The Islamic University of Gaza
Deanery of higher studies
Faculty of Engineering
Department of Civil Engineering

# Optimizing the schedule of resource-constrained construction projects using Genetic Algorithms

A thesis submitted in partial fulfillment of the requirements of the degree of Master of Science in Construction Management- Civil Engineering

By:

Tahreer Mohammed Fayyad

Supervisors:

Dr. Mohammad Arafa

Dr. Mamoun Alqedra

2010- 1431

# DEDICATION

I proudly dedicate this thesis to my beloved husband Eng. Ahmad AbdAlQader for his unconditional, total support in any endeavor of my life.

With all loves…

# ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and heartfelt thanks to **Dr. Mohammad Arafa** and **Dr. Mamoun AlQedra**; the supervisors of my thesis, for their strong support and guidance throughout the duration of this research.

Deep thanks and gratitude are also due to my father **Mr. Mohammed Fayyad** and my mother **Mrs. Seham Fayyad** for their infinite support and encouragement. I would like to express my thanks to my husband **Eng. Ahmad AbdAlqader** for his patience, support, encouragement and forbearance during the time in which this work was done. I also offer great thanks to my **brother** and my **sisters** for their love and encouragements.

I deeply thank **Eng. Rami Shamia** for his full help in preparing the GUI of the developed program in this research.

High appreciation is due to my colleagues in the **Civil Engineering Department** at the Islamic University of Gaza for their assistance during this research.

# ABSTRACT

In this research, an optimization technique was developed using Genetic Algorithms (GA) to optimize the schedule of construction project activities in order to minimize the total duration of the project, subjected to both precedence and resources constraints.

Genetic algorithms are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome like data structure and apply recombination operators to these structures so as to preserve critical information. In this research, a new approach was developed in generating the populations of the genetic algorithms generations; that is the “Feasible Solutions Developer operator; (FSD operator)”. This operator enables the user to create completely feasible solutions that satisfy all constraints, and this helps in getting a quick convergence toward the best solution during GA stages, without losing the GA feature of searching global maximum or minimum. Also, a new crossover operator was developed in this study; the procedure of the new crossover operator suit the scheduling problem formulation, and suit the type of the used chromosomes.

An implementation of the developed GA optimization model for resource-constrained construction projects scheduling has resulting in an application program called the CPS Optimizer. It treats the deficiencies of traditional scheduling systems such as CPM, that don’t take resource constraints in consideration, The CPS is a user friendly program with a graphical user interface . Verification and validation of the developed program were applied. A comparison with the results of previous studies and with the feature of “resource leveling” in the commercial construction management programs showed a strong efficiency of the developed model.

The results of the developed optimization model meet the intended goal of achieving the best schedule with the minimum efforts.

# الجدولة المُثلى للمشاريع الإنشائية مقيدة الموارد باستخدام الخوارزميات الجينية

## الملخص

في هذا البحث تم تطوير نموذج محوسب باستخدام الخوارزميات الجينية لعمل جدولة زمنية أمثل لأنشطة المشاريع الإنشائية بحيث يتم الحصول على أقل مدة زمنية للمشروع في وجود قيود زمنية على أنشطة المشروع وقيود على موارد المشروع.

الخوارزميات الجينية هي عائلة مكونة من عدة عمليات حسابية مستوحاة من عملية التطور، حيث يتم عمل ترميز لحلول لمشكلة معينة من خلال كروموسومات بسيطة ومن ثم تطبيق عمليات تهجين على الحلول للحفاظ على الصفات الحسنة والانتقال بها من جيل إلى جيل للوصول إلى الحل الأمثل. تم في هذا البحث تطوير نهجاً جديداً في توليد الحلول الأولية العشوائية، بحيث يتم توليد حلول صالحة تحقق القيود على المسألة، دون إغراق أنفسنا في عدد لانهائي من الحلول الغير صالحة والتي يلزم وقت طويل لتصفيتها عبر خطوات الحل، وهذا النهج يساعد في سرعة الحصول على الحل الأفضل دون الخوف من الانحسار في قيم عليا أو دنيا محلية. كم أنه خلال هذا البحث تم تطوير أداة تهجين تناسب طبيعة المشكلة قيد الدراسة وكذلك نوع الكروموسوم المستخدم.

لقد تم تطبيق النموذج المطور في هذا البحث من خلال برنامج تطبيقي يعالج أوجه القصور في نظم الجدولة التقليدية التي لاتأخذ في الاعتبار القيود المفروضة على الموارد، كما أن للبرنامج واجهة سهلة الاستخدام. وقد عقدت عدة مقارنات بين نتائج البرنامج ونتائج دراسات سابقة ونتائج أمر "تسوية الموارد" الموجود في برامج جدولة المشاريع، حيث أظهرت هذه النتائج كفاءة وتميز النموذج وأثبتت أنه قد حقق الهدف من عملية إنشائه من حيث تحقيق جدولة أمثل للمشاريع بأقل تكلفة في الجهد والوقت.

## Contents

## List of figures

## List of tables

## NOTATIONS

| | |
|---|---|
| *RCPSP:* | Resource- constrained project scheduling problem |
| *AOA:* | Activity on arrow diagram |
| *AON:* | Activity on node diagram |
| *CPM:* | Critical path method |
| *PERT:* | Program evaluation and review technique |
| *Act:* | Activity name |
| *Dur or D:* | The activity normal duration |
| *ES:* | Early start |
| *EF:* | Early finish |
| *LS:* | Late start |
| *LF:* | Late finish |
| *O.R:* | Operations Researches |
| *GA:* | Genetic Algorithms |
| $S_i$: | The starting date of activity i |
| $R_i$ | Resource number i |
| $R_j(t)$: | Required amount of resources in a given time by activity j |
| $R_j^a(t)$: | Available amount of the resource j |

| | |
|---|---|
| *T:* | Total duration of the project (makspan) |
| *$T_i$:* | Total duration of activity i |
| *f(i):* | Fitness value of chromosome i |
| *CPS optimizer:* | The construction projects scheduling optimizer |
| *FSD operator:* | Feasible solutions developer operator |
| *GUI:* | Graphical user interface |

# CHAPTER (1)

## Introduction

## 1.1.Motivation

Resource management is one of the most important aspects of construction project management in today's economy because the construction industry is resource-intensive and the costs of construction resources have steadily risen over the last several decades, (Patrick, 2004).

Every project schedule has its own precedence constraints, which means that each activity can be processed when all its predecessors are finished. In general the purpose of project scheduler is to minimize its completion time or makespan, subject to precedence constraints. A more general version assumes that to develop one or more activities, resources such as tools, equipment, machines, or human resources are needed. Each resource has limited capacity; consequently at a certain moments one activity may not begin their processing due to resource constraints even if all their predecessors are finished, (Franco et al, 2006). This type of problems is called Resource-constrained project scheduling problem (RCPSP) which involves assigning jobs or tasks to a resource or a set of resources with limited capacity in order to meet some predefined objective, (Yang et al, 2001).

Achieving optimal scheduling of a project attracted the attention of many researchers, who employed various techniques including linear programming, dynamic programming, expert systems, neural networks, and genetic algorithms, (Senouci, 2004).

A study on the optimization of the schedule of resource-constrained construction projects using Genetic Algorithm optimization Technique is carried out in this thesis.

## 1.2. Statement of the problem

The traditional scheduling methods such as Critical Path Method (CPM) are not enough for scheduling because they cannot take resource constraints into account. In a resource-constrained project, the amount of resources needed during one or more time periods is limited to less than required by two or more activities occurring simultaneously. A need arises for a new approach to optimize projects schedules which take into consideration the shortage of resource.

## 1.3. Aims and Objectives

The primary aim of this research is to develop a genetic algorithm optimization model to schedule project activities in order to minimize the total duration of the project, subjected to both precedence and resources constraints.

To achieve this aim, the following objectives were achieved:

1. Develop a computer model to automate the schedule of construction projects.
2. Develop a GA optimization model using Matlab to optimize the schedule of resource constrained construction projects scheduling.
3. Implement the model on cases study.

## 1.4. Methodology

The objectives of the research were achieved by developing a computer model solves resource-constrained schedules by implementing the following steps:

- Carrying out a survey on the previous researches that are related to the different approaches to schedule resource-constrained problems.
- Studying the Genetic algorithm approach in order to apply it in the current study.
- Developing a computer model to optimize the process of scheduling projects under a given resource-constraints using genetic algorithm technique.
- Making verification and validation of the model, implementing case studies.
- Comparing the results of the model with results of other approaches or techniques, i.e. MS project and Primavera leveling.

## 1.5. Thesis Outline

The thesis includes six chapters and three appendices. A brief description of the chapters' contents is presented below:

Chapter one highlights the need for research in the field of optimization. The aims and objectives of the research are described. At the end, the structure of the thesis is presented.

Chapter two reviews the construction projects planning and scheduling definitions and techniques as well as resource management.

Chapter three reviews the optimization techniques, including classical and heuristic search, especially the Genetic Algorithms optimization techniques. This chapter ends with a survey of studies carried out in the area of optimizing the schedule of resource-constrained construction projects.

Chapter four describes the formulation of scheduling optimization problem. This chapter introduces a new concept of creation of the initial population, which is called the feasible solutions developer operator (FSD operator).

Chapter five introduces an implementation of the developed GA optimization model for resource-constrained construction projects scheduling which is called the CPS Optimizer program, and shows the results of this program comparing with other approaches.

Achievements and main conclusions of the research with recommendations for future researches are given in chapter six.

# CHAPTER (2)

## Construction Projects Planning and Scheduling

## 2.1. Construction Project management

In construction industry, construction projects require knowledge of project and business management as well as in-depth understanding of the construction process. Project management is an objective-oriented process used throughout business and industry. The management of projects incorporates methods to effectively coordinate and utilize project resources for the accomplishment of one or more objectives. Project management is typically carried out under different types of constrains, including time, labor, material, equipment, cost, and performance of the end product, the planning, scheduling, evaluation, and control of these constraints present distinctive management problems varying from mere nuisance to true impediments for getting the project completed, (Jackson, 2004; Patrick, 2004).

A project management problem typically consists of planning and scheduling decisions, (Goncalves et al, 2006). Planning can be thought of as determining "what" is going to be done, "how", "where", and by "whom". In scheduling, this information is needed in order to determine "when", (Hinze, 2008). By scheduling it is possible to find the right sequence to do the jobs and right schedule, (Kamarainen, 2000).

## 2.2. Projects planning and scheduling

Project is a sequence of unique, complex, and connected activities that are conducted in a coordinated effort to accomplish one or more unique objectives, and must be completed by a specific time, within budget, and according to specifications. It can be stated that a project is a set of jobs, tasks, or activities that must be completed in order to achieve specific project objectives, typically which are unique and nonrecurring, (Patrick, 2004, Wysocki, 2007).

In order to mange a construction project effectively, we must have a plan which includes what is to be accomplished, the technology involved, the resources needed, and the expected time for construction. A very crucial part of this plan is the time-based schedule which is so important for all concerned. The goal of planning is to minimize resource expenditures while satisfactorily completing a given task. Planning aims at producing an efficient use of equipment, materials, and labor, and ensuring coordinated effort. (Steven, 1990, Schexnayder, 2006, Wysocki, 2007).

## 2.3. Construction network scheduling

Schedule is a time-based arrangement of activities planned to take place in order to efficiently complete the project. It helps in coping with complexities, masses of data, and tight deadlines, (Patrick, 2004, Kerzener, 2006).

Network scheduling is a method of scheduling activities by joining them in a series of interconnected links which reflect relationships of activities as assigned by the planner. In project management, decisions depend directly on quantity and quality of information and indirectly on application method and technique used to schedule and control. Until now, the time factor has played a huge rule in scheduling and has always been related to its logic structure, (Steven, 1990, Franco et al, 2006).

### 2.3.1. Definitions

*Activity:* is a discernible task or work function where a particular worker or a crew of workers completes a specific item of work within a prescribed time frame, or it is the element of work that must be accomplished, (Kerzener, 2006).

*Network:* A series of interconnected links with fixed logical relationships that provides valuable information for planning, time studies, scheduling and resource management, (Kerzener, 2006).

*Schedule:* A time-based arrangement of project activities, (Steven, 1990, Franco et al, 2006).

### 2.3.2. Advantages of network scheduling techniques

The advantages of the network scheduling as mentioned by Kerzner (2006) are:

- They form the basis for all planning and predicting and help management decide how to use its resources to achieve time and cost goals.
- They provide visibility and enable management to control projects.
- They help management evaluate alternatives by answering such questions as how time delays will influence project completion, where slack exists between elements, and what elements are crucial to meet the completion date.
- They provide a basis for obtaining facts for decision-making.

- They provide the basis structure for reporting information.
- They reveal interdependencies of activities.
- They identify the longest path or critical paths.
- They aid in scheduling risk analysis.

### 2.3.3. Traditional scheduling techniques

The most common traditional scheduling techniques are: (Jackson,2004, Schexnayder, 2004, Kerzner, 2006; Hinze, 2008),

- Gantt or bar charts.
- Activity on arrow diagram (AOA),
- Activity on node diagram (AON),
- Matrix scheduling,
- Program evaluation and review technique (PERT),
- Critical path method (CPM).

#### *2.3.3.1. Gantt chart*

A Gantt chart presented planned activities as stacked horizontal bands against a back ground of dates (along horizontal axis) as shown in figure (2.1). It is a simple and concise graphical picture for managing a project. It is also easy with a bar chart to compare planned production against actual production, but it does not clearly show the detailed sequence of the activities, and it does not show which activities are critical to the successful, timely completion of the project.

| Activity | Workdays | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| Site clearing | ▨ | | | | |
| Dig trench | | ▨ | | | |
| Assemble pipes | | ▨ | ▨ | | |
| Install pipes | | | | ▨ | |
| Backfills | | | | | ▨ |

*Figure (2.1): Gantt Chart*

### *2.3.3.2. Activity on arrow diagram*

In activity on arrow (AOA) diagram, activities are represented by arrows, and the start and finish of those activities are represented by nodes with AOA diagrams as shown in figure (2.2), activity start and finish times are clearly depicted as separate events, making it a good tool for showing activity sequence. However, the AOA's use of dummy activities is confusing, and time is not shown graphically.

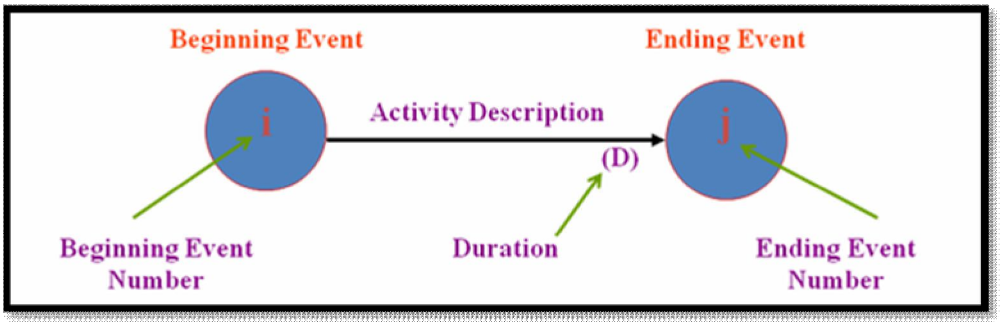


*Figure (2.2): AOA Diagram*

### *2.3.3.3. Activity on node diagram*

The activity on node (AON) diagram represents activities as nodes and the sequence of activities as arrows as shown in figure (2.3). The AON eliminates the need for dummy activities and easy to learn. However, time and project progress are not shown graphically.

The time-scaled network shows graphically the activity starts and finishes as well as the order of activities, but is awkward for large, complex projects.

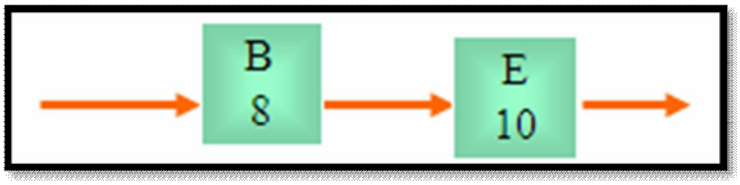


*Figure (2.3): AON Diagram*

### *2.3.3.4. The matrix schedule*

There are other methods that are employed to schedule and control the construction process. One of such method is the matrix schedule. A matrix schedule is one in which spreadsheet is used to show all of the activities on a particular project. This type of matrix approach is used most often on housing projects.

### *2.3.3.5. The critical path method*

The critical path method, or CPM, is a method developed in the 1950s that uses the network modeling process to schedule projects. The critical path represents the sequence or path of activities that takes the longest to complete. The length of the critical path or sum of all critical activity durations is equal to the minimum project duration. The CPM can be applied to those projects that can be divided into distinct activities with fixed durations and a well-defined sequence of work. However, its suitability has been widely criticized. Sriprasert et al (2002) pointed the *CPM major drawbacks as follows:*

- Inability to cope with non-precedence constraints – In the real world, construction posses various kinds of constraints ranging from physical constraints (i.e. topology, space, safety, and environment), contract constraints (i.e. time, cost, quality, and special agreement) to resources and information constraints (i.e. availability and perfection). Unfortunately, CPM considers only time and precedence constraints among activities. Its underlying network representation is proven to be inadequate to represent and integrate more problems in construction.
- Difficulty in plan evaluation and communication – The CPM schedule is graphically presented in either a form of Gantt chart (Bar chart with relationships) or a form of

precedence diagram. To evaluate and communicate the plan, project participants must mentally associate this schedule information with the description of the physical building (i.e. drawings and/or 3D project model) as well as other technical information (i.e. specifications and method statements). This has been proven difficult especially when there is a need to analyze effects of changes to the overall sequence of construction

- Inadequacy for work-face executions – As projects enter their construction phase, detailed short term planning is delegated to engineers, superintendents, or foremen. Rather than employing the CPM, simple Bar chart or activity lists are dominant techniques for this detailed planning. Existing CPM tools do not provide adequate support for analysis of constraints at operational level. Resource allocation, smoothing or leveling procedures are incapable of ensuring full continuity for a production crew or process. For complex projects, field personnel find the CPM schedules confusing and, therefore, less useful. Large amount of efforts are required to re-plan and redraw the network each time it was updated. Furthermore, the CPM has inflexibility and lack of expressiveness to cope with the varied pattern of construction in the field.

#### *2.3.3.6. The program evaluation and review technique*

Another Network-scheduling method that is often used in planning and design application is the program evaluation and review technique (PERT). PERT is a network similar to CPM but which has activity durations expressed as probability distribution rather than fixed values. PERT network also have critical paths, and the project duration can be calculated. However, this duration being a sum of distributions is itself a probability distribution, and hence it is termed stochastic. PERT is the more general of the two technique, but CPM is the one more widely used. CPM also have various and sundry cost and resource control capabilities.

The traditional scheduling methods such as Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT) are not enough for scheduling, because they produce infinite schedules, i.e. they cannot take resource constraints into account. Infinite scheduling may give results, which are not feasible. A schedule is called feasible, if the precedence relations are maintained and the resource and other constraints are satisfied. Efficient methods to solve resource constrained scheduling problems are needed, (Kamarainen et al, 2000). The CPM generates useful information for the project manager to plan and control the project more actively and efficiently, however the CPM has proven to be helpful only when the project deadline is not fixed and the resources are not constrained by either the availability or time, (Goncalves, 2006).

### 2.3.4. Network Definitions

*Early start:* The earliest an activity can start.

*Early finish:* The earliest an activity can finish.

*Late finish:* The latest an activity can finish without delaying project completion.

*Late start:* The latest an activity can start without delaying project completion.

*Free float:* The maximum time an activity can be delayed without delaying the start of the succeeding activity.

*Total float:* The maximum time an activity can be delayed without delaying completion of the project

In this research AON diagram is used to show the data about the activities, and the results obtained and the symbols used are as shown in figure (2.4).

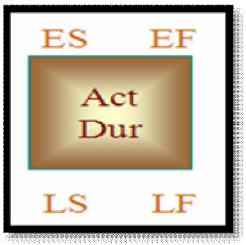


*Figure (2.4): AON symbols*

Where:

Act: is the activity name.

Dur: is the activity normal duration.

ES: Early start of the activity.

EF: Early finish of the activity.

LS: Late start of the activity.

LF: Late finish of the activity.

### 2.3.5. Construction project constraints

The order of activities is based on the timing of some activities relative to the occurrence of other activities. Ordering activities is not necessarily as straightforward as it may seem on the surface. The reason why two activities must be done in a particular order can be termed a constraint. Without constraints on a project, all activities can theoretically begin on the first day of construction. But constraints do exist in the real world and they must be considered in order for a network to be useful. Constraints, as described by Hinze (2008) are of variety of types as followed:

*Physical Constraints:* Physical constraints exist due to the physical process of construction, such as the need to erect forms before concrete can be placed. These are logic constraints that include those defined by “how” (construction methods) the project is to be carried out.

*Resource Constraints:* These constraints are conditions of availability that dictate that certain activities cannot be performed simultaneously because insufficient resources are available. For example, having only one crane available that must be used on two otherwise independent activities might require that the activities be scheduled so that they do not occur at the same time. Similarly, the amount of concrete that can be placed in a single day may be dictated by the production capacity of the concrete batch plant.

*Safety Constraints:* Safety requirement may dictate that activities not occur simultaneously (e.g., overhead and ground level work in the same area, drilling and blasting taking place concurrently) or that a specified sequence occur (e.g., erection of safety barriers before allowing work in an area). Safety considerations may also dictate defining nonworking days for extremely hot or cold days. Project lighting requirements may also dictate by safety concerns.

*Financial Constraints:* Monetary constraints can include the staggering of high-cost activities to minimize cash requirements during construction or the necessity of securing loans prior to undertaking certain portions of a project. Large cash flow items might also be scheduled so that they are incurred in a particular "tax year".

*Environmental Constraints:* Environmental constraints can include the need to carry out mitigation procedures prior to other activities and may also address restriction such as not working in certain areas during such times as spawning season, fish runs, or eagle nesting.

*Management constraints:* Sometimes referred to as "arbitrary", these can be defined simply as additional constrains not otherwise categorized here. They may relate to requirements of supervisory time, consequences of tax strategy decisions, cash flow needs, or the demands of other projects not reflected in the network.

*Contractual Constraints:* The owner may impose constraints on the construction process. The owner of condominium projects may require that a particular phase of the project be fully completed and occupied prior to beginning construction of the next phase. On a remodeling project, the owner may require that construction noise or dust be kept to a minimum if a portion of the facility will remain occupied and in operation.

*Regulatory constraints:* Governmental agencies are also known to impose constraints on the construction process. This includes regulations that are enforced by such federal agencies as the Environmental Protection Agency, land use restrictions that might be imposed by a municipal government, the other regulations enacted at the municipal, county, state, or federal levels. Some of these constraints would also be included among the environmental constraints.

## 2.4. Resources management

Resources management is one of the most important aspects of construction project management in today's economy because the construction industry is resource-intensive and the costs of construction resources have steadily risen over the last several decades. Often the project planner utilizes the time and precedence based schedule as a basis for the management of resources for the project, (Patrick, 2004)

### 2.4.1. Resource-constrained project scheduling

Every project schedule has its own precedence constraints, which means that each activity can be processed when all its predecessors are finished. In general the purpose of project scheduler is to minimize its completion time or makespan, subject to precedence constraints. A more general version assumes that to develop one or more activities, resources such as tools, equipment, machines, or human resources are needed. Each resource has limited capacity; consequently at a certain moments one activity may not begin their processing due to resource constraints even if all their predecessors are finished, (Franco et al, 2006). This type of problems is called *Resource-constrained project scheduling problem (RCPSP)* which involves assigning jobs or tasks to a resource or a set of resources with limited capacity in order to meet some predefined objective, (Yang et al, 2001). In its more general form, the resource-constrained scheduling problem asks the following: Given a set of activities, a set of resources, and a measurement of performance, what is the best way to assign the resources to the activities such that the performance is maximized?

### 2.4.2. Resource scheduling, leveling and allocation

In a resource-constrained project, the amount of resources needed during one or more time periods is limited to less than required by two or more activities occurring simultaneously. Resource management techniques must be employed to distribute resource usage period by period in order to minimize resource variations over time. Resource management can be divided into three main stages or steps for the analysis and solution of resource-constrained schedule as follows: (Patrick, 2004, Wysocki, 2007)

- Resource scheduling.
- Resource leveling.
- Resource allocation.

Each of these serves a slightly different purpose and is applied in the order listed as constraints on project resources become more critical.

*Resource scheduling*, simply provides a profile of resource usage during the duration of the project. It is often referred to as resource loading, and it is likely to be the most common resource management method used by project planners. The network model process and use of computerized scheduling provides the ability to organize resource information and present that information over the duration of the project.

*Resource leveling* is a broadly applied and common term in project resource management. The primary objective in resource leveling is to reduce the peaks and valleys in a resource profile without increasing the project duration. It uses the float time of noncritical activities to redistribute activity start and finish in order to eliminate or at least reduce resource conflicts. Because resource leveling is only applied to noncritical activities, the critical path remains untouched and the project duration is unchanged. This technique is often applied when the project completion date is specified and fixed.

*Resource allocation* is a method where maximum limits are set for each resource and specific amounts of resources are allocated to project activities according to certain scheduling heuristic. The intent is to schedule the activities so that resource limitations are not exceeded while keeping the project duration at minimum. In other words, the project finishes in as short a time as possible under the restrictions of activity precedence, resource limits, and time constraints, in that order. As a result, the project duration or completion date may be extended to keep resources within specified limits.

### 2.4.3. Resources in Gaza Strip

Many studies proved that a need arises for a new approach to schedule projects which take into consideration the limited resources and the special circumstances in Gaza Strip in order to keep the unavoidable extension of the project to a minimum. AL-Najjar (2008) studied the factors influencing time overruns on construction projects in Gaza Strip, the study showed that Gaza strip has special political conditions that leads to the shortage of resources which leads to project delays. Yahia (2004) showed that no attention is given to leveling and allocation of resources in the project scheduling in Gaza Strip, and there is a need to use specific management tools during the life cycle of the project.

