IOWA STATE UNIVERSITY
Digital Repository

Retrospective Theses and Dissertations

Iowa State University Capstones, Theses and Dissertations

1995

Manufacturing cell formation in a fuzzy environment

Chang-Chun Tsai
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

Part of the Industrial Engineering Commons

Recommended Citation

Tsai, Chang-Chun, "Manufacturing cell formation in a fuzzy environment " (1995). *Retrospective Theses and Dissertations*. 10989.
https://lib.dr.iastate.edu/rtd/10989

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

المنارة للاستشارات

www.manaraa.com

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Manufacturing cell formation in a fuzzy environment

by

Chang-Chun Tsai

A Dissertation Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Department: Industrial and Manufacturing Systems Engineering
Major: Industrial Engineering

Approved:

Signature was redacted for privacy.

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa

1995

UMI Number: 9540949

UMI Microform 9540949
Copyright 1995, by UMI Company. All rights reserved.

This microform edition is protected against unauthorized
copying under Title 17, United States Code.

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I wish to express my deep gratitude to my advisors, Professors Thomas Arnold Barta and Chao-Hsien Chu, for their assistance during my Ph.D. program. I am indebted to them for their continuous guidance, encouragement and support. They help me to rebuild up my confidence what I have lost. The countless discussions I have had with them have been both stimulating and fruitful.

My thanks go to all the members of my committee, Professors Doug Gemmier, Doug Mcbeth, and Vasant Honavel, for their valuable comments and encouragement. I also thank Professors Sharon Filipowski, Jo Min and Raymond Cheung for their interests in my work and helpful suggestions.

I would like to thank the financial supporters in Taiwan, National Science Council and National Cheng-Kung University. They reserved my teaching job for five years; thus I do not need to worry about the financial problem and can pay my attention on the study. I also thank ICEMT for the graduate fellowship I received last year.

I wish to thank my friends and fellow graduate students who have made my Ames years memorable ones. I thank Mike Moon, Tankut Atan, Robert Jay McCoy, Tai-Hung Yang, Cheng-Kang Chen, Sutthira Thanyavanich, Scott Singleton, Debra Bishop, Peter Brust, Anandanpillai Thirumalai, Juu-Hwa Ting, Pei-Shin Shih, and Kuo-Cheng Ko. In particular, Robert Jay McCoy, Mike Moon, and Tankut Atan deserve special thanks for the time and efforts they spend to help me in my study.

My gratitude extends to the IMSE office staff, Lori Hilpipre, Lynn Franco, Donna Cerka, and Mike Renze, for their administrative and facility assistance.

Special thanks go to my wife Huey-Lian Ferng, without her endless love, encouragement, understanding, patience, and fine food, I cannot finish my degree smoothly. I owe my son Tony and daughter Isabella much time that a father should dedicate to his children. Finally, I shall forever appreciate to my parents for their love and support. This thesis is dedicated to my parents, my wife and my children.

GENERAL INTRODUCTION

Cellular manufacturing (CM) is a manufacturing philosophy and strategy for improving both productivity and flexibility. Cell formation (CF), the first and key problem faced in designing an effective CM system, is a process whereby parts with similar design features or processing requirements are grouped into part families, and the corresponding machines into machine cells. Although the CF problem has attracted much attention, most approaches have failed to account for the imprecise linguistics and the uncertainty inherent in real-world situations. The goal of this dissertation is to develop useful CF approaches to solving the CF problem in a fuzzy environment.

Background

Cellular manufacturing is a promising manufacturing philosophy and strategy for engaging in international competition. The use of CM can result in many benefits, e.g., shortened throughput, reduced work-in-process, decreased material handling, increased production control, and decreased scrap rate [25]. This approach has been used widely in just-in-time (JIT) production and in flexible manufacturing systems (FMS). Cell formation, the process of grouping parts with similar design features or processing requirements into part families and machines into machine cells, is the first stage in designing an effective CM system.

The CF problem has attracted much attention, and great effort has been expended in the development of efficient procedures. Two categories can be discerned: (1) procedures based on design features and (2) procedures based on processing requirements [24]. Because the former type entails a classification and coding system that requires time and expertise to develop, most suggested procedures are of the latter type, which also can be divided among six categories based on the clustering method [4]: (1) array-based clustering, (2) hierarchical clustering, (3) nonhierarchical clustering, (4) mathematical programming, (5) graphic theoretic approach, and (6) heuristics. Or two categories can be used, viz., optimum or heuristic, according to the nature of the solution. Thorough review of the problem can be found in [5, 10, 18, 24].

Mathematical programming (MP) is one method of solving the CF problem. It can guarantee that the answer is optimal, and its model is the basis for other methods [5]. According to [5], six objective functions – total costs of machine investments, total costs of intercell movement, total number of intercell movements, total similarity coefficients between parts, total distances, and total subcontracting costs – have been used frequently in CF modeling. Most uses are related to the problem of exceptional elements (EEs). This indicates that dealing with EEs is an important objective for CF.

Many programming models for dealing with EEs have been proposed, but from Table 1 we know that key issues of the EE problem have not yet been addressed:

(1) Generation of an optimal CF solution and decision for the decision maker dealing with EEs. Four papers apply the MP approach; none, however, seems to search for the CF solution and to deal with EEs simultaneously.

Table 1. The properties of proposed approaches

Ref.	Approach	Required data (except the machine-part matrix)	Drawbacks
[2]	**Mathematical**	None.	* It didn't consider cost factors.
[17]	**Mathematical**	* Initial cell formation solution. * Acquisition cost of machines. * Subcontracting cost of parts. * Annual demand of parts. * Annual capacity of machines. * Processing time of parts. * Cost of intercell moving.	* The CF must be done before the optimization of eliminating EEs can be performed. * It didn't consider the capacity of machines when accepting transfer parts.
[23]	**Mathematical**	* Opportunity cost of parts. * Annual capacity of machines. * Annual demand of parts. * Processing time of parts.	* It can't figure out the exact # of EEs. Thus, it can't consider more about the duplication of machine and intercellular moving of parts. * The opportunity cost is difficult to obtain.
[3]	**Heuristic**	None.	* It can't obtain the optimal strategy to deal with EEs.
[8]	**Heuristic**	* Initial cell formation solution. * Batch size for producing parts. * Acquisition cost of machines. * Subcontracting cost of parts. * Annual demand of parts. * Annual capacity of machines. * Processing time of parts.	* The CF stage must be done before the comparing procedure. * It can't deal with the combination of machine duplication and subcontracting.
[9]	**Heuristic**	* Subcontracting cost of parts.	* It only considers subcontracting.
[11]	**Heuristic**	* Subcontracting cost of parts.	* It only considers subcontracting.
[13]	**Heuristic**	* Similarity coefficients. * Costs of intercell and intracell moving.	* It only considers intercell transfer.
[14]	**Heuristic**	None.	* It didn't consider cost factors.
[15]	**Heuristic**	* Acquisition cost of machines. * Cost and distance of intercell moving.	* It didn't considers subcontracting.
[20]	**Heuristic**	* Cost of intercell moving. * Annual demand of parts. * Processing time of parts. * Acquisition cost of machines.	* It didn't considers subcontracting. * It allocates parts to a set of existing cells.
[7]	**Simulated annealing (SA)**	* Cost of intercell moving. * Acquisition cost of machines.	* It didn't consider subcontracting.
[19]	**Tabu search & SA**	* Cost of intercell moving. * Acquisition cost of machines.	* It didn't considers subcontracting.
[21]	**Genetic algorithm (GA)**	* Annual demand of parts. * Annual capacity of machines. * Processing time of parts.	* It can't be applied in the MP approach.
[22]	**Network**	* Processing time of parts.	* It didn't considers the cost factors.

(2) Combination of the three policies (a) intercell moving, (b) machine duplication, and (c) subcontracting for particular EE parts. Only [17] considers these three policies, but it combines (a) and (b), not all three policies for a particular part.

(3) Calculation of the number of machines needed by considering the maximum utilization of machines and the available capacity of a machine that can be transferred for intercell moving. This is important information in the estimation of machine investment cost and intercell movement cost. So far, however, no papers have considered the available capacities of machines when accepting part transfers.

(4) Obtaining of a trade-off value between the total costs of dealing with EEs and group efficacy (GE), two conflicting performance measures in the CF results.

Furthermore, most approaches proposed thus far have been based upon unrealistic assumptions, e.g., (1) the coefficients used in the decision are known and constant, (2) the objective function and constraints can be defined precisely, and (3) the clustered families/cells are mutually exclusive. In practice, however, most objectives and constraints are quite difficult to define precisely. As a result, many models never have been used in practice.

The major benefit of fuzzy set theory, a new algorithmic approach that has been the subject of intense study over the past decade, is that it allows and accounts for the ambiguity, incomplete information, and uncertainty inherent in most real-world situations [27]. There has been research dealing with fuzziness of the CF problem. (See Table 2 for a summary) The scope of these studies, however, is limited in the sense that (1) most can be used only to form part families [1, 12, 26, 28] or to form part families and machine cell sequentially [6], (2)

Table 2. Fuzzy Methodologies for Manufacturing C.F.

Data		Clustering method	Type of clustering	Remarks	Refs.
Type	Source				
Fuzzy data	Part feature	Hierarchical clustering	Form part families only	* Threshold value (similarity) for grouping is fixed.	(1)
Crisp data	Process	Fuzzy clustering	Form part families/ machine cell sequ.	* Allow for disjoined cells * # of cells is fixed.	(6)
Fuzzy data	Part feature	Fuzzy clustering	Form part families only	* Allow for disjoined cells * # of cells is fixed.	(12)
Fuzzy data	Part feature	Fuzzy clustering	Form part families only	* Allow for disjoined cells * # of cells is fixed.	(26)
Fuzzy data	Process	Single linkage	Form part families only	* Threshold value (similarity) for grouping is fixed.	(28)
		Rank order clustering	Form part families/ machine cell simu.		

some apply only traditional approaches to deal with fuzzy data [1, 28], and (3) most deal only with fuzzy variables, i.e., with the possibility of disjoined cells [6, 26].

No study has attempted to deal with the following two main critical difficulties in modeling a real-world problem by means of the MP approach: (1) vagueness – either goal or constraints often are represented by a vague linguistic form, e.g., “We want the profit essentially larger than or equal to a dollars” and (2) ambiguity – the parameters are known inexactly, e.g., “The profit rate will be almost b (\$/min).”

This dissertation is motivated by studies of the following CF problems:

(1) Most models cannot group parts and machines to cells and solve the CF problem for dealing with EEs simultaneously.

(2) No model attempts to obtain the trade-off between minimizing total cost of dealing with EEs and maximizing GE.

(3) None has attempted to deal with vagueness and ambiguity by modeling a CF problem by means of the MP approach.

(4) No efficient heuristic algorithm has been developed to solve the CF problem of dealing with EEs.

Objectives of the Dissertation

The main objective of this dissertation is to develop useful fuzzy mathematical programming (FMP) models to solve the CF problems mentioned in the last section. Following the basic process of implementing FMP, the first two objectives are constructed. Furthermore, to develop a complimentary approach for the large problem, a third objective also is planned:

(1) To develop a linear programming (LP) model for simultaneously addressing two problems: (a) grouping parts and machines into cells and (b) solving the CF problem for dealing with EEs. Therefore, the fuzzy linear programming (FLP) methodology is applied to solve fuzzy CF problems involving situations such as those in which "The number of cells is around 3" or "The maximum number of machines in each cell is around 4." Because future demand and resources are fuzzy in nature and because most decision makers are unable to specify goals exactly, the FLP approach to solving the CF problem is new.

Fuzzy set theory will be applied to convert the FMP model to a conventional programming model.

(2) To develop a goal programming (GP) model able to obtain the trade-off between minimizing total cost of dealing with EEs and maximizing GE. Then, to use fuzzy multiobjective linear programming (FMLP) to find the optimal trade-off between two conflicting goals in the GP model and to compare the performance with that of GP approach.

(3) To develop an efficient heuristic genetic algorithm (HGA) to solve the problem and to compare the performances with those of traditional MP, FLP and FMLP. Fuzzy programming effectively deals with fuzziness embedded in a problem, but when the problem becomes bigger, mathematical formations require a great amount of execution time. Hence, an efficient heuristic algorithm such as the GA is needed. However, GA performance depends on proper parameters [16]. The wrong one will lead to poor computational performance. A new heuristic genetic algorithm will be proposed to improve executing efficiency.

Dissertation Organization

The remainder of this dissertation, which consists of four papers, is organized as follows. Paper I recently was submitted for publication after being revised based upon an IEEE FUZZY conference paper presented in June 1994. This paper illustrates how the FLP approach can be used to form manufacturing cells in a fuzzy environment. It also examines the impact of membership functions and operators on computational performance. The study shows that FLP not only provides a better and more flexible way of representing the problem domain but, with the newly proposed operator – min-add, also leads to improved overall performance. This paper is being reviewed by the *IIE Transactions Journal*.

Paper II is associated with the achievement of the second objective. An improved formula for computing the similarity coefficient (SC) between two parts is proposed to relate GE value and SC value. An FMLP approach also was applied to model CF problems with two conflicting objective functions based on the GP model. Like the first paper, this paper shows that FMLP with a min-add operator leads to improved overall performance.

Paper III proposes a new heuristic GA to optimize the total costs of dealing with EEs. New crossover and mutation heuristic operators strengthen GA performance to the extent that the GA outperforms MP and the traditional GA for all 17 different data sets. This paper verifies that the proposed heuristic GA can meet the critical demands of real-world applications.

Paper IV extends paper III by using the proposed heuristic GA to solve the same problems as are solved in papers I and II. The advantage of the heuristic GA again is confirmed.

In this dissertation, background and objectives are presented before the four papers just described. General conclusions, contributions of this dissertation, and future research are summarized after them. All references of these sections are listed at the end of the dissertation in the bibliography. Additional references are listed at the end of each paper. Literature reviews appear in their respective papers.

I. OPTIMIZATION OF MANUFACTURING CELL FORMATION WITH FUZZY LINEAR PROGRAMMING

A paper submitted to IIE Transactions

Chang-Chun Tsai, Chao-Hsien Chu, and Thomas Arnold Barta

Abstract

Cell formation (CF) has recently received much attention in education and industry because of its strategic importance to modern manufacturing theory. In this paper, a sophisticated linear programming model is proposed to simultaneously form manufacturing cells and minimize the total costs of dealing with exceptional elements. Also, we will illustrate how a fuzzy linear programming (FLP) approach can be used to solve the CF problem in a fuzzy environment, propose a new fuzzy operator, and examine the impact of different membership functions and operators on computational performance. Our study shows that FLP not only provides a better and more flexible way of representing the problem domain, it also leads to improved overall performance.

Introduction

Applying mathematical programming models to the solution of real-world problems is a challenging task because decision makers find it difficult to specify goals and constraints exactly and because the parameters used in these models cannot be estimated precisely. Over the past 25 years, fuzzy set theory has been applied to many disciplines, including operations research, control theory, and artificial intelligence/expert systems dealing with situations or problems involving ambiguities and fuzziness. Fuzzy linear programming is one area in which fuzzy set theory has been explored widely. For instance, FLP has been applied to problems regarding transportation [4], location planning [7], project networks [8], resource allocations [26], air pollution regulations [30], and media selection for advertising [36]. Although there has been intense study of the application of fuzzy set theory to industrial engineering [11] and to operations management [15], very few studies have attempted to use FLP in the design of manufacturing systems. This paper will focus attention on applying FLP to the design of cellular manufacturing systems, specifically in terms of CF problems.

Cell formation is the process of grouping parts with similar design features or processing requirements into parts families and the corresponding machines into machine cells. Over the past decade, the problem has received much attention because of its strategic importance to modern manufacturing theory. Extensive review of CF problems can be found, for example, in [5, 29, 33]. Most studies, however, have focused on the process of forming manufacturing cells. If any EE existed, it was removed manually [5, 17] or dealt with after initial cells were formed [28]. No study has attempted to handle EEs automatically during manufacturing CF. Furthermore, most studies have focused on using traditional analytical or heuristic methods to model the problem; a few studies have attempted to model the problem in a fuzzy environment, but their scope has been limited. Most studies form only part families [22, 37, 40] or form part families and machine cells sequentially [6]. Other studies apply only traditional approaches to fuzzy data [40], and most studies deal only with fuzzy variables [6, 37]. No study has attempted to deal with the CF problem with fuzzy goals, fuzzy constraints, or fuzzy parameters. Fuzzy constraints are the most important component of the CF problem because most algorithms are more sensitive to the number of cells that they can form and to the number of machines or parts allowed in each cell; that is, if these parameters are selected improperly, clustering results may be unacceptable [5].

The purposes of this study are twofold: first, to develop a more relatively sophisticated linear programming (LP) model able simultaneously to form manufacturing cells and to minimize the cost of eliminating EEs; second, to use FLP to model CF problems in a fuzzy environment. Because membership functions and operators tend to influence computational performance, this study also will assess the relative performances of two different types of membership functions. An improved fuzzy operator also will be proposed. Its performance will be evaluated and contrasted with that of three other commonly used operaiors.

Taxonomy of Fuzzy Linear Programming

In a fuzzy environment, mathematical programming models must take into consideration fuzzy constraints, vague goals, and ambiguous parameters. Many FLP approaches have been developed for these combinations and can be classified according to these criteria (a detailed discussion of the FLP procedure can be found, for example, in [20, 43]):

1. Problem style. Approaches can be classified as either vagueness or ambiguity, depending on whether goals, constraints, or parameters are fuzzified [14, 20]. If goals or constraints are defined subjectively by the decision maker, the programming is called *vagueness* or *flexibility*. The solution procedure for this type of model depends on model style. *Ambiguity* occurs when parameters are known inexactly and their membership functions are relative to

the degree of possibility of occurrence; the ambiguity problem, therefore, also is called *possibility* programming [14]. Different approaches have been developed for cases in which fuzzy parameters appear. For example, when they have appeared in the single objective function, parametric programming [2] and dual theory [32] have been applied. If there are multiobjective functions, α feasible and β efficiency [24] can be used. When the fuzzy parameters are in the constraints, fuzzy ranking [10, 19, 27, 31] and fuzzy addition [1] can be used.

2. Model style. The vagueness problem can be classified further as symmetric or asymmetric [42, 43]. A *symmetric model* is designed to solve problems with both a fuzzy objective and fuzzy constraints. Zimmermann's method [42] or Carlsson's parametric approach [2] can be applied to solve the problem. On the other hand, the asymmetric model is used to solve problems with either a fuzzy objective or fuzzy constraints. The solving approach can refer to Werners's approach [34] or to parametric approaches [3, 32], or can add a penalty function to the objective function [12].

3. Solving process. Both flexible and ambiguous styles of problems can be divided further into interactive or traditional styles, according to the solving procedure used [18, 20]. The interactive concept provides for the decision maker a learning process about the system. In such an environment, the decision maker can learn to establish suitable membership functions and to recognize the importance of factors in the system. Eventually, appropriate solutions can be devised by means of several interactions. Optimizing a given system [18, 20], which usually was done in a traditional programming approach, becomes unnecessary.

Key Issues of Applying Fuzzy Linear Programming

Although FLP is different from other fuzzy applications such as fuzzy inference and fuzzy ranking [43], two factors common to other fuzzy applications deserve attention.

1. Membership function. The *membership function* is a tool to incorporate fuzziness or to represent the linguistic variables for applications of fuzzy set theory. There have been many types of membership functions used in practice [20, 43]. A good membership function often has the following properties [9]: (a) it has a theoretical basis, (b) it can be calculated and fit to the problem easily, (c) it can be described in terms of a few parameters, (d) its parameters are meaningful to the problem, (e) it has a linear form, and (f) it has a closely connected membership and operator. Five types of membership function – linear nonincreasing, triangular, trapezoidal, exponential, and S-shaped – often have been used in FLP applications. The first three have linear forms. Most previous papers have applied the linear nonincreasing function [8, 26, 30] because they have had to minimize the objective function and to use vague linguistics such as "essentially no more than " or "less than" some value. The trapezoidal function and the triangular function [4, 36] normally are adapted when such vague linguistics as "almost," "about," or "nearly equal" are used. The remaining nonlinear membership functions – the exponential function [7, 23] and the S-shaped function [21] – are suitable representations for certain real-life cases such as cost or utility functions.

2. Operator. The *operator*, which defines the intersection of fuzzy sets, is a mechanism for converting FLP to traditional mathematical programming sc that the FLP can be solved by

traditional LP software. Because most vagueness programming approaches have the membership functions in the goal or constraints, a suitable operator must be chosen to aggregate all fuzzy membership functions. No operator is needed for ambiguity programming, however, because no fuzzy membership function needs to be aggregated. According to Zimmermann [43], a good operator normally has eight properties: (a) axiomatic strength, (b) empirical fit, (c) adaptability, (d) numerical efficiency, (e) compensation, (f) range of resulting set, (g) aggregating behavior, and (h) required scale level of membership functions. Table 1 shows those operators that often have been applied on FLP. The first five operators have simplistic linear forms after transformation. Their generic formats are given in Appendix A, and their characteristics are discussed next. (a) Even though the max-min operator is noncompensatory in nature, it has been used popularly in literature. This is because it uses the simple and easy-to-understand logic statement *and*. (b) Both min-bound sum and compensatory and are positively compensatory operators and follow the same aggregate rule combining min and max operations by means of a weighting factor. But these operators cannot guarantee an efficient alternative because their aggregated results fail to consider nonextreme values [35]. (c) *añd* is a positively compensatory operator proposed by Werners to remedy the aforementioned deficiency of min-bound-sum and compensatory and operators. (d) The add operator is negatively compensatory [16]. By summing all membership functions to the objective function, the add operator is much easier to understand; the range of its aggregated result, however, may be larger than one (1.0), and sometimes its clustering result may be affected by constraints. Because every

Table 1. Summary of Operators Used in the FLP Studies

Operator	Formulation [#]	Compensatory [+]	Format after Transformation	Reference
Max-min	$U_D = Min U_S$	No	Linear	[4,8,26,30,36]
Min-bound-sum	$U_D = \gamma\, Min U_S + (1-\gamma) Min(1, \sum_{s=0}^{t} U_S)$	Positive	Linear	[25]
Compensatory and	$U_D = \gamma\, Min U_S + (1-\gamma) Max U_S$	Positive	Linear	[41]
Fuzzy and $(a\tilde{n}d)$	$U_D = \gamma\, Min U_S + {}^{(1-\gamma)}/_{(t+1)} \times \sum_{s=0}^{t} U_S$	Positive	Linear	[35,41]
Add	$U_D = \sum_{s=0}^{t} U_S$	Negative	Linear	[30]
Product	$U_D = \prod_{s=1}^{t} U_S$	Negative	Nonlinear	[23,40]
γ	$U_D = (\prod_{s=0}^{t} U_S)^{1-\gamma} [1 - \prod_{s=0}^{t} (1 - U_S)]^{\gamma}$	Positive	Nonlinear	[25]

[#] The objective is to maximize U_D.

[+] The definitions of *compensatory* and of *negative compensatory* are adapted from [16].

membership function and operator has unique features, how to select a suitable membership function and/or operator long has been an issue deserving extensive study. In this paper, two membership functions – linear nonincreasing and triangular – will be examined because they are more appropriate for the CF problem. Also, among the five linear-form operators, three – max-min, $\widetilde{and}$, and add – will be examined because of their simplicity. The min-bound-sum and compensatory and operators were excluded from comparison because of the weaknesses already mentioned. The performances of these operators will be evaluated along with a new operator proposed in this study.

Notation Used in the Formulations

The following notation was used to model CF problems:

Index Set

i machine index; i=1,.....m

j part index; j =1,.....n

k cell index; k=1,.....c

l index of membership functions; l= 0,.....c

s index of fuzzy constraints; s= 1,.....c

Parameters

A_i periodic cost of acquiring machine type l

C_i periodic capacity of machine type i

D_j periodic forecast demand for part j

I_j incremental cost for moving a unit of part j within two cells

NM maximum number of machine types allowed in each cell

P_0 tolerance value for the fuzzy objective function (cost)

P_r tolerance values for the fuzzy constraints (*NM*)

P_{ij} processing time of machine type *i* needed to produce part *j*

γ parameter used in fuzzy modeling

S_j incremental cost of subcontracting a unit of part *j* for an operation

SP set of pairs (*i,j*) such that $a_{ij} = 1$

UC_{ij} utilization capacity of machine type *i* for parts *j*. Value can be calculated by means of the equation $P_{ij} \times D_j / C_i$

$U_D(x)$ membership function of aggregated results

$U_S(x)$ *sth* membership function

Z^0 optimal solution using the maximum value of *NM*

Z^1 optimal solution using the minimum value of *NM*

Decision Variables

IC_k =1, if cell *k* is formed; 0, otherwise

M_{ijk} number of machines *i* dedicated to cell *k* for producing part *j*

O_{ijk} units of part *j* to be subcontracted as a result of machines type *i* not being available within cell *k*

Q_i number of machines type *i* needed to process corresponding parts in machine cell

R_{ik} number of machines type *i* to be dedicated in cell *k*

U_{ijk} =1, if $X_{ik} = 1$, and $Y_{jk} = 0$; 0 otherwise

V_{ijk} =1, if Y_{jk}= 1, and X_{ik}= 0; 0 otherwise

X_{ik} =1, if machine *i* is assigned to cell *k*; 0, otherwise

Y_{jk} =1, if part *j* is assigned to cell *k*; 0, otherwise

Z_{ijk} number of intercellular transfers required by part *j* as a result of machine type *i* not being available within part cell *k*

λ minimum value of all membership functions

α_l extra variables used in the fuzzy and operator

Traditional Model

The proposed model is an extension of the model used in [28], in which two major weaknesses can be found. First, the CF stage must occur separately, before optimization through elimination of EEs. Second, the model does not consider machine capacity when accepting part transfers. The proposed model remedies both deficiencies. Not only is the best decision arrived at regarding assignment of parts and machines to cells such that the total cost of dealing with EEs can be minimized, but also the available capacity and the investment cost of the machines are considered. The needed number of machines consequently can be determined and minimized. The formulation is as follows:

$$MIN \sum_{k}\sum_{i} A_i R_{ik} + \sum_{k} \sum_{(i,j)\in sp} I_j Z_{ijk} + \sum_{k} \sum_{(i,j)\in sp} S_j O_{ijk} \qquad (1)$$

Subject to:

$$\sum_{k=1}^{c} X_{ik} = 1, \ \forall i \qquad (2)$$

$$\sum_{k=1}^{c} Y_{jk} = 1, \ \forall j \qquad (3)$$

$$\sum_{i=1}^{m} X_{ik} \leq NM, \forall k \tag{4}$$

$$X_{ik} - Y_{jk} + \frac{1}{D_j} Z_{ijk} + \frac{1}{D_j} O_{ijk} {}_{i} + \frac{1}{UC_{ij}} M_{ijk} - U_{ijk} = 0, \forall (i,j) \in sp, \forall k \tag{5}$$

$$\sum_{(i,j) \in sp} M_{ijk} \leq R_{ik}, \forall i, \forall k \tag{6}$$

$$Q_i \leq \sum_{(i,j) \in sp} UC_{ij}(1 - \sum_{k} V_{ijk}) + 1, \forall i \tag{7}$$

$$\sum_{k} \sum_{(i,j) \in sp} \frac{P_{ij}}{C_i} Z_{ijk} \leq Q_i - \sum_{(i,j) \in sp} UC_{ij}(1 - \sum_{k} V_{ijk}), \forall i \tag{8}$$

$$X_{ik}, Y_{jk}, U_{ijk}, V_{ijk} = 0 \text{ or } 1; \quad R_{ik}, Q_i = \text{general integer} \tag{9}$$

Three types of costs associated with EEs are to be minimized in equation (1). The first cost is that of duplicating a machine. This cost subsumes the purchase, maintenance, salvage, and machine life. The second cost includes intercellular transfers for the EE. The last cost is that of subcontracting. Constraints (2) and (3) ensure that each machine and part is assigned only one cell. Constraint (4) prevents the assignment of more than *NM* machines to each cell. This constraint also prevents all machines and parts from being assigned to a single cell. Constraint (5) combines two equations:

$$X_{ik} - Y_{jk} + U_{ijk} - V_{ijk} = 0, \ \forall (i,j) \in sp, \forall k$$

$$Z_{ijk} + O_{ijk} + \frac{C_i}{P_{ij}} M_{ijk} = D_j U_{ijk}, \ \forall (i,j) \in sp, \forall k$$

and ensures that an EE either is an *exceptional machine* (a machine to be duplicated) or an *exceptional part* (a part to be transferred or subcontracted). Furthermore, the constraint guarantees that the demand of exceptional part *j* can be shared by the duplicated machine *i*, transfer within cells, and subcontract. Constraint (6) calculates the number of machines type i needed to be dedicated within cell k to producing the EEs, where M_{ijk} is a real variable representing the utilization capacity of a machine type i dedicated to process part j in cell k. Constraint (7) determines the numbers of machines type i needed in each cell. The constraint sums the utilization capacity of machines type i for all relative parts ($\sum_{(i,j)\in sp} UC_{ij}$), not the exceptional elements ($-\sum_{k} V_{ijk}$). Constraint (8) ensures that the number of intercellular transfers between machines type i do not exceed available machine capacity. The maximum available capacity of machine type i (machine unit) for the relative parts to be transferred is equal to C_i/P_{ij} times the right side of Equation (8).

