# RESOURCE LEVELING USING IMPROVED

# GENETIC ALGORITHM

BY

MOHAMMED ABDUL RAHMAN

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

**KING FAHD UNIVERSITY OF PETROLEUM & MINERALS**

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the

Requirements for the Degree of

**MASTER OF SCIENCE**

**In**

CONSTRUCTION ENGINEERING AND

MANAGEMENT

MAY 2012

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by Mohammed Abdul Rahman under the direction of his thesis advisor and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE in CONSTRUCTION ENGINEERING MANAGEMENT**.

Thesis Committee

Dr. Ashraf M. Elazouni (Advisor)

Dr. Sadi Assaf (Member)

15/05/2012

Dr. Mohammed I. Al-Khalil (Member)

Dr. Khalaf A. Al-Ofi
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

10/6/12

Date

بسم الله الرحمن الرحيم

*Dedicated to*

*My parents and sister for their duas and constant support and encouragement throughout my life*

# ACKNOWLEDGEMENT

All praise belongs to Almighty Allah (s.w.t.) for bestowing me with health, knowledge and patience to carry out this work and complete my M.S. successfully at King Fahd University of Petroleum and Minerals, Dhahran. I am happy to have had a chance to glorify His name in the sincerest way through this small accomplishment and ask Him to accept my efforts. Acknowledgement is due to King Fahd University of Petroleum and Minerals for providing me financial support and good academic environment during the course of my master studies.

My deepest gratitude and appreciation goes to my thesis advisor Dr.Ashraf Elazouni for his constant help, guidance and motivation during the course of my study. His priceless suggestions made this work interesting and challenging for me. I also wish to express my deepest appreciation to Dr.Sadi Assaf & Dr. Ibrahim Al-Khalil for thier help, guidance, and constant encouragement during my master's program.

I would also like to acknowledge all the Construction Engineering Management faculty members with whom I took courses during my M.S. as they have helped me tremendously, guided me and encouraged me during my coursework and related research.

I am forever indebted and grateful to my parents & my sister for their continuous encouragement and support. I wish to appreciate and thank all my friends as they have my pillar of support throughout the course of my master's studies. Special thanks to Ahmed Zubair, Azfar Amaan, Riyaz bhai, Azhar bhai and Masood bhai for their help during the course of my completion of thesis and otherwise.

## CONTENTS

## LIST OF FIGURES

# LIST OF TABLES

# ABSTRACT (ARABIC)

الغرض من هذا البحث هو استخدام تقنية الخوارزميات الجينية (GA) لتسوية الموارد من عمالة ومعدات الخاصة بمشاريع التشييد. وسيتم فى هذا البحث توظيف اثنين من الخوارزميات الجينية (GA) من المؤلفات السابقة واللذان يعملان بحيث يتم الحفاظ على علاقات الأسبقية بين أنشطة المشروع أثناء عملية التطور وذلك بهدف تقييم أداءهما من حيث جودة الحلول والزمن المستغرق. ويهدف البحث إلى تحسين أداء تقنية GA في حل المشاكل الخاصة بجدولة مشاريع التشييد المتعلقة بتسوية الموارد من خلال تحسين نوعية الحلول وخفض الزمن المستغرق بحيث يمكن استخدامها بشكل فعال في المشاريع العملاقة.

.

# ABSTRACT (ENGLISH)

The purpose of this research is to use Genetic Algorithm (GA) technique for resource leveling of construction projects. Two different precedence-preserving crossover and mutation operators from the literature will be employed in the GA technique with the objective of evaluating their impact on the performance of the GA technique as measured by the quality of solutions and computation cost. The proposed improvement of GA technique will be conducted by using resource leveling problems as a bed to compare between the performances of the two precedence-preserving operators. The research aims at upgrading the performance of the GA technique in solving scheduling problems in general and recourse leveling problems in particular through upgrading the quality of solutions and reducing the computation cost so it can be used efficiently to solve large size problems.

# CHAPTER 1 INTRODUCTION

## 1.1 Problem Statement

Efficient resource leveling avoids disruptive work schedules, minimizes idle resource time and reduces release and re-hire of temporary workforce. The purpose of this research is to use Genetic Algorithm (GA) technique for resource leveling of construction projects. However, the evolution process of the GA technique in the domain of resource leveling is subject to yield offspring chromosomes wherein the precedence relationships are violated. Hence, checking the generated chromosomes and repairing the infeasible chromosomes are required which cause computational inefficiency of the GA technique.

## 1.2 Research Objective

1- Two different precedence-preserving crossover and mutation operators in the literature will be employed in the GA technique with the objective of evaluating their impact on the performance of the GA technique as measured by the quality of solutions and computation cost.

2- The proposed improvement of GA technique will be conducted by using resource leveling problems as a bed to compare between the performances of the two precedence-preserving operators.

## 1.3 Benefits

The research aims at upgrading the performance of the GA technique in solving scheduling problems in general and recourse leveling problems in particular through upgrading the quality of solutions and reducing the computation cost so it can be used efficiently to solve large size problems.

# CHAPTER 2 LITERATURE REVIEW

## 2.1 Definition of Resource Leveling

Resource Leveling is related to planning of construction projects which aims at reducing undesirable resource fluctuations so to produce a cost-effective and productive construction schedule [2]. Efficient resource leveling avoids disruptive work schedules, minimizes idle resource time and reduces release and re-hire of temporary workforce [3]. Hinze [4] describes and differentiate two terms in resource management: namely, resource allocation which refers to the allocation of resources that are limited; and resource leveling which refers to the efficient usage of required resources when the project duration is fixed . Resource allocation is also called resource-constrained scheduling or just resource scheduling and resource leveling is also called time-constrained scheduling [5].

The chief objective of this research is to develop a GA based resource leveling approach. However, the proposed approach also yields optimal resource profile beyond the fixed initial duration of project, provided the number of days the total project duration is to be extended is specified for investigation.

## 2.2 Existing Resource Leveling Methods

A literature review of the existing resource leveling methods is necessary to study the limitations that exist in the previous methods so as to understand the need to develop to an improved holistic approach for resource leveling. The existing methods can be broadly classified into three categories: Exact Methods, Simulated Evolutionary Algorithm Methods, Heuristic Methods and Meta-Heuristic Methods.

Exact Methods: These are the methods which give exact solutions whose algorithms are based upon enumeration, integer programming, or dynamic programming techniques [6]. Some of these type of models developed in the literature are by Petrovic [7], Ahuja [8], Easa [1], Younis and Saad [9] and Bandelloni, Tucci et al. [10]. These models can only be applied to schedules with limited number of activities or small size projects [2]. It is because the problem of "combinatorial explosion" occurs if exact methods are applied to large schedules i.e., with increase in number of activities, there is enormous increase in number of possible schedules; thus, considerably increasing the solution search space which consequently makes the task of finding the optimal solution by exact methods cumbersome [11]. Exact methods can handle only single-resource projects [6]. Hence, they cannot be used to solve large and complex problems effectively which limit their use for practical purposes.

Heuristic Methods: A few heuristic approaches previously developed in the existing literature are offered by Galbreath [12], Burgess and Killebrew [13], Son and Skibniewski [14], Ahuja [8] and Harris [15]. These heuristic methods can handle complex problems and have been extensively used in practice [2]. However, there are limitations which restrict the use of these methods in general for all construction schedules. These models are problem dependent, which implies that the rules specific to a model cannot be applied equally for all problems [11]. Moreover, majority of these methods can handle only single resource projects [2], and the optimal solution is not always guaranteed by these models pertaining to different characteristics exhibited by varying schedules. [11].

Meta-heuristic Methods: The resource leveling problems can be solved using computational techniques based on biological and animal pattern called as Meta-heuristic methods. These methods include genetic algorithm based techniques, taboo searches, simulated evolutionary algorithm and particle swarm optimization approaches [16]. An intelligent simulated evolutionary algorithm called ant colony optimization (ACO) was developed by several Italian authors. This method simulated the natural auto catalytic behavior of ants to find the shortest path [17]. The first model for resource leveling using ACO was developed by Ding and Wang [17]. Later, an improved directional ant colony optimization method has been proposed by Geng, Weng and Liu [16] for non linear resource leveling optimization problem. Particle swarm optimization has been developed for resource constrained project scheduling problems [18]. The Genetic Algorithm (GA) technique based method and other meta-heuristic methods provide a generalized and robust approach to resource leveling as they offset the limitations imposed by the earlier heuristic methods. The earliest GA technique based model dates back to year 1996 developed by Chan, Chua et al. [19]. Other GA methods include models developed by Hegazy [20], Leu and Yang [21], Senouci and Eldin [22] and Leu, Yang et al. [11]. El-Rayes and Jun [3] assert that the metrics used for reducing resource fluctuations by these GA based methods and other previous heuristic methods tend to favor a predetermined resource profile. This implies that these methods do not consider various other resource profiles which may generate schedules with more efficient resource consumption.

Therefore, a set of new metrics have been developed by El-Rayes and Jun [3] to induct generation of optimal solutions which are impartial to a predetermined resource profile shape. These new metrics will be used in this proposed research. However, the GA technique to be developed will be improved by employing two different precedence-preserving crossover and mutation operators from the literature. But, before analyzing these two different approaches, a brief background on Genetic Algorithms is necessary to contemplate its application.

### 2.3 Introduction to Genetic Algorithms (GA's)

Genetic Algorithms are inspired by the Darwin's theory of natural selection and the "survival of the fittest" principle through the process of computational evolutionary programming which tries to obtain optimal or near-optimal solution [23].

The cells of living organisms consist of same set of chromosomes. These chromosomes contains blocks of DNA (Deoxyribonucleic acid) called as genes. During the process of reproduction, crossover and mutation of parents' genes occur to form a new offspring. This biological background is the basis for Genetic Algorithm development. The initial population for the Genetic Algorithm is a set of solution (represented by chromosomes). New population (offspring) is created by reproduction, crossover and mutation of the current population (parents) with a hope that the new population will yield better solution. Therefore, new solution is generated according to the fitness parameter of parents chromosomes i.e., the better they are the more probability they have to reproduce. This whole process of reproducing new offspring is repeated until a specified condition is accomplished[24].

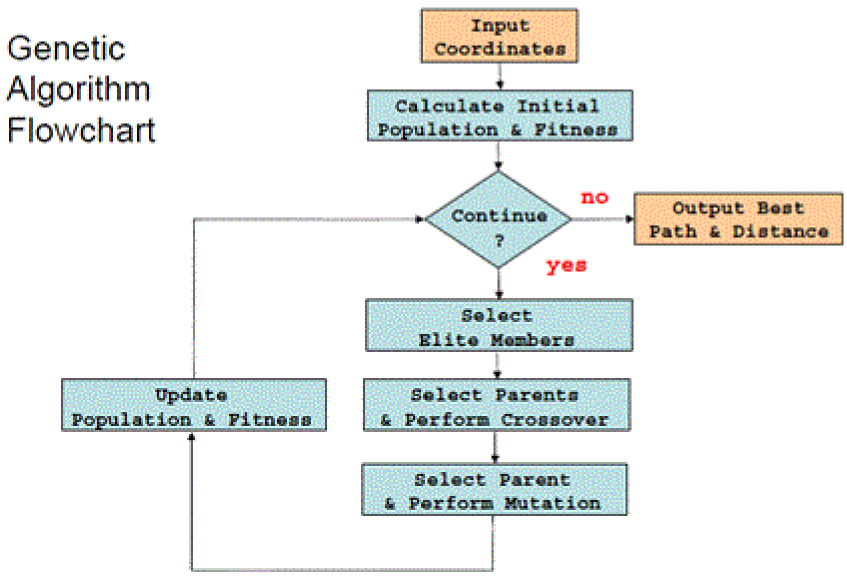



**Figure 2-1** GA Flowchart [25]

The conceptual model of Genetic Algorithm can be applied for resource leveling problems of construction project as it provides a general optimization method which is free from drawbacks of heuristic approaches [11]. As already mentioned, this research aims to improvise the already existing GA based methods by fostering holistic resource leveling. The proposed GA model will have the following advantages over the existing ones: better quality solutions; ease of application in solving complex construction project schedules; and holistic resource leveling by investigating optimal solution beyond the original duration.

# CHAPTER 3 RESEARCH METHODOLOGY

The following are the steps necessary to develop the proposed model:

1. Construct the daily resource utilization histogram.
2. Identify the metrics to measure and reduce undesirable resource fluctuation.
3. Develop two GA systems which employ the two different precedence-preserving operators.
4. Code the GA systems using MATLAB.
5. Compare the results of the two GA systems.
6. Validation of the model

An overview of the two models while analyzing the above steps related to these models is discussed in this chapter. The complete procedure for both these approaches is explained with the help of a small example in the next chapter.

## 3.1 Construct the daily resource utilization histogram

The daily consumption of resource can be calculated by summing up the number of resource units consumed by the activity/ activities during a particular day. A bar chart can be prepared by plotting the number of resources consumed per day over the duration of the project. The developed bar chart will represent the daily resource utilization histogram. The profile of the daily resource usage of a schedule shows the variation of how the resource is being utilized, and the aim of resource leveling is to minimize this variation. While coding the genetic algorithm in MATLAB software, daily resource histogram for the early schedule & the final optimized schedule will be generated.

## 3.2 Identify the metrics to measure and reduce undesirable resource fluctuation

In order to reduce the undesirable fluctuations in resource, two new resource leveling metrics have been developed by El-Rayes and Jun [3]. These are: (1) Release and re-hire (RRH) and (2) Resource Idle Days (RID).

Release and re-hire (RRH): This metric is to be employed as the fitness function for Genetic Algorithm when the construction project allows release and re-hire of resources (workers). RRH metric is calculated systematically by the formulae described by El-Rayes and Jun [3]. It measures the sum total number of resources that need to be temporarily released during low demand periods and re-hired during high demand periods. If the objective is also to reduce the Maximum Resource Demand (MRD), optimization function to be used in GA can be modified by incorporating MRD metric using equation (1). The factors $W_1$ and $W_2$ in the equation represent the planner defined weights or relative importance for RRH and MRD metrics respectively.

Optimization function to minimize resource fluctuation and peak demand:

**Min( $W_1$ RRH + $W_2$ MRD)……………………………………………………(Equation 1)**

Resource Idle Days (RID: This metric is to be employed as the fitness function for Genetic Algorithm when the construction project requires keeping the additional resources idle on site during low demand periods because release and rehire of resources is restricted. RRH metric is calculated by the formulae described by El-Rayes and Jun [3]. It measures the total number of idle and nonproductive resource days.

If the objective is also to reduce the peak demand value (MRD), optimization function to be used in GA can be modified by incorporating MRD metric using equation (2). The factors $W_1$ and $W_2$ in the equation represent the planner defined weights or relative importance for RID and MRD metrics respectively.

Optimization function to minimize resource fluctuation and peak demand:

**Min( $W_1$ RID + $W_2$ MRD)…………………………………………………..(Equation 2)**

During the execution of the GA, the user will have the option to choose between the RRH & RID metric. Further, the user can specify the factors: '$W_1$' & '$W_2$'.

### 3.3 Develop two GA systems which employ the two different precedence-preserving operators

The evolution process of the GA technique in the domain of resource leveling is subject to yield offspring chromosomes wherein the precedence relationships are violated. Hence, checking the generated chromosomes and repairing the infeasible chromosomes are required which cause computational inefficiency of the GA technique. To avoid this problem, two different precedence-preserving crossover and mutation operators differing from the literature will be employed. One of these methods encodes activities' start times and the other method encodes activities' shift values.

#### 3.3.1 Precedence-Preserving GA operators with Activities' Start Times Encoding

This method was originally developed by Abido and Elazouni [26] for finance based scheduling. The same procedure will be used for crossover and mutation for resource leveling of schedules in this model. The chromosomes structure consists of a string of genes and each chromosome represents one possible schedule. The number of genes in a chromosome will be equal to the number of activities and the gene values will represent the start times of the corresponding activities. The encoding of genes and employment of crossover and mutation operators involves the following: initial population generation; reproduction; employing crossover operator and employing mutation operator.

Initial Population Generation: The activities are arranged in a network with sequence steps [4]. All the activities are then randomly assigned their start time such that assignment of start times of any activity of a particular sequence step is not started unless and until all the activities of the previous sequence step are not finished. The range within which an activity start time is randomly assigned is to be governed such that there is no violation of precedence relationships [26]. The total floats of all activities must be adjusted if extension of the project duration is specified. In case of any resource constrains, if the resource leveling of existing total duration of project schedule does not suffice the specified resource limitation criteria, the relative importance factor for peak resource demand is either increased or the project schedule is extended and the limitation for maximum resource demand is specified.

Reproduction of Chromosomes: The reproduction of chromosomes implies choosing chromosomes for next generation based on their fitness parameter. The fitness parameter assigned to each chromosome is defined by Roulette Wheel Selection i.e., lower the optimization function for a chromosomes, higher the percentage it has to be selected in next generation. Infeasible chromosomes will be assigned zero percentage and thus, such chromosomes will be discarded in the next generation. After the percentages of selection are given for every chromosome, a random number is generated in the range of sum of percentage selections of all chromosomes. The random number which corresponds to a particular chromosome will be selected. This procedure will be described in the next chapter comprehensively.

Employing Crossover Operator: The crossover of two parents' chromosomes is done by single-point cut approach i.e., two chromosomes will be cut randomly at a single point between the genes and swapped to create child chromosomes. However, the child chromosomes created will violate precedence relationships. In order to avoid this problem, an algorithm has been proposed by Abido and Elazouni [26] which forms child chromosomes randomly by either the forward path or backward path approach.

