COMBINATORIAL OPTIMIZATION

Combinatorial Engineering of Decomposable Systems

Mark Sh. Levin

Springer Science+Business Media, LLC

Combinatorial Engineering of Decomposable Systems

COMBINATORIAL OPTIMIZATION

VOLUME 2

Through monographs and contributed works the objective of the series is to publish state of the art expository research covering all topics in the field of combinatorial optimization. In addition, the series will include books which are suitable for graduate level courses in computer science, engineering, business, applied mathematics, and operations research.

Combinatorial (or discrete) optimization problems arise in various applications, including communications network design, VLSI design, machine vision, airline crew scheduling, corporate planning, computer-aided design and manufacturing, database query design, cellular telephone frequency assignment, constraint directed reasoning, and computational biology. The topics of the books will cover complexity analysis and algorithm design (parallel and serial), computational experiments and applications in science and engineering.

The titles published in this series are listed at the end of this volume.

Combinatorial Engineering of Decomposable Systems

by

Mark Sh. Levin

Moscow, Russia

SPRINGER SCIENCE+BUSINESS MEDIA, LLC

A C.I.P. Catalogue record for this book is available from the Library of Congress.

ISBN 978-1-4613-7673-6 ISBN 978-1-4615-5831-6 (eBook)
DOI 10.1007/978-1-4615-5831-6

Printed on acid-free paper

This book is dedicated to the
Memory of My Parents:
Shmuil M. Levin, and
Revekka M. Levina (Bahnova).

Contents

Preface

In recent years, combinatorial building new complex systems has been an active area of research in design and planning. This situation depends on the fact that a lot of contemporary systems are very complicated and consist of various components which may be selected on the basis of catalogues. In the book, we describe this situation as combinatorial engineering of decomposable systems. We consider the following issues:

(a) a hierarchical combinatorial description of decomposable systems;

(b) functional operations of combinatorial engineering (e.g., analysis, design, comparison, transformation);

(c) basic combinatorial elements (e.g., chains, trees) and their proximity;

(d) approaches to structural modeling;

(e) compatibility of system components;

(f) basic problems of combinatorial synthesis (multicriteria selection, multiple-choice knapsack problem, morphological analysis, clique, morphological clique, etc.).

Mainly we examine the following: hierarchical system models, system components, design alternatives (DA's) for system components and their interconnections (Ins), estimates of DA's and Ins, and changes (transformation) in the systems.

Our material is based on the following fundamentals: (a) system analysis, systems engineering, and hierarchical approaches to the design and analysis of complex systems; (b) technology of decision making and artificial intelligence; and (c) combinatorial modeling and optimization.

The book presents the author's engineering/scientific experience and knowledge (e.g., engineering practice, management engineering, design of effective algorithms and solving schemes) in the analysis and design of complex multidisciplinary systems of various kinds (e.g., software, information support, organizations, planning, quality analysis, house-building, machine building, electronics, etc.).

Our investigation is based on

hierarchical morphological multicriteria design (HMMD)

that implements construction of a decomposable system (integrated system, composite software, plan, machine, etc.) from interconnected components. The research on HMMD consists of the following main parts:

(1) description of HMMD;

(2) support tools as follows: (a) basic problems, and models (design of hierarchical system models; multicriteria ranking of DA's, synthesis of composite DA's, etc.); (b) information aid; (c) proximity for complex objects; (d) organizational aid;

(3) application of HMMD to some combinatorial problems; and

(4) case studies.

Our text describes components of HMMD including interconnection among subsystems, multicriteria ranking, coordination of scales, composition problem, kinds of composite decisions, information support, procedures for analysis and refinement, comparison of system versions, and some organizational issues.

HMMD is related with various disciplines (e.g., decision making, cooperative work, combinatorial optimization, concurrent design, multi-agent systems, etc.).

Two main directions of our morphological approach can be pointed out:

1. Interactive Part. Interactive description, analysis, design, and transformation of decomposable systems. In this case, we intend the following:

1.1. Orientation to multidisciplinary studies, and education processes (graduate and post-graduate students, continuous education), including joint project execution.

1.2. A problem dimension is about 30...100 nodes, and 2...5 hierarchical levels of the system model.

1.3. Interactive user-oriented modes of solving processes.

1.4. Simple software with very limited required computer resources.

2. Computation Part. Solution of very large combinatorial problems on the basis of morphological macroheuristics. Note that these macroheuristics may be considered as a hierarchical modification of dynamic programming. This direction requires the use of powerful computer environments (including multi-processor systems). In the book, we only describe possible applications of some morphological heuristics for combinatorial optimization problems (e.g., traveling salesman problem, scheduling, multi-routing problem, and location problem). Evidently, this direction has to be continued on the basis of special computer experiments for very large problems including realistic ones. It is reasonable to compare the use of different approaches for the same problems, (e.g., traveling salesman problem, scheduling, and location):

(i) our morphological heuristics,
(ii) Branch-And-Bound Method, and
(iii) genetic and evolutionary methods, etc.

Thus this part is a research in progress.

The hopefulness of the author is based on the possible use of HMMD for student participation at the various stages of a multi-disciplinary project-oriented education. In this case, student teams can include students of different departments. The use of HMMD may improve the student skills in various significant domains:

(1) basic disciplines;
(2) communication skills;
(3) complex project execution and management; and
(4) system thinking.

Secondly, the monograph presents a set of essays on some significant topics as follows: (a) hierarchical design; (b) combinatorial models of synthesis; (c) comparison of structured systems; (d) transformation of system; etc. Also, the manuscript contains a battery of various applied examples (over 40) that may be useful for specialists of many domains. A special chapter of the book is oriented to educational issues.

Note that the book concurrently involves a bibliography of references central to hierarchical design problems, combinatorial synthesis, morphological approach, comparison of structural systems, system transformation, and some specific applications. For readers who are interested but unfamiliar with the references in these areas, the bibliography facilitates and encourages their researches.

In the main, our book addresses man-machine analysis and synthesis of complex systems on the basis of easy information processing by human. So our interactive viewpoint consists in the following:

Enabling to understand, to analyze, and to manage information by human at all stages of solving processes as follows:

(i) acquisition of initial information;
(ii) analysis/evaluation and management of intermediate information; and

(iii) analysis of resultant decisions.

For the above-mentioned goals we try to use the following:
1. Ordinal scales for initial, intermediate, and resultant information;
2. A limited volume of information (small dimension, number of presented elements as concepts, criteria, levels of scales, decisions, etc.); and
3. Easy presentation of information.

The next goal of the book consists of the following. In our opinion, needs of multi-disciplinary specialists are increasing. The book may be considered as a support material for preparation of the mutlidisciplinary specialists in the field of complex systems.

The additional goal centers the development of new software for HMMD. Three versions of the DSS COMBI for multicriteria ranking were developed in 1987, 1989, and 1991, accordingly ([294], [297], [317]). The authors of the system are Dr. M.Sh. Levin (general design, programming of prototype method, management, modeling, basic case studies) and A.A. Michailov (design, programming, modeling, some case studies). Note a morphological technique environment is realized in the DSS COMBI, including morphological graph-menu of solving process on the basis of the algorithms/procedures and data (estimates, preference relations). This system was presented at the Intl. Conf. on Subjective Probability, Utility and Decision Making (SPUDM) in Moscow (1989), at the Intl. Conf. on Multiple Criteria Decision Making in Fairfax (1990), and at the Intl. Conf. on Human-Computer Interaction EWHCI'93 in Moscow (1993), etc. The DSS COMBI was applied in education.

A hierarchical hypertext system for multicriteria analysis (methodology of multicriteria analysis; models; multicriteria descriptions of various objects; software packages; and well-known indices) was developed by M.Sh. Levin in 1988 [292]. The system is a simple attempt to hierarchical representation of information for complicated problem domains. The system was presented at the Intl. Conf. on Multi-objective Programming in Yalta (1988); at the SPUDM in Moscow (1989), at Intl. Conf. on HCI in Moscow (1993), etc.

The first software prototype of HMMD (interactive shell, base of case studies, heuristic algorithms for selection and composition, a helper) was developed within the scope of the project which was supported by Israeli Ministry of Trade and Industry (Jan.-Sept., 1992). The following team has executed the project: Dr. M.Sh. Levin (the author of project, general design, modeling, algorithms, management, case studies), Eng. B. Belayavsky (programming), and Eng. B. Sokolovsky (some case studies).

Our material is presented in an ***engineering style*** that includes a standard schematic description of realistic problems, formulations of corresponding mathematical models, solving schemes (algorithms, procedures), and numerical examples on the basis of standard tables, figures, and diagrams. Real-life applications are a result of or involve the implementation of materials presented in the book.

Thus each reader can understand basic problems, approaches to solve them, and approaches to build other close problems and solving schemes. Mathematical fundamentals may be found in referred literature.

Acknowledgments

The list of acknowledgments is the following:

1. Discussions on engineering design (1991/1992, Israel): Prof. Z. Gezelev, Eng. A. Kleiman, Prof. Yu. Kolodny, Prof. E. Levin, Dr. I. Levin, Dr. Yu. Lesohin, Eng. I. Mishne, and Prof. I. Shreiber.

2. Discussions on discrete mathematics, combinatorial optimization and bibliography aid:

(a) Dr. B.V. Cherkasskii, Dr. S.O. Kuznezov, Prof. I. Melamed, Dr. A. Rubinov, and Prof. I.Kh. Sigal (1993/1997, Russia);

(b) Prof. B.G. Mirkin (1994/97, Russia/USA/Germany);

(c) Dr. M.I. Rubinstein (1994/95, Germany); and

(d) Dr. V.G. Timkovsky (1992/97, Canada).

3. Benevolent attention at various stages of the author's research: Mr. I. Amihud (Israel), Prof. H. Branover (Israel), Dr. A. Cypher (USA), Eng. E. Danilova (Russia), Prof. Y. Dinitz (Israel), Mr. D. Dolev (Israel), Dr. A. El-Boher (Israel), Prof. A. Feinberg (USA), Prof. B. Fischhoff (USA), Dr. E.M. Furems (Russia), Dr. Ju. Gornostaev (Russia), Prof. A. Kapralski (Japan), Dr. M. Klein (USA), Dr. M. Kress (Israel), Dr. E.R. Lieberman (USA), Prof. B.G. Litvak (Russia), Dr. B.A. Nardi (USA), Eng. R. Ortiz Bernal (Germany), Prof. A. Pawlak (France), Prof. M.A. Pollatschek (Israel), Prof. Yu.S. Popkov (Russia), Dr. B. Prasad (USA), Eng. C. Ruckert (Germany), Prof. A.L. Sadovskii (USA), Prof. B. Shackel (UK), Dr. M. Sobolewski (USA), Prof. A.L. Stempkovsky (Russia), Prof. R.E. Steuer (USA), Prof. C. Unger (Germany), Prof. D.B. Yudin (Russia), and Prof. S. Zionts (USA).

4. Possibility for author's presentations and discussions at some universities as follows:

(a) Technical University of Berlin, Germany (Prof. W. Beitz);

(b) Ben-Gurion University of the Negev, Israel (Prof. V. Goldstein);

(c) The University of Aizu, Japan (Prof. S. Okawa, Prof. C. Nehaniv, and Prof. Q.-P. Gu);

(d) University of Konstanz, Germany (Prof. R. Vetschera).

5. The invitation of the President of The University of Aizu (Fukushima, Japan) Prof. T.L. Kunii to occupy a visiting research position at the above-mentioned university, financial and organizational aid of Fukushima Prefecture (Japan) and The University of Aizu, including excellent environment and well-disposed Professors of the University (1995/1996).

6. The invitation of Prof. I. Rival for a brief research visit to University of Ottawa (Canada). Personal contacts with Prof. I. Rival and members of his Algorithms Research Laboratory were very useful and important for the author.

7. The crucial English editing this book of Prof. Panos M. Pardalos (University of Florida, USA).

8. The efforts of the publishers (Kluwer Academic Publishers, Science & Technology Division; Boston, USA): Mr. John R. Martindale, Ms. Lorena Loubsky, Ms. Arlene Apone.

9. The aid and consulting in the use of LaTeX: (a) V.L. Bronshtein (Inst. for Forecasting of Russian Academy of Sciences); (b) V.V. Voloshinov (Inst. for System Analysis of Russian Academy of Sciences); and (c) Domingo Marin Ricoy and V. Romanov (Editorial URSS, Moscow, Russia).

10. The information environment of the Russia National Public Library for Science and Technology (*GPNTB*, Moscow) where the author has prepared basic surveys and bibliography materials (1992/1997).

1 DECOMPOSABLE SYSTEMS AND DESIGN

1.1 COMBINATORIAL APPROACH TO DECOMPOSABLE SYSTEMS

In recent years, community needs for design science have increased. There exist the following basic reasons:

1. Complexity of designed systems increases,

2. Designed systems consist of components and interconnection among them of various and different kinds,

3. The time of life-cycle of product (technology, etc.) is decreasing, and

4. Designed systems are very specific and unique.

As a result, we obtain a situation when a design team, including different specialists, have to design a new complex system in a very limited time interval, and soon this situation will be an usual one.

A lot of design directions exists. Some of these directions are discipline dependent, others are discipline-independent ones. However only two major approaches to system design are well-known [519]:

(i) improvement of an existent system (evolution); and

(ii) designing a new system.

As a rule, the first approach is used. Usually it consists in evolutionary improvement and multi-criteria selection of design alternatives ([22], [66], [115], [127], [235], [496], etc.).

In this book, the second approach, based on standard multicriteria decision making problems and the hierarchical solving scheme with weighted interconnection among subsystems, is studied.

We point out some basic combinatorial operations for decomposable systems:

(a) combinatorial description and presentation;

(b) analysis and evaluation;

(c) synthesis of composite DA's;

(d) revealing of bottlenecks by elements;

(e) comparison of system versions and/or their parts; and

(f) modification (e.g., improvement, adaptation) of system versions.

Partly the above-mentioned operations correspond to basic operations of traditional logic [533]. Note that composite alternatives have been analyzed by R. Vetschera as a selected subset of an initial alternative set in group decision making [521]. In our case, a composite alternative is a result of selecting and integrating the alternatives into a composite one. Our basic approach is represented as hierarchical multicriteria morphological analysis and design (HMMD). HMMD involves two major design phases:

(i) Top-Down development of hierarchical model for designed system; and

(ii) Bottom-Up scheme for the following: selection of the best design alternatives (alternatives, design choices, candidate designs) for system components and composing of the design alternatives into composite ones.

This approach outlines a common structured language of preliminary design stages for different specialists (designers, engineers, managers, economists, etc.). The following three types of a design process are the basic ones: creative design, innovative design, and routine design ([94], etc.). Tomiyama and Ten Hagen consider three similar types of design as follows: new design (creation of structure); combinatorial design (combination of parts), and parametric design [502]. HMMD is oriented to combinatorial design, and, partially, other ones. In addition, HMMD may provide conceptual unity, harmony of a designed system which are usually required in complex projects [64], and support not only to obtain a design decision but to develop a systematic knowledge base and a flexible environment for the design, analysis, improvement, and learning.

Fundamental system ideas involve several key components of *system* concepts ([82], [519], etc.):

(1) elements (parts, components, subsystems);

(2) hierarchy (levels, etc.);

(3) connections (interactions, compatibility);

(4) whole (system properties);

(5) role, function, objectives of a system (or requirements to the system and its parts); and

(6) environment.

A system is defined as an aggregation or an assembly of components united by some forms of regulated interaction to form an integral whole. In our opinion, logical step of systems engineering 'design synthesis' is a key one [440]. This step is implemented as combining (composing) the components in such a way as to construct a structure not clearly there before ([387], [432], [519], etc.). The important role of the composition for problem solving was pointed out by Newell and Simon ([380]. Main approaches to the design synthesis, parametric design are the following:

1. Evolutionary transformation of an existing system ([66], [95], [154], [235], [441], etc.).

2. Morphological analysis ([22], [119], [127], [235], [554], etc.).

3. Mathematical programming ([165], [166], [186], [545], etc.).

4. Multicriteria optimization and decision making ([269], [367], [448], [449], [482], [486], [488], [496], etc.).

5. Knowledge based systems ([26], [47], [94], [167], [168], [169], [337], [338], [349], [437], [496], [502], [510], etc.).

6. Constraint Satisfaction Problems approach ([109], [110], [328], [334], [335], [359], etc.).

7. Heuristic approaches (genetic algorithms, stochastic search techniques, simulated annealing algorithms, evolutionary computation, etc.) ([175], [275], [354], [391], etc.).

8. Special creative techniques including expert judgment (inventory machines, etc.) ([14], [119], [139], [170], [235], [378], [432], [472], [483], [525], [533], etc.).

Note Shenhar and Dvir have proposed three-level classification for complex products, that involves the following [461]:

(i) ***assembly*** as a collection of components;

(ii) ***system*** as a complex collection of many units and assemblies; and

(iii) ***array*** as a large collection or network of systems.

Here we will use hierarchical morphological approach to decomposable systems. The use of the hierarchic view had a long time ([471], [473], etc.). Hierarchy theory consists of a set of claims ([134], [471], [473], etc.). Demster et al. point out the following two fundamental reasons for using a hierarchical approach [114]:

(a) reducing complexity; and

(b) coping with uncertainty.

The significant idea of ***nearly decomposable systems*** is based on two observations ([471], [473]):

1. Complexity takes a form of a hierarchy, whereby a complex system is composed of interrelated subsystems that have their own subsystems, and so on.

2. Generally, interaction inside subsystems are stronger and/or more frequent than interactions among subsystems.

Here we consider an organic hierarchy for the representation of a functional/organ structure of designed systems ([86], [217], [337], etc.). An example of organic hierarchy is depicted in Fig. 1.1, where horizontal links correspond to interconnectibility between system parts [86].

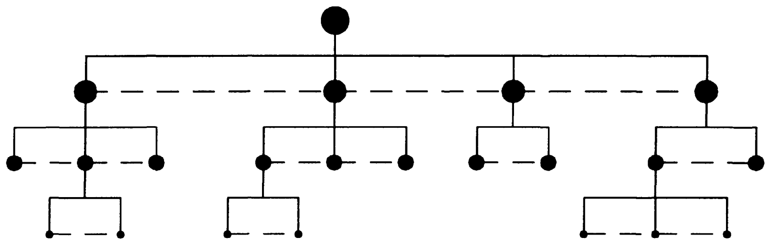

Fig. 1.1. Example of organic hierarchy (some horizontal links are suppressed)

We use "decoupled" decomposition (partition) approach that divides the original system into a set of more simple subsystems. This solving scheme realizes the well-known principle "divide-and-conquer".

The problem of decomposition is perhaps the most important approach towards complex situations [380]. Mainly, there exist three kinds of decomposition as follows:

(a) by physical components (objects);

(b) by functions, by knowledge domains, etc. (aspects); and

(c) by time stages/periods (series parts of a directed flow of elements, information).

The first two approaches are basic in the analysis of complex systems (e.g., engineering systems, complex software), and the third approach is fundamental for the analysis of processes (e.g., manufacturing systems, data flows, dynamical programming). Well-known object-oriented development is based on decomposition approach too [56]. Hubka and Eder have indicated some useful 'Propositions' for technical systems including the following [218]:

1. All types of system properties are achieved by mean of the elementary design properties;

2. Every system can be decomposed into partial systems which in their prescribed combination fulfill the partial and elementary functions of the system; and

3. The behavior of a system depends not only on the sum of behavior of the elements, but also on the coupling relationships between these elements.

Usually the hierarchic decomposition approach consists in the following stages: development of a hierarchic structure of the designed system or initial problem on the basis of hierarchical levels and decomposition (partitioning); solving local subproblems; and composition of global decision from local decisions ([189], [208], [338], [380], etc.).

Now let us consider an example of decomposable system (Fig. 1.2). Here we consider the following situation: initial system S consists of three components A, B, C with corresponding DA's: A_1, A_2, B_1, B_2, B_3, C_1, C_2.

At the design stage we can compose the following system versions, for example: $S_1 = A_1 * B_2 * C_1$, $S_2 = A_2 * B_1 * C_1$, $S_3 = A_2 * B_2 * C_2$.

In addition, it is possible to examine system changes as follows:

(1) to add DA's (A_3, B_4, C_3);

(2) to delete DA's (B_2, C_1); and

(3) to add a component D with DA's (D_1, D_2, D_3, D_4).

In this case, the following composite DA's may be under consideration, for example: $S_1^{'} = A_1 * B_4 * C_2 * D_1$, $S_2^{'} = A_3 * B_1 * C_2 * D_2$, and $S_3^{'} = A_2 * B_1 * C_3 * D_3$.

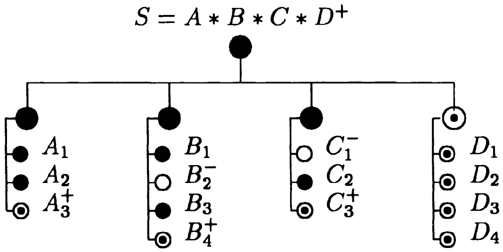

Fig. 1.2. Decomposable system

Usually design processes can be considered as a human-computer system (HCS) involving the following major components: goal part (requirement, objectives, etc.), information part (design information), operational part (design techniques, tools), and organizational part (design team, organizational structure) ([120], [214], [218], [294], [324], [378], [434], [500], etc.). Fig. 1.3 depicts basic hierarchies of a design process flow.

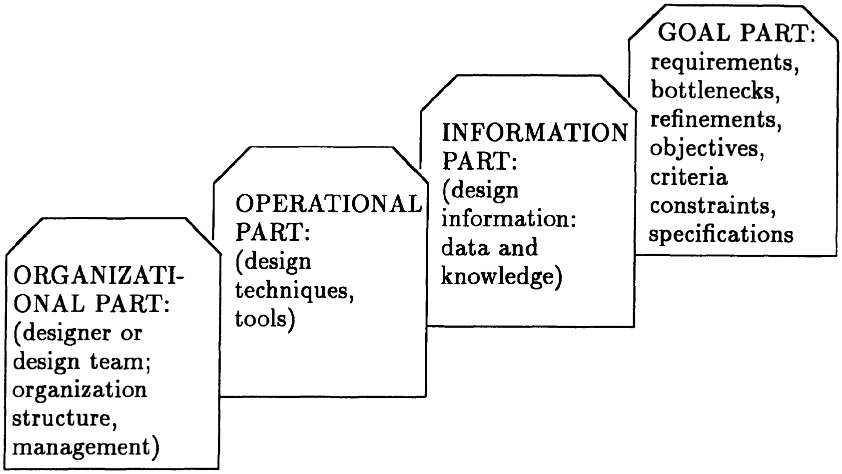

Fig. 1.3. Interconnected hierarchies of design process flow

Table 1.1 displays relationship of these components and creative levels corresponding to the components [14]:

(i) usage of initial element,

(ii) usage of selected element; and

(iii) usage of designed element.

Usually the above-mentioned components have high complexity and it is reasonable to use hierarchical representation to reduce this complexity. Hierarchical structures provide simple execution of the following operations: development, representation, processing (analysis, evaluation, correction, etc.). Hierarchies have the following features [294]:

(1) clarity and displayability;

(2) facility of analysis and study;

(3) ease of processing and correction; and

(4) decomposability, including facilities for concurrent execution of operations, distributed concurrent processing, etc.

HMMD provides an environment of distributed cooperative work. HMMD includes specification and analysis of several connected hierarchies: system components and corresponding alternative designs or design alternatives (DA's); criteria (Cr) and estimates of DA's; weighted interconnection (Ins) among DA's; etc. HMMD is based on several interconnected hierarchies (Fig. 1.4), where a hierarchical model of design system is a basic (leading) one.

Table 1.1. Components of Design Process

Part of design process flow	Creativity level of components by Altshuler		
	Initial	Selected	Designed
GOAL PART	Initial requirements, bottlenecks, refinements, objectives, criteria, constraints, specifications	Selected requirements, bottlenecks, refinements, objectives, criteria, constraints, specifications	Created @ requirements, bottlenecks, refinements, objectives, criteria, constraints, specifications
INFORMATION PART	Initial data and knowledge (from data and knowledge bases)	Selected @ data and knowledge	Combinations of data and knowledge, transformed data and knowledge
OPERATIONAL PART	Initial universal tools	Selected @ tools oriented to task and user	Designed tools
ORGANIZATIONAL PART	User, team; initial organization (responsibility allocation, planning, etc.)	Selected @ user/team (diagnostics & selection,) selected organization approaches	Trained user(s), established team, designed organization approaches

@ - example of system

Listed elements may be considered as components of declarative languages for design. Our proposed approach is similar to a hierarchical multi-blackboard architecture ([532], etc.).

The generalized scheme of HMMD involves the hierarchical description and design of system, analysis (quality assessment, revealing bottlenecks), and improvement. Design phases of HMMD are described in ([297], [300], etc.).

Our work depends on the direction of problem solving and decision making which was pointed out by Simon et al. [474]. HMMD may be considered as an useful element for various multi-disciplinary research directions including the following: computer supported cooperative work, concurrent engineering, cooperative distributed problem solving, group decision making, distributed decision making, distributed AI systems, and multi-agent systems ([113], [128], [147], [148], [238], [247], [251], [402], [412], [485]).

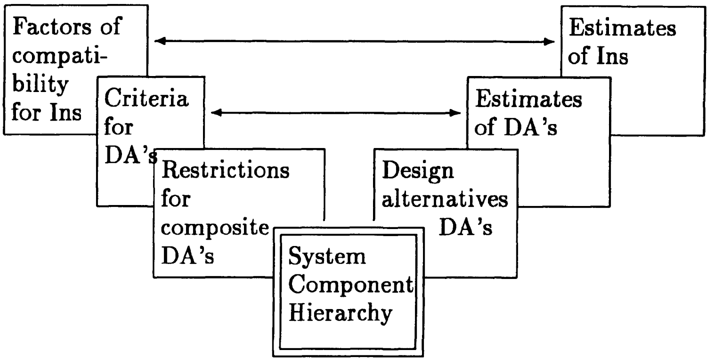

Fig. 1.4. Interconnected hierarchies of HMMD

1.2 COMBINATORIAL MODELING

Combinatorial optimization problems have been used in many applications as follows:

(1) engineering design in mechanics, electronics, architecture, software engineering, etc. ([41], [186], [202], [505], etc.);

(2) planning and scheduling in computer systems, in manufacturing and in other applications ([51], [85], etc.);

(3) VLSI and IC design: ([1], [38], [69], [189], [514], [538], etc.);

(4) network design and management ([197], [205], [450], [499], etc.); and

(5) information design ([59], [77], [78], [79], [159], [299], [304], [312], [515], etc.).

The list of basic combinatorial problems involves the following: partitioning, clustering, scheduling, knapsack problem, salesman problem, packing, routing, assignment, location, covering, placement, floorplanning, etc. ([7], [160], [197], [379], [425], etc.).

Roberts has pointed out that combinatorics is concerned with the study of arrangement, patterns, designs, assignment, schedules, connections, and configurations [425]. Here we present a systematic view to some problems of the combinatorial synthesis (composition problems), and accumulate some corresponding combinatorial models. We examine deterministic combinatorial problems to compose a decomposable system from components, for each of them there exists a set of design alternatives (DA's). In addition, we take into account pairwise interconnection (Ins) or compatibility between DA's. Note that our problem differs from traditional problems of combinatorial design theory ([117], etc.).

Program and data modules, elements of plans, models from design libraries, data paths, tests or verification tools, and their combinations may be considered as above-mentioned DA's. Our problems maybe considered as a composite one. Here we do not study synthesis problems in chemistry (the synthesis of products, etc.).

Now let us consider a set of engineering functional operations on the basis of above-mentioned Altshuller's levels of creative problems are as follows: selection of objects, modification of an object, design of a new object, and synthesis of a new system from a set of initial objects) [14]. Thus we may examine the following two-dimensional Cartesian space:

(i) kind of a system (the whole object or system, the decomposable system consisting of a set of simple objects);

(ii) functional engineering operations: description and/or presentation; analysis as evaluation, assessment; analysis as revealing bottlenecks; comparison; selection; synthesis; modification (correction, improvement, adaptation, reconstruction, re-engineering).

As a results, we obtain the following basic objects of our examination (e.g., system descriptions) and problems for the whole system (a), and decomposable system (b):

1. *Description and/or presentation:*

(1a) functional description, multidimensional representation;

(1b) tree-like system model, external requirements: criteria, constraints for the system and its elements, design alternatives (DA's) for the elements (nodes of the system model), interconnection (Ins) among DA's, estimates of DA's and Ins.

2. *Analysis and evaluation:*

(2a) assessment in multiparameter space;

(2b) multilevel assessment of the system and its elements, including assessment of composite DA's in a complex space of system excellence.

3. *Analysis as revealing of bottlenecks:*

(3a) revealing of critical parameters;

(3b) revealing of bottlenecks (by system parts, by Ins, by system structure).

4. *Comparison:*

(4a) multiparameter comparison of the objects;

(4b) comparison of system versions (by components and DA's, by Ins, by structure).

5. *Selection:*

(5a) multicriteria selection;

(5b) multilevel system's selection.

6. *Synthesis:*

(6a) optimal design;

(6b) two problems: (i) selection of the best system version; (ii) hierarchical synthesis (design of system model, specification of requirements, generation of DA's, assessment of DA's and Ins, composing of composite DA's).

7. *Transformation (e.g., modification, improvement, adaptation, change, approximation):*

(7a) parameter optimization;

(7b) generation of improvement actions (improvement of DA's and/or Ins, modification of the system model), and scheduling.

Approximation of a combinatorial object corresponds to well-known combinatorial problems, e.g., spanning tree problem, Steiner tree problem, constructing a minimum-weight two-connected spanning networks, covering problem, etc. ([160], [174],[347], [372], [379], [425], etc.).

Recently the problems above have been considered in important applications (e.g., network design on the basis of tree-like approximation). On the other hand, new practical approximation problems have been appeared, for example:

(i) approximation of information structures with the use of hierarchies, clusters, cliques ([59], [79], [292], [312], etc.);

(ii) processing of preference relations in decision making on the basis of ***step-by-step*** approximation problems ([35], [290], [317]).

Main two types of transformation may be used at different engineering levels as follows: (a) the local transformation; (b) the global modification, adaptation, improvement or reengineering of the system.

In our opinion, modification is now the most important kind of engineering activity (e.g., redesign, reengineering). Also, it is reasonable to point out basic changes of decomposable systems:

1. *Internal changes:*

(a) local evolution (DA's and/or Ins);

(b) global evolution (subsystems); and

(c) global modification (system model).

2. *External changes:* requirements.

Composing or synthesizing a problem is an important part of design processes in various engineering disciplines, and many design methodologies include this stage of the design framework, e.g, as follows:

morphological analysis ([22], [235], [554], etc.);

diakoptics [266];

design methodologies of Carnegie Mellon University ([113], [118], [189]);

JESSI COMMON FRAMEWORK for design process [324];
Hierarchical Decision Making [202];
concurrent engineering ([412], etc.);
modular design and manufacturing ([41], [505], etc.); and
Computer-Aided Cooperative Product Development ([485], etc.).

In addition, it is reasonable to point out close design approaches of Computing Center of the Russian Academy of Sciences:
(a) decomposition design of complex systems ([265], etc.), and
(b) approximation-combinatorial method [245].

Mathematical study "knowledge synthesis" consists in an investigation of composition problems also ([511], [512],[545], etc.). Many actions of systems integration may be examined as the composition problem too. For example, recent direction Information/Decision Fusion is oriented to aggregation/integration of various kinds of data and decisions and, as a result, system composition in several domains ([410], etc.). In our opinion, the number of applications that use composition problem is increasing.

In our case, the generalized list of composition problem stages is the following:
(1) specification of requirements (objectives, criteria, constraints);
(2) designing the structure of the target system;
(3) generation of DA's;
(4) evaluation of DA's, and Ins between DA's;
(5) selection of DA's;
(6) composing; and
(7) analysis of composite DA's, and improvement.

Usually the following basic approaches to problems above are used: (a) optimization models ([41], [186], [496], etc.); (b) knowledge base systems ([189], [349], [263], [496], etc.); and (c) hybrid approaches ([271], etc.).

1.3 COMBINATORIAL MODELS OF SYNTHESIS

Mainly, we examine two optimization problems as follows: selection, and composing (when a structure of the target system is specified). We do not consider space constraints between DA's. Table 1.2 outlines considered combinatorial problems. Our material will focus on the descriptions of the problems above, and solving schemes.

1.4 SOME DESIGN PRINCIPLES

Fundamentals of design processes are well-known, e.g., design science, systems engineering, system analysis, etc. However, contemporary design schemes have to involve some detailed principles, taking into account some additional properties.

Let us consider the following basic design principles for decomposable systems:

Principle 1. Decomposition of complex object, i.e., ***divide-and-conquer***.

Principle 2. Hierarchy as a main tool opposite to complexity of presentation, representation, analysis, execution of team activity, etc. Data and knowledge presentation, structure processing, etc.

Principle 3. Interconnected hierarchies as the main representation that implements both decomposition and hierarchical increasing the complexity.

Principle 4. Revealing and use of a leading (basic) hierarchy. As an example of a basic hierarchy we may point out a hierarchy of designed product or process, hierarchy of design/manufacturing process and organizational hierarchy of domain specialists team.

Principle 5. Multiple view description (multiple hierarchies, multiple objectives, multiple criteria).

Principle 6. Flexible scheme of data processing based on standard ***hard*** tools for data processing (e.g., selection and composition).

(a) standard (***hard***) type (e.g., 'cascade', tree-like, etc.) of scheme;

(b) flexible implementation of scheme (design framework, flow).

Principle 7. Limited dimension of design problem elements (e.g., human, computer procedure, tasks, objectives, operations, data/knowledge).

Evidently that these principles are projections of system analysis principles into design processes.

Table 1.2. Basic selection/composition problems

Problem	Description			
	Composite DA's (objectives)	DA's (kinds of estimates)	Ins (kinds of estimates)	Constraints
1.Ranking (multicriteria (selection) ([129], etc.)	Priority	Multicriteria		
2.Knapsack (selection) [347]	Additive function	Quantitative		Linear resource constraint
3.Multiple-choice knapsack problem (composing) [347]	Additive function	Quantitative		Linear resource constraint
4.Quadratic integer programming (composing [71]	Additive function	Quantitative	Quantitative	Linear resource constraint
5.Nonlinear integer programming (composing with redundancy) [41]	Nonlinear function	Quantitative		Resource constraint
6.Mixed integer nonlinear programming (composing) [186]	Nonlinear function	Quantitative		Resource constraint
7.Morphological analysis (composing) ([22], [554])			Binary	
8.Multicriteria morphological analysis (composing) [127]		Multicriteria	Binary	
9.Morphological clique (composing) ([296], etc.)	Lattice of excellence	Multicriteria /ordinal	Ordinal	
10.Morphological clique (composing with redundancy) [298]	Lattice of excellence	Multicriteria /ordinal	Ordinal	

1.5 HIERARCHICAL MORPHOLOGICAL DESIGN

Hierarchical Morphological Multicriteria Design (HMMD) was proposed for the description, design, analysis, and improvement of decomposable systems ([296], [300], etc.). The following applications of HMMD have been published:
(1) design of user interfaces [297];
(2) planning a solution strategy [298];
(3) information design in hypertext systems ([299], [312]);
(4) planning of information centers [301];
(5) transformation of information systems [310];
(6) planning of student career [313];
(7) design of vibration conveyors [314]; etc.

In this section, we describe basic assumptions, scheme, and interconnection of system components.

1.5.1 Assumptions

HMMD is described in ([296], [300]). Note, we examine a system consisting of components and a set (morphological class) of DA's that are generated for each component. Basic assumptions are the following:

1. Decomposability of a system, i.e. system structure, is central or tree-like one. It means that an entity of a designed system is represented by hierarchical structure.

2. The system effectiveness (synergistic features, excellence) may be represented as an aggregation of two parts:
(i) subsystems effectiveness; and
(ii) effectiveness of interconnectivity (interconnection, compatibility) among subsystems.

The latter may be considered as an additional system element (Fig. 1.5).

These two assumptions correspond to the above-mentioned propositions by Hubka and Eder [218].

3. System effectiveness may be described through the following three types of monotonic criteria ([39], etc.): (a) additive type (e.g., cost); (b) multiplicative type (e.g., reliability); and (c) supreme one, i.e., maximum or minimum (e.g., performance).

Let system (decision) S consists of m elements (subsystems and/ or interconnections)

$$\{S(j), j = 1, ..., m\} : \ S = S(1) * ... * S(m).$$

Denote system effectiveness (e.g., for $m = 3$, i.e, for subdecisions a, b, c) as follows:

$$F(S) = F(S_a * S_b * S_c) = g(F_a(S_a), F_b(S_b), F_c(S_c)),$$

where g is an aggregation function, S_a, S_b, S_c are subsystems, $F_a(S_a), F_b(S_b), F_c(S_c)$ are system effectiveness for subsystems. We assume that criteria satisfy to the following conditions:

$$g(F_a, F_b)) = g(F_b, F_a), \qquad (1.1)$$

$$g(F_a, g(F_b, F_c)) = g(g(F_a, F_b), F_c)), \qquad (1.2)$$

$$if \;\; F_b > F_c \;\; \forall F_a \;\; then \;\; g(F_a, F_b) > g(F_a, F_c). \qquad (1.3)$$

Obviously, if (1)-(3) are correct then the best system (design decision) consists of the best components ([39], etc.). Note that the distribution of requirements (i.e., criteria, preferences) and constraints to the subordinates (i.e., for a lower level of the system hierarchy) is an important problem for hierarchical description of systems. An approach in a multiagent framework for the problem is proposed in [102].

4. Decomposability of interconnection (compatibility), i.e., effectiveness of interconnection among subsystems, is equal to an aggregation of all pair interconnections between subsystems. The assumption is based on the independence of pairwise interconnection.

This assumption is a strong one, but it is in accordance with the above-mentioned propositions by Hubka and Eder [218].

Here we use positive ordinal scale of pairwise intercompatibility assessment (e.g., $0...l$; l is the best one, 0 corresponds to impossible interconnection) and *minimum* aggregation function for interconnection among subsystems. Afterhere we will use the symbol Δ to point out incompatibility of DA's too.

5. For all alternatives DA's of each morphological class it is possible to transform multi-criteria description (without possible external interconnection) into effectiveness that is evaluated upon the ordinal scale $1...k$, (1 corresponds to the best group, 2 corresponds to the second group, etc.).

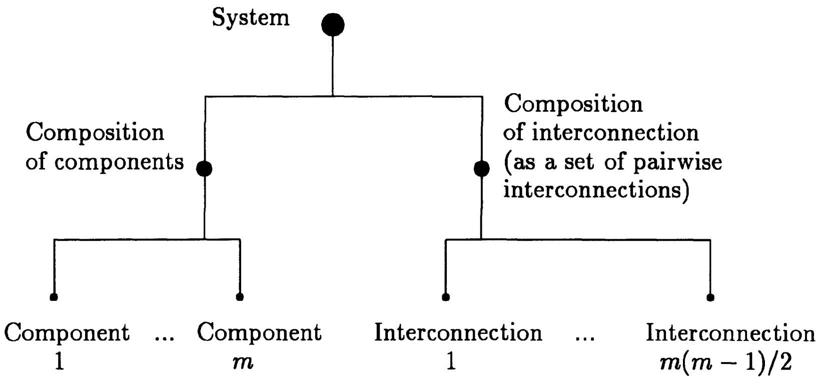

Fig. 1.5. Hierarchy of designed system

1.5.2 Scheme

Generalized HMMD consists of the following main phases:

Phase 1. Description of complex decomposable system.
Phase 2. Design.
Phase 3. Analysis of composite design alternatives and revealing bottlenecks (including measurement and evaluation of system perfection).
Phase 4. Improvement: generating the improving redesign activities and planning the improvement (redesign).
Phase 5. Comparison analysis of systems versions.

Table 1.3 depicts the design scheme of HMMD. Design processes of HMMD are based on two basic multicriteria problems: selection/ranking of DA's (Ins), and composing of DA's.

Note similar hierarchical design schemes have been applied in the following approaches:

(1) GARDEN framework [524];
(2) design methodology of The Engineering Design Research Center of Carnegie Mellon University 'EDRC' [113];
(3) Decision Support Problem Technique [367];
(4) multilevel selection development ([448], [449]);
(5) Analytic Hierarchy Process [438];

(6) Structured Design [543];
(7) Object Oriented Development [56];
(8) Hierarchical Decision Making [202];
(9) generation of management alternatives on the basis of morphological tables [213];
(10) JESSI COMMON FRAMEWORK for design process [324]; and
(11) concurrent engineering ([247], [272], [402]; [412], etc.).

Dixon has examined the following three basic series problems of systems design [119]:
(i) inventiveness or creative synthesis (generation of alternative design decisions);
(ii) engineering analysis (evaluation of a decision, i.e., computation of estimates); and
(iii) decision making (selection of the best decisions).

Dixon has pointed out that these problems often are applied and combined at each stage of design processes. Our HMMD is oriented to integration of problems above.

1.5.3 Interconnection of Components

Complex systems include different components and contain many types of interrelations among components. Interconnection of system components is a well-known description for large-scale systems design, e.g., software systems, multiple processor systems, network systems, organization structures, and technical systems ([154], [189], [268], [218], [223], [373], [453], etc.). Hubka and Eder point out couplings between technical system elements of different kinds e.g., as follows [218]:
(1) mechanical (scleronomic or rheonomic, holonomic or non-holonomic, unilateral or bilateral, force or force-free, etc.);
(2) thermal (heat transmitting or insulating);
(3) electrical (conductive connections or isolations);
(4) chemical (aggressive or neutral, acidic or basic, oxidizing or reducing); and
(5) magnetic (field-exciting or magnetically screening), time or space coupling.

Griffin and Hauser consider communication among product life cycle stages (marketing, engineering, manufacturing) and show that greater communication provides an enhancement of product development [184].

Table 1.4 contains some examples of compatibility for several systems of different kinds.

Interconnectivity analysis techniques derive multiple views of the structure of large-scale systems [453]. Ishii et al. propose 'Design Compatibility Analysis' (DCA) that employs AI techniques to evaluate the following compatibility: (i) proposed design with the required specification; and (ii) proposed design elements within the system ([223], [224]). DCA involves the use of the design experience on good or bad combinations of design alternatives that is represented as templates of design. Morse and Hendrickson indicate compatibility conditions among system components as follows: (i) relationship among interdependent data items (simple code restrictions; functional relationships among decision components); (ii) value range [373]. Gupta et al. consider a time-dependent interaction between design problems when a subproblem of a part design is possible only after the design of another subproblem of the part design [189].

Generally, interconnectibility may be specified as compatibility strength or/and structure requirements into composite decisions, e.g., precedence constraints. The latter are used for VLSI or circuits design ([189], [373], etc.). Various scales may be used for evaluation of compatibility strength, including the following:

(a) scalar value (binary, positive ordinal, positive quantitative, positive and negative ordinal; positive and negative quantitative;

(b) vector (binary, positive ordinal, positive quantitative, positive and negative ordinal; positive and negative quantitative); and

(c) taking into account the uncertainty nature of interconnection: fuzzy scale, probability scale.

It is reasonable to list the following well-known approaches to describe and analyze relationships among system components:

(1) ***sign graphs*** ([201], [423], etc.); and

(2) ***causal assertions*** (***positive***; ***negative***; ***neutral***; ***neutral*** or ***negative***; ***neutral*** or ***positive***; ***nonneutral***; ***positive***, ***neutral*** or ***negative***, ***universal***) in cognitive maps or causal graphs (e.g., for multi-agent systems) ([73], [262], [552], etc.).

In this book, nonnegative ordinal scales for compatibility strength are used.

Note that in recent years some authors have studied an emerging research area *coordination theory* ([75], [344], etc.). Also, it is reasonable to point out, that often it is necessary to take into account system components compatibility for a wide range of conditions/modes [154].

Table 1.3. Design scheme of HHMD

Phase	Step	Support procedure
Phase 1. Top-Down design and description of hierarchical system model	1.1 Design of tree-like model	Structural modelling
	1.2 Description of model:	
	1.2.1 Design of criteria hierarchy	Identification of objectives Expert judgement
	1.2.2 Specification of constraints for composition decisions	Identification of objectives Expert judgement
Phase 2. Bottom-Up hierarchical scheme of selection and composition of DA's	2.1 Initial step for leaf vertices - generation of DA's	Generation
	2.2 (Iterative) Major body of the schema	
	2.2.1 Evaluation of DA's on criteria	Expert judgement Modelling Computation
	2.2.2 Multicriteria comparison of DA's for selection (ranking)	Multicriteria selection (ranking)
	2.2.3 Specification of interconnection among DA's of subsystems (morphological classes)	Expert judgement Modelling Multicriteria ranking
	2.2.4 Coordination of ordinal scales for DA's and compatibility	Expert judgement Modelling
	2.2.5 Composition of the best DA's for higher hierarchical level	Composition problem
Phase 3. Analysis of composite DA's	3.1 Measurement of system perfection for DA's	Computation (e.g., N)
	3.2 Analysis of composite DA's, revealing bottlenecks	Analysis of morphological schemes
Phase 4. Improvement of composite DA's	4.1 Generation of improvement action set (*S-aggravating* elements and neighbors of DA's)	Generation of *S-aggravating* elements and neighbors of DA's
	4.2 Planning the improvement (forming the combination of improvement actions and their scheduling)	Selection of improvement combination Scheduling of actions
Phase 5. Analysis of system change	5.1 Comparison of system versions (structure, components, DA's, etc.)	Comparison analysis

Table 1.4. Some systems and types of compatibility

Application domain	Components	Types of compatibility
1.Technological process	Machine-tools or other equipment, personnel, etc.	Productivity, precision, reliability, maintenance, human qualification
2.Complex software	Program blocks, software packages, etc.	Productivity, data format, interface, common elements, experience of personnel
3.Biology system	Animals (beasts of prey, herbivorous animals, etc.) and plants	Usefulness of joint life
4.Material (e.g., metal)	Major components, auxiliary components	Compatibility (chemical, etc.)
5.Medical treatment	Patient, drugs, physiotherapy procedures, X-ray procedures, equipment	Compatibility of drugs & procedures, procedures & patient, procedures & equipment, etc.
6.Team	Manager, members of team, auxiliary persons	Psychological compatibility, etc.

1.6 AUXILIARY PROBLEMS

1.6.1 Design of Structural Models

In recent years a significance of designing the structural hierarchical models for complex systems has increased. Let us list the approaches which may be used for designing a hierarchical model:

(1) designing of hierarchical multilevel models for complex systems ([12], [353], [525], [536], [543], etc.);

(2) structural modeling ([164], [201], [285], etc.);

(3) selection, modification, and aggregation of standard frames, e.g., technological frame, that includes goals, elements under processing (data, raw), human resources, technological tools (methodology, techniques), results;

(4) design of configurations on the basis of expert systems ([349], [496], etc.);

(5) network languages for complex systems on the basis of linguistic geometry ([491], [492]);

(6) analysis of system elements and their clustering ([232], [325], [355], [368], [389], [402],[520], [541], etc.); and

(7) hierarchical approximation of a basic system structure ([59], etc.).

Let us point out some basic approaches and applications for approximation problems. First experience is accumulated in the following domains:

1. Approximation of networks for topological design, facility location, etc. (communication networks, transportation networks, etc.). Here there are several well-known basic problems as follows:

(i) spanning tree problem ([7], [156], [160], [423], etc.);

(ii) Steiner tree problem ([160], [535], etc.); and

(iii) construction of a minimum-weight two connected (or k-connected) spanning networks ([46], [372], etc.).

2. Approximation of preference relation in decision making or *set-to-set* transformation including series-parallel transformation schemes ([35], [290], [317], [342], [398], etc.).

3. Design of approximation tree-like structures for information, for example as follows:

(i) relational schema design in databases ([105], [230], [515], etc.); and

(ii) design of hierarchical structures (e.g., trees, pyramids) in hypermedia systems ([59], [292], etc.);

Secondly we consider two basic models of structural approximation ([35], [292], [365], etc.). Let G be an initial graph, and G^{ap} be a corresponding resultant approximating graph that has to be found. Then the problems are:

$$min \quad \rho(G - G^{ap}) \tag{1.4}$$

where ρ is a proximity of graphs G and G^{ap}.

$$max \quad q(G^{ap}), \qquad s.t. \quad \rho(G - G^{ap}) \leq \epsilon \tag{1.5}$$

where ϵ is a limitation for admissible difference between graphs G and G^{ap}.

Thirdly it is reasonable to investigate close problems for aggregation or integration of a set of initial graphs (or binary relations). Some material towards this problem will be presented in chapter 4.

1.6.2 Multicriteria Ranking

Let $A = \{1, ..., i, ..., p\}$ be a set of items which are evaluated upon criteria $K = 1, ..., j, ..., d$, and $z_{i,j}$ is an estimate (quantitative, ordinal) of item i on criterion j. The matrix $\{z_{i,j}\}$ may be mapped into a partial order on A. Often instead of $z_{i,j}$ a preference relation on A (e.g., on the basis of pair comparison) is applied as initial information. In the main, the following resultant kind of the poset (partially ordered set) as a partition with linear ordered subsets (a layered structure) is searched for (Fig. 1.6):

$$A = \bigcup_{k=1}^{m} A(k), \quad |A(k_1) \& A(k_2)| = 0 \quad if \ k_1 \neq k_2,$$

$$i_1 \succeq i_2 \quad \forall i_1 \in A(k_1), \ \forall i_2 \in A(k_2), \ k_1 \leq k_2.$$

Set $A(k)$ is called layer k. Thus each item $i \in A$ has a priority r_i, that equals the number of the corresponding layer.

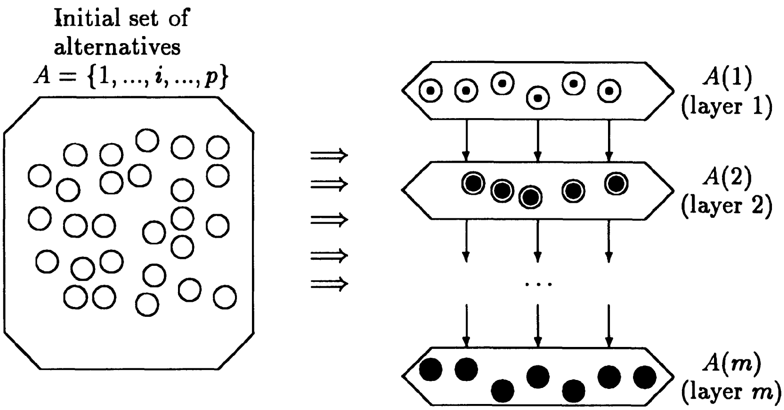

Fig. 1.6. Illustration for multicriteria ranking

A case $m = 2$ is often used for many problems. Here we may select the 1st (or 2nd) layer. Clearly, multicriteria ranking (construction of the above-mentioned layered structure) is a base for an ordinal assessment of DA's, and Ins.

The techniques for the multicriteria selection/ranking are well-known ([65], [74], [129], [369], [544], [550], etc.):

(1) statistical techniques [60];

(2) multi-attribute utility analysis ([150], [242]);

(3) interactive multi-criteria decision making ([256], [259], [260], [330], [484], [488], etc.);

(4) analytic hierarchy process [438];

(5) outranking techniques ([435], etc.);

(6) mathematical theory of choice ([10], etc.);

(7) knowledge bases ([322], etc.);

(8) neural network ([344], [447], [529], etc.);

(9) expert judgment ([277], [317], [327], etc.); and

(10) hybrid techniques [145].

Generally, it is reasonable to illustrate discrete multicriteria decision making problems on the basis of a specific morphological scheme (Fig. 1.7) [294]. We use the following notations for intermediate results: G_o, Q_o, G_{po}, C_o, C_o, S_o, S_{to}, R_o.

Several well-known problems are pointed out in Fig. 1.7: (a) holistic group ranking ♠; (b) unique choice problem *; and (c) regular group ranking problem ♡.

Main kinds of problems are the following:

(i) *clustering*: $A \Rightarrow Z \Rightarrow G_p \Rightarrow G_{po} \Rightarrow R_o$; and

(ii) *ranking*: $A \Rightarrow Z \Rightarrow G \Rightarrow G_o \Rightarrow (S_o, L_o) \Rightarrow S(S_t, L)$.

Planning of a series-parallel multi-period strategy on the basis of the method (algorithms/procedures) morphological environment is examined in section 3.6. This approach has been realized in a decision support system COMBI ([294], [297], [317]). Graphical interfaces of the system involve menu for a morphological method environment as a functional graph ([297], [317]) (Fig. 1.8). This functional graph consists of the two kinds of blocks:

(1) data (i.e., initial data as alternatives A, criteria K, and estimates Z; preference relation G; intermediate linear ordering L_o; intermediate layered structure S_o; results S and L; and

(2) transformation (articulation) operations (forming of preference relation; linear ranking; group ranking; aggregation; and direct solving).

In DSS COMBI, each transformation block allows to select an algorithm or a procedure from the corresponding algorithm/procedure base.

Bibliographic survey on multiple criteria decision making and many applied examples are contained in [489].

Type of data	Data	Specification			Deter-minis-tic	Fuzzy
		Com-plete	Incom-plete	None (under develop-ment)		
Initial	Alternatives $A = \{1, ..., i, ..., p\}$	♡ ♠		*		
	Criteria $K = \{1, ..., j, ..., d\}$	♡		*		
	Estimates of alternatives $Z = \{z_{ij}\}$	♡		*		
Inter-medi-ate	Preference structure of alternatives $G = (A, E)$		♡	* ♠		
	Proximity structure of alternatives $G_p = (A, E_p)$		♡	*		
	Hierarchy of criteria $Q = (K, V)$	♡		*		
	Proximity structure of criteria $Q_p = (K, V_p)$	♡		*		
Resul-tant	Layered structure (group ranking) S	♡ ♠				
	Two-layered structure S_t			*		
	Linear ranking L					
	Clustering R					

Fig. 1.7. Morphological scheme of discrete decision making problems

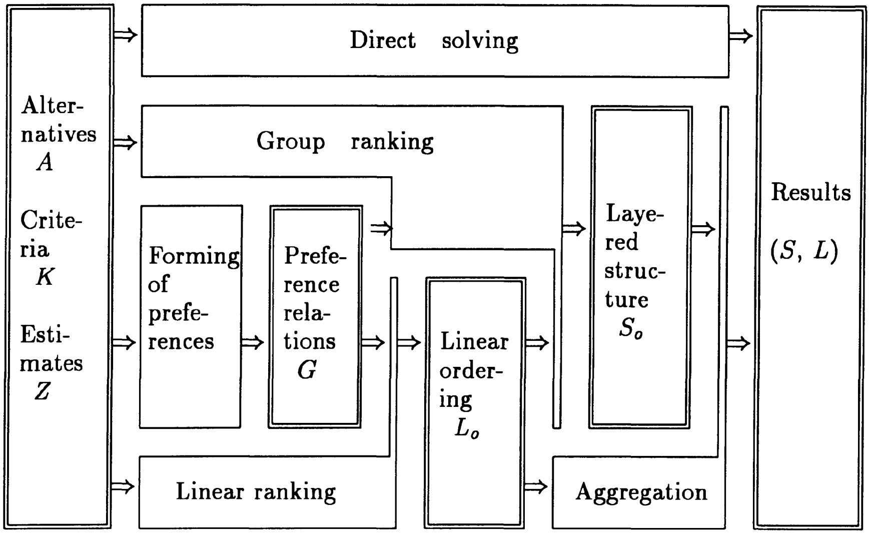

Fig. 1.8. Functional graph-menu for ranking

1.6.3 Coordination of Scales

Coordination of initial scales of sybsystems perfection is a sophisticated problem. In the main, we intend to utilize the following types of scales for measuring a perfection of system, and their elements (DA's, Ins):

(i) ordinal (including a case of interval estimates); and

(ii) poset.

Table 1.5 contains several basic situations of scales coordination.

Table 1.5. Situations of scales coordination

Initial scales of element perfection	Resultant coordinated scale of element perfection
1.Ordinal	Ordinal
2.Poset	Ordinal
3.Ordinal	Poset
4.Poset	Poset

Obviously, the problem involves the following main components: (i) initial different ordinal scales; (ii) additional information on correspondence between elements of the scale; (iii) models and procedures to build a resultant scale.

The same monotonicity of initial scales is assumed. It is reasonable to consider three basic series stages of the coordination process:

(1) analysis of initial scales $\{\Delta^k | i = 1, ..., m\}$;

(2) integration of initial scales into a preliminary aggregated scale Δ^a:

$$\{\Delta^k\} \Longrightarrow \Delta^a;$$

(3) approximation of the aggregated scale by a resultant scale of a required kind Δ^{ap}:

$$\{\Delta^a\} \Longrightarrow \Delta^{ap}.$$

Thus resultant processing is:

$$\{\Delta^k, k = 1, ..., m\} \Longrightarrow \{\Delta^a\} \Longrightarrow \Delta^{ap}.$$

In addition, we can use an index to indicate the type of the scale as follows: ordinal (o); poset (p). Note that an estimate for the scales may be an interval or fuzzy ones. So the following basic examples of scale coordination problems maybe considered:

(a) $\{\Delta^1_o, \Delta^2_o\} \Longrightarrow \Delta^a_o$;

(b) $\{\Delta^1_o, \Delta^2_o\} \Longrightarrow \Delta^a_p$;

(c) $\{\Delta^k_o, k = 1, ..., m\} \Longrightarrow \{\Delta^a_o\}$;

(d) $\{\Delta^k_o, k = 1, ..., m\} \Longrightarrow \{\Delta^a_p\}$;

(e) $\{\Delta^k_o, k = 1, ..., m\} \Longrightarrow \{\Delta^a_p\} \Longrightarrow \Delta^{ap}_o$;

(f) $\{\Delta^k_o, k = 1, ..., m\} \Longrightarrow \{\Delta^a_p\} \Longrightarrow \Delta^{ap}_p$; and

(g) $\{\Delta^k_p, k = 1, ..., m\} \Longrightarrow \{\Delta^a_p\} \Longrightarrow \Delta^{ap}_o$.

Fig. 1.9, 1.10, 1.11 illustrate problems of scale coordination (equivalence of scale elements is pointed out by double lines).

Clearly, the following factors may be taken into account:

(a) complexity of a procedure (i.e., comparison operations and their features) to obtain information on comparison of scale elements;

(b) usefulness of data presentation and expert perception;

(c) requirements of problems (e.g., synthesis) at the next stage of information processing;

(d) quality (precision, usability, etc.) of the resultant coordinated scale; and

(e) reliability of the results in the cases of incomplete, uncertain, and incorrect information.

Now we consider comparison operations for the case of two initial ordinal scales. Let $\Delta^1_o = \{a_1, ..., a_\ell, , ..., a_{l_1}\}$, and $\Delta^2_o = \{b_1, ..., b_\ell, , ..., b_{l_2}\}$ be initial scales, where $\forall \ell_1 > \ell_2\ a_{\ell_1} \succeq a_{\ell_2}$ (analogically for elements of Δ^2_o). Main operations to compare two elements $a_{\ell_1} \in \Delta^1_o$, $\ell_1 = 1, ..., l_1$ and $b_{\ell_2} \in \Delta^2_o$, $\ell_2 = 1, ..., l_2$ are the following:

(1) equivalence $a_{\ell_1} \sim b_{\ell_2}$; and

(2) preference $a_{\ell_1} \succ b_{\ell_2}$.

Moreover, this situation with several initial ordinal scales corresponds to the case 1 of Table 3. Here we can use the well-known algorithm for merging of several (m) ordered sets. Starting from the best elements of m sets we compare these elements (preference or equivalence) and form a resultant ordered set as the coordinated scale. This algorithm is based on the following number of pair comparisons:

$$O((l_1 + ... + l_k + ... + l_{m-1}) \lg m + (m-1)),$$

where $l_k = |\Delta^k_o|$ [252].

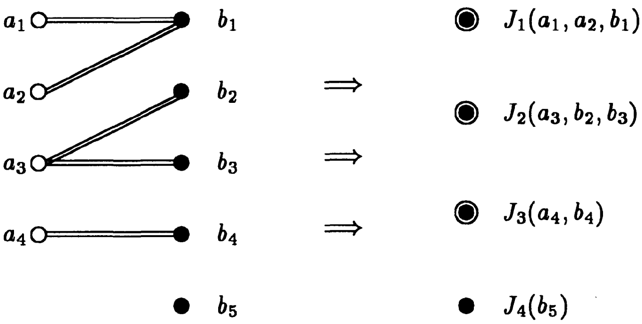

Fig. 1.9. Illustration for problem $\{\Delta^1_o, \Delta^2_o\} \Longrightarrow \Delta^a_o$

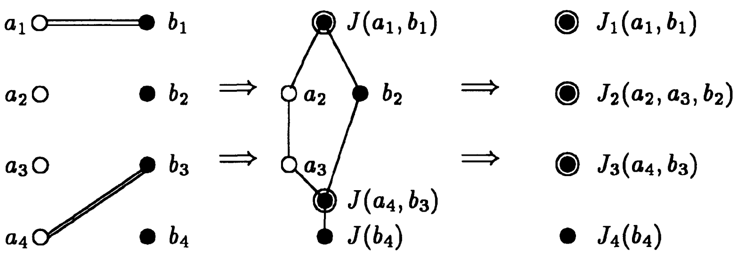

Fig. 1.10. Illustration for problem $\{\Delta^1_o, \Delta^2_o\} \Longrightarrow \Delta^a_p \Longrightarrow \Delta^{ap}_o$

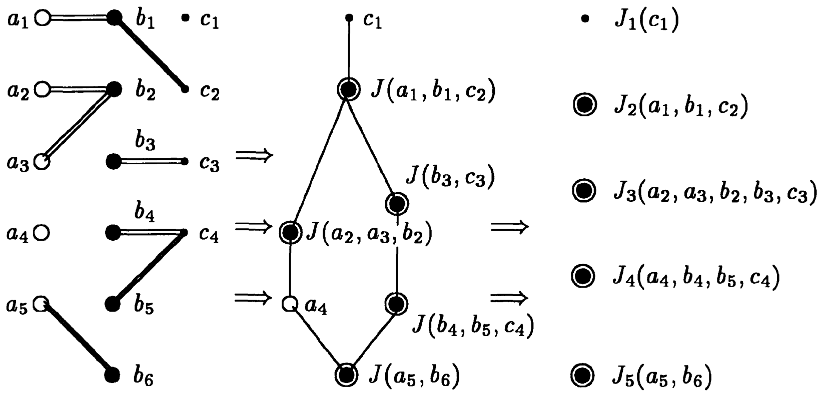

Fig. 1.11. Illustration for problem $\{\Delta_o^1, \Delta_o^2, \Delta_o^3\} \Longrightarrow \Delta_p^a \Longrightarrow \Delta_o^{ap}$

The problem with equivalence of scale elements may be considered as a specific matching problem for 2 or more initial linear ordered sets. Clearly that two initial scales may be analyzed as a bipartite graph, and m-scale coordination problem is based on m-partite graphs. In addition, it is reasonable to apply topological operations on scales (linear ordered sets), for example:

(a) compression (i.e., an element set is mapped into a resultant condensed element);

(b) expansion (i.e., an element is mapped into a set of several new ordered elements); and

(c) shifting.

Thus it is possible to transform initial ordered sets into resultant ones with the same number of elements (as alignment in mathematical biology). In our case, a vertex of the partite graph (a scale element) may correspond to several elements of another scale. Finally, we obtain the following properties of the problem:

(1) linear ordering for vertices of graph parts;

(2) possible correspondence of a vertex to several vertices of another graph part; and

(3) possible correspondence of a vertex to an additional vertex;

Some simple versions of the scale coordination problem are polynomial solvable, but mainly the problem is complicated. Note considered problem is close to the poset scheduling problem which is NP-hard [70]. It is necessary to take into account that the use of expert judgment (including uncertain information) transforms the problem into an ill-structured one. So more complicated comparison operations can be applied:

1. *Element-interval*:

1.1. equivalence $a_{\ell_1} \sim [b_{\ell'_2}, b_{\ell''_2}]$ (here and afterhere $\ell' < \ell''$);

1.2. preference: $a_{\ell_1} \succ [b_{\ell'_2}, b_{\ell''_2}]$;

1.3. inclusion $a_{\ell_1} \in [b_{\ell'_2}, b_{\ell''_2}]$.

2. *Interval-interval*:

2.1. equivalence $[a_{\ell'_1}, a_{\ell''_1}] \sim [b_{\ell'_2}, b_{\ell''_2}]$;

2.2. inclusion $[a_{\ell'_1}, a_{\ell''_1}] \subseteq [b_{\ell'_2}, b_{\ell''_2}]$;

2.3. strong preference: $[a_{\ell'_1}, a_{\ell''_1}] \succ [b_{\ell'_2}, b_{\ell''_2}]$ or $a_{\ell''_1} \succ b_{\ell'_2}$;

2.4. preference: $[a_{\ell'_1}, a_{\ell''_1}] \succeq [b_{\ell'_2}, b_{\ell''_2}]$

when the following two conditions are satisfied:

(a) $a_{\ell'_1} \succ b_{\ell'_2}$, and

(b) $\exists \ell_2 \in [\ell'_2, \ell''_2]$, $\ell_2 > \ell''_2$ and $a_{\ell''_1} \sim b_{\ell_2}$.

Note obtaining of a large resultant scale may be useful for the algorithm of searching for a morphological clique (composite decisions), because the large scale strains a space of global effectiveness of composite decisions. However a small and simple scale is preferable from the viewpoint of expert perception and data presentation.

In more complex situations, it is necessary to consider comparison and matching of several levels of perfection at the same time. In our opinion, the basic approach to this problem is based on expert judgement, and modeling. For example, a solving strategy is the following:

(1) to generate a hypothetical alternative for each scale element;

(2) to describe the above mentioned alternatives; and

(3) to solve a problem of multicriteria ranking for the alternatives (e.g., on the basis of pair comparison).

Thus we obtain resultant layers that correspond to elements of the resultant aggregated scale. One of the possible ways is presented in section 7.3 (solving of assignment problem on the basis of morphological metaheuristic).

Finally, let us point out that recently closed combinatorial problems (keyword matching, string matching, alignment, multidimensional matching, pattern matching, etc.) have been intensively studied in linguistics, measurement in sport, information retrieval, decision making, computer engineering, mathematical biology, etc. ([18], [35], [160], [425], [439], [501], [530], etc.).

In our opinion, the problems of scale coordination require additional investigation both from practical and theoretical viewpoints.

1.7 SUMMARY

In this chapter, we have identified the key questions answered by decomposable systems, their modeling, design principles. We have introduced a hierarchical morphological design approach and briefly described auxiliary problems (structural modeling, multicriteria ranking, and coordination of scales). Note that techniques of multicriteria ranking are well-known. In our opinion, composite systems, that implement many processes (ranking, clustering, combination of composite solving schemes), are prospective ones. The significance of structural modeling is increasing. Also, combinatorial problems of scale coordination require many additional studies.

2 SOME MATHEMATICAL PROGRAMMING PROBLEMS

2.1 KNAPSACK PROBLEM

Knapsack problem (KP) is a basic combinatorial NP-hard one ([160], [347], [379], etc.):

$$\max \quad \sum_{i=1}^{m} c_i x_i$$

$$s.t. \quad \sum_{i=1}^{m} a_i x_i \leq b$$

$$x_i = 0 \cup 1, \; i = 1, ..., m$$

and additional resource constraints

$$\sum_{i=1}^{m} a_{i,k} x_i \leq b_k; \; k = 1, ..., l;$$

where $x_i = 1$ if item i is selected, for ith item c_i is a value (profit), and a_i is a weight, etc. We assume the use of nonnegative coefficients in programming problems here and hereafter. KP is approximate solvable, e.g., a fully polynomial approximation algorithm exists which is polynomial both in the problem dimension (here in m), and $\frac{1}{\epsilon}$, where ϵ is an error ratio bound ([160], etc.). Fig. 2.1 illustrates KP.

The list of approaches for KP are the following (exact, near-optimal, and probabilistic) ([347], etc.):

(1) relaxations (continuous, Lagrangian, and surrogate);

(2) branch-and-bound algorithms;

(3) dynamic programming algorithms;

(4) polynomial-time (approximate) and fully polynomial-time (fast approximate) approximation schemes;

(5) heuristics (e.g., greedy algorithms);

(6) probabilistic methods; and

(7) composite techniques.

Note that similar approaches are used for other integer programming problems also, except fast approximate algorithms, which maybe applied for a limited number of problems (e.g., KP, multiple-choice knapsack problem, and some of their modifications) ([261], [442]).

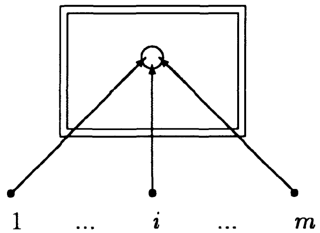

Fig. 2.1. Illustration for knapsack problem

There exist several ways of generalizing the KP:

(a) increasing the number of resource constraints;

(b) use of multicriteria descriptions for items;

(c) use of item dependencies, i.e., a special structure (poset) for items, e.g., KP with dependent items, multiple-choice knapsack problem, KP with compatibility, etc.; and

(d) use of a special structure (ordered set) for constraints (e.g., nested KP).

Now let us consider item dependence as the following structural requirements:

(1) compatibility (Ins) of items as a symmetric binary relation (i.e., the selected subset have to contain only compatible items); and

(2) dependence of items as an additional logical constraint of the following kind: $x_{i_1} \geq x_{i_2}$ $\forall$ i_1,i_2; or a dependence digraph (or graph) $Q = (I, D)$ consisting of these pairs (a set D).

We will take into account compatibility for multiple-choice knapsack problem, and for composition problems based on morphological analysis in further sections. The consideration of compatibility for KP leads to the search for a clique with the maximum total profit and a restricted total weight (profit clique). Tables 2.1, 2.2 present an example for the following problems: KP, profit clique (KP, and compatibility of selected items), limited clique (maximizing the number of selected compatible items, and constraint to total item weight as in KP); and maximal clique (maximizing the number of selected compatible items). The following notations are used: M is the number of items in a decision; $b = 1.3$ is the right-side constraint (additional constraints are not applied); selected items are shown by symbol $\star$; and for the decision

$$C = \sum_{i=1}^{m} c_i x_i; \quad C' = \sum_{i=1}^{m} a_i x_i.$$

Here we point out an important KP with item dependence as a tree-like digraph Q [228]. Johnson and Niemi have examined two cases for Q: (i) *outtree* (arcs corresponding to logical constraints directed from a corner), and (ii) *in-tree* (arcs direct to a corner). For the first case fully approximation scheme is proposed.

2.2 MULTIPLE-CHOICE KNAPSACK PROBLEM

Multiple-choice knapsack problem is based on grouping the items:

$$\max \quad \sum_{j=1}^{m} \sum_{i=1}^{q_j} c_{i,j} x_{i,j}$$

$$s.t. \quad \sum_{j=1}^{m} \sum_{i=1}^{q_j} a_{i,j} x_{i,j} \leq b$$

$$\sum_{i=1}^{q_j} x_{i,j} \leq 1; \; j = 1, ..., m$$

$$x_{i,j} = 0 \cup 1; \; i = 1, ..., q_j; \; j = 1, ..., m.$$

We can consider item group j $J_j = \{(i,j) \mid i = 1, ..., q_j\}$ as a set of DA's for a component j of a composite decision. Fig. 2.2 depicts a multiple-choice knapsack problem.

Table 2.1. Initial data and results

i	c_i	a_i	Knapsack	Profit clique	Limited clique	Maximal clique
1	0.3	0.1	⋆	⋆	⋆	⋆
2	0.5	0.2	⋆	⋆	⋆	⋆
3	1.2	0.5	⋆	⋆		⋆
4	0.4	0.2	⋆			
5	0.2	0.1	⋆	⋆	⋆	⋆
6	0.3	0.2	⋆			
7	0.4	0.3		⋆	⋆	⋆
8	0.1	0.2			⋆	⋆
9	0.1	0.3			⋆	⋆
C			2.9	2.6		
C'			1.3	1.2	1.2	
M			6	5	6	7

Table 2.2. Compatibility of items

i	1	2	3	4	5	6	7	8	9
1	.	1	1	1	1	1	1	1	1
2	1	.	1	0	1	0	1	1	1
3	1	1	.	0	1	1	1	1	1
4	0	1	0	.	1	1	1	1	1
5	1	1	1	1	.	1	1	1	1
6	1	0	0	1	0	.	1	1	0
7	1	1	1	0	1	0	.	1	1
8	1	1	1	1	1	1	1	.	1
9	1	1	1	0	1	0	1	1	.

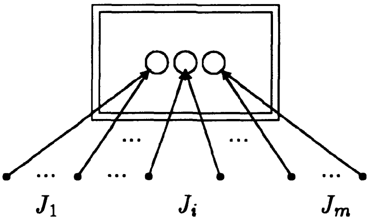

Fig. 2.2. Illustration for multiple-choice knapsack problem

Multiple-choice knapsack problem is approximate solvable ([347], etc.). It is the simple algebraic version of the composition problem. Clearly, above-mentioned approaches can be applied to multiple-choice knapsack problems too ([347], etc.).

Now let us consider multiple-choice knapsack problem with compatibility. Let $B = (\{J_j, j = 1, ..., m\}, V)$ be a ***condensed*** morphological graph describing a generalized interconnectivity, where V is a set of arcs of the following kinds:

(1) $v^{\star}(x, y)$, if all pair of elements from J_x and J_y are compatible;

(2) $v^{-}(x, y)$, if all pair of elements from J_x and J_y are not compatible; and

(3) $v^{+}(x, y)$, if some (not all) pair of elements from J_x and J_y (not all) are compatible, here we can consider feasible interconnection.

Obviously, $V = \{v^{\star}\} \cup \{v^{-}\} \cup \{v^{+}\}$. Thus we analyze the structure of V to reveal the following cases:

(a) $V = \{v^{\star}\}$, $\mid \{v^{-}\} \mid + \mid \{v^{+}\} \mid = 0$, (standard multiple-choice knapsack problem);

(b) $\mid \{v^{-}\} \mid > 0$, (decision are absent); and

(c) $V = \{v^{\star}\} \cup \{v^{+}\}$, $\mid \{v^{-}\} \mid = 0$.

In the last case, we introduce a special graph $B^{+} = (\{z_j\}, \{v^{+}\})$, where elements $\{z_j\}$ correspond to vertices which are connected by arcs of type v^{+}. Finally, we consider the following cases:

(i) B^{+} is a chain;

(ii) B^{+} is a tree;

(iii) B^{+} is the complete graph (standard multiple-choice knapsack problem); and

(iv) an arbitrary graph.

It follows in the standard way that if B^{+} is a chain or a tree, then the corresponding version of MCP has a fast approximate algorithm. Thus the kind of graph B^{+} can be used as a test for ϵ-approximation solvability.

2.3 KNAPSACK PROBLEM WITH SPECIFIC CONSTRAINT

Another specific knapsack-type problem with tree-like logical constraint was investigated in [289]. Fast approximate polynomial algorithm with relative errors of the objective function and constraint for the problem has been proposed. This problem corresponds to the design of an over-lay structure of complex modular software or data, but other applications are possible too (e.g., planning of quality analysis). The integration of software modules requires additional memory, but allows to decrease a time (i.e., frequency) of loading some corresponding modules.

Let us consider the combinatorial problem. Let $G = (A, \Upsilon)$ be an oriented tree, where A is a set of vertices (software or data modules). Υ is a multi-valued mapping of A into A. Arcs of G are oriented from the root $a_o \in A$ to leaf vertices. Each vertex $a \in A$ has a positive weight (required volume of RAM) $\beta(a) > 0$. Each arc $(a^{'}, a^{''})$ $(a^{'}, a^{''} \in A$ and $a^{''} \in \Upsilon a^{'})$ has a weight (i.e., an initial frequency of loading into RAM) $w(a^{'}, a^{''}) > 0$. This arc weight corresponds to the frequency of calling (and loading) from module $a^{'}$ to module $a^{''}$.

Let $\pi(a^1, a^l) =< a^1, ..., a^i, ..., a^l >$ be a path $(a^{j+1} \in \Upsilon a^j,\ \ j = 1, ..., l-1)$. We propose for each path a weight $\lambda(\pi(a^1, a^l) = \sum_{i=1}^{l} \lambda(a^i)$. Denote by a weight of graph G the value

$$\lambda(G) = \max_{a^{''} \in A^o} \{\lambda(\pi(a_o, a^{''}))\},$$

where $A^o = \{a \in A \mid |\Upsilon a| = 0\}$ is a set of leaf vertices. Let $G_a = (A_a, \Upsilon)$ is a subtree with a root $a \in A$, and A_a contains vertex a and all other vertices, that can be reached from a. Graph $(A_a \setminus a, \Upsilon)$ is called *tail* of vertex a, and a value $\lambda^-(a) = \lambda(G_a) - \lambda(a)$ is called a *tail* weight of vertex a. Clearly that

$$\lambda(a) = \max_{a^{'} \in \Upsilon a} \{\lambda(G_{a^{'}})\}.$$

We examine weight $w(a)$ and binary variable $x(a)$ $\forall a \in A \backslash a_o$ (1 corresponds to a situation when the arc, directed to a, is condensed). Now let us define a transformation of graph G on the basis of integrating the vertices $a^{'}$ and $a^{''}$ as follows:

(a) vertex $a^{'}$ is changed into $J(a^{'}, a^{''})$ with the following properties: $\lambda(J(a^{'}, a^{''})) = \lambda(a^{'}) + \lambda(a^{''})$ and $\Upsilon J(a^{'}, a^{''}) = (\Upsilon a^{'} \bigcup \Upsilon a^{''}) \setminus a^{''}$;

(b) vertex $a^{''}$ and arcs, which are oriented from the vertex, are deleted.

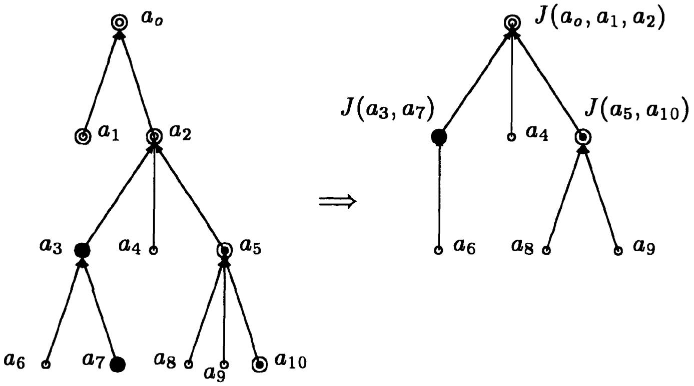

Fig. 2.3. Transformation of complex software by integration of modules (design of over-lay structure)

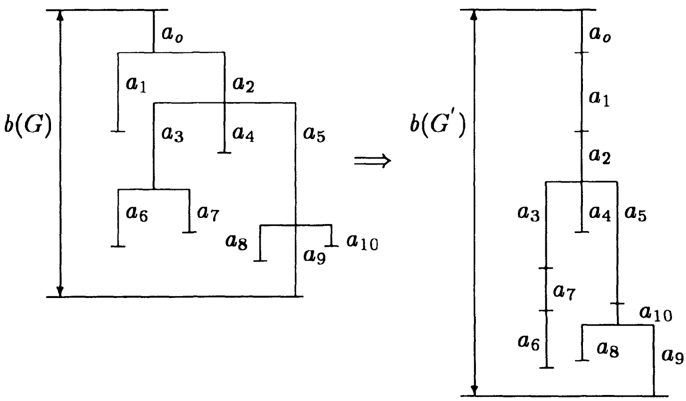

Fig. 2.4. Usage of memory (RAM)

For graph G we propose a binary vector $\kappa(a)$ that involves all $x(a)$ $\forall a \in A \setminus a_o$. Thus we examine the weights of vertex a and its *tail* as functions of vector κ: $\lambda(a, \kappa)$, $\lambda^-(a, \kappa)$. Now let us consider a problem (kind 1):

$$\max \qquad W(\kappa) = \sum_{a \in A \setminus a_o} x(a) w(a)$$

$$s.t. \qquad \lambda(a_o, \kappa) + \lambda^-(a_o, \kappa) \leq b,$$

where b is a positive constant (e.g., a volume of accessible RAM). Problem 1 is illustrated in Fig. 2.3 and 2.4 by an example of designing the over-lay structure on the basis of the module integration, when different software or data modules can apply the same parts of RAM. In addition, we examine analogical problem (kind 2) with other constraints as follows:

$$\lambda(a_o, \kappa) \leq b^-, \;\; \lambda^-(a_o, \kappa) \leq b^+, \;\; b^- + b^+ = b.$$

Let us consider some simple cases of the problem. Let $\Upsilon a_o = \{a_1, ..., a_i, ..., a_m\}$, and u_i corresponds to an arc (a_o, a_i), $w(u_i) = w_i$. Then corresponding problem 1 (equivalent to KP) is:

$$\max \quad \sum_{i=1}^{m} x_i w_i$$

$$s.t. \quad \lambda(a_o) + \sum_{i=1}^{m} x_i \lambda(a_i) \leq b,$$

$$x_i = 0 \cup 1.$$

The objective function in other simple cases (Fig. 2.5a, 2.5b, 2.5c, 2.5d) is analogical, and hereafter we examine only constraints.

Next problem 1.1 (Fig. 2.5a) has the following constraint:

$$\lambda(a_o) + \sum_{i=1}^{m} x_i \lambda(a_i) + \max_{1 \leq i \leq m} ((1 - x_i)\lambda(a_i)) \leq b.$$

This problem corresponds to a kernel load in many software packages. Similar problem 1.2 (Fig. 2.5b) of kind 2 is the following:

$$\lambda(a_o) + \sum_{i=1}^{m} x_i \lambda(a_i) \leq b^-,$$

$$\max_{1 \leq i \leq m} ((1 - x_i \lambda(a_i)) \leq b^+.$$

Problem 1.3 (Fig. 2.5c) is:

$$\lambda(a_o) + \sum_{i=1}^{m} x_i \lambda^-(a_i) + \max_{1 \leq i \leq m} ((1 - x_i \lambda^-(a_i)) + \lambda^+(a_i)) \leq b.$$

It is reasonable to point out the following properties of this problem:

(a) a_i ($\forall a_i \in \Upsilon a_o$) has the weight $\lambda^-(a_i)$;

(b) a_i ($\forall a_i \in \Upsilon a_o$) has the only one son with a weight $\lambda^+(a_i)$, and the value is the *tail* weight; and

(c) only condensing the following arcs (a_o, a_i) $(i = 1, .., m)$ is admissible.

Thus we have examined a sequence of simple problems on the basis of KP: KP or problem 1, 1.1, 1.2, 1.3. In the same way we consider a sequence of auxiliary problems on the base of multiple-choice knapsack problem (Fig. 2.5e, 2.5f, 2.5g, 2.5h). In our case, multiple-choice knapsack problem or problem 2 (Fig. 2.2) is the following:

$$\max \qquad W(\{x_{ij}\}) = \sum_{i=1}^{m}\sum_{j=1}^{q_i} w(a_{ij})x_{ij}$$

$$s.t. \qquad \lambda(a_o)\sum_{i=1}^{m}\sum_{j=1}^{q_i} x_{ij}\lambda(a_{ij}) \leq b,$$

$$\sum_{j=1}^{q_i} x_{ij} = 1, \; i = 1, ..., m; \; x_{ij} = 0 \cup 1.$$

Here we apply the following set of Boolean vectors in auxiliary problems:

$$X = \{\kappa = (x_{ij}^1; x_{ij}^2) | \; x_{ij}^1, x_{ij}^2 = 0 \cup 1; \; j = 1, ..., q_i; \; i = 1, ..., m\}$$

In addition, the following constraint has to be taken into account in all auxiliary problems:

$$\sum_{j=1}^{q_i} x_{ij}^1 = 1, \; \forall i; \; x_{ij}^2 \leq x_{ij}^1, \; \forall i, j.$$

Also, the following modified objective function is used:

$$W(X) = \sum_{i=1}^{m}\sum_{j=1}^{q_i}(x_{ij}^1 w^-(a_{ij}) + x_{ij}^2 w(a_{ij})).$$

Now, for example, let us consider auxiliary problem 2.4, that corresponds to kind 2 above (Fig. 2.5h):

$$\lambda(a_o) + \sum_{i=1}^{m}\sum_{j=1}^{q_i} x_{ij}^2 \lambda^-(a_{ij}) \leq b^-,$$

$$\max_{i,j} \; ((1 - x_{ij}^2)\lambda^-(a_{ij}) + \lambda^+(a_{ij})) \leq b^+.$$

For the sequence of simple problems above, we can apply approximation algorithms, which are based on an ϵ-approximate algorithm $(\epsilon \in [0, 1])$ for KP ([347], [442], etc.). In the case of these algorithms, an estimate of an operation number is similar to the estimate for KP ([347], etc.), and equals $O(\frac{m^2}{\epsilon})$ [289]. The algorithms apply ordering of elements from set Υa by non-decreasing of $\lambda(a_i)$ or $(\lambda^-(a_i) + \lambda^+(a_i))$.

Solutions of auxiliary problems are based on similar approximation approach to multiple-choice knapsack problem with the following estimates of operations and required memory accordingly ([347], etc.):

$$O(\frac{m}{\epsilon}\sum_{i=1}^{m} q_i), \quad O(\frac{m^2}{\epsilon}\max_{1\leq i\leq m}\{q_i\}).$$

Unfortunately, we could not construct an algorithm with similar estimates for the auxiliary problem 2.3 (Fig. 2.5g) [289]. As a result, the (ϵ, δ)-approximate algorithms with the following estimates (number of operations, and required memory) have been proposed:

$$O(\frac{m}{\epsilon\delta}\sum_{i=1}^{m} q_i), \quad O(\frac{m^2}{\epsilon}\max_{1\leq i\leq m}\{q_i\}),$$

where δ is a relative error for constraints.

In the more general case, when G is a k-level tree, the algorithm is based on series-parallel solving of simple and auxiliary problems above for parts (i.e., branches) of G [289]. Fig. 2.6 contains a Bottom-Up solving scheme for the tree-like G:

Step 1. Problems 1.2.
Step j $(j = 2, ..., k-2)$. Problems 2.4
Step $(k-1)$. Problem 2.3.

Estimates of the algorithms are as follows (i.e., operations, and memory):

$$O(\frac{n^2\eta^5(a_o)}{\epsilon\delta^4}), \quad O(\frac{m^2\eta^4}{\epsilon\delta^4}),$$

where $m(a) = |\Upsilon(a)|$, $m = \max_{a\in A} m(a)$, $\eta(a) =| A_a \backslash \{a' \in A_a \mid |\Upsilon a'| = 0\} |$.

In the case of 3-level tree, the estimate of the operation number is:

$$O(\frac{n^2\eta^4(a_o)}{\epsilon\delta^3}).$$

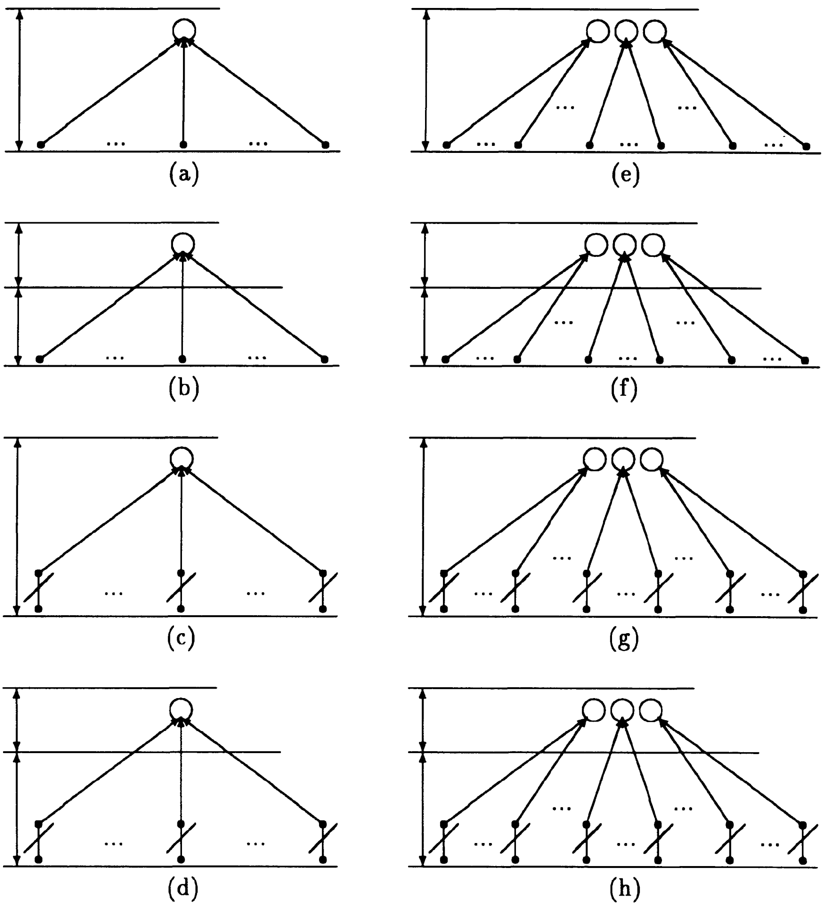

Fig. 2.5. Illustrations for simple and auxiliary problems

2.4 INTEGER QUADRATIC PROGRAMMING PROBLEM

Here we consider a generalization of multiple-choice knapsack problem by taking into account additive profits of item compatibility. Let a nonnegative value $d(i, j_1, k, j_2)$ be a profit of compatibility between item j_1 in group J_i and item j_2 in group J_k.

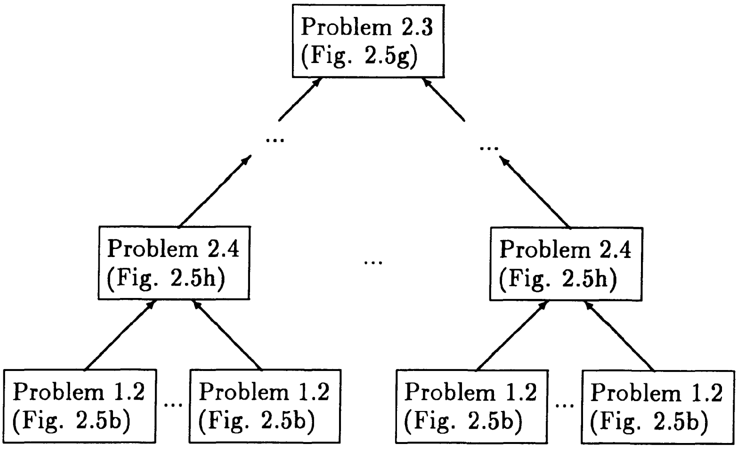

Fig. 2.6. Bottom-Up solving scheme for tree-like graph

Also, this value of compatibility is added to the objective function. Such quadratic programming problem is:

$$\max \quad \sum_{i=1}^{m} \sum_{j=1}^{q_i} c_{i,j} x_{i,j} + \sum_{l<k} \sum_{j_1=1}^{q_l} \sum_{j_2=1}^{q_k} d(l, j_1, k, j_2) x_{l,j_1} x_{k,j_2}$$

$$l = 1, ..., m; \;\; k = 1, ..., m$$

$$s.t. \quad \sum_{i=1}^{m} \sum_{j=1}^{q_i} a_{i,j} x_{i,j} \leq b$$

$$\sum_{j=1}^{q_i} x_{i,j} \leq 1; j = 1, ..., m$$

$$x_{i,j} = 0 \cup 1; \;\; i = 1, ..., m; \;\; j = 1, ..., q_i.$$

The compatibility profit above may be examined for KP too. In [71] a modification of KP is considered, in which compatibility profits and analogical quadratic elements in the resource constraint are used. However the quantitative assessment of these interactive impacts on the profit and/or resource is a complicated process.