# CHAPTER (3)

## Optimization of Construction Projects Scheduling

## 3.1. Introduction

Operation Research (O.R) is a professional discipline still without a fixed content or boundaries. Broadly speaking, it consists of using scientific methods and techniques for decision making in a variety of real life situations. A key word associated with O.R is "Optimization". Optimization means determining the best course of action amongst the different alternatives available in a decision making problem. Optimization problems arise in almost every sphere of human activity. These occur in almost every engineering discipline such as civil engineering.

## 3.2. Optimization

### 3.2.1. Optimization Definition

Optimization is the process of adjusting the inputs to or characteristics of advice, mathematical process, or experiment to find the minimum or maximum output or result as shown in figure (3.1). The input consists of variables; the process or function is known as the cost function, objective function, or fitness function; the output is the cost or fitness. Briefly, optimization is the study of how to find the 'best' (optimum) solution to a problem. The terminology 'best' solution implies that there is more than one solution and the solutions are not of equal value, (Haupt, 2004, Sivanandam et al, 2008).

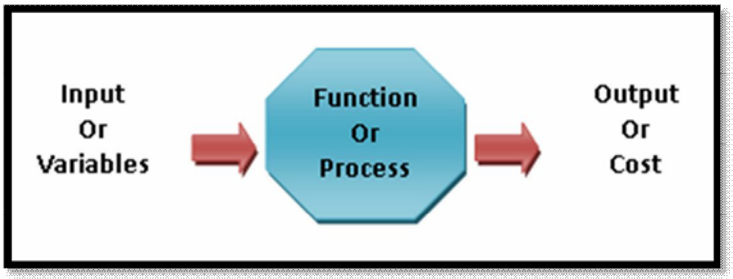


*Figure (3.1): Optimization process*

The objective function, or cost function, is the mathematical representation of the problem to be solved. The function should be formulated such that its values vary with parameter (or parameters). The goal is to find the value of the parameter at which the function obtains an extreme (lowest or highest) value. Many functions have many local minimum or maximum values such as Rastrigin's function as shown in figure (3.2) and (3.3), which is defined as:

$Ras(x) = 20 + X_1^2 + X_2^2 - 10(\cos 2\pi x_1 + \cos 2\pi x_2)$, (Matlab, 2005).

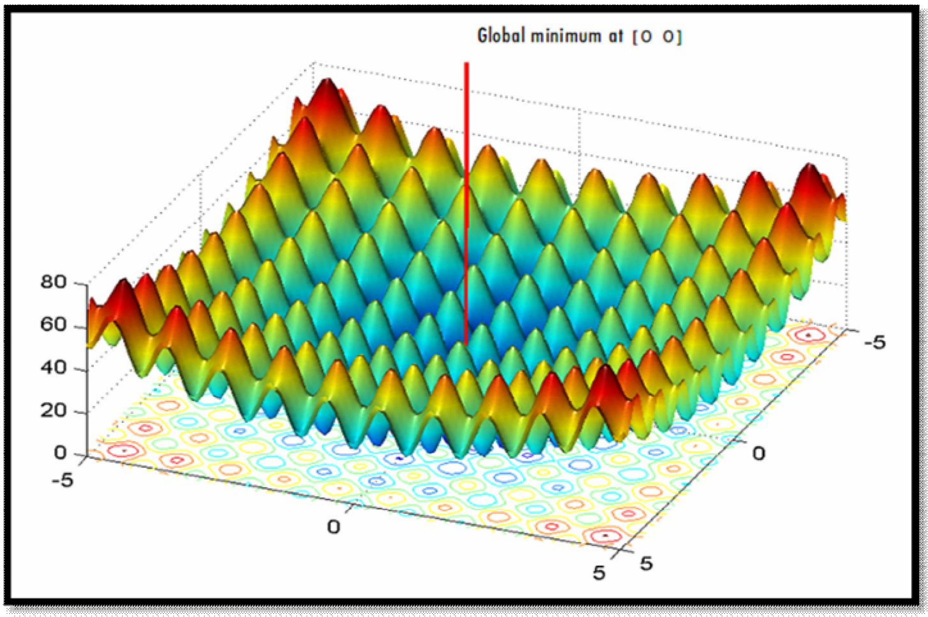


*Figure (3.2): Rastrigin's function*

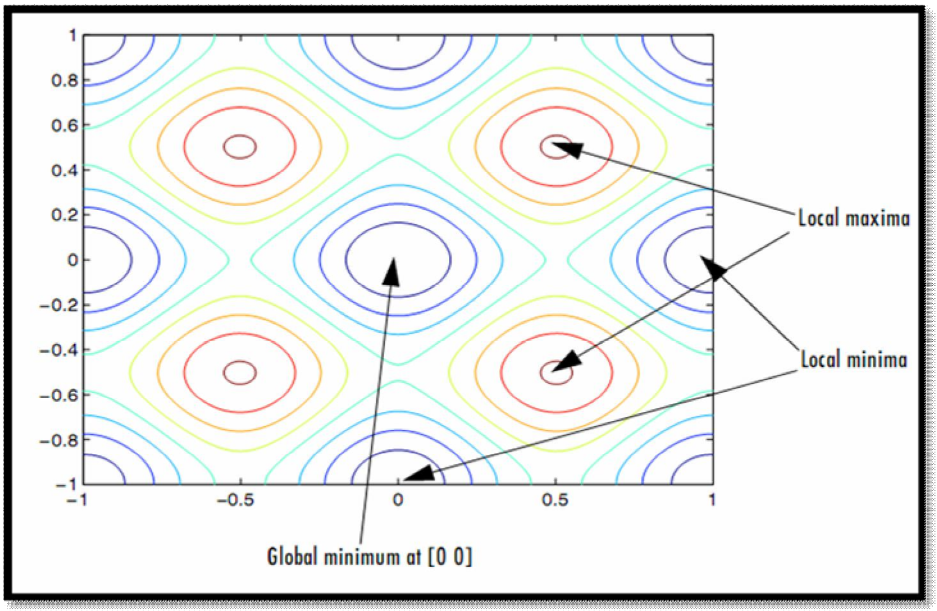


*Figure (3.3): Rastrigin's function local minima & maxima*

### 3.2.2. Categories of optimization

Optimization algorithms are divided into six categories as shown in Table (1). None of these six views or their branches is necessarily mutually exclusive. For instance, a dynamic optimization problem could be either constrained or unconstrained, (Haupt, 2004).

*Table (3.1): Categories of optimization, (Haupt, 2004).*

| Categories of optimization | Description |
|---|---|
| Trial & error<br>Or<br>function | Trial-and-error optimization refers to the process of adjusting variables that affect the output without knowing much about the process that produces the output. In contrast, a mathematical formula describes the objective function in function optimization. Various mathematical manipulations of the function lead to the optimal solution. |
| Single variable<br>Or<br>Multiple variable | If there is only one variable, the optimization is one-dimensional. A problem having more than one variable requires multidimensional optimization. Optimization becomes increasingly difficult as the number of dimensions increases. Many multidimensional optimization approaches generalize to a series of one-dimensional approaches. |
| Static<br>Or<br>Dynamic | Dynamic optimization means that the output is a function of time, while static means that the output is independent of time. In driving, the shortest route isn't necessarily the fastest route. Finding the fastest route is a dynamic problem whose solution depends on the time of day, the weather, accidents, and so on. The static problem is difficult to solve for the best solution, but the added dimension of time increases the challenge of solving the dynamic problem. |

| | |
|---|---|
| Discrete<br>Or<br>Continuous | Discrete variables have only a finite number of possible values, whereas continuous variables have an infinite number of possible values. If we are deciding in what order to attack a series of tasks on a list, discrete optimization is employed. However, if we are trying to find the minimum value of $f(x)$ on a number line, it is more appropriate to view the problem as continuous. |
| Constrained<br>Or<br>Unconstrained | Variables often have limits or constraints. Constrained optimization incorporates variable equalities and inequalities into the cost function. Unconstrained optimization allows the variables to take any value. A constrained variable often converts into an unconstrained variable through a transformation of variables. Most numerical optimization routines work best with unconstrained variables. Consider the simple constrained example of minimizing $f(x)$ over the interval $-1 \leq x \leq 1$. The variable converts x into an unconstrained variable u by letting $x = \sin(u)$ and minimizing $f(\sin(u))$ for any value of u. |
| Random<br>Or<br>Minimum seeking | Some algorithms try to minimize the cost by starting from an initial set of variable values. These minimum seekers easily get stuck in local minima but tend to be fast. They are the traditional optimization algorithms and are generally based on calculus methods. Moving from one variable set to another is based on some determinant sequence of steps. On the other hand, random methods use some probabilistic calculations to find variable sets. They tend to be slower but have greater success at finding the global minimum. |

### 3.2.3. Classical optimization techniques

The classical methods of optimization are useful in finding the optimum solution of continuous and differentiable functions. These methods are analytical and make use of the techniques of differential calculus in locating the optimum points. Since some of the practical problems involve objective functions that are not continuous and/or differentiable, the classical optimization techniques have limited scope in practical applications. The features of the classical optimization techniques as described by Kumar (2005) are as followed:

- The classical optimization techniques are useful in finding the optimum solution or unconstrained maxima or minima of continuous and differentiable functions.
- These are analytical methods and make use of differential calculus in locating the optimum solution.
- The classical methods have limited scope in practical applications as some of them involve objective functions which are not continuous and/or differentiable.
- Yet, the study of these classical techniques of optimization form a basis for developing most of the numerical techniques that have evolved into advanced techniques more suitable to today’s practical problems.
- These methods assume that the function is differentiable twice with respect to the design variables and the derivatives are continuous.
- Three main types of problems can be handled by the classical optimization techniques:
    - Single variable functions.
    - Multivariable functions with no constraints.
    - Multivariable functions with both equality and inequality constraints. In problems with equality constraints the Lagrange multiplier method can be used. If the problem has inequality constraints, the Kuhn-Tucker conditions can be used to identify the optimum solution.

- These methods lead to a set of nonlinear simultaneous equations that may be difficult to solve.

The common difficulties with most of the classical optimization techniques are:

- Convergence to an optimal solution depends on the chosen initial solution.
- Most algorithms tend to get stuck to a suboptimal solution.
- An algorithm efficient in solving one search and optimization problem may be not efficient in solving a different problem.
- Algorithms are not efficient in handling problems having discrete variables.

### 3.2.4. Heuristic optimization techniques

Whereas exact solution methods are guaranteed to find the optimal solution (if one exists), heuristic methods sometimes find optimal solutions, but more often find simply “good” solutions. Heuristic methods typically require far less time and/or space than exact methods. The heuristics specify how to make a decision given a particular situation; heuristics are rules for deciding which action to take. The main motivation for heuristic techniques:

- To deal with local optima and not get trapped in them
- To allow optimization for systems, where the design variables are not only continuous, but discrete or integer.

### 3.2.5. Discrete optimization techniques

Several algorithms for discrete optimization problems were developed, among them branch and bound method, penalty function approach, rounding–off, cutting plane, simulated annealing, genetic algorithms, neural networks, and Lagrangian relaxation methods. Some of the methods for discrete variable optimization use the structure of the problem to speed up the search for the discrete solution. The branch and bound method, simulated annealing, and genetic algorithm are the most used methods, (Mahfouz, 1999).

## 3.3. Genetic algorithm

### 3.3.1. Introduction to Genetic Algorithms

Several strategies have been developed to solve optimization problem, (Kolisch et al, 1999). The last quarter of the 20th century has witnessed the introduction and rise of optimization techniques of natural origin: such as Genetic algorithm (GA), (Toklu 2002). It is an adaptive method, which may be used to solve search and optimization problems. Genetic algorithms are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome like data structure and apply recombination operators to these structures so as to preserve critical information. Genetic Algorithms are often viewed as function optimizers, (Lin et al 2008).

The genetic algorithm is a method for solving optimization problems that is based on natural selection, the process that drives biological evolution. The genetic algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individuals at random from the current population to be parents and uses them produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution. You can apply the genetic algorithm to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear.

### 3.3.2. Historical Background

The genetic algorithm (GA) is an optimization and search technique based on the principles of genetic and natural selection. The theory of natural selection by Charles Darwin forms the base of the genetic algorithm. Over several generations, biological organisms evolve based on the principles of natural selection "survival of the fittest" to reach certain remarkable tasks. A GA allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the "fitness" (i.e. minimize the cost function). The method was developed by John Holland (1975) over the course of 1960s and 1970s and finally popularized by one of his students, David Goldberg, who was able to solve a difficult problem involving the control of gas pipeline transmission for his dissertation, (Mahfouz, 1999, Haupt, 2004).

GA handles a population of possible solutions. Each solution is represented through a chromosome, which is just an abstract representation. Coding all the possible solutions into a chromosome is the first part, but certainly not the most straightforward one of a Genetic Algorithm. A set of reproduction operators has to be determined, too. Genetic algorithms differ from conventional optimization techniques in following ways: (Sivanandam et al, 2008).

1. GAs operate with coded versions of the problem parameters rather that parameters themselves i.e. GA works with the coding of solution set and not with the solution itself.
2. Almost all conventional optimization techniques search from a single point but GA always operate on a whole population of points (strings) i.e. GA uses population of solutions rather than a single solution for searching. It improves the chance of reaching the global optimum and also helps in avoiding local stationary point.
3. GAs use fitness function for evaluation rather that derivatives. As a result, they can be applied to any kind of continuous or discrete optimization problem. The key point to be performed here is to identify and specify a meaningful decoding function.
4. GAs use probabilistic transition operates while conventional methods for continuous optimization apply deterministic transition operates i.e. GA does not use deterministic rules.

### 3.3.3. Components of a basic genetic algorithm

The GA begins like any other optimization algorithm, by defining the optimization variables, the cost function, and the cost. It ends like other optimization algorithms too, by testing for convergence. In between, however, this algorithm is quite different. A path through the components of the GA is shown in the flowchart figure (3.4).

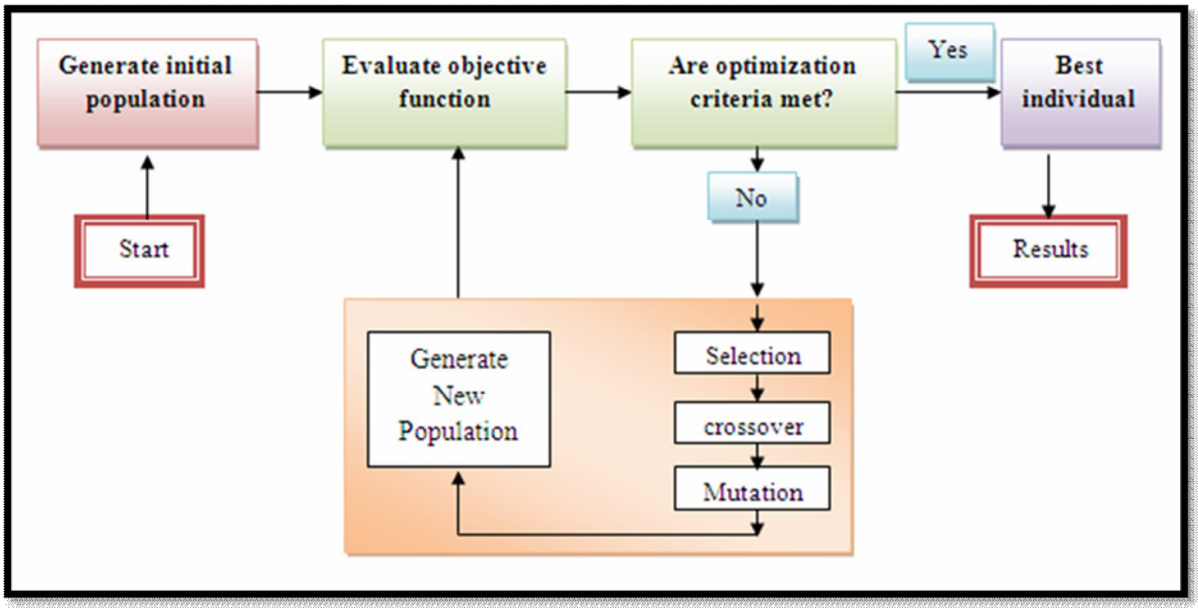

Generate initial population
Evaluate objective function
Are optimization criteria met?
Yes
Best individual
Start
No
Results
Generate New Population
Selection
crossover
Mutation

*Figure (3.4): Flowchart of GA*

Kamarainen et al (2000) described a simple version of genetic algorithm, and formulated it as follows:

1. The initial generation is created, i.e. a random integer is given to each gene of each chromosome of the population.

2. Create the next generation:

   ○ Initialize the new generation by duplicating the previous generation.

   ○ Find the best chromosome, i.e. the chromosome with the best evaluation value, of the previous generation.

   ○ Divide the population into pairs of chromosomes. Let l be the index of a pair. Repeat the next steps for all pairs l.

      ▪ Select randomly a chromosome from the previous generation and a crossover point. The crossover point lies between two consecutive genes.

      ▪ Crossover: Swap the values of genes after the crossover point between the randomly selected chromosome and the best chromosome with a probability called a crossover rate. Replace the pair l with the two new offspring chromosomes.

- Mutation: Swap two randomly selected genes of a chromosome with a probability called mutation rate. Apply the mutation to the both chromosomes in the pair l.

○ Replace the last chromosome of the population with the best chromosome of the previous generation. This is called elitism and it guarantees that the next generation will contain a chromosome, which is better than or equal to the best chromosome in the previous generation.

3. If the termination condition is met, then stop. Otherwise, go to step 2.

The GAs are implemented in a computer simulation in which a population of abstract representation (called chromosomes or the genotype of the genome) of candidate solution (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions. Solutions are represented in binary as strings of 0s and 1s. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to from a now population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

### 3.3.4. Terminologies of Genetic Algorithm

#### 3.3.4.1. Fitness Function

The fitness function is the function you want to optimize. For standard optimization algorithms, this is known as the objective function.

#### 3.3.4.2. ّIndividuals

An individual is any point to which you can apply the fitness function. The value of the fitness an individual is its score; an individual is a single solution. A chromosome is a set of parameters which define a proposed solution to the problem that the genetic algorithm is trying to solve. The chromosome is often represented as a simple string.

#### 3.3.4.3. Populations and Generations

A population is an array of individuals. At each iteration, the genetic algorithm performs a series of computations on the current population called parents to produce anew population called children. Each successive population is called a new generation. Typically, the algorithm is more likely to select parents that have better fitness values.

#### 3.3.4.4. Encoding

A chromosome is subdivided into genes. A gene is the GAs representation of a single factor for a control factor. The process of representing the solution in the form of a string that conveys the necessary information is called Encoding. Each gene controls a particular characteristic of the individual; similarly, each bit in the string represents a characteristic of the solution.

1010110011010

*Figure (3.5): The GA chromosome*

### 3.3.5. Encoding Methods

#### 3.3.5.1. Binary Encoding

Most common methods of encoding chromosomes are strings of 1s and 0s and each position in the chromosome represents a particular characteristic of the problem.

| | |
|---|---|
| Chromosome A | 1001110101011101 |
| Chromosome B | 0101011101010110 |

*Figure (3.6): Binary Encoding*

The binary GA works with bits where *N bits* of one solution = *N gene x Nvar*

Where: *N gene*: Number of bits for each variable in the problem. It is problem dependent

*N var:* is the number of variables in the fitness function

The population has N soultions, so the population matrix size = N bit * N pop

Chromosome = 1111001011 0111000101…. 1000110101

#### 3.3.5.2. Permutation Encoding (Real number encoding)

Every chromosome is a string of numbers; it is useful in ordering problems. In permutation encoding, every chromosome is a string of integer/real values, which represents number in a sequence. Permutation encoding is only useful for ordering problems. Even for this problems for some types of crossover and mutation corrections must be made to leave the chromosome consistent

| | |
|---|---|
| Chromosome A | 153264798 |
| Chromosome B | 856723149 |

*Figure (3.7): Permutation Encoding*

#### 3.3.5.3. Value Encoding

In value encoding, every chromosome is a string of some values. Values can be anything connected to problem, form numbers, real numbers or chars to some complicated objects. It is used in problems where complicated values are used where binary encoding would not suffice.

| | |
|---|---|
| Chromosome A | ABDJEIFJDHDIERJFDLDFLFEGT |
| Chromosome B | (back), (back), (right), (forward), (left) |

*Figure (3.8): Value Encoding*

### 3.3.6. Operators of Genetic Algorithm

#### 3.3.6.1. Selection

During each successive generation, a proportion of the existing is selected to breed a new generation. They are selected according to their fitness. The better the chromosomes are, the more chances to be selected they have.

Deciding how many chromosomes to keep is somewhat arbitrary. Letting only a few chromosomes survive to the next generation limits the available genes in the offspring. Keeping too many chromosomes allows bad performers a chance to contribute their traits to the next generation. Often 50% in the natural selection process are kept.

### 3.3.6.2. Crossover

Crossover is the process of taking two parent solutions and producing from them a child. After the selection process, the population is enriched with better individuals makes clones of good strings but does not creak new ones. Crossover operator is applied to the mating pool that it creates a better offspring.

There are many ways how to do crossover; single point crossover, (Haupt, 2004), two point crossover, multipoint crossover, uniform crossover, (sivanadam et al, 2008). Single point crossover is shown in figure (3.8).

| | |
|---|---|
| Parent 1 | 10110 010 |
| Parent 2 | *10101 111* |
| Child 1 | 10110 *111* |
| Child2 | *10101* 010 |

*Figure (3.9): Single Crossover*

### 3.3.6.3. Mutation

It is the process of randomly changing the genes of the individual parents. It operates independently on each individual. Mutation prevents the algorithm to be trapped in a local minimum. There are many different forms of mutation for the different kinds of representation, i.e. flipping, interchanging, and reversing, (Sivanadam and Deepa 2008) a simple can consist in inverting the value of each gene as shown in figure (3.9).

| | |
|---|---|
| Offspring | 11 0 11 00 1 00 11 0 11 0 |
| Mutated offspring | 11 0 1 0 00 1 00 1 0 0 11 0 |

*Figure (3.10): GA Mutation*

#### 3.3.6.4. Penalty method

Penalty method is that where a constrained optimization problem is transformed to an unconstrained optimization problem by associating a penalty with all constraint violations. This penalty is included in the objective function evaluation.

## 3.4. Techniques for project scheduling

Early techniques to solve constrained-resource problems used mathematical models, such as integer programming, branch-and-bound, linear programming or dynamic programming. Heuristic rules may perform well over a variety of problems and are widely used in practical cases because of their simple format and efficiency in application; nevertheless, optimal solutions are not guaranteed. Mathematical models guarantee optimal solutions on small-scale problems. The disadvantages of mathematical models are that it is difficult to create general mathematical models and extensive computational effort is required on larger problems (Leu and Yang, 1999).

Early approaches solved simplified versions of the problem exactly, but researchers quickly realized that real problems are too large and complicated for any exact solution. For example, decision trees were used to enumerate every possible choice. Heuristic methods were then devised to find good solutions, or to find simply feasible solutions for the really difficult problems. Most research now consists of designing better heuristics for specific instances of scheduling problems. However, heuristic solutions are typically limited to a specific set of constraints or problem formulation, and devising new heuristics is difficult at best (Wall, 1996).

Genetic algorithms have been applied to scheduling problems in the last few decades. Toklu (2002) made a survey about RCPSP made in 1998 , he stated that there is no mention of them, and in a similar survey made in 1999 they take a minimal space. Basic information about the application of GAs to project scheduling can be found in 1996. Then GAs had been applied to construction problems for crash schedule determination or time-cost trade-off scheduling problems, after that, improved heuristics are proposed and used with GAs for applications to resource allocation and leveling problems. Multicriteria optimization and GA application to repetitive activities are subjects recently treated, (Toklu, 2002).

## 3.5. Previous studies

Achieving optimal scheduling of a project attracted the attention of many researchers, who employed various techniques including linear programming, dynamic programming, expert systems, neural networks, and genetic algorithms (Senouci, 2004).

Leu and Yang (1999) presented a multi-criteria computational optimal scheduling model that integrated a time/cost trade off model, resource-limited model, and resource leveling model. A searching technique using Genetic algorithms was adopted in the model. The method produced suboptimal solutions and did not seek total cost minimization.

El Rays and Moselhi (2001) presented an automated model that utilized dynamic programming formulation and incorporated a scheduling algorithm and an interruption algorithm, so as to automate the generation of interruption during scheduling. The model is valid for repetitive construction projects only.

Toklu (2002) introduced a computer program to apply GA in order to schedule resource-constrained projects to minimize makespan. The strategy of solving a problem took a lot of time; a project consists of only eight activities was solved after one thousand generations, and took six minutes in a computer having Pentium III-500 processor.

Kim (2005) presented an optimal algorithm for resource allocation model, which was to be implemented into a framework for the development for an integrated model. There was no evidence that the model was capable of producing optimum scheduling.

Liu et al (2005) developed a new crossover operator to avoid producing illegal chromosomes comparing to traditional crossover methods used in scheduling optimization.

Shih and Liu (2006) proposed a new scheduling model aiming for the minimization of total project cost, including resource usage and idle cost. But the model depends on providing external resources to provide planners with additional flexibility in arranging schedules and provide planners with diversity of resource supply, and also the model needs in-depth discussion on issues of determining the limit of internal resource to minimize operational costs.

Mendes et al (2006) presented a genetic algorithm for the resource-constrained project scheduling problem. Priorities of activities are defined for the project not an exact schedule of the project.

Senouci and Derham (2008) presented a genetic algorithm based multi-objectives optimization model for the scheduling of construction projects. But the model is valid for linear construction projects only, which characterized by repetitive activities.

Deiranlou and jolai (2009) used a new crossover based on combination of order crossover and partially mapped crossover.