Employing mutation operator: To avoid the problem of violating precedence relationships, a randomly chosen activity start time (gene) in a chromosome is to be selected and mutated such that the replaced value must lie within the range of the forward free float (FFF) and backward free float (BFF) Abido and Elazouni [26].

### 3.3.2 Precedence-Preserving GA operators with Activities' Shift Values Encoding

The method was originally developed by Demeulemeester and Herroelen [27]. It has been used by AlGhazi [28] for finance based scheduling. The same procedure for crossover and mutation will be used for resource leveling of schedules in this model. The chromosomes structure consists of a string of genes and each chromosome represents one possible schedule. The number of genes in a chromosome will be equal to the number of activities and the gene values will represent the shift values of the corresponding activities. The shift value of an activity represents the difference of its start time with the highest finish time among all its predecessors or the activities' early start (minimum of the two). The encoding of genes and employment of crossover and mutation operators involves the following: initial population generation; reproduction employing crossover & mutation operators; and repair chromosome.

Initial Population generation: A pseudo code of the algorithm [28] will randomly generate start times of all the activities sequentially and then convert them into shift values to represent all the genes of the parent chromosome.

Employing Reproduction, Crossover and Mutation operators: The reproduction of chromosomes method is same for both the models. The crossover of two parents' chromosomes is done by single-point cut approach i.e., two chromosomes will be cut randomly at a single point between the genes and swapped to create child chromosomes.

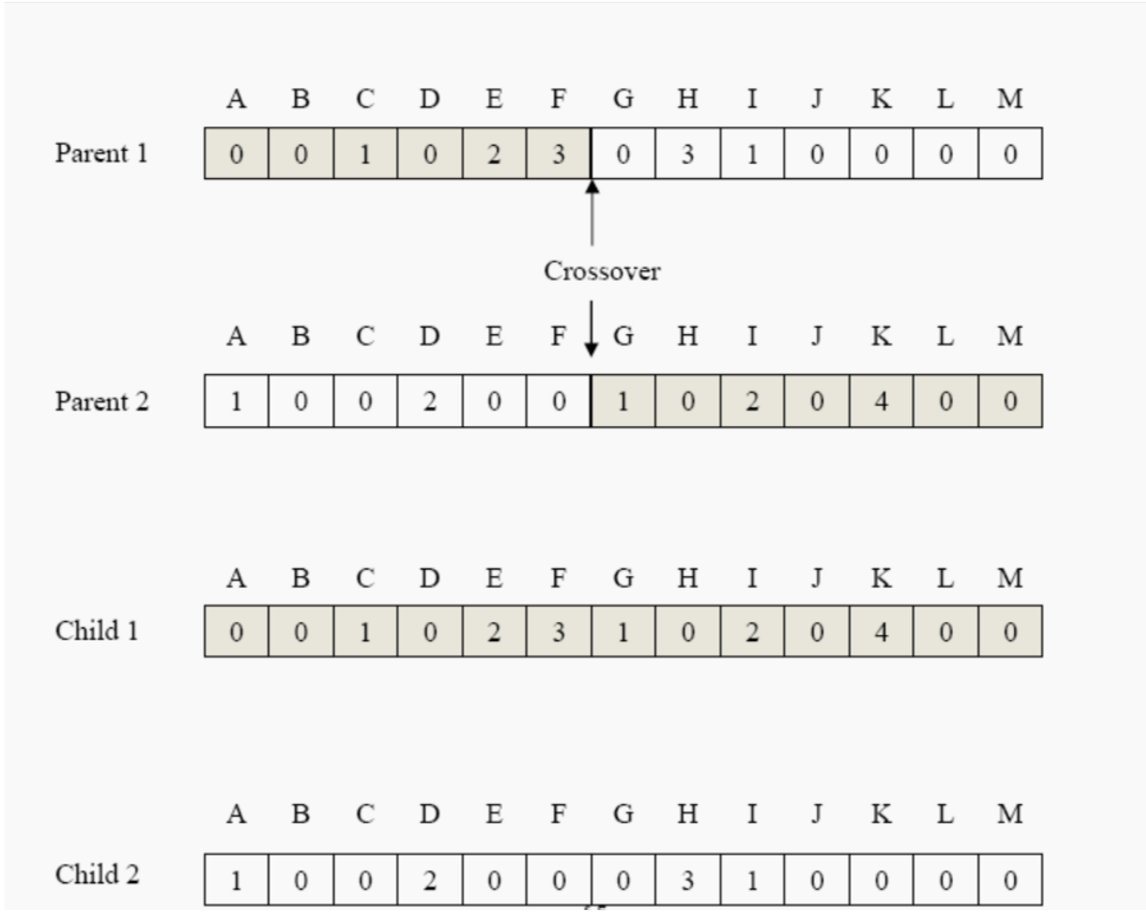



**Figure 3-1** Sample crossover of two chromosomes comprising of 13 activities (genes) [28]

Mutation is done to ensure a diverse population to avoid premature convergence [28]. Based on the mutation probability, a chromosome is mutated by randomly altering value of one gene in a chromosome. No alteration will be done to chromosomes after crossover and mutation as activities are being represented by shift values. The code which decodes the chromosomes to its early start times will handle the relationship violations, if any.

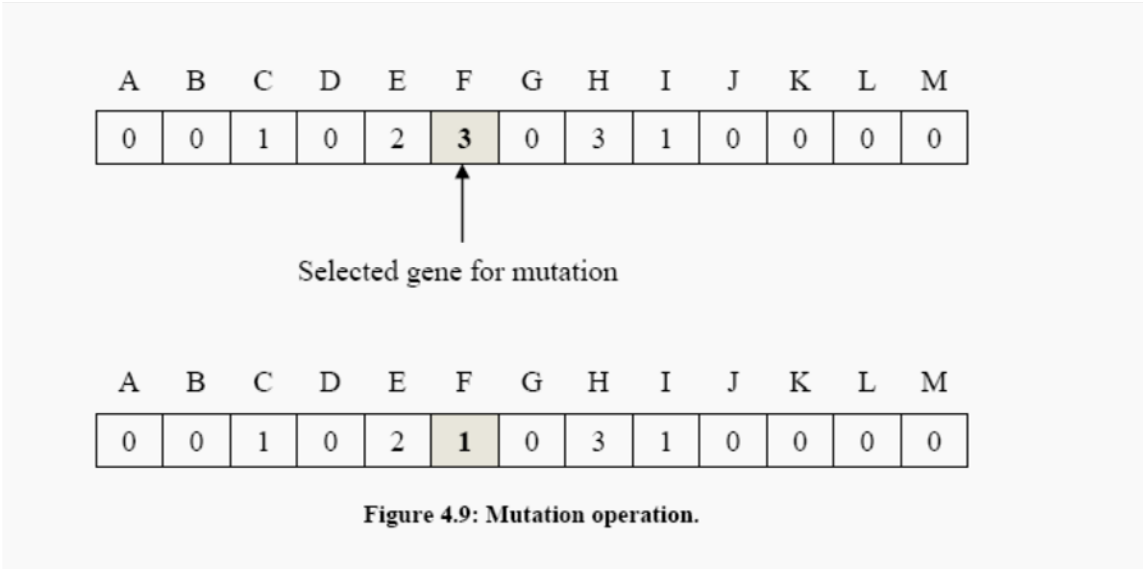



**Figure 3-2** Sample mutation of chromosomes comprising of 13 activities (genes) [28]

Decode chromosome: To avoid the problem of violating precedence relationships, a pseudo code is proposed has been developed to decode the chromosome [28]. The decoded chromosome will represent start times of the activities.

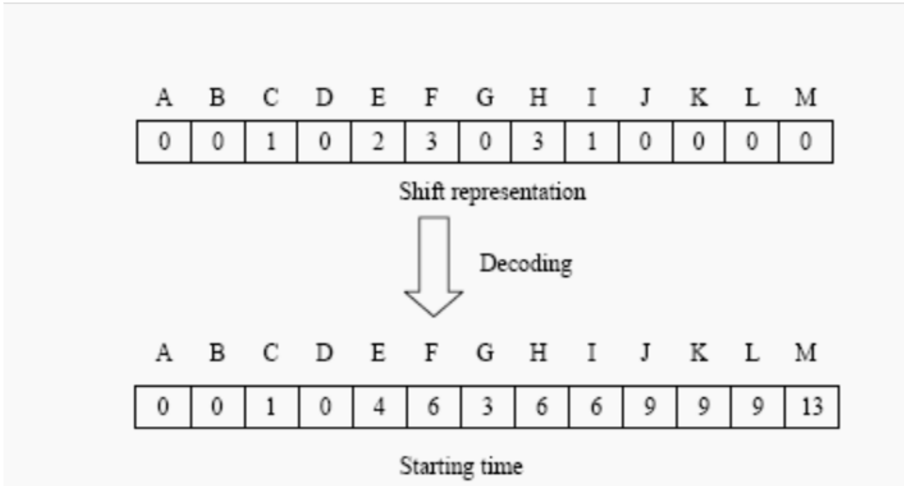



**Figure 3-3** Sample 'shift representing' chromosome decoded to show start times [28]

Elitism: During the creation of every new generation, a single chromosome representing the lowest optimal value is kept during the reproduction process to save the best chromosome in every generation. This is called elitism and it is carried out for both the methods.

## 3.4 Code the GA systems using MATLAB

The model for resource leveling using improved GA is to be developed in MATLAB, as it's a high performance language for technical computing and provides a specialized GA toolbox. An Initialization code is generated to accept the ID's, predecessors, duration and number of resource units for all activities of a schedule from the user. After accepting these values, it will calculate all floats and create resource histogram for early schedule. Then, it will prompt to specify the number of days by which the total project duration can be extended. It will also ask the user to specify the metric to be used for optimization: RRH or RID and the relative importance factors for optimization function and maximum resource demand. Further codes are developed separately for initial population generation, reproduction, crossover and mutation based on the procedure described for both models.

Integration of all codes: All the codes are tested separately so as to assess whether each of the code perform its functionality as necessary. After this procedure, all these codes will be integrated separately for each method so that it can perform as a single unit. The objective to write one single program for the whole GA process is to develop a single standalone application for resource leveling for the ease of user.

### 3.5 Compare the results of the two GA systems

The performance of the GA techniques employing the two different precedence-preserving crossover and mutation operators will be compared by evaluating their impact as measured by the quality of solutions and computation cost. The best among the two improved GA techniques will be proposed by using resource leveling problems as a bed to compare between the performances of the two precedence-preserving operators.

### 3.6 Validation of the model

Easa [1] proposed an integer-linear optimization model which produces exact solutions for resource leveling. A small project schedule will serve as a basis to validate the proposed model. The result of the completely optimized solution produced by Easa [1] has been used to verify the results of the proposed model. Two cases of optimal solutions are reported by Easa [1]: one being the optimal solution obtained for minimizing duration for a Uniform Histogram, and the other being the optimal solution obtained for minimizing daily resource variations. The solution obtained by first case will not correspond to the proposed Genetic Algorithm model as the GA model is impartial to a particular pre-determined resource profile (uniform histogram). The second case solution represents a better and improved solution and the solution obtained from the GA model should correspond to this solution. This is demonstrated by a five-day activity network[1] represented as an arrow diagram. A single ending activity is required for proper calculations of floats, and hence, an "end" activity having zero duration and null resources is added to the original five-day network. The precedence diagram of this network is shown. The resulting graph obtained by applying this proposed GA algorithm using both the models correspond to the solution obtained in [1] for the case of minimizing daily resource variations. Hence, the proposed model is verified.

Optimization Function: RRH/RID

Crossover Percentage: 80% & Mutation Percentage: 90%

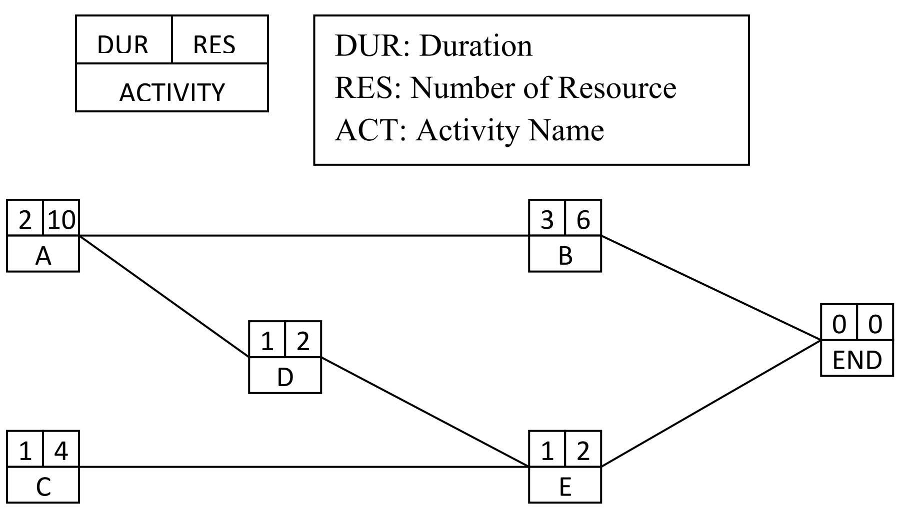



**Figure 3-4** Precedence Diagram of the Network in [1]

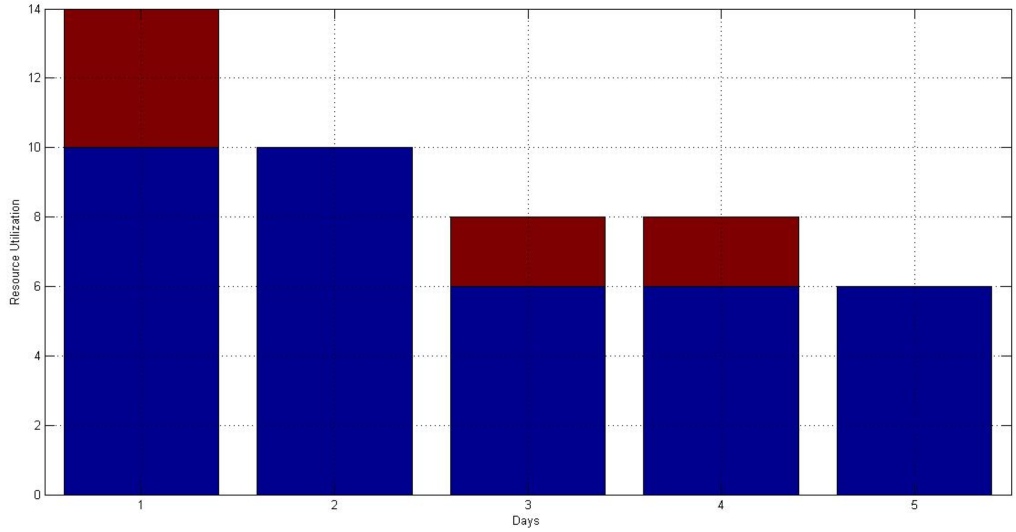



**Figure 3-5** Early Start Schedule of the Network

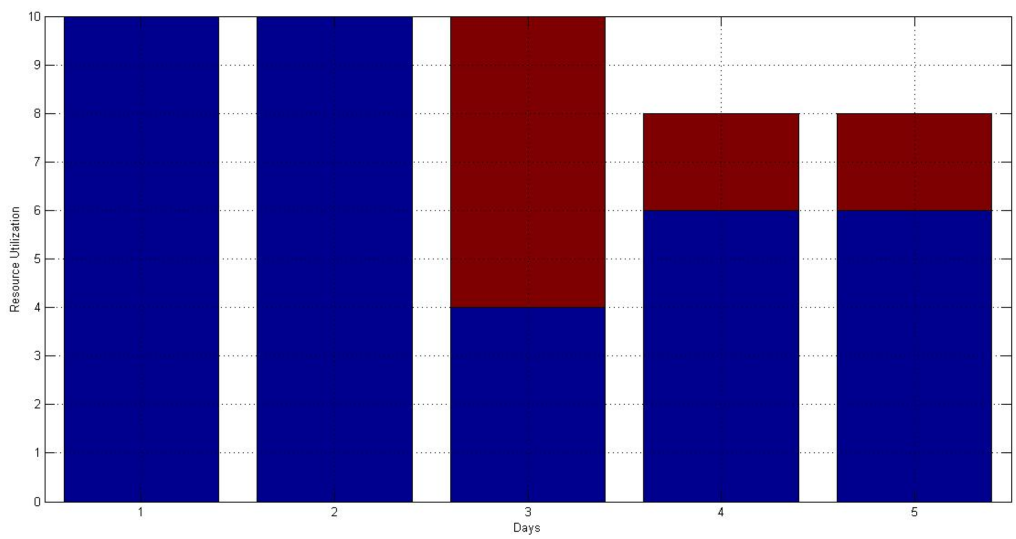



**Figure 3-6** Optimal resource histogram obtained by both GA models which corresponds to solution by Easa [1]

# CHAPTER 4 EXECUTION OF GENETIC ALGORITHM

The models created in MATLAB environment is examined in this chapter. The two models are separately investigated with the help of a small network consisting of 6 activities taken from El-Rayes and Jun [3]. The following flowchart represents the step-by-step execution of GA for both the models.