Fuzzy Models

The aforementioned traditional model assumes that objective functions and constraints can be defined precisely; they cannot be, however. For example, the right-hand side of constraint (4) often is fuzzy and can be expressed as

$$\sum_{i=1}^{m} X_{ik} \tilde{\leq} NM,\ \forall k \ \text{ or } \ \sum_{i=1}^{m} X_{ik} \cong NM,\ \forall k. \qquad (10)$$

And according to Werners' approach [34], (1) can be fuzzified as :

$$\sum_{k}\sum_{i} A_i R_{ik} + \sum_{k}\ \sum_{(i,j)\in sp} I_j Z_{ijk} + \sum_{k}\ \sum_{(i,j)\in sp} S_j O_{ijk} \tilde{\leq} Z^0 = Z^1 - P_0 \qquad (11)$$

Here, the value of Z^0 is the feasible value of the best goal, which can be obtained by solving the traditional model with the maximum value of NM. Z^1 is the feasible value of the worst goal, which can be obtained by solving the same model with the minimum value of NM. To convert the fuzzy model to a traditional formulation, two tolerance values – p_r, and p_0 – must be determined in advance. p_r, for equation (10), normally is determined by the decision maker, according to problem characteristic or experience. p_0 for equation (11) can be determined from the budget limit and its allowance or can be set as a value equal to Z^0 subtracted from Z^1. Fuzzy objective function (11) and fuzzy constraint (10) then can be converted to the conventional formulation by means of a suitable operator. As mentioned, two membership functions – linear nonincreasing and triangular – and four operators – max-min, *añd*, add, and a newly proposed operator – will be examined. Additionally, if the membership function is linear and nonincreasing, the transformation formula in Zimmermann [43] will be used; otherwise, the formula in Yang and Ignizio [38] will be. Consequently, there are eight cases needing to be examined for (10) and (11). The first two cases are the combinations of max-min operator and different membership functions, cases 3 and 4, the combinations of the *añd* operator and different membership functions. Cases 5 and 6 are the combinations of the add operator and different membership functions. Cases 7 and 8 combine the proposed operator with different membership functions. The generic formation can be found in Appendix A. Equivalent formulations for each case now are summarized. Please note that, for each case, complete formulation includes those equations noted below as well as equations (2), (3), and (5) - (9).

Case 1: Max-min operator and linear nonincreasing membership function. The equivalent formulation can be obtained as

$$MAX\ \lambda \tag{12}$$

subject to

$$\sum_k \sum_i A_i R_{ik} + \sum_k \sum_{(i,j) \in sp} I_j Z_{ijk} + \sum_k \sum_{(i,j) \in sp} S_j O_{ijk} + \lambda P_0 \leq Z^0 + P_0 \tag{13}$$

$$\sum_{i=1}^{m} X_{ik} + \lambda P_r \leq NM + P_r, \forall k \tag{14}$$

$$\sum_{k=1}^{c} IC_k \geq 2 \tag{15}$$

$$\sum_{i=1}^{m} X_{ik} \geq 2 IC_k, \forall k \tag{16}$$

$$0 \leq \lambda \leq 1. \tag{17}$$

Constraint (15) prevents the formation of fewer than two cells, and constraint (16) prevents the assignment of fewer than two machines in each cell formed. There, the IC_k variables in equations (15) and (16) are added either to allow a lower-bound value of the number of machines in each cell when there is a linear nonincreasing membership function or to prevent the grouping of parts and machines into exactly "C" cells when there is a triangular membership function. Consequently, the number of cells permitted ranges from "2" to "C," and when linear nonincreasing membership is used, the number of machines allowed in each cell ranged from "2" to "$NM + P_r$."

Case 2:Max-min operator and triangular membership function. The equivalent LP formulation consists of equations (12), (13), (14), (15), (17), and (18):

$$\sum_{i=1}^{m} X_{ik} - NM \times IC_k - \lambda P_r \geq -P_r, \forall k . \tag{18}$$

The difference between cases 1 and 2 is that the triangular membership function requires one additional constraint (18) to define the minimum value of $\sum_{i=1}^{m} X_{ik}$.

Case 3:The *añd* operator and the linear nonincreasing membership function. The equivalent LP formulation consists of equations (19), (15), (16), (20), (21), (22), and (23).

$$MAX\,\alpha+(1-\gamma)\frac{1}{c+1}\sum_{l=0}^{c}\alpha_l \tag{19}$$

subject to

$$\sum_k\sum_i A_i R_{ik}+\sum_k\sum_{(i,j)\in sp} I_j Z_{ijk}+\sum_k\sum_{(i,j)\in sp} S_j O_{ijk}+\alpha P_0+\alpha_0 P_0\leq Z^0+P_0 \tag{20}$$

$$\sum_{i=1}^{m} X_{ik}+\alpha P_r+\alpha_k P_r\leq NM+P_r,\ \forall k \tag{21}$$

$$\alpha+\alpha_l\leq 1,\ l=0,\ldots c \tag{22}$$

$$\alpha_l\geq 0,\ \ 0\leq\alpha\leq 1, \gamma<1. \tag{23}$$

Because no simple rule can be applied to decide the value of γ, determining it becomes a major bottleneck in the use of the *añd* operator. An experiment has been conducted to identify the nature of the γ value. Results are discussed later.

Case 4:The*añd* operator and triangular membership function. The equivalent LP formulation can be expressed as (24), (15), (20), (21), (25),(26), and (23).

$$MAX\,\alpha+(1-\gamma)\frac{1}{2c+1}\sum_{l=0}^{2c}\alpha_l \tag{24}$$

$$\alpha+\alpha_l\leq 1,\ l=0,\ldots 2c \tag{25}$$

$$\sum_{i=1}^{m} X_{ik}-NM\times IC_k-\lambda\times P_r-\alpha_{c+k}\times P_r\geq P_r,\ \forall k. \tag{26}$$

Case 5: Add operator and linear nonincreasing membership function. The equivalent LP formulation consists of equations (27), (15), (16), (28), and (29):

$$Min \frac{\sum_{k}\sum_{i} A_i R_{ik} + \sum_{k} \sum_{(i,j) \in sp} I_j Z_{ijk} + \sum_{k} \sum_{(i,j) \in sp} S_j O_{ijk} - Z_0}{P_0} + \sum_{s=1}^{c} \frac{S_s}{P_r} \tag{27}$$

subject to

$$\sum_{i=1}^{m} X_{ik} - S_k \leq NM, \forall k \tag{28}$$

$$S_s \leq P_r, \quad s = 1 \ldots c. \tag{29}$$

Case 6: Add operator and triangular membership function. The equivalent LP formulation consists of equations (30), (15), (28), (31), and (32):

$$Min \frac{\sum_{k}\sum_{i} A_i R_{ik} + \sum_{k} \sum_{(i,j) \in sp} I_j Z_{ijk} + \sum_{k} \sum_{(i,j) \in sp} S_j O_{ijk} - Z_0}{P_0} + \sum_{s=1}^{2c} \frac{S_s}{P_r} \tag{30}$$

$$\sum_{i=1}^{m} X_{ik} + S_{c+k} \geq NM * IC_k, \forall k \tag{31}$$

$$S_s \leq P_r, \quad s = 1 \ldots 2c. \tag{32}$$

Case 7: The proposed operator (denoted as Add – min) and linear nonincreasing membership function. The proposed operator applies the min operator for the fuzzy constraints. Thus, the aggregated membership function becomes

$$U_D = 0.5(U_G + \underset{s=1}{Min}\, U_s). \tag{33}$$

The range of U_D is [0, 1]. The model after aggregation is a linear form, and the compensatory property is better than the max-min and add operators. The equivalent transformed formulation is

$$MAX\ U_G + \lambda \qquad (34)$$

Subject to:

$$U_G = 1 - \frac{\sum_k \sum_i A_i R_{ik} + \sum_k \sum_{(i,j) \in sp} I_j Z_{ijk} + \sum_k \sum_{(i,j) \in sp} S_j O_{ijk} - Z_0}{P_0} \qquad (35)$$

$$1 - \frac{\sum_{i=1}^{m} X_{ik} - NM}{P_r} \geq \lambda\ ,\ \forall k. \qquad (36)$$

If we insert constraint (35) into (34), the fuzzy objective function becomes (37). The fuzzy constraints include (14), (15), (16), and (17).

$$MIN\ \sum_k \sum_i A_i R_{ik} + \sum_k \sum_{(i,j) \in sp} I_j Z_{ijk} + \sum_k \sum_{(i,j) \in sp} S_j O_{ijk} - \lambda P_0. \qquad (37)$$

Case 8:The proposed operator and triangular membership function. The objective function of the equivalent LP formulation is the same as in (37), and the constraints are equations (14), (15), (17), and (18).

Data Sets for Numerical Computations

To evaluate the performance of the proposed fuzzy models, three data sets with different percentages of EEs are used. Here, the percentage of EEs is defined as their number divided by the total number of elements in a data set. Data set I, adapted from [28], has the highest percentage of EEs (28.6%) and can be formed into two or three cells. Table 2 lists the processing time of each part, the costs involved, the part demand, and the machine capacity of this data set. Data set II, with an intermediate EE percentage (18.8%), also can be formed into

Table 2: Numerical Values for Data Set I

Parts

		1	2	3	4	5	6	7	8	9	10	A_i	C_i
M	1	2.95	0	2.2	0	0	0	0	0	0	4.61	$50,784	2000
a	2	2.76	5.18	1.89	3.89	0	5.14	0	0	0	0	$67,053	2000
c	3	5.54	4.29	0	0	0	0	0	0	0	0	$43,944	2000
h	4	2.91	0	0	1.97	2.59	4.01	0	2.7	0	0	$67,345	2000
i	5	0	0	0	4.28	0	4.51	0	0	0	0	$42,414	2000
n	6	1.92	0	0	0	0	0	2.23	0	5.52	0	$75,225	2000
e	7	0	0	0	0	3.4	0	1.16	4.72	0	2.49	$52,741	2000
s	8	0	5.32	0	0	0	0	0	3.75	3.85	0	$63,523	2000
	9	0	0	0	0	0	0	4.04	0	0	1.83	$50,632	2000
	S_j	$4.20	$4.30	$3.50	$4.40	$5.00	$3.90	$4.40	$4.60	$5.00	$5.00		
	D_j	32128	27598	20651	11340	18707	17040	46196	45384	16409	22000		
	I_j	$3.70	$2.80	$2.80	$3.30	$2.80	$3.50	$2.80	$2.60	$3.40	$3.20		

two or three cells. The machine/part matrix of this data set was from [13], but the other data were generated randomly by a computer program based on the mean value and the standard deviation of data set I. The percentage of EEs of data set III is 3.3%, the lowest percentage. Two, three, or four cells can be formed in this data set. The machine/part matrix of data set III was from [17], and the other related data also were generated randomly. For all data sets, the maximum number of machines allowed in each cell is “no more than four" or “around four.” Thus, for the traditional model, the right-hand side of equation (5) is a fixed number "four"; under the fuzzy environment, however, we can use fuzzy intervals to represent these numbers.

Computational Results

All models were solved by running the LINDO (linear interactive and discrete optimizer) package on an IBM compatible 486 DX2 66 MHZ computer. Clustering performances were measured in terms of EE numbers, executing pivots (CPU time), and total costs of dealing with EEs. Table 3 summarizes computational results from nine different cases. Cases 1 to 8 already have been discussed. Case 9 is the traditional CF model included for purposes of comparison. Several observations can be made based on the information in Table 3.

First, though the computational CPU times of the fuzzy models differed from case to case, clustering results, except for case 5 of data set III, are far better than those obtained for thetraditional model. This can be seen from the fact that EE number and total costs are smaller for all fuzzy models except case 5. The clustering results of case 5 actually are the same as those for the traditional model.

Table 3. Computational Results for Traditional and Fuzzy Models

Case		Data Set I [28]				Data Set II [13]				Data Set III [17]			
Operator	Membership Functions	NMc +	Pivots (CPU Time)	# of EE	Cost of Dealing with EE	NMc	Pivots (CPU Time)	# of EE	Cost of Dealing with EE	NMc	Pivots (CPU Time)	# of EE	Cost of Dealing with EE
1. Max-min	Linear nonincreasing	(4,5)	8652840 (31:28:39)	6	$371,297	(4,5)	1153709 (3:53:54)	5	$221,930	(5,5,4)	607414 (3:45:43)	2	$103,370
2. Max-min	Triangular	(4,5)	6548046 (25:41:09)	6	$371,297	(4,5)	892312 (2:48:54)	5	$221,930	(5,5,4)	127757 (0:46:19)	2	$103,370
3. $a\tilde{n}d$	Linear nonincreasing	(4,5)	2825289 (9:38:35)	6	$301,695	(4,5)	198635 (0:36:55)	5	$168,200	(5,5,4)	233673 (1:27:48)	2	$67,758
4. $a\tilde{n}d$	Triangular	(4,5)	416468 (1:23:34)	6	$301,695	(4,5)	64639 (0:13:02)	5	$168,200	(5,5,4)	152506 (0:59:06)	2	$67,758
5. Add	Linear nonincreasing	(4,5)	239216 (0:41:16)	6	$301,695	(4,5)	56757 (0:08:02)	5	$168,200	(4,4, 4, 2)	139872 (0:52:57)	4	$186,740
6. Add	Triangular	(4,5)	253607 (0:40:20)	6	$301,695	(4,5)	29462 (0:05:33)	5	$168,200	(6,5,3)	118359 (0:41:38)	0	$0
7. Add-min	Linear nonincreasing	(4,5)	208898 (0:30:26)	6	$301,695	(4,5)	25888 (0:04:52)	5	$168,200	(5,5,4)	117888 (0:41:27)	2	$67,758
8. Add-min	Triangular	**(4,5)**	**128741** **(0:18:59)**	6	$301,695	(4,5)	**20925** **(0:03:49)**	**5**	**$168,200**	(5,5,4)	**67402** **(0:23:18)**	2	$67,758
9. Traditional Model (NM = 4)		(4,3, 2)	327486 (0:41:28)	9	$441,233	(4,3, 2)	61054 (0:11:29)	6	$282,878	(4,4, 4,2)	177333 (1:07:22)	2	$209,660

+ NMc: Number of machine types in each cell. For instance, (4,3,2) means assigning four machines to cell 1, three machines to cell 2, and two machines to cells 3.

Second, even though the max-min operator was used most often in the literature, its performance was surprisingly unacceptable. Not only did it require the longest time to process, clustering results (in terms of cost and of EE number) also always are worse than those of the other operators. For instance, more than 31 hours and 25 hours were required to complete cases 1 and 2 of data set I, respectively, and the costs of dealing with EE are far higher than those in the other cases. Although the situation is improved in data set III, results still are unsatisfactory. For instance, nearly 4 hours are required for using the linear nonincreasing membership function, and 46 minutes when a triangular membership function is used.

Third, although the $a\tilde{n}d$ operator often arrived at good clustering results and required shorter CPU time than the max-min operator did, the performance time of the former still was far worse than that of either the add or the proposed operators. Meanwhile, determining a proper γ value is difficult. As mentioned, an experiment was conducted to determine the best possible values of γ . Table 4 summarizes the results of using data set II. According to the table, the best γ values for cases 3 and 4 are 0.8 and 0.2, respectively. Although they are not shown, the best values for data set I are 0.2 and 0.1, and for data set III are 0.3 and 0.8. The best value for γ clearly depends on the data set. Nevertheless, when the best γ values are used for comparison, the $a\tilde{n}d$ operator still performed worse in terms of CPU time than the add operator or the proposed operator does.

Fourth, although applying the add operator can shorten CPU time, this operator has two basic weaknesses: (1) It is time consuming to obtain the objective function because all membership functions must be summed up in the objective function and (2) clustering results are

Table 4. Results of Varying γ Values for the $a\tilde{n}d$ Operator

γ *value*	Data Set II (Linear Nonincreasing Membership Fn.)				Data Set II (Triangular Membership Function)			
	NMc+	Executing Pivots (CPU Time)	# of EE	Cost of Dealing with EE	NMc	Executing Pivots (CPU Time)	# of EE	Cost of Dealing with EE
0.1	(4,5)	415873 (1:02:09)	5	$161,930	(4,5)	101079 (0:16:14)	5	$161,930
0.2	(4,5)	758110 (1:53:17)	5	$161,930	**(4,5)**	**64639** **(0:13:02)**	**5**	**$161,930**
0.3	(4,5)	448390 (1:47:19)	5	$161,930	(4,5)	131369 (0:26:27)	5	$161,930
0.4	(4,5)	726365 (1:48:46)	5	$161,930	(4,5)	131388 (0:26:29)	5	$161,930
0.5	(4,5)	337578 (0:57:18)	5	$161,930	(4,5)	213858 (0:43:11)	5	$161,930
0.6	(4,5)	608503 (2:05:19)	5	$161,930	(4,5)	297232 (0:58:33)	5	$161,930
0.7	(4,5)	537940 (1:40:36)	5	$161,930	(4,5)	367246 (1:14:03)	5	$161,930
0.8	**(4,5)**	**198635** **(0:36:55)**	**5**	**$161,930**	(4,5)	414789 (1:23:37)	5	$161,930
0.9	(4,5)	870456 (2:33:04)	5	$161,930	(4,5)	436411 (1:27:59)	5	$161,930

+ NMc: Number of machine types in each cell.

affected by constraints because all membership functions of the constraints are put in the objective function. Poor clustering results can be seen from data set III of Table 3; the CF result of case 5 is the worst. It also is interesting that the result of case 6 of data set III is unique. All remaining machines or parts were assigned into a single cell, and thus the necessity of duplicating machines or of transferring parts was avoided. For this reason, cost was zero.

Fifth, the proposed operator (add-min) obviously is the most efficient no matter which membership function is used. For example, in data set I, the worst case of using the proposed operator still is better than the best cases of using the other three operators. It also is true that the proposed operator with any membership function consistently performs better than the conventional formulations do, and other operators often require more CPU time than the traditional model does.

Finally, the triangular membership function applied with any operator always resulted in shorter CPU times than the linear nonincreasing function did. This could be so because constraints of the former needed a lower-bound, and thus the possible range of solutions was narrowed. The triangular function may be much more compliant than the linear nonincreasing function or the triangular function may be more appropriate for representing the fuzzy constraints of the number of machines.

Sensitivity Analyses

To verify the sensitivity of these computational results, two follow-up analyses were performed. The first examines the impact of γ value on the computational performance of the *$\tilde{a}\tilde{n}d$* operator. Table 4 summarizes results using data set II. As shown, though CPU times depend upon γ values, clustering results are the same. Thus, the selection of γ value affects CPU time but not clustering result.

The second analysis concerned impact of p_0 value on the computational performance of the proposed operator. The best case was chosen from each data set (case 8), and the p_0 value ranged from 1/2 p_0 to 3/2 p_0 for each case. Results are summarized in Table 5. As can be seen, although p_0 value ranged, clustering results from three different data sets remained the same. Moreover, CPU time for the triangular membership function always was shorter than that of the linear nonincreasing membership function. The proposed operator therefore is robust, and the performance of the triangular membership function better than that of the linear nonincreasing membership function.

Concluding Remarks

This paper proposes an efficient mathematical programming formulation and corresponding FLP models simultaneously to form machine cells and to minimize the cost of eliminating EEs. Two membership functions with four operators, including a newly proposed operator, were applied and their results compared. Sensitivity analyses also were performed to test the robustness of the fuzzy models and of the proposed operator. From the computational analyses and sensitivity tests, a number of conclusions can be drawn.

First, the FMP approach not only provides a better and more flexible way of representing the problem domain, it also leads to improved clustering performance. The CPU time required for fuzzy models, however, depends on operator used. Clearly, the proposed operator always outperforms the traditional model whereas the performance of other operators depends

Table 5. Impact of Different p_0 Values on the Performance of the Proposed Operator

p_0 Value	Data Set I (p_0= \$166,000)				Data Set II (p_0= \$121,000)				Data Set III (p_0= \$209,660)			
	# of Cells	Executing Pivots (CPU Time)	# of EE	Cost of Dealing with EE	# of Cells	Executing Pivots (CPU Time)	# of EE	Cost of Dealing with EE	# of Cells	Executing Pivots (CPU Time)	# of EE	Cost of Dealing with EE
$\frac{1}{3}p_0$	2	83023 (0:13:13)	6	\$301,695	2	26921 (0:05:23)	5	\$168,200	2	65842 (0:25:19)	2	\$67,758
$\frac{1}{2}p_0$	2	92309 (0:14:37)	6	\$301,695	2	23268 (0:04:16)	5	\$168,200	3	61321 (0:21:41)	2	\$67,758
p_0	2	128741 (0:18:59)	6	\$301,695	2	20925 (0:03:49)	5	\$168,200	3	67402 (0:23:18)	2	\$67,758
$\frac{3}{2}p_0$	2	188214 (0:28:39)	6	\$301,695	2	20399 (0:04:02)	5	\$168,200	3	83308 (0:29:37)	2	\$67,758
$0p_0$ (Traditional Model)	3	327486 (0:41:28)	9	\$441,233	3	61054 (0:11:29)	6	\$282,878	4	177333 (1:07:22)	2	\$209,660

on the data used.

Second, performances (in terms of both clustering results and CPU time) of the popularly used max-min operator are worse than those of the other operators, no matter which membership function is used. Thus, a frequently used method may not necessarily be the best.

Third, though the *añd* operator has the best compensatory property, its performance is worse than that of the proposed operators. The compensatory property therefore is not necessarily the only major factor needing to be considered when operators are selected for the CF problem.

Fourth, the proposed operator always outperformed the other three operators, regardless of the measure used. It also was more stable and robust than the others, regardless of membership function or p_0 values.

Finally, the triangular membership function was a more appropriate membership function for solving the CF problem than the linear nonincreasing type of membership function was.

Using FLP to model CF problems is a promising approach. Although this paper has demonstrated through examples that FLP can be applied successfully to solve the designated problem, several issues require further study. For instance, the fuzziness considered in this paper was limited to fuzzy constraints. Several other parameters such as processing time, intercellular transfer cost, and subcontracting can be fuzzified further. Other objectives such as machine utilization and total similarity coefficient also can be considered as a fuzzy equation, or the problem can become a multiple fuzzy linear objective function.

Appendix: Generic Forms of Selected Fuzzy Operators

Notation

The following notation is used in the transformation of fuzzy models:

$U_G(x)$: the membership function of the objective function

$U_G = 1 - {(f(x) - Z^0)}/{P_0}$, where P_0 is the tolerance value for *f(x)*.

$U_{C_l}(x)$: the membership function of the constraints

$U_{C_l} = 1 - {(AX - b)}/{P_r}$, where P_r is the tolerance value for the fuzzy constraints.

1. Max-min operator:

Proposed by Zadeh [39], this operator uses min to define the intersection of an aggregated rule, i.e.,

$$U_D(X) = U_G(X) \wedge U_{C_l}(X) = U_G(X) \cap U_{C_l}(X). \tag{A1}$$

The aggregated membership function becomes

$$U_D(X) = Min[U_G(X), U_{C_l}(X)], \ \forall j. \tag{A2}$$

To find the maximum value of $U_D(x)$, the model can be defined as

$$Max\ \lambda \tag{A3}$$

subject to

$$\lambda \leq U_G(x) \tag{A4}$$

$$\lambda \leq U_{C_l}(x), \forall j \tag{A5}$$

$$0 \leq \lambda \leq 1. \tag{A6}$$

2. Min-bound sum operator:

Proposed by Luhandjula [25], this operator uses equation (A7) as the aggregation rule

$$U_D = \gamma \times U_{G \cap C_l} + (1-\gamma) \times U_{G \cup C_l}. \tag{A7}$$

It then uses the min operation to represent the intersection and the bounded sum operation to represent the union; aggregated membership function thus becomes

$$U_D = \gamma \, Min \, U_S + (1-\gamma) Min(1, \sum_{s=0}^{t} U_S). \tag{A8}$$

The equivalent form after using the operator is:

$$Max \;\; \gamma\lambda + (1-\gamma)u \tag{A9}$$

subject to

$$\lambda \leq U_G(x) \tag{A10}$$

$$\lambda \leq U_{C_l}(x), \; \forall j \tag{A11}$$

$$u \leq 1 \tag{A12}$$

$$u \leq U_G + \sum_{l} U_{C_l}(x) \tag{A13}$$

$$0 \leq \lambda \leq 1; \gamma \leq 1. \tag{A14}$$

3. Compensatory and operator:

This operator uses the same aggregation rule as the min-bound-sum operator does, but it then uses the min operation to represent the intersection and the max operation to represent the union. Thus, the result of $U_D(x)$ becomes

$$U_D = \gamma \, Min \, U_S + (1-\gamma) \, Max \, U_S. \tag{A15}$$

The crisp equivalent model after using this operator is:

$$Max\ \ \gamma\lambda + (1-\gamma)u \tag{A16}$$

subject to

$$\lambda \leq U_G(x) \tag{A17}$$

$$\lambda \leq U_{C_l}(x),\ \forall j \tag{A18}$$

$$u \leq U_G + MY_0 \tag{A19}$$

$$u \leq U_{C_l} + M\,Y_{l},\ \forall l \tag{A20}$$

$$\sum_{s=0}^{t} Y_S \leq t \tag{A21}$$

$$0 \leq \lambda \leq 1;\ \gamma < 1,\ Y_S = 0{,}1;\ M \text{ is a large number.} \tag{A22}$$

4. $\tilde{and}$ operator:

To address the deficiencies of aggregation rule (A7), Werners [35] suggested modification of the membership function of the resulting fuzzy set as

$$U_D = \gamma\, Min U_S + {}^{(1-\gamma)}\!\big/_{(t+1)} \times \sum_{s=0}^{t} U_S. \tag{A23}$$

The equivalent model after this operator is used becomes

$$Max\,\lambda + {}^{(1-r)}\!\big/_{(t+1)} \times \sum_{s=0}^{t} \alpha_S \tag{A24}$$

subject to

$$\lambda + \alpha_0 \leq U_G(x) \tag{A25}$$

$$\lambda + \alpha_l \leq U_{C_l}(x),\ \forall l \tag{A26}$$

$$\lambda + \alpha_S \leq 1, \ \forall s \tag{A27}$$

$$\lambda, \alpha_S \geq 0; \ \delta < 1. \tag{A28}$$

5. Add operator:

Proposed by Sommer [30], this operator first modifies the fuzzy constraints $AX \tilde{\leq} b$ into $AX - S_l \leq b$ and $S_l \tilde{\leq} 0$. As a result, the membership function of the constraint set becomes $U'_{cl} = 1 - {S_l}/{P_r}$. Therfore, all membership functions are sumed up in the objective functions; i.e.,

$$Max\, U_G + \sum_{l=1}^{t} U'_{cl} \tag{A29}$$

The equivalent LP model can be obtained as

$$Min\ {(f(x) - Z^0)}/{P_0} + \sum_{l=1}^{t} {S_l}/{P_r} \tag{A30}$$

subject to

$$AX - S_l \leq b, \forall j \tag{A31}$$

$$S_l \leq P_r, \forall j. \tag{A32}$$

References

[1] Buckle, J. J., "Solving Possibilistic Linear Programming Problems," *Fuzzy Sets and Systems*, 31, 329-341 (1989).