Okada et al (2010) proposed a random key-based genetic algorithm approach to solve optimization scheduling problems, but they did not show computational experiments to show the effectiveness of the proposed approach, and it was recommended in their publication to do this in the future research directions.

## 3.6. Concluding remarks

Judging from the state of past research, it is necessary to develop a more efficient algorithm to obtain good and near optimal solutions for practical construction projects in which computational efficiency and multiple objective issues are of concern, it is also noted that In the past few years, there has been a boom in applying Genetic Algorithms (GA) to solving the optimization problems.

# CHAPTER (4)

## Methodology of scheduling problem optimization

## 4.1.Introduction

One of the central challenges of computer science is to get a computer to do what needs to be done, without telling it how to do it. Genetic algorithm addresses this challenge by providing a method for automatically creating a working computer program from a high level problem statement of the problem. Genetic algorithms achieves this goal of automatic programming (also sometimes called program synthesis or program induction) by genetically breeding a population of computer programs using the principles of Darwinian natural selection and biologically inspired operations. The steps of genetic optimization programming as defined by Sivanandam and Deepa, (2008), are:

- Preparatory steps
- Executional steps

The human user communicates the high-level statement of the problem to the genetic algorithm programming system by performing certain well-defined preparatory steps. The five major preparatory steps for the basic version of genetic programming require the human user to specify:

1. The set of terminals (e.g., the fixed variables of the problem, zero argument functions, and random constants) for each branch of the to-be-evolved program,
2. The set of primitive functions for each branch of the to-be-evolved program,
3. The fitness measure (for explicitly or implicitly measuring the fitness of individuals in the population),
4. Certain parameters for controlling the run, and the termination criterion and method for designating the result of the run.

The preparatory steps are the problem-specific and domain-specific steps that are performed by the human user prior to launching a run of the problem-solving method.

The executional steps of genetic programming (that is, the flowchart of genetic programming) are as follows:

1. Randomly create an initial population (generation 0) of individual computer programs composed of the available functions and terminals.
2. Iteratively perform the following sub-steps (called a generation) on the population until the termination criterion is satisfied.
   - Execute each program in the population and ascertain its fitness (explicitly or implicitly) using the problem's fitness measure.
   - Select one or two individual programs from the population with a probability based on fitness (with reselection allowed).
   - Create new individual programs for the population by applying the following genetic operations with specified probabilities: Reproduction, crossover, and mutation.
3. After the termination criterion is satisfied, the single best program in the population produced during the run (the best-so-far individual) is harvested and designated as the result of the run. If the run is successful, the result may be a solution (or approximate solution) to the problem.

At this chapter, a formulation of optimization problem is presented for resource-constrained project scheduling problem (RCPSP) including preparatory steps and executional steps.

## 4.2. Modeling of construction project scheduling

A project such as a construction project consists of a network of activities and a node in network corresponds to an activity. Each activity in a project has a corresponding duration and also needs certain amount of resources such as labor material to execute itself with them. Activity duration is usually measured in integral increments of time called planning units. The normal duration of an activity refers to the time required to complete that activity under normal circumstances. Figure (4.1) shows a typical network of a construction project, and table (4.1) shows the resource usage of the project activities.

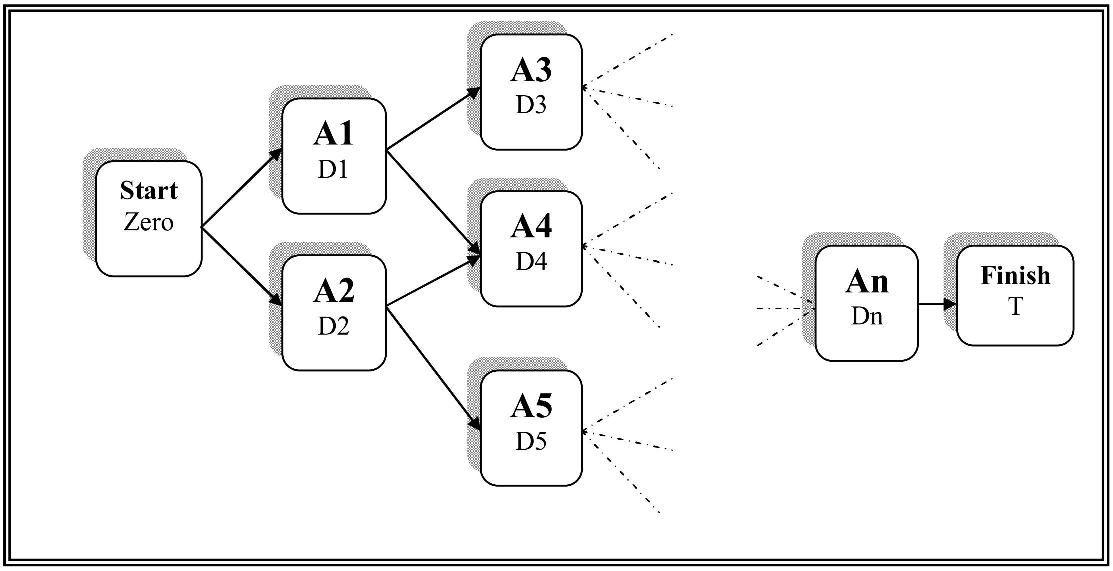


*Figure (4.1): A network of a construction project*

*Table (4.1): Resources usage of typical construction project activities*

| **Activity** | **Precedence** | **Duration** | **Resource Usage** | | | |
|---|---|---|---|---|---|---|
| | | | $R_1$ | $R_2$ | ….. | $R_n$ |
| **$A_1$** | -- | $D_1$ | 1 | -- | | -- |
| **$A_2$** | -- | $D_2$ | 2 | -- | | -- |
| **$A_3$** | $A_1$ | $D_3$ | -- | 1 | | -- |
| **$A_4$** | $A_1$,$A_2$ | $D_4$ | -- | 1 | | -- |
| **$A_5$** | $A_2$ | $D_5$ | 1 | -- | | -- |

| | | | | | | |
|---|---|---|---|---|---|---|
| .<br>.<br>. | | | | | | |
| **$A_n$** | $A_x$,$A_y$ | $D_n$ | -- | -- | | 1 |

Figure (4.1) and Table (4.1) show that any project network consists of the following componenets:

Activities

$$A = \llbracket[A]\rrbracket_1, A_2, A_3, \dots, A_n]$$

Where $A_1, A_2, A_3, \dots, A_n$ are the project activities

$n$ : the number of activities.

Durations

$$D = \llbracket[D]\rrbracket_1, D_2, D_3, \dots, D_n]$$

Where $D_1, D_2, D_3, \dots, D_n$ are the correspondent activity duration

$D_i$ : the duration of activity i.

Resources

$$Res = \llbracket[R]\rrbracket_1, R_2, R_3, \dots, R_r]$$

Where $R_1, R_2, R_3, \dots, R_r$ are the project resources

$r$ : the number of resources.

Precedence constraints

The construction project activities relate to each other through predefined precedence relations. The precedence relations between activities in Table (4.2) are modeled as shown in Table (4.3).

*Table (4.2): Precedence relations and resources usage of construction project activities*

| Activity | Precedence | Duration | Resource Usage | | |
|---|---|---|---|---|---|
| | | | $R_1$ | $R_2$ | $R_3$ |
| **$A_1$** | -- | $D_1$ | 1 | -- | -- |
| **$A_2$** | -- | $D_2$ | 2 | -- | -- |
| **$A_3$** | $A_1$ | $D_3$ | -- | 1 | -- |
| **$A_4$** | $A_1$,$A_2$ | $D_4$ | -- | 1 | -- |
| **$A_5$** | $A_2$ | $D_5$ | 1 | -- | -- |
| **$A_6$** | $A_3$ | $D_6$ | -- | -- | 2 |
| **$A_7$** | $A_4$,$A_5$ | $D_7$ | -- | -- | 1 |

*Table (4.3): Modeling the precedence relations of construction project activities*

| | | The succeeding activities | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | **$A_1$** | **$A_2$** | **$A_3$** | **$A_4$** | **$A_5$** | **$A_6$** | **$A_7$** |
| The preceding activities | **$A_1$** | a | a | a | a | a | a | a |
| | **$A_2$** | a | a | a | a | a | a | a |
| | **$A_3$** | y | a | a | a | a | a | a |
| | **$A_4$** | y | y | a | a | a | a | a |
| | **$A_5$** | a | y | a | a | a | a | a |
| | **$A_6$** | a | a | y | a | a | a | a |

This means that activity 5 depends on the finish of activity 2, but do not depend on activity

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **$A_7$** | a | a | a | y | y | a | a | 1, 3, 4, 5, 6 nor 7. |

$a_{(i,j)}$: the activity i do not depends on activity j.

$y_{(i,j)}$: the activity i depends on the finish of activity j

Resources requirements of activities

The resources requirements of the activities in Table (4.2) are modeled as shown in Table (4.4).

*Table (4.4): Modeling the resource requirements of construction project activities*

**The project activities**

| | | **$A_1$** | **$A_2$** | **$A_3$** | **$A_4$** | **$A_5$** | **$A_6$** | **$A_7$** |
|---|---|---|---|---|---|---|---|---|
| **The project resources** | **$R_1$** | 1 | 2 | 0 | 0 | 1 | 0 | 0 |
| | **$R_2$** | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| | **$R_3$** | 0 | 0 | 0 | 0 | 0 | 2 | 1 |

This means that activity 6

needs 2 units of resource 3

## 4.3. Formulation of RCPSP as a GA optimization problem

The first step in the formulation of an optimization problem is to identify the components of the problem, namely, the fixed parameters, design variables, objective function, and constraints. The design variables are determined in such a way that the value of the objective function becomes minimum.

### 4.3.1. Fixed parameters

The fixed parameters are assumed to be:

- Number of the activities in a project,
- Activities durations,
- Relation between activities (precedence relationships),
- Number of resources,
- Maximum units available of each resource, and
- Activity requirements of each resource.

Matrices of activities precedence and resources were used to introduce the project network to the proposed GA model.

### 4.3.2. Design variables

As design variables should be independent of each other, the design variables of resource-constrained project scheduling problem (RCPSP) are defined in this study as the starting dates of the project activities which define the schedule of the project where the first activity has a starting date of zero. This means that the number of the design variables depends on the project and equals to the number of the activities in the project.

$$\text{Project schedule} = S$$

$$\text{Where } S = S_1, S_2, S_3, \dots, S_n$$

$$S_i \text{ is the starting date of the activity i.}$$

### 4.3.3. Objective function

The key question addressed in RCPSP is to determine the starting date of each activity such that precedence and resource constraints are satisfied while achieving some objectives like shortest project duration or minimum resource investment (Xu et al, 2007). The objective function is to find the best schedule that gives the minimum makespan (total project duration (T)).

$$Minimize \ \ T$$

$$T = maximum\ (S_i + D_i)$$

Where *T* is the total Duration of the project.

$S_i$ is the starting date of activity i

$D_i$ is the duration of activity i

### 4.3.4. Constraints

In this research two types of constraints are considered; sequential constraints and resource constraints.

<u>Sequential constraints</u>

Sequential constrains are where an activity can not start until all its predecessors have finished.

$$S_j > S_i$$

for each activity i precede activity j.

<u>Resource constraints</u>

Processing every activity in the project requires a predefined amount of resources $\mathbf{R_j}$(▯), which are available in limited quantities in every time unit, where at time (t) the total consumption of the j'th resource cannot exceed the available amount of this resource ($\mathbf{R_j j^{\dagger} a}$ (▯))

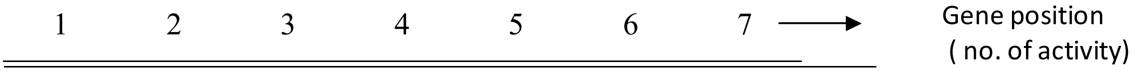
$$R_j(t) \leq R_j^a(t)$$

### 4.3.5. Chromosome Representation

A chromosome is subdivided into genes. A gene is the G.A representation of a single design variable. In this study each gene represents a starting date of an activity.

Value encoding, not binary encoding, is used in this study to express each gene, because it is suitable where complicated values, such as real numbers are used, (Sivanandam et al, 2008).

Figure (4.2) shows the chromosome representation used in this study.



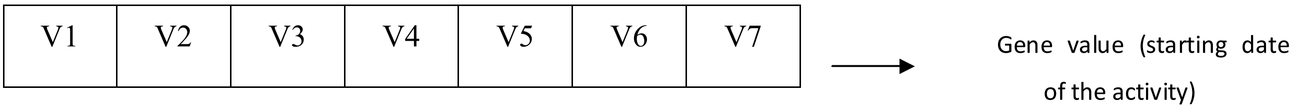


*Figure (4.2): Chromosome structure for resource-constrained project*

### 4.3.6. Determination of the fitness function

The fitness indicates how good the solution is. In the current study, good solutions have minimum project duration, so that the solution is more fit if it has a lesser duration. This is reflected in the fitness function which is decided to be the reciprocal of the total duration of the project. The fitness function of this study is defined as follows:

$$\mathbf{f_{(i)}} = \frac{\mathbf{1}}{\mathbf{T_{(i)}}}$$

Where $f_{(i)}$ is the fitness value of chromosome i , and $T_{(i)}$ is the total project duration defined by chromosome i.

## 4.4. Application of Genetic Algorithm to RCPSP

Figure (4.3) shows a flowchart of the developed GA optimization model, according to the following steps:

### 4.4.1. Creation of the population

#### 4.4.1.1. Modeling steps

Application of genetic algorithm technique as an optimization tool normally comprises three stages: coding the design variables into chromosomes that represent the problem, evaluating the fitness of each solution string, and applying genetic operators to produce the next generation of a solution.

Applying GA technique to the RCPSP in this research passed through the following steps:

- Create the first population to be completely random solutions (chromosomes) that do not necessary satisfy the time and resources constraints. This needed a very long time to produce one solution – not necessarily optimum- that satisfies time and resource constraints for each activity in the solution, so that a modification in the solution approach was required.

- To enhance the convergence toward the best solution, a new approach was followed: that is to create solutions that satisfy time constraints, and to apply a penalty function for the solution that does not satisfy resource constraints. It was noted that the solution still slow and it was difficult and time consuming to get the desired aim of an optimum solution.
- Finally, a new approach was developed to generate the first population, which is called "*Feasible Solutions Developer Operator (FSD Operator)*" that enables the user to create completely feasible solutions that satisfy both time and resource constraints which helps in getting a quick convergence toward the best solution during GA stages.

#### 4.4.1.2. Feasible solutions developer operator (FSD Operator)

The idea behind the Feasible Solutions Developer Operator (FSD Operator) is to inject a feasibility test during the internal steps of creating a solution; creation of one solution consists of many internal steps, the FSD Operator examines the feasibility of a solution at each step, if it is not satisfied, repeating this step is applied until a feasible solution is obtained.

This means that only feasible solutions comprise the initial population, infeasible solutions are discarded before entering the pool of initial solutions; this accelerates the convergence toward the best solution during the following GA operators. Figure (4.3) shows the idea of the FSD Operator.

To conclude, the first population in the GA must be randomly produced. In this research random but legal and feasible solutions are produced without losing the GA feature of searching global maximum or minimum. For example, any schedule must start with an activity that has no preceding activities; it is time consuming when creating random solutions that start illogically with an activity that has preceding activities.

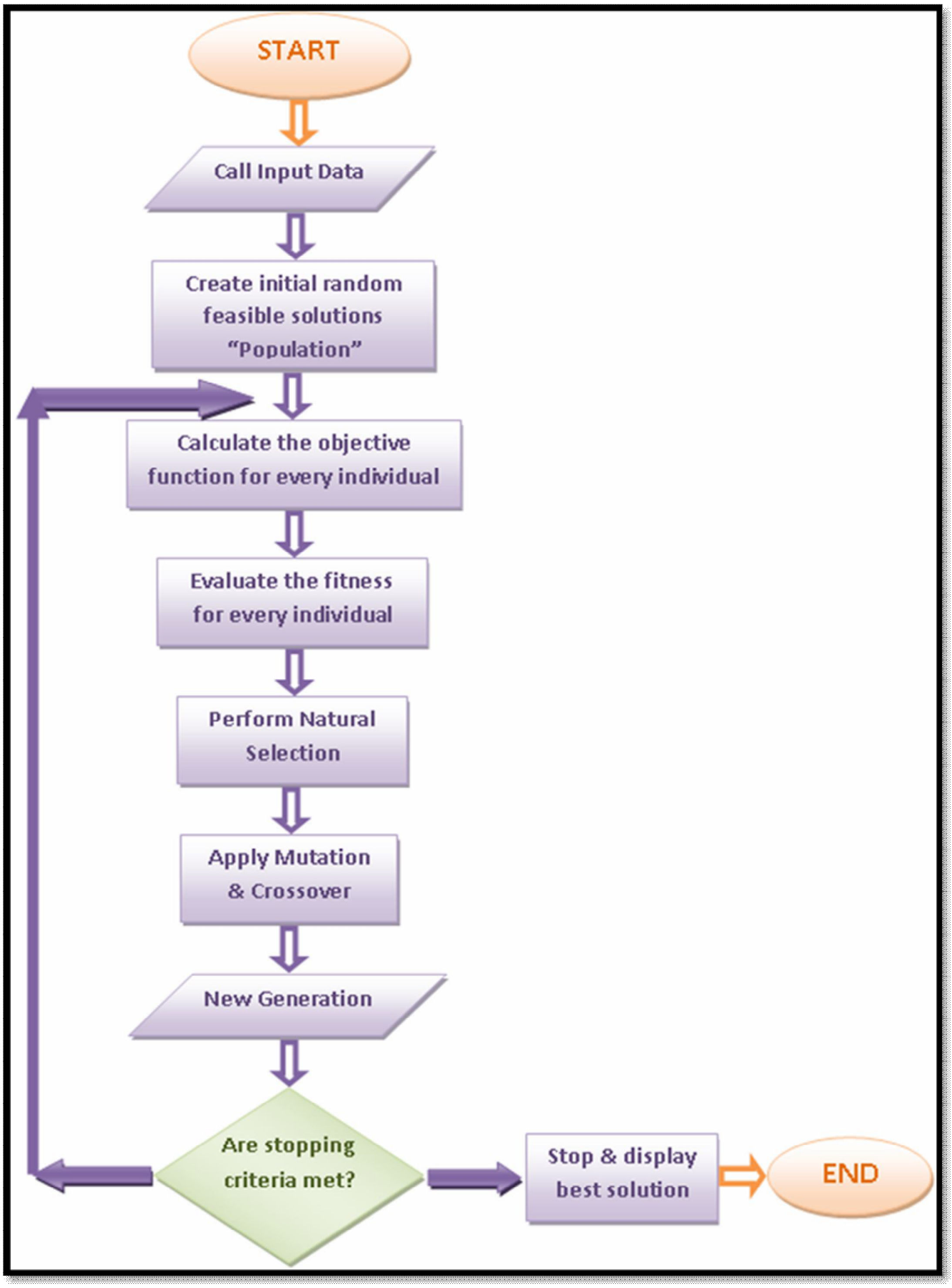

START
Call Input Data
Create initial random feasible solutions "Population"
Calculate the objective function for every individual
Evaluate the fitness for every individual
Perform Natural Selection
Apply Mutation & Crossover
New Generation
Are stopping criteria met?
Stop & display best solution
END

*Figure (4.3): Flowchart of the developed GA optimization model*

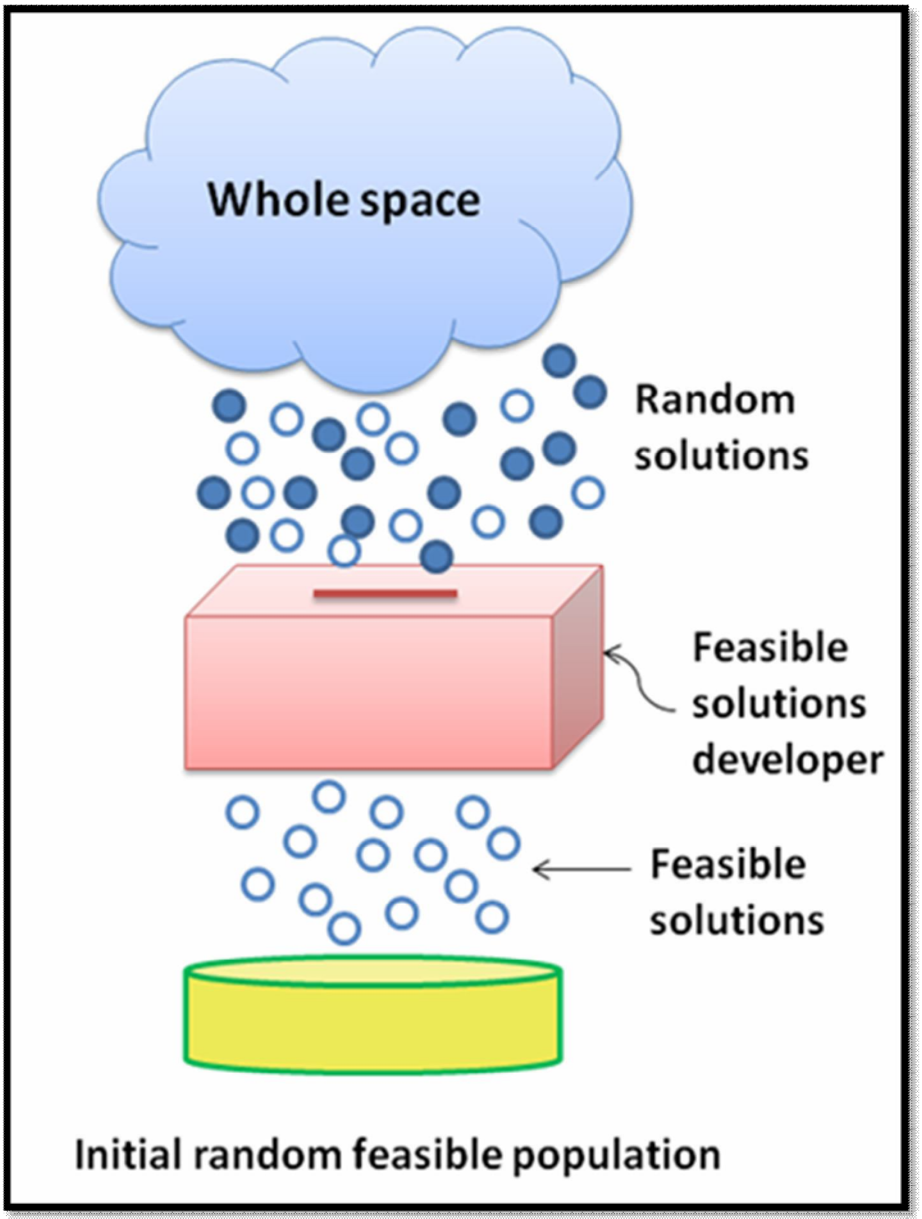

Whole space
Random solutions
Feasible solutions developer
Feasible solutions
Initial random feasible population

*Figure (4.4): Creation of population by Feasible Solutions Developer Operator*

### 4.4.2. Natural selection

Deciding how many chromosomes to keep is somewhat arbitrary. Letting only a few chromosomes survive to the next generation limits the available genes in the offspring. Keeping too many chromosomes allows bad performers a chance to contribute their traits to the next generation. Often 50% in the natural selection process are kept. In this study, 50 % of the chromosomes in the natural selection are kept.

### 4.4.3. Selection strategy

The selection strategy used in this study is Roulette Wheel, where chromosomes are selected to survive to the next generation according to their fitness. The better chromosomes are, the more chances to be selected they have. According to Roulette Wheel principle, the selection probability for a chromosome k is proportional to the ratio of $^{fk}/_{\Sigma fi}$ where fk is the fitness value of chromosome k.

### 4.4.4. Crossover operator

Because there generally exist some precedence relationships among activities in scheduling problem, several crossover operators intensively used in sequencing problem will generate illegal chromosomes (illegal solutions), to overcome this drawback, many researchers developed new crossover operators that suit their scheduling problem formulation, and suit the type of their chromosomes. A new crossover operator developed in this study, the procedure of the new crossover operator is performed as follows (As shown in Figure (4.5)):

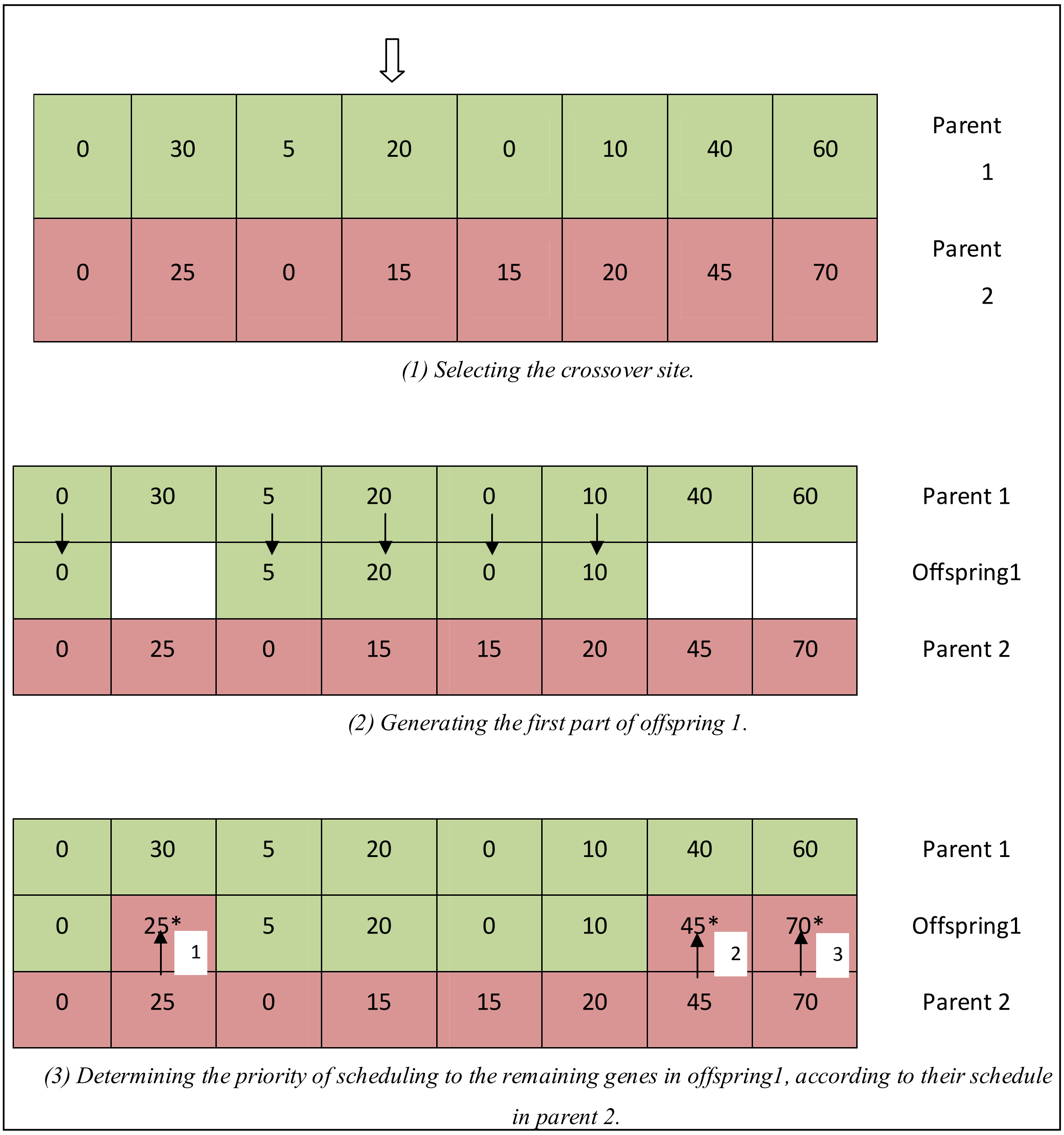
*Step 1:* Select one value of a gene from the first parent at random.