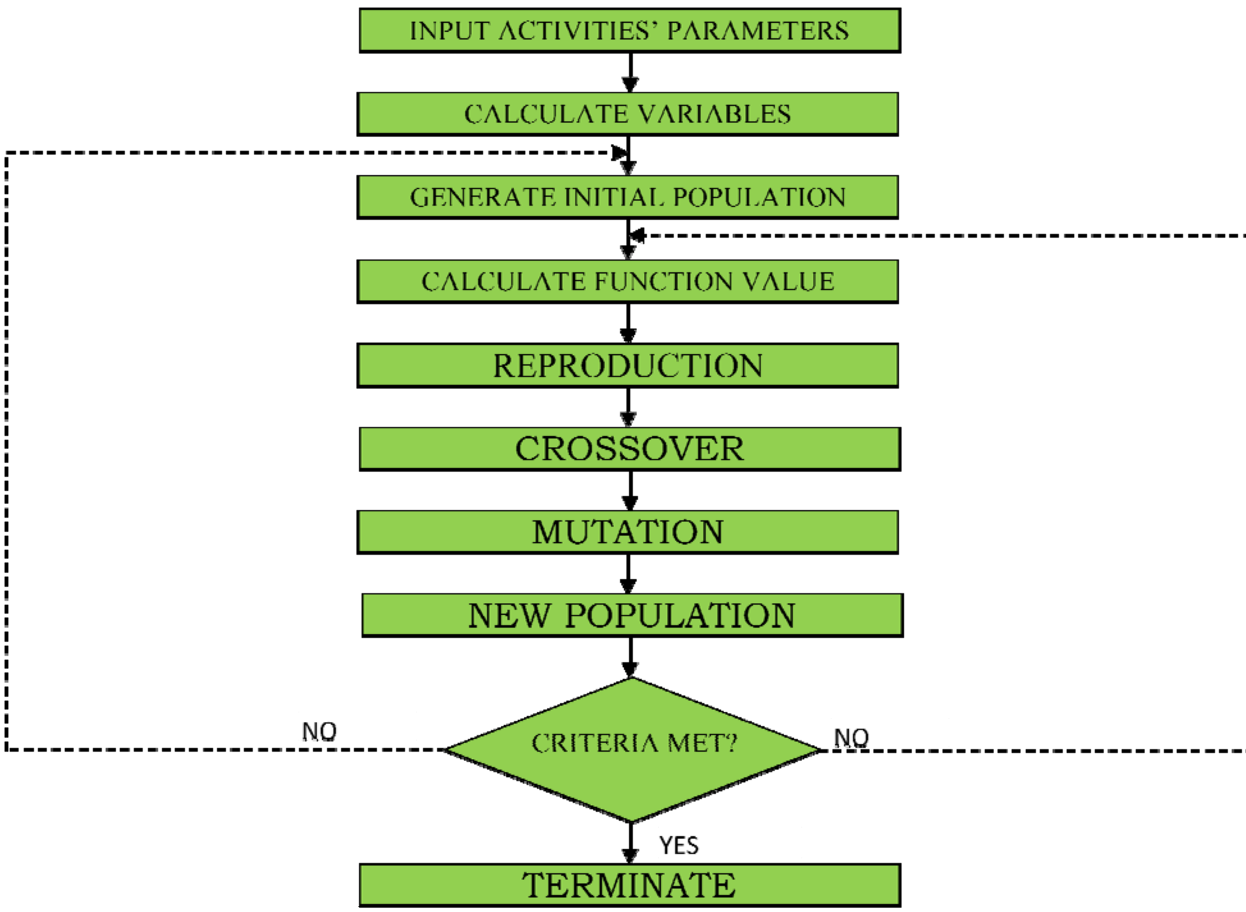
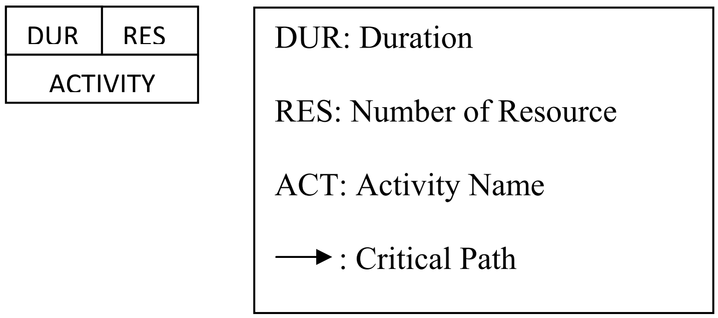



**Figure 4-1** Flowchart of the Genetic Algorithm

## 4.1 MODEL A – START TIMES ENCODED GA

The Genetic Algorithm for model A encodes the genes of its chromosomes as the start times of all activities. The complete procedure is illustrated with the help of the six activity schedule given below[3]:



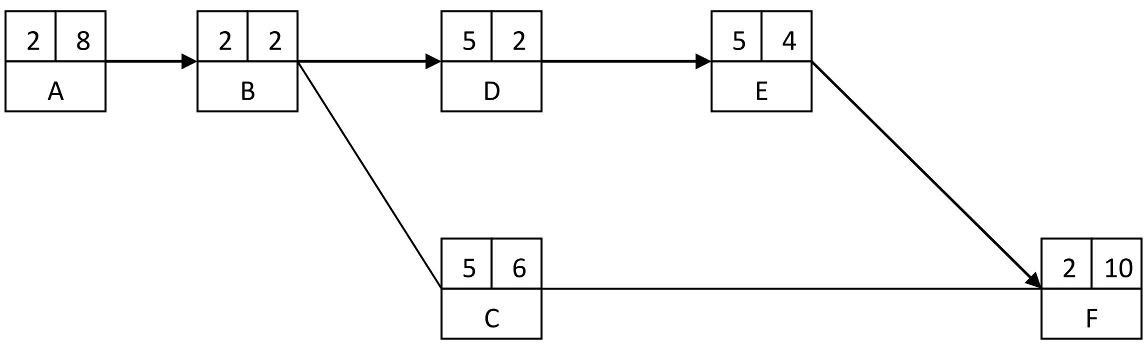




**Figure 4-2** Precedence Diagram of the 6 activity network

When the Genetic Algorithm is executed in MATLAB software, the user is prompted to input the following details of the network: Activity ID, Duration, Number of resources, Number of predecessors & the Predecessors of each activity. The table shown below is the input to be given for the above network.

**Table 4-1** Input for the Network

| Sl. No: | Activity ID | Duration | No. of resources | No. of Predecessors | Predecessors |
|---|---|---|---|---|---|
| 1 | A | 2 | 8 | 0 | - |
| 2 | B | 2 | 2 | 1 | A |
| 3 | C | 5 | 6 | 1 | B |
| 4 | D | 5 | 2 | 1 | B |
| 5 | E | 5 | 4 | 1 | D |
| 6 | F | 2 | 10 | 2 | C,E |

The code of the genetic algorithm arranges the activities entered in the sequential order. It also calculated the successors of each activity and stores in a matrix such that each column represents the successors of the activity corresponding to the single row matrix which stores all the activities. The activities arranged in sequence and their corresponding successors for the given network are represented in the MATLAB environment as:

```
ACT =
    'A'    'B'    'C'    'D'    'E'    'F'
```

ACT - The Activity ID matrix

```
succ =
    'B'    'C'    'F'    'E'    'F'    []
     []    'D'    []     []     []     []
```

succ – The successor matrix which corresponds to the activity ID matrix

SOA =

'A' 'B' 'C' 'E' 'F'

[] [] 'D' [] []

SOA - The Sequence Of Activities matrix

[] - Null matrix

The sequence step arrangement for the network can be graphically shown as[4]:

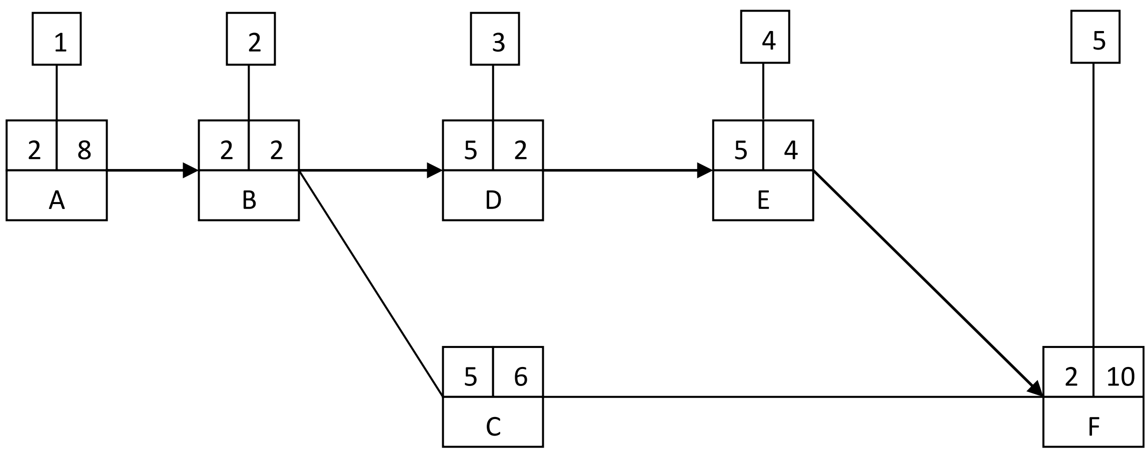



**Figure 4-3** Sequence Step Arrangement of the 6 activity network

After arranging the activities in sequence, the user is prompted to input if it likes to extend the project duration. If yes, it seeks the number of days the total project is to be extended. The user may not necessary extend the project duration initially. After analyzing the results obtained from the GA results based on the initial fixed total duration, the user is prompted again to extend the duration of project for another GA run, if the results from the initial GA run seem unsatisfactory to the user.

The GA then calculates early start, early finish, late start & late finish times. It also calculates total float and free float values. If the total project duration is extended, the total float and free float values will then represent the extended total float and free float values respectively. The matrix for early start(ES), early finish(EF), late start(LS), late finish(LF), total float(TF) & free float(FF) values for the network being analyzed in the MATLAB software are represented as the following:

```
ES =

    1     3     5     5    10    15

EF =

    3     5    10    10    15    17

LS =

    1     3    10     5    10    15

LF =

    3     5    15    10    15    17

TF =

    0     0     5     0     0     0

FF =

    0     0     5     0     0     0
```

Before generating the initial population, the user enters the size of population. Generally, higher the size of population, higher is the number of chromosomes generated and thus, higher is the variability of chromosomes. For example, if the size of population entered is ten, an equal number of chromosomes will be generated. Each chromosome generated will represent a different schedule constituting different characteristics and the sum matrix of all these chromosomes generated represents the Initial Population. The initial population generated for the six activity network is represent by the matrix ST as shown below:

Size of Population: 10

ST =

| | | | | | |
|---|---|---|---|---|---|
| 1 | 3 | 9 | 5 | 10 | 15 |
| 1 | 3 | 7 | 5 | 10 | 15 |
| 1 | 3 | 6 | 5 | 10 | 15 |
| 1 | 3 | 5 | 5 | 10 | 15 |
| 1 | 3 | 5 | 5 | 10 | 15 |
| 1 | 3 | 6 | 5 | 10 | 15 |
| 1 | 3 | 9 | 5 | 10 | 15 |
| 1 | 3 | 5 | 5 | 10 | 15 |
| 1 | 3 | 10 | 5 | 10 | 15 |
| 1 | 3 | 6 | 5 | 10 | 15 |

The metric to be used for optimization is an option for the user. If the user wants to level the resources with the objective of releasing the additional construction resources and rehire them at a later stage, the metric RRH (Release and re-hire) must be chosen by the user. However, if the user wants to level the resources with the objective of retaining the idle construction resources on site until needed, the metric RID (Reduce Idle Days) must be chosen by the user. The metric RRH is given by equation (3) which is based on equation (4) & (5):

$$\mathbf{RRH = H - MRD = \tfrac{1}{2} \times HR - MRD} \quad \text{……………………………………}\mathbf{Equation\ (3)}$$

$$\mathbf{HR = (r_1 + \sum_{t=1}^{T-1} |\, r_t - r_{t+1}| + r_T\ )} \quad \text{………………………….……………..}\mathbf{Equation(4)}$$

$$\mathbf{MRD = Max\ (r_{1,}\ r_2 ... r_T)} \quad \text{…………………….........................................}\mathbf{Equation\ (5)}$$

The metric RID is calculated using equation (6):

$$\mathbf{RID = \sum_{t=1}^{T-1} [Min\ \{Max\ (r_{1,}\ r_2 ... r_T),\ Max\ (r_{1,}\ r_2 ... r_T)\ \} - r} \quad \text{……………..}\mathbf{Equation\ (6)}$$

Where the parameters are:

HR - total daily resource fluctuations (it sums up all increases and decreases in the daily resource demand)

H – Total increases in the daily resource demand

T - Total project duration

$r_t$ - resource demand on day (t)

$r_{t+1}$ – resource demand on day (t+1)

MRD - Maximum resource demand during the entire project duration

The optimization function value for all the chromosomes in the initial population are calculated using the above equations. For the given network, the optimization function value for both the cases, RRH & RID for 10 chromosomes of Initial Population is given in the table below:

**Table 4-2** Optimization Function Values for Initial Population

| **OPT ( if RRH is chosen)** | **OPT ( if RID is chosen)** |
|---|---|
| 12 | 42 |
| 12 | 42 |
| 12 | 42 |
| 10 | 32 |
| 10 | 32 |
| 12 | 42 |
| 12 | 42 |
| 10 | 32 |
| 6 | 42 |
| 12 | 42 |

The user may specify the relative importance factor to maximum resource demand ($W_2$) if the objective is also to reduce the peak demand of the resource histogram. If the user specifies this factor, the function matrix to be optimized, represented by OPT is given by equation (1) & (2) depending upon the optimization function chosen, RRH or RID respectively. If $W_2$ is not specified, then the OPT value or optimization function matrix will be RRH or RID matrix shown in the above table depending upon the optimizing function RRH or RID chosen.

After calculating the optimization function value for each chromosome in the initial population, the genetic algorithm assigns the relative importance to each chromosome by assigning the chromosome corresponding to least value the highest weighted factor and the chromosome corresponding to the highest value the least weighted factor. The percentages for each chromosome reflect the probable percentage selection of that chromosome in the next generation. Assuming the chosen optimization factor is RRH, the percentage selection of each chromosome (PER) and the cumulative sum of percentages (cumPER) for the ten chromosomes generated in the initial population for the network is given in the table below:

**Table 4-3** Percentage Weighted Factor & their Cumulative Sum

| PER (Percentage Weighted factor) | cumPER (cumulative sum of percentage weighted factor) |
|---|---|
| 9.0000 | 9.0000 |
| 9.0000 | 18.0000 |
| 9.0000 | 27.0000 |
| 10.8000 | 37.8000 |
| 10.8000 | 48.6000 |
| 9.0000 | 57.6000 |
| 9.0000 | 66.6000 |
| 10.8000 | 77.4000 |
| 18.0000 | 95.4000 |
| 9.0000 | 104.4000 |

For example, the weighted factor for first chromosome is calculated by dividing the inverse of its RRH value, 12 (Table 2, Column 1, Row 1) with the inverse cumulative sum of RRH matrix, 108 (Table 2, Column 1). Therefore, percentage weight for first chromosome is:

(1/12) / (1/108) = 9.0000 as shown (Table 3, Column 1, Row 1)

The selection of chromosome by reproduction is based on the roulette wheel selection. This selection is to be understood by visualizing the roulette of a casino. The roulette wheel in casinos has sector of angles divided equally for all numbers. However, each chromosome selection in the genetic algorithm corresponds to the fixed sector it occupy relative to the percentage weighted factor calculated.