[2] Carlsson, C., and Korhonen P., "A Parametric Approach to Fuzzy Linear Programming," *Fuzzy Sets and Systems*, 20, 17-30 (1986).

[3] Chanas, S., "The Use of Parametric Programming in Fuzzy Linear Programming," *Fuzzy Sets and Systems*, 11, 243-251 (1983).

[4] Chanas, S., Kolodziejczyk, W., and Machaj, A., " A Fuzzy Approach to the Transportation Problem," *Fuzzy Sets and Systems*, 13, 211-221 (1984).

[5] Chu, C. H., "Cluster Analysis in Manufacturing Cellular Formation," *OMEGA*, 17, 289-295 (1989)

[6] Chu, C. H., and Hayya, J. C., "A Fuzzy Clustering Approach to Manufacturing Cell Formation," *International Journal of Production Research*, 29, 1475-1487 (1991).

[7] Darzentas, J., "On Fuzzy Location Model," in *Optimization Models Using Fuzzy Sets and Possibility Theory*, Kacprzyk, J. and Orlovski, S. A. (eds.), Dordrecht, D. Reidel Publishing Company, 328-341 (1987).

[8] Deporter, E. L., and Ellis, K. P., "Optimization of Project Networks with Goal Programming and Fuzzy Linear Programming," *Computers and Industrial Engineering*, 19, 500-504 (1990).

[9] Dombi, J., "Membership Function as an Evaluation," *Fuzzy Sets and Systems*, 35, 1-21 (1991).

[10] Dubois, D., "Linear Programming with Fuzzy Data," in *Analysis of Fuzzy Information (Vol. III): Applications in Engineering and Science*, Bezdek, J. C. (ed.), Boca Raton, CRC Press, 241-263 (1987).

[11] Evans, G. W., Karwowski, W. and wilhelm, M. R. (eds.), *Applications of Fuzzy Set Methodologies in Industrial Engineering*, Amsterdam, Elsevier (1989).

[12] Fabian, C., and Stoica, M., "Fuzzy Integer Programming," in *TIMS/Studies in the Management Sciences,* Vol. 20, Zimmermann, H. J., Zadah, L. A. and Gaines, B. R. (eds.), Amsterdam, Elsevier Science, 123-131 (1984).

[13] Gongaware, T. A. and Ham, I., "Cluster Analysis Applications for Group Technology Manufacturing Systems," in *Proceedings of the 9th North American Manufacturing Research Conference (NAMRC),* 503-508 (1981).

[14] Inuiguchi, M., Ichihashi, H. I., and Tanaka, H., "Fuzzy Programming: A Survey of Recent development," in *Stochastic Versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty*, Slowinski, R., and Teghem, J. (eds.), Norwell, Kluwer Academic Publishers, 45-70 (1990).

[15] Karwowski, W. and Evans, G. W., "fuzzy Concepts in Production Management Research: A Review," *International Journal of Production Research,* 24, 129-147 (1986).

[16] Kim, M. H., Lee, J. H., and Lee, Y. J., "Analysis of Fuzzy Operators for High Quality Information Retrieval," *Information Processing Letters*, 46, 252-256 (1993).

[17] King, J. R., "Machine-Component Group Formation in Group Technology," *OMEGA,* 8, 2, 193-199 (1980).

[18] Lai, Y.-J., and Hwang, C.-L., "Interactive Fuzzy Linear Programming," *Fuzzy Sets and Systems*, 45, 169-183 (1992).

[19] Lai, Y.-J., and Hwang, C.-L., "A New Approach to Some Possibilistic Linear Programming Problems," *Fuzzy Sets and Systems*, 49, 121-133 (1992).

[20] Lai, Y.-J., and Hwang, C.-L., *Fuzzy Mathematical Programming--Method and Application*, Berlin, Springer-Verlag (1992).

[21] Leberling, H., "On finding compromise solutions in multicriteria problems using the fuzzy min-operator," *Fuzzy Sets and Systems*, 6, 105-118 (1981).

[22] Li, J., Ding, Z., and Lei, W., "Fuzzy Cluster Analysis and Fuzzy Pattern Recognition Methods for Formation of Part Families," in *Proceedings of the 14th North American Manufacturing Research Conference (NAMRC)*, 558-563 (1986).

[23] Li, R. J., and Lee, E. S., " An Exponential Membership Function for Fuzzy Multiple Objective Linear Programming," *Computers for Mathematical Applications.*, 22, 12, 55- 60 (1991).

[24] Luhandjula, M. K., “Multiple Objective Programming Problems with Possibilistic Coefficients”, *Fuzzy Sets and Systems*, 34, 15-31 (1990).

[25] Luhandjula, M. K., "Compensatory Operators in Fuzzy Linear Programming with Multiple Objectives," *Fuzzy Sets and Systems*, 8, 245-282 (1982).

[26] Mjelde, K. M., "Fuzzy Resource Allocation," *Fuzzy Sets and Systems*, 19, 239-250 (1986).

[27] Negoita, C.V., and Sularia, M., "On Fuzzy Mathematical Programming and Tolerances in Planning," *Economic Computer and Economic Cybernetic Studies and Researches*, 1, 3-14 (1976).

[28] Shafer, S. M., Kern, G. M., and Wei, J. C., "A Mathematical Programming Approach for Dealing with Exceptional Elements in Cellular Manufacturing," *International Journal of Production Research*, 30, 1029-1036 (1992).

[29] Singh, N., "Design of Cellular Manufacturing Systems: An Invited Review," *European Journal of Operational Research*, 69, 3, 284-291 (1993).

[30] Sommer, G., and Pollatschek, M. A., "A Fuzzy Programming Approach to An Air Pollution Regulation Problem," *European Journal of Operational Research*, 10, 303-313 (1978).

[31] Tanaka, H., Ichihashi, H., and Asai, K., "A Formulation of Fuzzy Linear Programming Problem Based on Comparison of Fuzzy Numbers," *Control and Cybernetics*, 13, 3, 185-194 (1984).

[32] Verdegay, J. L., "A Dual Approach to Solve the Fuzzy Linear Programming Problem," *Fuzzy Sets and Systems*, 14, 131-141 (1984).

[33] Wemmerlöv, U., and Hyer, N. L., "Procedures for the Part Family/Machine Group Identification Problem in Cellular Manufacturing," *Journal of Operations Manufacturing*, 6, 125-147 (1986).

[34] Werners, B. M., "An Interactive Fuzzy Programming System," *Fuzzy Sets and Systems*, 23, 131-147 (1987).

[35] Werners, B. M., "Aggregation Models in Mathematical Programming," in *Mathematical Models for Decision Support*, Vol. 48, Mitra Gautan, and Greenberg Harvey J. (eds), Heidelberg, Spring-Verlag, 295-305 (1988).

[36] Wiedey, G., and Zimmermann H. J., "Media Selection and Fuzzy Linear Programming," *Journal of Operational Research Society,* 31, 342-349 (1978).

[37] Xu, H., and Wang, H. P., "Part Family Formation for GT Applications Based on Fuzzy Mathematics," *International Journal of Production Research*, 27, 1637-1651 (1989).

[38] Yang, Taeyong, and Ignizio, James P., "Fuzzy Programming with Nonlinear Membership Functions: Piecewise Linear Approximation," *Fuzzy Sets and Systems*, 41, 39-53 (1991).

[39] Zadeh, L. A., "Fuzzy Sets," *Information and Control*, 8, 338- 353, (1965).

[40] Zhang, C., and Wang, H. P., "Concurrent Formation of Part Families and Machine Cells Based on the Fuzzy Set Theory," *Journal of Manufacturing Systems*, 11, 61-67 (1992).

[41] Zimmermann, H.-J., "Fuzzy Programming and Linear Programming with Several Objective Functions," *Fuzzy Sets and Systems*, 1, 45-55 (1978).

[42] Zimmermann, H.-J., "Applications of Fuzzy Set Theory to Mathematical Programming," *Information Sciences*, 36, 29-58 (1985).

[43] Zimmermann, H.-J., *Fuzzy Set Theory and Its Applications*, Boston, Kluwer Academic Publishers (1991).

II. A FUZZY MULTIOBJECTIVE LINEAR PROGRAMMING MODEL FOR CELL FORMATION

A paper prepared for submission to IIE Transactions

Chang-Chun Tsai, Chao-Hsien Chu

Abstract

Cellular manufacturing (CM) has attracted much attention by both researchers and practitioners while retaining a high level of flexibility because it can lead to reduced throughput time and increased productivity. The realization of CM, however, requires several major decisions, among which cell formation (CF) is the most critical one. Mathematical programming is one of the most popular approaches for modeling the CF problem, but situations in the real world usually cannot be defined precisely, and data regarding them are uncertain. Fuzzy mathematical programming is a promising algorithmic approach to dealing with such a problem. This paper illustrates how a fuzzy multiobjective linear programming (FMLP) approach can be developed to model cell formation problems with two conflicting objective functions – minimizing the total costs of dealing with exceptional elements (EEs) and maximizing group efficacy. A new similarity coefficients formula is proposed to relate group efficacy with similarity coefficients.

Introduction

To produce families of similar parts, cellular manufacturing (CM) – an application of the group technology (GT) philosophy – groups dissimilar machines in physical proximity. The implementation of CM consists of five steps [37]: (1) form part families; (2) form machine cells; (3) choose tools, fixtures, and pallets; (4) select material handling equipment; and (5) layout the equipment. Cell formation (CF) covers the first two steps. Over the past decades, many analytical methods have been proposed and efficient procedures developed for solving the CF problems [8, 9, 32, 38]. Mathematical programming (MP) is one of the most popular methods for CF. According to [9], the following objective functions – total costs of machine investment, total costs of intercell movement, total number of intercell movements, total similarity coefficients between parts, total distances, and total subcontracting costs – have been used frequently in CF modeling. Most are related to the problem of dealing with exceptional elements (EEs). Thus, dealing with EEs is an important objective for CF. Three measures often have been applied to evaluate the performance of CF: cost, group efficacy (GE), and similarity coefficients (SCs). The studies dealing with EEs have arrived at two interesting results: minimizing the cost of dealing with EEs often results in smaller number of cells and lower GE value, as does maximizing total SC value. Clearly, the total costs of dealing with EEs and GE are difficult to optimize simultaneously and the optimal cell number is hard to determine either in the CF problem. Therefore, how to obtain a trade-off between GE and the cost of dealing with EEs becomes an attractive but thorny issue. By use of an improved similarity-coefficient formula, it is possible to solve the controversy between SCs and GE so that the trade-off between cost and GE can be obtained by maximizing total SC and minimizing total costs. Although a number of studies [23, 26, 29, 36] have used multiobjective linear programming (MLP) to solve the CF problem, none has solved this attractive but controversial issue. The current research is the first to propose a goal programming model to obtain the trade-off between the cost of dealing with EEs and GE. Meanwhile, past studies have been based on the unrealistic assumptions that objective functions and constraints of the mathematical model can be defined precisely. In practice, however, it is quite difficult for the decision maker to specify exact goals and constraints in the modeling of MLP. Furthermore, according to our previous study [35], fuzzy mathematical programming is more stable and efficient than traditional programming for modeling the single objective CF problem. In this instance, fuzzy set theory also can be applied to MLP. Zimmermann was the first researcher to use fuzzy multiobjective linear programming (FMLP) in vectors programming [42]. Since then, FMLP has been applied to many fields – project network [10], reliability optimization [27], and the transportation problem [1], to name a few. None, however, has attempted to apply the FMLP to CM system design.

The trade-off between minimizing the total costs of dealing with EEs and maximizing the total SCs of parts in all groups is unclear. In the CF problem, moreover, the number of machines in each cell is unknown. The objective of this research is to apply the FMLP to find the optimal trade-off between multiple goals under both crisp and fuzzy constraints. In the next section, the key issues of using FMLP will be examined. The details of formulating goal programming (GP) and FMLP with the new operator in [35] will be discussed in the third section. After illustrating and analyzing three numerical examples, conclusions will be discussed in the last section.

Fuzzy Multiobjective Linear Programming

Fuzzy mathematical programming (FMP) is one of the simpler and more efficient approaches to solving multiple-objective decision making (MODM) problems under a fuzzy environment. Alternative methods include preemptive fuzzy goal programming, interpolated membership function, weighted additive modeling, a preference structure on aspiration levels, and nested priority [16]. The problem style, model style, and solving process of FMLP modeling are similar to those of FLP modeling and were discussed in [35]. The only characteristic of FMLP different from that of FLP is the implementing process. But no matter which FMLP approach is used, the general four-stage solving procedure depicted in Figure 1 and outlined below can be followed:

1. Initialization: The first stage is to model the problem using the standard traditional MLP method.
2. Fuzzification: This stage consists of three-steps: (1) determining the attributes to be fuzzified; (2) developing the FMLP formulation according to the fuzzy attributes in MLP; and (3) choosing the corresponding membership functions for the linguistic input values and the fuzzy relations in FMLP. The process then can be classified into two problem styles – vagueness and ambiguity [35, 42]. For vagueness programming problems, most research has applied a linear nonincreasing membership function to the fuzzy objective functions [1, 10, 27, 42] and to the constraints [25] because this membership function has the linear form and its linguistic can be expressed as essentially "less than" or "more than" a certain objective value. One paper [17] used the S shape function, and another [18] the

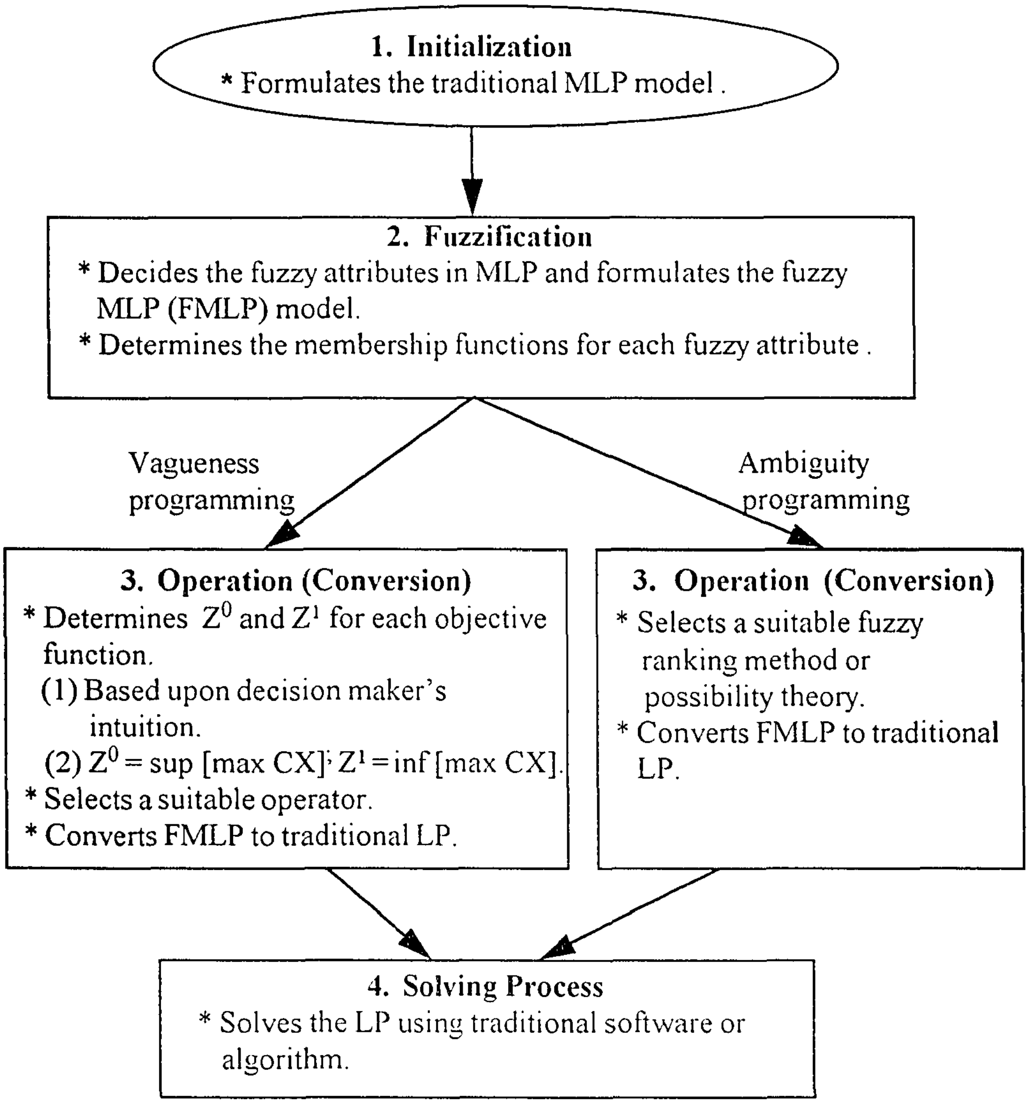

1. Initialization
* Formulates the traditional MLP model .
2. Fuzzification
* Decides the fuzzy attributes in MLP and formulates the fuzzy MLP (FMLP) model.
* Determines the membership functions for each fuzzy attribute .
Vagueness programming
Ambiguity programming
3. Operation (Conversion)
* Determines Z^0 and Z^1 for each objective function.
(1) Based upon decision maker's intuition.
(2) Z^0 = sup [max CX]; Z^1 = inf [max CX].
* Selects a suitable operator.
* Converts FMLP to traditional LP.
3. Operation (Conversion)
* Selects a suitable fuzzy ranking method or possibility theory.
* Converts FMLP to traditional LP.
4. Solving Process
* Solves the LP using traditional software or algorithm.

Figure 1: The Generic Procedure for FMLP Modeling

exponential distribution in their objective functions. Both have the nonlinear form and are more suitable for representing real-life cases such as cost or utility functions. For ambiguity programming, the trapezoid membership function often has been used as the L-R fuzzy number of parameters [24]. L and R are two functions for the left side and right side of a membership function, respectively.

3. Operation (Conversion): If traditional software is to be used to solve the FMLP, it must be converted to traditional mathematical programming. For the conversion process, selecting a suitable operator is a key issue in vagueness programming. On the other hand, selecting a suitable fuzzy ranking method or possibility theory also is necessary. Also vagueness programming must define the intersection, or "AND," of fuzzy sets; it thus needs an operator to aggregate all fuzzy objective functions.

Seven operators were listed and compared in [35] in the modeling of a single objective FLP. As for the FMLP, most studies [1, 10, 12, 17, 27, 39] applied the max-min operator; two [18, 42] used the product operator, γ ; and one paper examined min-bound sum operators [20]. Although the add-min operator is new, it has been verified as superior to other operators [35] and will be adapted and used in the FMLP model to compare with the performance of goal programming. Furthermore, if the FMLP models are to be solved, the aspired level of achievement, Z_k^0, and the least acceptable level of achievement, Z_k^1, must be decided for each objective k. They can be either assigned by the decision maker or obtained by running the LP with each single objective function.

To compare the fuzzy number in the constraints, some fuzzy ranking methods have been proposed in [7]. Two possibility theories, α feasible and β efficiency [19], also have been used to solve the model where the objective function has fuzzy parameters. Because the model to be discussed considers only fuzzy goals and fuzzy constraints, the FMLP model can be viewed as an example of vagueness programming.

4. Solving process. Choose a powerful software or algorithm to solve the traditional LP and obtain the optimal results. In this paper, a PC version of LINDO (Linear interactive and discrete optimizer) package is used.

Mathematical Formulations

We use the following notation to model our problems:

Indexing Sets

i machine index; $i = 1,\dots, m$

j, j' part index; $j, j' = 1,\dots, n$

k cell index; $k = 1, \dots, c$

s index of fuzzy constraints; $s = 1,\dots, c$

Parameters

A_i the periodic cost of acquiring a machine type i

a the number of machines, both parts j and j' tour

b the number of machines, only part j' tours

c the number of machines, only part j tours

C_i the periodic capacity of machine type i

D_j the periodic forecasted demand for part *j*

d the number of machines not required by either part *j* or *j'*

I_j incremental cost for moving a unit of part *j* within two cells

NM the maximum number of machine types allowed in each cell

P_c the tolerance value for the objective function of cost

P_s the tolerance value for the objective function of SCs

P_{ij} processing time of machine type *i* needed to produce part *j*

S_j incremental cost of subcontracting a unit of part *j* for an operation

$SC_{jj'}$ the similarity coefficients between part *j* and part *j'*

SP set of pairs (*i,j*) such that $a_{ij} = 1$

UC_{ij} the utilization capacity of machine type *i* for the parts *j*

The value can be calculated by the equation: $P_{ij} \times D_j / C_i$

$U_D(x)$ membership function of aggregated results

$U_G(x)$ membership function of the specified objective function

$U_s(x)$ *sth* membership function

Wc the priority weight for the cost function

Ws the priority weight for the similarity function

Z_c^1 the worst (largest) value of cost function

Z_s^1 the worst (smallest) value of similarity coefficient function

Z_c^0 the best (smallest) value of cost function

Z_s^0 the best (largest) value of similarity coefficient function

Decision variables

dc^+ the deviation variable for the cost function

ds^- the deviation variable for the similarity function

IC_k =1, if cell k is formed; 0, otherwise

M_{ijk} number of machine i dedicated to cell k for producing part j

O_{ijk} units of part j to be subcontracted as a result of machine type i not being available within cell k

Q_i number of machine type i needed to process it's corresponding parts in its machine cell

R_{ik} number of machine type i to be dedicated in cell k

$SY_{jj'}$ = 1, if $Y_j = 1$ and $Y_{j'} = 1$; 0, otherwise

U_{ijk} = 1, if $X_{ik} = 1$, and $Y_{jk} = 0$; 0 otherwise (i.e., machine i is an exceptional machine)

V_{ijk} = 1, if $Y_{jk} = 1$, and $X_{ik} = 0$; 0 otherwise (i.e., part j is an exceptional part)

X_{ik} = 1, if machine i is assigned to cell k; 0, otherwise

Y_{jk} = 1, if part j is assigned to cell k; 0, otherwise

Z_{ijk} number of intercellular transfers required by part j as a result of machine type i not being available within the part cell k

λ minimum value of all membership functions

The Goal Programming Model

The purpose of this model is to obtain a trade-off between the maximum total SCs of parts and the total costs of dealing with EEs. This model has three unique characteristics: (1) It can group parts into part families and form machines into machine cells simultaneously. (2) It obtains the minimum total costs of dealing with EE and also considers the available capacity of machines and their investment cost. (3) It can obtain a reasonable relation between two criteria (SCs and GE) – the larger the SC value, the larger the GE value. The formulation is given below:

$$Min\ W_c \cdot \frac{dc^+}{P_c} + W_s \cdot \frac{ds^-}{P_s} \tag{1}$$

subject to

$$\sum_k \sum_i A_i R_{ik} + \sum_k \sum_{(i,j)\in sp} I_j Z_{ijk} + \sum_k \sum_{(i,j)\in sp} S_j O_{ijk} - dc^+ = Z_c^0, \tag{2}$$

$$\sum_{k=1}^{c} \sum_{j=1}^{n} \sum_{j\neq j'}^{n} SC_{jj'} Y_{jk} Y_{j'k} + ds^- = Z_s^0, \tag{3}$$

$$\sum_{k=1}^{c} X_{ik} = 1,\ \forall i, \tag{4}$$

$$\sum_{k=1}^{c} Y_{jk} = 1,\ \forall j, \tag{5}$$

$$\sum_{i=1}^{m} X_{ik} \leq NM,\ \forall k, \tag{6}$$

$$X_{ik} - Y_{jk} + \frac{1}{D_j} Z_{ijk} + \frac{1}{D_j} O_{ijk} {}_{i} + \frac{1}{UC_{ij}} M_{ijk} - U_{ijk} = 0\ \forall (i,j) \in sp,\ \forall k, \tag{7}$$

$$\sum_{(i,j)\in sp} M_{ijk} \leq R_{ik},\ \forall i,\ \forall k, \tag{8}$$

$$Q_i \leq \sum_{(i,j)\in sp} UC_{ij}(1 - \sum_k V_{ijk}) + 1,\ \forall i, \tag{9}$$

$$\sum_k \sum_{(i,j)\in sp} \frac{P_{ij}}{C_i} Z_{ijk} \leq Q_i - \sum_{(i,j)\in sp} UC_{ij}(1 - \sum_k V_{ijk}),\ \forall i, \text{ and} \tag{10}$$

$$X_{ik}, Y_{jk}, U_{ijk}, V_{ijk} = 0 \text{ or } 1; \quad R_{ik}, Q_i = \text{general integer.} \tag{11}$$

The proposed model treats at the tolerance value of every objective function as a scale; thus, different unit objective functions can be aggregated. The formulations of the proposed model are similar to those of the weighted additive model of fuzzy goal programming in [17], which is more efficient and easier to understand. After dividing by the related scale, P_c and P_s, total deviation value for total costs and SCs is minimized in Equation (1), in which P_c and P_s must be calculated first. They are equal to the difference of the best goal (Z^0) and the worst goal (Z^1). Thus, Z_c^0 and Z_c^1 are the feasible minimum cost and maximum cost of dealing with EE, respectively. Similarly, Z_S^0 and Z_S^1 are the feasible largest and smallest SC values. Three types of costs associated with EEs are to be minimized to approach Z_c^0 in constraint (2). The first type assesses the cost of duplicating a machine. The second totals the costs of intercellular transfers for the EEs. The last evaluates subcontracting cost. Total similarity coefficients of the pairs of parts in all groups are maximized to Z_S^0 in (3). Constraint sets (4) and (5) ensure that each machine and part is assigned into only one cell. Constraint set (6) is designed to prevent assignation of more than "*NM*" machines to each cell. Constraint set (7) is the combination of equations (7a) and (7b):

$$X_{ik} - Y_{jk} + U_{ijk} - V_{ijk} = 0, \ \forall (i,j) \in sp, \forall k \tag{7a}$$

$$Z_{ijk} + O_{ijk} + \frac{C_i}{P_{ij}} M_{ijk} = D_j U_{ijk}, \ \forall (i,j) \in sp, \ \forall k, \tag{7b}$$

where (7a) ensures that an EE is either an exceptional machine (a machine to be duplicated) or an exceptional part (a part to be transferred or subcontracted) and where (7b) further guarantees that the demand of exceptional part j can be shared by the combination of duplicated machine i transferred within cells and subcontracted. Constraint set (8) calculates the number of machine type i needed to be dedicated in cell k for producing the EE, where M_{ijk} is a real variable representing the percentage of utilization capacity of a machine type i dedicated to process part j in cell k. Constraint set (9) determines the number of machine type i needed in each cell. It sums all the utilization capacities of machine type i for all relative parts j ($\sum_{(i,j)\in sp} UC_{ij}$) not belonging to the EEs ($-\sum_k V_{ijk}$). Constraint set (10) ensures that the numbers of intercellular transfers between machines type i do not exceed the available machine capacity, where $\frac{C_i}{P_{ij}}$ multiplied by the right side of equation (10) is the maximum available units of machine type i if the relevant parts are to transfer. Obviously, constraint sets (9) and (10) are the upper-bound and the lower-bound of Q_i, respectively.

