A. AHMED

ATILIM UNIVERSITY 2020

GENETIC-ALGORITHM-BASED OPTIMIZATION APPROACH FOR TIME-COST TRADE-OFF PROBLEM WITH GENERALIZED PRECEDENCE RELATIONSHIPS

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

ARY HAMA FARAJ AHMED AHMED

A MASTER OF SCIENCE THESIS
IN
THE DEPARTMENT OF CIVIL ENGINEERING

AUGUST 2020

GENETIC-ALGORITHM-BASED OPTIMIZATION APPROACH FOR TIME-COST TRADE-OFF PROBLEM WITH GENERALIZED PRECEDENCE RELATIONSHIPS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

ARY HAMA FARAJ AHMED AHMED

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
THE DEPARTMENT OF CIVIL ENGINEERING

AUGUST 2020

Approval of the Graduate School of Natural and Applied Sciences, Atilim University.

Prof. Dr. Ali KARA
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Civil Engineering, Atılım University.

Prof. Dr. Hasan Umur AKAY
Head of Department

This is to certify that we have read the thesis "Genetic-Algorithm-Based Optimization Approach for Time-Cost Trade-Off Problem with Generalized Precedence Relationships" submitted by Ary Hama Faraj Ahmed Ahmed and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Saman AMINBAKHSH
Supervisor

Examining Committee Members:

Prof. Dr. Rıfat Sönmez
Civil Eng. Dept., Middle East Technical University

Asst. Prof. Dr. Saman Aminbakhsh
Civil Eng. Depr., Atilim University

Assoc. Prof. Dr. Saeid Kazemzadeh Azad
Civil Eng. Depr., Atilim University

Date: 08.08.2020

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ARY HAMA FARAJ AHMED AHMED

Signature:

ABSTRACT

GENETIC-ALGORITHM-BASED OPTIMIZATION APPROACH FOR TIME-COST TRADE-OFF PROBLEM WITH GENERALIZED PRECEDENCE RELATIONSHIPS

Hama Faraj Ahmed Ahmed, Ary

M.S., Department of Civil Engineering

Supervisor: Asst. Prof. Dr. Saman Aminbakhsh

August 2020, 118 pages

Decreasing the overall cost of a project through reductions in its duration can be considered as one of the main goals of any construction project. Shortening the duration can be achieved by accelerating the construction project by utilization of more efficient construction techniques; that is, by allocation of additional and/or more productive resources. However, expedition of the project schedule will obviously incur extra costs due to implementation of high-price construction techniques. Meanwhile, this reduction of time might be plausible only up to a certain limit. Stemming from the highly challenging nature of this trade-off between the overall cost and the duration of the projects, still no planning software package provides any features for tackling it. The lack of such functionality has encouraged development of numerous optimization algorithm by several researchers for achieving a compromise between the conflicting objectives of time and cost. Despite a great deal of effort, a large body of literature ignore the various types of precedence relationships which are frequently incorporated in practice.

In this thesis a Simulated Annealing-based Genetic Algorithm is proposed for solution of this optimization problem which is referred to as the time-cost trade-off problem (TCTP) in the construction context. The proposed hybrid GA is designed to ensure fast convergence without sacrificing the quality of the solutions found. The proposed optimization method is capable of solving TCTPs with generalized precedence

relationships with realistic overlapping of activities. Performance of the hybrid GA proposed herein is tested over a wide range of frequently used problems and the results are compared with various existing methods. Practicality of this algorithm is also validated by fitting the model to a large-scale real-case construction project. As a result of validation, the utility of the proposed algorithm is illustrated which also highlights how it can help both the client and the contractor in speeding up the project without exceeding the budget.

Keywords: Generalized Precedence Relationships, Genetic Algorithm, Optimization, Simulated Annealing, Time-cost Trade-Off problem.

ÖZ

GENEL ÖNCÜLLÜK İLİŞKİLİ ZAMAN-MALİYET ÖDÜNLEŞİM PROBLEMİ İÇİN GENETİK ALGORİTMA TABANLI OPTİMİZASYON YÖNTEMİ

Hama Faraj Ahmed Ahmed, Ary

Yüksek Lisans, İnşaat Mühendisliği Bölümü

Tez Yöneticisi: Dr. Öğr. Üyesi Saman Aminbakhsh

Ağustos 2020, 118 sayfa

Bir projenin süresinin kısaltılarak toplam maliyetin düşürülmesi, bir inşaat projesinin ana hedeflerinden biri olarak düşünülebilir. Daha verimli inşaat tekniklerinin kullanılarak ve inşaat projesini hızlandırarak süre kısaltılabilir. Ek ve / veya daha verimli kaynakların tahsis edilmesiyle süre kısaltılması sağlanabilir. Bununla birlikte proje takviminin hızı, yüksek fiyatlı inşaat tekniklerinin uygulanması nedeniyle, ekstra maliyete neden olacaktır. Ayrıca, zamanın kısaltılması belli bir süreye kadar makul olabilir. Toplam maliyet ve projelerin süresi arasındaki bu denge, işin doğası gereği son derece zorlayıcıyı olup, henüz bununla başa çıkabilecek özelliklere sahip bir planlama yazılım paketi bulunmamaktadır. Bu işlevsellik eksikliği, zaman ve maliyetin çatışan hedefleri arasında bir denge sağlamak için çeşitli araştırmacıları sayısız optimizasyon algoritmasının geliştirilmesine teşvik etmiştir. Büyük bir çabaya rağmen, literatürün büyük çoğunluğu, pratikte sıklıkla dahil edilen çeşitli ilişkilerini göz ardı eder.

Bu tezde, inşaat bağlamında zaman-maliyet ödünleşim problemi (TCTP) olarak adlandırılan bu optimizasyon probleminin çözümü için Simüle Tavlama Bazlı Genetik Algoritma önerilmiştir. Önerilen hibrit GA, bulunan çözümlerin kalitesinden ödün vermeden hızlı çözüm sağlayacak şekilde tasarlanmıştır. Önerilen optimizasyon yöntemi TCTP'leri gerçekçi örtüşen aktiviteler ve genel bağımlılık ilişkileriyle çözme yeteneğine sahiptir. Burada önerilen hibrid GA'nın performansı, çokça ve sık kullanılan problemler üzerinde test edilir ve sonuçlar, mevcut çeşitli yöntemlerle karşılaştırılır. Bu algoritmanın pratikliği, modeli büyük ölçekli gerçek durum inşaat projesine oturtarak da doğrulanır. Doğrulamanın bir sonucu olarak, önerilen algoritmanın faydası şudur: hem müşterinin hem de müteahhitin bütçeyi aşmadan projeyi hızlandırmada bu algoritmanın nasıl yardımcı olabileceğini vurgulamaktadır.

Anahtar Kelimeler: Genelleştirilmiş Bağımlılık İlişkileri, Genetik Algoritma, Optimizasyon, Simüle Tavlama, Zaman Maliyet Denge problemi.

To my beloved family

ACKNOWLEDGMENTS

I would like to express my gratitude to Atilim University for providing such an opportunity. Especially, the Civil Engineering Department and all its members. I am blessed with having good friends and making new acquaintances, which made it feel like home.

Special thanks to my supervisor Asst. Prof. Dr. Saman Aminbakhsh for his support, commitment, and guidance throughout this study, for always having time to answer my questions despite his busy schedule, and for exemplifying the meaning of an excellent teacher.

I would like to thank my parents for being there every step of the way, and my soulmate for making it an easy, smooth journey, and for believing in my dreams as if they were hers.

Also, many thanks to my friend Mr. Nizar Ahmed, an IT developer for his help throughout this study.

Finally, my extreme love and sincere feelings to the one and only, my bright star and sunshine, my three years old daughter (Taniya) for the energy I get to carry on and overcome any obstacles just by looking in her eyes.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

ACO	Ant Colony Optimization
ACS	Ant Colony System
AoA	Activity on Arrow
AoN	Activity on Node
APD	Average Percent Deviation
Avg.	Average
C	Direct Cost
Cc	Daily incentive amount
Cd	Indirect Cost
Cn	Daily liquidated-damage amount
CPM	Critical Path Method
CPU	Central Processing Unit
DPSO	Discrete Particle Swarm Optimization
DTCTP	Discrete Time-Cost Trade-Off Problem
D	Duration
EC	Expected Count function
EF	Early Finish
ES	Early Start
FF	Finish to Finish
FMOPSO	Fuzzy Multi-Objective Particle Swarm Optimization
FS	Finish to Start
GA	Genetic Algorithm
GASA	Genetic Algorithm with Simulated Annealing
GHz	Gigahertz
HA	Hybrid Genetic Algorithm
HAGAQSA	Hybrid Genetic Algorithm with Quantum Simulated Annealing
HGASA	Hybrid Genetic Algorithm with Simulated Annealing
L	Lag time
LF	Late Finish

LP	Linear Programming
LS	Late Start
MA	Memetic Algorithm
MAWA	Modified Adaptive Weight Approach
NA-ACO	Non-dominated Archiving Ant Colony Optimization
NP-Hard	Non-Polynomial Hard
PD	Percent Deviation
PSO	Particle Swarm Optimization
QSA	Quantum Simulated Annealing
RAM	Random Access Memory
SA	Simulated Annealing
SAM	Siemens Approximation Method
SF	Start to Finish
SS	Start to Start
SFL	Shuffled Frog Leaping
TCT	Time-Cost Trade-Off
TCTP	Time-Cost Trade-Off Problem
USD	United States Dollar
WM-ACO	Weighted Method Ant Colony Optimization

CHAPTER 1

INTRODUCTION

In a construction project, a set of functions has to be performed while taking into consideration the logical relationship between each of them, which can be done in a several different ways, with different time and/or cost; however, due to this variety of work techniques, the decision making for the trade-off between time and cost is difficult especially for big size projects with too many activities. Finishing on time with minimum cost and maximum quality is literally the main objective of any project, furthermore for minimizing the cost we need to expedite the critical activities since the daily overhead cost will be cut as the duration is decreased. Generally, minimizing the duration of a project would simultaneously minimize the cost up to a specific point, after which the additional direct costs invested will start to surpass the amount of indirect cost saved due to duration reduction. With respect to this complicated relationship between time and cost of a project, the need for a definite optimization technique for proper duration reduction is undeniable. A method should be adopted or developed which facilitates perfect selection of activity time-cost alternatives. Finding the optimal selection of time-cost alternatives is known as the time-cost trade-off problem (TCTP). The general TCTP aims to guarantee the optimal duration with the minimum budget for the project.

In the present time, competition among construction companies is to provide a unique schedule to gain competition with a reasonable profit. It is obvious for construction projects to try finishing before defined deadlines in order to gain extra profits. Those who are in charge try to expedite the project with minimum extra expenses by conveying the recession times in networks as well as finding the best range of alternatives to perform activities. This is achieved by finding the best equilibrium between the direct and indirect costs of the project.

With respect to the state of the extant literature, it can be easily realized that TCTP is among the essential aspects of project management. Back in the 1960s Fondahl [1], TCTP research area emerged and ever since, many researchers (Siemens , Feng et al., Yang, Sonmez and Bettemir, Aminbakhsh and Sonmez, Agdas et al. [2]–[7]), have put their special attention to this domain. The TCT problems studied in this domain can basically be categorized into two classes of discrete and continuous problems which can further be grouped under three types of i) deadlines, ii) budget, and iii) time-cost curve problems. The deadline defined for the projects takes into consideration the upper limit for the project duration while minimizing the total cost. Some other projects have budget limit consideration. This type of project takes budget into consideration while minimizing the duration of the project. In any type of the problem, the search for the solution is tracked with as little cost or duration as possible. Since it is virtually impossible to solve this problem manually, many heuristics and algorithms concerning the optimal or near-optimal solution has been developed. The advancements made in computer science technologies have paved the way for establishing many optimization techniques, mainly including exact optimization methods (such as linear and dynamic programming), heuristic algorithms, and metaheuristic algorithms (such as Genetic Algorithm (GA) Feng et al. [3], Ant Colony Optimization (ACO) Elbeltagi [8], and Particle Swarm Optimization (PSO) Yang [4]. However, each proposed method has certain advantages and disadvantages compared against each other. The exact methods needing enormous arithmetic resources are unable to solve complex problems [9], [10]. The studies proposing heuristic methods argue that TCT problems cannot be addressed extensively and efficiently [2], [11]. Besides, a major deficiency was observed in current indicative meta-heuristic algorithms as they also fail to escape from the local Optima [3], [12], and [13].

The basic objective of this study is to develop a modified Simulated Annealing-based Genetic Algorithm for solution of discrete TCTPs that incorporate generalized precedence relationships, that is, Start-to-Start (SS), Start-to-Finish (SF), Finish-to-Start (FS), and Finish-to-Finish (FF). This study attempts to emphasize the effectiveness of optimization algorithms along with the development of computer science technologies which facilitate the project management in locating the minimum project duration. How selection of cost and duration of activities affect the total duration and budget of a project is also demonstrated. The comparisons with traditional approaches accentuate the importance of development of optimization algorithms.

The proposed GA is coded in C# programming language using the Microsoft Visual Studio 2019. Well-recognized benchmark instances obtained from the literature are fed into the proposed GA optimizer. For comparison purposes, optimal solutions of the practiced instances are also obtained from the literature. Performance of the proposed GA is evaluated based the average percent deviation from the optimal results as well as its convergence speed over every instance used. In addition to the existing problems, data of a large-scale real-case construction project involving generic logical relationships is also entered into the model results of which further validate practicality of the GA developed in this study.

CHAPTER 2

LITERATURE REVIEW

2.1 Scheduling

The project success highly depends on the how well the organization and how adequate the schedule are set. Although planning and scheduling are considered two terms which are frequently used interchangeably, scheduling is thought to be a part of planning. Scheduling is an outcome of asking ‘when’ during planning, and its goal is to integrate activities relations to complete the project with an acceptable time, minimum cost, and lowest risk. A poor scheduled project may cause project completion’s delay, cost overrun, and it will lead to claims, arguments, and disputes. Therefore, appropriate approaches should be taken in planning and scheduling [14], [15]. Construction projects consist of complex sequential processes, where technological solutions should be evaluated and improved continuously to handle the uniqueness of project activities. Therefore, more advanced methods have been developed to find solutions for most of the organizational problems. Meanwhile, network scheduling techniques are used frequently in planning and scheduling processes, and the most popular method is the Critical Path Method (CPM) [16]. The objective of project control is to guarantee the project completion time with minimum budget while satisfying all the project constraints. It is a complicated process for managers in execution, which needs frequent tracking of progress, comparing with the original plan, and taking actions rapidly when needed. As a result of advancements made in project planning and controlling methods – such as the Critical Path Method (CPM), Gantt Chart/Bar Chart, and Program Evaluation and Review Technique (PERT) – there have also been numerous software packages developed on basis of such methods. Microsoft Project, Primavera, Asta Power Project, etc. are some of the best know commercially available software for planning. Nonetheless, none of the existing software, despite their extensive usage in practice, come with any tools or features for time-cost analysis. Resultantly, cost overruns still remain a major issue in project management since no certain tool is commercially available [17].

2.2 Network Scheduling

Researchers in the field of management has always been looking for new and best control methods to deal with complexities, enormous data, and narrow deadlines that are features of competitive industries. Managers also aim for better techniques for showing technical and cost data to customers. Thus, so many methods are used for achieving these goals including the methods named in section 2.1. The most common method is obviously the Critical Path Method (CPM) for which two different diagramming methods of ADM (Arrow Diagramming Method) and PDM (Precedence Diagramming Method) are used [18].

2.3 Critical Path Method (CPM)

CPM became widely used as a reliable tool for planning and scheduling complex projects in the late 1950s [19]. It is essential to use CPM while scheduling the construction process for projects to be done gainfully and on time. Because of the advantage and importance of CPM, it's used in all industries differently, including construction, which has significantly been used for the last three decades; therefore, many improvements were made in both computer hardware and scheduling software [20], [21]. Although the calculations of CPM are simple and easy to understand, CPM-based scheduling is a challenging process. At the stage of planning, the CPM network may have complex relationships that complicate the scheduling process. Furthermore, the CPM algorithm cannot solve the multiple constraints in a project, such as a deadline and a resource limit, thus researchers have introduced some processing techniques such as time-cost trade-off, resource leveling, etc. This difficulty adds to the perception that CPM and existing software are useful for organizational and reporting purposes but not as a decision support tool to reflect and react to reality [22]. Critical Path Analysis (CPA) concept is simple. Jobs related to each other are laid out in paths with respect to their order of performance. Start time and finish time of each task is determined through consecutive forward and backward pass processes, then these times are summed along each path to find the longest path(s), that designates the project duration. In other terms, the critical path is the longest path through the network and determines the duration of the project. It is also the shortest amount of time necessary to accomplish the project, and also there may be more than one critical path because of identical lengths of different paths. Furthermore, tasks of the critical path are known as critical activities; any delay in those activities will lead to delays in the whole project. Meanwhile, the activities out of the critical path are known as noncritical activities that have relatively flexible start-finish times due to having floats. Thus, any delay within the flexibility constraint of those activities will not affect the project duration.

In light of this idea, CPA supply information to; show relationships between tasks, schedule the expected occurrence of every job in a project, calculate the chance of meeting deadlines, calculate resource usage and compare it against availability, find the best project duration, and to compute distribution of costs. In addition to the foregoing aspects, CPA also compile information which help; determining the feasibility of the project execution with regard to cost and resource needs, setting strategies, defining duties of different task groups, tracking progress in order to avoid deviations from original plans and project objective, predicting probable challenges and difficulties, and revising the plan when necessary [23].

2.4 Factors for Project Success

Success factors are defined as factors that affect and specify project success [24]. In the 1970s, project success concentrated on execution, time reduction, cost improvement, and delivery system [25]. Meanwhile, in the late 1980s, the planning quality and delivery were also specified to have significant roles in project success [26]. Furthermore, De Wit [27] and Cooke-Davies [28], considered the factors of success as those interventions for the system that leads directly or indirectly so that the project succeeds [29]. However, Critical Success Factors (CFS) added other factors into account, such as organizational and stakeholder views. More recently, the CSF structure improved on the basis that success depends on stakeholders and interactions between supplier and customers [30].

2.4.1 Cost Objective

Azhar et al. [31] state that "*Cost is among the major consideration throughout the project management life cycle and can be regarded as one of the most important parameters of a project and the driving force of project success*.". Furthermore, cost-effectiveness is an essential technique in project management effort.

2.4.2 Performance (Quality) Objective

Quality has usually been utilized as an elective (but less palatable) title for the execution venture objective. Besides, a common understanding of the project or item quality summons up to several things in our imagination. Characteristics of quality performance depend on the kind of project or product [32].

2.4.3 Time Objective

It is intuitive of project management that projects may be defined as 'succeeded' if the project is completed at a specified time, within calculated budget, without any disaster, within standards and quality specifications, and client contentment. Also, 'construction time' is now an essential factor because it usually serves as a critical benchmark for evaluating project performance and the capacity of project planning [33]. Construction time can be defined as the total time that the project spends from the start of the site works to the accomplishment and handover of a building to the client. Meanwhile, construction time is specified before the start of project execution. Furthermore, in view of achieving this objective, construction time can be compressed by applying different construction techniques [34].

2.4.4 Project Management Triangle or Iron Triangle

The Triangle of objectives created by Dr. Martin Barnes in the mid-1980s shows that time, cost, and quality objectives are interconnected. While focusing or settling, an angle of the triangle impacts the other two angles (Figure2.1) [35].

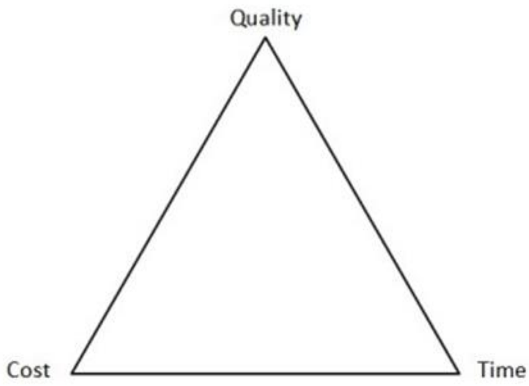

Figure 2.1 The Original Triangle of Objectives Introduced by Barnes [35]

The new version places the quality objective in the center, making an obvious difference between quality and scope. In this version of triangle of objectives, the scope is the specifications and deliverables; however, quality has moved to the core of this triangle which simply indicates quality affects and gets affected by the other objectives (Figure2.2) [36].

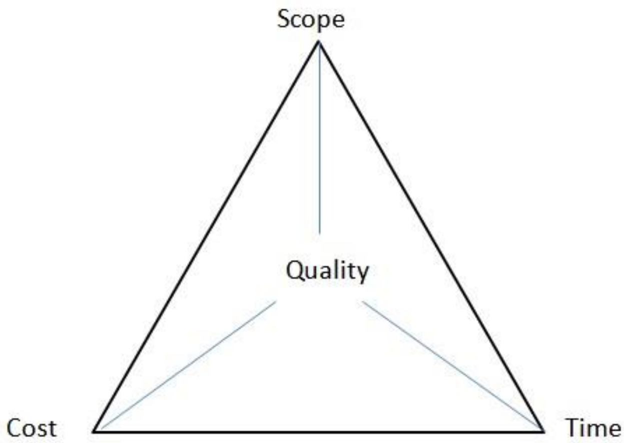

Figure 2.2 Revised Triangle of Objectives [36]

The triangle of objectives improved by Klein and Ludin put people at the core. They call it the four variables of project success. They believe if people are not treated as a critical element, the project will not succeed, despite the existence of perfect plans, organized structure, and sufficient controlling mechanisms (Figure2.3) [37].

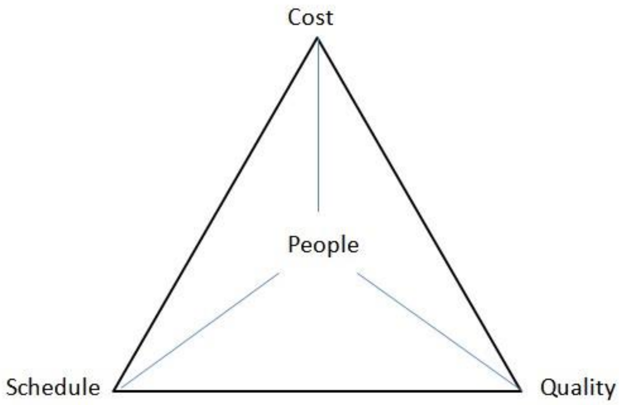

Figure 2.3 The Triangle of Objectives Proposed by Kliem and Ludin [37]

2.5 Time-Cost Relationship

Two of the main objectives in construction projects are time and cost. In the construction projects contractors commonly use historical information and experiences to estimate the project duration and cost. Generally, expediting the activity completion time need more resources to assign and, as a result, a higher cost [38]. Furthermore, the trade-off between time and cost make planners to challenge achieving the best construction plan for finding the optimum time and cost to complete a project [39].

2.5.1 Effect of Project Delays on Direct Costs and Indirect Costs

The direct project costs of equipment, machinery, and workers' wages are time-dependent. Time distension is one factor that can be predicted to inflate cost because of a latency in the start and finish of a job. This time extension would potentially cause a rise in material prices, and also to increase salary payments. Other factors such as idle time or time waste possibly brought by inefficient working due to a shortage in material, loss of information, or bad planning and organization are not easy to quantify. Such factors can also lead to a delayed work. Apparently, delays in any individual work item aggregates to a major delay in the whole project. On the other hand, the indirect costs which are generally assumed to remain constant over the life-span of the project – including overhead costs of administration, management, accommodation, and facilities – will be increased on daily basis as the project runs behind the schedule. That is, these costs will have to carry for a longer time than planned; as a result, the expenditures will presumably exceed the original budget [35].

2.5.2 Project Time-Cost Trade-off

Activity duration depends on the kind and size of the resources that are assigned and varies accordingly. Furthermore, allocating more workers to a specific activity will generally result in a shortening of the duration. Meanwhile, expediting the project time may lead to higher costs and poor quality. However, the compromise between time and cost is an interesting subject in economy markets nowadays. This relation between time and cost is called the time-cost trade-off [40].

2.5.3 Time-Cost Trade-off (TCT)

Since the development of the CPM, the time-cost trade-off has been the topic of comprehensive research in project scheduling. The trade-off includes an inverse pattern between the duration of a task and its quantity of non-renewable resources used, for example, for speeding up an activity, more resources need to be assigned to it which comes at an additional cost. This well-known subject has been studied under different presumptions (e.g., Linear, Non-Linear, and Discrete time-cost relation function) [41].

2.5.4 Activity Time-Cost Trade-Off

Generally, the less costly the resources, the more the duration it takes to do an activity. Due to this trade-off between time and the direct cost, the direct cost will usually increase while shortening the duration of a task, which involves: the labor cost, tools, and machinery. It should be considered that the number of resources applied and the activity duration is generally inversely linked to each other. Figure 2.4 demonstrates a simple explanation of a relationship between the activity duration and its direct costs. Taking into consideration only this task in isolation and without considering the deadline of the project, a decision-maker would choose a duration with minimum direct cost, named the normal duration. On the other hand, a manager may select the minimum duration to complete an activity which is called the crashed duration, and requires the maximum investment [42].

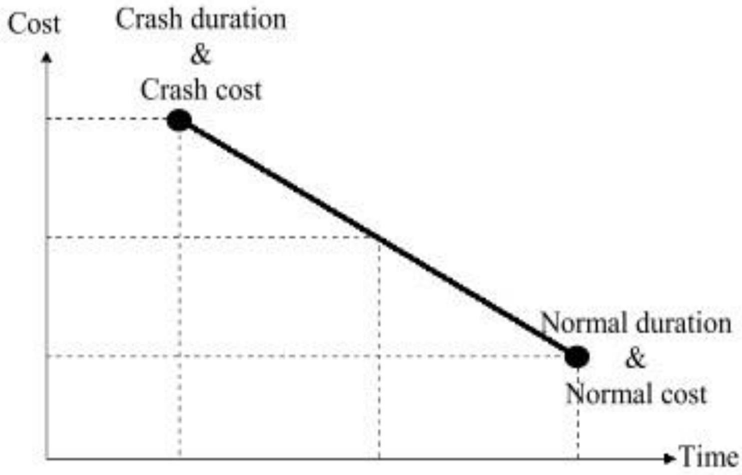

Figure 2.4 Linear Time-Cost Relation

In Figure 2.4 the continuous linear relationship indicates that any point between these two extreme points can be chosen. It happens that some of the middle points may show an ideal trade-off between time and cost for this task. The line aggregates the normal point (lower point) and the crash point (upper point). This slope of the line is the cost slope of the activity. The slope of a line can be found mathematically by having the coordinates of the normal and crash points. Cost slope can be computed through dividing deduction of the crash cost and normal cost by the deduction of the normal duration and crash duration.

As shown in Figures 2.4, 2.5, and 2.6, the minimum direct cost needed to complete a task is called normal cost, and the corresponding duration is called the normal duration. Crash duration is the shortest possible duration need for executing the activity, and the relevant cost is called crash cost. In Figure 2.5 it is shown that not necessarily the relation between time and cost of an activity is always non-linear. As shown in Figure 2.6, this relation can sometimes be discrete too, providing the decision makers only a limited number of options to choose from. Usually, a planner starts estimating and scheduling procedures by taking on the lower-cost option [40], [43].