2.5 INTEGER NONLINEAR PROGRAMMING

The design of a complex system with redundancy has been widely used in engineering. Berman and Ashrafi have examined integer nonlinear programming (INLP) problems for the modular design of independent DA's. They consider the following four versions of selecting an optimal module set:

(1) for one function system (without redundancy); it is a modification of multiple-choice knapsack problem;

(2) for one function system (with redundancy);

(3) for a system with k series functions (without redundancy); and

(4) for a system with k series functions (with redundancy).

Let us consider only model 2 above ($p(i,j)$ is the reliability of version j for module i (Fig. 2.7):

$$\max \quad \prod_{i=1}^{m}(1-\prod_{j=1}^{q_i}(1-p_{i,j}x_{i,j}))$$

$$s.t. \quad \sum_{j=1}^{q_i} x_{i,j} \geq 1, \; i=1,...,m$$

$$\sum_{i=1}^{m}\sum_{j=1}^{q_i} c_{i,j}x_{i,j} \leq b$$

$$x_{i,j} = 0 \cup 1, \; i=1,...,m, \; j=1,...,q_i.$$

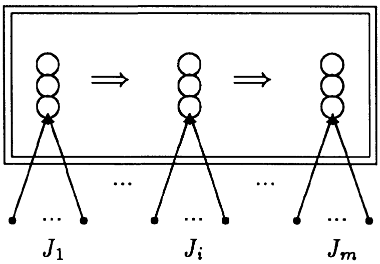

Fig. 2.7. Illustration for integer non-linear programming

Mainly for integer nonlinear programming the following techniques are used:

(1) branch-and-bound techniques;

(2) dynamic programming techniques; and

(3) heuristics including reducing to linear integer or continuous programming problem ([41], [379], etc.).

2.6 MIXED-INTEGER NONLINEAR PROGRAMMING

The use of mixed-integer nonlinear programming (MINLP) techniques is examined for the synthesis of engineering systems in ([113], [186]). Generally, this approach is based on the following steps:

Step 1. Generating the superstructure, and DA's.
Step 2. The structure is modeled as the MINLP problem:

$$\max \quad F(x,y)$$

$$s.t. \quad h(x,y) = 0$$

$$g(x,y) \leq 0.$$

Step 3. Solving the problem where x is a binary vector (existence or selection of the items or DA's, i.e, 0 not selected, 1 selected); y is a vector of continuous variables (e.g., sizes, parameters); $F(x,y)$ is the objective function (e.g., weight, cost); $h(x,y) = 0$ represents the performance or analysis equations; and $g(x,y) \leq 0$ represents the design specifications and logical constraints.

For many engineering applications in synthesis, the dominant structure is that the MINLP is most often linear in 0/1- variables with nonlinearities being present in the continuous variables. The following techniques are used for problems above ([186], [379], etc.):
(a) the branch and bound method for MILP;
(b) specialized combinatorial optimization techniques;
(c) the reduced gradient method; and
(d) interior point methods.

2.7 BILEVEL PROGRAMMING

Systematic analysis of mathematical models for hierarchical management systems has been proposed by Mesarovic, Macko and Takaharo in [353]. This book has described many issues including bilevel programming problems, and global optimization problems. In recent decades, there exists a new trend to analyze and to apply some generalized optimization problems (e.g., global optimization, hierarchical optimization) from two basic viewpoints as follows: theoretical investigation, and applications including multi-processor computing ([212], [356], etc.). In this context, let us consider basic problems of bilevel/multilevel programming, that are under intensive study of many investigators ([16], [522], etc.). A bibliography review on the topic is presented in [522]. The (continuous) bilevel programming problem (BPP) is formulated as follows ([356], [522], etc.):

$$\min_{x,y} \quad F(x,y)$$

$$s.t. \quad g(x,y) \leq 0,$$

where y (for each value of x) is the solution of the so-called lower level problem:

$$\min_{y} \quad f(x,y)$$

$$s.t \quad h(x,y) \leq 0,$$

with $x \in R^{nx}$, $y \in R^{ny}$; $F, f : R^{nx+ny} \rightarrow R$; $g : R^{nx+ny} \rightarrow R^{nu}$; and $h : R^{nx+ny} \rightarrow R^{nl}$. In addition, let us denote the following:

(1) *upper* level: variables: x, constraints: $g(x,y) \leq 0$, objective function: $F(x,y)$;

(2) *lower* level: variables: y, constraints: $h(x,y) \leq 0$, objective function: $f(x,y)$.

The BPP is convex if $f(x,y)$ and $h(x,y)$ are convex functions of y $\forall x$, and basic classes of BPP are the following [522]: linear BPP (all functions involved are affine), linear-quadratic BPP (f is a convex quadratic function), quadratic BPP (F and f are convex quadratic functions). The simplest version of BPP, i.e linear BPP, is HP-hard ([227], [199], etc.). Some problems of mathematical programming (e.g., minimax problems, linear integer problems, bilinear and quadratic problems) can be rewritten as a linear BPP [522].

Note that many applications of BPP (e.g., network design/management, facility location) have *discrete* nature, and are similar to combinatorial synthesis.

Mainly the following basic classes of algorithms for BPPs are pointed out ([15], [522], etc.):

(1) extreme point algorithms (linear BPPs, linear-quadratic BPPs);

(2) branch and bound algorithms (convex BP, integer linear BPP, integer quadratic BPP);

(3) complimentary pivot algorithms (linear BPPs, linear-quadratic BPPs);

(4) descent methods (convex BP, nonlinear BPPs);

(5) penalty function methods; and

(6) artificial intelligent based approaches.

2.8 SUMMARY

In this chapter, we have described basic optimization problems, that can be applied for the selection and synthesis of items (in particular, DA's). Also it is reasonable to point out the importance of the following: (i) study of versions for described problems under uncertainty, and (ii) and real applications of the problems.

3 MORPHOLOGICAL APPROACH AND MODELS

3.1 MORPHOLOGICAL ANALYSIS

Let us consider composition problems on the base of morphological analysis. Initial information is the following: a set of morphological classes; DA's of each morphological class; binary compatibility between elements for each pair of elements, that are of different morphological classes (0 corresponds to an infeasible pair). Also, we search for the following composition of DA's (one representative of each morphological class) with nonzero compatibility

$$S = \{S(1), ..., S(i), ..., S(m)\},$$

where $S(i)$ is component i. The problem was proposed by Zwicky [554]. Howard has proposed morphological tables to generate management decisions [213]. Ayres has examined the problem on the basis of searching for the nearest feasible composition S to a given one [22].

An analysis of analogical models has been executed in ([127], [254], [476], etc.). Note that close problem of distinct representatives has been studied in ([194], [254], [425], etc.). Generally, this problem is NP-complete ([254], etc.).

3.2 MORPHOLOGICAL CLIQUE PROBLEM

3.2.1 Preliminary Description of Basic Problem

Now let us consider a basic version of morphological clique problem as a generalization of morphological analysis ([295], [296], [300] etc.):

(a) the common ordinal scale for DA's (priority $r = 1, ..., l$; 1 corresponds to the best priority);

(b) the ordinal scale for compatibility $0, ..., u$ (u corresponds to the best one); and

(c) the vector of system excellence for decision S:

$$N(S) = (w(S); n(S)),$$

where $w(S)$ is the minimum of pairwise compatibilities in S,

$$n(S) = (n(1), ..., n(r), ..., n(l)),$$

where $n(r)$ is the number of components of the rth quality in S (a histogram). Clearly, $n(S)$ is a fuzzy estimate ([34], etc.).

Thus we search for feasible decisions which are nondominated by $N(S)$. Fig. 3.1 demonstrates two equivalent presentations of an instance of the lattice on $N = (w; n(1), n(2), n(3))$ for the following case: $w = const$, $m = 3$, $l = 3$. The complete lattice on N contains u interconnected lattices for $w = 1, ..., u$. In addition, we can use layers of the system excellence, e.g., ideal decisions (of the best DA's), Pareto-effective by N DA's, etc.

Trivially, a basic solving scheme is:

Step 1. To construct feasible compositions.
Step 2. To select Pareto-effective composite DA's.

Evidently, the problem is NP-hard. Usually for constructing the feasible compositions there are used backtracking, and the following two operations ([254], [476]):

(a) rejecting the infeasible DA's; and

(b) revealing the situation when the composite decision does not exist.

In our case, more complex scales for DA's, Ins, and S allow to apply a direct solving scheme both from $w = u$, and $r = l$. Moreover, experts can participate at all stages of the solving process.

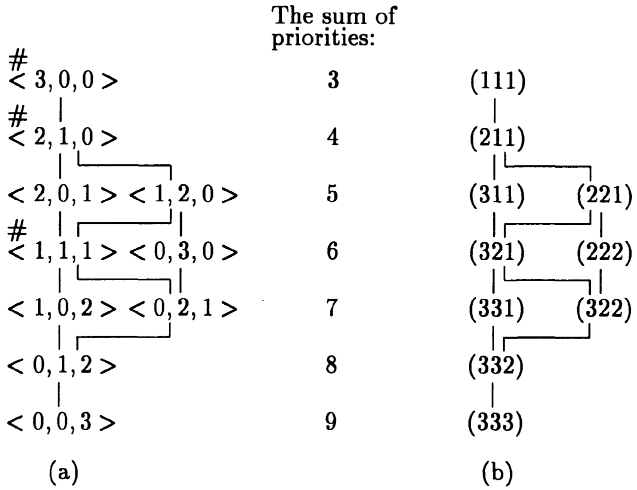

Fig. 3.1. Two equivalent presentations of an instance of the lattice on $N = (w; n(1), n(2), n(3))$, w =const, $m = 3$, $l = 3$;
(a) position (histogram) presentation with the numbers of DA's $(n(S))$,
(b) direct presentation with the priorities of DA's.

The considered morphological clique problem is the basic one in HMMD. Fig. 3.2 depicts a relationship of several well-known combinatorial problems and morphological clique problem.

3.2.2 Formulation of Problem

Let $G = (\{A(j), j = 1, ..., m\}, \{\{E(j), j = 1, ..., m\} \cup E_c\})$ be a morphological graph, where $A(j)$ is a set of vertices (design alternatives DA's) in morphological class j, $\mid A(i) \& A(j) \mid = 0$, $E(j)$ is a set of arcs inside the morphological class j (partial order), E_c is a set of arcs (compatibility, interconnection Ins) between DA's of different morphological classes (symmetric binary relation). We will use priorities besides of $E(j)$. The problem is:

Find a morphological clique (morphological scheme, composite decision or DA S) that consists of the best DA's (one representative of each $A(j)$) with taking into account admissibility.

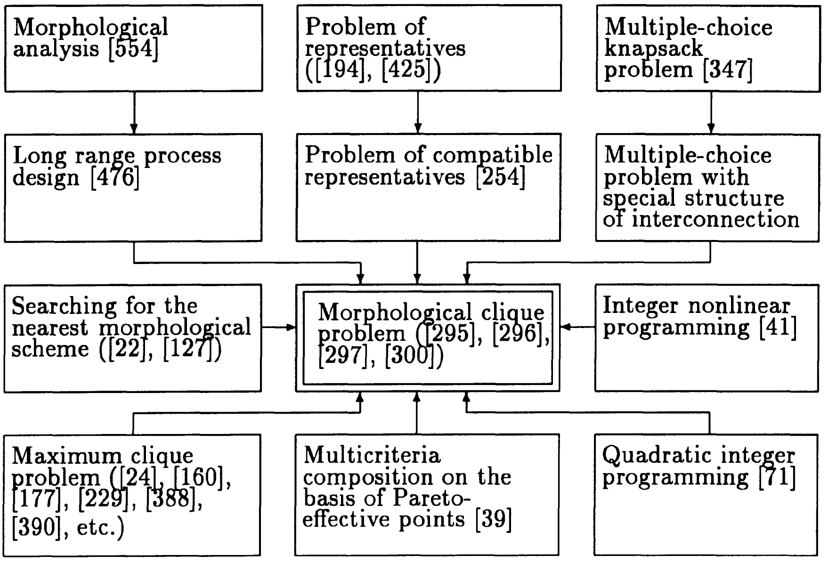

Fig. 3.2. Some combinatorial models of synthesis

Fig. 3.3 contains an example of the morphological clique.

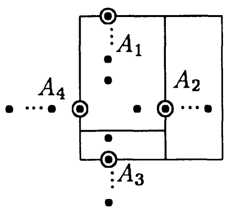

Fig. 3.3. Example of morphological clique

Morphological space is the following: $\Lambda_o = A(1) \star ... \star A(j) \star ... \star A(m)$. Let $S = \{S(1), ..., S(j), ..., S(m)\}$ be an admissible decision (morphological scheme), where component $S(j)$ is an element of $A(j)$, $w(e_c(S(a), S(b)))$ is the arc weight for each component pair $S(a), S(b) \in S$, and

$$w(S) = \min \{w(e_c(S(a), S(b))) \mid \forall S(a), S(b) \in S\} > 0$$

corresponds to admissibility by Ins. Denote by $Z = \{S\}$ a set of admissible decisions. We will use parametric space $\Lambda(w_o)$, where $w > w_o$. Quality vector of S by components is the following: $n(S) = (n(1), ..., n(i), ..., n(k))$, where $n(i)$ is the number of elements with a priority $r = i$ ($r = 1$ corresponds to the best level). $N(S) = (w(S); n(S))$ corresponds to a generalized quality (performance). Additional designations are:

I. Decisions:
(1) Y is the set of composite decisions which are Pareto-effective by N;
(2) Id is the set of ideal composite decisions (of the best components).

II. Parameters of problems:
(1) $w_o = \max\{w\}, ..., 1$ is a limit for the minimum of $w(S)$;
(2) $i = 1, ..., k$ is a limit for the worst design alternative in S, we examine $A(i,j) = \{a \in A(j) \mid r(a) < i+1\}$ and

$$B_i = A(i_1) \star ... \star A(i_j) \star ... \star A(i_m);$$

(3) $\Gamma = \{N''\}$ or $\{n'\}$ is the set of reference points.

III. Basic problems:
(1) Φ is constructing of Z;
(2) Ψ is searching for Y.

IV. Parametric problems and decisions:
(1) $\Phi_i(w_o, \Gamma) \Rightarrow Z_i(w_o, \Gamma)$;
(2) $\Psi_i(w_o, \Gamma) \Rightarrow Y_i(w_o, \Gamma)$.

Note the following lemma is evident:

Lemma 3.1.
Suppose $S \in Y(w_o = l)$, and $S' \in Y(w_o = l-1)$; then $N(S') \succeq N(S)$.

Now let us consider the following problem:

$$Arg\ \{N(S_o) \mid N(S_o) \succeq N(S) \ \ \forall S \in Z\}. \tag{3.1}$$

3.2.3 Algorithms

Basic solving scheme for (3.1) consists of the stages (algorithm 1):

Step 1 (Problem Φ). To construct $Z = Z(\Lambda)$.

Step 2 (Problem Ψ). To select Pareto-effective point set $Y = Y(Z)$.

Step 2 of algorithm 1 has complexity that is a polynomial in $| Z |$. Since Φ is NP-complete, it is reasonable to reduce its dimension, e.g., to use several subproblems with lower dimension. Consider the problem $\Psi(\Gamma)$:

$$Arg\{N(S_o) \mid (N(S_o) \succeq N(S), S \in \Lambda_o)\&$$

$$((incomparable\ N(S_o), N(S)) \cup (N(S_o) \succ N' \in \Gamma)\}. \qquad (3.2)$$

Further, we examine the set $Y(N')$ and partial order on the vectors $\{N(S)\}$, and by inclusion for $Y(N(S))$. It is evidently the following:

Property 1. If $N' \succeq N''$, then $Y(N'') \ni Y(N')$.

Let $A_{i,j} = \{a \in A_j \mid r(a) < r+1\}$ and, accordingly, $\Lambda_i = A_{i,1} \star ... \star A_{i,j} \star ... \star A_{1,m}$, $\Lambda_i(N')$, Ψ_i for (3.1), and $\Psi_i(N')$, $Y_i(N')$ for (3.2). Finally, consider algorithm 2 [300]:

Step 0. Initialization: $w' := \max(w)$; $w'' := 1$; $i' := 1$, $i'' := k$. The set of decision Y is empty ($|Y| = 0$); and $\Gamma = \{(0; 0, ..., 0, m)\}$.
Step 1. $w_o := w''$.
Step 2. $i := i'$.
Step 3. Solving of problem $\Phi_i(w_o, \Gamma)$, i.e., construction of $Z_i(w_o, \Gamma)$.
Step 4. Solving of problem $\Psi_i(w_o, \Gamma)$, i.e., selection of $Y_i(w_o, \Gamma)$.
Step 5. If $| Y_i(w_o) | > 0$ then *begin* $Y := Y \cup Y_i(w_o)$; $\Gamma := \Gamma \cup Y_i(w_o)$; if $i = 1$ then *Stop*; $i'' := i - 1$; go to *Step 2*, *end.*
Step 6. If $w_o = w''$ then go to *Stop* else $w_o := w_o - 1$ and go to *Step 2.*
Step 7. If $i = i''$ then go to *Step 1* else $i := i + 1$ and go to *Step 3.*

Algorithm 2 contains subproblems and the only subproblem $\Phi_i(w_o, \Gamma)$ is NP-complete ([254], etc.). All other parts of algorithm 2 are polynomial in problem parameters and $| Z_i(w_o, \Gamma) |$. Mainly, the dimension of problem $\Phi_i(w_o, \Gamma)$ is not high, and it is possible to use for this problem backtracking.

In addition, it is reasonable to use algorithmic rules, e.g., as follows:

(i) to reject dominated elements or decisions; and

(ii) to reveal the situation when admissible decisions do not exist ([300], [476]).

Fig. 3.4 demonstrates a subgraph that can not be included into a decision (*negative* example).

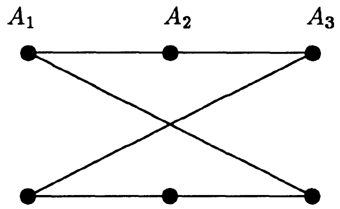

Fig. 3.4. *Negative* example for morphological clique

Fig. 3.5 illustrates a generalized idea of algorithm 2. For beginning we try to search for the best decisions near the ideal point (stage 1, domain 1), and so on.

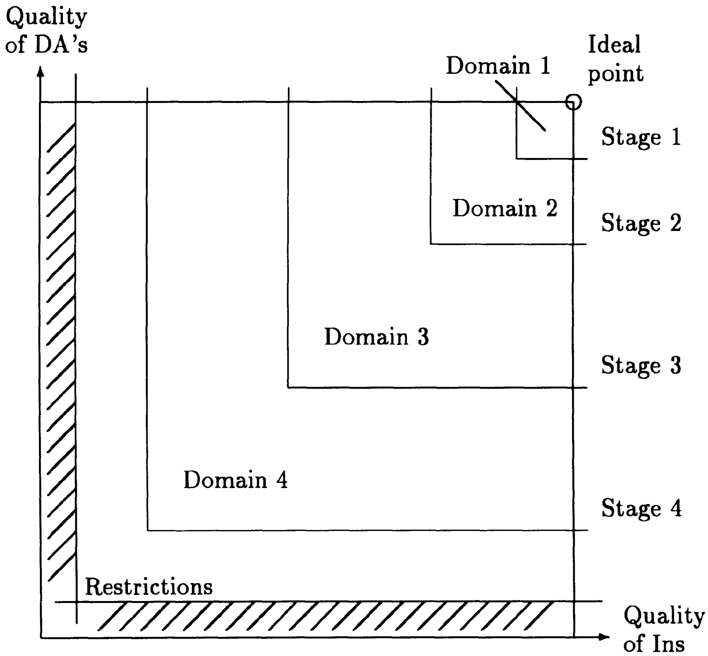

Fig. 3.5. Generalized idea of algorithm 2

3.2.4 Improvement of Composite Decisions

In our opinion, now the analysis and improvement of complex decisions become crucial problems in the design and planning processes. Let us consider the analysis and improvement of composite DA's. Suppose $S(j)$ is a component of composite decision

$$S = S(1) \star ... \star S(j) \star ... \star S(m),$$

$$S^{'} = S(1) \star ... \star S(j-1) \star S(j+1) \star ... \star S(m)$$

is an auxiliary composite decision without element $S(j)$ (for a reduced problem). It is assumed that n is a standardized vector. Thus we can consider the following:

Definition 3.1. $S(j)$ is called:
(a) S-*improving* element, if $N(S) \succ N(S^{'})$;
(b) S-*neutral* element, if $N(S) = N(S^{'})$; and
(c) S-*aggravating* element, if $N(S^{'}) \succ N(S)$.

We will use the analogical definition for w. Note that these elements of composite DA's are close to graphs elements proposed in [201]. Now let us consider a proximity for composite DA's. Let θ be a variation by w or n for $S^{'}, S^{''}$:

$$\theta_w(S^{'}, S^{''}) = w(S^{'}) - w(S^{''}), \quad \theta_n(S^{'}, S^{''}) = n(S^{'}) - n(S^{''}).$$

Thus we obtain the integer metric θ_w and a vector proximity θ_n.

Here we use the integer metric both for θ_w and θ_n as a sum of variation units.

Similar approach may be used for a sets of composite DA's. Now it is possible to introduce the definition of a neighbor for the composite decisions:

Definition 3.2. A composite decision $\{S^{''}\}$ is called a θ_o-neighbor of a composite decision $S^{'}$ by w or n, if the following condition is correct:

$$\theta_w(S^{'}, S^{''}) + \theta_n(S^{'}, S^{''}) = \theta_o;$$

The definitions of a neighborhood of $S^{'}$ by w or n are:

$$D_w(S^{'}, \theta_o) = \{S^{''} \mid (\theta_w(S^{'}, S^{''}) = \theta_o) \& (\theta_n(S^{'}, S^{''}) = 0)\},$$

$$D_n(S^{'}, \theta_o) = \{S^{''} \mid (\theta_w(S^{'}, S^{''}) = 0) \& (\theta_n(S^{'}, S^{''}) = \theta_o)\}.$$

In this way, $D(S, \theta_o)$ is said to be a θ_o-neighborhood of S. The neighborhoods of a set $\{S\}$ are: $D_w(\{S\}, \theta_o)$, $D_n(\{S\}, \theta_o)$, and $D(\{S\}, \theta_o)$. Also, we use the neighborhood of Pareto-effective layer Y as follows: $Y_w(\theta_o)$, $Y_n(\theta_o)$, and $Y(\theta_o)$.

Finally, by $Id(\theta_o)$ denote a neighborhood of an ideal point set Id. Thus we get the following layers of a system perfection: Id; $Id(1)$; Y; $Y(1)$.

Two main parameters may be considered to classify the bottlenecks of composite DA's: (a) a result of improving the bottleneck; and (b) a constructive type of the bottleneck.

The following kinds of improvement actions are examined:

(1) the generation of Id;

(2) the improvement of Y;

(3) the extension of Y, i.e., improving of elements of $Y(1)$; and

(4) the restriction of Y by building a new Pareto-effective point, that dominates by N some elements of Y.

In general, there are the following two cases: (a) $|Y| \neq 0$; and (b) $|Y| = 0$. In the last case, we can consider only a construction of an admissible composite decision as an extension of the Y. In this case, it is necessary to search for an inadmissible interconnection, the improvement of which enables to construct an admissible decision. Constructing an admissible decision(s) is based on examining all inadmissible interconnections. Now we examine the existence of Y. Two following constructive types of bottlenecks are:

1. *S-**aggravating*** elements of the points of Y. The improvement of the element enables to improve a point of Y. As a result, we can obtain a restriction of Y, when a new built point dominates by N some points of Y.

2. Corresponding elements of the neighborhood, e.g., $Y(1)$. The improvement of the bottlenecks allows to extend Y.

Let us consider algorithms to reveal bottlenecks:

Algorithm 3.

Step 1. To find all *S-**aggravating*** elements (components, interconnections) for the decisions of Y.

Step 2. To analyze each *S-**aggravating*** element.

Algorithm 4.

Step 1. To construct $Y(1)$.

Step 2. To analyze each element in $Y(1)$.

Generally, planning of the improvement process consists of the following:

(a) selection and combining of the improvement actions; and

(b) scheduling of the actions.

The first problem is similar to knapsack problem or morphological clique. Scheduling may be based on well-known models ([50], [85], [160], etc.).

Fig. 3.6 depicts a quality lattice, Pareto-effective points (#), neighbors, and two kinds of improvement actions.

In this section, we have described an analysis and improvement of the graph system. Note that close approach to analyze a graph, that corresponds to a team, has been proposed in ([201], [430]). In ([498]) this approach has been generalized for fuzzy graphs, when fuzzy estimates are used for compatibility.

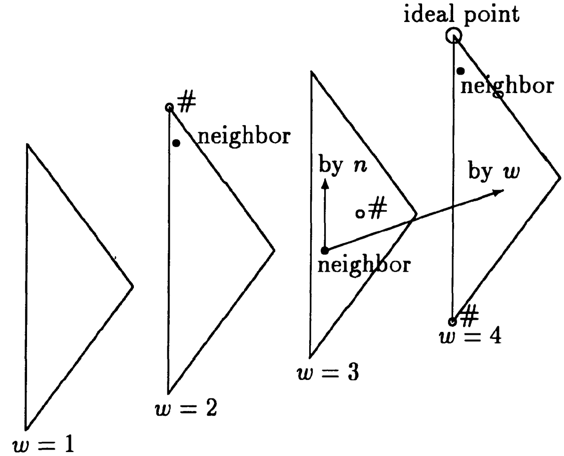

Fig. 3.6. Quality lattice, Pareto-effective points (#), neighbors, and improvements ($\longrightarrow$)

3.2.5 *Example of Solving and Improvement*

Finally, let us consider a numerical example. The structure of the designed system, and DA's are the following (priorities of DA's are shown in brackets):

(1) subsystem H: $H_1(2)$, $H_2(3)$, $H_3(1)$, $H_4(2)$;

(2) subsystem T: $T_1(1)$, $T_2(3)$; and

(3) subsystem U: $U_1(1)$, $U_2(1)$, $U_3(2)$.

Table 3.1 contains compatibility of DA's. Pareto-effective composite DA's are the following:

(a) $N = (4; 1, 1, 1)$: $S_1 = H_3 \star T_2 \star U_2$;

(b) $N = (3; 2, 1, 0)$: $S_2 = H_3 \star T_1 \star U_2$; $S_3 = H_3 \star T_1 \star U_3$; and

(c) $N = (2; 3, 0, 0)$: $S_4 = H_3 \star T_1 \star U_1$.

Points, corresponding these vectors (N), are indicated in Fig. 3.1 (Fig. 3.5) by symbol #. Some bottlenecks and improvements for these composite DA's are presented in Table 3.2.

Table 3.1. Compatibility of DA's

	T_1	T_2	T_3	U_1	U_2	U_3
H_1	3	3	3	3	2	4
H_2	3	4	3	4	4	3
H_3	4	4	4	2	4	3
H_4	2	4	2	3	4	4
$H_5 = H_1\&H_4$	2	3	2	3	2	4
$H_6 = H_2\&H_3$	3	4	3	2	4	3
T_1				2	3	4
T_2				1	4	3
$T_3 = T_1\&T_2$				1	3	3

Table 3.2. Bottlenecks and improvement actions

Composite DA's	Bottleneck		Action
	DA's	Ins	w/r
$S_1 = H_3 \star T_2 \star U_2$	T_2		$3 \Rightarrow 2$
$S_1 = H_3 \star T_2 \star U_2$	U_2		$2 \Rightarrow 1$
$S_2 = H_3 \star T_1 \star U_2$	T_2		$2 \Rightarrow 1$
$S_2 = H_3 \star T_1 \star U_2$		(T_1, U_2)	$3 \Rightarrow 4$
$S_3 = H_3 \star T_1 \star U_3$	U_3		$2 \Rightarrow 1$
$S_3 = H_3 \star T_1 \star U_3$		(H_3, U_3)	$3 \Rightarrow 4$

Now let us include into our example a redundancy by the following additional aggregate DA's (with the 1st priorities): $H_5 = H_1\&H_4$; $H_6 = H_2\&H_3$; $T_3 = T_1\&T_2$. Compatibility of these aggregate DA's equals the minimum of corresponding compatibility estimates of included elements (see also Table 3.1). Additional Pareto-effective composite DA's are the following:

(a) $N = (4;1,1,1)$: $S_5 = H_6 \star T_2 \star U_2$;

(b) $N = (3;2,1,0)$: $S_6 = H_6 \star T_1 \star U_2$; $S_7 = H_6 \star T_1 \star U_3$; $S_8 = H_6 \star T_3 \star U_2$; $S_9 = H_6 \star T_3 \star U_3$; and

(c) $N = (2;3,0,0)$: $S_{10} = H_5 \star T_1 \star U_1$; $S_{11} = H_6 \star T_1 \star U_1$.

Fig. 3.7 demonstrates a concentric presentation of composite DA's.

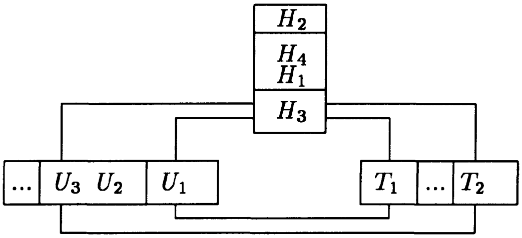

Fig. 3.7. Concentric presentation of composite DA's

3.3 TAXONOMY OF MODELS

Let $B = (\{a(j), j = 1, ..., m\}, V)$ be the condensed morphological graph describing a generalized interconnectivity: each vertex $a(j)$ corresponds to morphological class j in G; V is a set of arcs of the following kinds:

(1) $v^{\star}(i,j)$, if all pairs of elements of $A(i)$ and $A(j)$ are compatible;

(2) $v^{-}(i,j)$, if all pairs of elements of $A(i)$ and $A(j)$ are not compatible; and

(3) $v^{+}(i,j)$, if some (not all) pairs of elements of $A(i)$ and $A(j)$ are compatible.

Clearly, $V = \{v^{\star}\} \cup \{v^{+}\} \cup \{v^{-}\}$. Now we analyze the following:

(a) $V = \{v^{\star}\}, \quad | \{v^{+}\} | + | \{v^{-}\} | = 0$ (e.g., basic multiple-choice knapsack problem);

(b) $| \{v^{-}\} | > 0$ (admissible decisions do not exist); and

(c) $V = \{v^{\star}\} \cup \{v^{+}\}, \quad |\{v-\}| = 0$.

In the last case, we use graph $B^{+} = (\{b_j\}, \{v^{+}\})$, where vertices are connected by arcs of type v^{+}. Analogical morphological graph, and its components were introduced in section 2.2

Our description of a taxonomy for composition problems is the following:

$Problem := Result \mid Components \mid Compatibility \mid Restriction,$
$Result := t, a;$
$Components := num, a;$
$Compatibility := rel, a, tB^{+};$
$Restriction := ine, str;$

where type $t := S \mid F \mid S_r \mid M$; S corresponds to the best composite decision; F corresponds to a family of the best decisions; S_r corresponds to the best decision with redundancy, i.e., with a duplication of some component(s); M is a morphological decision; assessment $a := amd \mid N \mid ad \mid nl \mid rel \mid vec \mid b \mid$

$o \mid int$; adm corresponds to admissibility; $N = (w(S); n(S))$; ad corresponds to an additive function; nl corresponds to a nonlinear function; pr corresponds to a preference relation; vec corresponds to a vector description; b corresponds to a binary description; o corresponds to an ordinal description; int corresponds to a function with interval ordinal value; num corresponds to the number of morphological classes; rel corresponds to binary relation E_c, e.g., symmetric & reflexive ($sym\&ref$), asymmetric, etc.; $tB^+ := empty \mid chain \mid tree \mid G$ (general case of graph) or C (complete graph); ine corresponds to inequalities; str is structural restrictions.

The known examples are the following:

(a) basic morphological clique problem ([295], [296], [300], etc.):

$$S, N \mid m, o \mid sym\&ref, o, G \mid$$

(b) morphological analysis ([22], [254], [476], etc.):

$$S, adm \mid m, \mid sym\&ref, b, G \mid$$

(c) multiple-choice knapsack problem [347]:

$$S, a \mid m, a \mid sym\&ref, b, empty \mid ine$$

Let us consider some versions of morphological clique problem. First let us note that the solving scheme for the version $F, N \mid m, o \mid sym, o, ge \mid$ is based on the basic version above. Secondly we propose the use of an interval (fuzzy) scale for the assessment of DA's: $S_o, N \mid m, int \mid sym, o, ge \mid$. Here the preference relation by $n(S)$ will be a little more complex. This version is the basic one for the following:

$S_r, N \mid m, o \mid sym, o, ge \mid$, and

$M, N \mid m, o \mid sym, o, ge \mid$.

In this case, we consider an aggregated design alternative (DA), that includes several original ones. Interconnection of aggregated DA will be equal to the intersection of original interconnection of initial DA's. The priority of the aggregated DA will be a histogram on the basis of original priorities. Thirdly fast approximate algorithms exist for the problems as follows:

$S_o, ad \mid m, ad \mid sym, b, co \mid ine$ (multiple-choice knapsack problem [347]);

$S_o, ad \mid m, ad \mid sym, b, chain|ine$; and

$S_o, ad \mid m, ad \mid sym, b, tree \mid ine$.

3.4 ISSUES OF COMPLEXITY

Clearly considered combinatorial composition problems are very hard. Let us point out some basic factors of complexity as follows: (a) evaluation of

DA's; (b) evaluation of Ins; (c) integrated evaluation of composite decisions S (d) structural kind of Ins (e.g., graph B^+), and its properties; (e) approach to define a pair compatibility; (f) dependency of pair compatibility; and (g) additional constraints and requirements.

Note that a role of item compatibility in combinatorial problems is increasing. Some simple polynomial solvable cases of the problem of compatible representatives (e.g., with transitive Ins) have been studied in [254]. Approximation algorithms have been proposed for a close problem in [475]. Approximation schemes are prospective ones, and our attempt to design an approximation algorithm is presented in the next section. Contemporary problems in approximation approaches for combinatorial models have been studied in ([539], [540]). Other ways are based on heuristics ([28], etc.).

3.5 ISSUES OF APPROXIMATION

Approximation approaches for optimization problems are based on approximating the following: (a) goal function(s); (b) restrictions.

For example, similar techniques are applied for knapsack problem ([347], [289], etc.). In the case of morphological clique problem, we can approximate the following two main elements of composite decision S: (i) quality of DA's; and (ii) quality of Ins.

Here we examine an approximation of the $2nd$ element by two ways as follows:

(1) approximation of initial structural requirements that are mapped into compatibility structure B^+; and

(2) weakening the requirements to pair compatibility of DA's of different morphological classes.

The $1st$ way above may provide polynomial solvability of morphological clique problem because we can use simple structural approximations (e.g., *chains*, *trees*). In the $2nd$ case, we can consider a decision (quasi morphological clique problem):

$$S^a = S^a(1), ..., S^a(i), ..., S^a(m),$$

where $S^a(i)$ is a set of items of morphological class i. Let

$$K(S^a) = (w(S^a); k(S^a))$$

be a vector-like system excellence for S^a, where $w(S^a)$ is the minimum of compatibility between each item of $S^a(i)$ and an item for each other morphological class, $k(S^a) = (k(1, ..., k(r), ..., k(l))$, where $k(r)$ is the number of components of the rth quality in S^a (a histogram). Here it is possible to reject an item j, if there exists a morphological class that does not contain an item which is connected with item j with compatibility weight more or equal to $w(S^a)$. Evidently, that some other versions of rejection rules may be applied too. An example of quasi morphological clique ($m = 4$) and $w(S^a) = 2$) is presented in Fig. 3.8.

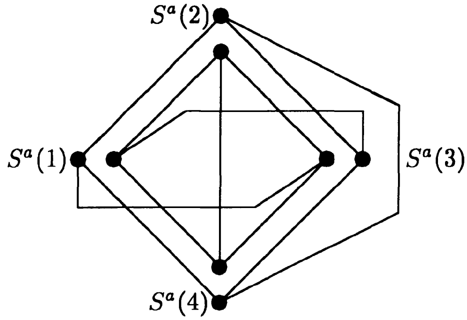

Fig. 3.8. Example of quasi morphological clique

In addition, we can describe a relationship of S and S^a:

Lemma 3.2.
1. Each S is contained into corresponding S^a.
2. There exists situations, when S^a does not include corresponding S.

The proof of the $1st$ part is trivial because each S is S^a. The $2nd$ part may be proved by example (see numerical example 5). This lemma 3.2 has a significant corollary, because each existing S is contained in corresponding S^a. Thus we can search for S on the basis of a preliminary construction of corresponding S^a.

Let us examine a numerical example for a 5-component system ([304], [312]):
1. H: $H_1(2)$, $H_2(1)$, $H_3(3)$, $H_4(1)$.
2. T: $T_1(2)$, $T_2(3)$, $T_3(1)$, $T_4(1)$.
3. I: $I_1(1)$, $I_2(3)$, $I_3(1)$, $I_4(2)$.
4. J: $J_1(1)$, $J_2(3)$, $J_3(2)$, $J_4(1)$.
5. F: $F_1(2)$, $F_2(1)$, $F_3(2)$, $F_4(3)$, $F_5(1)$.
Compatibility is presented in Tables 3.3, 3.4. Thus we compute composite decisions for the following cases:

Example 1. Basic morphological clique problem:
$S_1 = H_1 * T_1 * I_2 * J_2 * F_4$, $N(S_1) = (2; 0, 2, 3)$; $S_2 = H_3 * T_3 * I_4 * J_3 * F_1$, $N(S_2) = (1; 1, 3, 1)$.

Example 2. Morphological clique problem with the approximation of structural requirements (*chain*: $I - T - H - J - F$):

$S_3^a = H_1 * T_1 * I_3 * J_3 * F_3$, $N(S_3^a) = (3; 1, 4, 0)$.

Example 3. Morphological clique problem with the approximation of structural requirements (*tree*: $T - H - J - F$, and $J - I$):

$S_4^a = H_1 * T_2 * I_2 * J_3 * F_1$, $N(S_4^a) = (4; 0, 3, 2)$.

Example 4. Morphological clique problem with redundancy. Here we use additional pairs of items as follows: $H_5 = H_1\&H_2(1)$, $T_5 = T_1\&T_3(1)$, $J_5 = J_1\&J_3(1)$, $F_6 = F_1\&F_5(1)$, and assume that attributes (including compatibility) of composite items are an aggregation of attributes of initial items. In our example, the priority and compatibility of a composite item are equal to the best initial ones (Table 3.4). As a result we obtain the following:

$S_5^a = H_5 * T_5 * I_3 * J_5 * F_6 = (H_1\&H_2) * (T_1\&T_3) * I_3 * (J_1\&J_3) * (F_1\&F_5)$, $N(S_5^a) = (4; 5, 0, 0)$.

Example 5. Morphological clique problem with weakening the requirements to pair compatibility of items:

$S_6^a = \{H_2, H_4\} * \{T_3, T_4\} * \{I_1, I_3\} * \{J_1, J_4\} * \{F_2, F_5\}$, $N(S_6^a) = (5; 10, 0, 0)$.

Note that similar approximation approach may be useful for information design ([299], [312]).

Table 3.3. Compatibility of DA's

	T_1	T_2	T_3	T_4	I_1	I_2	I_3	I_4	J_1	J_2	J_3	J_4	F_1	F_2	F_3	F_4	F_5
H_1	4	0	0	0	0	2	0	0	0	2	4	0	0	0	0	2	0
H_2	0	0	0	5	0	0	5	0	0	0	0	5	0	0	0	0	5
H_3	0	0	1	0	0	0	0	2	0	0	1	0	1	0	0	0	0
H_4	0	0	5	0	5	0	0	0	5	0	0	0	0	5	0	0	0
T_1					0	2	3	0	0	3	0	0	0	0	0	2	0
T_2					0	0	0	0	0	0	0	0	0	0	0	0	0
T_3					0	0	5	1	0	0	4	5	3	0	0	0	5
T_4					5	0	0	0	5	0	0	0	0	5	0	0	0
I_1									0	0	0	5	0	0	0	0	5
I_2									0	3	4	0	0	0	0	2	0
I_3									5	0	0	0	0	5	0	0	4
I_4									0	0	1	0	1	0	0	0	0
J_1													0	0	0	0	5
J_2													0	0	0	0	0
J_3													4	0	3	0	0
J_4													0	5	0	0	0

Table 3.4. Compatibility of DA's

	T_5	I_3	J_5	F_6
H_5	4	5	4	5
T_5		5	4	5
I_3			5	4
J_5				5

3.6 SERIES-PARALLEL MULTI-PERIOD STRATEGY

3.6.1 Operational Environment

Multi-period planning/decision making has been used in economical planning, in production planning and scheduling, in problem solving, in model management, in method engineering, etc. ([62], [106], [141], [171], [298], [375], [462], [542], etc.). Note that in recent years there exists a trend from software libraries to problem-solving environment *PSE* (i.e., advanced solution methods, automatic or semiautomatic selection of solution methods, and ways to incorporate novel solution methods) ([49], [62], [157], [418], etc.). As a result, the significance of model management and method engineering is increasing. Fig. 3.9 depicts two examples of series-parallel strategies for a multi-period problem.

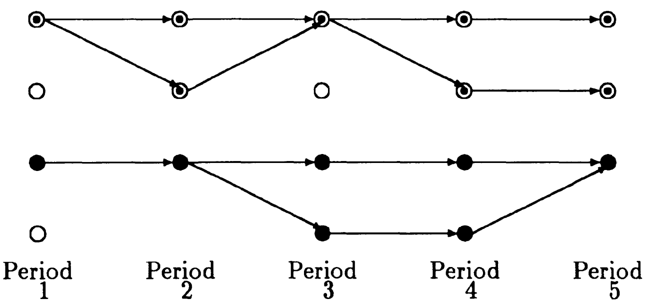

Fig. 3.9. Examples of multi-period series-parallel solving strategies

Note series-parallel processes correspond to tree-decomposable graphs, i.e., it is possible to decompose an initial graph on the basis of a derivation tree. Other tree-decomposable graphs (e.g., trees, partial k-trees, etc.) have been studied in ([58], etc.). In our case, we apply a structure which is similar to *AND-OR* tree.

In this section, we examine the synthesis of the three-period series-parallel solving strategy for multicriteria ranking [304]. Another similar example is contained in [298].

Usually planning of decision making processes is based on the selection, integration, and sequencing of models of a model base ([171], [375], [462], etc.). Note close procedures often are based on the following: (i) various heuristics [106]; and (ii) special model knowledge including descriptions of basic submodels, their connections ([375], etc.). For example, issues of the method selection on the basis of an expert system have been studied in [385].

On the other hand, a two-stage series-parallel procedure for choice problem has been proposed by C.R. Plott [398]. In this case, the problem of choice or ***set-to-set*** transformation is considered, where a set of admissible alternatives is transformed into the set of chosen (best) alternatives. Plott has suggested ***path independence*** for series-parallel data processing. The condition requires the following: results of data processing (two-stage or more general multi-stage) would not depend on the form of the procedure, but should only depend on preference relations and on the set of alternatives. ***Path independence*** in series-parallel data processing has been studied by many researchers ([237], [342], [343], etc.).

In this section, we consider composition problem for a solving scheme on the basis of HMMD, including series and/or parallel composing of the composite DA's for data processing (articulation of preferences) with taking into account Ins among DA's. In this application, Ins correspond to compatibility of series DA's, and independence of parallel DA's.

Usually information part of DSS involves the following:

(1) data (alternatives or basic items, criteria, multicriteria estimates of alternatives upon criteria, preference relations);

(2) tools for maintaining data (DBMS and interfaces with commercial DBMSs);

(3) support information for learning (e.g., a helper, etc.).

Here the following information (a preference relation or a matrix) $R_j (j = 0, ...3)$ is examined ([294], [317]):

(1) basic data as alternatives (items), criteria, multicriteria estimates of alternatives upon criteria (R_0);

(2) a preference relation of alternatives (R_1);

(3) an intermediate linear ordering of alternatives (R_2); and

(4) a ranking of alternatives (R_3), when an ordinal priority for each alternative is defined.

We assume the following kinds of basic operations: (i) data processing (series and/or parallel); (ii) data aggregation; (iii) parallelism of solving process on the basis of considering the different elements (alternatives, criteria, techniques, experts).

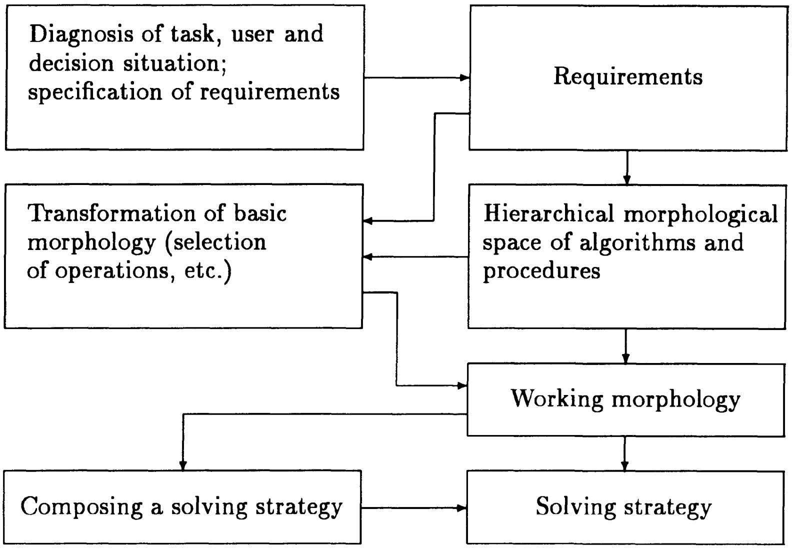

Fig. 3.10. Design scheme for solving strategy

The solving process may be presented as a hierarchy with the following levels ([35], [294], [317]):

(1) algorithms and man-machine interactive procedures for data transformation;

(2) strategies (*step-by-step* schemes of data transformation, particularly series-parallel ones);

(3) scenarios (complexes of strategies with their analysis and feedback).

We use two problems for the design of strategies:

(a) composing a series data processing (morphological clique);

(b) composing a parallel data processing (maximal clique with weighted).

Our basic scheme (a morphology) of data processing (a composite solving strategy) is the following (three periods):

$$R_0 \Longrightarrow R_1 \Longrightarrow R_2 \Longrightarrow R_3$$

or $S = H \star T \star U$, where H corresponds to forming R_1 (an algorithm or a procedure); T corresponds to forming R_2; U corresponds to forming R_3. Fig. 3.10 depicts a process of designing a solving strategy for multicriteria ranking.

A set of examined DA's (algorithms and procedures) and their estimates are presented in Tables 3.5, and 3.6. The ordinal scale [0...5] is applied for each criterion.

Table 3.5. Criteria

Criteria	Weight		
	H	T	U
1.Required computer resources	−2	−1	−1
2.Required human resources	−4	−3	−4
3.Quality of ranking (robustness, etc.)	5	5	5
4.Possibility for data representation	4	4	4
5.Possibility for an analysis of intermediate data	4	2	3
6.Usability (easy to learn, understanding, etc.)	5	5	5

Table 3.6. DA's and estimates

DA's	Description	Criteria						r
		1	2	3	4	5	6	
H_1	Pairwise comparison	4	5	4	4	5	3	2
H_2	ELECTRE-like technique	3	3	4	4	5	3	1
H_3	Additive utility function	2	3	2	3	1	4	3
H_4	Expert stratification	5	5	4	4	4	3	3
T_1	Line elements sum of preference matrix	2	0	2	2	2	4	3
T_2	Additive utility function	2	3	2	3	1	4	3
T_3	Series revealing of maximal elements	3	0	4	4	4	3	1
T_4	Series revealing of Pareto elements	3	0	4	4	4	3	1
T_5	Expert stratification	5	5	4	4	4	3	2
U_1	Series revealing of maximal elements	3	0	4	4	4	3	1
U_2	Series revealing of Pareto elements	3	0	4	4	4	3	1
U_3	Dividing of linear ranking	4	3	3	2	3	3	3
U_4	Expert stratification	5	5	4	4	4	3	2

3.6.2 Design of Series Strategies

Now let us construct a series solving strategy on the basis of the morphological clique problem. First we rank DA's. DSS COMBI is used for multicriteria ranking in our example here and afterhere. However, results of other techniques will be similar. Resultant priorities of DA's are shown in Table 3.6. Table 3.7

contains compatibility of DA's. Finally we combine composite Pareto-effective DA's which are the following:

(a) $S_1 = H_4 * T_5 * U_4$, $N(S_1) = (5; 0, 2, 1)$;
(b) $S_2 = H_2 * T_1 * U_3$, $N(S_2) = (5; 1, 0, 2)$;
(c) $S_3 = H_2 * T_3 * U_1$, $N(S_3) = (4; 3, 0, 0)$.

Table 3.7. Compatibility of DA's

	T_1	T_2	T_3	T_4	T_5	U_1	U_2	U_3	U_4
H_1	4	0	3	4	3	3	4	4	3
H_2	5	0	4	3	3	4	3	5	3
H_3	0	5	0	0	0	0	0	4	0
H_4	0	0	0	0	5	0	0	0	5
T_1						0	0	5	0
T_2						0	0	4	0
T_3						5	0	0	0
T_4						0	5	0	0
T_5						0	0	0	5

3.6.3 Design of Parallel Strategies

We examine the following DA's to design parallel solving strategies:

(1) H_1 (we assume that there are 7 different experts; the 2rd index is used to point out a parallel version; an analogical situation we examine for H_4, T_5, and U_4);

(2) H_2 (4 modifications of technique, e.g., on the basis of the use different thresholds in an outranking technique); and

(3) 3 series strategies S_1, S_2, and S_3.

Tables 3.8, 3.9, 3.10 contain estimates of an independence for DA's above. Resultant composite parallel DA's, found on the basis of maximal clique with ordinal weighted Ins, are presented in Table 3.11.

Table 3.8. Independence of DA's

	H_{11}	H_{12}	H_{13}	H_{14}	H_{15}	H_{16}
H_{11}	.	0	4	1	4	1
H_{12}	0	.	3	4	2	3
H_{13}	4	3	.	3	4	3
H_{14}	1	4	3	.	2	4
H_{15}	4	2	4	2	.	2
H_{16}	1	3	3	4	2	.

Table 3.9. Independence of DA's

	H_{21}	H_{22}	H_{23}	H_{24}	H_{25}
H_{21}	.	1	2	3	4
H_{22}	1	.	1	2	3
H_{23}	2	1	.	1	2
H_{24}	3	2	1	.	1
H_{25}	4	3	2	1	.

Table 3.10. Independence of DA's

	S_1	S_2	S_3
S_1	.	3	3
S_2	3	.	2
S_3	3	3	.

3.6.4 Resultant Composite Series-parallel Strategies

Finally we combine resultant strategies. Our design morphology is depicted in Fig. 3.11. We delete DA's which are covered by some others (e.g., $S_1\&S_3$ is covered by $S_1\&S_2\&S_3$). It is assumed that properties (estimates, compatibility, except required resources) of parallel DA's correspond to properties of basic ones. Resultant series-parallel solving strategies are demonstrates in Fig. 3.12. Parallel DA's are included into corresponding series strategies.

Table 3.11. Composite parallel DA's

Basic DA	$(w; M)$	Parallel DA's
1.H_1	(4;3)	$H_{11}\&H_{13}\&H_{15}$
	(3;4)	$H_{12}\&H_{13}\&H_{14}\&H_{16}$
	(2;5)	$H_{12}\&H_{13}\&H_{14}\&H_{15}\&H_{16}$
2.$H_4(T_5, U_4)$	(4;3)	$H_{41}\&H_{43}\&H_{45}$
	(3;4)	$H_{42}\&H_{43}\&H_{44}\&H_{46}$
	(2;5)	$H_{42}\&H_{43}\&H_{44}\&H_{45}\&H_{46}$
3.H_2	(4;2)	$H_{21}\&H_{25}$
	(2;3)	$H_{21}\&H_{23}\&H_{25}$
4.S	(3;2)	$S_1\&S_2$, $S_1\&S_3$
	(2;3)	$S_1\&S_2\&S_3$

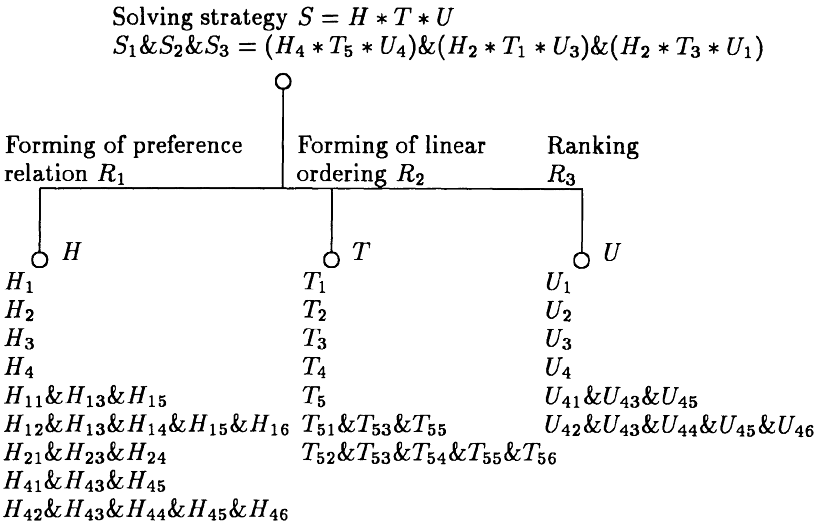

Fig. 3.11. Resultant design morphology

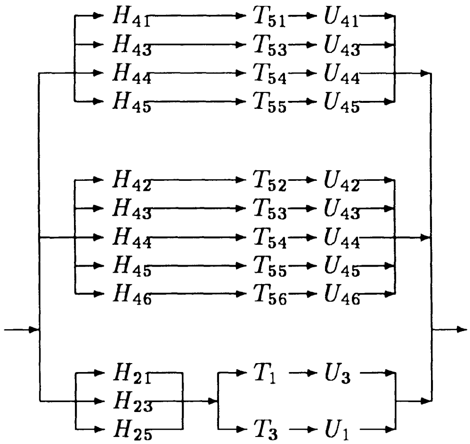

Fig. 3.12. Series-parallel solving strategies

3.7 NOTE ON CONSTRAINT SATISFACTION PROBLEMS

In recent years, Constraint Satisfaction Problems (CSPs) approach has been applied for many practical problems, e.g., parametric design, resource allocation, planning of genetics experiments, machine vision, graph problems, scheduling ([50], [328], [335], etc.). In our opinion, CSPs approach is close to morphological analysis. CSPs are formulated as follows:

(a) a set of domains $D = \{D_1, ..., D_j, ..., D_m\}$, for example, $D_j \in R^n$;

(b) a set of variables $X = \{x_1, ..., x_j, ..., x_m\}$, $x_j \in d_j \ \forall j$; and

(c) a set of constraints $C = \{c_1, ..., c_l, ..., c_k\}$,

$\forall c_l \subseteq \Delta_l = \{\delta_1^l \times ... \times \delta_j^l \times ... \times \delta_m^l\}$, where $\forall l \ \delta_j^l \subseteq D_j$, Δ_l is a Cartesian product.

A solution x^* to a CSP is an assignment of values for all variables

$$x^* = \{x_1^*, ..., x_j^*, ..., x_m^*\}, \quad \forall j \ \ x_j \in D_j$$

$$s.t. \quad c_l, \quad l = 1, ..., k.$$

In numerical CSPs, constraints can be presented by numerical relations between quantitative variables too. Fig. 3.13 illustrates CSPs.

In CSPs it is necessary to find a feasible decision or to show that for a given constraint set no such solution exists. *Conflict resolution* techniques are used for the latter case. CSPs correspond to an NP-complete problem (like morphological analysis) ([160], etc.).

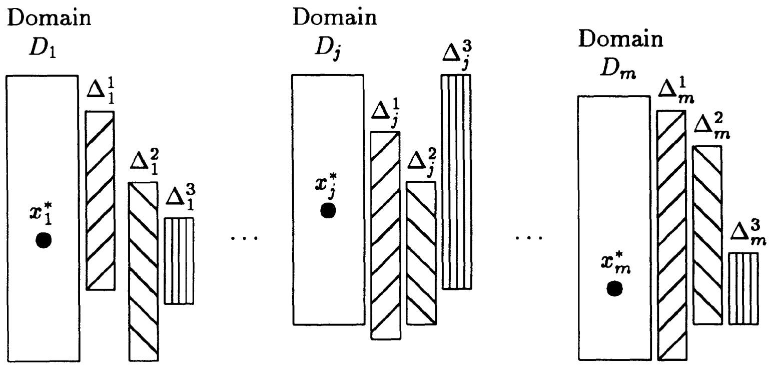

Fig. 3.13. Illustration to Constraint Satisfaction Problem

Clearly, that CSPs are similar to morphological analysis. On the other hand, our morphological approach involves weighted priorities of DA's, and Ins. Thus it is possible to examine in CSPs some additions as follows:

(i) weighted preference relation on constrains;

(ii) a preferences on values of variables.

Traditionally, the following solving algorithms are used for CSPs:

(1) backtracking and networks consistency algorithms ([109], [335], [377], etc.);

(2) heuristic revision [359];

(3) decomposition ([110], etc.);

(4) distributed constraint satisfaction search or multiagent approach ([219], [328], [334], [493], etc.).

In the last case, the following algorithms are classified and discussed in [334]:

1. Distribuited-agent-based Strategies and Algorithms:

1.1. A Root Agent Algorithm.

1.2. A Distributed Hill Climbing Method.

1.3. Depth First with A Self-stabilising Protocol.

1.4. Multistage Negotiation.

1.5. An Asynchronous Backtracking Method.

1.6. Distributed Extended Forward Checking while Conflict-directed Backjumping.

1.7. A Hybrid Search Algorithm.

2. Parallel-agent-based Strategies and Algorithms:

2.1. Parallel Searching Separated Spaces.

2.2. Parallel Search Separated Spaces while Sharing Information.

2.3. Dynamic Load Balancing by way of Dead-Ends.

3. Function-agent-based Strategies and Algorithms.

3.8 SUMMARY

In this chapter, we have proposed our morphological approach on the basis of morphological clique problem, and close issues. In our opinion, structural approaches to approximation may be very important for many combinatorial problems on graphs. The design of series-parallel multi-period strategies is the second basic application of our morphological approach (i.e., for planning). Note relations of our morphological approach and well-known Constraint Satisfaction Problems approach require additional investigations.

4 COMPARISON OF DECOMPOSABLE SYSTEMS

This chapter focuses on the comparison of decomposable systems and their parts ([297], [305], etc.). Mainly, these issues are only under consideration. In addition, we describe vector-like proximity for rankings, an aggregation problem for ranking ([35], [291], [305]). Also, examples of comparison for sets and trees are presented.

4.1 BASIC APPROACHES

In our opinion, the following two measurement problems may be considered as basic ones:

1. Measurement of an excellence of an object (system).

2. Measurement of a distance (similarity, closeness, dissimilarity, etc.) for combinatorial objects.

These problems are used in many domains and for many other complicated problems, e.g., classification or clustering of objects, aggregation of objects or computation of consensus, etc. In this section, we examine issues of the analysis and comparison of decomposable systems.

Many authors have been investigated processes of the analysis of complex systems on the basis of a system parameter space, where system versions corresponds to points ([66], [113], [127], [387], [440], [477], etc.). For example, Fig. 4.1 illustrates complex problems of system evolution, including system modification, and formation of new kind of systems ([66], etc.).

Usually the following basic system analytical problems have been examined to analyze complex systems ([22], [66], [113], [387], [218], [440], [519], [477], [441], etc.):

(1) to compare two system versions;

(2) to classify system versions;

(3) to evaluate a system version set (e.g., aggregation, construction of consensus, etc.);

(4) to analyze tendencies of system changes (evolution, etc.);

(5) to reveal the most significant system parameters; and

(6) to plan an improvement process for a system.

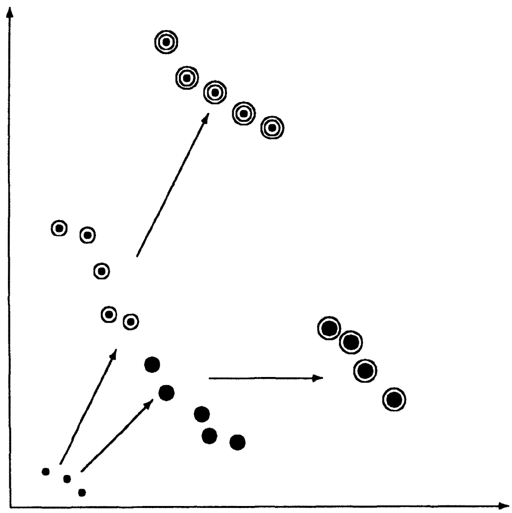

Fig. 4.1. System versions in a parameter space

Traditionally statistical and classification approaches have been used to analyze complex systems in parameter spaces. But, as A. Tversky has pointed out, the geometrical approach is not always the adequate one [508]. Here we examine decomposable systems and use system representations on the basis of structural or combinatorial objects. We assume that a system is a decomposable one, and can have several versions.

Thus the following kinds of problems are basic ones:

(a) description of a system and its parts; and

(b) operations for analysis and transformation of a system.

We take into account the following interconnected hierarchies for decomposable systems: (1) a system tree-like model; (2) requirements (criteria, restrictions) to system components (nodes of the model); (3) design alternatives (DA's) for nodes; (4) interconnection (Ins) or compatibility between DA's of different components; and (5) factors of compatibility.

The system proximity can be examined as the following structure, corresponding to the system model:

(a) proximity of hierarchical tree-like models;

(b) proximity of requirement hierarchies (criterion hierarchies; restriction hierarchies; compatibility factors hierarchies);

(c) proximity of DA's (sets of DA's, estimates on criteria, and priorities); and

(d) proximity of Ins (set of Ins with priorities).

We consider three levels of combinatorial descriptions: (1) basic combinatorial objects (points in a space; vectors; sets; partitions; rankings, strings; trees; posets; etc.); (2) elements of the system description: leaf nodes; set of DA's and/or Ins; tree-like system model; criteria for DA's; etc.; and (3) basic system descriptions, e.g., complete description; external requirement.

Note that the measurement theory, and the representational theory of measurement have been presented in ([264], [394], [424]), and [426], respectively.

4.2 MEASUREMENT OF PROXIMITY

First let us consider approaches to modeling a proximity (distance, similarity, closeness, dissimilarity, etc.) for combinatorial objects. Note that these investigations have been executed in various disciplines (e.g., statistics, mathematical psychology; decision making; chemistry; linguistics; morphological schemes of system in technological forecasting; biology; genetics; data and knowledge engineering; network engineering; architecture; and combinatorics). A survey of coefficients for measures of similarity, dissimilarity, and distance from viewpoint of statistical sciences is presented in [182].

From a system viewpoint, it is reasonable to examine some functions, operations, and corresponding requirements to mathematical models of the proximity (Table 4.1).

Formal requirements to proximity models are based on three Freshe's axioms specifying metrics, and sometimes on additional axioms ([243], etc.). In some cases, the triangulation axiom is rejected, e.g., for architectural objects [549], for rankings [35]. Measuring the proximity between combinatorial objects is based on the following approaches:

(1) a metric in a parameter space;

(2) attributes of the largest common part of objects (intersection) or an unification (the minimal covering construction); and

(3) minimum of changes (change path), which allows to transform an initial object into a target one.

Secondly let us consider scales of measuring. Traditionally R_1, $[0,1]$ or an ordinal scale are applied. Hubert and Arabie use measures of agreement or consensus indices, e.g., in $[-1,1]$ [216]. Recently some extensions of metric spaces have been proposed ([30], [400]), for examples:

(a) graphs and ordered sets [225];

(b) conceptual lattices for complex scaling [158];

(c) ordered sets and semilattices for partitions [31]; and

(d) simplices for rankings [35].

Generally, Arabie and Hubert have examined three approaches to compare combinatorial objects (sequences, partitions, trees, graphs) through given matrices [21]: (i) axiomatic approach to construct "good" measures; (ii) usage of structural representations; and (iii) usage of an optimization task.

Finally, the following approaches may be used in complicated cases:

(a) multidimensional scaling ([267], [504], etc.);

(b) usage of graphs and ordered sets as a kind of a metric space ([30], [31], [158], [225]); and

(c) integration or composition of a global proximity on the basis of distances or proximities for system components.

4.3 COMBINATORIAL OBJECTS AND SYSTEM'S DESCRIPTION

Main approaches to compare combinatorial objects are the following ([21], [400], etc.):

1. Points in a space: traditional metrics.

2. Sets, systems of representatives: metrics ([345], etc.).

3. Partitions: metrics ([364], [362], etc.), multidimensional scaling [20], measures of agreement or consensus indices [216], ordered sets, semilattices [31].

4. Linear (ordinal) rankings: metrics ([89], [243], [244], [362]).

5. Strings: maximum subsequence ([523], [455]), minimum supersequence ([501] etc.), metrics [198].

6. Linear rankings with values (set of numbers): metrics [431], a distance as the minimum cost transformation ([327], etc.).

7. Group rankings (elements with ordinal priorities): metrics ([89], [244], [362]), vector-like proximity [35].

8. Sets of strings: measures in [0, 1] [284].

9. Trees: metrics ([57], [345], [427]), distance as the minimum cost transformation, and the largest common substructure ([160]), [497]), proximity ([30], etc.).

10. Trees with labeled leafs: metrics ([107], [207]).

11. Hierarchies: metrics [59].

12. Graphs and posets: metrics ([55], [96], [411], etc.), structural representations of proximity [21].

Table 4.1. Some operations and requirements to mathematical models

Functional phase	Operations	Requirements
Development	Selection Design	Problem relevantness Completeness Universality Easiness Generalizability Habitulness
Representation	Mathematical description Text description Graphical presentation Animation Composite approach	Easy to visualize Understandability Operationability Relevantness to use
Study and learn	Understanding Remembering Evaluation of features Identification Processing	Ability to process/analyze Habitualness Coordination with intuition Understandability Simpleness Easy to visualize
Utilization (processing)	Processing Transformation Composing Coordination	Easy to process (by human, by computer) Composability

Distances in graphs (mainly as the shortest path) are described in [24]. Aggregation problems for combinatorial objects are studied in ([21], [31], [76], [107], [181], [216], [243], [327], [362], etc.).

Finally, a relationship of basic elements of our system model and combinatorial objects above is the following:

(1) points in a space: leaf nodes, DA's, Ins, priorities of DA's, estimates of Ins, requirements;
(2) sets: leaf nodes, DA's, Ins, requirements;
(3) partitions: requirements;
(4) ordinal rankings: DA's, Ins, requirements;
(5) strings: DA's, Ins;
(6) linear rankings: DA's, Ins, requirements;
(7) sets of strings: DA's, requirements;
(8) trees: system model, requirements; and
(9) posets: estimates of DA's and Ins, requirements.

Clearly it is reasonable to apply typical system descriptions of decomposable systems as follows: (i) external requirements: criteria, factors, restrictions; (ii) system model: structural (tree-like) model, leaf nodes (components), DA's; (iii) extended system model: structural (tree-like) model, leaf nodes (components), DA's, Ins, priorities of DA's, ordinal estimates of Ins.

4.4 VECTOR-LIKE PROXIMITY

4.4.1 *Preliminaries*

Let $G = (A, E)$ be a digraph, where $A = \{1, ..., i, ..., n\}$ is a set of vertices (i.e., objects, discrete information units), E is a set of arcs corresponding to a preference. By the above we may examine the following kinds of the digraphs:
(1) tree (denoted as T);
(2) parallel-series graph (P);
(3) acyclic graph or partial ordering (R);
(4) chain or linear ordering (L);
(5) layered structure S (group ordering, ranking or stratification), in which the set A is divided into m subsets (layers) without intersections as follows:
(a) $A(l)$, $l = 1, ..., m$, and $\forall l_1, l_2$, $l_1 \neq l_2$, $|A(l_1) \& A(l_2)| = 0$,
(b) if $l_1 < l_2$ then $i_1 \succ i_2$ $\forall i_1 \in A(l_1)$ and $\forall i_2 \in A(l_2)$;
(6) fuzzy layered structure S_f allowing any object to belong to the group of successive layers: the number or the interval of layers, which the ith object belongs to, is defined $\forall i \in A$: π_i or $d_i = [d_i^1, d_i^2]$, $1 \leq d_i^1 \leq d_i^2 \leq m$ and $\pi_i = d_i^1$ if $d_i^1 = d_i^2$.

Thus the system of intervals $\{d_i\}$ is specified. By the analogy of definitions above it is possible to specify clusters, and fuzzy clusters. Sometimes the comparison of structures representing the union of similar graphs (e.g. 'chains'-NL, layered structures - NS) has a particular interest in practice. An example of the comparison for two layered structures is demonstrated in Fig. 4.2.

$$A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

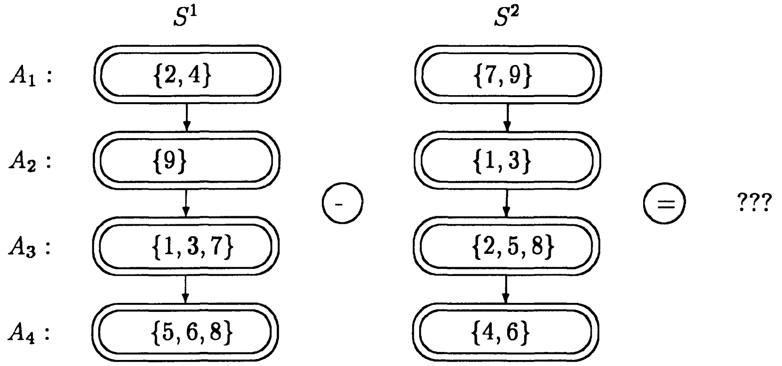

Fig. 4.2. Comparison of rankings

Usually the proximity measures have been used as scalar functions, which satisfy to Freshe's axioms for metrics (pseudo-metrics). Kendall's proximity measure is the most widely used [244]. Let $\|g_{ij}\|$, $(i, j \in A)$ be an adjacency matrix for graph G:

$$g_{ij} = \begin{cases} 1, & \text{if } i \succ j, \\ 0, & \text{if } i \sim j, \\ -1, & \text{if } i \prec j. \end{cases}$$

Then Kendall proximity metric for graphs G^1 and G^2 is the following:

$$\rho_K(G^1, G^2) = \sum_{i<j} \mid g_{ij}^1 - g_{ij}^2 \mid,$$

where g_{ij}^1, g_{ij}^2 are elements of adjacency matrices of graphs G^1 and G^2 accordingly. Often this metric has been used for rankings (linear and layered structures). Further, we describe a vector-like proximity for rankings ([291], [35]). In this case, a measurement scale is a simplex in which for a set of measured objects a poset may be constructed.

4.4.2 Definitions

Let $\Theta(S)$ be a set of all layered structures on A.

Definition 4.1. We say that

$$\delta_\pi(i, S, Q) = \pi(i, S) - \pi(i, Q),$$

$$\delta_\pi(i,j,S,Q) = \pi(i,S) - \pi(j,S) - (\pi(i,Q) - \pi(j,Q)),$$

where $\pi(i,S) = l \ \ \forall i \in A(l)$ in S, are the first order error $\forall i \in A$, and the second order error $\forall (i,j) \in \{A * A | i \neq j \ \forall S, Q \in \Theta(S)\}$ respectively.

Thus we get an integer-valued scale with the following ranges for an estimate of a discordance between the structures $S, Q \in \Theta(S)$ with respect to i and (i,j): $-(m-1) \le r \le m-1$ for $\delta_i^\pi(S,Q)$, and $-2(m-1) \le r \le 2(m-1)$ for $\delta_{i,j}^\pi(S,Q)$.

Definition 4.2. Let

$$x(S,Q) = (x^{-(m-1)}, ..., x^{-1}, x^1, ..., x^{m-1}), \tag{4.1}$$

$$y(S,Q) = (y^{-2(m-1)}, ..., y^{-1}, y^1, ..., y^{2(m-1)}), \tag{4.2}$$

be vectors of an error (proximity) $\forall S, Q \in \Theta(S)$ with respect to components i (the 1st order), and the pairs (i,j) (the 2nd order). The components of the above-mentioned vectors are defined as follows:

$$x^r = |\{i \in A | \delta_i^\pi(S,Q) = r\}|/n,$$

$$y^r = 2|\{(i,j) \in \{A * A | i \neq j\} | \delta_{ij}^\pi(S,Q) = r\}|/(n(n-1)).$$

It may be reasonable to define similar vectors of the higher order also. Moreover, it is possible to examine the weighted errors of the first and the second orders while taking into account the dependence on corresponding number l of layer $A(l)$ for the definition of vector components. Now let us denote a set of arguments for the components of vectors x and y as follows: $\Omega = \{-k, ..., k\}$, negative values as Ω^-, and positive ones as Ω^+. In addition, we will use the vectors x with aggregate components of the following type (similarly, for y):

$$x^{k_1;k_2} = \sum_{r=k_1}^{k_2} x^r,$$

$$x^{\le -k} = \sum_{r=-(m-1)}^{-k} x^r,$$

$$x^{\ge k} = \sum_{r=k}^{m+1} x^r, k > 0,$$

$$x^{|r|} = x^r + x^{-r}.$$

Definition 4.3. Let $|x(S,Q)| = \sum_{r\in\Omega} x^r$, $|y(S,Q)| = \sum_{r\in\Omega} y^r$ be modules of the vectors.

Afterhere we will consider vector x as a basic one.

Definition 4.4. We will call vectors truncated ones if

(1) the part of terminal components is rejected, e.g.

$$x(S,Q) = (x^{-k_1}, x^{-(k_1-1)}, ..., x^{-1}, x^1, ..., x^{k_2-1}, x^{k_2}),$$

and one or both of following conditions are satisfied: $k_1 < m-1$, $k_2 < m-1$;

(2) aggregate components are used as follows:

$$x(S,Q) = (x^{\leq k_1}, ..., x^{k_a-1}, x^{k_a,k_b}, x^{k_b+1}, ..., x^{\geq k_2}),$$

$$x(S,Q) = (x^{|1|}, ..., x^{|r|}, ..., x^{|k|}). \tag{4.3}$$

Definition 4.5. Let us call vector x (y): (a) the two-side one if $|\Omega^+| \neq 0$ and $|\Omega^-| = 0$; (b) the one-side one if $|\Omega^+| = 0$ or $|\Omega^-| = 0$; (c) the symmetrical one if $-r \in \Omega^-$ exists $\forall r \in \Omega^+$ and vice versa; (d) the modular one if it is defined with respect of definition 4.4 (4.3).

Moreover, we obtain a pair of linear orders on the components of vectors x (4.1) and y (4.2): *component* $1(-1) \prec ... \prec$ *component* $k(-k)$.

Clearly, if the components are aggregate ones, the orders will be analogues ones. Fig. 4.3 depicts examples of vector domains, where α, β, and γ denote examples of indifference sets (Fig. 4.3a).

Definition 4.6. $x_1(S,Q) \succeq x_2(S,Q)$, $\Omega(x_1) = \Omega(x_2)$, $\forall S, Q \in \Theta(S)$ if any decreasing of weak components x_1 in the comparison with x_2 is compensated by corresponding increasing of it's 'strong' components

$(r, p \in \Omega^+$ or $-r, -p \in \Omega^-)$:

$$\sum_{r\geq u}^{r} x_1^r - \sum_{r\geq u}^{r} x_2^r \geq 0, \ \forall u \in \Omega^+ \ (\forall -u \in \Omega^-, -r \leq -u). \tag{4.4}$$

It is possible to force condition (4.4) by using a right side that is equal to a parameter $\upsilon > 0$.

Definition 4.7. Let $M = \{x \in X | \sum_{r\in\Omega} x^r = 1\}$ be a marginal set (similarly for y).

Note that $\forall x$ (y) there exists a dominating subset $D(x) = \{\eta \in M | \eta \succeq x\}$.

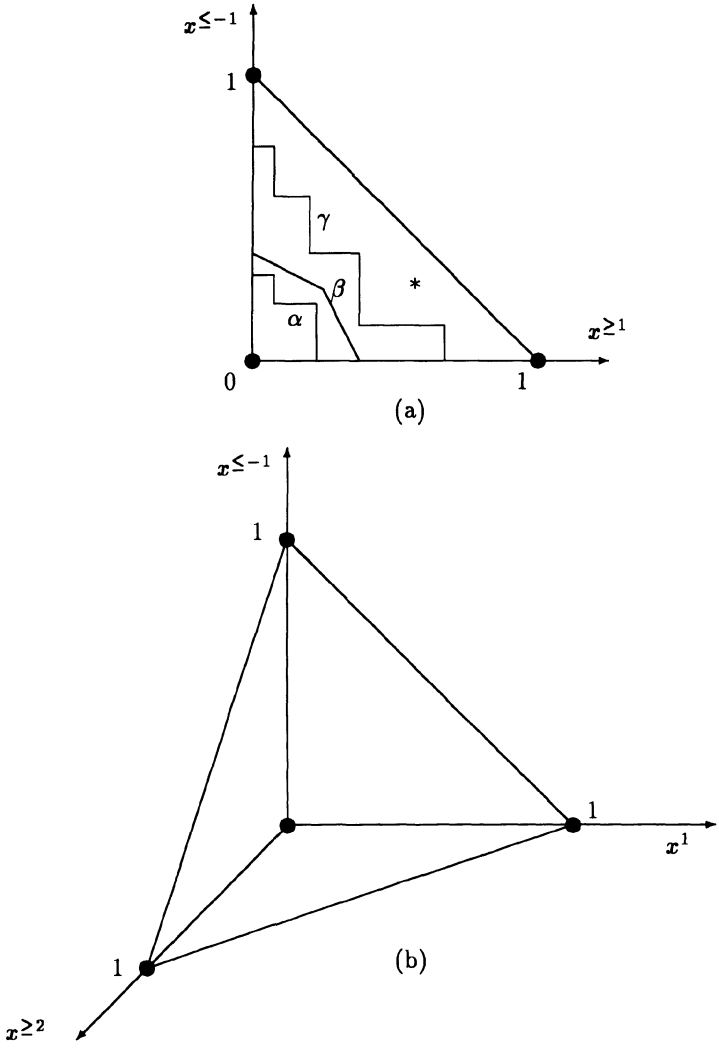

Fig. 4.3. Examples of vector domains

Definition 4.8. Let a pair of vectors x_1, x_2 (y_1, y_2) be:
(i) comparable ones if $x_1 \succeq x_2$ (therefore $D(x_2) \supseteq D(x_1)$ and vice versa);
(ii) strongly incomparable ones if $|D(x_1, x_2)| = |D(x_1) \& D(x_2)| = 0$;
(iii) weakly incomparable ones if $|D(x_1, x_2)| \neq 0$, and
$D(x_1, x_2)$ does not include $D(x_1)$, $D(x_2)$.

Fig. 4.4 illustrates Definition 4.8.

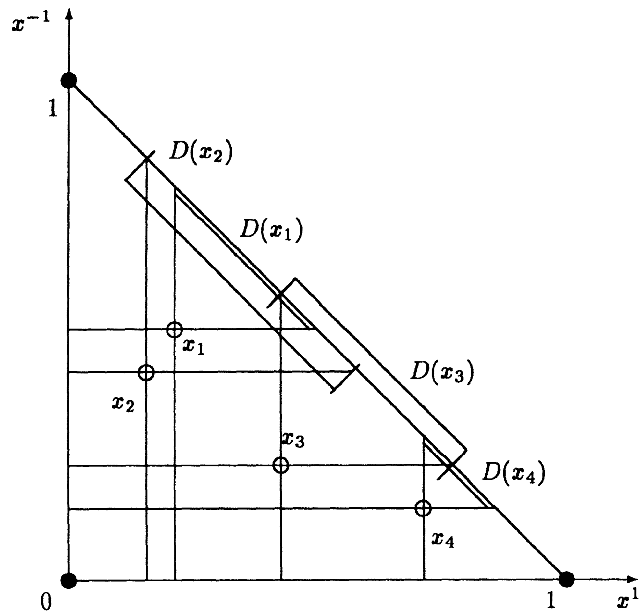

Fig. 4.4. Examples of dominating sets

4.4.3 Properties

Finally let us consider properties of our vector-like proximity as follows:

1. Condition (6.4) defines a poset.

2. $0 \leq |x(S,Q)| \leq 1$, $0 \leq |y(S,Q)| \leq 1$ $\forall S, Q \in \Theta(S)$.

3. $x(S,Q) \succeq (0, ..., 0)$, $y(S,Q) \succeq (0, .., 0)$ $\forall S, Q \in \Theta(S)$.

4. The following condition is true for one-side vectors:
$x(S,Q) \prec (0,0,...,0,1)$, $y(S,Q) \prec (0,0,...,0,1)$.

5. The following condition is true for any two-side vector $x(S,Q)$, $\forall S, Q \in \Theta(S)$:
there exists such vector $e = (e^{-k_1}, 0, ..., 0, e^{k_2}) \in M$ $(k_1, k_2 > 0)$, *that* $x(S,Q) \succeq e$ (similarly, for y).

6. For any modular vector the following is true:
$x(S,Q) = x(Q,S)$ (similarly, for y).

7. For any two-side symmetrical vector the following is true:
$x(S,Q) = x^{*}(Q,S)$, where $x^{*r} = x^{-r}$ (similarly, for y).

8. $\forall x(S,Q)$, $\forall S, Q \in \Theta(S)$, the following is true:
if $x(S,Q) = (0,..,0)$ then $S = Q$.

4.4.4 Numerical Example

Here let us consider rankings S^1 and S^1 from Fig. 4.2. Corresponding adjacency matrices are as follows:

$$|g_{ij}(S^1)| = \begin{pmatrix} . & -1 & 0 & -1 & 1 & 1 & 0 & 1 & -1 \\ 1 & . & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & -1 & . & -1 & 1 & 1 & 0 & 1 & -1 \\ 1 & 0 & 1 & . & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & . & 0 & -1 & 0 & -1 \\ -1 & -1 & -1 & -1 & 0 & . & -1 & 0 & -1 \\ 0 & -1 & 0 & -1 & 1 & 1 & . & 1 & -1 \\ -1 & -1 & -1 & -1 & 0 & 0 & -1 & . & -1 \\ 1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & . \end{pmatrix}$$

$$|g_{ij}(S^2)| = \begin{pmatrix} . & 1 & 0 & 1 & 1 & 1 & -1 & 1 & -1 \\ -1 & . & -1 & 1 & 0 & 1 & -1 & 0 & -1 \\ 0 & 1 & . & 1 & 1 & 1 & -1 & 1 & -1 \\ -1 & -1 & -1 & . & -1 & 0 & -1 & -1 & -1 \\ -1 & 0 & -1 & 1 & . & 1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & . & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & . & 1 & 0 \\ -1 & 0 & -1 & 1 & 0 & 1 & -1 & . & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & . \end{pmatrix}$$

Also, Kendall's distance is:

$$\rho_K(S^1, S^2) = 31.$$

Proposed vector-like proximities allow to describe more prominent the dissimilarity between two structures:

$$\pi(S^1) = (\pi_1(S^1), ..., \pi_i(S^1, ..., \pi_4(S^1) = (3,1,3,1,4,4,3,4,2),$$

$$\pi(S^2) = (\pi_1(S^2), ..., \pi_i(S^2, ..., \pi_4(S^2) = (2,3,2,4,3,4,1,3,1),$$

$$\delta_i^\pi(S^1, S^2) = (1,-2,1,-3,1,0,2,1,1),$$

$$\delta_{ij}^\pi(S^1, S^2) = \begin{pmatrix} . & 3 & 0 & 4 & 0 & 1 & -1 & 0 & 0 \\ -3 & . & -3 & -1 & -3 & -2 & -4 & -3 & -3 \\ 0 & 3 & . & -4 & 0 & 1 & 1 & 0 & 0 \\ -4 & 1 & 4 & . & -4 & -3 & -5 & -4 & -4 \\ 0 & 3 & 0 & 4 & . & 1 & -1 & 0 & 0 \\ -1 & -2 & -1 & 3 & -1 & . & -1 & -1 & -1 \\ 1 & 4 & -1 & 5 & 1 & 1 & . & 1 & 1 \\ 0 & 3 & 0 & 4 & 0 & 1 & -1 & . & 0 \\ 0 & 3 & 0 & 4 & 0 & 1 & -1 & 0 & . \end{pmatrix}$$

$$x(S^1, S^2) = (x^{-3}, x^{-2}, x^{-1}, x^1, x^2, x^3) = (1,1,0,5,1,0),$$

$$y(S^1, S^2) = (y^{-6}, y^{-5}, y^{-4}, y^{-3}, y^{-2}, y^{-1}, y^1, y^2, y^3, y^4, y^5, y^6) =$$
$$(0,1,5,5,1,6,6,0,1,1,0,0).$$

Note here we do not use in x and y the coefficients: $\frac{1}{n}$ and $\frac{2}{n(n-1)}$.

In addition, we can examine vector-like proximity with aggregate components, for example:

$$x(S^1, S^2) = (x^{\leq -1}, x^{\geq 1}) = (2,6).$$

This vector is shown in Fig. 4.3a by *. Clearly, the vector demonstrates a change of elements of A, mainly, to the top layer. Thus we face a new problem: construct for an applied task the best vector-like proximity or a set of the proximities.

4.4.5 *Application*

Proposed vector-like proximity may be used at standard stages of decision making procedures, for example:

(1) comparison of results;

(2) aggregation of results;

(3) sensitivity analysis, e.g., assessment of result's changes, when initial information (criteria, estimates, etc.) is changed;

(4) analysis of influence (e.g., influence of a criterion to a fragment of the decision);

(5) analysis of possibilities (definition of certain values for elements of initial information, when we can obtain required results); and

(6) risk analysis (e.g., assessment of result's changes in the case of probabilistic changes of initial information).

Moreover, one can build certain vector-like proximity. Consider an example. Let $\Phi^*(A)$ be a standard choice function to obtain a two-level resultant structure S^* on a set of alternatives A. We can use vector $x(S,S^*) = (x^{-1}, x^1)$ and vector constraint $x_o = (x_o^{-1}, x_o^1)$ to evaluate a quality of a choice function $\Phi(A)$, which defines a two-level structure S.

The following two cases may be examined:

(a) $x_o = (\tau, 0)$, $\tau \in [0,1]$ if we should like to select all effective decisions, and agree to include into resultant decision set some non-effective decisions; and

(b) $x_o = (0, \tau)$, otherwise.

Finally, it is reasonable to use sets of vector-like proximity for composite decisions (e.g., composite DA's).

4.4.6 *Comparison of Fuzzy Layered Structures*

An assessment of proximity between fuzzy layered structures is a more complicated problem. Let us consider an example of qualitative vector-like proximity for any fuzzy structures $S_f, Q_f \in \Theta(S_f)$, where $\Theta(S_f)$ is a set of all fuzzy layered structures on A.

Definition 4.9. Let $z(S_f, Q_f) = (z^{-(m-1)}, ..., z^{-1}, z^1, ..., z^{m-1})$, where

$$z^r = |\{i \in A | d_i^2(S_f) - d_i^1(Q_f) = r, |(d_i(S_f) \& d_i(Q_f))| = 0\}|/n, r > 0;$$

$$z^r = |\{i \in A | d_i^1(S_f) - d_i^2(Q_f) = -r, |(d_i(S_f) \& d_i(Q_f))| = 0\}|/n, r < 0;$$

be a vector of the first order proximity between $\forall S_f, Q_f \in \Theta(S_f)$ with respect to $\forall i \in A$.

In the same way, we may describe properties for vectors z, which are similar to those of vector x (y) (besides the 8th one).

4.4.7 *Aggregation of Layered Structures*

Let us consider an aggregation of layered structures. The problem consists in building of a consensus for a set of initial structures. We use the following conversation:

$$F : \{G^{\lambda} = (A, E^{\lambda}), \lambda = 1, ..., \Lambda\} \Longrightarrow G^{a} = (A, E^{a}).$$

In our opinion, it is reasonable to consider the following standard problems: $\{L\} \Rightarrow S$, $\{T\} \Rightarrow P$, $\{S\} \Rightarrow S$, $\{S\} \Rightarrow S_f$, $\{S_f\} \Rightarrow S_f$.

There exist the following three approaches to the aggregation problem above [273]:

(i) axiomatic;
(ii) criterial (e.g., concordance criteria, majority rules); and
(iii) modeling (i.e., usage of physical model in a space).

We will use a modeling approach to problem $\{S\} \Rightarrow S_f$ as follows:

$$h(S^{a} \mid s^{a} \in \{S \mid \eta(S, S^{\lambda}) \preceq \eta_o \ \forall \lambda = 1, ..., \Lambda\} \longrightarrow max,$$

where h is an attribute (quality) of the resultant ('overage') structure S^{a}; η_o is a vector-like proximity. Fig. 4.5 depicts this problem.

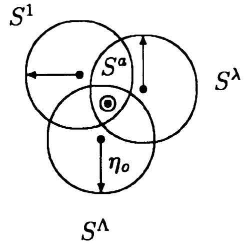

Fig. 4.5. Aggregation of structures, S^{a} is denoted by ⦿

Now let us examine problem $\{S\} \Rightarrow S_f$ in details. We use the following notations:

a_{il} is the number of initial structures, in which $i \in A_l, i = 1, ..., m$;

vector $\xi_i = (\xi_{i1}, ..., \xi_{il}, ..., \xi_{im})$ defines frequencies of correspondence of element i to layers $(1, ..., m)$, where $\xi_{il} = \frac{a_{il}}{\Lambda}$ (it is a membership function of element i to layer $l = 1, .., m$).