The Fuzzy Model

The aforementioned traditional model assumes that objectives and constraints can be defined precisely; however, they cannot. For instance, constraint sets (2) and (3) are fuzzy and can affect each other. Constraint set (6) is fuzzy, too, because the NM value is difficult to determine in practice. They therefore can be fuzzified and expressed as

$$\sum_k \sum_i A_i R_{ik} + \sum_k \sum_{(i,j)\in sp} I_j Z_{ijk} + \sum_k \sum_{(i,j)\in sp} S_j O_{ijk} \mathrel{\tilde{\leq}} Z_c^1 - P_c, \qquad (12)$$

$$\sum_{k=1}^{c} \sum_{j=1}^{n} \sum_{j' \neq j}^{n} SC_{jj'} Y_{jk} Y_{j'k} \tilde{\geq} Z_S^1 + p_S, \text{ and} \qquad (13)$$

$$\sum_{i=1}^{m} X_{ik} \cong NM, \ \forall k. \qquad (14)$$

The two fuzzy inequations, (12) and (13), can be converted easily to the conventional formulation by applying a suitable operator after p_c and p_S are calculated. The values of p_c, p_S, Z_c^1, and Z_S^1 are the same as those in goal programming. Different operators can result in different formulations and thus achieve different results. Add-min operator was applied in this paper due to its superior performance than others for the single-objective CF models [35].

Two fuzzy cases are considered in this study. The first case is an asymmetric model, which only fuzzifies the two objective functions (12) and (13). The second case, called a *symmetric model*, further considers the possible impacts of the fuzzy constraint (14). Properties of several membership functions involving linear nonincreasing, triangular, trapezoidal, exponential, and S shapes were discussed in [35]. Because the triangular membership function is a reasonable one to represent "around" ($\cong$) and has been shown to perform better than the linear nonincreasing membership function for the fuzzy constraints [35], the second case considers only the triangular membership function in fuzzy constraints. For the objective function, the linear nonincreasing function is appropriate for the minimum function, and the linear nondecreasing function is suitable for the maximum function. Hence, this paper applies the linear nonincreasing function and the linear nondecreasing function in

(12) and (13), respectively. The transformation formula in Zimmermann [43] will be used for these two membership functions.

Case 1: An asymmetric model with an add-min operator. According to the properties of this operator (see Appendix), formulation after membership function aggregation yields

$$U_D = U_G + \underset{s=1}{Min}\, U_s . \qquad (15)$$

To apply (15) for the FMLP, we have to select and keep only one objective as U_G and dispose the remaining objective functions as fuzzy constraints. In this paper, considering computational efficiency, the cost membership function is selected as U_G, and the membership function of SCs is merged in U_S. The equivalent transformed formulation, according to the transferred results in the Appendix, becomes

$$MIN \sum_{k}\sum_{i} A_i R_{ik} + \sum_{k}\sum_{(i,j)\in sp} I_j Z_{ijk} + \sum_{k}\sum_{(i,j)\in sp} S_j O_{ijk} - \lambda P_c \qquad (16)$$

subject to

$$\sum_{k=1}^{c}\sum_{j=1}^{n}\sum_{j'\neq j}^{n} SC_{jj'} Y_{jk} Y_{j'k} - \lambda P_s \geq Z_S^1 . \qquad (17)$$

The complete formulation includes (16), (17), (4), (5), and (7) - (11).

Case 2: A symmetric model with an add-min operator. The difference between cases 1 and 2 is that the latter case further considers the fuzzy constraint of (14), which uses the triangular membership function. The transformation formula in [41] is modified to prevent certain machines' being assigned to cells not yet formed. Equivalent formulations can be obtained as

$$\sum_{i=1}^{m} X_{ik} + \lambda P_r - NM \times IC_k \le P_r, \forall k, \tag{18}$$

$$\sum_{i=1}^{m} X_{ik} - \lambda P_r - NM \times IC_k \ge -P_r, \forall k, \text{ and} \tag{19}$$

$$\sum_{k=1}^{c} IC_k \ge 2 \ , \tag{20}$$

where the IC_k variables in equations (18) and (19) are added to ensure that $\sum_{i=1}^{m} X_{ik}$ is equal to zero when cell k is not yet formed. Constraint set (20) prevents clustering results in fewer than two cells. The complete formulation includes (16), (18) - (20), (4), (5), and (7) - (11).

Please note that Equations (3) and (17) have a nonlinear term, $Y_{jk} Y_{jk}'$, and cannot be solved by a linear programming package such as LINDO. Theoretically, we can transform the nonlinear term into a linear one by assuming that $Y_{jk} Y_{jk}'$ is equal to SY_{jj}' and subsequently adding the two constraints (21) and (22):

$$Y_{jk} + Y_{j'k} \ge 2 SY_{jj}' \text{ and} \tag{21}$$

$$Y_{jk} + Y_{j'k} \le 1 + SY_{jj}'. \tag{22}$$

The Proposed Similarity Coefficients

Total similarity and group efficacy (GE) are two popular criteria for measuring the performance of CF problems. The SC describes the relation between pairs of parts. In general, the closer the relation, the higher the SC value. Group efficacy is a quantitative criterion for measuring the goodness of block diagonalization of binary matrices [14]. Shafer and Rogers [31] surveyed and analyzed most SCs used previously. But most SC formulae produce a common result, i.e., the fewer the number of cells, the higher the SCs and the lower the GE. In other words, one cell is the best answer when maximizing total SCs. This result is inconsistent with the grouping principle of CF. Hence, to rectify this weakness, the current research redefines the following formula for SCs between parts *j* and *j'*:

$$\begin{aligned} SC_{jj'} &= (2a-b-c)/(2a+b+c) && \text{for all } j \neq j' \\ &= 0 && \text{otherwise.} \end{aligned} \tag{23}$$

According to [14], GE is equal to (total number of operations - number of EE)/ (total number of operations + number of voids). Thus, Equation (23) can be derived as follows: for parts *j* and *j'*, the total number of operations is "$2a + b + c$." If we assume that only "*a*" machines are assigned in the diagonal blocks, then the number of EEs is "$b + c$," and the number of voids in the diagonal blocks is zero. Furthermore, to ensure that the range of SCs is between -1 and +1, we assume that the penalty weight of EEs is "two"; hence, the GE formula becomes $[(2a+b+c)-2(b+c)]/(2a+b+c+0)$.

According to equation (23), the SC value is equal to -1 if two parts are entirely dissimilar, e.g., $(-b-c)/(b+c)$. The SC value is 1 ($2a/2a$) when two parts are entirely similar.

In general, total SC will be positive if GE value exceeds 0.6. Total SCs will be negative if GE value is less than 0.45.

To assess the appropriateness of our proposed similarity coefficient, 16 data sets from the open literature are used for the evaluation. We adapted two different numbers of cells for each data set. Their performances are contrasted with six other popular similarity coefficients [22, 31], including

(1) Sorenson: $2a/(2a+b+c)$,

(2) Jaccord: $a/(a+b+c)$,

(3) Russell and Rao (R & R): $a/(a+b+c+d)$,

(4) Phi: $(ad-bc)/\sqrt{(a+b)(a+c)(b+d)(c+d)}$,

(5) Yule: $(ad-bc)/(ad+bc)$, and

(6) Hamann: $[(a+d)-(b+c)]/[(a+d)+(b+c)]$.

The range of SC values for the first three formulae is [0, 1]; for the last three, it is [-1, 1]. Table 1 summarizes comparison results. As shown, the proposed formula is the only one that can meet the principle disputed early; namely, SC value grows as GE does. The three coefficients with the range [0, 1] (Sorenson, Jaccord, and R & R) cannot be used because their similarity values are not always correlated with GE value although they are in data sets 8 and 12, in which their GE values decreased when cell number increased. The other three coefficients with SCs ranging from -1 to 1 are performed better but still cannot meet the principle for all cases. The ratios of match for Yule, Phi, and Hamann are 13/16, 11/16, and 5/16, respectively. So if we use the proposed SCs to present the GE value in goal

Table 1. Comparison Among Different Similarity Coefficients

Data set [Ref.] (Size)	CF results (# of Machines in each cell)	No. EE	GE	Total Similarity						
				Proposed (-1, 1)	Sorenson (0, 1)	Jaccord (0, 1)	R & R (0, 1)	Phi (-1, 1)	Yule (-1, 1)	Hamann (-1, 1)
1. [4] (8 X 20)	(2,6) (4,2,2)	5 9	0.63 0.85	24.59 42.73	57.8 51.87	49.65 46.03	21.75 19.0	36.43 47.87	40.79 58.78	36.5 47.25
2. [11] (9 X 9)	(4,5) (4,3,2)	3 6	0.66 0.74	3.207 4.63	9.602 7.317	7.483 6.05	3.885 2.997	6.093 5.935	8.751 8.418	6.224 5.78
3. [30] (9 X 10)	(5,4) (5,2,2)	5 8	0.45 0.48	-2.65 -0.39	8.67 6.804	6.12 4.97	2.64 2.33	3.867 3.586	5.82 5.53	5.775 4.443
4. [28] (10 X 12)	(3,7) (2,3,5)	0 1	0.56 0.78	1.908 10.265	16.454 15.632	13.493 13.017	5.2 4.9	11.165 14.44	11.87 19.72	13.4 14.8
5. [21] (12 X 10)	(9,3) (5,4,3)	0 5	0.53 0.72	-2.827 5.26	10.586 8.63	8.38 7.2	3.667 2.83	4.416 7.70	2.945 10.74	6.0 7.67
6. [40] (12 X 19)	(9,3) (7,3,2)	9 17	0.48 0.52	2.728 10.715	44.86 36.857	33.14 27.68	16.0 13.67	24.22 21.98	35.3 33.53	29.33 24.67
7. [13] (14 X 24)	(5,5,4) (4,4,3,3)	2 2	0.47 0.67	2.389 20.934	49.21 46.48	39.581 37.913	9.275 8.707	42.17 43.56	45.67 58.62	64.134 54.702
8. [3] (15 X 10)	(5,5,5) (5,5,3,2)	0 7	0.92 0.83	9.39 6.83	10.69 7.41	9.8 7.0	3.27 2.33	10.24 7.218	11.81 7.935	10.53 7.33
9. [33] (16 X 30)	(8,6,2) (6,4,4,2)	16 19	0.46 0.67	-49.85 20.396	68.57 61.2	53.11 48.61	18.06 16.31	38.7 51.5	16.219 70.3	83.75 64.75
10. [2] (16 X 43)	(7,7,2) (7,4,3,2)	17 21	0.38 0.42	-72.75 -61.07	136.12 132.46	100.02 97.16	27.31 26.69	101.93 101.6	102.83 110.98	198.63 193.13
11. [2] (20 X 35)	(10,5,5) (5,5,5,5)	1 2	0.53 0.76	2.647 63.96	101.82 100.97	86.19 85.72	20.75 20.55	83.411 97.52	63.13 126.5	127.4 112.9
12. [5] (24 X 40)	(5,4,4,4,3,2,2) (5,4,4,3,2,2,2, 2)	0 9	0.93 0.88	83.75 78.8	92.88 86.4	88.08 82.58	11.29 10.38	92.63 86.32	99.78 91.87	98.42 91.25
13. [5] (24 X 40)	(6,5,3,3,3,2) (5,4,4,4,3,2,2)	9 10	0.64 0.83	40.74 65.004	86.37 83.5	76.37 74.5	11.375 11.04	79.751 81.93	77.398 97.3	105.75 91.99
14. [15] (30 X 41)	(11,9,8,2) (9,9,5,2,5)	8 9	0.31 0.39	-118.88 -72.52	62.56 58.74	44.32 41.78	7.67 7.4	46.58 47.157	-13.45 25.56	173.4 136.13
15. [34] (30 X 50)	(13,7,6,4) (11,7,6,4,2)	0 10	0.36 0.39	-98.34 -70.2	105.83 97.4	74.82 69.34	11.03 9.97	83.07 78.99	99.132 106.57	221.93 198.87
16. [6] (40X100)	(8,6,5,5,4 4,3,3,2) (6,5,5,5,4,4, 3,3,3,2)	35 36	0.76 0.84	317.52 350.11	437.28 435.55	385.94 384.93	48.53 48.38	9488.3 9543.5	488.33 514.12	504.3 481.8
No. of Match				16	2 @	2 @	2 @	11 *	13 *	5 @

@ Cases in shadow indicate that they match the principle.

* Cases in shadow indicate that they do not match the principle.

programming and FMLP models, a trade-off between total costs and GE would be achievable. The optimal number of cells also would be obtainable automatically.

Numerical Illustration

To illustrate the performance of the proposed fuzzy CF models, this paper considers three data sets from [30, 11, 13]. These sets were examined in [35] for fuzzy CF problems with single-objective. The complexity of data sets I, II, and III in terms of percentage of EEs, the number of EEs divided by the total number of elements, is 28.6%, 18.8%, and 3.3%, respectively. It is assumed that the desired maximum number of cells "*C*" for the data set is "2" or "3" and that the maximum number of machines "*NM*" allowed in each cell is "4." Table 2 summarizes the results from seven different cases, where cases 1 and 2 have been discussed in the previous section and where cases 3, 4, and 5 are based on the goal programming model. The ratio among the weight of cost function and the similarity function in cases 3, 4, and 5 are 0.5, 2, and 1, respectively. Case 6 is a single objective formulation designed to maximize the total SCs of parts. As such, it considers only constraint (5) in goal programming and uses (3) as the objective function. Similarly, the objective of case 7 is to minimize the total costs of dealing with EEs; thus, constraints (1) and (3) should be deleted from the goal programming model and considers (2) as the objective function.

These results were obtained by running the LINDO (Linear Interactive and Discrete Optimizer) package in an IBM compatible 486 DX2 66MHZ computer. Computational performances are measured in terms of GEs, SCs, number of EEs, executing times, and total costs of dealing with EEs. Several observations can be made based on the results.

Table 2: Summary of Results for Traditional and Fuzzy Models

Data set (Size)	**Cases**	**Costs of dealing with EE**	**SCs**$^{+}$	**GE**$^{\&}$	**Executing CPU time (h: m: s)**	**# of EE**	**{Machine/Parts}**

I
(9 X 10)
Pc
(226,609)
Ps
(5.87)
	1. FMLP (Asymmetric)	$325,892	-2.66	0.48	(1:25:38)	5	C1:{1,2,3,4,5/1,2,3,4,6}
							C2:{6,7,8,9/5,7,8,9,10}

	2. FMLP (Symmetric)	$325,892	-2.66	0.48	(1:26:09)	5	Same as case 1.

	3. GP (Ws=2Wc)	$436,447	-0.39	0.49	(68:57:30)	8	C1:{1,2,3,4,5/1,2,3,4,6}
							C2:{7,8/5,8}
							C3:(6,9/7,9,10}

	4. GP (Ws=Wc)	$325,892	-2.66	0.48	(68:54:51)	5	Same as case 1.
	5. GP (Ws=Wc/2)	$325,892	-2.66	0.48	(67:06:19)	5	Same as case 1.

	6. LP (Max SCs)	$526,834	0.41	0.54	(1:05:35)	9	C1:{1,2,3,5/1,3,4,6}
							C2:{4,7,9/5,7,8,10}
							C3:{6,8/2,9}

	7. LP (Min Cost)	$300,125	-5.46	0.43	(0:18:10)	5	C1:{2,3,4,5,7,8/1,2,4,5, 6,8}
							C2:{1,6,9/3,7,9,10}

II
(9 X 9)
Pc
(143,527)
Ps
(3.26)
	1. FMLP (Asymmetric)	$168.200	3.21	0.66	(0:13:42)	3	C1:{1,2,6,9/1,2,6,9}
							C2:{3,4,5,7,8/3,4,5,7,8}

	2. FMLP (Symmetric)	$168.200	3.21	0.66	(0:13:46)	3	Same as case 1.
	3. GP (Ws=2Wc)	$168,200	3.21	0.66	(0:41:48)	3	Same as case 1.
	4. GP (Ws=Wc)	$168,200	3.21	0.66	(0:37:24)	3	Same as case 1.
	5. GP (Ws=Wc/2)	$168,200	3.21	0.66	(0:52:48)	3	Same as case 1.

	6. LP (Max SCs)	$289,160	4.63	0.74	(0:11:26)	6	C1:{1,5/1,5}
							C2:{2,6,9/2,6,9}
							C3:{3,4,7,8/3,4,7,8}

	7. LP (Min Cost)	$145,533	1.37	0.60	(0:02:45)	3	C1:{1,3,4,5,7,8/1,3,4,5, 7,8}
							C2:{2,6,9/ 2,6,9}

Table 2: (Continued)

Data set	Cases	Costs of dealing with EE	SCs[+]	GE[&]	Executing CPU time (h: m: s)	# of EE	{Machine/Parts}

III
(14 X 24)
Pc
(209,660)
Ps
(18.52)
	1. FMLP
	(Asymmetric)
		$67,758	2.38	0.527	(18: 45:34)	2	C1:{1,4,5,7,13/1,2,6,7,8, 17,18,19,20,23}
							C2:{2,3,10,11/3,4,21,24}
							C3:{6,8,9,12,14/5,9,10, 11,12,13,14,15, 16,22}

	2. FMLP (Symmetric)	$67,758	2.38	0.527	(19: 27:39)	2	Same as case 1.
	3. GP **(Ws=2Wc)**	–	–	–	–	–	–
	4. GP **(Ws=Wc)**	–	–	–	–	–	–
	5. GP **(Ws=Wc/2)**	–	–	–	–	–	–

	6. LP
	(Max SCs)
		$209,660	20.9	0.67	(2:23:00)	2	C1:{1,12,13/6,7,8,18}
							C2:{2,3,10,11/3,4,21,24}
							C3:{4,5,7/1,2,17,9, 20,23}
							C4:{6,8,9,14/5,9,10, 11,12,13,14,15, 16,22}

	7. LP
	(Min Cost)
		0	2.38	0.526	(0:15:22)	0	C1:{1,4,5,7,12,13/1,2,6, 7,8,17,18,19,20,23}
							C2:{2,3,10,11/3,4,21,24}
							C3:{6,8,9,14/5,9,10,11, 12,13,14,15,16,22}

Note:

+ SCs: Similarity coefficients.

& GE: Group efficacy.

- The executing CPU time is more than four days.

First, When SC and GE values are compared in all cases, they are correlated significantly. This result satisfies the principle mentioned earlier – as SC value grows, so does GE value. For instance, Table 2 shows that in all data sets no SC decreases as GE grows. The worse situation occurs in data set III, where GE values of cases 1 and 2 are larger than the value of case 7 (0.527 vs. 0.526) whereas their SC values are the same. This occurs because cases 1 and 2 in data set III have the same number of part families and the same parts even though the machines in machine cells are different. The clustering result, however, also obtains a trade-off value between cost and GE. This fact proves that Equation (26) is a feasible formation to be applied in the objective function to optimize GE evaluation.

Second, SC and GE values and the total costs of dealing with EEs in cases 6 and 7 are, respectively, the upper bound and the lower bound of these in all cases. Based on the results of these two cases, *Pc* and *Ps* can be calculated. As can be seen from Table 2, all costs and GE values of cases 1 - 5 are between those of cases 6 and 7.

Third, only in the higher complex data set will changing W_S/W_C value obtain different clustering results. For instance, in data set I, which has the greatest complexity of all sets, the clustering results of case 3 which have the highest SC and GE values and costs as well as the related W_S/W_C value. The GE and SC values of case 3 are closer to those of case 7 than to those of cases 4 and 5 because the W_S/W_C value is the highest in case 3. In data sets II and III, which are less complex, the clustering results of all three cases are no different, because few of their clustering results can satisfy the trade-off condition.

Fourth, in terms of clustering results, the proposed FMLP is equivalent to goal programming, when *Wc* is equal to *Ws* (cases that *Wc* is not equal to *Ws* will be discussed in the section of sensitivity analysis). But in terms of computational efficiency, FMLP models outperform goal programming, as can be seen from Table 2, in data sets I and II. The clustering results of cases 1 and 2 are the same as those of cases 4. Execution time of FMLP in data set I are much shorter than that of goal programming (85.5 minutes and 4134.8 minutes, respectively). In data set III, the execution time of goal programming is more than four days that is much longer than that of FMLP (18.5 hours).

Fifth, in terms of computational efficiency, the asymmetric models performed better than the symmetric model. Table 2 shows that all clustering results of these two models are the same, but the execution time of the asymmetric model (case 1) in all data sets is shorter than that of the symmetric model (case 2). This result will be further confirmed in the next section. Furthermore, from a theoretical point of view, the asymmetric model is much easier to implement than the symmetric model because the former does not involve the fuzzy constraints of the *NM* value. Thus, the asymmetric model may be a better choice.

Finally, even though FMLP can be used to model the CF problem in a fuzzy environment and can improve computational performance, it still requires much time to solve larger data sets. For instance, data set III has 14 machines and 24 parts that can be assigned to three or four cells; thus, the model requires 2600 constraints and 1570 variables, which makes execution time uneconomical. Such complexity is, however, a common weakness of MP.

Sensitivity Analysis

To examine the impacts of P_c/P_s values on clustering results, a sensitivity analysis was conducted. Table 3 summarizes the results with different P_c/P_s values and models. For the convenience of implementation, we fixed the P_s value and changed the P_c value only. A number of observations can be drawn.

First, P_c/P_s value will affect execution time significantly. Each data set affects execution time uniquely. For instance, in data set I, a complex set, a higher P_c/P_s value will obtain a greater number of cells. Thus, computation time will increase. In other words, the higher the P_c/P_s value, the longer the execution time. For data set II, execution time is longer when P_c/P_s is lower.

Second, different P_c/P_s values will obtain different clustering results for the higher complex data set. For instance, cases 1 and 4 in data set I perform better in term of GE but worse in terms of cost and EE numbers. However, clustering results for all cases in data sets II and III are entirely the same. In other words, for the lower complex data set, P_c/P_s value has no major impact.

Finally, there is a correlation between varying the P_c/P_s value and the W_s/W_c value. Both have the same effect on overall performance, such as GE, SC, EE numbers, and cost. Furthermore, clustering results obtained by changing P_c/P_s in the fuzzy asymmetric model are similar to those obtained by changing W_s/W_c in goal programming model, for neither

Table 3: Summary of Results for Fuzzy Models with Different P_c/P_s Values

Data set	P_c/P_s	Model style	Cost of dealing with EE	SCs	GE	Executing time (h: m:s)	# of EE	{Machine/Parts}

I
Pc
(226609)
Ps
(5.87)
	1. 2	Asymmetric	436,447	-0.39	0.49	1:27:30	8	C1:{1,2,3,4,5/1,2,3,4,6}
								C2:{7,8/5,8}
								C3:(6,9/7,9,10}

	2. 1	Asymmetric	325,892	-2.66	0.48	1:25:38	5	C1:{1,2,3,4,5/1,2,3,4,6}
								C2:{6,7,8,9/5,7,8,9,10}

	3. $\frac{1}{2}$	Asymmetric	325,892	-2.66	0.48	1:02:15	5	Same as case 2.

	4. 2	Symmetric	469,430	-0.39	0.50	1:38:19	9	C1:{1,2,3/1,2,3}
								C2:{4,5,7,8/4,5,6,8}
								C3:(6,9/7,9,10}

	5. 1	Symmetric	325,892	-2.66	0.48	1:26:09	5	Same as case 2.
	6. $\frac{1}{2}$	Symmetric	325,892	-2.66	0.48	1:28:08	5	Same as case 2.

II
Pc
(143527)
Ps
(3.26)
	1. 2	Asymmetric	168,200	3.21	0.66	0:10:12	3	C1:{1,2,6,9/1,2,6,9}
								C2:{3,4,5,7,8/3,4,5,7,8}

	2. 1	Asymmetric	168,200	3.21	0.66	0:13:42	3	Same as case 1
	3. $\frac{1}{2}$	Asymmetric	168,200	3.21	0.66	0:26:59	3	Same as case 1
	4. 2	Symmetric	168,200	3.21	0.66	0:17:11	3	Same as case 1
	5. 1	Symmetric	168,200	3.21	0.66	0:13:46	3	Same as case 1
	6. $\frac{1}{2}$	Symmetric	168,200	3.21	0.66	0:36:59	3	Same as case 1

Table 3: (Continued)

Data set	P_c/P_s	Model style	Cost of dealing with EE	SCs	GE	Executing time	# of. EE	{Machine/Parts}

III
Pc
(209660)
Ps
(18.52)
	1. 2	Asymmetric	67,758	2.38	0.53	22: 06:37	2	C1:{1,4,5,7,13/1,2,6,7, 8,17,18,19,20,23}
								C2:{2,3,10,11/3,4, 21,24}
								C3:{6,8,9,12,14/5,9,10, 11,12,13,14,15, 16,22}

	2. 1	Asymmetric	67,758	2.38	0.53	18: 45:34	2	Same as case 1
	3. $\frac{1}{2}$	Asymmetric	67,758	2.38	0.53	19: 59:21	2	Same as case 1
	4. 2	Symmetric	67,758	2.38	0.53	23: 35:14	2	Same as case 1
	5. 1	Symmetric	67,758	2.38	0.53	19: 27:39	2	Same as case 1
	6. $\frac{1}{2}$	Symmetric	67,758	2.38	0.53	21: 05:43	2	Same as case 1

considers fuzziness in the constraints. Clustering results among cases 1, 2, and 3 in Table 3 (FMLP) are the same as those among cases 3, 4, and 5 in Table 2 (goal programming).

Concluding Remarks

A goal programming formulation and corresponding FMLP models were proposed to solve the CF problems with two conflicting objectives – minimizing the total costs of dealing with the EEs, and maximizing the SCs of parts. A new part SC formula also was proposed to optimize GE by way of maximizing total SC. Symmetric and asymmetric models with the add-min operator were investigated through FMLP.

Three conclusions can be drawn from the numerical illustrations. First, the FMLP approach not only can obtain the same clustering results but also is more efficient computationally than the goal programming models are. Second, the effect of changing P_c/P_s ratio in FMLP is similar to that of changing the W_s/W_c ratio in goal programming models. Generally, if the data set was complicated, that is, if there was a great percentage of EEs, the larger the P_c/P_s or W_s/W_c value, the better the GE value and the worse the cost performance. If the EE percentage is relatively small, clustering results would be the same, however. Third, since the asymmetric model always could obtain the same clustering results as the symmetric model could in the lower complex data set or slightly different clustering results in the complex data set, the asymmetric model is worthy of consideration because of its computational efficiency.

Using FMLP to trade off conflicting objectives in CF problems is a better approach than goal programming method to designing workcells. However, like other mathematical programming, the executing efficiency of FMLP deteriorates when the problem size grows. Hence, developing an efficient heuristic algorithm based on the FMLP model to solve larger problems remain an interesting topic for future studies.

Appendix: Generic Form of the Add-min Operator

The add-min operator is an improved operator remedying the weakness of add and min operators. According to [35], add-min applies the min operator for fuzzy constraints and combines these with the fuzzy objective function by using the add operator. Thus, the aggregated membership function becomes

$$U_D = 0.5(U_G + \underset{s=1}{Min}\, U_S). \tag{A1}$$

The range of U_D is [0, 1]. The model after aggregation is linear, and the compensatory property is better than the max-min and add operators. For a standard FLP model,

$$M\tilde{I}N\ f(x) = CX \tag{A2}$$

subject to

$$AX \tilde{\leq} b. \tag{A3}$$

The equivalent transformed formulation after the add-min operator is applied becomes

$$MAX\ U_G + \lambda \tag{A4}$$

subject to

$$U_G = 1 - \frac{CX - Z^0}{P_0} \text{ and} \tag{A5}$$

$$1 - \frac{AX - b}{P_r} \geq \lambda . \qquad \text{(A6)}$$

If constraint (A5) is inserted into (A4), the fuzzy objective function becomes (A7) and the fuzzy constraint (A8).

$$MIN\ CX - \lambda P_0 \qquad \text{(A7)}$$

subject to

$$AX + \lambda P_r \leq b + P_r . \qquad \text{(A8)}$$

References

[1] Bit, A. K., Biswal, M. P., and Alam, S. S., "Fuzzy Programming Approach to Multicriteria Decision Making Transportation Problem," *Fuzzy Sets and Systems*, 50, 135-141 (1992).

[2] Burbidge, J. L., *The Introduction of Group Technology*, John Wiley and Sons, New York (1975).