**Figure 4-4** Roulette Wheel of a Casino

(http://www.rouletteexperts.net/roulette_blog.jpg )

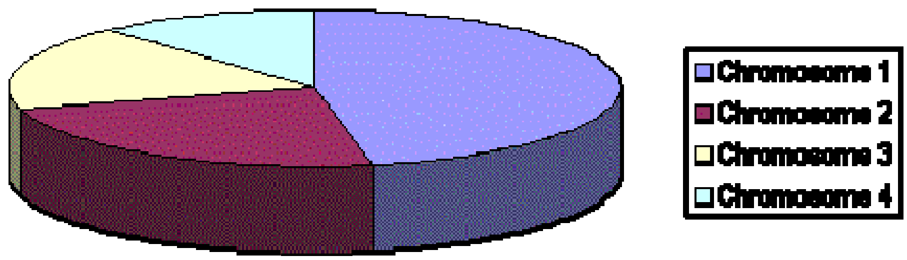



**Figure 4-5** Roulette Wheel for Chromosome Selection (Unequal Sectors) [24]

The roulette wheel for the network being examined, if RRH metric is the optimization function, can be graphically shown as (where 1-10 represent ten chromosomes of initial population):

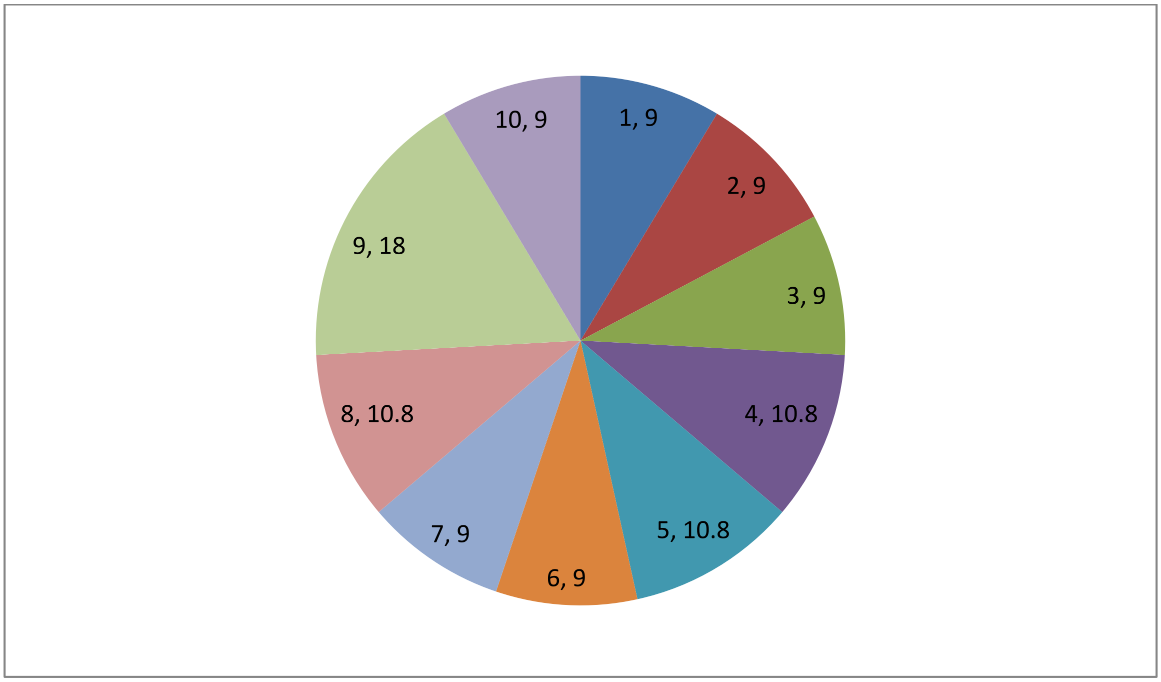



**Figure 4-6** Chromosome Weighted Percentages on Roulette Wheel for Initial Population (Chromosome Number, Percentage Weight)

The selection of chromosome by reproduction in next chromosomes depends on these relative weighted percentages on roulette wheel. Similar to spinning a wheel on a casino roulette which randomly stops on any given number, a random number is generated between 0 to the cumulative sum of percentage weights (104.4). Every chromosome in the next generation to be reproduced is created by choosing a random number, and selecting that chromosome which corresponds to the random number falling in its range. The range for each chromosome is between the previous chromosome cumulative percentage weight and its own cumulative percentage weight. For example, chromosome 2 has range from 9 to 18. Hence, if the random number generated is in-between this range (for example – 12), then this chromosome will be selected. For the first chromosome, the range is between 0 to its percentage weight. If the size of population is N, the random number is generated N-1 times for filling N-1 spaces of reproduction chromosome matrix. The last chromosome, N, is filled with the chromosome corresponding to the least value of the initial population (to consider Elitism). The range for all the ten chromosomes in the network is given below:

**Table 4-4** The Chromosome Number & Its Range

| **Chromosome Number of Initial Population** | **Range within which the random number must come for it to be selected** |
| --- | --- |
| 1 | 0 - 9 |
| 2 | 9 - 18 |
| 3 | 18 - 27 |
| 4 | 27 - 37.8 |
| 5 | 37.8 - 48.6 |
| 6 | 48.6 - 57.6 |
| 7 | 57.6 - 66.6 |
| 8 | 66.6 - 77.4 |
| 9 | 77.4 - 95.4 |
| 10 | 95.4 - 104.4000 |

These are the random numbers generated and chromosome chosen for the given network:

**Table 4-5** Random Number Generated & the Corresponding Chromosome Selected

| **Random Number Generated** | **Chromosome Selected from Initial/ Parent Population** |
|---|---|
| 27.9737 | 4 |
| 45.9281 | 5 |
| 97.4449 | 10 |
| 71.3399 | 8 |
| 22.1912 | 3 |
| 87.6165 | 9 |
| 65.6451 | 7 |
| 13.9659 | 2 |
| 21.6247 | 3 |
| - | **9** (Elitism – chromosome corresponds to least RRH value of 6) |

Thus, the reproduction chromosome matrix, RCR, is represented based on the above selection in MATLAB environment as:

RCR =

| | | | | | |
|---|---|---|---|---|---|
| 1 | 3 | 5 | 5 | 10 | 15 |
| 1 | 3 | 5 | 5 | 10 | 15 |
| 1 | 3 | 6 | 5 | 10 | 15 |
| 1 | 3 | 5 | 5 | 10 | 15 |
| 1 | 3 | 6 | 5 | 10 | 15 |
| 1 | 3 | 10 | 5 | 10 | 15 |
| 1 | 3 | 9 | 5 | 10 | 15 |
| 1 | 3 | 7 | 5 | 10 | 15 |
| 1 | 3 | 6 | 5 | 10 | 15 |
| 1 | 3 | 10 | 5 | 10 | 15 |

It is to be observed that all the reproduced chromosomes come from the parent chromosome i.e., no new chromosome has been generated. In order to introduce variability or generate chromosomes of new characteristics representing new & different schedule, crossover and mutation of chromosomes is necessary. Crossover is done between two chromosomes while mutation occurs on a single chromosome. The user specifies the percentage of crossover and mutation. However, these percentages should lie within a range such that elitism is not discarded i.e., the best chromosome in every generation is not lost. For example, if the size of the population is 10 as given in the network, the maximum crossover and mutation percentage to be given is 90%. However, if crossover percentage is given 90%, it is taken as 80% as even number of chromosomes crossover. If mutation percentage is given 90%, all the 9 mutate, thus, saving the last best chromosome.

The procedure for crossover and mutation is as described by Abido and Elazouni [26]. For crossover of two activities, the backward free float (BFF) & forward free float

(FFF) for both the chromosomes are calculated. FFF represents the normal free float i.e., the number of days an activity can be shifted forward without effecting the start times of its succeeding activities. BFF represents the number of days an activity can be advanced without effecting the finish times of its preceding activities[26]. Both the chromosomes are cut randomly at a certain point in-between. The new crossover chromosome to be generated can be formed either by applying forward path or backward path. If forward path is applied, the left part of the random cut point for both the chromosome remains same while the right part of the first chromosome is calculated by forwardly applying the BFF values of second chromosome and vice versa. If backward path is applied, the right part of the random cut point for both the chromosome remains same while the left part of the first chromosome is calculated by backwardly applying the FFF values of second chromosome and vice versa. The forward path or backward path to be used is chosen randomly by the genetic algorithm. For crossover percentage of 80%, i.e, 8 crossover of chromosomes for the 6 activity network, following was the set of crossover matrix obtained (SCR):

SCR =

| | | | | | |
|---|---|---|---|---|---|
| 1 | 3 | 5 | 5 | 10 | 15 |
| 1 | 3 | 5 | 5 | 10 | 15 |
| 1 | 3 | 6 | 5 | 10 | 15 |
| 1 | 3 | 5 | 5 | 10 | 15 |
| 1 | 3 | 10 | 5 | 10 | 15 |
| 1 | 3 | 6 | 5 | 10 | 15 |
| 1 | 3 | 7 | 5 | 10 | 15 |
| 1 | 3 | 9 | 5 | 10 | 15 |
| 1 | 3 | 6 | 5 | 10 | 15 |
| 1 | 3 | 10 | 5 | 10 | 15 |

In order to illustrate the crossover operation, consider fifth and sixth chromosome in the reproduction chromosomes matrix (RCR).

1 3 6 5 10 15
1 3 10 5 10 15

These two chromosomes have undergone crossover operation and the result is as obtained in SCR matrix:

1 3 10 5 10 15
1 3 6 5 10 15

The randomly selected point for these chromosomes was after the first gene. Forward path crossover was applied and swapping occurred. It can be illustrated by the given figure.

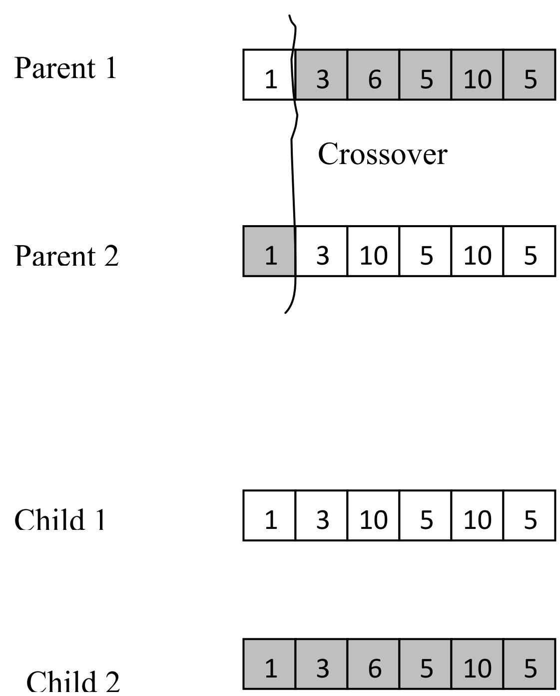



**Figure 4-7** Crossover of Chromosomes – Model A

The mutation process is also applied to produce new chromosomes which will exhibit variability. The BFF and FFF for the chromosome to be mutated are calculated. Then a random gene is selected in the chromosome is selected and mutated such that the activity is shifted forward or backward within the range defined by the FFF & BFF of that corresponding activity. If the random gene selected has 0 total float or the activity corresponding to that gene lies on critical path, than the mutated chromosome will remain same. The SCR matrix of the given network is mutated (mutation percentage – 90%) and the resulting set of mutated chromosome matrix, SMR in the MATLAB is:

SMR =

| | | | | | |
|---|---|---|---|---|---|
| 1 | 3 | 5 | 5 | 10 | 15 |
| 1 | 3 | 5 | 5 | 10 | 15 |
| 1 | 3 | 6 | 5 | 10 | 15 |
| 1 | 3 | 5 | 5 | 10 | 15 |
| 1 | 3 | 10 | 5 | 10 | 15 |
| 1 | 3 | 6 | 5 | 10 | 15 |
| 1 | 3 | 9 | 5 | 10 | 15 |
| 1 | 3 | 9 | 5 | 10 | 15 |
| 1 | 3 | 6 | 5 | 10 | 15 |
| 1 | 3 | 10 | 5 | 10 | 15 |

For example, the randomly generated gene for the seventh chromosome is third gene has both, BFF & FFF. It is shifted by 2 days forwardly by consuming FFF as illustrated.

| | | | | | |
|---|---|---|---|---|---|
| 1 | 3 | 7 | 5 | 10 | 5 |

Mutation of third gene

| | | | | | |
|---|---|---|---|---|---|
| 1 | 3 | 9 | 5 | 10 | 5 |

**Figure 4-8** Mutation of Chromosome – Model A

After reproduction, crossover and mutation; the set of mutated chromosomes SMR represents the parent chromosome for next generation. This new generation is again iteratively reproduced, crossover and mutated to yield a subsequent new generation and this process is continued until a repetitive optimal value is achieved for a certain number of times or the optimal value for the elite chromosome becomes ZERO.

After the number of iterations inputted by the user is finished, the user is prompted to execute one or more of the following operations by soliciting the response to the following questions: a) If the user wants to run more number of iterations? If yes, specify the number of generations to be created or iterations to be performed? b) If the user is not satisfied with the result, then it is prompted to run the GA again by creating a new Initial Population? c) If the user wants to vary the percentage weight $W_2$ for MRD to minimize or maximize the peak resource demand? AND d) If the user wants to extend the total project duration for minimizing peak resource demand or obtaining a better resource profile for cost benefits or other reasons?

After the above mentioned queries are satisfactory answered and the genetic algorithm is run multiple times, if required, the results will be generated by choosing the elite chromosome of last generation and drawing the resource histogram based on the start time values. The early start schedule based resource histogram is also displayed to compare the graphs. Thus, this is the complete procedure for Model A (start times encoded chromosomes method) and the execution of the genetic algorithm occurs as described in this section for every network. The early start schedule based resource histogram for the given network is as shown in the figure below:

Note: The alphabets represented on bars represent activity ID's & correspond to the resources consumed by that activity during that particular day.

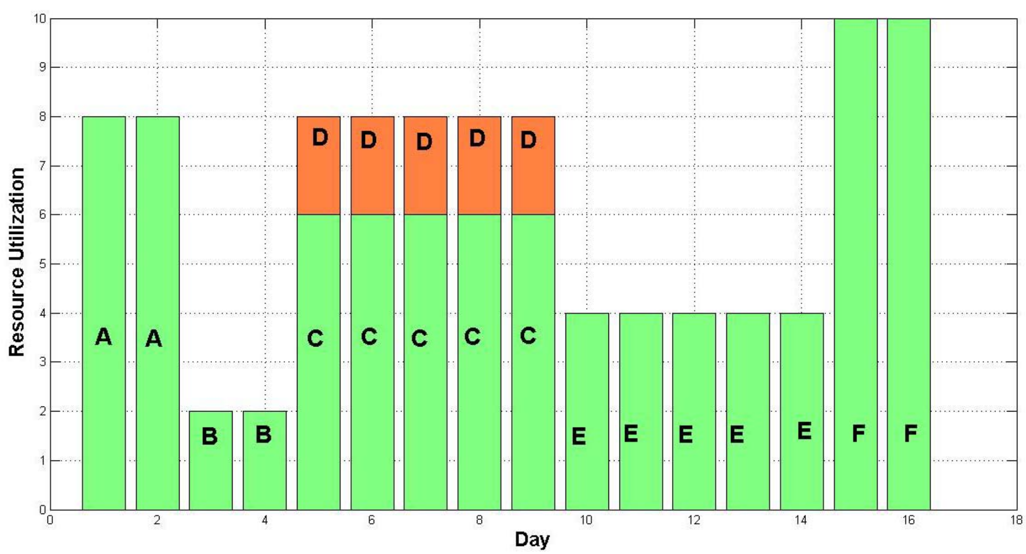



**Figure 4-9** Early-Start Schedule Resource Histogram

RRH Value: 12 & RID Value:32

If the optimization function chosen was RRH, the resource histogram for the network is as shown below (RRH = 6, RID = 42):

Optimal Value (RRH) - 6

Size of Population – 10

Crossover Percentage – 80%

Mutation Percentage – 90%

Number of Iterations performed - 10

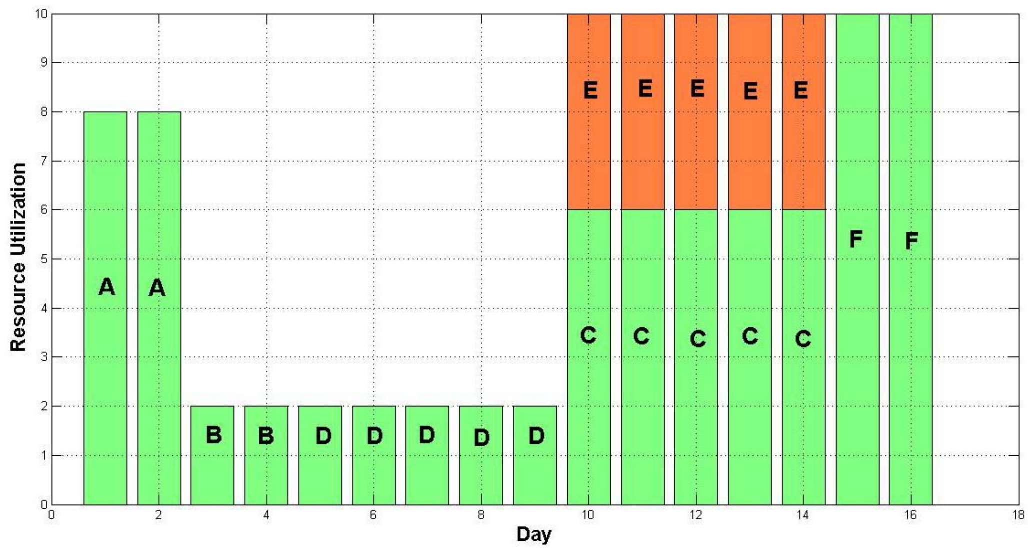



**Figure 4-10** Optimal Resourse Histogram If RRH Metric Is Used

If the optimization function chosen was RID, the resource histogram for the network is as shown below (RID = 32 , RRH = 10):

Optimal Value (RID) - 32

Size of Population – 10

Crossover Percentage – 80%

Mutation Percentage – 90%

Number of Iterations performed - 10

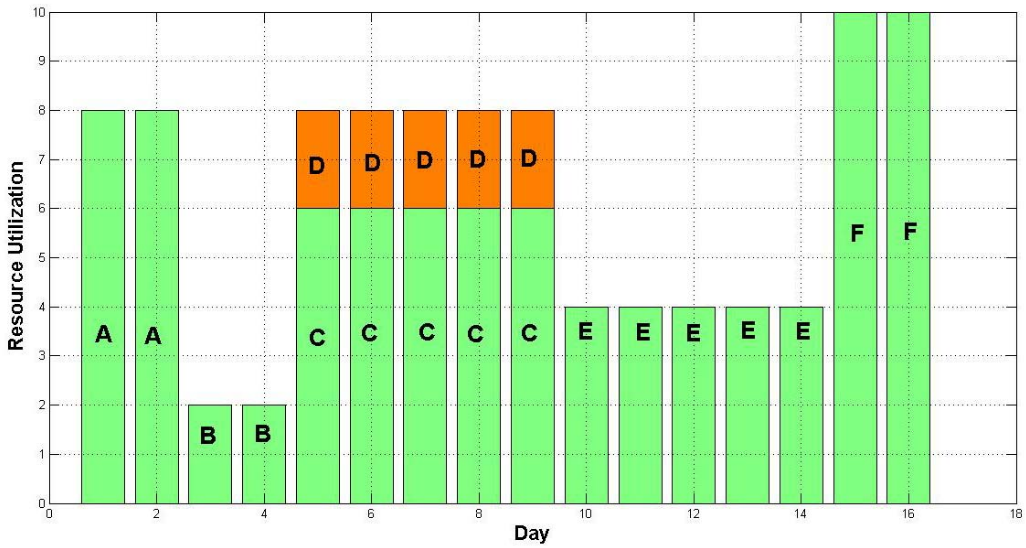



**Figure 4-11** Optimal Resourse Histogram If RID Metric Is Used

Hence, it is deduced that the optimal solution reached by the two metrics may be different. However, both the solutions represent optimal value and either can be used depending upon the project requirement.