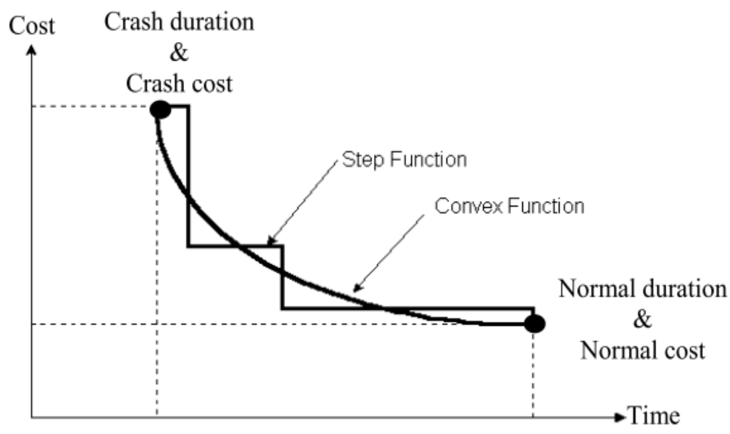

Figure 2.5 Non-Linear Time-Cost Relation

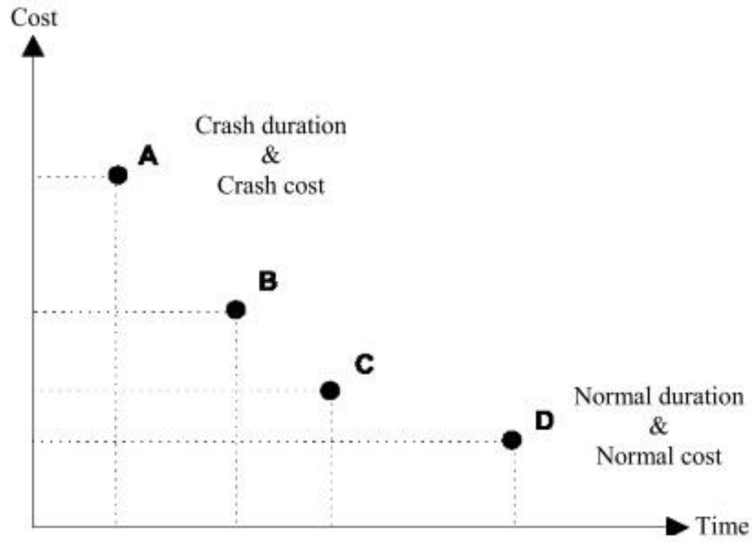

Figure 2.6 Discrete-Time-Cost Relation

2.5.5 Project Time-Cost Relationship

It is the summation of direct and indirect costs needed for performing the tasks of the project that represents the total project cost. While direct costs are costs of equipment, labor, machinery, and subcontractors, the indirect costs are the costs necessary for performing a work that are neither directly related to a specific activity nor concerned with a particular project. If the project is scheduled by choosing the least-direct-cost option for each task, then the project duration might be longer than the limitation agreed in the contract. Such a delay may lead to significant penalties related to late project completion time. So, the planners do a trade-off between time and cost by performing what is named as the time-cost trade-off analysis to minimize the project duration. Besides, this process can be achieved by reducing the duration of certain critical activities on the critical path [44]. Figure 2.7 explains the relationship between direct costs, indirect costs, and the project duration. The sum of the direct cost of the project activities equals the direct cost of the project; consequently, decreasing the project duration will cause the project's direct cost to increase. Moreover, by reducing project duration, the indirect cost will decrease linearly since the indirect cost is often a function of the project duration.

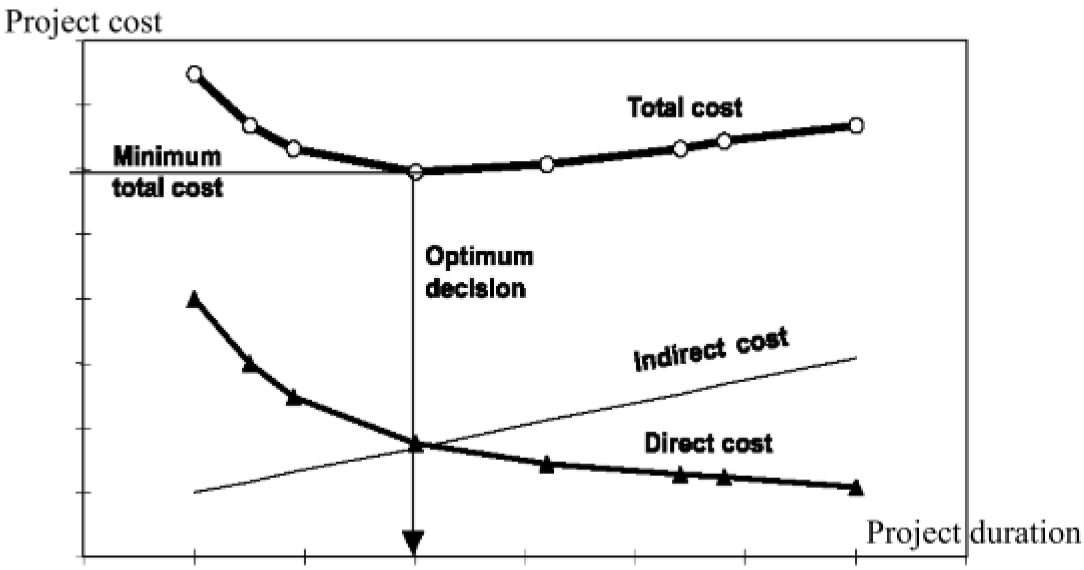

Figure 2.7 Relation between Duration and Total Cost of a Project [45]

The total time-cost relationship of a project can be determined by aggregating the direct cost and indirect cost amounts together, as explained in Figure 2.7. The optimum project duration correspond to the point for which the total cost is the minimum possible amount [43].

2.5.6 Shortening Project Duration

Crash time is the minimum time to complete a project; this minimization can be achieved by setting minimum duration options for all the activities. This schedule will also represent the all-crashed cost of the project. As mentioned before, the critical path has critical activities in which any change in their completion time will directly affect project duration. On the other hand, for some activities that are not over the critical path, i.e., for some of the non-critical activities, cheaper options with longer duration amounts can be selected. This duration extension may only be applied until the overall project duration remains unchanged. That is, any extension beyond the float amount of an activity is not preferable since it will delay the project completion. By using different techniques and assumptions, the optimum crash schedule can be determined and thereby costs reduced. Furthermore, some of these techniques are used for shortening the duration of the project without exceeding a certain budget; in other terms, minimum project duration with optimal cost. The classic procedure for shortening project duration can be summarized as the following steps:

- Constructing the project network, carrying out CPM calculations, and finding the critical path using all regular durations and costs for all the activities.
- Calculating the cost slope for every activity, identifying the critical activities with the minimum cost slopes that still have accelerated options to choose from.
- Proceeding with the duration reduction of the critical; if multiple critical paths exist, the activity that provides larger duration reduction is shortened first.
- Updating cost-slopes and carrying out CPM calculations, terminating when either the is goal achieved, or no further shortening is possible. Results can then be presented graphically by drawing project duration against cumulative cost.
- Total cost curve can be obtained by adding the constant amount of daily indirect cost to the direct costs and by using this graphical representation the relation between duration and total project cost can clearly be visualized [46].

2.6 Time Cost Trade-off Problem (TCTP)

Time-cost trade-off problem (TCTP) is a multi-objective problem. It involves finding the best combination of execution modes for the activities to come up with a desired balance between the project time and cost [47]. Mathematical and heuristic-based approaches were early efforts for finding TCTP. Despite their strength in finding the optimal and/or near-optimal results for problems with linear time-cost relations, such solutions had major weakness in solving discrete TCTP. On the other hand, meta-heuristic techniques were demonstrated by Feng et al. [3] to have higher performance in this area. Despite this fact, it is argued that meta-heuristic methods may not always be able to achieve a global optimum solution. They are designed to search only portions of the solution space intelligently rather than exploring the whole solution space. This provide such methods with the ability to produce comparatively reasonable solutions for the large-size problems [3]. Various techniques of solving TCTP are described thoroughly within the following sections.

2.6.1 Mathematical Programming Models

Mathematical programming strategies model TCTP with mathematical formulations and use linear programming, dynamic programming, or integer programming methodologies [48]. Linear time-cost relationship of activities used as a tool for solving the TCTP by Kelly [49]. Likewise, other techniques too used linear programming as a tool to solve the TCTP such as the approaches used by Hendrickson and Au in [50] and Pagnoni [3]. There is a general consensus that the appropriate approach for problems with linear time-cost relation is the linear programming technique; though, expectedly it fails to solve problems with discrete time-cost relationships. There have been studies (e.g., Meyer and Shaffer [51] and Patter-son and Huber [52]) on solving problems with both linear and discrete relationships by using mixed-integer programming. However, it has been emphasized that for projects with higher number of activities and/or complex networks substantially more computational effort is required. Burns et al. [9] followed the hybrid approach, which applied linear programming to find the exact solution for any desired duration.

Robinson [53], Elmagraby [54], and De et al. [55] utilized dynamic programming to solve the TCT problem for networks that split to parallel subnetworks [3], [56].

2.6.2 Heuristic Methods

Heuristic methods depend on rules of thumb which are prejudicially deemed to be deficient in mathematical accuracy. They give reasonable solutions but they do not guarantee optimality. Examples of a heuristic approach are Fondahl's method [1], Prager's structural model [57], Siemens's effective cost slope model [2], Moselhi's structural stiffness method [11], and a recently proposed Activity Uncrashing Heuristic method [58]. These methods give satisfactorily good solutions, and in some cases, they are even able to find the optimal solutions. Most heuristic methods presume only linear-time-cost relationships for activities and the solutions achieved by heuristic methods do not provide the range of possible solutions, making it hard to test with various scenarios for what-if analyses [59]. Though, none of these remarks would be valid for the Activity Uncrashing Heuristic proposed by Sonmez et al. [58] which actually is designed to provide virtually the full range of possible solutions for the discrete problems.

2.6.3 Meta-Heuristic Methods

Meta-heuristics are general-purpose techniques that could be applied for solution of a vast domain of problems especially where exact or heuristic methods fail to provide adequate solutions within reasonable computation time. The major problem with the meta-heuristics though is their inability in untrapping themselves from the local optima chiefly for the complex optimization problems. Some meta-heuristics methods have preliminarily been exercised and been quite successful in optimization problems such as Genetic Algorithm (GA) [3], Simulated annealing (SA) [60], Scatter and Tabu Search [36], Ant Colony Optimization (ACO) [12], Particle Swarm Optimization (PSO) [8], Memetic algorithm (MA) [8], and other techniques used for optimization purposes.

2.7 Optimization

Optimization is the process of achieving the best solution (maximum or minimum of a function) to a given objective or objectives while satisfying certain constraints. Moreover, optimization techniques define the best alternative among the solutions obtained under several constraints. The plants and animals use optimization instinctually in such a manner that minimizes the path for finding foods or maximize energy for hunting, which is briefly illustrated in Figure 2.8 [61].

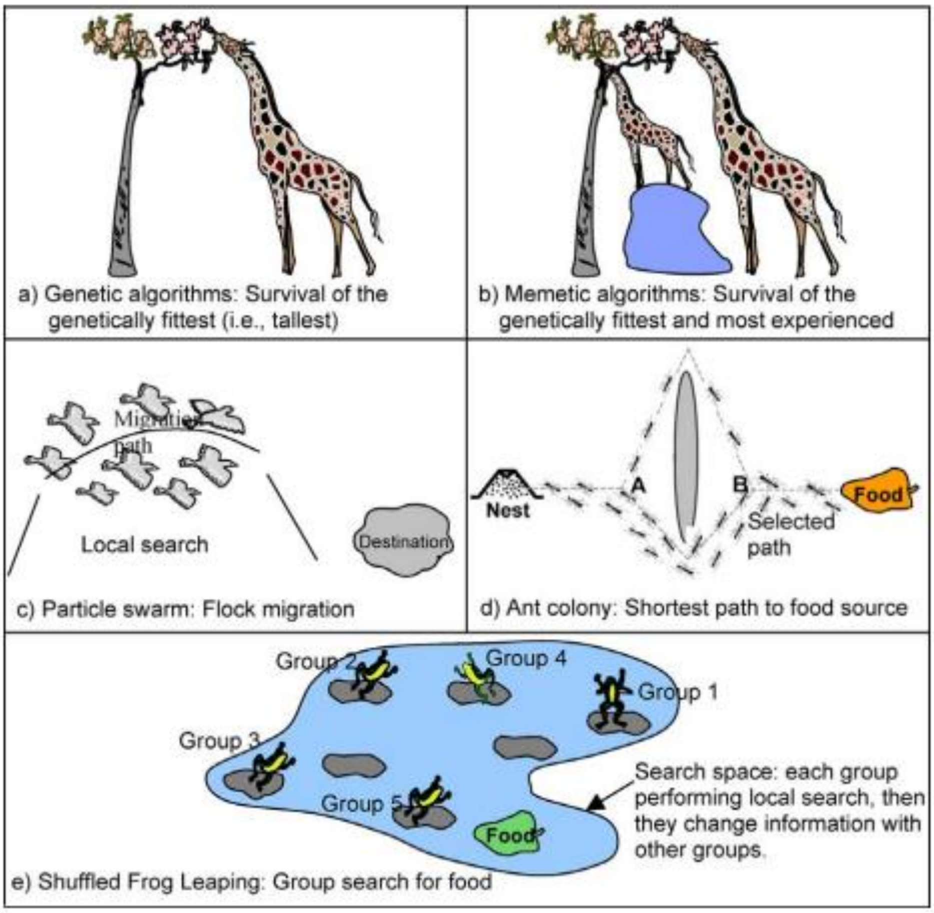

Figure 2.8 Natural Evolutionary Systems Diagram [8]

2.7.1 Optimization Methods for Solving TCTP

Every day we face different kinds of decision-making problems as directors, executors, designers, planners, administrative officers, etc. In such situations, the conclusive decision is generally made over several district processes; however, sometimes, they might not be realized explicitly. Figure 2.9 displays the decision-making process model [62]. It implies that the last decision is made during three models; the structure model, the impact model, and the evaluation model.

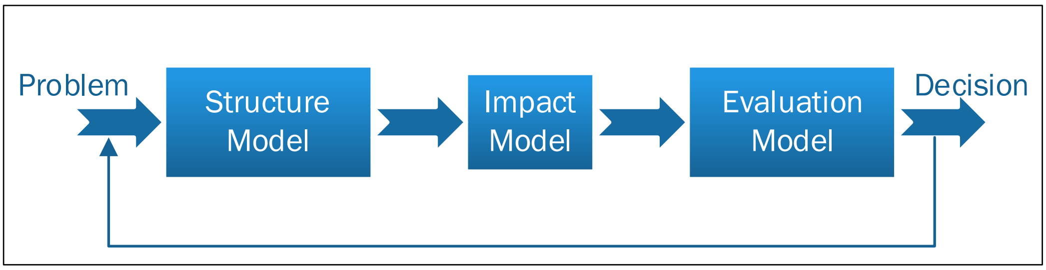

Figure 2.9 Conceptual Model of the Decision-Making Process [62]

For finding the optimal or near-optimal solution, the decision-makers need to find a suitable technique to fit to the problem; in this vein, advancements were made in optimization techniques for solving single and multi-objective problems [62]. Similarly, a vast array of techniques has been adapted for solving TCT problems, some of them give the exact solution (optimal solutions), but they have a very limited performance for implementation of large-scale problems. One of the exact approaches was proposed in 1965 by Mayer and Shaffer , they tried an application of mixed-integer programming for solving time-cost curve problem [63]. At the beginning of the 20th century, the emphasis was on time-based completion [64]. In 1995 De et al. revised the discrete time-cost tradeoff problem suggested by Blackburn, and Bockerstette and Shell [55]. They also pointed out different versions of the problem discussing their complexities, thereby they proposed an approach for the deadline problem and introduced a new dynamic programming formulation considering DTCT problems. While in the late 20th-century, more accurately in 1998, Demeulemeester et al. [10] introduced a new exact procedure for DTCT problems based on the Activity-on-the-Arc (AoA) network. They used Visual C++ coding language for implementing the algorithm. The objective of the procedure was presentation and validation of an alternate optimal procedure for the DTCT problems. Besides, they used a horizon-varying approach, which included the iterative optimal solution of the problem by minimizing the sum of all activities' resources used, while satisfying the project deadline. Furthermore, they used the branch-and-bound algorithm which depends on the logic of calculating a convex piecewise linear underestimation of its DTCT curve for each activity. This computation is done by measuring vertical distance for each activity and its correlated mode, which measures convex underestimation quality. They used two basics to perform the calculation, the first rule is computing each activity and its correlated mode as a minimum between two vertical distances, and the second one is computing the vertical distance between the correlated mode and the linear interpolation between the nearest mode. Moreover, branching is complete by distinguishing the activity with the largest vertical distance and splitting its collection of modes into two subsets. The validation of this approach is confirmed by comparing the achieved results with the results obtained earlier by Demeulemeester et al.'s study [10].

After the first introduced branch-and-bound algorithm by Demeulemeester et al. [10] for solving DTCT problems, another branch-and-bound algorithm was also suggested by Yang and Chen [65]. Vanhoucke [66] codded an algorithm in Visual C++ for time/switch constrained discrete TCT problems; their study mixed the branching algorithm in which activities that had lead times greater than their durations were considered. Also, their analysis at that time showed the difference between classical activity network in some management point of view and the critical path and float in this context. The results obtained by Yang and Chen [65], Vanhoucke [67], and Vanhoucke [66] validated effectiveness of this algorithm [65]–[67]. Although the exact algorithms need significantly more time for computing large-scale problems, they are widely used by researchers for checking performance and validation of the performance of heuristics and meta-heuristics algorithms. For instance, Sönmez and Bettemir [5] used mixed integer programming formulation of De et al. [55] for comparing their results for large-scale problems. Also, Chassiakos and Sakellaropoulos [68] used linear-integer programming (LP/IP) formulation for finding optimal solutions and comparing the exact result with the obtained results by the approximate method [68], [69].

As mentioned earlier, heuristic methods give good solutions but not always optimality of the results is guaranteed. In 1961, Fondahl presented different versions of CPM; Fondahl's method became the preferred critical path method in the recent years [1].

Moreover, another heuristic method was proposed in 1971 by Siemens [2], which offers an approach known as Siemens Approximation Method (SAM). This algorithm aims for the shortening project duration when the project duration exceeds the desired limit. The procedure is for calculating which activity to expedite and by what amount. Moreover, implementation of SAM is considerably less complex than analytical methods. From the comparison of SAM's results with other complex analytical methods, it was concluded that SAM achieved good results and nearly the same solutions as the methods in comparison. Because of its simplicity, SAM is claimed to be easily implemented by hand or by computer and it is also capable of solving convex non-linear cost slopes [2]. During a search for better performance of heuristics method, Sonmez et al. [58] proposed an uncrashing heuristic with noncritical activity rescheduling method for solving DTCTP. They implemented different types and size of instances focusing on large-scale problems, reaching solutions within a reasonable CPU time. Despite its high performance, it requires long CPU times for very large size projects, particularly for the Pareto front problem.

Meta-heuristic techniques are vastly used by researchers for solving simple TCTP. Numerous meta-heuristic methods have been implemented to find optimal solutions for different versions of time-cost trade-off problems. Although the main benefit of using meta-heuristics is deemed to be their ability in solving large-scale problems, some of them are only able to solve TCTP to global optimality for small size problems. No meta-heuristic can be claimed to locate global optima at each run because they rely on random search and the results of each run would most likely be different than other trials. In this respect, we cannot say they always guarantee optimality. Furthermore, meta-heuristic techniques are relatively simple to understand and implement. Some present meta-heuristic algorithms have the main drawback of incapability in improving the existing best solution even with an increased iteration number. In other terms, these methods tend to stuck into local optima.

As one of the best know precursors in the nature inspired optimization algorithms, Genetic algorithms of Holland [70] can be mentioned. This method was inspired by the natural evolution and the process of natural selection and has ever since been extensively applied for numerous optimization problems under various fields. Feng et al. [3] implemented GAs for TCTP and developed a novel algorithm for optimal selection of construction time-cost alternatives. Validation of this method was made by examining many test cases; they introduced and implemented a widely used 18-activity network as a test case. For this problem their GA was able to find varied results in each trial due to the nature of the algorithm; nevertheless, it is reported to be able to find 95% of the optimum solutions [3].

As mentioned earlier, GA is widely applied by many researchers as a potent and efficient procedure to find global optimum. Zheng et al. [71], for instance, developed a new multi-objective GA-based method to optimize time and cost simultaneously. They modified GA by introducing the Modified Adaptive Weight Approach (MAWA) which helps extend GA beyond single objective time-cost optimization. This method was tested over a 7-activity network which revealed efficiency of the proposed method in solving multi-objective problems. Though, MAWA is discussed to share the inherent randomness problem of GA-based models that could impact their reliability which ultimately makes decision-makers determine the final best solution without having adequate data for supporting their selection [71]. Zheng et al. [72] presented an upgraded version of previously proposed GA-based time-cost optimization model. Three different modules of the proposed GA were implemented using the 18-activity network described by Feng et al. [3]. Promising results were reported especially when more than 300 generations were used [72]. Eshtehardian et al. [73] proposed a GA-based approach which also incorporated Fuzzy set theory of Zadeh [74] to assist the decision-making process. It helped choosing a Pareto front solution based on the risk attitude of the decision maker. For experimenting and verifying the performance of this approach, two different models were implemented for different values of α using the 18-activity network of Feng et al. [3], [73].

It is observed that utilization of sole-GAs for optimization purposes would require higher iteration numbers for achieving suitable outcomes. To cope with this issue, in 2009 Bettemir [75] developed hybridized GAs with improved convergence efficacy for TCT problems. Bettemir [75] examined complementary methods to increase convergence by integrating Simulated Annealing (SA) and Quantum Simulated Annealing (QSA) with the GA. Consequently, significant savings have been gained in terms of computation time. For checking the validation and performances of the proposed algorithms, the authors implemented different types of instances including 7-activity network, 18-activity network, 29-activity network, 63-activity network, and 108-activity network, 360-activity network, 290-activity network, and 630-activity network problems with two different cases for each problem. It is shown that sole-GA can only solve the 7-activity network to optimality while the proposed hybrid methods exhibit better performance with less computation time [75]. Also, Sonmez and Bettemir [5] proposed a Hybrid algorithm in which they integrated GAs together with SA and QSA to increase the convergence possibility of optimality, especially for the discrete TCTP. They implemented different types of instances including 18-activity network, 29-activity network, 63-activity network, 290 activity, and 630-activity network problems with two different cases for each problem. Results further proved the better performance of HA compared with sole-GA [69].

In 2017 Albayrak [76] made a comprehensive review on metaheuristic methods used to solve TCTP. He found that GA had kept its usability in subsequent studies for a long time and was still preferred with major modifications and hybridization. Furthermore, Agdas et al. [7] proposed an approach to solve TCTPs by using GAs, with the aid of graph theory method for solving the critical path problem; thereby, increasing the GAs computational efficiency significantly. This study focused on large-scale construction TCTP by implementing several types of large instances including 630-activity network, 1800-activity network, 3150-activity network, and 6300-activity. Agdas et al.'s proposed algorithm had better results than those found by Sonmez and Bettermir [5].

Meta-heuristic methods are not limited to various interpretations of GA. There exist other meta-heuristic algorithms which too are developed for solving optimization problems. One of these methods include Ant Colony Optimization (ACO) introduced by Colorni et al. [77]. ACO imitates the behavior of ants searching for food by taking the shortest possible path between the target and their nests. For solving the discrete Time-Cost Optimization (TCO) problems Ng and Zhang [12] proposed a multi-objective approach known as the Ant Colony System (ACS) also for which the Modified Adaptive Weight Approach (MAWA) is adopted. This algorithm was compared with other analytical methods over the well-known 18-activity example of Feng et al. [3]. Its tendency to premature convergence was stated to be the main drawback of this algorithm [12]. A similar approach was introduced by Afshar et al. [78] known as the Nondominated Archiving ACO (NA-ACO) method. This method assigns separate ant colonies to each of the objectives and information exchange is made possible between the individual colonies. Each colony evaluates and uses the exchanged information in the pheromone update process. The 18-activity instance of Feng et al. [3] is fitted to NA-ACO using three different indirect cost values and the results are compared with Weighted Method ACO (WM-ACO) and GA of Zheng et al. [72]. Although NA-ACO is shown to perform better than the algorithms in comparison, it is not able to find the full Pareto front of the practiced problems [79]. In 2012 Zhang and Ng [80] developed the ACO-based TCO model for time-cost optimization problems. By implementing a well-known 18-activity network it provided better solutions compared with the results of Zheng et al. [72].

The competition for finding a suitable algorithm to solve TCT problems properly makes many researchers propose different optimization techniques. In 2007 Elbeltagi et al. [81] proposed Modified Shuffled Frog-Leaping (MSFL) algorithm for unraveling complicated large-size problems. This method involves several local searches together with a mechanism for information exchange among them. An acceleration parameter is also introduced to help MSFL untrap itself from the local optima which governs the balance between the local search hand and the global exploration. Performance of MSFL is experimented over the 18-activity problem of Feng et al. [3] and two well-known benchmark test problems of F8 and F10. Results are shown to be better than those of SFL and GA algorithms [81].

Particle Swarm Optimization (PSO) is yet another evolutionary algorithm (EA) frequently used for finding the solutions for different optimization problems. This method is inspired by the behavior of the bird flocks that share information as they search for food on their path to their destination. PSO was first introduced by Kennedy and Eberhart [82] which was slightly modified later by Shi and Elberhat [83]. The modified version incorporated a parameter known as inertia weight which regulated the balance between the local and the global search. PSO was compared against four other meta-heuristic algorithms by Elbeltagi et al. [8]. The four EAs included GA, MA (Memetic Algorithms which is basically the same as GA but applies local search on chromosomes and offsprings), ACO, and SFL. The performance of all the five EAs was tested using the 18-activity problem of Feng et al. [3] and two well-known benchmark test problems of F8 and F10. PSO was shown to perform better than any of the practiced EAs in the terms of solution quality and convergence speed [8]. In 2007 Yang [4] proposed an elitist PSO algorithm to facilitate bicriteron time-cost trade-off analysis. the elitist PSO algorithm has been considered as effective and efficient algorithm in obtaining the Pareto front. On the other hand, PSO has also been practiced for DTCTP optimization by Aminbakhsh and Sonmez [6], proposing a Discrete Particle Swarm Optimization (DPSO) method for solving large-scale problems. They implemented different types of instances including 18-activity network, 63-activity network, and 630-activity network problems. Their proposed DPSO achieved high-quality solutions for small-, medium-, and large-scale DTCTP with high-performance, especially for medium and large-scale problems.

Table 2.1 illustrates the optimization techniques proposed by various researchers for solving TCTP most which were elaborated in this section.