Step 2: Select all the genes in the first chromosome that has a gene value smaller than the selected value and transfer them to offspring 1 .

*Step 3:* Take the values of the empty genes in the offspring 1 from the corresponding genes in parent 2 considering the following:

- Begin with the genes that have smaller values among the remaining genes.
- Before transferring the value of the selected gene from parent 2 to offspring 1, check the time constraint (precedence constraint) and the resource constraint, if there is no violation, transfer the value, otherwise generate a value that satisfy the constraints, at this situation, the priority of schedule, rather than the value of the genes, was transferred from parent 2.

With the same steps, the second offspring is produced from the same parents.



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 30 | 5 | 20 | 0 | 10 | 40 | 60 | Parent 1 |
| 0 | 25 | 0 | 15 | 15 | 20 | 45 | 70 | Parent 2 |

*(1) Selecting the crossover site.*

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 30 | 5 | 20 | 0 | 10 | 40 | 60 | Parent 1 |
| 0 | | 5 | 20 | 0 | 10 | | | Offspring1 |
| 0 | 25 | 0 | 15 | 15 | 20 | 45 | 70 | Parent 2 |

*(2) Generating the first part of offspring 1.*

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 30 | 5 | 20 | 0 | 10 | 40 | 60 | Parent 1 |
| 0 | 25* | 5 | 20 | 0 | 10 | 45* | 70* | Offspring1 |
| 0 | 25 | 0 | 15 | 15 | 20 | 45 | 70 | Parent 2 |

*(3) Determining the priority of scheduling to the remaining genes in offspring1, according to their schedule in parent 2.*

*Figure (4.5): Crossover operator steps*

*: means that we have to transfer the value of the gene (if it forms feasible solution), or the priority of scheduling and to calculate a new value based on the transferred priority of scheduling (to form feasible solution).

### 4.4.5. Mutation

For simplicity, the mutation operator used in this research is uniform mutation. Uniform mutation replace a gene with a randomly gene. Testing the feasibility of the introduced solution is applied. If any solution does not satisfy the feasibility test, it is discarded from the next generation; instead, a random generated solution is produced as the solutions in the first population.

- A summary of how the next generation is produced using the above operators is shown figure (4.4).

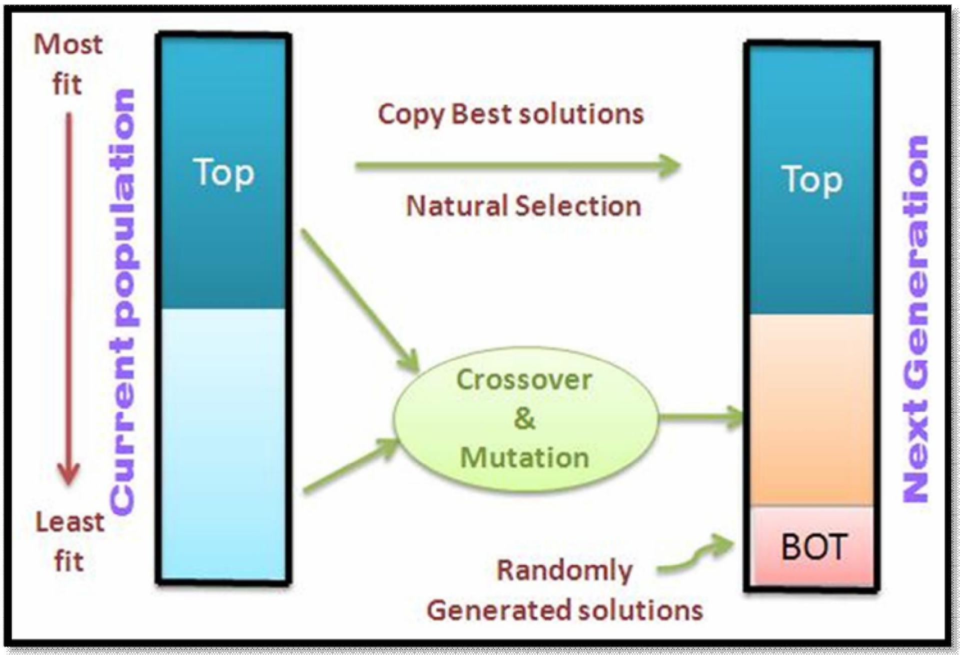


*Figure (4.6): Creation of the next generation during GA solution*

# CHAPTER (5)

## Analysis of results and discussion

## 5.1. Introduction

An implementation of the developed GA optimization model for resource-constrained project scheduling has resulted in an application program called the Construction Projects Scheduling Optimizer program “CPS Optimizer”. This program is to implement genetic algorithm to resource constrained project scheduling problems in order to further investigate the practical benefits to real world projects.

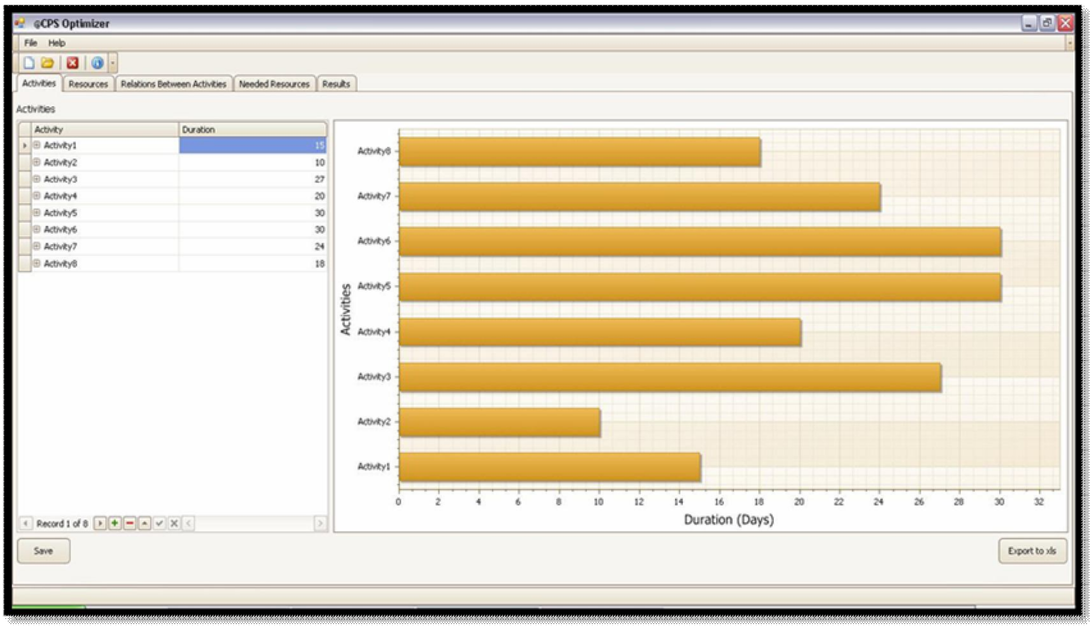
To verify and validate the model, a comparison with the solution obtained for scheduling problems solved by other researchers will be carried out through this chapter.

## 5.2. The CPS Optimizer

A flow chart of the developed GA optimization model for the resource constrained project scheduling problem is shown in figure (4.3) as explained in the chapter (4). The model is called The Construction Projects Scheduling Optimizer program (CPS Optimizer). It was initially developed within MATLAB; MATLAB is a numerical computing environment and many-generation programming language. Developed by The MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, Matlab (2009).

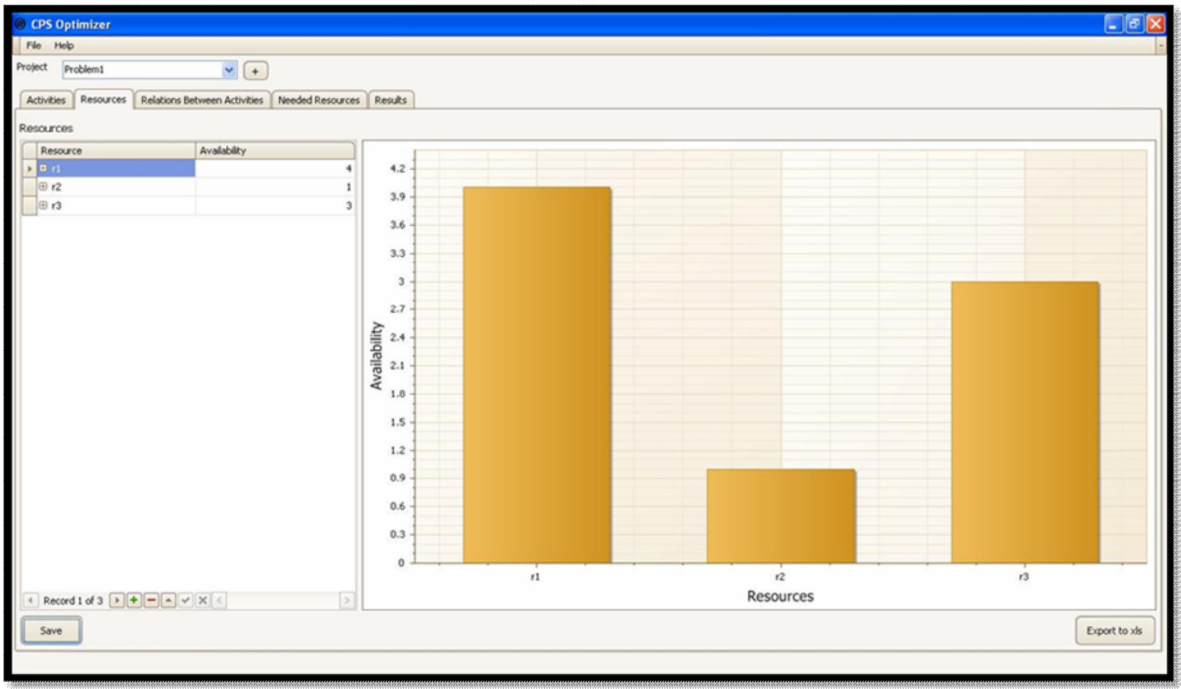
The GA algorithm was modeled by the researcher to suit the special nature of the studied problem and the type of coding used; that instead of using binary coding, a real coding was used. Later, C-Sharp programming language was used for the Graphical User Interface (GUI). The interface of the CPS Optimizer is shown in figure (5.1).

The CPS Optimizer interface contains several tabs, where the first tab – that is shown in figure (5.1) - enables the user to add a new project, the names of the activities, and the duration of each of them. A chart showing the activities and the duration of each activity is created automatically by the program. The CPS Optimizer resources input sheet is shown and activities relationships input sheet are shown in figure (5.2) and figure (5.3).



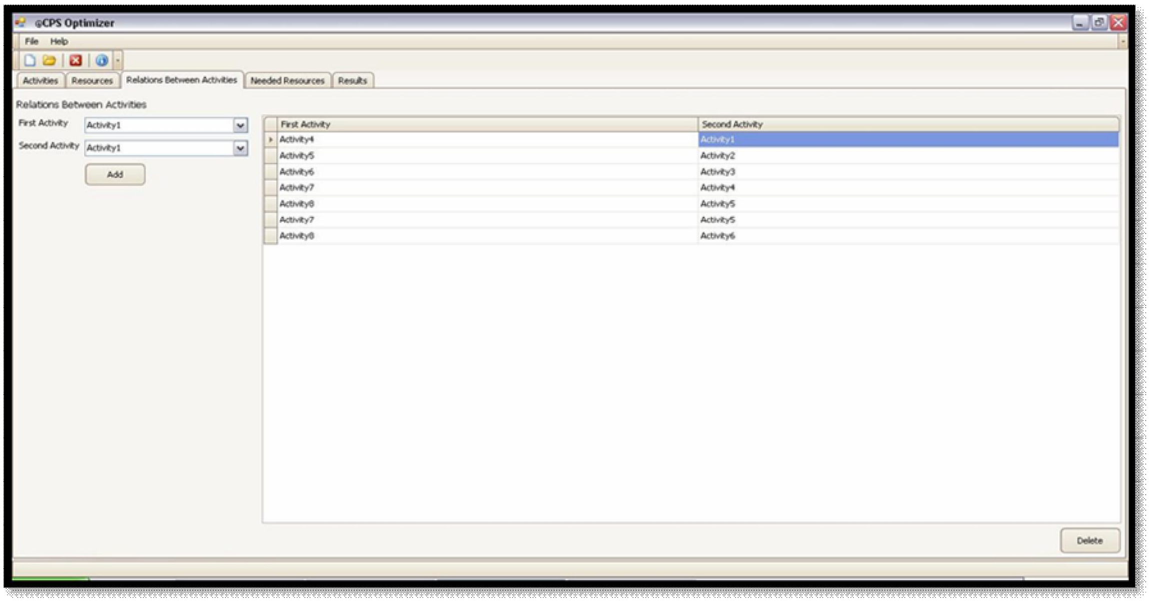
| Activity | Duration |
|---|---|
| Activity1 | 15 |
| Activity2 | 10 |
| Activity3 | 27 |
| Activity4 | 20 |
| Activity5 | 30 |
| Activity6 | 30 |
| Activity7 | 24 |
| Activity8 | 18 |

*Figure (5.1): The CPS Optimizer interface*



| Resource | Availability |
|---|---|
| r1 | 4 |
| r2 | 1 |
| r3 | 3 |

*Figure (5.2): The CPS Optimizer resources input sheet*



Relations Between Activities

First Activity: Activity1

Second Activity: Activity1

| First Activity | Second Activity |
|---|---|
| Activity4 | Activity1 |
| Activity5 | Activity2 |
| Activity6 | Activity3 |
| Activity7 | Activity4 |
| Activity8 | Activity5 |
| Activity7 | Activity5 |
| Activity8 | Activity6 |

*Figure (5.3): The CPS Optimizer activities relationships input sheet*

## 5.3. Verification and validation of the model

Model verification and validation are essential parts of the model development process if the model to be accepted and used to support decision making.

Verification is done to ensure that:

- The model is programmed correctly, and there are no syntax errors.
- The algorithms have been implemented properly.
- The model does not contain errors, oversights, or bugs.

No computational model will ever be fully verified guaranteeing 100% error free implementation. A high degree of statistical certainty is all that can be realized for any model as more cases as tested.

The developed model was verified by more tests performed, errors were identified, and corrections were made to encountered errors. This took a long period of time, and a huge effort by the researcher to get a high degree of certainty about the model verification.

Validation ensures that the model meets its intended requirements in terms of the methods employed and the results obtained. The ultimate goal of the model validation is to make the model useful in the sense that the model addresses the right problem, provides accurate information about the system being modeled, and finally to make the model accurately used.

In order to validate the developed model, two problems are applied and the results are analyzed and compared with the results of previous studies and with the feature of “resource leveling” in the commercial construction management programs such as Microsoft project (2002) and Primavera (P5).

*Resource leveling feature* is a feature based on application of some heuristics rules to CPM schedule so that (a) the resource usage does not exceed prescribed limits and (b) the resource usage is as uniform as possible. Resource leveling can be used to some extent in the sample problem by rearranging the CPM schedule so that the resource limits are not exceeded.

## 5.4. Problem 1(Eight activities project)

The following problem was introduced by Toklu (2002). Toklu applied genetic algorithms to construction scheduling with or without resource, applying it to eight activities project problem. In the current research, the CPS Optimizer is used to solve the problem, and the result is compared with Toklus model results.

### 5.4.1. Description of the problem

A simplified breakdown of the works necessary for construction of a bridge is considered to yield the activities shown in table (1), where the precedence relations and durations are presented.

*Table (5.1): The precedence relations and durations of the first problem*

| Activity | Description | Precedence | Duration | Resource Usage | | |
|---|---|---|---|---|---|---|
| | | | | R1 | R2 | R3 |
| **A1** | Excavation at pier A | -- | 15 | 1 | -- | -- |
| **A2** | Pier works A | A1 | 20 | -- | 1 | -- |
| **B1** | Excavation at pier B | -- | 10 | 1 | -- | -- |
| **B2** | Pier works B | B1 | 30 | -- | 1 | -- |

| | | | | | | |
|---|---|---|---|---|---|---|
| **C1** | Excavation at pier C | -- | 27 | 1 | -- | -- |
| **C2** | Pier works C | C1 | 30 | -- | 1 | -- |
| **D1** | Deck AB | A2,B2 | 24 | -- | -- | 1 |
| **D2** | Deck BC | B2,C2 | 18 | -- | -- | 1 |

It is assumed that there are ***three resources*** used in this project, the excavation team (R1), the pier team (R2) and the deck team (R3), and that there is ***only one of each team***.

### 5.4.2. CPM Solution of the problem

The CPM solution for this problem (supposing unlimited resources) is as shown in figure (5.4). It indicates that the makespan (Total duration) for the given project is T = 75 days. The reader can return back to Figure (2.3) for the meaning of each number in the shown schedule.

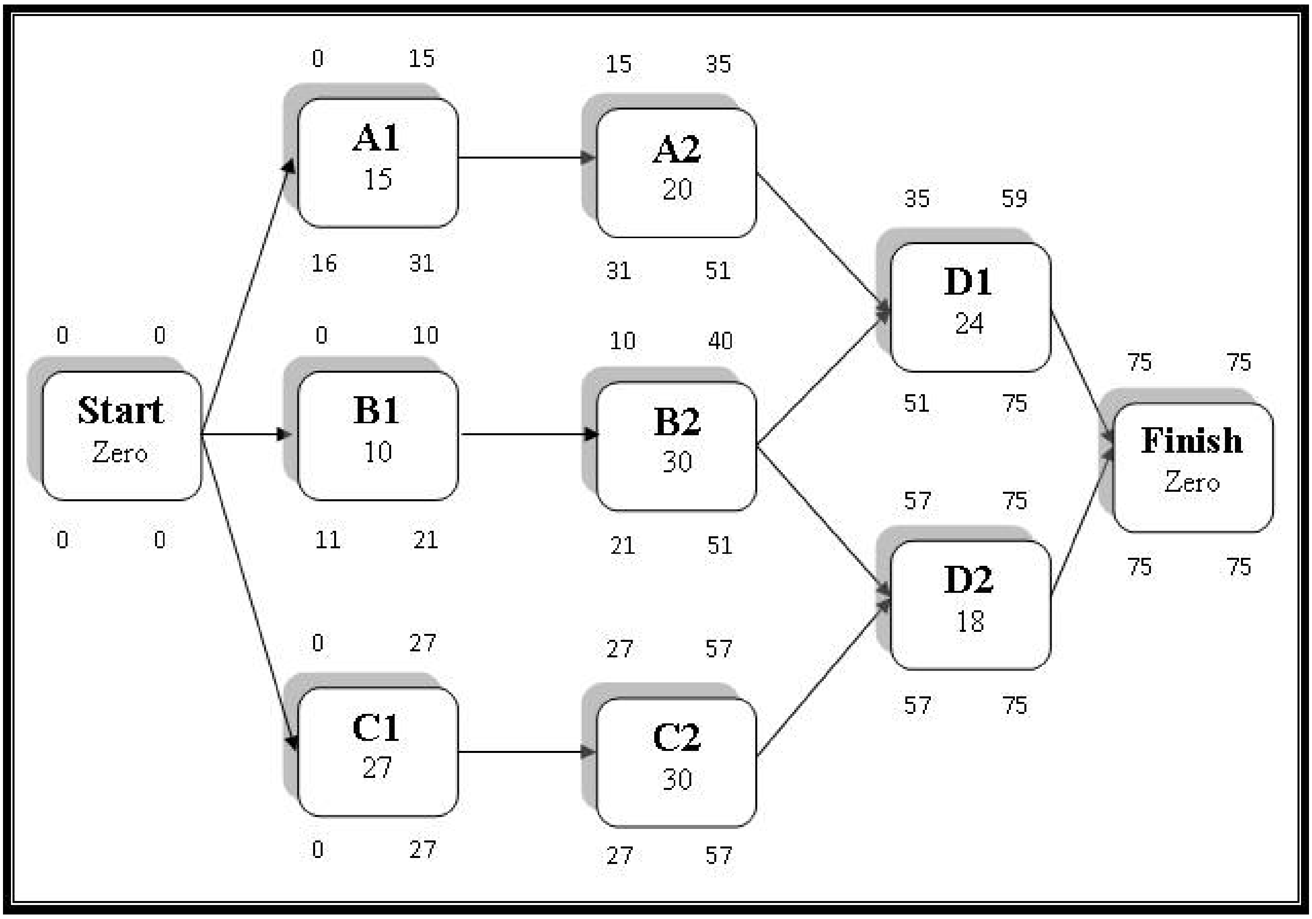
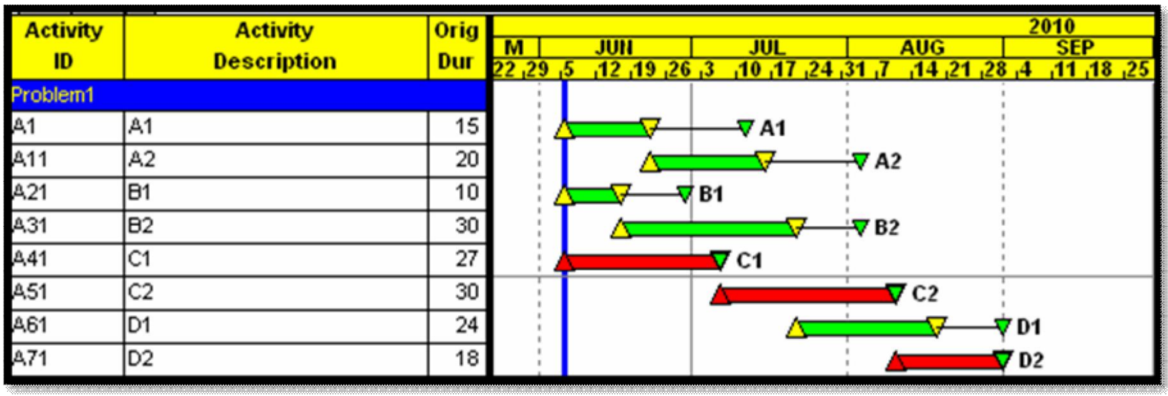


*Figure (5.4): CPM schedule of problem 1*

### 5.4.3. Primavera Solution of the problem

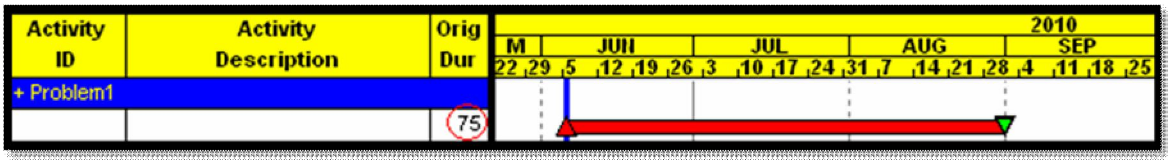
The eight activities project is also scheduled using Primavera program yielding a schedule with Total duration equals to 75 days with the existence of resources overrun as shown in figures (5.5), (5.6) and (5.7).