## 4.2 MODEL B – SHIFT TIMES ENCODED GA

The genetic algorithm of this model encodes the genes of chromosomes as the shift values of the activities. In essence, the shift value of an activity represents the difference of its start time and the highest finish time among all its predecessors. The step-by-step algorithmic development of this model is demonstrated with the help of same six activity network used for Model A.

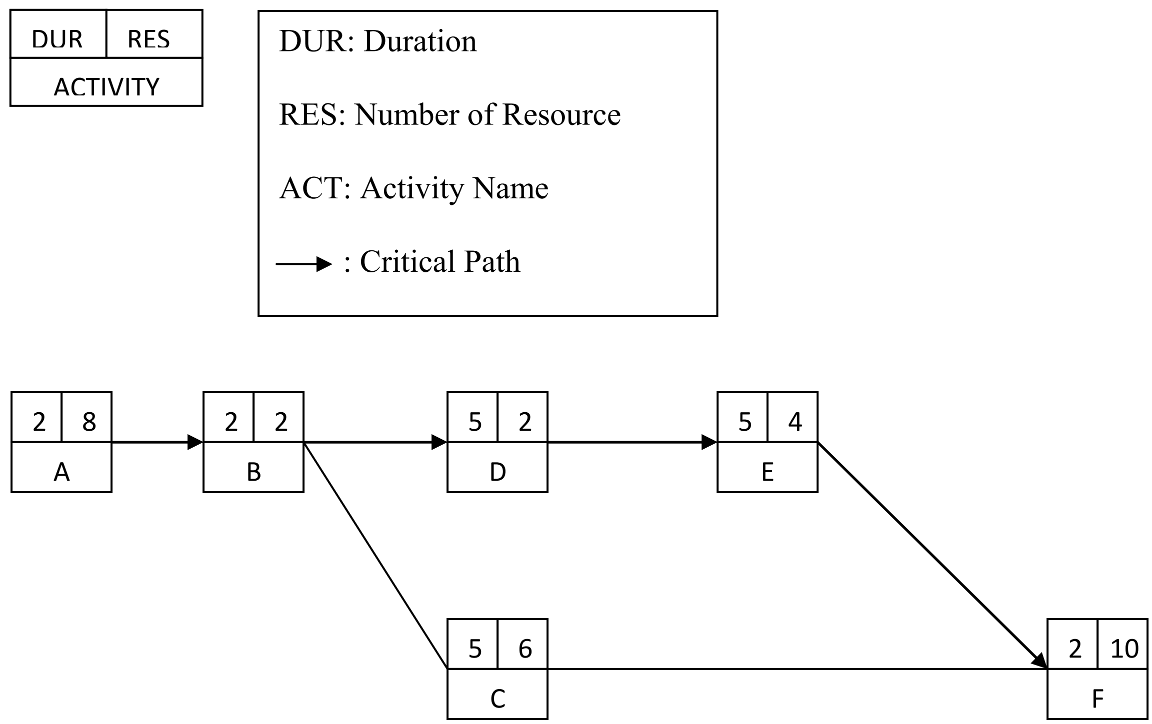



**Figure 4-12** Six Activity Network

The inputs for this model are same as described for Model A. Also, the GA for Model B also does sequence-step-arrangement, identifies successors and calculates early starts, early finishes, late starts, and late finishes, total & free floats exactly similar to the procedure explained for Model A. However, the generation of Initial Population differs. This method is illustrated by AlGhazi [28]. A base random number is generated for every chromosome to be generated in Initial Population. A set of random numbers for every gene of this chromosome is also generated. The idea is to randomly shift within its free float only those genes whose corresponding random number generated is less than the base random number generated. For all those genes (activities) whose base random numbers are more than the base random number, the shift will be 0 i.e., the start times of these activities will be equal to the maximum finish times of their corresponding predecessors. Based on this concept, the initial population generated for the network being analyzed is as shown (SV).

SV =

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 5 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 4 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |

The start & finish times for these chromosomes are also recorded. These values are useful for calculating optimal value for each chromosome directly. After calculating optimal values, reproduction of chromosomes occur as exemplified for Model A. Once, the reproduction matrix is created constituting start times, a new matrix (represented by RCR) is again generated which will constitute the shift value chromosomes for the reproduced chromosomes. This RCR matrix for the given network having ten as the size of population (Optimization function – RRH) is given as:

RCR =

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 4 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 0 | 0 | 0 |

Then, the reproduced matrix undergoes mutation. The mutation & crossover order is interchanged as mutation on RCR over last generations of GA iterations help bring more variability than crossover operation. The mutation is performed such that compulsory mutation occurs over the chromosome i.e., at least and only one gene has to mutate every time a chromosome mutate unless the whole chromosome has all zero shift values. Thus, if one analyzes the above RCR matrix for the given network, the seventh and eighth chromosome will mutate (mutation percentage – 90%). The set of mutated chromosome is set represented as SMR.

SMR =

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 2 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 0 | 0 | 0 |

The mutation of chromosome for this model is simpler than the mutation process for Model A. If the gene to be mutated value is x, the mutated value is randomly selected from the range 0 to x-1. Thus, the eighth chromosome mutated can be graphically represented.

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 |

Mutation of third gene

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 4-13** Mutation of Chromosome – Model B

The crossover of chromosomes is the next step which the GA algorithm performs. As already mentioned, the crossover occurs between two chromosomes. A random point is selected for crossover in-between the chromosome. Then, the two generated child crossover chromosome contain the same genes on the left side of the random point while the right part of the two chromosomes will interchange i.e., the first child will have second parent's part and vice versa. Although crossover occur for all the eight chromosomes for the given network (crossover percentage – 80%), the chromosome crossover can be visualized only for the seventh and eight chromosomes (SCR), and this can be graphically represented.

SCR =

$$
\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 5 & 0 & 0 & 0
\end{matrix}
$$

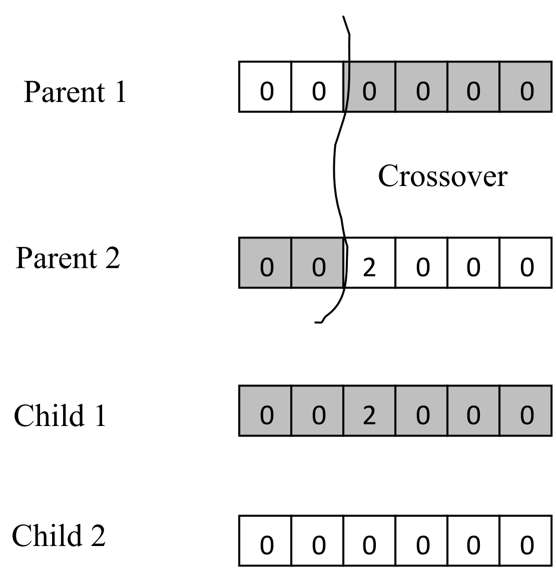



**Figure 4-14** Crossover of Chromosomes – Model B

The SCR matrix becomes the set of parent chromosomes for the next generation and this process is iteratively performed until the specified number of iterations is completed or the optimal value reaches 0. After the GA runs the required iterations, it prompts the user for the same questions as asked by the GA for Model-A if the user wants multiple run of GA for a desired purpose. The GA can also accept from the user a range of values within which the daily resource demand must lie. However, the convergence will be slow and the solution obtained will not be most optimal.

The result using RRH & RID optimization function obtained for the network with Model-B is same as obtained for Model-A as it is a small network. The results may vary for bigger networks. This is investigated in the next chapter.

# CHAPTER 5 APPLICATION EXAMPLES

There are various parameters governing the Genetic Algorithm for both the models. These are: Size of the population, Number of iterations, crossover & mutation percentage, the optimization function chosen (RRH & RID), the model chosen (Model-A or Model-B), Extension of project scheme and the percentage weight or relative importance to maximum resource demand ($W_2$). The results obtained by varying all these parameters may or may not be different. The performance of the Genetic Algorithm must be understood by varying these parameters with the help of illustrated examples. An analysis of the effect the variable parameters have on the functioning of the Genetic Algorithm is necessary. To demonstrate this, a 20-activity schedule to be used for multiple analyses will be utilized from El-Rayes and Jun [3].

| DUR | RES |
|---|---|
| ACTIVITY | |

DUR: Duration

RES: Number of Resource

ACT: Activity Name

⟶ : Critical Path

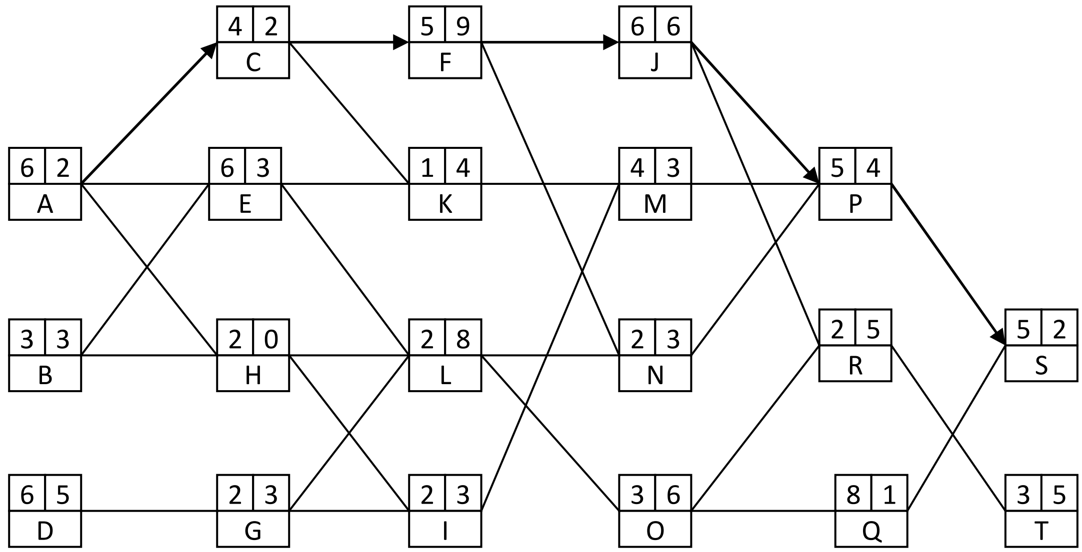



**Figure 5-1** Precedence Diagram of Twenty Activity Network

## 5.1 COMPARISION OF MODELS

The models are compared by executing the above network using the genetic algorithms of both models. The results obtained will be analyzed. The fixed parameters used for both models are:

Optimization Function: RRH

Crossover Percentage: 80%

Mutation Percentage: 90%

Extension of Schedule: None

Model – A: Initially, the size of population was kept 10. The solution was obtained after 113th iteration i.e., one hundred and thirteenth generation; the optimal value of RRH equal to zero was achieved. However, if the size of population is kept 100, the solution was obtained quickly at 5th generation. The start times of activities (arranged in sequence step arrangement) corresponding to the optimal solution are:

| **Activity** | A | B | D | C | E | G | H | F | I | K | L | J | M | N | O | P | Q | R | S | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Start Time** | 1 | 4 | 5 | 7 | 7 | 12 | 9 | 11 | 16 | 13 | 14 | 16 | 18 | 19 | 16 | 22 | 19 | 22 | 27 | 24 |

Model – B: When Model B was run with 10 size of population, even after 150 iterations, the least optimal value i.e., RRH was 1 was the most optimal chromosome. However, if the size of population was increased to 100 and the GA was run again, the least optimal value of 0 was achieved at 7th generation. The start times of activities (arranged in sequence step arrangement) corresponding to this optimal solution are:

| **Activity** | A | B | D | C | E | G | H | F | I | K | L | J | M | N | O | P | Q | R | S | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Start Time** | 1 | 2 | 5 | 7 | 7 | 12 | 8 | 11 | 16 | 13 | 14 | 16 | 18 | 16 | 16 | 22 | 19 | 22 | 27 | 24 |

If the crossover & mutation percentages were kept lower, it was observed that the convergence was slower. The higher is the size of population, faster is the convergence of solution and better is the optimal value achieved. Although the optimal value (RRH = 0) obtained by both the models is same (not necessary always), the resource profile obtained from the two models may differ as in this case. The resource histogram graph for both the models is displayed by the GA at the end.

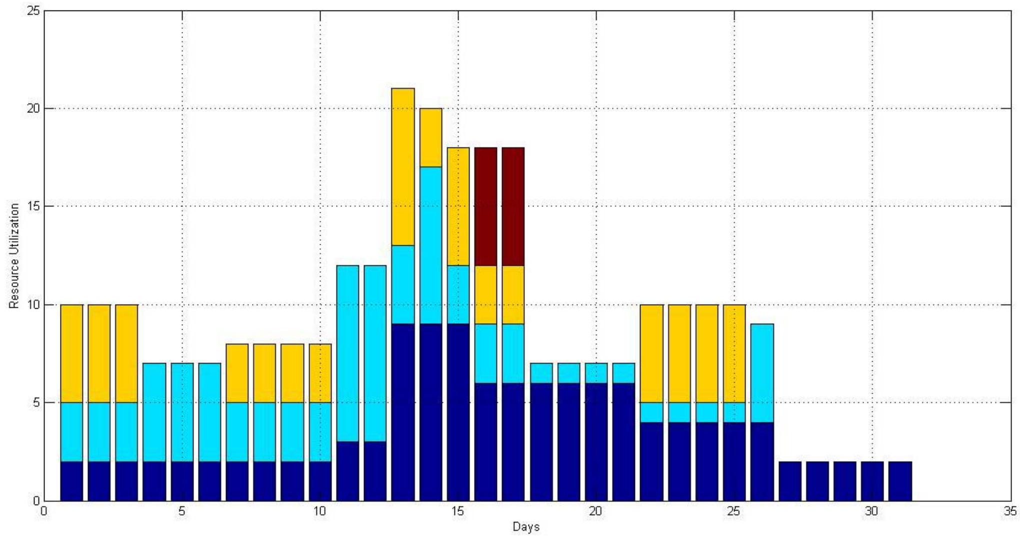



**Figure 5-2** Early Start Schedule Histogram

RRH Value for Early Start Schedule Histogram: 6

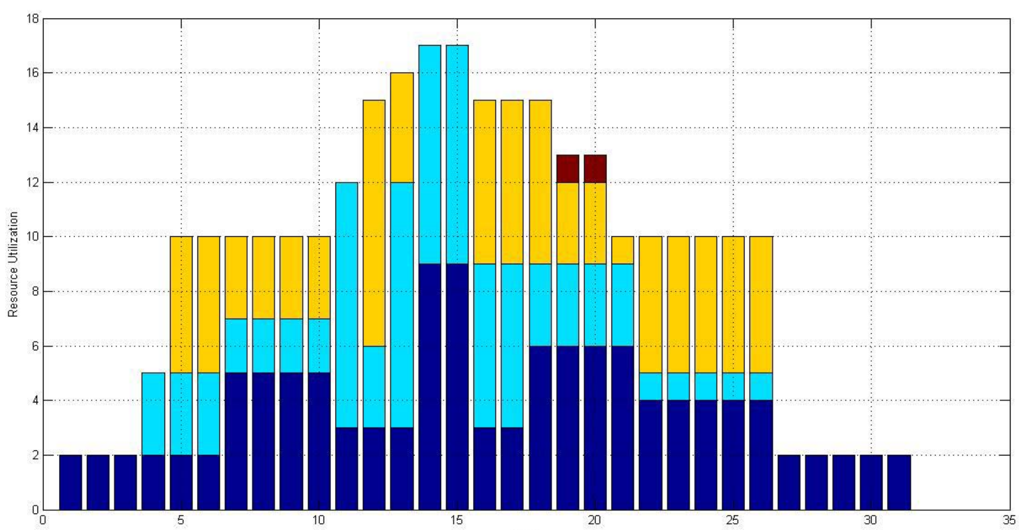



**Figure 5-3** Resource Profile obtained from Model - A

RRH Value of the above graph: 0

Graph Interpretation:

After the complete GA is run, a graph showing the resource histogram is displayed. Also, two variables ACTIVITY & RESOURCE are given for graph interpretation.