Denote S^{a}_{f} as a set of intervals $\{d_i\}$. So we evaluate a result (S^{a}_{f}) quality on the basis of the following entropy-like function:

$$\sum_{i=1}^{n} H_i = \sum_{i=1}^{n} \frac{1}{d_i^2 - d_i^1 + 1} \longrightarrow max.$$

Next, we use as proximity the following modular vector

$$z_o = (z^1, ..., z^k, 0, ..., 0).$$

As a result, our problem is:

$$\sum_{i=1}^{n} H_i(S_f) \longrightarrow max,$$

$$z(S^\lambda, S_f) \preceq z_o, \ \forall \lambda = 1, ..., \Lambda.$$

With respect to zero-valued components $z^{k+1}, ..., z^{k^+}$ it is possible to define a set of admissible variants for intervals $(d_{i\theta} \mid \theta = 1, ..., q_i)$. Thus we reduce our model to the following modification of multiple choice knapsack problem:

$$\sum_{i=1}^{n} \sum_{\theta=1}^{q_i} H_{i\theta}(d_{i\theta}) \kappa_{i\theta} \longrightarrow max,$$

$$\sum_{r \geq p} \sum_{i=1}^{n} \sum_{\theta=1}^{q_i} b_{i\theta}^r \kappa_{i\theta} \leq \sum_{r \geq p} z^r, \ p = 1, ..., k,$$

$$\sum_{\theta=1}^{q_i} \kappa_{i\theta} = 1, \ i = 1, ..., n, \ \kappa_{i\theta} = 0 \cup 1,$$

where $b_{i\theta}^r$ is the sum of components ξ_i, which are differed from $d_{i\theta}^1$ $(d_{i\theta}^2)$ by r. A version of the described aggregation scheme has been implemented in DSS COMBI ([294], [317]).

In the case we use two-side vector z_o, the first constraint of the model has to be transformed into two constraints for negative and positive components respectively. Note that examined modification of multiple choice knapsack problem is interesting for cases, when resources in knapsack-like problems are ordered by their importance and may be replaced.

4.5 COMPARISON OF TREE-LIKE MODELS

This section focuses on some approaches and examples for comparison of system versions. In this case, we should like to analyze above-mentioned basic components of system descriptions (i.e., set of DA's, leaf nodes for a system tree-like model, tree-like system model, etc.).

Our basic comparison problems are as follows:

(a) comparison of sets, including cases as follows:

(i) *set-set*,

(ii) *weighted set-weighted set*,

(iii) *ranking-ranking*,

(iv) *set-set & element closeness*,

(v) *set-set & comparison structure*, and

(vi) *set-set & element closeness & comparison structure*; and

(b) comparison of trees.

4.5.1 Subsets

The vector-like proximity for rankings has been analyzed in previous section, and can be applied for ordered DA's. Here we consider comparison of subsets, which is oriented to the following cases:

(1) *set-set* (ρ_s^o);

(2) *weighted set-weighted set* (ρ_s^1);

(3) *set-set & element closeness* (ρ_s^2);

(4) *set-set & comparison structure* (ρ_s^3); and

(5) *set-set & element closeness & comparison structure* (ρ_s^4).

Clearly, it is possible to consider modified versions of the above-mentioned set distances too. Fig. 4.6 illustrates a basic approach to the analysis of similarity/dissimilarity for subsets $S^1, S^2 \subseteq S = \{1, ..., i, ..., n\}$ ([55], [243], [411], etc.).

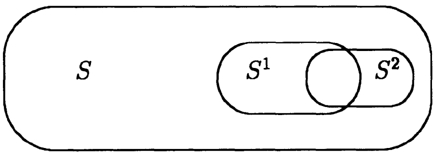

Fig. 4.6. Comparison of subsets

Thus it is reasonable to apply the following (*set-set*) distance (dissimilarity):

$$\rho_s^o(S^1, S^2) = \frac{|(S \setminus S^1) \cup (S \setminus S^2)|}{|(S^1 \cup S^2)|} = 1 - \frac{|(S^1 \& S^2)|}{|(S^1 \cup S^2)|}.$$

Here and hereafter we assume that $|(S^1 \cup S^2) > 0|$. Clearly, that corresponding similarity is:

$$d_s(S^1, S^2) = 1 - \rho_s^o(S^1, S^2).$$

Let $v_i \geq 0$ be a weight of element $i \in S$. In this case, we can use the following modified formulae (***weighted set-weighted set***):

$$\rho_s^1(S^1, S^2) = \frac{\sum_{i \in ((S \setminus S^1) \cup (S \setminus S^2))} v_i}{\sum_{i \in (S^1 \cup S^2)} v_i}.$$

Now let us introduce initial closeness for elements of set S: $0 \leq \zeta(i,j) \leq 1$, $\forall i, j \in S$. As a result, the following metric can be applied (***set-set & element closeness***):

$$\rho_s^2(S^1, S^2) = 1 - \frac{\sum_{(i \in S^1)\&(j \in S^2)} \zeta(i,j)}{\sum_{(i \in S^1 \cup S^2)\&(j \in S^1 \cup S^2)} \zeta(i,j)}.$$

In addition, it is reasonable to take into account a prohibition to compare for some element pair of S. In other words, there is a subset (a specific comparison binary relation) $\Omega(S) \subseteq S \times S$, which limits the element pairs set for comparison. Binary relation $\Omega(S)$ specifies comparison of corresponding elements (e.g., the same nature). For example, we should like to compare two teams, and each of that consists of the following kinds of specialists: managers, researchers, engineers, technicians, secretaries. Evidently, that it is reasonable to compare persons of corresponding subgroups. It is sure that we can consider analogical situations for other systems (e.g., engineering systems). Also, we can introduce $\Omega(S') = \{(i,j) \in \Omega(S) | (i \in S')\&(j \in S')\}$ $\forall S' \in S$. Note the above-mentioned subsets and their correspondence may be based on special matching problems. Thus we obtain the following (***set-set & element closeness & comparison structure***):

$$\rho_s^4(S^1, S^2) = 1 - \frac{\sum_{(i \in S^1)\&(j \in S^2)\&((i,j) \in \Omega(S^1 \cup S^2))} \zeta(i,j)}{\sum_{(i,j) \in \Omega(S^1 \cup S^2)} \zeta(i,j)}.$$

Clearly, the case ***set-set & comparison structure*** is more easy one ($\zeta(i,j) = 1$ $\forall i, j$):

$$\rho_s^3(S^1, S^2) = 1 - \frac{|\{(i,j) | (i \in S^1)\&(j \in S^2)\&((i,j) \in \Omega(S^1 \cup S^2))\}|}{|\{(i,j) \in \Omega(S^1 \cup S^2)\}|}.$$

Table 4.2 contains a numerical example. Some computed distances above are the following:

(a) $\rho_s^o(S^1, S^2) = 0.6$;
(b) $\rho_s^1(S^1, S^2) = 0.544$;

(c) $\rho_s^2(S^1, S^2) = 0.485$;
(d) $\rho_s^4(S^1, S^2) = 0.61$,
for $\Omega(S) = \{(1,1),(1,2),(2,1),(2,2),(3,3),(4,4),(4,5),(5,4),(5,5)\}$.

Clearly, in case (d) we examine a structure of S, that consists of the following three blocks: $\{1,2\}$, $\{3\}$, $\{4,5\}$. So this approach allows to compare DA's for leaf nodes of a system tree-like model.

Table 4.2. Numerical example for comparison of subsets

S: i	S^1	S^2	v_i	$\zeta(i,j)$				
				1	2	3	4	5
1	*	*	0.9	1.0	0.5	0.7	0.2	0.1
2		*	0.8	0.5	1.0	0.4	0.5	0.9
3	*	*	0.7	0.7	0.4	1.0	0.8	0.1
4	*		0.6	0.2	0.5	0.8	1.0	0.7
5	*		0.5	0.1	0.9	0.1	0.7	1.0

Thus the distances above can be used for comparison of DA's. To compare DA's with priorities we can apply the following:
(i) Kendall's metric,
(ii) traditional metrics for strings,
(iii) proposed vector-like proximity, and
(iv) metrics ρ_s^1, or ρ_s^4.

Now let us consider an applied example (comparison of teams). First, we specify the following initial set: 1. John (manager M_1); 2. David (researcher C_1); 3. Brad (engineer E_1); 4. Deborah (researcher C_2); 5. Michael (researcher C_3); 6. Claus (secretary R_1); 7. Alan (engineer E_2); 8. Laura (manager M_2); 9. Julia (secretary R_2); 10. Jonathan (manager M_3); 11. Olav (engineer E_3); 12. Stephan (technician T_1); 13. Thomas (engineer E_4); and 14. Franz (technician T_2).

Secondly, the element closeness is introduced on the basis of the following relations:
(a) for the same element the closeness is equal to 1.0;
(b) for relationship *brothers/sisters* the closeness is equal to 0.7;
(c) for relationship *cousins* the closeness is equal to 0.4.
(d) for relationship *second cousins* the closeness is equal to 0.2.

Clearly, that other relationships can be applied (e.g., friendship, the same educational background, close employment histories).

In our example, relationships are as follows:
(1) *brothers/sisters*: {John, Deborah, Alan}, {Michael, Jonathan}, {Laura, Julia, Olav};

(2) *cousins*: {John, Michael}, {John, Jonathan}, {Deborah, Michael}, {Deborah, Jonathan}, {Alan, Michael}, {Alan, Jonathan}, {David, Laura}, {David, Julia}, {David, Olav}, {Brad, Claus}, {Stephan, Thomas};

(3) *second cousins*: {John, Stephan}, {John, Thomas}, {Deborah, Stephan}, {Deborah, Thomas}, {Alan, Stephan}, {Alan, Thomas}.

As a result, we get the element closeness, which is presented in Table 4.3.

Table 4.3. Element closeness

	M_1	C_1	E_1	C_2	C_3	R_1	E_2	M_2	R_2	M_3	E_3	T_1	E_4	T_2
1.M_1	1.0	0	0	0.7	0.4	0	0.7	0	0	0.4	0	0.2	0.2	0
2.C_1	0	1.0	0	0	0	0	0	0.4	0.4	0	0.4	0	0	0
3.E_1	0	0	1.0	0	0	0.4	0	0	0	0	0	0	0	0
4.C_2	0.7	0	0	1.0	0.4	0	0.7	0	0	0.4	0	0.2	0.2	0
5.C_3	0.4	0	0	0.4	1.0	0	0.4	0	0	0.7	0	0	0	0
6.R_1	0	0	0.4	0	0	1.0	0	0	0	0	0	0	0	0
7.E_2	0.7	0	0	0.7	0.4	0	1.0	0	0	0.4	0	0.2	0.2	0
8.M_2	0	0.4	0	0	0	0	0	1.0	0.7	0	0.7	0	0	0
9.R_2	0	0.4	0	0	0	0	0	0.8	1.0	0	0.7	0	0	0
10.M_3	0.4	0	0	0.4	0.7	0	0.4	0	0	1.0	0	0	0	0
11.E_3	0	0.4	0	0	0	0	0	0.7	0.7	0	1.0	0	0	0
12.T_1	0.2	0	0	0.2	0	0	0.2	0	0	0	0	1.0	0.4	0
13.E_4	0.2	0	0	0.2	0	0	0.2	0	0	0	0	0.4	1.0	0
14.T_2	0	0	0	0	0	0	0	0	0	0	0	0	0	1.0

So we can examine two subsets (Fig. 4.7):

(a) $S^1 = \{4,5,6,7,8,11,12\} = \{C_2, C_3, R_1, E_2, M_2, E_3, T_1\}$;

(b) $S^2 = \{2,4,6,8,9,10,11,13,14\} = \{C_1, C_2, R_1, M_2, R_2, M_3, E_3, E_4, T_2\}$.

In other words, we get the following structured sets:

(i) $S^1 = M_2 * R_1 * (C_2 \& C_3) * (E_2 \& E_3) * T_1$;

(ii) $S^2 = (M_2 \& M_3) * (R_1 \& R_2) * (C_1 \& C_2)(E_3 \& E_4) * T_2$.

Note that $S^1 \cup S^2 = \{2,4,5,6,7,8,9,10,11,12,13,14\}$.

We examine a structure of S that consists of the following blocks: $\{8,10\}$, $\{6,9\}$, $\{2,4,5\}$, $\{7,11,13\}$, $\{12,14\}$. Here

$\{(i,j) \in \Omega(S^1 \cup S^2) | i \in S^1 \& i \in S^1\} = \{(4,2),(4,4),(5,2),$
$(5,4),(6,6),(6,9),(7,11),(7,13),(8,8),(8,10),(11,11),(11,13)\}$, and

$\Omega(S^1 \cup S^2) = \{(2,2),(2,4),(2,5),(4,2),(4,4),(4,5),(5,2),(5,4),(5,5),$
$(6,6),(6,9),(7,7),(7,11),(7,13),(8,8),(8,10),(9,6),(10,8),(10,10),(11,7),$
$(11,11),(11,13),(12,12),(12,14),(13,7),(13,11),(13,13),(14,12),(14,14)\}$.

Finally, distances between teams S^1 and S^1 are as follows:
(a) $\rho_s^o(S^1, S^2) = 0.66$;
(b) $\rho_s^2(S^1, S^2) = 1 - 11.4/27/1 = 0.579$;
(c) $\rho_s^4(S^1, S^2) = 1 - 8.7/13.6 = 0.361$.

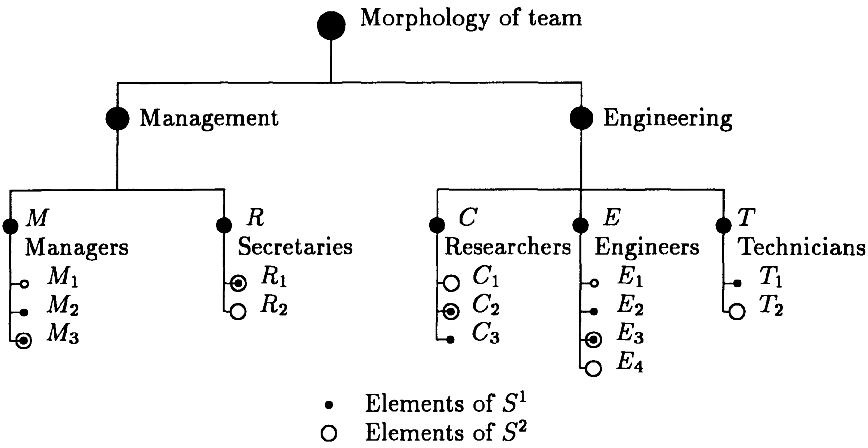

Fig. 4.7. Structure for comparison of teams

4.5.2 Comparison of Trees

Let us examine a distance for two labeled ordered trees T and T'. The distance from T to T' can be measured by the following two main approaches:

1. A composition of sets (subsets) differences ([55], [243], [411], etc.).

2. The minimum cost sequence of ***edit*** operations to transform T into T' ([18], [366], [454], [497], etc.).

Fundamentals of the first approach have been considered previously.

The problem is to determine, for two labeled ordered trees T and T'. Fig. 4.8 and 4.9 illustrate a generalized glance to tree comparison on the basis of tree parts.

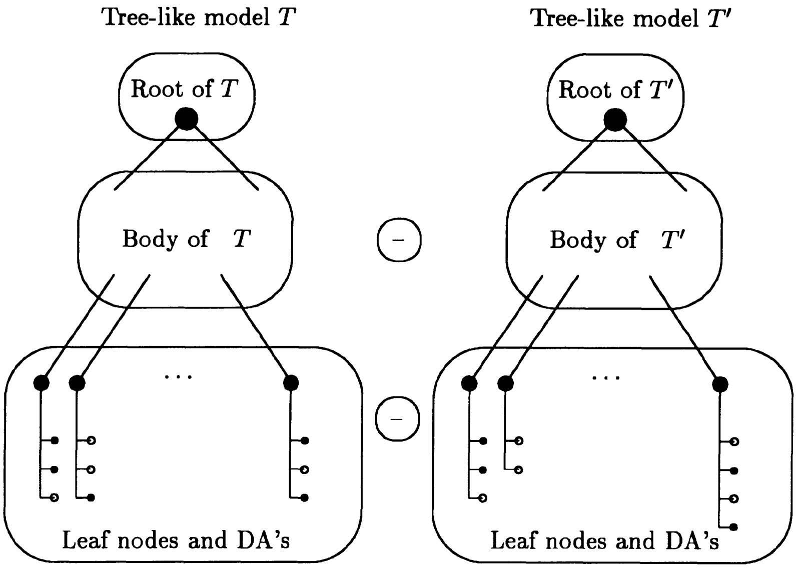

Fig. 4.8. Comparison of trees

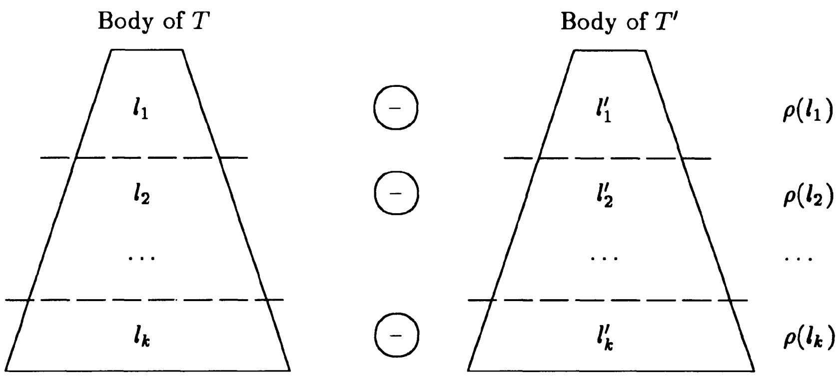

Fig. 4.9. Layer comparison of tree bodies

Thus we can consider the following approach to compare two tree bodies (Fig. 4.9):

(1) to divide tree bodies into layers;

(2) to match the layers, as a result we will obtain layers $\{l_i | i = 1...k\}$;

(3) to compare corresponding layers (l_i) on the basis of the following:

(i) comparison of node partitions (e.g., $\rho_s^3(l^i)$);

(ii) comparison of nodes with corresponding DA's (this case corresponds to comparison of leaf nodes with DA's on the basis of $\rho_s^4(l_i)$); and

(iii) composite case with the use of vector-like proximity as follows: $\rho(l_i) = (\rho_s^3(l_i), \rho_s^4(l_i))$.

Finally, we get a vector distance for body comparison

$$\rho_b(T, T') = (\rho(l_1), .., \rho(l_i), ..., \rho(l_k),$$

and for tree-like model comparison

$$\rho(T, T') = (\rho_{lna}(T, T'), \rho_b(T, T')),$$

where $\rho_{lna}(T, T')$ is a distance for levels of leaf nodes and DA's (e.g., ρ_s^2, ρ_s^4).

The second approach is based on the tree-to-tree correction problems. Let us consider some examples of combinatorial operations to the transformations. Tai has examined the following set of allowable edit operations [497]:

(1) changing one node into another one (changing the label of the node);

(2) deleting one node from a tree, and

(3) inserting a node into a tree.

Selkow et al. have analyzed a case with more simple edit operations as follows [454]:

(1) deleting a leaf node from a tree, and

(2) inserting a leaf node into a tree.

A new dissimilarity measure for evolutionary trees, has been proposed in [366]. The measure is based on the following: (a) duplications of nodes; (b) losses of nodes; and (c) gaps of nodes.

Ramesh et al. have considered more complicated changes while taking into account two operations of replacement [409]:

(1) identification of subtrees, which can be replaced (*tree pattern matching*);

(2) selection of one or more of these subtrees for replacement (*reduction strategy*).

Mirkin et al. have pointed out that often "there is no formal difference between inconsistencies that arise from different techniques applied to the same data or the difference in the data themselves" [366].

In our case, we try to analyze changes of a system model as follows:

(i) insertion of a system component/node;

(ii) deletion of a system component/node; and

(iii) change of a system component/node.

Note we have presented the following cases in previous subsection:

(a) insertion, deletion, and change of a design alternative for a system component/node;

(b) insertion, deletion, and change of a leaf node(s) for the system model; and

(c) composite case.

Thus it is necessary to examine the insertion, deletion, change, and replacement of a system component/node (with corresponding subtrees) at hierarchical levels between a root and leaf nodes, i.e., for tree bodies (Fig. 4.8).

From application viewpoints, it is reasonable to investigate the following set of transformation operations for systems (Fig. 4.10):

(a) ***change*** (change of a sybsystem);

(b) ***separation*** (separation of an additional subsystem);

(c) ***duplication*** (change of a subsystem into new two ones);

(d) ***sewing*** (unification of two subsystems into a new integrated one); and

(e) ***absorption*** (inclusion of a subsystem into a corresponding "father" system).

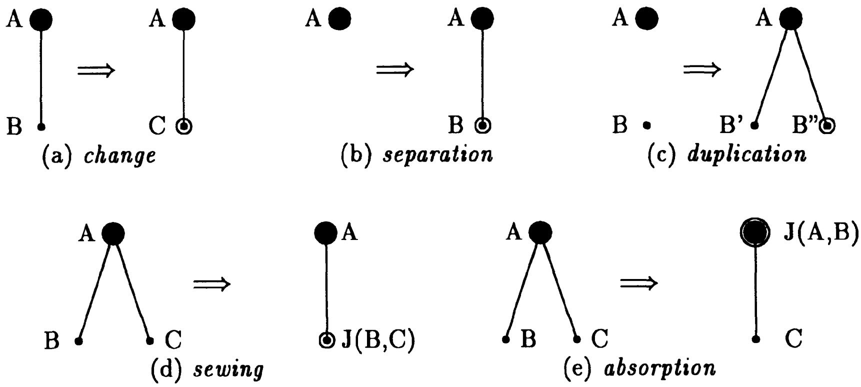

Fig. 4.10. Transformation operations

4.6 SUMMARY

In the chapter, only some significant issues of system comparison are briefly considered. Note that often similar problems are examined in classification theory and its applications, social science, statistics, data analysis, graph theory and orders, information engineering ([30], [90], [243], [244], [273], [345], [362], [363], [365], [400], [423], [425], [439], [441], [444], [530], etc.).

In our opinion, the importance of comparison problems for decomposable systems is increasing for many crucial application domains (e.g., biology, engineering, management). The situation is based on a fact that often composite systems and composite decisions are applied. For example, distances for software components (e.g., stacks, sets, lists) have been proposed and applied for software reuse processes in [80].

On the other hand, studies of comparisons for various combinatorial objects are prospective ones for discrete mathematics too.

5 TRANSFORMATION OF DECOMPOSABLE SYSTEMS

In this chapter, we examine issues of transformation (improvement, adaptation) of decomposable systems [310], including the following: structure of transformation proceses, transformation trajectories, and a numerical example of an information system transformation.

5.1 GENERALIZED APPROACH

Problems of transformation (improvement, etc.) of complex systems have been studied in various disciplines ([42], [103], [188], etc.). This section addresses the description, and improvement of decomposable systems. We analyze the use of HMMD (designing a new system) [300] to represent and to design an improvement process. Note that hierarchical approaches to plan or to schedule have been studied many years, for example:

(1) hierarchical planning systems [114];

(2) hierarchical decision making in manufacturing [202];

(3) hierarchical tasks network (HTN) decomposition ([136], [137], etc.).

Here we use HMMD not only for the design and analysis of a system, but to design a change system (a hierarchy of improvement actions) and to plan a system improvement process too. Similar processes are basic in the quality improvement, and redesign or reengineering. We analyze decomposable systems, main elements of the improvement process, our generalized framework of the improvement, and support combinatorial models to schedule improvement actions (e.g., clique, morphological clique, etc.). Our numerical example demonstrates stages of the improvement process.

The following situation of a system change is presented in Fig. 5.1:

1. An initial system is:

$S = A * B * C * D$ with corresponding DA's (A_1, A_2, A_3, A_4; B_1, B_2, B_3, B_4; C_1, C_2, C_3, C_4; D_1, D_2, D_3, D_4).

2. Change actions are the following:

(i) changing of the system structure:

(a) removal of component D; and

(b) addition of component E;

(ii) changing of DA's:

(a) removal of B_2, C_1; and

(b) addition of A_3, A_4, B_4, C_3, C_4.

$$S = A * B * C * D^- \Rightarrow S = A * B * C * E^+$$

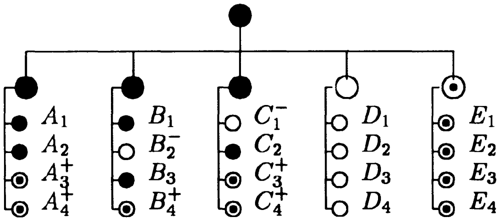

Fig. 5.1. Example of modified system

Here we analyze the following layers of system excellence:

(1) an ideal decision;

(2) Pareto-effective points; and

(3) a neighborhood of Pareto-effective DA's (e.g., a decision of this set maybe transformed into a Pareto-effective point on the basis of the only one improvement step).

An improvement of the system is illustrated in Fig. 5.2. Here we point out the following:

(a) points: initial point S_o; the ideal point I; four Pareto-effective points; target point S^*; S_{o1} and S_{o2}, that are intermediate points of improvements (these points may be examined as neighbors of the Pareto-layer); and

(b) series trajectories of the improvements:

$\alpha =< S_o, S_{o1}, S^* >$ and $\beta =< S_o, S_{o2}, S^* >$.

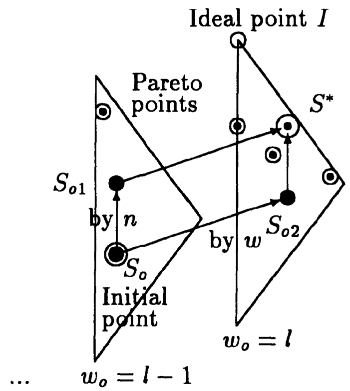

Fig. 5.2. Excellence lattice, improvements ($\longrightarrow$)

So we propose a similar stage for the statement and implementation of the improvement process on the basis of HMMD. At this stage, we have to examine new kind of DA's as improvement actions, their interconnection (compatibility), and scheduling of these actions.

Generally, we examine the system improvement as a series of steps of the representation and processing of the following:

(a) initial system;

(b) hierarchical morphological design space;

(c) hierarchical change system (system of improvement actions); and

(d) schedule of change actions.

Thus we need the description for elements above, and methods to their processing. Note that a hierarchical approaches to plan are the basic ones ([114], [202], etc.).

5.2 STRUCTURE OF TRANSFORMATION PROCESS

Now let us consider the following interconnected levels:

(1) a space of system excellence, for example on the basis of the lattice above;

(2) a set of compositions (composite DA's); and

(3) a set of improvement trajectories, including a set of elementary improvement actions, and their series-parallel combinations (i.e., series-parallel trajectories).

Spaces of objects and their effectiveness are depicted in Fig. 5.3. Here we point out kinds of correspondences between elements of spaces above too.

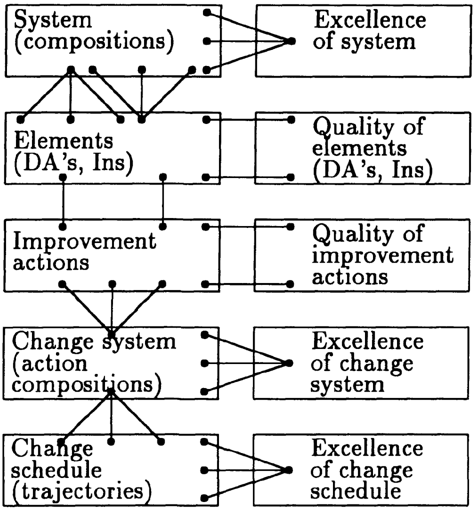

Fig. 5.3. Spaces of objects and effectiveness

Clearly, we have to take into account the following cases for a point of the system excellence space:

(1) a corresponding composition does not exist;

(2) there exists only one corresponding composition; and

(3) there exist a set of corresponding compositions.

Analogically for two compositions as source/destination points of the improvement process we have got the same three cases.

Now we can point out several attempts to describe and use close multi-level descriptions of complex processes, for example:

(1) hierarchical task network planning ([136], [137], etc.);

(2) network languages for complex systems ([491], [492]); and

(3) families of coordination algorithms for multi-agent environments ([111], [112], etc.)

Here we examine the main stages of the improvement process which are shown in Table 5.1.

Generally, we can examine the following types of system changes:

1. Internal changes:

1.1 micro-level: (1) change of a subsystem (submodel, requirements); (2) change of DA's; (3) change of Ins;

1.2 macro-level: change of a system structure.

2. External changes:

2.1 requirements to the system;

2.2 searching for morphological decisions.

Table 5.1. Series improvement process

Object	Operations	Methods
1.Hierarchical description of existing system	Analysis of system, partitioning/ decomposition	Engineering techniques
2.Initial hierarchical morphological design space	Generation of new DA's (concurrently) Change of system structure	Engineering techniques Searching for new data
3.Extended hierarchical morphological design space	Generation of aggregate DA's (concurrently)	Clique problem with ordinal item compatibility
4.Hierarchical description of change system (improvement actions)	Analysis of system excellence and generation of improvement actions	HMMD for basic system, its analysis (morphological clique and improvement analysis)
5.Change system (selected improvement actions)	Design of change system and its analysis	HMMD for change system (morphological clique and improvement analysis)
6.Change schedule system (e.g., series-parallel schedule of improvement actions)	Design of series-parallel schedule or trajectory	HMMD for series-parallel schedule, dynamic programming, network planning, etc.

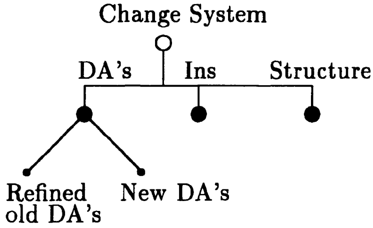

Fig. 5.4. Hierarchy of improvements

Now it is reasonable to investigate new types of requirements to new DA's (i.e., the improvement actions), their interconnection, and a structure of the system changes (Fig. 5.4).

5.3 PHASES OF TRANSFORMATION

Basic phases of the improvement process are the following:

Phase 1. Analysis of the initial system:
(1.1) analysis of the existing system;
(1.2) generation of new DA's and/or new system structure;
(1.3) generation of aggregate DA's;
(1.4) assessment of components (DA's, Ins);
(1.5) evaluation of the system versions (i.e., composite DA's).

Phase 2. Generation of the improvement action set:
(2.1) generation of improvement actions on the basis of the following: (a) expert judgment, (b) examination of bottlenecks, (c) examination of the neighborhood of the Pareto-effective points, and (d) examination of series neighborhood layers (i.e., the Pareto-effective points layer, the neighborhood of the Pareto-effective points layer, etc.);
(2.2) evaluation of the improvement actions including the following: (a) profit of the actions, (b) required resources (time, etc.), (c) analysis of equivalent actions and their integration, and (d) pair precedence relation between the actions;
(2.3) selection of admissible actions and building of an action hierarchy (Fig. 5.3);
(2.4) building of a precedence digraph on the action set.

Phase 3. Design of an improvement implementation plan (trajectory) on the basis of the following approaches: traditional network planning, dynamic programming, multistage planning, and scheduling.

List of basic support procedures is the following:
(1) ordinal assessment of DA's, and Ins;
(2) building of aggregate DA's;
(3) building of composite DA's;
(4) analysis of composite DA's;
(5) generation of improvement actions; and
(6) design of transformation trajectories in a layered network of transformation actions (an operational network).

5.4 TRANSFORMATION TRAJECTORY

At the transformation phase 3 from previous section (design of trajectory), we can examine the following types of problems:

(i) *optimization 1* (**P'**): Find the best improvement plan while taking into account results (an excellence of the target system) and required resources;

(ii) *recognition* (**P"**): Define the possibility (i.e., *Yes* or *No*) of reaching a specified target decision(s) on the basis of the specified set of the improvement actions;

(iii) *optimization 2* (**P"'**): Define the best improvement plan to reach the target system(s) in the case of the existence of the possibility (from problem *recognition*).

It is reasonable to use parallelism and/or coordination of improvement actions for problem **P'**. In this case, we can design a multiperiod series-parallel improvement strategy on the basis of HMMD ([298], [304]). The above-mentioned hierarchy of improvements may be analyzed for each period while taking into account precedence relation of the actions (a basic morphological change system).

For problem **P"** and **P"'**, we can propose an analysis of series neighborhood layers (a layered network) and searching for an improvement trajectory on the basis of two basic strategies (dynamic programming):

(a) from the target system(s) to the initial one; and
(b) from the initial system to the target one(s).

Fig. 5.5, 5.6, 5.7, and 5.8 depict strategies to find the best trajectory for four basic problems as follows:

(a) ***point-to-point***;
(b) ***point-to-set***;
(c) ***set-to-point***; and

(d) *set-to-set.*

Evidently, these problems correspond to searching for an extremal path(s) in a digraph. Layers, shown in the Figures, correspond to nodes with the same distance (e.g., a number of edges) from a basic point(s). Polynomial algorithms for the above-mentioned problems are well-known (***breadth-first search***): case (a) ([7], [414], etc.); cases (b) and (c) ([414], [495], etc.); and case (d) ([414], [388], [495], etc.).

Analogical path problems with vector (multicriteria) weights of edges (vertices) are more complicated and can be NP-hard [517].

Generally, the following methods or their combinations can be used for the design of transformation trajectories:

(1) searching for the best trajectory in an operational network on the basis of the above-mentioned network methods (extremal paths); and

(2) design of series-parallel schedules on the basis of morphological clique problem (series-parallel multiperiod strategy);

(3) design of schedules on the basis of traditional approaches ([51], [160], etc.); and

(4) interactive procedures for multicriteria trajectory optimization ([470], [488], etc.).

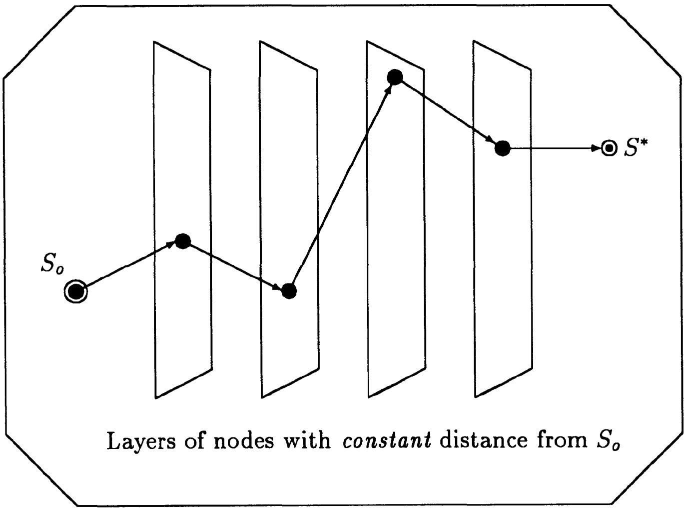

Fig. 5.5. Trajectory problem (a) *point-to-point*

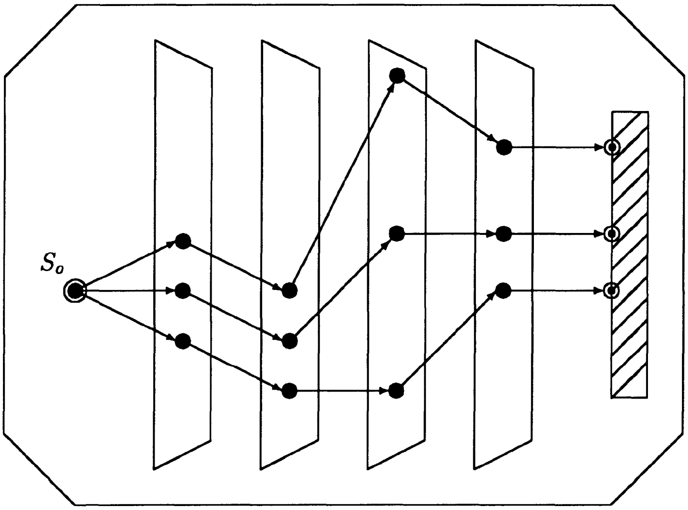

Fig. 5.6. Trajectory problem (b) *point-to-set*

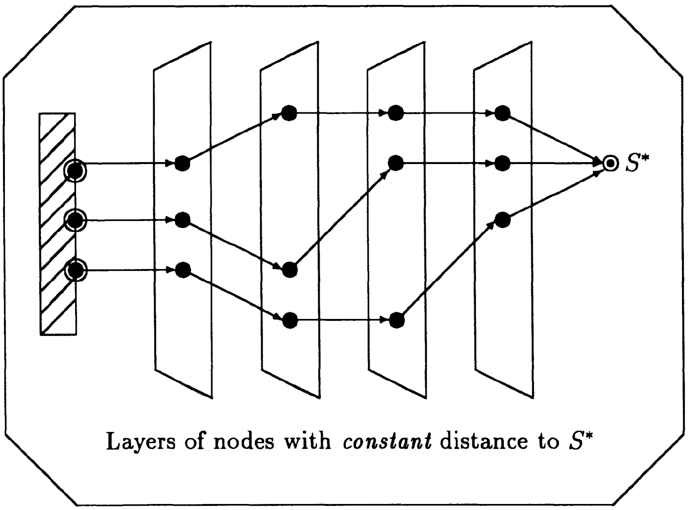

Fig. 5.7. Trajectory problem (c) *set-to-point*

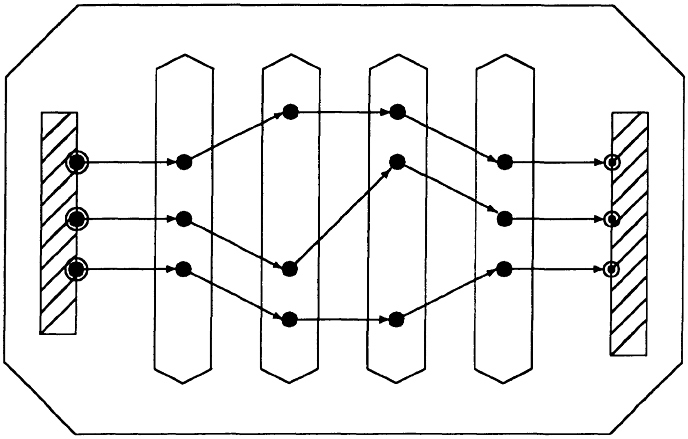

Fig. 5.8. Trajectory problem (d) *set-to-set*

5.5 PRESENTATION ISSUES

The importance of a complex object's presentation is increasing. In our case, we have to analyze several kinds of the systems (i.e., initial system, design space, change system, and improvement schedule), and their processing. Main presentation approaches for objects are as follows:

(1) structural modeling ([164], etc.);

(2) morphological presentation of complex objects and hierarchical alternatives ([300], [392], etc.); and

(3) diagrams and flowcharts (e.g., for scheduling).

Techniques for presentation of processes are mainly based on flowcharts, the use of languages, and special multi-media environments, for example:

(1) representation of complex technological processes (e.g., nets, bar diagrams, dataflow diagrams ([331], [384], etc.);

(2) morphological flowchart presentation of operational environments [294];

(3) special languages ([161], etc.); and

(4) complex presentation of algorithms/technique environments on the basis of texts, animation, movements ([361], etc.).

5.6 EXAMPLE OF SYSTEM TRANSFORMATION

Here we consider system transformation as designing a multiperiod technology strategy. Note design of long-term competitive strategies is an important problem for many companies ([2], [17], [63], [83], [91], [234], [509], etc.). On the other hand, our example corresponds to reengineering for various systems (e.g., business) too.

5.6.1 System and its Analysis

We examine the following initial computer system S: hardware (J), software (V), information (Y), and personnel (H). A detailed investigation of an information center has been executed in [301] (section 4.3). Here the initial system is a composition of DA's as follows: $S_o = J_0 * V_0 * Y_0 * H_0$.

At the next stage, we consider the following:

(a) generation of new DA's;

(b) design of a new system structure (an additional component ***communication*** C);

(c) generation of aggregate DA's; and

(d) deletion of Y_2.

Clearly, that now $S_0 = J_0 * V_0 * Y_0 * H_0 * C_0$.

Table 5.2 contains descriptions of DA's (priorities are shown in brackets). Compatibility of DA's are presented in Table 5.3.

Thus $N(S_o) = (2; 0, 4, 1)$. Also, a resultant Pareto-effective point set consists of the following elements:

1. $N = (2; 4, 1, 0)$:
$S_1 = J_1 * V_1 * Y_2 * H_1 * C_1$,
$S_2 = J_2 * V_1 * Y_2 * H_1 * C_1$,
$S_3 = J_1 * V_1 * Y_2 * H_2 * C_1$,
$S_4 = J_2 * V_1 * Y_2 * H_2 * C_1$.

2. $N = (3; 2, 3, 0)$:
$S_5 = J_1 * V_1 * Y_0 * H_1 * C_0$,
$S_6 = J_2 * V_1 * Y_0 * H_1 * C_0$,
$S_7 = J_1 * V_1 * Y_0 * H_2 * C_0$,
$S_8 = J_2 * V_1 * Y_0 * H_2 * C_0$.

Table 5.4 contains improvement actions, which are obtained on the basis of bottlenecks (i.e., S-***aggravating*** elements). Note we use the following types of improvements by results: (1) generation of an ideal point; (2) improvement of Pareto-effective points; (3) refinement of neighbors of the Pareto-effective points layer; and (4) improvement and compression of the Pareto-effective points layer.

Table 5.2. DA's

DA's	
Several personal computers	$J_o(3)$
Workstation	$J_1(2)$
LAN	$J_2(2)$
Initial DBMS	$V_o(2)$
New DBMS	$V_1(1)$
Expert system	$V_2(3)$
New DBMS and expert system	$V_3 = V_1 \& V_2(2)$
Initial database	$Y_o(2)$
Initial database and special intellectual interface	$Y_1(1)$
Knowledge base	$Y_2(3)$
Initial data base and knowledge base	$Y_3 = I_1 \& I_2(2)$
Initial personnel	$H_o(2)$
Trained personnel	$H_1(1)$
Trained personnel and a knowledge engineer	$H_2(1)$
New personnel oriented to knowledge engineering	$H_3(2)$
None	$C_o(2)$
Access to external databases in certain time	$C_1(1)$
Real-time communication	$C_2(2)$

Table 5.3. Compatibility of DA's

	Y_o	Y_1	Y_3	Y_0	J_1	J_2	J_0	V_1	V_2	V_3	V_0	C_1	C_2
H_0	3	2	1	3	3	3	3	3	0	1	3	2	1
H_1	3	3	1	3	3	3	3	3	0	2	3	3	2
H_2	3	3	1	3	3	3	3	3	0	2	3	3	3
H_3	3	3	3	2	2	2	3	3	3	3	3	3	3
Y_o				2	3	3	3	3	0	3	3	1	1
Y_1				2	3	3	3	3	0	3	1	3	3
Y_3				1	3	3	2	2	1	0	2	3	3
J_o							3	3	2	2	3	2	1
J_1							3	3	3	3	3	3	3
J_2							3	3	3	3	3	3	3
V_o											3	0	0
V_1											3	2	2
V_2											0	3	3
V_3											2	3	3

5.6.2 *Change System*

Generally, the change system consists of the following subsystems (we point out possible improvement actions for our example):

1. Improvement of components: J, V, Y, H, C.

2. Improvement of compatibility: (J, V), (J, Y), etc.

Our consideration in previous section is the base to compress the change space, because we will examine only improvements of S_o, and Pareto-effective points.

5.6.3 *Trajectories*

Now let us consider improvements in the space of compositions. We recall that S_o is our basic point, and for each other improvement trajectory it is necessary to add a start part as follows: from S_o to a point (e.g., S_1, etc.). Thus we can examine the following three kinds of improvement trajectories:

(1) from S_o directly to ideal point I;

(2) from S_o to points S_1 or S_2 or S_3 or S_4, and from the point to I; and

(3) from S_o to points S_5 or S_6 or S_7 or S_8, and from the point to I.

Table 5.4. Bottlenecks and improvements

Composite DA's	Bottlenecks		Actions	
	DA's	Ins	w/r	Type
S_1	J_1		$2 \to 1$	1
S_1		V_1, C_1	$2 \to 3$	1
S_2	J_2		$2 \to 1$	1
S_2		V_1, C_1	$2 \to 3$	1
S_3	J_1		$2 \to 1$	1
S_3		V_1, C_1	$2 \to 3$	1
S_4	J_2		$2 \to 3$	1
S_4		V_1, C_1	$2 \to 3$	1
S_5	J_1		$2 \to 1$	2
S_5	Y_0		$2 \to 1$	2
S_5	V_0		$2 \to 1$	2
S_6	J_2		$2 \to 1$	2
S_6	Y_0		$2 \to 1$	2
S_6	V_0		$2 \to 1$	2
S_7	J_1		$2 \to 1$	2
S_7	Y_0		$2 \to 1$	2
S_7	V_0		$2 \to 1$	2
S_8	J_2		$2 \to 1$	2
S_8	Y_0		$2 \to 1$	2
S_8	V_0		$2 \to 1$	2

We consider two types of improvement actions as follows: (a) a replacement of an element ($\to$) and (b) an improvement of an element ($\uparrow$). In our example, basic improvement actions are the following: $J_0 \to J_1$, $J_0 \to J_2$, $V_0 \to V_1$, $Y_0 \to Y_1$, $H_0 \to H_2$, $H_0 \to H_2$, and $C_0 \to C_1$.

The improvement actions for Pareto-effective points are presented in Table 5.4. Finally, we can consider series-parallel trajectories of the 2nd kind above:

1. $\alpha_1 = (S_o \to S_1) * (S_1 \uparrow) =$

(a)
$$((J_o \to J_1)\&(V_o \to V_1)\&(Y_o \to Y_2)\&(H_o \to H_1)\&(C_o \to C_1)) * (J_1 \uparrow);$$

(b)
$$((J_o \to J_1)\&(V_o \to V_1)\&(Y_o \to Y_2)\&(H_o \to H_1)\&(C_o \to C_1)) * ((V_1, C_1) \uparrow).$$

2. $\alpha_2 = (S_o \to S_2) * (S_2 \uparrow) =$

(a)
$$((J_o \to J_2)\&(V_o \to V_1)\&(Y_o \to Y_2)\&(H_o \to H_1)\&(C_o \to C_1)) * (J_2 \uparrow);$$

(b)
$$((J_o \to J_1)\&(V_o \to V_1)\&(Y_o \to Y_2)\&(H_o \to H_1)\&(C_o \to C_1)) * ((V_1, C_1) \uparrow).$$

3. $\alpha_3 = (S_o \to S_3) * (S_3 \uparrow) =$

(a)
$$((J_o \to J_1)\&(V_o \to V_1)\&(Y_o \to Y_2)\&(H_o \to H_2)\&(C_o \to C_1)) * (J_1 \uparrow);$$

(b)
$$((J_o \to J_1)\&(V_o \to V_1)\&(Y_o \to Y_2)\&(H_o \to H_2)\&(C_o \to C_1)) * ((V_1, C_1) \uparrow).$$

4. $\alpha_4 = (S_o \to S_4) * (S_4 \uparrow) =$

(a)
$$((J_o \to J_2)\&(V_o \to V_1)\&(Y_o \to Y_2)\&(H_o \to H_2)\&(C_o \to C_1)) * (J_2 \uparrow);$$

(b)
$$((J_o \to J_1)\&(V_o \to V_1)\&(Y_o \to Y_2)\&(H_o \to H_2)\&(C_o \to C_1)) * ((V_1, C_1) \uparrow).$$

In addition, improvement trajectory as a parallel-series chain $\alpha_5 = (S_o \to S_5) * (S_5 \to I)$ is shown in Fig. 5.9. Fig. 5.10 illustrates trajectories of kinds α_1 and α_5 in the space of system excellence.

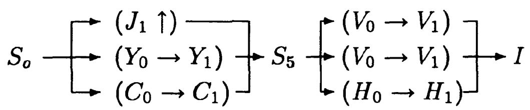

Fig. 5.9. Series-parallel improvement trajectory

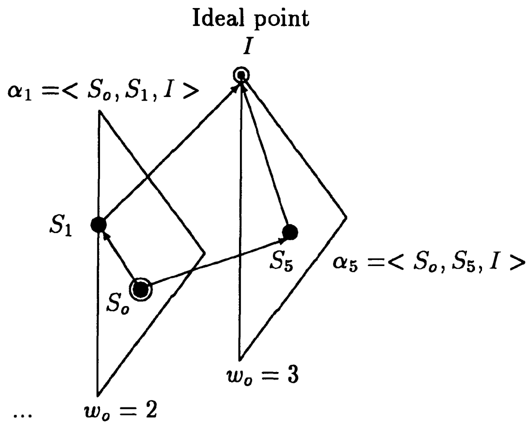

Fig. 5.10. Examples of improvement trajectories

5.7 SUMMARY

We have proposed our viewpoint to reengineering of decomposable systems. Our examination may be used for various applications, for example:

(1) distributed information systems (modification, improvement, redesign);

(2) improvement of network systems through modification; and

(3) reengineering of business processes.

In addition, it is reasonable to point out the significance of the kinds of optimization problems, when we search for the best improvement of a combinatorial system. Similar approach can be used for many well-known combinatorial problems on graphs. Note Roberts has examined close problems: to design a strategy for improvement of systems described by weighted graphs [423].

Finally, let us emphasize the following significant research directions:

(1) development of special knowledge based systems to design of the change system;

(2) development of tools for the presentation of complex systems, and improvement processes;

(3) investigation of system transformation trajectories with feedback;

(4) study of corresponding scheduling problems;

(5) development of special knowledge based systems to evaluate the change system while taking into account estimates of the schedule change system; and

(6) application of examined issues in engineering education.

6 SPECIFIC ESTIMATES OF SYSTEM ELEMENTS

Strategic design and technological forecasting are significant parts of joint engineering and management activities. Technology forecasting techniques can be classified in four major categories ([2], [17], [22], [234], [509], etc.):

1. Search and evaluation of information.

2. Technical trend exploration (projective).

3. Normative or goal oriented (e.g., generation and selection of relevant technologies).

4. Integrative (cross-impact analysis, scenarios, mathematical modeling, etc.).

Many approaches are applied for above-mentioned problems of forecasting: time series data analysis; expert judgement; decision theory (e.g., decision trees); multicriteria analysis, etc.

Strategic system design/planning is often based on multi-stage multicriteria decision making, decision theory while taking into account uncertainty. For example, Cook and Kress proposed selecting a new technology decision using the following: series of technological decisions, decision trees, multiple alternatives, multiple criteria, and uncertainty [91]. Bard and Feinberg apply multi-phase methodology for technology selection [29].

Our composition approaches can be used as components of normative and integrative above-mentioned techniques of forecasting. In this chapter, we start to take into account uncertainty of system characteristics (DA's, Ins). It is readily seen that our problems under uncertainty are oriented to forecasting and strategic planning. Four approaches are used for the representation of uncertainty ([125], [179], [250], [251], etc.):

(1) probability theory ([146], etc.);

(2) Dempster-Shafer evidence theory ([458], etc.);

(3) fuzzy set theory ([34], [250], [537], [547], etc.); and

(4) possibility theory ([125], [236], [537], [548], etc.).

Here we consider only some specific estimates of DA's and Ins as follows:

(a) functions of time,

(b) probabilistic estimates, and

(c) fuzzy environment.

Note that Prof. G. Rzevski (UK) has described close design problems during his presentations at Intl. conf. "Information Technology in Design EWITD'96" (Moscow, Russia).

6.1 TIME DEPENDENCE FOR ESTIMATES

Here let us consider time dependence of system description components (external requirements, composite DA's, elements) of time. Table 6.1 depicts main functions of time and basic sources of changes.

In previous section, we have examined external system changes as changes of composite DA's on the basis of external improvement actions. Now let us analyze internal system changes of elements: (i) Ins; and (ii) DA's. We will consider estimates of the elements as functions of time. Here μ-stage ($\mu = 4$) time interval is assumed, and the estimates are evaluated on ordinal scales (Fig. 6.1):

(a) for priority of DA i: $r_i(t)$, and

(b) for compatibility of DA's i_1 and i_2: $w_{i_1 i_2}(t)$.

It is reasonable to investigate some basic kinds of functions (increasing / improvement function, decreasing / aggravation function, etc.). Thus we get vector-like estimates:

$r_i(t) = (r_i^1, ..., r_i^\xi, ..., r_i^\mu)$,

$w_{i_1 i_2}(t) = (w_{i_1 i_2}^1, ..., w_{i_1 i_2}^\xi, ..., w_{i_1 i_2}^\mu)$,

where ξ is the number of a time stage ($\xi = 1, ..., \mu$).

Table 6.1. Sources of time dependence

Functions of time	Source of changes	
	Internal	External
1. Composite DA's (trajectories at system excellence space)	Self-improvement	Improvement actions on the basis of external resources
2. Elements		
2.1 DA's:		
(a) Set of DA's		Improvement actions on the basis of external resources
(b) estimates	Self-improvement (self-education, etc.)	Improvement actions on the basis of external resources
2.2. Ins (estimates)	Self-improvement (self-education, etc.)	Improvement actions on the basis of external resources
3. External requirements	Internal conflicts (correspondence to constraints, etc.)	Changes of external environment

Clearly, constraints for DA's and Ins correspond to inequalities, and constraints for composite DA's correspond to preference relations.

In the main these functions can be considered as a result of expert judgement (a forecast). Also, note that internal changes of the estimates above may be based on self-improvement processes, for example, as follows:

(1) increasing of professional skills (personnel, pair of personnel and tools); and

(2) improvement of compatibility for mechanical system parts or chemical system components.

We can point out the following basic possibilities of solving process:

1. To solve a problem (design, improvement, etc.) for a certain time stage. In this case, our basic approach can be used.

2. To solve a problem (design, improvement, etc.) for several time stages.

In the second case, the following two ways can be considered:

Strategy 1. To map (to aggregate) multi-stage vector estimates into a resultant ordinal estimate (for example, on the basis of multicriteria ranking).

Strategy 2. To solve the problem for each time stage, and to search for a composite decision that consists of local decisions for each time stage.

Let us examine *strategy 1* on the basis of an example [311]. We assume that a human-computer system (HCS) consists of four main parts as follows:
(a) goals (tasks);
(b) operational part;
(c) factual part (information as data and/or knowledge); and
(d) human (user's) part.
Let us examine the following:
(1) basic design for the only one stage;
(2) design for the multi-stage case; and
(3) kinds of constraints.

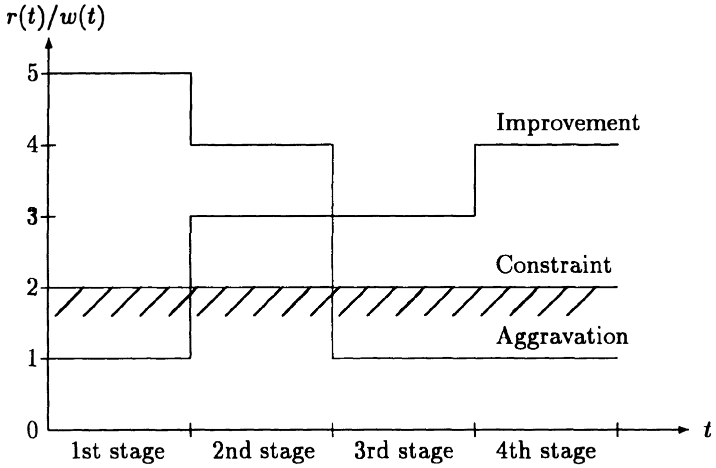

Fig. 6.1. Example of priority/compatibility functions

6.1.1 Basic Example

Our system morphology is depicted in Fig. 6.2. Basic DA's are presented in Table 6.2. As an input we consider a certain task, for example, T_2 (we reject T_1 and T_3). In addition, we reject U_3, and add aggregate DA's as follows: $U_4 = U_1 \& U_2$, $O_4 = O_1 \& O_3$. An example of compatibility is presented in Table 6.3.

Resultant composite DA's are the following (two Pareto-effective points):
(a) $N_1 = (3;3,1,0)$: $S_1 = T_2 * I_2 * O_4 * U_2$;
(b) $N_2 = (2;4,0,0)$: $S_2 = T_2 * I_3 * O_2 * U_2$, $S_3 = T_2 * I_3 * O_4 * U_2$, $S_4 = T_2 * I_3 * O_4 * U_4$, $S_5 = T_2 * I_3 * O_2 * U_4$.

Fig. 6.3 illustrates the lattice of system excellence and Pareto-effective points above.

Now we examine bottlenecks for the following composite DA's:
(1) S_1, improvement of I_2 (obtaining the best decision); and
(2) S_4, improvement of (O_4, U_4) (obtaining the best decision).

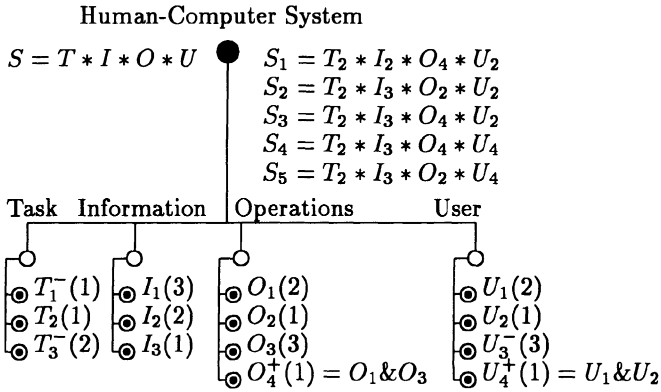

Fig. 6.2. Structure of HCS (hypothetical priorities of DA's are shown in brackets)

Table 6.2. Basic DA's

DA's	Description
I_1	Basic information
I_2	Results of an additional searching
I_3	Results of an information design
O_1	Basic techniques
O_2	Additional selected techniques
O_3	New designed techniques
U_1	Novice
U_2	Trained user
U_3	Expert

Table 6.3. Compatibility

	I_1	I_2	I_3	O_1	O_2	O_3	O_4	U_1	U_2	U_3	U_4
T_2	1	3	3	1	2	3	3	1	3	3	3
I_1				2	3	3	3	2	1	1	1
I_2				2	2	2	3	1	3	1	3
I_3				2	3	3	3	3	3	3	3
O_1								3	2	1	2
O_2								2	2	3	2
O_3								1	2	3	2
O_4								2	3	3	2

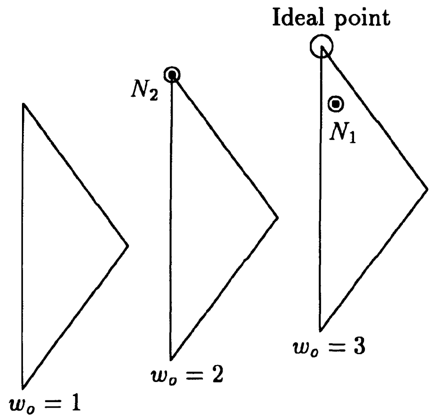

Fig. 6.3. Lattice of system excellence

6.1.2 Multi-stage Vector Estimates

Here we consider vector estimates. It is assumed that the following compatibility will be self-improved: between human (user's) part and other ones (on the basis of accumulated skills).

Table 6.4 contains 4-component vector estimates and resultant estimates for compatibility between U and other DA's.

In this case, resultant composite DA's are the following:

(a) $N_1 = (3;3,1,0)$: $S_1 = T_2 * I_2 * O_4 * U_2$, $S_2 = T_2 * I_2 * O_4 * U_4$;

(b) $N_2 = (2;4,0,0)$: $S_3 = T_2 * I_3 * O_2 * U_4$, $S_4 = T_2 * I_3 * O_4 * U_2$, $S_5 = T_2 * I_3 * O_4 * U_4$.

Let us remark that the set of Pareto-effective DA's is changed (see composite decisions for basic example in previous section). Two Pareto-effective points are the same.

Table 6.4. Four-stage compatibility for U

	U_1	U_2	U_3	U_4
T_2	$1,1,2,2/2$	$3,3,3,3/3$	$3,3,3,3/3$	$3,3,3,3/3$
I_1	$2,2,2,2/2$	$1,2,2,1/1$	$1,2,2,3/2$	$1,2,2,1/1$
I_2	$1,2,2,2/2$	$3,3,3,3/3$	$1,2,3,3/3$	$3,3,3,3/3$
I_3	$3,2,1,1/1$	$3,3,2,2/2$	$3,3,3,3/3$	$3,3,2,2/2$
O_1	$3,3,3,3/3$	$2,3,2,2/2$	$1,2,3,3/3$	$2,3,2,2/2$
O_2	$2,2,2,2/2$	$2,1,1,1/1$	$3,3,3,3/3$	$2,2,2,2/2$
O_3	$1,2,2,2/2$	$2,1,1,1/1$	$3,3,3,3/3$	$2,1,1,1/1$
O_4	$2,2,2,2/2$	$3,3,3,3/3$	$3,3,3,3/3$	$2,3,3,3/3$

6.1.3 Constraints

Constraints for the functions of time $r_i(t)$, and $w_{i_1 i_2}(t)$ have an important applied entity. Table 6.5 presents basic kinds of constraints and approaches to solve corresponding problems. Here the following constraints are considered:

(i) r^* corresponds to a numerical constraint of priorities for DA's,

(ii) w^* corresponds to a numerical constraint of compatibility Ins, and

(iii) N^* corresponds to a vector constraint of system excellence for composite DA's.

In many cases, the use of the constraints provides formulating and solving of real applied problems. We can use constraints to construct and to manage required models and solving schemes. For example, the following two types of system improvements on the basis of multi-stage improvement processes may be considered: (a) improvement trajectory from an initial state to a resultant one; and (b) improvement trajectory from an initial state to a resultant one in accordance with special requirements to intermediate states. In the second case, we can specify the following requirement (i.e., specific constraints to intermediate composite DA's):

Designed system has to work at each intermediate stage.

Similar requirements are significant for reengineering activities (e.g., for teams, for organizations, for manufacturing systems, and for computer systems).

Note it is reasonable to examine real world situations while taking into account the following aspects [126]:
(a) "softness" of constraints;
(b) some constraint have priorities over others; and
(c) constraints may involve uncertain parameters.

Table 6.5. Basic Kinds of Constraints

Type of Constraint	Entity	Approach to Solve
1. Left side for DA's/Ins ($\xi = 1$, $r_i^1 > r^*$ or $w_{i_1 i2}^1 > w^*$)	Requirements to start situation	Rejection of inadmissible DA's/Ins
2. Right side for DA's/Ins ($\xi = \mu$, $r_i^\mu > r^*$ or $w_{i_1 i2}^\mu > w^*$)	Requirements to resultant situation	Rejection of inadmissible DA's/Ins
3. Multi-stage for DA's/Ins ($\forall \xi = 1, ..., \mu$ $r_i^\xi > r^*$ or $w_{i_1 i2}^\xi > w^*$)	Requirements to multi-stage process	1.Rejection of inadmissible DA's/Ins 2.Mapping of vector estimates into integrated ordinal scales
4. Left side for composite DA's ($\xi = 1$, i.e., $N(S^{\xi=1}) \succ N^*$)	Requirements to start situation	Taking into account additional constraints to morphological clique at the 1st stage
5. Right side for composite DA's ($\xi = \mu$, i.e., $N(S^{\xi=\mu}) \succ N^*$)	Requirements to resultant situation	Taking into account additional constraints to morphological clique at the last stage
6. For composite DA's at intermediate stages ($1 < \xi^+ < \mu$ $N(S^{\xi^+}) \succ N^*$)	Requirements to intermediate situations	Taking into account additional constraints to morphological clique at intermediate stages
7. Multi-stage for composite DA's ($\forall \xi$ $N(S^\xi) \succ N^*$)	Requirements to multi-stage process	Taking into account additional constraints to morphological clique at all stages

6.2 PROBABILISTIC ESTIMATES

Often it is necessary to consider probabilistic estimates of DA's and Ins. In these cases, problems are close to traditional approaches for uncertainty (e.g., stochastic programming, fuzzy sets). The first approach to solve problems with probabilistic estimates is based on an aggregation of the estimates, for example:

(a) mathematical expectations;

(b) integrated values on the basis of thresholds (constraints);

(c) integrated composite values on the basis of complex aggregation techniques.

In this section, we examine four important realistic design situations:

(1) probabilistic estimates for DA's/Ins;

(2) probabilistic appearance of technological innovations (i.e., DA's) at time stages;

(3) probability density of DA's/Ins for one-stage design;

(4) probability density of DA's/Ins for multi-stage design.

6.2.1 Probabilistic Estimates of Elements

In the case of probabilistic estimates of DA's/Ins, we have to study composition problems, when elements (vertices and arcs) of the initial morphological graph have probabilistic character. Thus the problem is:

Find a morphological clique in probabilistic morphological graph.

An engineering approach to the problem is the following: to include probabilistic estimates into multicriteria estimates of DA's/Ins, to map the extended multicriteria estimates into ordinal scales, and to solve a basic problem. Clearly, in this solving scheme the step ***mapping of multicriteria estimates*** is crucial one. Also, this operation may include the use of constraints.

Now let us consider an example which is oriented to synthesis of a composite project for a company [302]. Structure of the project is depicted in Fig. 6.4. Table 6.6 presents criteria for DA's.

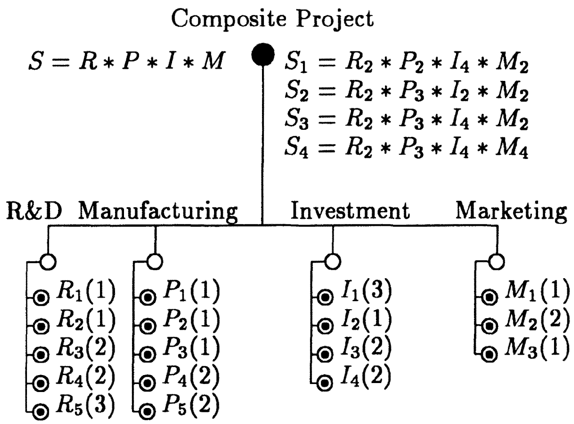

Fig. 6.4. Structure of system (priorities of DA's are shown in brackets)

Table 6.6. Criteria

Criteria	Weights			
	R	P	I	M
1.Expenditure/cost (-)	3	5		
2.Resultant quality/preliminary results	5	5		
3.Possible complexity of joint business (-)	5	5	5	5
4.Required quality of business-plan (-)			3	
5.Possible volume of market				3
6.Possible extension of market				5

Table 6.7 involves the following: DA's, their estimates on criteria, resultant priorities of DA's (r), probabilistic estimates of DA's (θ), and resultant priorities while taking into account probabilistic estimates (r_θ). Table 6.8 contains compatibility.

Table 6.7. DA's and estimates

DA's	Criteria						r	θ	r_θ
	1	2	3	4	5	6			
R_1 Independent research	5	3	0				2	1.00	2
R_2 Joint research with Japanese company	2	4	5				2	0.60	3
R_3 Joint research with Korean company	2	3	5				3	0.70	3
R_4 Joint research with German company	2	5	2				1	0.80	1
R_5 Joint research with Japanese and German companies	1	5	3				1	0.48	2
P_1 Independent manufacturing	1	3	0				1	0.70	2
P_2 Manufacturing joint with German company	4	5	2				2	0.90	1
P_3 Joint manufacturing with Brazilian company	2	3	3				2	0.60	2
P_4 Joint manufacturing with Japanese company	5	5	5				2	0.50	3
P_5 Joint manufacturing with American company	5	4	4				3	0.50	3
I_1 Self-investment			0	1			1	0.70	1
I_2 European financial groups			3	3			3	0.80	2
I_3 American banks			3	4			3	0.60	3
I_4 Arabian investors			4	2			3	0.60	3
M_1 South America			4		4	2	2	0.90	1
M_2 Europe			2		3	0	3	0.80	3
M_3 South-East Asia			3		4	3	1	0.75	2

Table 6.8. Compatibility

	P_1	P_2	P_3	P_4	P_5	I_1	I_2	I_3	I_4	M_1	M_2	M_3
R_1	1	1	1	0	0	3	3	3	3	0	1	0
R_2	0	0	2	3	1	0	1	1	1	0	1	3
R_3	2	0	1	0	0	0	1	1	2	1	1	3
R_4	1	3	3	1	1	1	3	1	3	3	3	2
R_5	1	3	3	3	1	0	3	3	3	1	3	3
P_1						3	3	3	3	2	2	2
P_2						0	3	2	3	3	3	2
P_3						0	3	3	3	3	0	0
P_4						0	2	2	3	2	2	3
P_5						0	1	3	3	3	1	1
I_1										1	1	2
I_2										2	3	2
I_3										3	1	2
I_4										2	2	2

Now let us consider the following three schemes of design:

Scheme 1. Basic design problem. In this case, Pareto-effective composite DA's are the following:

(a) $N = (3;1,1,2)$: $S_1^{'} = R_5 * P_2 * I_2 * M_2$;

(b) $N = (2;2,1,1)$: $S_2^{1} = R_5 * P_2 * I_2 * M_3$, $S_3^{1} = R_5 * P_2 * I_3 * M_3$; and

(c) $N = (1;4,0,0)$: $S_4^{1} = R_4 * P_1 * I_1 * M_3$.

Scheme 2. Design with taking into account probabilistic estimates of DA's (probabilistic estimates are included into multicriteria estimates of DAs). Here Pareto-effective composite DA's are the following:

(a) $N = (3;2,1,1)$: $S_1^{''} = R_4 * P_2 * I_2 * M_2$; and

(b) $N = (2;2,2,0)$: $S_2^{''} = R_4 * P_2 * I_2 * M_3$.

Scheme 3. Probabilistic analysis of resultant composite DA's obtained for the basic design problem (computation of probabilistic estimates for composite decisions, and selection of the best ones). Computation of probabilistic estimates for composite DA's may be based on the following simple formulae:

$$\theta(S) = \prod_{e \in S} \theta_e,$$

where e is a component of S. Here we examine Pareto-effective decisions for *scheme 1*: $\theta(S_1^{'}) = 0.27648$, $\theta(S_2^{'}) = 0.25920$, $\theta(S_3^{'}) = 0.19440$, $\theta(S_4^{'}) = 0.27440$. Obviously, in this case, it is reasonable to select two composite decisions $S_1^{'}$ and $S_4^{'}$.

Finally, we can analyze the following obtained decisions:
scheme 2: S_1'', S_2''; and
scheme 3: S_1', S_4'.
Note that these decisions may be considered as input information for *Decision Maker*.

6.2.2 Probabilistic Appearance of Alternatives

Now we examine the following design multi-stage problem. DA's for system parts may appear with probabilities for each time stage. Basic information for our 4-stage example is presented in Fig. 6.5, Tables 6.9, 6.10, 6.11, 6.12, and 6.13.

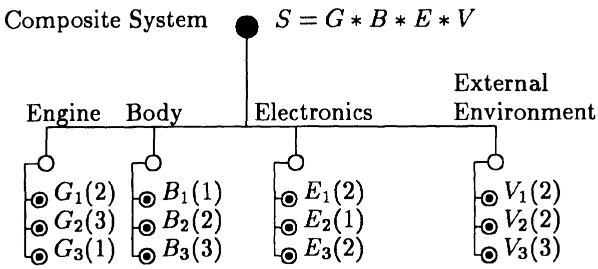

Fig. 6.5. Structure of system (priorities of DA's are shown in brackets)

Table 6.9. Compatibility (stage 1)

	B_1	B_2	B_3	E_1	E_2	E_3	V_1	V_2	V_3
G_1	0	1	1	1	1	1	1	1	1
G_2	1	1	1	0	2	1	1	3	1
G_3	0	1	0	1	0	2	0	0	1
B_1				1	2	1	2	1	2
B_2				3	2	1	2	2	1
B_3				1	3	1	1	1	1
E_1							2	1	1
E_2							0	0	1
E_3							1	2	3

Table 6.10. Compatibility (stage 2)

	B_1	B_2	B_3	E_1	E_2	E_3	V_1	V_2	V_3
G_1	0	1	1	2	1	1	1	1	1
G_2	1	1	1	0	2	1	1	3	1
G_3	0	1	0	2	1	2	1	2	1
B_1				1	2	1	2	1	2
B_2				3	2	1	2	2	1
B_3				1	3	2	1	1	1
E_1							2	1	1
E_2							2	2	2
E_3							1	2	3

Table 6.11. Compatibility (stage 3)

	B_1	B_2	B_3	E_1	E_2	E_3	V_1	V_2	V_3
G_1	0	1	2	3	1	1	1	2	2
G_2	2	1	1	0	3	1	1	3	1
G_3	2	2	1	2	1	3	2	2	1
B_1				2	2	2	2	2	2
B_2				3	2	1	3	3	1
B_3				1	3	2	1	2	2
E_1							2	1	1
E_2							3	2	2
E_3							2	3	3

Table 6.12. Compatibility (stage 4)

	B_1	B_2	B_3	E_1	E_2	E_3	V_1	V_2	V_3
G_1	1	3	3	3	3	2	2	3	2
G_2	2	2	1	0	3	1	1	3	2
G_3	3	2	2	2	3	3	2	3	1
B_1				3	3	2	2	3	3
B_2				3	3	2	3	3	1
B_3				1	3	2	1	2	2
E_1							3	2	1
E_2							3	3	2
E_3							3	3	3

Table 6.13. Probability of appearance for DA's

DA's	Stage 1	Stage 2	Stage 3	Stage 4
G_1	0.60	0.80	1.00	1.00
G_2	0.80	0.85	0.90	1.00
G_3	0.10	0.40	0.60	0.70
B_1	0.20	0.60	0.80	1.00
B_2	0.50	0.70	1.00	1.00
B_3	1.00	1.00	1.00	1.00
E_1	0.90	1.00	1.00	1.00
E_2	0.40	0.50	0.60	0.80
E_3	0.70	0.90	1.00	1.00
V_1	1.00	1.00	1.00	1.00
V_2	0.60	0.70	0.90	1.00
V_3	0.80	1.00	1.00	1.00

Here we have to take into account probability of the appearance for DA's. This situation is realistic for long-term design/planning, forecasting, investment. We assume that priorities of DA's are not dependent on time (Fig. 6.5).

In general, we can examine the following two main approaches:

A. *Design and selection of a composite alternatives*:

To design a composite DA's, and to select a decision at a stage with sufficient quality of the decision and its probability. Thus, in this case, we can obtain a set of composite DA's, and their estimates: (i) quality N, (ii) a number of stage (time interval), and (iii) probabilistic estimates of appearance. As a result it is possible to examine a final multicriteria ranking to select a resultant decision at the certain stage.

B. *Design of a trajectory of composite alternatives*:

To design a trajectory consisting of composite DA's for each stage. In this case, it is necessary to take into account an additional information: a cost to move from an initial composite decision to another one. Thus we should like to select the following:

(a) trajectories consisting of the same composite DA's at each stage; and

(b) trajectories, in which the movement from a composite decision at a stage to another one at a next stage is very cheap.

In addition, it is reasonable to study multi-stage vector priorities of DA's. In this case, to design a composite DA's at a stage it is possible to use two approaches which were considered in previous section (i.e., inclusion of probabilistic estimates into multicriteria estimates, or an additional probabilistic analysis of resultant composite DA's).

Note that a process of trajectories construction has to be based on techniques that were examined in previous chapter (e.g., design of a change system, etc.).

Table 6.14 presents resultant composite DA's (top index corresponds to a stage number).

Table 6.14. Resultant composite DA's

Composite DA's	N	Stage	Probability $\theta(S)$
$S^1_1 = G_3 * B_2 * E_3 * V_3$	$(1;1,2,1)$	1	0.02800
$S^1_2 = G_3 * B_2 * E_1 * V_3$	$(1;1,2,1)$	1	0.03600
$S^1_3 = G_2 * B_1 * E_3 * V_2$	$(1;1,2,1)$	1	0.06720
$S^1_4 = G_1 * B_2 * E_2 * V_3$	$(1;1,2,1)$	1	0.09600
$S^1_5 = G_1 * B_3 * E_2 * V_1$	$(1;1,2,1)$	1	0.24000
$S^1_6 = G_1 * B_3 * E_2 * V_2$	$(1;1,2,1)$	1	0.14400
$S^1_7 = G_2 * B_2 * E_2 * V_1$	$(1;1,2,1)$	1	0.16000
$S^1_8 = G_2 * B_2 * E_2 * V_2$	$(1;1,2,1)$	1	0.09600
$S^2_1 = G_3 * B_2 * E_2 * V_1$	$(2;2,2,0)$	2	0.14000
$S^2_2 = G_3 * B_2 * E_2 * V_2$	$(2;2,2,0)$	2	0.09800
$S^3_1 = G_3 * B_1 * E_2 * V_1$	$(2;3,1,0)$	3	0.28800
$S^3_2 = G_3 * B_1 * E_2 * V_2$	$(2;3,1,0)$	3	0.28800
$S^4_1 = G_3 * B_1 * E_2 * V_2$	$(3;3,1,0)$	4	0.56000

At each stage it is reasonable to select reliable composite DA's, for example, S^1_5, S^1_6, S^1_7, S^2_1, S^3_1, S^3_2, S^4_1. Further, this set of DA's is a base for multicriteria ranking and selection of a resultant decision. Fig. 6.6 depicts corresponding Pareto-effective points:

$$N^1 = (1;1,2,1),\quad N^2 = (2;2,2,0),\quad N^3 = (2;3,1,0),\quad N^4 = (3;3,1,0).$$

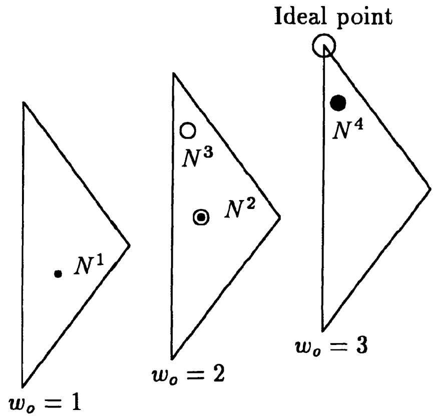

Fig. 6.6. Lattice of system excellence

In the case of the trajectory approach (**B**), it is reasonable to use for the design of trajectories above morphological clique problem as for the design of multi-period strategy (see section 3.6). Fig. 6.7 depicts an initial obtained system for composing of trajectories.

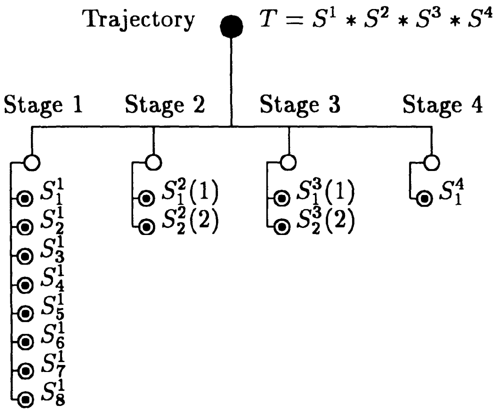

Fig. 6.7. Morphology of trajectory

In this case, estimates of compatibility are based on a cost of transforming (change) of an alternative into another. As a result, we can consider, for example, the following trajectories:

$$\alpha_1 = (S^1_8 \Rightarrow (G_2 \rightarrow G_3)) \Longrightarrow (S^2_2 \Rightarrow (B_2 \rightarrow B_1)) \Longrightarrow S^3_2 \Longrightarrow S^4_1,$$

$$\alpha_2 = (S^1_5 \Rightarrow ((G_1 \rightarrow G_3)\&(B_3 \rightarrow B_2))) \Longrightarrow$$
$$\Longrightarrow (S^2_1 \Rightarrow ((V_1 \rightarrow V_2)\&(B_2 \rightarrow B_1))) \Longrightarrow S^3_2 \Longrightarrow S^4_1,$$

$$\alpha_3 = (S^1_6 \Rightarrow ((G_1 \rightarrow G_3)\&(B_3 \rightarrow B_2))) \Longrightarrow (S^2_1 \Rightarrow (B_2 \rightarrow B_1)) \Longrightarrow S^3_2 \Longrightarrow S^4_1.$$

Obviously, *Decision Maker* is the most important participant at the final above-mentioned phases.

6.3 FUZZY ESTIMATES

6.3.1 Notations and Basic Example

Here we consider fuzzy estimates for DA's or/and Ins. Notations are the following:

ι is an index corresponding to a design alternative;

$\mu^r(\iota)$ is a membership function of the priority $r(\iota)$, we consider the following set: $\{\mu^r_l(\iota),\ l = 1, ..., 3\}$; and

$\mu^w(\iota)$ is a membership function of compatibility $w(\iota)$, we use the following set: $\{\mu^w_k(\iota),\ k = 0, ..., 3\}$.

Let $\{\mu^w_k(\iota_1, \iota_2)\}$ be the following vector:
$(\mu^w_3(\iota_1, \iota_2), \mu^w_2(\iota_1, \iota_2), \mu^w_1(\iota_1, \iota_2), \mu^w_0(\iota_1, \iota_2))$.

Now let $r^a(\iota)$ and $w^a(\iota_1, \iota_2)$ be aggregated estimates for a design alternative ι, and for a pair of design alternatives (ι_1, ι_2), accordingly.

Our basic example is depicted in Fig. 6.8. Table 6.15 contains normalized fuzzy priorities $\{\mu^r_l(\iota)\}$, and realistic aggregated priorities $\{r^a(\iota)\}$. Table 6.16 contains normalized fuzzy compatibility $\{\mu^w_k(\iota_1, \iota_2)\}$, and Table 6.17 presents realistic aggregated compatibility $\{w^a(\iota_1, \iota_2)\}$.

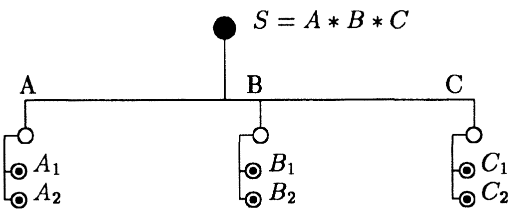

Fig. 6.8. Example of Composite System

Table 6.15. Fuzzy priorities

DA's ι	$\mu_1^r(\iota)$	$\mu_2^r(\iota)$	$\mu_3^r(\iota)$	$r^a(\iota)$
A_1	1.00	0.00	0.00	1
A_2	0.00	0.05	0.95	3
B_1	0.15	0.65	0.20	2
B_2	0.85	0.15	0.00	1
C_1	1.00	0.00	0.00	1
C_2	0.00	1.00	0.00	2

Table 6.16. Fuzzy compatibility

	B_1	B_2	C_1	C_2
A_1	0.5; 0.2; 0.3; 0.0	0.0; 0.3; 0.4; 0.3	0.0; 0.4; 0.5; 0.1	0.2; 0.4; 0.3; 0.1
A_2	0.1; 0.2; 0.1; 0.6	0.7; 0.3; 0.0; 0.0	0.4; 0.4; 0.2; 0.0	0.0; 0.7; 0.3; 0.0
B_1			0.6; 0.3; 0.1; 0.0	0.0; 0.5; 0.5; 0.0
B_2			0.0; 0.1; 0.3; 0.6	0.2; 0.5; 0.3; 0.0

Table 6.17. Aggregated compatibility

	B_1	B_2	C_1	C_2
A_1	3	1	1	2
A_2	0	3	2	2
B_1			3	1
B_2			0	2

Evidently, that on the basis of fuzzy priorities we can obtain the following relationship on the set of DA's:

(a) $A_1 \succeq A_2$;

(b) $B_1 \succeq B_2$; and

(c) C_1 and C_2 are non-comparable.

6.3.2 Generalized Glance

Now let us study the following four cases:

Case 1: deterministic (aggregated) estimates of priorities for DA's $\{p^a(\iota)\}$, and deterministic (aggregated) estimates of compatibility for Ins, and $\{w^a(\iota_1, \iota_2)\}$ (a basic case):.

Case 2: estimates of DA's are aggregated (deterministic) $\{r^a(\iota)\}$, and estimates of Ins are fuzzy $\{\mu_k^w(\iota_1,\iota_2)\}$, $\forall(\iota_1,\iota_2)$.

Case 3: estimates of DA's are fuzzy $\{\mu_l^r(\iota)\}$ $\forall\iota$, and estimates of Ins are aggregated (deterministic) $\{w^a(\iota_1,\iota_2)\}$, $\forall(\iota_1,\iota_2)$.

Case 4: estimates of DA's are fuzzy $\{\mu_l^r(\iota)\}$ $\forall\iota$, and estimates of Ins are fuzzy $\{\mu_k^w(\iota_1,\iota_2)\}$, $\forall(\iota_1,\iota_2)$.

Fig. 6.9 illustrates the cases above (top index of composite DA's corresponds to the case).

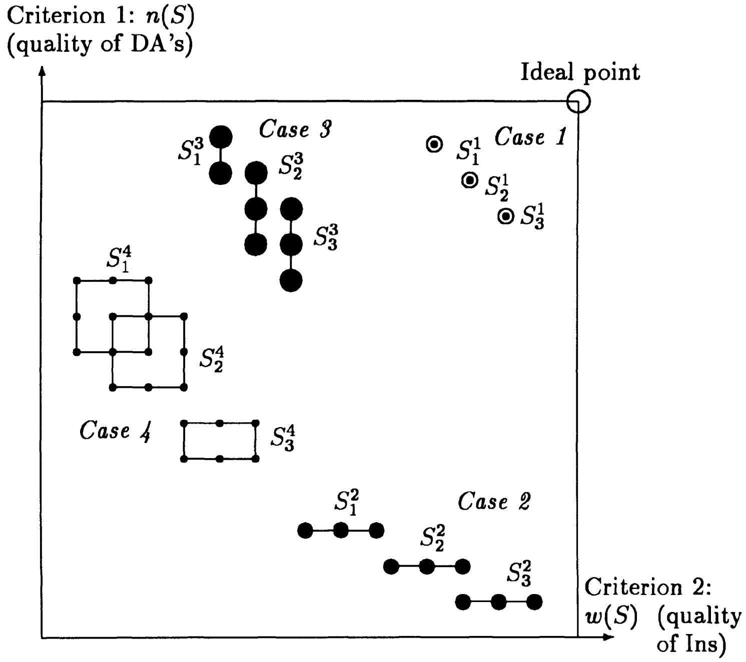

Fig. 6.9. Illustrative space of system excellence and composite DA's

Clearly, that main method is based on two stages:

(1) generation of feasible composite decision; and

(2) selection of Pareto-effective decisions.

Unfortunately, it is reasonable to point out the following two significant features of our synthesis problem with fuzzy estimates:

(a) complexity of corresponding combinatorial problems is increasing because a number of analyzed composite decisions is more than in deterministic case; and

(b) it is necessary to construct a preference rule to select the best fuzzy decisions.

Now we will examine several numerical examples to illustrate four above-mentioned cases.

6.3.3 *Deterministic Case*

In the *case 1*, we obtain the following two Pareto-effective decisions:

(1) $S_1^1 = A_2 * B_2 * C_2$, $N(S_1^1) = (2; 1, 2, 0)$; and

(2) $S_2^1 = A_1 * B_1 * C_1$, $N(S_2^1) = (1; 2, 1, 0)$.

Fig. 6.10 illustrates these composite DA's.

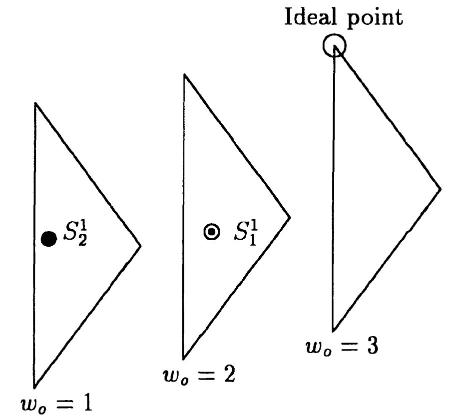

Fig. 6.10. Lattice of system excellence (*case 1*: without uncertainty)

6.3.4 *Fuzzy Compatibility*

Now we illustrate *case 2*. Let $S_1^2 = S_1^1$, and $S_2^2 = S_2^1$. We obtain 12 possible situations or elementary compositions (Table 6.18) for S_1^2. Note that we consider probability of an elementary composition as the following product:

$\mu^w(A_2, B_2)\ \mu^w(A_2, C_2)\ \mu^w(B_2, C_2)$.

Table 6.18. Basic compositions for S_1^2 and its probability

Number	$w(A_2, B_2)$	$w(B_2, C_2)$	$w(A_2, C_2)$	Probability
1	2	1	1	0.027
2	3	1	1	0.063
3	2	2	1	0.045
4	3	2	1	0.105
5	2	3	1	0.018
6	3	3	1	0.042
7	2	1	2	0.063
8	3	1	2	0.147
9	2	2	2	0.105
10	3	2	2	0.245
11	2	3	2	0.042
12	3	3	2	0.098

It is follows easily that $N(S_1^2) = (0, 0.49, 0.51, 0; 1, 2, 0)$.
In the same way, we obtain the following for S_2^2 (Table 6.19):
$N(S_2^2) = (0.1, 0.252, 0.648, 0; 2, 1, 0)$.

Table 6.19. Basic compositions for S_2^2 and its probability

Number	$w(A_1, B_1)$	$w(B_1, C_1)$	$w(A_1, C_1)$	Probability
1	1	1	1	0.015
2	1	1	2	0.012
3	1	2	1	0.045
4	1	2	2	0.036
5	1	3	1	0.090
6	1	3	2	0.072
7	2	1	1	0.010
8	2	1	2	0.008
9	2	2	1	0.030
10	2	2	2	0.024
11	2	3	1	0.060
12	2	3	2	0.048
13	3	1	1	0.025
14	3	1	2	0.020
15	3	2	1	0.075
16	3	2	2	0.060
17	3	3	1	0.150
18	3	3	2	0.120

As a result, we have to examine a set of points at system excellence space for each composite decision with taking into account a membership functions (Fig. 6.11).

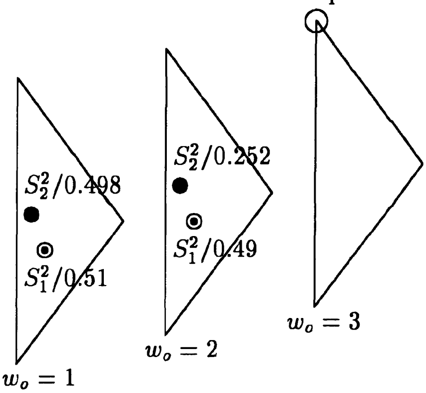

Fig. 6.11. Lattice of system excellence (*case 2*): fuzzy compatibility

Finally, it is necessary to analyze all feasible composite DA's, and to reveal Pareto-effective decision by the selected above-mentioned preference rule.

6.3.5 Fuzzy Priorities

Here let us construct fuzzy decision sets. We examine the following basic composite DA's: $S_1^3 = S_1^1 = A_2 * B_2 * C_2$ and $S_2^3 = S_2^1 = A_1 * B_1 * C_1$.