Table 2.1 Techniques for TCTP Analysis

Method	**Description**	**Author(s)**	**Remarks**	**Drawbacks**	**Performance Comparison**	**Year**
Exact	Linear programing	Kelly [49].	May provide an optimal solution.	-Difficult to formulate. -Used to small problems only. -Generally, suppose linear, instead of discrete, relationship between time and cost.	Na	1961
Heuristic	Siemens Approximation Method	Siemens [2].	-Simple to comprehend -Gives good solutions. -Applied for large-size projects	-Shortage of mathematical rigor. -Do not provide optimal solutions. - Generally, suppose linear, instead of the discrete, relationship between time and cost	In 1971 Siemens compared SAM with linear programing results, these comparisons revealed that the solution equally good nearly the same as the solution.	1971
Meta Heuristic	Genetic Algorithms (GA)	Feng, Liu, and Burns [3].	-Strong search method. -It is able to use a discrete relationship between time and cost. -utilizable to large problems	-Random search is time-consuming. -unable to point when or if an optimal solution is reached.	Case study compared with the cost-slope of siemens (1971). The GA is capable of giving several equally good solutions compared to the single solution by the heuristic method	1997

Table 2.1 Techniques for TCTP Analysis (continued)

Method	Description	Author(s)	Remarks	Drawbacks	Performance Comparison	Year
Meta Heuristic	Genetic Algorithms (GA)	Zheng, Ng, and Kumaraswamy [72].	A genetic algorithm that enrolls MAWA, Pareto ranking, and Niche formation to prevent genetic deviation, manage selection pattern, and spend diversifier, respectively.	Resources are considered to be unlimited with the situation that an extra amount can only be reached at a higher price.	Na	2005
	Genetic Algorithms (GA), Memetic Algorithms (MA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Shuffled frog leaping Algorithms (SFL).	Elbeltagi, Hegazy, and Grierson [8].	Meta-heuristic algorithms (GA, Ma, PSO, ACO, and SFL) are compared showing weak performance of GA along with the strength of PSO methods	Na	Different evolutionary-based search approaches were showed include GA, MA, PSO, ACO, and SFL. Using Visual Basic programs for each technique, two referenced continuous optimization problems were implemented. Among the compared algorithms	2005

Table 2.1 Techniques for TCTP Analysis (continued)

Method	Description	Author(s)	Remarks	Drawbacks	Performance Comparison	Year
Meta Heuristic					the PSO algorithm generally showed a better performance than other methods in terms of success rate and being second best in terms of processing time.	
	Shuffled Frog-Leaping algorithm (SFL)	Elbeltagi, Hegazy, and Grierson [81].	A modified shuffle frog leaping (SFL) algorithm that combines a time-variant parameter to prevent sticking into local optimal.	Na	Na	2007
	Particle Swarm Optimization (PSO).	Yang [4].	-The Pareto front can be reinforced to the total cost curve by summating the indirect cost and the total cost curve. - What-if analysis can be accomplished regarding various deadlines or budgets.	Na	Na	2007

Table 2.1 Techniques for TCTP Analysis (continued)

Method	Description	Author(s)	Remarks	Drawbacks	Performance Comparison	Year
Meta Heuristic	Ant-Colony Optimization (ACO)	Ng and Zhang [12].	-The Ant Colony System (ACS) model can solve the multi-objective Time Cost Optimization, (TCO)problem successfully by applying the MAWA, which incorporates the time and cost into a single objective. - the ACS model can finish the searching process by using fewer ants and minimal iterations. This has shown that the ACS has a better searching ability than the ACO model.	-The ACS-TCO's weaknesses are its inclination to early convergence. With the increase of iteration, the solutions reach a group of values that are not globally optimal. - Parameters selection is another side that could impact the performance of the ACS-TCO model is. , as they can affect the convergence speed and/or the quality of solutions.	Comparing the ACS-TCO model and Elbeltagi's ACO- based time–cost trade-off model, the results show the proposed model can obtain the best solution with a fewer number of iterations, and this could minimize the computation time, especially for more complex networks.	2008

Table 2.1 Techniques for TCTP Analysis (continued)

Method	Description	Author(s)	Remarks	Drawbacks	Performance Comparison	Year
Meta Heuristic	Ant-Colony Optimization (ACO)	Xiong and Kuang [84]	An ant colony optimization (ACO) algorithm for TCTP is pre- Integrated with the modified adaptive weight approach (MAWA), the ACO algorithm can find out the optimal solutions.	Na	To compare the performance, the Pareto optimal solutions obtained from GAs (Zheng et al. 2004). It shows that all Pareto optimal solutions taken from the ACO algorithm are better than from the GAs.	2008
	Simulated Annealing (SA)	Anagnostopolus and Kotsikas [60].	Five types of the SA algorithm are interpreted	Na	Na	2010
	Particle Swarm Optimization (PSO)	Zhang and Xing [13].	A fuzzy-based PSO with quality considerations that uses fuzzy inputs utility to produce composite values.	Generates only a solo optimal solution rather than the Pareto front	The comparison shows that the proposed Fuzzy Multi-Objective Particle Swarm Optimization FMOPSO is capable of finding the TCQT problem with almost the same competence as the GA	2010

Table 2.1 Techniques for TCTP Analysis (continued)

Method	Description	Author(s)	Remarks	Drawbacks	Performance Comparison	Year
Meta Heuristic	Hybrid-Genetic Algorithm (GA)	Sonmez and Bettemir [5]	- HA can find optimal or near-optimal solutions by searching a small part of the solution space. - The hybrid GA has a significant potential for solving large-scale DTCTPs	Na	- Comparing HA with GA showed better performance of the HA. - The results of HA for large scale problems show that HA gives sufficient optimal and near-optimal solutions for the DTCTP	2012
	Hybrid-Particle Swarm Optimization (PSO)	Aminbakhsh [85].	Can find optimum or near-optimum solutions for the medium-sized problems with minimal deviations from the optimal solutions.	- A slight deterioration in the quality of the obtained solutions for larger-scale problems are found when exposed to higher daily indirect costs	The results compared with HA (Sonmez and Bettemir 2012) the comparison shows that the HPSO has better performance and more provide appropriate optimal and near-optimal solutions for the DTCTP	2013
	An Advanced Stochastic Time-Cost Tradeoff (ASTCT)	Guk Lee, Yong Yi, and Eun Lee [86].	- ASTCT searches for near-optimal solutions enormously by adapting simulation-based scheduling with GA.	-First, the fictitious construction method table that includes the best-fit-PDFs of activity durations and costs were used	Na .	2015

Table 2.1 Techniques for TCTP Analysis (continued)

Method	Description	Author(s)	Remarks	Drawbacks	Performance Comparison	Year
Meta Heuristic			-ASTCT decrease the calculation time while providing adequate maturity by using a CPM-guided initialization of the chromosome, and by assuming the minimum number of generations. -ASTCT depends on well-established systems (e.g., Primavera P3 or P6, MS Project, SureTrak, etc.) with which practitioners are already familiar.	In the case studies rather than actual data collected from job sites. -ASTCT presumes that the resources and related costs are divided evenly on the time of each task and not shared by multiple tasks. -The precedence obstacles between tasks will differ with the change of construction techniques.		
	Discrete Particle Swarm Optimization (DPSO)	Aminbakhsh, and Sonmez [6].	DPSO can reach high-quality solutions for small, medium, and large-scale DTCTP and outperforms	Limitations for representing the complexity of some large-scale construction projects	The DPSO performance-tested three types of instances. Small-, Medium-, and Large-scale instances.	2016

Table 2.1 Techniques for TCTP Analysis (continued)

Method	Description	Author(s)	Remarks	Drawbacks	Performance Comparison	Year
Meta Heuristic			the state-of-the-art technique regarding solution quality and computation time especially for medium and large-scale problems		DPSO having high-quality solutions for the small instances up to 18 activities, the DPSO algorithm was highly successful for achieving high-quality solutions for the medium-scale test problems, and the performance of the DPSO for large-scale problems reached very successful results. To compare with the other researchers' studies	
	Genetic Algorithms (GA)	Agdas, Warne, Norgaard, and, Forrest J [7].	The model was improved with minimum software improvement utilizing open source tools, and processing demands can be reached by a personal computer while conserving a manageable computational time	The parameter sweeps analyses done show that the model performance will be extremely rely on problem formulation and GA parameter selection.	Na	2018

CHAPTER 3

THE PROPOSED HYBRID GENETIC ALGORITHM (HGA)

In this chapter, a hybrid Genetic Algorithm-based method will be described for finding multi-mode time-cost trade-off problems (TCTP). The proposed method is coded in C# programming language for which CPM technique is embedded for finding total duration and total cost of the practiced problems. The Hybrid Genetic Algorithm described herein aims at finding near-optimum, if not the optimum, solutions to come to the aid of decision-makers in selecting possibly the best time-cost modes available. This is facilitated by optimizing the duration while minimizing the total cost of the TCT problems.

3.1 CPM Calculations for Networks with Generalized Precedence Constraints

CPM was vastly used in the mid-1980s in the construction industry. Initially, the Arrow Diagramming Method (ADM) was developed by Du Pont in 1956. Later on, in 1961 Fondahl proposed the Precedence Diagramming Method (PDM), which basically replaced ADM since not only it was easier to implement, but also it made defining various logical relationships possible among the activities for proper overlapping purposes [87]. Another key difference between ADM and PDM was the way the network was represented. In PDM activities were represented as nodes rather than arrows, and arrows were simply used for connecting nodes to secure the logical relationships. Regardless of the representation method used, CPM allows calculation of four basic times for each activity. The four basic times include Early Start (ES), Early Finish (EF), Late Start (LS), and Late Finish (LF). ES and EF denote the earliest time an activity can start and finish, respectively. LS and LF, on the other hand, denote the latest time an activity may start and finish, respectively, without disturbing the completion time of the project.

The difference between the ES and LS or the EF and LF is known as the Total Float (TF) which virtually implies the degree of flexibility in timing of each activity. Activities with zero degree of flexibility are known as the critical activities which constitute the critical path(s) of the network. In other terms, if TF for an activity is zero, that means any changes in the timing of that activity will change the length of the critical path, and thus the project duration will change.

In this study, an adept CPM calculator is embedded to the algorithm which is capable of handling networks with all the different relationship types. That is, the networks with general relationships of Finish to Start (FS), Finish to Finish (FF), Start to Finish (SF), and Start to Start (SS) with inclusion of either lead or lag times.

3.1.1 CPM Calculator for the Proposed Algorithm

The calculation of the critical path can be achieved by two consecutive processes of forward pass and backward pass calculations. Forward pass is done to find Early Start (ES) and Early Finish (EF) times of each activity. Backward pass is then carried out to calculate the Late Start (LS) and Late Finish (LF) times of the activities. After these two processes, critical activities are detected with regard to their Total Float (TF) amounts. The following notation will be used hereafter in this study:

ES_j: The earliest time activity j can be started;

EF_j: The earliest time activity j can be finished and is equal to $ES_j + D_j$;

LF_j: The latest time activity j can be finished without delaying completion of the project;

LS_j: The latest time activity j can be started without delaying completion of the project and is equal to $LF_j – D_j$;

TF_j: The amount of time activity j can be delayed without delaying completion of the project and is equal to $LF_j – EF_j = LS_j – ES_j$;

where; j is the activity number.

The CPM calculator module of the proposed algorithm receives the activity information including duration, immediate predecessor, relationship with the immediate preceding activity/activities. Of the information, activity duration is retrieved based on the time-cost mode selected for each activity. The coding process can be summarized as follows:

Step 1: Define activities;

Step 2: Retrieve duration of each activity based on the mode selected;

Step 3: Define relationship with immediate preceding activities;

Step 4: Calculate successors;

Step 5: Perform forward pass and find ES and EF of each activity;

Step 6: Perform backward pass and find LF and LS of each activity;

Step 7: Find critical activities that has equal ES/EFs and LS/LFs;

Step 8: Detect all the critical paths based on the activities found in Step 7.

3.2 Simulated Annealing-based Hybrid Genetic Algorithm (HGA)

In this study, a hybrid Genetic Algorithm for solving multi-mode TCT problems is proposed which incorporates the principles of the Simulated Annealing (SA) method. Most of the activities can be executed using more than one construction method or can be executed more rapidly by allocating extra resources, so, for each activity more than one duration and cost combination might be available. These duration and cost combinations are named as modes; thus, each activity may have more than one mode for execution. Large number of modes coupled with large number of activities make the mathematical computations complicated for finding a minimum total cost within a minimum total duration.

In the Discrete-Time-Cost Trade-off (DTCT), each activity has a number of time-cost pairs. Time-cost pair for *i*th mode of an activity can be denoted by (C_i, D_i); where, C_i is the cost the activity when mode *i* is selected, and D_i is respected duration of that mode. DTCTP aims at assigning a mode for each activity by considering total cost and project duration criteria. This study, by using a Simulated Annealing-based GA attempts to address the goal of finding practically the best combination of time-cost modes for DTCT problems.

3.2.1 Genetic Algorithm (GA)

As discussed in Chapter 2, Genetic Algorithms are generic optimization approaches which can be applied for solution of a wide range of optimization problems. Optimization of TCTP is among the many application areas of GAs which is why it has also been used as the basis of the method developed in this study.

3.2.2 Genetic Algorithm (GA) for Construction TCTP

In general, to find the shortest completion time and the least cost, we need to properly select the best time-cost modes available. Selection of appropriate combination of time-cost modes entails making decisions on the resources and tools for execution of each task of the project. In multi-objective TCTPs, generally there are more than one execution mode for each activity making the mode selection process complicated. Thus, the optimal project duration and total cost cannot be found short of an adequate optimization method. In light of this fact, genetic algorithm is used as the foundation for the method introduced in this thesis. The series of decisions are dubbed as the chromosomes (strings) in the genetic algorithm which record the selections made on the execution modes of the activities in a project. GA evaluates implementing several different decisions to find potentially the best scenario for obtaining optimum/near-optimum solution to a TCT problem. GA applies techniques inspired by the natural selection to improve the selections made by updating the chromosome structure [88].

A simple version of GA can be abstracted as follows. As shown in Figure 3.2, every string k denotes a possible solution to the problem with m activities by holding values for m tuples in the form of $k = (x_1, x_2,\ldots, x_m)$; where, each gene x_i denotes a decision on the execution mode of an activity out of a possible range of xi = (x_1, $x_2\ldots$, x_m).

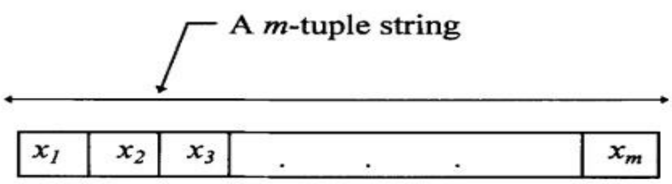

Figure 3.1 Genetic Algorithm's Chromosome Structure

Genetic algorithms include several components. These components include: P_c which denotes swap (crossover) probability; this parameter regulates the probability of crossover between the modes of two distinct strings. P_m which stands for mutation probability; this parameter regulates the probability of a random change in one or more genes of a selected string. F which denotes fitness function; this is a function utilized to measure the survival ability of a string.

Steps involved in a simple GA can be described as follows:

Step 0: Generate a parent population randomly; randomly select strings and evaluate their fitness function;

Step 1: Generational Reproduction; generate the offspring generation; new generation is created by selecting a chromosome from the old generation with respect to fitness function and with replacement;

Step 2: Crossover or Swapping; this step is counted as the leading operator in a genetic algorithm [6]. New offspring are generated by selecting new genes from parent chromosomes. Crossover operation is performed by choosing a pair of chromosomes (strings) randomly and executing the crossover at P_c rate. As illustrated in Figure 3.3, crossover operation is choosing a crossover site randomly and swapping the tails of the selected two strings [3];

Step 3: Mutation; this operation allows for introducing new genetic material into the population. This operation may introduce the previously unknown genes into the population where better candidate solutions might be achieved [89]. As demonstrated in Figure 3.4, mutation will randomly change the value of an element in the string at P_m rate;

Step 4: Evaluation; in this step, the fitness function values of the strings will be evaluated;

Step 5: Termination; repeating steps 1-4 till either no improvements are made in the last couple of iterations or the iterations are complete.

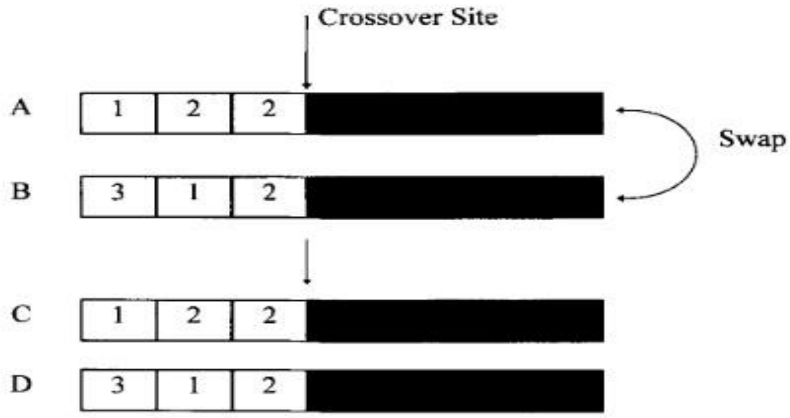

Figure 3.2 An Illustration of Crossover Operator of Genetic Algorithm [3]

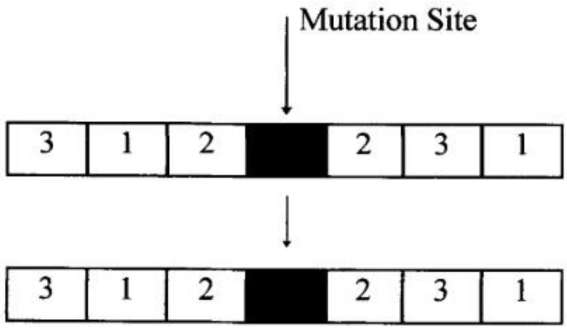

Figure 3.3 An Illustration of Mutation Operator of Genetic Algorithm [3]

3.2.3 Genetic Algorithm for Discrete Time-Cost Trade-Off

In real-life projects the time-cost relations are seldom continuous linear. Rather, execution modes are available in a discrete space. In addition, by utilizing discretization, any time-cost function can be estimated [90]. Due to these very reasons, in this study the discrete version of TCTP is studied. As depicted before, GA is shown to successfully perform in many engineering fields as it helps decision-makers to search for optimal solutions for complex and large-scale problems. To adopt GA for DTCTP, a roulette wheel selection method is also implemented to give greater chance to the survival of the fitter individuals, i.e., those with better fitness functions. Moreover, the fitness function calculations are carried out with respect to the objective function that follows, which results in the minimization of the total project cost.

$$C_j = \sum_{i=1}^{n} C_{ij} + T_j C_d \tag{3.1}$$

$$O_j = \text{Minimize} \sum_{j=1}^{s} C_j \tag{3.2}$$

$$F_j = M - O_j \tag{3.3}$$

$$P_j = \frac{F_j}{\sum_{j=1}^{n} F_j} \tag{3.4}$$

where; C_j is the total cost of chromosome j; i (1 to n) is a project activity, having a direct cost C_{ij} in the j's chromosome; T_j is the total project duration of j's chromosome, which is found by the CPM calculator module of the algorithm; C_d is the amount of daily indirect cost; O_j is the objective function for j's chromosome to minimize the total project cost – the higher the fitness value, the more fit is the chromosome (less total cost); F_j is the fitness function for j's chromosome, and M is a constant which is taken as the maximum fitness value number among all strings; P_j is the probability of the selection of each chromosome using the roulette wheel selection technique.

Expected count equation given in Eq. (3.5) is then used which secures discarding the chromosomes with lower probability values and duplicating those with higher probabilities.

$$EC_j = \frac{F_j}{F_{avg.}} \tag{3.5}$$

where; EC_j is the expected count function, and $F_{avg.}$ is the average of all fitness function values which is found by dividing total fitness values by the number of chromosomes. Figure 3.5 briefly shows how the implemented roulette wheel approach gives higher probabilities of survival for the chromosomes with greater fitness function values.

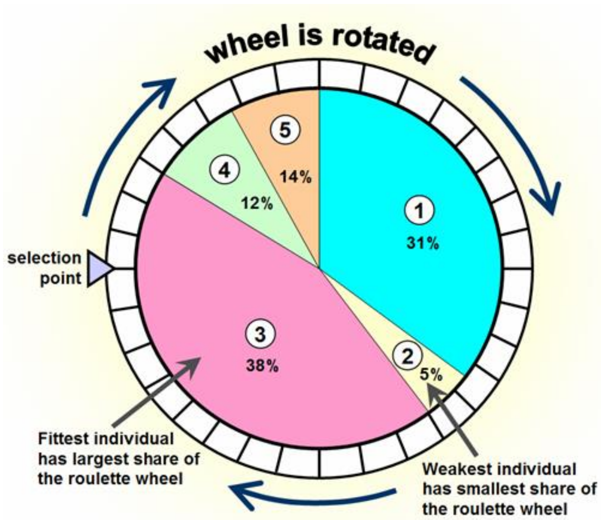

Figure 3.4 Illustration of the Roulette Wheel Selection Method [91].

The next step will be starting for reproduction after selecting chromosomes with high fitness values and using expected count for discarding low-fit chromosomes. The fundamental operation for genetic reproduction is crossover operation. Some genes will be selected for reproducing new genes by random selection [75]. Generally, crossover operation is not applied to all pairs of selected individuals. The selection of pairs is made randomly by assigning a random number between [0,1] to each chromosome and comparing the given random number with the probability value. If the random number assigned is smaller than P_c then the chromosome is selected for crossover operation. The probability applied typically ranges between 0.60 and 1.0. However, if the crossover operator is not applied, the parents duplicate for producing offspring [92]. In this study, a crossover probability variable is defined which only takes values between 0.6 and 1.0 from the users. This enables examining different crossover probabilities for comparison purposes.

In this step, another measure is taken to guarantee only the worst chromosomes in the selected generation will be replaced by fitter reproduced chromosomes in the crossover operation. If the chromosomes in the crossover process have higher fitness function values after reproduction, they will be replaced. On the other hand, if the chromosome fitness function value is lower than the parent's fitness value, the offspring generated after crossover will be discarded as according to condition (3.6) as follows:

$$Crossover\ Opeartion \begin{cases} Replace\ if & F_{j0} \leq F_{jc} \\ Discard\ Crossover\ if & F_{j0} > F_{jc} \end{cases} \tag{3.6}$$

where; F_{j0} is the fitness function value before crossover operation, and F_{jc} is fitness function value after crossover operation. After reproduction in crossover operation, roulette wheel selection is applied as a result of which the population size might change due to low-fit chromosomes being discarded. In this study, population size will be calibrated after the roulette wheel selection is applied by finding the difference between the parent and offspring population sizes using Eq. (3.7). With regard to condition (3.8), chromosomes reproduced in the crossover operation will be used to make up for any deficiencies in the population size.

$$\Delta_{Pop\ Size} = \text{Parent}_{Pop\ Size} - \text{Offspring}_{Pop\ Size} \tag{3.7}$$

$$Pop\ \ Size\ Calibration \begin{cases} if \quad \Delta_{Pop\ Size} = 0 & Dismiss\ Calibration \\ Otherwise, & Start\ Calibration \end{cases} \tag{3.8}$$

For calibration, first, all the chromosomes – reproduced prior to and after crossover operation – will be sorted in descending order respecting their fitness values and then the first $\text{Parent}_{Pop\ Size}$ number of chromosomes will be selected. After the crossover process, the mutation operation is applied to promote the diversity of the population. The mutation rate is considered adjust the probability of a change in the values of genes in a chromosome [93]. The mutation operation process is done by assigning a random number for each gene in chromosomes and comparing these random numbers with the given probability value. If the random number is smaller than probability value so the gene will be nominated for the mutation; otherwise, the mutation will be rejected. The choice of P_m value plays a chief role in the behavior of GA. A large value of P_m converts the GA into a purely random search algorithm. At the same time some mutation is desired by selection of non-zero values for P_m to prevent the early convergence of the GA to suboptimal solutions. Usually, for P_m a value is chosen within the range of 0.005 to 0.05 [94]. P_m value can be set by using Eq. (3.9) as follows:

$$P_m = \frac{1}{No.\ of\ Genes\ in\ the\ Chromosome} \tag{3.9}$$

where; P_m is the mutation probability and ‘No. of Genes in the Chromosome’ in this study is set to be equal to the number of activities.

3.2.4 Genetic Algorithm and Simulated Annealing (GASA)

If a sole-GA method is to be implemented for solving an optimization problem, more iterations would be necessary to achieve adequate results. Figure 3.6 illustrates how a classical GA can get stuck at a local optimum point in a complex solution space. In this respect, some complementary computation methods can be applied to increase the convergence capabilities of GA for escaping local optima which ultimately will also lead to shortened computation time. Simulated Annealing (SA) is one of the chief methods used toward this aim. SA is a probabilistic meta-heuristic algorithm used for the optimization problems which was inspired by the cooling schedule of alloys subjected to tempering. In higher temperatures, molecules can move freely in any direction. While, as the alloy cools down, the movement of the molecules gets restricted depending on the temperature [95], [96]. Establishing such an interaction between the SA method and the GA reproduction mechanism via developing a hybrid algorithm seems promising. By merging both the algorithms the computation time can be shortened as SA will reject those mutations leading to a worse solution as the algorithm reaches larger iteration number [69], [97]. Controlling the mutation operator of the genetic algorithm by using the simulated annealing method is known as the Genetic Algorithm Simulated Annealing (GASA) [75].

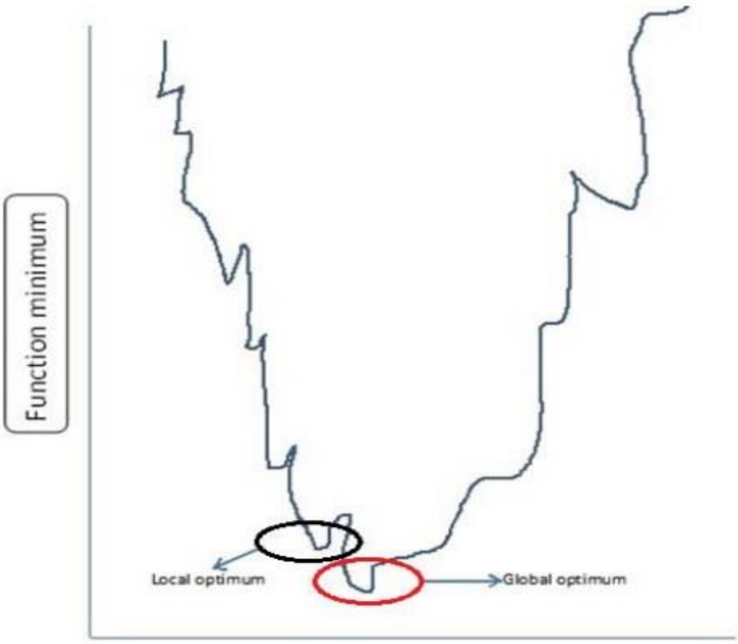

Figure 3.5 Local and Global Optima in a Complex Solution Space [97].

Both genetic and simulated annealing algorithms have strengths and weaknesses. GA's strength lies within its implicit parallelism and keeping only useful information from previous generations. It facilitates generations of high-quality solutions by capturing and modifying components of existing good solutions through crossover operators. On the other hand, GAs might lose direction as their operators are susceptible to discarding useful solutions. Immature early convergence of GA is stemming from this weakness. By taking advantage of the cooling schedule and local selection concept of SA method a naturally parallel evaluation structure can be obtained. Such a hybrid approach maintains the generality of GA and can easily be used to solve various optimization problems [98]. In the HGA developed for DTCTP, every mutation that leads to a better offspring chromosome will be accepted while a bad mutation may or may not be rejected. Condition (3.10) given below, is used to regulate acceptance or rejection of the bad mutations.

$$\begin{cases} if\ R_n \leq\ e^{\frac{(\Delta f)t}{\beta t f_0}} & Accept \\ if\ R_n >\ e^{\frac{(\Delta f)t}{\beta t f_0}} & Reject \end{cases} \tag{3.10}$$

where; R$_n$ represents a random number between 0 and 1; f_0 is the initial fitness function value before mutation operation; f_m is the fitness function value after mutation process; $\Delta f = (f_m - f_0)$; β the Boltzmann Constant used to set the rate of cooling; and t is the current iteration number. Effect of Eq. (3.10) can be illustrated as shown in Figure 3.7 which is adapted from [75]. In this figure the acceptance probability of a bad mutation is plotted against the iteration number for Boltzmann Constant of 1.0.

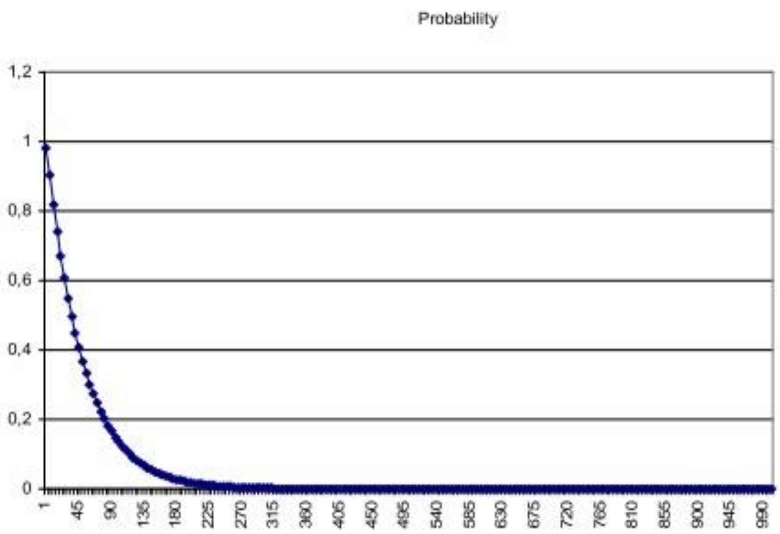

Figure 3.6 Acceptance Probability of Harmful Mutation versus the Iteration Number [75]

As shown in Figure 3.7, the probability of accepting a bad mutation reaches zero after 200th iteration. This practically prevents harmful mutations since only the better neighbors are sought for mutation of the chromosomes.

3.2.5 Termination Conditions

It is a common practice to terminate the algorithm when the maximum number of iterations is reached [99]. Particularly the following three types of termination conditions are often employed for classical GAs [100]:

i. Reaching the upper bound of the number of generations;
ii. Reaching the upper bound of the number of fitness function evaluations;
iii. Having no improvement made for the last rounds of iterations.

In this study, two of the methods above (i.e., i and iii) were testes as the stopping criteria; though, only one of them was kept for carrying out the analysis. The criterion employed for analysis (i) stops the algorithm whenever the last rounds of iterations – entered by the user – is complete. Criterion (iii) was tested by not included in the final version the developed algorithm since it intervened the hill-climbing strategy of Simulated Annealing (SA). That is, before SA kicked in, the algorithm was already stopped due to making no improvements in the last certain number of iterations. The flowchart of the proposed GA-based model is illustrated in Figure 3.8.