ACTIVITY

Columns 1 through 14

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 'A' | 'A' | 'A' | 'A' | 'A' | 'A' | 'D' | 'D' | 'D' | 'D' | 'E' | 'E' | 'G' | 'F' |
| [] | [] | [] | 'B' | 'B' | 'B' | 'C' | 'C' | 'C' | 'C' | 'F' | 'G' | 'F' | 'L' |
| [] | [] | [] | [] | 'D' | 'D' | 'E' | 'E' | 'E' | 'E' | [] | 'F' | 'K' | [] |
| [] | [] | [] | [] | [] | [] | [] | [] | 'H' | 'H' | [] | [] | [] | [] |

Columns 15 through 28

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 'F' | 'I' | 'I' | 'J' | 'J' | 'J' | 'J' | 'P' | 'P' | 'P' | 'P' | 'P' | 'S' | 'S' |
| 'L' | 'J' | 'J' | 'M' | 'M' | 'M' | 'M' | 'Q' | 'Q' | 'Q' | 'Q' | 'Q' | [] | [] |
| [] | 'O' | 'O' | 'O' | 'N' | 'N' | 'Q' | 'R' | 'R' | 'T' | 'T' | 'T' | [] | [] |
| [] | [] | [] | [] | 'Q' | 'Q' | [] | [] | [] | [] | [] | [] | [] | [] |

Columns 29 through 31

| | | |
|---|---|---|
| 'S' | 'S' | 'S' |
| [] | [] | [] |
| [] | [] | [] |
| [] | [] | [] |

RESOURCE =

Columns 1 through 17

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 3 | 3 | 3 | 9 | 9 | 3 | 3 |
| 0 | 0 | 0 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 9 | 3 | 9 | 8 | 8 | 6 | 6 |
| 0 | 0 | 0 | 0 | 5 | 5 | 3 | 3 | 3 | 3 | 0 | 9 | 4 | 0 | 0 | 6 | 6 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Columns 18 through 31

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 6 | 6 | 6 | 4 | 4 | 4 | 4 | 4 | 2 | 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 6 | 3 | 3 | 1 | 5 | 5 | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

ACTIVITY & RESOURCE – Each column shows the contribution from each activity & the number of resources contributed on the corresponding bar/day (bottom to up convention). For example, in Model – A, Column fifteen has two alphabets, F & L (see ACTIVITY) and values 9 & 8 (see RESOURCE). F represents contribution by Activity F represented on bar by dark blue (9 resources) & L represents contribution by Activity L represented on bar by sky blue (8 resources).

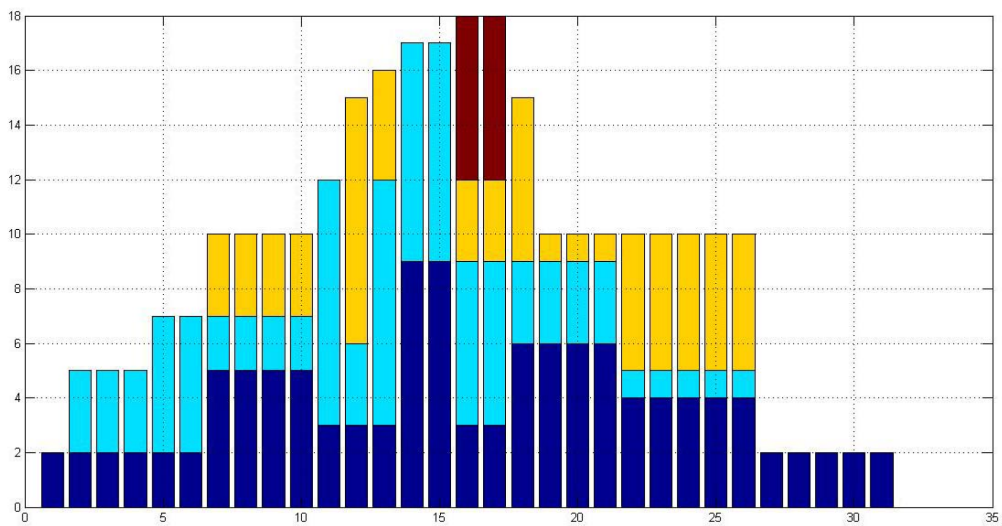



**Figure 5-4** Resource Profile Obtained from Model - B

RRH Value of the above Graph: 0

ACTIVITY =

Columns 1 through 14

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 'A' | 'A' | 'A' | 'A' | 'A' | 'A' | 'D' | 'D' | 'D' | 'D' | 'E' | 'E' | 'G' | 'F' |
| [] | 'B' | 'B' | 'B' | 'D' | 'D' | 'C' | 'C' | 'C' | 'C' | 'F' | 'G' | 'F' | 'L' |
| [] | [] | [] | [] | [] | [] | 'E' | 'E' | 'E' | 'E' | [] | 'F' | 'K' | [] |
| [] | [] | [] | [] | [] | [] | [] | 'H' | 'H' | [] | [] | [] | [] | [] |

Columns 15 through 28

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 'F' | 'I' | 'I' | 'J' | 'J' | 'J' | 'J' | 'P' | 'P' | 'P' | 'P' | 'P' | 'S' | 'S' |
| 'L' | 'J' | 'J' | 'M' | 'M' | 'M' | 'M' | 'Q' | 'Q' | 'Q' | 'Q' | 'Q' | [] | [] |
| [] | 'N' | 'N' | 'O' | 'Q' | 'Q' | 'Q' | 'R' | 'R' | 'T' | 'T' | 'T' | [] | [] |
| [] | 'O' | 'O' | [] | [] | [] | [] | [] | [] | [] | [] | [] | [] | [] |

Columns 29 through 31

| | | |
|---|---|---|
| 'S' | 'S' | 'S' |
| [] | [] | [] |
| [] | [] | [] |
| [] | [] | [] |

RESOURCE =

Columns 1 through 17

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 3 | 3 | 3 | 9 | 9 | 3 | 3 |
| 0 | 3 | 3 | 3 | 5 | 5 | 2 | 2 | 2 | 2 | 9 | 3 | 9 | 8 | 8 | 6 | 6 |
| 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 0 | 9 | 4 | 0 | 0 | 3 | 3 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 6 |

Columns 18 through 31

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 6 | 6 | 6 | 4 | 4 | 4 | 4 | 4 | 2 | 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 6 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

These are the following characteristics that were identified while analyzing the two models:

MATLAB® Version: 2007

System Properties: 4 GB RAM; Windows 7 (64-bit) & Processor – Intel® Core™ i3 CPU M330 @ 2.13 GHz

**Table 5-1** Characteristics of two models

| **MODEL – A** | **MODEL – B** |
|---|---|
| Start Times Encoding of Chromosomes | Shift Values Encoding of Chromosomes |
| The solution was obtained in approximately 7 minutes (much slower than other model). | The optimal value was reached is less than one minutes (45 sec average). |
| Higher the crossover & mutation percentages, faster the convergence | Higher the crossover & mutation percentages, faster the convergence |
| Less iterations are required | More iterations are required comparatively |
| Optimal Value reached after repetitive iterations, which may not be the lowest as compared to other model. | Optimal Value reached after repetitive iterations |
| The higher is the size of population, faster is the convergence of solution and better is the optimal value achieved | The higher is the size of population, faster is the convergence of solution and better is the optimal value achieved |
| Relatively Slow Convergence | Fast Convergence |
| Take more time for crossover and mutation but yields high chromosomes of high variability | Take comparatively much less time for crossover and mutation but yields less variable chromosomes |
| Discards infeasible chromosomes during reproduction | Discards infeasible chromosomes during reproduction |

| | |
|---|---|
| Gives different resource profile if the optimization function is changed (RRH/RID). Besides, may give different resource profile and optimal value than the other model. | Gives different resource profile if the optimization function is changed (RRH/RID). Besides, may give different resource profile and optimal value than the other model. |

## 5.2 VARIATION THROUGH MULTIPLE GA RUN

The Genetic Algorithm does not give one single optimal solution, especially for large schedules. Multiple runs of the GA yield different optimal values. Even if the optimal value is same, the maximum resource demand value will vary. Hence, by executing the genetic algorithm multiple times and after analyzing all the results, the best resource profile could be chosen in terms of lowest optimal value (RRH/RID) and MRD. All the programs in this research have been run a number of times unless and until a repeating most optimal value is obtained. The varying results of running multiple genetic algorithm on a network is demonstrated by executing the genetic algorithm on the given 20-activity network 10 times. All the results vary and the resource profile obtain differ for most solutions. Following table shows the start times corresponding to the optimal solution obtained in each run.

Size of Population – 100

Crossover Percentage – 90%

Mutation Percentage – 90%

Optimization Function – RRH

Number of Iterations Performed – 10

**Table 5-2** Start Times Obtained for Multiple Run of GA

| |
|---|
| ST =<br>1 4 5 7 7 12 8 11 16 13 14 16 18 16 16 22 19 22 27<br>24 |
| ST =<br>1 2 5 7 7 11 7 11 14 13 13 16 18 16 15 22 18 22<br>27 24 |
| ST =<br>1 5 5 7 8 12 11 11 16 15 14 16 18 19 16 22 19 22 27<br>24 |
| ST =<br>1 5 5 7 8 12 11 11 15 17 14 16 18 16 16 22 19 22 27<br>24 |
| ST =<br>1 5 5 7 8 11 11 11 13 14 14 16 18 16 16 22 19 22 27<br>24 |
| ST =<br>1 1 1 1 1 1 1 1 1 2 1 2 2 2<br>1 4 5 7 7 2 9 1 6 3 4 6 8 9 6 2 9 2 7 4 |
| ST =<br>1 1 1 1 1 1 1 1 1 2 1 2 2 2<br>1 2 5 7 7 2 8 1 6 3 4 6 8 6 6 2 9 2 7 4 |
| ST =<br>1 2 5 7 7 12 12 11 16 13 14 16 18 19 16 22 19 22<br>27 24 |
| ST =<br>1 3 6 7 8 12 7 11 16 16 14 16 18 17 16 22 19 22 27<br>24 |
| ST =<br>1 2 5 7 7 12 9 11 16 13 14 16 18 16 16 22 19 22 27<br>24 |

The optimal value (RRH) achieved, corresponding MRD value and the generation after which the optimal value is obtained is also shown.

**Table 5-3** Optimal Values obtained for Multiple Runs

| RUN # | RRH VALUE | MRD VALUE | Value Achieved @ Generation # |
|---|---|---|---|
| 1 | 0 | 18 | 3 |
| 2 | 0 | 21 | 10 |
| 3 | 0 | 21 | 5 |
| 4 | 1 | 20 | 10 |
| 5 | 0 | 24 | 10 |
| 6 | 0 | 17 | 5 |
| 7 | 0 | 18 | 7 |
| 8 | 0 | 17 | 4 |
| 9 | 2 | 19 | 10 |
| 10 | 0 | 18 | 8 |

The graphs for all the ten cases as obtained from the result after executing the program is displayed.

Legend:

X Axis:  Time in Days

Y Axis: Resource Utilization from multiple activities differentiated by varying shades.