[3] Chan, H. M., and Milner, D. A., "Direct Clustering Algorithm for Group Formation in Cellular Manufacturing," *Journal of Manufacturing System*, 1, 65-74 (1982).

[4] Chandrasekharan, M. P., and Rajagopalan, R., "An Ideal Seed Non-Hierarchical Clustering Algorithm for Cellular Manufacturing," *International Journal of Production Research*, 24, 2, 451-464 (1986).

[5] Chandrasekharan, M. P., and Rajagopalan, R., "Groupability: An Analysis of the Properties of Binary Data Matrices for Group Technology," *International Journal of Production Research*, 27, 6, 1035-1052 (1989).

[6] Chandrasekharan, M. P., and Rajagopalan, R., "ZODIAC-An Algorithm for Concurrent Formation of Part-Families and Machine-Cells, " *International Journal of Production Research*, 25, 6, 835-850 (1987).

[7] Chen, S. J., and Hwang, C. L., *Fuzzy Multiple Attribute Decision Making: Methods and Applications*, Heidelberg, Springer-Verlag (1991).

[8] Chu, C. H., "Clustering Analysis in Manufacturing Cellular Formation," *OMEGA*, 17, 289-295 (1989).

[9] Chu, C. H., "Recent Advances in Mathematical Programming for Cell Formation," in Kamran, A. K. and Liles, D. H. (eds.), *Planning , Design and Analysis of Cellular manufacturing System*, The Netherlands, Elsevier science B. V., 3-46.

[10] Deporter, E. L., and Ellis, K. P., "Optimization of Project Networks with Goal Programming and Fuzzy Linear Programming," *Computers Ind. Engng*, 19, 500- 504 (1990).

[11] Gongaware, T. A., and Ham, I., "Cluster Analysis Applications for Group Technology Manufacturing Systems," in *Proceedings of the 9th North American Manufacturing Research Conference (NAMRC)*, 503-508 (1981).

[12] Hannan, E. L., "Linear Programming with Multiple Fuzzy Goals," *Fuzzy Sets and Systems*, 6, 235-248 (1981).

[13] King, J. R., “Machine-Component Group Formation in Group Technology,” *OMEGA*, 8, 2, 193-199 (1980).

[14] Kumar, C. S., and Chandrasekharan, M. P., “Grouping Efficacy: A Quantitative Criterion for Goodness of Block Diagonal Forms of Binary Matrices in Group Technology,” *International Journal of Production Research*, 28, 233-243 (1990).

[15] Kumar, K. R., and Vannelli, A., "Strategic Subcontracting for Efficient Disaggregated Manufacturing," *International Journal of Production Research*, 25, 1715-1728 (1987).

[16] Lai, Y.-J., and Hwang, C.-L., *Fuzzy Multiple Objective Decision Making*, Berlin, Springer-Verlag (1994).

[17] Leberling, H., "On Finding Compromise Solutions in Multicriteria Problems Using the Fuzzy Min-Operator," *Fuzzy Sets and Systems*, 6, 105-118 (1981).

[18] Li, R. J., and Lee, E. S., "An Exponential Membership Function for Fuzzy Multiple Objective Linear Programming," *Computers Math. Applic.*, 22, 55-60 (1991).

[19] Landjula, M. K., “Multiple Objective Programming Problems with Possibilistic Coefficients,” *Fuzzy Sets and Systems*, 34, 15-31 (1990).

[20] Luhandjula, M. K., "Compensatory Operators in Fuzzy Linear Programming with Multiple Objectives," *Fuzzy Sets and Systems*, 8, 245-282 (1982).

[21] McAuley, J., "Machine Grouping for Efficient Production," *The Production Engineering*, 51, 53-57 (1972).

[22] Mosier, C. T., “An Experiment Investigating the Application of Clustering Procedures and Similarity Coefficients to the GT Machine Cell Formation Problem,” *International Journal of Production Research*, 27, 10, 1811-1835 (1989).

[23] Nagi, R., Harhalakis G., and Proth, J.M., "Multiple Routing and Capacity Considerations in Group Technology Application," *International Journal of Production Research*, 28, 2243-2257 (1992).

[24] Okada, S., Gen, M., and Ida, K., “A Method for Solving Multiobjective Linear Programming Problems with Trapezoidal Fuzzy Coefficients,” *Japanese Journal of Fuzzy Theory and Systems*, 5, 15-27 (1993).

[25] Pickens, J. B., and Hop, J. G., “Fuzzy Goal Programming in Forestry: An Application with Special Solution Problems,” *Fuzzy Sets and Systems*, 39, 239-246 (1991).

[26] Sankaran, S., “Multiple Objective Decision Making Approach to Cell Formation: A Goal Programming Model,” *Mathematical Computational Modeling*, 3, 71-82 (1990).

[27] Sasaki, M., Gen, M., and Ida, K., "A Method for Solving Reliability Optimization Problem by Fuzzy Multiobjective 0 - 1 Linear Programming," *Electronics and Communications in Japanese*, 3, 74, 106 - 116 (1991).

[28] Seiforddini, H., and Wolef, P. M., "Application of the Similarity Coefficient Method in Group Technology," *IIE Transaction*, 18, 271-277 (1986).

[29] Shafer S. M., and Rogers, D.F., “A Goal Programming Approach to the Cell Formation Problem,” *Journal of Operations Management*, 10, 28-43 (1991).

[30] Shafer S. M., Kern, G. M., and Wei, J. C., "A Mathematical Programming Approach for Dealing with Exceptional Elements in Cellular Manufacturing," *International Journal of Production Research*, 30, 1029-1036 (1992).

[31] Shafer S. M., and Rogers, D.F., "Similarity and Distance Measures for Cellular Manufacturing. Part I. A Survey," *International Journal of Production Research*, 31, 1133-1142 (1993).

[32] Singh, N., "Design of Cellular Manufacturing Systems: An Invited Review," *European Journal of Operational research*, 69, 284-291 (1993).

[33] Srinivasan, G. S., Narendarn, T. T., and Mahadevan, B., "An Assignment Model for the Part-Families Problem in Group Technology," *International Journal of Production Research*, 28, 1, 145-152 (1990).

[34] Stanfel, L. E., " Machine Clustering for Economic Production," *Engineering Costs and Production Economics*, 9, 73-81 (1985).

[35] Tsai,C. C., Chu, C. H., and Barta, T. A., "Optimization of Manufacturing Cell Formation with Fuzzy Linear Programming," *Working Paper*, Department of Industrial Manufacturing System Engineering, Iowa State University (1994).

[36] Wei, J. C., and Gaither, N., "A Capacity Constrained Multiobjective Cell Formation Method," *Journal of Manufacturing Systems*, 9, 222-232 (1990).

[37] Wemmerlöv, U., and Hyer, N. L., "Research Issues in Cellular Manufacturing," *International Journal of Production Research*, 25, 413-432 (1987).

[38] Wemmerlöv, U., and Hyer, N. L., "Procedures for the Part Family, Machine Group Identification Problem in Cellular Manufacturing," *Journal of Operations Management*, 6, 125-147 (1986).

[39] Werners, B. M., "Interactive Multiple Objective Programming Subject to Flexible Constraints," *European Journal of Operational Research*, 31, 342-349 (1987).

[40] Witte, J. D., "The Use of Similarity Coefficients in Production Flow Analysis," *International Journal of Production Research*, 18, 4, 503-514 (1992).

[41] Yang, T., and Ignizio, J. P., "Fuzzy Programming with Nonlinear Membership Functions: Piecewise Linear Approximation," *Fuzzy Sets and Systems*, 41, 39-53 (1991).

[42] Zimmermann, H. J., "Fuzzy Programming and Linear Programming with Several Objective Functions," *Fuzzy Sets and Systems*, 1, 45-55 (1978).

[43] Zimmermann, H. J., *Fuzzy Set Theory and Its Applications*, Boston, Kluwer Academic Publishers (1991).

III. OPTIMIZING THE PROBLEM OF DEALING WITH EXCEPTIONAL ELEMENTS IN CELL FORMATION: A HEURISTIC GENETIC ALGORITHM APPROACH

A paper prepared for submission to Decision Science

Chang-Chun Tsai, Chao-Hsien Chu, and Thomas Arnold Barta

ABSTRACT

Genetic algorithms (GA) have been recognized as an efficient search procedure for solving difficult combinatorial problems. The performance of GA, however, depends quality on suitably selected parameters. Improperly selected parameters can lead to inconclusive results and to increased computational time.

This paper demonstrates how GA can be used to solve a cell formation (CF) problem so as to minimize the total costs of dealing with exceptional elements (EEs). New heuristic crossover and mutation operators are proposed to enhance computational effectiveness, and a heuristic algorithm for computing total costs is developed to evaluate the fitness values of chromosomes. The proposed heuristic GA clearly outperforms both a mathematical programming (MP) model and a traditional GA, in terms of clustering results, computational time, and ease of use.

INTRODUCTION

Cellular manufacturing (CM), an application of the group technology (GT) philosophy, groups dissimilar machines in physical proximity to produce families of similar parts. This philosophy has affected the way that people think about and perform their jobs [42]. Cell formation, the first step and the key task of designing CM, concerns the process of grouping parts with similar design features or processing requirements into part families and grouping machines into machine cells. Many analytical methods have been proposed or developed for solving CF problems [9] [13] [40] [49]. Mathematical programming, one of the most popular methods, not only can guarantee an optimal solution but also can serve as a basis on which to develop heuristic methods. When problems increase in size, however, the solving efficiency of MP deteriorates. A state-of-the-art survey on the use of MP in CF can be found in [13].

To remedy the aforementioned weakness, efficient heuristic algorithms have been developed. The GA, which imitates natural selection and biological evolutionary process, is one of these search algorithms. Because the GA has a number of properties that other search procedures do not have [24], it can be an attractive algorithm with which to solve complicated combinatorial problems. These properties include (a) many feasible points considered and evaluated simultaneously in search space; (b) strings of characters representing the parameter set that are dealt with directly, not with the parameters themselves; and (c) probabilistic theory, not random search, used to direct the search. Thus, a GA search has the capability to process points in parallel, which decreases the probability of its being trapped in a local minimum, and a high-quality solution can be reached.

Although GA has been applied successfully to a variety of disciplines, we have been able to find only one paper [47] that has applied the GA to solve a CF problem. But there were two weaknesses in that GA study: (1) results did not compare with those of the mathematical programming approach because the mathematical model was not well developed for running in any mathematical software package, and (2) as with the traditional GA, improper parameters could cause poor computational performance. To overcome these weaknesses and to retain the advantages of the novel approach, heuristic crossover and heuristic mutation operators were developed based on the special characteristics of the CF problem.

A comprehensive CF problem that can simultaneously form manufacturing cells and minimize the total cost of dealing with EEs is studied. An EE is produced when either certain parts must be processed through more than one machine cell or certain machines are required by more than one part family. An EE increases the tangible and the intangible costs of developing manufacturing cells. The heuristic GA is based on the mathematical programming model proposed in [45]. Seventeen data sets from the open literature will be used, and their performance will be compared with that of the MP and that of the traditional GA.

OVERVIEW OF THE GENETIC ALGORITHM

The GA is one of the best-known structured random search methods. As stressed by Holland [22], one of its important aspects is that even in a large and complicated search space, given certain conditions on the problem domain, the GA will converge on solutions globally optimal or nearly so. The GA has been used in a wide variety of research fields including design, scheduling, configuration, financial portfolio management, adaptive control systems, and noisy data interpretation [18] [21]. The implementing procedure for GA can be seen in [14], and the key components covered in the procedure follow [18].

Chromosomal Representation

Defining variables is the major step in implementing MP. Similarly, defining and representing variables is a key step in implementing GA. Here, two decisions are involved: (1) The style and the physical interpretation of representation. Different problems have different ways of representation and interpretation, and these differences will affect chromosome size. For example, in the layout problem, the string represents the layout structure. In the CF problem, each bit may represent the cell number for related machines and parts. (2) The coding scheme for each bit position. Two schemes are used: (a) binary (0 and 1) and (b) integer (0 to 9). The first is the traditional coding method and can be found in problems of fuzzy control [16] [17] [20] [24] [25] [34], optimal knapsack packing [15], and neural networks [1] [31]. The second is a hybrid string representation up until now found primarily in problems of facility layout [5] [44], traveling salesman [23] [33], scheduling [2] [29], CF [47], and mixed discrete nonlinear optimization [30]. This latest scheme decreases the length of the chromosome, thus simplifying string processing and improving computational efficiency dramatically.

Generation of Initial Population

Initial population normally is generated randomly. For feasible solutions to be obtained, however, each chromosome must satisfy problem constraints. Two methods are used to handle these constraints: (1) The variable restriction method, which chooses only the populations satisfying the feasible region of constraints; and (2) the penalty function method, which allows populations to violate constraints but imposes a penalty when they do. These violated chromosomes therefore have a higher probability of rejection as a result of their penalized fitness. An unsuitably defined penalty function may produce an infeasible final solution or may make the optimal solution difficult to obtain.

Selection of Fitness Function

A fitness function is needed so that better solutions can be scrutinized. The objective function (to maximize or to minimize) of the MP model can be used directly for such a purpose.

Genetic Operators

Operators, which process the evolution of populations from generation to generation, are the key aspect of GA. Three basic operators – reproduction, crossover, and mutation – and their variations are discussed in most GA literature [18].

(1) Reproduction. The reproduction operator generates a mating pool of populations selected from the current existing chromosomes with respect to the probability distribution of fitness values. Several methods have been used for reproduction. Roulette wheel selection [17] [18] is the easiest and most popular method and normally achieves effective reproductions. With this approach, the better-fitted populations are assigned larger slots on the wheels and thus have greater probabilities of being selected for reproduction in the next generation than other populations do. The second approach is

tournament selection [24], whereby pairs of populations are compared with each other on a head-to-head basis to determine survival in the next generation. The larger evaluation value in each pair is selected for the maximum problem; the smaller value, for the minimum problem. Another alternative is called "steady-state without duplicates" [30]. Under this strategy, only a certain number of populations instead of all populations, are replaced each time.

(2) Crossover. After the better-fitted populations are produced, the crossover operator is used. Each pair of chromosomes is exchanged according to a special rule, and new strings are produced. Three kinds of crossover operators can be found: (a) The basic crossover operator includes single-point crossover and two-point crossover [47]. The former first generates randomly a number, $\mathcal{X}$, that indicates the position of the crossing point for each pair of populations; then the offspring combines the pre-$\mathcal{X}$ section of one parent with the post-$\mathcal{X}$ section of another. The latter generates randomly two numbers, $\mathcal{X}$ and $\mathcal{Y}$, that indicate the positions of the crossing point for each pair of populations. The offspring exchange between each other's $\mathcal{X}$-$\mathcal{Y}$ section. (b) Permutation crossover operators. Four methods have been proposed so far [5]. The first is partially matched crossover (PMX), which is based on the aforementioned two-point method. Because the results of two-point crossover sometimes are illegal, the improved step is to replace these repeated genes with related genes from the exchanging section (between the $\mathcal{X}$-$\mathcal{Y}$ section). The second method is Order Crossover (OX), which also generates randomly two cut points and copies the genes bound by the cut points to the offspring. The method

then uses a sliding procedure to fill the remaining places with numbers not occurring in the copied section. The third method is Cycle Crossover (CX), which does not need to select cut points. Each gene of the offspring comes from the corresponding position of either parent. An initial value is selected first from parent 1, and the related position's value in parent 2 is checked. Therefore, the position of this value is found in parent 1, and the related position's value in parent 2 is checked. This procedure stops when the initial value is met again. A cycle is formed by the positions of these elements, and the use of these operators can be found in the layout, traveling salesman, and sequencing problems. The last method is proposed in [35] to solve the sequencing problem. It was modified from the aforementioned one-point method. This method also replaces repeated genes with related genes from the exchanging section (the pre-$\boldsymbol{\mathcal{X}}$ section). (c) Heuristic operator. A heuristic crossover operator was developed in [23] to solve the traveling salesman problem. It uses a heuristic to combine good subpaths from both parents. Because it has a clear direction by which to find the better generations, it is more efficient than random crossover.

(3) Mutation. Mutation involves the change of a single gene value with a mutation probability when new chromosomes are formed. The new value is a randomly generated number other than itself.

Determination of System Parameters

Certain parameters must be determined in the initialization of every GA problem. Population size, for example, is required for defining the variable. Maximum number of generations and the probabilities of crossover and of mutation also are needed for the evolution process. These values are very sensitive to the performance of implementing GA. It is difficult to determine appropriate parameter values. Guidelines for selecting parameters can be found in [20]. But extensive experimentation with tedious trial and error is suggested for the selection process, because parameters depend greatly on problem characteristics.

MATHEMATICAL MODEL OF CELL FORMATION

Finding the best combination of machine cells and part families to reach a specific goal (objective) in a context of limited of resources (constraints) is the spirit of CF. Several popular objective functions can be found in [13]; most functions frequently used in CF modeling have been related to EE. For example, the top two are total costs of machine investment and total costs of intercell movement. Hence, for practical purposes, this paper applies GA to a more comprehensive CF model, which was adapted from [38] and used in [45]. The proposed model not only can group part families and machines cells simultaneously so as to minimize the total cost of dealing with EEs, but also considers the available capacity of machines; thus, the needed number of machines can be determined and minimized. For a detailed mathematical programming model, please see the Appendix and refer to [45].

THE PROPOSED GENETIC ALGORITHM HEURISTIC

One major weakness of the optimization approach is that computational efficiency deteriorates as problem size increases. In this section, the GA heuristic developed to resolve this dimensional problem is discussed. Its performance will be compared with that of the optimization model and that of the traditional GA approach.

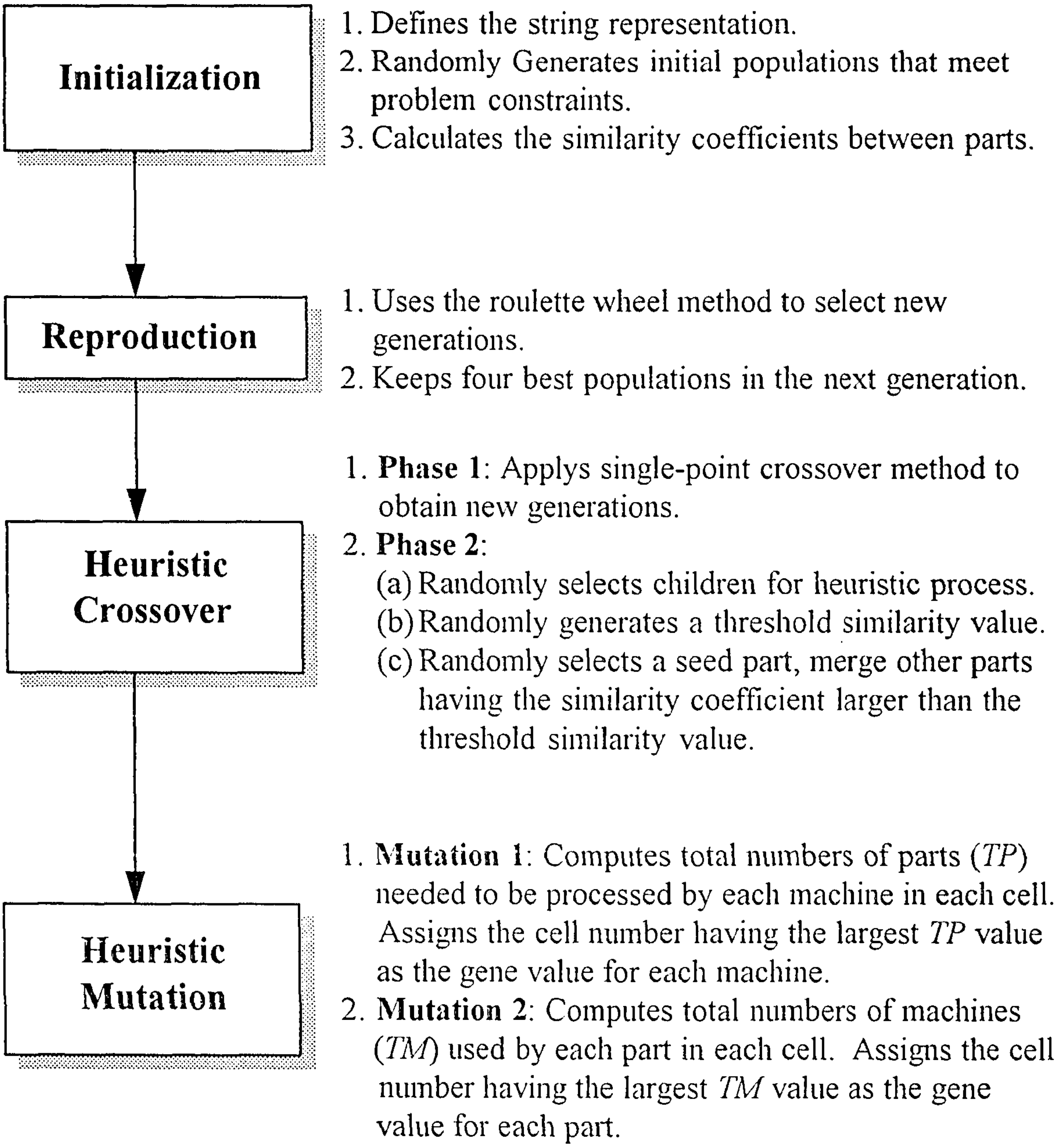
The traditional GA uses three operators: reproduction, crossover, and mutation. Random selection of the processing objects is used in the last two operators. Thus, results depend on crossover probability, mutation probability, and the seed for generating random numbers. When inappropriate values are assigned these parameters results may be inconclusive [37]. The proposed GA heuristic uses the traditional reproduction operator but modifies the crossover operator and develops a new heuristic mutation operator to decrease the dependence of parameters and to increase computational efficiency. Table 1 summarizes the differences between the proposed GA heuristic and the traditional GA. A complete implementing procedure for the proposed heuristic GA appears in Figure 1. Key components and technical details follow:

Chromosomal Representation

The chromosome consists of $(m + n)$ genes, in which the first m genes correspond to machines and the next n genes correspond to parts. For example, if seven machines and ten parts are to be grouped into three cells, then chromosome can be represented as (1 1 2 3 1 2 3 | 1 3 2 3 2 1 1 3 2 1), where all strings are integers. As such, this chromosome indicates that cell 1 consists of machines 1, 2, and 5 and of parts 1, 6, 7, and 10; cell 2 has machines 3 and 6, and parts 3, 5, and 9; remaining machines and parts belong to cell 3.

Table 1: Differences between the Traditional GA and the Proposed GA Heuristic

Key Components	The Traditional GA	The Proposed GA Heuristic
• Chromosomal Representation	• Each gene represents the cell number to which machines and parts belong.	• Same as the traditional GA.
• Generation of Initial Populations	• Variable restriction method. (*NM* needs to be given)	• Same as the traditional GA, except the *NM* is not required.
• Selection of Fitness Function	• Uses the objective function of MP.	• Same as the traditional GA.
• Reproduction Operator	• Roulette wheel selection method. • Keeps four best populations in the next generation.	• Same as the traditional GA.
• Crossover Operator	• Uses a single-point crossover operator. The crossover range includes all genes.	• Uses a single-point crossover operator. The crossover range is limited to part genes. • Develops a heuristic recombination method based on the SCs between parts, for part genes.
• Mutation Operator	• Mutates genes' values randomly, according to the probability of mutation.	• Develops a heuristic method to determine the gene value for each machine, based on crossover results and then reconsiders the gene value of each part.
• System Parameters	• Probability of crossover. • Probability of mutation. • Seed for random number generation. • Population size. • Number of generations.	• Seed. (1234 can solve all data sets) • Population size ≤ 100. • Number of generations ≤ 5.



1. Defines the string representation.
2. Randomly Generates initial populations that meet problem constraints.
3. Calculates the similarity coefficients between parts.

1. Uses the roulette wheel method to select new generations.
2. Keeps four best populations in the next generation.

1. **Phase 1**: Applys single-point crossover method to obtain new generations.
2. **Phase 2**:
   (a) Randomly selects children for heuristic process.
   (b) Randomly generates a threshold similarity value.
   (c) Randomly selects a seed part, merge other parts having the similarity coefficient larger than the threshold similarity value.

1. **Mutation 1**: Computes total numbers of parts (*TP*) needed to be processed by each machine in each cell. Assigns the cell number having the largest *TP* value as the gene value for each machine.
2. **Mutation 2**: Computes total numbers of machines (*TM*) used by each part in each cell. Assigns the cell number having the largest *TM* value as the gene value for each part.

Figure 1: Implementation Procedure of the Proposed GA Heuristic

Generation of Initial Population

This study uses a variable restriction method to generate initial population. Initialization population is generated randomly to satisfy the requirement of problem constraints. Two constraints are considered: (1) the total number of machines in each cell must be less than or equal to *NM*, and (2) the number of machines and of parts in each cell must be greater than one.

Selection of Fitness Function

This study uses the objective function, minimum total costs, of the mathematical programming model as the fitness function for evaluation. The following rules are deduced and applied to compute the optimal costs of dealing with EEs for each chromosome:

(1) Rule 1. If more than one exceptional part that must be processed by the same machine, *i*, are in the same cell, then the following procedure is followed:

- Compute the total number of machine *i* needed to process EEs in cell *k*.
- Sort the cost ratios (cost of moving a unit of part *j*/ processing time of machine *i* needed to process part *j*) of all EEs in ascending order.
- If the total number of machine *i* needed to process EEs (*TCM*) is greater than one, then duplicate machine *i* to cell *k* for the truncate value of *TCM*, and assign the remaining units to the EEs with the highest cost ratios.
- If more than one EE still is unresolved, then choose the minimum total costs of intercellular movement and subcontracting from all situations that might occur, and

compare them with the duplicating cost of machine *i*. Select the alternative with the minimum value.

(2) Rule 2. If more than one exceptional part that must be processed by the same machine, *i*, are in different cells, then compute the total costs of dealing with each EE and all its possible combinations recursively by using Rule 3; then select the optimal alternative with the minimum value.

(3) Rule 3. If only one exceptional part in certain cells still must be dealt with, then the following process is followed:

- If the available capacity of machine *i* is greater than the number of machines *i* needed to process part *j*, then the intercell units are equal to the latter, and the subcontracting units to zero. Otherwise, the intercell units are equal to the former, and the subcontracting units are equal to the subtraction value between them.
- Calculate the total cost of intercell movement and subcontract and compare them to the cost of duplicating machine *i*. The optimal policy is the smaller one.

Genetic Operators

The reproduction operator is the same as in the traditional GA approach, which used the roulette wheel selection method. But crossover and mutation operators have been revised to improve efficiency.

(1) Crossover operator. A two-phase processes was developed for the crossover. The first phase applies the single-point crossover operator to each pair of better-fit chromosomes. This process is similar to that in the traditional GA, except that the range of crossover

includes only part genes. Machine genes will be improved by means of mutation. To remedy the weakness of the traditional crossover operator, random crossover, the second phase applies the similarity coefficients (SCs) among parts to adjust the gene's value of parts. The SC presents the relation between pairs of parts. The closer the relation, the higher the SC value. In other words, if a pair of parts has a high SC value, they will have a high probability of being grouped together. The detailed heuristic process involved in the second phase is as follows:

- Children to be performed are selected randomly according to a probability of 0.5.
- A number between the range of [mean value ± variance] is generated randomly as the threshold similarity for comparison.
- Starting from cell 1, a part belonging to that cell as the seed is selected randomly. Then the SCs between the seed part and the other parts are compared with the threshold similarity. If the SCs are larger, the part is merged with the seed part in that cell; otherwise, they are put in the next cell. This process repeats until all cells are examined.

Regarding the approach to computing SCs, Shafer and Rogers [39] surveyed and analyzed most SCs proposed in the literature. Most SC formulae produce an unacceptable result – the fewer the number of cells, the higher the total SCs. That is, all parts are grouped together when total SCs are maximized. To eliminate this weakness, the current research follows the modification in [46], where the SCs between part *j* and part *j'* are defined as

$$SC_{jj'} = (2a - b - c)/(2a + b + c) \qquad \text{for all } j \neq j'$$

$$= 0 \qquad \text{otherwise,} \qquad (1)$$

where a is the number of machines that both part j and part j' tour, b is the number of machines that part j' tours, and c is the number of machines that part j tours.