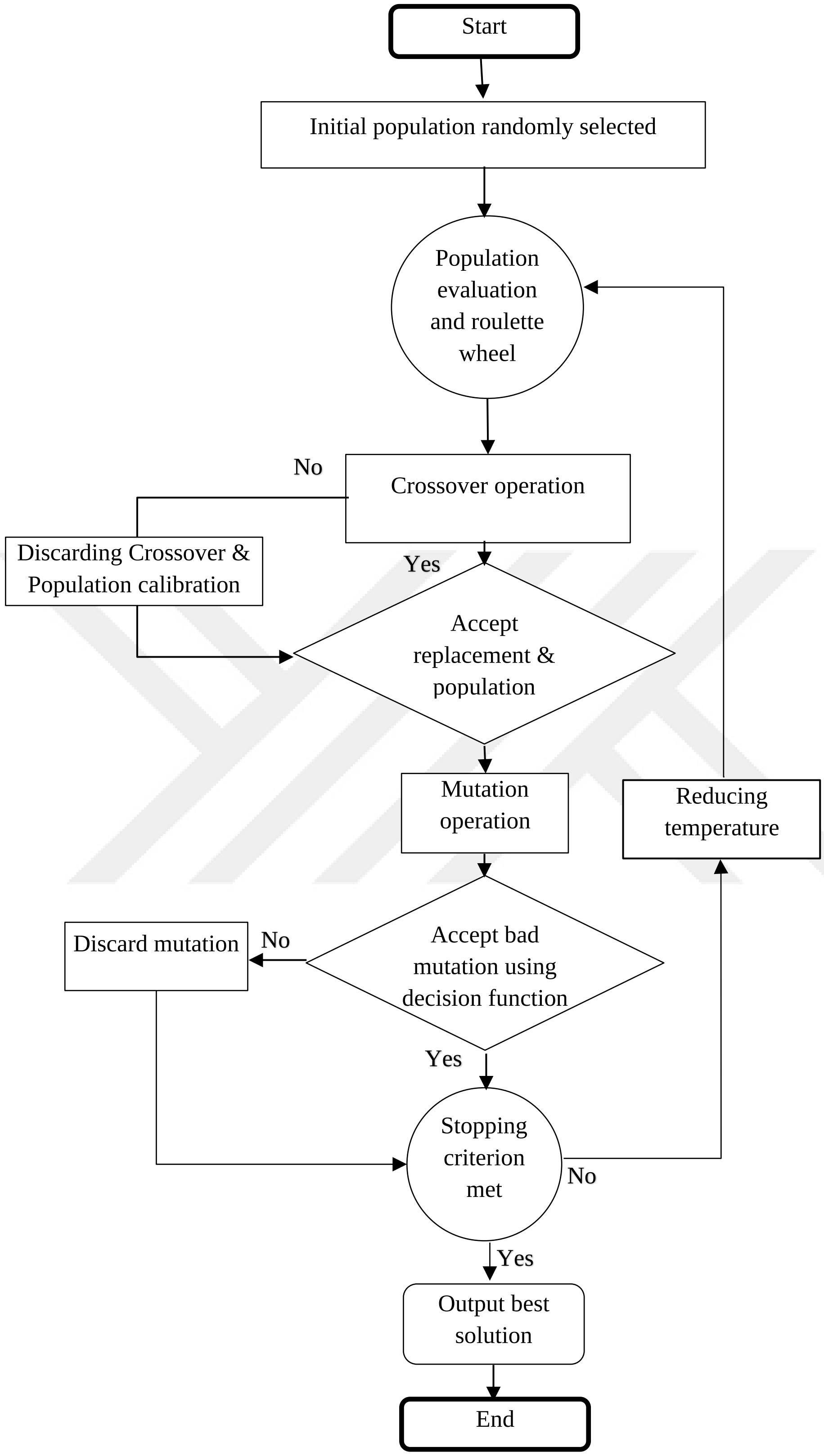

Figure 3.7 Flow chart of the Proposed Hybrid Genetic Algorithm (HGA)

3.3 Simulated Annealing-based Hybrid Genetic Algorithm (HGA) for Time-Constraint Discrete Time-Cost Trade-Off Problem

In time-constraint time-cost trade-off problems, there is an upper bound set for the completion time of the projects. This upper bound is introduced to reflect the completion deadlines stipulated in the contracts. When parties agree to such terms in the contract, the contractor affirm they will pay a certain amount of delay penalty on daily basis in case project delays, and the owner guarantees to pay certain amount of bonus payment on daily basis if the project finishes earlier than the set date. The hybrid GA proposed in section 3.2 is slightly modified to also accommodate such contractual conditions as the deadline, the daily liquidated damage, and the daily incentives payments. In this regard, the three new conditions are introduced as parameters to the formulation of the fitness function as given in Eq. (3.11) [45].

$$\begin{cases} C_j = \sum_{i=1}^{n} C_{ij} + T_j C_d + (T - T_d) C_c & if\ T_d \geq T \\ C_j = \sum_{i=1}^{n} C_{ij} + T_j C_d + (T - T_d) C_n & if\ T_d < T \end{cases} \quad (3.11)$$

where; C_j is the total cost of chromosome j; i (1 to n) is a project activity, having a direct cost C_{ij} in the j's chromosome; T_j is the total project duration of j's chromosome, which is found by the CPM calculator module of the algorithm; C_d is the amount of daily indirect cost; Td denotes the desired completion time; daily incentive payment and daily liquated damages are expressed using C_c and C_n, respectively. Therefore, the higher the fitness function value, the more fit is the chromosome (less total cost) which will help the algorithm find solutions with durations equal or less than the deadline. The flowchart of this the proposed HGA for time-constraint TCTP is demonstrated in Figure 3.9 which remains virtually identical to Figure 3.8 with the exception of the conditions added to reflect Eq. (3.11).

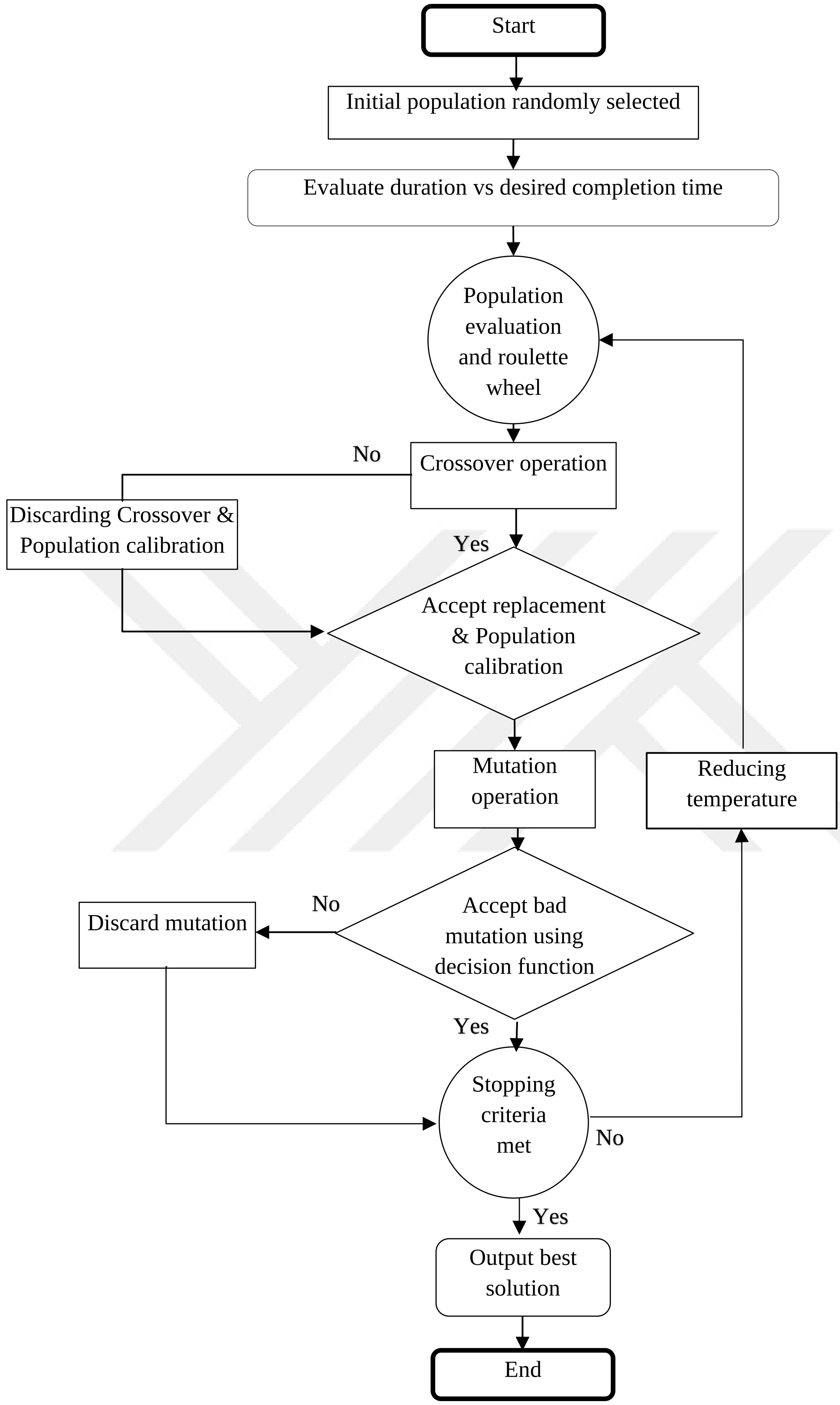

Figure 3.8 Flow chart of the Proposed HGA for time-constraint TCTP

CHAPTER 4

PERFORMANCE MEASUREMENT AND VALIDATION OF THE PROPOSED ALGORITHM

In this chapter performance measurement and validation of the developed algorithm is carried out for both discrete TCTP and time-constraint DTCTP using instances widely used in the literature. Then a real-life construction project is used to demonstrate applicability and effectiveness of the developed HGA model as the real project includes over four hundred activities with generalized precedence relationships.

4.1 Performance Measurement of the Proposed HGA

The authors of this study believe that the CPM calculator module of a practical optimization algorithm should be able to tackle networks with general relationships among its activities. For performance analysis of the proposed method, initially the problems derived from the literature are used. Through these analyses, parameters of the proposed model are adjusted and fine-tuned for each problem following several test runs which led to improved convergence speed and higher quality solutions. In this study, the parameter settings used formerly in the literature [3], [8], [45], [69] have all been tested. As discussed earlier, the number of generations is used as the only termination criterion [101]. Eventually, the parameter values listed below have been adopted for the proposed algorithm:

- Population size of 30 to 100,000 was used;
    - Population size between (30-300) is recommended to be used for small-scale problems with regard to the complexity of the network and the number of activities;

  - Population size between (3,000-100,000) is recommended to be used for medium-scale problems with regard to the complexity of the network and the number of activities;
  - Population size between (3,000-25,000) is recommended to be used for large-scale problems with regard to the complexity of the network and the number of activities. It is to be noted that for getting a better solution for large-scale problems with complex networks and more than three modes, a higher population should be used; however, more CPU time is required;
- Crossover probability (P_c) was set a value within the range 0.6 to 1.0;
- Mutation probability (P_m) was set using Eq. (3.9) within the range 0.005 to 0.05;
- Boltzmann constant was set a value within the range 0.01 to 1.5;
  - In order to get better solutions with acceptable CPU time, it is important for users to limit probability of bad mutation acceptance by selecting a lower than one Boltzmann's constant value especially for large-scale problems with a complex (more than three modes) network. In other terms, using a high population number will obviously increase the chances of locating higher-quality solutions at the cost of increased CPU time. Therefore, in order to obtain satisfactory results within acceptable CPU times, restricting acceptance of bad mutations is recommended;
- Iteration number (Generation number) was set a value within range 30 to 1,300;
  - Iteration number between (30-50) is recommended to be used for small-scale problems;
  - Iteration number between (250-500) is recommended to be used for medium-scale problems;
- Iteration number between (500-1500) is recommended to be used for large-scale problems.

The parameter selection significantly affects the solution quality and performance of HGA. In this respect, it is recommended to experiment different parameter values within the ranges proposed above.

It is well-established that the optimality of results of any heuristic or meta-heuristic cannot be evaluated unless they are compared to those of an exact procedure. Moreover, as explained earlier, it is widely acknowledged that the heuristic and meta-heuristic methods do not guarantee optimally of the results obtained. Thus, optimal results obtained from the literature will be used for comparison purposes. Eleven benchmark problems, twelve new instances, and an actual case project are fed into the proposed HGA model. Due to the randomness of the proposed HGA model, it is ran for ten consecutive times for each problem and the Average Percent Deviation (APD) for each instance is reported.

4.2 Computer Implementation

Microsoft Visual Studio 2019 was used for implementation of the proposed HGA algorithm using C# programming language. The proposed algorithm is occupied with the generalized CPM calculator to deal with the networks with FS, SS, SF, and FF relationships and lead/lags in between. A laptop computer running on Windows 10 Pro (64-bit) operating system with Intel(R) Core (TM) i7-8550U CPU @ 1.80 GHz, and 8 Gigabytes of RAM was used for all the experiments. Processing times for executions of the main blocks of code are given in Tsp.Hours, Tsp.Minutes, Tsp.Seconds, and Tsp.Milliseconds in the following sections.

4.3 Performance of HGA for solving DTCTP and time-constraint DTCTP

The 18-activity network which was generated by Feng et al. [3] and slightly modified by Hegazy [57] is a small-scale problem widely used by numerous researchers [5], [6], [8], [12], [58], [72], [79], [81], [84], [102]. This small-scale problem which provides ground for evaluating the performance of TCTPs is used in this study. In addition, a larger problem with 29 activities is also used which is derived from Chassiakos and Sakellaropoulos [68]. The 29-activity problem incorporates generalized logical relationships among the activities which was later used as the basis for generation of larger problems by Sonmez and Bettemir [5]. Sonmez and Bettemir [5] constructed 290-activity network by copying 29-network nine times and connecting the ten projects using finish to start relationships. Another test problem introduced by Sonmez and Bettemir [5] which includes 63 activities is also used. Based this network Sonmez and Bettemir [5] also developed a large-scale problem with 630 activities. This large-scale problem was generated by copying the 63-activity network nine times in serial and connecting them using finish to start relationships. The largest and the most complicated problems solved by HGA includes 12 instances with 990 activities which were generated by Aminbakhsh [102] and Sonmez et al. [58]. 990-activity problems combine four different network complexity indices with three different time-cost alternative sizes (three, six, and nine modes). The 23 instances will be implemented under different assumptions of daily indirect cost, etc. More information on the practiced instances will be provided in the following sections.

4.3.1 18-activity Instances

Feng et al.'s [3] network which contains 18 activities with different mode numbers is given in Table 4.1. The table contains the activities' immediate predecessors and time-cost alternatives. This problem includes ten activities with three time-cost alternatives, two with four alternatives, and five activities with five alternatives summing up to about 5.9×10^9 possible schedules. The precedence diagram of this instance for the all-normal (longest and the cheapest) schedule is illustrated in Figure 4.1. This problem is studied under three conditions known as18A, 18B, and 18C. The 18A and 18B problem – with no imposed deadline – are used for implementation of HGA discussed in section 3.2. 18A problem [85] assumes zero daily indirect cost while an indirect cost of $1,500/day is assumed for 18B problem [5]. Also, for implementing HGA detailed in section 3.3, 18C problem [45]– time constraint DTCTP – is used. For the 18C problem, an indirect cost of $200/day, liquidated damages of $20,000/day, daily incentive payment of $1,000/day, and a desired completion duration of 110 days are assumed.

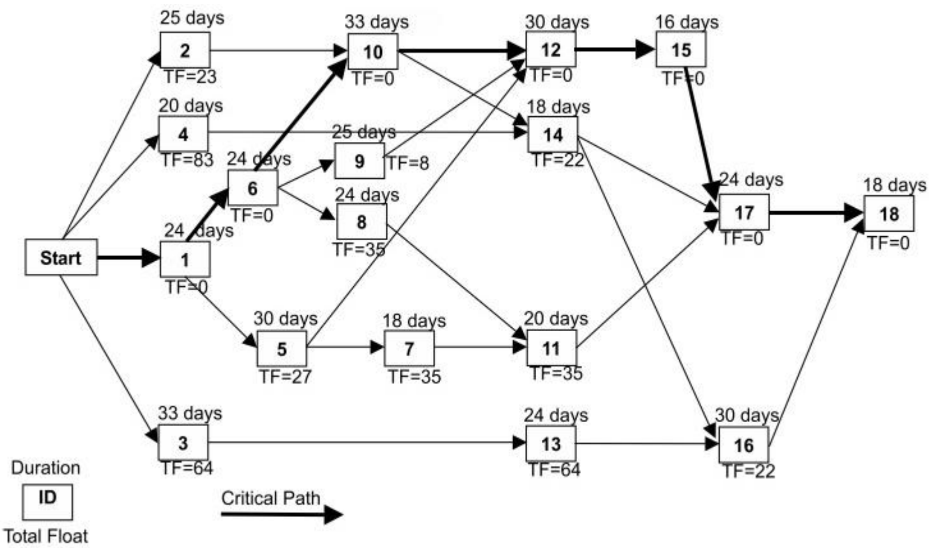

Figure 4.1 Network Diagram for 18-activity Problem [5]

Table 4.1 Activity Table for Problems 18A, 18B, and 18C [5]

Activity	**Pred.**	**Mode 1**		**Mode 2**		**Mode 3**		**Mode 4**		**Mode 5**	
		Dur. (day)	**Cost $**	**Dur. (day)**	**Cost $**	**Dur. (day)**	**Cost $**	**Dur. (day)**	**Cost $**	**Dur. (day)**	**Cost $**
1	–	14	2,400	15	2,150	16	1,900	21	1,500	24	1,200
2	–	15	3,000	18	2,400	20	1,800	23	1,500	25	1,000
3	–	15	4,500	22	4,000	33	3,200	–	–	–	–
4	–	12	45,000	16	35,000	20	30,000	–	–	–	–
5	1	22	20,000	24	17,500	28	15,000	30	10,000	–	–
6	1	14	40,000	18	32,000	24	18,000	–	–	–	–
7	5	9	30,000	15	24,000	18	22,000	–	–	–	–
8	6	14	220	15	215	16	200	21	208	24	120
9	6	15	300	18	240	20	180	23	150	25	100
10	2, 6	15	450	22	400	33	320	–	–	–	–
11	7, 8	12	450	16	350	20	300	–	–	–	–
12	5, 9, 10	22	2,000	24	1,750	28	1,500	30	1,000	–	–
13	3	14	4,000	18	3,200	24	1,800	–	–	–	–
14	4, 10	9	3,000	15	2,400	18	2,200	–	–	–	–
15	12	12	4,500	16	3,500	–	–	–	–	–	–
16	13, 14	20	3,000	22	2,000	24	1,750	28	1,500	30	1,000
17	11, 14, 15	14	4,000	18	3,200	24	1,800	–	–	–	–
18	16, 17	9	3,000	15	2,400	18	2,200	–	–	–	–

As there is no direct way to set the parameters of a GA [32, 37, 44, 68, 70, 71], the parameters for each test instance are set after numerous trials with different values. The parameters are configured with regard to the solution quality and convergence speed of HGA. The significant parameters impacting the solution process of HGA are population size, generation number, crossover probability rate (P_c), and mutation probability rate (P_m). Obviously, for obtaining higher-quality solutions, larger population sizes and large number of generations are needed which increase processing time significantly. Thus, for the small-scale problems, smaller population sizes, and the contrary, for larger problems larger populations sizes are used in this study. The generation number which is also considered as the termination condition, can be decided and entered by the users. Figure 4.2 illustrates the window using which the users will enter their desired parameter values.

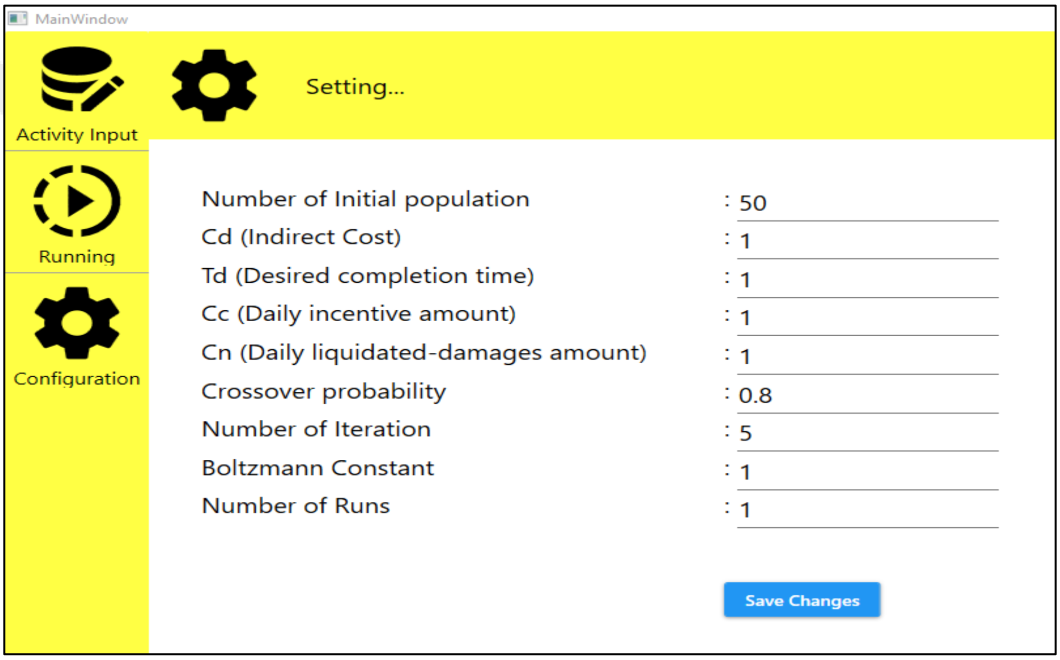

Figure 4.2 Parameter Setting Window of the Proposed HGA

Table 4.2 summarizes the parameter setup of HGA for the small-scale benchmark test instances.

Table 4.2 Parameter Values for the 18-activity Instances

Description	Problem 18A	Problem 18B	Problem 18C
Population Size	50	300	300
Boltzmann Constant	1	1	1
Generation Number	50	35	35
Pc (0.6-1)	0.80	0.80	0.80
P_m (0.005-0.05)	1⁄no.Gene (18) ≈ 0.05	1⁄no.Gene (18) ≈ 0.05	1⁄no.Gene (18) ≈ 0.05

As described in Table 4.2, for the 18A problem 50 offspring is used as the population size. In contrast, for the other two problems, 18B and 18C, 300 offspring is experienced to guarantee the optimality of solutions for all the trials. Since the number of generations remarkably affects the CPU time, values between 35-50 is preferred with Boltzmann's Constant assumed to be 1. Also, the crossover probability rate is set as 0.8 for all 18-activity problems. The mutation probability rate is often selected randomly in the previous GA-based methods; whereas, HGA uses Eq. (3.9) to set the mutation probability within the range 0.005 to 0.05.

Table 4.3 summarizes the results obtained by HGA for 18-activity problems using the parameter settings discussed above. In the second column of this table the number of successive runs is given. Under 'Best Solution' column, the best solutions obtained are presented. The last two columns tabulate the average processing times and the average percent deviations from the optima, respectively.

Table 4.3 Results Achieved for 18-activity Problems by HGA

Instance	No. of Runs	Conditions				Best Solution		Avg. CPU Time (S)	APD (%)
		C_d ($)	C_c ($)	C_n ($)	T_d (Day)	Total Cost ($)	Total Duration (Day)		
18A	10	0	0	0	0	99,740	169	0.05	0
18B	10	1,500	0	0	0	271,270	110	0.27	0
18C	10	200	1,000	20,000	110	128,270	110	0.27	0

Performance of HGA over 18-activity problems is compared to some of the selected existing methods. Table 4.4 compiles the results of comparisons made.

Table 4.4 Performance Comparisons over 18-activity Problems

Instance	Best Solution									
	GA (Hegazy, 1999)		GASA (Bettemir, 2009)		HA (Sonmez and Bettemir, 2012)		DPSO (Aminbakhsh and Sonmez, 2016)		HGA (this study)	
	APD (%)	CPU Time (S)	APD (%)	CPU Time (S)	APD (%)	CPU Time (S)	APD (%)	CPU Time (S)	APD (%)	CPU Time (S)
18A	-	-	-	-	-	-	0	0.4	0	0.05
18B	-	-	-	-	0	-	0	0.4	0	0.27
18C	7.72	6.5	0	8	0	-	0	0.4	0	0.27

HGA's results are compared to both the optimal solutions and the solutions provided by many researchers (Hegazy [45] , Bettemir [75], Sonmez and Bettemir [5], and Aminbakhsh and Sonmez [6]). The APD values for 10 successive runs are also given for the proposed algorithm and other researchers' methods in Table 4.4. HGA is able to obtain the optimum solutions for all of the ten experimental trials for problems 18A, 18B, and 18C. Besides, the proposed algorithm proves to be more efficient since it converges to global optima within 0.05 seconds for problem 18A and within 0.27 seconds for problems 18B and 18C. Figure 4.3, Figure 4.4, and 4.5 illustrate the minimum total cost achieved by HGA per each iteration for problems 18A, 18B, and 18C, respectively. It can be clearly observed that the best solutions were obtained even before reaching the maximum iteration number. Though, using smaller iteration number was experienced to rule out locating the global optimum solution per all the 10 serial runs.

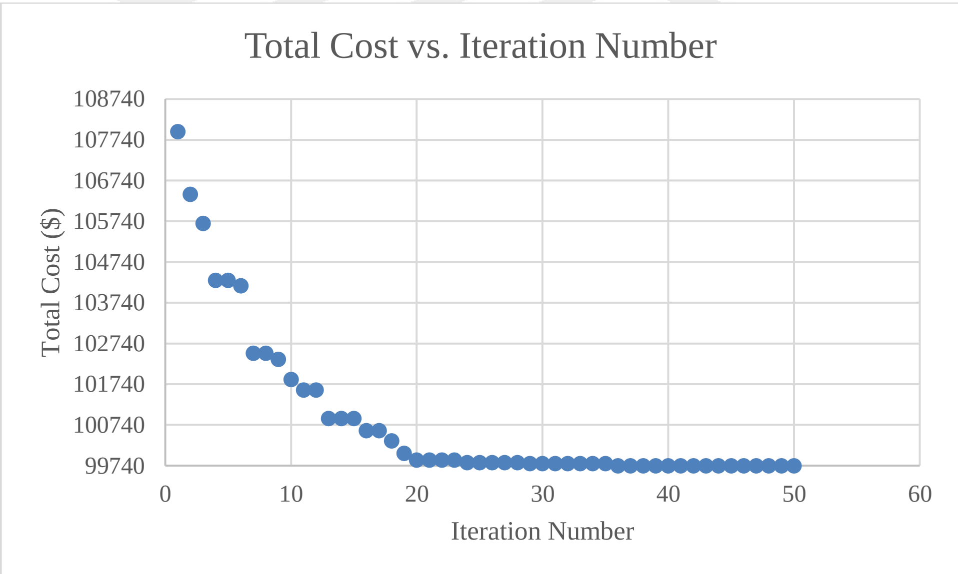

Figure 4.3 Total Cost versus Iteration Number for 18A Problem

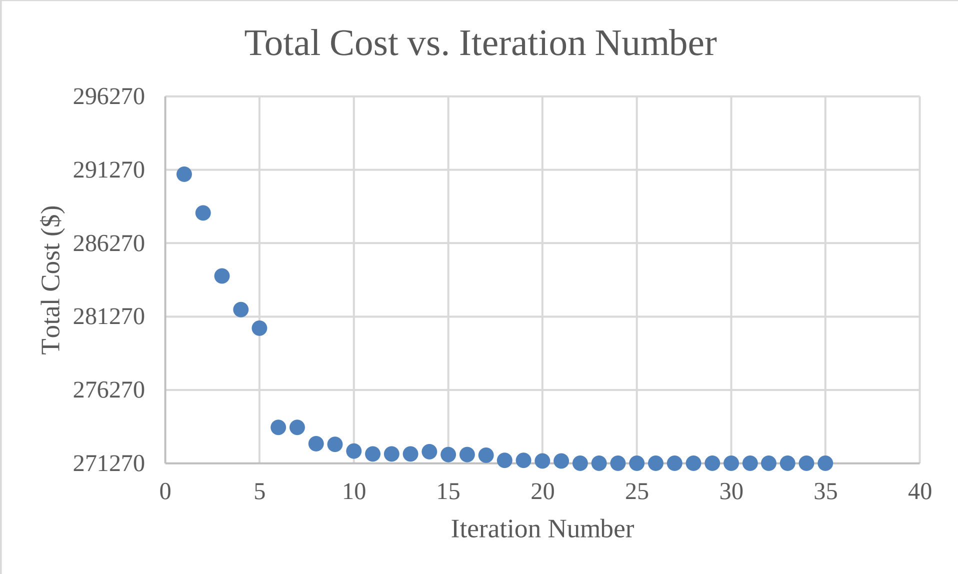

Figure 4.4 Total Cost versus Iteration Number for 18B Problem

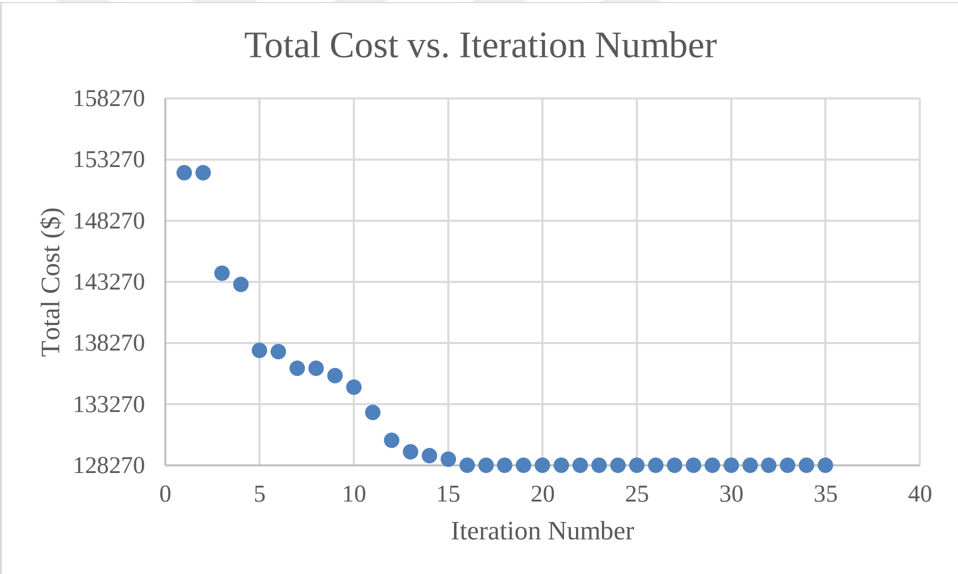

Figure 4.5 Total Cost versus Iteration Number for 18C Problem

4.3.2 29-activity and 290-activitiy Instances

The proposed HGA is implemented for a more complex problem which contains all sorts of logical relationships as Finish to Start (FS), Start to Start (SS), and Finish to Finish (FF) except for Start to Finish (SF). Details of this problem, covering the general precedence relations of activities and the time-cost modes are given in Table 4.5.