Using Table 6.15, we get the following situations for S_1^3:

(a) S_{11}^3, $n(S_{11}^3) = (1, 2, 0)$ and a resultant value of membership function (as a product of corresponding values for included DA's) is equal to 0.0425;

(b) S_{12}^3, $n(S_{11}^3) = (0, 3, 0)$ and the resultant value of membership function is equal to 0.0075;

(c) S_{13}^3, $n(S_{11}^3) = (1, 1, 1)$ and the resultant value of membership function is equal to 0.8075;

(d) S_{14}^3, $n(S_{11}^3) = (0, 2, 1)$ and the resultant value of membership function is equal to 0.1425.

Analogically, we get the following situations for S_2^3:

(a) S_{21}^3, $n(S_{21}^3) = (3, 0, 0)$ and the resultant value of membership function is equal to 0.15;

(b) S_{22}^3, $n(S_{21}^3) = (2, 1, 0)$ and the resultant value of membership function is equal to 0.65;

(c) S_{23}^3, $n(S_{21}^3) = (2, 0, 1)$ and the resultant value of membership function is equal to 0.20.

Fig. 6.12 illustrates composite decisions with fuzzy priorities.

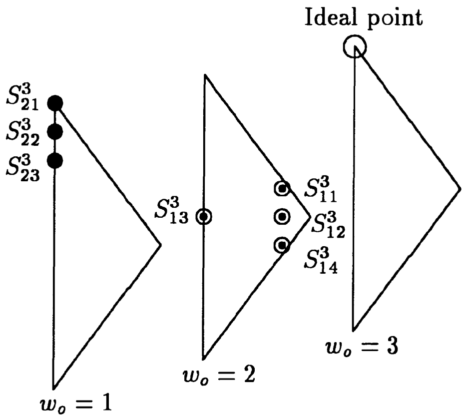

Fig. 6.12. Lattice of system excellence (*case 3*): fuzzy priorities

6.3.6 Fuzzy Compatibility and Priorities

It is easy to prove that *case 4* is a combination (a product) of *case 2* and *case 3*.

6.4 SUMMARY

Evidently, we have considered only several basic simple problems with time dependence and probability. It is reasonable to study more complex problems, for example, the following: (a) multi-stage problems with uncertainty for all elements of the problems; (b) problems with time dependence of probability; (c) multi-stage problems while taking into account human behavior.

In our opinion, the significance of considered in this chapter problems is increasing.

7 MORPHOLOGICAL METAHEURISTICS

The importance of combinatorial optimization problems is increasing, because many real-world applications are based on the following: (i) discrete variables; (ii) combinatorial structures, and corresponding combinatorial descriptions; and (iii) combinatorial nature of solving processes. Let us point out some basic monographs in this field:

I. Fundamentals of combinatorics and algorithms design:

(1.1) combinatorial analysis [194];

(1.2) problems and algorithms ([7], [358], [379], [388], [414], [425], etc.);

(1.3) complexity ([160], [240], etc.);

(1.4) networks and graphs ([24], [40], [68], [201], [200], [177], [383], [534], etc.);

(1.5) orders ([421], [422], etc.); and

(1.6) matroids ([280], etc.);

II. Some main problems and applications:
(2.1) scheduling ([25], [50],[51], [85], [88], [155], [286], [420], etc.);
(2.2) travelling salesman problem ([24], [281], [413], etc.);
(2.3) knapsack-like problems [347];
(2.4) location problem ([104], [197], [205], [360], [386], etc.);
(2.5) Steiner tree problem ([43], [160], [180], [452], [481], [535], etc.).
(2.6) networks design ([104], [123]; [152]; etc.);
(2.7) flow and commodity problems ([24], [160], [329], [358], [495], etc.);
(2.8) matching problems ([18], [24], [160], [209], etc.);
(2.9) classification and clustering ([363], [436], etc.);
(2.10) unification/standardization problems [176]; and
(2.11) combinatorial modeling in social sciences ([243], [362], [423], etc.).

In recent years, many complicated combinatorial problems have been studied and applied. Thus it is reasonable to use heuristics including the following approaches: simulated annealing algorithms; evolution strategies; evolutionary and genetic algorithms; stochastic search techniques; greedy algorithms; decomposition schemes; etc. ([13], [23], [175], [211], [239], [275], [354], [469], [518], etc.). Also, special techniques of artificial intelligence are based on Constraint Satisfaction Problems (CSPs) and solving schemes for them: backtracking and networks consistency algorithms ([109], [335], [377], etc.); heuristic revision [359]; and decomposition ([110], etc.); distributed constraint satisfaction search or multiagent approach ([219], [328], [334],[493], etc.).

Note that more and more attention is oriented to multi-criteria combinatorial optimization problems (for example, see survey [516], etc.).

This chapter addresses a possible implementation of our hierarchical morphological approach to some well-known combinatorial optimization problems. In this case, generalized solving scheme (our metaheuristic) consists of the following stages:

Stage 1. Hierarchical decomposition of initial problems into a set of subproblems. Note that often the decomposition may be based on partitioning of an initial graph (e.g., location problem, travelling salesman problem) or an initial set of elements as units/items, positions, and processors (e.g., scheduling, routing).

Stage 2. Generation of local decisions (DA's) for subproblems.

Stage 3. Assessment of interconnectivity of DA's for different subproblems.

Stage 4. Composing the best composite decision(s) of the problem at the higher hierarchical level.

In fact, this scheme realized dynamic programming. On the other hand, this scheme is close to genetic algorithms and evolutionary approach ([175],

[211], [354], etc.). In some cases, the scheme above may be a polynomial or ϵ-approximate polynomial one (e.g., the algorithm for knapsack problem with specific constraints in section 2.3 [289]). Mainly these hierarchical heuristics have to be proven on the basis of computing experiments. Here the following two main sources of errors in composite decisions can be pointed out:

(a) incorrect partitioning of the initial problem (*stage 1*); and

(b) incorrect composing of composite decisions (*stage 4*).

Note that subproblems have often small dimension and allow to apply exact enumerative algorithms (*stage 2*). On the other hand, evaluation of compatibility between DA's (*stage 3*) requires correct problem analysis and formulation.

In our opinion, usefulness of the hierarchical morphological approach maybe considered by the reasons as follows:

(a) understandable presentation of a combinatorial problem;

(b) partitioning of a problem into interconnected parts;

(c) possibility to apply concurrent (parallel) algorithms;

(d) possibility to include experts into solving process at various stages;

(e) a good algorithm for problems; and

(f) auxiliary techniques.

7.1 MULTI-ROUTE PROBLEM

7.1.1 Description

In our multi-route problem, we have to find a combination of a set of routes (trajectories). This is *point-to-point connection problem* that arises, for example, in circuit switching and VLSI design ([174], [321]).

In this case, we also can apply morphological clique problem. A set of the best decisions for each route can be examined as DA's, and compatibility of routes is based on the use of common resources (in nodes or arcs of a network). Fig. 7.1 depicts a network situation with an incompatibility of routes.

Note an analogical problem was proposed for the routing of packets in a network [283]:

Find paths for the packets and then schedule the transmission of the packets subject to the following condition: no two packets traverse the same edge simultaneously. The objective is to minimize the time by which the packages will deliver their final points.

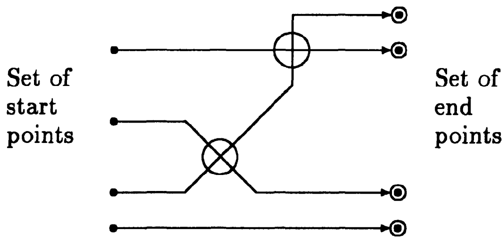

Fig. 7.1. Illustration for multi-route problem

7.1.2 *Formulation of Problem*

Let $G = (A, E)$ be a basic network, where A is a set of nodes, and E is a set of arcs. Let a set $X = \{x_i | i = 1, ..., m\} \in A$ corresponds to start points (***sources***), and a set $Y = \{y_i | i = 1, ..., m\} \in A$ corresponds to end (target) points (***destinations***). Thus we can investigate a set of node pairs $Z = \{z_i = (x_i, y_i) | i = 1, ..., m\}$.

We assume that $\forall z_i = (x_i, y_i)$ there exists a way (trajectory) from $\chi^i_1 = x_i$ to $\chi^i_{k_i} = y_i$ as follows: $\alpha_i =< x_i, ..., y_i >=< \chi^i_1, ..., \chi^i_\iota, ..., \chi^i_{k_i} >$ or each y_i is accessible from x_i, and $\Lambda = \{\alpha_i\}$ is a set of trajectories for pair z_i.

Also, we take into account a time parameter for each trajectory α_i at each used node $\tau_i(\chi^i_\iota)$, $\iota = 1, ..., k_i$ and at each arc: $\tau_i(\chi^i_\iota, \chi^i_{\iota+1})$, $\iota = 1, ..., k_i - 1$.

Clearly, that for each node χ^i_ι and for each arc $(\chi^i_\iota, \chi^i_{\iota+1})$ we can find (compute) a time interval for the use of the node χ^i_ι (analogically for arcs):

$$\Theta_i(\chi^i_\iota) = [\Theta'(\chi^i_\iota), \Theta''(\chi^i_\iota)],$$

where $\Theta'(\chi^i_\iota) = \sum_{l=1}^{l=\iota-1} \tau(\chi^i_l) + \sum_{l=1}^{l=\iota-1} \tau(\chi^i_l, \chi^i_{l+1})$, $\Theta''(\chi^i_\iota) = \Theta'(\chi^i_\iota) + \tau(\chi^i_\iota)$.

Let $\Theta^*(\alpha_i) = \Theta''(y_i = \chi^i_{k_i})$ denote a total time for trajectory α_i. Also, we will consider for pair z_i the best (***shortest***) trajectory α^+_i and the corresponding best total time $\Theta^*(\alpha^+_i)$, when we do not take into account compatibility with other trajectories.

Now compatibility for trajectories α_i and α_j by node χ_o is described as follows:

the trajectories are compatible if $|\Theta_i(\chi_o) \& \Theta_i(\chi_o)| = 0$, and incompatible ones otherwise.

Evidently, we can build for each pair z_i a set of the best trajectories (DA's) and use an ordinal scale to assess quality of the trajectories on the basis of total times:

(1) the 1st class: $\Theta^*(\alpha_i) - \Theta^*(\alpha^+) = 0$;

(2) the 2nd class: $\Theta^*(\alpha_i) - \Theta^*(\alpha^+) \leq \Delta\Theta,\ \Delta\Theta \geq 0$; and

(3) the 3rd class: $\Delta\Theta < \Theta^*(\alpha_i) - \Theta^*(\alpha^+) \leq 2\Delta\Theta$; etc.;

Note that building of the best trajectories for pair z_i mainly corresponds to polynomial (or pseudo-polynomial) problems ([160], [280], [414], etc.). Thus, ***basic multi-route problem*** (composition problem with binary compatibility) is:

Find a set of compatible trajectories $S = \{\alpha_i | i = 1, ..., m\}$ ***with the best total quality*** $n(S)$.

Evidently, we assume $\exists \alpha_i \ \forall z_i = (x_i, y_i)$.

Further, we consider the next realistic step as follows: a possibility for each trajectory at each node to wait (discrete time of wait $t(\chi_\iota^i) = \nu\delta$, $\nu = 0, 1, 2, ...,\ \delta \geq 0$). It is a way to resolve some conflicts at nodes. As a result, we use a compatibility scale: $\nu = 0, 1, ...$ ($\nu = 0$ corresponds to the best level). Also, the set of trajectories is increasing by the analysis of additional trajectories with waiting. Thus we get ***multi-route problem with waiting*** that corresponds to the basic composition problem. In this case, it is necessary to point out the following features of the problem:

1. Quality of global decision depends on waiting. Here we can assume that wait time intervals are very small.

2. Dependence of pair compatibility may exist. In other words, improvement of a pair compatibility may aggravate another pair compatibility.

7.1.3 Generalization

Now let us consider a classification of proposed problems. Table 7.1 presents several main situations.

Furthermore, it is reasonable to emphasize the following notes:

1. Multi-route problem may be considered for the case, when the following levels of intersection are examined:

(i) without intersection by nodes (i.e., binary compatibility), it is ***disjoint paths problem*** ([160], [393], etc.);

(ii) admissible intersection of trajectories by the only one node (i.e., with a scale of compatibility [0,1,many]); and

(iii) admissible intersection of trajectories by several nodes (i.e., with a scale of compatibility [0,1,2,...]).

2. Evidently, our problems are modifications of multicommodity flow problems ([140], [329], etc.)

3. On the other hand, in the case of several target points, when target point set is the same for all start point, we face a specific matching problem ([160], [388], etc.).

Now we point out some generalizations of our multi-route problem:

1. Resources for nodes γ_a $\forall a \in A$ and arcs ζ_e $\forall e \in E$. In this case, each trajectory α_i requires resources of corresponding nodes and arcs, and total times for trajectories depend on the use of resources.

2. Node conflicts for trajectories when the number of trajectories is more than 2.

3. Arc conflicts for trajectories.

Table 7.1. Types of multi-route problems

Situation	Kind of end set	
	1 target point for each trajectory	Several target points for each trajectory
1. Without intersection of trajectories by nodes	*Disjoint paths problem*	*Disjoint paths problem with target set*
2. Joint nodes are admissible for trajectories, but at different time	*Basic multi-route problem* (with binary compatibility)	*Basic multi-route problem with target set*
3. Waiting at nodes is admissible	*Multi-route problem with waiting*	*Multi-route problem with waiting and target set*

7.2 LOCATION/ASSIGNMENT PROBLEM

Location problem is very important for many applications, for example ([104], [176], [197], [358], [360], [379], [386], etc.):

(1) facility location in networks;
(2) location of information;
(3) location of computer tasks in computer networks;
(4) design of circuits;
(5) design of standards; and
(6) architectural design; etc.

Over 50 representative problems in location research (objectives, decision variables, system parameters, algorithms) are investigated in [61]. Note well-known assignment problems and covering problems are very close to location problem.

In location problem, we can consider a set of local alternative decisions (i.e., DA's) for each location place. In this case, compatibility may be based on resource constraints, and on the regions of "service". Similar practical problem has been examined to locate medical hospitals in a region. Close problem is described in ([104], p. 320). Practical significance of multicriteria location/assignment problems is increasing ([99], [276], [516], etc.). Let us consider a formulation for a specific location/assignment problem.

Let $H = \{1, ..., i, .., n\}$ be a finite set of elements under location/assignment (facility, people, etc.). We are given a set of places $Q = \{1, ..., i, .., m\}$ for potential location/assignment of above-mentioned elements (H). In addition, we take into account the following possible information:

(1) weighted relationships on Q (e.g., closeness);

(2) relationships on H;

(3) correspondence between elements of H and Q (weights, vector estimates, preference relations, restrictions, etc.); and

(4) information for some assignment subdecisions of two kinds:

(a) correspondence between a composite element (as a subset of H that are assigned to an element of Q) and places Q; the following data may be applied: weights, vector estimates, preference relations, restrictions, rules, etc.;

(b) correspondence between assignment subdecisions, for example, pairs as follows: $i_1 \in H$ is assigned to $j_1 \in Q$ and $i_2 \in H$ is assigned to $j_2 \in Q$; the following data may be applied: weights, vector estimates, preference relations, restrictions, rules, etc.

In the case, when several elements of H are assigned to the only one place, we can build an additional aggregate element of H with all corresponding attributes.

Thus the problem is:

Locate elements of H to places Q while taking into account additional information (relations, preferences, rules, etc.).

Clearly, the problem may be considered as a satisfaction or optimization one. Quality of assignment decisions can be formulated as integrated goodness of local decisions (on the basis of estimates, preference relation, etc.) subject to restrictions, rules, etc.

Many versions of this problem are examined in combinatorial optimization. Recently, approaches of artificial intelligence are applied too. The case of uncertainty (ordinal scales, probabilistic or fuzzy estimates, etc.) is very important too.

Now let us consider an example, in that we use our morphological approach to assignment of offices (i.e., places and corresponding morphological classes of potential personnel). Note that close problems are examined in ([276], [326]).

Also, it is reasonable to point out the book [190] that describes very interesting problems as follows:

(a) the stable marriage problem,

(b) the resident/hospital problem; and

(c) the stable roommates problem.

The book [253] describes the stable marriage problem and its relation to many combinatorial problems. The above-mentioned problems are not optimization ones: given a group of people, where every person has a preference list of others. The problems are:

Find all sets of so called 'stable' matchings, i.e., partitions of these people into pairs such that each person prefers to have his or her partner.

There are optimization versions of this problem too.

7.2.1 Preliminary Information

In our example, the following main concepts are examined: (1) research projects; (2) personnel; (3) offices; and (4) relationships of the concepts above.

We consider four research projects as follows:

(a) large project R_1 (3 specialists and secretary);

(b) project R_2 (2 specialists);

(c) project R_3 (1 specialists); and

(d) project R_4 (1 specialists).

Also, one secretary participates at projects R_1, R_2, and R_4.

Structure of projects, a plan of offices (A, B, C, D, X, Y, Z) and their closeness (two levels: dot-and-dash lines correspond to weak closeness), and description of personnel are presented in Fig. 7.2, Fig. 7.3 and Table 7.2, respectively.

Note that in Table 7.2 feasible assignment of 1 or 2 people to offices is shown in brackets [...].

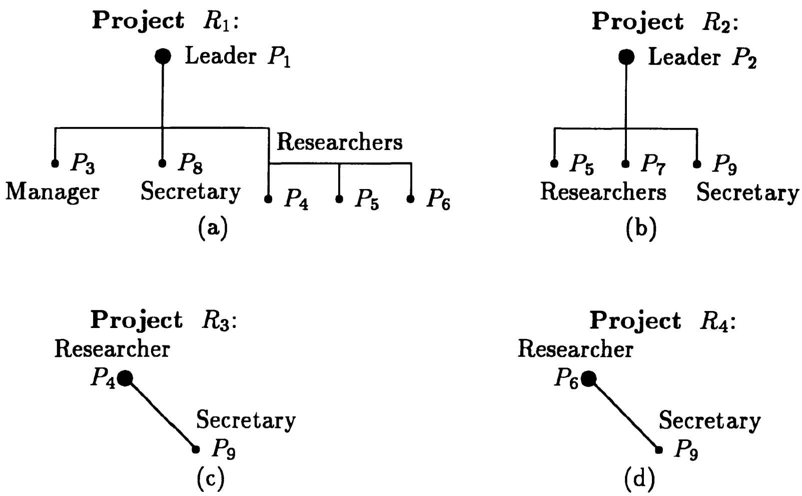

Fig. 7.2. Structure of projects

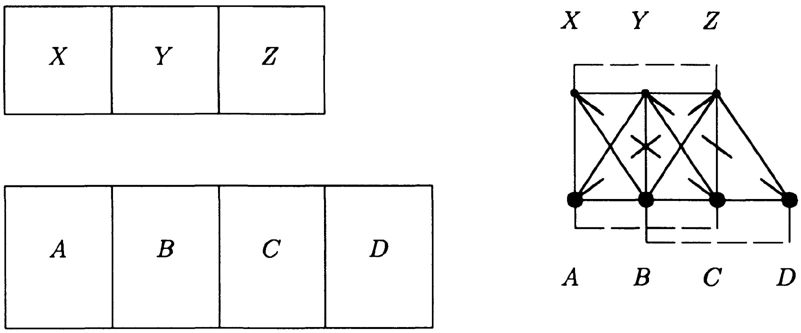

Fig. 7.3. Plan of offices and graph of their closeness

Table 7.2. Description of personnel

People	Role	Project	Smoke	Corresponding offices	Friendship
P_1	Leader of large project	R_1	Yes	$A[1]$, $B[1]$, $C[1]$, $D[1]$	P_2, P_3
P_2	Leader of project	R_2	No	$A[1]$, $B[1]$, $C[1]$, $D[1]$	P_1
P_3	Manager of large project	R_1	Yes	$A[2]$, $B[2]$, $C[2]$, $D[2]$, $X[1]$, $Y[1]$, $Z[1]$	P_1, P_5 P_8
P_4	Researcher	R_1, R_3	No	$A[2]$, $B[2]$, $C[2]$, $D[2]$, $X[1]$, $Y[1]$, $Z[1]$	P_6
P_5	Researcher	R_2	Yes	$A[2]$, $B[2]$, $C[2]$, $D[2]$	P_3, P_8
P_6	Researcher	R_1, R_4	No	$A[2]$, $B[2]$, $C[2]$, $D[2]$, $X[1]$, $Y[1]$, $Z[1]$	P_4
P_7	Researcher	R_1, R_2	Yes	$A[2]$, $B[2]$, $C[2]$, $D[2]$	P_5, P_9
P_8	Secretary	R_1	Yes	$A[2]$, $B[2]$, $C[2]$, $D[2]$, $X[2]$, $Y[2]$, $Z[2]$	P_3, P_5
P_9	Secretary	All	Yes	$A[2]$, $B[2]$, $C[2]$, $D[2]$, $X[2]$, $Y[2]$, $Z[2]$	P_7

In addition, we consider the following rules:

Rule 7.2.1. The leader of project has to be close to all members of the group (researchers, manager, secretary).

Rule 7.2.2. The leader of project has to be located in a large office (single).

Rule 7.2.3. The manager of project has to be close to the leader of the group, and secretary.

Rule 7.2.4. The manager of project can be located in a small office (single) or in a large office (twin).

Rule 7.2.5. The researcher, who conducts and manages his/her research project, can be located in a small office (single) or in a large office (twin).

Rule 7.2.6. People of the same project have to be located in the same office or in close offices.

Rule 7.2.7. Secretary can be located in a large office (twin) or a small office (twin).

Rule 7.2.8. Smoking and non-smoking people can not be located in the same office.

Rule 7.2.9. Friends have to be located in the same office or in close offices.

In particular, these rules were taken into account to specify a correspondence of people to offices (Table 7.2).

7.2.2 Composite Decision

First, let us consider complementability of people, who can be located to a twin office. Table 7.3 contains the following vector estimate and a resultant ordinal estimate:

1. Vector estimate consists of the following binary elements:

(1) smoking (1 corresponds to the same smoke/non-smoke, and 0 otherwise);

(2) joint work (1 corresponds to a joint project, and 0 otherwise);

(3) friendship (1 corresponds to friendship, and 0 otherwise).

2. The following scale is used for the resultant complementability estimate: $[0, 1, 2]$, where 0 corresponds to incompatibility, 1 corresponds to compatibility, 2 corresponds to good compatibility.

Table 7.3. Complementability

	P_4	P_5	P_6	P_7	P_8	P_9
P_3	(010)/0	(101)/1	(010)/0	(110)/1	(111)/2	(100)/1
P_4		(000)/0	(111)/2	(010)/0	(010)/0	(010)/0
P_5			(000)/0	(111)/2	(100)/1	(110)/1
P_6				(010)/0	(010)/0	(010)/0
P_7					(110)/1	(111)/2
P_8						(110)/1

As a results, we get the following good aggregate pair DA's:
$P_{38} = P_3\&P_8$, $P_{46} = P_4\&P_6$, $P_{79} = P_7\&P_9$, $P_{57} = P_5\&P_7$.

Note that here it is possible to apply the special model: profit clique with taking into account complementability of selected elements (see chapter 2).

We consider special estimates of two kinds to describe relationship between personnel and offices: (1) correspondence of personnel to offices; and (2) compatibility of local decisions.

Local decisions consist in assignment of the person to an office. In each case above, we can examine the following stages: (a) verbal description and rules; (b) multicriteria assessment; and (c) resultant ordinal estimates. Our resultant structure of composite decision is presented in Fig. 7.4. We assume that ordinal estimates of the correspondence of personnel to offices is based on expert judgement (while taking into account above-mentioned rules, and personal preferences).

Table 7.4 presents two relationships between people on the basis of Fig. 7.2 and Table 7.2 as follows: joint project (p), and friendship (f).

$S = A * B * C * D * X * Y * Z$

$S' = A_3 * B_1 * C_2 * D_5 * X_4 * Y_3 * Z_2 =$
$(P_3\&P_8) * P_1 * P_2 * (P_7\&P_9) * P_5 * P_6 * P_4$

X	Y	Z
$X_1 = P_3(2)$	$Y_1 = P_3(1)$	$Z_1 = P_3(2)$
$X_2 = P_4(1)$	$Y_2 = P_4(2)$	$Z_2 = P_4(1)$
$X_3 = P_6(1)$	$Y_3 = P_6(2)$	$Z_3 = P_6(1)$
$X_3 = P_5(1)$		

A	B	C	D
$A_1 = P_1(1)$	$B_1 = P_1(1)$	$C_1 = P_1(1)$	$D_1 = P_1(1)$
$A_2 = P_2(1)$	$B_2 = P_2(1)$	$C_2 = P_2(1)$	$D_2 = P_2(1)$
$A_3 = P_3\&P_8(2)$	$B_3 = P_3\&P_8(3)$	$C_3 = P_3\&P_8(3)$	$D_3 = P_3\&P_8(2)$
$A_4 = P_4\&P_6(2)$	$B_4 = P_4\&P_6(3)$	$C_4 = P_4\&P_6(3)$	$D_4 = P_4\&P_6(2)$
$A_5 = P_7\&P_9(1)$	$B_5 = P_7\&P_9(2)$	$C_5 = P_7\&P_9(2)$	$D_5 = P_7\&P_9(1)$
$A_6 = P_5\&P_7(1)$	$B_6 = P_5\&P_7(2)$	$C_6 = P_5\&P_7(2)$	$D_6 = P_5\&P_7(1)$

Fig. 7.4. Structure of composite decision

Tables 7.5, 7.6, 7.7, 7.8, 7.9, and 7.10 contain compatibility of local decisions. We use the following estimates:

1. Vector estimate consisting of two components as follows:

(1) fulfillment of rules for joint projects:

(i) the variable equals 2 if there exists the fulfillment of rules 7.2.1, 7.2.3, and 7.2.6;

(ii) the variable equals 1 if there exists the weak fulfillment of rules 7.2.1, 7.2.3, and 7.2.6; and

(iii) the variable is denoted by Δ otherwise (incompatibility, ***prohibition***);

(2) fulfillment of rules for friendship:

(i) the variable equals 2 if there exists the fulfillment of rules 7.2.9;

(ii) the variable equals 1 if there exists the weak fulfillment of rules 7.2.9; and

(iii) 0 otherwise.

2. Resultant ordinal estimate:
(i) the best compatibility:
(a) 5 (vector estimate equals $(4,4)$ or $(8,0)$) or
(b) ◇ (independence of DA's);
(ii) quasi best compatibility 4 (vector estimate equals $(4,2)$);
(iii) good compatibility 3 (vector estimate equals $(4,0)$ or $(2,2)$);
(iv) satisfied compatibility 2 (vector estimate equals $(1,1)$, $(2,0)$) $(0,2)$, and $(2,1)$); and
(v) bad compatibility 1 (vector estimate equals $(0,1)$, $(1,0)$);
(vi) incompatibility or ***prohibition***:
(a) 0 (vector estimate equals $(0,0)$);
(b) Δ (if the following component of the vector estimate is denoted by Δ).

Table 7.4. Two relationships

	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
P_1	f	p,f	p	p	p	-	p	-
P_2		-	-	p	-	p	p	-
P_3			p	p,f	p	-	p,f	-
P_4				p	p,f	-	p	p
P_5					p	f	p,f	-
P_6						-	p	p
P_7							-	p,f
P_8								-

Table 7.5. Compatibility

	B_1	B_2	B_3	B_4	B_5	B_6
A_1	$/\Delta$	$(0,2)/2$	$(4,2)/4$	$(4,0)/3$	/◇	$(2,0)/2$
A_2	$(0,2)/2$	$/\Delta$	/◇	/◇	$(4,0)/3$	$(4,0)/3$
A_3	$(4,2)/4$	/◇	$/\Delta$	$(8,0)/5$	/◇	$(4,4)/5$
A_4	$(4,0)/3$	/◇	$(8,0)/5$	$/\Delta$	$(4,0)/3$	$(4,0)/3$
A_5	/◇	$(4,0)/3$	/◇	$(4,0)/3$	$/\Delta$	$/\Delta$
A_6	$(2,0)/2$	$(4,0)/3$	$(4,4)/5$	$(4,0)/3$	$/\Delta$	$/\Delta$

Table 7.6. Compatibility

	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3
A_1	(2,2)/3	(2,0)/2	(2,0)/2	(2,0)/2	(2,2)/3	(2,0)/2	(2,0)/2
A_2	(0,2)/2	/◇	/◇	(2,0)/2	(0,2)/2	/◇	/◇
A_3	/△	(4,0)/3	(4,0)/3	(4,4)/8	/△	(4,0)/3	(4,0)/3
A_4	(4,0)/3	/△	/△	(4,0)/3	(4,0)/3	/△	/△
A_5	/◇	(2,0)/2	(2,0)/2	(0,2)/2	/◇	(2,0)/2	(2,0)/2
A_6	(2,2)/3	(2,0)/2	(2,0)/2	/△	(2,2)/3	(2,0)/2	(2,0)/2
B_1	(2,2)/3	(2,0)/2	(2,0)/2	(2,0)/2	(2,2)/3	(2,0)/2	(2,0)/2
B_2	(0,2)/2	/◇	/◇	(2,0)/2	(0,2)/2	/◇	/◇
B_3	/△	(4,0)/3	(4,0)/3	(4,4)/5	/△	(4,0)/3	(4,0)/3
B_4	(4,0)/3	/△	/△	(4,0)/3	(4,0)/3	/△	/△
B_5	/◇	(2,0)/2	(2,0)/2	(0,2)/2	/◇	(2,0)/2	(2,0)/2
B_6	(2,2)/3	(2,0)/2	(2,0)/2	/△	(2,2)/3	(2,0)/2	(2,0)/2
C_1	(1,1)/2	(1,0)/1	(1,0)/1	(1,0)/1	(2,2)/3	(2,0)/2	(2,0)/2
C_2	(0,1)/1	/◇	/◇	(1,0)/1	(0,2)/2	/◇	/◇
C_3	/△	(2,0)/2	(2,0)/2	(2,2)/4	/△	(4,0)/3	(4,0)/3
C_4	(2,0)/2	/△	/△	(2,0)/2	(4,0)/3	/△	/△
C_5	/◇	(1,0)/1	(1,0)/1	(0,2)/2	/◇	(2,0)/2	(2,0)/2
C_6	(1,1)/2	(1,0)/1	(1,0)/1	/△	(2,2)/3	(2,0)/2	(2,0)/2
D_1	(0,0)/0	(0,0)/0	(0,0)/0	(0,0)/0	(1,1)/2	(1,0)/1	(1,0)/1
D_2	(0,0)/0	/◇	/◇	(0,0)/0	(0,1)/1	/◇	/◇
D_3	/△	(0,0)/0	(0,0)/0	(0,0)/0	/△	(2,0)/2	(2,0)/2
D_4	(0,0)/0	/△	/△	(0,0)/0	(2,0)/2	/△	/△
D_5	/◇	(0,0)/0	(0,0)/0	/◇	/◇	(1,0)/1	(1,0)/1
D_6	(0,0)/0	(0,0)/0	(0,0)/0	/△	(1,1)/2	(1,0)/1	/△

Table 7.7. Compatibility

	C_1	C_2	C_3	C_4	C_5	C_6
A_1	$/\triangle$	$(0,1)/1$	$(2,1)/2$	$(2,0)/2$	$/\diamond$	$(1,0)/1$
A_2	$(0,1)/1$	$/\triangle$	$/\diamond$	$/\diamond$	$(2,0)/2$	$(2,0)/2$
A_3	$(2,1)/2$	$/\diamond$	$/\triangle$	$(4,0)/3$	$/\diamond$	$(2,2)/3$
A_4	$(2,0)/2$	$/\diamond$	$(4,0)/3$	$/\triangle$	$(2,0)/2$	$(2,0)/2$
A_5	$/\diamond$	$(2,0)/2$	$/\diamond$	$(2,0)/2$	$/\triangle$	$/\triangle$
A_6	$(2,0)/2$	$(2,0)/2$	$(2,2)/3$	$(2,0)/2$	$/\triangle$	$/\triangle$
B_1	$/\triangle$	$(0,2)/2$	$(4,2)/4$	$(4,0)/3$	$/\diamond$	$(2,0)/2$
B_2	$(0,2)/2$	$/\triangle$	$/\diamond$	$/\diamond$	$(4,0)/3$	$(4,0)/3$
B_3	$(4,2)/4$	$/\diamond$	$/\triangle$	$(8,0)/5$	$/\diamond$	$(4,4)/5$
B_4	$(4,0)/3$	$/\diamond$	$(8,0)/5$	$/\triangle$	$(4,0)/3$	$(4,0)/3$
B_5	$/\diamond$	$(4,0)/3$	$/\diamond$	$(4,0)/3$	$/\triangle$	$/\triangle$
B_6	$(2,0)/2$	$(4,0)/3$	$(4,4)/5$	$(4,0)/3$	$/\triangle$	$/\triangle$

Table 7.8. Compatibility

	D_1	D_2	D_3	D_4	D_5	D_6
A_1	$/\triangle$	$(0,0)/0$	$(0,0)/0$	$(0,0)/0$	$/\diamond$	$(0,0)/0$
A_2	$(0,0)/0$	$/\triangle$	$/\diamond$	$/\diamond$	$(0,0)/0$	$(0,0)/0$
A_3	$(0,0)/0$	$/\diamond$	$/\triangle$	$(0,0)/0$	$/\diamond$	$(0,0)/0$
A_4	$(0,0)/0$	$/\diamond$	$(0,0)/0$	$/\triangle$	$(0,0)/0$	$(0,0)/0$
A_5	$/\diamond$	$(0,0)/0$	$(0,0)/0$	$(0,0)/0$	$/\triangle$	$/\triangle$
A_6	$(0,0)/0$	$(0,0)/0$	$(0,0)/0$	$(0,0)/0$	$/\triangle$	$/\triangle$
B_1	$/\triangle$	$(0,1)/1$	$(2,1)/2$	$(2,0)/2$	$/\diamond$	$(1,0)/1$
B_2	$(0,1)/1$	$/\triangle$	$/\diamond$	$/\diamond$	$(2,0)/2$	$(2,0)/2$
B_3	$(2,1)/2$	$/\diamond$	$/\triangle$	$(4,0)/3$	$/\diamond$	$(2,2)/3$
B_4	$(2,0)/2$	$/\diamond$	$(4,0)/3$	$/\triangle$	$(2,0)/2$	$(2,0)/2$
B_5	$/\diamond$	$(2,0)/2$	$/\diamond$	$(2,0)/2$	$/\triangle$	$/\triangle$
B_6	$(2,0)/2$	$(2,0)/2$	$(2,2)/3$	$(2,0)/2$	$/\triangle$	$/\triangle$
C_1	$/\triangle$	$(0,2)/2$	$(4,2)/4$	$(4,0)/3$	$/\diamond$	$(2,0)/2$
C_2	$(0,2)/2$	$/\triangle$	$/\diamond$	$/\diamond$	$(4,0)/3$	$(4,0)/3$
C_3	$(4,2)/4$	$/\diamond$	$/\triangle$	$(8,0)/5$	$/\diamond$	$(4,4)/5$
C_4	$(4,0)/3$	$/\diamond$	$(8,0)/5$	$/\triangle$	$(4,0)/3$	$(4,0)/3$
C_5	$/\diamond$	$(4,0)/3$	$/\diamond$	$(4,0)/3$	$/\triangle$	$/\triangle$
C_6	$(2,0)/2$	$(4,0)/3$	$(4,4)/5$	$(4,0)/3$	$/\triangle$	$/\triangle$

Table 7.9. Compatibility

	Z_1	Z_2	Z_3
A_1	(1,1)/2	(1,0)/1	(1,0)/1
A_2	(0,1)/1	/◇	/◇
A_3	/△	(2,0)/2	(2,0)/2
A_4	(2,0)/2	/△	/△
A_5	/◇	(1,0)/1	(1,0)/1
A_6	(1,1)/2	(1,0)/1	(1,0)/1
B_1	(2,2)/3	(2,0)/2	(2,0)/2
B_2	(0,2)/2	/◇	/◇
B_3	/△	(4,0)/3	(4,0)/3
B_4	(4,0)/3	/△	/△
B_5	/◇	(2,0)/1	(2,0)/1
B_6	(2,2)/3	(2,0)/2	(2,0)/2
C_1	(2,2)/3	(2,0)/2	(2,0)/2
C_2	(0,2)/2	/◇	/◇
C_3	/△	(4,0)/3	(4,0)/3
C_4	(4,0)/3	/△	/△
C_5	/◇	(2,0)/2	(2,0)/2
C_6	(2,2)/3	(2,0)/2	(2,0)/2
D_1	(2,2)/3	(2,0)/2	(2,0)/2
D_2	(0,2)/2	/◇	/◇
D_3	/△	(4,0)/3	(4,0)/3
D_4	(4,0)/3	/△	/△
D_5	/◇	(2,0)/2	(2,0)/2
D_6	(2,2)/3	(2,0)/2	(2,0)/2

Table 7.10. Compatibility

	Y_1	Y_2	Y_3	Z_1	Z_2	Z_3
X_1	/△	(2,0)/2	(2,0)/2	/△	(1,0)/1	(1,0)/1
X_2	(2,0)/2	/△	(2,2)/3	(1,0)/1	/△	(1,1)/2
X_3	(2,0)/2	(2,2)/3	/△	(1,0)/1	(1,1)/2	/△
X_4	(2,2)/3	(2,0)/2	(2,0)/2	(1,1)/2	(1,0)/1	(1,0)/1
Y_1				/△	(2,0)/2	(2,0)/2
Y_2				(2,0)/2	/△	(2,2)/3
Y_3				(2,0)/2	(2,2)/3	/△

As a results, we consider the following composite decision (Fig. 7.5):

$$S' = A_3 * B_1 * C_2 * D_5 * X_4 * Y_3 * Z_2 =$$
$$(P_3 \& P_8) * P_1 * P_2 * (P_7 \& P_9) * P_5 * P_6 * P_4,$$

where $N(S') = (2; 5, 2)$.

X_4 :	Y_3 :	Z_2 :
P_5	P_6	P_4

A_3 :	B_1 :	C_2 :	D_5 :
$P_3 \& P_8$	P_1	P_2	$P_7 \& P_9$

Fig. 7.5. Illustration for composite decision

7.3 TRAVELING SALESMAN PROBLEM

7.3.1 Formulation

The traveling salesman problem (TSP) is well-known one ([24], [281], [379], [413], etc.). This combinatorial model is used in many application domains (scheduling of manufacturing and computer systems, transportation systems, network design, etc.). The basic formulation in the plane is the following. Let $A = \{a_1, ..., a_n\}$ be a given set of points/vertices (e.g., cities). Here for each pair of points $i, j \in A$ we consider an arc with a nonnegative weight $c_{i,j}$. Let V be a set of the arcs. A weight of the arc (i, j) corresponds to distance between cities i, j (e.g., a direct travel time, etc.). The problem is:

Find a Hamiltonian circuit (tour) / path, that visits each vertex (city) exactly once and takes the least total weight (e.g., travel time.)

We introduce binary variables $x_{ij} = 1$ if j immediately follows i on the path, $x_{ij} = 0$ otherwise. Thus we get the following:

$$\min \sum_{(i,j) \in V} c_{ij} x_{ij}$$

$$s.t. \quad \sum_{\{i:\ (i,j)\in V\}} x_i = 1 \ \ \forall j \in A$$

$$\sum_{\{j:\ (i,j)\in V\}} x_i = 1 \ \ \forall i \in A$$

$$\sum_{\{(i,j)\in V:\ i\in B,\ j\in A\setminus B\}} x_{ij} \geq 1, \ \forall B \subset A, \ 2 \leq |B| \leq ||A| - 2.$$

Other formulations of TSP may be based, for example, on the following ([281], etc.):

(a) specific digraphs instead of the plane;

(b) various metrics/proximity;

(c) approaches to compute a total path weight; and

(d) vector weights of arcs; etc.

Partitioning solving schemes are often used for TSP ([239], [281], [413], [469], [518], etc.). In this case, solving schemes are called ***cluster first-route second*** [53].

7.3.2 Scheme of Heuristic

Clearly, we can examine a set of subgraphs $\{G'\}$ as a result of partitioning (clustering) an initial graph G into parts. Let us consider the following local problem of generating alternative local decisions for G'.

First we reveal a convex envelope of vertices in G' and construct the best Hamiltonian paths for each pair of vertices of the envelope. Thus we get DA's for G'.

Secondly we use our approach for all levels of our decomposition. Note that centers of subgraphs as G' of lower levels are often used as an aggregated vertex at the corresponding higher level. If the number of vertices in each G' is limited by p, we have about $O(p!)$ operations to construct each Hamiltonian path (at the bottom hierarchical level), and $O(p(p-1)/2)$ Hamiltonian paths.

Thirdly note that composition of a composite Hamiltonian path (circuit) for a higher hierarchical level of decomposition on the basis of paths of corresponding lower hierarchical levels ('sons') may be based on the following problems (with chain-like compatibility structure):

(i) 'shortest' (e.g., *set-to-set*) path problem; and

(ii) morphological clique problem.

Now we list basic problems as follows:

Problem 1 (**P1**). Partitioning of a graph into subgraphs (clustering of the graph vertices with taking into account distances/proximity between them).

Problem 2 (**P2**). Construction of a *convex* envelope for vertices of a subgraph.

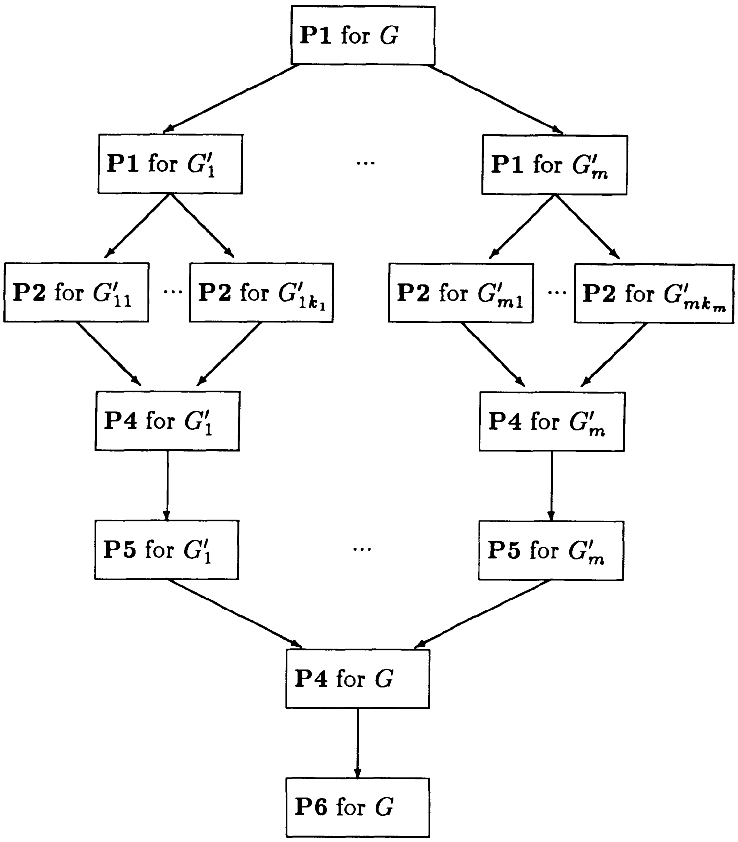

Fig. 7.6. Scheme of morphological heuristic

Problem 3 (**P3**). Building of Hamiltonian path (local decision for the subgraph) for a vertex pair (start/end points) of the vertex set of the *convex* envelope. Note that here it is reasonable to take into account location of the subgraph at graph because, in some cases, we can get the vertex pair for only one/two sides of graph.

Problem 4 (**P4**). Computation and analysis of compatibility between local paths.

Problem 5 (**P5**). Composing of a composite Hamiltonian path(s) (composing of local paths into a composite path) for a composite graph (***shortest path*** problem or morphological clique problem).

Problem 6 (**P6**). Composing of a composite Hamiltonian circuit (composing of local paths into a composite path) for a composite graph (***shortest path*** problem and selection of the best path or morphological clique problem).

A scheme of morphological ***cascade-like*** heuristic for 3-level hierarchy is presented in Fig. 7.6, where G'_{el} is part l $(l = 1, ..., k_e)$ of G'_e $(e = 1, ..., m)$.

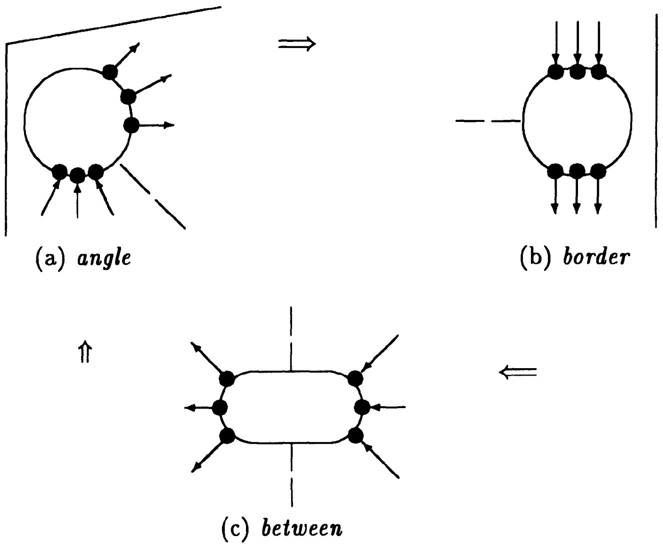

Fig. 7.7. Situations for coordination

7.3.3 Coordination of Hierarchical Levels

Here we consider evident situations to coordinate paths at different hierarchical levels. The following two hierarchical levels are assumed: (i) ***high level***; and (ii) ***low level***. For example, graph G may be examined as ***high level***, its parts $\{G'_i\}$ correspond to ***low level***, and so on.

Trivially, a Hamiltonian path (circuit) at ***high level*** defines basic directions of paths at ***low level***. As a result, we can get several typical situations as follows: ***angle***, ***border***, ***between***. Fig. 7.7 illustrates the situations (indicated vertices belong to a ***convex*** envelope) of a vertex cluster. Thus a Hamiltonian path (circuit) at ***high level*** is realized on the basis of paths, which interconnect an input vertex and an output vertex at (*low level*). In fact, the problem is: to divide vertices of the ***convex*** envelope into input and output ones, and to build corresponding paths. Note, building of paths is close to or the same as *set-to-set* shortest path problem.

7.3.4 Numerical Example

Now let us consider a numerical example of TSP in the plan for an initial graph G (Table 7.12, and Fig. 7.8). Note our example is close to the example of Karp [240]. Fig. 7.9 presents two-level partitioning of G.

At each stage of our morphological heuristic, we examine several local decisions as Hamiltonian paths (DA's). Evidently that at the last stage, it is necessary to consider Hamiltonian circuit for G. Basic Hamiltonian paths (DA's) for G'_1, G'_2, G'_3, and G'_4 are depicted in Fig. 7.10, Fig. 7.11, Fig. 7.12, and Fig. 7.13, respectively.

Fig. 7.14 contains Hamiltonian paths (DA's) for G'_{11}, G'_{12}, G'_{13}, G'_{14}, G'_{15}; Fig. 7.15 contains Hamiltonian paths (DA's) for G'_{21}, G'_{32}, G'_{33}; Fig. 7.16 contains Hamiltonian paths (DA's) for G'_{31}, G'_{32}, G'_{33}, G'_{34}; and Fig. 7.17 contains Hamiltonian paths (DA's) for G'_{41}, G'_{42}, G'_{43}, G'_{44}. Lengths of elementary paths are presented in Table 7.13. Tables 7.14, 7.15, and 7.16 demonstrate proximity of paths for G'_1 as an example (chain-like compatibility). Clearly, that for more complicated situations (e.g., ordinal scales, multiattribute estimates, and basic morphological clique problem), the tables are very important. Here we use symbol $\diamond$ to point out the case when DA's are not connected in corresponding composite DA's at the higher hierarchical level. In our example, we take into account proximity between paths at the stage of computing of paths for the 2nd hierarchical level. Tables 7.17, 7.18, and 7.19 contain paths for the 2nd level of hierarchy.

Table 7.12. Vertex coordinates of graph G

No.	x	y	No.	x	y
1	01.00	88.00	21	65.00	68.00
2	09.00	88.00	22	65.00	58.00
3	01.00	78.00	23	42.00	45.00
4	09.00	78.00	24	42.00	35.00
5	35.00	88.00	25	10.00	37.00
6	45.00	88.00	26	05.00	30.00
7	40.00	78.00	27	10.00	23.00
8	25.00	79.00	28	25.00	35.00
9	25.00	71.00	29	25.00	25.00
10	10.00	65.00	30	35.00	15.00
11	10.00	55.00	31	50.00	15.00
12	44.00	64.00	32	67.00	40.00
13	36.00	56.00	33	67.00	30.00
14	73.00	90.00	34	98.00	47.00
15	65.00	85.00	35	89.00	43.00
16	73.00	80.00	36	98.00	39.00
17	90.00	84.00	37	66.00	19.00
18	100.00	84.00	38	74.00	11.00
19	90.00	70.00	39	95.00	25.00
20	100.00	70.00	40	95.00	15.00

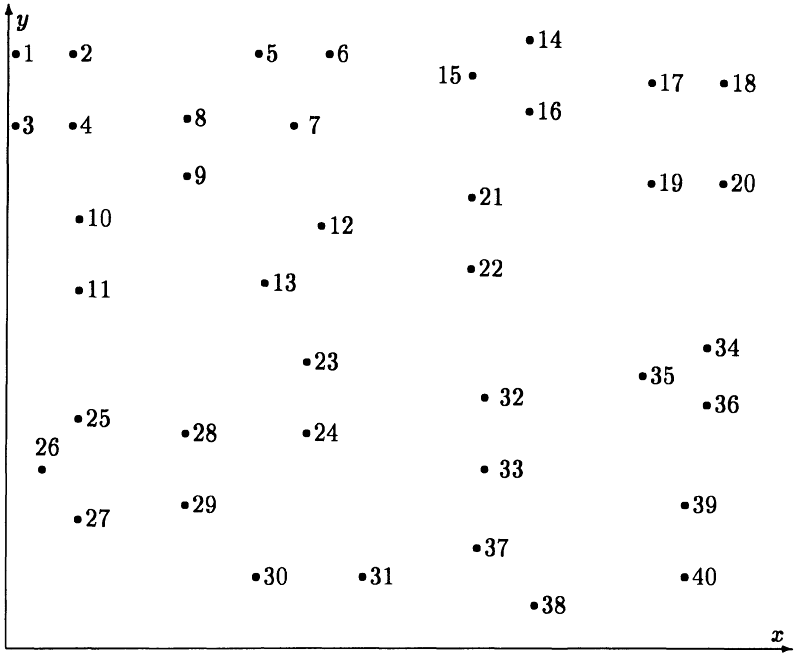

Fig. 7.8. Example for TSP (initial graph G)

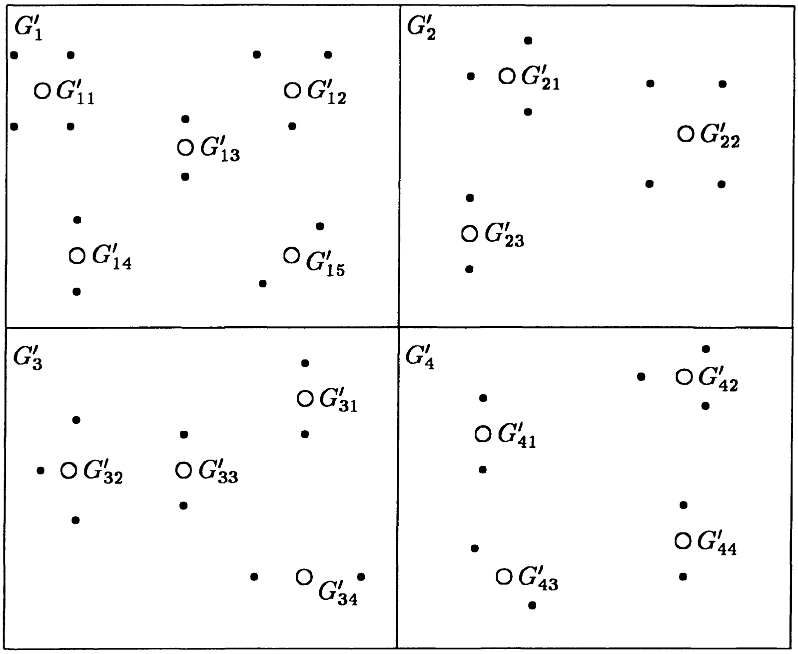

Fig. 7.9. Two-level partitioning of G

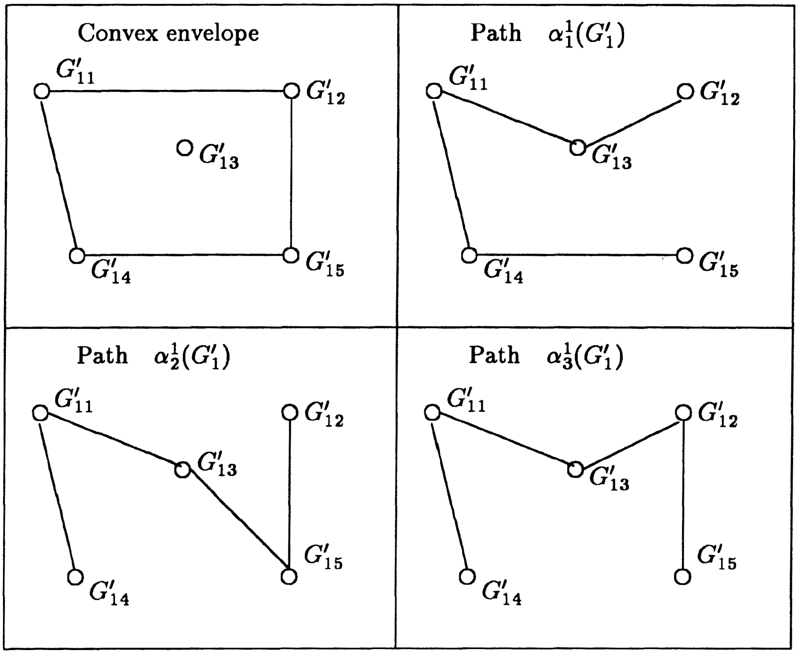

Fig. 7.10. Basic Hamiltonian paths for G'_1

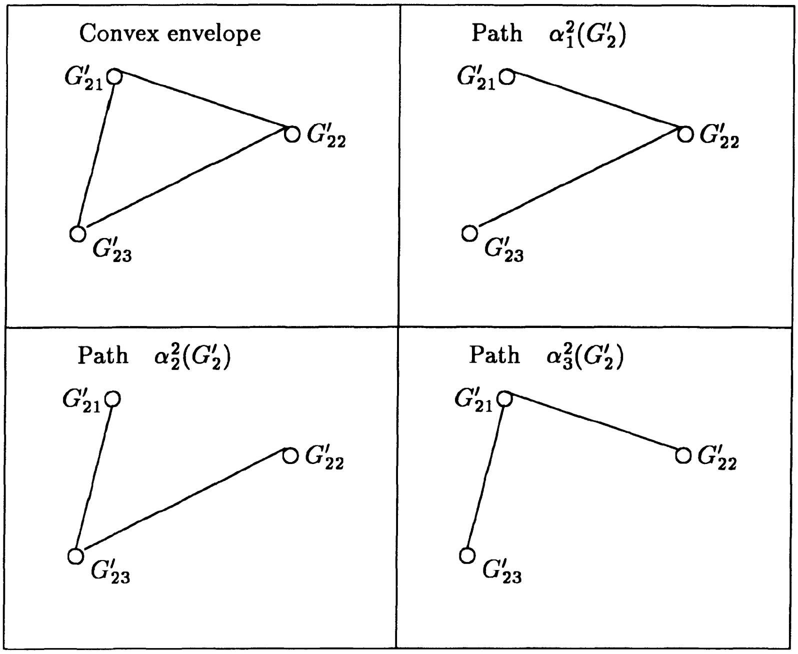

Fig. 7.11. Basic Hamiltonian paths for G'_2

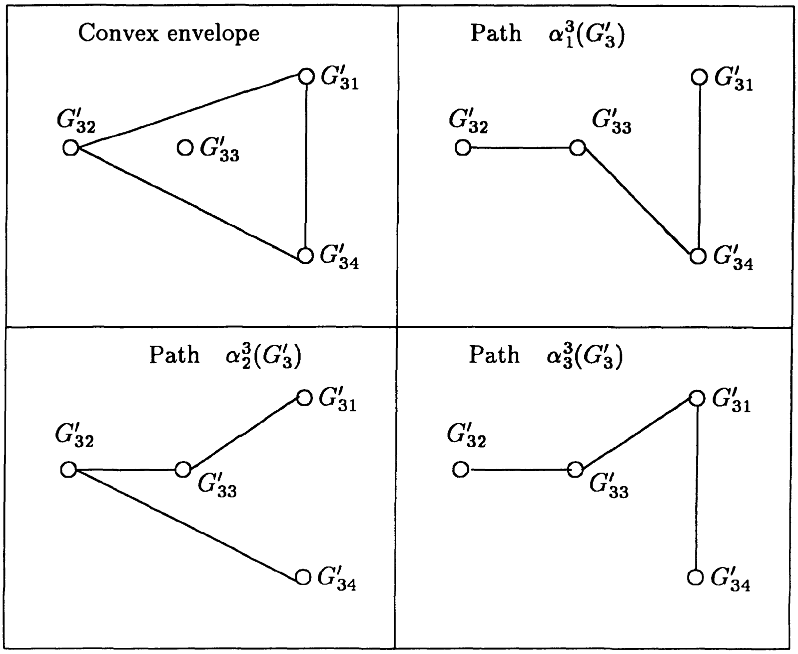

Fig. 7.12. Basic Hamiltonian paths for G'_3

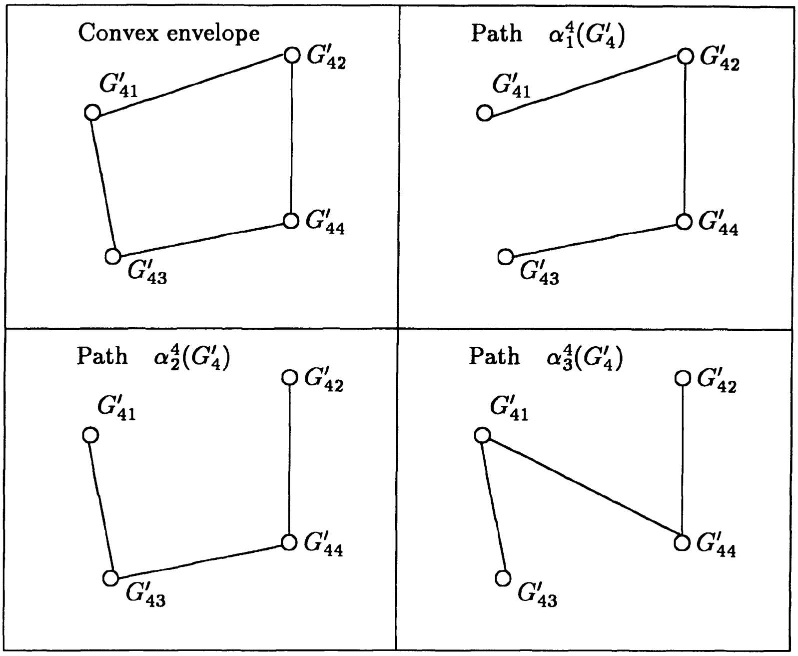

Fig. 7.13. Basic Hamiltonian paths for G'_4

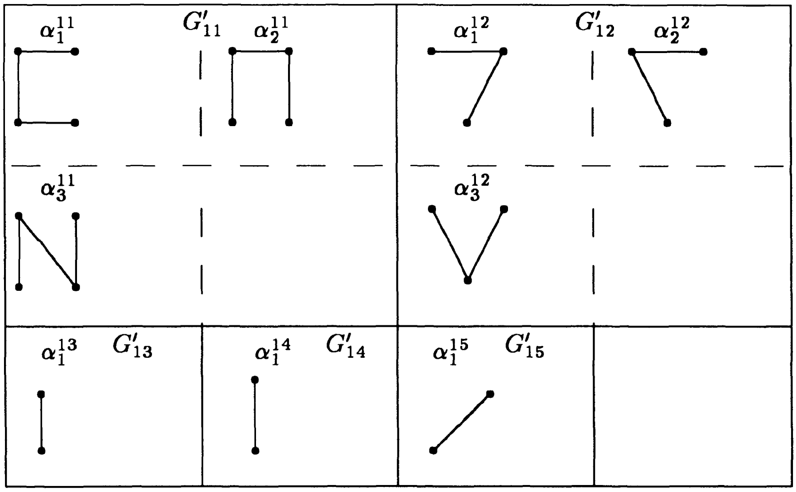

Fig. 7.14. Basic Hamiltonian paths for components of G'_1

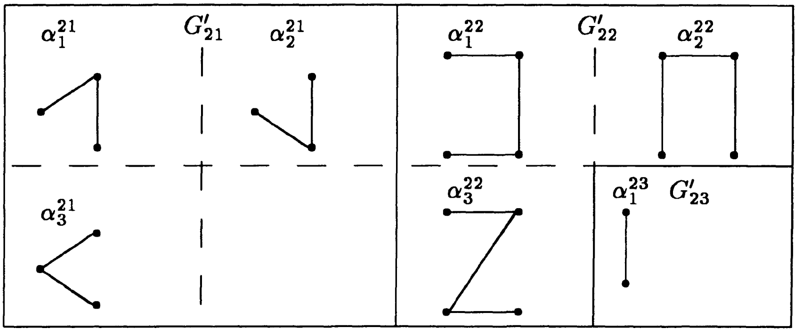

Fig. 7.15. Basic Hamiltonian paths for components of G'_2

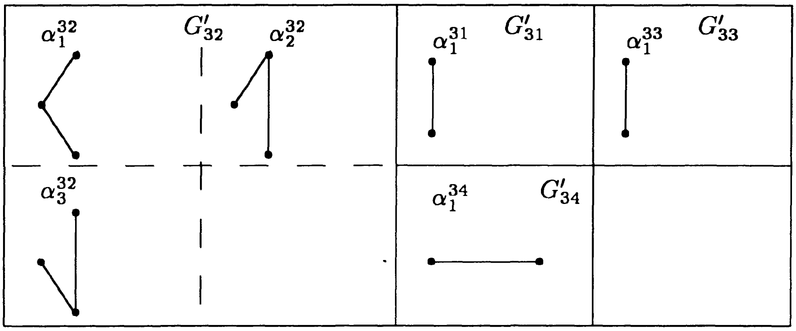

Fig. 7.16. Basic Hamiltonian paths for components of G'_3

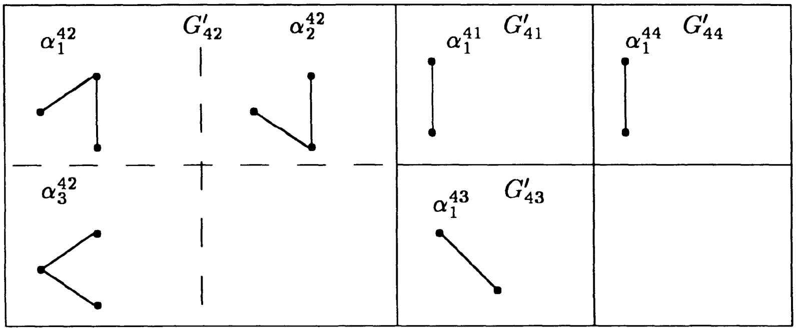

Fig. 7.17. Basic Hamiltonian paths for components of G'_4

Table 7.13. Lengths of basic elementary paths

No.	Path	Length	No.	Parth	Length
1	α_1^{11}	26.00	15	α_3^{22}	37.20
2	α_2^{11}	28.00	16	α_1^{23}	10.00
3	α_3^{11}	32.80	17	α_1^{31}	10.00
4	α_1^{12}	21.90	18	α_1^{32}	17.20
5	α_2^{12}	21.90	19	α_2^{32}	22.60
6	α_3^{12}	23.80	20	α_3^{32}	22.60
7	α_1^{13}	08.00	21	α_1^{33}	10.00
8	α_1^{14}	10.00	22	α_1^{34}	15.00
9	α_1^{15}	13.70	23	α_1^{41}	10.00
10	α_1^{21}	16.93	24	α_1^{42}	16.81
11	α_2^{21}	16.93	25	α_2^{42}	16.81
12	α_3^{21}	17.86	26	α_3^{42}	17.62
13	α_1^{22}	34.00	27	α_1^{43}	11.30
14	α_2^{22}	38.00	28	α_1^{44}	10.00

Table 7.14. Proximity of paths for α_1^1 (G_1')

	α_1^{12}	α_2^{12}	α_3^{12}	α_1^{13}	α_1^{14}	α_1^{15}
α_1^{11}	◇	◇	◇	23.16	13.10	◇
α_2^{11}	◇	◇	◇	17.50	15.82	◇
α_3^{11}	◇	◇	◇	23.60	15.82	◇
α_1^{12}				13.80	◇	◇
α_2^{12}				15.00	◇	◇
α_3^{12}				13.80	◇	◇
α_1^{13}					◇	◇
α_1^{14}						26.00

Table 7.15. Proximity of paths for α_2^1 (G_1')

	α_1^{12}	α_2^{12}	α_3^{12}	α_1^{13}	α_1^{14}	α_1^{15}
α_1^{11}	$\diamond$	$\diamond$	$\diamond$	18.32	13.10	$\diamond$
α_2^{11}	$\diamond$	$\diamond$	$\diamond$	17.50	15.82	$\diamond$
α_3^{11}	$\diamond$	$\diamond$	$\diamond$	18.32	15.82	$\diamond$
α_1^{12}				$\diamond$	$\diamond$	14.60
α_2^{12}				$\diamond$	$\diamond$	14.60
α_3^{12}				$\diamond$	$\diamond$	25.60
α_1^{13}					$\diamond$	19.70
α_1^{14}						$\diamond$

Table 7.16. Proximity of paths for α_3^1 (G_1')

	α_1^{12}	α_2^{12}	α_3^{12}	α_1^{13}	α_1^{14}	α_1^{15}
α_1^{11}	$\diamond$	$\diamond$	$\diamond$	23.16	13.10	$\diamond$
α_2^{11}	$\diamond$	$\diamond$	$\diamond$	17.50	15.82	$\diamond$
α_3^{11}	$\diamond$	$\diamond$	$\diamond$	23.16	15.82	$\diamond$
α_1^{12}				15.00	$\diamond$	14.60
α_2^{12}				20.40	$\diamond$	14.60
α_3^{12}				13.80	$\diamond$	24.00
α_1^{13}					$\diamond$	$\diamond$
α_1^{14}						$\diamond$

Table 7.17. Paths for the 2nd level of hierarchy (G'_1)

Structure	Vertex route	Length
$\beta_1(\alpha_1^1) = \alpha_1^{12} * \alpha_1^{13} * \alpha_1^{11} * \alpha_1^{14} * \alpha_1^{15}$	$7,6,5,8,9,2,1,3,4,10,11,13,12$	155.56
$\beta_2(\alpha_1^1) = \alpha_2^{12} * \alpha_1^{13} * \alpha_1^{11} * \alpha_1^{14} * \alpha_1^{15}$	$6,5,7,8,9,2,1,3,4,10,11,13,12$	156.76
$\beta_3(\alpha_1^1) = \alpha_3^{12} * \alpha_1^{13} * \alpha_1^{11} * \alpha_1^{14} * \alpha_1^{15}$	$6,7,5,8,9,2,1,3,4,10,11,13,12$	157.56
$\beta_4(\alpha_1^1) = \alpha_1^{12} * \alpha_1^{13} * \alpha_2^{11} * \alpha_1^{14} * \alpha_1^{15}$	$7,6,5,8,9,4,2,1,3,10,11,13,12$	140.00
$\beta_5(\alpha_1^1) = \alpha_2^{12} * \alpha_1^{13} * \alpha_2^{11} * \alpha_1^{14} * \alpha_1^{15}$	$6,5,7,8,9,4,2,1,3,10,11,13,12$	143.20
$\beta_6(\alpha_1^1) = \alpha_3^{12} * \alpha_1^{13} * \alpha_2^{11} * \alpha_1^{14} * \alpha_1^{15}$	$6,7,5,8,9,4,2,1,3,10,11,13,12$	143.90
$\beta_7(\alpha_1^1) = \alpha_1^{12} * \alpha_1^{13} * \alpha_3^{11} * \alpha_1^{14} * \alpha_1^{15}$	$7,6,5,8,9,2,4,1,3,10,11,13,12$	155.18
$\beta_8(\alpha_1^1) = \alpha_2^{12} * \alpha_1^{13} * \alpha_3^{11} * \alpha_1^{14} * \alpha_1^{15}$	$6,5,7,8,9,2,4,1,3,10,11,13,12$	156.38
$\beta_9(\alpha_1^1) = \alpha_3^{12} * \alpha_1^{13} * \alpha_3^{11} * \alpha_1^{14} * \alpha_1^{15}$	$6,7,5,8,9,2,4,1,3,10,11,13,12$	157.08
$\beta_1(\alpha_2^1) = \alpha_1^{12} * \alpha_1^{15} * \alpha_1^{13} * \alpha_1^{11} * \alpha_1^{14}$	$7,6,5,12,13,9,8,2,1,3,4,10,11$	156.32
$\beta_2(\alpha_2^1) = \alpha_2^{12} * \alpha_1^{15} * \alpha_1^{13} * \alpha_1^{11} * \alpha_1^{14}$	$6,5,7,12,13,9,8,2,1,3,4,10,11$	145.12
$\beta_3(\alpha_2^1) = \alpha_3^{12} * \alpha_1^{15} * \alpha_1^{13} * \alpha_1^{11} * \alpha_1^{14}$	$6,7,5,12,13,9,8,2,1,3,4,10,11$	158.22
$\beta_4(\alpha_2^1) = \alpha_1^{12} * \alpha_1^{15} * \alpha_1^{13} * \alpha_2^{11} * \alpha_1^{14}$	$7,6,5,12,13,9,8,4,2,1,3,10,11$	160.22
$\beta_5(\alpha_2^1) = \alpha_2^{12} * \alpha_1^{15} * \alpha_1^{13} * \alpha_2^{11} * \alpha_1^{14}$	$6,5,7,12,13,9,8,4,2,1,3,10,11$	149.02
$\beta_6(\alpha_2^1) = \alpha_3^{12} * \alpha_1^{15} * \alpha_1^{13} * \alpha_2^{11} * \alpha_1^{14}$	$6,7,5,12,13,9,8,4,2,1,3,10,11$	162.12
$\beta_7(\alpha_2^1) = \alpha_1^{12} * \alpha_1^{15} * \alpha_1^{13} * \alpha_3^{11} * \alpha_1^{14}$	$7,6,5,12,13,9,8,2,4,1,3,10,11$	165.84
$\beta_8(\alpha_2^1) = \alpha_2^{12} * \alpha_1^{15} * \alpha_1^{13} * \alpha_3^{11} * \alpha_1^{14}$	$6,5,7,12,13,9,8,2,4,1,3,10,11$	154.64
$\beta_9(\alpha_2^1) = \alpha_3^{12} * \alpha_1^{15} * \alpha_1^{13} * \alpha_3^{11} * \alpha_1^{14}$	$6,7,5,12,13,9,8,2,4,1,3,10,11$	170.74
$\beta_1(\alpha_3^1) = \alpha_1^{15} * \alpha_1^{12} * \alpha_1^{13} * \alpha_1^{11} * \alpha_1^{14}$	$13,12,7,6,5,8,9,2,1,3,4,10,11$	143.72
$\beta_2(\alpha_3^1) = \alpha_1^{15} * \alpha_2^{12} * \alpha_1^{13} * \alpha_1^{11} * \alpha_1^{14}$	$13,12,7,5,6,8,9,2,1,3,4,10,11$	149.84
$\beta_3(\alpha_3^1) = \alpha_1^{15} * \alpha_3^{12} * \alpha_1^{13} * \alpha_1^{11} * \alpha_1^{14}$	$13,12,6,7,5,8,9,2,1,3,4,10,11$	155.56
$\beta_4(\alpha_3^1) = \alpha_1^{15} * \alpha_1^{12} * \alpha_1^{13} * \alpha_2^{11} * \alpha_1^{14}$	$13,12,7,6,5,8,9,4,2,1,3,10,11$	143.32
$\beta_5(\alpha_3^1) = \alpha_1^{15} * \alpha_2^{12} * \alpha_1^{13} * \alpha_2^{11} * \alpha_1^{14}$	$13,12,7,5,6,8,9,4,2,1,3,10,11$	149.44
$\beta_6(\alpha_3^1) = \alpha_1^{15} * \alpha_3^{12} * \alpha_1^{13} * \alpha_2^{11} * \alpha_1^{14}$	$13,12,6,7,5,8,9,4,2,1,3,10,11$	154.62
$\beta_7(\alpha_3^1) = \alpha_1^{15} * \alpha_1^{12} * \alpha_1^{13} * \alpha_3^{11} * \alpha_1^{14}$	$13,12,7,6,5,8,9,2,4,1,3,10,11$	153.78
$\beta_8(\alpha_3^1) = \alpha_1^{15} * \alpha_2^{12} * \alpha_1^{13} * \alpha_3^{11} * \alpha_1^{14}$	$13,12,7,5,6,8,9,2,4,1,3,10,11$	159.90
$\beta_9(\alpha_3^1) = \alpha_1^{15} * \alpha_3^{12} * \alpha_1^{13} * \alpha_3^{11} * \alpha_1^{14}$	$13,12,7,5,6,8,9,2,4,1,3,10,11$	161.70

Table 7.18. Paths for the 2nd level of hierarchy (G'_2)

Structure	Vertex route	Length
$\beta_1(\alpha_1^2) = \alpha_1^{21} * \alpha_1^{22} * \alpha_1^{23}$	15, 14, 16, 17, 18, 20, 19, 21, 22	103.48
$\beta_2(\alpha_1^2) = \alpha_1^{21} * \alpha_2^{22} * \alpha_1^{23}$	15, 14, 16, 19, 17, 18, 20, 21, 22	119.73
$\beta_3(\alpha_1^2) = \alpha_1^{21} * \alpha_3^{22} * \alpha_1^{23}$	15, 14, 16, 17, 18, 19, 20, 21, 22	165.23
$\beta_4(\alpha_1^2) = \alpha_2^{21} * \alpha_1^{22} * \alpha_1^{23}$	15, 16, 14, 17, 18, 19, 20, 21, 22	114.13
$\beta_5(\alpha_1^2) = \alpha_2^{21} * \alpha_2^{22} * \alpha_1^{23}$	15, 16, 14, 19, 17, 18, 20, 21, 22	128.33
$\beta_6(\alpha_1^2) = \alpha_2^{21} * \alpha_3^{22} * \alpha_1^{23}$	15, 16, 14, 17, 18, 19, 20, 21, 22	117.33
$\beta_7(\alpha_1^2) = \alpha_3^{21} * \alpha_1^{22} * \alpha_1^{23}$	14, 15, 16, 17, 18, 19, 20, 21, 22	114.41
$\beta_8(\alpha_1^2) = \alpha_3^{21} * \alpha_2^{22} * \alpha_1^{23}$	14, 15, 16, 19, 17, 18, 20, 21, 22	120.66
$\beta_9(\alpha_1^2) = \alpha_3^{21} * \alpha_3^{22} * \alpha_1^{23}$	14, 15, 16, 17, 18, 19, 20, 21, 22	117.61
$\beta_1(\alpha_2^2) = \alpha_1^{21} * \alpha_1^{23} * \alpha_1^{22}$	15, 14, 16, 21, 22, 17, 18, 20, 19	110.98
$\beta_2(\alpha_2^2) = \alpha_1^{21} * \alpha_1^{23} * \alpha_2^{22}$	15, 14, 16, 21, 22, 19, 17, 18, 20	107.35
$\beta_3(\alpha_2^2) = \alpha_1^{21} * \alpha_1^{23} * \alpha_3^{22}$	15, 14, 16, 21, 22, 17, 18, 19, 20	114.18
$\beta_4(\alpha_2^2) = \alpha_2^{21} * \alpha_1^{23} * \alpha_1^{22}$	14, 16, 15, 21, 22, 17, 18, 20, 19	113.18
$\beta_5(\alpha_2^2) = \alpha_2^{21} * \alpha_1^{23} * \alpha_2^{22}$	14, 16, 15, 21, 22, 19, 17, 18, 20	109.55
$\beta_6(\alpha_2^2) = \alpha_2^{21} * \alpha_1^{23} * \alpha_3^{22}$	14, 16, 15, 21, 22, 17, 18, 19, 20	116.33
$\beta_7(\alpha_2^2) = \alpha_3^{21} * \alpha_1^{23} * \alpha_1^{22}$	14, 15, 16, 21, 22, 17, 18, 20, 19	111.91
$\beta_8(\alpha_2^2) = \alpha_3^{21} * \alpha_1^{23} * \alpha_2^{22}$	14, 15, 16, 21, 22, 19, 17, 18, 20	108.30
$\beta_9(\alpha_2^2) = \alpha_3^{21} * \alpha_1^{23} * \alpha_3^{22}$	14, 15, 16, 21, 22, 17, 18, 19, 20	115.11
$\beta_1(\alpha_3^2) = \alpha_1^{23} * \alpha_1^{21} * \alpha_1^{22}$	22, 21, 15, 14, 16, 17, 18, 20, 19	095.38
$\beta_2(\alpha_3^2) = \alpha_1^{23} * \alpha_2^{21} * \alpha_1^{22}$	22, 21, 15, 16, 14, 17, 18, 20, 19	096.03
$\beta_3(\alpha_3^2) = \alpha_1^{23} * \alpha_3^{21} * \alpha_1^{22}$	22, 21, 16, 15, 14, 17, 18, 20, 19	094.76
$\beta_4(\alpha_3^2) = \alpha_1^{23} * \alpha_1^{21} * \alpha_2^{22}$	22, 21, 15, 14, 16, 19, 17, 18, 20	101.63
$\beta_5(\alpha_3^2) = \alpha_1^{23} * \alpha_2^{21} * \alpha_2^{22}$	22, 21, 15, 16, 14, 19, 17, 18, 20	108.13
$\beta_6(\alpha_3^2) = \alpha_1^{23} * \alpha_3^{21} * \alpha_2^{22}$	22, 21, 16, 15, 14, 19, 17, 18, 20	106.86
$\beta_7(\alpha_3^2) = \alpha_1^{23} * \alpha_1^{21} * \alpha_3^{22}$	22, 21, 15, 14, 16, 17, 18, 19, 20	098.58
$\beta_8(\alpha_3^2) = \alpha_1^{23} * \alpha_2^{21} * \alpha_3^{22}$	22, 21, 14, 16, 15, 17, 18, 19, 20	102.53
$\beta_9(\alpha_3^2) = \alpha_1^{23} * \alpha_3^{21} * \alpha_3^{22}$	22, 21, 14, 15, 16, 17, 18, 19, 20	105.91

Table 7.19. Paths for the 2nd level of hierarchy $(G'_3,\ G'_4)$

Structure	Vertex route	Length
$\beta_1(\alpha_1^3) = \alpha_1^{31} * \alpha_1^{34} * \alpha_1^{33} * \alpha_1^{32}$	23, 24, 31, 30, 29, 28, 27, 26, 25	127.17
$\beta_2(\alpha_1^3) = \alpha_1^{31} * \alpha_1^{34} * \alpha_1^{33} * \alpha_2^{32}$	23, 24, 31, 30, 29, 28, 27, 25, 26	132.57
$\beta_3(\alpha_1^3) = \alpha_1^{31} * \alpha_1^{34} * \alpha_1^{33} * \alpha_3^{32}$	23, 24, 31, 30, 29, 28, 26, 27, 25	133.37
$\beta_4(\alpha_2^3) = \alpha_1^{34} * \alpha_1^{32} * \alpha_1^{33} * \alpha_1^{31}$	31, 30, 27, 26, 25, 28, 29, 24, 23	113.20
$\beta_5(\alpha_2^3) = \alpha_1^{34} * \alpha_2^{32} * \alpha_1^{33} * \alpha_1^{31}$	31, 30, 27, 25, 26, 28, 29, 24, 23	124.10
$\beta_6(\alpha_2^3) = \alpha_1^{34} * \alpha_3^{32} * \alpha_1^{33} * \alpha_1^{31}$	31, 30, 26, 27, 25, 28, 29, 24, 23	126.20
$\beta_7(\alpha_3^3) = \alpha_1^{34} * \alpha_1^{31} * \alpha_1^{33} * \alpha_1^{32}$	31, 30, 24, 23, 28, 29, 27, 26, 25	115.97
$\beta_8(\alpha_3^3) = \alpha_1^{34} * \alpha_1^{31} * \alpha_1^{33} * \alpha_2^{32}$	31, 30, 24, 23, 28, 29, 27, 25, 26	121.37
$\beta_9(\alpha_3^3) = \alpha_1^{34} * \alpha_1^{31} * \alpha_1^{33} * \alpha_3^{32}$	31, 30, 24, 23, 28, 29, 25, 27, 26	125.47
$\beta_1(\alpha_1^4) = \alpha_1^{41} * \alpha_1^{42} * \alpha_1^{44} * \alpha_1^{43}$	32, 33, 35, 34, 36, 39, 40, 38, 37	109.41
$\beta_2(\alpha_1^4) = \alpha_1^{41} * \alpha_2^{42} * \alpha_1^{44} * \alpha_1^{43}$	32, 33, 34, 36, 35, 39, 40, 38, 37	123.91
$\beta_3(\alpha_1^4) = \alpha_1^{41} * \alpha_3^{42} * \alpha_1^{44} * \alpha_1^{43}$	32, 33, 34, 35, 36, 39, 40, 38, 37	120.02
$\beta_4(\alpha_2^4) = \alpha_1^{42} * \alpha_1^{44} * \alpha_1^{43} * \alpha_1^{41}$	35, 34, 36, 39, 40, 38, 37, 33, 32	094.81
$\beta_5(\alpha_2^4) = \alpha_2^{42} * \alpha_1^{44} * \alpha_1^{43} * \alpha_1^{41}$	34, 36, 35, 39, 40, 38, 37, 33, 32	099.51
$\beta_6(\alpha_2^4) = \alpha_3^{42} * \alpha_1^{44} * \alpha_1^{43} * \alpha_1^{41}$	34, 35, 36, 39, 40, 38, 37, 33, 32	095.62
$\beta_7(\alpha_3^4) = \alpha_1^{42} * \alpha_1^{44} * \alpha_1^{41} * \alpha_1^{43}$	35, 34, 36, 39, 40, 32, 33, 38, 37	120.41
$\beta_8(\alpha_3^4) = \alpha_2^{42} * \alpha_1^{44} * \alpha_1^{41} * \alpha_1^{43}$	34, 36, 35, 39, 40, 32, 33, 38, 37	125.11
$\beta_9(\alpha_3^4) = \alpha_3^{42} * \alpha_1^{44} * \alpha_1^{41} * \alpha_1^{43}$	34, 35, 36, 39, 40, 32, 33, 38, 37	121.21

Mainly, we examine the only one way for each elementary path. Evidently, we can select the best path while taking into account their length, and start and end points, for example:

$\beta_5(\alpha_1^1) \succ \beta_2(\alpha_1^1) \succ \beta_3(\alpha_1^1)$; and $\beta_4(\alpha_1^1) \succ \beta_1(\alpha_1^1)$.

Thus we select the following nondominated composite paths (the 2nd level):

(a) G'_1: $\beta_4(\alpha_1^1)$, points: 7, 12; $\beta_5(\alpha_1^1)$, points: 6, 12; $\beta_1(\alpha_2^1)$, points: 7, 11; $\beta_2(\alpha_2^1)$, points: 6, 11; $\beta_4(\alpha_3^1)$, points: 13, 11;

(b) G'_2: $\beta_4(\alpha_1^2)$, points: 15, 22; $\beta_7(\alpha_1^2)$, points: 14, 22; $\beta_1(\alpha_2^2)$, points: 15, 19; $\beta_2(\alpha_2^2)$, points: 15, 20; $\beta_7(\alpha_2^2)$, points: 14, 19; $\beta_8(\alpha_2^2)$, points: 15, 20; $\beta_3(\alpha_3^2)$, points: 22, 19; $\beta_7(\alpha_3^2)$, points: 22, 20;

(c) G'_3: $\beta_1(\alpha_1^3)$, points: 23, 25; $\beta_2(\alpha_1^3)$, points: 23, 26; $\beta_4(\alpha_2^3)$, points: 31, 23; $\beta_7(\alpha_3^3)$, points: 31, 25; $\beta_8(\alpha_3^3)$, points: 31, 26; and

(d) G'_4: $\beta_1(\alpha_1^4)$, points: 32, 37; $\beta_4(\alpha_2^4)$, points: 35, 32; $\beta_6(\alpha_2^4)$, points: 34, 32; $\beta_9(\alpha_3^4)$, points: 34, 37.

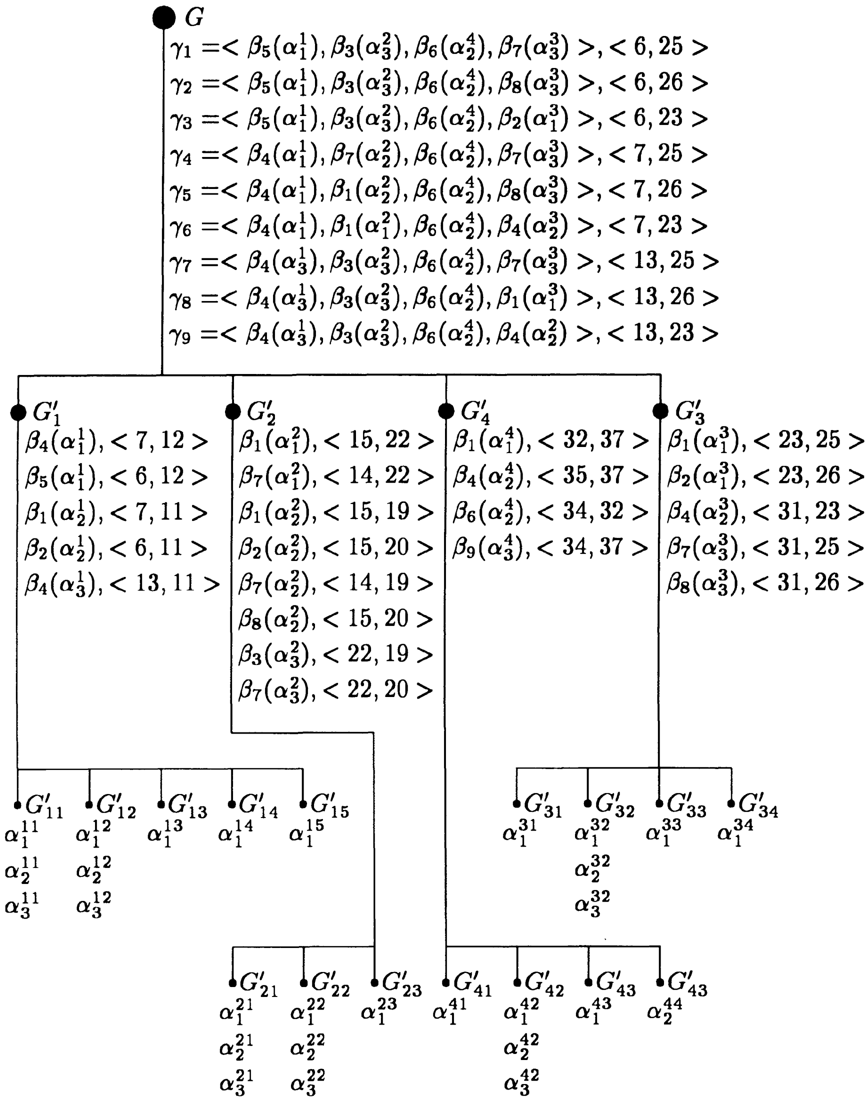

Fig. 7.18. Structure of composite decision

Clearly, building of the paths (the second level of hierarchy) corresponds to a well-known ***set-to-set shortest path*** problem that is polynomial one. We have considered all steps to demonstrate solving scheme for more complicated situations above (e.g., ordinal scales, vector estimates).

Fig. 7.18 depicts a hierarchical structure of our composite decision (start and end points of paths at the higher hierarchical level are pointed out too).

In the example, a simple Hamiltonian circuit of the higher level can be examined as follows: $G'_1, G'_2, G'_4, and G'_3$. Here we use ***set-to-set shortest path*** problem also:

(i) start points: 6, 7, 13; and

(ii) end points: 25, 26, 23.

As a result, the following 9 best paths are built:

(1) $< 6, 25 >$: $\gamma_1 =< \beta_5(\alpha_1^1), \beta_3(\alpha_3^2), \beta_6(\alpha_2^4), \beta_7(\alpha_3^3) >$ (522.85);

(2) $< 6, 26 >$: $\gamma_2 =< \beta_5(\alpha_1^1), \beta_3(\alpha_3^2), \beta_6(\alpha_2^4), \beta_8(\alpha_3^3) >$ (528.25);

(3) $< 6, 23 >$: $\gamma_3 =< \beta_5(\alpha_1^1), \beta_3(\alpha_3^2), \beta_6(\alpha_2^4), \beta_2(\alpha_1^3) >$ (520.08);

(4) $< 7, 25 >$: $\gamma_4 =< \beta_4(\alpha_1^1), \beta_7(\alpha_2^2), \beta_6(\alpha_2^4), \beta_7(\alpha_3^3) >$ (525.25);

(5) $< 7, 26 >$: $\gamma_5 =< \beta_4(\alpha_1^1), \beta_1(\alpha_2^2), \beta_6(\alpha_2^4), \beta_8(\alpha_3^3) >$ (530.65);

(6) $< 7, 23 >$: $\gamma_6 =< \beta_4(\alpha_1^1), \beta_1(\alpha_1^2), \beta_6(\alpha_2^4), \beta_4(\alpha_2^3) >$ (511.83);

(7) $< 13, 25 >$: $\gamma_7 =< \beta_4(\alpha_3^1), \beta_3(\alpha_3^2), \beta_6(\alpha_2^4), \beta_7(\alpha_3^3) >$ (560.97);

(8) $< 13, 26 >$: $\gamma_8 =< \beta_4(\alpha_3^1), \beta_3(\alpha_3^2), \beta_6(\alpha_2^4), \beta_1(\alpha_1^3) >$ (566.37); and

(9) $< 13, 23 >$: $\gamma_9 =< \beta_4(\alpha_3^1), \beta_3(\alpha_3^2), \beta_6(\alpha_2^4), \beta_4(\alpha_2^2) >$ (558.20).