(2) Heuristic mutation operator. The traditional mutation operator mutates gene value randomly, according to the probability of mutation; thus, it might not offer a positive direction to reach the global optimal point directly. The proposed heuristic mutation consists of two steps. The first step, called mutation 1, determines the gene value for each machine. The process works as follows:

- The total number of parts needing to be processed by every machine in each cell is computed, and the maximum number selected. If there is a tie, one of them is chosen randomly from the ties.
- Eventually the selected value is assigned to the related machine chromosome.

The second step, called mutation 2, updates the gene value for each part. The process is similar to mutation 1, except that the total number of parts needed to be processed by every machine in each cell is replaced with the total number of machines used by each part in each cell.

Determination of System Parameters

The proposed heuristic GA is fairly robust. The dependence on system parameters has decreased. For instance, good results always can be generated in fewer than three generations with the population size fixed at 10, and there is no need of using a mutation probability.

Also, the crossover probability simply can be generated randomly, ranging between 0.5 to 1.0, without impacting the results.

NUMERICAL ILLUSTRATION

To explain the fundamental logic of heuristic crossover and mutation, a simple problem from [48] is selected for illustration. The data set consists of 5 machines and 7 parts and can be grouped into two cells. Table 2 shows the machines/parts matrix, the unit cost values randomly generated based on the mean value and the standard deviation in [38], and the corresponding part similarity coefficients.

Crossover

Suppose that the two following chromosomes are selected from initial populations:

initial 1: 2 1 1 2 2 2 2 1 2 2 1 1 and

initial 2: 1 1 2 2 2 2 1 1 1 1 2 1.

(1) Phase 1: Single-point crossover operator. Assuming that the crossover point is selected randomly between positions 8 and 9, the results of crossover are

Parent 1: 2 1 1 2 2, 2 2 1| 2 2 1 1 Child 1: 2 1 1 2 2, 2 2 1| **1 1 2 1**
X =>
Parent 2: 1 1 2 2 2, 2 1 1| **1 1 2 1** Child 2: 1 1 2 2 2, 2 1 1| 2 2 1 1

(2) Phase 2: Heuristic recombination. Three steps are implemented for part genes:

- Select children for recombination. Suppose that only child 1 is selected; then Child 2 is unchanged.

Table 2: Numerical Data for Illustration

(a) The Machines/Parts Matrix

					PARTS			
		1	2	3	4	5	6	7
	1	0	1	0	1	0	0	1
M	2	0	0	1	0	1	0	0
/	3	1	1	0	1	0	0	1
C	4	1	0	1	0	0	1	0
	5	0	0	1	1	1	1	0

(b) Cost Information

					PARTS					
		1	2	3	4	5	6	7	A(*i*)	C(*i*)
	1	0.00	4.82	0.00	3.69	0.00	0.00	3.69	58352	2000
M	2	0.00	0.00	2.83	0.00	1.94	0.00	0.00	72451	2000
/	3	3.72	1.78	0.00	2.60	0.00	0.00	3.54	70153	2000
C	4	2.32	0.00	3.46	0.00	0.00	2.54	0.00	47670	2000
	5	0.00	0.00	2.81	4.50	3.73	3.56	0.00	64495	2000
	D(*j*)	37537	35051	28443	40300	34731	29405	37416		
	S(*j*)	1.71	2.00	1.81	2.37	3.85	2.06	3.09		
	I (*j*)	5.33	6.24	6.68	4.51	6.02	4.76	6.61		

(c) Similarity Coefficients Between Parts

					PARTS			
		1	2	3	4	5	6	7
	1	0.00	0.00	-0.20	-0.20	-1.00	0.00	0.00
P	2	0.00	0.00	-1.00	0.60	-1.00	-1.00	1.00
A	3	-0.20	-1.00	0.00	-0.33	0.60	0.60	-1.00
R	4	-0.20	0.60	-0.33	0.00	-0.20	-0.20	0.60
T	5	-1.00	-1.00	0.60	-0.20	0.00	0.00	-1.00
S	6	0.00	-1.00	0.60	-0.20	0.00	0.00	-1.00
	7	0.00	1.00	-1.00	0.60	-1.00	-1.00	0.00

Mean value of SCs : -0.1932
Variance of SCs: 0.3664

- Compute the possible range for generating the threshold similarity. Because the mean value of SCs in the example is -0.1932 and the variance 0.3664, the possible range is [-0.5596, 0.1732].
- Implement the heuristic crossover. Currently, in child 1, parts 3, 4, 5, and 7 are in cell 1. Hence, the list of possible seed candidates is { 3, 4, 5, 7 }. Assume that part 3 was selected randomly as the seed part. After comparing the SCs between part 3 and the other parts with threshold similarity (0.04 is generated randomly), only parts 5 and 6 are qualified. Thus, the new cell # 1 contains parts 3, 5, and 6; and the other parts belong to cell # 2. The result after crossover are

  Child 1: 2 1 1 2 2, 2 2 1 2 1 1 2 and

  Child 2: 1 1 2 2 2, 2 1 1 2 2 1 1 (Child 2 is not changed).

Mutation

Two steps are involved in the mutation heuristic. Mutation 1 is applied for the machine genes and then the part genes are adjusted by means of mutation 2. Tables 3 and 4 respectively illustrate the process of heuristic mutation for children 1 and 2. Final clustering results are summarized in Table 5, where child 1 is the best solution that can be achieved in data set 1.

Table 3: The Process of Heuristic Mutation for Child 1

Child 1: 2 1 1 2 2, 2 2 1 2 1 1 2

Mutation 1: Total number of parts needing to be processed by each machine in each cell are:

Machines Cell	M1	M2	M3	M4	M5
C1	0	2	0	2	3
C2	3	0	4	1	1
Gene value	2	1	2	1	1

After mutation 1: **2 1 2 1 1**, 2 2 1 2 1 1 2

Mutation 2: Total number of machines used by each part in each cell are:

Parts Cell	P1	P2	P3	P4	P5	P6	P7
C1	1	0	3	1	2	2	0
C2	1	2	0	2	0	0	2
Gene value	2*	2	1	2	1	1	2

After mutation 2: 2 1 2 1 1, **2 2 1 2 1 1 2**

* This value was selected randomly because of the tie.

Table 4: The Process of Heuristic Mutation for Child 2

Child 2: 1 1 2 2 2, 2 1 1 2 2 1 1

Mutation 1: Total number of parts needing to be processed by each machine in each cell are:

Machines Cell	M1	M2	M3	M4	M5
C1	2	1	2	2	2
C2	1	1	2	1	2
Gene value	1	1 *	2*	1	2*

After mutation 1 :**1 1 2 1 2** , 2 1 1 2 2 1 1

Mutation 2: Total number of machines used by each part in each cell are:

Parts Cell	P1	P2	P3	P4	P5	P6	P7
C1	1	1	2	1	1	1	1
C2	1	1	1	2	1	1	1
Gene value	1*	1*	1	2	1*	2*	1*

After mutation 2: 1 1 2 1 2, **1 1 1 2 1 2 1**

* These values were selected randomly because of the tie.

Table 5: The Output of the Proposed GA for Children 1 and 2

Child 1: 2 1 2 1 1, **2 2 1 2 1 1 2**

		Parts						
		3	5	6	1	2	4	7
M/C	2	1	1	0	0	0	0	0
	4	1	0	1	1	0	0	0
	5	1	1	1	0	0	1	0
	1	0	0	0	0	1	1	1
	3	0	0	0	1	1	1	1

Group efficacy = 0.700
Number of EEs = 2
Total costs of dealing with EEs = $ 151,848

Child 2: 1 1 2 1 2, **1 1 1 2 1 2 1**

		Parts						
		1	2	3	5	7	4	6
M/C	1	0	1	0	0	1	1	0
	2	0	0	1	1	0	0	0
	4	1	0	1	0	0	0	1
	3	1	1	0	0	1	1	0
	5	0	0	1	1	0	1	1

Group efficacy = 0.346
Number of EEs = 7
Total costs of dealing with EEs = $ 413,576.25

COMPUTATIONAL EXPERIENCE

Data Sets for Comparison

To evaluate the relative performance among the proposed heuristic GA, the MP model, and the traditional GA approach, 17 data sets collected from the open literature are used. These data sets have been explored in a number of studies [10] [11] [12] [43], and their related characteristics are listed in Table 6. The additional numerical values for required parameters, such as the processing time of each part, the cost involved, and the part demand are generated randomly by a computer program, according to the mean value and the variance of data set 4 in [38].

Computational Results

To solve the MP models, this study used the LINDO (linear interactive and discrete optimizer) package running on an IBM compatible 486 DX2 66 MHZ computer. The GAs are coded in ANSI C and run on two different computers – the DECstation 5000 model 240 workstation (for comparisons with the traditional GA) and the IBM compatible 486 DX2 66 MHZ computer (for comparisons with MP). Clustering performances were measured in terms of EEs [10], CPU times, and grouping efficacies (GE) [27]. The desired cell number and the *NM* values were quoted from related references. Table 6 also summarizes the computational results of MP and the proposed heuristic GA. To examine that the heuristic crossover operator or the heuristic mutation operator enhances the computational effectiveness, Table 7 compares computational results of the proposed heuristic GA with those of the traditional GA

Table 6: Summary of Computational Results
(Mathematical Programming vs. the Proposed GA Heuristic)

NO.	Size + (m x n)	Matrix density	# of. cells	Ref.	Mathematical Programming				The Proposed GA Heuristic				
					NM	No. of EEs	Total costs ($)	PC CPU time (sec.)	Max. popu. (generations)	NM	No. of EEs	Total costs ($)	PC CPU time (sec.)
1	5 x 7	0.457	2	[48]	3	2	151,864	5.12	10(1)	3	2	151,848	0.22
2	8 x 20	0.381	3	[6]	4	9	600,553	5602	20 (1)	4	9	600,719	0.275
3	9 x 9	0.395	3	[19]	4	6	289,159	607.1	20 (1)	4	6	289,148	0.495
4	9 x 10	0.322	3	[38]	**4**	**9**	**441,233**	**4979**	60 (1)	**5**	**8**	**434,316**	**0.64**
5	10 x 12	0.30	3	[36]	5	1	65,478	177.2	10 (2)	5	1	65,478	0.22
6	12 x 10	0.317	3	[32]	5	6	237,477	**1024**	10 (2)	**6**	**5**	**220,948**	**0.659**
7	12 x 19	0.33	3	[50]	6	**19**	891,827	**21723**	30 (3)	**7**	**17**	**875,470**	**2.143**
8	14 x 24	0.182	4	[26]	4	2	209,660	6082	20 (4)	4	2	209,660	1.923
9	15 x 10	0.307	3	[4]	5	0	0	42.6	10 (2)	5	0	0	0.495
10	16 x 30	0.242	4	[43]	6	19	–	–	30 (4)	6	19	1,076,797	4.725
11	16 x 43	0.183	4	[3]	7	21	–	–	30 (5)	7	21	984,119	8.132
12	20 x 35	0.194	4	[3]	6	2	100,550	57882	10 (2)	6	2	100,550	1.099
13	24 x 40	0.127	7	[7]	6	0	0	6100	10 (2)	6	0	0	2.253
14	24 x 40	0.135	7	[7]	6	10	–	–	30 (2)	6	10	570,654	5.165
15	30 x 41	0.104	5	[28]	9	10	–	–	60 (5)	10	10	459,014	37.75
16	30 x 50	0.102	4	[41]	13	0	0	3289	10 (2)	13	0	0	3.74
17	40x100	0.105	10	[8]	8	36	–	–	#	#	#	#	#

+ m: the total numbers of machine types; n: the total numbers of part types.
– The computer running time is larger than one day (86,400 seconds).
# Cannot be run in a PC due to the memory overflow.

Table 7: Summary of Computational Results
(TheTraditional GA vs. the Proposed GA Heuristic without Using SCs or the Proposed GA Heuristic)

Data Set	Traditional GA *				The Proposed GA Heuristic ** (Only Using Heuristic Mutation)				The Proposed GA Heuristic **			
	Max. popu. (generations)	Cost (# of EE)	CPU time in W.S. (seconds)	Match MP	Max. popu. (generations)	Cost (# of EE)	CPU time in W.S. (seconds)	NM	Max. popu. (generations)	Cost (# of EE)	CPU time in W.S. (seconds)	NM
1	20 (5)	$151,848 (2)	0.14	Yes	10 (1)	$151,848 (2)	0.031	3	10 (1)	$151,848 (2)	0.051	3
2	800 (48)	$600,719 (9)	217.8	Yes	20(1)	$600,719 (9)	0.235	4	20(1)	$600,719 (9)	0.285	4
3	200 (15)	$289,150 (6)	8.019	Yes	30(1)	$289,148 (6)	0.227	4	30(2)	$289,148 (6)	0.340	4
4	800 (100)	$447,877 (10)	306.9	No	100(12)	$441,233 (10)	2.945	4	50 (5)	$434,316 (8)	0.977	5
5	500 (19)	$65,478 (1)	36.54	Yes	50(2)	$65,478 (1)	0.543	5	10 (2)	$65,478 (1)	0.082	5
6	500 (5)	$237,447 (6)	12.2	Yes	40(5)	$237,447 (6)	0.687	5	10 (2)	$220,948 (5)	0.121	6
7	800 (100)	$1,124,925 (22)	519.8	No	500(3)	$891,827 (19)	14.066	6	30 (3)	$875,470 (17)	0.966	7
8	800 (100)	$494,869 (7)	671.5	No	50(4)	$209,660 (2)	2.301	4	20 (1)	$209,660 (2)	0.418	4
9	50 (1)	$0 (0)	0.664	Yes	10(5)	$0 (0)	0.34	5	10 (1)	$0 (0)	0.098	5

Table 7: (Continued)

Data	Traditional GA *				The Proposed GA Heuristic ** (Only Using Heuristic Mutation)				The Proposed GA Heuristic **			
Set	Max. popu. (generations)	Cost (# of EE)	CPU time in W.S. (seconds)	Match MP	Max. popu. (generations)	Cost (# of EE)	CPU time in W.S. (seconds)	NM	Max. popu. (generations)	Cost (# of EE)	CPU time in W.S. (seconds)	NM
10	800(100)	$1,774,097 (38)	994.8	No	100(9)	$1,076,797 (19)	8.581	6	20 (4)	$1,076,797 (19)	0.678	6
11	800(100)	$2,381,996 (42)	1293.7	No	100(5)	$984,119 (21)	7.742	7	50 (3)	$984,119 (21)	3.875	7
12	800(100)	$1,698,190 (39)	1321.6	No	50(1)	$100,550 (2)	2.172	6	10 (1)	$100,550 (2)	0.418	6
13	800(100)	$2,791,835 (49)	1979.1	No	50(6)	$0 (0)	8.222	6	10 (2)	$0 (0)	1.469	6
14	800(100)	$2,395,065 (46)	2098.2	No	50(5)	$570,654 (10)	8.054	6	20 (4)	$570,654 (10)	4.828	6
15	800(100)	$2,062,305 (32)	2022.7	No	500(12)	**$464,680** **(10)**	**228.58**	9	50 (10)	**$459014** **(10)**	**33.15**	**10**
16	800(100)	$3,531,819 (35)	1303.8	No	50(14)	$0 (0)	18.378	13	30 (3)	$0 (0)	2.715	13
17	800(100)	$16,502,678 (273)	17751.	No	100(2)	$2,347,667 (36)	93.1	6	50 (3)	$2,347,667 (36)	45.3	6

* This result was the best obtained from running algorithm ten times with different seed values.

** All results matched the optimal solution of mathematical programming.

and those of the proposed heuristic GA not using the heuristic crossover (We called this a Minor Modification of the proposed GA Heuristic). For supposes of comparison, maximum population size and maximum generations for the traditional GA are set at 800 and 200, respectively. Certain parameters such as seed, crossover probability, and mutation probability are adjusted to produce a better-fitting solution. Several observations can be made regarding these computational results:

First, when the performances of computational efficiency and clustering results are compared, the proposed heuristic GA performs best, the minor modification of the heuristic GA (MHGA) second best and the traditional GA and the MP approaches third and fourth best. This ranking can be justified as follows:

(1) The proposed heuristic GA not only can obtain the same clustering solutions as most MP approaches can, but its CPU computational times also are much shorter, especially if the data set has more than 10 EEs. Moreover, once problem size increases, it is impracticable to use the MP to solve the problem; the proposed heuristic GA takes less than four seconds in a 486/66 PC to solve small and medium size data sets and less than one minute in a workstation to solve a large problem (data set 17).

(2) As for the other three approaches, the performance of the MHGA is much better than that of the MP or of the traditional GA. Like the proposed heuristic GA, the MHGA not only can obtain the exact optimal costs, as can the MP, but also has a much shorter CPU running time. The larger the problem, the better MHGA's efficiency.

(3) Although the traditional GA can reach the optimal point easily in small data sets, it fails to find the solution under the default parameters when data sets need to be grouped into more than three cells. An optimal solution might be obtained, however, by adjusting system parameters. For instance, in data set 8, the values of population sizes, numbers of generations, and CPU running time to obtain the optimal solution are 3,000, 23, and 1,072 seconds, respectively. General computational performances still are better than those of the MP under suitable parameters.

The first two results just described demonstrate that the contribution of the proposed heuristic – whether crossover or mutation – is positively significant. Applying these two heuristic operators simultaneously makes the GA useful in real world applications.

Second, the *NM* value is difficult to specify in practice. If it is too small or too large, the clustering results will be unreasonable. The proposed heuristic has the advantage of obtaining the better solution under an unrestricted *NM* value. As can be seen from Table 6, in data sets 4, 6, 7, and 15, without specifying NM value, the proposed heuristic automatically can obtain a lower total cost value (the objective function of the mathematical model) and a smaller EE number than the MP approach can. The main reason for this advantage is that the results of the latter approach are somewhat limited by *NM* value. Meanwhile, the proposed heuristic not only can provide a feasible answer, it also lists two of the best alternative solutions. This provides flexibility for users in the selection of final results to fit their own environment.

Third, the computational time of the MP approach increased significantly when the data set had more EEs (was more complex); the CPU time for the proposed GA, however, increased only slightly. For example, data set 14 is modified from data set 13 (add ten more EEs). The PC CPU time for running the MP model increased from 6,100 seconds to more than one day, but both data sets were solved in 2.253 seconds and in 5.165 seconds by using the proposed GA. This demonstrates that the proposed GA heuristic is suitable for large scale applications.

Fourth, the proposed GA is less sensitive to the parameters used than the traditional GA procedure is. For instance, most data sets can be solved under the following given parameter values: (1) population sizes less than or equal to 40 (except data sets 4 and 17); (2) generations less than or equal to 3; (3) seed for random number generation fixed at 1234; (4) the probability of crossover randomly generated; and (5) no probability of mutation needed. As can be seen from Table 7, traditional GA suffers from the determination of these parameters. Even after much fine tuning, most data sets still cannot obtain the same results as the proposed GA heuristic can. In this regards, the proposed GA is a more robust and friendly algorithm for decision makers because they do not need to decide the values of parameters.

Finally, although cost results of the proposed GA and of the MP might be somewhat different because of the rounding error in the MP model, the clustering results and the alternative method for dealing with EEs are the same (e.g. machine duplication, intercell movement, and subcontracting), a fact demonstrating that the proposed heuristic algorithm for computing the optimal cost of dealing with EE does work and can evaluate the exact fitness value of every chromosome.

CONCLUDING REMARKS

An improved GA was proposed to solve a CF problem. The proposed heuristic reflected a major enhancement in three directions: (1) considering the similarity coefficients between parts in the heuristic crossover to reassign the gene's value for each part; (2) creating a heuristic mutation operator to mutate the genes' values of machines and parts; and (3) developing a rule-based heuristic algorithm to compute the fitness value for each chromosome. From the computational analyses, two conclusions are drawn.

First, the proposed genetic algorithm heuristic outperforms both the mathematical programming and the traditional GA in terms of computational efficiency and of clustering results. In particular, even if problem sizes become large, the proposed heuristic still can complete the search in less time than the other two approaches can.

Second, the proposed heuristic has a number of user-friendly properties: (1) It does not require specification of system parameters, such as the maximum number of machines allowed in each cell and the crossover and mutation probabilities. (2) It displays all feasible solutions in the report. In this case, a solution might not be obtained from the MP approach. (3) It can obtain the optimal solution for any possible cell number in one execution. Thus, it also relaxes the need for deciding the number of cells, a very difficult task in CF. All these benefits promote use in real-world applications.

Considering problem characteristics in the GA to optimize the total cost of dealing with EEs seems a promising approach. Based on the results of this research, it should be possible to modify the heuristic fitness algorithm to solve any other CF problems or to modify related heuristic operators to solve other complicated combinatory problems.

APPENDIX: THE FORMULATION OF MATHEMATICAL PROGRAMMING MODEL

Notation Used

The following notation was used to model the CF problem:

Index Set

i	machine index; $i = 1,..., m$
j	part index; $j = 1,..., n$
k	cell index; $k = 1, ..., c$

Parameters

A_i	periodic cost of acquiring machine type i
C_i	periodic capacity of machine type i
D_j	periodic forecast demand for part j
I_j	incremental cost for moving a unit of part j within two cells
NM	maximum number of machine types allowed in each cell
P_{ij}	processing time of machine type i needed to produce part j
S_j	incremental cost of subcontracting a unit of part j for an operation
SP	set of pairs (i,j) such that $a_{ij} = 1$

UC_{ij} utilization capacity of machine type i for parts j. Value can be calculated by means of the equation $P_{ij} \times D_j / C_i$

Decision Variables

IC_k =1, if cell k is formed; 0, otherwise

M_{ijk} number of machines i dedicated to cell k for producing part j

O_{ijk} units of part j to be subcontracted as a result of machines type i not being available within cell k

Q_i number of machines type i needed to process corresponding parts in machine cell

R_{ik} number of machines type i to be dedicated in cell k

U_{ijk} = 1, if X_{ik} = 1, and Y_{jk} = 0; 0 otherwise

V_{ijk} = 1, if Y_{jk} = 1, and X_{ik} = 0; 0 otherwise

X_{ik} = 1, if machine i is assigned to cell k; 0, otherwise

Y_{jk} = 1, if part j is assigned to cell k; 0, otherwise

Z_{ijk} number of intercellular transfers required by part j as a result of machine type i not being available within cell k

Mathematical Programming Model

The formulation is as follows:

$$Min \sum_k \sum_i A_i R_{ik} + \sum_k \sum_{(i,j) \in sp} I_j Z_{ijk} + \sum_k \sum_{(i,j) \in sp} S_j O_{ijk} \qquad (A1)$$

Subject to:

$$\sum_{k=1}^{c} X_{ik} = 1, \ \forall i \tag{A2}$$

$$\sum_{k=1}^{c} Y_{jk} = 1, \forall j \tag{A3}$$

$$\sum_{i=1}^{m} X_{ik} \leq NM, \forall k \tag{A4}$$

$$X_{ik} - Y_{jk} + \frac{1}{D_j} Z_{ijk} + \frac{1}{D_j} O_{ijk} {}_{i} + \frac{1}{UC_{ij}} M_{ijk} - U_{ijk} = 0, \ \forall (i,j) \in sp, \forall k \tag{A5}$$

$$\sum_{(i,j) \in sp} M_{ijk} \leq R_{ik}, \ \forall i, \ \forall k \tag{A6}$$

$$Q_i \leq \sum_{(i,j) \in sp} UC_{ij} (1 - \sum_{k} V_{ijk}) + 1, \ \forall i \tag{A7}$$

$$\sum_{k} \sum_{(i,j) \in sp} \frac{P_{ij}}{C_i} Z_{ijk} \leq Q_i - \sum_{(i,j) \in sp} UC_{ij} (1 - \sum_{k} V_{ijk}), \ \forall i \tag{A8}$$

$$X_{ik}, Y_{jk}, U_{ijk}, V_{ijk} = 0 \ or \, 1; \quad R_{ik}, Q_i \ = \text{general integer} \tag{A9}$$

REFERENCES

[1] Austin, S. Genetic Solutions to XOR Problems, *AI EXPERT*, 1990, December, 53-57.

[2] Biegel, J. E., & Davern, J. J. Genetic Algorithms and Job Shop Scheduling, *Computers Industrial Engineering*, 1990, 19, 81-91.

[3] Burbidge, J. L. *The Introduction of Group Technology*, John Wiley and Sons, New York, 1975.

[4] Chan, H. M., & Milner, D. A. Direct Clustering Algorithm for Group Formation in Cellular Manufacturing, *Journal of Manufacturing System*, 1982, 1, 65-74.

[5] Chan, K. C., & Tansri, H. A Study of Genetic Crossover Operations on the Facilities Layout Problem, *Computers Industrial Engineering*, 1994, 26(3), 537-550.

[6] Chandrasekharan, M. P., & Rajagopalan, R. An Ideal Seed Non-Hierarchical Clustering Algorithm for Cellular manufacturing, *International Journal of Production Research*, 1986, 24(2), 451-464.

[7] Chandrasekharan, M. P., & Rajagopalan, R. Groupability: An Analysis of the Properties of Binary Data Matrices for Group Technology, *International Journal of Production Research*, 1989, 27(6), 1035-1052.

[8] Chandrasekharan, M. P., & Rajagopalan, R. ZODIAC-An Algorithm for Concurrent Formation of Part-Families and Machine-Cells, *International Journal of Production Research*, 1987, 25(6), 835-850.

[9] Chu, C. H. Clustering Analysis in Manufacturing Cellular Formation, *OMEGA*, 1989, 17, 289-295.

[10] Chu, C. H., & Tsai, M. A Comparison of Three Array-Based Clustering Techniques for Manufacturing Cellular Formation, *International Journal of Production Research*, 1990, 28, 1417-1433.

[11] Chu, C. H., & Hayya, J. C. Fuzzy Clustering Approach to Manufacturing Cell Formation, *International Journal of Production Research*, 1991, 29, 1475-1478.

[12] Chu, C. H. Manufacturing Cell Formation by Competitive Learning, *International Journal of Production Research*, 1993, 31, 829-843.

[13] Chu, C.H. Recent Advances in Mathematical Programming for Cell Formation, in *Planning, Design and Analysis of Cellular manufacturing System* (eds. Kamrani, A.K. Parsaei, H.R., and Liles, D.H.), The Netherlands, Elsevier Science B. V., 1995, 3-46.

[14] Davis, L., & Steenstrup, M. Genetic Algorithms and Simulated Annealing: An Overview, in *Genetic Algorithms and Simulated Annealing* (ed. L. Davis), San Mateo, Morgan Kaufmann Publishers, Inc., 1987, 1-11.

[15] Denning, P. J. Genetic Algorithm, *American Scientist*, 1992, 80, 12-14.

[16] Epsy, T., Vimbrack, E., & Aldridge, J. Application of Genetic Algorithms to Tuning Fuzzy Control Systems, *NASA International workshop on Neural Networks and Fuzzy Logic*, 1992, 237-248.

[17] Freeman, L. M., Krishnakumar, K., Karr, C., & Meredith, D. Tuning of Fuzzy Logic Controllers using Genetic Algorithms-Aerospace Applications, *Presented at the AAAIC'90 Conference*, Dayton, Ohio, Oct. 1990.

[18] Goldberg, D. E. *Genetic Algorithms in Search, Optimization, And Machine Learning*, Addison-Wesley: Reading, MA, 1989.

[19] Gongaware, T. A., & Ham, I. Cluster Analysis Applications for Group Technology Manufacturing Systems, *Proceedings of the 9th North American Manufacturing Research Conference (NAMRC)*, 1981, 503-508.

[20] Grefenstette, J. J. Optimization of Control Parameters for Genetic Algorithm, *IEEE Transactions on Systems, Man, and Cybernetics*, 1986, 16(1), 122-128.

[21] Hedberg, S. Emerging Genetic Algorithms, *AI EXPERT*, 1994, September, 25-37.