Figure (5.5) shows the Gantt chart of problem 1 using Primavera, where the total duration of the project equals to 75 days before using resource leveling feature in Primavera as shown in the summary graph in figure (5.6). Figure (5.7) shows the resources profile of problem 1 before leveling the resources, so there is an overrun in the resources usage during the project duration.



| Activity ID | Activity Description | Orig Dur |
|---|---|---|
| Problem1 | | |
| A1 | A1 | 15 |
| A11 | A2 | 20 |
| A21 | B1 | 10 |
| A31 | B2 | 30 |
| A41 | C1 | 27 |
| A51 | C2 | 30 |
| A61 | D1 | 24 |
| A71 | D2 | 18 |

*Figure (5.5): Gantt chart of the problem 1 using primavera*



| Activity ID | Activity Description | Orig Dur |
|---|---|---|
| + Problem1 | | |
| | | 75 |

*Figure (5.6): Total duration of the problem1 before resource leveling using primavera*

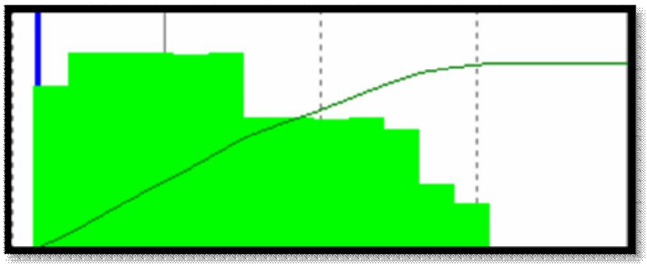
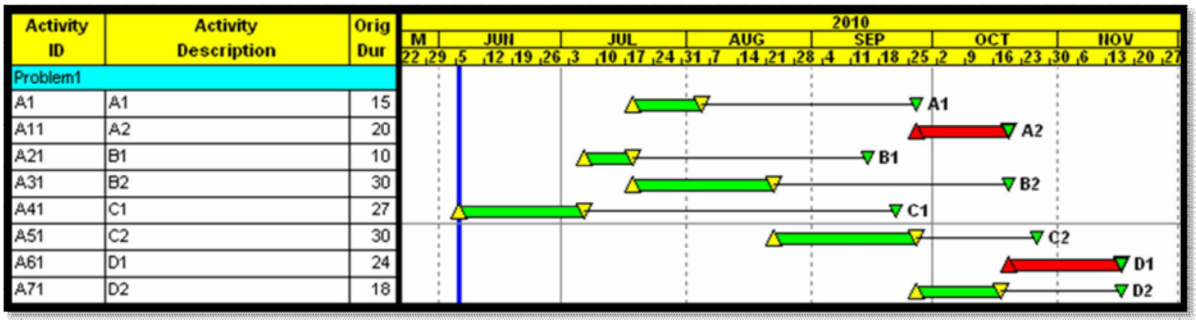


*Figure (5.7): The resources profile of the problem1 before leveling using primavera*

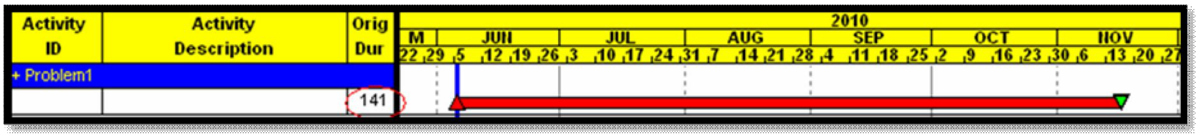
Running resource leveling with default options using primavera program yields a schedule with T = 141 days, which suggested a schedule of activities that C1→B1→A1 in order to overcome the resource overrun, as shown in figures (5.8), (5.9) and (5.10).

Figure (5.8) shows the schedule of problem 1 after resource leveling using primavera, where many activities moved comparing with the schedule in figure (5.5), for example the activities C1,B1 and A1 have the same starting date in figure (5.5), but there is an overrun in the resources because each of them needs resource 1 (R1) and there is only one unit available of R1, so that Primavera solved this resource constrained by moving activities B1 and A1, and the schedule is as shown in figure (5.8) and has a total duration of the project equals to 141 days as shown in figure(5.9). Figure (5.10) shows the resources profile of the problem 1 after leveling the resources, where the resources distributed over the duration of the project.



| Activity ID | Activity Description | Orig Dur |
|---|---|---|
| Problem1 | | |
| A1 | A1 | 15 |
| A11 | A2 | 20 |
| A21 | B1 | 10 |
| A31 | B2 | 30 |
| A41 | C1 | 27 |
| A51 | C2 | 30 |
| A61 | D1 | 24 |
| A71 | D2 | 18 |

*Figure (5.8): Schedule of the problem 1 after resource leveling using primavera*



| Activity ID | Activity Description | Orig Dur |
|---|---|---|
| + Problem1 | | |
| | | 141 |

*Figure (5.9): Total duration of problem 1 after resource leveling using primavera*

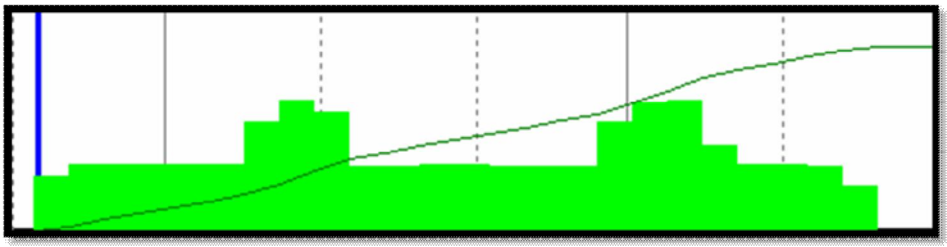
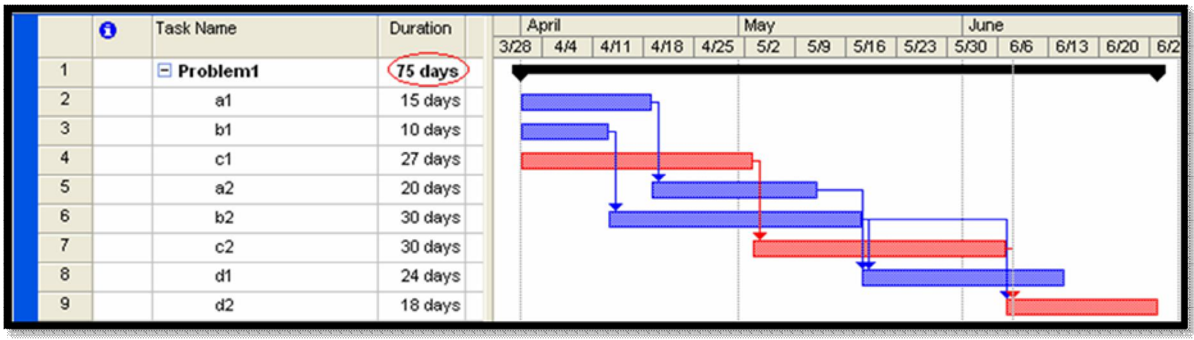


*Figure (5.10): The resources profile of problem 1 after leveling using primavera*

### 5.4.4. Microsoft Project Solution of the problem

Problem 1 is scheduled using MS Project program yielding a schedule with Total duration equals to 75 days with the existence of resources overrun as shown in figures (5.11), and (5.12).

Figure (5.11) shows the Gantt chart of problem 1 using MS Project, where the total duration of the project equals to 75 days before using resource leveling feature in MS.



| | ⓘ | Task Name | Duration |
|---|---|---|---|
| 1 | | ⊟ Problem1 | 75 days |
| 2 | | a1 | 15 days |
| 3 | | b1 | 10 days |
| 4 | | c1 | 27 days |
| 5 | | a2 | 20 days |
| 6 | | b2 | 30 days |
| 7 | | c2 | 30 days |
| 8 | | d1 | 24 days |
| 9 | | d2 | 18 days |

*Figure (5.11): Gantt chart of problem 1 using MS Project before leveling*

As shown in figure (5.12), the red color shows the resource overrun, it is noted that there is overrun over the duration of the project, at the beginning of the project the overrun is one unit, and there is an overrun two units at the end of the project.

Running resource leveling with default options using Microsoft project yields a schedule with T = 114 days, which suggested a schedule of activities that B1→C1→A1 in order to overcome the resource over allocation as shown in figure (5.11) and (5.12).

Figure (5.13) shows the schedule of problem 1 after resource leveling using MS Project, where many activities moved comparing with the schedule in figure (5.11), for example the activities C1,B1 and A1 have the same starting date in figure (5.11), but there is an overrun in the resources as shown in figure (5.12) because each of them needs resource 1 (R1) and there is only one unit available of R1, so that MS Project solved this resource constrained by moving activities A1 and C1, and the schedule is as shown in figure (5.13) and has a total duration of the project equals to 114  where the activities A1,B1, and C1 have a schedule of B1→C1→A1

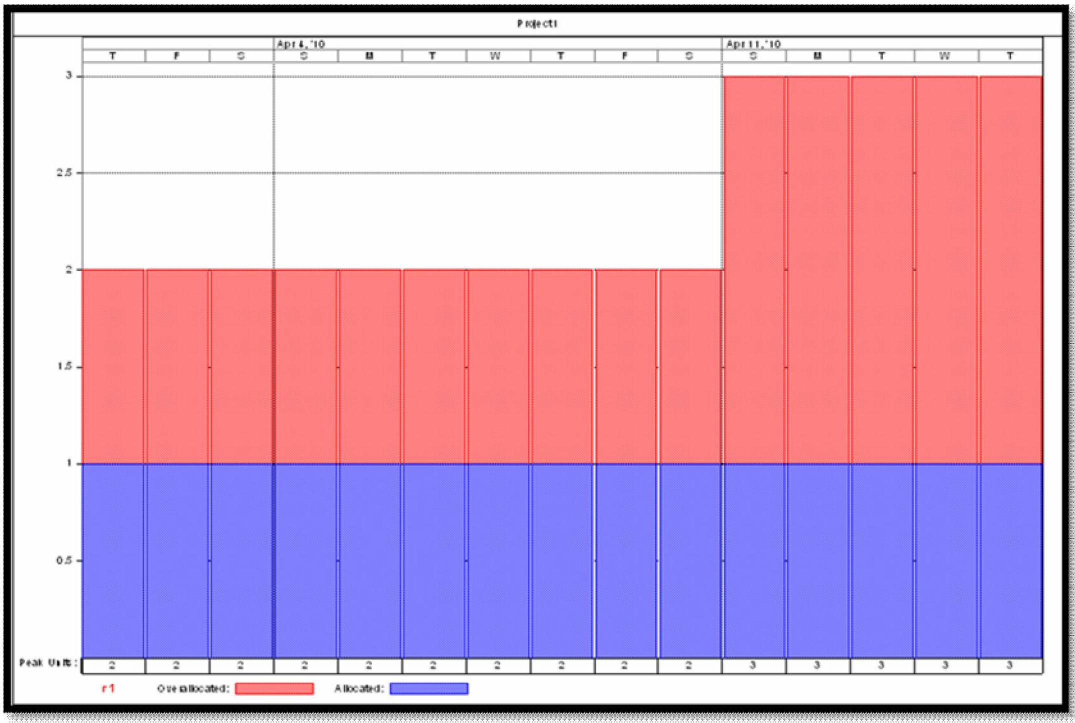
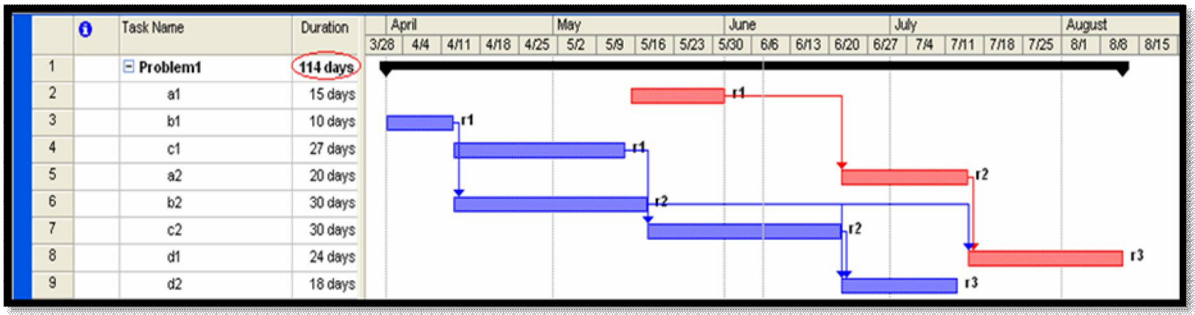


*Figure (5.12): The resources profile of problem 1 before leveling using MS Project*



| | | Task Name | Duration |
|---|---|---|---|
| 1 | | ⊟ Problem1 | 114 days |
| 2 | | a1 | 15 days |
| 3 | | b1 | 10 days |
| 4 | | c1 | 27 days |
| 5 | | a2 | 20 days |
| 6 | | b2 | 30 days |
| 7 | | c2 | 30 days |
| 8 | | d1 | 24 days |
| 9 | | d2 | 18 days |

*Figure (5.13): Gantt chart of problem 1 using MS Project after leveling*

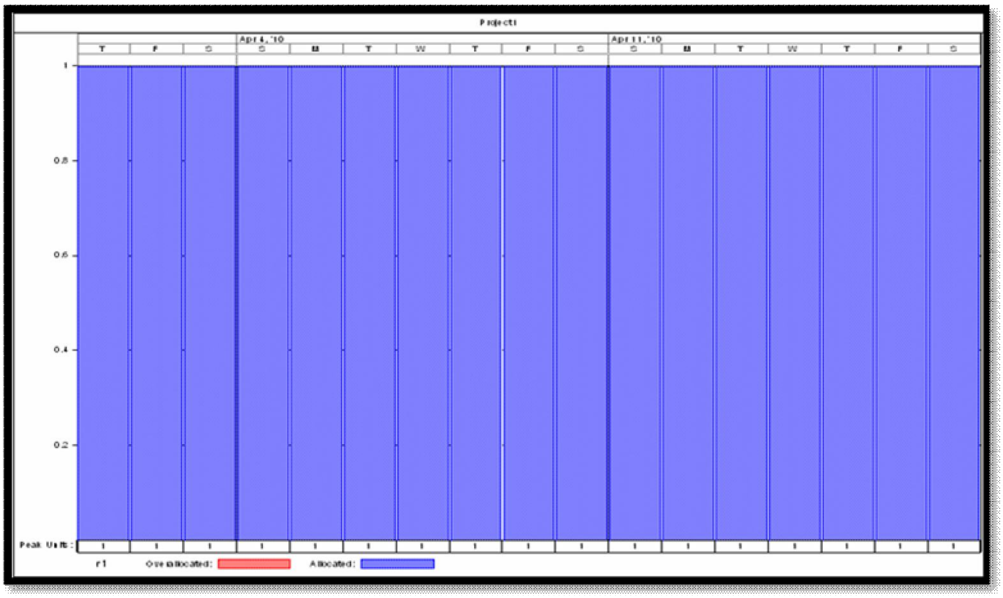


*Figure (5.14): The resources profile of problem 1 after leveling using MS Project*

As shown in figure (5.14), there is no overrun in the resources after leveling using MS Project.

### 5.4.5. CPS Optimizer Solution of problem 1

Running CPS Optimizer yields a best schedule with T = 108 days, where the schedule is B1→A1→C1, the best solution was obtained using 10 generations only. Toklu (2002) obtained the best solution using his GA approach after 1000 generations.

The starting dates of the activities are: 10, 0, 25, 40, 10, 60, 60, and 90 as noted in figure (5.15).

From the solution shown, it is noted that the starting date of the last activity equals to 90, and since this activity has duration of 18 days, the total duration of project is 108 days. The duration, starting date and ending date of each activity are shown in figure (5.16)

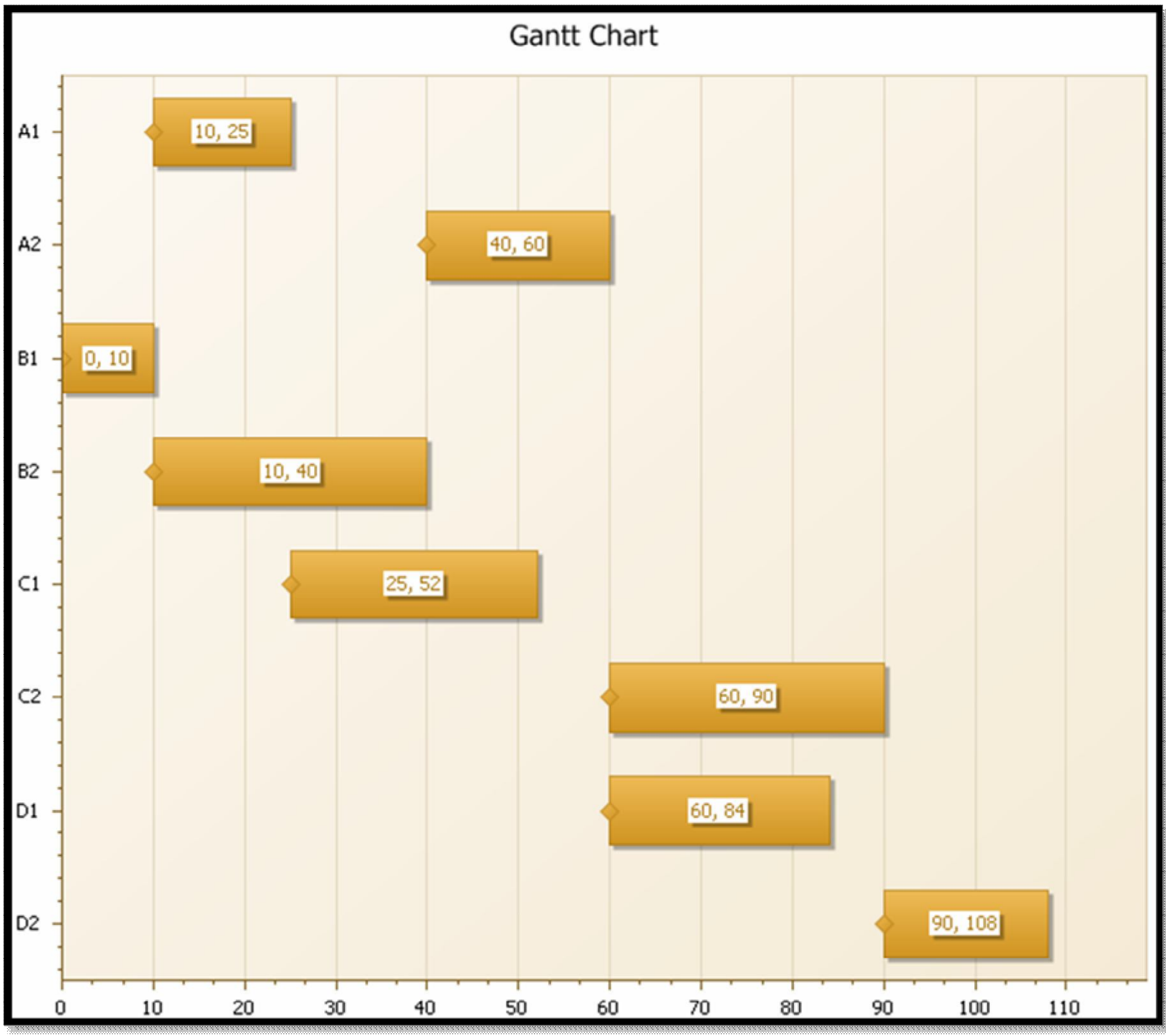
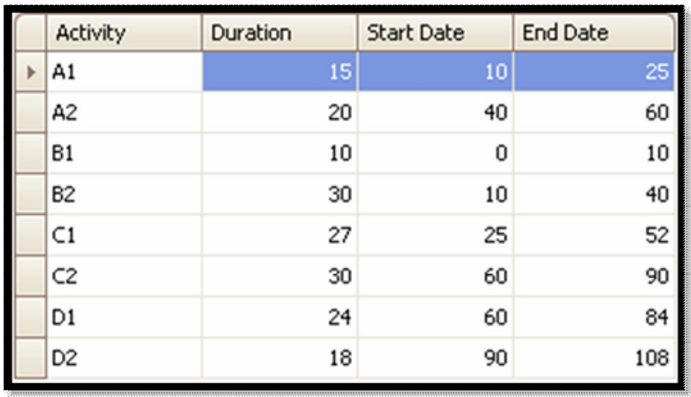


*Figure (5.15): Gantt chart of the best schedule of problem 1*



| Activity | Duration | Start Date | End Date |
|---|---|---|---|
| A1 | 15 | 10 | 25 |
| A2 | 20 | 40 | 60 |
| B1 | 10 | 0 | 10 |
| B2 | 30 | 10 | 40 |
| C1 | 27 | 25 | 52 |
| C2 | 30 | 60 | 90 |
| D1 | 24 | 60 | 84 |
| D2 | 18 | 90 | 108 |

*Figure (5.16): Starting and ending dates of the best schedule*

### 5.4.6. Comparison between the solutions

Table (5.2) shows a comparison between the different mentioned solutions, which shows that the CPS Optimizer results meet the intended goal of achieving the best schedule with the minimum efforts (10 iterations only).

*Table (5.2): Comparison between the solutions of problem 1.*

| Solution | Total duration (days) | No. of iterations | Order in Excavation | Order in Piers | Order in Deck |
|---|---|---|---|---|---|
| **CPM** | 75 | -- | Parallel | Parallel | Parallel |
| **Primavera** | 139 | -- | C1→B1→A1 | B2→C2→A2 | D2→D1 |
| **MS Project** | 114 | -- | B1→C1→A1 | B2→C2→A2 | D2→D1 |
| **Toklu (2002) solution** | 108 | 1000 | B1→A1→C1 | B2→A2→C2 | D1→D2 |
| **CPS Optimizer** | 108 | 10 | B1→A1→C1 | B2→A2→C2 | D1→D2 |

## 5.5. Problem 2 (twelve activities project)

### 5.5.1. Description of the problem

A construction project with twelve activities is considered in this problem. A simplified breakdown of the activities of problem 2 is shown in table (5.3), where the precedence relations and durations are presented.

*Table (5.3): The precedence relations and durations of the second problem*

| Activity | Precedence | Duration | Resource Usage | | | | |
|---|---|---|---|---|---|---|---|
| | | | R1 | R2 | R3 | R4 | R5 |
| **A** | -- | 11 | 1 | -- | -- | -- | -- |
| **B** | -- | 14 | 1 | -- | -- | 1 | -- |
| **C** | -- | 15 | 1 | -- | -- | -- | -- |
| **D** | A | 10 | -- | 1 | -- | -- | -- |
| **E** | A | 10 | -- | 1 | -- | -- | -- |
| **F** | B, C | 14 | -- | 1 | -- | -- | -- |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **G** | D | 13 | -- | -- | -- | 1 | -- |
| **H** | E | 12 | -- | -- | 1 | -- | -- |
| **I** | E | 7 | -- | -- | 1 | -- | -- |
| **J** | G | 8 | -- | -- | -- | 1 | -- |
| **K** | H | 8 | -- | -- | -- | 1 | -- |
| **L** | I, F | 5 | 2 | -- | 1 | -- | -- |
| **M** | J, K, L | 6 | -- | 2 | -- | 1 | 3 |

It is assumed that there are ***five resources*** used in this project, and the available units of each of them are as shown in table (5.4)

*Table (5.4): The availability of resources of problem 2*

| Resource name | Available units |
|---|---|
| **R1** | 2 |
| **R2** | 2 |
| **R3** | 1 |
| **R4** | 1 |
| **R4** | 3 |

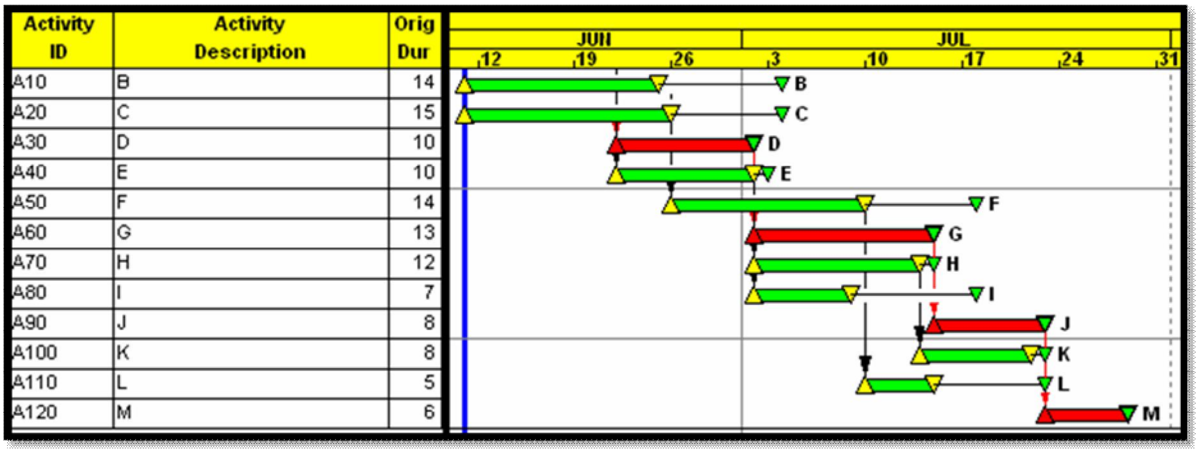
### 5.5.2. CPM Solution of the problem

The CPM solution for this problem (supposing unlimited resources) gives a makespan for the given project T = 48 days.

### 5.5.3. Primavera Solution of problem 2

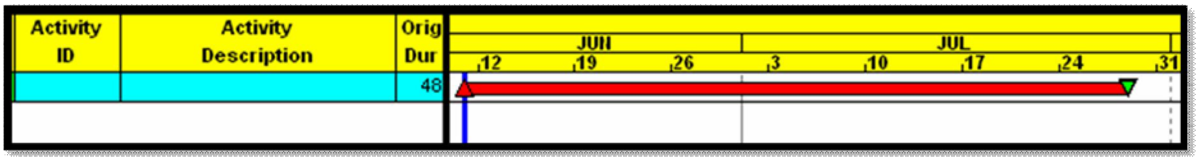
Problem 2 is scheduled using primavera program yielding a schedule with Total duration equals to 48 days with the existence of resources overrun as shown in figures (5.17), (5.18) and (5.19).