Lengths of closing arcs are the following: $\rho(6, 25) = 61.00$; $\rho(6, 26) = 70.40$; $\rho(6, 23) = 43.00$; $\rho(7, 25) = 50.70$; $\rho(7, 26) = 59.30$; $\rho(7, 23) = 33.00$; $\rho(13, 25) = 32.10$; $\rho(13, 26) = 40.40$; and $\rho(13, 23) = 12.50$.

Thus the best Hamiltonian circuit (a path takes into account a length of the closing arc) is (Fig. 7.19):

$\gamma_6 =< \beta_4(\alpha_1^1), \beta_4(\alpha_2^2), \beta_6(\alpha_2^4), \beta_4(\alpha_2^3) >$ (544.83).

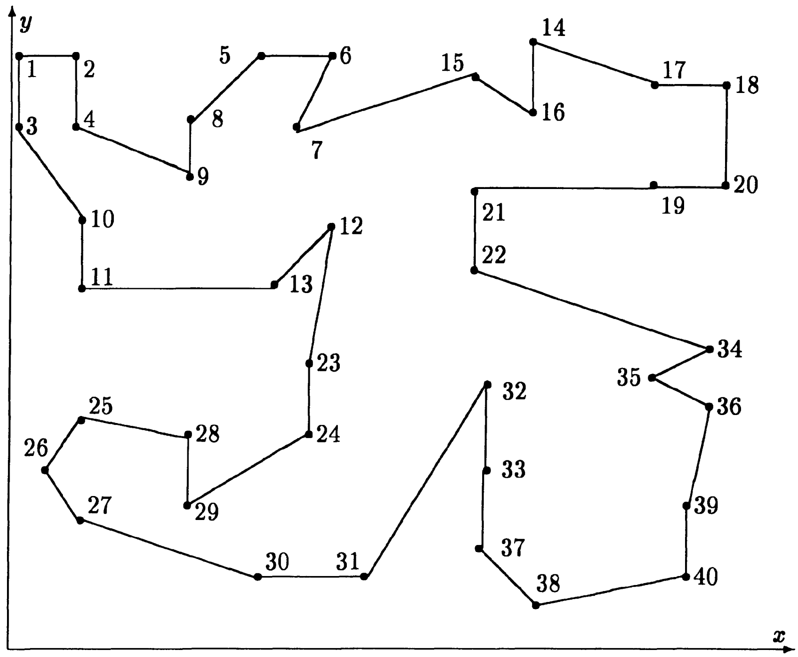

Fig. 7.19. Resultant Hamiltonian circuit

7.3.5 Towards Generalized Problems

We have demonstrated the numerical example for TSP in the plane (i.e., scalar parameters), but our solving framework is oriented to a general case. The following problem classification parameters may be considered:

(a) kinds of scales (scalar, ordinal, vector-like);

(b) types of objective function (additive, vector-like); and

(c) kinds of metrics or proximity between vertices.

Let us recall that our morphological approach is often based on mapping vector-like estimates into ordinal scales. Table 7.20 illustrates correspondence between kinds of scales/objectives and methods (algorithmic complexity is pointed out in brackets).

It can easily be checked that many problems with vector-like scales have polynomial complexity (as corresponding problems with scalar scales), if a number of Pareto-effective points is limited by a certain value. For example, this note is correct for modifications of the vector path problem.

Table 7.20. Stages, scales and problems

Stage	Scalar estimates, and additive objective	Ordinal/vector estimates, and vector objective
1. Partitioning of graph (**P1**)	Hierarchical clustering (polynomial heuristics)	Hierarchical clustering (polynomial heuristics)
2. Generation of local DA's, i.e. Hamiltonian paths (**P2, P3**)	Generation of start/ end points for paths (effective algorithms)	Generation of start/ end points for paths (effective schemes)
	Building of paths (NP-hard, enumerative algorithms)	Building of paths (NP-hard, enumerative algorithms)
3. Computation of compatibility between DA's (**P4**)	Computation of compatibility as distance between paths	Expert judgement Multicriteria ranking
4. Composing of composite Hamiltonian paths (**P5**)	*Set-to-set* path problem (polynomial algorithms)	*Set-to-set* vector path problem (NP-hard, polynomial heuristics) Morphological clique problem (chain-like compatibility structure, polynomial scheme for some cases)
5. Composing of resultant Hamiltonian circuit (**P6**)	*Set-to-set* path problem, and selection of the best path (polynomial algorithms)	Morphological clique problem (chain-like compatibility structure, polynomial schemes for some cases)

7.4 STEINER MINIMAL TREE PROBLEM

A Steiner Minimal Tree Problem (SMTP) in the plane is the following.

Let $A = \{a_1, ..., a_n\}$ be a given point set. It is required to build a tree which interconnects the points of A, and total length of the tree is minimum. To achieve the minimum, the tree may contain other additional vertices, which are called *Steiner points*.

In addition, the following two close problems are well-known:
(i) SMTP with rectangular metric; and
(ii) Graph Steiner Tree Problem (on a graph with weighted arcs).

These problems have a lot of applications (e.g., network design, VLSI design, mechanical engineering, computation geometry, and mathematical biology). Within the last decades, SMTP has been very intensively studied. Basic surveys are contained in ([81], [180], and [535]). SMTP and the majority of its modifications are NP-hard ([160], etc.). The following approaches are applied to the problems:

(1) exact algorithms on the basis of the following: (i) enumerative schemes and reduction techniques (to reduce an n-point problem to a set of $(n-1)$-point problems) ([84], [352], etc.); (ii) dynamic programming ([122], [192], [280], etc.);

(2) approximation algorithms ([43], [350], [551], etc.);

(3) heuristics including local optimization, greedy algorithms, decomposition ([123], [494], etc.); and

(4) polynomial algorithms for some simple cases (e.g., parallel-series graphs).

In this section, we illustrate the use of our morphological heuristic to an example of SMTP in the plane. Note that this approach is similar to the algorithm of Dreyfus and Wagner (series partitioning of the problem into small ones and computing of local solutions for resultant small problems on the basis of the shortest paths for vertex pairs) [122].

We consider 22 points which were used in previous section for TSP (Table 7.12.), but their coordinates on axis y are reduced on 40 (Fig. 7.20).

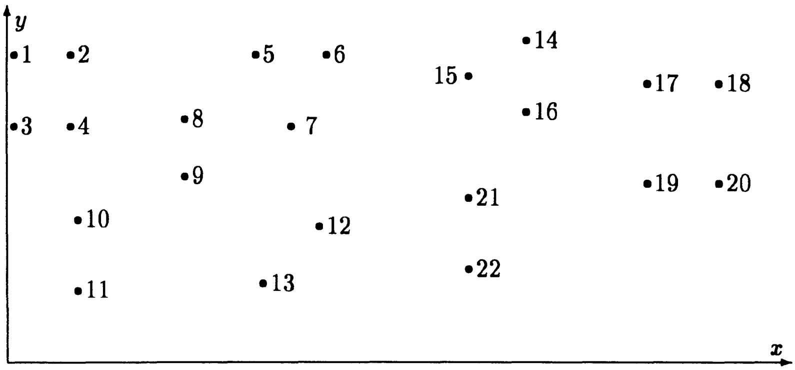

Fig. 7.20. Example for SMTP (initial graph G)

Analogically, we consider the partitioning of G into the following parts (Fig. 7.9):

(a) the 1st level: G'_1, and G'_2; and

(b) the 2nd level: G'_{11}, G'_{12}, G'_{13}, G'_{14}, G'_{15}, and G'_{21}, G'_{21}, G'_{21}.

Fig. 7.21 depicts DA's for the 1st hierarchical level (G'_1, G'_2).

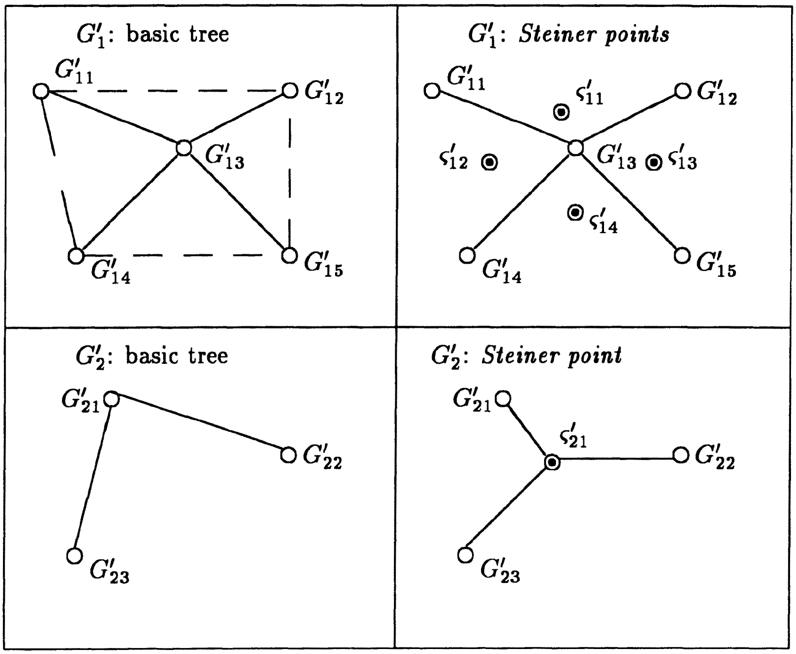

Fig. 7.21. Basic DA's G'_1, G'_2

Thus it is reasonable to consider the following trees at the 1st level:

1. G'_1: basic tree β'_{10}; modified basic tree with Steiner point (ς'_{11}) β'_{11}; modified basic tree with Steiner point (ς'_{12}) β'_{12}; modified basic tree with Steiner point (ς'_{13}) β'_{13}; modified basic tree with Steiner point (ς'_{14}) β'_{14}; modified basic tree with Steiner points (ς'_{11} and ς'_{14}) β'_{15}; and modified basic tree with Steiner points (ς'_{12} and ς'_{13}) β'_{16}.

2. G'_2: basic tree β'_{20}; modified basic tree with Steiner point β'_{21}.

Analogically, we can get the following trees at the 2nd level (Fig. 7.14, 7.15):

1. G'_{11}: basic tree α_1^{11}; basic tree α_2^{11}; basic tree α_3^{11}; a modified basic tree for G'_{11} with a Steiner point at the 'center' of the corresponding rectangle α_0^{11}.

2. G'_{12}: basic tree α_1^{12}; basic tree α_2^{12}; basic tree α_3^{12}; modified basic tree with a Steiner point at the 'center' of the corresponding triangle α_0^{12}.

3. G'_{21}: basic tree α_1^{21}; basic tree α_2^{21}; basic tree α_3^{21}; modified basic tree with a Steiner point at the 'center' of the corresponding triangle α_0^{21}.

4. G'_{22}: basic tree α_1^{22}; basic tree α_2^{22}; basic tree α_3^{22}; a modified basic tree for G'_{22} with a Steiner point at the 'center' of the corresponding rectangle α_0^{22}.

Finally, we get the structure of a composite decision (Fig. 7.22).

Here $\kappa_1 = 0, 1, 2, 3$; $\kappa_2 = 0, 1, 2, 3$; $\iota_1 = 0, 1, 2, 3$; $\iota_2 = 0, 1, 2, 3$.

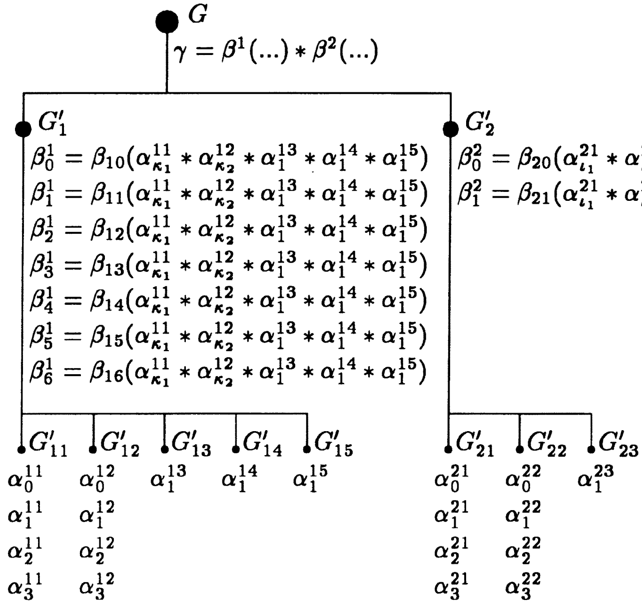

Fig. 7.22. Structure of composite decision

Fig. 7.23 depicts the composite decision as the resultant Steiner tree with 6 Steiner points as follows:

(a) 2 points at the 1st level (ς'_{13}, and ς'_{21}), and

(b) 6 points at the 2nd level.

The formal description of the decision is:

$$\gamma = \beta'_{13}(\alpha^{11}_0 * \alpha^{12}_0 * \alpha^{13}_1 * \alpha^{14}_1 * \alpha^{15}_1) * \beta'_{21}(\alpha^{21}_0 * \alpha^{22}_0 * \alpha^{23}_1).$$

Note that often the number of Steiner points is limited. It follows easily that our heuristic can be used for the situation too: to select at each stage decisions with a restriction to the number of Steiner points.

7.5 SCHEDULING

7.5.1 Generalized Description

Let us examine a static scheduling problem: to schedule (to order) n jobs. In this case, we assume that a design system consists of n components (positions) of a resultant schedule. Also, we can consider an initial set of jobs (or tasks) $J = \{J_1, ..., J_i, ..., J_n\}$ as a set of DA's for each position above. Let

$$S =< s[1], ..., s[k], ..., s[n] >$$

be a schedule of the jobs, where $s[k]$ is the number of a job at the kth position (Fig. 7.24).

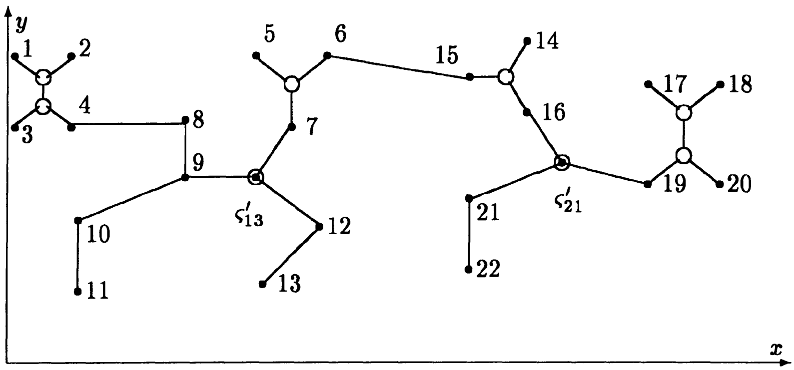

Fig. 7.23. Example of Steiner tree with 8 Steiner points

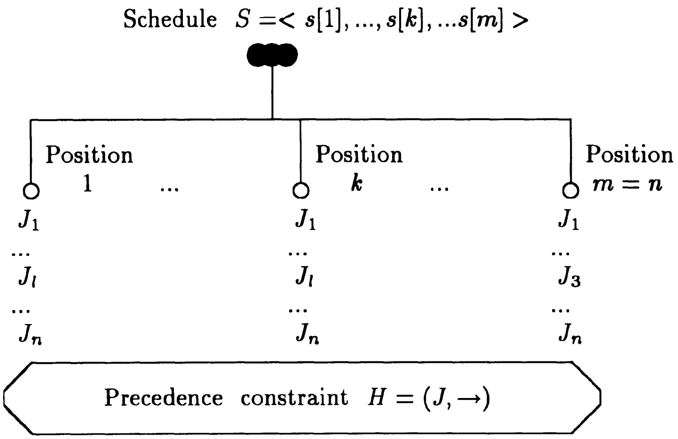

Fig. 7.24. Illustration for m-position scheduling

Quality (priority) of an assignment of a job to position l ($l = 1, ..., n$) may be based on some problem parameters. For example, in some scheduling problems (e.g., Bellman-Johnson problem, problems with minimizing a schedule's length or *makespan*) we can apply a priority function for linear ordering of jobs, and the order number of a job will be its priority ([51], [233], [288], [479], etc.). Other problem parameters (e.g., precedence constraint as a digraph $H = (J, \rightarrow)$, where J is a set of vertices-jobs) may be used to compute Ins. Thus we can obtain the basic morphological clique problem, or its modifications. Note that some investigators study compatibility of jobs in scheduling problems ([54], [222], etc.). Now let us describe the following three rules:

Rule 7.5.1. J_i and J_i are incompatible, when they occupy different positions.

Rule 7.5.2. If $J_i \rightarrow J_j$ then
$w(s[k_1], s[k_2]) = 0, \ \forall k_1, k_2, \ k_1 < k_2$, and $s[k_1] = J_j, \ s[k_2] = J_i$.

Let $\alpha(J_{i_1}, J_{i_2}) =< J_{i_1}, ..., J_{i_2} >$ be a path from J_{i_1} to J_{i_2}, a length of the path $|\alpha|$ be equal to the number of its vertices, and $\alpha^m(J_{i_1}, J_{i_2})$ be the path of maximal length.

Rule 7.5.3. If $s[k_1] = J_{i_1}, \ s[k_2] = J_{i_2}, \ \exists \ \alpha(J_{i_1}, J_{i_2})$, then
$|k_1 - k_2| \geq |\alpha^m(J_{i_1}, J_{i_2})| - 1$ otherwise $w(J_{i_1}, J_{i_2}) = 0$.

Furthermore, we obtain the following evident lemma:

Lemma 7.1. Rule 7.5.1, 7.5.2 and 7.5.3 are sufficient conditions to generate admissible schedules.

7.5.2 Class of Problems with Precedence Constraints

In this section, the following 6 scheduling problems are considered ([233], [288], [479], etc.):

Problem 1. There is one machine. Each job J_i is characterized by processing time τ_i and penalty function $\varphi(t) = a_i t + b_i \quad a, b \geq 0$. It is required to minimize the total penalty $f(S) = \sum_{i=1}^{n} \varphi_i(C_i)$, where C_i is a moment of the completion of job J_i (all jobs start at the moment $t = 0$).

Problem 2. This problem differs from **Problem 1** by another penalty function: $\varphi_i(t) = a_i exp(\lambda t) + b_i, \ \lambda \geq 0$.

Problem 3. This problem differs from **Problem 1** by another total penalty function: $f(S) = \sum_{i=1}^{n} \varphi_i(t_i)$, where $t_i = \prod_{l=1}^{i} \tau_{p[l]}$.

Problem 4. This problem differs from **Problem 3** by another penalty function: $\varphi_i(t) = a_i ln(\lambda t) + b_i$.

Problem 5. This is a Bellman-Johnson problem (two machines). All jobs must be processed at the same linear order (the 1st machine and the 2nd machine). Processing times at machines are the following: a_i and b_i, respectively. The objective is to minimize the total processing time for all jobs.

Problem 6. There is one machine. Each job J_i is characterized by processing time τ_i and probability of stoppage of the processing η_i. The objective is to minimize the average cost of all jobs $f(S) = \sum_{l=1}^{n} \tau_{s[l]}/\delta_{s[l]}$, where $\delta_{s[l]} = \prod_{k=1}^{s[l]}(1 - \eta_k)$. Note this problem can be reduced to **Problem 1** by a change of variables.

In the case there is no precedence constraint, the above-mentioned problems can be solved by the following algorithm:

Step 1. Computation of real-valued function for each job i:
$\varrho(i) = \tau_i/a_i$ (**Problem 1**);
$\varrho(i) = a_i exp(\lambda\tau_i)[1 - exp(\lambda\tau_i)]^{-1}$ (**Problem 2**);
$\varrho(i) = a_i\tau_i/(1 - \tau_i)$ (**Problem 3**);
$\varrho(i) = a_i ln(\lambda\tau_i)/(1 - ln(\lambda\tau_i))$ (**Problem 4**);
$\varrho(i) = sign(a_i - b_i)[\sum_{k=1}^{n}(a_k - b_k) - min(a_i, b_i)]$ (**Problem 5**); and
$\varrho(i) = \tau_i/\eta_i$ (**Problem 6**).
Step 2. Linear ordering of the jobs by nondecreasing of $\varrho(i)$.

Clearly, this optimization algorithm runs in polynomial time $O(n \ln n)$. Moreover, an extension of the algorithm can be used for the above-mentioned six problems with series-parallel precedence constraints too ([221], [288], [370], [468], etc.). Investigations of polynomial-time algorithms for scheduling problems with more general precedence constraints are presented in ([58], [67], [371], [376], etc.).

Now let us consider a heuristic on the basis of our morphological approach for the problems with precedence constraints as a digraph $H = (J, \rightarrow)$. In the previous section it was pointed out that, H is a base to construct compatibility. Also, computation of priorities for DA's may be based on the following:

(1) to find an optimal schedule in the case of a *good* structure (e.g., without precedence constraint, a spanning series-parallel digraph):

$$S^* =< s^*[1], ..., s^*[k], ..., s^*[n] >;$$

(2) to compute for each job J_i and each position l a priority $r(J_i, l)$ as follows:

$$r(J_i, l) = m - |l - s^*(J_i)|,$$

where $s^*(J_i)$ equals the number of a position that is occupied by job J_i in schedule S^*.

Thus we obtain a problem that may be solved on the basis of morphological clique problem.

7.5.3 Towards Priorities of Jobs

Let us consider a situation when job J_i can not occupy all positions $1, ..., m$, for example, by the following reasons:

(a) limited time interval to obtain external resources (e.g., equipment);

(b) plans of contractors; and

(c) specified ***arrival time*** $t_s(J_i)$ and ***due date*** $t_d(J_i)$.

As a result, it is reasonable to examine a special ***feasible interval*** (or ***boundary*** [328]) and ***desirability function*** to set time and preference for processing of job J_i. In the simplest case, we can consider time interval

$$t(J_i) = [t_s(J_i), t_d(J_i)].$$

to process J_i. In addition, it is possible to use a preference (probability) to process J_i as a nonnegative function χ_i, defined at the interval $t(J_i)$.

In the case of our morphological approach, we have to map χ_i to integer axis $1, ..., n$ (Fig. 7.25, case of single machine).

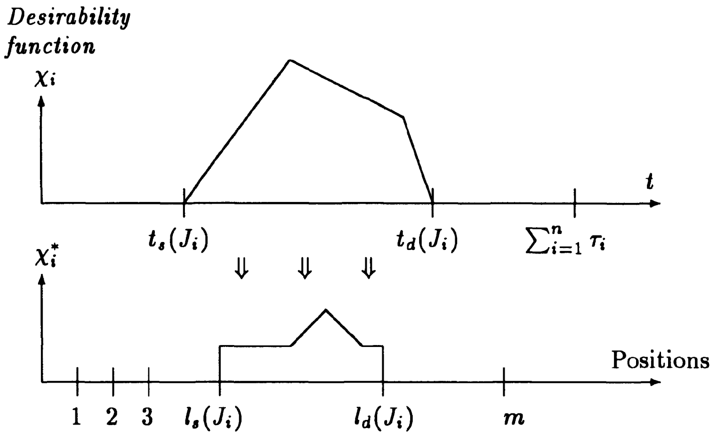

Fig. 7.25. Illustration for ***feasible interval*** and ***desirability function***

Now we describe an example for single machine scheduling. Let υ_κ be κth minimum of the ordered list of processing times $\{\tau_i\}$. Then $l_s(J_i) = l' + 1$, where l' is defined as a minimal integer number that satisfies to the following inequality:

$$\sum_{\kappa=1}^{l'} v_\kappa \geq t_s(J_i).$$

In the same way, $l_d(J_i) = n - (l'' + 1)$, where l'' is defined as a minimal integer number that satisfies to the following inequality:

$$\sum_{\kappa=1}^{l''} v_\kappa \geq \sum_{i=1}^{n} \tau_i - t_d(J_i).$$

Thus we obtain *feasible interval* of positions that will provide correct processing of job J_i.

It is obvious that we can specify admissible positions (on the basis of *feasible interval* $[l_s(J_i), l_d(J_i)]$ and priorities at positions above (on the basis of priorities function $\chi_i(J_i)$) for each job J_i.

Analogically, we examine the second way to define *feasible intervals* on the basis of precedence constraints. First, let $J^- \subseteq J$ be a ***start subset*** if $\forall J_{i_1} \in J^-$ has no incoming arcs, and $J^+ \subseteq J$ be an ***end subset*** if $\forall J_{i_2} \in J^-$ has no outgoing arcs. Fig. 7.26 depicts an example of precedence constraints for five-element scheduling problem.

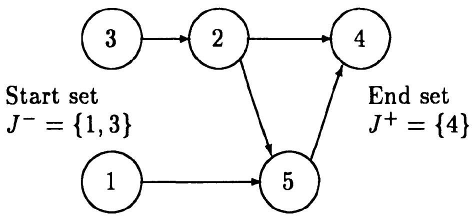

Fig. 7.26. Example of precedence constraints

In this case, $J^- = \{J_1, J_3\}$ and $J^+ = \{J_4\}$. In addition, we define $\forall J_i \in J$ two kinds of paths:

$$\alpha^+(J_i) =< J_i, ..., J_e >, \quad J_e \in J^+,$$

$$\alpha^-(J_i) =< J_s, ..., J_i >, \quad J_s \in J^-.$$

Finally, it is easy to prove that:

Lemma 7.2. Necessary condition for a schedule to be admissible is that the following *feasible interval* system

$$[\, d^-(J_i) - 1,\ m - d^+(J_i) + 1 \,],\quad \forall J_i \in J$$

where

$$d^+(J_i) = \max_i\{|\alpha^+(J_i)|\},\quad d^-(J_i) = \max_i\{|\alpha^-(J_i)|\}$$

be fulfilled.

Now we assume that our scheduling problem (Fig. 7.26) corresponds to **Problem 1**. Evidently, without taking into account precedence constraints the optimal schedule is the following (J_3 and J_3 are equivalent):

$$S^* =< J_5,\ J_2,\ J_3,\ J_4,\ J_1 >,\quad T(S^*) = 19.0.$$

As before, the *desirability function* (priority) is:

$$r(J_i, l) = m - |l - s^*(J_i)|,$$

where $s^*(J_i)$ be the position number of J_i in optimal schedule S^*, $l \in [d^+(J_i), d^-(J_i)]$. Table 7.22 presents parameters and priorities of jobs.

Table 7.22. Parameters and priorities

J_i	τ_i	a_i	$\varrho_i = \tau_i / a_i$	$r(J_i, l)$				
			$l =$ 1	2	3	4	5	
J_1	2.0	1.0	2.0	1	2	3	–	–
J_2	1.5	1.5	1.0	–	5	4	3	–
J_3	1.5	1.0	1.5	3	4	5	–	–
J_4	1.5	1.0	1.5	–	–	–	–	4
J_5	0.5	1.0	0.5	–	4	3	2	–

Thus we get the possibility to apply morphological clique problem. Clearly, our situation does not exactly correspond to Fig. 7.24., because each morphological class can include only a subset of the initial job set. Fig. 7.27 depicts resultant morphology of schedule (*alternative jobs* J'_{12} and J'_{23} will be described in next section). Compatibility of jobs is presented in Table 7.23.

$S_1 = J_{31} * J_{22} * J_{13} * J_{54} * J_{45}$
$S_2 = J_{11} * J_{32} * J_{23} * J_{54} * J_{45}$
$S_3 = J_{31} * J_{12} * J_{23} * J_{54} * J_{45}$

Position 1	Position 2	Position 3	Position 4	Position 5
$J_{11}(1)$	$J_{12}(2)$	$J_{13}(3)$	$J_{24}(3)$	$J_{45}(4)$
$J_{31}(3)$	$J_{22}(5)$	$J_{23}(4)$	$J_{54}(2)$	
	$J_{32}(4)$	$J_{33}(5)$		
	$J_{52}(4)$	$J_{53}(3)$		
	$J'_{12}(2)$	$J'_{23}(4)$		

Fig. 7.27. Schedule morphology

Table 7.23. Compatibility of jobs

	J_{12}	J_{22}	J_{32}	J_{52}	J_{13}	J_{23}	J_{33}	J_{53}	J_{24}	J_{54}	J_{45}
J_{11}	0	1	1	1	0	1	1	1	1	1	1
J_{31}	1	1	0	0	1	1	0	1	1	1	1
J_{12}					0	1	1	1	1	1	1
J_{22}					1	0	0	1	0	1	1
J_{32}					1	1	0	0	1	1	1
J_{52}					0	0	0	0	0	0	1
J_{13}									1	1	1
J_{23}									0	1	1
J_{33}									1	0	0
J_{53}									0	0	1
J_{24}											1
J_{54}											1

Finally, we obtain the following three composite DA's (admissible schedules) that are Pareto-effective:

(1) $S_1 = J_{31} * J_{22} * J_{13} * J_{54} * J_{45}$, $N(S_1) = (1; 1, 1, 2, 1, 0)$, $T(S_1) = 23.5$;

(2) $S_2 = J_{11} * J_{32} * J_{23} * J_{54} * J_{45}$, $N(S_2) = (1; 0, 3, 0, 1, 1)$; $T(S_2) = 25.5$;

and

(3) $S_3 = J_{31} * J_{12} * J_{23} * J_{54} * J_{44}$, $N(S_3) = (1; 0, 2, 1, 2, 0)$; $T(S_3) = 25.0$.

Clearly, S_1 is optimal. It is obvious that our example is only illustrative and can be transformed (by assignment of J_4 at the 5th position) to the problem with tree-like precedence constraints.

7.5.4 Alternative Jobs

In addition, it is reasonable to examine ***alternative jobs***. In this case, we can consider $\forall\ J_i$ several alternative processing ways: $\{J_i^{\iota},\ \iota = 1,2,3,...\}$. Each ***alternative job*** J_i^{ι} can have specific characteristics (e.g., processing time, ***feasible interval***, ***desirability function***, etc.). Evidently, that morphological classes for schedule positions are extended by ***alternative jobs***. Also, resultant scheduling problem consists of two kinds of basic operations as follows:

(i) to select the best ***alternative jobs***; and

(ii) to schedule jobs.

This problem is realistic. It follows easily that morphological clique problem is the good base to take into account ***alternative jobs***. Let us add in the example from previous section two ***alternative jobs***:

(1) J_1', $[l_s(J_1'), l_e(J_1')] = \{2\}$, $r(J_1', 2) = 2$, $\tau(J_1') = 1.8$, $a(J_1') = 1.0$; and

(2) J_2', $[l_s(J_2'), l_e(J_2')] = \{3\}$, $r(J_2', 3) = 4$, $\tau(J_2') = 1.4$, $a(J_2') = 1.5$.

In this case, schedule morphology involves two DA's: J_{12}' and J_{23}' (Fig. 7.27) compatibility of that are equal to compatibility of J_{12} and J_{23}, respectively. Also, additional Pareto-effective composite DA's are the following:

(a) $S_4 = J_{31} * J_{12}' * J_{23} * J_{54} * J_{44}$, $N(S_3) = (1; 0, 2, 1, 2, 0)$; $T(S_4) = 24.1$;

(b) $S_5 = J_{11} * J_{32} * J_{23}' * J_{54} * J_{45}$, $N(S_2) = (1; 0, 3, 0, 1, 1)$; $T(S_2) = 24.15$; and

(c) $S_6 = J_{31} * J_{12} * J_{23}' * J_{54} * J_{44}$, $N(S_3) = (1; 0, 2, 1, 2, 0)$; $T(S_3) = 23.65$.

7.5.5 Flow Shop Problem

Let $\tau(i, j)$ be a processing time for the ith job, and jth machine ($j = 1, ..., \nu$). We assume a series processing of each job i at machines $1, ..., j, ..., \nu$, i.e., the same chain for each job $(1, 2, 3, ...)$. The objective is to produce schedule S that minimizes maximum total completion time $T(S)$. There are the following three basic types of the ***flow shop*** problem ($\nu \geq 3$) ([51], [85], [88],[160], [464], etc.):

Problem A. Basic problem (Bellman-Johnson problem), i.e. with intermediate capacities. Very interesting approximate algorithm with limited absolute error has been proposed for the problem ([36], [456], [457]).

Problem B. It is **Problem A** subject to a restriction

'without intermediate capacities'.

The problem corresponds to well-known plan approach ***just-in-time***. In this case, some enumerative exact methods have been used (e.g., Branch-and-Bound algorithm in [318]).

Problem C. Continuous process (***no-wait constraints***). The problem corresponds to traveling salesman problem ([172], etc.).

These three problems (**A**, **B**, **C**) are NP-hard ([160], [428], etc.). In [318] grid-like digraph Q has been proposed for problem **B**. We analyze the graph (Fig. 7.28) for three kinds of problems above:

1. Vertices of Q correspond to processing operations, i.e., each job at each machine: (i) row j corresponds to machine j, (ii) column k corresponds to a job that occupies position k in schedule S.

2. Arcs correspond to a way from one operation to another, but specification of the arc set is dependent on problem kinds (**A**, **B**, **C**).

Let the vertices have weights that equal processing times. In addition, we use layers for each $l = k + j = const$ $(l = 2, ..., n + \nu)$.

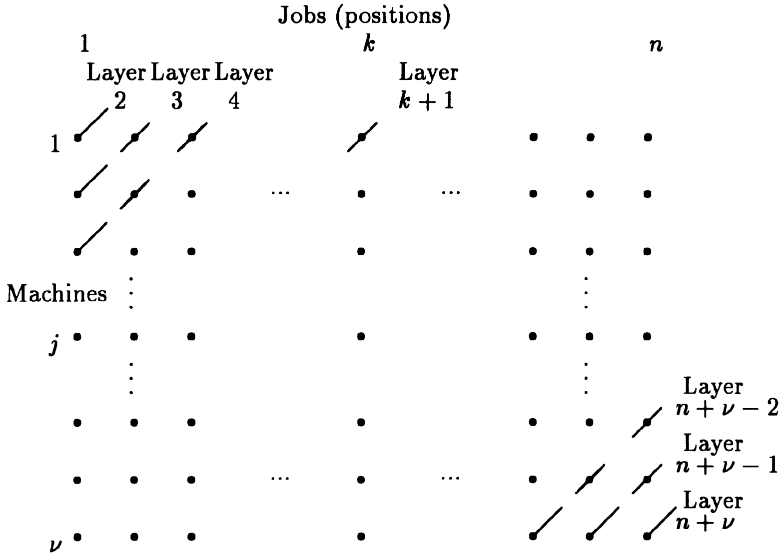

Fig. 7.28. Graph for Flow Shop Schedule (without arcs)

Now let us consider the following types of arcs:

(1) E_1 involves arcs of the following kinds ($\forall\ j = 1, ..., \nu,\ k = 1, ..., n$): $(j, k) \rightarrow (j, k + 1)$ or $(j, k) \rightarrow (j + 1, k)$;

(2) E_2 involves arcs of the following kinds: $a \rightarrow b$, where $a = (j_a, k_a)$, $b = (j_b, k_b)$, $l_a = j_a + k_a$, $l_b = j_b + k_b$, $l_b = l_a + 1$, $k_b > k_a$; and

(3) E_3 involves arcs of the following kinds: $a \rightarrow b$, where $a = (j_a, k_a)$, $b = (j_b, k_b)$, $l_a = j_a + k_a$, $l_b = j_b + k_b$, $l_b = l_a$, $k_b = k_a + 1$.

Fig. 7.29, 7.30, and 7.31 illustrate three problems above **A**, **B**, **C** (*Gantt charts*). Processing times $\{t(i,j)\}$ are pointed out in Fig. 7.29.

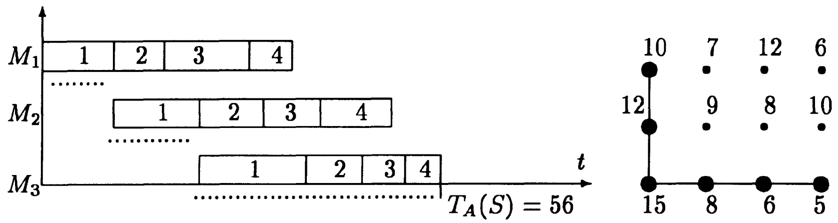

Fig. 7.29. Illustration for **Problem A**

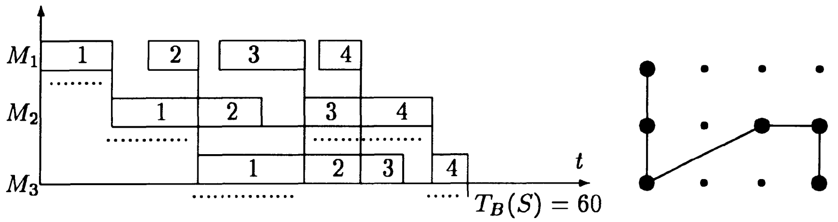

Fig. 7.30. Illustration for **Problem B**

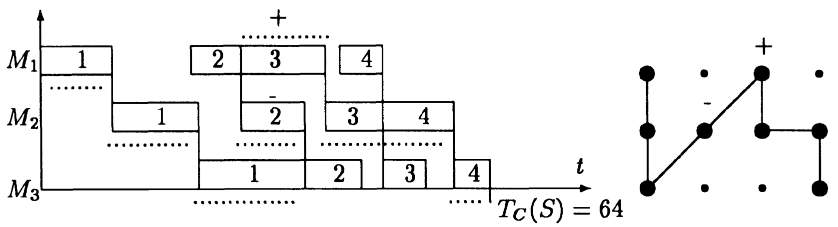

Fig. 7.31. Illustration for **Problem C**

Now let us consider the following:

Lemma 7.3. [318]. Optimal schedule (linear ordering of jobs/columns) for **Problem B** corresponds to minimum of the maximal (critical) path (with sum of vertex weights) from vertex $(1,1)$ to vertex (n,ν) in digraph Q with the following arc set: $R_b = E_1 \bigcup E_2$.

We add the following:

Lemma 7.4. For **Problem C** it is necessary to consider the following arc set: $R_C = E_1 \bigcup E_2 \bigcup E_3$. In the case when the path consists of several elements at a layer, corresponding weights of the layer elements are added with alternative signs (+, -, ...).

Proof of the necessity to examine E_3 is based on the example that is presented in Fig. 7.31.

A maximal processing time at a layer l for schedule S is

$$\mu(S,l) = \max_{i+j=l}\{\tau(s[i], s[j])\}.$$

Let μ_κ be the κth maximum or *bottleneck* of the ordered list of $\{\tau(i,j)\}$. The following is evident for **Problem A**, and **Problem B**:

Lemma 7.5. $T(S) \leq \Xi(S) = \sum_{l=2}^{n+\nu} \mu(S,l) \leq \sum_{\kappa=1}^{n+\nu-1} \mu_\kappa$.

It is reasonable to decrease $\Xi(S)$ by an *inclusion* of several maximal *bottlenecks* ($\{\mu_\kappa\}$) into the same layers.

Now let us consider a kind of the resultant graph.

Definition 7.1. Digraph $G = (A, E)$ is called *k-transitive chain* or *path* ($k \geq 2$)

if it has the following structure:

(1) elements $A =< a_1, ..., a_i, ..., a_n >$ are linear ordered $a_i \rightarrow a_{i+1}$, $i = 1, ..., n-1$ and,

(2) $\forall a_i$ is connected with κ 'right' (consequent) neighbors (elements) $\kappa \leq k-1$ and with κ_1 'left'(preceding) neighbors (elements) $\kappa \leq k-1$.

An example of 3-*transitive path* is presented in Fig. 7.32. Clearly, each sequence of l neighbors ($l \leq k$) in G is l-element clique. Let $b(A) = \sum_{e \in E} b(e)$ be a weight of *k-transitive path* G where $b(e)$ is a weight of arc e. Further, *k-transitive path* consisting of all vertices of graph G is *k-transitive Hamiltonian path* of G. Note that similar *n-transitive graphs* have been introduced in [507].

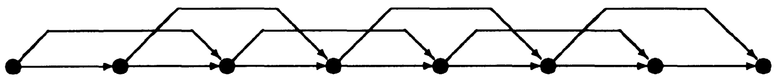

Fig. 7.32. Example of 3-*transitive path*

Fig. 7.32 corresponds to *flow shop problem* with $n = 8$, $\nu = 3$. Thus we consider each schedule as a $(\nu - 1)$-*transitive Hamiltonian path* and try to minimize (or maximize) its weight. Here arcs weights are equal to corresponding pair proximity between jobs.

Hence, the generalized solving scheme consists of the following two phases:

Phase 1. To construct and compute a proximity (i.e., to past together maximal elements into the same layers) for each pair of jobs $J_{i_1}, J_{i_1} \in J$ and for the following differences of positions:

$$-(\nu+1), ..., -1, 1, ..., \nu-1.$$

As a result, we get a complete oriented multigraph $G' = (J, \Rightarrow)$ in that proximity corresponding to arc weights, and for each vertex $\forall J_i \in J$ there exist $\nu - 1$ arcs of two kinds (for two directions).

Phase 2. To find *ν-transitive Hamiltonian path* with minimal (or maximal) total weight in G'. Evidently, this phase can be based on morphological clique problem.

Note that we can consider simple analogue problems as follows:
1. To find the shortest *ν-transitive path* in a digraph.
2. To find the shortest *ν-transitive Hamiltonian path* in a digraph.
Let us define a linear ordered (nonincreasing) set of $\{\tau(i,j)\}$:

$$\mu_1, ..., \mu_{n+\nu-1}, \mu_{n+\nu}, ..., \mu_{n\times\nu}.$$

First $n+\nu-1$ elements of the sequence are called *maximal elements*. Moreover, we consider binary $\tau'(i,j)$ that equals 1, if it corresponds to an maximal element, and 0 otherwise. As a result, we obtain a binary matrix.

Thus we can define an asymmetric jobs proximity (phase 1):

$$\rho(s[i_1], s[i_2]) = \begin{cases} 0, & \text{if } \delta \geq \nu, \\ \sum_{j=\delta+1}^{\nu} \tau'(i_1,j)\tau'(i_2,j-\delta), & \text{if } 1 \leq \delta \leq \nu-1, \\ \sum_{j=1}^{\nu-\delta} \tau'(i_1,j)\tau'(i_2,j+\delta), & \text{if } -(\nu-1) \leq \delta \leq -1, \\ 0, & \text{if } \delta \leq -\nu. \end{cases}$$

where $s[i_1]$ is a job at position i_1 and $s[i_2]$ is a job at position i_2, $(i_1 = 1, ..., n,\ i_2 = 1, ..., n,\ i_1 \neq i_2)$, $(\delta = i_2 - i_1)$. ρ corresponds to a level of complimentability of $s[i_1]$ and $s[i_2]$.

Fig. 7.34a presents the binary matrix that corresponds to the schedule example of Fig. 7.30. Thus, $\rho(s[i], s[j])$ equals the number of layers, when 1 of job $s[i_1]$ and 1 of $s[i_2]$ are at the same layer l (a diagonal from ***left-bottom*** to ***right-up***).

Let us consider an implementation of phase 2. Taking into account Lemma 7.5, we will minimize

$$T(S) \le \Xi(S) \le \sum_{\kappa=1}^{n+\nu-1} \mu_\kappa .$$

Here $\Xi^*(S) = \sum_{\kappa=1}^{n+\nu-1} \mu_\kappa$ is the upper bound.
Note h ***inclusions*** of maximal elements provide the following:

$$\Xi^*(S) = \sum_{\kappa=1}^{n+\nu-1-h} \mu_\kappa + \sum_{\kappa=n+\nu}^{n+\nu+h} \mu_\kappa .$$

Thus we get

$$\Delta(S,S') = T(S) - T(S') \ge \sum_{\kappa=1}^{h} (\mu_{n+\nu-1-h+\kappa} - \mu_{n+\nu-1+\kappa}).$$

On the other hand, we define

$$h(S') = \sum_{\forall i_1 < i_2} \rho(s[i_1], s[i_2]).$$

Evidently, that $h(S)$ is the weight of ***ν-transitive Hamiltonian path*** S, and we search for the ***maximal Hamiltonian path***.

Reduction of the above-mentioned problem to morphological clique is based on the following:

(1) n series positions are examined;
(2) all jobs are considered as DA's for each position;
(3) $\rho(s[i_1], s[i_2])$ corresponds to job compatibility; and
(4) all jobs are compatible at the higher level for $\delta > \nu$.

Note that proposed heuristic satisfies basic requirements as follows [28]: robustness, interactive computing, flexibility, simplicity and analyzability.

Now let us consider an example for the proposed heuristic on the basis of **Problem B** (example from Fig. 7.30). Fig. 7.33 depicts a schedule morphology (the 2nd index corresponds to the position number). Here all jobs have the same priority.

$S' = J_{11} * J_{22} * J_{33} * J_{44}$
$S'' = J_{41} * J_{12} * J_{23} * J_{34}$

Position 1	Position 2	Position 3	Position 4
J_{11}	J_{12}	J_{13}	J_{14}
J_{21}	J_{22}	J_{23}	J_{24}
J_{31}	J_{32}	J_{33}	J_{34}
J_{41}	J_{42}	J_{43}	J_{44}

Fig. 7.33. Schedule morphology for example from Fig. 7.30

Here $\max\{\tau(i,j)\} = \tau(1,3) = 15$ and $\min\{\tau(i,j)\} = \tau(4,3) = 5$. The ordered list of $\{\tau(i,j)\}$ is:

$$15,\ 12,\ 12,\ 10,\ 10,\ 9,\ 8,\ 8,\ 7,\ 6,\ 6,\ 5.$$

In this case, $l = 2, ..., 7$ (i.e., six layers). Thus the upper bound for the objective function (six maximal elements) is: 68.

Fig. 7.34a depicts a schedule S' that corresponds to Fig. 7.30 with the upper bound $\Xi^*(S') = 62$. Fig. 7.34b depicts another schedule S'' with $\Xi^*(S'') = 56$.

J_1	J_2	J_3	J_4
1	0	1	0
1	1	0	1
1	0	0	0

(a)

J_4	J_1	J_2	J_3
0	1	0	1
1	1	1	0
0	1	0	0

(b)

Fig. 7.34. Illustration for pasting together of maximal elements

Table 7.24 presents compatibility of jobs (Δ corresponds to incompatible DA's).

Clearly, it is possible to apply other techniques to compute job proximity and to use other approaches, including more exact ones, to evaluate upper bounds.

Table 7.24. Proximity of jobs

	J_{12}	J_{22}	J_{32}	J_{42}	J_{13}	J_{23}	J_{33}	J_{43}	J_{14}	J_{24}	J_{34}	J_{44}
J_{11}	Δ	1	1	1	Δ	0	1	0	Δ	0	0	0
J_{21}	1	Δ	1	0	0	Δ	0	0	0	Δ	0	0
J_{31}	0	0	Δ	0	0	0	Δ	0	0	0	Δ	0
J_{41}	1	0	1	Δ	0	0	0	Δ	0	0	0	Δ
J_{12}					Δ	1	1	1	Δ	0	1	0
J_{22}					1	Δ	1	0	0	Δ	0	0
J_{32}					0	0	Δ	0	0	0	Δ	0
J_{42}					1	0	1	Δ	0	0	0	Δ
J_{13}									Δ	1	1	1
J_{23}									1	Δ	0	0
J_{33}									0	0	Δ	0
J_{43}									1	0	1	Δ

7.6 GRAPH COLORING PROBLEM

The graph coloring problem arises in many applications. Traditional versions of graph coloring problems have been studied in ([160], [173], [177], [226], [229], [423], etc.). Our composition problem may be considered as a similar problem.

Let us examine a graph $G = (A, E)$ where A is a set of vertices and E is a set of edges. Let a set of colors correspond to a set of DA's, and for each $a \in A$ we can apply priority r of DA's as a profit of the assignment of the design alternative for vertex a. In addition, we consider an ordinal estimate of quality for a neighborhood (Ins) of each two colors (DA's) for the case. If they are assigned for neighbor vertices $a^{'}, a^{''} \in A$ (i.e., if there exists an edge $(a^{'}, a^{''})$). Thus we obtain our basic morphological clique problem.

An illustrative example is depicted in Fig. 7.35. Here we consider the following colors for each node: (a) black (X_1); (b) blue (X_2); (c) cyan (X_3); (d) green (X_4); (e) yellow (X_5); and (f) white (X_6).

Our compatibility of colors for neighbor vertices is presented in Table 7.25. Note that we can specify different compatibility for different vertices pairs too. Priorities of DA's (colors) are shown in Fig. 7.35 (in brackets).

Finally, composite decisions are the following ($N = (2; 4, 0)$):

(a) $S_1 = A_1 * B_4 * C_5 * D_6$;

(b) $S_2 = A_2 * B_4 * C_5 * D_6$; and

(c) $S_3 = A_1 * B_3 * C_5 * D_6$.

Note that the morphological approach (including graph coloring problem) has been applied in the design of user interfaces in [297] (chapter 11).

Table 7.25. Compatibility for colors

	X_1	X_2	X_3	X_4	X_5	X_6
X_1	$\triangle$	1	2	3	4	3
X_2		$\triangle$	1	2	3	4
X_3			$\triangle$	1	2	4
X_4				$\triangle$	3	2
X_5					$\triangle$	2
X_6						$\triangle$

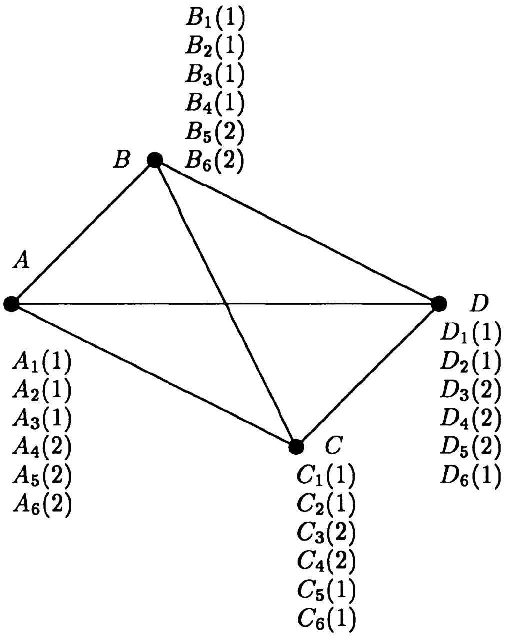

Fig. 7.35. Example of graph coloring

7.7 SUMMARY

This chapter has presented applications of morphological metaheuristics to several well-known combinatorial optimization problems. The material is a basis for **computation part** of our investigation. Evidently, these efforts have to be continued with the use of many computing experiments and real examples. It may be an excellent fundamental for student exercises and projects.

8 INFORMATION SUPPORT AND DESIGN

8.1 ISSUES OF INFORMATION SUPPORT

Main functional operations for design, utilization, and maintenance of design information systems are the following ([113], [294], [408]):

(1) acquisition of new data and knowledge;
(2) structuring and modeling;
(3) representation;
(4) learning;
(5) access, control;
(6) analysis, evaluation, and correction; and
(7) maintenance.

The issues of design data and knowledge representation are crucial ([11], [79], [93], [120], [130], [131], [217], [218], [238], [337], [387], [408], [459], etc.). HMMD uses hierarchical tree-like structure of designed system as a basic hierarchy. Representation of the hierarchy is easy. Many researchers apply similar approaches ([131], [238], [337], [247], etc.).

DB, KB, hypermedia systems and their combinations may be considered as information support tools. Some basic elements of the tools are presented in Table 8.1.

The following information systems may be pointed out as examples:

(a) NASA STI project [48];

(b) Information Design Model for engineering design [131];

(c) Designer's Electronic Guidebook for mechanical engineering in Cambridge Engineering Center [132];

(d) distributed hypermedia system (KMS) for managing knowledge in organizations [11];

(e) hierarchical hypertext system (HHS) involving components of different kinds and their criterial descriptions for various problem domains ([292], [294]);

(f) Management System "TOSCANA" on the basis of conceptual structures ([258], etc.); and

(g) Interspace (Engineering Concept Spaces): Digital Library Infrastructure for an University Engineering Community [79].

The development of complete information systems for supporting all phases of hierarchical design processes is the significant future direction. Hayes-Roth has investigated benchmarks which may be fruitful for the engineering design systems [203].

8.2 SOME PROBLEMS OF DATA RETRIEVAL

Contemporary information environments consist of many databases. As a result, information retrieval has a multi-stage and distributed character and has to include stages for selection of required databases and planning of searching processes. For example, similar approaches are implemented in *Index and Search Machines* for World Wide Web ([546], etc.). It is reasonable to consider the following main stages of data retrieval processes ([45], [27], [59], [77], [78], [87], [92], [143], [204], [292] [299], [339], [397], [419], [443], [500], [546], etc.):

Stage 1. Analysis of user's information needs, and forming a structured information query.

Stage 2. Analysis of information query and planning of data retrieval processes.

Stage 3. Searching for data.

Stage 4. Analysis and ranking selected data, evaluation of selected information.

Stage 5. Synthesis (integration) of resultant information clusters.

Table 8.1. Information components and support tools

Information components	Basic elements of information support
1.Hierarchical system model	Standard frames
2.Design module	
2.1 Requirements	
2.1.1 Criteria	Standard criteria
2.1.3 Compatibility factors	Standard factors
2.1.2 Restrictions	Standard restrictions
2.2. DA's	
2.2.1 Set of DA's	Standard design alternatives
2.2.2 Estimates	
2.2.3 Priorities	
2.3. Interconnection Ins	Standard interconnection
2.3.1 Estimates on factors	
2.3.2 Resultant estimates	
3.Composite decisions	Standard constraints
4.Improvement (combination, characteristics, schedule)	Basic examples, strategies of improvement
5.Systems versions (examples, requirements, tendencies)	Basic examples of systems versions, tendencies for various problem domains

Thus we consider the following significant phases of combinatorial data processing:

Preliminary phase 1 (stages 1, and 2 above). An analysis of information needs and planning of retrieval processes.

Final phase 2 (stages 5 and 6 above). An analysis of selected information and combinatorial synthesis of a resultant information cluster that corresponds to user's needs.

In the section, we present a short description of the above-mentioned phases.

8.2.1 Planning of Data Retrieval Process

Analysis of information needs and planning data retrieval processes are based on the following steps ([315], [546], etc.):

(1) analysis of information needs and design of structured information requirements;

(2) formation of a preliminary list of databases and their attributes (resource database);

(3) analysis of databases profiles;

(4) ranking of databases and selection of the more useful databases;

(5) design of a composite/distributed data retrieval process; and

(6) scheduling of data retrieval processes.

Clearly, that steps 4, 5, and 6 may be based on combinatorial models as follows (respectively):

(a) knapsack like problems and multicriteria ranking;

(b) knapsack like problem and morphological clique; and

(c) knapsack like problem, traditional scheduling problems, and morphological clique.

Approaches to intelligent information retrieval have been discussed in [97]. Some combinatorial problems of scheduling for data retrieval processes have been examined in [315].

8.2.2 Information Synthesis

Let us examine problems for designing a resultant information cluster (complex) in hypertext or hypermedia systems (hyper-information systems). We assume that a hyper-information system consists of discrete information units (items), their attributes (e.g., keywords), and item links of different kinds.

Many investigators have used similar formal descriptions as information networks ([77], [78],[79], [87], [292], [503], [531], etc.). Usually the following kinds of information processing are considered:

(a) search (e.g., automatic searching for, browsing);

(b) selection of data on the basis of comparison of data or data and specification (results: an item; an item set; an item set with attributes; and an item set with attributes and links);

(c) design or approximation of an information structure (e.g., a set of clusters and an information hierarchy); and

(d) design (synthesis) of an information complex (results: an item set; an item set with structure).

Here we consider combinatorial problems for designing an information complex ([299] and [312]). In recent years, similar problems of information integration have been investigated ([159], [220], [374], [397], etc.). Aggregate hypertext objects that are similar to our information complexes have been studied in [526]. In this section, the following steps are examined as crucial:

(1) specification and/or approximation of an initial structure for a resultant information cluster;

(2) selection of items as alternative units (DA's) including comparison, multicriteria selection; and

(3) synthesis or composition of an item set (i.e., an information cluster) from interconnected items (DA's) on the basis of combinatorial problems (e.g., clique problem, an approximation of a clique; designing an item hierarchy).

Finally, we consider two basic optimization problems as follows: the selection and composition (when a structure of the target system is specified). Also, we take into account an ordinal item compatibility or relationship (i.e., interconnection Ins) and a correspondence of items to components of a search specification (a structure of the resultant information cluster). As a result, we get versions of a morphological clique problem. Our information design process is illustrated in Fig. 8.1.

Note that close data retrieval schemes (search, selection, ranking, etc.) are used in many information systems, for example,

(a) large scale retrieval systems based on concept spaces ([77], [78], and [79]); and

(b) integration of heterogeneous information sources [159];

(c) cooperative database systems [374].

Usually the following basic kinds of information elements for hypertext are assumed ([87], [292], [503], [531], etc.).

(1) a node (a discrete cognitive unit or an item with attributes);

(2) a link between two nodes;

(3) a node set (an item group);

(4) a link set;

(5) a node and a corresponding link set; and

(6) a node set, and a corresponding link set.

We can use the following main structural types of information elements (an item set with a structure and an information complex):

(a) a pair of items;

(b) an item chain (path);

(c) a layered structure (a linear ordered set of item groups);

(d) a hierarchy of items or item groups;

(e) a cluster or a group of interconnected items; and

(f) combined cases.

Grouping operations of information items may be based on various relations as follows; equivalence, proximity, similarity, dissimilarity, implication, prerequisite, complementability, and independence. Relation graphs for relationships among agents on the basis of cognitive maps (communication structures and causal assertions) are studied in ([73], [552], etc.).

Some standard semantic frames and relations between data units have been considered in [399]. An attempt to describe the typical information complexes is contained in [299]. Here we assume that an information complex is a cluster of data units which consists of the following:

(i) a structure (e.g., complete graph) in which each vertex corresponds to a component of the designed information complex;

(ii) data units are divided into a set of item groups and each group corresponds to a component of the complex; and

(iii) connections of data items from different groups corresponding to edges of the structure.

Table 8.2 contains basic operations for processing of data, structures, and information complex.

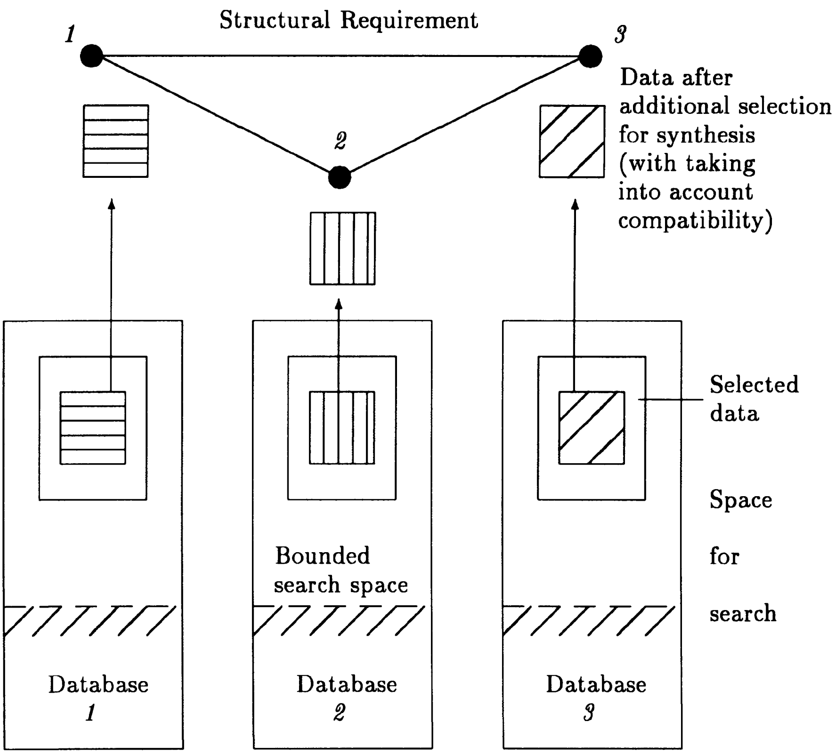

Fig. 8.1. Scheme of combinatorial information design

Table 8.2. Operations and information elements

Stage	Basic information elements		
	Data unit	Information structure	Information complex
1.Analysis/ evaluation	Definition of features, e.g. correspondence to specification and kind of connection	Standard graph analysis/ evaluation operations	Definition of features (e.g., element interconnection, etc.)
2.Comparison	Semantic comparison by attributes	Structural comparison	Semantic and structural comparison
3.Approximation	Possible change	Graph approximation	Semantic and structural approximation
4.Aggregation/ construction of a consensus	Unification, choice or design of represen- tatives	Structural unification/ design of a consensus, e.g., median	Semantic and structural unification, design of a consensus
5.Design	Specification, search and selection	Graph construction	Selection and integration of initial data units and their links, approximation and aggregation

Thus a process of designing an information complex consists of the following:

Step 1. Specifying a structure of information complex, and requirements to information items for each component of the information complex.

Step 2. Searching for and selection of data items for each component (accumulation of initial information).

Step 3. Analysis and evaluation of selected information items, e.g., indexing ([79], [444], etc.).

Step 4. Construction of the required information complex or its approximation: (a) selection of information items while taking into account their compatibility; (b) design of the required complex; and (c) design of an approximate complex if the required complex does not exist or its design is impossible.

Some basic problems of designing the complexes are presented in Table 8.3.

Table 8.3. Problems of information design

INPUT (search/design specification)	OUTPUT	KIND OF PROBLEM
1.Specification for an item	Item of item set	Search/ selection
2.Specification for an item & restriction	Set of connected items	Knapsack problem with compatibility
3.Structure of resultant complex and specifications for each its component	Group of connected item sets	Morphological clique problem

Let us examine a numerical example. Here we assume that a structural specification consists of 5 components. After searching for, selecting, and evaluating a correspondence to specification components (a computation of priorities are shown in brackets), the following data items (DA's) are obtained:

1. Hypermedia systems H: $H_1(2)$, $H_2(1)$, $H_3(3)$, and $H_4(1)$.
2. Graphs G: $G_1(2)$, $G_2(3)$, $G_3(1)$, and $G_4(1)$.
3. Combinatorial optimization C: $C_1(1)$, $C_2(3)$, $C_3(1)$, and $C_4(2)$.
4. Decision making D: $D_1(1)$, $D_2(3)$, $D_3(2)$, and $D_4(1)$.
5. Fuzzy sets F: $F_1(2)$, $F_2(1)$, $F_3(2)$, $F_4(3)$, and $F_5(1)$.

On the basis of the analysis and evaluation of data items we may define attributes of DA's above. This information instance corresponds to the numerical example in section 3.5 (compatibility and resultant composite decision for 5 cases of approximation approach).

Our results are preliminary. However, we think that our approach based on combinatorial models may be used for other problems in the fields of data processing. Note that in recent years, many close approaches based on special information spaces or complexes, are studied and realized in digital libraries and enterprise information modeling, for example: (a) engineering concept spaces for assisting of user-specific concept based information retrieval ([78], [79], etc.); (b) conceptual structures in information management ([258], etc.); (c) common aggregate context for assisting of semantic integration via a common ontology [220]; etc.

8.3 PLANNING OF INFORMATION CENTER

8.3.1 Strategic Planning for Information Systems

Problems of strategic planning and design for information systems (IS) have been investigated for about four decades. For instance, Hackathorn and Karimi compared information engineering methods [191]. Ledered and Sethi considered three methodologies for strategic IS planning as follows: business systems planning (Top-Down planning with Bottom-Up implementation); strategic system planning on the basis of main functional areas; and information engineering on the basis of models for enterprise, data, and processing [282]. Premkumar and King examined an evaluation of strategic IS planning [403]. Raghunathan and King studied three phases of IS planning as follows: (a) strategic planning; (b) system planning; (c) plan implementation [406]. In recent years, many IS (e.g., university libraries, firm databases) are transformed to large information conglomerations (LIC) which are based on the following: (1) contemporary communication networks; (2) distribution/sharing of users, information, and processing; (3) intelligent support all stages of data processing, and services; and (4) use and development of new kinds of IS (e.g., hypermedia systems).

In fact, some LIC's play a role of regional or problem domain information centers. An example of a complex distributed information system is depicted in [248]. Note that LIC's start to manufacture new information products not only for their users but for national and/or international information market. Rainet and Carr describe a big catalog of services offered by information centers for end users [407].

Usually, the following main problems of IS development are examined: (a) data design (selection, structuring, composition, sharing, etc.); (b) service planning (planning of query sequencing, etc.); (c) network design and management (topology design, facility location, routing, etc.); (d) user modeling, diagnosis, and training; etc.

In the main, traditional approaches to IS development are based on the generation of design alternatives, and their selection [246]. Note that usually only one or two kinds of above-mentioned IS components are examined. For instance, the following approaches have been used for the selection of IS components: (a) criteria like cost/benefit ([5], and [466]); and (b) empirical models [183]; (c) combination of knapsack-like models and multicriteria ranking ([316], [351]); (d) analytic hierarchy process [142].

A flexible approach to information systems development has been proposed in [6]. In this section, we investigate a problem of information center development (design and/or redesign) on the basis of HMMD. Thus it is necessary to take into account various components as follows: data; software; hardware; communication support; interfaces; human relation; etc. This type of problem is a complicated one, and it requires both selection (i.e., analysis) and composition (i.e., synthesis) approaches.

8.3.2 Structure of Information Center and Requirements

Here we examine a structure for information centers that is presented in Fig. 8.2. Table 8.4 contains compatibility factors for all connected components. We use the following three techniques for assessment of Ins: (a) direct expert judgement; (b) constant compatibility (5) for independent components pairs; (c) evaluation of factors, and (d) multi-factor ranking to obtain resultant ordinal estimates of Ins. In the last case, the factors are indicated by weights (in brackets). Requirements to IS are a key part of IS investigations. In the main, the requirements relate with the following: goals and basic technologies in the organization; budget restriction; experience of users and personnel; quality of services; market needs; etc. For instance, Magal et al. investigated 26 critical success factors for information centers managers, and extract 5 composite critical success factors as follows [336]: (1) commitment to information centers; (2) quality of support services for information centers; (3) facilitation of end-user computing; (4) role clarity; and (5) coordination of end-user computing.

The study of IS by Raghunathan and Raghunathan is based on the following 8 planning objectives [405]: (1) predicting of future trends; (2) improving short-term IS performance; (3) improving long-term IS performance; (4) improving decision making; (5) avoiding problem areas; (6) increasing user satisfaction; (7) improving system integration; and (8) improving resource allocation. Keen has described a set of requirements to DSS from a user viewpoint [241]: (a) increasing the initial set of alternatives; (b) better understanding of an analyzed business; (c) adaptation to expected situations; (d) ability to carry out *ad hoc* analyses; (e) new insights and learning; (f) improved communication and teamwork; (g) better control; (h) saving of resources (e.g., money and time); (i) receiving better decisions; etc. Hamilton and Chervany examined several viewpoints to IS evaluation as follows: user personnel, MIS development personnel, management personnel, and internal audit personnel [195]. Farn and Lo executed the detail analysis of selection requirements for IS including independence of files, data security, high level language interface, installation experience, etc. [142]. Reimann and Waren proposed user-oriented criteria for the selection of DSS software [415]. Lucas proposed a special framework to DSS evaluation [333]. An analysis of effectiveness measures for DSS has been executed in [513].

In our consideration, outlined criteria sets correspond to examined parts of IS (software, hardware, data, communication, etc.).

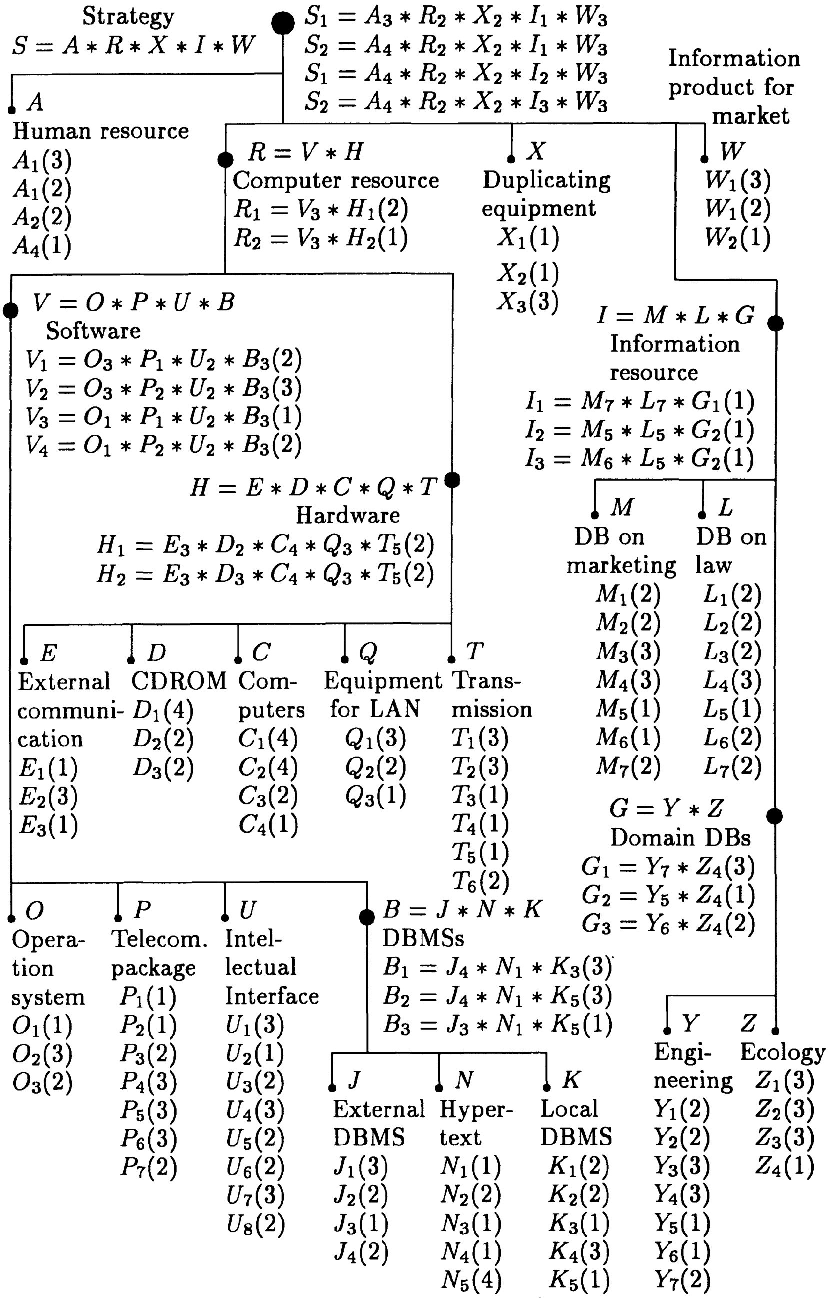

Fig. 8.2. Structure of strategy for information center

Table 8.4. Compatibility factors for connected components

Components	Compatibility factors
1. Operation System (O) & Telecommunication Package (P)	Multi-tasks
2. Operation System (O)&Interface (U)	Software compatibility
3. Operation System (O)& DBMS (B)	Software compatibility
4. External (J)& local DBMSs (K)	Software and data format interface
5. External DBMSs (J)& Hypertext system (N)	Software and data format interface
6. Internal DBMSs (K) & Hypertext system (N)	Software and data format interface
7. Telecommunication Package (P) & Interface (U)	Constant (5)
8. Telecommunication Package (P) & DBMS (B)	Constant (5)
9. Interface (U) & DBMS (B)	Constant (5)
10.External Communication (E) & CD ROM (D)	Constant (5)
11.External Communication (E) & Computers (C)	Required resource for interface
12.External Communication (E) & Equipment for LAN (Q)	Constant (5)
13.External Communication (E) & Transmission (T)	Constant (5)
14.CD ROM (D) & Computers (C)	Constant (5)
15.CD ROM(D)& Equipment for LAN(Q)	Constant (5)
16.CD ROM (D) & Transmission (T)	Constant (5)
17.Computers (C)& Equipment for LAN (Q)	Required resource for interface
18.Computers (C) & Transmission (T)	Required resource for interface

Continuation of Table 8.4

Components	Compatibility factors
19.Equipment for LAN (Q) & Transmission (T)	Software and hardware interface
20.Software (V) & Hardware (H)	Constant (5)
21.Engineering (Y) & ecology DBs (Z)	Computer resource (-2), semantic whole (5), speed (4)
22.Marketing DB (M) & Law DB (L)	Computer resource (-2), semantic whole (5), speed (4)
23.Marketing DB (M) & Problem DB (G)	Computer resource (-2), semantic whole (5), speed (4)
24.Law DB (M) & Problem DB (G)	Computer resource (-2), semantic whole (5), speed (4)
25.Computer resource (R) & duplicating equipment (X)	Constant (5)
26.Computer resource (R) & Information (I)	Constant (5)
27.Duplicating equipment (X) & Information (I)	Constant (5)
28.Human resource (A) & Computer resource (R)	Experience of staff
29.Human resource (A) & Duplicating equipment (X)	Experience of staff
30.Human resource (A) & Information (I)	Experience of staff
31.Product for market W & Human resource (A)	Experience of staff
32.Product for market W & Computer resource (R)	Technical support
33.Product for market W & duplicating equipment (X)	Technical support
34.Product for market W & Information (I)	Semantic whole

8.3.3 Composing of Integrated DBMSs

Criteria and their weights, aggregate criteria for DBMSs, estimates, compatibility, and composite DA's are presented in Tables 8.5, 8.6, 8.7, 8.7, and 8.8, accordingly.

Table 8.5. Criteria

Criteria	Weights		
	K	N	J
Cost	-3	-3	-3
Experience	5	5	
Developed applications	5	1	
Speed		1	
Complexity of installation			2
Universal access			5
Multi-base mode			6

Table 8.6. Aggregate criteria for B

Criteria	Weight	Specification
F_{b1} Cost	-1	$F_{j1} + F_{k1} + F_{n1}$
F_{b2} Complexity of installation	3	$\min(F_{k2}, F_{j5})$
F_{b3} Multi-base access mode	4	F_{j7}
F_{b4} Developed applications	3	$\max(F_{k3}, F_{n3})$
F_{b5} Compatibility	4	w

Table 8.7. Estimates

DA's	Criteria						
	1	2	3	4	5	6	7
K_1	5	2	2	1			
K_2	5	3	2	2			
K_3	1	2	2	3			
K_4	3	0	1	3			
K_5	5	2	3	3			
N_1	3	3	2				
N_2	3	1	1				
N_3	7	2	3				
N_4	1	0	3				
N_5	5	0	0				
J_1	0				0	0	0
J_2	2				5	1	0
J_3	4				3	3	1
J_4	6				1	5	0

Table 8.8. Compatibility

	K_1	K_2	K_3	K_4	K_5	N_1	N_2	N_3	N_4	N_5
J_1	2	2	2	2	2	3	2	1	1	2
J_2	3	3	3	2	3	3	2	1	1	2
J_3	2	2	2	2	3	3	2	1	1	2
J_4	4	4	4	2	4	4	2	1	1	2
K_1						5	2	1	1	2
K_2						5	2	1	1	2
K_3						5	2	1	1	2
K_4						4	2	1	1	2
K_5						5	2	1	1	2

Table 8.9. Composite DA's for B

DA's	Criteria				N
	1	2	3	4	
$B_1 = J_4 * K_3 * N_1$	10	1	0	2	$(4;2,1,0,0)$
$B_2 = J_4 * K_5 * N_1$	14	1	0	3	$(4;2,1,0,0)$
$B_3 = J_3 * K_5 * N_1$	12	2	1	3	$(3;3,0,0,0)$

8.3.4 *Composing of Software*

Criteria and their weights, aggregate criteria for software, estimates, compatibility, and composite DA's are presented in Tables 8.10, 8.11, 8.12, 8.13, and 8.14, accordingly.

Table 8.10. Criteria

Criteria		Weights		
		U	P	O
F_{u1}	Cost	-1	-1	-3
F_{u2}	Experience	1		5
F_{u3}	Knowledge base	1		
F_{u4}	Multi-base access	1		
F_{u5}	User's model	1		
F_{p2}	Volume of required memory		-3	
F_{p3}	Scenario of dialogue		4	
F_{p4}	Multi-equipment		4	
F_{p5}	Consulting service		6	
F_{o3}	Multi-task mode			5

Table 8.11. Aggregate criteria for V

Criteria		Weight	Specification
F_{v1}	Cost	-1	$F_{o1} + F_{p1} + F_{u1} + F_{b1}$
F_{v2}	Experience, service, installation	3	$\min(F_{o2}, F_{p8}, F_{u2}, F_{b2})$
F_{v3}	Possible extension	3	expert judgement

Table 8.12. Estimates

DA's	Criteria									
	1	2	3	4	5	6	7	8	9	10
U_1	4	0	3	0	0					
U_2	4	2	0	4	0					
U_3	10	0	2	5	2					
U_4	2	0	1	0	0					
U_5	6	1	2	3	0					
U_6	8	1	4	3	0					
U_7	6	0	2	3	0					
U_8	8	1	3	4	0					
P_1	10					10	1	0	1	
P_2	10					10	1	0	1	
P_3	5					5	0	0	1	
P_4	10					15	1	0	0	
P_5	5					15	1	0	0	
P_6	20					10	0	0	1	
P_7	20					20	1	0	1	
O_1	1	1								0
O_2	6	0								1
O_3	4	0								1

Table 8.13. Compatibility

	$P_1 \ldots P_7$	O_1	O_2	O_3	B_1	B_2	B_3
$U_1..U_8$	5	3	4	3	5	5	5
$P_1..P_5$		5	1	5	5	5	5
P_6		2	5	3	5	5	5
P_7		2	5	3	5	5	5
O_1					5	2	2
O_2					2	4	4
O_3					3	4	4

Table 8.14. Composite DA's of V

Composite DA's	Cr			N
	1	2	3	
$V_1 = O_3 * P_1 * U_2 * B_3$	32	0	4	$(5;3,1,0,0)$
$V_2 = O_3 * P_1 * U_2 * B_3$	32	0	3	$(5;3,1,0,0)$
$V_3 = O_1 * P_1 * U_2 * B_3$	27	1	3	$(2;4,0,0,0)$
$V_4 = O_1 * P_2 * U_2 * B_3$	27	1	2	$(2;4,0,0,0)$

8.3.5 Composing of Hardware

Criteria and their weights, aggregate criteria for hardware, estimates, compatibility, and composite DA's are presented in Tables 8.15, 8.16, 8.17, 8.18, and 8.19, accordingly.

Table 8.15. Criteria

Criteria		Weights				
		T	Q	C	D	E
F_{t1}	Cost	−3				
F_{t2}	Multi-network mode	5				
F_{q1}	Network dimension		−1			
F_{q2}	Speed		1			
F_{q3}	Interface with network		3			
F_{q4}	Quantity of PCs		4			
F_{c1}	Cost			1	1	1
F_{c2}	Automation level			4		
F_{c3}	Internal base support			7		
F_{c4}	Multi-user mode			6		
F_{c5}	Work in external network			5		
F_{d2}	Read mode				5	
F_{d3}	Write mode				2	
F_{e2}	Start interval					−3
F_{e3}	Quality of communication					2

Table 8.16. Estimates

DA's	Criteria														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
T_1 None	0	0													
T_2 Modem	3	0													
T_3 Modem&fax-plate	4	1													
T_4 Fax-modem	4	1													
T_5 Modem&network plate	5	1													
T_6 Modem&equipment for X.25	7	1													
Q_1 Adapter for Ethernet			1	1	1	2									
Q_2 Adapter for Arcnet			3	2	3	3									
Q_3 Adapter for Pcnet			3	4	3	5									
C_1 2 IBM PC							2	1	1	2	1				
C_2 5 IBM PC							3	2	3	5	1				
C_3 7 PC& LAN							8	5	10	10	5				
C_4 10 PC&Mini computer							13	6	15	15	5				
D_1 None							0					0	0		
D_2 Read mode							4					1	0		
D_3 Read&write modes							30					1	1		
E_1 None							0							0	0
E_2 Switch channel							3							5	1
E_3 Network channel							1							1	5

Table 8.17. Compatibility

	$Q_1 \ldots Q_3$	C_1	C_2	C_3	C_4	$D_1 \ldots D_3$	E_1	E_2	E_3
$T_1..T_4$	2	2	1	5	5	5	5	5	5
$T_5..T_6$	5	2	1	5	5	5	5	5	5
$Q_1..Q_3$		2	1	5	5	5	5	5	5
C_1						5	1	4	4
C_2						5	1	3	3
$C_3..C_4$						5	1	5	5
$D_1..D_3$							5	5	5

Table 8.18. Aggregate criteria for H

Criteria	Weight	Specification
F_{h1} Cost	-1	$F_{e7} + F_{d7} + F_{c7} + F_{t7}$
F_{h2} Experience, service, installation	3	expert judgement
F_{h3} Possible extension	4	expert judgement
F_{h4} Possibility of new product or service development	4	F_{d13}

Table 8.19. Composite DA's for H

DA's	Criteria				N
	1	2	3	4	
$H_1 = E_3 * D_2 * C_4 * Q_3 * T_5$	26	2	3	0	$(5;4,1,0,0)$
$H_2 = E_3 * D_3 * C_4 * Q_3 * T_5$	52	1	3	1	$(5;4,1,0,0)$

8.3.6 Composing of Computer Resource

Compatibility between DA's of V and H equal to 5, because all DA's are compatible. Aggregate criteria for computer resource (R) are presented in Table 8.20.

Table 8.20. Aggregate criteria for R

Criteria	Weight	Specification
F_{r1} Cost	-1	$F_{h1} + F_{v1}$
F_{r2} Possible extension	4	$\min(F_{h3}, F_{v3})$
F_{r3} Possibility of new product or service development	4	$(F_{h4}$

Table 8.21. Composite DA's for R

DA's	Criteria			N
	1	2	3	
$R_1 = V_3 * H_1$	53	3	1	$5; 1, 1, 0, 0$
$R_2 = V_3 * H_2$	79	3	1	$5; 1, 1, 0, 0$

8.3.7 Composing of Problem Databases

We examine problem databases for the two following domains: engineering (Y) and ecology (Z). Criteria and their weights, estimates, and aggregate criteria for G are presented in Tables 8.22, 8.23, and 8.24, accordingly. Table 8.25 contains compatibility between DA's of Y and Z. Each element of this table consists of preliminary estimates upon three compatibility factors (Table 8.4) and a resultant ordinal estimate (after symbol '/') that was computed by multicriteria ranking.

Table 8.22. Criteria

Criteria	Weight	
	Y	Z
Cost	−3	−3
Quality of service	5	5
Up-date support	−5	−5
Required computer resource	−3	−3
Completeness of data	6	

Table 8.23. Aggregate criteria

Criteria	Weight	Specification
F_{g1} Cost	−3	$F_{y1} + F_{z1}$
F_{g2} Quality of service	5	$F_{y2} + F_{z2}$
F_{g3} Up-date support	−5	$\min(F_{y3}, F_{z3})$
F_{g4} Required computer resource	−3	$F_{y4} + F_{z4}$

Table 8.24. Estimates

DA's	Criteria				
	1	2	3	4	5
Y_1 None	0	0	0	0	0
Y_2 Catalog	1	1	1	1	1
Y_3 Internal base	3	2	3	3	2
Y_4 Internal base & catalog	4	3	4	4	2
Y_5 External base	3	3	1	1	5
Y_6 External base & catalog	5	4	1	2	5
Y_7 Internal & external bases, catalog	7	7	4	4	5
Z_1 None	0	0	0	0	
Z_2 Internal base	3	2	3	3	
Z_3 External base	3	3	1	1	
Z_4 Internal & external bases	7	7	4	4	

Table 8.25. Compatibility

	Z_1	Z_2	Z_3	Z_4
Y_1	0, 0, 0/1	5, 0, 1/1	0, 0, 0/1	4, 0, 0/1
Y_2	1, 0, 0/1	5, 1, 1/2	1, 1, 1/2	5, 2, 1/2
Y_3	4, 0, 0/1	8, 4, 6/3	4, 4, 3/3	8, 5, 6/4
Y_4	5, 0, 0/1	9, 4, 6/3	5, 4, 3/3	9, 5, 6/4
Y_5	0, 0, 0/1	4, 5, 3/4	0, 5, 2/4	4, 6, 3/4
Y_6	1, 0, 0/1	5, 5, 3/4	1, 5, 2/4	5, 6, 3/4
Y_7	5, 0, 0/1	9, 6, 6/4	5, 6, 3/4	9, 7, 6/5

Table 8.26. Composite DA's for G

DA's	Criteria				N
	1	2	3	4	
$G_1 = Y_7 * Z_4$	14	14	4	8	$(5; 1, 1, 0, 0)$
$G_2 = Y_5 * Z_4$	10	10	1	5	$(4; 2, 0, 0, 0)$
$G_3 = Y_6 * Z_4$	12	11	2	6	$(4; 2, 0, 0, 0)$

8.3.8 Composing of Information Resource

In our example, we consider information resource as a composition of problem DB's and two specific DB's as follows: database on *marketing* and database on *law*. Criteria, aggregate criteria for I, estimates upon criteria are presented in Tables 8.27, 8.28, 8.29. Tables 8.30, 8.31 contain compatibility. Each element of these tables consists of preliminary estimates upon three factors (Table 8.4) and a resultant ordinal estimate (after symbol '/') that was computed by multicriteria ranking. Compatibility for G and L, and G and M are coincided. Table 8.32 presents composite DA's for I.

Table 8.27. Criteria

Criteria	Weight	
	M	L
Cost	-3	-3
Quality of service	5	5
Up-date support	-5	-5
Required computer resource	-3	-3
Access to data relations	5	7

Table 8.28. Aggregate criteria

Criteria	Weight	Specification
F_{i1} Cost	-3	$F_{l1} + F_{m1} + F_{g1}$
F_{i2} Quality of service	5	$F_{l2} + F_{m2} + F_{g2}$
F_{i3} Up-date support	-5	$\min(F_{l3}, F_{m3}, F_{g3})$
F_{i4} Required computer resource	-3	$F_{l4} + F_{m4} + F_{g4}$
F_{i5} Compatibility	5	w

Table 8.29. Estimates

DA's	Criteria				
	1	2	3	4	5
M_1 None	0	0	0	0	0
M_2 Catalog	1	1	1	1	0
M_3 Internal base	3	2	3	3	1
M_4 Internal base & catalog	4	3	4	4	1
M_5 External base	3	3	1	1	3
M_6 External base & catalog	5	4	1	2	3
M_7 Internal & external bases, catalog	7	7	4	4	3
L_1 None	0	0	0	0	0
L_2 Catalog	1	1	1	1	0
L_3 Internal base	3	2	3	3	1
L_4 Internal base & catalog	4	3	4	4	1
L_5 External base	3	3	1	1	3
L_6 External base & catalog	5	4	1	2	3
L_7 Internal & external bases, catalog	7	7	4	4	3

Table 8.30. Compatibility

	M_1	M_2	M_3	M_4	M_5	M_6	M_7
L_1	0,0,0/0	1,0,0/0	4,0,0/0	5,0,0/0	0,0,0/0	1,0,0/0	5,0,0/0
L_2	1,0,0/0	2,1,0/2	5,1,1/3	6,2,3/3	1,2,0/3	2,2,0/3	6,2,3/4
L_3	4,0,0/0	5,4,1/3	8,4,6/3	9,6,6/3	4,6,3/3	5,5,1/3	9,5,6/4
L_4	5,0,0/0	6,4,4/3	9,4,6/3	10,7,6/5	5,7,4/4	6,5,5/4	10,7,7/5
L_5	0,0,0/0	1,5,2/3	4,5,3/3	5,6,3/4	0,6,3/4	1,6,4/4	5,6,3/4
L_6	1,0,0/0	2,5,2/3	5,5,3/3	6,6,3/4	1,6,3/4	2,6,4/4	6,6,3/4
L_7	5,0,0/0	6,6,4/4	9,6,6/4	10,8,7/5	5,7,4/4	6,7,5/4	10,8,7/5

Table 8.31. Compatibility

	L_1/M_1	L_2/M_2	L_3/M_3	L_4/M_4	L_5/M_5	L_6/M_6	L_7/M_7
G_1	5,0,0/1	6,2,3/2	9,6,6/4	10,7,6,4	5,6,3/4	6,6,3/4	10,7,7/5
G_2	0,0,0/1	1,2,0/2	4,6,3/4	5,7,4/4	0,6,3/4	1,6,3/4	5,7,4/4
G_3	1,0,0/1	2,2,0/2	5,5,1/3	6,5,5/4	1,6,4/4	2,6,4/4	6,7,5/4

Table 8.32. Composite DA's for I

DA's	Criteria				N
	1	2	3	4	
$I_1 = M_7 * L_7 * G_1$	28	28	4	16	$(5;0,2,1,0)$
$I_2 = M_5 * L_5 * G_2$	16	16	1	7	$(4;0,3,0,0)$
$I_3 = M_6 * L_5 * G_2$	18	17	1	9	$(4;0,3,0,0)$

8.3.9 Composing of Resultant Strategies

Criteria and their weights for components of resultant strategy are presented in Table 8.33. Table 8.34 contains estimates of DA's. Compatibility of DA's, and resultant composite DA's are presented in Tables 8.35 and 8.36 accordingly. Fig. 8.3. depicts composite DA's at the space of excellence. Here we do not examine estimates and ranking of resultant composite DA's. An additional analysis of resultant decisions may involve expert judgement, multicriteria ranking, bottleneck revelation, and action improvement. Table 8.37 presents some bottlenecks and improvement actions.