[22] Holland, J. H. *Adaptation in Natural and Artificial System*, Ann Arbor: University of Michigan Press, 1975.

[23] Jog, P., Suh, J. Y., & Gucht, D. V. The Effects of Population Size, Heuristic Crossover and Local Improvement on a Genetic Algorithm for the Traveling Salesman Problem, *Proc. of the 3rd Int. Conf. on Genetic Algorithms*, 1989, 110 - 115.

[24] Karr, C. L., & Gentry, E. J. Fuzzy Control of PH Using Genetic Algorithms, *IEEE Transactions on Fuzzy Systems*, 1993, 1(1), 46-53.

[25] Karr, C. L. Design of an Adaptive Fuzzy Logic Controller using a Genetic Algorithm, *The Fourth International Conference on Genetic Algorithms*, 1991, 450-457.

[26] King, J. R. Machine-Component Group Formation in Group Technology, *OMEGA*, 1980, 8(2), 193-199.

[27] Kumar, K. R., & Chandrasekharan, K. P. Grouping Efficacy: A Quantitative Criterion for goodness of Block Diagram Forms of Binary Matrices in Group Technology, *International Journal of Production Research*, 1990, 28, 233-243.

[28] Kumar, K. R., & Vannelli, A. Strategic Subcontracting for Efficient Disaggregated Manufacturing, *International Journal of Production Research*, 1987, 25, 1715-1728.

[29] Lawton, G. Genetic Algorithms for Schedule Optimization, *AI EXPERT*, 1992, May, 23-27.

[30] Lin, S. - S. Mechanical Tolerancing for Integrated Design and Manufacturing, Unpublished Ph. D. *Dissertation*, Dep. of Mechanical Engineering, The Florida State University, 1994.

[31] Maniezzo, V. Genetic Evolution of the Topology and Weight Distribution of Neural Networks, *IEEE Transactions on Neural Networks*, 1994, 5(1), 39- 53.

[32] McAuley, J. Machine Grouping for Efficient Production, *The Production Engineering*, 1972, 51, 53-57.

[33] Oliver, I. M., Smith, D. J., & Holland, J. R. C. A Study of Permutation Crossover Operator on the Traveling Salesman Problem, *Proceedings of the Second International Conference on Genetic Algorithm*, 1987, 224-230.

[34] Pham, D. T., & Karaboga, D. Optimum Design of Fuzzy Logic Controller Using Genetic Algorithms, *Journal of Systems Engineering*, 1991, 1, 114-118.

[35] Reeves, Colin R. A Genetic Algorithm for flowshop Sequencing, *Working Paper*, Department of Statistics and Operations Research, Coventry University, 1994.

[36] Seiforddini, H., & Wolef, P. M. Application of the Similarity Coefficient Method in Group Technology, *IIE Transactions*, 1986, 18, 271-277.

[37] Sen, M. K., Gupta, A. D., Stoffa, P. L., Lake, L. W., & Pope, G. A. Stochastic Reservoir Modeling Using Simulated Annealing and Genetic Algorithm, *Proceeding of the 1992 SPE Annual Technical Conference and Exhibition*, 1992, 939-950.

[38] Shafer, S. M., kern, G. M., & Wei, J. C. A Mathematical Programming Approach for Dealing With Exceptional Elements in Cellular Manufacturing, *International Journal of Production Research*, 1992, 30, 1029-1036.

[39] Shafer, S. M., & Rogers, D. F. Similarity and Distance Measures for Cellular Manufacturing. Part I. A Survey, *International Journal of Production Research*, 1993, 31, 1133-1142.

[40] Singh, N. Design of Cellular Manufacturing Systems: An Invited Review, *European Journal of Operational Research*, 1993, 69, 284-291.

[41] Stanfel, L. E. Machine Clustering for Economic Production, *Engineering Costs and Production Economics*, 1985, 9, 73-81.

[42] Steudel, H. J. The Role and Design of Workcells for world-Class Manufacturing, *The Journal of Applied Manufacturing Systems*, 1991, Winter, 47-55.

[43] Srinivasan, G. S., Narendarn, T. T., & Mahadevan, B. An Assignment Model for the Part-Families Problem in Group Technology, *International Journal of Production Research*, 1990, 28(1), 145-152.

[44] Tam, K. Y. Genetic Algorithms, Function Optimization, and Facility Layout Design, *European Journal of Operational Research*, 1992, 63, 322-346.

[45] Tsai, C.- C., Chu, C. H., & Barta, T. A. Fuzzy Linear Programming Approach to Manufacturing Cell Formation, *Proceedings of Third IEEE International Conference of Fuzzy Systems*, 1994, 2, 1406-1409.

[46] Tsai, C.- C., & Chu, C. H. A Fuzzy Multiobjective Linear Programming Model for Cell Formation, *Working Paper*, Department of Industrial and Manufacturing Systems Engineering, Iowa State University, 1994.

[47] Venugopal, V., & Narendran T. T. A Genetic Algorithm Approach to the Machine Component Grouping Problem with Multiple Objectives, *Computer and Industrial Engineering*, 1992, 22, 4, 469-480.

[48] Waghodekar, P. H., & Sahu, S. Machine-Component Cell Formation in Group Technology: MACE, *International Journal of Production Research*, 1984, 22(6), 937-948.

[49] Wemmerlöv, U., & Hyer, N. L. Procedures for the Part Family, Machine Group Identification Problem in Cellular Manufacturing, *Journal of Operations Management*, 1986, 6, 125-147.

[50] Witte, J. D. The Use of Similarity Coefficients in Production Flow Analysis, *International Journal of Production Research*, 1992, 18(4), 503-514.

IV. OPTIMIZING FUZZY CELL FORMATION PROBLEMS BY A HEURISTIC GENETIC ALGORITHM

A paper prepared for submission to Decision Science

Chang-Chun Tsai, Chao-Hsien Chu, and Thomas Arnold Barta

ABSTRACT

Fuzzy linear programming (FLP) and fuzzy multiobjective linear programming (FMLP) have been shown in [22] and [23] not only to provide a better and more flexible way of presenting the problem domain, but also to improve overall performance. In larger problems, however, fuzzy mathematical programming (FMP) cannot meet the demands of real-world applications. A heuristic genetic algorithm (HGA) as proposed in [24] is used to correct this weakness.

The current paper is an extension of the HGA that is based on new heuristic crossover and mutation operators to remedy the weakness of the traditional GA, namely that inappropriately selected parameters lead to poor clustering results and require added computational time. The HGA is based on different FMP models, and its performance is compared with FMP performance in terms of clustering results and computational efficiency. The HGA clearly outperforms the FMP model in terms of clustering results, computational time, and user friendliness.

INTRODUCTION

Mathematical programming (MP) is a useful approach to modeling and optimizing a problem because it not only can obtain the optimal solution but also can be the basis on which heuristic approaches are developed. But most required elements in MP, e.g., goals, constraints, and coefficients, cannot be decided precisely with ease. For this reason, fuzzy set theory has been applied widely in MP during the last 25 years, and several problems have been solved very well. For instance, transportation [4], location planning [10], air pollution [19], animal husbandry systems [25], and cell formation [22] all are single-objective problems. Reliability [16], project networks [11], transportation [1], forests [15], and cell formation [23] are multiple objective problems.

Applying FMP to the cell formation (CF) problem is new. Cell formation is the first step and the key task in implementing cellular manufacturing (CM), which is not only one of the major applications of the group technology (GT) philosophy but also a valuable concept in world-class manufacturing and furthers the objectives of just-in-time (JIT) manufacturing and total-quality-management [21]. The main aim of CF is to form suitable workcells by grouping similar parts and their related machines together. The CF process can be illustrated as (1) using part attributes (GT codes always used) or part routings (sets of parts with production similarities) to form part families; (2) using part routings or production data to form machine groups; and (3) combining part families with machine groups to form cells. A review of CF problems can be found in [7] [8] [18] [26]. A major factor causing the clustering bottleneck in the CF problem is the *exceptional element* (EE), a part that needs to be processed by a machine not in the same cell. Most prior research, however, removes EEs manually or deals with them after forming the initial cells. No study has attempted to deal with EEs and to form manufacturing cells simultaneously except two of our previous working papers [22] [23]. Furthermore, only these two papers model the CF problem in a fuzzy environment and verify that FMP performances excel over the traditional MP performance in terms of clustering results and executing efficiency. When problem size increases, as with the data set adapted from [13] and used in [22] [23], FMP performing efficiency deteriorates. Not-withstanding, HGA, a powerful heuristic search approach proposed in [22], can remedy the problem.

The HGA retains the spirit of the traditional GA, which imitates natural selection and biological evolutionary processes [3], and by applying the characteristics of CF problem, the HGA replaces random procedures in crossover and mutation operators with a heuristic property. Hence, the HGA improves executing efficiency dramatically and decreases dependence on system parameters. The HGA was relatively efficient computationally and met the critical demands of real-world applications in [24]. The main motivation of this paper is to extend and to apply the HGA to the solving of FMP problems in [22] and [23], respectively. To solve the related CF problems, the HGA will be based on an FLP model, an FMLP model, and a goal programming (GP). The performance of HGA will be compared with the performances of the traditional FMP models.

MATHEMATICAL FORMATION OF FUZZY CELL FORMATION PROBLEMS

Achieving optimal clustering results by grouping machine cells and part families according to a *fuzzy* (imprecise) *goal* (objective) and *resources* (constraints) is the intention of the fuzzy CF problem. Fuzzy mathematical programming is a suitable new approach to modeling this problem. Two of our previous papers [22] [23] have attempted to deal with EEs and to form workcells simultaneously under the fuzzy environment. There, two FMP models and a GP model were considered. The first FMP model was the FLP model in [22]. A min-add operator and a linear nonincreasing membership function were applied to optimize total cost of dealing with EEs under the fuzzy constraint. The second FMP model is the FMLP model in [23]. The purpose of this model is to trade off two conflicting objective functions in the cell formation (CF) problems – minimizing total costs of dealing with EEs and maximizing group efficacy (GE). Since, according to [23], the clustering results of the asymmetric model are the same as those of the symmetric model when the ratio *Pc/Ps* is equal to one, this paper considers only the asymmetric model. A GP model also was discussed in [23] because it can obtain the same trade-off value as the FMLP model can. Hence, to verify that the HGA also can be applied in fuzzy CF problems, the three models just mentioned are used in this paper. Their MP formulations are listed in the Appendix. For a detailed explanation, refer to [22] [23].

Notation

The following notation was used in this paper.

Index Sets

Ch_p chromosome index; $p = 1,\ldots, pop\_size$

i machine index; $i = 1,\ldots, m$

j, j' part index; $j, j' = 1,\ldots, n$

k cell index; $k = 1, \ldots, c$

p population size index; $p = 1,\ldots, pop\_size$

Parameters

A_i periodic cost of acquiring machine type i

C_i periodic capacity of machine type i

D_j periodic forecast demand for part j

F the total fitness of populations

f_p the fitness value for each chromosome

I_j incremental cost for moving a unit of part j within two cells

NM maximum number of machine types allowed in each cell

NMC_k total number of machine types in cell k

P_c the tolerance value for the fuzzy cost objective function

P_r tolerance values for the fuzzy NM constraints

P_s the tolerance value for the fuzzy similarity coefficient objective function

Pb_c the probability of crossover

Pb_m the probability of mutation

Pb_p the probability of reproduction

q_p the cumulative reproduction probability for each chromosome Ch_p

$SC_{jj'}$ the similarity coefficients between part j and part j'

Wc the priority weight for the cost function

Ws the priority weight for the similarity coefficient function

Z_c^1 the worst (largest) value of cost function

Z_s^1 the worst (smallest) value of similarity coefficient function

Z_c^0 the best (smallest) value of cost function

Z_s^0 the best (largest) value of similarity coefficient function

<u>Decision Variables</u>

dc^+ the deviation variable for the cost function

ds^- the deviation variable for the similarity coefficient function

M_{ijk} number of machines i dedicated to cell k for producing part j

O_{ijk} units of part j to be subcontracted as a result of machines type i not being available within cell k

X_{ik} =1, if machine i is assigned to cell k; 0, otherwise

Y_{jk} =1, if part j is assigned to cell k; 0, otherwise

Z_{ijk} number of intercellular transfers required by part j as a result of machine type i not being available within part cell k

λ minimum value of all membership functions

HEURISTICAL GENETIC ALGORITHM

The Introduction of Heuristical Genetic Algorithm

The HGA is an improvement over traditional GAs, which are methods of direct random search that work according to the biological reproduction process. Based on a powerful searching ability, that is, (1) considering and evaluating many feasible points in the search space simultaneously and (2) applying probabilistic theory to direct the search, GAs are employed to solve certain nonpolynomial (NP) hard class problems – for instance, facility layout problems [24]. Figure 1 shows the traditional GA procedure. Key steps include (1) selecting the fitness function from the mathematical model; (2) defining the representational chromosome scheme; (3) determining system parameters; (4) generating initial populations; (5) using the reproduction operator to produce better-fitted populations; (6) using the crossover operator; and (7) using the mutation operator. Two major weaknesses in GAs can be found. First, unsuitable system parameters lead to poor performance. Second, random crossover and mutation diminish executing efficiency. Based on the structure in Figure 1, the HGA applies certain heuristics related to characteristics of the CF problem in the last two steps to remedy these two weakness and to improve executing efficiency.

(1) Crossover. Two modifications are included. First, about the crossover range for using the single-point crossover operator, the HGA is limited only to parts genes. Second, a heuristic based on similarity coefficients (SCs) between parts is applied in the HGA after

```
* Initialization.
    1. Develop a mathematical model for the problem.
    2. Define the chromosome and its representational scheme.
    3. Determine system's parameters, such as population size, number of generations,
       crossover probability, and mutation probability.
    4. Generate initialize populations which satisfy constraints.
* Reproduction.
    5. Compute the fitness value, $f_p$, for each chromosome $Ch_p$.
    6. Calculate the total fitness of populations $F$, $F = \sum_{p=1}^{pop\_size} f_p$.
    7. Compute the reproduction probability for each $Ch_p$, $Pb_p = f_p / F$.
    8. Calculate the cumulative reproduction probability for each $Ch_p$, $q_p = \sum_{i=1}^{p} Pb_i$.
    9. For each population $p$, elect the best new population, $p \leq$ pop_size.
       Generate a random number $n$ from a range of [0,1].
       If $(n<q_1)$ then choose $Ch_l$
       Else, if $(q_{p-1}<n<q_p)$ then choose $Ch_p$.
* Crossover.
    10. Selecting two chromosomes each time, $Ch_p$ and $Ch_{p+1}$, $p \leq$ pop_size-1.
        Generate a random number $n$ from range of [0,1].
        If $(n < Pb_c)$ then choose a suitable operator and do the crossover.
        Otherwise, keep the same values of $Ch_p$ and $Ch_{p+1}$.
* Mutation.
    11. Randomly select a bit $i$ for mutation, ($i \leq$ pop_size $\times$ total length of a chromosome).
        Generate a random float number $n$ from the range [0,1].
        If $(n < Pb_m)$ then mutate bit $i$.
```

Figure 1. The Generic Procedure of Genetic Algorithm

the single-point crossover is processed. This heuristic selects a part randomly (from cell one to cell *NM*-1) and merges other parts with an SC value larger than the threshold similarity value. The procedure groups higher similarity parts automatically. Hence, replacing the heuristic based on the problem characteristics with random crossover can ensure a positive search direction.

(2) Mutation. Unlike the traditional GA, which mutates genetic values randomly, the HGA applies two heuristic methods to determine the gene value for each machine and to update the gene value for each part. For instance, in mutation 1, the first step is to compute total parts numbers (TP) required for processing by each machine in each cell. The second step is to assign the cell number with the largest TP value as the gene value for each machine. This new heuristic method is based on a characteristic of the CF problem – that of minimizing the number of EEs in clustering results – and can decrease search space and save time for random mutation.

These two improvements are, in fact, the main advantages of the HGA. They also lead to a number of user-friendly characteristics that the traditional GA does not have.

(1) Generation of initial populations. As with those of the traditional GA, initial populations of the HGA normally are generated randomly, and the method for handling constraints is the variable restriction method. The only difference between the HGA and the traditional GA is that the decision maker does not need to consider the *NM* value, which is not determined easily. This improvement improves usability for the decision maker.

(2) The determination of system parameters. The values of certain required parameters such as population size, maximum generation numbers, crossover and mutation probabilities, and seeds are quite sensitive to GA implementation performance. Improper parameters will not lead to the optimal solution. The HGA is developed to remedy this weakness, for it need not consider the probabilities of crossover and mutation. The parameters required by the HGA are one seed, population size less than 100, and generation number less than five. This parameters lead to a HGA that is user-friendly.

The Implementation of Heuristic Genetic Algorithm for Fuzzy Cell Formation Problems

The HGA implementation procedure can be summarized as [24]: (1) Initialization. The string representation is defined, system parameters are decided, the fitness function is selected, and initial populations are generated randomly to satisfy problem constraints. (2) Reproduction. The roulette method is applied to select new better-fitted generations. (3) Heuristic crossover. (4) Heuristic mutation. The procedures for solving two FMP models and one GP model by means of a HGA are similar, the only difference lies in the method of calculating the fitness function for reproducing the best generation. The processes for computing the fitness function of three models follow:

(1) The FLP model. The objective function of the FLP is to minimize the value of subtracting $\lambda P_{()}$ from the total costs of dealing with EEs:

$$Min \sum_{k}\sum_{i} A_i R_{ik} + \sum_{k}\sum_{(i,j)\in sp} I_j Z_{ijk} + \sum_{k}\sum_{(i,j)\in sp} S_j O_{ijk} - \lambda P_{()}, \qquad (1)$$

in which λ is calculated from the fuzzy constraints:

$$\sum_{i=1}^{m} X_{ik} + \lambda P_r \leq NM + P_r, \forall k. \tag{2}$$

According to (2), the possible range of total machine numbers in each cell, NMC_k, is NM through $NM + P_r$. The possible range of k is m / NM through C, the maximum cell number. The value of λ, therefore can be calculated by means of the formula

$$\lambda = Min[(NM + P_r - NMC_k)/P_r]; \forall k. \tag{3}$$

Insert (3) in (1), and the fitness value f_p for each chromosome Ch_p can be calculated. Because all possible cell numbers, k, and NM values must be considered for finding the optimal solution, the HGA algorithm must be run through two loops. The outloop is for checking all possible k values increased from m/NM to C; the inloop is for all possible NMC_k values increased from NM to $NM + P_r$.

(2) The FMLP model. The fitness function of FMLP is the same as that of Equation (1). But the equation for calculating λ value is different because FMLP involves fuzzy inequations (4) to compute total SCs. Furthermore, in this paper, FMLP does not have fuzzy constraints (an asymmetric model), and Equation (5) is used to calculate λ value. Therefore, the fitness value f_p for each chromosome Ch_p can be obtained by inserting (5) in (1):

$$\sum_{k=1}^{c} \sum_{j=1}^{n} \sum_{j' \neq j}^{n} SC_{jj'} Y_{jk} Y_{j'k} - \lambda P_s \geq Z_s^1. \tag{4}$$

$$\lambda = (\sum_{k=1}^{c} \sum_{j=1}^{n} \sum_{j' \neq j}^{n} SC_{jj'} Y_{jk} Y_{j'k} - Z_s^1) / P_s . \tag{5}$$

(3) Goal programming. As with FLP and FMLP, based on the objective function GP (A14), the fitness value f_p of HGA for each chromosome Ch_p becomes

$$f_p = W_c \times dc^+ / P_c + W_s \times ds^- / P_s . \tag{6}$$

where dc^+ and ds^- are calculated according to the related constraints – Equations (7) and (8):

$$\sum_{k} \sum_{i} A_i R_{ik} + \sum_{k} \sum_{(i,j) \in sp} I_j Z_{ijk} + \sum_{k} \sum_{(i,j) \in sp} S_j O_{ijk} - dc^+ = Z_c^0, \text{ and} \tag{7}$$

$$\sum_{k=1}^{c} \sum_{j=1}^{n} \sum_{j \neq j'}^{n} SC_{jj'} Y_{jk} Y_{j'k} + ds^- = Z_s^0 . \tag{8}$$

Modifying the related algorithm of computing f_p, the entire implementation procedures of HGA based on GP and FMLP models are the same as those based on FLP model.

COMPUTATIONAL EXPERIENCE

To aid in the comparison of HGA performance with FMP and GP performances, nine data sets were used. The first three were copied from [22] [23]. The other six were chosen from [24] according to these critical factors: (1) number of EEs greater than five, (2) number of machine types greater than 10, and (3) *NM* value not greater than 8. The clustering results of data sets, therefore, will be affected greatly by fuzzy constraints or objectives. If these parameters were not enforced, it would not make sense to apply FMP, for the clustering results of FMP always would be the same as those of traditional MP. This paper assumes the *NM* value of the first five data sets to be "4," that of the last set to be "6," and that of the remaining three sets to be "5." The other relevant characteristics of these data set parameters are listed in the Table 1. Regarding the numerical values of required data sets, such as processing time for each part, costs involved, and part demand, these are generated randomly by a computer program based on the mean value and the variance of data set 1 in [17].

Based on three related FMP models, the evaluation function of HGA was modified and implemented to facilitate its comparison with FLP, FMLP, and GP in Tables 1, 2, and 3, respectively. This study uses the LINDO (linear interactive and discrete optimizer) package running on an IBM compatible 486 DX2 66 MHZ computer to solve all models. The formulations required for running LINDO were generated by a generator coded in BASIC language. The GAs were coded in ANSI C and run in two different computer platforms: the DECstation 5000 model 240 workstation (for data set 9) and the IBM compatible 486 DX2 66 MHZ computer (for the remaining sets). Performances were measured in terms of EE number [9], CPU time, similarity coefficient [23], and cost. Several observations can be made about the computational results.

First, in terms of computational efficiency, the HGA is much better than FMP or GP in that CPU computational time is much shorter for the HGA. Moreover, as problem size increases, it becomes impracticable to use FMP or GP to solve the problem (most data sets require more than one day). The HGA required less than four minutes in a 486/66 PC to

Table 1. Summary of Computational Results
(The Heuristic Genetic Algorithm vs. Fuzzy Linear Programming)

NO.	SIZE +	Matrix	NM	Z_c^0	P_c		The Heuristic GA						FLP @	
	(m X n)	density	(P_r)	(in $)	(in $)	Ref.	Max. popu. (Generations)	Cost (in $)	CPU time# (seconds)	NMc &	No. EEs	SCs	Cost (in $)	CPU time (seconds)
1	9 x 10	0.322	4 (2)	300,250	166,000	[17]	70 (5)	301,630	5.44	(4,5)	8	-6.59	301,695	1,139
2	9 x 9	0.395	4 (2)	161,930	121,000	[12]	30 (4)	168,200	1.868	(4,5)	3	3.21	168,200	229
3	14 x 24	0.182	4 (2)	0	209,600	[13]	30 (4)	67,774	5.714	(4,5,5)	2	2.38	67,758	1,398
4	12 x 10	0.317	4 (3)	154,487	225,305	[14]	30 (4)	237,477	4.341	(3,4,5)	5	5.26	237,477	1,359
5	12 x 19	0.33	4 (3)	543,286	695,473	[27]	30 (4)	675,239	6.154	(6,6)	14	3.34	675,796	55,748
6	16 x 30	0.242	5 (3)	645,599	742,194	[20]	30 (4)	970,574	15.714	(6,4,6)	17	-12.7	–	–
7	16 x 43	0.183	5 (4)	542,663	801,735	[2]	80 (5)	565,408	124.83	(3,6,7)	19	-94.6	–	–
8	24 x 40	0.135	5 (3)	288,280	282,373	[5]	30 (4)	445,427	184.56	(4,4,5,5,6)	8	13.99	–	–
9	40x100	0.105	6 (3)	1,929,433	418,234	[6]	90 (5)	2,043,226	206.11	(7,6,5,5,5, 5,4,3)	31	253.3	–	–

+: m: the total numbers of machine types; n: the total numbers of part types.

# : Data set 9 was run in the workstation owing to memory limitations of the PC.

&: NMc: Number of machine types in each cell.

@: The values of EE and NMc are same as those of the heuristic GA.

–: Computer running time exceeds one day (86,400 seconds).

Table 2. Summary of Computational Results

(The Heuristic Genetic Algorithm vs. Fuzzy Multiobjective Linear Programming)

NO.	SIZE	NM	Z_c^0	P_c			The Heuristic GA#						FMLP @	
	(m X n)	(P_r)	(in $)	(in $)	Z_s^0	P_s	Cost (in $)	CPU time (seconds)	Max. popu. (Generations)	NMc &	No. EEs	SCs	Cost (in $)	CPU time (seconds) PC
1	9 x 10	4 (2)	300,125	226,709	0.41	5.87	325,784	7.198	70 (4)	(4,5)	8	-2.66	325,892	5,138
2	9 x 9	4 (2)	145,533	143,627	4.64	3.26	168,200	2.857	30 (4)	(4,5)	3	3.21	168,200	822
3	14 x 24	4 (2)	0	209,600	20.94	18.5	67,774	8.791	30 (4)	(4,5,5)	2	2.38	67,758	67,534
4	12 x 10	4 (3)	154,487	82,990	5.263	7.37	220,947	5.824	30(4)	(3,3,6)	5	5.263	220,908	1,359
5	12 x 19	4 (3)	543,286	649,518	12.55	9.21	788,385	24.945	70 (4)	(6,6)	14	7.391	–	--
6	16 x 30	5 (3)	645,599	448,941	22.73	97.48	939,332	10.67	20 (4)	(7,4,5)	16	-4.293	–	--
7	16 x 43	5 (4)	542,663	860,799	13.63	153.5	1,102,827	155.28	70 (5)	(7,4,3,2)	21	-21.85	–	–
8	24 x 40	5 (3)	288,280	310,084	67.22	74.32	520,178	249.94	50 (4)	(6,5,4,4, 3,2)	9	49.81	–	–
9	40x100	6 (3)	1.929.433	418,234	350.1	196.2	2,108,586	229.55	60 (5)	(6.5,5,5,5 ,4,4.3,3,)	32	310.31	–	--

: Data set 9 was run in the workstation owing to memory limitations of the PC.

&: NMc: Number of machine types in each cell.

@: The values of EE and NMc are same as those of the heuristic GA.

– : Computer running time exceeds one day (86,400 seconds).

Table 3. Summary of Computational Results

(The Heuristic Genetic Algorithm vs. Goal Programming)

NO.	SIZE	NM	Z_c^0	P_c			The Heuristic GA [#]						Goal Programming[@]	
	(m X n)	(P_r)	(in $)	(in $)	Z_s^0	P_s	Cost (in $)	CPU time (seconds)	Max. popu. (Generations)	NMc [&]	No. EEs	SCs	Cost (in $)	CPU time (seconds)
1	9 x 10	4 (2)	300,125	226,709	0.41	5.87	325,784	7.473	70 (4)	(4, 5)	8	-2.66	325,892	265,982
2	9 x 9	4 (2)	145,533	143,627	4.64	3.26	168,200	3.022	30 (4)	(4,5)	3	3.21	168,200	2,244
3	14 x 24	4 (2)	0	209,600	20.94	18.5	67,774	8.846	30 (4)	(4,5,5)	2	2.38	–	–
4	12 x 10	4 (3)	154,487	82,990	5.263	7.37	220947.8	6.099	30 (4)	(3,3,6)	5	5.063	220,908	43,231
5	12 x 19	4 (3)	543.286	649,518	12.55	9.21	788.385	32.69	90 (5)	(6.6)	14	7.391	–	–
6	16 x 30	5 (3)	645.599	448.941	22.73	97.48	939.332	26.429	30 (4)	(7,4.5)	16	-4.293	–	–
7	16 x 43	5 (4)	542,663	860,799	13.63	153.5	1,102,827	205.8	100 (5)	(7,4,3,2)	21	-21.85	–	–
8	24 x 40	5 (3)	288,280	310,084	67.22	74.32	520,178	264.95	50 (5)	(6.5,4,4, 3,2)	9	49.81	–	–
9	40x100	6 (3)	1.929.433	418.234	350.1	196.2	2,108.586	266.28	60 (5)	(6.5,5,5,5 ,4,4,3,3,)	32	310.31	–	–

#: Data set 9 was run in the workstation owing to memory limitations of the PC.