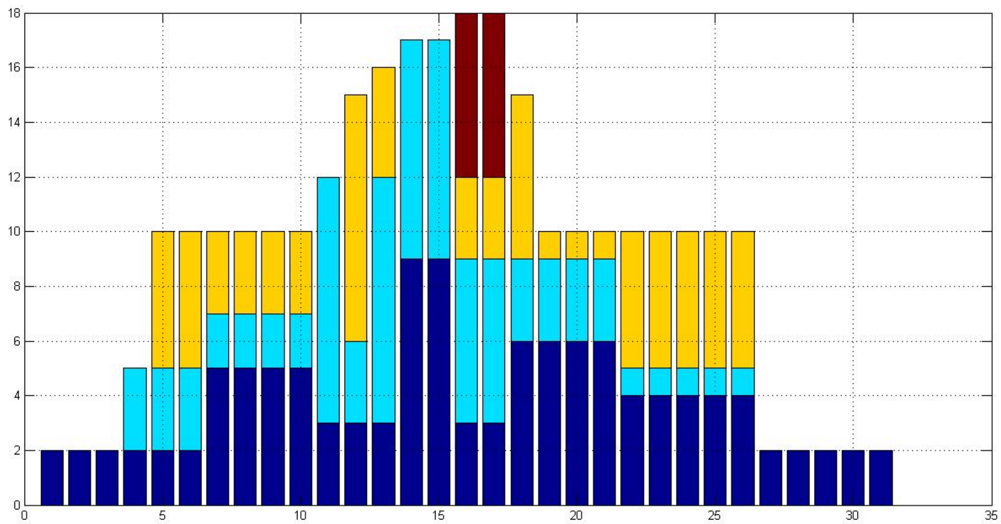



**Figure 5-5** Resource Histogram corresponding to GA Run #1

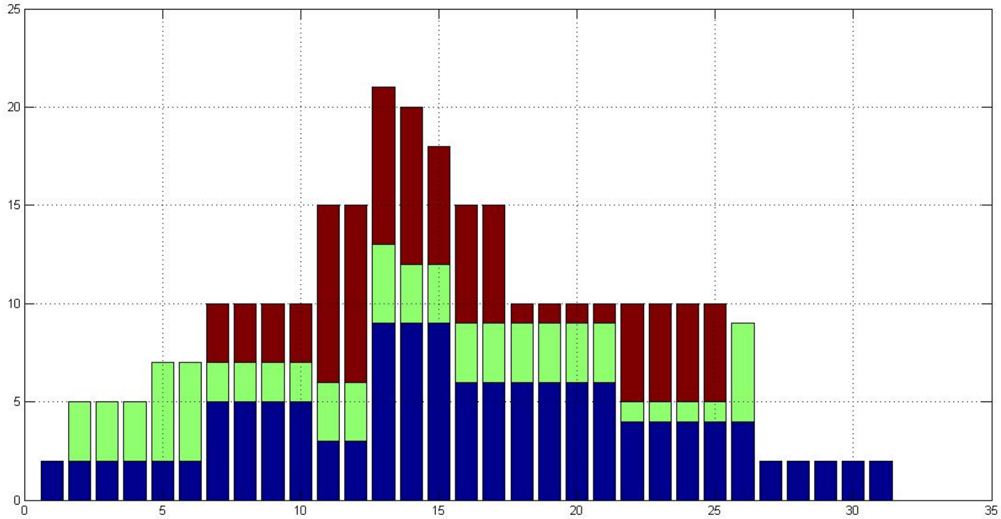



**Figure 5-6** Resource Histogram corresponding to GA Run #2

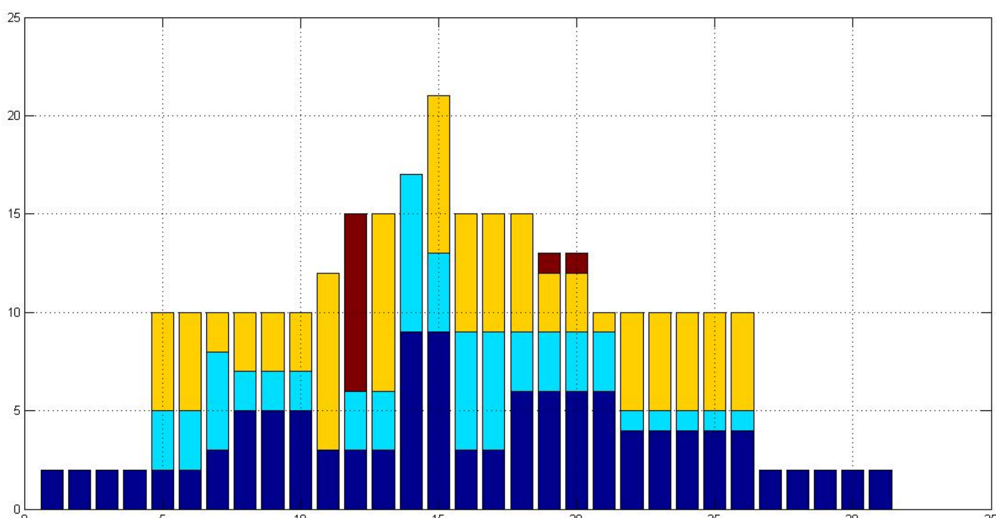



**Figure 5-7** Resource Histogram corresponding to GA Run #3

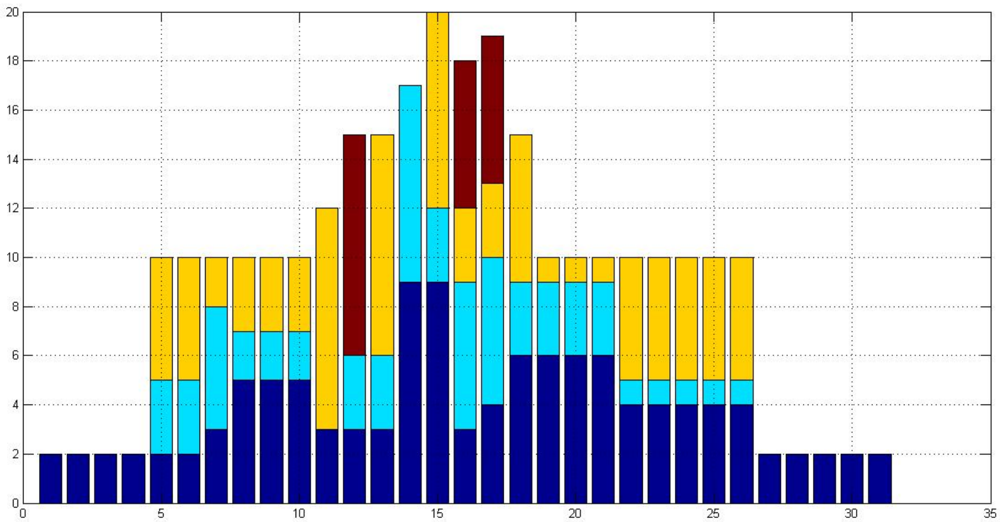



**Figure 5-8** Resource Histogram corresponding to GA Run #4

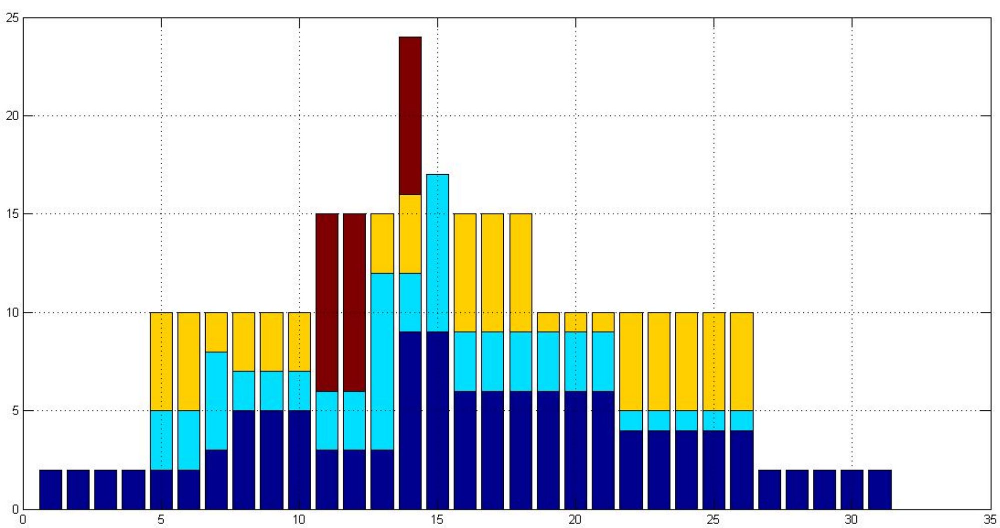



**Figure 5-9** Resource Histogram corresponding to GA Run #5

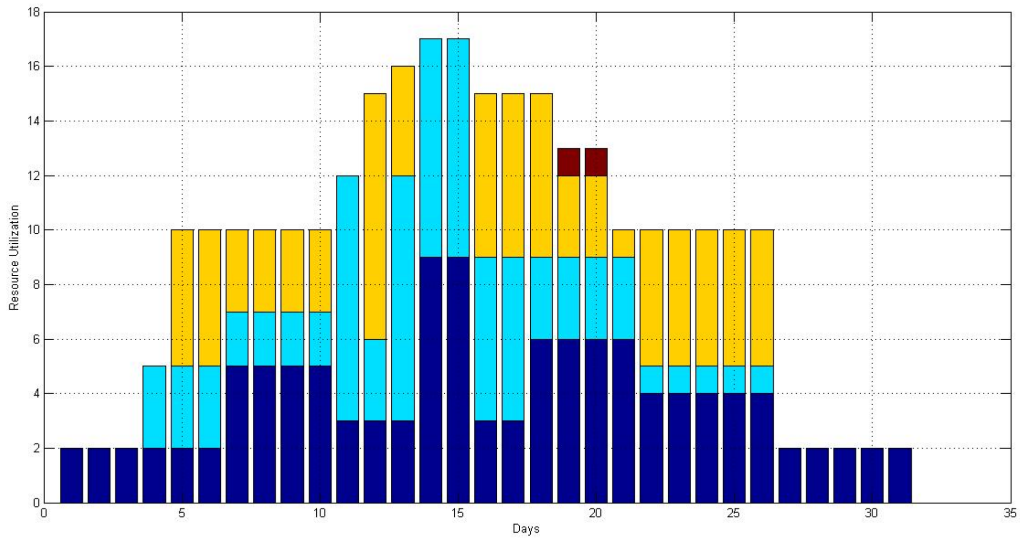



**Figure 5-10** Resource Histogram corresponding to GA Run #6

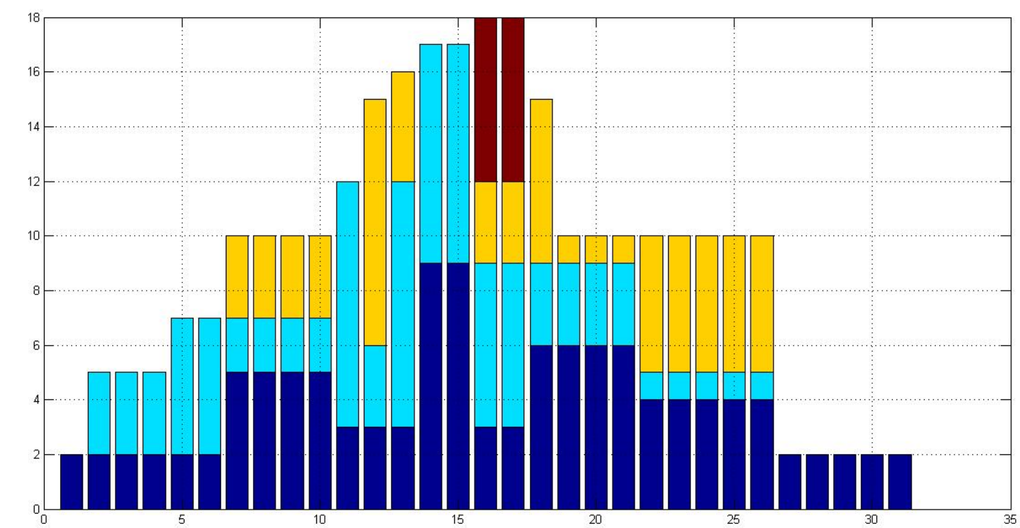



**Figure 5-11** Resource Histogram corresponding to GA Run #7

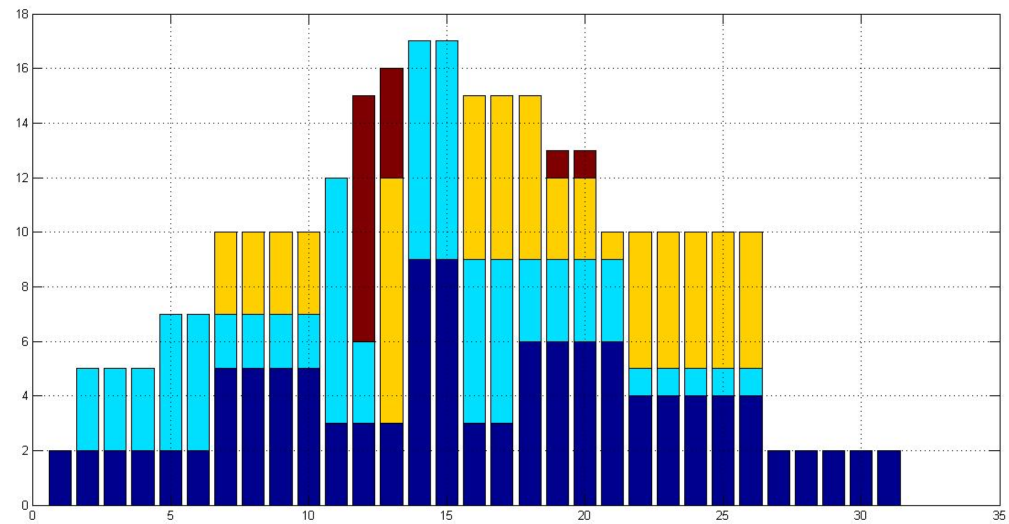



**Figure 5-12** Resource Histogram corresponding to GA Run #8

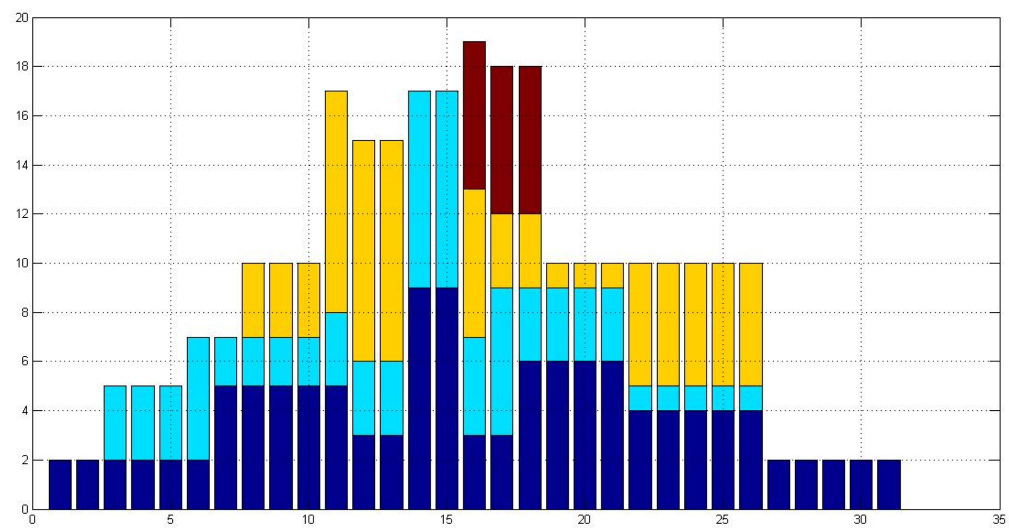



**Figure 5-13** Resource Histogram corresponding to GA Run #9

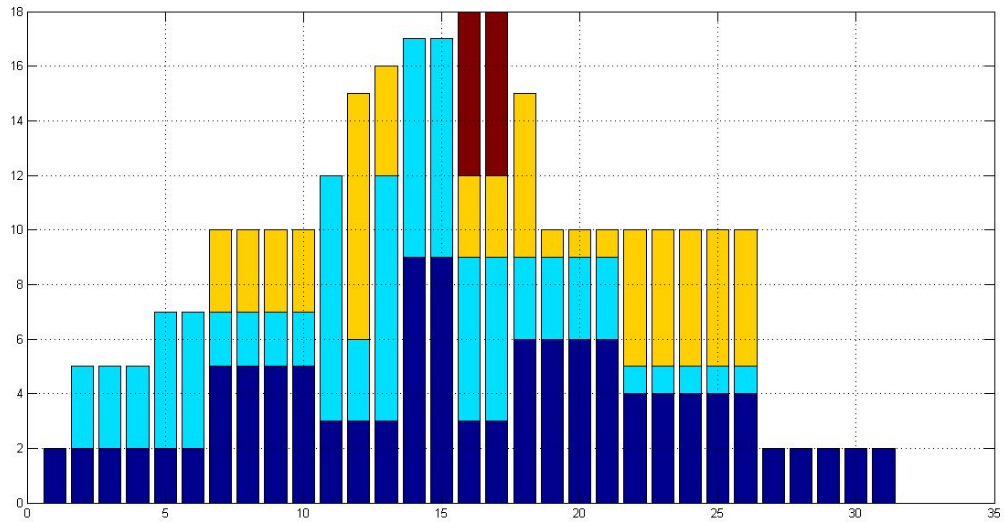



**Figure 5-14** Resource Histogram corresponding to GA Run #10

## 5.3 VARIATION BY EXTENSION OF SCHEDULE

Extension of schedule is generally allowed to obtained a smoother resource distribution profile or for cost benefits. It may also be required to reduce peak demand value. The schedule may also be extended due to delay. Hence, it is important to investigate resource leveling of network beyond their fixed project duration. This program gives the user flexibility to enter the amount of days the network can be extended to look for a more optimal schedule. Multiple GA runs are necessary for this approach and the schedule corresponding to the least optimal function value obtained may have a peak value higher than the fixed duration schedule. However, the schedule obtained by extension will have fewer variations, i.e., smoother resource curve than the fixed duration resource profile. The same 20-activity schedule is allowed to extend by 5 days. Following is the result achieved.

Size of Population: 100

Mutation Percentage: 80%

Crossover Percentage: 90%

Optimization Function: RRH

Number of Iterations Performed: 50

RRH Value Achieved: 2

MRD Value Achieved: 18

Duration: 32 days

Start Times of the Optimal Value:

ST =
1 1 4 8 7 10 12 12 16 13 14 17 19 17 16 23 19 23
28 25

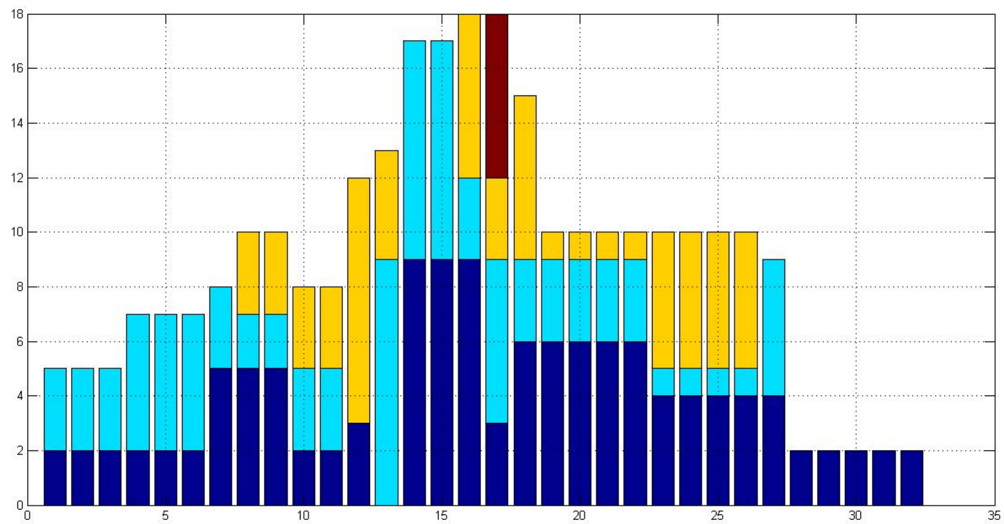



**Figure 5-15** Resource Profile of the Extended Schedule

## 5.4 VARIATION DUE TO MULTIPLE FACTORS

A small and a large schedule, consisting of 30 and 120 activities respectively, are run multiple times (five) without & with extension of schedules to assess the comparative relationship of both models. A total number of ten days of extension is entered for 30 activity schedule whereas the 120 activity schedule is extended by 20 days. Each case is run five times to seek the lowest optimal value. RRH metric was used as the optimization function and iterative process is not stopped until an iterative value is achieved ten days repetitively. Mutation percentage is 99% and crossover percentage is 99%. The size of the population is hundred.

The following table represents the input for the 30 activity schedule.

**Table 5-4** Input for the 30 Activity Schedule

| Activity | Duration | Resources | No. of Predecessors | Predecessors | | | |
|---|---|---|---|---|---|---|---|
| Aa | 2 | 10 | 0 | - | - | - | - |
| Ab | 4 | 10 | 1 | Aa | - | - | - |
| Ac | 1 | 13 | 1 | Ab | - | - | - |
| Ad | 4 | 10 | 1 | Ab | - | - | - |
| Ae | 3 | 11 | 1 | Ad | - | - | - |
| Af | 2 | 15 | 1 | Ac | - | - | - |
| Ag | 1 | 10 | 2 | Ac | Ad | - | - |
| Ah | 3 | 10 | 1 | Ag | - | - | - |
| Ai | 2 | 11 | 2 | Ae | Af | - | - |
| Aj | 2 | 10 | 2 | Ae | Ah | - | - |
| Ak | 2 | 10 | 2 | Ai | Aj | - | - |
| Al | 1 | 13 | 2 | Af | Ah | - | - |
| Am | 5 | 10 | 2 | Ak | Al | - | - |
| An | 4 | 20 | 1 | Ah | - | - | - |
| Ao | 3 | 19 | 1 | Ag | - | - | - |
| Ap | 5 | 10 | 2 | Am | Ao | - | - |
| Aq | 2 | 28 | 2 | Aj | Ao | - | - |
| Ar | 3 | 20 | 1 | Am | - | - | - |
| As | 3 | 15 | 2 | An | Ap | - | - |
| At | 5 | 10 | 2 | An | Ap | - | - |
| Au | 3 | 10 | 1 | At | - | - | - |
| Av | 3 | 23 | 2 | Ap | Aq | - | - |
| Aw | 1 | 17 | 3 | Aq | Ar | At | - |
| Ax | 5 | 10 | 4 | Aq | Ar | As | Au |
| Ay | 2 | 20 | 2 | Av | Aw | - | - |
| Ba | 1 | 10 | 2 | Aw | Ax | - | - |
| Bb | 2 | 12 | 1 | Ba | - | - | - |
| Bc | 4 | 10 | 1 | Ba | - | - | - |
| Bd | 4 | 11 | 2 | Aw | Ax | - | - |
| Be | 5 | 10 | 4 | Ay | Bd | Bb | Bc |

The following table represents the input for the 120 activity schedule.