Table 8.33. Criteria

Criteria	Weights		
	A	X	W
Cost	−3	−3	−3
Quality	4		
Time before start	−4		
Phychological conflict	−5		
Productivity		2	
Service		5	
Requirements to staff	3		4
Possibility of new market	3		3

Table 8.34. Estimates

DA's	Criteria							
	1	2	3	4	5	6	7	8
A_1 Previous staff	0	1	7	3			0	1
A_2 Training previous staff	2	3	4	2			1	2
A_3 Combination of previous training staff, and new staff	4	4	1	1			2	2
A_4 New staff	6	5	0	5			3	1
X_1 None	0				0	2		
X_2 Equipment 1	2				2	2		
X_3 Equipment 2	3				4	1		
W_1 None	0						0	0
W_2 Print materials (e.g., brochure)	2						3	1
W_3 Hypertext system	2						5	3

Table 8.35. Compatibility

	X_1	X_2	X_3	R_1	R_2	W_1	W_2	W_3	I_1	I_2	I_3
A_1	5	4	4	1	1	5	3	0	1	1	1
A_2	5	5	5	3	3	5	5	3	3	3	3
A_3	5	5	5	5	5	5	5	5	5	5	5
A_4	5	5	5	5	5	5	5	5	4	4	4
X_1				5	5	5	0	3	5	5	5
X_2				5	5	5	4	5	5	5	5
X_3				5	5	5	5	5	5	5	5
R_1						5	4	3	5	5	5
R_2						5	5	5	5	5	5
W_1									5	5	5
W_2									5	2	2
W_3									5	2	2

Table 8.36. Composite DA's for S

DA's	N
$S_1 = A_3 * W_3 * R_2 * X_2 * I_1$	$(5;4,1,0,0)$
$S_2 = A_4 * W_3 * R_2 * X_2 * I_1$	$(4;5,0,0,0)$
$S_3 = A_4 * W_3 * R_2 * X_2 * I_2$	$(4;5,0,0,0)$
$S_4 = A_4 * W_3 * R_2 * X_2 * I_3$	$(4;5,0,0,0)$

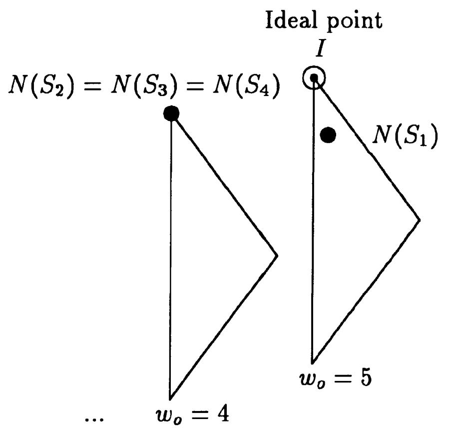

Fig. 8.3. Space of system excellence

Table 8.37. Bottlenecks and improvements

Composite DA's	Bottleneck		Action w/r
	DA	Ins	
$S_1 = A_3 * W_3 * R_2 * X_2 * I_1$	A_3		$2 \Longrightarrow 1$
$A_4 * W_3 * R_2 * X_1 * I_1$		(W_3, X_1)	$3 \Longrightarrow 4$
$A_4 * W_3 * R_2 * X_1 * I_2$		(W_3, X_1)	$3 \Longrightarrow 4$
$A_4 * W_3 * R_2 * X_1 * I_3$		(W_3, X_1)	$3 \Longrightarrow 4$

8.4 SUMMARY

In this chapter, we have described three issues as follows:
(i) information support of morphological design,
(ii) combinatorial planning and synthesis in information retrieval, and
(iii) design/planning and reengineering of information systems.

The importance of the second issue may be increased for multimedia systems and Internet applications. In this case, it will be reasonable to use multi-period series-parallel planning too. Our example for information systems is understandable for many readers and can be considered as a basic one for close applications.

9 SOME ORGANIZATIONAL ISSUES

9.1 DESIGN FLOW

Our hierarchical morphological approach (e.g., HMMD) implements an environment of distributed cooperative work. Organizational issues of cooperative work include the following ([128], [153], [251], [274], [480], [485], [490], etc.):

(a) organizational structuring on the basis of general framework;
(b) control and planning;
(c) coordination;
(d) conflict avoidance and resolution; and
(e) negotiation.

Structuring the design tasks in a complex project is the crucial basis of design processes ([98], [135], [490], etc.). HMMD implements ***cascade*** flexible design strategy (***reconvergent***) that is based on a hierarchical (tree-like) model of designed system and consists of two major stages which are sequenced in an iterative manner (Fig. 9.1):

(i) start stage (***divergent part***), and
(ii) work stage (***convergent part***).

This specifies the type of design flow. Many investigators have been examined similar design schemes ([98], [113], [332], [367], [387], [480], [528], etc.).

Let us note that the first stage is executed by specialists with system thinking, and at the second stage decision-oriented specialists have to be used. HMMD provides an iterative combination of their activities. Essentially, at the first stage of HMMD the system specialist develops a hierarchical model of the design system and directs design processes into domains and decisions including sets of local decisions.

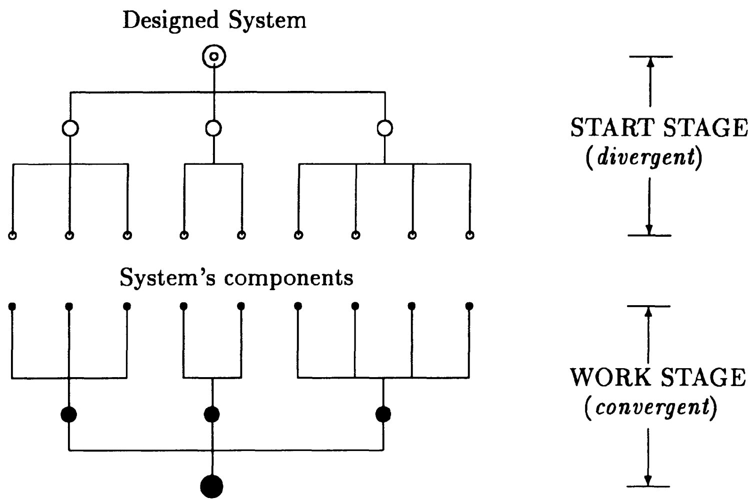

Fig. 9.1. Design phase of HMMD as *cascade* strategy

9.2 ROLES OF SPECIALISTS

The design team can have many tasks, including the following: problem definition and exploration; specification of criteria and constraints; and generation, evaluation and selection of alternatives ([116], [215], etc.). Adelson introduced the following impasses in group work situations: goal decomposition, goal reformulation, allocation of limited resources, role interdependence, and role conflict [3]. In fact, there exist two main types of conflicts by the reason: (i) psychological conflict and (ii) professional conflict. Evidently, these reasons are mixed in real situations.

Usually we face the following professional conflict situations:

1. Between high-level specialists (system thinking) and bottom-level ones. The specialists see different things into height (different system thinking and design styles).

2. Between specialists from different product life cycle stages. The specialists see things from different viewpoints (R & D, manufacturing, marketing, and utilization).

3. Between specialists from different design disciplines. The specialists see different things and different professional aspects (e.g., mechanics, software, human engineering, etc.).

4. Between specialists in a design discipline (domain). The specialists see different levels of a thing into depths because they have different professional levels, knowledge and experience.

To avoid, and to resolve the professional conflicts it is reasonable to use special organizational approaches and support procedures for modeling and information aid ([218], etc.).

On other hand, the issues of human factors require the special study. Let us reveal the following major tasks of this kinds ([193], [324], [373], etc.):

(i) user's diagnosis;

(ii) user's training; and

(iii) selection and allocation of specialists into stages and roles of design process.

First let us consider some issues of design team organization. It is very important to develop environments which provide effective integration of several domain specialists for complex multi-disciplinary synthesis problems ([113], [247], [373], etc.).

Note that the tree-like model of designed systems and its hierarchical description, which plays a central information role in HMMD, is an analogue of a hierarchical blackboard for collaboration of design team participants and support relationships and channels of communication among the various participants. Here we consider a four level fragment of a hierarchical system (Table 9.1) including corresponding basic tasks. In this fragment, we consider the following types of specialists:

(1) specialists of system level - top level specialists or personnel (P^t);

(2) specialists of stages of life cycle product (stage level personnel - P^s);

(3) specialists of subsystem level (for each stage) - middle level personnel (P^m); and

(4) domain specialists of component level - bottom level personnel (P^c).

In addition, we introduce external personnel P^e. Obviously, considered four-level fragment may correspond to different system hierarchical fragments. In Table 9.1, formulae $P^{'}\&P^{''}$ denotes major personnel ($P^{'}$) and auxiliary personne ($P^{''}$) who participates in the process under consideration also. It is implied that initial generation of a decision model is executed by P^t. Fig. 9.2 depicts an example of specialists allocation for four-levels system (the 2nd index of P denotes subsystems 1, 2, 3, 4 and system components a, b, c, d, e, f, g, and h).

We use the following designations:

Cr - specification of criteria;

Co - specification of constraints;

In - specification/evaluation of interconnection (compatibility between components);

Ra - ranking DA's;

Ev - evaluation of DA's upon criteria;

Cm - composition of DA's at the higher levels;

An - analysis of composite DA's; and

Ge - generation of DA's at the bottom level.

Table 9.1. Relationship of task&specialists

Task	Specialists	Task	Specialists
	Top system level		
Cr	$P^e\&P^t$	*Ra*	$P^t\&P^s$
Co	$P^e\&P^t$	*Ev*	$P^t\&P^s$
In	$P^t\&P^s$	*Cm*	$P^t\&P^s$
		An	$P^t\&P^s$
	Life cycle stage system level		
Cr	$P^t\&P^s$	*Ra*	$P^s\&P^m$
Co	$P^t\&P^s$	*Ev*	$P^s\&P^m$
In	$P^s\&P^m$	*Cm*	$P^s\&P^m$
		An	$P^s\&P^m$
	Middle system level		
Cr	$P^s\&P^m$	*Ra*	$P^m\&P^b$
Co	$P^s\&P^m$	*Ev*	$P^m\&P^b$
In	$P^m\&P^b$	*Cm*	$P^m\&P^b$
		An	$P^m\&P^b$
	Bottom system level		
Cr	$P^m\&P^b$	*Ra*	P^b
Co		*Ev*	P^b
In		*Ge*	P^b

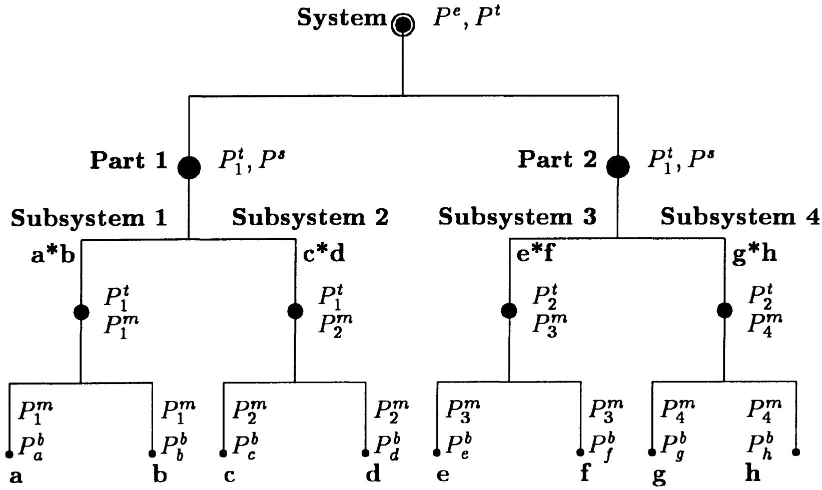

Fig. 9.2. Specialist allocation for four-levels design process

The following characteristics may be used for an analysis of designers ([98], [294], [440], etc.):

(a) professional level (knowledge, experience, skills);
(b) design style (convergent, divergent);
(c) creativability; and
(d) ability to system thinking.

Quinn et al. have analyzed ***professional intellect*** that consists of the following analogical components [404]:

(1) advanced skill (***know-how***);
(2) cognitive knowledge (***know-what***);
(3) self-motivated creativity (*care-why*); and
(4) systems understanding (***know-why***).

Morse and Hendrickson examined the following major roles in design processes [373]:

(a) project manager (issues of budget, project scope and definition, schedule, etc.);
(b) project engineer or design coordinator (maintains the global perspective and decision-making authority to direct an integration of local decisions in accordance with overall project objectives); and
(c) design agent (technical knowledge to produce locally efficient designs).

Note that F.P. Brooks has considered close basic roles (e.g., system architect, coordinators, and local specialists) for very large projects [64]. In HMMD, these three roles correspond to the following: (a) P^e; (b) P^t; and (c) P^m and P^b. Liebisch and Jain consider four similar roles: framework administrator, design methodology manager, project manager, and design engineer [324]. Hales identified nine roles for design processes as follows [193]:

(a) general roles (project promoter; team builder and leader; and coordinator and negotiator); and

(b) roles being linked to specific design phases of design processes (cross-examining consultant; design project manager; creative designer; concept developer; detail designer; and drafting supervisor and draftperson).

In our case, the role of specialist can be referred to the system level, system component, and phase of the design process. Note that the correlations above are not synonymous: P^e corresponds to general roles, ans P^b corresponds to detail designers. System specialists (P^t) correspond to cross-examining consultants (exploration of compatibility, etc.) and concept developers. Some roles are required at all levels:

(a) the coordinator and negotiator are required at every level for the tasks as follows: In, Cm, Co, and An;

(b) creative designers can execute operation Ge;

(c) concept developers design the system model, a hierarchy of criteria (Cr), etc.

The decomposition of a role set is oriented to take into account limits of specialists ([357], etc.).

9.3 ANALYSIS AND TRAINING OF SPECIALISTS

Here let us consider approaches to the analysis, diagnosis, and training of specialists (users). It is an important auxiliary stage of design team organization. Generally, it is reasonable to list basic operations of traditional logic as follows [533]:

1. *Definition.*
2. *Comparison and discrimination.*
3. *Analysis.*
4. *Abstraction.*
5. *Generalization.*
6. *Forming close concepts.*
7. *Subsumption.*
8. *Forming proposition.*
9. *Forming inference.*
10. *Forming syllogisms.*

Wertheimer has suggested that the ability to carry out these operations corresponds to a mark of intelligence [533]. Thus it is reasonable to analyze the ability of a specialist to operations above. On the other hand, often it is necessary to investigate various professional fields. Also, we take into account human knowledge on several technological components of human-computer systems as follows: (1) domain; (2) conceptual solving strategy; (3) mathematical models; (4) algorithms and man-machine procedures; (5) software; (6) hardware; and (7) information.

Many investigations use a similar approach and it relates with conceptual modeling ([72], [433], etc.). Fig. 9.3 demonstrates a morphological scheme of a specialist. Our morphological description is based on three levels of specialist knowledge by Piaget [395]: (a) preobjective and preoperational; (b) certain objects and operations; and (c) abstract objects and operations.

Similar levels

detailed/specific/concrete operations $\Longleftrightarrow$ global/general/abstract ones

are applied in some other studies ([433], etc.). For example, Powell examined three approaches: (i) experimental learning style model of Kolb [257], (ii) model for learning, and (iii) cognitive development of Piaget [396]. As a result, he proposed the following designer's self-informing styles: (1) contemplative; (2) focused; (3) dynamic; and (4) rigorous [401].

In our opinion, an analysis of the morphological scheme for certain specialist(s) is significant preliminary stage for the selection, training and allocation. In addition, revelation of user's profiles is the basis for system adaptation in on-line modes.

Thus it is reasonable to use the following factors to analyze kinds of specialists ([294], [98], [401], [416], [417], etc.):

(a) domains (type, stage, and level of design information);

(b) major type of self-informing styles: (i) contemplative; (ii) focused; (iii) dynamic; and (iv) rigorous;

(c) knowledge on design strategies, design algorithms, and/or procedures;

(d) two types of thinking: (i) analytical, systemic or scientific, and (ii) problem-oriented or decision-oriented thinking;

(e) ***cognitive style***, i.e., ability to think relatively context free ***field-independent*** or only ***field-dependent***; (it is interesting to note that it is possible to meet some specialists of the third kind: ***technique/method-oriented***).

Strategies of training are based on the following transformation:

$$Initial\ Specialist(s) \Longrightarrow Target\ Specialist(s).$$

The planning of the transformation may be considered as a combination of the following standard operations [382]: (i) directive indications; (ii) explanations; (iii) observation of examples; and (iv) discovery. Also, the HMMD may be used for the construction of the transformation learn strategy.

Knowledge & experience on technological components	Level of specialist knowledge (by J. Piaget)		
	Preobjective & preoperational (A)	Certain objects& operations (B)	Abstract objects& operations (C)
Problem domain	@ ⇒	*	
Conceptual solving schemes		@ *	
Models	@ ⇒		*
Algorithms, man-machine procedures	@ ⇒	*	
Software	* ⇒	@	
Hardware		@ *	
Design information		@ *	

* and @ correspond to two examples

Fig. 9.3. Morphological scheme of a specialist

9.4 DESIGN OF TEAM

Here we consider a team consisting of the four elements (Fig. 9.4):

1. Leader L (e.g., principal researcher, author of project, and "surgeon"). This role corresponds to the following: generation of basic ideas, specification to project elements, preparation of technical documentation, etc.

2. Researcher R or the 2nd pilot (the 2nd "I" of the leader).

3. Manager M (responsibility for personnel, salaries, premises, marketing strategies, etc.).

4. Detail designer D (design of project parts).

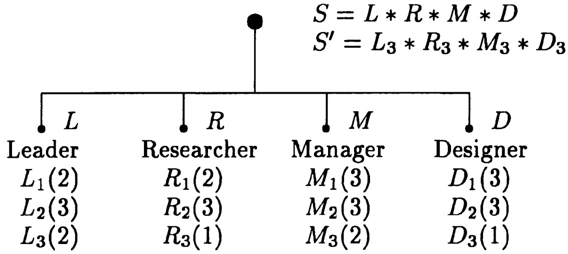

Fig. 9.4. Hierarchical structure of team (priorities of DA's are shown in brackets)

Our basic requirements are the following:

I. Professional requirements:

1.1 Educational background and experience in the following: engineering (F_1), scientific research (F_2), management (F_3), and foreign languages (F_4).

1.2 Erudition (F_5).

1.3 Ability to the analysis and diagnosis of engineering and scientific situations and directions (including possible implementation, conflicts, and forecasting) (F_6).

1.4 Creativity, including the following: generation of new ideas, concepts, non-habitual decisions (creativability) (F_7), and ability to system thinking (F_8).

II. Organizational requirements:

2.1 Planning (F_9).

2.2 Allocation of specialists (F_{10}).

2.3 Education and training of team members (F_{11}).

III. Socio-psychological features:
3.1 Justice and honesty (F_{12}).
3.2 Politeness (F_{13}).
3.3 Attention to colleagues (F_{14}).
3.4 Communication skills (F_{15}).
3.5 Humor (F_{16}).
Table 9.2 contains weights of criteria with taking into account roles above.

Table 9.2. Criteria

Criteria	Weights			
	L	R	M	D
F_1	5	5	0	5
F_2	3	3	0	0
F_3	3	0	5	0
F_4	3	2	4	0
F_5	4	0	0	0
F_6	4	3	4	0
F_7	5	3	0	0
F_8	4	3	2	0
F_9	4	0	4	2
F_{10}	4	0	4	0
F_{11}	5	0	3	0
F_{12}	5	5	5	5
F_{13}	4	0	5	0
F_{14}	3	0	5	0
F_{15}	3	3	5	3
F_{16}	3	3	3	0

The following basic DA's are considered: L_1, L_2, R_1, R_2, R_3, M_1, M_2, D_1, D_2, and D_3. In addition, we examine a situation when person L_1 will occupy two positions of the leader and manager. For this case DA's L_3 and M_3 are added with corresponding estimates. Note that to obtain a required resultant structure the following compatibility is used:

(1) $w(L_3, M_3)$ equals maximum one, and

(2) compatibility of L_3 with other DA's of M and M_3 with other DA's of L are equal to 0.

In our opinion, it is reasonable to use the following compatibility factors: (a) age, (b) course of life, (c) professional background and orientation, and (d) basic hobby (sport, tourism, etc.).

Hypothetical estimates and compatibility for our example are contained in Tables 9.3 and 9.4. Here we use ordinal scale (0...5). Note that in our example it is possible to examine negative compatibility too. Fig. 9.4 presents priorities of DA's. The best composite design alternative is the following: $S' = L_3 * R_3 * M_3 * D_3$ with $N(S') = (5; 2, 2, 0)$. Table 9.5 illustrates bottlenecks and improvement actions.

Table 9.3. Estimates

DA's	Criteria															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
L_1	4	5	4	3	5	5	4	5	4	4	5	5	3	4	4	3
L_2	5	5	4	5	5	3	4	3	2	3	2	3	3	3	3	2
L_3	4	5	4	3	5	5	4	5	4	4	5	5	3	4	4	3
R_1	4	3		4		3	2	3				4			5	4
R_2	2	5		5		4	2	4				4			3	2
R_3	3	5		5		5	4	5				5			5	3
M_1			4	5		2		3	3	3	3	4	4	4	5	3
M_2			5	5		3		4	3	4	2	4	3	3	5	3
M_3			4	3		5		5	4	4	5	5	3	4	4	3
D_1	3								2			5			4	
D_2	3								3			3			5	
D_3	4								3			5			4	

Table 9.4. Compatibility of DA's

	R_1	R_2	R_3	M_1	M_2	M_3	D_1	D_2	D_3
L_1	4	3	5	3	4	5	5	3	5
L_2	5	3	3	4	4	0	3	5	3
L_3	4	3	5	0	0	5	5	3	5
R_1				5	5	4	4	5	4
R_2				4	4	3	4	4	4
R_3				5	5	5	5	4	5
M_1							4	5	4
M_2							4	5	4
M_3							5	3	5

Table 9.5. Bottlenecks and actions

Composite DA's	Bottleneck		Action w/r
	DA	Ins	
$S' = L_3 * R_3 * M_3 * D_3$	L_3		$2 \Longrightarrow 1$
$S' = L_3 * R_3 * M_3 * D_3$	M_3		$2 \Longrightarrow 1$

In addition, we consider a transformation process (i.e., improvement) of the team. A structure of the transformation process is depicted in Fig. 9.5. Criteria, compatibility and DA's are presented in Tables 9.6, 9.7, and 9.8, respectively. Resultant composite DA's are the following (Fig. 9.6):

(1) $N_1 = (5;2,0,1)$: $S_1 = A_3 * B_1 * C_2$, $S_2 = A_3 * B_1 * C_3$, $S_3 = A_3 * B_4 * C_2$, $S_4 = A_3 * B_4 * C_3$; and

(2) $N_2 = (5;2,1,0)$: $S_5 = A_1 * B_1 * C_2$, $S_6 = A_1 * B_1 * C_3$, $S_7 = A_1 * B_3 * C_2$, $S_8 = A_1 * B_3 * C_3$; $S_9 = A_1 * B_4 * C_2$, $S_{10} = A_1 * B_4 * C_3$, $S_{11} = A_2 * B_1 * C_2$, $S_{12} = A_2 * B_1 * C_3$; $S_{13} = A_2 * B_3 * C_2$, $S_{14} = A_2 * B_3 * C_3$, $S_{15} = A_2 * B_4 * C_2$, $S_{16} = A_2 * B_4 * C_3$.

Clearly, it is reasonable to investigate multi-stage improvement processes.

Plan for improvement of team
$S = A * B * C$
$S_1 = A_3 * B_1 * C_2$
$S_2 = A_3 * B_1 * C_3$
...

A New members
$A_1(2)$
$A_2(2)$
$A_3 = A_1 \& A_2(3)$

B Professional courses
$B_1(1)$
$B_2(2)$
$B_3(1)$
$B_4(1)$
$B_5 = B_3 \& B_4(2)$
$B_6 = B_1 \& B_4(3)$
$B_7 = B_1 \& B_2 \& B_4(3)$

C Trips & communication
$C_1(2)$
$C_2(1)$
$C_3(1)$
$C_4 = C_1 \& C_3(3)$

Fig. 9.5. Structure of team improvement process (priorities of DA's are shown in brackets)

Table 9.6. Criteria

Criteria	Weights		
	A	B	C
1.Cost (-)	5	3	4
2.Correspondence to personnel		4	5
3.Required time (-)	3	3	2
4.Improvement of output	5	5	
5.Improvement of friendship		2	5
6.Improvement of creativity and system thinking	4	3	2

Table 9.7. Compatibility of DA's

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	C_1	C_2	C_3	C_4
A_1	5	2	5	5	5	5	2	4	4	4	4
A_2	3	4	4	3	3	3	3	5	4	5	5
A_3	5	4	5	5	5	5	3	5	4	5	5
B_1								2	1	5	2
B_2								4	2	5	4
B_3								5	4	4	5
B_4								4	5	5	5
B_5								4	4	4	5
B_6								2	1	5	2
B_7								2	1	5	2

Table 9.8. DA's and estimates

DA's	Criteria					
	1	2	3	4	5	6
A_1 New leader	4		4	3		5
A_2 New manager	3		3	2		4
$A_3 = A_1\&A_2$	7		4	4		6
B_1 Course on advances in science & engineering	5	3	2	2	3	1
B_2 Course on foreign languages	3	2	3	1	3	1
B_3 Course on system analysis	4	2	1	2	3	1
B_4 Course on creativity	5	2	1	3	3	1
$B_5 = B_3\&B_4$	9	2	2	4	3	1
$B_6 = B_1\&B_4$	10	2	3	4	3	1
$B_7 = B_1\&B_2\&B_4$	13	2	6	5	3	1
C_1 Course on human relation	4	2	1		3	0
C_2 Joint trip to rest-home	7	5	2		5	0
C_3 Joint participation at professional exhibition	6	4	2		4	1
$C_4 = C_1\&C_2$	10	3	3		6	1

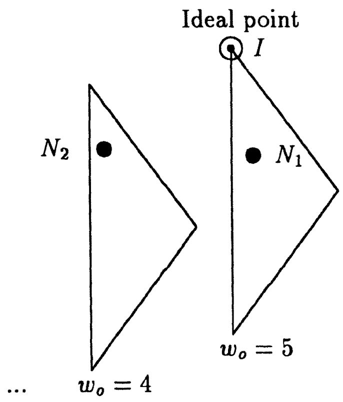

Fig. 9.6. Space of system excellence

9.5 PLANNING OF PRODUCT LIFE CYCLE

Let us consider an example that corresponds to a prototype of HMMD. We select only three components of product life cycle:

(1) research and development of models and case studies;

(2) development of system support tools, including: (a) information (data and knowledge) support, and (b) shell/interface;

(3) marketing, including marketing planning, prices, activities (in universities, industry), and publications.

Note similar studies have been applied in many research directions (e.g., strategic planning, technology management, mutliphase decision making, systems engineering, and concurrent engineering) ([17], [402], [440], [451], [542], etc.). Our tree-like system model is depicted in Fig. 9.7.

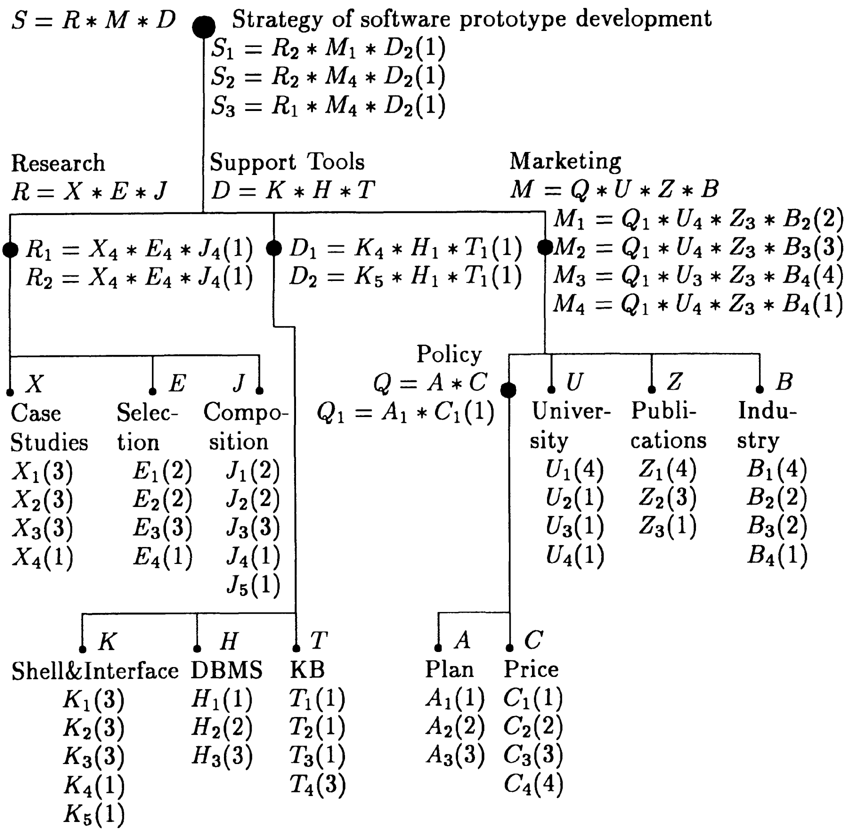

Fig. 9.7. Hierarchical structure of software development process (priorities of DA's are shown in brackets)

Table 9.9 contains criteria for DA's (ordinal scale 0...5, negative monotonicity is shown in brackets). Aggregate criteria for composite DA's are presented in Table 9.10, where for each criterion (F) the 1st letter index corresponds to the component and the 2nd index corresponds to the number of the criterion. Table 9.11 contains DA's and their estimates on criteria, Table 9.12, 9.13, 9.14, and 9.15 present estimates of compatibility, and Table 9.16 contains composite DA's. Fig. 9.8 depicts a concentric presentation of the composition problem. Composite DA's at the space of system excellence are illustrated in Fig. 9.9. The computation of estimates and multicriteria ranking are executed only for market part M. Table 9.17 presents the results of an analysis (bottlenecks, improvement actions).

Table 9.9. Criteria

Criteria	Weights														
	K	H	T	X	E	J	R	D	A	C	U	Z	B	Q	M
1.Cost (-)	3	4	4	2	1	1	1	3	1	1	1	1	2	1	1
2.Time of preparation (-)	4	4	4	4	4	4	4	4	2	2	1	2	2	2	2
3.Easy of use	5	2	3	5	5	5	4	4							
4.Learnability	1	1	2	5	5	5	4	5							
5.Creative level	4	1	1		1	5	5	4							
6.Usefullness for advertisement	5	1	2	5	1	5	5	3	5	5	4	5	4	5	5
7.Possibility to develop a market									5	5	1	5	1	5	5
8.Learning of potential consumers									5	5	2	5	2	4	4
9.Possibility of untimely disseminating of methods (waste of author right) (-)									4	4	4	5	4	4	4
10.Usefullness for new applications									5	5	5	5	3	4	4
11.Professional acknowledgement									3	1	5	1	5	4	4
12.Possibility for profit									4	5				3	3

Table 9.10. Aggregate criteria

Cr	Specification
F_{q1}	$F_{a1} + F_{c1}$
F_{q2}	$\max(F_{a2}, F_{c2})$
F_{q6}	$\min(F_{a6}, F_{c6})$
F_{q7}	$\min(F_{a7}, F_{c7})$
F_{q8}	$\max(F_{a8}, F_{c8})$
F_{q9}	$\max(F_{a9}, F_{c9})$
F_{q10}	$\min(F_{a10} + F_{c10})$
F_{q11}	$\max(F_{a11}, F_{c11})$
F_{q12}	$F_{a12} + F_{c12}$
F_{m1}	$F_{p1} + F_{u1} + F_{z1} + F_{b1}$
F_{m2}	$\max(F_{q2}, F_{u2}, F_{z2}, F_{b2})$
F_{m6}	$\min(F_{q6}, F_{u6}, F_{z6}, F_{b6})$
F_{m7}	$\min(F_{q7}, F_{u7}, F_{z7}, F_{b7})$
F_{m8}	$\max(F_{q8}, F_{u8}, F_{z8}, F_{b8})$
F_{m9}	$\max(F_{q9}, F_{u9}, F_{z9}, F_{b9})$
F_{m10}	$\min(F_{q10}, F_{u10}, F_{z10}, F_{b10})$
F_{m11}	$\max(F_{q11}, F_{u11}, F_{z11}, F_{b11})$
F_{m12}	F_{q12}

Table 9.11. DA's and estimates

DA's	Criteria											
	1	2	3	4	5	6	7	8	9	10	11	12
K_1 None	0	0	0	0	0	0						
K_2 Hierarchical menu system	1	2	1	3	1	0						
K_3 Menu& data presentation	2	3	2	2	1	1						
K_4 New shell (interface)	4	4	4	2	4	4						
K_5 New shell& data presentation	5	5	5	1	4	4						
H_1 Files	1	1	2	1	0	0						
H_2 Interface with DBMSs	2	2	1	3	1	2						
H_3 Special DBMS	4	4	1	2	2	1						
T_1 None	0	0	0	0	0	0						
T_2 Simple support of rules	1	1	4	3	1	1						
T_3 Interface with KBMSs	2	1	3	3	1	3						
T_4 Shell for special KBMS	6	4	3	2	4	5						
E_1 Heuristic	1	0	3	1	0	0						
E_2 Well-known model	2	0	3	1	0	0						
E_3 Several well-known models	3	2	2	3	1	1						
E_4 New model	3	4	3	5	4	4						
J_1 Heuristic	1	0	3	1	0	0						
J_2 Well-known model	2	0	3	1	0	0						
J_3 Several well-known models	3	2	2	3	1	1						
J_4 New model	3	4	3	5	4	4						
J_5 New model& analysis	4	6	3	5	5	6						
X_1 None	0	0	0	0		0						
X_2 One basic case study	1	1	1	0		0						
X_3 Several case studies	2	2	2	1		1						
X_4 Library of case studies	5	6	7	6		6						
Z_1 None	0	0				0	0	0	0	0	0	
Z_2 A case study (a company)	1	1				1	1	1	1	2	1	
Z_3 Case studies (companies)	2	3				5	5	4	2	5	2	
U_1 None	2	3				5	5	4	2	5	2	
U_2 Seminar	1	1				1	1	3	3	2	1	
U_3 Course	2	2				1	1	3	3	1	2	
U_4 Several students	2	3				1	1	2	2	4	1	
B_1 None	0	0				0	0	0	0	0	0	
B_2 Presentation at conferences	4	2				2	4	1	4	3	2	
B_3 Publishing a paper(s)	2	2				2	3	1	4	2	3	
B_4 Publishing a book	2	4				4	4	3	3	3	3	

Table 9.11. Continuation

DA's	Criteria											
	1	2	3	4	5	6	7	8	9	10	11	12
A_1 Conquest of market	3	4				5	5	5	4	5	4	1
A_2 Balancing	3	3				4	4	3	3	3	3	3
A_3 Obtaining a high profit	4	4				2	1	2	4	1	1	5
C_1 Charge-free	2	1				5	5	5	5	5	5	0
C_2 Superlow price		2				3	4	3	3	3	3	1
C_3 Low price	3	3				2	3	2	3	2	3	2
C_4 High price	4	4				1	2	1	3	1	3	5

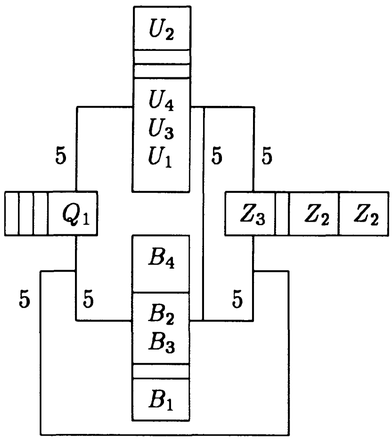

Fig. 9.8. Concentric presentation of composite DA

Table 9.12. Compatibility of DA's

	H_1	H_2	H_3	T_1	T_2	T_3	T_4
K_1	4	0	0	3	2	0	0
K_2	5	4	2	4	4	4	3
K_3	5	5	5	4	4	4	3
K_4	5	4	5	1	5	5	5
K_5	5	4	5	1	5	5	5
H_1				5	5	3	3
H_2				5	3	5	4
H_3				5	5	4	5

Table 9.13. Compatibility of DA's

	C_1	C_2	C_3	C_4	D_1	D_2	M_1	M_2	M_3	M_4
A_1	5	4	2	1						
A_2	0	4	4	2						
A_3	0	2	3	5						
R_1					3	4	4	3	4	4
R_2					3	5	5	4	5	4
D_1							4	4	3	4
D_2							5	5	4	5

Table 9.14. Compatibility of DA's

	E_1	E_2	E_3	E_4	J_1	J_2	J_3	J_4	J_5
X_1	0	1	1	1	0	0	0	0	0
X_2	2	3	2	2	2	2	2	1	1
X_3	3	3	4	3	3	4	4	5	5
X_4	5	5	5	5	5	4	4	5	5
E_1					5	3	3	4	4
E_2					3	5	3	3	3
E_3					2	2	5	5	5
E_4					1	3	3	5	5

Table 9.15. Compatibility of DA's

	Z_1	Z_2	Z_3	B_1	B_2	B_3	B_4	Q_1
U_1	0	1	1	2	2	2	2	5
U_2	2	3	5	2	3	3	3	5
U_3	2	3	4	2	3	3	5	5
U_4	2	3	5	2	5	5	4	5
Z_1				1	1	1	1	5
Z_2				1	3	2	2	5
Z_3				1	3	5	5	5
B_1								5
...								..
B_4								5

Table 9.16. Composite DA's

Composite DA's	N
$R_1 = X_4 * E_4 * J_4$	$(5;3,0,0,0)$
$R_2 = X_4 * E_4 * J_5$	$(5;3,0,0,0)$
$D_1 = K_4 * H_1 * T_2$	$(5;3,0,0,0)$
$D_2 = K_5 * H_1 * T_2$	$(5;3,0,0,0)$
$Q_1 = A_1 * C_1$	$(5;2,0,0,0)$
$M_1 = Q_1 * U_4 * Z_4 * B_2$	$(5;4,1,0,0)$
$M_2 = Q_1 * U_4 * Z_4 * B_3$	$(5;4,1,0,0)$
$M_3 = Q_1 * U_3 * Z_4 * B_4$	$(4;5,0,0,0)$
$M_4 = Q_1 * U_4 * Z_4 * B_4$	$(4;5,0,0,0)$
$S_1 = R_2 * D_2 * M_1$	$(5;2,1,0,0)$
$S_2 = R_2 * D_2 * M_4$	$(4;3,0,0,0)$
$S_3 = R_1 * D_2 * M_4$	$(4;3,0,0,0)$

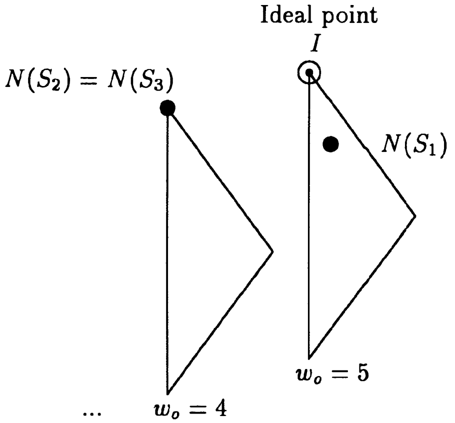

Fig. 9.9. Space of system excellence

Table 9.17. Bottlenecks and actions

Composite DA's	Bottleneck		Action w/r
	DA	Ins	
$M_1 = Q_1 * U_4 * Z_4 * B_2$	B_2		$2 \Longrightarrow 1$
$M_2 = Q_1 * U_4 * Z_4 * B_3$	B_3		$2 \Longrightarrow 1$
$M_3 = Q_1 * U_3 * Z_4 * B_4$		(U_3, Z_4)	$4 \Longrightarrow 5$
$M_4 = Q_1 * U_4 * Z_4 * B_4$		(U_4, B_4)	$4 \Longrightarrow 5$
$S_1 = R_2 * D_2 * M_1$	M_1		$2 \Longrightarrow 1$
$S_2 = R_2 * D_2 * M_4$		(R_2, M_4)	$4 \Longrightarrow 5$

9.6 SUMMARY

In this chapter, we have considered several key problems of personnel management, and planning of product life cycle. In our opinion, presented examples can be modified and applied for many close situations in business as well as in education processes.

10 EDUCATIONAL ISSUES

This chapter addresses educational issues on the basis of combinatorial synthesis ([303], [304], [306], [307], [309]). In the main we examine education of information engineering.

In this chapter, we consider the following: (a) issues of intellectual activity; (b) approaches to solve complicated problems; (c) a classification of education processes on the basis of kinds of target specialists; (d) a structure of a course on information technology on the basis of decision making and composing; (e) design of a course on hierarchical design; and (f) forming a career plan.

It is reasonable to point out special computer systems for student advising ([100], [178], [187], etc.). In particular, *Knowledge-Based Tutoring Systems* (KBTSs) are oriented to the following: *what*, *when*, *how*, and *whom* they are teaching and can tailor their contents to a certain individual student [553]. KBTSs usually consist of the following three parts: (a) domain knowledge base; (b) student model; and (c) pedagogical knowledge base (e.g., tutoring strategies).

Clearly, our study and examples may be applied as components for the systems above. We use our hierarchical approach to represent information on educational courses. Note that a hierarchical architecture to represent and to manipulate curriculum knowledge has been proposed in [553].

10.1 INTELLECTUAL ACTIVITY

In our opinion, the following recent tendencies may be pointed out:

(1) a part of composite technologies which involve contemporary results from different domains (e.g., engineering, biology, computer science, mathematics, etc.) is increasing;

(2) acceleration of implementing the innovations;

(3) macro-technological cycle (technological changing of complex product's generations) is decreasing from 12 to 2-3 years.

Note that now decision making is a technological process at each work place for each employee (clerk, engineer, worker, etc.) ([196] etc.). Moreover, one can see that intellectual levels of operations at each stage of product life cycle (not only management or R & D, but maintenance and utilization too, etc.) is increasing.

Let us consider a scale to evaluate an intelligence. First we examine creative levels by Altshuller as the following ordinal scale [14]:

1. The use of a well-known object (product, technology, management decision, strategy, innovation, etc.).

2. Searching for and selection of an object.

3. Analysis and modification of an initial well-known object.

4. Design of a new object.

5. Design of a new object's system.

Only the 4th and 5th levels above are really creative ones, and level 3 has an intermediate character. The level 2 (search and selection) is the basic one in decision making. Hereafter we will consider levels 2 and 3 as quasi-intellectual or quasi-creative. Note that an experience in decision making may be an excellent basis for specialists with an intermediate creative level to solve problems of levels 2 and 3. Moreover, the correct organization and support of a team work allow to solve problems for levels 4 and 5 on the basis of a set of quasi-intellectual problems (when intellectual decisions are combinations of local quasi-intellectual decisions).

It is reasonable to point out the following two possible ways:

1. Searching for creative persons who can solve intellectual problems. This process involves searching for, selection, training. This way is limited because a limited number of persons have an inclination to creative activities and the process of the selection and training of creative persons has also some limitations.

2. Construction of global creative decisions on the basis of composition of local quasi-creative decisions. This way is similar to the Shannon's approach to the design of reliable systems from unreliable components. In our case, we are oriented to compose decisions of levels 4 and 5 from local decisions of level 2 and 3. Note that analogical organizational approach has been applied in intermediate centuries in masteries of famous painters, when some details of oil-paintings were executed by apprentices.

Finally, we point out the significance of the following:

(a) composite decision making problems;

(b) technological solving schemes for composite decision making problems;

(c) support of team work in decision making; and

(d) special tools for composing of local decisions of different domains.

10.2 COMPLICATED PROBLEMS

The majority of applied problems are complex and composite ones. As a result, it is reasonable to apply special approaches as follows [293]:

(1) division of initial problems into parts (decomposition, partitioning);

(2) choice of corresponding tools (e.g., techniques, experts, and techniques of information presentation);

(3) aggregation of results; and

(4) planning of decision making process (i.e., design of technological process of decision making, including the following stages: information accumulation, analysis of information, decomposition and parallelism of information processing, and aggregation of results, etc.).

Now let us examine several basic parallel-series composite problems. First a problem of revealing the bottlenecks in a complex system corresponds to mutli-criteria choice/ranking. But we have pointed out the problem as a specific class because the problem is very important for applications. Usually this problem consists of generating a criterion set to reveal the most significant system parts (bottlenecks). This approach is fruitful for complex engineering systems, transportation and communication networks. On the other hand, the problem may be implemented as searching for a place for possible impacts, for example, in organizational-economical or ecological systems. An example of a similar bottleneck problem has been described in [35]. This example involves an analysis of 34 elements of a gas pump while taking into account 6 criteria (frequency of refusals, required time and cost of restoration, impacts to other elements, usage of the examined element in various technical systems, and influence to human safety). As a result of problem solving, we obtain a set of more important system elements. Clearly, it is reasonable to improve these elements while taking into account some factors (technical, economical, ecological, etc.).

Next complex problem consists in hierarchical composing (or combinatorial synthesis). This problem is very complicated at the applied level, and at the organizational level. The problems of hierarchical composition correspond to an approach when we generate high-creative system decisions on the basis of an integration of results of quasi-intellectual local subproblems.

Here we would like to point out the significance of reengineering. Recently, similar problems existed in all problem domains. In our opinion, each activity is executed in existing environments (systems, traditions, habits, etc.). This situation exists in engineering, housing, economical and financial domain, law, ecology, etc., and one has to take it into account.

We can see the situation of reengineering, for example, in urban conglomerations, in electronics (re-use of scheme design's decisions, etc.), in software development and in information engineering (re-use of components), and in management ([149], [196], etc.).

In the main, the following three possible strategies exist:

1. Design a new system instead of an old one. Similar strategy has been applied in ancient Rome: to destroy old buildings, to clear and to smooth out a place, and to build a new city on the basis of a standard design decision.

2. Modification of an existing system (mainly, to change a part of the system).

3. Design of a new system with the use of some old system components (e.g., 70...80 percents).

Evidently, complex, creative and adaptive approach is required in the third strategy, In this case, it is necessary to execute the following: (a) an analysis of the old (existing) and new (under design) systems; (b) an analysis of all components of the old and new systems; (c) revelation of bottlenecks; and (d) specification of requirements to the design of the new system while taking into account the re-use of the old components, etc. Thus we have to apply hierarchical decision making (including decomposing and composing).

Finally let us note important basic operations to search for and to process information (Table 10.1). Basic operations from Table 10.1 may be used as a fundamental for education process.

Table 10.1. Basic operations and their implementation

Basic operations	Approaches to implementation
1.Information retrieval (logical selection of data)	Traditional logical search
2.Multicriteria choice/ranking	Multicriteria analysis
3.Approximation, aggregation, modification	Structural approximation, computation of consensus Parametric optimization Organizational techniques as brain storm
4.Designing and maintenance of structures	Synthesis of hierarchical structures Structural modeling
5.Composing of new composite concept	Organizational techniques of solving schemes (e.g., brain storm, etc.) Morphological analysis Mathematical programming Hierarchical morphological multicriteria design

10.3 ORIENTATION OF EDUCATION PROCESS

In our opinion, teaching of complex problem solving is a very difficult problem. It is very important when we have students who can work and think only at the levels of certain objects and discipline-dependent descriptions. Let us consider basic problems in the field of information technology (IT) as follows:

(1) multicriteria description and analysis of IT components;

(2) comparison and selection/choice of IT components;

(3) composition of several IT components into a whole system while taking into account quality of element and their compatibility (interconnection);

(4) assignment/location of IT components in a network.

These problems may be examined at different formal levels as follows:

(1) user's level (choice of a decision);

(2) engineer's level (mainly, conceptual description and design as a modification); and

(3) mathematical level (mathematical modeling).

Finally it is reasonable to examine the following three basic strategies of teaching for three types of students as follows (Table 10.2): (i) economists or managers/users *ec*; (ii) engineers/developers of IT components *en*; and (iii) systems engineers (e.g., specialists in system integration) *se*.

Table 10.2. Relationship of education strategies and student types

Basic problems of IT teaching	Level of teaching		
	Consumer	Engineering	Mathematical
1.General description of IT (components, their interrelation, tendencies)	*ec*, *en*	*ec*, *en*	
2.Multicritertia description, analysis of IT components	*ec*	*en*, *se*	*en*, *se*
3.Comparison and choice of IT components	*ec*	*en*, *se*	*en*, *se*
4.Composing of IT components (system integration)	*ec*, *en*	*se*	*se*
5.Assignment/location IT components in networks	*ec*, *en*	*se*	*se*
6.Reengineering of an existing system	*ec*, *en*	*se*	*se*

In our opinion, we can use several basic combinatorial models for the analysis and/or synthesis of IT components, i.e., discrete decision making problems ([293], [300], [304], etc.) to teach basic problems above (multicriteria description and analysis of IT components; comparison and selection; composing; and assignment/location of IT components) (Table 10.3).

10.4 COURSE ON INFORMATION ENGINEERING

Our basic course was oriented to students-managers and involved the following:

(1) generalized system material to acquaint with the contemporary development, marketing, and utilization/maintenance of IT;

(2) to acquaint with IT components (hardware, communication systems, main types of software packages, mathematical modeling, etc.) on the basis

of multicriteria descriptions, multicriteria comparison of alternatives for the above-mentioned IT components.

The multicriteria analysis and comparison of alternatives for IT components allows teaching of basic requirements to main IT components. In our course, the multicriteria analysis was implemented on the basis of DSS COMBI-PC (IBM PC) [294]. The course was oriented to system integration (to order IT components, to organize a competition and complex evaluation of IT systems).

Table 10.3. Relationship of basic IT problems and combinatorial models

Basic IT problems	Combinatorial models
1.Comparison, selection of components	Multicriteria analysis Knapsack-like problems
2.System integration on the basis of estimates of components and their compatibility	Combinatorial synthesis, including knapsack-like problems, quadratic programming, mixed integer programming, morphological analysis, morphological clique
3.Allocation/location of IT components in an existing system (evaluation of components, location with taking into account their interconnection)	Location and covering problems
4.Reengineering of IT systems (structural description of existing system, analysis of its components and their interconnection, revealing of bottlenecks and generation of improvement actions, scheduling of the re-engineering process	Multicriteria analysis Combinatorial synthesis Location and combinatorial synthesis Location and covering Scheduling

The program consists of the following 9 parts:

Part 1. Introduction: IT components, properties of IT, bottlenecks, real-time systems and their applications, DSS COMBI.

Part 2. Paradigm of decision making and DSS (classification of problems by H.Simon, stages of decision making, realistic examples of multicriteria analysis).

Part 3. Data bases, hypertext systems, multicriteria analysis and comparison of systems.

Part 4. Knowledge based systems (structure, approaches to knowledge representation, stages of developing the knowledge based systems, multicriteria analysis and comparison of the systems).

Part 5. Hardware, communication networks (types of computers, e.g., PC, workstation, mainframe; scheme of data processing in communication networks, examples of applications in banks, management of communication networks, multicriteria analysis and comparison of computers, communication systems).

Part 6. Human-computer interaction (structure, example of human-computer systems, human limitations in the information processing, user modeling, user interface adaptation, evaluation of human-computer interaction).

Part 7. Mathematical modeling, optimization software packages (structure of the systems, types of mathematical models, examples of systems, comparison of the systems).

Part 8. Distributed systems (distributed data bases and DSS, group and cooperative work, multi-agent systems, applied examples of distributed systems).

Part 9. Design of technologies (system integration, evaluation of components and their interconnection, integrated evaluation of composite systems, location of IT components).

In addition, our course includes tasks for each student at the following two levels/stages: (1) solving of several local problems of multicriteria analysis/comparison of IT components (with the use of DSS COMBI); and (2) composing of several IT components into a whole system (multicriteria selection of alternatives for several IT components and system integration).

The tasks above are based on the extended example which was described in [301] (section 8.3). Our students prepared their results as a reports.

Note, at the second stage, some local parts of student tasks are common, and it stimulates communication between students. Clearly, the above-mentioned education scheme may be implemented as a distance education process.

10.5 COURSE ON HIERARCHICAL DESIGN

Let us consider the problem of forming the education program for a course which corresponds to materials on HMMD. For each element of the course we satisfy the following list of options (DA's): without consideration (O_{p0}); brief consideration (O_{p1}); learning at the intermediate level (O_{p2}); deepening learning (O_{p3}); and deepening learning with special additional work, e.g., development of software, preparation of a paper, and solving sample problems (O_{p4}).

The following factors may be used for assessment of the options: (1) connection with future work; (2) basic professional orientation of a student group; (3) education background of students; (4) opportunity for executing an additional work; etc.

In our example, the priorities of the options are obtained on the basis of expert judgment. A structure of initial material and DA's is the following (priorities of DA's are shown in brackets):

0. Course $S = H * R * B * X$:
1. Description of HMMD $H = D * F * P * M * C * A$:
  1.1. Design approaches D: $D_1(1)$, $D_2(3)$, $D_4(2)$.
  1.2. Fundamentals of HMMD F: $F_2(1)$, $F_3(2)$.
  1.3. Components of HMMD P: $P_2(2)$, $P_3(1)$.
  1.4. Composition problem M: $M_2(2)$, $M_3(1)$, $M_4(3)$.
  1.5. Selection problem C: $C_1(1)$, $C_3(2)$, $C_4(2)$.
  1.6. Analysis and improvement A: $A_2(3)$, $A_3(1)$, $A_4(1)$.
2. Relative Domains $R = E * N * Q * L$:
  2.1. Engineering information E: $E_0(1)$, $E_1(3)$, $E_4(2)$.
  2.2. Conflict situations N: $N_1(2)$, $N_4(1)$.
  2.3. Quality analysis Q: $Q_1(1)$, $Q_4(2)$.
  2.4. Education and private life L: $L_1(2)$, $L_2(3)$, $L_4(2)$.
3. Combinatorial Optimization $B = K * T * W * U$:
  3.1. Knapsack problem K: $K_2(2)$, $K_3(2)$, $K_4(1)$.
  3.2. Traveling salesman problem T: $T_1(2)$, $T_4(1)$.
  3.3. Location problem W: $W_1(1)$, $W_4(3)$.
  3.4. Routing problem U: $U_0(2)$, $U_1(1)$, $U_4(3)$.
4. Sample Problems $X = O * I * J * V * G * Y * Z$:
  4.1. Integration of software O: $O_0(3)$, $O_1(1)$, $O_4(2)$.
  4.2. Design of HCI I: $I_0(2)$, $I_1(1)$, $I_4(3)$.
  4.3. Development of information center J: $J_0(12)$, $J_1(2)$, $J_4(2)$.
  4.4. Design of conveyor V: $V_0(3)$, $V_1(1)$, $V_4(2)$.
  4.5. Exploration of oil/gas fields G: $G_0(2)$, $G_1(2)$, $G_2(1)$.
  4.6. Planning a career Y: $Y_0(3)$, $Y_2(1)$, $Y_4(2)$.
  4.7. Selection of investment portfolio Z: $Z_0(2)$, $Z_1(2)$, $Z_3(1)$, $Z_4(3)$.

Compatibility between DA's are shown in Tables 10.4, 10.5, 10.6, 10.7, and 10.8. Composite DA's are shown in Table 10.9.

Resultant composition of the course is the following: $S_1 = H_2 * R_2 * B_1 * X_1 = (D_4 * F_3 * P_3 * M_3 * C_3 * A_4) *$

$(N_4 * Q_4 * L_4 * E_4) * (K_4 * T_4 * W_4 * U_4) * (V_1 * O_1 * I_4 * J_4 * Z_3 * Y_2 * G_2)$.
Table 10.10 contains bottlenecks, improvement actions.

Table 10.4. Compatibility of DA's

	F_2	F_3	P_2	P_3	M_2	M_3	M_4	C_1	C_3	C_4	A_2	A_3	A_4
D_1	1	2	1	2	2	3	2	1	2	2	5	5	5
D_2	4	4	4	4	4	4	5	4	4	4	5	5	5
D_4	1	5	2	5	1	5	4	3	4	4	5	5	5
F_2			4	5	4	4	5	2	4	4	4	3	3
F_3			5	5	4	5	5	3	5	5	4	5	5
P_2					4	4	5	5	4	4	5	3	3
P_3					4	5	5	4	5	5	4	5	5
M_2								5	4	4	5	3	3
M_3								3	5	4	3	5	5
M_4								3	5	4	3	5	5
C_1											5	5	5
C_3											5	5	5
C_4											5	5	5

Table 10.5. Compatibility of DA's

	F_2	F_3	P_2	P_3	M_2	M_3	M_4
E_0	1	1	2	2	5	1	1
E_1	4	4	2	3	3	5	4
E_4	4	5	3	5	4	5	5
N_1			4	5	5	4	4
N_4			5	4	4	5	5
Q_1					5	5	5
Q_4					5	5	5

Table 10.6. Compatibility of DA's

	T_1	T_4	W_1	W_4	U_0	U_1	U_4
K_2	3	4	3	4	3	5	5
K_3	4	5	4	4	3	5	5
K_4	4	5	4	5	3	5	5
T_1			5	5	2	4	3
T_4			5	5	2	3	5
W_1					5	5	5
W_4					5	5	5

Table 10.7. Compatibility of DA's

	I_0	I_1	I_4	J_0	J_4	V_0	V_1	V_4	G_0	G_2	Y_0	Y_2	Y_4	Z_0	Z_1	Z_4	Z_4
O_0	5	2	2	5	4	5	5	5	5	5	5	5	5	5	5	5	5
O_1	4	4	5	4	5	5	5	5	5	5	5	5	5	5	5	5	5
O_4	2	5	5	4	5	5	5	4	5	5	5	5	5	5	5	5	5
I_0				5	1	5	5	5	5	5	5	5	5	5	5	5	5
I_1				5	3	5	5	5	5	5	5	5	5	5	5	5	5
I_4				3	5	5	5	5	5	5	5	5	5	5	5	5	5
J_0						5	5	5	5	5	5	5	5	5	4	3	3
J_4						5	5	5	5	5	5	5	5	2	3	5	5
V_0									5	5	5	5	5	5	5	5	5
V_1									5	5	5	5	5	5	5	5	5
V_4									5	5	5	5	5	5	5	5	5
G_0											5	5	5	5	3	3	2
G_2											5	5	5	4	4	5	5
Y_0														4	4	3	3
Y_2														4	4	5	5
Y_4														4	4	5	5

Table 10.8. Compatibility of DA's

	R_1	R_2	R_3	R_4	R_5	R_6	B_1	B_2	B_3	X_1	X_2
H_1	4	5	2	3	1	1	4	2	2	3	2
H_2	5	5	3	4	1	1	5	4	4	5	3
H_3	4	5	3	4	1	1	4	2	2	3	2
H_4	5	5	3	4	1	1	5	4	4	5	3
H_5	4	5	2	3	1	1	4	2	2	3	2
H_6	4	5	2	3	1	1	5	4	4	5	3
H_7	2	3	1	2	1	1	4	2	2	3	2
H_8	2	3	1	2	1	1	5	4	4	5	3
R_1							5	2	2	4	2
R_2							5	4	2	5	3
R_3							3	2	2	2	2
R_4							3	4	2	4	2
R_5							3	2	2	2	1
R_6							3	2	2	4	2
B_1										5	3
B_2										4	3
B_3										3	5

Table 10.9. Composite DA's

DA's	N
$H_1 = D_4 * F_3 * P_3 * M_3 * C_3 * A_3$	$(4;3,3,0)$
$H_2 = D_4 * F_3 * P_3 * M_3 * C_3 * A_4$	$(4;3,3,0)$
$H_3 = D_4 * F_3 * P_3 * M_3 * C_4 * A_3$	$(4;3,3,0)$
$H_4 = D_4 * F_3 * P_3 * M_3 * C_4 * A_4$	$(4;3,3,0)$
$H_5 = D_4 * F_3 * P_3 * M_3 * C_1 * A_3$	$(3;4,2,0)$
$H_6 = D_4 * F_3 * P_3 * M_3 * C_1 * A_4$	$(3;4,2,0)$
$H_7 = D_1 * F_2 * P_3 * M_3 * C_1 * A_3$	$(1;6,0,0)$
$H_8 = D_1 * F_2 * P_3 * M_3 * C_1 * A_4$	$(1;6,0,0)$
$X_1 = V_1 * O_1 * I_4 * J_4 * Z_3 * Y_2 * G_2$	$(5;5,1,1)$
$X_2 = V_1 * O_1 * I_1 * J_0 * Z_3 * Y_2 * G_2$	$(4;7,0,0)$
$R_1 = N_4 * Q_4 * L_1 * E_4$	$(4;1,3,0)$
$R_2 = N_4 * Q_4 * L_4 * E_4$	$(4;1,3,0)$
$R_3 = N_4 * Q_1 * L_1 * E_4$	$(3;2,2,0)$
$R_4 = N_4 * Q_1 * L_4 * E_4$	$(3;2,2,0)$
$R_5 = N_4 * Q_1 * L_1 * E_0$	$(1;3,1,0)$
$R_6 = N_4 * Q_1 * L_4 * E_0$	$(1;3,1,0)$
$B_1 = K_4 * T_4 * W_4 * U_4$	$(5;2,0,2)$
$B_2 = K_4 * T_4 * W_1 * U_4$	$(4;3,0,1)$
$B_3 = K_4 * T_4 * W_1 * U_1$	$(3;4,0,0)$
$S_1 = H_2 * R_2 * B_1 * X_1$	$(5;4,0,0)$

Table 10.10. Some bottlenecks and improvement actions

Composite DA's	Bottleneck		Action (w/r)
	DA	Ins	
$H_1 = D_4 * F_3 * P_3 * M_3 * C_3 * A_3$	D_4		$2 \Rightarrow 1$
$H_1 = D_4 * F_3 * P_3 * M_3 * C_3 * A_3$	F_3		$2 \Rightarrow 1$
$H_1 = D_4 * F_3 * P_3 * M_3 * C_3 * A_3$	C_3		$2 \Rightarrow 1$
$H_2 = D_4 * F_3 * P_3 * M_3 * C_3 * A_4$	D_4		$2 \Rightarrow 1$
$H_2 = D_4 * F_3 * P_3 * M_3 * C_3 * A_4$	F_3		$2 \Rightarrow 1$
$H_2 = D_4 * F_3 * P_3 * M_3 * C_3 * A_4$	C_3		$2 \Rightarrow 1$
$H_3 = D_4 * F_3 * P_3 * M_3 * C_4 * A_3$	D_4		$2 \Rightarrow 1$
$H_3 = D_4 * F_3 * P_3 * M_3 * C_4 * A_3$	F_3		$2 \Rightarrow 1$
$H_3 = D_4 * F_3 * P_3 * M_3 * C_4 * A_3$	C_4		$2 \Rightarrow 1$
$H_4 = D_4 * F_3 * P_3 * M_3 * C_4 * A_4$	D_4		$2 \Rightarrow 1$
$H_4 = D_4 * F_3 * P_3 * M_3 * C_4 * A_4$	F_3		$2 \Rightarrow 1$
$H_4 = D_4 * F_3 * P_3 * M_3 * C_4 * A_4$	C_4		$2 \Rightarrow 1$
$H_5 = D_4 * F_3 * P_3 * M_3 * C_1 * A_3$	D_4		$2 \Rightarrow 1$
$H_5 = D_4 * F_3 * P_3 * M_3 * C_1 * A_3$	F_3		$2 \Rightarrow 1$
$H_6 = D_4 * F_3 * P_3 * M_3 * C_1 * A_4$	D_4		$2 \Rightarrow 1$
$H_6 = D_4 * F_3 * P_3 * M_3 * C_1 * A_4$	F_3		$2 \Rightarrow 1$
$R_1 = N_4 * Q_4 * L_1 * E_4$	Q_4		$2 \Rightarrow 1$
$R_1 = N_4 * Q_4 * L_1 * E_4$	L_1		$2 \Rightarrow 1$
$R_1 = N_4 * Q_4 * L_1 * E_4$	E_4		$2 \Rightarrow 1$
$R_2 = N_4 * Q_4 * L_4 * E_4$	L_4		$2 \Rightarrow 1$
$R_2 = N_4 * Q_4 * L_4 * E_4$	L_4		$2 \Rightarrow 1$
$R_2 = N_4 * Q_4 * L_4 * E_4$	L_4		$2 \Rightarrow 1$
$R_2 = N_4 * Q_4 * L_4 * E_4$		(N_4, Q_4)	$4 \Rightarrow 5$
$R_3 = N_4 * Q_1 * L_4 * E_4$	L_1		$2 \Rightarrow 1$
$R_3 = N_4 * Q_1 * L_1 * E_4$	E_4		$2 \Rightarrow 1$
$R_3 = N_4 * Q_1 * L_4 * E_4$		(Q_1, E_4)	$3 \Rightarrow 4$
$R_4 = N_4 * Q_1 * L_4 * E_4$	L_4		$2 \Rightarrow 1$
$R_4 = N_4 * Q_1 * L_4 * E_4$	E_4		$2 \Rightarrow 1$
$R_4 = N_4 * Q_1 * L_4 * E_4$		(Q_1, E_4)	$3 \Rightarrow 4$
$R_5 = N_4 * Q_1 * L_1 * E_0$	L_1		$2 \Rightarrow 1$
$R_5 = N_4 * Q_1 * L_1 * E_0$		(N_4, E_0)	$1 \Rightarrow 2$
$R_6 = N_4 * Q_1 * L_4 * E_0$	L_1		$2 \Rightarrow 1$
$B_1 = K_4 * T_4 * W_4 * U_4$	W_4		$3 \Rightarrow 2$
$B_1 = K_4 * T_4 * W_4 * U_4$	U_4		$3 \Rightarrow 2$
$B_2 = K_4 * T_4 * W_1 * U_4$	U_4		$3 \Rightarrow 2$
$B_2 = K_4 * T_4 * W_1 * U_4$		(K_4, W_1)	$4 \Rightarrow 5$
$B_3 = K_4 * T_4 * W_1 * U_1$		(T_4, U_1)	$3 \Rightarrow 4$
$X_1 = V_1 * O_1 * I_4 * J_4 * Z_3 * Y_2 * G_2$	J_4		$2 \Rightarrow 1$
$X_1 = V_1 * O_1 * I_4 * J_4 * Z_3 * Y_2 * G_2$	I_4		$3 \Rightarrow 2$

10.6 EXAMPLE OF STUDENT BUSINESS

Let us consider a simple example that may be used to introduce HMMD. The material is based on real student seminars. We consider a new business that consists of four basic components (Fig. 10.1).

Structure of Student Business

$S = T * I * P * M$

$S_1 = T_2 * I_1 * P_1 * M_1$	$S_1' = T_4' * I_1 * P_1 * M_1$
$S_2 = T_2 * I_1 * P_3 * M_1$	$S_2' = T_4' * I_1 * P_4 * M_1$

T Product	I Investment	P Manufacturing	M Market
$T_1(3/3)$	$I_1(1/1)$	$P_1(3/2)$	$M_1(1/1)$
$T_2(1/2)$	$I_2(2/3)$	$P_2(3/3)$	$M_2(2/2)$
$T_3(3/1)$	$I_3(3/2)$	$P_3(1/3)$	$M_3(1/1)$
$T_4'(/1)$	$I_4(3/2)$	$P_4(3/1)$	$M_4(2/3)$
			$M_5(3/3)$

Fig. 10.1. Structure of student business (priorities of DA's are shown in brackets)

DA's for the components, criteria, estimates are presented in Tables 10.11, 10.12, 10.13, 10.14. Compatibility is contained in Table 10.15. Resultant composite DA's are the following:

(1) $S_1 = T_2 * I_1 * P_1 * M_1$, $N(S_1) = (3; 3, 0, 1)$;

(2) $S_2 = T_2 * I_1 * P_3 * M_1$, $N(S_2) = (1; 4, 0, 0)$.

Fig. 10.2 illustrates corresponding Pareto-effective points. As a rule, the above-mentioned components, criteria, DA's, and estimates are discussed and generated with students.

Clearly, that all elements of this example may be changed (components, criteria, alternatives, and estimates). Let us consider forecasting of student business. Forecasting problems for decomposable systems are presented in Table 10.16. Here we add forecasting estimates to our example (see estimates after a letter "/") for weights of criteria, estimates, and priorities of DA's in Tables 10.11, 10.12, 10.13, 10.14, and Fig. 10.1). We use expert judgment for the following forecasting problems:

(a) *forecasting of set* for DA's of T;

(b) *forecasting of values* for weights of criteria; and

(c) *forecasting of values* for vector estimates of DA's.

Table 10.11. DA's, criteria and estimates for product T (weights of criteria are shown in brackets)

DA's	Criteria		
	F_{t1} (5/5) Demand	F_{t2} (−4/3) Cost	F_{t3} (3/5) Prospective
T_1 Chocolate	4/2	3/3	2/2
T_2 Database for students (job)	5/5	3/3	5/4
T_3 Communication center for interface with Internet	4/5	5/5	5/5
T_4' Consulting activity	/5	/2	/5

The same compatibility of DA's is applied, and compatibility of T_4' with other DA's equals 3. Resultant forecasting composite DA's are the following (Fig. 10.1, Fig. 10.2):

(1) $S_1' = T_4' * I_1 * P_1 * M_1$, $N(S_1') = (3; 3, 1, 0)$; and

(2) $S_2' = T_4' * I_1 * P_4 * M_1$, $N(S_2') = (1; 4, 0, 0)$.

Table 10.12. DA's, criteria and estimates for investment I (weights of criteria are shown in brackets)

DA's	Criteria		
	F_{i1} (4/5) Contact	F_{i2} (2/3) Possible volume	F_{i3} (−5/3) Responsibility
I_1 Self-investment	5/5	2/4	1/2
I_2 Parents	4/3	3/3	2/3
I_3 Bank	3/4	5/5	5/5
I_4 Firm	3/4	4/5	5/5

Table 10.13. DA's, criteria and estimates for manufacturing (place) P (weights of criteria are shown in brackets)

DA's	Criteria	
	F_{p1} (−3/2) Cost	F_{p2} (3/4) Resource
P_1 Apartment	1/2	1/2
P_2 University	2/2	2/1
P_3 Premises located in city	5/5	3/3
P_4 Premises located near city	3/3	4/4

Table 10.14. DA's, criteria and estimates for market M (weights of criteria are shown in brackets)

DA's	Criteria		
	F_{m1} (−4/4) Transport	F_{m2} (5/5) Volume	F_{m3} (4/5) Prospective
M_1 City	1/2	5/5	5/5
M_2 Near regions	1/2	4/4	3/4
M_3 Far regions	2/3	5/5	5/5
M_4 Near countries	2/3	4/4	4/4
M_5 Far countries	3/4	3/4	5/5

Table 10.15. Compatibility

	I_1	I_2	I_3	I_4	P_1	P_2	P_3	P_4	M_1	M_2	M_3	M_4	M_5
T_1	2	2	3	3	1	1	2	3	3	2	3	2	1
T_2	3	3	3	3	3	2	3	2	3	1	2	2	1
T_3	0	0	2	3	0	1	3	2	3	1	1	1	3
I_1					3	3	1	1	3	3	2	2	0
I_2					3	3	1	1	3	3	2	2	0
I_3					0	1	3	2	3	3	3	3	3
I_4					0	3	3	2	3	3	3	3	3
P_1									3	2	1	1	1
P_2									3	2	1	1	1
P_3									3	2	1	1	1
P_4									3	2	1	1	1

Table 10.16. Problems of forecasting for decomposable systems

Elements of description for decomposable systems	Objects under forecasting				
	Value	Function	Set	Order	Set & Order
1.Structural (tree-like) system model					*
2.DA's			*	*	
3.Criteria			*	*	
4.Vector estimates of DA's	*	*			
5.Priorities of DA's	*	*			
6.Complementability of DA's	*				
7.Factors of Ins			*	*	
8.Estimates of Ins	*	*			

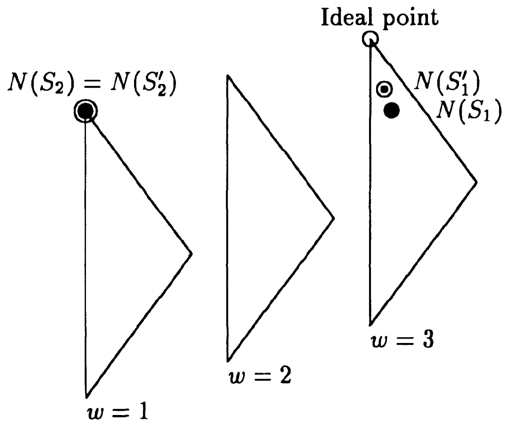

Fig. 10.2. Quality lattice and Pareto-effective points

10.7 PLANNING OF STUDENT CAREER

Individual decision making problems include various selection problems, e.g., car selection, house selection, school/university selection, private investment policy, etc. [474]. One of the most important decision making problems from the field of individual life is career planning ([162], [206], [445], [460], [527], etc.). Effectiveness of computer-assisted career guidance systems is analyzed in [446]. Often this problem may include various components which are under multicriteria selection ([162]. Here we consider a blend of several professional domains (computer science, management, and engineering, etc.) to combine a composite career plan for students [313].

Fig. 10.3 depicts the structure of our investigated fragment of the career plan. We consider a simple situation without resource constraints (e.g., financial, time, etc.), multicriteria evaluation and ranking of resultant composite decisions. The following solving stages are used:

(1) hierarchical design of career strategy;

(2) revelation of bottlenecks by DA's/Ins; and

(3) formation of improvement actions.

Criteria for leaves of examined structure are contained in Table 10.17. Table 10.18 contains estimates of DA's. Tables 10.19, 10.20, 10.21, 10.22, and 10.23 contain estimates of compatibility. The following factors for compatibility assessment are used:

(a) common theoretical fragments;

(b) causal relation;

(c) common application; and

(d) components may compose a wholeness.

Composite DA's are presented in Table 10.24, and bottlenecks and improving actions are presented in Table 10.25. The following designations for types of improvement actions are used:

(i) a generation of the ideal point (1);

(ii) an improvement of a Pareto-effective point (2); and

(iii) an extension of the Pareto-effective point set (3).

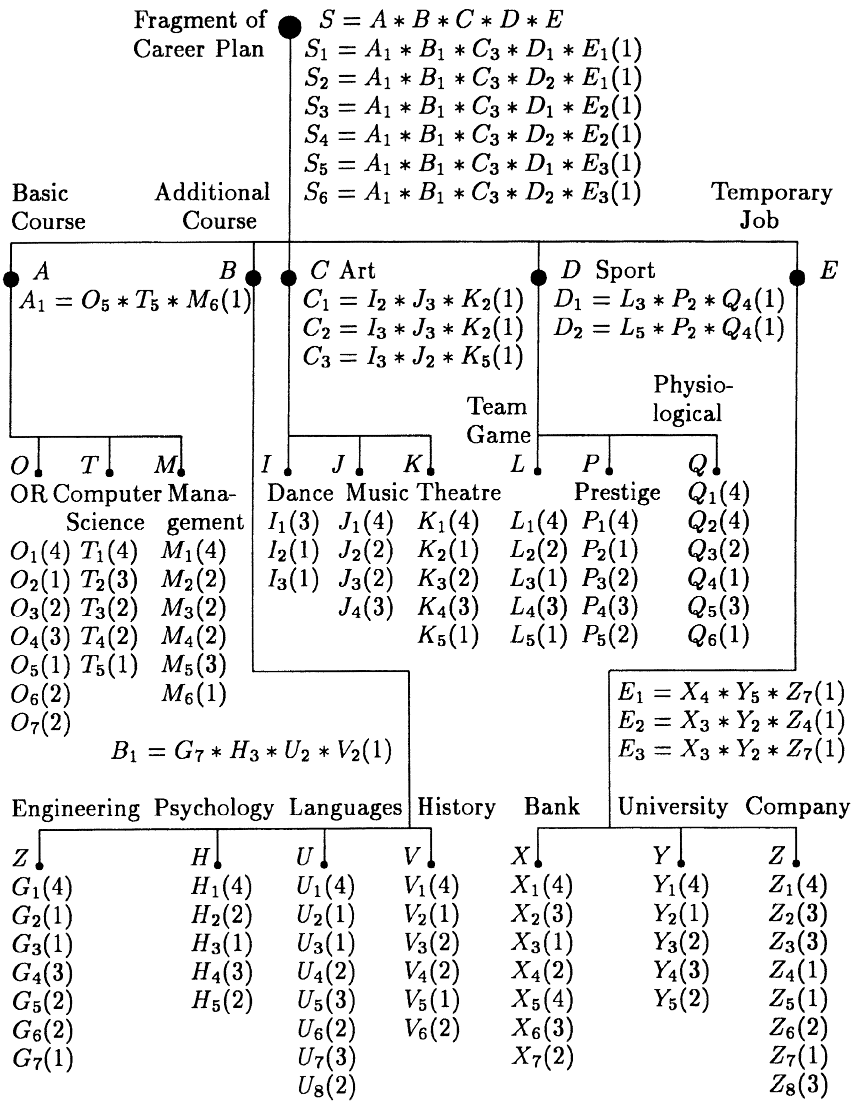

Fig. 10.3. Structure of career plan fragment (priorities of DA's are shown in brackets)

Table 10.17. Criteria for DA's

Criteria	Weights																				
	O	*T*	*M*	*G*	*H*	*U*	*V*	*I*	*J*	*K*	*L*	*P*	*Q*	*X*	*Y*	*Z*	*A*	*B*	*C*	*D*	*E*
1.Cost (-) / salary	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2.Possibility to meet useful persons														3	3	3	4	4	4	4	4
3.Possibility to meet friends	2	2	2	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2
4.Possibility to meet boy/girl friend	2	2	2	3	3	3	3	4	4	4		3		1	1	1	2	2	2	2	2
5.Accordance to inclinations	2	2	2	2	2	2	2	2	2	2	4	4	4	3	3	3	3	3	3	3	3
6.Usefulness for future career	3	3	3	4	4	4	4	2	2	2		2	2	4	4	4	4	4	4	4	4
7.Usefulness for health								2	2	2	4	4	4				3	3	3	3	3
8.Usefulness for for future life				1	1	1	1	3	3	3	3	3	3	4	4	4	2	2	2	2	2

Table 10.18. Estimates of DA's

DA's	Criteria							
	1	2	3	4	5	6	7	8
O_1 None	0		0	0	0	0		
O_2 Probability and statistics	1		1	1	4	4		
O_3 Decision analysis	1		1	1	2	4		
O_4 Simulation techniques	1		1	1	1	2		
O_5 MCDM	1		1	1	3	4		
O_6 Mathematical programming	1		1	1	3	2		
O_7 Networks&combinatorial optimization	1		1	1	4	3		
T_1 None	0		0	0	0	0		
T_2 Programming languages and OS	1		1	1	2	3		
T_3 Computation theory	1		1	1	2	5		
T_4 AI	1		1	1	2	4		
T_5 HCI	1		1	1	4	4		
M_1 None	0		0	0	0	0		
M_2 Production/operations management	1		1	1	3	3		
M_3 Quality assurance	1		1	1	3	3		
M_4 Investment	1		1	1	2	4		
M_5 Marketing	1		1	1	2	3		
M_6 Project management	1		1	1	4	4		
G_1 None	0		0	0	0	0		0
G_2 Chemical engineering	1		1	1	1	4		3
G_3 Biotechnology	1		1	1	1	5		1
G_4 Mechanical engineering	1		1	1	1	2		2
G_5 Electrical engineering	1		1	1	3	2		2
G_6 Civil engineering	1		1	1	2	2		3
G_7 Software engineering	1		1	1	4	3		1
H_1 None	0		0	0	0	0		0
H_2 Human engineering	1		1	1	4	3		1
H_3 Cognitive psychology	1		1	3	2	4		1
H_4 Activity theory	1		1	2	2	2		2
H_5 Empirical studies	1		1	3	2	2		3

Continuation of Table 10.18

DA's	Criteria							
	1	2	3	4	5	6	7	8
U_1 None	0		0	0	0	0		0
U_2 French	1		1	3	2	3		2
U_3 German	1		1	2	3	4		2
U_4 Spanish	1		1	2	2	2		2
U_5 Portuguese	2		1	2	1	2		1
U_6 Japanese	1		1	2	1	4		2
U_7 Chinese	2		1	2	1	2		1
U_8 Classical	2		1	2	3	3		2
V_1 None	0		0	0	0	0		0
V_2 Modern history and politics	1		1	1	3	2		3
V_3 Ancient history	1		1	2	2	1		1
V_4 History of Christianity	1		1	1	2	1		2
V_5 History of Judaism	1		1	2	3	1		2
V_6 History of Islam	1		1	0	1	1		1
I_1 None	0		0	0	0	0	0	0
I_2 Ball dance	2		1	2	2	2	2	2
I_3 Ensemble	1		1	3	1	1	2	2
J_1 None	0		0	0	0	0	0	0
J_2 Classic	2		1	1	3	2	2	2
J_3 Jazz	1		1	2	1	2	1	2
J_4 Singing	2		1	1	1	1	1	2
K_1 None	0		0	0	0	0	0	0
K_2 Actor	2		1	3	1	4	1	5
K_3 Producer	4		1	2	3	2	3	3
K_4 Technical staff	1		1	1	2	2	2	1
K_5 Author	0		1	1	3	3	3	3
L_1 None	0		0		0		0	0
L_2 Basket-ball	1		1		1		2	2
L_3 Football	1		1		2		2	2
L_4 Rowing	2		1		1		2	1
L_5 Volley-ball	1		1		2		3	2

Continuation of Table 10.18

DA's	Criteria							
	1	2	3	4	5	6	7	8
P_1 None	0		0	0	0	0	0	0
P_2 Tennis	1		1	2	2	2	2	3
P_3 Golf	3		1	1	1	4	3	3
P_4 Airplane	4		1	1	1	3	1	1
P_5 Maintain sky	3		2	2	2	1	2	2
Q_1 None	0		0		0	0	0	0
Q_2 Karate	2		1		1	2	3	3
Q_3 Box	2		1		1	2	1	3
Q_4 Jogging	1		2		3	2	3	2
Q_5 Sky	2		1		3	1	3	1
Q_6 Gymnastics	2		1		1	2	2	2
X_1 None	0	0	0	0	0	0		0
X_2 Operational job	1	3	1	2	1	3		2
X_3 Capital budgeting	1	4	1	2	2	4		4
X_4 Insurance	1	3	1	2	1	3		3
X_5 Data Base	2	1	1	1	1	1		1
X_6 Computer Communication	2	1	1	1	2	1		1
X_7 Modeling	2	1	2	1	3	4		0
Y_1 None	0	0	0	0	0	0		0
Y_2 Modeling	1	2	1	1	3	3		1
Y_3 Software development	2	1	1	1	2	2		1
Y_4 Physical experiments	2	1	1	1	1	1		1
Y_5 Psychological research	1	1	1	3	2	2		2
Z_1 None	0	0	0	0	0	0		0
Z_2 Manufacturing	2	1	1	1	1	2		1
Z_3 Nonmanufacturing	2	1	1	1	1	1		1
Z_4 R & D	1	3	2	1	4	3		1
Z_5 Operations management	1	3	2	1	3	3		1
Z_6 Human engineering	1	1	1	2	2	2		3
Z_7 Marketing	1	3	1	2	1	3		3
Z_8 Maintenance	1	1	1	1	1	1		2

Table 10.19. Compatibility of DA's

	T_1	T_2	T_3	T_4	T_5	M_1	M_2	M_3	M_4	M_5	M_6
O_1	2	2	2	2	2	2	2	2	2	2	2
O_2	2	3	4	4	3	2	3	5	5	5	4
O_3	2	3	3	4	4	2	3	4	5	4	4
O_4	2	5	3	3	5	2	5	3	4	3	4
O_5	2	3	3	5	4	2	4	4	4	4	4
O_6	2	3	4	3	3	2	5	3	5	3	4
O_7	1	2	5	4	3	2	5	3	4	4	4
T_1						2	2	2	2	2	2
T_2						2	4	3	3	3	3
T_3						2	4	3	4	3	3
T_4						2	5	4	4	5	5
T_5						2	4	3	4	4	4

Table 10.20. Compatibility of DA's

	Y_1	Y_2	Y_3	Y_4	Y_5	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8
X_1	2	2	2	2	2	2	2	2	2	2	2	2	2
X_2	2	3	3	3	5	2	3	3	3	3	4	4	3
X_3	2	5	3	3	3	2	4	4	4	3	3	5	3
X_4	2	5	3	3	5	2	4	4	3	3	4	5	5
X_5	2	3	5	3	3	2	4	4	3	4	3	4	3
X_6	2	3	5	3	3	2	4	4	3	4	3	4	3
X_7	2	3	3	3	3	2	4	4	4	4	3	4	3
Y_1						2	2	2	2	2	2	2	2
Y_2						2	5	5	5	5	4	4	3
Y_3						2	3	3	4	5	3	3	3
Y_4						2	4	3	3	3	3	3	5
Y_5						2	3	4	3	3	5	5	4

Table 10.21. Compatibility of DA's

	P_1	P_2	P_3	P_4	P_5	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
L_1	2	2	2	2	2	2	2	2	2	2	2
L_2	2	3	3	3	3	2	4	4	3	2	2
L_3	2	3	3	3	3	2	3	3	4	4	3
L_4	2	3	3	3	3	2	3	3	3	3	3
L_5	2	3	3	3	4	2	4	4	3	2	3
P_1						2	2	2	2	2	2
P_2						2	4	3	4	3	4
P_3						2	3	3	3	3	3
P_4						2	3	3	3	3	3
P_5						2	3	3	4	3	4

Table 10.22. Compatibility of DA's

	H_1	H_2	H_3	H_4	H_5	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	V_1	V_2	V_3	V_4	V_5	V_6
G_1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
G_2	2	4	2	2	3	2	4	5	3	3	5	4	3	2	4	3	4	3	3
G_3	2	4	2	2	3	2	3	5	3	3	5	3	3	2	5	3	3	3	3
G_4	2	4	2	2	3	2	3	5	3	3	5	3	3	2	4	4	3	3	3
G_5	2	3	2	2	3	2	3	3	3	3	5	3	3	2	5	3	3	3	3
G_6	2	5	2	2	3	2	4	3	3	4	4	3	4	2	4	4	4	3	4
G_7	2	5	4	4	3	2	4	4	3	3	4	3	4	2	5	3	3	4	3
H_1						2	2	2	2	2	2	2	2	2	2	2	2	2	2
H_2						2	3	4	3	3	5	3	3	2	5	3	3	3	3
H_3						2	5	5	3	3	4	4	5	2	4	4	4	4	4
H_4						2	3	3	3	3	4	3	3	2	4	3	4	4	3
H_5						2	4	4	3	3	4	3	3	2	5	3	4	3	3
U_1														2	2	2	2	2	2
U_2														2	5	3	5	2	2
U_3														2	5	3	5	2	2
U_4														2	5	3	5	2	2
U_5														2	4	3	5	2	2
U_6														2	5	1	1	2	1
U_7														2	4	1	1	1	1
U_8														2	2	5	5	5	5

Table 10.23. Compatibility of DA's

	B_4	C_3	C_3	C_4	D_4	D_4	E_5	E_5	E_5
A_1	2	3	3	3	3	2	3	3	3
B_1		3	3	5	4	4	5	4	5
C_1					4	4	5	4	5
C_2					5	5	5	4	5
C_3					5	5	5	5	5
D_1							5	5	5
D_2							5	5	5

Table 10.24. Composite DA's

DA's	N
$A_1 = O_5 * T_5 * M_6$	$(4;3,0,0,0)$
$B_1 = G_7 * H_3 * U_2 * V_2$	$(4;4,0,0,0)$
$C_1 = I_2 * J_3 * K_2$	$(4;2,1,0,0)$
$C_2 = I_3 * J_3 * K_2$	$(4;2,1,0,0)$
$C_3 = I_3 * J_2 * K_5$	$(4;2,1,0,0)$
$D_1 = L_3 * P_2 * Q_4$	$(3;3,0,0,0)$
$D_2 = L_5 * P_2 * Q_4$	$(3;3,0,0,0)$
$E_1 = X_4 * Y_5 * Z_7$	$(5;1,2,0,0)$
$E_2 = X_3 * Y_2 * Z_4$	$(4;3,0,0,0)$
$E_3 = X_3 * Y_2 * Z_7$	$(4;3,0,0,0)$
$S_1 = A_1 * B_1 * C_3 * D_1 * E_1$	$(4;5,0,0,0)$
$S_2 = A_1 * B_1 * C_3 * D_2 * E_1$	$(4;5,0,0,0)$
$S_3 = A_1 * B_1 * C_3 * D_1 * E_2$	$(4;5,0,0,0)$
$S_4 = A_1 * B_1 * C_3 * D_2 * E_2$	$(4;5,0,0,0)$
$S_5 = A_1 * B_1 * C_3 * D_1 * E_3$	$(4;5,0,0,0)$
$S_6 = A_1 * B_1 * C_3 * D_2 * E_3$	$(4;5,0,0,0)$

Table 10.25. Some bottlenecks and improvement actions

Composite DA's	Bottleneck		Action	
	DA	Ins	(w/r)	Type
$O_2 * T_5 * M_6$		(T_5, O_2)	$3 \Rightarrow 4$	3
$O_4 * T_4 * M_6$	T_4		$2 \Rightarrow 1$	1
$G_2 * H_3 * U_2 * V_2$		(H_3, G_2)	$2 \Rightarrow 4(5)$	3
$G_2 * H_3 * U_3 * V_2$		(H_3, G_2)	$2 \Rightarrow 4(5)$	3
$C_1 = I_2 * J_3 * K_2$	J_3		$2 \Rightarrow 1$	2
$C_2 = I_3 * J_3 * K_2$	J_3		$2 \Rightarrow 1$	2
$C_3 = I_3 * J_2 * K_5$	K_5		$2 \Rightarrow 1$	2
$I_2 * J_2 * K_2$		(J_2, K_2)	$3 \Rightarrow 5$	3
$D_2 = L_3 * P_2 * Q_4$		(L_3, P_2)	$3 \Rightarrow 4$	2
$X_4 * Y_5 * Z_4$	X_4		$2 \Rightarrow 1$	3
$X_4 * Y_5 * Z_4$	Y_5		$2 \Rightarrow 1$	3
$E_2 = X_3 * Y_2 * Z_4$		(X_3, Z_4)	$4 \Rightarrow 5$	1
$E_3 = X_3 * Y_2 * Z_7$		(X_2, Z_7)	$4 \Rightarrow 5$	1
$X_3 * Y_2 * Z_5$		(X_3, Z_5)	$3 \Rightarrow 4$	3

Note that the tutoring strategy may be based on optimization of specialist's profiles [44]. In this case, target profiles of required specialists are represented as a hierarchical graph in which nodes correspond to disciplines, and arcs of two kinds correspond to the following: (a) inclusion; and (b) relations between disciplines or their parts. Thus we can consider the following series problems:

(1) design of the required target profile(s);

(2) description of initial student profiles;

(3) evaluation of proximity between target profile(s) and the profile(s) of students; and

(4) design of a multistage transformation of initial student(s) profile(s) into target profile(s).

10.8 SUMMARY

In this section, we have proposed our viewpoints to intellectual activities, and basic problems of educational management. Problems above are a basis to extend old or to develop new software for educational applications.

11 ADDITIONAL APPLIED PROBLEMS

This chapter consists of various applied problems that are solved on the basis of HMMD. The battery of the examples demonstrates possible applications in computer science, information engineering, mechanical engineering, finance, forecasting, etc.

11.1 DESIGN OF USER INTERFACE

In this section, we examine hierarchical design of user interfaces on the basis of HMMD [297]. In recent years, special User Interface Development Environments (UIDE) and User Interface Management Systems (UIMS) are considered ([32], [33], [210] [231], [279], etc.). Hix investigated four generations of UIMSs [210]. Usually functional/task analysis and models of human computer interaction (HCI) are used as the main approaches to user interface design ([231], [348], etc.). The list of basic HCI models includes the following ([231] and [348]):

(1) Command Language Grammar (CLG);
(2) Task Action Language (TAL);
(3) Task Analysis for Knowledge Description (TAKD);

(4) GOMS approach (Goals, Operators, Methods and Selection Models); and
(5) Task Mapping Model (TMM); etc.