Figure (5.17) shows the Gantt chart of problem 2 using Primavera, where the total duration of the project equals to 48 days before using resource leveling feature in Primavera as shown in the summary graph in figure (5.18). Figure (5.19) shows the resources profile of problem 2 before leveling the resources, so there is an overrun in the resources usage during the project duration.



| Activity ID | Activity Description | Orig Dur |
|---|---|---|
| A10 | B | 14 |
| A20 | C | 15 |
| A30 | D | 10 |
| A40 | E | 10 |
| A50 | F | 14 |
| A60 | G | 13 |
| A70 | H | 12 |
| A80 | I | 7 |
| A90 | J | 8 |
| A100 | K | 8 |
| A110 | L | 5 |
| A120 | M | 6 |

*Figure (5.17): Gantt chart of problem 2 using primavera*



| Activity ID | Activity Description | Orig Dur |
|---|---|---|
| | | 48 |

*Figure (5.18): Total duration of problem 2 before leveling using primavera*

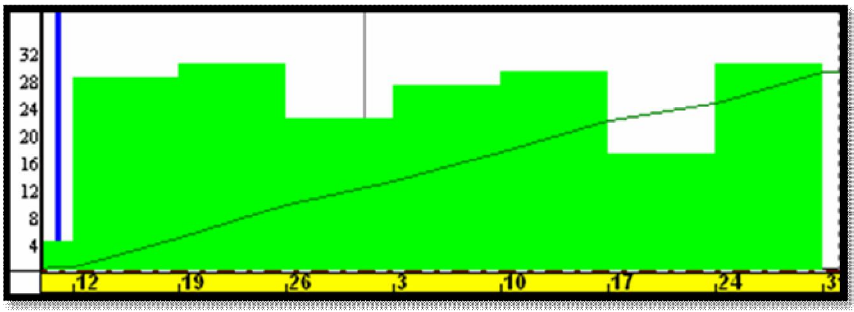
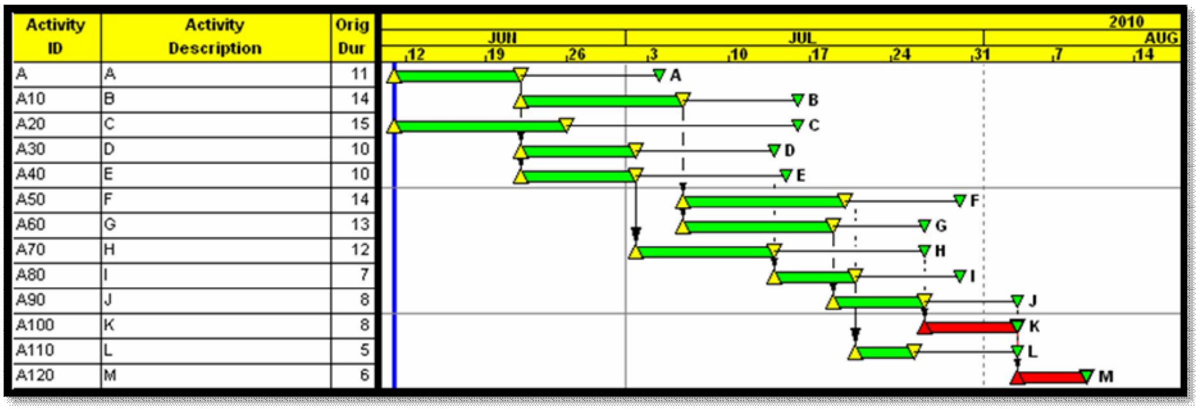


*Figure (5.19): The resources profile of problem 2 before leveling using primavera*

Running resource leveling with default options using primavera program yields a schedule with T = 60 days, which suggested a schedule of activities that A&C → B in order to overcome the resource overrun, as shown in figures (5.20), (5.21) and (5.22).

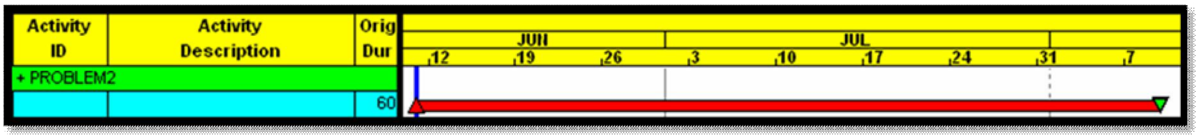
Figure (5.20) shows the schedule of problem 2 after resource leveling using primavera, where many activities moved comparing with the schedule in figure (5.17), for example the activities

C,B and A have the same starting date in figure (5.17), but there is an overrun in the resources because each of them needs resource 1 (R1) and there is only two units available of R1, so that Primavera solved this resource constrained by moving activity B, and the schedule is as shown in figure (5.20) and has a total duration of the project equals to 60 days as shown in figure(5.21). Figure (5.21) shows the resources profile of the problem 2 after leveling the resources, where the resources distributed over the duration of the project.



| Activity ID | Activity Description | Orig Dur |
|---|---|---|
| A | A | 11 |
| A10 | B | 14 |
| A20 | C | 15 |
| A30 | D | 10 |
| A40 | E | 10 |
| A50 | F | 14 |
| A60 | G | 13 |
| A70 | H | 12 |
| A80 | I | 7 |
| A90 | J | 8 |
| A100 | K | 8 |
| A110 | L | 5 |
| A120 | M | 6 |

*Figure (5.20): Schedule of problem 2 after leveling using primavera*



| Activity ID | Activity Description | Orig Dur |
|---|---|---|
| + PROBLEM2 | | |
| | | 60 |

*Figure (5.21): Total duration of problem 2 after leveling using primavera*

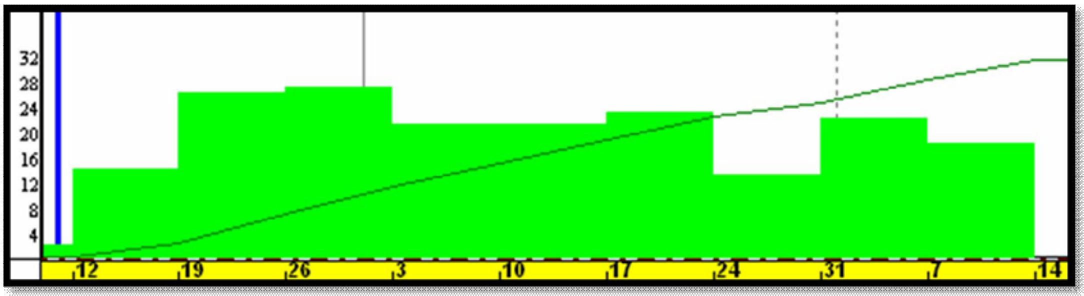
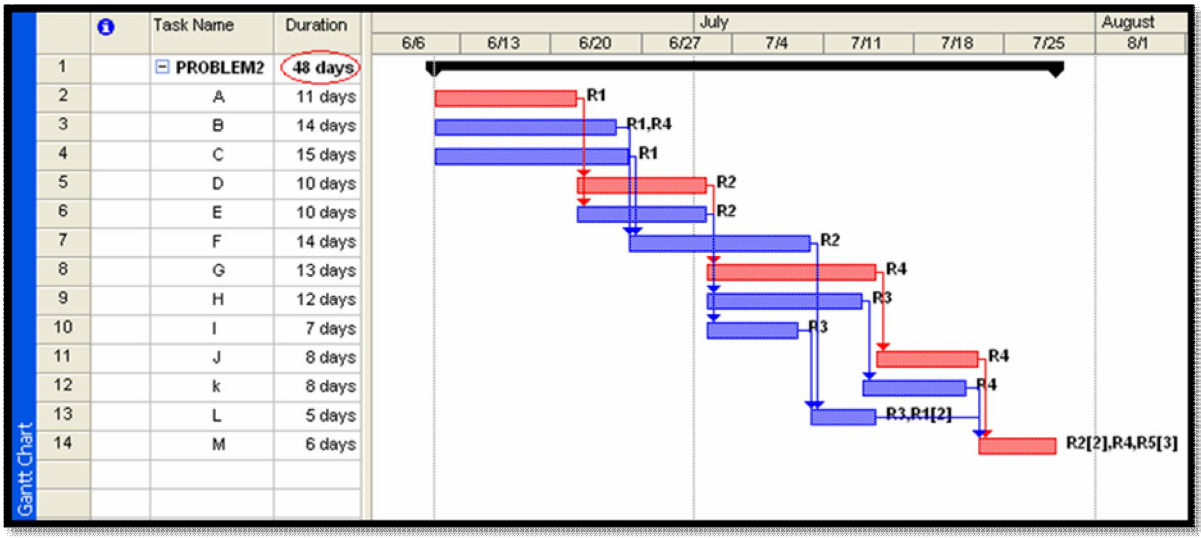


*Figure (5.22): The resources profile of problem 2 after leveling using primavera*

### 5.5.4. Microsoft Project Solution of the problem

Problem 2 is scheduled using MS Project program yielding a schedule with Total duration equals to 48 days with the existence of resources overrun as shown in figures (5.22) and (5.23).

Figure (5.23) shows the Gantt chart of problem 2 using MS Project, where the total duration of the project equals to 48 days before using resource leveling feature in MS.



| | Task Name | Duration |
|---|---|---|
| 1 | ⊟ PROBLEM2 | 48 days |
| 2 | A | 11 days |
| 3 | B | 14 days |
| 4 | C | 15 days |
| 5 | D | 10 days |
| 6 | E | 10 days |
| 7 | F | 14 days |
| 8 | G | 13 days |
| 9 | H | 12 days |
| 10 | I | 7 days |
| 11 | J | 8 days |
| 12 | k | 8 days |
| 13 | L | 5 days |
| 14 | M | 6 days |

*Figure (5.23): Gantt chart of problem 2 using MS Project before leveling*

As shown in figure (5.24), the red color shows the resource overrun, it is noted that there is overrun over most days of the project.

Running resource leveling with default options using Microsoft project yields a schedule with T = 60 days, which suggested a schedule of activities that A & C →B in order to overcome the resource over allocation as shown in figure (5.25) and (5.26).

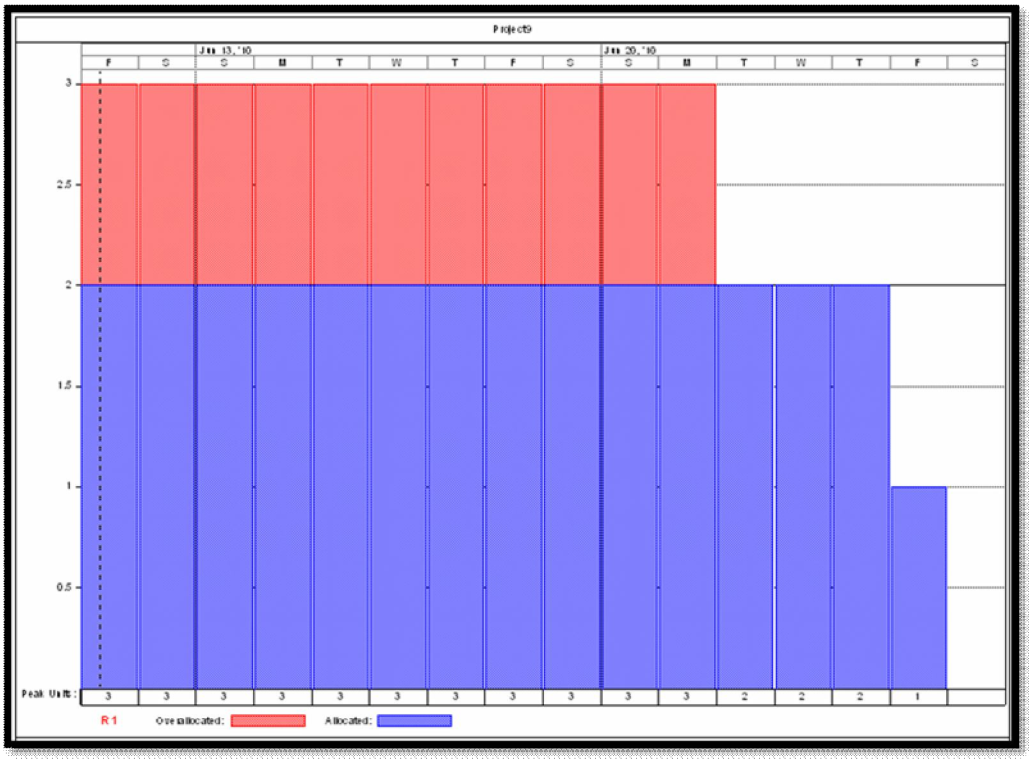
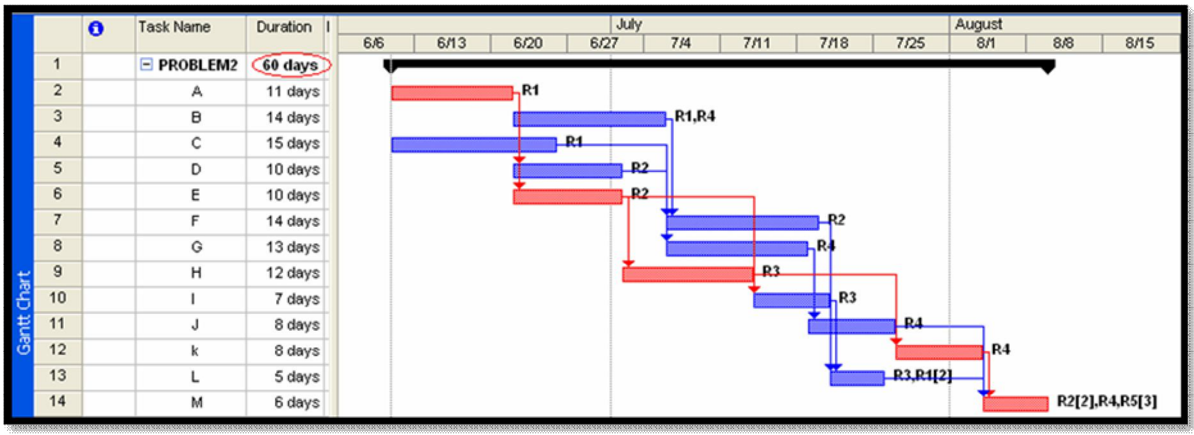


*Figure (5.24): The resources profile of problem 2 before leveling using MS Project*

Figure (5.25) shows the schedule of problem 2 after resource leveling using MS Project, where many activities moved comparing with the schedule in figure (5.23), for example the activity C,B and A have the same starting date in figure (5.23), but there is an overrun in the resources as shown in figure (5.24) because each of them needs resource 1 (R1) and there is only two units available of R1, so that MS Project solved this resource constrained by moving activity B, and the schedule is as shown in figure (5.25) and has a total duration of the project equals to 60 days where the activities A,B, and C have a schedule of A & C →B



|  | Task Name | Duration |
|---|---|---|
| 1 | ⊟ PROBLEM2 | 60 days |
| 2 | A | 11 days |
| 3 | B | 14 days |
| 4 | C | 15 days |
| 5 | D | 10 days |
| 6 | E | 10 days |
| 7 | F | 14 days |
| 8 | G | 13 days |
| 9 | H | 12 days |
| 10 | I | 7 days |
| 11 | J | 8 days |
| 12 | k | 8 days |
| 13 | L | 5 days |
| 14 | M | 6 days |

*Figure (5.25): Gantt chart of problem 2 using MS Project after leveling*

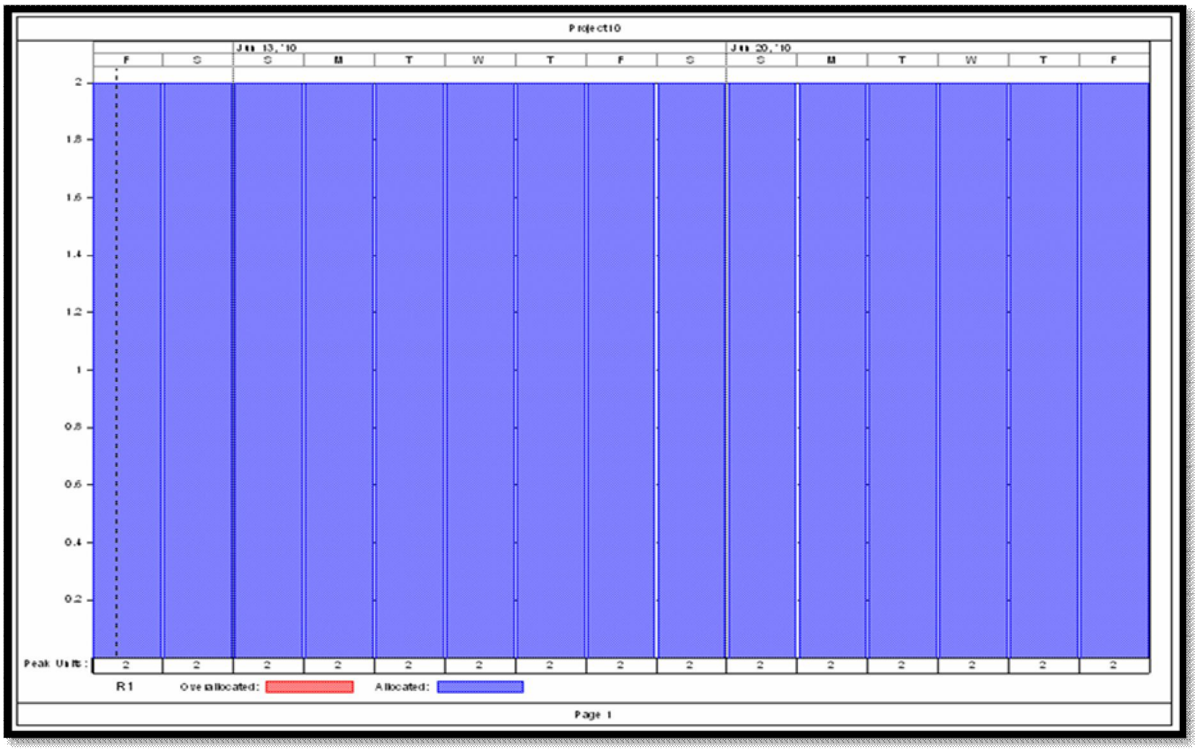


*Figure (5.26): The resources profile of problem 2 after leveling using MS Project*

As shown in figure (5.26), there is no overrun in the resources after leveling using MS Project.

### 5.5.5. CPS Optimizer Solution of the problem

Running CPS Optimizer yield a best schedule with T = 56 days, where the schedule is A & B → C the best solution was obtained using 10 generations only.

The problem was solved using 10, 20, 50 generations, to be sure about the results. Each time the best schedule was obtained with T = 56 days

The starting dates of the activities are shown in figure (5.27) and (5.28).

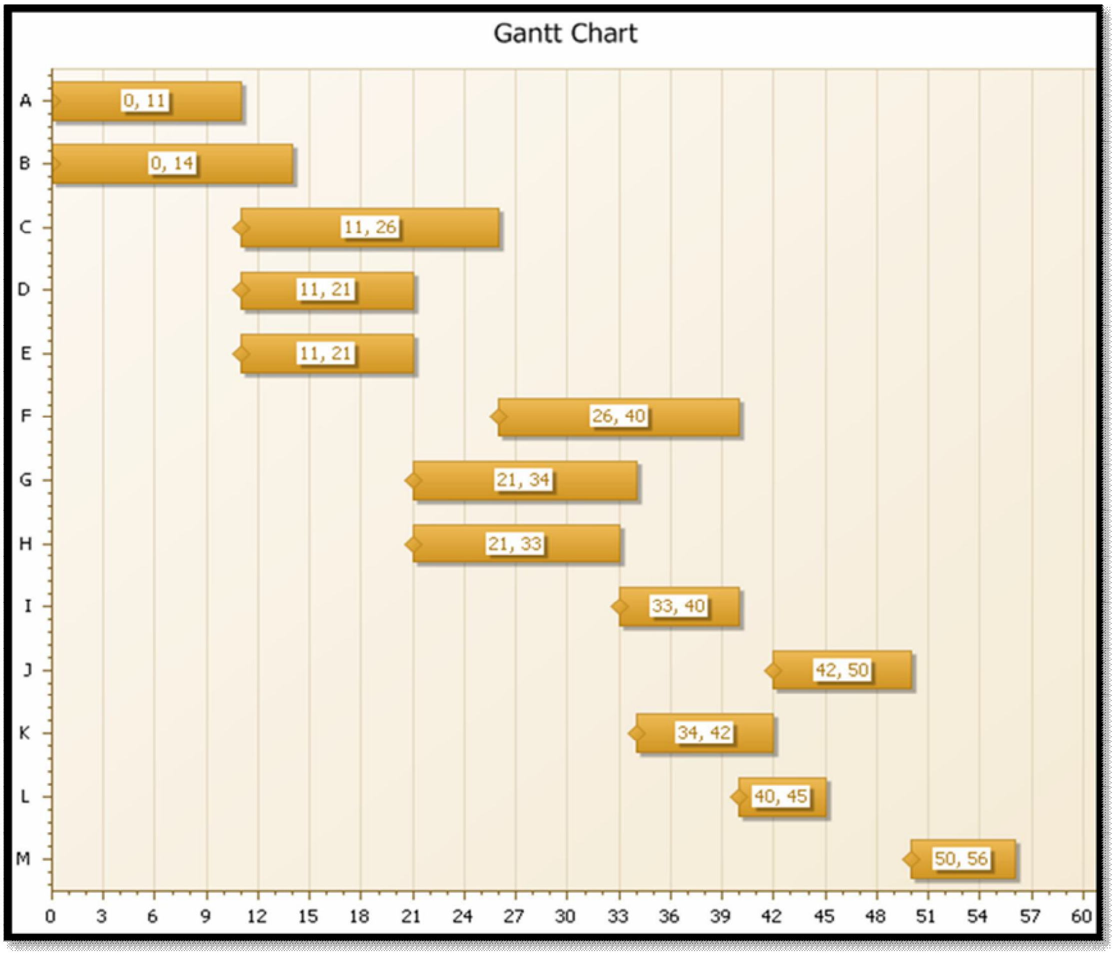
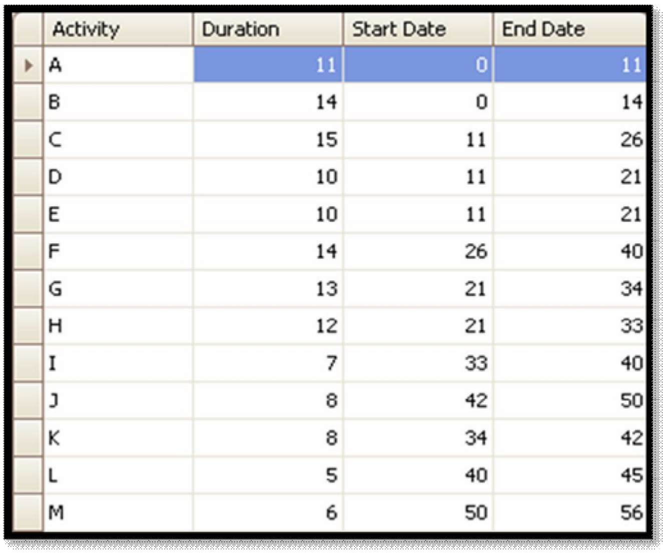


*Figure (5.27): Gantt chart of the best schedule of problem 2*



| Activity | Duration | Start Date | End Date |
|---|---|---|---|
| A | 11 | 0 | 11 |
| B | 14 | 0 | 14 |
| C | 15 | 11 | 26 |
| D | 10 | 11 | 21 |
| E | 10 | 11 | 21 |
| F | 14 | 26 | 40 |
| G | 13 | 21 | 34 |
| H | 12 | 21 | 33 |
| I | 7 | 33 | 40 |
| J | 8 | 42 | 50 |
| K | 8 | 34 | 42 |
| L | 5 | 40 | 45 |
| M | 6 | 50 | 56 |

*Figure (5.28): Starting and ending dates of the best schedule of problem 2*

### 5.5.6. Comparison between the solutions

Table (5.5) shows a comparison between the different mentioned solutions, which shows that the CPS Optimizer results meet the intended goal of achieving the best schedule with the minimum efforts ( 10-20 iterations).

*Table (5.5): Comparison between the solutions of the second problem*

| Solution | Total duration (days) | No. of iterations | Order in the first 3 activities |
|---|---|---|---|
| **CPM** | 48 | -- | Parallel |
| **Primavera** | 60 | -- | A&C → B |
| **MS Project** | 60 | -- | A&C → B |
| **CPS Optimizer** | 56 | 10-20 | A & B → C |

## 5.6. The CPS Optimizer features

The CPS Optimizer has many features:

- It treats the deficiencies of traditional scheduling systems such as CPM that do not take resource constraints in consideration.
- CPS Optimizer interface is easy to use, it contains many tabs, where each tab acts as a sheet of inputs as shown in figures (5.1) and (5.2). Tabs enable user to add different categories of project information ranging from activities durations, activities relationships, resources, and GA parameters. In figure (5.2), resources input sheet is shown where the user can add all the resources available in the project and the available units of each resource. Figure (5.3) shows the activities relationships input sheet, where the user can add the relationships between the project activities. Appendix (1): explain how to use the CPS Optimizer step by step.

CPS Optimizer is a time saving as it has the ability to create only legal random solutions without losing the GA feature of searching global maximum or minimum.

## 5.7. The CPS Optimizer limitations

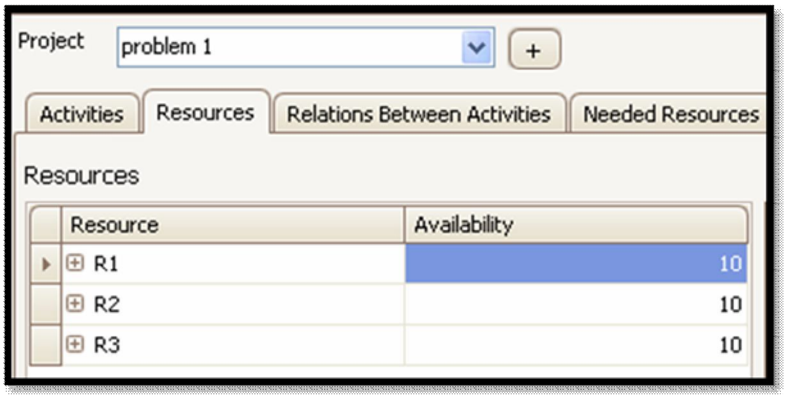
The design of the CPS Optimizer is based on the application of the GA concepts in optimization. It is a general tool that optimizes the schedule of resource constrained construction projects *assuming* the following:

- The relations between activities are supposed to be “finish to start” relationships.
- The resource usage during the activity is supposed to be in a constant rate over the duration of the activity.