Different combinations of the time-cost alternatives in this problem allows for creation of 8.3×10^9 particular schedules. This problem is studied under two conditions known as 29A and 29B. For the 29A problem the indirect cost is assumed as €1,200/day. Exact solution of this problem is derived from Chassiakos and Sakellaropoulos [68] as €1,226,200. While for the ten times duplicated version of this problem, i.e., 290A, the result is obtained from Sonmez and Bettemir [5] as €12,262,000. Problems 29A and 290A are used for implementation of HGA discussed in section 3.2. Furthermore, the 29B and 290B problems are used for testing HGA detailed in section 3.3. For the 29B problem the indirect cost is assumed to be €1,200/day, liquidated damages as €1,500/day, incentive payment as €500/day, and a desired completion duration of 240 days. The optimal solution of the 29B problem is obtained from Chassiakos and Sakellaropoulos [68] as €1,220,700. Besides, for the ten times duplicated version of this problem, i.e., 290B with a desired completion duration of 2,400 days, the result is obtained from Sonmez and Bettemir [5] as €12,262,000.

Table 4.5 Activity Table for 29-activity Problem [5]

Activity	Pred.	Mode 1		Mode 2		Mode3	
		Dur. (Day)	Cost (€1,000)	Dur. (Day)	Cost (€1,000)	Dur. (Day)	Cost (€1,000)
1	-	15	60	12	68	-	-
2	1SS+5, 1FF+0	25	30	20	38	15	44
3	1SS+10, 1FF+3, 2SS+10, 2FF+3	25	50	20	54	15	60
4	3FS+0	12	17	9	21	-	-
5	4FS+0	6	3	-	-	-	-
6	1FS+0	12	27	9	32	-	-
7	6SS+0, 6FF+0	6	8	-	-	-	-
8	7FS+0	20	44	15	48	12	54
9	4FS+0, 8FS+0	12	15	9	22	-	-
10	5FS+0, 9FS+0	6	3	-	-	-	-
11	5FS+0, 10FS+0	1	0.5	-	-	-	-
12	11FS+0	25	95	20	105	15	109
13	12SS+6	15	34	12	41	9	51
14	12FF+8, 13FF+8	12	9	9	13	-	-
15	12SS+6, 12FF+0, 14FS-6, 14FF+10	25	30	15	38	12	42
16	15SS+10, 15FF+0	40	78	35	85	30	90
17	15FS-10, 15FF+0	25	23	20	26	12	35
18	15FS+0	20	14	15	18	12	24
19	15FS+12	25	14	20	19	15	24
20	17SS+10, 17FF+5	20	38	15	42	-	-
21	16FS-10, 16FF+0, 20SS+12, 20FF+2	40	42	35	50	30	58
22	21SS+15, 21FF+2	40	36	30	48	25	56
23	22FS-15, 22FF+0	40	65	35	74	25	79
24	23SS+15, 23FF+5	9	7	-	-	-	-
25	23FS-10, 23FF+0	25	45	20	51	15	59
26	25FS-10, 25FF+0	25	50	20	58	15	64
27	26FS-10, 26FF+0	30	60	25	72	20	78
28	27FS-10, 27FF+0	12	9	9	13	7	18
29	18FS+0, 19FS+0, 24FS+0, 28FS+0	1	0.5	-	-	-	-

As explained before, the sensitivity of meta-heuristics to parameter selection is vastly documented in the extant literature [3], [7], [8], [45], [69], [103]. The parameters for 29-activity-based problems are also set after numerous trials. The parameters are configured with regard to the solution quality and convergence speed of HGA. The parameter values set for 29-activity problems are tabulated in Table 4.6.

Table 4.6 Parameter Values for the 29-activity Instances

Description	**Problem 29A**	**Problem 29B**
Population Size	60	150
Boltzmann Constant	1	1
Generation Number	30	40
Pc (0.6-1)	0.80	0.80
P_m (0.005-0.05)	1⁄no.Gene (29) ≈ 0.035	1⁄no.Gene (29) ≈ 0.035

As shown in Table 4.6, for the 29-activity problems, the selected population size is set to range between 60 to 150. Also, for the generation number a value is selected from the range 30 to 40; the Boltzmann's Constant is selected to be equal to 1; the crossover probability is set as 0.8; and the mutation probability is set by using Eq. (3.9) with a permissible range of 0.005 to 0.05.

As seen in Table 4.7, for the 290-activity problems different parameter setups are used. Due to higher number of activities, larger population size and iteration number were used to achieve better solutions at the cost of increased processing time.

Table 4.7 Parameter Values for the 290-activity Instances

Description	**Problem 290A**	**Problem 290B**
Population Size	3,000	3,000
Boltzmann Constant	1.2	1.2
Generation Number	500	500
Pc (0.6-1)	0.80	0.80
P_m (0.005-0.05)	1⁄no.Gene (290) ≈ 0.005	1⁄no.Gene (290) ≈ 0.005

As summarized in Table 4.7, a population size of 3,000 is used for both 290A and 290B problems. Similarly, the generation number is also increased to 500 for both problems. Since the 290-activity problems have significantly larger solution spaces compared to the 29-activity problems, larger values were used for the number of population and generation number parameters. Similarly, due to the larger dimension of the solution space, the Boltzmann's Constant was also increased by 20% so as to prevent HGA from getting stuck into the local optima. The same crossover probability of 0.8 is used while the mutation probability is set as 0.005 since Eq. (3.9) results in a value lower than the allowable minimum amount.

The results of the experiments for the 29-activity problems are abstracted in Table 4.8. In the second column of this table the number of successive runs is given. Under 'Best Solution' column, the best solutions obtained are presented. The last two columns tabulate the average processing times and the average percent deviations from the optima, respectively.

Table 4.8 Results Achieved for 29-activity Problems by HGA

Instance	No. of Runs	Conditions				Best Solution		Avg. CPU Time (S)	APD (%)
		C_d (€)	C_c (€)	C_n (€)	T_d (Day)	Total Cost (€)	Total Duration (Day)		
29A	10	1,200	0	0	0	1,226,200	246	0.06	0
29B	10	1,200	500	1,500	240	1,220,700	221	0.22	0

Performance of HGA over 29-activity problems is compared to both optimal solutions and results of the existing methods ([75], [5]). Table 4.9 compiles the results of comparisons made.

Table 4.9 Performance Comparisons over 29-activity Problems

Instance	Best Solution							
	GASA (Bettemir, 2009)		HGAQSA (Bettemir, 2009)		HA (Sonmez and Bettemir, 2012)		HGA (this study)	
	APD (%)	CPU Time (S)	APD (%)	CPU Time (S)	APD (%)	CPU Time (S)	APD (%)	CPU Time (S)
29A	0.00	5.00	0.00	2.00	0.00	-	0.00	0.06
29B	0.00	5.00	0.00	2.00	0.00	-	0.00	0.22

As seen in Table 4.9, the proposed HGA is able to obtain the optimum solutions for all of the ten experimental trials over both the problem types. Besides, the proposed algorithm proves to be more efficient compared to GASA and HGAQSA methods of Bettemir [75] since it converges to global optima within 0.22 seconds; whilst, it takes 2 seconds for both Bettemir's [75] methods to solve the same problems. This experiment further highlights the capabilities of the proposed HGA in solving small size DTCT problems with general logical relationships. Figure 4.6 and Figure 4.7 illustrate the minimum total cost achieved by HGA per each iteration for problems 29A and 29B, respectively. It can be observed that at the initial stages, HGA makes huge leaps toward better quality solutions then it starts improving the solutions rather gradually towards the final rounds of iterations.

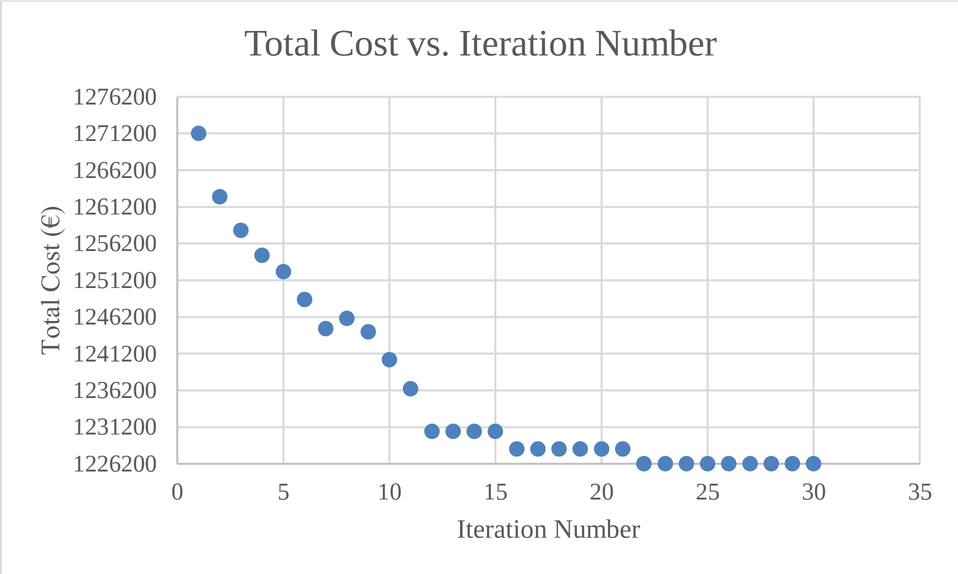

Figure 4.6 Total Cost versus Iteration Number for 29A Problem

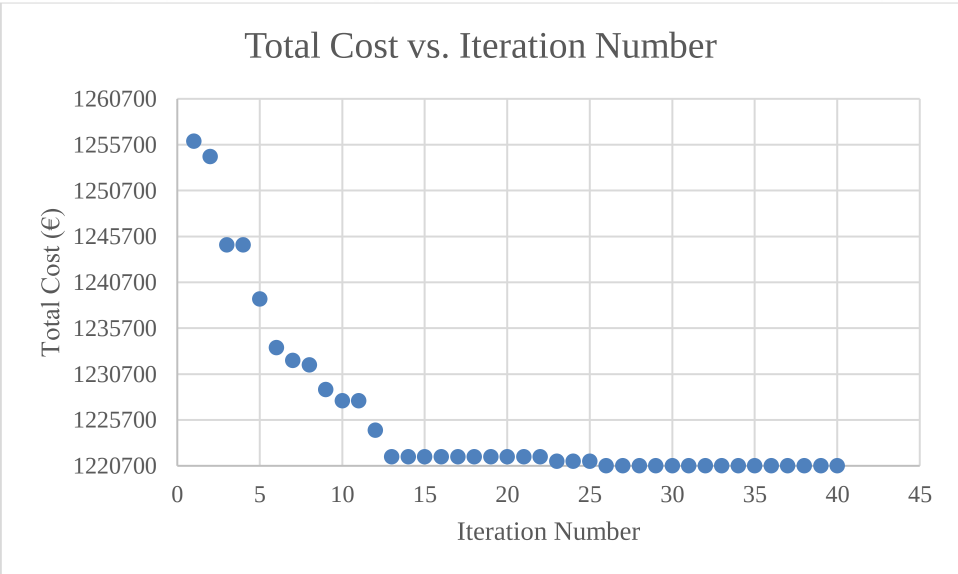

Figure 4.7 Total Cost vs. Iteration Number for 29B problem.

The results of the experiments for the 290-activity problems are demonstrated in Table 4.10. In the second column of this table the number of runs is given. Under 'Best

Solution' column, the best solutions obtained at each run are presented. The last two columns tabulate the processing times and the percent deviations from the optima for each run, respectively.

Table 4.10 Results Achieved for 290-activity Problems by HGA

Instance	**No. of Runs**	**Conditions**				**Best Solution**		**Avg. CPU Time (Min:S)**	**PD (%)**
		***C_d* (€)**	***C_c* (€)**	***C_n* (€)**	***T_d* (Day)**	**Total Cost (€)**	**Total Duration (Day)**		
290A	1	1,200	0	0	0	12,264,000	2,495	13:20	0.02
	2					12,264,000	2,480	13:30	0.02
	3					12,262,000	2,500	13:37	0.00
	4					12,262,000	2,475	13:59	0.00
	5					12,262,000	2,495	12:08	0.00
	6					12,264,000	2,505	11:37	0.02
	7					12,264,000	2,460	11:37	0.02
	8					12,262,000	2,495	11:34	0.00
	9					12,264,000	2,500	11:44	0.02
	10					12,262,000	2,490	11:42	0.00
	Avg.							12:24	0.01
290B	1	1,200	500	1,500	2,400	12,209,900	2,207	11:12	0.02
	2					12,207,900	2,207	11:27	0.01
	3					12,212,300	2,219	11:44	0.04
	4					12,210,300	2,219	11:54	0.03
	5					12,210,900	2,217	11:33	0.03
	6					12,208,400	2,212	11:30	0.01
	7					12,208,400	2,212	11:49	0.01
	8					12,207,000	2,197	11:49	0.00
	9					12,213,800	2,194	12:03	0.06
	10					12,217,300	2,189	12:21	0.08
	Avg.							11:52	0.03

290-activity problems have not been implemented by many researchers before. Hence, the solutions obtained by HGA are compared to those found by Bettemir [75], and Sonmez and Bettemir [5] . Results of the comparisons are presented in Table 4.11.

Table 4.11 Performance Comparisons over 290-activity Problems

Instance	Best Solution							
	GASA (Bettemir, 2009)		HGAQSA (Bettemir, 2009)		HA (Sonmez and Bettemir, 2012)		HGA (this study)	
	APD (%)	CPU Time (Min:S)	APD (%)	CPU Time (Min:S)	APD (%)	CPU Time (Min:S)	APD (%)	CPU Time (Min:S)
290A	4.62	-	0.68	-	0.74	-	0.01	12:24
290B	3.45	-	0.38	-	0.43	-	0.03	11:52

The results provided in Table 4.11 continue to support robustness of HGA since it is able to find high-quality solutions with meager deviations from the optima. On the other hand, HGA is quite successful at finding better solutions than the ones reported by Bettemir [75] ,and Sonmez and Bettemir [5]. It takes about 12 minutes for HGA to unravel the 290-activity problems with the parameter setting discussed earlier. Results of Bettemir [75], and Sonmez and Bettemir [5] provide no grounds for comparing the convergence speeds since in none of the previous attempts the processing times have been reported. Though, 12 minutes can be considered an acceptable CPU time for tackling relatively complex problems that associate several different logical relationships. Figure 4.8 and Figure 4.9 display the minimum total cost achieved by HGA per each iteration for problems 290A and 290B, respectively. Setting Boltzmann's Constant as 1.2 for these problems has caused survival of some bad mutations especially towards the final rounds of the iterations. This, in turn, has facilitated exploration of some unvisited portions of the search domain and has helped HGA to untrap itself from the local optimum points.

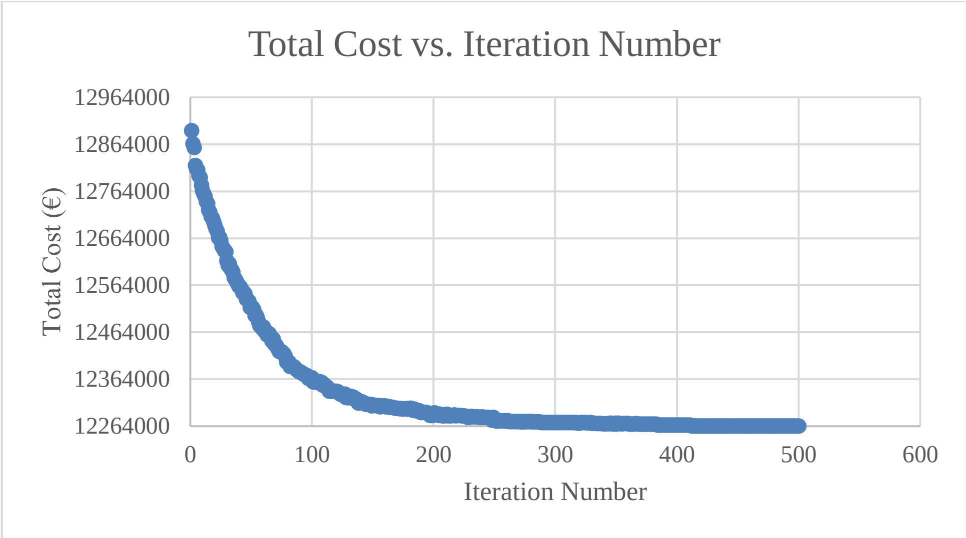

Figure 4.8 Total Cost versus Iteration Number for 290A Problem

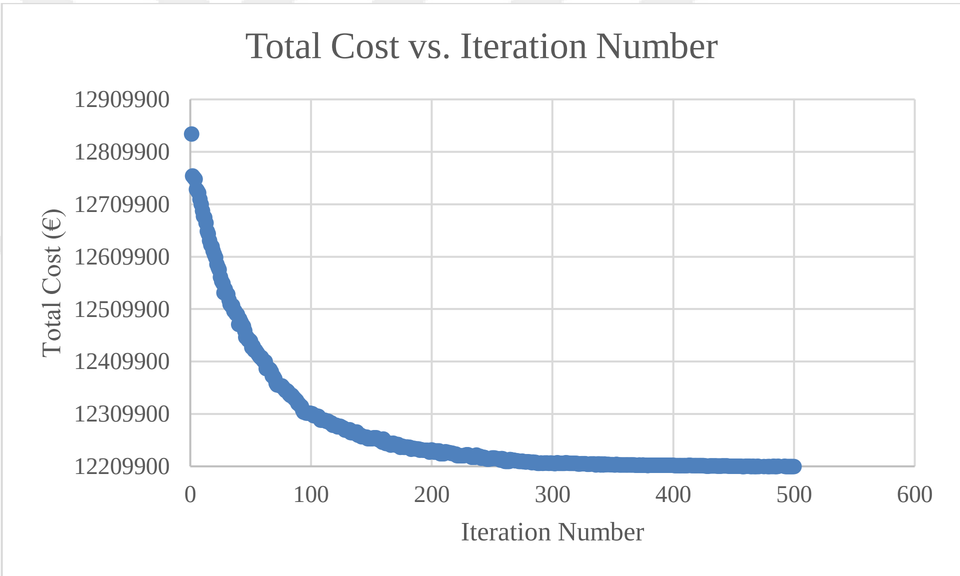

Figure 4.9 Total Cost versus Iteration Number for 290B Problem

For each 290-activity problem, the distribution of the solutions over the search domain at the end of the last iteration are shown in Figures 4.10 and 4.11. Solutions clustered along the convergence curves roughly reveal the pattern for the time-cost curves of the practiced instances.

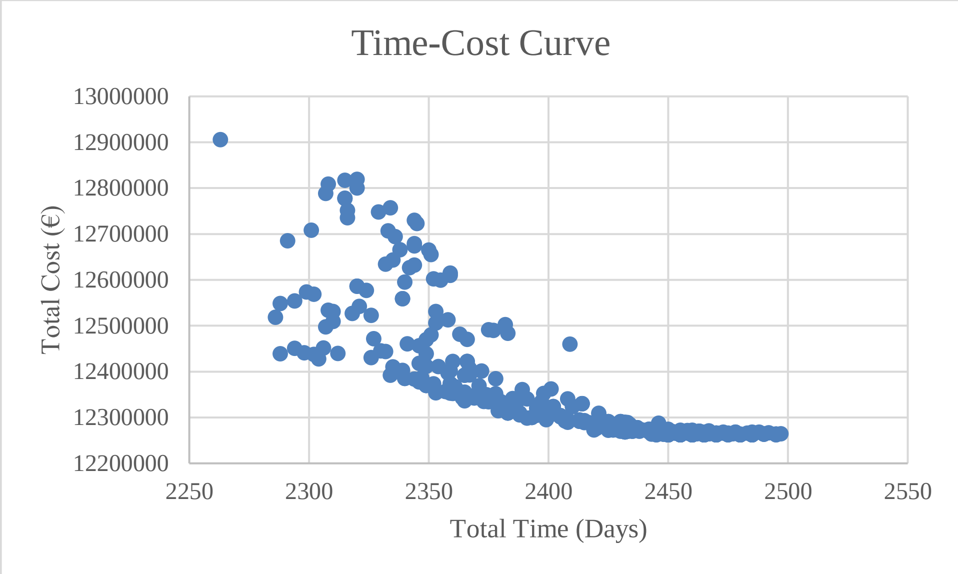

Figure 4.10 Convergence Curve for 290A Problem

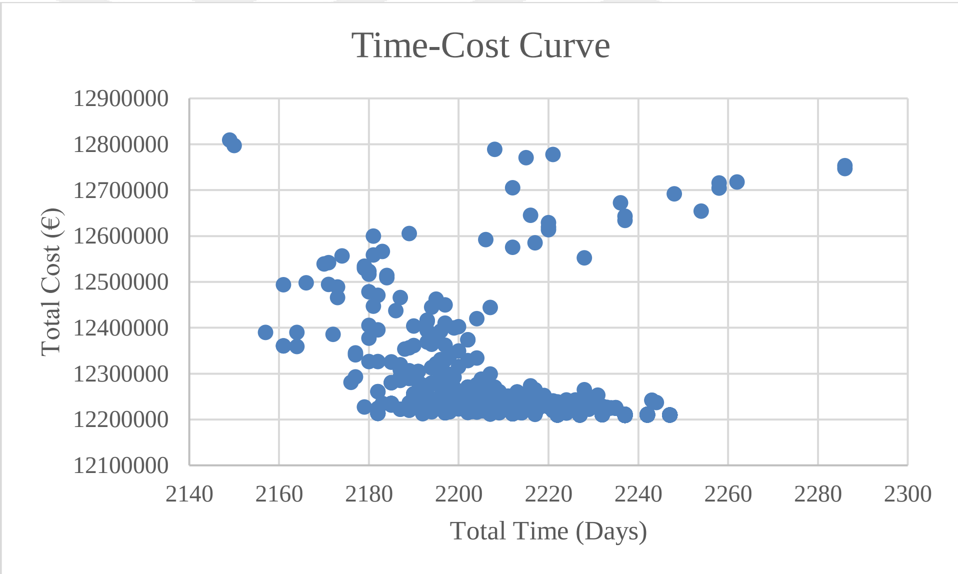

Figure 4.11 Convergence Curve for 290B Problem

4.3.3 63-activity and 630-activity Instances

A 63-activity problem and its derivates have also been frequently used in the literature for performance evaluation of different optimization algorithms. This problem was originally introduced by Sonmez and Bettemir [5] with all the activities having the classical Finish to Start (FS) type of logical relationship. The precedence diagram of this instance for the all-normal (longest and the cheapest) schedule is shown in Figure 4.12. Details of this problem including precedence relations of activities and the time-cost alternatives are given in Table 4.12. Different combinations of the time-cost modes in this problem allows for formation of 1.4×10^{24} particular schedules. This problem is studied under two conditions known as 63A and 63B. These problems are used for implementation of the HGA described in section 3.2. For 63A and 63B problems the indirect costs are assumed to be $2300/day and $3500/day, respectively. Exact solutions of these problems are derived from Sonmez and Bettemir [5] as $5,421,120 for 63A and $6,176,170 for 63B problems. In addition, the ten times duplicated versions of these problems, i.e., 630A and 630B are practiced for implementing the HGA described in section 3.3. The optimal solutions are obtained by Sonmez and Bettemir [5] using an exact method as $54,211,200 and $61,761,700 for 630A and 630B problems, respectively.