Traditional approaches to interface design are oriented to the analysis, evaluation and selection of design alternatives ([32], [231], [279], etc.). Mainly, comparison and selection of various versions of interface components (e.g., icon, text, direct manipulation, menu, etc.) is based on techniques as follows: (a) experimental investigation [37]; (b) rules ([32], [52], [279], etc.); and (c) Analytic Hierarchical Process (AHP) [438].

In recent years, object oriented development technique is used for user interface design too ([56], [381], etc.). Fisher reviews some investigations in optimal design of human interfaces including combinatorial optimization problems [151]. The use of object-oriented development, TMM, and optimal design approaches point out that there is a movement to an implementation of synthesis techniques in human interface design.

This section addresses a composition design of an user interface fragment consisting of presentation elements. Investigated hypothetical example is based on the interface of DSS COMBI for multicriteria ranking ([294], [317]) and may be considered as *a posteriori* analysis.

11.1.1 Fragment of User Interface

We study the only one basic flow of data in DSS COMBI, when a basic element of information is preference relation or matrix, and problem solving processes are represented as series transformations of data D_v (see sections 1.6.2, 3.6): (1) alternatives, criteria, multicriteria estimates of alternatives upon criteria (D_0); (2) preference relation of alternatives (D_1); (3) intermediate linear ordering of alternatives (D_2); (4) intermediate group ordering of alternatives (D_3); and (5) result data (ranking of alternatives, e.g., result group ordering, fuzzy group ordering) (D_4).

The list of functional operations realizes processing of the following kind:
$D_v \rightarrow D_j$, when $v = 0, ..., 3$; $j = 1, ..., 4$; $j > v$.

Data edition operations are: (a) input, (b) correction, (c) presentation, (d) output, and (e) import/export.

DSS COMBI involves the following main modes: (a) choice or creation of applied problems; (b) problem solving (multicriteria ranking) on the basis of various techniques; (c) result's analysis; (d) help (learning, training); (e) data import/export; and (f) quit. Thus we consider the following components of the user interface fragment: (1) management of system's modes; (2) planning / management of solving processes; (3) operations of data processing; and (4) edition and presentation of data.

11.1.2 Composing of User Interface

Let consider the steps of user interface design and analysis on the basis of HMMD. Here we do not use constraints. Our model of user interface fragment is depicted in Fig. 11.1. Notations of leaf nodes are the following:

Y is a presentation of main modes (solving, help, quit, etc.);

Z is a basic colour composition for the presentation of main modes (screen background, background for presentation element, text, and frame);

J is a presentation of functional schemes for data processing;

L is a presentation of functional operations for data processing ($D_v \rightarrow D_j$);

H is a presentation of connections among functional operations;

K is an acoustic effect (sound) for operational part;

X is a colour composition for presentation of operational part (background, text, and frame);

E is a presentation of data scheme;

R is a presentation of data (numbers, table, flowchart, etc.);

Q is a presentation of data edition operations (input, correction, etc.);

B is a sound for factual part; and

C is a colour composition for the presentation of factual part (background, text, and frame).

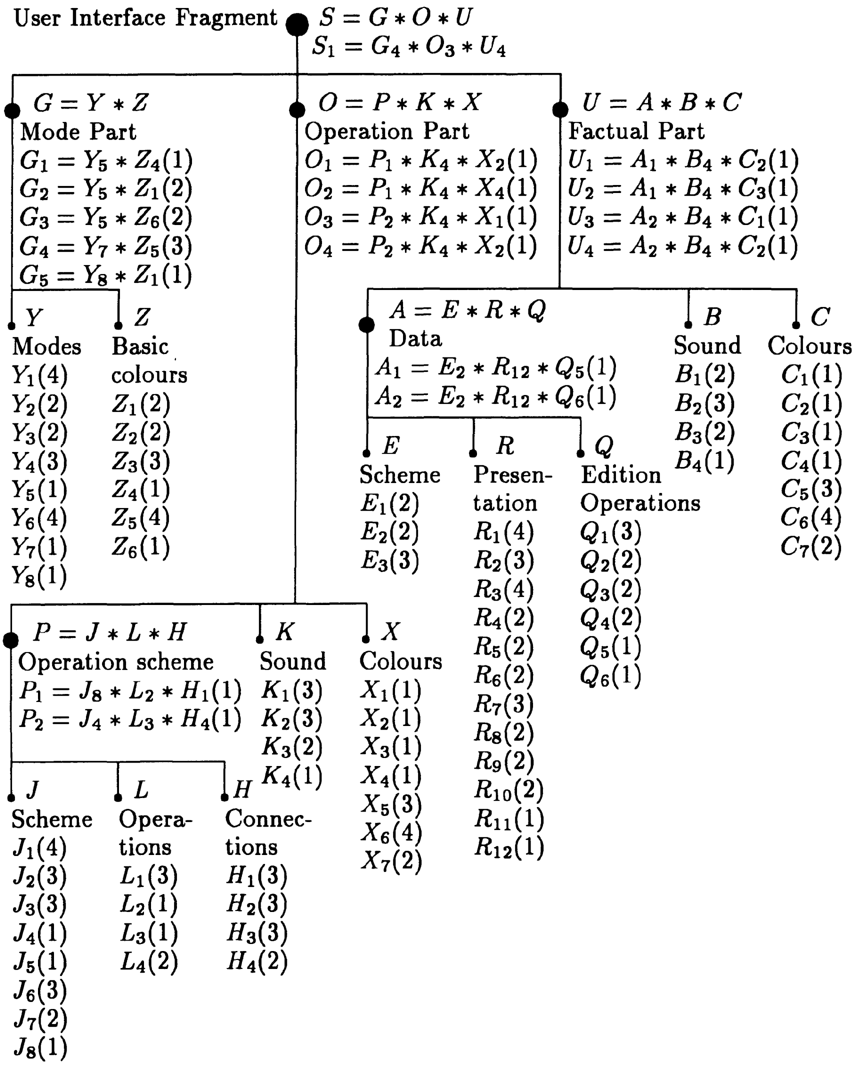

Fig. 11.1. Hierarchical scheme of user interface fragment priorities of DA's are shown in brackets)

Table 11.1. Criteria and their weights

Criteria	Weights											
	Y	J	L	H	E	R	Q	Z	X	C	K	B
1.Tradition (habits)	5	3	5	5	5	3	2					
2.Volume of information	1	6	4	2	4		4					
3.Complexity of development (negative)	3	3	3	4	4	2	3				2	2
4.Usability	6	6	6	5	5	6	5	6	6	6		
5.Easy of use	6	6	6	5	5	6	5					
6.Possibility for extension	4	2	2	3	5	5	3					
7.Learnability	2	6	6	4	6	6	4				3	3
8.Acceptability								5	5	5	5	5
9.Efficiency											3	3

Table 11.2. DA's and their estimates (part 1)

DA's	Criteria						
	1	2	3	4	5	6	7
Y_1 Command language	4	6	5	2	2	5	2
Y_2 Vertical menu	4	6	2	4	4	3	4
Y_3 Horizontal menu	4	6	2	4	4	3	4
Y_4 Flowchart	2	3	4	6	5	2	6
Y_5 Icon	3	4	3	5	6	3	6
Y_6 Matrix menu	1	5	3	4	4	2	4
Y_7 Pop-Up menu	4	6	5	5	4	5	3
Y_8 Catalogue menu	3	6	4	5	4	5	4
J_1 Command language	4	6	5	2	2	5	2
J_2 Vertical menu	4	4	2	4	4	3	4
J_3 Horizontal menu	4	4	2	4	4	3	4
J_4 Flowchart	3	4	3	5	6	2	6
J_5 Icon	3	4	3	5	6	3	6
J_6 Matrix menu	1	5	3	4	4	2	4
J_7 Pop-Up menu	3	6	5	5	4	5	3
J_8 Catalogue menu	3	6	4	5	4	5	4
L_1 Command language	4	6	5	2	2	5	2
L_2 Icon	3	3	3	5	6	3	6
L_3 Elements of flowchart	2	3	3	5	6	2	6
L_4 Elements of menu	3	6	5	5	4	3	3
H_1 Table (matrix)	4	5	4	3	2	4	3
H_2 Element (pair)	3	6	2	2	3	5	2
H_3 List of elements	2	6	3	3	3	5	2
H_4 Graph	3	3	5	5	4	3	6

Table 11.3. DA's and their estimates (part 2)

DA's	Criteria						
	1	2	3	4	5	6	7
E_1 Table of connections	4	3	3	2	2	2	2
E_2 Flowchart	3	3	4	5	5	2	6
E_3 Catalogue	3	6	5	5	4	5	3
R_1 Numbers	3		1	1	3	1	4
R_2 Table (matrix)	3		2	3	3	2	4
R_3 Preference graph	2		5	4	2	3	3
R_4 Vertical bar chart	3		3	3	4	3	4
R_5 Horizontal bar chart	3		3	3	4	3	4
R_6 Pie chart	3		3	3	4	3	4
R_7 Star chart	2		3	4	4	2	5
R_8 Graphical presentation of layers (e.g., Pareto-effective points	2		4	5	4	4	5
R_9 Animation	2		6	5	3	3	6
R_{10} Table & bar chart	3		4	5	4	3	4
R_{11} Table & bar chart & pie chart	3		5	6	4	3	4
R_{12} All versions	3		8	8	5	4	2
Q_1 Command language	4	6	5	2	2	5	2
Q_2 Vertical menu	4	5	2	4	4	3	4
Q_3 Horizontal menu	4	5	2	4	4	3	4
Q_4 Matrix menu	2	6	3	4	4	3	5
Q_5 List of icons	3	4	3	5	6	3	6
Q_6 Pop-Up menu	3	6	4	5	4	5	3

Table 11.4. DA's and their estimates (part 3)

DA's	Criteria				
	3	4	7	8	9
Z_1 Black-blue-white-white		6		5	
Z_2 White-blue-yellow-black		5		2	
Z_3 Cyan-blue-white-black		3		6	
Z_4 Cyan-blue-cyan-black		4		4	
Z_5 Cyan-white-blue-black		5		5	
Z_6 Blue-white-black-black		6		4	
$X_1(C_1)$ Green-black-black		5		4	
$X_2(C_2)$ Blue-yellow-black		5		4	
$X_3(C_3)$ Blue-white-black		6		4	
$X_4(C_4)$ Blue-cyan-black		6		4	
$X_5(C_5)$ Blue-cyan-white		4		3	
$X_6(C_6)$ White-blue-black		3		2	
$X_7(C_7)$ Grey-black-black		4		4	
K_1 None	0		0	3	0
K_2 Start	1		1	1	2
K_3 End	1		2	1	2
K_4 Process	3		2	2	1
B_1 None	0		0	3	0
B_2 Start	1		1	1	1
B_3 End	1		2	1	1
B_4 Process	3		2	2	1

Table 11.5. Compatibility

	L_1	L_2	L_3	L_4	H_1	H_2	H_3	H_4
J_1	5	0	0	0	1	2	2	1
J_2	0	0	0	5	1	2	2	1
J_3	0	0	0	5	1	2	2	1
J_4	1	1	5	1	3	2	2	4
J_5	0	5	0	0	3	2	2	3
J_6	1	2	0	4	4	2	2	2
J_7	1	2	0	5	0	1	1	3
J_8	2	5	0	4	5	2	2	2
L_1					4	3	3	2
L_2					5	3	3	4
L_3					3	2	2	5
L_4					4	3	3	3

Table 11.6. Compatibility

	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8
Z_1	3	1	1	3	4	3	3	4
Z_2	2	1	1	2	4	3	2	2
Z_3	2	4	4	4	4	4	4	3
Z_4	2	4	4	4	5	3	3	3
Z_5	2	4	4	4	3	4	3	3
Z_6	3	2	2	3	4	3	3	3

Usually the following main criteria for interface evaluation are considered: acceptability; usability; learnability; efficiency; and ease of use ([231], [381], etc.). We apply similar criteria (Table 11.1).

DA's and their estimates on criteria for leaf nodes are shown in Tables 11.2, 11.3, 11.4. Priorities of DA's are shown in Fig. 11.1 (in brackets).

Tables 11.5, 11.6, 11.7, 11.8, 11.9, and 11.10 contain compatibility of DA's. Composite DA's are presented in Table 11.11. Table 11.12 contains bottlenecks and improvement actions.

Table 11.7. Compatibility

	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	E_1	E_2	E_3
R_1	3	3	3	4	4	4	0	0	2
R_2	3	3	3	5	4	4	3	5	1
R_3	3	3	3	4	4	4	2	5	2
R_4	3	4	2	4	3	2	4	4	4
R_5	3	2	4	4	3	2	4	4	4
R_6	3	3	4	4	4	3	4	4	4
R_7	3	3	4	4	4	3	4	4	4
R_8	3	3	4	4	4	3	4	4	4
R_9	3	3	4	3	4	4	4	4	4
R_{10}	3	4	4	5	4	4	4	5	4
R_{11}	3	4	4	5	4	4	4	5	4
R_{12}	5	5	5	5	5	5	5	5	5
Q_1							3	4	3
Q_2							4	3	5
Q_3							4	4	5
Q_4							3	5	3
Q_5							4	5	3
Q_6							4	5	3

Table 11.8. Compatibility

	K_1	K_2	K_3	K_4	P_1	P_2
X_1	5	5	5	5	4	5
X_2	5	5	5	5	5	5
X_3	5	5	5	5	4	4
X_4	5	5	5	5	5	4
X_5	5	5	5	5	4	4
X_6	5	5	5	5	4	4
X_7	5	5	5	5	4	4
K_1					5	5
...					...	...
K_4					5	5

Table 11.9. Compatibility

	O_1	O_2	O_3	O_4	U_1	U_2	U_3	U_4
G_1	1	3	5	1	1	3	5	1
G_2	2	3	5	2	2	2	5	2
G_3	2	4	5	2	2	2	5	2
G_4	5	3	5	5	5	3	5	5
G_5	2	4	5	2	2	2	5	2
O_1					0	2	5	0
O_2					2	0	5	2
O_3					5	5	0	5
O_4					0	2	5	0

Table 11.10. Compatibility

	B_1	B_2	B_3	B_4	A_1	A_2
C_1	5	5	5	5	4	5
C_2	5	5	5	5	5	5
C_3	5	5	5	5	5	4
C_4	5	5	5	5	4	4
...	...	...	...	...	...	...
C_7	5	5	5	5	4	4
B_1					5	5
...					...	...
B_4					5	5

Table 11.11. Composite DA's

DA's	N	DA's	N
$G_1 = Y_5 * Z_4$	$5;1,1,0,0$	$A_1 = E_2 * R_{12} * Q_5$	$5;2,1,0,0$
$G_2 = Y_5 * Z_1$	$4;2,0,0,0$	$A_2 = E_2 * R_{12} * Q_6$	$5;2,1,0,0$
$G_3 = Y_5 * Z_6$	$4;2,0,0,0$	$U_1 = A_1 * B_4 * C_2$	$5;3,0,0,0$
$G_4 = Y_7 * Z_5$	$4;2,0,0,0$	$U_2 = A_1 * B_4 * C_3$	$5;3,0,0,0$
$G_5 = Y_8 * Z_1$	$4;2,0,0,0$	$U_3 = A_2 * B_4 * C_1$	$5;3,0,0,0$
$P_1 = J_8 * L_2 * H_1$	$5;2,0,1,0$	$U_4 = A_2 * B_4 * C_2$	$5;3,0,0,0$
$P_2 = J_4 * L_3 * H_4$	$4;2,1,0,0$	$S_1 = G_4 * O_3 * U_4$	$5;3,0,0,0$
$O_1 = P_1 * K_4 * X_2$	$5;3,0,0,0$		
$O_2 = P_1 * K_4 * X_4$	$5;3,0,0,0$		
$O_3 = P_2 * K_4 * X_1$	$5;3,0,0,0$		
$O_4 = P_2 * K_4 * X_2$	$5;3,0,0,0$		

Table 11.12. Bottlenecks and improvements

Composite DA's	Bottlenecks		Action	
	DA's	Ins	w/r	Type
$G_1 = Y_5 * Z_4$	Z_4		$2 \Rightarrow 1$	1
$G_2 = Y_5 * Z_1$		(Y_5, Z_1)	$4 \Rightarrow 5$	1
$G_3 = Y_5 * Z_6$		(Y_5, Z_6)	$4 \Rightarrow 5$	1
$G_4 = Y_7 * Z_5$		(Y_7, Z_5)	$4 \Rightarrow 5$	1
$G_5 = Y_8 * Z_1$		(Y_8, Z_1)	$4 \Rightarrow 5$	1
$Y_2 * Z_5$	Y_2		$2 \Rightarrow 1$	3
$Y_3 * Z_5$	Y_3		$2 \Rightarrow 1$	3
$Y_5 * Z_3$	Z_3		$2 \Rightarrow 1$	3
$Y_7 * Z_3$	Z_3		$2 \Rightarrow 1$	3
$P_1 = J_8 * L_2 * H_1$	H_1		$3 \Rightarrow 1$	1
$P_2 = J_4 * L_3 * H_4$	H_4		$2 \Rightarrow 1$	2
$P_2 = J_4 * L_3 * H_4$		(J_4, H_4)	$4 \Rightarrow 5$	2
$A_1 = E_2 * R_{12} * Q_5$	E_2		$2 \Rightarrow 1$	1
$A_2 = E_2 * R_{12} * Q_6$	E_3		$2 \Rightarrow 1$	1

11.1.3 Comparison of System Versions

Our example of the human interface design corresponds to real development stages of DSS COMBI (Table 11.13).

The development of the first package was oriented to construction of DSS that consists of various techniques for multicriteria ranking. At the second stage of system development, flowchart of solving processes as functional graph menu, including both operations and data, was realized [294]. This effort was based on an attempt to improve the interface by diagram with direct manipulation for operations and data ([278], [465], etc.).

Table 11.13. Versions of DSS

Version	Modes	Operations	Data presentation	Data edition
A version for minicomputer (1987)	Command language Y_1	Command language $J_1\&L_1\&H_3$	Numbers $E_1\&R_1$	Command language Q_1
The 1st version for PC (1989)	Menu & catalogue $Y_2\&Y_8$	Flowchart of solving scheme $J_4\&L_3\&H_4$	Table $E_2\&R_2$	Matrix menu Q_4
The 2nd version for PC (1991)	Pop-Up menu Y_1	Pop-Up menu $J_7\&L_4\&H_4$	Table & bar, pie charts $E_2\&R_{11}$	Pop-Up menu Q_6

However, utilization experience of the package shown that the complete morphological solving scheme presented in the screen was often difficult for many users. As a result, the next step of DSS COMBI development included the following:

(a) operational part was implemented at the easier level; and

(b) various bar & pie charts for data presentation were added.

An evolution of user interfaces for the DSS COMBI versions is illustrated in Fig. 11.2.

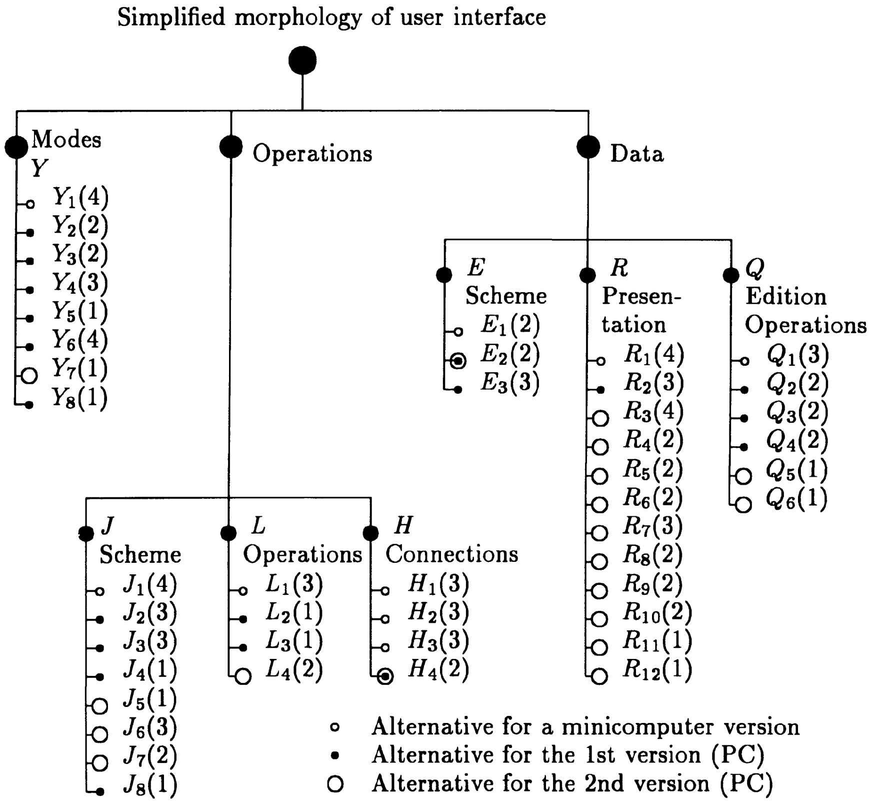

Fig. 11.2. Evolution of user interface for DSS

11.2 SERIES-PARALLEL STRATEGY FOR USER INTERFACE

Problem solving methodology has been examined by a number of well known authors ([380], [382], [471], etc.). The problem decomposition is perhaps the most important approach towards complex situations in various domains [380]. As a result, we face problems of composing the models. In the last decade, various investigations in the field of model management have been conducted ([19], [121], [185], [375], etc.). The investigations address the development and use of model management systems as model bases or libraries for application domains as follows: (a) databases; (b) decision support systems; and (c) expert systems ([375], etc.).

This section focuses on composing a solution strategy from basic components as follows [298]: (i) operations of data processing, (ii) operations of knowledge acquisition or transformation, and (iii) training of user and data/knowledge representation. Our design scheme consists of the following:

1. Forming a basic hierarchical morphological space of operations (HMSO).

2. Analysis of an initial situation (user, task) and adaptation of HMSO (design of a working version of HMSO) as follows:

(a) selection or identification of appropriate operations on the basis of constraints;

(b) parallelization of operations on the basis of parametrization of techniques and use of different experts; and

(c) training the user;

3. Design of a composite solving strategy including the following two phases:

(a) selection of operations (multicriteria ranking);

(b) synthesis on the basis of steps as follows:

(i) composing a series strategy or constructing an operation chain (morphological clique);

(ii) composing parallel strategies (knapsack-like and/or clique-like problems); and

(iii) composing parallel-series strategies (step-by-step use of parallel-series synthesis above).

Here we examine designing the series, parallel, and parallel-series strategies for supporting a dialogue in the development and use of knowledge based systems. Our example demonstrates composition of a parallel-series strategy.

11.2.1 Solving Scheme

There exist sources of an user interface adaptation as follows:

(a) user;

(b) task; and

(c) intermediate results of problem solving (information for feedback).

Table 11.14 contains adapted objects and some basic operations. An adaptation process is depicted in Fig. 11.3. The requirements are based on the following:

(1) kinds of tasks;

(2) types of users;

(3) available resources (e.g., human, computer, time); and

(4) features of a decision situation (e.g., uncertainty, required precision and robustness of results).

Studies of the user modeling and adaptation are considered in detail in ([108], [255], etc.). In the main, the following techniques have been applied for the model selection and sequencing (synthesis of series or parallel-series strategies):

(a) artificial intelligence techniques ([121], [144], [323], etc.);

(b) linear programming problems [375];

(c) integer programming problems as searching for an optimal path to a required output [185]; and

(d) nonlinear integer programming for the modular design with redundancy, i.e., with parallel fragments [41].

Here we examine conceptual design, and combinatorial models for the selection, modification, and composition of the solution strategies and their elements. Table 11.15 contains the relationship of the basic actions and stages of our strategy design. The stage of HMSO generation is based on the conceptual analysis and design ([163], [429]).

Table 11.14. Adapted objects and operations

Adapted object	Operations of adaptation
1.User	Training
2.Task	Modification
3.Hierarchical morphological space of operations HMSO:	
(1) structure of HMSO	Modification
(2) operations	Selection, modification, forming new ones
4.User interface	Selection of modes and presentation elements
5.Intermediate results	Modification of task, user, HMSO, interface

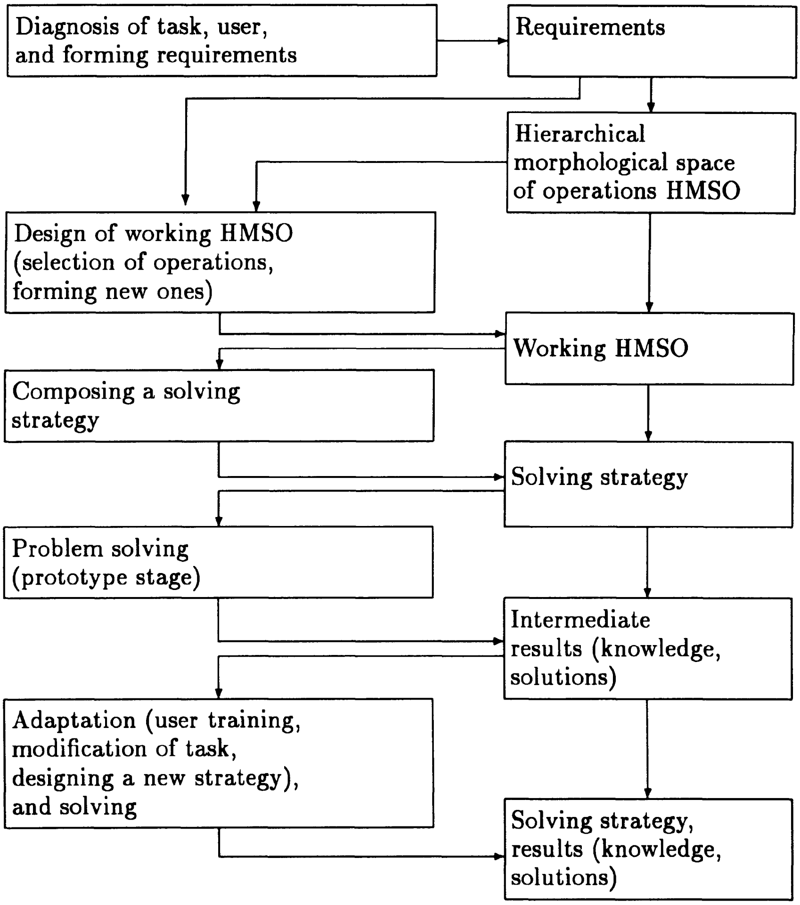

Fig. 11.3. Scheme of adaptation

Table 11.15. Stages of strategy design and actions

Processing	Actions (models, techniques)
1.Generation of HMSO	Conceptual design
2.Selection of operations	Multicriteria ranking
3.Modification of operations and their parallelization	Use of several experts, parametrization of techniques
4.Synthesis of a strategy	
(a) composing a series strategy	Morphological clique
(b) composing a parallel strategy	Knapsack-like and/or clique-like (or "maximum independent set") problem; conceptual design
(c) composing a parallel-series strategy	Step-by-step use of parallel-series synthesis

11.2.2 Example

Our example is based on the examination of a system for the knowledge acquisition and use. We examine HMSO as follows:

1. Prototyping P:

1.1 Problem identification I.

1.2 Selection/generation of terminology T.

1.3 Knowledge acquisition V.

2. Design of work version D.

2.1 Enhancement of problem description (type, terminology) E.

2.2 Acquisition of additional knowledge W.

3. Utilization U.

Generally, HMSO is based on the following kinds of operations: (1) knowledge acquisition (interaction); (2) data/knowledge processing (computation); (3) data/knowledge representation (interaction); and (4) training of user (interaction).

The following criteria for the evaluation of DA's are used ([185], [297], [381], etc.): required computer resources; required human resources; quality of ranking (robustness, etc.) possibility for data representation; possibility for an analysis of intermediate data; and usability (easy to learn, understanding, acceptability, habits, etc.).

In our example, we apply representation elements or design alternatives (DA's) for leaf nodes as follows: command language (X_1); menu (X_2); icons (X_3); and graphic / animation (X_4). Here the index corresponds to the number of DA's, and instead of X we use a certain notation of the leaf (e.g., T_1, E_3). Also, we consider the following working HMSO:

I_1, I_2; T_1, T_2; V_1, V_2; E_1, E_2; W_1, W_2; U_1, U_2, U_4.

In addition, we will use the following parallel aggregate (parallel combined) DA's:

$I_5 = I_1\&I_2$; $T_5 = T_1\&T_2$; $V_5 = V_1\&V_5$; $W_5 = W_1\&W_2$; $U_5 = U_1\&U_2$; and $U_6 = U_1\&U_2\&U_4$.

An obtained working HMSO is presented in Fig. 11.4.

Strategy $S = P * D * U$

$S_2 = P_2 * D_1 * U_4 = (I_2 * T_1 * V_5) * (E_2 * W_5) * U_4 =$
$((I_2 * T_1) * (V_1\&V_2)) * (E_2 * (W_1\&W_2)) * U_4$

$S_5 = P_3 * D_1 * U_6 = (I_5 * T_1 * V_5) * (E_2 * W_5) * (U_1)\&U_2\&U_4) =$
$((I_1\&I_2) * T_1 * ((V_1)\&V_2))) * (E_2 * (W_1\&W_2)) * (U_1\&U_2\&U_4)$

$S_6 = S_2\&S_5$

Prototyping — Design — Utilization

$P = I * T * V$
$P_1 = I_2 * T_1 * V_5(1)$
$P_2 = I_2 * T_5 * V_5(1)$
$P_3 = I_5 * T_1 * V_5(1)$
$P_4 = I_5 * T_5 * V_5(2)$

$D = E * W$
$D_1 = E_2 * W_5(1)$

U
$U_1 = (3)$
$U_2 = (2)$
$U_4 = (1)$
$U_5 = U_1\&U_2(1)$
$U_6 = U_1\&U_2\&U_4(1)$

I Problem Identification
$I_1(2)$
$I_2(1)$
$I_5 = I_1\&I_2(1)$

T Generation of terminology
$T_1(1)$
$T_2(2)$
$T_5 = T_1\&T_2(1)$

V Knowledge acquisition
$V_1(3)$
$V_2(2)$
$V_5 = V_1\&V_2(1)$

E Enhancement
$E_1(3)$
$E_2(2)$

W Knowledge acquisition
$W_1(3)$
$W_2(2)$
$W_5 = W_1\&W_2(1)$

Fig. 11.4. Design morphology

Let us construct a series solution strategy on the basis of morphological clique problem. Priorities of DA's ($r = 1, ..., 3$) received by expert judgment are presented in brackets in Fig. 11.4. Tables 11.16, 11.17, 11.18, and 11.19 contain compatibility.

Table 11.16. Compatibility

	T_1	T_2	T_5	V_1	V_2	V_5
I_1	3	1	3	3	1	3
I_2	2	3	3	2	3	3
I_5	3	3	3	3	3	3
T_1				3	1	3
T_2				2	3	3
T_5				3	3	3

Table 11.17. Compatibility

	W_1	W_2	W_5
E_1	3	1	3
E_2	3	3	3

Table 11.18. Compatibility

	R_1	R_2	R_5
G_1	3	2	3
G_2	1	3	3
G_4	1	3	3
G_5	3	3	3

Table 11.19. Compatibility

	D_1	U_2	U_3	U_4	U_5	U_5
P_1	1	2	3	2	2	3
P_2	2	2	3	2	2	3
P_3	2	2	2	2	2	3
P_4	3	3	3	2	3	3
D_1		1	3	2	3	3

Table 11.20. Independence

	S_1	S_2	S_3	S_4	S_5
S_1	.	3	1	3	1
S_2	3	.	1	2	4
S_3	1	1	.	1	1
S_4	3	2	1	.	1
S_5	1	4	1	1	.

Now we combine composite Pareto-effective (by N) DA's:

(a) $P_1 = I_2 * T_1 * V_5$, $P_2 = I_2 * T_5 * V_5$, $P_3 = I_5 * T_1 * V_5$, $P_4 = I_5 * T_5 * V_5$, $N = (3; 3, 0, 0)$;

(b) $D_1 = E_2 * W_5$, $N(D_1) = (3; 1, 1, 0)$; and

(c) $S_1 = P_4 * D_1 * U_6$, $N(S_1) = (3; 2, 1, 0)$; and $S_2 = P_2 * D_1 * U_4$, $S_3 = P_2 * D_1 * U_6$, $S_4 = P_3 * D_1 * U_4$, $S_5 = P_3 * D_1 * U_6$, $N = (2; 3, 0, 0)$.

Fig. 11.5 illustrates two above-mentioned Pareto-effective points for S. Let us point out the following bottlenecks for some composite DA's: P_4 for S_1, (P_2, D_1) for S_3, and (P_3, D_1) for S_5. Each improvement of bottlenecks above leads to the ideal solution.

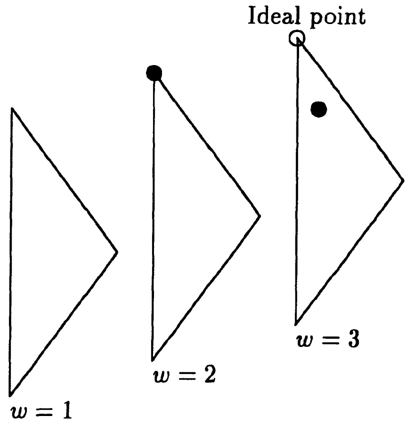

Fig. 11.5. Quality lattice and Pareto-effective points

Now we may design a parallel strategy for the top level (S). Table 11.20 contains estimates of independence or complementability between DA's for S. We search for a maximal clique which is Pareto-effective one by the following vector: $M(Y) = (h(Y); K(Y))$, where Y is a set of selected elements, and h is the minimum of pairwise independence in Y, and $K = |Y|$.

Finally, a resultant parallel strategy for S, which is computed on the basis of clique problem with the maximum element independence, is the following (Fig. 11.6): $S_2 \& S_5$, $M = (4; 2)$.

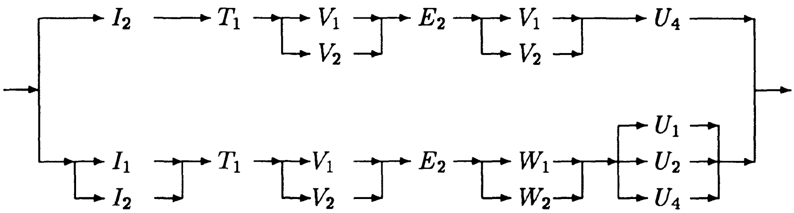

Fig. 11.6. Parallel-series solving strategies

11.3 DESIGN OF VIBRATION CONVEYOR

Design of mechanical machines is the basic direction of engineering design ([138], [165], [167], [168], [169], [217], [218], [287], [387], [463], [467], etc.). Parametric optimization is a traditional application of the multicriteria approach to the design of mechanical systems ([138], [486], etc.). However, hierarchical composition of systems is important for mechanical engineering practice too, especially for the design of complex systems ([113], etc.).

The use of hierarchical approaches to manufacturing planning has a long time ([114], [487], etc.). Sause and Powell examined hierarchical design process model for computer integrated structural engineering ([448], [449]). Recently hierarchical design schemes based on multicriteria analysis and selection have been used for many complex manufacturing systems ([202], [238], [269], [270], etc.). Tsukune et. al. proposed hierarchical modular manufacturing to integrate intelligent and complex machines [505].

This section addresses the implementation of HMMD to vibration conveyors [314]. Reduction gears, vibration machines are often basic examples for applications of machine design optimization ([133], [463], [482], etc.). Yet, our proposed example outlines the hierarchical composition of a mechanotronical system including the following: development of tree-like structure; multicriteria description of components and their interconnection; and selection, composition, and refinement of DA's.

11.3.1 Structure of Conveyor

We consider a hierarchical structure of vibration conveyor that is shown in Fig. 11.6.

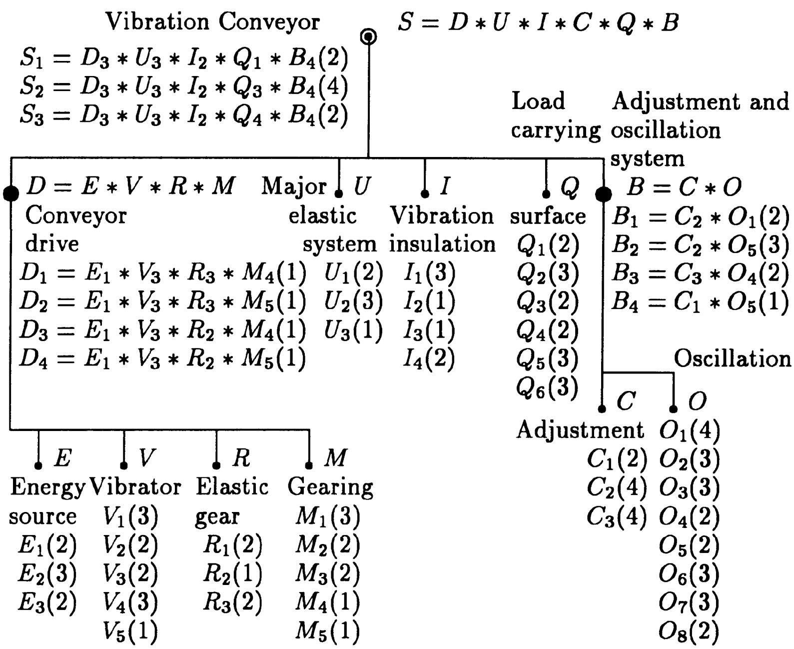

Fig. 11.6. Structure of vibration conveyor (ordinal priorities of DA's are shown in brackets)

11.3.2 Composition of Conveyor Drive

Table 11.21 contains criteria of components. DA's and estimates of components are presented in Tables 11.22, 11.23, 11.24, and 11.25. Table 11.26 contains compatibility between DA's for components of conveyor drive (D). Criteria for conveyor drive are presented in Table 11.27. Composite DA's are shown in Table 11.28.

11.3.3 Composition of Adjustment and Oscillation System

Table 11.29 contains criteria of components of B. DA's and estimates for components of B are presented in Tables 11.30 and 11.31. Criteria for adjustment and oscillation system are presented in Table 11.32. Table 11.33 contains compatibility between DA's for components of B. Composite DA's are shown in Table 11.34.

11.3.4 Composition of Vibration Conveyor

Table 11.35 contains criteria for U, I, and Q. DA's and estimates for components U, Q, and I are presented in Tables 11.36, 11.37, 11.38. Table 11.39 contains compatibility between DA's for components of vibration conveyor S. Criteria for vibration conveyor are presented in Table 11.40. Composite DA's are shown in Table 11.41. Fig. 11.7 depicts a concentric presentation of composite decision (S_1).

Table 11.21. Criteria for components of D

Criteria		Weight
Power	F_{e1}	−1
Starting torque	F_{e2}	−1
Reliability	F_{e3}	1
Cost	F_{e4}	−1
Maintenance	F_{e5}	1
Controllability	F_{e6}	1
Power	F_{v1}	1
Weight	F_{v2}	−1
Reliability	F_{v3}	1
Cost	F_{v4}	−1
Maintenance	F_{v5}	1
Clearance	F_{v6}	−1
Reliability	F_{r1}	1
Strength	F_{r2}	1
Mass	F_{r3}	−1
Clearance	F_{r4}	−1
Cost	F_{r5}	−1
Capacity	F_{r6}	1
Mass	F_{m1}	−1
Clearance	F_{m2}	−1
Efficiency	F_{m3}	1
Reliability	F_{m4}	1
Cost	F_{m5}	−1
Maintenance	F_{m6}	1

Table 11.22. DA's of E

DA's		Criteria					
		1	2	3	4	5	6
Electric energy	E_1	3	1	3	2	2	3
Hydraulic source	E_2	2	0	2	2	1	1
Pneumatic source	E_3	1	0	1	1	1	1

Table 11.23. DA's of R

DA's		Criteria					
		1	2	3	4	5	6
None	R_1	0	0	0	0	0	0
Linear system	R_2	2	2	1	1	1	1
Non-linear system	R_3	1	1	2	2	2	2

Table 11.24. DA's of V

DA's		Criteria					
		1	2	3	4	5	6
Hydraulic vibrator	V_1	4	3	2	3	1	3
Pneumatic vibrator	V_2	2	2	1	2	2	2
Eccentric vibrator	V_3	5	2	3	1	3	2
Electromagnetic vibrator	V_4	3	3	2	4	1	3
Piezoelectric vibrator	V_5	1	1	2	4	2	1

Table 11.25. DA's of M

DA's		Criteria					
		1	2	3	4	5	6
None	M_1	0	0	0	0	0	0
Reduction gear box	M_2	3	3	3	3	1	2
Worm gear box	M_3	3	2	1	3	1	2
Planet gear box	M_4	2	1	3	2	2	2
Wave gear box	M_5	1	1	2	2	3	1

Table 11.26. Compatibility for elements of D

	V_1	V_2	V_3	V_4	V_5	R_1	R_2	R_3	M_1	M_2	M_3	M_4	M_5
E_1	0	0	7	7	7	1	3	4	0	3	3	4	4
E_2	7	3	1	0	0	4	2	3	3	1	1	1	1
E_3	3	7	1	0	0	5	2	3	3	1	1	1	1
V_1						3	2	2	7	1	1	1	1
V_2						3	2	2	7	1	1	1	1
V_3						1	7	5	1	7	7	7	7
V_4						3	5	6	7	1	1	1	1
V_5						5	2	1	5	1	1	2	4
$R_1..R_3$									7	7	7	7	7

Table 11.27. Criteria for D

Criteria		Weight	Specification
Power	F_{d1}	-1	$F_{e15} + F_{v1}$
Starting torque	F_{d2}	-1	F_{e2}
Reliability	F_{d3}	1	$min(F_{e3}, F_{v3}, F_{r1}, F_{m4})$
Cost	F_{d4}	-1	$F_{e5} + F_{v4} + F_{r5} + F_{m5}$
Maintenance	F_{d5}	1	$min(F_{e5}, F_{v5}, F_{m6})$

Table 11.28. Composite DA's of D

DA's	Criteria					N
	1	2	3	4	5	
$D_1 = E_1 * V_3 * R_3 * M_4$	8	1	1	7	2	$4;1,3,0,0$
$D_2 = E_1 * V_3 * R_3 * M_5$	8	1	1	6	1	$4;1,3,0,0$
$D_3 = E_1 * V_3 * R_2 * M_4$	8	1	2	6	2	$3;2,2,0,0$
$D_4 = E_1 * V_3 * R_2 * M_5$	8	1	2	5	1	$3;2,2,0,0$

Table 11.29. Criteria for components of B

Criteria		Weight
Power drain	F_{c1}	-1
Starting torque	F_{c2}	-1
Cost	F_{c3}	-1
Reliability	F_{c4}	1
Maintenance	F_{c5}	1
Power drain	F_{o1}	-1
Capacity	F_{o2}	1
Controllability	F_{o3}	1
Destroyability	F_{o4}	-1

Table 11.30. DA's of C

DA's		Criteria				
		1	2	3	4	5
Quasi-resonance (after)	C_1	2	2	1	3	3
Resonance	C_2	1	1	3	1	1
Quasi-resonance (before)	C_3	3	3	2	2	2

Table 11.31. DA's of O

DA's		Criteria			
		1	2	3	4
Straightforward/harmonic/with throw up	O_1	3	2	1	2
Elliptic/harmonic/with throw up	O_2	4	3	2	2
Straightforward/harmonic/without throw up	O_3	1	1	1	1
Elliptic/harmonic without/throw up	O_4	3	2	2	1
Straightforward/non-harmonic/with throw up	O_5	5	4	2	1
Elliptic/non-harmonic/with throw up	O_6	6	5	3	3
Straightforward/non-harmonic/without throw up	O_7	2	2	2	2
Elliptic/non-harmonic/without throw up	O_8	3	3	3	2

Table 11.32. Criteria for B

Criteria	Weight	Specification
Energy drain F_{b1}	-1	$F_{c1} + F_{o1}$
Starting torque F_{b2}	-1	F_{c2}
Cost F_{b3}	-1	F_{c3}
Reliability F_{b4}	1	F_{c4}
Maintenance F_{b5}	1	F_{c5}

Table 11.33. Compatibility for elements of B

	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8
C_1	5	4	3	2	4	3	2	3
C_2	7	6	5	5	6	5	3	3
C_3	3	3	6	6	2	1	5	4

Table 11.34. Composite DA's of B

DA's	Criteria					N
	1	2	3	4	5	
$B_1 = C_2 * O_1$	4	1	3	1	1	$7; 0, 0, 0, 2$
$B_2 = C_2 * O_5$	6	1	3	1	1	$6; 0, 1, 0, 1$
$B_3 = C_3 * O_4$	6	3	2	2	2	$6; 0, 1, 0, 1$
$B_4 = C_1 * O_5$	7	2	1	3	3	$4; 0, 2, 0, 0$

Table 11.35. Criteria for U, I, Q

Criteria		Weight
Reliability	F_{u1}	1
Strength	F_{u2}	1
Mass	F_{u3}	−1
Clearance	F_{u4}	−1
Cost	F_{u5}	−1
Maintenance	F_{u6}	1
Rigidity	F_{q1}	1
Weight	F_{q2}	−1
Cost	F_{q3}	−1
Reliability	F_{q4}	1
Maintenance	F_{q5}	1
Capacity	F_{q6}	1
Decreasing of amplitude of vibration conveyor oscillation	F_{i1}	1
Weight	F_{i2}	−1
Cost	F_{i3}	−1
Reliability	F_{i4}	1
Maintenance	F_{i5}	1

Table 11.36. DA's of U

DA's		Criteria					
		1	2	3	4	5	6
Torsion-elastic system	U_3	2	2	2	2	3	2
Spring-elastic system	U_2	1	1	3	3	2	3
Rubber-metal elastic system	U_1	3	3	1	1	1	1

Table 11.37. DA's of I

DA's		Criteria				
		1	2	3	4	5
None	I_1	0	0	0	0	0
Shock absorption vibration oscillation	I_2	1	3	1	2	3
Dynamic dampers of oscillation	I_3	2	2	2	2	2
Active vibration with active active element in control circuit (feedback)	I_4	3	1	3	1	1

Table 11.38. DA's of I

DA's		Criteria					
		1	2	3	4	5	6
Trough-like surface (small)	I_1	1	1	1	1	0	1
Trough-like surface (middle)	I_2	1	2	2	1	0	2
Trough-like surface (big)	I_3	1	3	3	1	0	3
Box-like surface (small)	I_4	1	1	1	1	0	1
Box-like surface (middle)	I_5	1	2	2	1	0	2
Box-like surface (big)	I_6	1	3	3	1	0	3

Table 11.39. Compatibility for elements of S

	$U_1..U_3$	I_1	I_2	I_3	I_4	$Q_1..Q_6$	$B_1..B_4$
$Q_1..Q_6$	6	6	6	6	6	6	6
$I_1..I_4$		6	7	6	6	6	6
$U_1..U_3$						6	6
$D_1..D_4$							6

Table 11.40. Criteria for S

Criteria		Weight	Specification
Capacity	F_{s1}	1	F_{q6}
Clearance	F_{s2}	-1	
Weight	F_{s4}	-1	$F_{i2} + F_{q2}$
Metal drain	F_{s4}	-1	
Energy drain	F_{s5}	-1	F_{b1}
Reliability	F_{s6}	1	$min(F_{d3}, F_{b1}, F_{i4}, F_{q4}, F_{b4})$
Strength	F_{s7}	1	F_{u4}
Controllability	F_{s8}	1	
Cost	F_{s9}	-1	$F_{d4} + F_{u5} + F_{i3} + F_{q3} + F_{b3}$
Maintenance	F_{s10}	1	$min(F_{d5}, F_{u6}, F_{i5}, F_{q5}, F_{b5})$

Table 11.41. Composite DA's of S

DA's	Criteria										N
	1	2	3	4	5	6	7	8	9	10	
$S_1 = P_3 * U_3 * I_2 * Q_1 * B_4$	1	0	1	0	0	1	0	0	1	1	$6;4,1,0,0$
$S_2 = P_3 * U_3 * I_2 * Q_3 * B_4$	3	0	3	0	0	1	0	0	3	1	$6;4,1,0,0$
$S_3 = P_3 * U_3 * I_2 * Q_4 * B_4$	1	0	1	0	0	1	0	0	1	1	$6;4,1,0,0$

11.3.5 Revelation of Bottlenecks and Refinement

Finally, let us study bottlenecks and improving actions for conveyor drive (Table 11.42). The following types of improvement actions are used: generation of an ideal point (1), improvement of a Pareto-effective point (2); extension of a Pareto-effective point set by means of additional point(s) (3); and construction of a Pareto-effective point set by means of dominance relation (4).

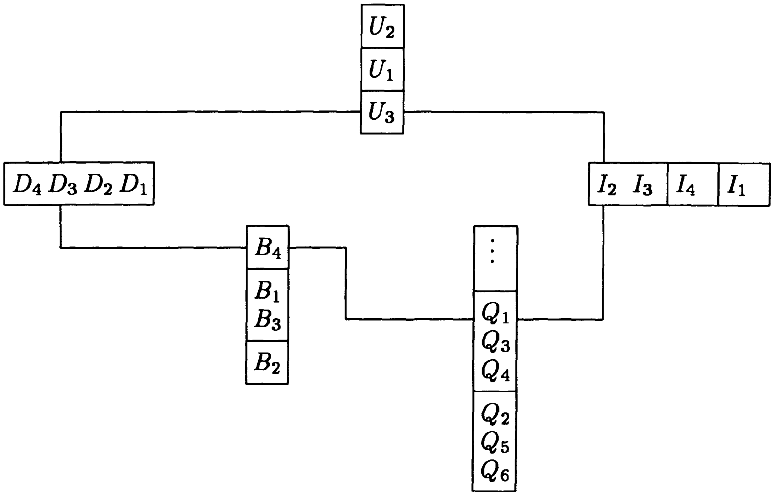

Fig. 11.7. Concentric presentation of composite DA's (S_1)

Table 11.42. Some bottlenecks and improvements

Composite DA's	Bottlenecks		Action	
	DA's	Ins	w/r	Type
$D_1 = E_1 * V_3 * R_3 * M_4$	E_1		$2 \Rightarrow 1$	2
$D_1 = E_1 * V_3 * R_3 * M_4$	V_3		$2 \Rightarrow 1$	2
$D_1 = E_1 * V_3 * R_3 * M_4$	R_3		$2 \Rightarrow 1$	2, 4
$D_2 = E_1 * V_3 * R_3 * M_5$	E_1		$2 \Rightarrow 1$	2
$D_2 = E_1 * V_3 * R_3 * M_5$	V_3		$2 \Rightarrow 1$	2
$D_2 = E_1 * V_3 * R_3 * M_5$	R_3		$2 \Rightarrow 1$	2, 4
$D_3 = E_1 * V_3 * R_2 * M_4$	E_1		$2 \Rightarrow 1$	2
$D_3 = E_1 * V_3 * R_2 * M_4$	V_3		$2 \Rightarrow 1$	2
$D_4 = E_1 * V_3 * R_2 * M_5$	E_1		$2 \Rightarrow 1$	2
$D_4 = E_1 * V_3 * R_2 * M_5$	V_3		$2 \Rightarrow 1$	2
$D_3 = E_1 * V_3 * R_2 * M_4$		(E_1, R_2)	$3 \Rightarrow 4$	2, 4
$D_4 = E_1 * V_3 * R_2 * M_5$		(E_1, R_2)	$3 \Rightarrow 4$	2, 4
$E_1 * V_3 * R_2 * M_2$	M_2		$2 \Rightarrow 1$	3
$E_1 * V_3 * R_2 * M_3$	M_3		$2 \Rightarrow 1$	3
$E_1 * V_3 * R_2 * M_5$		(V_5, R_2)	$2 \Rightarrow 3$	3, 4

11.4 DEVELOPMENT PHASES OF A DSS

Now we consider an illustration for system development on the basis of DSS COMBI (Fig. 11.8, 11.9, 11.10, and 11.11) ([294], [297], [317], etc.). Some descriptions of the system were presented in the chapters: 1 (morphological graph-menu for techniques), 3 (morphological environment for techniques and composition of series-parallel solving strategies), and 11 (graphical user interface). Here we examine five phases (generation of system versions: 0, 1, 2, 3, 4) with corresponding structures and DA's.

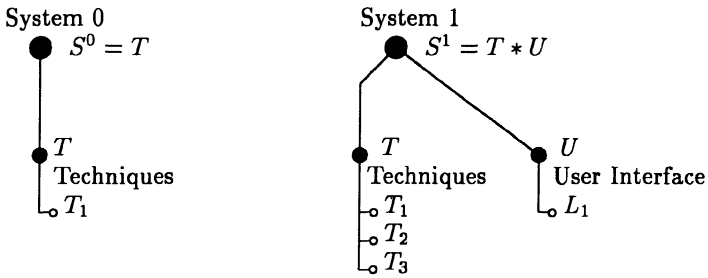

Fig. 11.8. Development of DSS Structure (1st step): phase 1

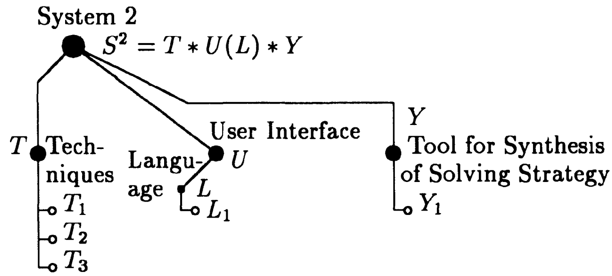

Fig. 11.9. Development of DSS Structure (2nd step): phase 2

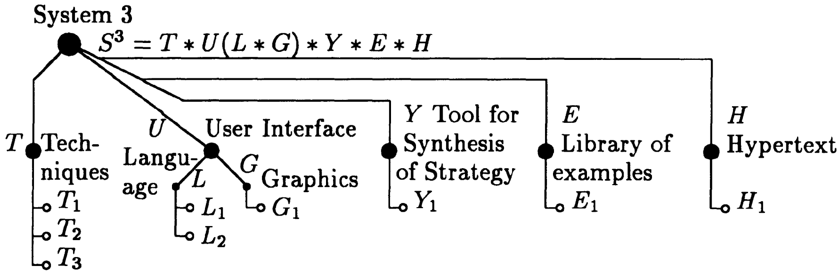

Fig. 11.10. Development of DSS Structure (3nd step): phase 3

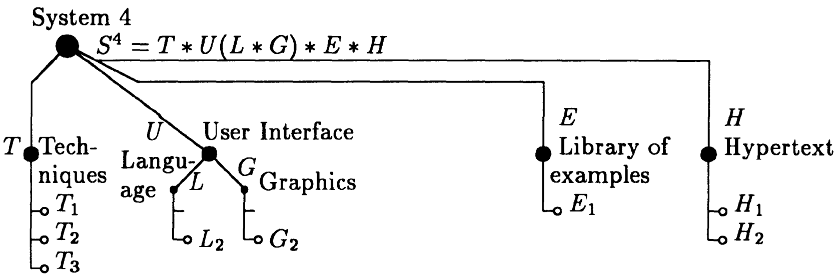

Fig. 11.11. Development of DSS Structure (4nd step): phase 4

Note that phases 1, 2, and 3 are based on an extension of previous system versions (i.e., extension of structure and addition of DA's). Phase 4 is based on an extension and a change of previous system version, including deletion of DA's.

Here we use representatives of DA's as follows: ELECTRE-like technique T_1; utility function technique T_2; expert stratification technique T_3; command language L_1; menu for techniques L_2; morphological graph-menu for techniques G_1; graphics for data G_2; tool for composing a composite solution strategy Y_1; database of examples E_1; hypertext system (description of multicriteria models, examples of criteria, references, etc.) H_1; and hypertext system on investment H_2.

Note that phase 4 was based on the following:

(a) mainly users did not use the mode of strategy synthesis (the corresponding component was deleted); and

(b) it was required to improve usability of the system by additional graphical presentation of information.

Our example demonstrates a possibility to investigate dynamics of decomposable systems, including the following:

(i) trajectories of development;

(ii) similarity/dissimilarity of system versions; and

(iii) quality analysis of complex decomposable systems.

Clearly, dynamics of decomposable systems can be applied as a fundamental to design special data and knowledge bases for research and education in the field of complex systems.

In addition, it is reasonable to note an interesting attempt to modeling dynamics for hierarchical systems (graph dynamics) that has been proposed by Aizerman et al. in ([8], and [9]). In these papers, the following kinds of operations have been introduced:

(a) unary operations;

(b) additive operations;

(c) binary and more complex operations; and

(d) n-ary operations.

As a result, functional inequalities and equations to describe system changes are proposed.

11.5 SELECTION OF INTERRELATED PROJECTS

Selection of interrelated research projects has been intensively studied by many scientists ([71], [101], [478], etc.). The problem is used in scientific foundations, in investment processes and in educational management. In our opinion, interrelated research projects may be very useful in organizing a group of students with different professional orientations for joint project execution.

In this section, we consider a simple illustrative example that is oriented to a composite project consisting of three interrelated studies in management, ecology, and urban planning (Fig. 11.12). Clearly our example may be extended by the addition of several other topics (e.g., operations research, computer science, sociology, technological forecasting, information science, and decision science). Here morphological approach is applied. On the other hand, it is reasonable to examine the same numerical example on the basis of several different models.

Table 11.43 contains criteria for alternative projects. Table 11.44 presents DA's, and their estimates on criteria. Factors of compatibility and corresponding estimates (initial and resultant) are contained in Tables 11.45, 11.46, and 11.47. Resultant estimates of compatibility is based on multicriteria ranking. Composite DA's, bottlenecks and improvement actions are presented in Table 11.48 and 11.49. Fig. 11.13 depicts the space of system excellence and composite DA's.

Composite project $S = E * R * U$

$S_1 = E_4 * R_4 * U_3(1)$

$S_2 = E_4 * R_6 * U_3(1)$

E	R	U
Management	Ecology	Urban planning
$E_1(1)$	$R_1(1)$	$U_1(2)$
$E_2(2)$	$R_2(2)$	$U_2(4)$
$E_3(3)$	$R_3(3)$	$U_3(1)$
$E_4(1)$	$R_4(2)$	$U_4(1)$
$E_5(2)$	$R_5(4)$	
$E_6(2)$	$R_6(1)$	

Fig. 11.12. Structure of composite project (priorities of DA's are shown in brackets)

Table 11.43. Criteria

Criteria	Weights		
	E	R	U
1.New scientific result	3	3	5
2.Possible utilization	5	5	4
3.Executed part of research	3	4	4
4.Personnel characteristic (background, experience,etc.)	4	2	2

Table 11.44. DA's and their estimates

DA's	Criteria			
	1	2	3	4
E_1 Analysis of taxation policy	4	5	3	4
E_2 International marketing for conversion production	2	5	4	3
E_3 International marketing for small business	2	5	3	3
E_4 Strategy planning	3	5	5	4
E_5 Support of new technologies	3	4	3	5
E_6 Economics of information technology	4	3	4	4
R_1 Risk analysis of chemical manufacturing	2	5	4	4
R_2 Environmental impact to pulmonary diseases	5	5	3	3
R_3 Ambient conditions in highway neighborhood	3	4	3	3
R_4 Ecological activities after the end of oil/gas field utilization	3	5	3	4
R_5 Water resources for urban conglomeration	3	3	3	4
R_6 Computer system development for ecological analysis in urban regions	2	4	4	5
U_1 Transport-traffic control system for for urban conglomeration	3	4	4	4
U_2 DSS for urban planning	3	4	3	3
U_3 AI tools for analysis of urban regions	4	5	3	3
U_4 System analysis of house-building for dwelling	4	4	4	4

Table 11.45. Factors of compatibility

Factors	Weight
1.Common methodology, techniques	1
2.Common information	3
3.Common investigated subject (region, etc.)	5

Table 11.46. Compatibility

	R_1	R_2	R_3	R_4	R_5	R_6
E_1	(000)/0	(000)/0	(000)/0	(011)/3	(001)/2	(001)/2
E_2	(000)/0	(000)/0	(000)/0	(011)/3	(011)/3	(012)/4
E_3	(000)/0	(000)/0	(000)/0	(000)/0	(000)/0	(000)/0
E_4	(111)/3	(111)/3	(112)/4	(222)/5	(211)/3	(211)/3
E_5	(001)/2	(001)/2	(000)/0	(001)/2	(001)/2	(001)/2
E_6	(111)/3	(111)/3	(000)/0	(111)/3	(000)/0	(221)/4

Table 11.47. Compatibility

	U_1	U_2	U_3	U_4
E_1	(000)/0	(000)/0	(000)/0	(000)/0
E_2	(011)/3	(011)/3	(011)/3	(000)/0
E_3	(000)/0	(000)/0	(000)/0	(000)/0
E_4	(112)/5	(111)/4	(111)/4	(111)/4
E_5	(011)/4	(000)/0	(000)/0	(011)/4
E_6	(111)/4	(111)/4	(111)/4	(011)/4
R_1	(000)/0	(000)/0	(000)/0	(000)/0
R_2	(000)/0	(300)/1	(300)/1	(011)/0
R_3	(101)/3	(000)/0	(211)/5	(000)/0
R_4	(300)/1	(300)/1	(300)/1	(200)/0
R_5	(000)/0	(000)/0	(210)/3	(000)/0
R_6	(100)/0	(300)/1	(300)/1	(000)/0

Table 11.48. Composite DA's

DA's	N
$S_1 = E_4 * R_3 * U_3$	$4; 2, 0, 1, 0$
$S_2 = E_4 * R_6 * U_3$	$1; 3, 0, 0, 0$

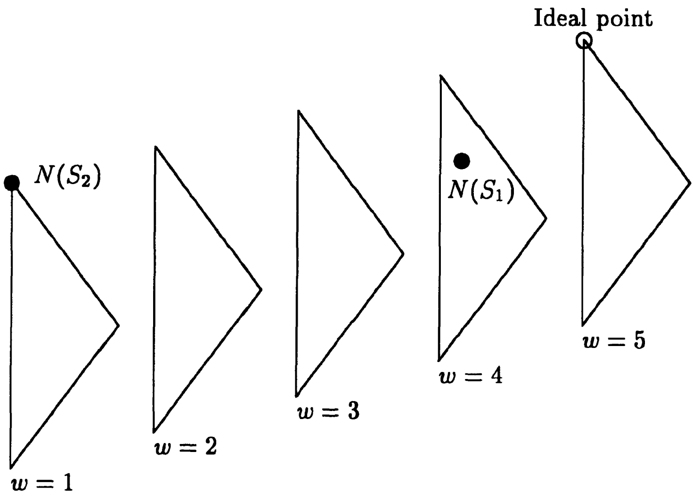

Fig. 11.13. Quality lattice and Pareto-effective points

Table 11.49. Some bottlenecks and improvement actions

Composite DA's	Bottlenecks		Action	
	DA's	Ins	w/r	Type
$S_1 = E_4 * R_3 * U_3$	R_3		$3 \Rightarrow 1$	2
$S_1 = E_4 * R_6 * U_3$		(E_4, U_3)	$4 \Rightarrow 5$	2

11.6 INVESTMENT

Investment problems are often based on knapsack-like models, multicriteria comparison of basic investment alternatives, and probabilistic analysis. This field has been intensively studied for many years ([319], [320],[346], [484], etc.).

Composite portfolio $S = A * B * C$

$S_1 = A_4 * B_4 * C_3(1)$

$S_2 = A_4 * B_6 * C_3(1)$

A Short-time investment	B Middle-time investment	C Long-time investment
$A_1(3)$	$B_1(3)$	$C_1(3)$
$A_2(3)$	$B_2(4)$	$C_2(3)$
$A_3(3)$	$B_3(2)$	$C_3(2)$
$A_4(2)$	$B_4(3)$	$C_4(1)$
	$B_5(1)$	

Fig. 11.14. Structure of composite investment portfolio (priorities of DA's are shown in brackets)

Table 11.50. Criteria

Criteria	Weights		
	A	B	C
1.Profit (0,...,100)	5	7	6
2.Risk (-; 0,...,100)	4	5	4
3.Prestige (0,1,2)	4	3	3
4.Possibility for continuation (0,...,5)	3		
5.Possibility to found a new company (0,1,2)		4	4
6.Obtaining a new experience (0,...,7)		2	2
7.Possibility to organize a new market (0,...,7)		4	4
8.Possibility to obtain *NAME* (0,...,7)	2	2	2
9.Connection with previous activity (0,...,7)	3	2	3

Here let us consider a simple hypothetical illustrative example that is based on morphological approach. Fig. 11.14 depicts a structure of a composite investment portfolio. Table 11.50 contains criteria. DA's and their estimates are shown in Table 11.51.

Table 11.51. DA's and their estimates

DA's	Criteria								
	1	2	3	4	5	6	7	8	9
A_1 State bonds	10	0	2	2				0	1
A_2 Bank deposit	12	0	2	2				2	0
A_3 Speculation on the stock exchange	300	40	3	1				2	0
A_4 Oil shares	20	25	3	1				2	0
B_1 State bonds	15	0	2		0	0	0	0	1
B_2 Bank deposit	18	0	2		0	0	0	0	1
B_3 Immovables	20	10	4		1	2	1	6	3
B_4 Jewelry	30	30	4		0	1	1	2	2
B_5 Biotechnology (shares)	28	20	4		2	6	5	6	7
C_1 State bonds	20	0	2		0	0	0	0	1
C_2 Bank deposit	24	0	2		0	0	0	0	1
C_3 Antique	30	20	4		1	1	1	5	2
C_4 Airspace companies (shares)	26	12	4		1	3	2	4	4

Table 11.52. Compatibility

	B_1	B_2	B_3	B_4	B_5	C_1	C_2	C_3	C_4
A_1	4	3	3	3	5	4	3	5	5
A_2	5	4	4	3	5	5	4	5	4
A_3	5	5	4	3	2	5	5	4	3
A_4	5	5	5	4	2	5	5	4	3
B_1						4	3	5	5
B_2						4	4	5	5
B_3						5	5	4	4
B_4						5	5	3	3
B_5						5	5	4	3

Table 11.53. Composite DA's

DA's	N
$S_1 = A_4 * B_3 * C_1$	$5; 0, 2, 1, 0$
$S_2 = A_2 * B_5 * C_1$	$5; 1, 0, 2, 0$
$S_3 = A_2 * B_3 * C_4$	$4; 1, 1, 1, 0$
$S_4 = A_2 * B_5 * C_4$	$3; 2, 1, 0, 0$

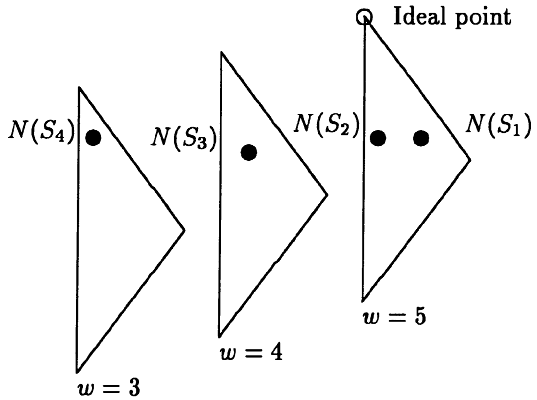

Fig. 11.15. Quality lattice and Pareto-effective points

Compatibility is presented in Table 11.52. Table 11.53 involves composite DA's, Table 11.54 contains bottlenecks and improvement actions. Fig. 11.15 depicts Pareto-effecive points that correspond to composite DA's.

Table 11.54. Some bottlenecks and improvement actions

Composite DA's	Bottlenecks		Action	
	DA's	Ins	w/r	Type
$S_1 = A_4 * B_3 * C_1$	A_2		$2 \Rightarrow 1$	2
$S_1 = A_4 * B_3 * C_1$	B_3		$2 \Rightarrow 1$	2
$S_1 = A_4 * B_3 * C_1$	C_1		$3 \Rightarrow 2$	2
$S_2 = A_2 * B_5 * C_1$	A_2		$3 \Rightarrow 2$	2
$S_2 = A_2 * B_5 * C_1$	C_1		$3 \Rightarrow 2$	2
$S_3 = A_2 * B_3 * C_4$	A_2		$3 \Rightarrow 2$	2
$S_3 = A_2 * B_3 * C_4$	B_3		$2 \Rightarrow 1$	2
$S_3 = A_2 * B_3 * C_4$		(A_2, B_3)	$4 \Rightarrow 5$	2
$S_4 = A_2 * B_5 * C_4$	A_2		$3 \Rightarrow 2$	2
$S_4 = A_2 * B_5 * C_4$		(B_5, C_4)	$3 \Rightarrow 4$	2

11.7 PLANNING OF EXPLORATION FOR OIL/GAS FIELDS

This section is based on a material that was prepared by the author and Dr. V.I. Poroskun (Moscow, Russia). Initial data on oil/gas fields in peninsula Yamal (Siberia) were taken from the handbook [340].

The following hierarchy of geological objects is considered: (i) a basic object (bottom level); (ii) group of geological objects; (iii) oil/gas fields; and (iv) geological region.

The following attributes for geological objects above are taken into account:

(1) characteristic of deposit as follows: deposit (3), perspective geological object (2), and perspective object in geological snare (1);

(2) depth of deposit (m);

(3) type of fluid (classification factor of hydrocarbon as follows: gas, gas & condensate, and oil);

(4) volume of supply or resource (cubic m);

(5) oil/gas output (cubic m in 24 hours);

(6) complexity of mining and geological situation;

(7) reliability (risk) of obtaining an output (0...100);

(8) category of supply (S1 - 0.2, S2 - 0.5, S3 - 0.8, etc.); and

(9) proximity to a technological base.

The following list of alternative actions (DA's) is studied for each geological object:

(a) without action (X_1);

(b) additional exploration action for definition of supply (X_2);

(c) independent use of geological object (gas) (X_3);

(d) independent use of geological object (oil) (X_4);

(e) independent use of geological object (gas and oil) (X_5);

(f) joint use of geological object (gas) (X_6);

(h) joint use of geological object (oil) (X_7); and

(g) joint use of geological object (oil+gas) (X_8).

Further, for each geological object we will select on the basis of expert judgment a subset of these alternative actions (with corresponding index). Thus these selected DA's are a base to combine composite exploration actions (composite DA's) for a group of geological objects and, then, for oil/gas fields.

The basic material involves a geological region, that consists of five oil/gas fields as follows [340]: "Harosovey", "Arkticheskoe", "Neitinskoe", "Krusensternskoe", and "Bovanenkovskoe".

Exploration plans were built for each oil/gas field, and the composite plan for the region was combined. Here we present only an example for oil/gas fields "Harosovey".

A structure of the exploration plan shown in Fig. 1.16.

The expert, who has participated into the solving process in the considered problem, had excellent knowledge and intuition in geology, and it was satisfied for the specification of the following initial results:

(1) selection of DA's for each geological object, i.e. each leaf node of the hierarchical model (Fig. 11.16);

(2) ranking the DA's for each geological object; and
(3) specification of compatibility between DA's (in ordinal scale).

Table 11.55 presents factors that are basic for the specification of compatibility between strategy elements. Compatibility between DA's that were specified by expert are presented in the Tables 11.56, 11.57 and 11.58.

Here we obtain the only one composite decision for each component B, C, D, and, as result, for top-level too (Table 11.59).

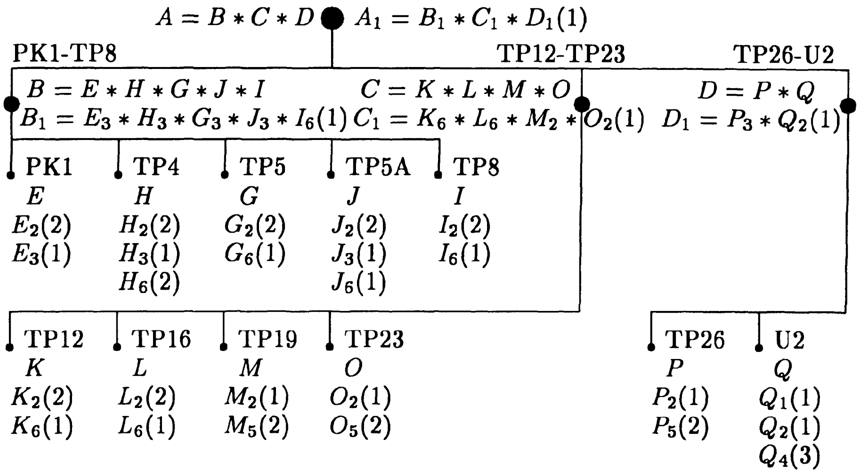

Fig. 11.16. Structure of the exploration plan

Table 11.55. Factors of compatibility

DA's	Factors
1. PK1 E & TP4 H	Proximity, supply
2. PK1 E & TP5 G	Proximity, supply
3. PK1 E & TP5A J	Proximity, supply
4. PK1 E & TP8 I	Proximity, supply
5. TP4 H & TP5 G	Supply, proximity
6. TP4 H & TP5A J	Supply, proximity
7. TP4 H & TP8 I	Supply, proximity
8. TP5 G & TP5A J	Supply, proximity
9. TP5 G & TP8 I	Supply, proximity
10.TP5A J & TP8 I	Supply, proximity
11.TP12 K & TP16 L	$S_5 + V$, proximity
12.TP12 K & TP19 M	Supply, $S_5 + V$, proximity
13.TP12 K & TP23 O	Supply, $S_5 + V$, proximity
14.TP16 L & TP19 M	Supply, $S_5 + V$, proximity
15.TP16 L & TP23 O	Supply, $S_5 + V$, proximity
16.TP19 M & TP23 O	Supply, $S_5 + V$, proximity
17.TP26 P & U2 Q	Supply, $S_5 + V$, proximity

Table 11.56. Compatibility for PK1-TP8 B

	H_2	H_3	H_6	G_2	G_6	J_2	J_3	J_6	I_2	I_6
E_2	2	2	1	2	1	2	2	3	2	3
E_3	2	4	1	1	2	2	4	1	2	4
H_2				3	3	2	2	3	3	4
H_3				3	4	2	3	2	3	4
H_6				3	4	1	3	4	3	4
G_2						3	4	3	2	3
G_6						1	3	3	3	4
J_2									3	3
J_3									2	2
J_6									3	4

Table 11.57. Compatibility TP12-TP23 C

	L_2	L_6	M_2	M_5	O_2	O_5
K_2	3	3	4	3	4	3
K_6	3	4	4	0	4	0
L_2			4	3	4	3
L_6			3	0	3	0
M_2					4	3
M_5					3	2

Table 11.58. Compatibility TP26-U2 D

	Q_1	Q_2	Q_6
P_2	4	4	1
P_5	4	2	1

Table 11.59. Composite DA's

DA's	N
$B_1 = E_3 * H_3 * G_3 * J_3 * I_6$	$2; 5, 0, 0, 0$
$C_1 = K_6 * L_6 * M_2 * O_2$	$3; 4, 0, 0, 0$
$D_1 = P_3 * Q_2$	$4; 2, 0, 0, 0$
$A_1 = B_1 * C_1 * D_1$	$5; 3, 0, 0, 0$

11.8 SUMMARY

This chapter has demonstrated additional applications of morphological approaches to decomposable systems. Our examples can be used as basic ones for new applications and for learning of potential users.

12 CONCLUSION

This book is based on the following viewpoint:

A number of creative problems will be increasing in all applied and scientific domains, and combinatorial morphological appoarches may be successfully used for many of them.

We have examined combinatorial engineering of decomposable systems as an engineering implementation of combinatorial morphological approaches to many applied systems. The following basic operations for decomposable systems were considered:

1. *Description and/or presentation*, including tree-like system model, external requirements: criteria, constraints for the system and its elements, design alternatives (DA's) for the elements (nodes of the system model), interconnection (Ins) among DA's, and estimates of DA's and Ins.

2. *Analysis and evaluation*, i.e., multi-level assessment of the system and its elements, including assessment of composite DA's in a complex space of system excellence.

3. *Analysis as revelation of bottlenecks* (by system parts, by Ins, by and system structure).

4. *Comparison* of system versions (by components and DA's, by Ins, and by structure).

5. *Selection* of system or their components.

6. *Synthesis*, including two problems as follows: (i) selection of the best system version; (ii) hierarchical synthesis (design of system model, specification of requirements, generation of DA's, assessment of DA's and Ins, composition of composite DA's).

7. *Transformation (e.g., modification, improvement, adaptation, change, approximation)*, including generation of improvement actions (improvement of DA's and/or Ins and modification of the system model) and scheduling.

We have investigated approaches to the above-mentioned operations including many realistic examples. On the other hand, our description of decomposable systems and operations for them require many new investigations. The proposed materials (models, examples, etc.) may be considered as the first step to develop **ABC** of decomposable systems. So there is an additional goal of the book:

To attract scientists, researhers, and practioners attentions to engineering of decomposable systems.

Note combinatorial optimization problems have been investigated for many years in various applications. However, the use of these problems for composing the complex design alternatives on the basis of standard designs is a significant opportunity for the improvement of design and planning processes. This is the key problem.

In our book, we have examined several combinatorial composition problems. In our opinion, proposed morphological clique problem is very useful by two reasons as follows:

(a) as a basis for system design and planning (many applied domains); and

(b) as a basis for algorithm design (i.e., morphological approach to macroheuristics).

Obviously, it is reasonable to study various other modifications of the composition problems too.

The list of several significant issues for future investigations is the following:

1. Exploration of many new application fields (e.g., medical treatment and innovation processes).

2. Modeling of system development phases (development trajectories) including applications for various domains (e.g., information systems, software packages, transportation systems, communication systems, and software packages).

3. The use of morphological metaheuristics for many other combinatorial problems.

4. The use of morphological approaches at various phases of knowledge engineering (problem identification, knowledge acquisition, testing, explanation, and utilization of knowledge).

5. Approximation and comparison of structures.

6. Graphical representation of the composite DA's and the process of composing.

7. Education in the field of decomposable systems and their applications.

References

[1] M. Abramovici, M. Breuer, and A. Friedman, *Digital System Testing and Testability Design.* New York: Computer Society Press, 1990.

[2] R.L. Ackoff, *Creating the Corporate Future.* New York: J. Wiley & Sons, 1981.

[3] B. Adelson, Impediments to the Process of Group Work. In: *Proc. Intl. Conf. on Human-Computer Interaction EWHCI'93*, Moscow, Vol. I, pp. 191-207, 1993.

[4] A. Agrawal, P. Klein, and R. Ravi, When Trees Collide: An Approximation Algorithm for the Generalized Steiner Problem on Networks. *SIAM J. on Computing*, Vol. 24, No. 3, pp. 440-456, 1995.

[5] N. Ahituv, and M. Igbaria, A Model to Facilitate Cost, Pricing and Budget of Computer Services, *Information & Management*, Vol. 14, pp. 235-241, 1988.

[6] N. Ahituv, and S. Neumann, A Flexible Approach to Information System Development. *MIS Quarterly*, Vol. 8, No. 2, pp. 69-78, 1984.

[7] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, *The Design and Analysis of Computer Algorithms,* Mass: Addison-Wesley, Reading, 1974.

[8] M.A. Aizerman, L.A. Gusev, S.V. Petrov, and I.M. Smirnov, A Dynamic Approach to Analysis of Structured Representatives as Graphs (Fundamentals of Graph Dynamics). I. *Avtomatika i Telemekhanika*, No. 7, pp. 135-151, 1977 (in Russian).

[9] M.A. Aizerman, L.A. Gusev, S.V. Petrov, and I.M. Smirnov, A Dynamic Approach to Analysis of Structured Representatives as Graphs (Fundamentals of Graph Dynamics). II. *Avtomatika i Telemekhanika*, No. 9, pp. 123-136, 1977 (in Russian).

[10] M. Aizerman, and F. Aleskerov, *Theory of Choice.* Amsterdam: Elsevier, 1995.

[11] R.M. Akscyn, D.L. McCracken, and E.A. Yoder, KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations. *Comm. of the ACM*, Vol. 31, No. 7, pp. 820-825, 1988.

[12] C. Alexander, *Notes on the Synthesis of Form.* Cambridge, Mass.: Harvard University Press, 1964.

[13] J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, (eds.), *Artificial Evolution.* LNCS, Vol. 1063, Berlin: Springer-Verlag, 1996.

[14] G.S. Altshuller, *Creativity as Exact Science. The Theory of the Solution of Inventive Problems*, London: Gordon and Breach Science Publisher, 1984.

[15] G. Anandalingam, R. Mathieu, L. Pittard, and N. Sinha, Artificial Intelligence Based Approaches to Solving Hierarchical Optimization Problems. In: R. Sharda, B. Golden, E. Wasil, O. Balci and W. Stewart (eds.) *Impacts of Recent Computer Advances on Operations Research*, Elsevier, pp. 289-301, 1983.

[16] G. Anandalingam, and T.L. Friesz (eds.) *Hierarchical Optimization*, Annals of Operations Research, Vol. 34, Basel: Baltzer, 1992.

[17] H.I. Ansoff, *Implanting Strategic Management.* NJ.: Prentice Hall, 1984.

[18] J. Aoe, (ed.), *Computer Algorithms: String Pattern Matching Strategies.* Los Alamitos: IEEE Computer Society Press, 1994.

[19] L.M. Applegate, B.R. Konsynski, and J.F. Nunamaker, Model Management Systems: Designs for Decision Support. *Decision Support Systems*, Vol. 2, No. 1, pp. 81-91, 1986.

[20] Ph. Arabie, and S.A. Boorman, Multidimensional Scaling of Measures of Distance between Partitions. *J. of Math. Psychology*, Vol. 10, No. 2, pp. 148-203, 1973.

[21] Ph. Arabie, and L.J. Hubert, Combinatorial Data Analysis. *Annual Review of Psychology*, Vol.43, pp. 169-203, 1992.

[22] R.U. Ayres, *Technological Forecasting and Long-Time Planning.* New York: McGraw-Hill Book Company, 1969.

[23] T. Back, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* New York: Oxford University Press, 1996.

[24] F. Backley, and F. Harary, *Distance in Graphs.* Redwood City, Ca.: Addison-Wesley, 1990.

[25] K. Baker, *Introduction to Sequencing and Scheduling.* New York: J.Wiley & Sons, 1974.

[26] V.E. Baker, and D.E. O'Connor, Expert Systems for Configuration at Digital: XCON and BEYOND. *Commun. of the ACM.* Vol. 32, No. 3, pp. 298-318, 1989.

[27] M. Balabanovic, and Y. Shoham, Fab: Content-Based, Collaborative Recommendation. *Commun. of the ACM.* Vol. 40, No. 3, pp. 66-72, 1997.

[28] M. Ball, and M. Magazine, The Design and Analysis of Heuristics. *Networks*, Vol. 11, No. 2, pp. 215-219, 1981.

[29] J.F. Bard, and A. Feinberg, A Two-Phase Methodology for Technology Selection and System Design, *IEEE Transactions on Engineering Management*, Vol. 36, No. 1, pp. 28-36, 1989.

[30] J.-P. Barthélemy, and A. Guenoche, *Trees and Proximity Representation*, Chichester: J.Wiley & Sons, 1991.

[31] J.-P. Barthélemy, B. Leclerc, and B. Monjardet, On the Use of Ordered Sets in Problems of Comparison and Consensus of Classifications. *J. of Classification*, Vol. 3, No. 2, 17-24, 1986.

[32] L. Bass, and J. Coutaz, *Developing Software for the User Interface. Series in Software Engineering.* MA: Addison Wesley, Reading, 1991.

[33] L. Bass, J. Coutaz, and C. Unger, A Reference Model for Interactive System Construction. In: *Proc. of Intl. Conf. on Human-Computer Interaction EWHCI'92*, Moscow, Part 1, pp. 23-30, 1992.

[34] R.E. Bellman, and L.A. Zadeh, Decision-Making in Fuzzy Environment. *Management Science*, Vol. 17, No. 4, 141-164, 1970.

[35] A.R. Belkin, and M.Sh. Levin, *Decision Making: Combinatorial Models of Information Approximation.* Moscow: Nauka, 1990 (in Russian).

[36] I.S. Belov, and Ya.N. Stolin, An Algorithm in a Single-Path Operations Scheduling Problem. In: *Mathematical Economics and Functional Analysis*, Moscow: Nauka, pp. 244-253, 1974 (in Russian).

[37] I. Benbasat, and P. Todd, An Experimental Investigation of Interface Design Alternatives: Icon vs. Text and Direct Manipulation vs. Menus. *Int. J. Man-Machine Studies*, Vol. 38, pp. 369-402, 1993.

[38] R.G. Bennetts, *Design of Testable Logic Circuits.* London: Addison-Wesley, 1984.

[39] B.A. Berezovskii, and L.A. Kontorer, Efficient Algorithm to Construct the Set of Undominated Alternatives of Decomposable Objects. *Autom. and Rem. Control,* Vol. 48, No. 1, Part 2, pp. 95-100, 1987.

[40] C. Berge, *Graphs and Hypergraphs.* Amsterdam: North-Holland, 1973.

[41] O. Berman, and N. Ashrafi, Optimization Models for Reliability of Modular Software Systems. *IEEE Trans. on Software Engineering,* Vol. 19, No. 11, pp. 1119-1123, 1993.

[42] O. Berman, D.I. Ingco, and A. Odoni, Improving the Location of Minimax Facilities Through Network Modification. *Networks*, Vol. 24, pp. 31-41, 1994.

[43] P. Berman, and V. Ramaiye, An Approximation Algorithm for the Steiner Tree Problem. *J. of Algorithms*, Vol. 17, pp. 381-408, 1994.

[44] M.A. Bermant, and A.K. Vaytkin, Modeling of Profile for Specialist Preparation. *Avtomatika i Telemekhanika*, Vol. 10, pp. 95-103, 1978 (in Russian).

[45] M.P. Bieber, Automating Hypermedia for Decision Support, *Hypermedia,* Vol. 4, No. 2, pp. 83-110, 1992.

[46] D. Bienstock, E.F. Brickell, and C.L. Monma, On the Structure of Minimum-Weight k-connected Spanning Networks. *SIAM J. on Discrete Mathematics*, Vol. 3, No. 3, pp. 329-329, 1990.

[47] W.P. Birmingham, A.P. Gupta, and D.P. Siewiorek, The Micon System for Computer Design. *IEEE Micro.* Vol. 9, No. 5, pp. 61-67, 1989.

[48] W. Blades, and G. Cotter, Modernization and Unification: The Strategic Goals for NASA STI Program. In: *Proc. Intl. Conf. on Inform. Technology and People ITAR'93*, Moscow, Vol.I, pp. 18-29, 1993.

[49] R. Blanning, Model Management Systems: An Overview. *Decision Support Systems.* Vol. 9, No. 1, pp. 9-18, 1993.

[50] J. Blazewicz, W. Cellary, R. Slowinski, and J. Weglarz, *Scheduling under Resource Constraints: Deterministic Models.* Basel: Baltzer, 1986.

[51] J. Blazewicz, K.H. Ecker, G. Schmidt, and J. Weglarz, *Scheduling in Computer and Manufacturing Systems,* 2nd ed., Berlin: Springer-Verlag, 1994.

[52] F. Bodart, A.-M. Hennebert, J.M. Leheureux, I. Sacre, and J. Vanderdonckt, Architecture Elements for Highly-Interactive Business- Oriented Applications. In: L.E. Bass, J. Gornostaev, and C. Unger (Eds.) *Human-Computer Interaction.* Lecture Notes in Computer Science, Vol. 753, Berlin: Springer-Verlag, pp. 83-104, 1993.

[53] L. Bodin, and B. Golden, Classification in Vehicle Routing and Scheduling. *Networks*, Vol. 11, No. 2, pp. 97-108, 1981.

[54] H.L. Bodlaender, K. Jansen, and G.J. Woeginger, Scheduling with Incompatible Jobs, Report No. 203, Institute for Mathematics, Technical University of Graz, January 1992.

[55] K.P. Bogart, Preference Structures I: Distance between Transitive Preference Relations. *J. Math. Soc.*, Vol. 3, pp. 49-67, 1973.

[56] G. Booch, *Object Oriented Development.* Redwood City, California: The Benjamin / Cummings Publishing Company, 1991.

[57] S.A. Boorman, and D.C. Oliver, Metrics on Spaces of Finite Trees. *J. of Math. Psychology*, Vol. 10, No. 1, pp. 26-59, 1973.

[58] R.B. Borie, Generation of Polynomial-Time Algorithms for Some Optimization Problems on Tree-Decomposable Graphs. *Algorithmica,* Vol. 14, No. 2, pp. 123-137, 1995.

[59] R.A. Botafogo, E. Rivlin, and B. Shneiderman, Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics. *ACM Trans. on Information Systems,* Vol. 10, No. 2, pp. 83-110, 1992.

[60] D.E. Boyce, A. Farni, and R. Weischedel, *Optimal Subset Selection: Multiple Regression, Interdependence and Optimal Network Algorithms.* LNEMS, Vol. 103, Berlin: Springer-Verlag, 1974.

[61] M.L. Brandeau, and S.S. Chiu, An Overview of Representative Problems in Location Research. *Management Science*, Vol. 35, No. 6, pp. 645-674, 1989.

[62] S. Brinkkemper, Method Engineering: Engineering of Information Systems Development Methods and Tools. In: *Proc. of 5th Intl. Conf. on Information Systems Development*, Gdansk, pp. 69-78, 1996.

[63] D.J. Brooks, and C.W. Kirkwood, Decision Analysis to Select a Microcomputer Networking Strategy: A Procedure and a Case Study. *J. of the Operational Research Society*, Vol. 39, No. 1, pp. 23-32, 1988.

[64] F.P. Brooks, Jr, *The Mythical Man-Month. Essays on Software Engineering*, Mass.: Addison-Wesley Publishing Co., Reading, 1975.

[65] D.M. Buede, Software Review. Overview of MCDA Software Market. *J. of Multi-Criteria Decision Analysis*, Vol. 1, No. 1, pp. 59-61, 1992.

[66] D.M. Buede, and R.W. Choisser, Providing an Analytical Structure for Key System Choices. *J. of Multi-Criteria Decision Analysis*, Vol. 1, No. 1, pp. 17-27, 1992.

[67] H. Buer, and R.H. Mohring, A Fast Algorithm for the Decomposition of Graphs and Posets. *Mathematics of Operations Research*, Vol. 8, pp. 170-184, 1983.

[68] R. Busacker, and T.L. Saaty, *Finite Graphs and Networks*. New York: McGraw-Hill, 1965.

[69] R. Camposano, From Behavior To Structure: High-Level Synthesis. *IEEE Design & Test of Computers,* October, pp. 8-19, 1990.

[70] G.J. Chang, and J. Edmonds, The Poset Scheduling Problem. *Order*, Vol. 2, No. 2, pp. 113-118, 1985.