## 5.8. Parametric study on CPS Optimizer

### 5.8.1. Projects without resource constraints

The developed model can deal with construction projects with or without resource constraints. In the previous section, two problems with resource constraints were studied. The developed model can deal with projects without resource constraints; that is the available resources are not limited. In the absence of resource constraints, the best schedule giving the minimum makespan is the CPM schedule. This was tested by solving problem 1 supposing unlimited resources ; large numbers of the available resources were added to the CPS optimizer to ensure that there is no resource constraints as shown in figure (5.29).



Project: problem 1

Resources

| Resource | Availability |
|---|---|
| R1 | 10 |
| R2 | 10 |
| R3 | 10 |

*Figure (5.29): Adding large numbers to the available resources of problem 1*

After solving the problem 1 without resource constraints using CPS Optimizer, the resulted schedule was the CPM schedule with total duration equals 75 days, and a schedule of activities A1, B1, and C1 in a parallel manner as shown in figure (5.30), where the starting date of each activity is shown in figure (5.31).

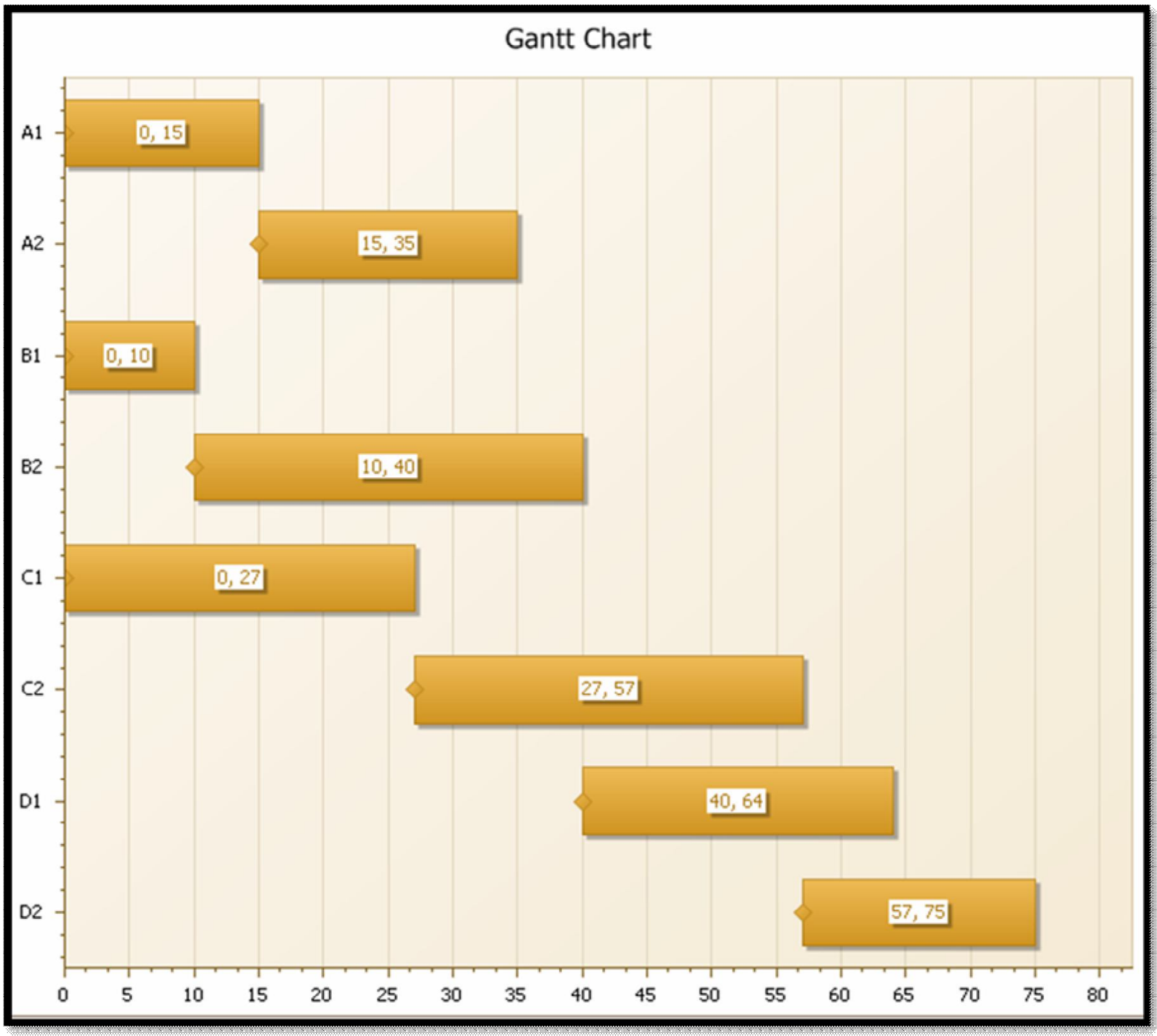
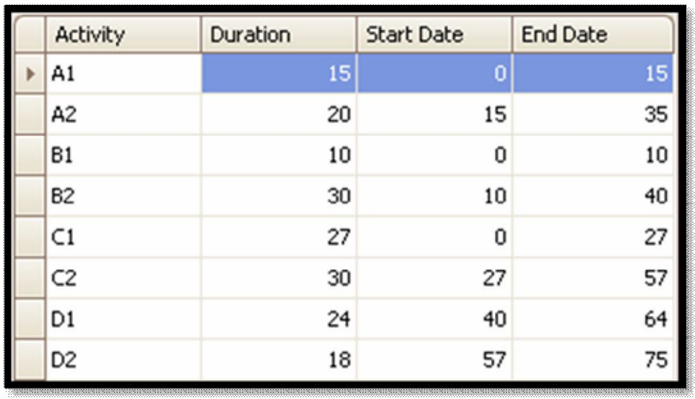


*Figure (5.30): CPS optimizer schedule of problem 1 supposing unlimited resources*



| Activity | Duration | Start Date | End Date |
|---|---|---|---|
| A1 | 15 | 0 | 15 |
| A2 | 20 | 15 | 35 |
| B1 | 10 | 0 | 10 |
| B2 | 30 | 10 | 40 |
| C1 | 27 | 0 | 27 |
| C2 | 30 | 27 | 57 |
| D1 | 24 | 40 | 64 |
| D2 | 18 | 57 | 75 |

*Figure (5.31): CPS optimizer schedule of problem 1 supposing unlimited resources*

### 5.8.2. Suitable number of iterations

The number of iterations is one of the stopping criteria used in the GA problems which highly affect the probability of getting best solutions. More iterations mean more time and effort but also mean high probability of getting best solution, so that the number of iterations should be chosen accurately.

Problem 1 was solved many times with many numbers of iterations for testing purposes. When the number of iteration was smaller than 10 (for example 3), the final solution was not optimum as shown in figure (5.32) and (5.33).

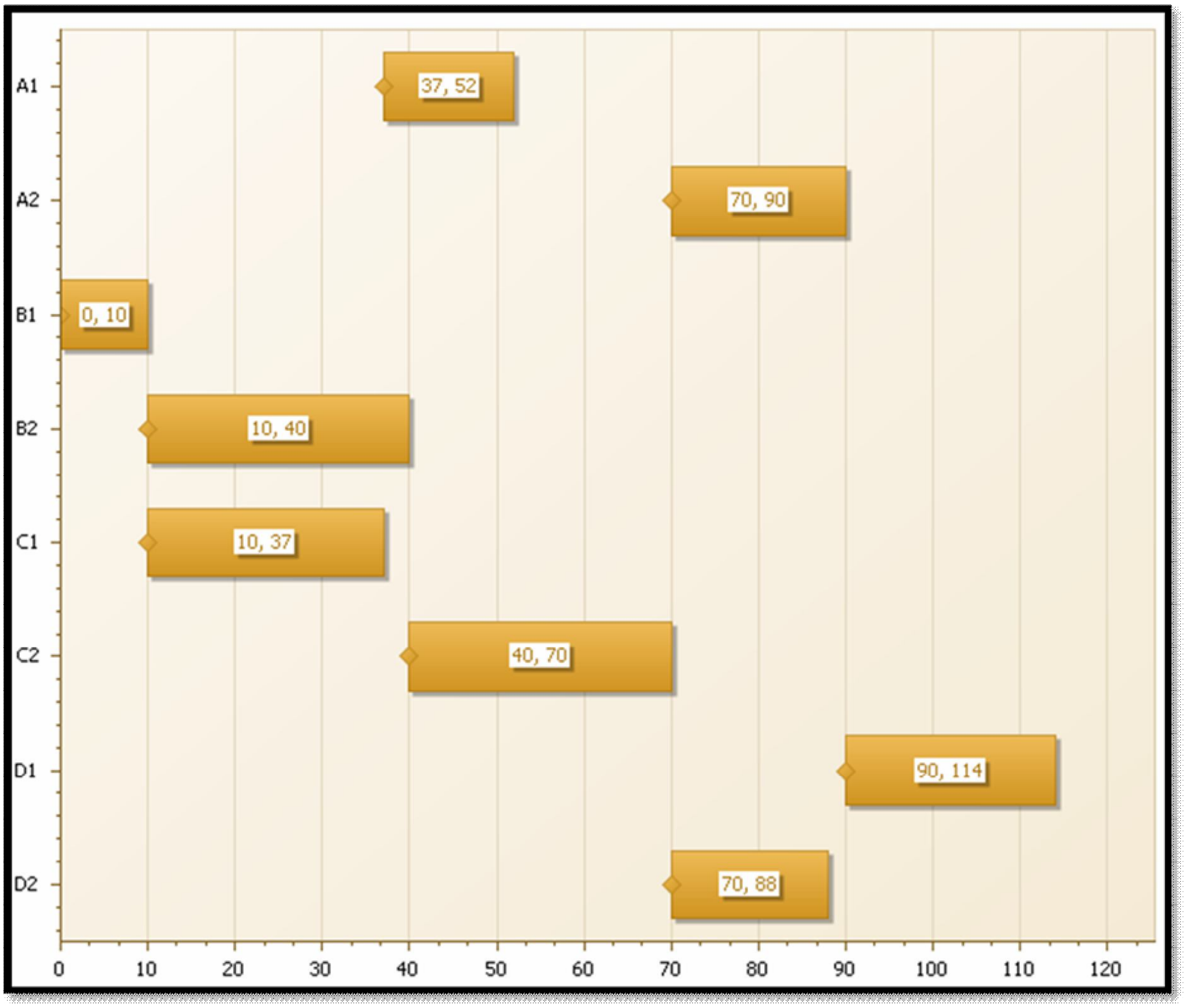


*Figure (5.32): CPS optimizer schedule of problem 1 using 3 iterations*

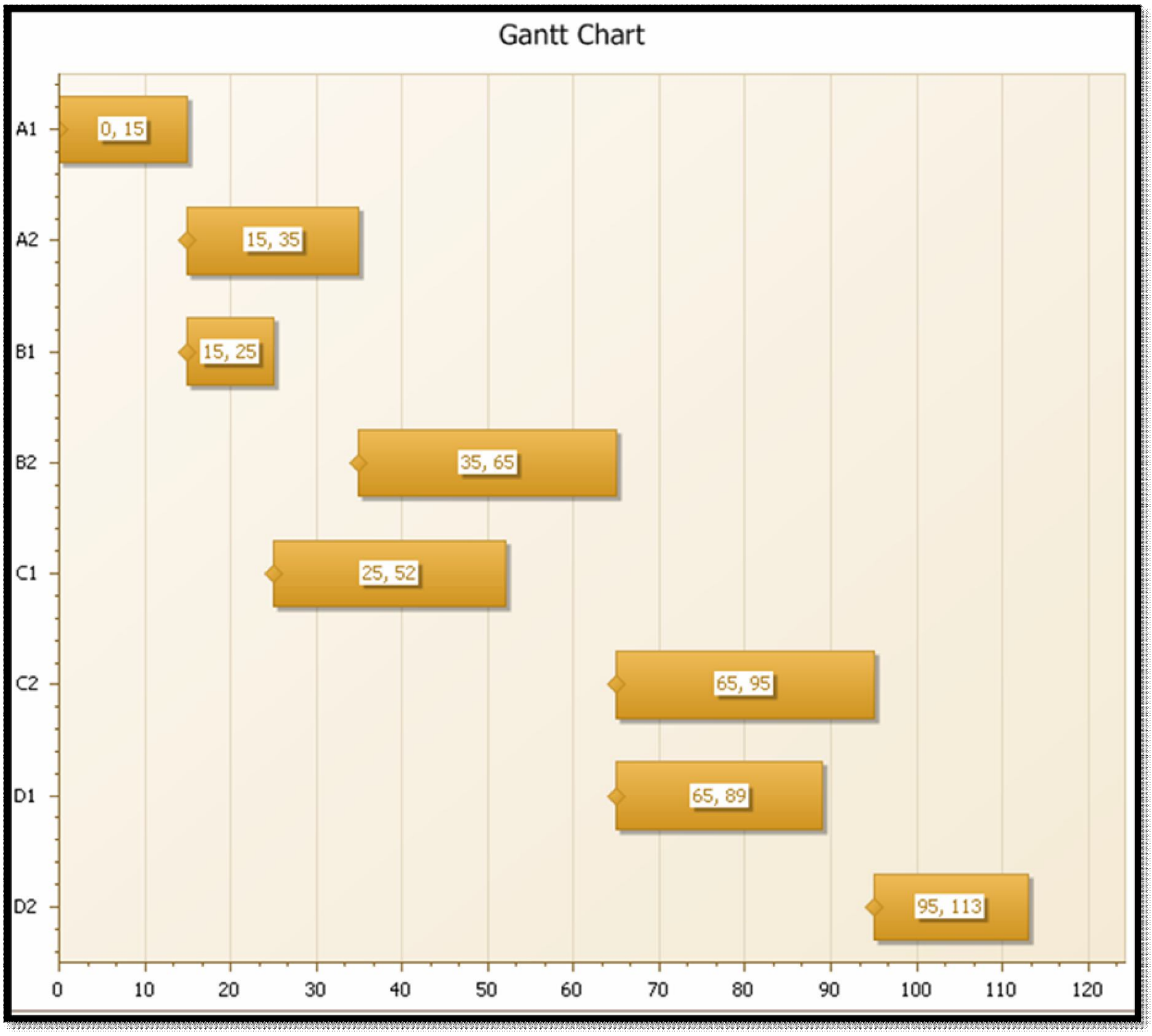


*Figure (5.33): CPS optimizer schedule of problem 1 using 4 iterations*

When increasing the number of iterations (more than 10), each time we got a best schedule with total duration 108 days as described in the previous sections. This makes the user surer about the best schedule.

### 5.8.3. Feasible solutions developer operator effect

The high speed of getting the best schedule using the developed model comparing with other models returns to the using of the developed operator (FSD Operator). Returning back to the programming of the CPS Optimizer, the feasible solutions developer operator was stopped to test the effect of using it. This needed too much time, and too much iterations to get the best schedule. 1000 iterations was an average number of the needed iterations, so that using feasible solutions developer operator highly enhance the convergence toward the feasible solution.

# CHAPTER (6)

## Conclusion and Recommendations

## 6.1. Achievements

The primary aim of this research is to develop a genetic algorithm optimization model to schedule project activities in order to minimize the total duration of the project, subjected to both precedence constraints and resources constraints. The main achievements of this project can be summarized as follows:

- A new approach was developed to create the population of the genetic algorithm individuals, that is the *"feasible solutions developer operator*" that enables the user to create completely feasible solutions that satisfy all constraints, and this helps in getting a quick convergence toward the best solution during GA stages, without losing the GA feature of searching global maximum or minimum.
- A new crossover operator was developed in this study; the procedure of the new crossover operator suit the scheduling problem formulation, and suit the type of the chromosomes encoding used.
- An implementation of the GA developed model for resource-constrained project scheduling has resulted in application program called the CPS Optimizer "The Construction Projects Scheduling Optimizer program".
- The CPS Optimizer treats the deficiencies of traditional scheduling systems such as CPM that don't take resource constraints in consideration, it has an interface which is easy to use, it contains many tabs, and each tab acts as a sheet of inputs. Tabs enable user to add different categories of project information ranging from activities durations, activities relationships, resources, and GA parameters. It is time saving program because it uses the feasible solutions developer operator.
- Verification and validation of the developed program were applied. A comparison with the results of previous studies and with the feature of "resource leveling" in the commercial construction management programs showed a strong efficiency of the developed model.

## 6.2. Conclusion

The main conclusions drawn from the current research can be summarized as follows:

- One of the major drawbacks of GAs; that is they are computationally expensive has been overcome by using the “feasible solutions developer operator” that enables the user to create completely random feasible solutions
- Using “feasible solutions developer operator” accelerate getting a quick convergence toward the best solution during GA stages comparing with traditional approach of creating random solutions in the initial population.
- The performance of the developed model enhanced a lot when using a new suitable crossover operator that suits the studied problem and creates feasible solutions.
- An implementation of the GA developed model for resource-constrained project scheduling has resulted in application program called the CPS Optimizer “The Construction Projects Scheduling Optimizer program. This program is to implement genetic algorithm to resource constrained project scheduling problems within software in order to further investigate the practical benefits to real world projects.
- The developed model was verified by more tests performed, errors were identified, and corrections were made to encountered errors. This took a long period of time, and a huge effort by the researcher to get a high degree of certainty about the model verification.
- Validation of the developed CPS Optimizer was done. Two problems are applied and the results are studied and compared with the results of previous studies and with the feature of “resource leveling” in the commercial construction management programs, which showed that the CPS Optimizer results meet the intended goal of achieving the best schedule with the minimum efforts.

## 6.3. Recommendations

- Many problems in construction have been identified by many researchers, so the coming researches should focus on finding solutions for these problems.
- The researcher recommends that additional studies should be done on the developed model in order to develop his ability to solve resource-constrained problems and overcome the limitations of the program.
- Studies should be done to enlarge the type of relationships between activities that the program can solve other than finish to start relationships.
- In practical projects, the activity does not necessary has a constant rate of resources over his total duration. Actual resource usage must be taken in consideration in future researches.
- Researchers should integrate the efforts toward an integrated program for optimizing scheduling, that optimize time, resources, costs and other measurements of project management performance.

# REFERENCES

# References

Al-Najjar, J., 2008, “Factors influencing time and cost overruns on construction projects in the Gaza Strip”, MSc thesis in construction management", The Islamic University of Gaza.

Coa, Y. J., Wu, Q. H., 1999, “Teaching genetic algorithm using Matlab.", Int. J. Elect. Enging. Educ., Vol. 36, pp. 139–153. Manchester U.P.

Deiranlou, M., and Jolai, F., 2009, “A new efficient Genetic Algorithm for project scheduling under resource constraints”, World Applied Sciences Journals 7 (8): 987-997.

EL Rayes, k. and moselhi, O. 2001, “Optimization resource utilization for repetitive construction projects”, Journal of construction engineering and management, January/Febrauary 2001.

Franco, E. G.,Zurita, F. L., Delgadillo, G. M., 2006 " A genetic algorithm for the resource-constrained project scheduling problem.", Report, School of Industrial Engineering, Universidad de La Sabana.

Goncalves, J. F., Mendes, J. J. M., Resende, M. G. C., 2006, " A genetic algorithm for the resource-constrained multi-project scheduling problem.", AT&T Labs Research Technical Report: TD-668LM4, Revised January 20.

Guo, Z. X., Wong, W. K., Leung, S. Y., Fan, J. T., Chan, S. F., " Genetic optimization of order scheduling with multiple uncertainties.", Expert systems with applications 35 (2008) 1788-1801.

Hartmann, S., 1998, “A competitive genetic algorithm for resource-constrained project scheduling.”, Naval Research Logistics 45:733-750.

Haupt, R. L., Haupt, S. E., 2004, “Practical genetic algorithms", Second edition, A john Wiley and sons , Inc., New Jersey.

Hinze, J. W., 2008, “Construction planning and scheduling”, third edition, Pearson Education, Inc., Upper Saddle River, New Jersey

Jackson, B., 2004 “Construction Management, Jump start”, First edition SYBEX Inc,. USA.

Kamarainen, O., Ek, V., Nieminen, K., Ruuth, S., 2000, " Large scale generalized resource-constrained scheduling problems: Genetic Algorithm approach." Imperial College, London, UK.

Kerzner, H., 2006, " Project management, A system approach to planning, scheduling and controlling", Ninth edition, John wiley & Sons Inc.

Kim, J. L.,Ellise, R. D., 2005, " A framework for integration model of resource-constrained scheduling using genetic algorithm.", Proceedings of the 2005 Winter Simulation Conference.

Kolisch, R., Hartmann, S., 1999, " Heuristic Algorithms for solving the resource-constrained project scheduling problem: classification and computational analysis.", Project scheduling: Recent models, algorithms and applications, pages 147–178, Kluwer, Amsterdam, the Netherlands.

Kumar, D. N., 2005 "Optimization Methods: Introduction and Basic Concepts", IISc, Bangalore..

Lin, C. M., Gen, M., 2008, "Multi-criteria human resource allocation for solving multistage combinatorial optimization problems using multiobjective hybrid genetic algorithm.", Expert systems with application 34 (2008) 2480 – 2490.

Leu, S. S., Yang, C. H., 1999, "GA-based multi-criteria optimal model for construction scheduling", Journal of Construction Engineering and management, Nonember-December 1999.

Liu, Y., Zhao, S. L.,Du, X. K.,Li, S. Q., 2005, " Optimization of resource allocation using genetic algorithms.", Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21, August.

Mahfouz, S. Y., 1999, "Design Optimization Of Structural Steelwork", Ph.D Thesis, University of Bradford, UK.

Matlab, 2005 User's guide:" Genetic algorithm and direct search toolbox, for use with matlab.", the web site: www.mathworks.com.

Matalb, 2009, version 2009a, mathworks company.

Mendes, J. J. M., Goncalves, J. F., Resende, M. G. C., 2006, “A random key based genetic algoritm for the resource-constrained project scheduling problem.", AT&T Labs Research Technical Report: TD-6DUK2C, Revised November 8.

Neapolitan, R., Naimipour, K., 1998, “Foundations of algorithms, using C++ pseudo code.", Second edition, Jones and Bartlett publishers, USA.

Okada, I., Zhang, F., Yang, H. Y., and Fujimura, S., 2010, “A random key-based Genetic algorithm approach for resource-constrained project scheduling problem with multiple modes”, Proceedings of the international multi-conference of engineers and computer scientists, Vol I.

Patrick, C. 2004, “Construction project planning and scheduling”, Pearson Education, Inc., Upper Saddle River, New Jersey.

Primavera 5, software packages, project and program management software, Primavera Systems Inc.

Schexnayder, C., and Mayo, R., 2004, “Construction Management fundamentals”, McGraw Hill, New Yrok.

Senouci, A., Al-Derham, H. R., 2008, " Genetic algorithm-based multi-objective model for scheduling of linear construction projects.", Advances in Engineering Software 39 (2008) 1023- 1028.

Senouci, A., Eldin, N., 2004," Use of genetic algorithms in resource scheduling of construction projects." Journal of construction engineering and management, November-December.

Shih, K. C., and Liu, S. S., 2006, “Optimization model of external resource allocation for resource constrained project scheduling problem”, ISARC.

Sivanandam, S. N., Deepa, S. N., 2008, “Introduction to genetic algorithms”, Springer-Verlag Berlin Heidelberg.

Sriprasert, E., Dawood, N., 2002, “Next Generation Of Construction Planning And Control System”, Proceedings IGLC-10, Gramado, Brazil.

Steven, J. D., 1990, “Techniques for construction network scheduling”, McGraw hill publishing company.

Toklu, Y. C., 2002, “Application of genetic algorithms to construction scheduling with or without resource constrains.” Canadian Journal of civil Engineering, Vol.29 No:3,pp.421-429.

Wall, M. B., 1996, "A genetic algorithm for resource-constrained scheduling.", Ph.D thesis in Mechanical Engineering , Massachusetts Institute of Technology..

Wysocki, R., 2007, “Effective project management”, Fourth edition, Wiley publishing, Inc., Indiana.

Xu, N., Nozick, L., Bernstein, O., Jones, D., 2007, “Stochastic rollout and justification to solve the resource-constrained project scheduling problem.", Proceedings of the 2007 Winter Simulation Conference.

Yahia, A., 2004, “Time schedule preparation by predicting production rate using simulation”, MSc thesis in construction management", The Islamic University of Gaza.

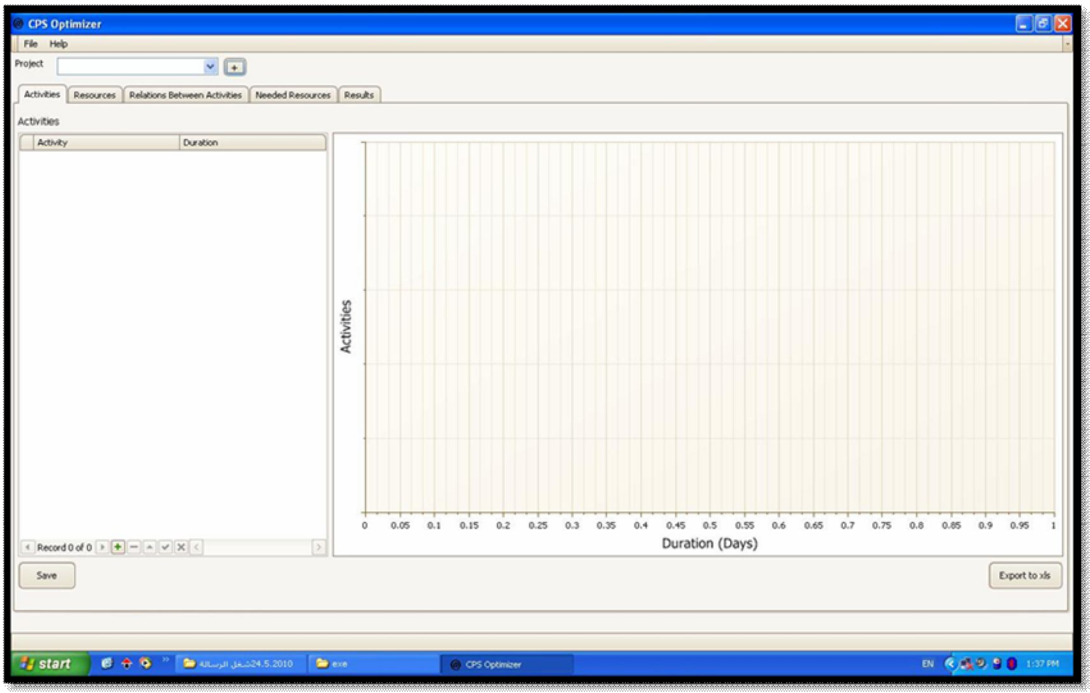
Yang, B., Genus, J.,O'Brien, W., 2001, " Resource-constrained project scheduling: Past work and new directions.", Research Report, Department of Industrial and Systems Engineering, University of Florida.