&: NMc: Number of machine types in each cell.

@: The values of EE and NMc are same as those of the heuristic GA.

–: Computer running time exceeds one day (86,400 seconds).

solve data sets of small or medium size (the first 8 data sets) and less than two minutes in a workstation to solve the largest problem (data set 9). Thus, the GA is an efficient algorithm and can meet real-world application demands.

Second, in terms of clustering results as indicated by cost, NMc, EE, and SC, those of the HGA are the same as those of FLP, FMLP and GP. As in Tables 1, 2, and 3, when the available data sets are compared, the HGA can produce exactly what FMP and GP can. The HGA, therefore, is a suitable means of solving CF dealing with EEs. Note that only five of data sets in Table 1, four of data sets in Table 2, and three of data sets are available for comparison, because in later data sets FMP and GP have poor computational efficiency.

Third, the HGA is user friendly. As stated in [24], compared with the traditional GA, HGA seldom requires system parameters. Furthermore, there are three advantages to the HGA that FMP and GP cannot achieve: (1) relaxation of the requirement to execute the generator to build formulations; (2) display of the performances such as cost, EE, and SC in the output; and (3) easy plotting and reading of clustering results. (For the partial output achieved by implementing HGA, see Appendix B.) These differences demonstrate how friendly an algorithm HGA is for the decision maker.

Fourth, in terms of all clustering results, the HGAs based on FMLP and GP models are exactly the same. Execution time for the former also is shorter than that for the latter. As can be inferred from Tables 2 and 3, all clustering results in terms of cost, NMc, EE, and SC are the same; and the execution times of all data sets in Table 2 are shorter than those in Table 3.

These findings confirm the conclusion in [23] that the FMLP approach outperforms GP in terms of execution time.

Finally, clustering results can be obtained by applying FMLP that cannot be by applying FLP. Based on the FLP model, FMLP considers one more objective function, namely maximum SCs, to trade off cost and GE, and additional results are available to the decision maker. After the HGA is implemented, the clustering results of FLP and FMLP for all data sets can be obtained and compared. The SC value of FLP is not larger than that of FMLP except in data sets 2 and 3, which have a small complexity. (See Tables 1 and 2.).

CONCLUDING REMARKS

In this paper, the HGA proposed in [24] was extended to solve a CF problem in a fuzzy environment. The HGA evaluation function was modified according to FLP, FMLP, and GP models, and its results were compared with the results of the models. From the computational experience analyses, a number of conclusions can be drawn.

First, the HGA is a good, practical algorithm. It not only can optimize fuzzy CF problems as the FMP and the GP do, it also outperforms these approaches in terms of computational efficiency. The HGA can satisfy the three required conditions of a good algorithm: (1) global convergence. The optimal solution should be found for any starting point. No matter what the initial randomly generated points, the HGA will obtain the global optimal solution. (2) Convergence rate. The faster from X^0 to X^*, the better the algorithm. Two special heuristic operators make the HGA converge to the optimal solution quite rapidly.

(3) Efficiency. A good algorithm is expected to calculate X^K from X^{K-1} easily. No complicated formulation or recurrent algorithm exists in the HGA, and new generations can be computed quite easily by execution of a straightforward heuristic algorithm.

Second, the HGA is a user-friendly algorithm. It has two user-friendly characteristics: (1) It displays all feasible solutions in terms of performance in the computational results, in which certain solutions might not have been obtained by FMP or GP approaches. (2) It considers only a few system parameters, e.g., seed, population size, and generation numbers, and does not need to execute a generator to build the formulation.

These advantages promote usage of the HGA in real-world applications. Obviously, using the HGA to optimize fuzzy CF problems is a promising approach. Although scheduling and layout problems have been solved by the GA approach in previous studies [24], such problem have been small; and computational efficiency still is a weakness in the application of GA to larger problems. Hence, based on the results of this paper, we are confident that the HGA can be applied to solve scheduling and layout in workcells that also are CM problems or to modify related heuristic operators to solve other complicated combinatory problems.

APPENDIX A: THE FORMULATION OF FUZZY MATHEMATICAL PROGRAMMING MODELS

The formulations of two fuzzy MP models and a GP used in this paper, with their related operator and membership function, are listed as follows:

(1) Fuzzy Linear Programming

After the min-add operator and the linear non-increasing membership function are used, the formulations of FLP are

$$Min \sum_{k}\sum_{i} A_i R_{ik} + \sum_{k}\sum_{(i,j)\in sp} I_j Z_{ijk} + \sum_{k}\sum_{(i,j)\in sp} S_j O_{ijk} - \lambda P_0, \tag{A1}$$

subject to

$$\sum_{i=1}^{m} X_{ik} + \lambda P_r \le NM + P_r, \ \forall k \tag{A2}$$

$$\sum_{k=1}^{c} X_{ik} = 1, \ \forall i \tag{A3}$$

$$\sum_{k=1}^{c} Y_{jk} = 1, \ \forall j \tag{A4}$$

$$\sum_{k=1}^{c} IC_k \ge 2 \tag{A5}$$

$$\sum_{i=1}^{m} X_{ik} \ge 2\, IC_k, \ \forall k \tag{A6}$$

$$X_{ik} - Y_{jk} + \frac{1}{D_j} Z_{ijk} + \frac{1}{D_j} O_{ijk} + \frac{1}{UC_{ij}} M_{ijk} - U_{ijk} = 0, \ \forall (i,j)\in sp, \forall k \tag{A7}$$

$$\sum_{(i,j)\in sp} M_{ijk} \le R_{ik}, \ \forall i, \ \forall k \tag{A8}$$

$$Q_i \le \sum_{(i,j)\in sp} UC_{ij}\Big(1 - \sum_{k} V_{ijk}\Big) + 1, \ \forall i \tag{A9}$$

$$\sum_{k}\sum_{(i,j)\in sp} \frac{P_{ij}}{C_i} Z_{ijk} \le Q_i - \sum_{(i,j)\in sp} UC_{ij}\Big(1 - \sum_{k} V_{ijk}\Big), \ \forall i \tag{A10}$$

$$X_{ik}, Y_{jk}, U_{ijk}, V_{ijk} = 0 \ or\ 1; \ R_{ik}, Q_i = \text{general integer} \tag{A11}$$

(A1) and (A2) are the equivalent transformed formulations of (A12) and (A13) after applying the add-min operator. Constraint sets (A3) and (A4) ensure that each machine

$$\sum_{i=1}^{m} X_{ik} \tilde{\leqq} NM , \forall k \tag{A12}$$

$$\sum_{k}\sum_{i} A_i R_{ik} + \sum_{k}\sum_{(i,j)\in sp} l_j Z_{ijk} + \sum_{k}\sum_{(i,j)\in sp} S_j O_{ijk} \tilde{\leqq} Z^0 = Z^1 - P_0 \tag{A13}$$

and part is assigned into only one cell. Constraint (A5) prevent cell number less than "2." Constraint set (A6) prevents clustering results in fewer than two cells. Constraint set (A7) guarantees that the demand of exceptional part j can be shared by the combination of duplicated machine i transferred within cells and subcontracted. Constraint set (A8) calculates the number of machine type is needed to be dedicated in cell k for producing exceptional parts. Constraint set (A9) determines the number of machine type i needed in each cell. It sums all the utilization capacities of machine type i for all relative parts j ($\sum_{(i,j)\in sp} UC_{ij}$) not belonging to the EEs ($1 - \sum_{k} V_{ijk}$). Constraint set (A10) ensures that the numbers of intercellular transfers between machines type i do not exceed the available machine capacity.

(2) Fuzzy Multiobjective Linear Programming:

The formulations of FMLP with the asymmetric model are the same as those of FLP, except replacing (A14) with (A2).

$$\sum_{k=1}^{c} \sum_{j=1}^{n} \sum_{j' \neq j}^{n} SC_{jj'} Y_{jk} Y_{j'k} - \lambda P_s \geq Z_s^1 \tag{A14}$$

Constraint set (A14) is transformed from (A15) by using the linear nondecreasing membership function. Constraint set (A15) represents the total similarity coefficients is not less than Z_s^0.

$$\sum_{k=1}^{c} \sum_{j=1}^{n} \sum_{j \neq j'}^{n} SC_{jj'} Y_{jk} Y_{j'k} \tilde{\geq} Z_s^0 \qquad \text{(A15)}$$

(3) Goal programming:

Replacing the objective function (A16) with (A1) and constraints (A17) and (A18) with (A2), the formulations of GP are the same as those of the FLP.

$$Min\ W_c \cdot \frac{dc^+}{P_c} + W_s \cdot \frac{ds^-}{P_s} \qquad \text{(A16)}$$

$$\sum_k \sum_i A_i R_{ik} + \sum_k \sum_{(i,j) \in sp} I_j Z_{ijk} + \sum_k \sum_{(i,j) \in sp} S_j O_{ijk} - dc^+ = Z_c^0 \qquad \text{(A17)}$$

$$\sum_{k=1}^{c} \sum_{j=1}^{n} \sum_{j \neq j'}^{n} SC_{jj'} Y_{jk} Y_{j'k} + ds^- = Z_s^0 \qquad \text{(A18)}$$

After dividing by the related scale, P_c and P_s, total deviation value for total costs and SCs is minimized in Equation (A16). Three types of costs associated with EEs are to be minimized to approach Z_c^0 in constraint (A17). Total similarity coefficients of the pairs of parts in all groups are maximized to Z_s^0 in (A18).

APPENDIX B: AN EXAMPLE OF THE PARTIAL HEURISTICAL GENETIC ALGORITHM OUTPUT

The partial output of the HGA based on the FMLP for solving data set 6 (16 X 30) is listed. All clustering results in terms of execution time, EE, GP, SC, and the plot of machines and parts in cells are involved in an optimal solution.

```
====> The execution time is 10.67 seconds

 The optimal solution is:

 1 1 3 1 2 3 1 1 3 2 1 1 3 2 3 2 1 1 2 1 3 2 1 2 1 1 2 1 3 2 2 1 2 1 3 1 2
 1 3 2 3 2 3 3 3 1  614843.25    16     0.556    -4.293

In cell  1==>

Machines=> 1, 2, 4, 7, 8, 11, 12,
Parts====> 1, 2, 4, 7, 9, 10, 12, 16, 18, 20, 22, 30,

In cell  2==>

Machines=> 5, 10, 14, 16,
Parts====> 3, 6, 8, 11, 14, 15, 17, 21, 24, 26,

In cell  3==>

Machines=> 3, 6, 9, 13, 15,
Parts====> 5, 13, 19, 23, 25, 27, 28, 29,

 The total numbers of machine i are set up in its machine cell:

 Q[ 1] =  5, Q[ 2] =  5, Q[ 3] =  5, Q[ 4] =  6, Q[ 5] =  7, Q[ 6] =  4,
 Q[ 7] =  7, Q[ 8] =  7, Q[ 9] =  6, Q[10] =  7, Q[11] =  4, Q[12] =  5,
 Q[13] =  2, Q[14] =  6, Q[15] =  5, Q[16] =  4,

 The total numbers of machine i are duplicated in cell k:

 dup[ 1][ 3]=   1, dup[ 7][ 3]=   1, dup[ 8][ 3]=   1, dup[11][ 2]=   1,
dup[12][ 3]=  1, dup[13][ 1]=  3, dup[13][ 2]=  1, dup[14][ 3]=  1,
dup[15][ 1]=  1, dup[15][ 2]=  1, dup[16][ 1]=  1, dup[16][ 3]=  1,
The total units of part j are subcontracted :

o[ 4][ 3]=      4800, o[19][12]=      5159, o[27][16]=      6205,

Total units of part j are transferred :

z[ 4][ 3]=   10238.295, z[18][10]=   22122.000, z[11][11]=     161.147,
z[19][12]=    3222.446, z[ 7][16]=    6963.386, z[27][16]=   10929.722,

Total costs for dealing with EEs = 939332.2500

    1 2 4 7 910121618202230 3 6 811141517212426 513192325272829
  1,0 0 1 1 1 0 1 0 1 0 1 1|0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
  2,1 0 1 0 0 1 0 1 1 1 0 0|0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  4,0 1 1 1 1 0 0 0 1 0 1 1|0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  7,0 1 1 1 0 0 1 0 1 0 1 1|0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
  8,0 1 1 1 1 0 1 0 1 0 1 0|0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
 11,0 1 0 1 1 0 1 0 1 0 0 1|0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 12,0 1 1 1 0 0 1 0 1 1 1 0|0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0

  5,0 0 0 0 0 0 0 0 0 0 0 0|1 1 1 1 1 1 0 1 1 1|0 0 0 0 0 0 0 0
 10,0 0 0 0 0 0 0 0 1 0 0 0|0 1 1 1 1 1 0 1 1 1|0 0 0 0 0 0 0 0
 14,0 0 0 0 0 0 0 0 0 0 0 0|0 1 1 1 1 1 1 1 0 0|0 0 0 0 0 1 0 0
 16,0 0 0 1 0 0 0 0 0 0 0 0|0 1 1 0 1 1 0 0 1 1|0 0 0 0 0 1 0 0

  3,0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0|0 1 0 1 1 1 1 1|
  6,0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0|0 0 0 1 1 1 0 1|
  9,0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0|1 0 1 1 1 1 1 0|
 13,0 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0|0 1 0 0 0 0 0 1|
 15,0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1|1 0 0 1 1 1 0 1|
```

REFERENCES

[1] Bit, A. K., Biswal, M. P., & Alam, S. S. Fuzzy Programming Approach to Multicriteria Decision Making Transportation Problem, *Fuzzy Sets and Systems*, 1992, 50, 135-141.

[2] Burbidge, J. L. *The Introduction of Group Technology*, John Wiley and Sons, New York, 1975.

[3] Chan, K. C., & Tansri, H. A Study of Genetic Crossover Operations on the Facilities Layout Problem, *Computers and Industrial Engineering*, 1994, 26(3), 537-550.

[4] Chanas, S., Kolodziejczyk, W., & Machaj, A. A Fuzzy Approach to the Transportation Problem, *Fuzzy Sets and Systems*, 1984, 13, 211-221.

[5] Chandrasekharan, M. P., & Rajagopalan, R. Groupability: An Analysis of the Properties of Binary Data Matrices for Group Technology, *International Journal of Production Research*, 1989, 27(6), 1035-1052.

[6] Chandrasekharan, M. P., & Rajagopalan, R. ZODIAC-An Algorithm for Concurrent Formation of Part-Families and Machine-Cells, *International Journal of Production Research*, 1987, 25(6), 835-850.

[7] Chu, C. H. Clustering Analysis in Manufacturing Cellular Formation, *OMEGA*, 1989, 17, 289-295.

[8] Chu, C. H. Mathematical Programming approaches to Cell Formation: A Survey and Assessment, *Proceedings of the National Decision Sciences Conference*, 1994.

[9] Chu, C. H., & Hayya, J. C. Fuzzy Clustering Approach to Manufacturing Cell Formation, *International Journal of Production Research*, 1991, 9, 1475-1478.

[10] Darzentas, J. On Fuzzy Location Model, in *Optimization Models Using Fuzzy Sets and Possibility Theory*, Kacprzyk, J. and Orlovski, S. A. (eds.), Dordrecht, D. Reidel Publishing Company, 1987, 328-341.

[11] Deporter, E. L., & Ellis, K. P. Optimization of Project Networks with Goal Programming and Fuzzy Linear Programming, *Computers and Industrial Engineering*, 1990, 19, 500-504.

[12] Gongaware, T. A., & Ham, I. Cluster Analysis Applications for Group Technology Manufacturing Systems, in *Proceedings of the 9th North American Manufacturing Research Conference (NAMRC)*, 1981, 503-508.

[13] King, J. R. Machine-Component Group Formation in Group Technology, *OMEGA*, 1980, 8(2), 193-199.

[14] McAuley, J. Machine Grouping for Efficient Production, *The Production Engineering*, 1972, 51, 53-57.

[15] Pickens, J. B., & Hof, J. G. Fuzzy Goal Programming in Forestry: An Application with Special Solution Problems, Fuzzy Sets and Systems, 1991, 39, 239-246.

[16] Sasaki, M., Gen, M., & Ida, K. A Method for Solving Reliability Optimization Problem by Fuzzy Multiobjective 0 - 1 Linear Programming, *Electronics and Communications in Japanese*, 1991, 74(3), 106 - 116.

[17] Shafer, S. M., kern, G. M., & Wei, J. C. A Mathematical Programming Approach for Dealing With Exceptional Elements in Cellular Manufacturing, *International Journal of Production Research*, 1992, 30, 1029-1036.

[18] Singh, N. Design of Cellular Manufacturing Systems: An Invited Review, *European Journal of Operational Research*, 1993, 69, 284-291.

[19] Sommer, G., & Pollatschek, M. A. A Fuzzy Programming Approach to An Air Pollution Regulation Problem, *European Journal of Operational Research*, 1978, 10, 303-313.

[20] Srinivasan, G. S., Narendarn, T. T., & Mahadevan, B. An Assignment Model for the Part-Families Problem in Group Technology, *International Journal of Production Research*, 1990, 28(1), 145-152.

[21] Steudel, H. J., & Desruelle, P. *Manufacturing in the Nineties*, Van Nostrand Reinhold, New York, 1992.

[22] Tsai, C.- C., Chu, C. H., & Barta, T. A. Optimization of Manufacturing Cell Formation with Fuzzy Linear Programming, *Working Paper*, Department of Industrial and Manufacturing Systems Engineering, Iowa State University, 1994.

[23] Tsai, C.- C., & Chu, C. H. A Fuzzy Multiobjective Linear Programming Model for Cell Formation, *Working Paper*, Department of Industrial and Manufacturing Systems Engineering, Iowa State University, 1994.

[24] Tsai, C.- C., Chu, C. H., & Barta, T. A. Optimizing the problem of Dealing with Exceptional Elements in Cell Formation: A Heuristic Genetic Algorithm Approach, *Working Paper*, Department of Industrial and Manufacturing Systems Engineering, Iowa State University, 1995.

[25] Wang, Z., & Dang, Y. Application of Fuzzy Decision Making and Fuzzy Linear Programming in Prediction and Planning of Animal Husbandry System in Farming Region, *Cybernetics and Systems Research*, 1984, 2, 563-566.

[26] Wemmerlöv, U., & Hyer, N. L. Procedures for the Part Family, Machine Group Identification Problem in Cellular Manufacturing, *Journal of Operations Management*, 1986, 6, 125-147.

[27] Witte, J. D. The Use of Similarity Coefficients in Production Flow Analysis, *International Journal of Production Research*, 1992, 18(4), 503-514.

GENERAL CONCLUSIONS

The goal of this dissertation is to develop useful CF approaches to solving CF problems in a fuzzy environment. This thesis has proposed a traditional MP model for grouping parts and machines to cells and for solving the generic problem of dealing with EEs simultaneously. We also have developed a GP model that can obtain the trade off between minimizing total costs of dealing with EE and maximizing GE. Fuzzy linear programming and FMLP are successfully applied, and the results were shown to be better than traditional LP and GP. Eventually, a heuristic GA is proposed to remedy the weakness of FMP when solving big problems. Our experimental results have shown efficient performances, in particular, for big problems.

This chapter summarizes the contributions of the dissertation. Several directions for future research also are suggested.

Contributions

The traditional mathematical model and FLP were proposed and compared in the first paper, which makes four major contributions. First, it proposes a sophisticated linear programming model that has the following functions: (1) it solves the generic CF problem and minimizes the total cost of dealing with EEs simultaneously. (2) It considers the model's available machine capacity. Then the policy for dealing with EEs is known exactly and is economical. (3) It determines the number of machines needed from the model. The model can calculate the costs of dealing with EEs and the minimum investment cost for machines.

Second, the paper applies the FLP approach for the first time to model CF problems. A state-of-the-art review of FLP is presented. Fuzzy linear programming implementation procedure and related key issues are discussed. After reading this review, a novice should be able to apply FLP easily. Third, the paper proposes a new fuzzy operator that is efficient and robust. Fourth, it assesses the relative performances of different membership functions and operators. Results suggest a basis on which to select membership functions and fuzzy operators.

The second paper makes three contributions. First, it proposes a suitable formula for making the relation between GE and SC in direct proportion – that is, for making SC value grow as GE does. This formula therefore can optimize GE value by maximizing total SC in the MP model. Second, the paper applies GP for the first time to obtain the trade-off between total costs of dealing with EEs and GE. Optimal cell numbers then can be deduced. Third, the paper compares an FMLP model with the new operator (proposed in the first paper) with GP performances. Fuzzy multiobjective linear programming is shown to outperform GP. The eminence of fuzzy set theory in MLP is confirmed.

An efficient heuristic GA is proposed and implemented in the third and fourth papers, whose two major contributions can be summarized as follows. First, new heuristic crossover and mutation operators are proposed to enhance the efficiency of implementing GA and to reduce the sensitivity of system parameters. Second, a heuristic algorithm for computing minimum total costs of dealing with EEs is developed to evaluate the fitness values of chromosomes. These two contributions make the HGA a powerful and user-friendly algorithm for solving a CF problem – for dealing with EEs even in the fuzzy environment.

The HGA not only can produce the same clustering results as the MP does, it also can outperform the MP approach in terms of computational efficiency.

Future Research

Applying FMP to CF problems is a relatively new strategy, and many research opportunities remain to be explored. There are at least four logical extensions of this thesis.

First, the required modules for easily creating specified MP models of the CF problem can be developed. These modules can include objective functions, constraints, input data, and their generators. Mathematical programming is a basic approach to obtaining an optimal solution and is a basis for developing heuristic methods. But developing an MP model to solve CF problems in the real world is quite complicated. Hence, designing all required modules and using an interface to create the MP model becomes essential.

Second, application of FMP and comparison of the performances of all fuzzy operators in solving different practical problems can be attempted. The fuzzy operator is a key to implementing FLP and obtaining good results. It would be helpful to compare the performances of all operators in a number of different problems to determine which operator is most robust and efficient.

Third, the HGA can be applied to solve the other CM problems. According to the results of applying the HGA obtained in this dissertation, it should be possible to modify the heuristic evaluation algorithm to solve other CM problems such as scheduling and layout in workcells or to modify related heuristic operators to solve other complicated combinatory problems.

Finally, an intelligent fuzzy simulator can be developed. With the assistance of a fuzzy simulator, a user can plot and examine the membership function, study its impact and parameters, and display CF results on a real-time basis. This simulator will be capable of expanding (1) to allow users to call a generator to formulate an FMP model and to convert it to the conventional MP and (2) to integrate the GA algorithm to solve the MP problem.

BIBLIOGRAPHY

[1] Ben-Arieh, D. and Triantaphyllou, E., "Quantifying Data for Group Technology with Weighted Fuzzy Features," *International Journal of Production Research*, 30, 6, 1285-1299 (1992).

[2] Boctor, F. F., "A Linear Formulation of the Machine-Part Cell Formation Problem," *International Journal of Production Research*, 29, 343-356 (1990).

[3] Chan, H. M. and Milner, D. A., "Direct Clustering Algorithm for Group Formation in Cellular Manufacturing," *Journal of Manufacturing System*, 1, 65-74 (1982).

[4] Chu, C. H., "Clustering Analysis in Manufacturing Cellular Formation," OMEGA: *International Journal of Management Sciences*, 17, 289-295 (1989).

[5] Chu, C. H. "Recent Advances in Mathematical Programming for Cell Formation," in Kamran, A. K. and Liles, D. H. (eds.), *Planning , Design and Analysis of Cellular manufacturing System* (in press).

[6] Chu, C. H. and Hayya, J. C., "A Fuzzy Clustering Approach to manufacturing Cell Formation," *International Journal of Production Research*, 29, 7, 1475-1487, (1991).

[7] Chu, C. H. and Shih, W. L., "Simulated Annealing for Manufacturing Cell Formation," *Proceedings of the National Decision Sciences Conference*, San Francisco, CA., (1992).

[8] Kern , G. M. and Wei, J. C., "The Cost of Eliminating Exceptional Elements in Group Technology Cell Formation," *International Journal of Production Research*, (1990).

[9] Kumar, K. R. and Vannelli, A., "Strategic Subcontracting for Efficient Disaggregated Manufacturing," *International Journal of Production Research*, 25, 1715-1728 (1987).

[10] Kusiak, A., "The Generalized Group Technology Concept," *International Journal of Production Research*, 25, 561-569 (1987).

[11] Kusiak, A. and Chow, W. S., "Efficient Solving of the Group Technology Problem," *Journal of Manufacturing System*, 6, 117-124 (1987).

[12] Li, J., Ding, Z. and Lei, W., "Fuzzy Cluster Analysis and Fuzzy Pattern Recognition Methods for Formation of Part Families," *Proceedings of the 14th North American Manufacturing Research Conference* (NAMRC), 558-563 (1986).

[13] McAuley, J., "Machine Grouping for Efficient Production," *The Production Engineering*, 51, 53-57 (1972).

[14] Seiforddini, H. and Wolef, P. M., "Application of the Similarity Coefficient Method in Group Technology," *IIE Transaction*, 18, 271-277 (1986).

[15] Seiforddini, H., "Duplication Process in Machine Cells Formation in Group Technique," *IIE transaction*, 21, 382-388 (1989).

[16] Sen, M. K., Gupta, A. D., Stoffa, P. L., Lake, L. W., and Pope, G. A., "Stochastic Reservoir Modeling Using Simulated Annealing and Genetic Algorithm," *Proceeding of the 1992 SPE Annual Technical Conference and Exhibition*, 939-950 (1992).

[17] Shafer, S. M., kern, G. M., and Wei, J. C., "A Mathematical Programming Approach for Dealing With Exceptional Elements in Cellular Manufacturing," *International Journal of Production Research*, 30, 1029-1036 (1992).

[18] Singh, N., "Design of Cellular Manufacturing Systems: An Invited Review," *European Journal of Operational Research*, 69, 284-291 (1993).

[19] Vakharia, A. J., Chang, Y. L. and Selim, H. M., "Cell Formation in Group Technology: A Combinatorial Search Approach," *Proceedings of the National Decision Sciences Institute*, 1310-1312 (1992).

[20] Vakharia, A. J., Kaku, B. K. "Redesigning a Cellular Manufacturing System to Handle Long-Term Demand Changes: A Methodology and Investigation" *Decision Sciences*, 24, 5, 909-930 (1993).

[21] Venugopal, V. and Narendran T. T., "A genetic Algorithm Approach to the Machine component Grouping Problem with Multiple Objectives," *Computer and Industrial Engineering*, 22, 4, 469-480 (1992).

[22] Vohra, T., Chen, D. S., Chang, J. C. and Chen, H. C., " A Network Approach to Cell Formation in Cellular Manufacturing," *International Journal of Production Research*, 28, 11, 2075-2084 (1990).

[23] Wei, J. C. and Gaither, N., "An Optimal Model for Cell Formation Decisions," *Decision Sciences*, 21, 2, 416-433 (1990).

[24] Wemmerlöv, U. and Hyer, N. L., "Procedures for the Part Family, Machine Group Identification Problem in Cellular Manufacturing," *Journal of Operations Management*, 6, 125-147 (1986).

[25] Wemmerlöv, U. and Hyer, N. L., "Research Issues in Cellular Manufacturing," *International Journal of Production Research*, 25, 3, 413-431 (1987).

[26] Xu, H. and Wang, H. P., "Part Family Formation for GT Applications Based on Fuzzy Mathematics," *International Journal of Production Research*, 27, 9, 1637-1651 (1989).

[27] Zadeh, L. A., "Fuzzy Sets," *Information and Control*, 8, 338- 353 (1965).

[28] Zhang, C. and Wang, H. P., "Concurrent Formation of Part Families and Machine Cells Based on the Fuzzy Set Theory," *Journal of Manufacturing Systems*, 11, 1, 61-67 (1992).