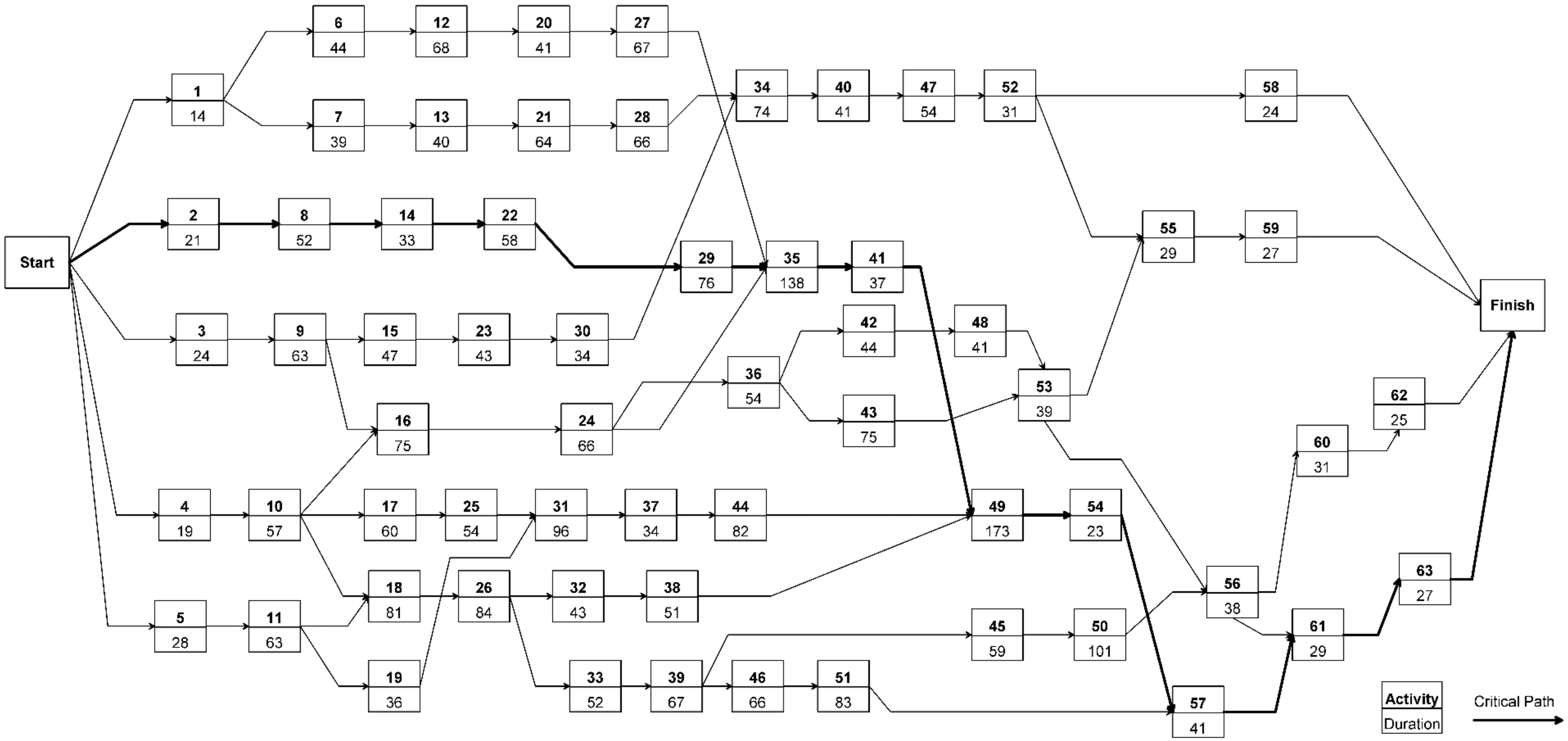

Figure 4.12 Network Diagram for 63-activity Problem [85]

Table 4.12 Activity Table for 63-activity Problem [85]

Activity	Pred.	Mode 1		Mode 2		Mode 3		Mode 4		Mode 5	
		Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $
1	-	14	3,750	12	4,250	10	5,400	9	6,250	-	-
2	-	21	11,250	18	14,800	17	16,200	15	19,650	-	-
3	-	24	22,450	22	24,900	19	27,950	17	31,650	-	-
4	-	19	17,800	17	19,400	15	21,600	-	-	-	-
5	-	28	31,180	26	34,200	23	38,250	21	41,400	-	-
6	1	44	54,260	42	58,450	38	63,225	35	68,150	-	-
7	1	39	47,600	36	50,750	33	54,800	30	59,750	-	-
8	2	52	62,140	47	69,700	44	72,600	39	81,750	-	-
9	3	63	72,750	59	79,450	55	86,250	51	91,500	49	99,500
10	4	57	66,500	53	70,250	50	75,800	46	80,750	41	86,450
11	5	63	83,100	59	89,450	55	97,800	50	104,250	45	112,400
12	6	68	75,500	62	82,000	58	87,500	53	91,800	49	96,550
13	7	40	34,250	37	38,500	33	43,950	31	48,750	-	-
14	8	33	52,750	30	58,450	27	63,400	25	66,250	-	-
15	9	47	38,140	40	41,500	35	47,650	32	54,100	-	-
16	9, 10	75	94,600	70	101,250	66	112,750	61	124,500	57	132,850
17	10	60	78,450	55	84,500	49	91,250	47	94,640	-	-
18	10, 11	81	127,150	73	143,250	66	154,600	61	161,900	-	-
19	11	36	82,500	34	94,800	30	101,700	-	-	-	-
20	12	41	48,350	37	53,250	34	59,450	32	66,800	-	-
21	13	64	85,250	60	92,600	57	99,800	53	107,500	49	113,750
22	14	58	74,250	53	79,100	50	86,700	47	91,500	42	97,400
23	15	43	66,450	41	69,800	37	75,800	33	81,400	30	88,450
24	16	66	72,500	62	78,500	58	83,700	53	89,350	49	96,400
25	17	54	66,650	50	70,100	47	74,800	43	79,500	40	86,800
26	18	84	93,500	79	102,500	73	111,250	68	119,750	62	128,500
27	20	67	78,500	60	86,450	57	89,100	56	91,500	53	94,750
28	21	66	85,000	63	89,750	60	92,500	58	96,800	54	100,500
29	22	76	92,700	71	98,500	67	104,600	64	109,900	60	115,600
30	23	34	27,500	32	29,800	29	31,750	27	33,800	26	36,200
31	19, 25	96	145,000	89	154,800	83	168,650	77	179,500	72	189,100
32	26	43	43,150	40	48,300	37	51,450	35	54,600	33	61,450

Table 4.12 Activity Table for 63-activity Problem (continued) [85]

Activity	Pred.	Mode 1		Mode 2		Mode 3		Mode 4		Mode 5	
		Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $
33	26	52	61,250	49	64,350	44	68,750	41	74,500	38	79,500
34	28, 30	74	89,250	71	93,800	66	99,750	62	105,100	57	114,250
35	24, 27, 29	138	183,000	126	201,500	115	238,000	103	283,750	98	297,500
36	24	54	47,500	49	50,750	42	56,800	38	62,750	33	68,250
37	31	34	22,500	32	24,100	29	26,750	27	29,800	24	31,600
38	32	51	61,250	47	65,800	44	71,250	41	76,500	38	80,400
39	33	67	81,150	61	87,600	57	92,100	52	97,450	49	102,800
40	34	41	45,250	39	48,400	36	51,200	33	54,700	31	58,200
41	35	37	17,500	31	21,200	27	26,850	23	32,300	-	-
42	36	44	36,400	41	39,750	38	42,800	32	48,300	30	50,250
43	36	75	66,800	69	71,200	63	76,400	59	81,300	54	86,200
44	37	82	102,750	76	109,500	70	127,000	66	136,800	63	146,000
45	39	59	84,750	55	91,400	51	101,300	47	126,500	43	142,750
46	39	66	94,250	63	99,500	59	108,250	55	118,500	50	136,000
47	40	54	73,500	51	78,500	47	83,600	44	88,700	41	93,400
48	42	41	36,750	39	39,800	37	43,800	34	48,500	31	53,950
49	38, 41, 44	173	267,500	159	289,700	147	312,000	138	352,500	121	397,750
50	45	101	47,800	74	61,300	63	76,800	49	91,500	-	-
51	46	83	84,600	77	93,650	72	98,500	65	104,600	61	113,200
52	47	31	23,150	28	27,600	26	29,800	24	32,750	21	35,200
53	43, 48	39	31,500	36	34,250	33	37,800	29	41,250	26	44,600
54	49	23	16,500	22	17,800	21	19,750	20	21,200	18	24,300
55	52, 53	29	23,400	27	25,250	26	26,900	24	29,400	22	32,500
56	50, 53	38	41,250	35	44,650	33	47,800	31	51,400	29	55,450
57	51, 54	41	37,800	38	41,250	35	45,600	32	49,750	30	53,400
58	52	24	12,500	22	13,600	20	15,250	18	16,800	16	19,450
59	55	27	34,600	24	37,500	22	41,250	19	46,750	17	50,750
60	56	31	28,500	29	30,500	27	33,250	25	38,000	21	43,800
61	56, 57	29	22,500	27	24,750	25	27,250	22	29,800	20	33,500
62	60	25	38,750	23	41,200	21	44,750	19	49,800	17	51,100
63	61	27	9,500	26	9,700	25	10,100	24	10,800	22	12,700

As the number of the activities increase, the project becomes more complicated. Hence, population size and iteration number are configured according to the complexity of these problems. The values selected for the parameters are given in Table 4.13.

Table 4.13 Parameter Values for the 63-activity Instances

Description	Problem 63A	Problem 63B
Population Size	100,000	100,000
Boltzmann Constant	1.5	1.5
Generation Number	250	250
Pc (0.6-1)	0.80	0.80
P_m (0.005-0.05)	1⁄no.Gene (63) ≈ 0.015	1⁄no.Gene (63) ≈ 0.015

As tabulated in Table 4.13, identical parameter values are set for both problems 63A and 63B. With regard to the solution quality and convergence speed of HGA, a population size of 100,000 and a generation number of 250 are used. Similarly, due to the large dimension of the solution space, a Boltzmann's Constant of 1.5 was used to provide more exploration capabilities to HGA. Crossover probability of 0.8 is used with the mutation probability being set by using Eq. (3.9) with an allowable range of 0.005 to 0.05.

Evidently, as the number of activities raise to 630, the dimensions of the solution space expand exponentially. Thus, for such a complex problem, larger population size and/or generation number will be needed to search over the solution space However, in order to keep the CPU time within half-an-hour range, the reduction of the population size was a prerequisite. Thus, choosing a smaller population size was preferred to obtain an acceptable performance with satisfactory results. For this reason, the parameter values were re-adjusted as given in Table 4.14 for the 630-activity problems.

Table 4.14 Parameter Values for the 630-activity Instances

Description	Problem 630A	Problem 630B
Population Size	1,500	1,500
Boltzmann Constant	0.5	0.5
Generation Number	1,300	1,300
Pc (0.6-1)	0.80	0.80
P_m (0.005-0.05)	1⁄no.Gene (630) ≈ 0.005	1⁄no.Gene (630) ≈ 0.005

As summarized in tale 4.14, the same set of parameter values are used for both 630A and 630B problems. A population size of 1,500 with an increased iteration number of 1,300 are used. The Boltzmann's Constant for these problems was decreased to 0.5 in order to provide a more focused exploration by limiting the selection of bad mutations. The same crossover probability of 0.8 is used while the mutation probability is set as 0.005 since Eq. (3.9) results in a value lower than the permissible minimum amount.

The results of the experiments for the 63-activity and the 630-activity problems are illustrated in Tables 4.15 and 4.16, respectively. In the second column of these tables the number of runs is given. Under 'Best Solution' column, the best solutions obtained at each run are presented. The last two columns tabulate the processing times and the percent deviations from the optima for each run, respectively.

Table 4.15 Results Achieved for 63-activity Problems by HGA

Instance	No. of Runs	Conditions				Best Solution		Avg. CPU Time (Min:S)	PD (%)
		C_d (€)	C_c (€)	C_n (€)	T_d (Day)	Total Cost (€)	Total Duration (Day)		
63A	1	2,300	0	0	0	5,422,220	635	46:16	0.02
	2					5,427,770	630	34:10	0.12
	3					5,422,770	633	49:01	0.03
	4					5,425,870	635	34:02	0.09
	5					5,424,340	633	47:41	0.06
	6					5,428,810	632	40:06	0.14
	7					5,429,270	633	41:36	0.15
	8					5,425,120	633	46:34	0.07
	9					5,424,820	632	45:10	0.07
	10					5,430,580	629	54:22	0.17
	Avg.							39.03	0.09
63B	1	3,500	0	0	0	6,181,290	619	48:39	0.08
	2					6,190,820	619	49:07	0.24
	3					6,184,970	618	49:40	0.14
	4					6,185,620	616	51:24	0.15
	5					6,185,020	619	48:50	0.14
	6					6,184,670	617	47:05	0.14
	7					6,186,630	615	46:49	0.17
	8					6,179,370	624	47:10	0.05
	9					6,181,620	621	46:47	0.09
	10					6,186,630	616	47:04	0.17
	Avg.							48:07	0.14

Table 4.16 Results Achieved for 630-activity Problems by HGA

Instance	No. of Runs	Conditions				Best Solution		Avg. CPU Time (Min:S)	PD (%)
		C_d (€)	C_c (€)	C_n (€)	T_d (Day)	Total Cost (€)	Total Duration (Day)		
630A	1	2,300	0	0	0	55,040,970	6,238	35:52	1.53
	2					55,093,300	6,272	34:09	1.63
	3					55,151,240	6,255	34:46	1.73
	4					55,123,635	6,222	34:10	1.68
	5					55,117,185	6,240	32:40	1.67
	6					55,151,650	6,277	33:11	1.73
	7					55,159,865	6,222	33:22	1.75
	8					55,092,850	6,219	33:31	1.63
	9					55,141,080	6,224	33:09	1.72
	10					55,111,910	6,239	33:01	1.66
	Avg.							33:63	1.67
630B	1	3,500	0	0	0	62,753,345	5,982	32:08	1.61
	2					62,712,545	6,013	33:19	1.54
	3					62,903,635	6,020	33:39	1.85
	4					62,756,605	6,049	32:35	1.61
	5					62,796,890	6,061	32:34	1.68
	6					62,840,210	6,036	32:54	1.75
	7					62,856,220	5,967	33:17	1.77
	8					62,793,895	5,963	32:56	1.67
	9					62,831,515	6,003	31:30	1.73
	10					62,738,425	6,070	33:25	1.58
	Avg.							32:61	1.68

Performance of HGA over 63-activity problems is compared to both optimal solutions and results of the existing methods ([75], [5], [6], [104]). Table 4.17 compiles the results of comparisons made.

Table 4.17 Performance Comparisons over 63-activity Problems

Instance	Best Solution									
	GASA (Bettemir, 2009)		HA (Sonmez and Bettemir, 2012)		DPSO (Aminbakhsh and Sonmez, 2016)		MAWA-TLBO (Eirgash, 2018)		HGA (this study)	
	APD (%)	CPU Time (Min:S)	APD (%)	CPU Time (Min:S)	APD (%)	CPU Time (Min:S)	APD (%)	CPU Time (Min:S)	APD (%)	CPU Time (Min:S)
63A	0.01	-	2.61	-	0.02	0:1.3	3.53	-	0.09	39:03
63B	0.07	-	2.50	-	0.05	0:1.3	1.17	-	0.14	48:07

The comparisons reveal that HGA is able to achieve higher-quality solutions than those reported by Sonmez and Bettemir [5] and Eirgash [104]. Though, both DPSO and GASA are able to locate solutions with smaller deviation amounts than the ones found by HGA. Results confirm the capability of the proposed HGA in providing satisfactory results with slim PDs for complex networks. As seen in Table 4.17, the processing times for none of the existing methods are available in the literature except for DPSO. This PSO-based algorithm, due to its fast convergence can unravel 63-activity problems in 1.3 minutes; whereas, this duration is around half an hour for HGA.

Similarly, the results of HGA for 630- activity problems are evaluated taking into account both the optimal solutions and the results available in the literature. As presented in Table 4.18, HGA's results are compared with Bettemir [75], Sonmez and Bettemir [5], Aminbakhsh and Sonmez [6], and Eirgash [104].

Table 4.18 Performance Comparisons over 630-activity Problems

Instance	Best Solution									
	GASA (Bettemir, 2009)		HA (Sonmez and Bettemir, 2012)		DPSO (Aminbakhsh and Sonmez, 2016)		NDS-TLBO (Eirgash, 2018)		HGA (this study)	
	APD (%)	CPU Time (Min:S)	APD (%)	CPU Time (Min:S)	APD (%)	CPU Time (Min:S)	APD (%)	CPU Time (Min:S)	APD (%)	CPU Time (Min:S)
630A	8.37	73:00	2.41	-	0.33	00:14.6	1.10	-	1.67	33:63
630B	7.32	73:00	2.47	-	0.34	00:14.6	1.51	-	1.68	32:61

Results show that HGA is able to obtain higher-quality solutions when compared to those published by Bettemir [75]and Sonmez and Bettemir [5]. Although HGA exhibits an acceptable performance for solution of large-scale 630-activity problems, it is not able to excel DPSO or NDS-TLBO at least when the parameters are configured as given in Table 4.14.

Figure 4.13 and Figure 4.14 illustrate the convergence history of HGA for 63A and 63B problems, respectively. Similarly, Figures 4.15 and 4.16 presents the minimum total cost obtained by HGA per each iteration for problems 630A and 630B, respectively. Figures 4.13 to 4.16 reveal HGA performs relatively the same over the solution spaces of 63-activity and 630-activity problems. For 63-activity problems it can be observed that HGA has converged to the best solution of that specific run at the 150th iteration which is well ahead of reaching the maximum number of iterations. Though, using smaller iteration number was experienced to deteriorate the overall quality of the solutions over the 10 consecutive runs. In addition, as discussed earlier in Section 3.2.5, HGA is not designed to terminate if no improvements are made within the last rounds of iterations since it intervenes with the hill-climbing strategy of Simulated Annealing (SA). More precisely, SA module of HGA is sometimes experienced to kick in after a long – e.g., 50 iterations – tally of identical solutions. For 630-activity problems, due to high numbers of activities and time-cost alternatives, the best solution is constantly found toward the final rounds of iterations over the 10 successive runs.

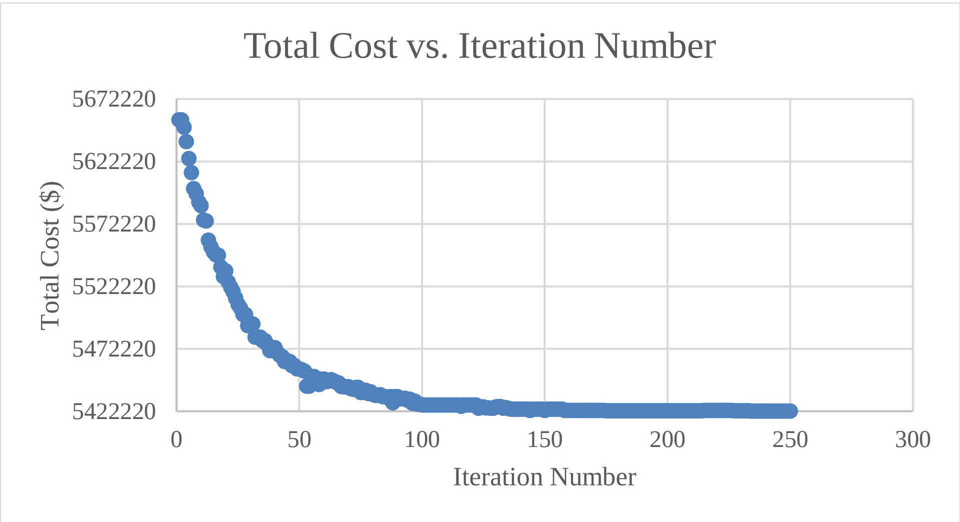

Figure 4.13 Total Cost versus Iteration Number for 63A Problem

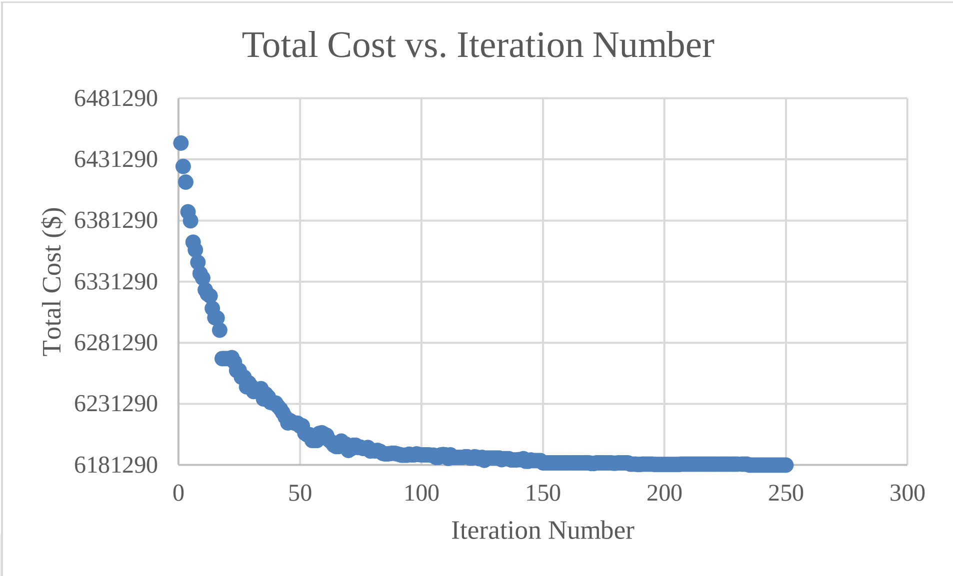

Figure 4.14 Total Cost versus Iteration Number for 63B Problem

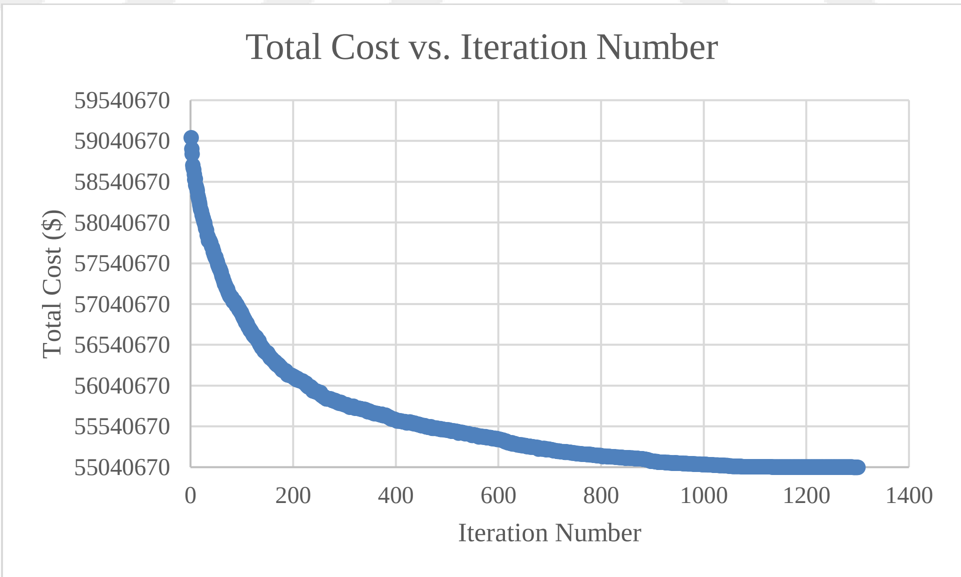

Figure 4.15 Total Cost versus Iteration Number for 630A Problem

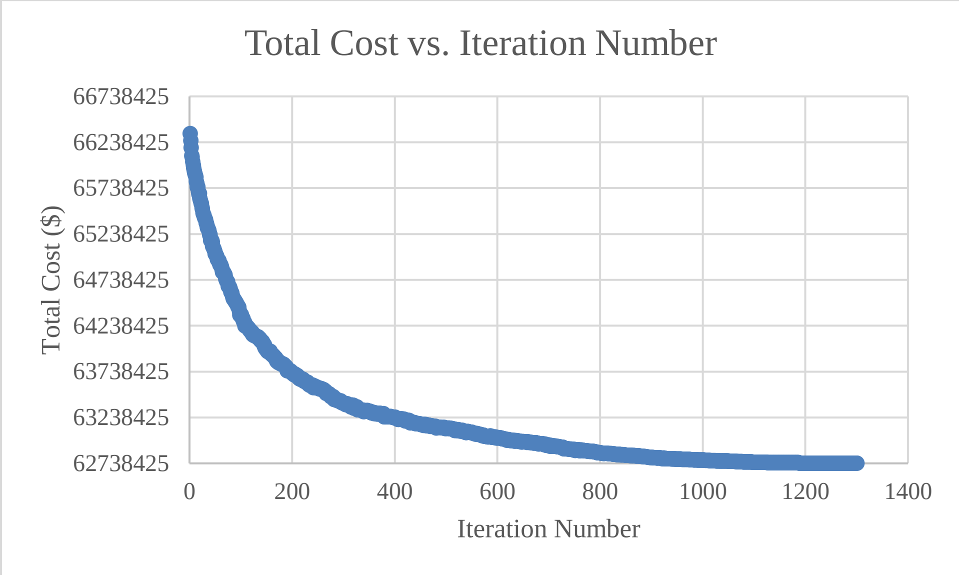

Figure 4.16 Total Cost versus Iteration Number for 630B Problem

For each 63-activity and 630-activity problems, the distribution of the solutions over the search domain at the end of the last iteration are shown in Figures 4.17 to 4.20. Distributions of the solutions along the convergence curves practically reveal the topography of the solution space for each of the practiced instances

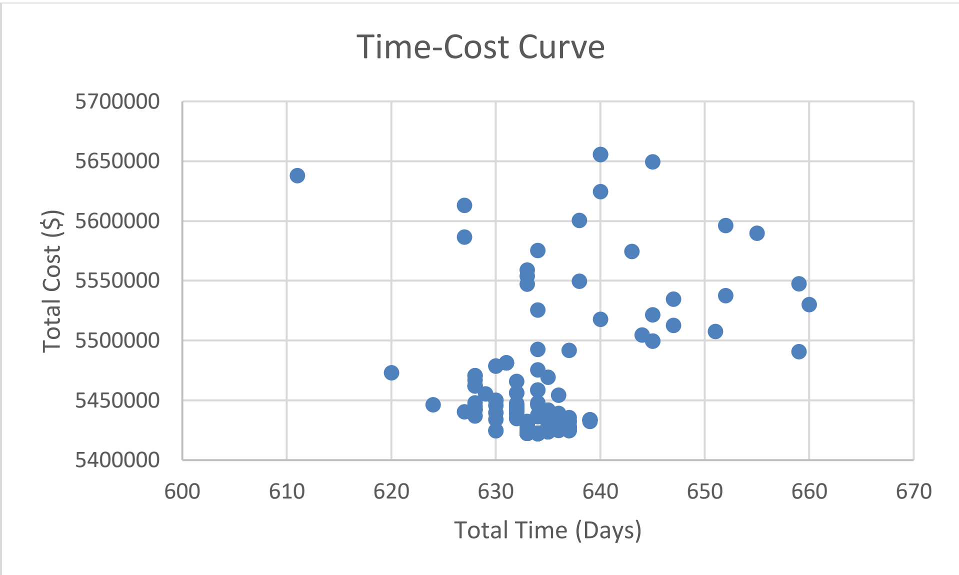

Figure 4.17 Convergence Curve for 63A Problem

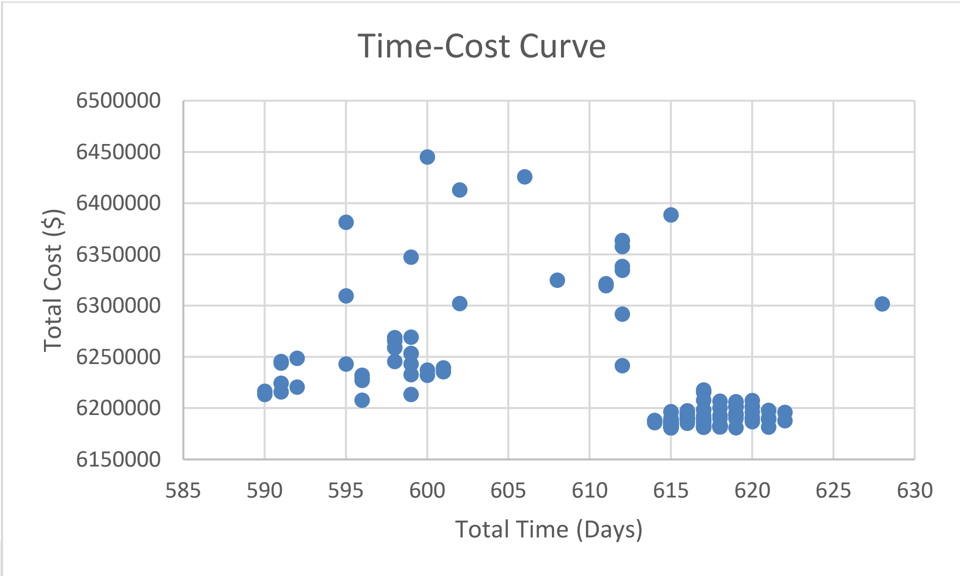

Figure 4.18 Convergence Curve for 63B Problem

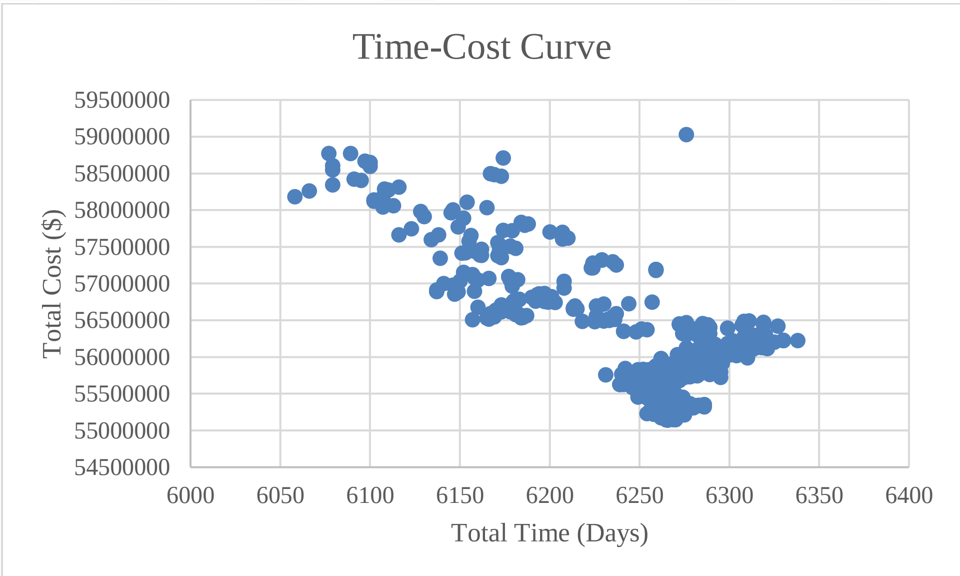

Figure 4.19 Convergence Curve for 630A Problem

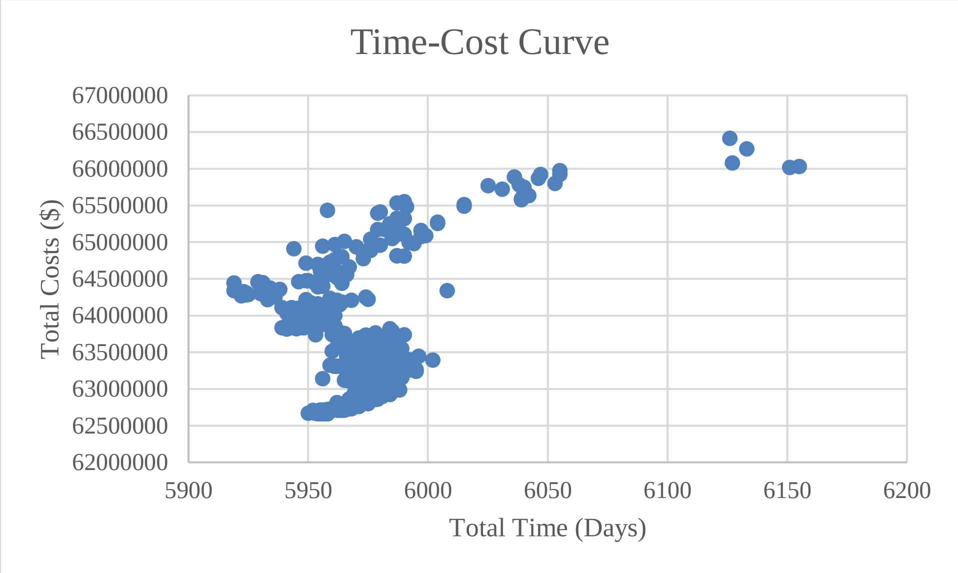

Figure 4.20 Convergence Curve for 630B Problem

4.3.4 990-activity Instances

The performance and capabilities of HGA is further evaluated by implementing twelve large-scale instances with 990 activities including T990_01, T990_11, T990_21, T990_31, T990_41, T990_51, T990_61, T990_71, T990_81, T990_91, T990_101, and T990_111 problems. These instances were constructed by Aminbakhsh [102] and later used by Sonmez et al. [58]. 990-activity problems combine four different network complexity indices with three different time-cost alternative sizes (three, six, and nine modes). For all these instances, the desired completion duration is set to be equal to the average of the earliest and the latest permissible times for completing the project. The project indirect cost is fixed at $1,000/day and a delay payment rate of $2,000/day is assumed for all the problems.