# APPENDIX

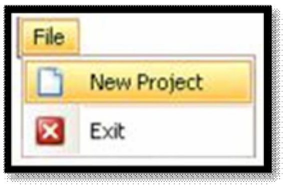
## How to use CPS Optimizer?

**You can follow the following steps to schedule a construction project using CPS Optimizer:**

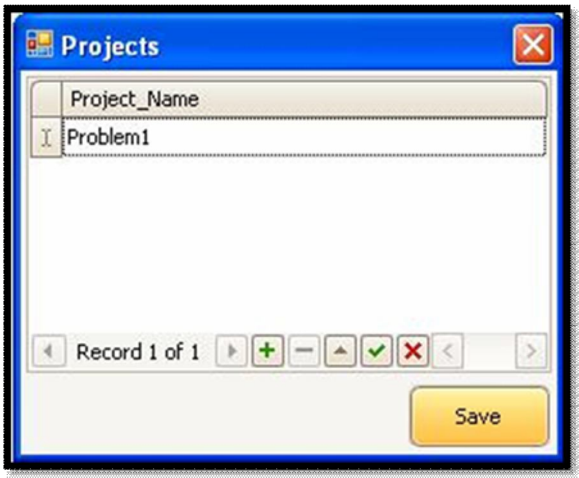
- ➢ After opening the program, the interface will be as shown in the figure below.



- ➢ To enter a new project, open the file menu, and choose “ New project”

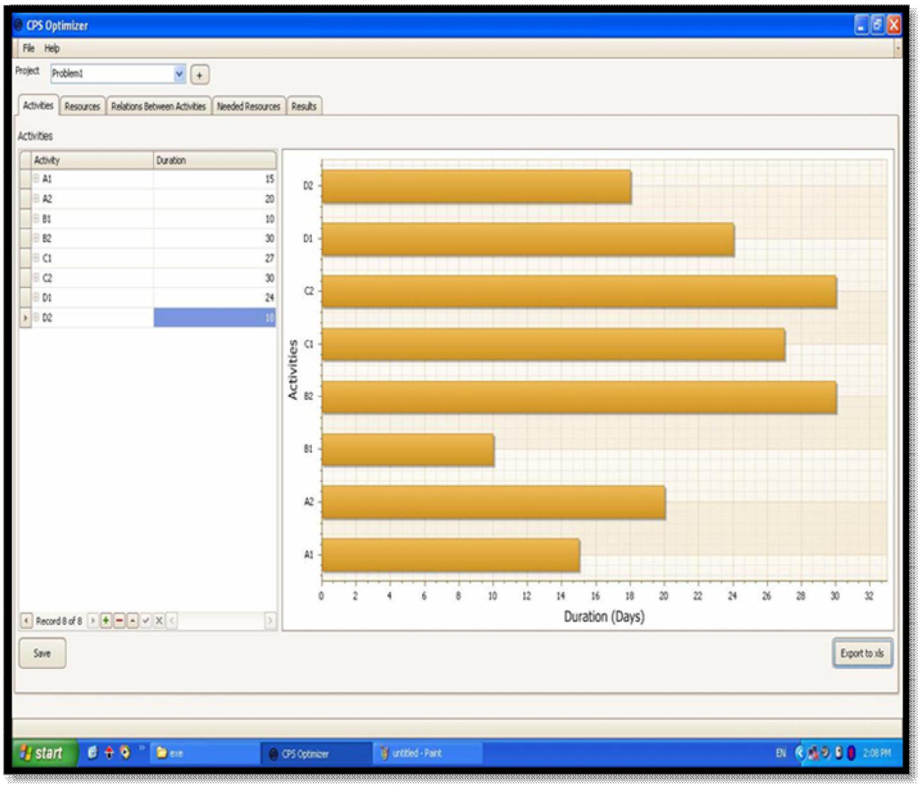


- ➢ Add a new project and name it as shown in the figure



| | Project_Name |
|---|---|
| | Problem1 |

- Add the activities and the corresponding duration of each activity in the rows on the left, automatically the program will produce a chart on the right that shows the added activities and their durations

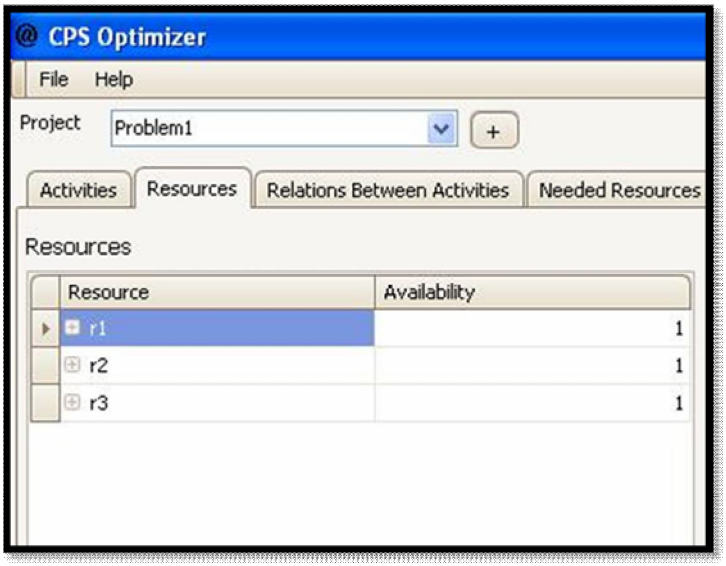


Project Problem1

Activities

| Activity | Duration |
|---|---|
| A1 | 15 |
| A2 | 20 |
| B1 | 10 |
| B2 | 30 |
| C1 | 27 |
| C2 | 30 |
| D1 | 24 |
| D2 | 18 |

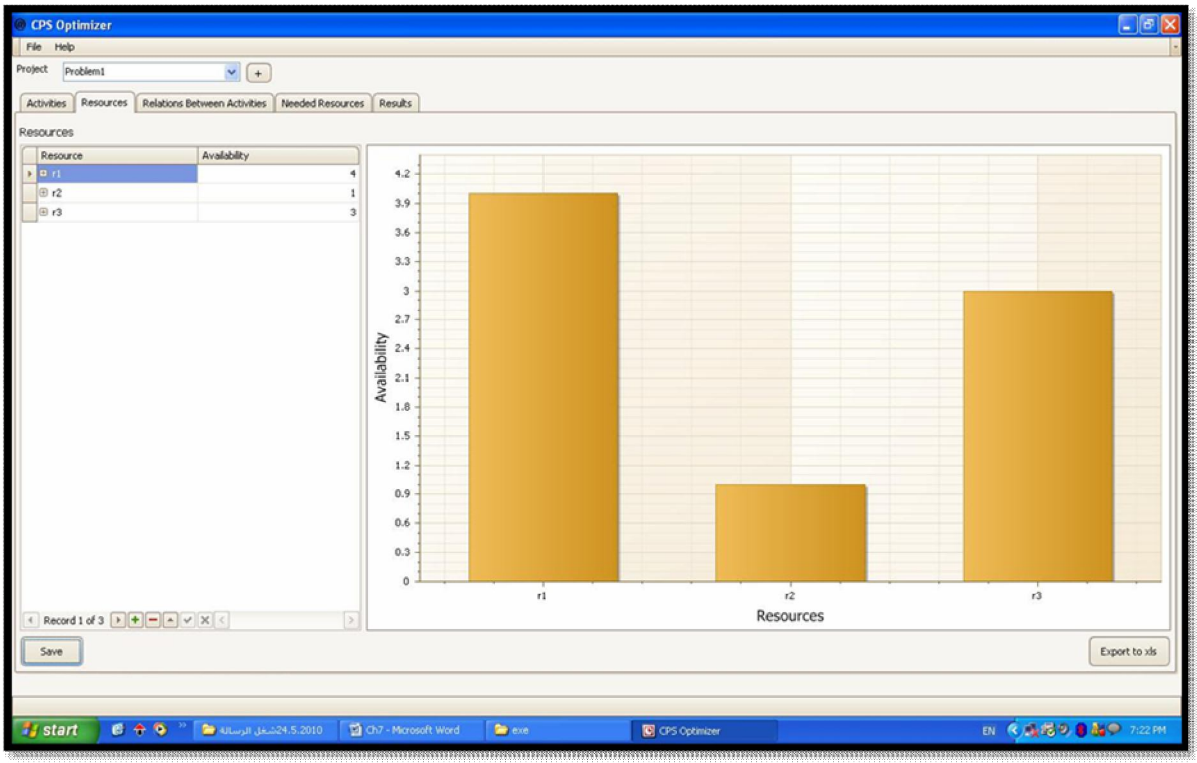
- Move to the next tab “Resources”, and add the resources you have in your project as shown in the following figure, where you can add the name and the available amount of each resource.



Project: Problem1

Resources

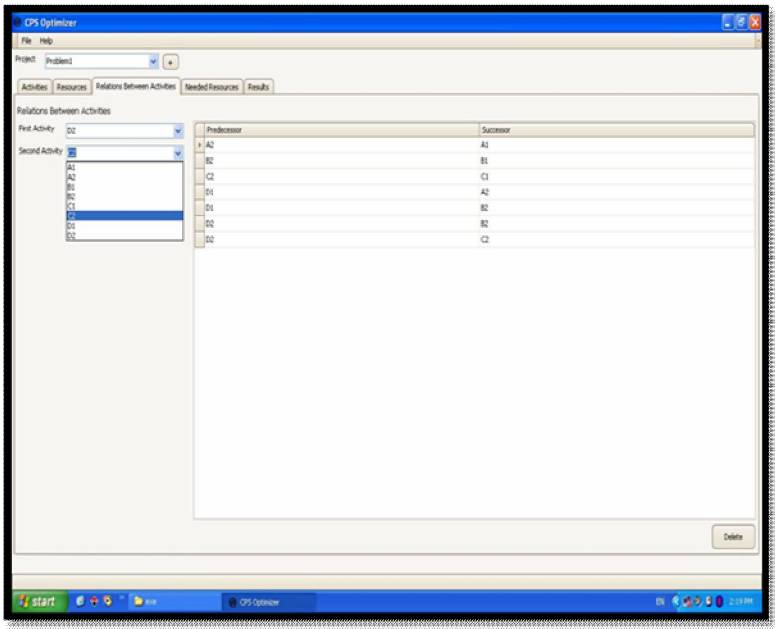
| Resource | Availability |
|---|---|
| r1 | 1 |
| r2 | 1 |
| r3 | 1 |



Resources

| Resource | Availability |
|---|---|
| r1 | 4 |
| r2 | 1 |
| r3 | 3 |

- ➢ To add the relations between activities, you have to move to the next tab which is called “Relations between activities”, you will find all the added activities are available, you can choose the predecessor and the successor activities from the left to add them to the list on the right.



Project: Problem1

Relations Between Activities

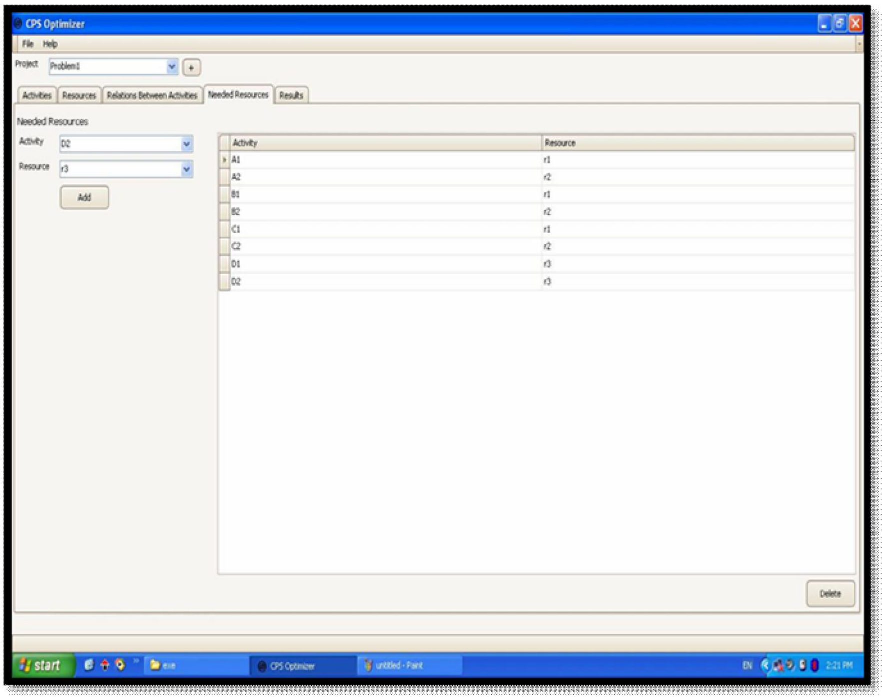
First Activity: D2

Second Activity: C2

A1
A2
B1
B2
C1
C2
D1
D2

| Predecessor | Successor |
|---|---|
| A2 | A1 |
| B2 | B1 |
| C2 | C1 |
| D1 | A2 |
| D1 | B2 |
| D2 | B2 |
| D2 | C2 |

- You can add the resources requirements of each activity, by moving to the following tab “Needed resources, where you will find all the added resources and activities are available on the left, so you can assign the needed resources for each activity, and add them to the list on the right.



Project: Problem1

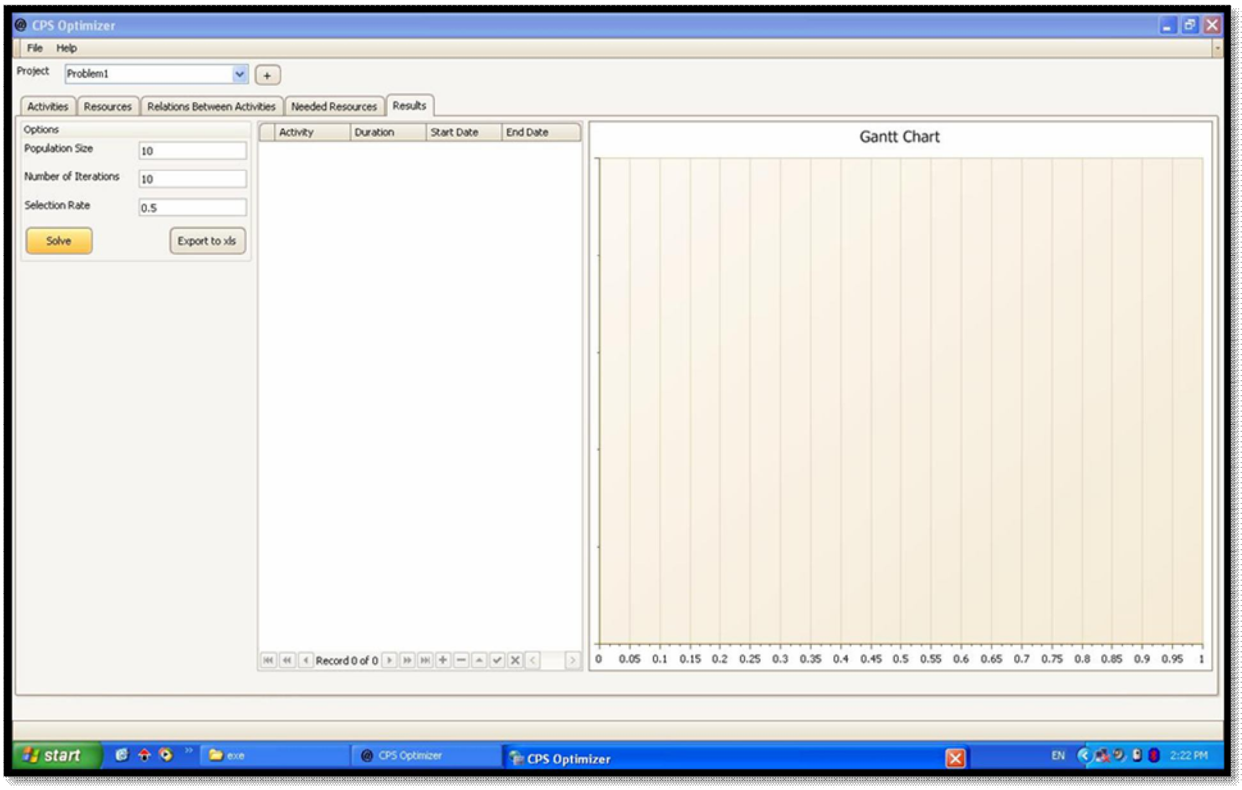
Needed Resources

Activity: D2

Resource: r3

| Activity | Resource |
|---|---|
| A1 | r1 |
| A2 | r2 |
| B1 | r1 |
| B2 | r2 |
| C1 | r1 |
| C2 | r2 |
| D1 | r3 |
| D2 | r3 |

- To find the optimum schedule of your project, move to the tab “Results”.

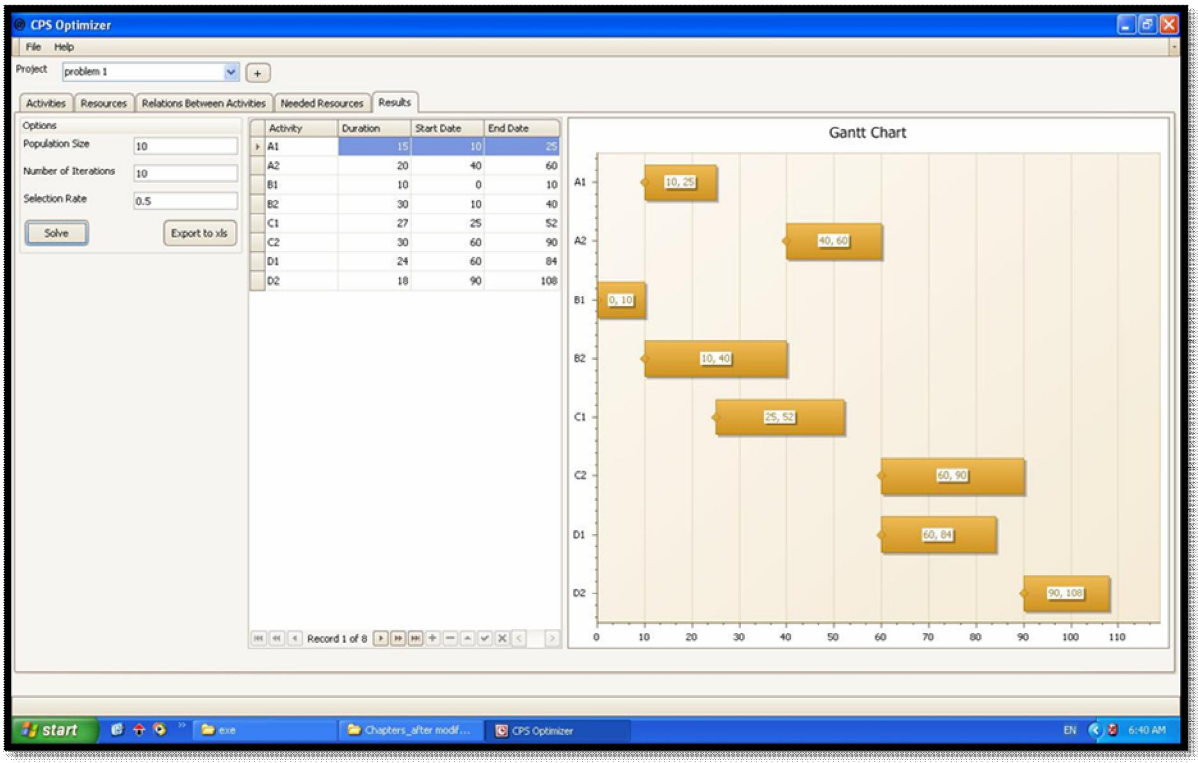


Population Size 10

Number of Iterations 10

Selection Rate 0.5

- You can enter your own GA parameters in the options field, or use the default values.
- Press "Solve" button to solve your schedule as shown in the following figure.



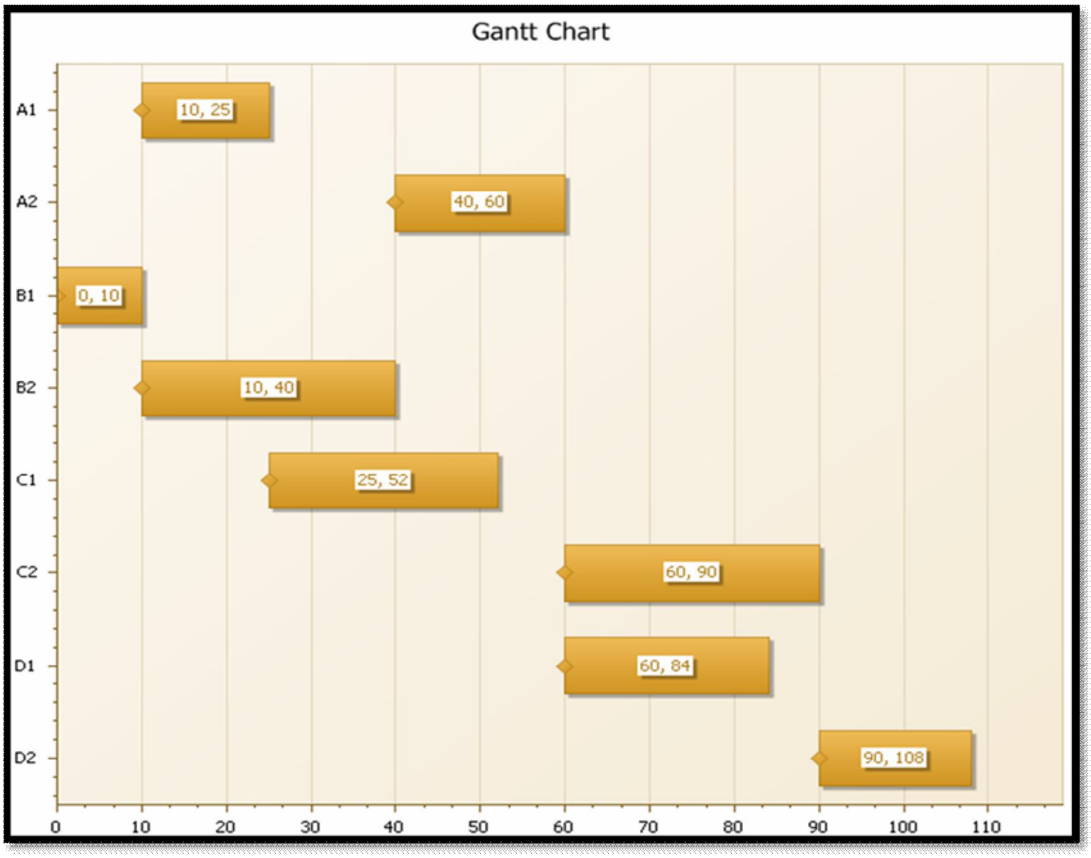
Project: problem 1

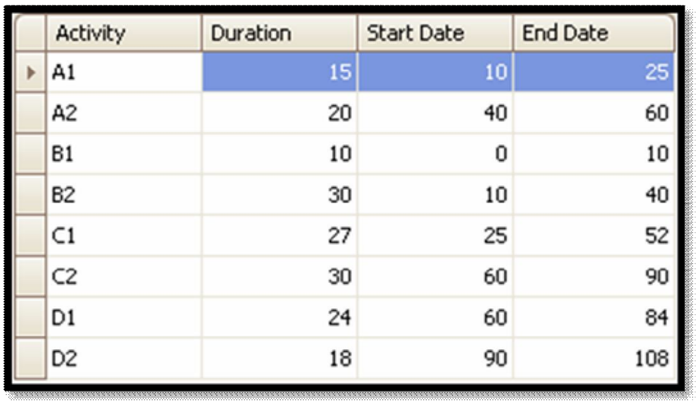
Options

| | |
|---|---|
| Population Size | 10 |
| Number of Iterations | 10 |
| Selection Rate | 0.5 |

| Activity | Duration | Start Date | End Date |
|---|---|---|---|
| A1 | 15 | 10 | 25 |
| A2 | 20 | 40 | 60 |
| B1 | 10 | 0 | 10 |
| B2 | 30 | 10 | 40 |
| C1 | 27 | 25 | 52 |
| C2 | 30 | 60 | 90 |
| D1 | 24 | 60 | 84 |
| D2 | 18 | 90 | 108 |

- The results are obtained are the Gantt chart of the best schedule of your project and a table shows the starting date and ending date of each activity





| Activity | Duration | Start Date | End Date |
|---|---|---|---|
| A1 | 15 | 10 | 25 |
| A2 | 20 | 40 | 60 |
| B1 | 10 | 0 | 10 |
| B2 | 30 | 10 | 40 |
| C1 | 27 | 25 | 52 |
| C2 | 30 | 60 | 90 |
| D1 | 24 | 60 | 84 |
| D2 | 18 | 90 | 108 |

- You can save your work or modify it to re-solve it.