[71] R.L. Carraway, and R.L. Schmidt, An Improved Discrete Dynamic Programming Algorithm for Allocating Resources Among Interdependent Projects. *Manag. Sci.*, Vol. 37, No. 9, pp. 1195-1200, 1991.

[72] J.M. Carrol, and J. Reitman-Olson (Eds.), *Mental Models in Human-Computer Interaction: Research Issues About What the User of Software Knows*, Washington, D.C.: National Academy Press, 1987.

[73] B. Chaib-draa, J. Desharnais, and S. Lizotte, A Relation Graph Formulation for Relationships Among Agents. In: *Papers from the Thirteenth*

Intl. Workshop "Distributed Artificial Intelligence", TR WS-94-02, Menlo Park, Ca.: AAAI Press, pp. 17-31, 1994.

[74] V. Chankong, and Y.Y. Haimes, *Multiobjective Decision Making: Theory and Methodology.* New York, Elsevier, North Holland Publishing, 1983.

[75] A. Chaudhury, P.-S. Deng, and S. Rathnam, A Computational Model of Coordination. *IEEE Trans. of Systems, Man, and Cybernetics. Part A: Systems and Humans*, Vol. 26, No. 1, pp. 132-141, 1996.

[76] P.Yu. Chebotarev, Aggregation of Preferences by the Generalized Row Sum Method. *Mathematical Social Sciences*, Vol. 27, pp. 293-320, 1994.

[77] H. Chen, and K.J. Lynch, Automatic Construction of Networks of Concepts Characterizing Document Databases. *IEEE Trans. of Systems, Man, and Cybernetics*, Vol. 22, No. 5, pp. 885-902, 1992.

[78] H. Chen, K.J. Lynch, K. Basu, and N.T. Hg, Generating, Integrating, and Activating Thesauri for Concept-Based Document Retrieval. *IEEE Expert*, Vol. 8, No. 2, pp. 25-34, 1993.

[79] H. Chen, B. Schatz, T. Hg., J. Martinez, A. Kirchhoff, and C. Lin, A Parallel Computing Approach to Creating Engineering Concept Spaces for Semantic Retrieval: The Illinois Digital Library Initiative Project. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 8, pp. 771-782, 1996.

[80] B.H.C. Cheng, and J.-J. Jeng, Reusing Analogous Components. *IEEE Trans. on Knowledge and data Engineering*, Vol. 9, No. 2, pp. 341-349, 1997.

[81] S. Chopra, and M.R. Rao, The Steiner Tree Problem I: Formulations, Compositions and Extension Facets. Mathematical Programming, Vol. 64, pp. 209-229, 1994.

[82] C.W. Churchman, *The System Approach*, New York: Dell, 1968.

[83] K. Clark, D. Ford, and M. Saren, Company Technology Strategy. *R & D Management*, Vol. 19, p. 215, 1989.

[84] E.J. Cockane, and D.E. Hewgill, Exact Computation of Steiner Minimal Trees in the Plane. *Information Processing Letters*, Vol. 22, No. 3, pp. 151-156, 1986.

[85] E.G. Coffman, Jr. (Ed.), *Scheduling in Computer and Job Shop Systems,* New York: J.Wiley & Sons, 1976.

[86] R.C. Conant, Information Flows in Hierarchical Systems, *Intl. J. of General Systems*, Vol. 1, No. 1, pp. 9-18, 1974.

[87] J. Conklin, Hypertext: An Introduction and Survey, *IEEE Computer*, Vol. 20, No. 9, pp. 17-41, 1987.

[88] R.W. Conway, W.L. Maxwell, and L.W. Miller, *Theory of Scheduling.* Mass.: Addison-Wesley, Reading, 1967.

[89] W.D. Cook, and M. Kress, Relationships between l^1 Metrics on Linear Rankings Space. *SIAM J. on Appl. Math.*, Vol. 44, No. 1, pp. 209-220, 1984.

[90] W.D. Cook, and M. Kress, *Ordinal Information and Preference Structures: Decision Models and Applications*, NJ: Prentice Hall, Englewood Cliffs, 1992.

[91] W.D. Cook, D.A. Johnston, and M. Kress, Selecting a New Technology Strategy Using Multiattribute, Multilevel Decision Trees. *J. of Multi-Criteria Decision Analysis*, Vol. 2, pp. 129-144, 1993.

[92] M.D. Cooper, *Design of Library Automation Systems. File Structures, Data Structures, and Tools.* New York: J.Wiley & Sons, 1996.

[93] A. Court, S. Culley, and C. McMahon, The Information Requirements of Engineering Designers, *Engineering Designer*, Vol. 20, No. 1, pp. 16-17, 1994.

[94] R.D. Coyne, M.A. Rosenman, A.D. Radford, and J.S. Gero, Innovation and Creativity in Knowledge-Based CAD. In: J.S.Gero (Ed.) *Expert Systems in Computer-Aided Design*, Elsevier Science Publishers B.V., IFIP, 1987.

[95] J. Crinnion, *Evolutionary System Development. A Practical Guide to the Use of Prototyping within a Structured Systems Methodology.* New York: Plenum Press, 1992.

[96] D.E. Critchlow, *Metric Methods For Analyzing Partially Ranked Data.* Berlin: Springer-Verlag, 1985.

[97] W.B. Croft, Approaches to Intelligent Information Retrieval. *Information Processing & Management*, Vol. 23, No. 4, pp. 249-254, 1987.

[98] N. Cross, Styles of Learning, Designing and Computing. *Design Studies*, Vol. 6, No. 3, pp. 157-162, 1985.

[99] J. Current, H. Min, and D. Schilling, Multiobjective Analysis of Facility Location Decisions. *European J. of Operational Research*, Vol. 49, pp. 295-300, 1990.

[100] K.W. Cutright, R. Williams, and D. Debald, Design of a PC-Based Expert System for Academic Advising. *Computers & Industrial Engineering*, Vol. 21, No. 1-4, pp. 423-427, 1991.

[101] A.F. Czajkowski, and S. Jones, Selecting Interrelated R & D Projects in Space Technology Planning, *IEEE Transactions on Engineering Management*, Vol. 33, No. 1, pp. 17-24, 1986.

[102] J. D'Ambrosio, T. Darr, and W. Birmingham, Hierarchical Concurrent Engineering in a Multiagent Framework. *Concurrent Engineering: Research and Applications,* Vol. 4, No. 1, pp. 47-57, 1996.

[103] D. Das, S.K. Gupta, and D.Nau, Reducing Setup Cost by Automatic Generation of Redesign Suggestions. In: *Proc. of ASME Conference "Computers in Engineering"*, 1994.

[104] M.S. Daskin, *Networks and Discrete Location. Models, Algorithms, and Applications.* New York: J.Wiley & Sons, 1995.

[105] C.J. Date, *An Introduction to Database Systems.* New York: Addison-Wesley, 1995.

[106] S. Dausere-Peres, and J.-B. Lassere, *An Integrated Approach in Production Planning and Scheduling.* Lecture Notes in Economics and Mathematical Systems, Vol. 411, Berlin: Springer-Verlag, 1994.

[107] W.H.E. Day, Optimal Algorithms for Comparing Trees with Labeled Leafs. *J. of Classification*, Vol. 2, No. 1, pp. 7-28, 1985.

[108] M. Debevc, D. Donlagic, R. Svecko, J. Spasovski and B. Meyer, Flexible User Support with Adaptive Bar, in: *Proc. of Intl. Conf. on Human-Computer Interaction EWHCI'94*, St. Petersburg, Vol. 1, pp. 129-134, 1994.

[109] R. Dechter, Network Based Heuristic for Constraint Satisfaction Problems. *Artificial Intelligence*, Vol. 34, No. 1, pp. 1-38, 1988.

[110] R. Dechter, I. Meiri, and J. Pearl, Temporal Constraint Networks. *Artificial Intelligence*, Vol. 49, No. 1-3, pp. 61-95, 1991.

[111] K.S. Decker, and V.R. Lesser, Generalizing the Partial Global Planning Algorithm.]i Intl. J. of Intelligent and Cooperative Systems, Vol. 1, No. 2, pp. 319-346, 1992.

[112] K.S. Decker, and V.R. Lesser, Designing a Family of Coordination Algorithms. In: *Papers from the Thirteenth Intl. Workshop "Distributed Artificial Intelligence"*, TR WS-94-02, Menlo Park, Ca.: AAAI Press, pp. 32-51, 1994.

[113] G.H. Demes, S.J. Fenves, I.E. Grossmann, C.T. Hendrickson, T.M. Mitchell, F.B. Prinz, D.P. Sewiorek, E. Subrahmanian, S. Talukdar, and A.W. Westerberg, The Engineering Design Research Center of Carnegie Mellon University. In: *Proc. of the IEEE,* Vol 81, No. 1, pp. 10-23, 1993.

[114] M.A.H. Dempster, M.L. Fisher, L. Jansen, B.J. Lageweg, J.K. Lenstra, and A.H.G. Rinnooy Kan, "Analytical Evaluation of Hierarchical Planning Systems". *Operations Research*, Vol. 29, No. 4, pp. 707-716, 1981.

[115] R.L. De Neufville, and D. Marks, (Eds.), *Systems Planning and Design Case Studies in Modeling, Optimization, and Evaluation.* NJ: Prentice Hall, Englewood Cliffs, 1974.

[116] G. De Santis, and R.B. Gallupe, A Foundation for the Study of Group Decision Support Systems. *Management Science*, Vol. 33, No. 5, pp. 589-609, 1987.

[117] J.H. Dinitz, and D.R. Stinson, (Eds.), *Contemporary Design Theory*, New York: J.Wiley & Sons, 1992.

[118] S.W. Director et al, A Design Methodology and Computer Aids for Digital VLSI Systems. *IEEE Trans. on Circuits and Systems,* Vol. 28, No. 7, pp. 634-645, 1981.

[119] J.R. Dixon, *Design Engineering: Inventiveness, Analysis and Decision Making.* New York: McGraw-Hill, 1966.

[120] J.R. Dixon, On Research Methodology Towards a Scientific Theory of Engineering Design. *AI EDAM*, Vol. 1, No. 3, pp. 145-157, 1987.

[121] D.R. Dolk and B.R. Konsynski, Knowledge Representation for Model Management Systems. *IEEE Trans. on Software Engineering*, Vol. 10, No. 6, pp. 619-628, 1984.

[122] S.E. Dreyfus, and R.A. Wagner, The Steiner Problem in Graphs. *Networks*, Vol. 1, p. 195-207, 1972.

[123] D.-Z. Du, On Greedy Heuristics for Steiner Minimum Trees. *Algorithmica*, Vol. 13, No. 4, pp. 381-385, 1995.

[124] D.-Z. Du, and D.F. Hsu, *Combinatorial Network Theory*, Boston: Kluwer Academic Publishers, 1996.

[125] D. Dubois, and H. Prade, *Possibility Theory: An Approach to Computerized Processing of Uncertainty*, New York: Plenum Press, 1988.

[126] D. Dubois, H. Fargier, and H. Prade, Possibility Theory in Constraint Satisfaction Problems: Handling Priority, Preference and Uncertainty. *Applied Intelligence*, Vol. 6, No. 4, pp. 287-309, 1996.

[127] Yu.A. Dubov, S.I. Travkin, and V.N. Iakimets, *Multiobjective Models for Formation and Choice of Systems Variants.* Moscow: Nauka, 1986 (in Russian).

[128] E.H. Durfee, V.R. Lesser, and D.D. Corkill, Trends in Cooperative Distributed Problem Solving. *IEEE Trans. on Knowledge and Date Engineering*, Vol. 1, No. 1, pp. 63-83, 1989.

[129] J.S. Dyer, P.C. Fishburn, R.E. Steuer, J. Wallenius, and S. Zionts, Multiple Criteria Decision Making, Multi-attribute Utility Theory: The Next Ten Years. *Manag. Sci.*, Vol. 38, No. 5, pp. 645-654, 1992.

[130] C.M. Eastman, Recent Development in Representation on the Science of Design. *Design Science*, Vol. 3, No. 1, pp. 45-52, 1982.

[131] C.M. Eastman, A.H. Bond, and S.C. Chase, Application and Evaluation of an Engineering Data Model. *Research in Engineering Design*, Vol. 2, No. 4, pp. 185-207, 1991.

[132] K.L. Edwards, Improved Design through Guidelines Support, *Engineering Designer*, Vol. 19, No. 5, pp. 22-23, 1993.

[133] T. Elpern, I. Weissberg, E. Zahavi, Machine Design Optimization by the Monte Carlo Annealing Method. *Engineering Optimization*, Vol. 15, No. 3, pp. 193-203, 1990.

[134] J.L. England, Hierarchies, Theories, and Methodologies. In: R. Trappl (ed.) *Cybernetics and Systems'88*, Kluwer Academic Publishers, pp. 271-278, 1988.

[135] S.D. Eppinger et al., Organizing the Tasks in Complex Design Projects. In: Sriram, D., Logcher, R., and Fukuda, S. (Eds.) *Computer-Aided Cooperative Product Development*, LNCS, Vol. 492, New York: Springer-Verlag, pp. 230-252, 1991.

[136] K. Erol, D. Nau, and J. Hendler, HTN Planning: Complexity and Expressivity. In: *AAAI-1994*, Seattle, July 1994.

[137] K. Erol, J. Hendler, and D. Nau, Complexity Results for HTN Planning. *Annals of Mathematics and Artificial Intelligence*, 1995.

[138] H.A. Eschenauer, Structural Optimization - a Need in Design Process ? In: *Engineering Optimization in Design Process*, H.A.Eschenauer, C.Mattheck, N.Olhoff (Eds.), Lecture Notes in Engineering 63, Berlin: Springer-Verlag, pp. 1-13, 1991.

[139] J.R. Evans, A Review and Synthesis of OR/MS and Creative Problem Solving (Part 1 and 2). *OMEGA Intl. J. of Management Science*, Vol. 17, No. 6, pp. 499-524, 1989.

[140] S. Even, A. Itai, and A. Shamir, On the Complexity of Timetable and Multicommodity Problems. *SIAM J. Computing*, Vol. 5, pp. 691-703, 1976.

[141] B.H. Faaland, The Multiperiod Knapsack Problem. *Operations Research,* Vol. 29, No. 3, pp. 612-616, 1981.

[142] K.-J. Farn, and D.-H. Lo, A Method for Selection of Enterprise Information System Based on the Combination of the Analytic Hierarchy Process and the Information System Management Planning. In: *Proc. of the Tenth Int. Conf. on MCDM*, Taiwan, Vol. 3, pp. 221-236, 1992.

[143] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Comm. of the ACM*, Vol. 39, No. 11, pp. 27-34, 1996.

[144] J. Fedorovicz and G.B. Williams, Representing Modeling Knowledge in an Intelligent Decision Support System, *Decision Support Systems*, Vol. 2, No. 1, pp. 3-14, 1986.

[145] K. Fedra, C. Zhao, and L. Winkelbauer, Interactive Multicriteria Decision Support: Combining Rule-based and Numerical Approaches. In: L. Wessels and A.P. Wierzbicki A.P (eds.), *User Oriented Methodology and Techniques of Decision Analysis and Support.* LNEMS, Vol. 397, Vienna: Springer-Verlag, pp. 48-64, 1993.

[146] W. Feller, *An Introduction to Probability Theory and Its Applications*, Vol. 1 and 2, New York: J.Wiley & Sons, 1971.

[147] S. Finger, M.S. Fox, F.B. Prinz, and J. Rinderle, Concurrent Design. *Applied Artificial Intelligence*, Vol. 6, pp. 257-283, 1992.

[148] B. Fischhoff, and S. Johnson, The Possibility of Distributed Decision Making. In: *Distributed Decision Making*. Washington: National Academy Press, pp. 25-61, 1990.

[149] G. Fischer, Cognitive View of Reuse and Redesign, *IEEE Software*. pp. 60-72, July 1987.

[150] P.C. Fishburn, *Utility Theory for Decision Making,* New York: J.Wiley & Sons, 1970.

[151] D.L. Fisher, Optimal Performance Engineering: Good, Better, Best. *Human Factors*, Vol. 35, No. 1, pp. 115-139, 1993.

[152] L.R. Ford, and D.R. Fulkerson, *Flows in Networks*, Princeton: Princeton University Press, 1962.

[153] M.S. Fox, An Organizational View of Distributed Systems. *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 11, No. 1, pp. 70-80, 1981.

[154] M. French, *Invention and Evolution. Design in Nature and Engineering*. 2nd edition, Cambridge: Cambridge University Press, 1994.

[155] S. French, *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-shop*, New York: J.Wiley & Sons, 1982.

[156] H.N. Gabow, Two Algorithms for Generating Weighted Spanning Trees in Order. *SIAM J. on Computing*, Vol. 6, No. 1, pp. 139-150, 1977.

[157] E. Gallopoulos, E.N. Houstis, and J.R. Rice, Workshop on Problem-Solving Environments: Findings and Recommendations. *ACM Computing Surveys*, Vol. 27, No. 2, pp. 277-279, 1995.

[158] B. Ganter, and R. Wille, Conceptual Scaling. In: *Applications of Combinatorics and Graph Theory to the Biological and Social Sciences*. Roberts F.S. (ed.), New York: Springer-Verlag, pp. 140-167, 1989.

[159] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, and J. Widom, The TSIMMIS Approach to Mediation: Data Models and Languages. *NGITS Symposium,* Naharia, Israel, June 1995.

[160] M.R. Garey, and D.S. Johnson, *Computers and Intractability. The Guide to the Theory of NP-Completeness*. San Francisco: W.H.Freeman and Company, 1979.

[161] D. Garlan, C.W. Krueger, and B.S. Lerner, TransformGen: Automating the Maintenance of Structured Environments. *ACM Trans. on Programming Languages and Systems*, Vol. 16, No. 3, pp. 727-774, 1994.

[162] I. Gati, N. Fassa, and D. Houminer, Applying Decision Theory to Career Counseling Practice: The Sequential Elimination Approach. *The Career Development Quarterly*, Vol. 43, No. 3, pp. 211-220, 1995.

[163] T.A. Gavrilova and E.V. Zudilova, Conceptual Design of Interfaces for Knowledge Acquisition Systems, *Engineering Cybernetics*, No. 2, pp. 3-11, 1994 (in Russian).

[164] A.M. Geoffrion, An Introduction to Structured Modeling. *Management Science*, Vol. 33, No. 5, pp. 547-588, 1987.

[165] J.S. Gero, (Ed.), *Design Optimization*. New York: Academic Press, 1985.

[166] J.S. Gero, (Ed.), *Optimization in Computer-Aided Design*. New York: Elsevier, 1985.

[167] J.S. Gero, (Ed.), *Expert Systems in Computer-Aided Design*. Amsterdam: North Holland, 1987.

[168] J.S. Gero, (Ed.), *Artificial Intelligence in Engineering: Design*. Amsterdam: Elsevier, 1988.

[169] J.S. Gero, (Ed.), *Applications of Artificial Intelligence in Engineering V*, Vol I: Design, Berlin: Springer-Verlag, 1990.

[170] J.S. Gero, M.L. Maher, and F. Sudweeks, (Eds.), *Computational Models of Creative Design*. Preprints of the Third Intl. Conf. on Computational Models of Creative Design, University of Sydney, 1995.

[171] D. Ghosh, and R. Agarwal, Model Selection and Sequencing in Decision Support Systems. *OMEGA*, Vol. 19, No. 2/3, pp. 157-167, 1991.

[172] P.C. Gilmore, and R.E. Gomory, Sequencing a One State-variable Machine: A Solvable Case of the Traveling Salesman Problem. *Operations Research*, Vol. 12, pp. 655-679, 1964.

[173] R.K. Gjertsen, Jr., M.T. Jones, and P.E. Plassmann, Parallel Heuristics for Improved, Balanced Graph Colorings. *J. of Parallel and Distributed Computing*, Vol. 37, No. 2, pp. 171-186, 1996.

[174] M.X. Goemans, and D.P. Williamson, A General Approximation Technique for Constrained Forest Problems. *SIAM J. on Computing*, Vol. 24, No. 2, pp. 296-317, 1995.

[175] D.E. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning.* Mass: Addison-Wesley, Reading, 1989.

[176] B.I. Goldengorin, *Requirements of Standards: Optimization Models and Algorithms.* Hoogezand, The Netherlands: Operations Research Co., 1995.

[177] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, New York: Academic Press, 1980.

[178] M.C. Golumbic, M. Markovich, S. Tsur, and U.J. Schild, A Knowledge-Based Expert System for Student Advising. *IEEE Trans. on Education*, Vol. 29, No. 2, pp. 120-124, 1986.

[179] I.R. Goodman, and H.T. Nguyen, Uncertainty Models for Knowledge-Based Systems, Amsterdam: North-Holland, 1986.

[180] E.N. Gordeev, and O.G. Taraszow, Steiner Problem. Survey. *Discrete Mathematics*, Vol. 5, No. 2, pp. 3-28, 1993 (in Russian).

[181] A.D. Gordon, Consensus Supertrees: The Synthesis of Rooted Trees Containing Overlapping Sets of Labeled Leafs. *J. of Classification*, Vol. 3, No. 2, pp. 335-348, 1986.

[182] J.C. Gower, Measures of Similarity, Dissimilarity, and Distance. In: S. Kotz, and N.L. Johnson, (Eds.) *Encyclopedia of Statistical Sciences*, Vol. 5, New York: J.Wiley & Sons, pp. 397-405, 1985.

[183] J. Griese, and R. Kurpicz, Investigating the Buying Process for the Introduction of Data Processing in Small and Medium-sized Firms, *Information & Management*, Vol. 8, pp. 41-51, 1985.

[184] A. Griffin, and J.P. Hauser, Pattern of Communication Among Marketing, Engineering and Manufacturing - A Comparison Between Two New Product Teams. *Management Science,* Vol. 38, No. 2, pp. 36-373, 1992.

[185] D. Grosh and R. Agarwal, Model Selection and Sequencing in Decision Support Systems, *OMEGA*, Vol. 19, No. 2-3, pp. 157-167, 1991.

[186] I.E. Grossmann, Mixed-Integer Non-linear Programming Techniques for the Synthesis of Engineering Systems. *Research in Engineering Design,* Vol. 1, No. 2/3, pp. 205-228, 1990.

[187] H. Gunadhi, K.-H. Lim, and W.-Y. Yeong, PACE: A Planning Advisor on Curriculum and Enrollment. In: *Proc. of 28th Annual Hawaii Intl. Conf.*

on System Sciences, Vol. III "Information Systems - Decision Support and Knowledge Based Systems", J.F. Nunamaker and R.H.Sprague, Jr. (eds.), Los Alamitos: IEEE Computer Society Press, pp. 23-31, 1995.

[188] A. Gunasekaran, A.R. Korukonda, I. Virtanen, and P. Yli-Olli, Improving Productivity and Quality in Manufacturing Organizations. *Intl. J. of Production Economics*, Vol. 36, No. 2, pp. 169-183, 1994.

[189] A.P. Gupta, W.P. Birmingham, and D.P. Siewiorek, Automating the Design of Computer Systems. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems,* Vol. 12, No. 4, pp. 473-487, 1993.

[190] D. Gusfield, and R.W. Irving, *The Stable Marriage Problem: Structure and Algorithms* Boston: The MIT Press, 1989.

[191] R.D. Hackathorn, and J. Karimi, A Framework for Comparing Information Engineering Methods. *MIS Quarterly*, Vol. 12, No. 2, pp. 203-220, 1989.

[192] S.L. Hakimi, Steiner Problem in Graphs and Its Implications. *Networks*, Vol. 2, pp. 113-133, 1971.

[193] C. Hales, Designer as Chameleon. *Design Studies*, Vol.6, No. 2, pp. 111-114, 1985.

[194] M. Hall, Jr., *Combinatorial Theory,* 2nd ed., New York, J.Wiley & Sons, 1986.

[195] S. Hamilton, and N.L. Chervany, Evaluating Information System Effectiveness - Part II: Comparing Evaluator Viewpoints. *MIS Quarterly*, Vol. 5, No. 4, pp. 79-86, 1981.

[196] M. Hammer, and J. Champy, *Reengineering the Corporation: A Manifesto for Business Revolution*, New York: Harper Collins, 1993.

[197] G.Y. Handler, and P.B. Mirchandani, *Location on Networks: Theory and Algorithms.* Cambridge, MA: MIT Press, 1979.

[198] S. Hannenhalli, and P. Pevzner, *Transforming Men into Mice*, Technical Report CSE-95-012. Dept. of Computer Science and Engineering, Pennsylvania State University, 1995.

[199] P. Hansen, B. Jaumard, and G. Savard, New Branch-and-Bound Rules for Linear Bilevel Programming. *SIAM J. on Scientific and Statistical Computing*, Vol. 13, pp. 1194-1217, 1992.

[200] F. Harary, *Graph Theory.* Mass.: Addison-Wesley, Readings, 1969.

[201] F. Harary, R.Z. Norman, and D. Carthwright, *Structural Models: An Introduction to the Theory of Directed Graphs.* New York: J.Wiley & Sons, 1965.

[202] G. Harhalakis, C.P. Lin, R. Nagi, and J.M. Proth, Hierarchical Decision Making in Computer Integrated Manufacturing Systems. In: *Proc. of the Third Inlt. Conf. on CIM.* CS Press, pp. 15-24, 1992.

[203] F. Hayes-Roth, Towards Benchmarks for Knowledge Systems and Their Implications for Data Engineering. *IEEE Trans. on Knowledge and Data Engineering*, Vol. 1, No. 1, pp. 101-110, 1989.

[204] S. Hayne, and S. Ram, Group Data Base Design: Addressing the View Modeling Problem, *J. of Systems and Software,* Vol. 28, No. 2, pp. 98-115, 1995.

[205] G. Held, *Network Management (Techniques, Tools and Systems).* New York: J.Wiley & Sons, 1992.

[206] A.A. Helwig, and P.J. Snodgres, Computer-Assisted Career Guidance Use in Higher Education. *Journal of Career Development*, Vol. 16, pp. 249-259, 1990.

[207] M.D. Hendy, C.H.C. Little, and D. Penny, Comparing Trees with Pendant Vertices Labelled. *SIAM J. on Appl. Math.*, Vol. 44, No. 5, pp. 1054-1065, 1984.

[208] D.M. Himmelblau, Morphology of Decomposition. In: Himmelblau, D.M. (Ed.). *Decomposition of Large-Scale Problem.* New York: North-Holland, pp. 1-13, 1973.

[209] D. Hirschberg, and G. Myers, (eds.), *Combinatorial Pattern Matching.* LNCS, Vol. 1075, Berlin: Springer-Verlag, 1996.

[210] D. Hix, Generations of User-Interface Management Systems. *IEEE Computer*, Vol. 5, No. 5, pp. 77-87, 1990.

[211] J.H. Holland, *Adaptation in Natural and Artificial Systems.* Ann Arbor, Michigan: Univ. of Michigan Press, 1975.

[212] R. Horst, and P.M. Pardalos, *Handbook of Global Optimization*, Dordrecht-Boston-London: Kluwer Academic Publishers, 1995.

[213] R.A. Howard, Decision Analysis: Practice and Promise. *Management Science*, Vol. 34, No. 6, pp. 679-696, 1988.

[214] C. Hubel, D. Ruland, and E. Siepmann, On Modeling Integrated Design Environments. In: *Proc. of European Design Automation Conference EURO-DAC'92*, Los Alamitos: IEEE CS Press, pp. 452-458, 1992.

[215] G.P. Huber, Issues in the Design of Group Decision Support Systems. *MIS Quarterly*, Vol. 8, No. 3, pp. 195-204, 1984.

[216] L. Hubert, and P. Arabie, Comparing Partitions. *J. of Classification*, Vol. 2, No. 3, pp. 193-218, 1985.

[217] V. Hubka, and W.E. Eder, A Scientific Approach to Engineering Design. *Design Studies*, Vol. 8, No. 3, pp. 123-137, 1987.

[218] V. Hubka, and W.E. Eder, *Theory of Technical Systems. A Total Concept Theory for Engineering Design.* Berlin: Springer-Verlag, 1988.

[219] M.N. Huhns, and D. Bridgeland, Multiagent truth Maintenance. *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 21, No. 6, 1437-1445, 1991.

[220] M.N. Huhns, M.P. Singh, T. Ksiezyk, and N. Jacobs, Global Information Management via Local Autonomous Agents. In: *Papers from the Thirteenth Intl. Workshop "Distributed Artificial Intelligence"*, TR WS-94-02, Menlo Park, Ca.: AAAI Press, pp. 120-141, 1994.

[221] T. Ichimori, H. Ishii, and T. Nishida, Algorithms for One Machine Job Sequencing with Precedence Constraints. *J. of the Operations Research Society of Japan*, Vol. 24, pp. 159-169, 1981.

[222] J.P. Ignizio, A Generalized Goal Programming Approach to the Minimal Interference Multicriteria $N \times 1$ Scheduling Problem. *IIE Trans.*, Vol. 16, No. 4, pp. 316-322, 1984.

[223] K. Ishii, R. Adler, and P. Barkan, Knowledge-based Simultaneous Using Design Compatibility Analysis. In: Gero, J.S. (ed.) *Artificial Intelligence in Engineering: Design.* Amsterdam: Elsevier Science Publisher B.V., pp. 361-377, 1988.

[224] K. Ishii, R.A. Miller, and L.R. Maloney, Compatibility-Based Concurrent Engineering for Forging. In: G. Rzevski, (ed.). *Applications of Artificial Intelligence in Engineering V*, Vol.2: Manufacture and Planning. Berlin: Springer-Verlag, pp. 343-359, 1990.

[225] E.M. Jawhari, M. Pouzet, and D. Missane, Retracts: Graphs and Ordered Sets From The Metrics Point of View. In: *Contemporary Mathematics*, Vol. 57, Rival, I. (ed.), Providence: American Mathematical Society, 1986.

[226] T.R. Jensen, and B. Toft, *Graph Coloring Problems*. New York: J.Wiley & Sons, 1995.

[227] R. Jeroslow, The Polynomial Hierarchy and a Simple Model for Competitive Analysis. *Mathematical Programming*, Vol. 32, pp. 146-164, 1985.

[228] D.S. Johnson, and K.A. Niemi, On Knapsacks, Partitions, and a New Dynamic Programming Techniques for Trees. *Mathematics of Operations Research*, Vol. 8, No. 1, pp. 1-14, 1983.

[229] D.S. Johnson, and M.A. Trick, (Eds.) *Cliques, Coloring, and Satisfiability*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 26, Providence: American Mathematical Society, 1996.

[230] J.A. Johnson, Rough Set Schemas. In: *Proc. of 5th Intl. Conf. Information Systems Development ISD'96*, Gdansk, pp. 193-203, 1996.

[231] P. Johnson, *Human Computer Interaction. Psychology, Task Analysis and Software Engineering*. London: McGraw-Hill, 1992.

[232] S.C. Johnson, Hierarchical Clustering Schemes. *Psychometrika*, Vol. 32, No. 3, pp. 241-254, 1967.

[233] S.M. Johnson, Optimal Two-and-Three-Stage Production Schedules. *Naval Res. Logist. Quart.*, Vol. 1, No. 1, pp. 61-68, 1954.

[234] H. Jones, and B.C. Twiss, *Forecasting Technology of Planning Decisions*. New York: PBI, 1978.

[235] J.C. Jones, *Design Methods*. New York: J.Wiley & Sons, 1981.

[236] C. Joslyn, Possibilistic Semantic and Measurement Methods in Complex Systems. In: *Proc. of The 2nd Intl. Symposium on Uncertainty Modeling and Analysis ISUMA'93*, Los Alamitos: IEEE CS Press, pp. 208-215, 1993.

[237] E. Kalai, and N. Megiddo, Path Independent Choice. *Econometrica*, Vol. 48, pp. 781-784, 1980.

[238] S.Z. Kamal, H.M. Karandikar, F. Mistree, and D. Muster, Knowledge Representation for Discipline-Independent Decision Making. In: Gero J.S.

(ed.) *Expert Systems in Computer-Aided Design*, Amsterdam: Elsevier, pp. 289-318, 1987.

[239] R.M. Karp, Probabilistic Analysis of Partitioning Algorithms for the Traveling-Salesman Problem in the Plane. *Mathematics of Operations Research,* Vol. 2, No. 3, pp. 209-224, 1977.

[240] R.M. Karp, Combinatorics, Complexity, and Randomness. *Commun. of the ACM*, Vol. 29, No. 2, pp. 98-109, 1986.

[241] P.G.W. Keen, Value Analysis: Justifying Decision Support Systems. *MIS Quarterly,* Vol. 5, March, pp. 1-14, 1981.

[242] R.L. Keeney, and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs.* New York: J.Wiley & Sons, 1976.

[243] J.G. Kemeny, and J.L. Snell, *Mathematical Models in the Social Sciences*, Cambridge, Mass.: MIT Press, 1972.

[244] M. Kendall, *Rank Correlation Methods*, 3rd ed., New York: Hafner, 1962.

[245] V.R. Khachaturov, "Approximation-Combinatorial Method of Decomposition and Composition of Systems and Limited Topological Spaces, Lattices, Optimization". *J. of Computer Mathematics and Mathematical Physics*, Vol. 25, No. 12, pp. 1777-1794, 1985 (in Russian).

[246] W.R. King, Alternative Designs in Information System Development. *MIS Quarterly*, Vol. 6, No. 4, pp. 31-42, 1982.

[247] M. Klein, Capturing Design Rationale in Concurrent Engineering Teams, *Computer*, Jan., pp. 39-47, 1993.

[248] L. Kleinrock, Distributed Systems. *Commun. of the ACM*, Vol. 28, No. 11, pp. 1200-1213, 1985.

[249] R. Kling, Cooperation, Coordination and Control in Computer Supported Work, *Comm. of the ACM*, Vol. 34, No. 12, pp. 83-88, 1991.

[250] G. Klir, and T. Folger, *Fuzzy Sets, Uncertainty, and Information*, New York: Prentice Hall, 1987.

[251] G. Klir, Generalized Information Theory, *Fuzzy Sets and Systems*, Vol. 40, pp. 127-142, 1991.

[252] D.E. Knuth, *The Art of Computer Programming*, Vol. I, Vol. II, Vol. III. Mass: Addison Wesley, Reading, 1968, 1969, 1973.

[253] D.E. Knuth, *Stable Marriage and Its Relation to Other Combinatorial Problems*, Providence: American Mathematical Society, 1997.

[254] D.E. Knuth, and A. Raghunathan, "The Problem of Compatible Representatives". *SIAM J. on Disc. Math.,* Vol. 5, No. 3, pp. 422-427, 1992.

[255] A. Kobsa A. and W. Wahlster (Eds.), *User Models in Dialog Systems*, Berlin: Springer-Verlag, 1989.

[256] M.M. Koksalan, M.H. Karwan, and S. Zionts, An Improved Method for Solving Multiple Criteria Problems Involving Discrete Alternatives. *IEEE Trans. on Systems, Man, Cybernetics,* Vol. SMC-14, pp. 24-34, 1984.

[257] D.A. Kolb, *The Learning Style Inventory: Technical Manual.* Boston, Mass.: McBer and Co., 1978.

[258] W. Kollewe, C. Sander, R. Schmiede, R. Wille, TOSKANA als Instrument der Bibliothekarischen Sacherschliessung. In: H. Havenkost, H.-J. Watjen (Hrsg.) *Aufbau und Erschliessung Begrifflicher Datenbanken*, BIS-Verlag, Oldenburg, pp. 95-114, 1995 (in German).

[259] P. Korhonen, J. Wallenius, and S. Zionts, Solving the Discrete Multiple Criteria Problem Using Convex Cones. *Management Science*, Vol. 30, No. 11, pp. 1336-1345, 1984.

[260] P. Korhonen, and J. Laakso, A Visual Interactive Method for Solving the Multiple Criteria Problem. *European J. of Operational Research*, Vol. 24, No. 2, pp. 277-287, 1986.

[261] B. Korte, and R. Schrader, On the Existence of Fast Approximation Schemes. *Nonlinear Programming,* Vol. 4, pp. 415-437, 1980.

[262] B. Kosko, Fuzzy Cognitive Maps. *Intl. J. of Man-Machine Studies*, Vol. 24, pp. 65-75, 1985.

[263] T.J. Kowalski, *An Artificial Intelligence Approach to VLSI Design.* Boston: Kluwer Academic Publishers, 1986.

[264] D.H. Krantz, R.D. Luce, P. Suppes, and A. Tversky, *Foundations of Measurement.* Vol. 1, New York: Academic Press, 1971.

[265] P.S. Krasnoshekov, V.V. Morozov, and V.V. Fedorov, Decomposition in Design Problems. *Engineering Cybernetics,* No. 2, pp. 7-17, 1979 (in Russian).

[266] G. Kron, *Diakoptics.* London: MacDonald, 1963.

[267] J.B. Kruskal, The Relationship Between Multidimensional Scaling and Clustering. In: *Classification and Clustering*, J. Van Ryzin, (ed.). New York: Academic Press, pp. 17-44, 1977.

[268] J.G. Kuhl, and S.M. Reddy, Fault-Tolerance Considerations in Large, Multiple-Processor Systems. *Computer*, Vol. 18, No. 3, pp. 56-67, 1986.

[269] N. Kuppuraju, P. Ittimakin, and F. Mistree, *Design Through Selection: A Method That Works. Design Studies*, Vol. 6, No. 2, pp. 91-106, 1985.

[270] N. Kuppuraju, S. Ganesan, F. Mistree, and J.S. Sobiesli, Hierarchical Decision Making in System Design, *Engineering Optimization*, 1985, Vol. 8, No. 3, pp. 223-252, 1985.

[271] A. Kusiak, Artificial Intelligence and Integrated Manufacturing Systems. In: A. Kusiak (Ed.) *Artificial Intelligence. Implications for CIM.* Berlin: IFS (Publications) Ltd. and Springer-Verlag, 1988.

[272] A. Kusiak, Concurrent Engineering: Models and Solutions Approaches. In: I. Mezgar and P. Bertok (eds.) *Knowledge Based Hybrid Systems*, North Holland, Amsterdam: Elsevier Science Publishings B.V., pp. 27-49, 1993.

[273] V.B. Kuzjmin, *Building of Group Decisions in Space of Deterministic and Fuzzy Relations*, Moscow: Nauka Publishing House, 1982 (in Russian).

[274] M. Kyng, Designing for Cooperation: Cooperating in Design. *Comm. of the ACM*, Vol. 34, No. 12, pp. 65-73, 1991.

[275] G. Laporte, and I. Osman, (Eds.) *Metaheuristics in Combinatorial Optimization.* Annals of Operations Research, Vol. 69, Amsterdam: Baltzer, 1995.

[276] O.I. Larichev, and A.N. Kozhukharov, A Multi-Criterial Assignment Problem. *Automatika i Telemehanika*, No. 7, pp. 71-84, 1977 (in Russian).

[277] O.I. Larichev, H.M. Moshkovich, A.I. Mechitov, and D.L. Olson, Experiments Comparing Qualitative Approaches to Rank Ordering of Multiattribute Alternatives. *J. of Multi-Criteria Decision Analysis,* Vol. 2, No. 1, pp. 5-26, 1993.

[278] J. Larkin, and H. Simon, Why a Diagram is (sometimes) Worth 10,000 Words. *Cognitive Science*, Vol. 11, pp. 65-99, 1987.

[279] J.A. Larson, *Interactive Software - Tools for Building Interactive User Interfaces.* New York: Yourdon Press Computing Series, and Prentice Hall, Englewood Cliffs, 1992.

[280] E.L. Lawler, *Combinatorial Optimization: Networks and Matroids*, New York: Holt, Rinehart and Winston, 1976.

[281] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys (eds.), *The Traveling Salesman Problem*, Chichester: J.Wiley & Sons, 1985.

[282] A.L. Lederer, and V. Sethi, The Implementation of Strategic Information Systems Planning Methodologies. *MIS Quarterly*, Vol. 12, No. 3, pp. 445-461, 1988.

[283] T. Leighton, B. Maggs, and S. Rao, Universal Packet Routing Algorithms. In: *Proc. of the 29th Annual Symposium on Foundation of Computer Science*, pp. 256-269, 1988.

[284] K.A. Lemone, Similarity Measures Between Strings Extended to Sets of Strings. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 4, No. 3, pp. 345-347, 1982.

[285] G.G. Lendaris, Structural Modeling - A Tutorial Guide. *IEEE Trans. on Systems, Man, and Cybernetics,* Vol. 10, No. 12, pp. 807-840, 1980.

[286] J.K. Lenstra, *Sequencing by Enumerative Methods.* Amsterdam: Mathematical Centre Tracts 69, 1977.

[287] R.R. Levary, Introduction to Engineering Design Using Operations Research Methods. In: R.R. Levary (Ed.) *Engineering Design. Better Results through Operations Research Methods.* New York: North Holland, pp. 1-16, 1988.

[288] M.Sh. Levin, Effective Solution of Certain Problems of Theory of Schedulings of Nets, *Cybernetics (translated from Kibernetika),* Vol. 16, No. 1-6, pp. 148-153, 1980.

[289] M.Sh. Levin, An Extremal Problem of Organization of Data. *Engineering Cybernetics,* Vol. 19, No. 5, pp. 87-95, 1982.

[290] M.Sh. Levin, Combinatorial Models in Decision Making. In: *Procedures for Evaluation of Multicriteria Objects*, Inst. for System Analysis, Moscow, No. 9, pp. 35-41, 1984 (in Russian).

[291] M.Sh. Levin, Vector-like Criterion for Proximity of Structures. *In: Abstracts of The Third Conf. "Problems and Approaches of Decision Making in Organizational Management Systems"*, Inst. for System Analysis, Moscow, pp. 20-21, 1988 (in Russian).

[292] M.Sh. Levin, A Hierarchical Hypertext System. *Automatic Documentation and Mathematical Linguistics*, Vol.23, No.3, pp. 52-59, 1989.

[293] M.Sh. Levin, Some Complicated Discrete Multi-Criteria Decision Making Problems. In: *Proc. of 10th Intl. Conf. on Multiple Criteria Decision Making*. Taiwan, Vol. 3, pp. 449-458, 1992.

[294] M.Sh. Levin, Hierarchical Components of Human-Computer Systems. In: L.E. Bass, J. Gornostaev, C. Unger (eds.), *Human Computer Interaction.* LNCS, Vol. 753, Berlin: Springer-Verlag, pp. 37-52, 1993.

[295] M.Sh. Levin, Morphological Clique Problem. In: *Abstracts of Intl. Conf. on Combinatorial Optimization CO94,* Amsterdam, April 1994.

[296] M.Sh. Levin, Towards Design of Complex Decomposable System. In: *Abstracts of the 4th Intl. Design Automation Workshop*. Moscow, pp. 2-5, June 1994.

[297] M.Sh. Levin, Hierarchical Design of User Interfaces. In: B. Blumenthal, J. Gornostaev, C. Unger (eds.), *Human Computer Interaction.* LNCS, vol.876, Berlin: Springer-Verlag, pp. 140-151, 1994.

[298] M.Sh. Levin, A Hierarchical Planning and Problem Solving: Approach to Interface Design. In: *Proc. of Intl. Conf. on Human-Computer Interaction EWHCI'95,* Moscow, Vol. 1, pp. 180-187, July 1995.

[299] M.Sh. Levin, Towards Hypertexts and Decision Making. In: *Proc. of Intl. Conf. on Human-Computer Interaction EWHCI'95,* Moscow, Vol. 2, pp. 22-37, July 1995.

[300] M.Sh. Levin, Towards Hierarchical Morphological Design of Decomposable Systems. In: *Proc. of The Second Conf. on Concurrent Engineering*, Virginia, pp. 65-75, August 1995.

[301] M.Sh. Levin, Problem of Information Center Planning, *Nauchno-Tekhnicheskaya Informatsiya*, Ser. 2, No. 2, pp. 16-24, 1995 (in Russian).

[302] M.Sh. Levin, Typical Decision Making Problems for Preparation of Business Plans, *Scientific and Technical Information Processing*, Vol. 22, No. 5, pp. 24-34, 1995.

[303] M.Sh. Levin, The Third Literacy, *Automatic Documentation and Mathematical Linguistics*, Vol. 29, No. 3, pp. 66-81, 1996.

[304] M.Sh. Levin, *Towards Combinatorial Models of Synthesis*, Technical Report 96-1-002, The University of Aizu, 1996.

[305] M.Sh. Levin, Towards Comparison of Decomposable Systems, *5th Conf. of Intl. Federation of Classification Societies IFCS-96*, Kobe, Vol. 1, pp. 218-221, March 1996.

[306] M.Sh. Levin, Towards Combinatorial Engineering of Decomposable Systems, *13th European Meeting on Cybernetics and Systems*, Vienna, Vol. 1, pp. 265-270, April 1996.

[307] M.Sh. Levin, Hierarchical Decision Making for Education in Information Engineering. In: *Proc. of 7th Annual Conf. of European Ass. for Education in Electrical and Information Engineering*, Oulu, pp. 301-306, June 1996

[308] M.Sh. Levin, Hierarchical Morphological Multicriteria Design of Decomposable Systems. *Concurrent Engineering: Research and Applications,* Vol. 4, No. 2, pp. 111-117, June 1996.

[309] M.Sh. Levin, On Teaching of Information Technology, *Nauchno-Tekhnicheskay Informatsiya*, Ser. 1, No. 5, pp. 14-23, 1996 (in Russian).

[310] M.Sh. Levin, Improvement of Decomposable Systems. In: *"Advances in Concurrent Engineering", presented at 3rd ISPE Intl. Conf. on Concurrent Engineering CE96*, Toronto, pp. 319-325, Aug. 1996.

[311] M.Sh. Levin, Combinatorial Analysis and Adaptation of Human-Computer Systems. In: *Proc. of Intl. Conf. on Human-Computer Interaction EWHCI'96,* Moscow, pp. 104-112, Aug. 1996.

[312] M.Sh. Levin, Combinatorial Design in Hypertext, In: *Proc. of 5th Intl. Conf. on Information Systems Development ISD'96*, Gdansk, pp. 255-263, 1996.

[313] M.Sh. Levin, Hierarchical Combinatorial Approach to Planning of Career, *Nauchno-Tekhnicheskaya Informatsiya*, Ser. 2, No. 12, pp. 22-27, 1996 (in Russian).

[314] M.Sh. Levin, and Yu.T. Kaganov, Hierarchical Design of Vibration Conveyor. In: *Proc. of Intl. Conf. on "Information Technology in Design"*, Moscow, pp. 164-169, July 1996.

[315] M.Sh. Levin, and L.S. Levinskii, Planning of Services for User Remote Access to Resources of a Computerized Information Network, *Automatic Documentation and Mathematical Linguistics*, Vol. 23, No. 2, pp. 12-21, 1989.

[316] M.Sh. Levin, D.B. Magidson, and E.P. Samsonova, Choice Problem of Communication Equipment in Computer-Information System. *Automatic Documentation and Mathematical Linguistics*, Vol. 24, No. 5, pp. 55-64, 1990.

[317] M.Sh. Levin, and A.A. Michailov, *Technology's Fragments of Object Set's Stratification*. Inst. for System Analysis, Preprint, Moscow, 1988 (in Russian).

[318] E.V. Levner, Optimal Planning of Object Processing on the Base of Machine Sequence. *Automatika i Telemekhanika*, No. 12, pp. 94-100, 1969 (in Russian).

[319] H. Levy, and M. Sarnat, *Investment and Portfolio Analysis*. New York: J. Wiley & Sons, 1972.

[320] H. Levy, and M. Sarnat, *Capital Investment and Financial Decisions*. Englewood Cliffs, NJ: Prentice Hall, 1986.

[321] C.-L. Li, S.T.McCormick, and D. Simchi-Levi, The Point-to-Point Delivery and Connection Problems: Complexity and Algorithms. *Applied Discrete Mathematics*, Vol. 36, pp. 267-292, 1992.

[322] H.L. Li, Solving Discrete Multicriteria Decision Problems Based on Logic-Based Decision Support Systems. *DSS,* Vol. 3, pp. 101-118, 1987.

[323] T.-P. Liang, Development of a Knowledge-Based Model Management System. *Operations Research*, Vol. 36, No. 6, pp. 849-863, 1988.

[324] D.C. Liebisch, and A. Jain, *JESSI COMMON FRAMEWORK* Design Management - The Means to Configuration and Execution of the Design Process. In: *Proc. of EURO DAC'92*. Los Alamitos: IEEE CS Press, pp. 552-557, 1992.

[325] N. Linial, and M. Saks, Low Diameter Graph Decompositions. *Combinatorica*, Vol. 13, No. 4, pp. 441-454, 1993.

[326] M. Linster, (Ed.), *Sisyphus '92: Models of Problem Solving*. Working paper 630, Institut f"ur Angewandte Informationstechnik der Gesellschaft f"ur Mathematik und Datenverarbeitung, Sankt Augustin, 1992.

[327] B.G. Litvak, *Expert Information: Methods of Acquisition and Analysis.* Moscow: Radio i Svazj, 1982 (in Russian).

[328] J.S. Liu, and K. Sucara, Distributed Problem Solving through Coordination in a Society of Agents. In: *Papers from the Thirteenth Intl. Workshop "Distributed Artificial Intelligence"*, TR WS-94-02, Menlo Park, Ca.: AAAI Press, pp. 169-185, 1994.

[329] M.V. Lomonosov, Combinatorial Approaches to Multiflow Problems. *Discrete Applied Mathematics*, Vol. 11, No. 1, pp. 3-93, 1985.

[330] V. Lotfi, T.J. Stewart, and S. Zionts, An Aspiration-level Interactive Model for Multiple Criteria Decision Making. *Computers and Operations Research*, Vol. 19, No. 7, pp. 671-682, 1992.

[331] M.L. Lu, Y. Naka, K. Shibao, X.Z. Wang, and C. McGreavy, A Multi-Dimensional Object-Oriented Information Model for Chemical Engineering. In: *Proc. of The Second Conf. on Concurrent Engineering*, McLean, pp. 21-29, 1995.

[332] M.L. Lu, H.S. Li, and Y. Naka, A Genetic Multi-dimensional Concurrent Tasking Activity Model. In: *"Advances in Concurrent Engineering", 3rd ISPE Intl. Conf. on Concurrent Engineering CE96*, Toronto, pp. 297-303, Aug. 1996.

[333] H.C. Lucas, Jr., Performance and The Use of Information Systems. *Management Science,* Vol. 16, pp. 157-162, 1989.

[334] Q.Y. Luo, P.G. Hendry, J.T. Buchanan, Strategies for Distributed Constraint Satisfaction Problems. In: *Papers from the Thirteenth Intl. Workshop "Distributed Artificial Intelligence"*, TR WS-94-02, Menlo Park, Ca.: AAAI Press, pp. 186-200, 1994.

[335] A.K. Mackworth, Constraint Satisfaction. In: S.C. Shapiro (ed.), *Encyclopedia on Artificial Intelligence*, New York: J.Wiley & Sons, pp. 205-211, 1987.

[336] S.R. Magal, H.H. Carr, and H.J. Watson, Critical Success Factors for Information Center Managers. *MIS Quarterly*, Vol. 12, No. 3, pp. 413-425, 1988.

[337] M.L. Maher, Engineering Design Synthesis: A Domain Independent Representation. *AI EDAM*, Vol. 1, No. 3, pp. 207-213, 1987.

[338] M.L. Maher, Process Models for Design Synthesis, *AI Magazine*, Vol. 11, No. 4, pp. 49-58, 1990.

[339] D. Maier, *The Theory of Relational Databases*, Los Alamitos: IEEE CS Press, 1983.

[340] S.P. Maksimov, (Ed.), *Oil and gas fields of the USSR. Handbook.* Moscow: Nedra, 1987 (in Russian).

[341] B. Malakooti, and Y. Zhou, Feedforward Artificial Neural Networks for Solving Discrete Multiple Criteria Decision Making Problems. *Management Science*, Vol. 40, No. 11, 1542-1561, 1994.

[342] A.V. Malishevski, Path Independence in Series-parallel Data Processing. *Mathematical Social Sciences*, Vol. 27, pp. 335-367, 1994.

[343] A.V. Malishevski, Structural Characterizations of the Path Independence Property for Set Transformations. In: E. Diday, Y. Ledievallier, O. Opitz (Eds.) *Ordinal and Symbolic Data Analysis*, Berlin: Springer-Verlag, pp. 319-327, 1996.

[344] T.W. Malone, and K. Crowston, The Interdisciplinary Study of Coordination. *ACM Computing Surveys*, Vol. 26, No. 1, pp. 87-119, 1994

[345] T. Margush, Distance Between Trees. *Discrete Applied Mathematics*, Vol. 4, pp. 281-290, 1982.

[346] H.M. Markowitz, *Portfolio Selection: Efficient Diversification of Investment.* New York: J. Wiley & Sons, 1959.

[347] S. Martello, and P. Toth, *Knapsack Problem: Algorithms and Computer Implementation,* New York: J.Wiley & Sons, 1990.

[348] K.A. Mayo, and H.R. Hartson, Synthesis-Oriented Situational Analysis in User Interface Design. In: L.E. Bass, J. Gornostaev, and C. Unger (Eds.) *Human-Computer Interaction*, Lecture Notes in Computer Science, Vol. 753, Berlin: Springer-Verlag, pp. 53-67, 1993.

[349] J. McDermott, R1: A Rule-Based Configurer of Computer Systems. *Artificial Intelligence*, Vol. 19, No. 2, pp. 39-88, 1982.

[350] K. Mehlhorn, A Faster Approximation Algorithm for the Steiner Problem in Graphs. *Information Processing Letters*, Vol. 27, pp. 125-128, 1988.

[351] A. Mehrez, G.S. Howard, Y. Lugassi, and S. Peretz, Information System Planning and Selection: A Multiattribute Theoretic Approach. *The Computer Journal*, Vol. 36, No. 6, pp. 525-541, 1993.

[352] Z.A. Melzak, On the Problem of Steiner. *Canad. Math. Bull.* Vol. 4, pp. 143-148, 1961.

[353] M.D. Mesarovic, D. Macko, and Y. Takahara, *Theory of Hierarchical, Multilevel Systems.* New York and London: Academic Press, 1970.

[354] Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolutionary Programs*, 3rd edition, Berlin: Springer-Verlag, 1996.

[355] N.F. Michelena, and P.Y. Papalambros, Optimal Model-based Decomposition of Powertrain System Design. *J. of Mechanical Design*, Vol. 117, pp. 499-505, 1995.

[356] A. Migdalas, and P. M. Pardalos, Hierarchical and Bilevel Programming. *J. of Global Optimization*, Vol. 8, No. 3, pp. 209-215, 1996.

[357] G.A. Miller, Magical Number Seven, Plus or Minus Two: Some Limits of Our Capacity for Processing Information. *Pshychol. Rev.*, Vol. 63, No. 2, pp. 81-97, 1956.

[358] E. Minieka, *Optimization Algorithms for Networks and Graphs.* New York: Marcel Dekker, 1978.

[359] S. Minton, M. Johnston, A. Philips, and P. Laird, Minimizing Conflicts: A Heuristics Repair Method for Constraint Satisfaction and Scheduling Problems. *Artificial Intelligence*, Vol. 58, pp. 161-205, 1992.

[360] P.B. Mirchandani, and R.L. Francis, (Eds.) *Discrete Location Theory*, New York: J.Wiley & Sons, 1990.

[361] N. Mirenkov, Visualization and Sonification of Methods. In: *Proc. of The First Aizu Intl. Symposium on Parallel Algorithms/Architecture Synthesis*, March 1995, Japan, IEEE CS Press, Los Alamitos, pp. 63-71, 1995.

[362] B.G. Mirkin, *Group Choice.* New York: Winston, 1979.

[363] B.G. Mirkin, *Mathematical Classification and Clustering.* Boston: Kluwer Academic Publishers, 1996.

[364] B.G. Mirkin, and L.B. Chernyi, On a Distance Measure Between Partitions of a Finite Set, *Automation and Remote Control*, Vol. 31, No. 5, pp. 786-792, 1970.

[365] B.G. Mirkin, and S.N. Rodin, *Graphs and Genes.* Bonn: Springer-Verlag, 1984.

[366] B. Mirkin, I. Muchnik, and T. Smith, A Biologically Consistent Model for Comparing Molecular Phylogenies, *J. of Computational Biology*, Vol. 2, No. 4, pp. 493-507, 1995.

[367] F. Mistree, and D. Muster, The Decision Support Problem Technique for Design. In: *ASEE Annual Conference Proceedings*, Cincinnati, pp. 117-126, 1986.

[368] G.E. Moiseenko, Optimal Dividing of System into Subsystems. *Automatika i Telemekhanika*, No. 7, pp. 103-112, 1977 (in Russian).

[369] M. Mollaghasemi, and J. Pet-Edwards, *Making Multiple-Objective Decisions.* Los Alamitos: IEEE CS Press, 1997.

[370] C.L. Monma, and J.B. Sidney, Sequencing with Series-Parallel Precedence Constraints. *Mathematics of Operations Research*, Vol. 4, pp. 215-224, 1979.

[371] C.L. Monma, and J.B. Sidney, Optimal Sequencing via Modular Decomposition: Characterization of Sequencing Functions. *Mathematics of Operations Research*, Vol. 12, pp. 22-31, 1987.

[372] C.L. Monma, B.S. Munson, and W.R. Pulleyblank, Minimum-Weight Two-Connected Spanning Networks. *Mathematical Programming A*, Vol. 46, No. 2, pp. 153-171, 1990.

[373] D.V. Morse, and C. Hendrickson, A Communication Model to Aid Knowledge-Based Design Systems. *AI EDAM*, Vol. 4, No. 2, pp. 99-115, 1990.

[374] A. Motro, Cooperative Database Systems. *Intl. J. of Intelligent Systems*, Vol. 11, No. 10, pp. 717-731, 1996.

[375] W.A. Muhanna, and R.A. Pick, Meta-modeling Concepts and Tools for Model Management: A Systems Approach. *Management Science*, Vol. 40, No. 9, pp. 1093-1123, 1994.

[376] J.H. Muller, and J. Spinrad, Incremental Modular Decomposition. *J. Assoc. Comput. Mach.*, Vol. 36, pp. 1-19, 1989.

[377] B.A. Nadel, Constraint Satisfaction Algorithms. *Computational Intelligence*, Vol. 5, pp. 188-224, 1989.

[378] G. Nadler, Systems Methodology and Design. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 15, No. 6, pp. 685-697, 1985.

[379] G.L. Nemhauser, and L.A. Wolsey, *Integer and Combinatorial Optimization*. New York: J.Wiley & Sons, 1988.

[380] A. Newell, and H.A. Simon, *Human Problem Solving*, NJ: Prentice Hall, Englewood Cliffs, 1972.

[381] J. Nielsen, *Usability Engineering*, Academic Press, 1993.

[382] N.J. Nilsson, *Problem Solving Methods in Artificial Intelligence*, New York: MacGraw-Hill Book Company, 1971.

[383] O. Ore, *Theory of Graphs*. Providence: American Mathematical Society, 1962.

[384] R. Ortiz, and P. Dadam, The Concurrency Model: Activating and Engineering Database through and Integrated Product and Process Data Model. In: *Proc. of the 6th Intl. Conference and Workshop on Database and Expert Systems Applications DEXA'95*, London, 1995.

[385] V.M. Ozernoy, Some Issues in Designing an Expert System for Multiple Criteria Decision Making. *Acta Psychologica*, Vol. 68, pp. 237-253, 1988.

[386] M. Padberg, and M.P. Rijal, *Location, Scheduling, Design and Integer Programming*. Boston: Kluwer Academic Publishers, 1996.

[387] G. Pahl, and W. Beitz, *Engineering Design. A Systematic Approach*. London: The Design Council & Springer-Verlag, 1988.

[388] C.H. Papadimitriou, and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. NJ: Prentice-Hall, Englewood Cliffs, 1982.

[389] P.Y. Papalambros and N.F. Michelena, Model-Based Partitioning in Optimal Design of Large Engineering Systems. In: N. Alexandrov and M.Y. Hussani (eds.), *Multidisciplinary Design Optimization: State-of-the-Art*, Philadelphia: SIAM, pp. 209-226, 1997.

[390] P.M. Pardalos, and J. Xue, The Maximum Clique Problem. *Journal of Global Optimization*, Vol. 4, No. 3, pp. 301-328, 1994.

[391] I. Parmee, and P. Booker, Applying the Genetic Algorithm to Design Problems: Progress at the Plymouth Engineering Design Centre. *Engineering Designer*, Vol. 19, No. 3, pp. 17-18, 1993.

[392] U.K. Patel, and A.G. Sutcliffe, Three-Dimensional Visualization of Knowledge Structures: Prototyping for Design Evaluation. In: L.E. Bass,

J. Gornostaev, and C. Unger (Eds.) *Human Computer Interaction*, LNCS, Vol. 753, Berlin: Springer-Verlag, pp. 163-180, 1993.

[393] Y. Perl, and Y. Shiloach, Finding Two Disjoint Path between Two Pairs of Vertices in a Graph. *J. Ass. Comput. Mach.*, Vol. 25, pp. 1-9, 1978.

[394] J. Pfanzagl, *Theory of Measurement.* New York: J.Wiley & Sons, 1968.

[395] J. Piaget, *Biology and Knowledge.* Chicago: The University of Chicago Press, 1971.

[396] J. Piaget, and B. Inhelder, *The Psychology of the Child*, New York: Basic Books, 1975.

[397] G. Piatetsky-Shapiro, Data Mining and Knowledge Discovery in Business Databases. In: Z.W. Ras, M. Michalewicz (eds.), *Foundations of Intelligent Systems.* LNAI, Vol. 1079, Berlin: Springer-Verlag, pp. 56-67, 1996.

[398] C.R. Plott, Path Independence, Rationality, and Social Choice. *Econometrica*, Vol. 41, pp. 1075-1091, 1973.

[399] D.A. Pospelov, *Situation Control: Theory and Practice,* Moscow: Nauka Publishing House, 1986 (in Russian).

[400] M. Pouzet, and I.G. Rosenberg, General Metrics and Contracting Operations. *Discrete Mathematics*, Vol. 130, No. 1-3, pp. 103-169, 1994.

[401] J.A. Powell, Informing Multimedia - a Sensitive Interface to Data for Design Professionals. In: *Proc. of Intl. Conf. of Information Technology in Design EWITD'94*, Moscow, Vol. 1, pp. 46-77, 1994.

[402] B. Prasad, *Concurrent Engineering Fundamentals: Integrated Product and Process Organization.* N.J.: Prentice Hall, 1996.

[403] G. Premkumar, and W.R. King, The Evaluation of Strategic Information System Planning. *Information and Management*, Vol. 26, No. 6, pp. 327-340, 1994.

[404] J.B. Quinn, P. Andersen, and S. Finkelstein, Managing Professional Intellect: Making the Most of the Best. *Harvard Business Review*, pp. 71-80, March-April, 1996.

[405] B. Raghunathan, and T.S. Raghunathan, Information Systems Planning and Effectiveness: An Empirical Analysis. *OMEGA*, Vol. 19, No. 2/3, pp. 125-135, 1991.

[406] T.S. Raghunathan, and W.R. King, Impacts of Information Systems Planning on Organization. *OMEGA*, Vol. 16, No. 2, pp. 85-94, 1988.

[407] R.K. Rainer, Jr., and H.H. Carr, Are Information Centers Responsive to End Users Needs? *Information and Management*, Vol. 22, No. 2, pp. 113-121, 1992.

[408] C.V. Ramamoorthy, and B.W. Wah, Knowledge and Data Engineering. *IEEE Trans. on Knowledge and Data Engineering*, Vol. 1, No. 1, pp. 9-16, 1989.

[409] R. Ramesh, and I.V. Ramakrishnan, Nonlinear Matching in Trees. *J. of the ACM*, Vol. 39, No. 2, pp. 295-316, 1992.

[410] N. Rao, V. Protopopescu, J. Barhen, and G. Seetharaman, Eds., *Proc. of Workshop on Foundations of Information/Decision Fusion with Applications to Engineering Problems.* Washington, 1996.

[411] A.M. Rappoport, Measurement of Distance Between Weighted Graphs of Expert Judgement. In: *Multicriteria Choice for Solving of Ill-structured Problems*, Inst. for System Analysis, Moscow, No. 5, pp. 97-108, 1978 (in Russian).

[412] Y.V.R. Reddy, K. Srinivas, V. Jagannathan, and R. Karinthi, Computer Support for Concurrent Engineering. *Computer,* Vol. 26, No. 1, January, pp. 12-15, 1993.

[413] G. Reinelt, *The Traveling Salesman,* Berlin: Springer-Verlag, 1994.

[414] E.M. Reingold, J. Nievegelt, and N. Deo, *Combinatorial Algorithms. Theory and Practice*, NJ: Prentice-Hall, Englewood Cliffs, 1977.

[415] B.C. Reiman, and A.D. Waren, User-oriented Criteria for the Selection of DSS Software. *Commun. of the ACM*, Vol. 28, No. 2, pp. 166-179, 1985.

[416] R.J. Riding, and I. Cheema, Cognitive Styles - an Overview and Integration. *Educational Psychology,* Vol. 11, pp. 193-215, 1991.

[417] R.J. Riding, and E. Sadler-Smith, Types of Instructional Material, Cognitive Styles and Learning Performance. *Educational Psychology,* Vol. 18, pp. 323-340, 1992.

[418] J.R. Rice, and R.F. Boisvert, From Scientific Software Libraries to Problem-Solving Environments. *IEEE Computational Science & Engineering*, Vol. 3, No. 3, pp. 44-53,1996.

[419] C.J. van Rijsbergen, *Information Retrieval*, London: Butterworths, 1979.

[420] A.H.G. Rinnooy Kan, *Machine Scheduling Problems: Classification, Complexity and Computations.* The Hague: Martinus Nijhoff, 1976.

[421] I. Rival, Ed. *Graphs and Order*, D.Reidel, Dordrecht, 1985.

[422] I. Rival, Ed. *Algorithms and Order*, D.Reidel, Dordrecht, 1989.

[423] F.R. Roberts, *Discrete Mathematical Models with Applications to Social, Biological and Environmental Problems.* NJ: Prentice Hall, Englewood Cliffs, 1976.

[424] F.R. Roberts, *Measurement Theory, with Applications to Decisionmaking, Utility, and the Social Sciences.* MA: Addson-Wesley, Reading, 1979.

[425] F.R. Roberts, *Applied Combinatorics.* NJ: Prentice Hall, Englewood Cliffs, 1984.

[426] F.R. Roberts, Issues in the Theory of Uniqueness in Measurement. In: I. Rival (ed.), *Graphs and Orders.* NATO Series C, Vol. 147, D. Reidel Publishing Company, pp. 415-444, 1985.

[427] D.F. Robinson, and L.R. Foulds, Comparison of Phylogenetic Trees. *Mathematical Biosciences*, Vol. 53, No. 1/2, pp. 131-147, 1981.

[428] H. Rock, The Three Machine No-wait Flow Shop Problem is NP-complete. *J. Ass. Comput. Mach.*, Vol. 31, pp. 336-345, 1984.

[429] F.W. Rook and J.W. Croghan, The Knowledge Acquisition Activity Matrix: A Systems Engineering Conceptual Framework. *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 19, no. 3, pp. 586-597, 1989.

[430] C.I. Ross, and F. Harary, A Description of Strengthening and Weakening Members of a Group. *Sociometry*, Vol. 22, No. 2, 139-197, 1959.

[431] G. Rote, Computing the Minimum Hausdorff Distance between Two Point Sets on a Line under Translation. *Information Processing Letters*, Vol. 38, No. 3, pp. 123-127, 1991.

[432] W.B. Rouse, A Note on the Nature of Creativity in Engineering: Implications for Supporting System Design. *Information Processing & Management*, Vol. 22, No. 4, pp. 279-285, 1986.

[433] W.B. Rouse, J.A. Cannon-Bowers, and E. Salas, The Role of Mental Models in Team Performance in Complex Systems, *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 22, No. 6, pp. 1296-1308, 1992.

[434] W.B. Rouse, and K.R. Boff, (Eds.), *System Design: Behavior Perspectives on Designers, Tools, and Organizations*, New York: North Holland, 1987.

[435] B. Roy, The Outranking Approach and Foundations of *ELECTRE* Methods. In: C.A. Bana e Costa (ed.), *Readings in Multi-Criteria Decision Aid.* Berlin: Springer-Verlag, pp. 155-183, 1990.

[436] M.I. Rubinstein, *Optimal Grouping of Interconnected Objects*, Moscow: Nauka, 1989 (in Russian).

[437] G. Rzevsky, (Ed.), *Applications of Artificial Intelligence in Engineering V*, Vol 2: Manufacture and Planning. Berlin: Springer-Verlag, 1990.

[438] T.L. Saaty, *The Analytic Hierarchy Process.* New York: MacGraw-Hill, 1988.

[439] L.E. Sadovskii, and A.L. Sadovskii, *Mathematics and Sports*, Providence: American Mathematical Society, 1994.

[440] A.P. Sage (Ed.), *Systems Engineering: Methodology and Applications*, IEEE Press, 1977.

[441] D. Sahal, *Patterns of Technological Innovations.* Mass.: Addison-Wesley, Reading, 1981.

[442] S. Sahni, General Techniques for Combinatorial Approximation. *Operations Research,* Vol. 25, No. 6, pp. 920-936, 1977.

[443] A. Salminen, J. Tague-Sutcliffe, and C. McClellan, From Text to Hypertext by Indexing. *ACM Trans. on Information Systems*, Vol. 13, No. 1, pp. 69-99, 1995.

[444] G. Salton, *Automating Text Processing.* Mass.: Addison Wesley, Reading, 1989.

[445] J.P. Sampson, Jr., and R.C. Reardon (Eds.), Enhancing the Design and Use of Computer-Assisted Career Guidance Systems. In: *Proc. of Intl. Teleconference on Technology and Career Development*, Alexandria: National Career Development Association, 1990.

[446] J.P. Sampson, Jr., Factors Influencing the Effective Use of Computer-Assisted Careers Guidance: the North American Experience. *British Journal of Guidance and Counselling*, Vol. 22, No. 1, pp. 91-106, 1994.

[447] M. Sun, A. Stam, and R.E. Steuer, Solving Multiple Objective Programming Problems Using Feed-forward Artificial Neural Networks: The Interactive FFANN Procedure. *Management Science*, Vol. 42, No. 6, pp. 835-849, 1996.

[448] R. Sause, and G.H. Powell, A Design Process Model for Computer Integrated Structural Engineering, *Engineering with Computers*, Vol. 6, No. 3, pp. 129-143, 1990.

[449] R. Sause, and G.H. Powell, A Design Process Model for Computer Integrated Structural Engineering: Design Phases and Tasks, *Engineering with Computers*, Vol. 7, No. 3, pp. 145-160, 1991.

[450] M. Schwartz, *Computer-Communication Network Design and Analysis.* NJ: Englewood Cliffs, Prentice-Hall, 1977.

[451] G.M. Scott, Managing the New-technology Cycle from R & D to Product Introduction: A Framework and Overall Technology Strategy. In: *Proc. of 28th Annual Hawaii Intl. Conf. on System Sciences*, Vol. III "Information Systems - Decision Support and Knowledge Based Systems", J.F. Nunamaker and R.H.Sprague, Jr. (eds.), Los Alamitos: IEEE Computer Society Press, pp. 482-493, 1995.

[452] A. Segev, The Node-Weighted Steiner Tree Problem. *Networks*, Vol. 17, pp. 1-17, 1987.

[453] R.W. Selby, Interconnectivity Analysis Techniques or Error Localization in Large Systems. *J. Syst. Software,* Vol. 20, No. 3, pp. 267-272, 1993.

[454] S.M. Selkow, The Tree-to-Tree Editing Problem. *Information Processing Letters.* Vol. 6, No. 6, pp. 184-186, 1977.

[455] P.H. Sellers, An Algorithm for the Distance between two Finite Sequences. *J. of Combinatorial Theory, Ser. A*, Vol. 16, pp. 253–258, 1974.

[456] S.V. Sevast'janov, On an Asymptotic Approach to Some Problems in Scheduling Theory. In: *Abstracts of 3rd All-Union Conf. on Problems of Theoretical Cybernetics*, Inst. Mat. Sibirs. Otdel. Akad. Nauk SSSR, Novosibirsk, pp. 67-69, 1974 (in Russian).

[457] S.V. Sevast'janov, On Some Geometric Methods in Scheduling Theory: a Survey. *Discrete Applied Mathematics*, Vol.55, pp. 59-82, 1994.

[458] G.A. Shafer, *Mathematical Theory of Evidence*, NJ: Princeton University Press, 1976.

[459] N.K. Shaw, M.S. Bloor, and A. de Pennington, Product Data Models. *Research in Engineering Design*, Vol. 1, No. 1, pp. 43-50, 1989.

[460] J.E. Shea, and T.M. West, A Methodology for Curriculum Development Using Multi-Objective Programming. *Computers & Industrial Engineering*, Vol. 31, No. 1/2, pp. 25-28, 1996.

[461] A.J. Shenhar, and D. Dvir, Managing Technology Projects: A Contingent Exploratory Approach. In: *Proc. of 28th Annual Hawaii Intl. Conf. on System Sciences*, Vol. III "Information Systems - Decision Support and Knowledge Based Systems", J.F. Nunamaker and R.H.Sprague, Jr. (eds.), Los Alamitos: IEEE Computer Society Press, pp. 494-503, 1995.

[462] B. Shetty, H.K. Bhargava, and R. Krishnan (eds.), *Model Management in Operations Research,* Annals of Operations Research, Vol. 38, Basel: J.C. Baltzer AG Science Publishers, 1992.

[463] J.E. Shigley, *Mechanical Engineering Design.* New York: McGraw-Hill, 1985.

[464] D.B. Shmoys, C. Stein, and J. Wein, Improved Approximation Algorithms for Shop Scheduling Problems. *SIAM J. on Computing*, Vol. 23, No. 3, pp. 617-632, 1994.

[465] B. Shneiderman, *Designing User Interface. Strategies for Effective Human-Computer Interaction.* New York: Addison-Wesley, 1987.

[466] P. Shoval, and U. Lugasi, Computer Systems Selection: The Graphical Cost-Benefit Approach, *Information and Management*, Vol. 15, pp. 163-172, 1988.

[467] J.N. Siddal, *Optimal Engineering Design. Principles and Applications.* Marcel Dekker: NY, 1985.

[468] J.B. Sidney, Decomposition Algorithms for Single-machine Sequencing with Precedence Relations and Deferral Costs. *Operations Research*, Vol. 23, pp. 283-298, 1975.

[469] I.Kh. Sigal, Decomposition Approach to Solving of Large Scale Traveling-Salesman Problem and Some Applications. *Engineering Cybernetics,* No. 6, pp. 143-155, 1990 (in Russian).

[470] J. Silverman, R.E. Steuer, and A.W. Whisman, A Multi-period, Multi Criteria Optimization System for Manpower Planning. *European J. of Operational Research*, Vol. 34, pp. 160-170, 1988.

[471] H.A. Simon, *The Sciences of the Artificial*, Cambridge, Mass.: MIT Press, 1969.

[472] H.A. Simon, *Models of Discovery and other Topics in the Methodology of Science*, Dordrecht, Boston: D.Reidel Publishing Company, 1977.

[473] H.A. Simon, and A. Ando, Aggregation of Variables in Dynamic Systems. *Econometrica*, Vol. 29, pp. 111-138, 1961.

[474] H.A. Simon, G.B. Dantzig, R. Hogarth, C.R. Plott, H. Raiffa, T.C. Schelling, K.A. Shepsle, R. Thaler, A. Tversky, and S. Winter, Decision Making and Problem Solving, *Interfaces*, Vol. 17, No. 5, pp. 11-31, 1987.

[475] H.U. Simon, Approximation Algorithms for Channel Assignment in Cellular Radio Networks. In: J. Czirik, J. Demetrovics, F. Gecseg (eds.) *Fundamentals of Computation Theory*, LNCS, Vol. 380, Berlin: Springer-Verlag, pp. 405-415, 1989.

[476] J. Singhal, and J.L. Katz, A Branch-And-Fathom Algorithm for the Long Range Process Design Problem. *Management Science*, Vol. 36, No. 4, pp. 513-516, 1990.

[477] D.S. Sink, *Product Management: Planning, Measurement and Evaluation, Control and Improvement.* New York: J.Wiley & Sons, 1985.

[478] Z. Sinuany-Stern, and A. Merhez, Discrete Multiattribute Utility Approach to Project Selection. *Journal of the Operational Research Society*, Vol. 38, No. 12, pp. 1133-1139, 1987.

[479] W.E. Smith, Various Optimizers for Single Stage Production, *Naval Res. Logist. Quart.*, Vol. 3, No. 1, pp. 59-66, 1956.

[480] R.G. Smith, and R. Davis, Frameworks for Cooperation in Distributed Problem Solving. *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 11, No. 1, pp. 61-70, 1981.

[481] W.D. Smith, and P.W. Shor, Steiner Tree Problems. *Algorithmica*, Vol. 7, pp. 329-332, 1992.

[482] I.M. Sobol, An Efficient Approach to Multicriteria Optimum Design Problems. *Surveys on Mathematics in Industry*, Vol. 1, pp. 259-281, 1992.

[483] W.E. Souder, and R.W. Ziegler, A Review of Creativity and Problem Solving Techniques. *Research Management*. Vol. 20, No. 4, pp. 34-42, 1977.

[484] J. Spronk, *Interactive Multiple Criteria Goal Programming.* Boston: Martinius Nijhoff, 1981.

[485] D. Sriram, R. Logcher, and S. Fukuda (Eds.), *Computer-Aided Cooperative Product Development.* LNCS, Vol. 492, New York: Springer-Verlag, 1991.

[486] W. Stadler, (Ed.), *Multicriteria Optimization in Engineering and in the Sciences*, New York: Plenum Press, 1988.

[487] M.K. Starr, *Operations Management*, NJ: Prentice Hall, 1978.

[488] R.E. Steuer, *Multiple Criteria Optimization: Theory, Computation, and Application*, New York: J.Wiley & Sons, 1986.

[489] R.E. Steuer, L.R. Gardiner, and J. Gray, A Bibliographic Survey of the Activities and International Nature of Multicriteria Decision Making. *J. of Multi-Criteria Decision Analysis*, Vol. 5, 1996.

[490] D.V. Steward, The Design Structure Systems: A Method for Managing the Design of Complex Systems. *IEEE Trans. on Eng. Manag.*, Vol. 28, No. 3, pp. 71-74, 1981.

[491] B. Stilman, Network Languages for Complex Systems. *Intl. J. Computers and Mathematics with Applications,* Vol. 26, No. 8, pp. 51-79, 1993.

[492] B. Stilman, A Formal Language for Hierarchical System Control. *Languages of Design,* Vol. 1, No. 4, pp. 333-356, 1993.

[493] K. Sucara, S. Roth, N. Sadeh, and M. Fox, Distributed Constraint Heuristic Search. *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 21, No. 6, 1446-1461, 1991.

[494] A. Suzuki, and M. Iri, A Heuristic Method for the Euclidean Steiner Problem as a Geographical Optimization Problem. *Asia-Pacific J. on Operations Research*, Vol. 3, No. 2, pp. 109-122, 1986.

[495] M.N.S. Swamy, and K. Thulasiraman, *Graphs, Networks, and Algorithms.* New York: J.Wiley & Sons, 1981.

[496] E.A. Sykes, and C.C. White, III, Multiobjective Intelligent Computer-Aided Design. *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 21, No. 6, pp. 1498-1511, 1989.

[497] K.-C. Tai, The Tree-to-Tree Correction Problem. *J. of the ACM*, Vol. 26, No. 3, pp. 422-433, 1979.

[498] E. Takeda, Connectivity in Fuzzy Graphs, *Techn. Reports*, Osaka University, Vol. 23, 343-352, 1963.

[499] A.S. Tanenbaum, *Computer Networks,* NJ: Englewood Cliffs, Prentice Hall, 1988.

[500] T.J. Teorey, and J.P. Fry, *Design of Database Structures*, NJ: Prentice-Hall, Inc., Englewood Cliffs, 1982.

[501] V.G. Timkovskii, Complexity of Common Subsequence and Supersequence Problems and Related Problems, *English translation from Kibernetika*, Vol. 5, pp. 1-13, 1989.

[502] T. Tomiyama, and P.J.W. Ten Hagen, Organization of Design Knowledge in an Intelligent CAD Environment. In: Gero, J.S. (ed.) *Expert Systems in Computer-Aided Design*, Amsterdam: Elsevier Publishers B.V., pp. 119-147, 1987.

[503] F.W. Tompa, A Data Model for Flexible Hypertext Database Systems, *ACM Trans. on Information Systems,* Vol. 7, No. 1, pp. 85-100, 1989.

[504] W.S. Torgenson, *Theory and Method of Scaling.* New York: J.Wiley & Sons, 1958.

[505] H. Tsukune, M. Tsukamoto, T. Matsushita, F. Tomita, K. Okada, T. Ogasawara, K. Takase, and T. Yuba, Modular Manufacturing. *J. of Intel. Manufacturing,* Vol. 4, No. 2, pp. 163-181, 1993.

[506] E. Turban, *Decision Support Systems and Expert Systems: Management Support Systems*, New York: Macmillan, 1990.

[507] W.T. Tutte, *The Connectivity Graphs.* Toronto: Toronto University Press, 1967.

[508] A. Tversky, Features of Similarity. *Psychological Review*, Vol. 84, No. 4, pp. 327-352, 1977.

[509] B.C. Twiss, *Managing Technological Innovation*, 3rd edition, London: Longman.

[510] E. Tyugu, Large Engineering Knowledge Bases. *AI in Engineering Design*, Vol. 8, No. 4, pp. 265-271, 1993.

[511] E.V. Tzoy, and D.B. Yudin, Direct Problem of Knowledge Synthesis in Decision Making Theory. *Engineering Cybernetics*, No. 1, pp. 130-139, 1990 (in Russian).

[512] E.V. Tzoy, A.D. Yudin, and D.B. Yudin, Problems of Enrichment and Synthesis of Knowledge. *Automation and Remote Control*, Vol. 55, No. 7, Part 1, pp. 927-953, 1994.

[513] G. Udo, Rethinking the Effectiveness Measures of Decision Support Systems. *Information & Management,* Vol. 22, No. 2, pp. 123-135, 1992.

[514] J.D. Ullman, *Computational Aspects of VLSI.* Los Alamitos: CS Press, 1984.

[515] J.D. Ullman, *Principles of Database and Knowledge Base Systems.* New York: Computer Science Press, 1988.

[516] E.L. Ulungu, and J. Teghem, Multi-objective Combinatorial Optimization Problems: A Survey. *J. of Multi-Criteria Decision Analysis*, Vol. 3, No. 2, pp. 83-104, 1994.

[517] A. Vainshtein, The Vector Shortest Path Problem and Other Vector Optimization Problems on Graphs. In: *Proc. of the 17th Yugoslavian Symposium on Operations Research*, Belgrade, pp. 169-172, 1990.

[518] A.I. Vakhutinsky, and B.L. Golden, A Hierarchical Strategy for Solving Traveling Salesman Problems Using Elastic Nets. *Journal of Heuristics*, Vol. 1, No. 1, pp. 67-76, 1995.

[519] J.P. Van Gigch, *Applied General Systems Theory.* 2nd ed., Vol. 1 and 2, Harper and Row Publishers, 1978.

[520] J. Van Ryzin, Ed., *Classification and Clustering.* New York: Academic Press, 1977.

[521] R. Vetschera, Composite Alternatives in Group Decision Support. *Annals of Operations Research*, Vol. 51, pp. 197-215, 1994.

[522] L.N. Vicente, and P.H. Calamai, Bilevel and Multilevel Programming: A Bibliography Review. *J. of Global Optimization*, Vol. 5, pp. 291-306, 1994.

[523] R.A. Wagner, and M.J. Fisher, The String-to-String Correction Problem. *J. of the ACM*, Vol. 21, No. 1, 168–173, 1974.

[524] F.R. Wagner, and A.H. Viegas de Lima, Design Version Management in the GARDEN Framework. In: *Proc. of the 28th DAC'91*, Los Alamitos, IEEE CS Press, pp. 704-710, 1991.

[525] R.J. Waller, The Synthesis of Hierarchical Structures: Techniques and Applications. *Decision Sciences,* Vol. 7, pp. 659-674, 1976.

[526] J. Wan, M. Bieber, J.L. Wang, and P.A. Ng, LHM: A Logic-based Hypertext Data Model for Integrating Hypertext and Information Systems. In: *Proc. of 28th Annual Hawaii Intl. Conf. on System Sciences*, Vol. III "Information Systems - Decision Support and Knowledge Based Systems", J.F. Nunamaker and R.H. Sprague, Jr. (eds.), Los Alamitos: IEEE Computer Society Press, pp. 350-359, 1995.

[527] W.B. Walsh, and S.H. Osipow (Eds.), *Career Decision Making*, Hillsdale, NJ: Erlbaum, 1988.

[528] F. Wang, and J. Deng, Work Process Management in Concurrent Engineering. In: *"Advances in Concurrent Engineering", 3rd ISPE Intl. Conf. on Concurrent Engineering CE96*, Toronto, pp. 287-294, Aug. 1996.

[529] J. Wang, and B. Malakooti, A Feedforward Neural Network for Multiple Criteria Decision Making. *Computers and Operations Research,* Vol. 19, No. 2, pp. 151-167, 1992.

[530] M.S. Waterman, *Introduction to Computational Biology: Maps, Sequences and Genomes*, New York: Chapman Hall, 1995.

[531] C. Watters, and M.A. Shepherd, A Transient Hypergraph-Based Model for Data Access, *ACM Trans. on Information Systems,* Vol. 7, No. 1, pp. 85-100, 1989.

[532] M. Weiss, and F. Stetter, A Hierarchical Blackboard Architecture for Distributed AI Systems. In: *Proc. of the 4th Intl. Conf. on Software Engineering and Knowledge Engineering*, Los Alamitos: IEEE CS Press, pp. 349-355, 1992.

[533] M. Wert, *Productive Thinking.* London: Tavistock Publications, 1961.

[534] R.J. Wilson, *Introduction to Graph Theory.* New York: Academic Press, 1972.

[535] P. Winter, Steiner Problem in Networks: A Survey. *Networks*, Vol. 17, pp. 129-167, 1987.

[536] N. Wirth, *Systematic Programming. An Introduction.* NJ: Prentice Hall, Englewood Cliffs, 1976.

[537] R.R. Yager, Ed., *Fuzzy Set and Possibility Theory. Recent Developments*, New York: Pergamon Press, 1982

[538] S. Yamada, A Mathematical Formulation of Allocation and Flooplanning Problem in VLSI Data Path Synthesis. *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences,* Vol. E77-A, No. 6, pp. 1043-1049, 1994.

[539] M. Yannakakis, Recent Developments on the Approximability of Combinatorial Problems, In: K.W. Ng, P. Raghavan, N.V. Balasubramanian and F.Y.L. Chin (eds.) *Algorithms and Computation,* LNCS, Vol. 762, Berlin: Springer-Verlag, pp. 363-368, 1993.

[540] M. Yannakakis, Some Open Problems in Approximation. In: M. Bonucceli, P. Crescenzi and R. Petreschi (eds.) *Algorithms and Complexity,* LNCS, Vol. 778, Berlin: Springer-Verlag, pp. 33-39, 1994.

[541] L. Yelowitz, An Efficient Algorithm for Constructing Hierarchical Graphs. *IEEE Trans. on System, Man, Cybernetics*, Vol. 6, No. 4, pp. 327-329, 1976.

[542] M. Yoshimura, A. Takeuchi, Multi-Phase Decision-Making Method of Integrated Computer-Aided Design and Manufacturing for Machine Products. *Intl. J. of Production Research*, Vol.31, No. 11, pp. 2603-2621, 1993.

[543] E. Yourdon, *Techniques of Program Structure and Design,* NJ: Prentice-Hall, Englewood Cliffs, 1975.

[544] P.L. Yu, *Multiple Criteria Decision Making: Concepts, techniques, and Extensions.* New York: Plenum, 1985.

[545] D.B. Yudin, Systems for Synthesis of Knowledge. *Reports of Russian Academy of Sciences*, Vol. 315, No. 4, pp. 809-812, 1990 (in Russian).

[546] B. Yuwono, and D.L. Lee, WISE: A World Wide Web Resource Data Base System. *IEEE Trans. on Knowledge and data Engineering,* Vol. 8, No. 4, pp. 548-554, 1996.

[547] L.A. Zadeh et al., Ed., *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, New York: Academic Press, 1975.

[548] L.A. Zadeh, Fuzzy Sets as a Basis for a Theory of Possibility, *Fuzzy Sets and Systems*, Vol. 1, pp. 3-28, 1978.

[549] J. Zeitoun, *Trames Planes. Introduction a une etude architecturale des trames.* Paris: Bordas, 1977 (in French).

[550] M. Zeleny, *Multiple Criteria Decision Making.* New York: McGraw-Hill, 1982.

[551] A.Z. Zelikovsky, The 11/6-approximation Algorithm for the Steiner Problem on Networks. *Algorithmica*, Vol. 9, pp. 463-470, 1993.

[552] W.R. Zhang, S.S. Chen, and R.S. King, A Cognitive Map Based Approach to the Coordination of Distributed Cooperative Agents. *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 22, No. 1, pp. 103-113, 1992.

[553] G. Zhou, J.T.-L. Wang, and P.A. Ng, Curriculum Knowledge Representation and Manipulation in Knowledge-Based Tutoring Systems. *IEEE Trans. on Knowledge and Data Engineering*, Vol. 8, No. 5, pp. 679-689, 1996.

[554] F. Zwicky, *Discovery Invention, Research Through the Morphological Approach.* New York: McMillan, 1969.

About the Author

Mark Sh. Levin is an Independent Researcher. He received M.S. degree in electronics from Moscow Technological University for Communication and Informatics (1970), M.S. degree in mathematics from Moscow State University(1975), and PhD degree (system analysis and combinatorial optimization) from Institute for System Analysis of Russian Academy of Sciences. He was with various Moscow R & D organizations as Engineer, Senior Engineer, and Head of Laboratory (1970-1983). After that he was Senior/Leading Scientist Researcher with Institute of Russian Committee of Standards, and with Institute for Computer-Aided Design of Russian Academy of Sciences, the Author of Project with Ofakim Center of Technology (Israel), a Visiting Researcher at The University of Aizu (Fukushima, Japan) and at School of Information Technology and Engineering of Ottawa University (Canada).

Index

Combinatorial Optimization

1. E. Çela: *The Quadratic Assignment Problem.* Theory and Algorithms. 1998
ISBN 0-7923-4878-8
2. M.Sh. Levin: *Combinatorial Engineering of Decomposable Systems.* 1998
ISBN 0-7923-4950-4

KLUWER ACADEMIC PUBLISHERS – DORDRECHT / BOSTON / LONDON