The parameters are configured after numerous trials with regard to the solution quality and convergence speed of HGA. Table 4.19 displays parameter setup and the assumed contractual conditions for each 990-activity problem.

Table 4.19 Project Information and Parameter Values for the 990-activity Instances

Instance	Parameters					Conditions		
	Population Size	Boltzmann Constant	Generation Number	*Pc* (0.6-1)	P_m (0.005-0.05)	C_d ($)	C_n ($)	T_d (Day)
T990_1	1,500	0.01	500	0.8	0.005	1,000	2,000	5,724
T990_11	1,500	0.01	750	0.8	0.005	1,000	2,000	5,568
T990_21	2,000	0.01	750	0.8	0.005	1,000	2,000	5521
T990_31	1,000	0.10	1,500	0.8	0.005	1,000	2,000	11,360
T990_41	1,000	0.10	1,500	0.8	0.005	1,000	2,000	11,114
T990_51	2,000	0.10	1,500	0.8	0.005	1,000	2,000	11,003
T990_61	2,000	0.10	1,000	0.8	0.005	1,000	2,000	16,996
T990_71	2,000	0.05	2,000	0.8	0.005	1,000	2,000	16,648
T990_81	1,000	0.10	1,500	0.8	0.005	1,000	2,000	16,523
T990_91	1,000	0.10	1,500	0.8	0.005	1,000	2,000	22,317
T990_101	1,500	0.10	2,000	0.8	0.005	1,000	2,000	21,911
T990_111	1,500	0.10	2,000	0.8	0.005	1,000	2,000	21,942

HGA is found to be very sensitive to Boltzmann's Constant for the 990-activity problems. Better results are obtained when a very small number of badly mutated strings are accepted. This is facilitated by setting a lower Boltzmann's Constant value. The results achieved for 990-activity instances are expressed in Table 4.20.

Table 4.20 Results Achieved for 990-activity Problems by HGA

Instance	Best Solution			
	No. of Time-Cost Modes	Total cost ($)	Total duration (Day)	CPU time (Hr:Min)
T990_1	3	64,108,971	5,057	00:25
T990_11	6	72,000,420	4,674	00:38
T990_21	9	74,223,084	4,439	00:51
T990_31	3	69,997,070	7,643	00:27
T990_41	6	76,090,722	6,074	00:43
T990_51	9	78,731,122	6,068	01:23
T990_61	3	75,138,471	9,792	00:55
T990_71	6	80,763,052	7,212	01:33
T990_81	9	82,791,300	6,479	00:37
T990_91	3	76,312,870	9,887	00:31
T990_101	6	82,837,470	5,878	01:05
T990_111	9	85,597,883	5,116	01:11

Not for all the 990-activity problems the optimal solutions are available. Global optimum solutions for problems T990_1 to T990_81 have not been reported yet in the literature. Thus, for this set of problems, results of HGA are only compared with the findings of Aminbakhsh [102]. Consequent to comparisons, it is concluded that HGA, by running for relatively the same amount of processing time, cannot locate solutions as good in quality as the CS-Heuristic. Solutions achieved by HGA deviates from those obtained using CS-Heuristic by 0.71 to 4.3%. Results of the comparisons are presented in Table 4.21.

Table 4.21 Performance Comparisons over Problems T990_1 to T990_81

Instance	Best Solution			
	CS-Heuristic (Aminbakhsh, 2018)		HGA (this study)	
	Total cost ($)	CPU time (Hr:Min)	Total cost ($)	CPU time (Hr:Min)
T990_1	62,941,272	00:17	64,108,971	00:25
T990_11	69,421,199	00:38	72,000,420	00:38
T990_21	71,154,452	00:52	74,223,084	00:51
T990_31	69,127,419	00:28	69,997,070	00:27
T990_41	74,375,755	00:52	76,090,722	00:43
T990_51	76,708,820	01:14	78,731,122	01:23
T990_61	74,610,640	00:34	75,138,471	00:55
T990_71	79,952,754	00:57	80,763,052	01:33
T990_81	81,676,727	01:06	82,791,300	00:37

For the instances that have previously been solved to global optimality, i.e., problems T990_91, T990_101, and T990_111, average percent deviations are calculated and compared to the results of CS-Heuristic. It is noted that HGA slightly misses the global optima of the foregoing three instances by finding solutions marginally lower in quality than those reported in Aminbakhsh [102]. Nevertheless, solid results obtained by HGA even for such real-life-size problems further substantiate practicality of this model. Comparison of HGA's results with the optima for 990-activity problems are tabulated in Table 4.22.

Table 4.22 Performance Comparisons over Problems T990_91 to T990_111

Instance	Best Solution				
	Optimum Solution	CS-Heuristic (Aminbakhsh, 2018)		HGA (this study)	
	Total Cost ($)	APD (%)	CPU time (Hr:Min)	APD (%)	CPU time (Hr:Min)
T990_91	76,021,561	0.01	00:15	0.38	00:31
T990_101	82,463,696	0.02	00.21	0.45	01:05
T990_111	84,982,380	0.05	00:24	0.72	01:11

4.3.5 Real-life Case Project

Sequel to the performance evaluations of HGA for problems obtained from the literature, an actual construction project is also used to demonstrate the real-life applicability and effectiveness of the proposed method. As mentioned earlier, only a few studies have exercised problems incorporating the generalized logical relations with realistic overlapping among the activities. Notwithstanding the limited number of test problems in this domain, this study takes on the task of exerting an optimization method capable of tackling an actual construction project with properly overlapped activities. The real project is fed into HGA and the obtained time-cost alternative selections are compared to the experienced-based decisions made by the experts during the planning stage. Quality of the solution obtained by HGA for the real project firmly suggests that this technique can come to the aid of decision makers and project managers in proper selection of different construction methods available. In the following section, the actual construction project which includes over four hundred activities with generalized precedence relationships will be described.

4.3.5.1 Real-life Project Information

The project was located in one of the governorates of Iraq the scope of which contains six identical 25-storey residential buildings. Only one of the identical buildings will be used for TCT analysis since they share the same schedule and are associated with the same set of execution modes. The project under consideration is handed over and the related data are all retrieved from the company's archives.

The collected data include information related to all the phases from site preparation to the final hand-over of the building. The sources of project data are listed below:

- Project schedule prepared in Microsoft Project;
- Estimated Bill of Quantity (B.o.Q);
- Contractor tenders;
- Vendor quotations;
- Priced Bill of Quantities of the awarded tenderer;
- The actual total project cost and the actual total project duration.

Details of the real-life construction project, covering the general precedence relations of activities and the time-cost modes are given in Table 4.23. The first mode denotes the actual time and cost spent to execute the tasks; while, the second and the third modes represent the bid offers of alternative contractors. Some activities have only one mode due to direct execution by the employer, and some others have only two modes due to the uniqueness of the work item, with only two tenders received from the bidders. The precedence relationships among the activities are obtained from the baseline schedule prepared in Microsoft Project. According the baseline schedule, most of the activities are defined to have a Finish to Start (FS) type of relation, some of which also use lag time or lead time concepts to secure the time to be delayed or to be advanced, respectively. In the schedule there is also a single activity with a Start to Start (SS) type of relationship.

Table 4.23 Activity Table for the 447-activity Case Project

Activity	Pred.	Mode 1		Mode 2		Mode 3	
		Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $
1	-	1	-	-	-	-	-
2	1FS+0	3	229,523.71	-	-	-	-
3	2FS+0	3	382,500.00	-	-	-	-
4	1FS+0	8	698,750.00	7	758,916.60	-	-
5	4FS+0	1	42,430.00	1	47,044.64	-	-
6	5FS+0	1	42,430.00	1	47,044.64	-	-
7	6FS+0	1	42,430.00	1	47,044.64	-	-
8	7FS+0	1	42,430.00	1	47,044.64	-	-
9	6FS+0	6	57,510.00	5	62,707.91	-	-
10	8FS+0	6	57,510.00	5	62,707.91	-	-
11	9FS+0	1	17,445.00	1	19,262.89	-	-
12	11FS+0	1	17,445.00	1	19,292.89	-	-
13	10FS+0	1	17,445.00	1	19,292.89	-	-
14	13FS+0	1	17,445.00	1	19,292.89	-	-
15	12FS+0	6	48,641.28	5	53,023.90	-	-
16	14FS+0	6	48,641.28	5	53,023.90	-	-
17	15FS+0	1	17,443.10	1	19,016.47	-	-
18	17FS+0	1	17,443.10	1	19,016.47	-	-
19	16FS+0	1	17,443.10	1	19,016.47	-	-
20	19FS+0	1	17,443.10	1	19,016.47	-	-
21	18FS+0	6	49,553.17	5	53,793.56	-	-
22	20FS+0	6	49,553.17	5	53,793.56	-	-
23	21FS+0	1	17,443.16	1	18,884.56	-	-
24	23FS+0	1	17,443.16	1	18,884.56	-	-
25	22FS+0	1	17,443.16	1	18,884.56	-	-
26	25FS+0	1	17,443.16	1	18,884.56	-	-
27	24FS+0	6	49,553.17	5	53,793.56	-	-
28	26FS+0	6	49,553.17	5	53,793.56	-	-
29	27FS+0	1	17,443.16	1	18,884.56	-	-
30	29FS+0	1	17,443.16	1	18,884.56	-	-
31	28FS+0	1	17,443.16	1	18,884.56	-	-
32	31FS+0	1	17,443.16	1	18,884.56	-	-
33	30FS+0	6	49,553.17	5	53,793.56	-	-
34	32FS+0	6	49,553.17	5	53,793.56	-	-
35	33FS+0	1	17,443.16	1	18,884.56	-	-
36	35FS+0	1	17,443.16	1	18,884.56	-	-
37	34FS+0	1	17,443.16	1	18,884.56	-	-
38	37FS+0	1	17,443.16	1	18,884.56	-	-
39	36FS+0	6	49,553.17	5	53,793.56	-	-
40	38FS+0	6	49,553.17	5	53,793.56	-	-
41	39FS+0	1	17,443.16	1	18,884.56	-	-

Table 4.23 Activity Table for the 447-activity Case Project (continued)

Activity	Pred.	Mode 1		Mode 2		Mode 3	
		Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $
42	41FS+0	1	17,443.16	1	18,884.56	-	-
43	40FS+0	1	17,443.16	1	18,884.56	-	-
44	43FS+0	1	17,443.16	1	18,884.56	-	-
45	42FS+0	6	49,553.17	5	53,793.56	-	-
46	44FS+0	6	49,553.17	5	53,793.56	-	-
47	45FS+0	1	17,443.16	1	18,884.56	-	-
48	47FS+0	1	17,443.16	1	18,884.56	-	-
49	46FS+0	1	17,443.16	1	18,884.56	-	-
50	49FS+0	1	17,443.16	1	18,884.56	-	-
51	50FS+0	6	49,553.17	5	53,793.56	-	-
52	51FS+0	6	49,553.17	5	53,793.56	-	-
53	51FS+0	1	17,443.16	1	18,884.56	-	-
54	53FS+0	1	17,443.16	1	18,884.56	-	-
55	52FS+0	1	17,443.16	1	18,884.56	-	-
56	55FS+0	1	17,443.16	1	18,884.56	-	-
57	54FS+0	6	49,553.17	5	53,793.56	-	-
58	56FS+0	6	49,553.17	5	53,793.56	-	-
59	57FS+0	1	17,443.16	1	18,884.56	-	-
60	59FS+0	1	17,443.16	1	18,884.56	-	-
61	58FS+0	1	17,443.16	1	18,884.56	-	-
62	61FS+0	1	17,443.16	1	18,884.56	-	-
63	60FS+0	6	49,553.17	5	53,793.56	-	-
64	62FS+0	6	49,553.17	5	53,793.56	-	-
65	63FS+0	1	17,443.16	1	18,884.56	-	-
66	65FS+0	1	17,443.16	1	18,884.56	-	-
67	64FS+0	1	17,443.16	1	18,884.56	-	-
68	67FS+0	1	17,443.16	1	18,884.56	-	-
69	66FS+11	7	49,553.17	6	53,793.56	-	-
70	68FS+14	7	49,553.17	6	53,793.56	-	-
71	69FS+0	1	17,443.16	1	18,884.56	-	-
72	71FS+0	1	17,443.16	1	18,884.56	-	-
73	70FS+0	1	17,443.16	1	18,884.56	-	-
74	73FS+0	1	17,443.16	1	18,884.56	-	-
75	72FS+1	7	49,553.17	6	53,793.56	-	-
76	74FS+6	7	49,553.17	6	53,793.56	-	-
77	75FS+0	1	17,443.16	1	18,884.56	-	-
78	77FS+0	1	17,443.16	1	18,884.56	-	-
79	76FS+0	1	17,443.16	1	18,884.56	-	-
80	79FS+0	1	17,443.16	1	18,884.56	-	-
81	78FS+4	7	49,553.17	6	53,793.56	-	-
82	80FS+4	7	49,553.17	6	53,793.56	-	-

Table 4.23 Activity Table for the 447-activity Case Project (continued)

Activity	Pred.	Mode 1		Mode 2		Mode 3	
		Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $
83	81FS+0	1	17,443.16	1	18,884.56	-	-
84	83FS+0	1	17,443.16	1	18,884.56	-	-
85	82FS+0	1	17,443.16	1	18,884.56	-	-
86	85FS+0	1	17,443.16	1	18,884.56	-	-
87	84FS+0	7	49,553.17	6	53,793.56	-	-
88	86FS+0	7	49,553.17	6	53,793.56	-	-
89	87FS+0	1	17,443.16	1	18,884.56	-	-
90	89FS+0	1	17,443.16	1	18,884.56	-	-
91	88FS+0	1	17,443.16	1	18,884.56	-	-
92	91FS+0	1	17,443.16	1	18,884.56	-	-
93	90FS+0	7	49,553.17	6	53,793.56	-	-
94	92FS+0	7	49,553.17	6	53,793.56	-	-
95	93FS+0	1	17,443.16	1	18,884.56	-	-
96	95FS+0	1	17,443.16	1	18,884.56	-	-
97	94FS+0	1	17,443.16	1	18,884.56	-	-
98	97FS+0	1	17,443.16	1	18,884.56	-	-
99	96FS+0	7	49,553.17	6	53,793.56	-	-
100	98FS+0	7	49,553.17	6	53,793.56	-	-
101	99FS+0	1	17,443.16	1	18,884.56	-	-
102	101FS+0	1	17,443.16	1	18,884.56	-	-
103	100FS+0	1	17,443.16	1	18,884.56	-	-
104	103FS+0	1	17,443.16	1	18,884.56	-	-
105	102FS+0	7	49,553.17	6	53,793.56	-	-
106	104FS+0	7	49,553.17	6	53,793.56	-	-
107	105FS+0	1	17,443.16	1	18,884.56	-	-
108	107FS+0	1	17,443.16	1	18,884.56	-	-
109	106FS+0	1	17,443.16	1	18,884.56	-	-
110	109FS+0	1	17,443.16	1	18,884.56	-	-
111	108FS+0	7	49,553.17	6	53,793.56	-	-
112	110FS+0	7	49,553.17	6	53,793.56	-	-
113	111FS+0	1	17,443.16	1	18,884.56	-	-
114	113FS+0	1	17,443.16	1	18,884.56	-	-
115	112FS+0	1	17,443.16	1	18,884.56	-	-
116	115FS+0	1	17,443.16	1	18,884.56	-	-
117	114FS+0	7	49,553.17	6	53,793.56	-	-
118	116FS+0	7	49,553.17	6	53,793.56	-	-
119	117FS+0	1	17,443.16	1	18,884.56	-	-
120	119FS+0	1	17,443.16	1	18,884.56	-	-
121	118FS+0	1	17,443.16	1	18,884.56	-	-
122	121FS+0	1	17,443.16	1	18,884.56	-	-
123	120FS+0	7	49,553.17	6	53,793.56	-	-

Table 4.23 Activity Table for the 447-activity Case Project (continued)

Activity	Pred.	Mode 1		Mode 2		Mode 3	
		Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $
124	122FS+0	7	49,553.17	6	53,793.56	-	-
125	123FS+0	1	17,443.16	1	18,884.56	-	-
126	125FS+0	1	17,443.16	1	18,884.56	-	-
127	124FS+0	1	17,443.16	1	18,884.56	-	-
128	127FS+0	1	17,443.16	1	18,884.56	-	-
129	126FS+0	7	49,553.17	6	53,793.56	-	-
130	128FS+0	7	49,553.17	6	53,793.56	-	-
131	129FS+0	1	17,443.16	1	18,884.56	-	-
132	131FS+0	1	17,443.16	1	18,884.56	-	-
133	130FS+0	1	17,443.16	1	18,884.56	-	-
134	133FS+0	1	17,443.16	1	18,884.56	-	-
135	132FS+0	7	49,553.17	6	53,793.56	-	-
136	134FS+0	7	49,553.17	6	53,793.56	-	-
137	135FS+0	1	17,443.16	1	18,884.56	-	-
138	137FS+0	1	17,443.16	1	18,884.56	-	-
139	136FS+0	1	17,443.16	1	18,884.56	-	-
140	139FS+0	1	17,443.16	1	18,884.56	-	-
141	138FS+0	7	49,553.17	6	53,793.56	-	-
142	140FS+0	7	49,553.17	6	53,793.56	-	-
143	141FS+0	1	17,443.16	1	18,884.56	-	-
144	143FS+0	1	17,443.16	1	18,884.56	-	-
145	142FS+0	1	17,443.16	1	18,884.56	-	-
146	145FS+0	1	17,443.16	1	18,884.56	-	-
147	144FS+0	7	49,553.17	6	53,793.56	-	-
148	146FS+0	7	49,553.17	6	53,793.56	-	-
149	147FS+0	1	17,443.16	1	18,884.56	-	-
150	149FS+0	1	17,443.16	1	18,884.56	-	-
151	148FS+0	1	17,443.16	1	18,884.56	-	-
152	151FS+0	1	17,443.16	1	18,884.56	-	-
153	150FS+0	7	49,553.17	6	53,793.56	-	-
154	152FS+0	7	49,553.17	6	53,793.56	-	-
155	153FS+0	1	17,443.16	1	18,884.56	-	-
156	155FS+0	1	17,443.16	1	18,884.56	-	-
157	154FS+0	1	17,443.16	1	18,884.56	-	-
158	157FS+0	1	17,443.16	1	18,884.56	-	-
159	156FS+0	7	49,557.17	6	53,635.81	-	-
160	158FS+0	7	49,557.17	6	53,635.81	-	-
161	160FS+0	2	29,723.98	2	32,075.50	-	-
162	161FS+0	2	8,579.19	2	9,312.42	-	-
163	1FS+0	12	100.00	-	-	-	-
164	152FS+0	3	21,912.00	2	38,557.21	2	42,183.38

Table 4.23 Activity Table for the 447-activity Case Project (continued)

Activity	Pred.	Mode 1		Mode 2		Mode 3	
		Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $
165	164FS+0	3	21,912.00	2	38,557.21	2	42,183.38
166	165FS+0	3	21,912.00	2	38,557.21	2	42,183.38
167	166FS+0	3	21,912.00	2	38,557.21	2	42,183.38
168	167FS+0	3	21,912.00	2	38,557.21	2	42,183.38
169	168FS+0	3	21,912.00	2	38,557.21	2	42,183.38
170	169FS+0	3	21,912.00	2	38,557.21	2	42,183.38
171	170FS+0	3	21,912.00	2	38,557.21	2	42,183.38
172	171FS+0	3	21,912.00	2	38,557.21	2	42,183.38
173	172FS+0	3	21,912.00	2	38,557.21	2	42,183.38
174	173FS+0	3	21,912.00	2	38,557.21	2	42,183.38
175	174FS+0	3	21,912.00	3	38,557.21	2	42,183.38
176	175FS+0	3	21,912.00	3	38,557.21	2	42,183.38
177	176FS+0	3	21,912.00	3	38,557.21	2	42,183.38
178	177FS+0	3	21,912.00	3	38,557.21	2	42,183.38
179	178FS+0	3	21,912.00	3	38,557.21	2	42,183.38
180	179FS+0	3	21,912.00	3	38,557.21	2	42,183.38
181	180FS+0	3	21,912.00	3	38,557.21	3	42,183.38
182	181FS+0	3	21,912.00	3	38,557.21	3	42,183.38
183	182FS+0	3	21,912.00	3	38,557.21	3	42,183.38
184	183FS+0	3	21,912.00	3	38,557.21	3	42,183.38
185	184FS+0	3	21,912.00	3	38,557.21	3	42,183.38
186	185FS+0	3	21,912.00	3	38,557.21	3	42,183.38
187	186FS+0	3	21,912.00	3	38,557.21	3	42,183.38
188	187FS+0	3	21,912.00	3	38,557.21	3	42,183.38
189	162FS+25	2	2,448.00	1	2,000.00	2	3,500.00
190	1FS+0	12	100.00	-	-	-	-
191	171FS+0	4	24,690.00	4	19,057.98	4	18,521.41
192	191FS+0	4	24,690.00	4	19,057.98	4	18,521.41
193	192FS+0	4	24,690.00	4	19,057.98	4	18,521.41
194	193FS+0	4	24,690.00	4	19,057.98	4	18,521.41
195	194FS+0	4	24,690.00	4	19,057.98	4	18,521.41
196	195FS+0	4	24,690.00	4	19,057.98	4	18,521.41
197	196FS+0	4	24,690.00	4	19,057.98	4	18,521.41
198	197FS+0	4	24,690.00	4	19,057.98	4	18,521.41
199	198FS+0	4	24,690.00	4	19,057.98	4	18,521.41
200	199FS+0	4	24,690.00	4	19,057.98	4	18,521.41
201	200FS+0	4	24,690.00	4	19,057.98	4	18,521.41
202	201FS+0	4	24,690.00	5	19,057.98	4	18,521.41
203	202FS+0	4	24,690.00	5	19,057.98	4	18,521.41
204	203FS+0	4	24,690.00	5	19,057.98	4	18,521.41
205	204FS+0	4	24,690.00	5	19,057.98	4	18,521.41

Table 4.23 Activity Table for the 447-activity Case Project (continued)

Activity	Pred.	Mode 1		Mode 2		Mode 3	
		Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $
206	205FS+0	4	24,690.00	5	19,057.98	4	18,521.41
207	206FS+0	4	24,690.00	5	19,057.98	4	18,521.41
208	207FS+0	4	24,690.00	5	19,057.98	6	18,521.41
209	208FS+0	4	24,690.00	5	19,057.98	6	18,521.41
210	209FS+0	4	24,690.00	5	19,057.98	6	18,521.41
211	210FS+0	4	24,690.00	5	19,057.98	6	18,521.41
212	211FS+0	4	24,690.00	5	19,057.98	6	18,521.41
213	212FS+0	4	24,690.00	5	19,057.98	6	18,521.41
214	213FS+0	4	24,690.00	5	19,057.98	6	18,521.41
215	214FS+0	4	24,690.00	5	19,057.98	6	18,521.41
216	189FS+0	2	3,000.00	3	4,000.00	4	3,000.00
217	1FS+0	27	100.00	-	-	-	-
218	192FS+0	4	7,769.00	5	7,109.60	6	7,043.60
219	218FS+0	4	7,769.00	5	7,109.60	6	7,043.60
220	219FS+0	4	7,769.00	5	7,109.60	6	7,043.60
221	220FS+0	4	7,769.00	5	7,109.60	6	7,043.60
222	221FS+0	4	7,769.00	5	7,109.60	6	7,043.60
223	222FS+0	4	7,769.00	5	7,109.60	6	7,043.60
224	223FS+0	4	7,769.00	5	7,109.60	6	7,043.60
225	224FS+0	4	7,769.00	5	7,109.60	6	7,043.60
226	225FS+0	4	7,769.00	5	7,109.60	6	7,043.60
227	226FS+0	4	7,769.00	5	7,109.60	6	7,043.60
228	227FS+0	4	7,769.00	5	7,109.60	6	7,043.60
229	228FS+0	4	7,769.00	5	7,109.60	6	7,043.60
230	229FS+0	4	7,769.00	5	7,109.60	6	7,043.60
231	230FS+0	4	7,769.00	5	7,109.60	6	7,043.60
232	231FS+0	4	7,769.00	5	7,109.60	6	7,043.60
233	232FS+0	4	7,769.00	5	7,109.60	6	7,043.60
234	233FS+0	4	7,769.00	6	7,109.60	6	7,043.60
235	234FS+0	4	7,769.00	6	7,109.60	6	7,043.60
236	235FS+0	4	7,769.00	6	7,109.60	6	7,043.60
237	236FS+0	4	7,769.00	6	7,109.60	6	7,043.60
238	237FS+0	4	7,769.00	6	7,109.60	6	7,043.60
239	238FS+0	4	7,769.00	6	7,109.60	6	7,043.60
240	239FS+0	4	7,769.00	6	7,109.60	6	7,043.60
241	240FS+0	4	7,769.00	6	7,109.60	6	7,043.60
242	241FS+0	4	7,769.00	6	7,109.60	6	7,043.60
243	242FS+0	1	799.00	1	700.00	2	600.00
244	1FS+0	1	100.00	-	-	-	-
245	222FS+0	4	31,525.94	6	22,905.12	5	24,574.32

Table 4.23 Activity Table for the 447-activity Case Project (continued)