**Table 5-5** Input for the 120 Network

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Aa | 2 | 5 | 0 | - | - | - | - |
| Ab | 3 | 5 | 1 | Aa | - | - | - |
| Ac | 1 | 15 | 1 | Ab | - | - | - |
| Ad | 4 | 5 | 1 | Ab | - | - | - |
| Ae | 5 | 5 | 1 | Ad | - | - | - |
| Af | 3 | 8 | 1 | Aa | - | - | - |
| Ag | 4 | 8 | 1 | Af | - | - | - |
| Ah | 3 | 12 | 2 | Ac | Ag | - | - |
| Ai | 5 | 5 | 2 | Ae | Ag | - | - |
| Aj | 2 | 10 | 3 | Ac | Ae | Af | - |
| Ak | 5 | 8 | 1 | Ag | - | - | - |
| Al | 2 | 8 | 1 | Ak | - | - | - |
| Am | 2 | 12 | 1 | Ah | - | - | - |
| An | 2 | 5 | 1 | Ai | - | - | - |
| Ao | 1 | 5 | 3 | Aj | Am | An | - |
| Ap | 1 | 17 | 3 | Aj | Ak | An | - |
| Aq | 3 | 8 | 1 | Al | - | - | - |
| Ar | 2 | 5 | 2 | Ao | Aq | - | - |
| As | 5 | 5 | 1 | Ar | - | - | - |
| At | 5 | 5 | 1 | As | - | - | - |
| Au | 4 | 10 | 2 | Ap | As | - | - |
| Av | 1 | 9 | 1 | As | - | - | - |
| Aw | 4 | 10 | 1 | Au | - | - | - |
| Ax | 3 | 10 | 2 | Av | Aw | - | - |
| Ay | 5 | 5 | 2 | At | Av | - | - |
| Ba | 2 | 10 | 2 | Ax | Ay | - | - |
| Bb | 2 | 5 | 1 | Ay | - | - | - |
| Bc | 3 | 13 | 1 | Av | - | - | - |
| Bd | 4 | 10 | 2 | Ba | Bc | - | - |
| Be | 2 | 5 | 2 | Bb | Bc | - | - |
| Bf | 5 | 5 | 1 | Be | - | - | - |
| Bg | 3 | 5 | 1 | Bf | - | - | - |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Bh | 1 | 13 | 1 | Be | - | - | - |
| Bi | 2 | 13 | 2 | Bd | Be | - | - |
| Bj | 1 | 5 | 2 | Bd | Bg | - | - |
| Bk | 1 | 13 | 1 | Bi | - | - | - |
| Bl | 4 | 5 | 2 | Bh | Bj | - | - |
| Bm | 1 | 5 | 1 | Bl | - | - | - |
| Bn | 3 | 7 | 1 | Bl | - | - | - |
| Bo | 4 | 10 | 2 | Bi | Bj | - | - |
| Bp | 4 | 5 | 2 | Bk | Bm | - | - |
| Bq | 2 | 5 | 3 | Bn | Bo | Bp | - |
| Br | 5 | 5 | 1 | Bq | - | - | - |
| Bs | 1 | 10 | 1 | Bq | - | - | - |
| Bt | 1 | 5 | 1 | Br | - | - | - |
| Bu | 3 | 6 | 2 | Bs | Bt | - | - |
| Bv | 3 | 11 | 2 | Br | Bs | - | - |
| Bw | 5 | 5 | 2 | Bs | Bt | - | - |
| Bx | 1 | 6 | 1 | Bu | - | - | - |
| By | 2 | 6 | 1 | Bx | - | - | - |
| Ca | 4 | 6 | 1 | By | - | - | - |
| Cb | 1 | 11 | 1 | Bv | - | - | - |
| Cc | 4 | 5 | 2 | Bw | Bx | - | - |
| Cd | 2 | 5 | 2 | Cb | Cc | - | - |
| Ce | 3 | 7 | 1 | Bx | - | - | - |
| Cf | 5 | 5 | 2 | Ca | Cd | - | - |
| Cg | 3 | 7 | 1 | Ce | - | - | - |
| Ch | 5 | 6 | 3 | Ca | Cd | Cg | - |
| Ci | 5 | 6 | 2 | Cf | Ch | - | - |
| Cj | 4 | 5 | 2 | Cf | Cg | - | - |
| Ck | 1 | 8 | 2 | Cf | Cg | - | - |
| Cl | 4 | 7 | 2 | Cf | Ch | - | - |
| Cm | 3 | 11 | 1 | Ck | - | - | - |
| Cn | 4 | 6 | 2 | Ci | Ck | - | - |
| Co | 1 | 5 | 2 | Cj | Ck | - | - |
| Cp | 5 | 5 | 1 | Co | - | - | - |
| Cq | 4 | 7 | 2 | Ck | Cl | - | - |
| Cr | 1 | 5 | 3 | Cm | Cp | Cq | - |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Cs | 2 | 13 | 1 | Cn | - | - | - |
| Ct | 1 | 6 | 4 | Cm | Cn | Co | Cq |
| Cu | 4 | 13 | 2 | Co | Cq | - | - |
| Cv | 5 | 8 | 1 | Cq | - | - | - |
| Cw | 5 | 5 | 2 | Cr | Ct | - | - |
| Cx | 1 | 8 | 2 | Cs | Cw | - | - |
| Cy | 2 | 7 | 1 | Cw | - | - | - |
| Da | 5 | 13 | 1 | Cu | - | - | - |
| Db | 5 | 5 | 2 | Cv | Cw | - | - |
| Dc | 4 | 13 | 1 | Da | - | - | - |
| Dd | 1 | 7 | 4 | Cu | Cv | Cx | Cy |
| De | 2 | 17 | 2 | Cw | Da | - | - |
| Df | 3 | 17 | 2 | Dd | De | - | - |
| Dg | 4 | 5 | 2 | Db | Dd | - | - |
| Dh | 2 | 9 | 2 | Dc | Dg | - | - |
| Di | 1 | 5 | 1 | Dg | - | - | - |
| Dj | 5 | 5 | 1 | Di | - | - | - |
| Dk | 4 | 13 | 2 | Df | Di | - | - |
| Dl | 3 | 13 | 2 | De | Dj | - | - |
| Dm | 4 | 5 | 2 | Dh | Dj | - | - |
| Dn | 3 | 5 | 1 | Dm | - | - | - |
| Do | 2 | 7 | 1 | Dn | - | - | - |
| Dp | 4 | 5 | 2 | Dk | Dn | - | - |
| Dq | 1 | 12 | 2 | Dl | Dm | - | - |
| Dr | 2 | 6 | 2 | Dp | Dq | - | - |
| Ds | 3 | 5 | 3 | Dl | Do | Dp | - |
| Dt | 4 | 7 | 1 | Dn | - | - | - |
| Du | 2 | 7 | 2 | Dq | Ds | - | - |
| Dv | 2 | 5 | 2 | Dq | Ds | - | - |
| Dw | 4 | 6 | 2 | Do | Dr | - | - |
| Dx | 3 | 5 | 1 | Dv | - | - | - |
| Dy | 3 | 7 | 1 | Dt | - | - | - |
| Ea | 1 | 7 | 1 | Du | - | - | - |
| Eb | 3 | 7 | 1 | Dy | - | - | - |
| Ec | 5 | 6 | 1 | Dw | - | - | - |
| Ed | 5 | 5 | 3 | Dx | Ea | Eb | - |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Ee | 3 | 12 | 1 | Ea | - | - | - |
| Ef | 1 | 5 | 1 | Ed | - | - | - |
| Eg | 1 | 13 | 2 | Eb | Ee | - | - |
| Eh | 2 | 6 | 1 | Ec | - | - | - |
| Ei | 5 | 8 | 2 | Ee | Eh | - | - |
| Ej | 1 | 8 | 1 | Ei | - | - | - |
| Ek | 5 | 5 | 2 | Ef | Eh | - | - |
| El | 4 | 10 | 2 | Ed | Eg | - | - |
| Em | 4 | 5 | 2 | Ee | Ek | - | - |
| En | 1 | 13 | 1 | El | - | - | - |
| Eo | 1 | 8 | 2 | Ej | El | - | - |
| Ep | 5 | 5 | 2 | Em | Eo | - | - |
| Eq | 5 | 12 | 3 | Ej | Ek | En | - |
| Er | 3 | 5 | 1 | Ep | - | - | - |
| Es | 1 | 12 | 2 | El | Em | - | - |
| Et | 4 | 5 | 3 | Er | Eq | Es | - |

The following were the results obtained:

**Table 5-6** Results from the GA Run – 30 Activities Schedule

| **30 ACTIVITY NETWORK** | | | | |
|---|---|---|---|---|
| **EXTENSION = ZERO DAYS** | | | | |
| | **MODEL A** | | **MODEL B** | |
| RUN # | **RRH** | DURATION | **RRH** | DURATION |
| 1 | 45 | 52 | 64 | 52 |
| 2 | 66 | 52 | 55 | 52 |
| 3 | 51 | 52 | 61 | 52 |
| 4 | 58 | 52 | 59 | 52 |
| 5 | 51 | 52 | 40 | 52 |
| **EXTENSION = TEN DAYS** | | | | |
| | **MODEL A** | | **MODEL B** | |
| RUN # | **RRH** | DURATION | **RRH** | DURATION |
| 1 | 40 | 56 | 55 | 57 |
| 2 | 46 | 54 | 71 | 57 |
| 3 | 64 | 57 | 29 | 57 |
| 4 | 54 | 57 | 33 | 57 |
| 5 | 73 | 56 | 43 | 57 |

**Table 5-7** Results from the GA Run – 30 Activities Schedule

| 120 ACTIVITY NETWORK | | | | |
|---|---|---|---|---|
| **EXTENSION = ZERO DAYS** | | | | |
| | **MODEL A** | | **MODEL B** | |
| RUN # | **RRH** | DURATION | **RRH** | DURATION |
| 1 | 231 | 163 | 211 | 163 |
| 2 | 239 | 163 | 212 | 163 |
| 3 | 231 | 163 | 223 | 163 |
| 4 | 239 | 163 | 191 | 163 |
| 5 | 231 | 163 | 174 | 163 |
| **EXTENSION = TWENTY DAYS** | | | | |
| | **MODEL A** | | **MODEL B** | |
| RUN # | **RRH** | DURATION | **RRH** | DURATION |
| 1 | 251 | 175 | 194 | 167 |
| 2 | 228 | 174 | 191 | 167 |
| 3 | 227 | 167 | 155 | 167 |
| 4 | 233 | 167 | 190 | 167 |
| 5 | 217 | 167 | 211 | 167 |

It can be inferred from the results that Model B gives lowest optimal values and thence, best results or better resource profiles then Model A. Even if time is considered, Model B requires only 25 minutes on average for 120 activity network to obtain the result, whereas Model A requires more than 1800 minutes to get the result in each case. Therefore, Model B yields much faster & better results and thus, it does not suffer from poor exploitation like Model A.

# CHAPTER 6 CONCLUSIONS AND RECOMMENDATIONS

A comprehensive analysis of the quantitative method being proposed is a prerequisite to the robust development of the models. The genetic algorithms for both models programmed in MATLAB® were subjected to multiple runs, extensive analysis and scrutinized, corrected and verified at every step of its development. The codes have written in a series of small modules connected together through a web of network such that every module can perform individually, and also possess the capacity to easily flux with one another to perform as a unit. The salient characteristics of the two models are inferred from multiple executions of the two programs while varying different parameters.

- Both the genetic models have been programmed for resource leveling of networks. One model encodes start times of activities (Model-A) while another model (Model-B) encodes shift values.
- The attainment of an optimized schedule for resource leveling is characterized by low optimization function value. The optimization value is a function of be Release & Re-hire Metric (RRH), Resource Idle Days Metric (RID) or a combination of the either two (RRH/RID) with maximum resource demand (MRD).
- The optimal value obtained from both the models may not be same. Even if the optimal value obtained is same, the resource profile may differ.

- Model-B shows faster convergence than Model-A. Model-B also gives better results than Model A.
- The crossover and mutation operation time is much slower in Model-A than Model-B because of the complexity involved during this operation.
- Both the models discard infeasible chromosomes, if any, due to extension of project duration beyond the limit or negative start times. Also, both models yield different result if the optimization function is varied.
- The optimal solution reached by the two metrics may be different. However, both the solutions represent optimal value and either can be used depending upon the project requirement.
- The higher is the size of population, faster is the convergence of solution and better is the optimal value achieved.
- Higher are the crossover & mutation percentages, faster is the convergence of optimal solution.
- Multiple runs of the genetic algorithm are possible and the various parameters can be varied after the required number of iterations is accomplished, if necessary.
- Decreasing the MRD factor will reduce the MRD value until a certain optimum point. After the optimal value is achieved, the MRD value may increase or remain same.
- Extension of schedule is an option for the user and this yields smoother profiles.
- Multiple runs is recommended to obtain the lowest optimal value with the least MRD as only a single run may not yield the most optimal value.

- The GA can also accept from the user a range of values within which the daily resource demand must lie. However, the convergence will be slow and the solution obtained will not be most optimal.
- The shift value encoded model (Model B) can be used for practical schedules to get efficient, better and faster results of resource leveling of networks even with extension scheme.

## REFERENCES


1. Easa, S.M., *Resource Leveling in Construction by Optimization.* Journal of Construction Engineering and Management, 1989. **115**(2): p. 302-316.
2. Sheng-Li, Z., et al. *GA-Based Resource Leveling Optimization for Construction Project*. in *Machine Learning and Cybernetics, 2006 International Conference on*. 2006.
3. El-Rayes, K. and D.H. Jun, *Optimizing Resource Leveling in Construction Projects.* Journal of Construction Engineering and Management, 2009. **135**(11): p. 1172-1180.
4. Hinze, J., *Construction planning and scheduling*. 2008, Harlow: Prentice Hall.
5. Savin, D., S. Alkass, and P. Fazio, *A procedure for calculating the weight-matrix of a neural network for resource leveling.* Advances in Engineering Software, 1997. **28**(5): p. 277-283.
6. Neumann, K. and J. Zimmermann, *Resource levelling for projects with schedule-dependent time windows.* European Journal of Operational Research, 1999. **117**(3): p. 591-605.
7. Petrovic, R. *On optimization of resource leveling in project plans, In: Lombaers, H.J. (Ed.), Project Planning by Network Analysis*. in *Proceedings of the 2nd International Congress*. 1969. North-Holland, Amsterdam pp. 268–273.
8. Ahuja, H.N., *Construction Performance Control by Networks*. 1976, New York Wiley.

9. Younis, M.A. and B. Saad, *Optimal resource leveling of multi-resource projects.* Computers & Industrial Engineering, 1996. **31**(1-2): p. 1-4.

10. Bandelloni, M., M. Tucci, and R. Rinaldi, *Optimal resource leveling using non-serial dyanamic programming.* European Journal of Operational Research, 1994. **78**(2): p. 162-177.

11. Leu, S.-S., C.-H. Yang, and J.-C. Huang, *Resource leveling in construction by genetic algorithm-based optimization and its decision support system application.* Automation in Construction, 2000. **10**(1): p. 27-41.

12. Galbreath, R.V., *Computer program for leveling previous resource usage.* Journal of the Construction Division, 1965. **91**: p. pp. 107–124.

13. Burgess, A.R. and J.B. Killebrew, *Variation in activity level on a cyclical arrow diagram.* Journal of Industrial Engineering, 1962 **13**: p. pp. 76–83

14. Son, J. and M.J. Skibniewski, *Multiheuristic Approach for Resource Leveling Problem in Construction Engineering: Hybrid Approach.* Journal of Construction Engineering and Management, 1999. **125**(1): p. 23-31.

15. Harris, R.B., *Packing Method for Resource Leveling (Pack).* Journal of Construction Engineering and Management, 1990. **116**(2): p. 331-350.

16. Geng, J.-q., L.-p. Weng, and S.-h. Liu, *An improved ant colony optimization algorithm for nonlinear resource-leveling problems.* Computers & Mathematics with Applications, 2011. **61**(8): p. 2300-2305.

17. Ding, J.-y. and Z.-f. Wang, *Applying Ant Colony Optimization to Multiple Resources Leveling Problem.* Energy Procedia, 2011. **13**(0): p. 2645-2653.

18. Zhang, H., H. Li, and C.M. Tam, *Particle swarm optimization for resource-constrained project scheduling.* International Journal of Project Management, 2006. **24**(1): p. 83-92.
19. Chan, W.-T., D.K.H. Chua, and G. Kannan, *Construction Resource Scheduling with Genetic Algorithms.* Journal of Construction Engineering and Management, 1996. **122**(2): p. 125-132.
20. Hegazy, T., *Optimization of Resource Allocation and Leveling Using Genetic Algorithms.* Journal of Construction Engineering and Management, 1999. **125**(3): p. 167-175.
21. Leu, S.-S. and C.-H. Yang, *GA-Based Multicriteria Optimal Model for Construction Scheduling.* Journal of Construction Engineering and Management, 1999. **125**(6): p. 420-427.
22. Senouci, A.B. and N.N. Eldin, *Use of Genetic Algorithms in Resource Scheduling of Construction Projects.* Journal of Construction Engineering and Management, 2004. **130**(6): p. 869-877.
23. Renner, G. and A. Ekárt, *Genetic algorithms in computer aided design.* Computer-Aided Design, 2003. **35**(8): p. 709-726.
24. Obitko, M. *Introduction to Genetic Algorithms*. 1998; Available from: http://www.obitko.com/tutorials/genetic-algorithms/index.php.
25. Cramster.com. Available from: http://www.cramster.com/answers-oct-10/electrical-engineering/matlab-robotic-pa-implement-genetic-algorithm-matlab-function-named_991073.aspx?rec=0.

26. Abido, M.A. and A.M. Elazouni, *Precedence-Preserving GAs Operators for Scheduling Problems with Activities' Start Times Encoding.* Journal of Computing in Civil Engineering, 2010. **24**(4): p. 345-356.

27. Demeulemeester, E. and W. Herroelen, *Project scheduling: a research handbook*. 2002: Kluwer Academic Publisher.

28. AlGhazi, A., *Finance-Based Scheduling of Activity Networks*. 2009, King Fahd University of Petroleum and Minerals.

# VITAE

| | | |
|---|---|---|
| Name | : | Mohammed Abdul Rahman |
| Address | : | 3-8-67, LB Nagar, |
| | | Hyderbad, A.P., India |
| Telephone | : | +966-531564313 |
| Date of Birth | : | 05-01-1988 |
| Nationality | : | Indian |
| Languages | : | English, Urdu, Hindi, Telugu and Arabic (Basic) |
| Contact Address | : | King Fahd University of Petroleum and Minerals |
| | | P.O.BOX-8661, Dhahran-31261, Kingdom of Saudi Arabia |
| Email | : | kashif012@gmail.com, |
| | | marahman@kfupm.edu.sa |
| Education | : | Master of Science (M.S.) MAY 2012 |
| | | Construction Engineering and Management Department |
| | | King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia |
| | | Bachelor of Engineering (B.E.) MAY 2009 |
| | | Civil Engineering Department |
| | | Osmania University, Hyderabad, India. |