Activity	Pred.	Mode 1		Mode 2		Mode 3	
		Dur. (Day)	**Cost $**	**Dur. (Day)**	**Cost $**	**Dur. (Day)**	**Cost $**
246	245FS+0	4	31,525.94	6	22,905.12	5	24,574.32
247	246FS+0	4	31,525.94	6	22,905.12	5	24,574.32
248	247FS+0	4	31,525.94	6	22,905.12	5	24,574.32
249	248FS+0	4	31,525.94	6	22,905.12	5	24,574.32
250	249FS+0	4	31,525.94	6	22,905.12	5	24,574.32
251	250FS+0	4	31,525.94	6	22,905.12	5	24,574.32
252	251FS+0	4	31,525.94	6	22,905.12	5	24,574.32
253	252FS+0	4	31,525.94	6	22,905.12	5	24,574.32
254	253FS+0	4	31,525.94	6	22,905.12	5	24,574.32
255	254FS+0	4	31,525.94	6	22,905.12	5	24,574.32
256	255FS+0	4	31,525.94	6	22,905.12	5	24,574.32
257	256FS+0	4	31,525.94	6	22,905.12	5	24,574.32
258	257FS+0	4	31,525.94	6	22,905.12	6	24,574.32
259	258FS+0	4	31,525.94	6	22,905.12	6	24,574.32
260	259FS+0	4	31,525.94	6	22,905.12	6	24,574.32
261	260FS+0	4	31,525.94	6	22,905.12	6	24,574.32
262	261FS+0	4	31,525.94	6	22,905.12	6	24,574.32
263	262FS+0	4	31,525.94	7	22,905.12	6	24,574.32
264	263FS+0	4	31,525.94	7	22,905.12	6	24,574.32
265	264FS+0	4	31,525.94	7	22,905.12	6	24,574.32
266	265FS+0	4	31,525.94	7	22,905.12	6	24,574.32
267	266FS+0	4	31,525.94	7	22,905.12	6	24,574.32
268	267FS+0	4	31,525.94	7	22,905.12	6	24,574.32
269	268FS+0	4	31,525.94	7	22,905.12	6	24,574.32
270	269FS+0	1	21,105.59	4	15,000.00	3	11,000.00
271	242FS+0	10	52,000.00	25	46,800.00	-	-
272	242FS+0	12	67,079.25	-	-	-	-
273	1FS+0	12	100.00	-	-	-	-
274	272FS-10	4	7,578.24	7	6,690.00	9	5,920.00
275	274FS+0	4	7,578.24	7	6,690.00	9	5,920.00
276	275FS+0	4	7,578.24	7	6,690.00	9	5,920.00
277	276FS+0	4	7,578.24	7	6,690.00	9	5,920.00
278	277FS+0	4	7,578.24	7	6,690.00	9	5,920.00
279	278FS+0	4	7,578.24	7	6,690.00	9	5,920.00
280	279FS+0	4	7,578.24	7	6,690.00	9	5,920.00
281	280FS+0	4	7,578.24	7	6,690.00	9	5,920.00
282	281FS+0	4	7,578.24	7	6,690.00	9	5,920.00
283	282FS+0	4	7,578.24	8	6,690.00	9	5,920.00
284	283FS+0	4	7,578.24	8	6,690.00	9	5,920.00
285	284FS+0	4	7,578.24	8	6,690.00	9	5,920.00

Table 4.23 Activity Table for the 447-activity Case Project (continued)

Activity	Pred.	Mode 1		Mode 2		Mode 3	
		Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $
286	285FS+0	4	7,578.24	8	6,690.00	9	5,920.00
287	286FS+0	4	7,578.24	8	6,690.00	9	5,920.00
288	287FS+0	4	7,578.24	8	6,690.00	9	5,920.00
289	288FS+0	4	7,578.24	8	6,690.00	9	5,920.00
290	289FS+0	4	7,578.24	8	6,690.00	9	5,920.00
291	290FS+0	4	7,578.24	9	6,690.00	9	5,920.00
292	291FS+0	4	7,578.24	9	6,690.00	9	5,920.00
293	292FS+0	4	7,578.24	9	6,690.00	9	5,920.00
294	293FS+0	4	7,578.24	9	6,690.00	9	5,920.00
295	294FS+0	4	7,578.24	9	6,690.00	9	5,920.00
296	295FS+0	4	7,578.24	9	6,690.00	9	5,920.00
297	296FS+0	4	7,578.24	9	6,690.00	9	5,920.00
298	297FS+0	4	7,578.24	9	6,690.00	9	5,920.00
299	298FS+0	2	1,136.82	4	1,000.00	6	2,000.00
300	1FS+0	12	100.00	-	-	-	-
301	231FS+0	2	3,305.10	1	4,265.00	-	-
302	301FS+0	2	3,305.10	1	4,265.00	-	-
303	302FS+0	2	3,305.10	1	4,265.00	-	-
304	303FS+0	2	3,305.10	1	4,265.00	-	-
305	304FS+0	2	3,305.10	1	4,265.00	-	-
306	305FS+0	2	3,305.10	1	4,265.00	-	-
307	306FS+0	2	3,305.10	1	4,265.00	-	-
308	307FS+0	2	3,305.10	1	4,265.00	-	-
309	308FS+0	2	3,305.10	1	4,265.00	-	-
310	309FS+0	2	3,305.10	1	4,265.00	-	-
311	310FS+0	2	3,305.10	1	4,265.00	-	-
312	311FS+0	2	3,305.10	1	4,265.00	-	-
313	312FS+0	2	3,305.10	1	4,265.00	-	-
314	313FS+0	2	3,305.10	1	4,265.00	-	-
315	314FS+0	2	3,305.10	1	4,265.00	-	-
316	315FS+0	2	3,305.10	1	4,265.00	-	-
317	316FS+0	2	3,305.10	1	4,265.00	-	-
318	317FS+0	2	3,305.10	2	4,265.00	-	-
319	318FS+0	2	3,305.10	2	4,265.00	-	-
320	319FS+0	2	3,305.10	2	4,265.00	-	-
321	320FS+0	2	3,305.10	2	4,265.00	-	-
322	321FS+0	2	3,305.10	2	4,265.00	-	-
323	322FS+0	2	3,305.10	2	4,265.00	-	-
324	323FS+0	2	3,305.10	2	4,265.00	-	-
325	324FS+0	2	3,305.10	2	4,265.00	-	-

Table 4.23 Activity Table for the 447-activity Case Project (continued)

Activity	Pred.	Mode 1		Mode 2		Mode 3	
		Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $
326	325FS+0	1	100.00	-	-	-	-
327	293FS+0	1	9,000.00	2	7,944.00	-	-
328	327FS+0	1	8,540.00	2	7,944.00	-	-
329	328FS+0	1	10,440.00	2	7,944.00	-	-
330	329FS+0	1	10,440.00	2	7,944.00	-	-
331	330FS+0	1	10,440.00	2	7,944.00	-	-
332	331FS+0	1	10,440.00	2	7,944.00	-	-
333	332FS+0	1	10,440.00	2	7,944.00	-	-
334	333FS+0	1	10,440.00	2	7,944.00	-	-
335	334FS+0	1	10,440.00	2	7,944.00	-	-
336	335FS+0	1	10,440.00	2	7,944.00	-	-
337	336FS+0	1	10,440.00	2	7,944.00	-	-
338	337FS+0	1	10,440.00	2	7,944.00	-	-
339	338FS+0	1	10,440.00	2	7,944.00	-	-
340	339FS+0	1	10,440.00	2	7,944.00	-	-
341	340FS+0	1	10,440.00	2	7,944.00	-	-
342	341FS+0	1	10,440.00	2	7,944.00	-	-
343	342FS+0	1	10,440.00	2	7,944.00	-	-
344	343FS+0	1	10,440.00	2	7,944.00	-	-
345	344FS+0	1	10,440.00	2	7,944.00	-	-
346	345FS+0	1	10,440.00	2	7,944.00	-	-
347	346FS+0	1	10,440.00	2	7,944.00	-	-
348	347FS+0	1	10,440.00	2	7,944.00	-	-
349	348FS+0	1	10,440.00	2	7,944.00	-	-
350	349FS+0	1	10,440.00	2	7,944.00	-	-
351	350FS+0	1	10,440.00	2	7,944.00	-	-
352	329FS+0	3	4,917.00	2	6,820.00	-	-
353	352FS+0	3	4,917.00	2	6,820.00	-	-
354	353FS+0	3	4,917.00	2	6,820.00	-	-
355	354FS+0	3	4,917.00	2	6,820.00	-	-
356	355FS+0	3	4,917.00	2	6,820.00	-	-
357	356SF+0	3	4,917.00	2	6,820.00	-	-
358	357FS+0	3	4,917.00	2	6,820.00	-	-
359	358FS+0	3	4,917.00	2	6,820.00	-	-
360	359FS+0	3	4,917.00	2	6,820.00	-	-
361	360FS+0	3	4,917.00	2	6,820.00	-	-
362	361FS+0	3	4,917.00	2	6,820.00	-	-
363	362FS+0	3	4,917.00	2	6,820.00	-	-
364	363FS+0	3	4,917.00	2	6,820.00	-	-
365	364FS+0	3	4,917.00	2	6,820.00	-	-

Table 4.23 Activity Table for the 447-activity Case Project (continued)

Activity	Pred.	Mode 1		Mode 2		Mode 3	
		Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $
366	365FS+0	3	4,917.00	3	6,820.00	-	-
367	366FS+0	3	4,917.00	3	6,820.00	-	-
368	367FS+0	3	4,917.00	3	6,820.00	-	-
369	368FS+0	3	4,917.00	3	6,820.00	-	-
370	369FS+0	3	4,917.00	3	6,820.00	-	-
371	370FS+0	3	4,917.00	3	6,820.00	-	-
372	371FS+0	3	4,917.00	3	6,820.00	-	-
373	372FS+0	3	4,917.00	3	6,820.00	-	-
374	373FS+0	3	4,917.00	3	6,820.00	-	-
375	374FS+0	3	4,917.00	3	6,820.00	-	-
376	375FS+0	3	4,917.00	3	6,820.00	-	-
377	292FS+0	1	60,000.00	4	45,000.00	-	-
378	377FS+0	1	60,000.00	4	45,000.00	-	-
379	378FS+0	1	40,000.00	4	33,280.00	-	-
380	379FS+0	1	40,000.00	4	33,280.00	-	-
381	380FS+0	1	40,000.00	4	33,280.00	-	-
382	381FS+0	1	40,000.00	4	33,280.00	-	-
383	382FS+0	1	40,000.00	4	33,280.00	-	-
384	383FS+0	1	40,000.00	4	33,280.00	-	-
385	384FS+0	1	40,000.00	4	33,280.00	-	-
386	385FS+0	1	40,000.00	4	33,280.00	-	-
387	386FS+0	1	40,000.00	4	33,280.00	-	-
388	387FS+0	1	40,000.00	4	33,280.00	-	-
389	388FS+0	1	40,000.00	4	33,280.00	-	-
390	389FS+0	1	40,000.00	4	33,280.00	-	-
391	390FS+0	1	40,000.00	4	33,280.00	-	-
392	391FS+0	1	40,000.00	4	33,280.00	-	-
393	392FS+0	1	40,000.00	4	33,280.00	-	-
394	393FS+0	1	40,000.00	4	33,280.00	-	-
395	394FS+0	1	40,000.00	4	33,280.00	-	-
396	395FS+0	1	40,000.00	4	33,280.00	-	-
397	396FS+0	1	40,000.00	4	33,280.00	-	-
398	397FS+0	1	40,000.00	4	33,280.00	-	-
399	398FS+0	1	40,000.00	4	33,280.00	-	-
400	399FS+0	1	40,000.00	4	33,280.00	-	-
401	400FS+0	1	40,000.00	4	33,280.00	-	-
402	345FS+0	9	15,000.00	-	-	-	-
403	216FS+0	15	470,000.00	45	196,667.28	-	-
404	403SS+0	7	60,000.00	-	-	-	-
405	403FS+0	1	9,601.94	2	10,050.00	1	8,040.00

Table 4.23 Activity Table for the 447-activity Case Project (continued)

Activity	Pred.	Mode 1		Mode 2		Mode 3	
		Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $
406	405FS+7	2	14,120.50	5	7,825.00	-	-
407	406FS+0	1	5,648.20	-	-	-	-
408	407FS+0	1	7,907.48	1	8,040.00	-	-
409	408FS+0	10	21,463.16	7	29,900.00	-	-
410	402FS+0	12	198,360.00	20	85,138.00	-	-
411	402FS+0	12	34,048.00	20	27,050.00	-	-
412	410FS-3	6	306,976.00	25	247,500.00	-	-
413	412FS-3	9	36,890.00	20	33,800.00	26	30,600.00
414	413FS+18	32	101,395.00	27	123,692.00	21	153,883.00
415	414FS+0	25	440,800.00	23	431,630.00	21	432,925.00
416	415FS+15	75	1,094,100.00	90	977,500.00	130	935,000.00
417	416FS+0	8	29,176.00	-	-	-	-
418	415FS+0	50	440,800.00	45	589,952.00	35	661,100.00
419	418FS+0	5	25,000.00	-	-	-	-
420	240FS+0	9	201,600.00	8	290,200.00	-	-
421	412FS+10	4	10,400.00	-	-	-	-
422	271FS+10	7	22,880.00	-	-	-	-
423	412FS+10	10	49,920.00	7	53,000.00	9	41,600.00
424	184FS+0	70	131,807.41	60	151,459.76	130	112,703.67
425	184FS+0	70	1,176,372.33	130	951,914.72	160	888,013.39
426	184FS+0	70	650,466.67	60	744,274.02	130	553,826.40
427	184FS+0	70	27,333.33	60	31,352.54	130	23,329.84
428	184FS+0	70	199,033.33	60	253,930.00	-	-
429	184FS+0	70	64,400.00	90	52,700.00	-	-
430	184FS+0	70	16,200.00	-	-	-	-
431	184FS+0	70	300,000.00	130	248,570.27	60	372,895.56
432	184FS+0	70	16,666.67	130	13,809.45	60	20,716.42
433	184FS+0	70	23,333.33	130	19,333.24	60	29,002.98
434	184FS+0	70	33,333.33	130	27,618.91	60	41,432.84
435	184FS+0	70	40,000.00	130	33,142.70	60	49,719.40
436	184FS+0	70	10,000.00	130	8,285.67	60	12,429.85
437	184FS+0	70	28,333.33	130	23,476.80	60	35,217.91
438	4FS+0	43	10,000.00	70	9,200.00	35	12,429.85
439	441FS-10	1	10,000.00	1	7,371.34	1	12,429.85
440	184FS+0	70	300,000.00	130	248,570.67	60	372,895.56
441	184FS+0	70	240,000.00	130	198,856.22	60	298,316.45
442	184FS+0	70	83,333.33	130	69,047.29	60	103,582.10
443	184FS+0	70	600,000.00	130	497,140.55	60	745,791.13
444	184FS+0	70	33,333.33	130	27,618.91	60	41,432.84
445	1FS+0	20	41,666.67	-	-	-	-

Table 4.23 Activity Table for the 447-activity Case Project (continued)

Activity	Pred.	Mode 1		Mode 2		Mode 3	
		Dur. (Day)	Cost $	Dur. (Day)	Cost $	Dur. (Day)	Cost $
446	1FS+0	110	25,000.00	-	-	-	-
447	3FS+0, 48FS+0, 159FS+0, 163FS+0, 188FS+0, 190FS+0, 215FS+0, 217FS+0, 243FS+0, 244FS+0, 270FS+0, 273FS+0, 299FS+0, 300FS+0, 326FS+0, 351FS+0, 376FS+0, 401FS+0, 404FS+0, 409FS+0, 411FS+0, 417FS+0, 419FS+0, 420FS+0, 421FS+0, 422FS+0, 423FS+0, 424FS+0, 425FS+0, 426FS+0, 427FS+0, 428FS+0, 429FS+0, 430FS+0, 431FS+0, 432FS+0, 433FS+0, 434FS+0, 435FS+0, 436FS+0, 437FS+0, 438FS+0, 439FS+0, 440FS+0, 442FS+0, 443FS+0, 444FS+0, 445FS+0, 446FS+0	0	0	-	-	-	-

As mentioned earlier, total cost of any construction project consists of direct and indirect costs. As shown in Table 4.24, the actual direct costs of different phases of the case project add up to $17,498,342. Besides, the principal items considered within the indirect cost of the project include the personnel, security, and administration costs. For the actual case project, the indirect cost is calculated by collecting all the costs spent on management, accommodation, security, utility and energy (electricity, water, gasoline, etc.), and other administrative expenses (Table 4.25). By dividing the total indirect cost (i.e., $812,590) by the total project duration (i.e., 684 days), the daily indirect cost is calculated as $1,188/day. In addition, as per the contract clauses, a delay penalty of $50,000/day applies if the project completion duration exceeds 684 days.

Table 4.24 Total Direct Cost for the 447-activity Case Project

Housing Real Estate Project – Sulaymaniyah / Iraq **Total Area per each Building = 25,000 m²**			
Major Project Phases	**Total Direct Cost ($)**	**Start Date**	**Finish Date**
Construction Works	5,853,926.063	12/04/2012	08/02/2014
Finishing Works	7,583,803.530	30/08/2012	17/05/2014
Mechanical Works	2,265,613.073	17/02/2013	12/04/2014
Electrical Works	1,728,333.333	20/04/2012	12/04/2014
Mobilization	41,666	30/06/2012	27/10/2012
Other Work Items	25,000	30/06/2012	17/05/2014
Grand Total	**17,498,342**	**12/04/2012**	**17/05/2014**

Table 4.25 Total Indirect Cost for the 447-activity Case Project

Item	**Indirect Cost**
Engineering Staff	$12,850/month
Company Lawyer	$5,500/month
Administration	$7,925/month
Laboratory Staff	$2,850/month
Electricity, Water, Gasoline, etc.	$6,205/month
Total Monthly Indirect Cost	$35,330/month
Total Indirect Cost for 684 Days	$812,590
Daily Indirect Cost	$1,188/day

4.3.5.2 Real-life Project Analysis

Bid evaluation is undoubtedly one of the major phases of the tendering cycle during which decision-makers, generally, by relying on their experiences make careful decisions. These decisions involve making choices among the competing bid offers with different time-cost implications. The significance of time-cost trade-off analysis and the effectiveness of the proposed optimization algorithm is studied herein by using the data archived for the actual construction project. HGA's parameters for the case project are configured subsequent to numerous experiments with various combinations of values and the best setting is achieved as given in Table 4.26.

Table 4.26 Parameter Values for the 447-activity Case Project

Description	Value
Population Size	3,000
Boltzmann Constant	1.2
Generation Number	500
Pc (0.6-1)	0.80
P_m (0.005-0.05)	1⁄no.Gene (447) ≈ 0.005

As summarized in Table 4.26, a population size of 3,000 with an iteration number of 500 are used. Boltzmann's Constant of 1.2 was experienced to suffice as higher values for this parameter improves exploration capabilities by promoting the selection of bad mutations. A crossover probability of 0.80 is used while the mutation probability is clamped to the feasible range and set as 0.005 since Eq. (3.9) results in a value lower that the allowable limit.

The results of the experiments for the case project are abstracted in Table 4.27. In the second column of this table the number of successive runs is given. Under 'Best Solution' column, the best solutions obtained is presented. The last column tabulates the average processing time as 16 minutes.

Table 4.27 Results Achieved for the 447-activity Case Project by HGA

Instance	No. of Runs	Conditions				Best Solution		Avg. CPU Time (Min:S)
		C_d (\$)	C_c (\$)	C_n (\$)	T_d (Day)	Total Cost (\$)	Total Duration (Day)	
Case Project	10	1,188	0	50,000	684	16,539,149	684	16:00

It is concluded that for the time-constrained actual construction project, HGA is able to come up with a solution with a total cost lower than the amount actually spent on the project. As given in Table 4.27, HGA locates a solution with a total cost of \$16,539,149 which is \$1,771,783 lower than the actual cost of the project. In other terms, a hefty saving of almost 10% would have been possible should an optimization model such as HGA had been used by the decision makers earlier. Meanwhile, by applying the same technique to all the six buildings of this housing real estate project, the owner could have saved \$10,630,710 for a project with an estimated budget of \$110,000,000.

Table 4.28 shows the chromosome structure of the best solution found by HGA for the actual case project. Separated by commas, the values in the string designate the mode selections for each of the 447 activities.

Table 4.28 Chromosome Structure of the Best Solution for 447-activity Case Project

Selected Time-Cost Modes for Activities 1 to 447
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 3, 2, 2, 2, 3, 3, 3, 3, 2, 3, 2, 2, 2, 2, 3, 3, 3, 3, 2, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 3, 3, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 3, 2, 1, 1, 1, 2, 2, 1, 1, 3, 3, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 4, 3, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1

Figure 4.21 displays the minimum total cost achieved by HGA per each iteration for the real-life case project. As shown in this chart, in the early stages of the optimization the probability of accepting badly mutated chromosomes is higher, then, it gradually decreases. This trend is more evident after the 200th iteration where more exploitation is applied rather than exploration as less badly mutated chromosomes are allowed to be accepted randomly.

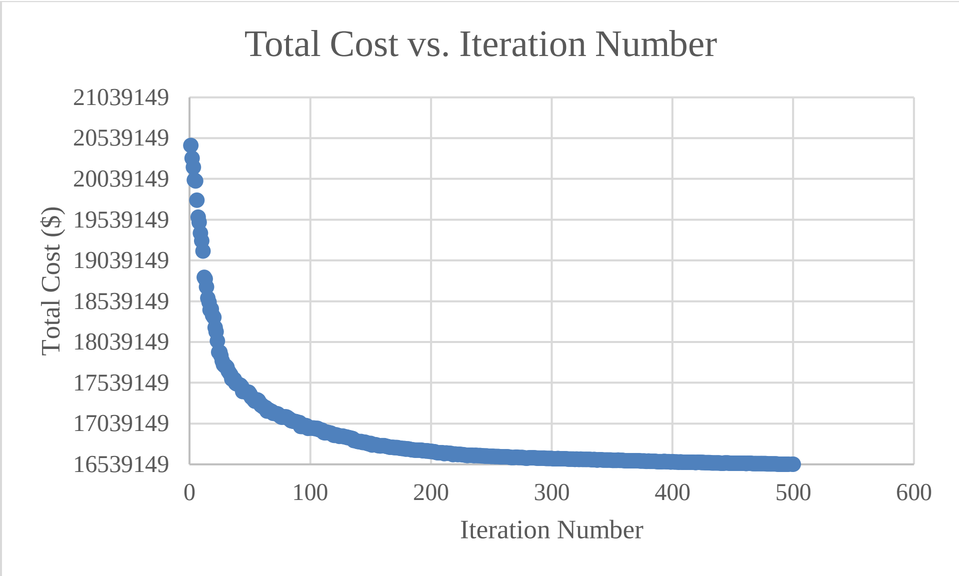

Figure 4.21 Total Cost versus Iteration Number for the 447-activity Case Project

CHAPTER 5

CONCLUSIONS

This thesis study presents a hybrid Simulated Annealing-based Genetic Algorithm (HGA) for solving an optimization problem referred to as the discrete time-cost trade-off problem (DTCTP) in the construction context. It is observed there exist only a few studies that have exercised problems incorporating the generalized logical relationships with realistic overlapping among the activities. The optimization method proposed herein is designed to provide capability of exerting DTCTPs with properly overlapped activities that incorporate generalized precedence relations.

Since classical genetic algorithms tend to require large number of iterations for achieving adequate results, and that they are susceptible to convergence to local optima, a complementary computation method is embedded to the proposed GA-based method. As the complementary method, Simulated Annealing (SA), a probabilistic meta-heuristic algorithm, is often used for the optimization problems. Wherefore, in this study, too, SA is applied to improve the convergence capabilities of GA by rejecting mutations leading to worse solutions as GA reaches larger iteration numbers. In the reproduction process of the proposed hybrid GA, a roulette wheel selection method is implemented to give greater chance to the survival of the fitter individual. As a result of this method, low-fit chromosomes are discarded, promoting the need for population size calibrations. Thus, the population size is calibrated by sorting all the chromosomes – reproduced before and after crossover operation – in descending order with respect to fitness values and then a population of equal size to parent is formed by selecting the fitter first chromosomes.

The proposed HGA which is coded in C# environment is able to solve two different extensions of DTCTP, namely, the cost-minimization and the time-constraint problems. For both these extensions, problems with different complexities are practiced including eleven benchmark problems widely used in the literature, twelve large-scale problems obtained from the recent literature, and an actual construction project introduced as a part of this study. Parameters of the proposed method are configured for each problem individually following several trials with respect to solution quality and convergence speed of HGA. Of the parameters, mutation probability is set automatically as the multiplicative inverse of the number of activities. For the crossover probability, a constant value leading to an improved performance is used across all the instances. Due to the inherent randomness of the proposed HGA model, it is ran for ten consecutive times for each of the test problems, thereby, the average values are reported. HGA is shown to perform with a satisfactory to exceptional degree of accuracy within acceptable processing time over the practiced instances.

During the performance evaluations, HGA's results are compared to solutions of other researchers and, depending on their availability, to the optimal solutions as well. HGA is discovered to be able to obtain the optimum solutions for all of the ten experimental trials for the small-scale 18-activity and 29-activity problems. Besides, the proposed algorithm proves to be more efficient than the existing methods since it converges to the global optima faster. The results achieved for 290-activity problems further highlight the capabilities of the proposed HGA in solving DTCTPs. It is shown to be able to find high-quality solutions with meager deviations from the optima for such complex problems that associate several different logical relationships. HGA is also fitted to 63-activity and 630-activity problems, outperforming most of the existing optimization methods which confirm the capability of HGA in providing satisfactory results with slim deviations for these instances. Large-scale problems including 990 activities are also fed into HGA. It is noted that HGA slightly misses the global optima of these instances by finding solutions marginally lower in quality than those reported in the literature. Nevertheless, solid results obtained by HGA even for such real-life-size problems further substantiate practicality of this model.

Sequel to the performance evaluations of HGA over the problems available in the literature, an actual construction project is also used to demonstrate the real-life applicability and effectiveness of the proposed method. The real project is solved by HGA and the obtained time-cost alternative selections are compared to the experienced-based decisions made by the experts during the planning stage. Quality of the solution obtained by HGA for the real project firmly suggests that this technique can come to the aid of decision makers and project managers in proper selection of different construction methods. It is realized that for the time-constrained actual construction project, HGA is able to come up with a solution with a much lower total cost than what was actually spent on the project. It is concluded that a saving of almost 10% would have been possible should an optimization model such as HGA had been used by the decision makers earlier, especially during the planning stage.

Implementation of different types of instances affirm HGA's sound performance for small-, medium-, and large-scale problems. Though, there remains room for further development of the proposed algorithm. For instance, parameter self-configuration can be studied for automatic selection of population size and generation number. Proposed HGA can also be extended for solution of truly multi-objective Pareto front time-cost trade-off problems. Improving the convergence capabilities of HGA by inclusion of a fast heuristic as an initial population generator seems another promising area.

REFERENCES

[1] J.W. Fondahl, *A non-computer approach to the critical path method for the construction industry* . California: The construction institute,1962,pp.5-85.

[2] N.Siemens, (1970, May), "A Simple CPM TimeCost Tradeoff Algorithm." *Management science* ,[Online],17(6),pp.B354-B363.Available:https://doi.org/10.1287/mnsc.17.6.B354 [Feb. 1,1971]

[3] C.W. Feng, L. Liu, and S.A. Burns, (1996, Sep.), "Using genetic algorithms to solve construction time-cost trade-off problems." *Journal of Computing in Civil Engineering,* [Online], 11(3),pp.184–189.Available: https://doi.org/10.1061/(ASCE)0887-3801(1997)11:3(184) [July 01,1997].

[4] I. Yang, (2006, Dec.)" Using elitist particle swarm optimization to facilitate bicriterion time-cost trade-off analysis." *Journal of construction engineering and management* [Online], 133(7), pp. 498–505.Available: https://doi.org/10.1061/(ASCE)0733-9364(2007)133:7(498),[July 01,2007].

[5] R. Sonmez, Ö.H. Bettemir, (2012, April), "A hybrid genetic algorithm for the discrete time–cost trade-off problem." *Expert Systems with Applications*, [Online],39(13), pp.11428–11434.Available: https://doi.org/10.1016/j.eswa.2012.04.019 [Apirl 10,2012]

[6] S. Aminbakhsh and R. Sonmez, (2015, Dec.) "Discrete particle swarm optimization method for the large-scale discrete time–cost trade-off problem." *Expert Systems with Applications*, [Online], 51, pp. 177–185.Available: https://doi.org/10.1016/j.eswa.2015.12.041 [Jan. 06,2016].

[7] D. Agdas, D.J. Warne, , J. Osio-norgaard, F.J. Masters,(2017, May), "Utility of Genetic Algorithms for Solving Large-Scale Construction Time-Cost Trade-Off Problems." *Journal of Computing in Civil Engineering,* [Online], 32(1), pp.1–10.Available: https://doi.org/10.1061/(ASCE)CP.1943-5487.0000718[Oct. 31,2017].

[8] E. Elbeltagi, T. Hegazy, and D.Grierson,(2005, Jan.), "Comparison among five evolutionary-based optimization algorithms." *Advanced engineering informatics,* [Online], 19(1), pp. 43–53.Available: https://doi.org/10.1016/j.aei.2005.01.004 [July 16,2005].

[9] S. A. Burns, L. Liu, and C. Feng, (1995, Dec.)"The LP/IP hybrid method for construction time-cost trade-off analysis." *Construction Management & Economics,* [Online], 14(3), pp.265-276.Available: http://dx.doi.org/10.1080/014461996373511[Oct. 21,2010]

[10] E. Demeulemeester, B. Reyck, B. Foubert, W. Herroelen, M. Vanhoucke,(1998, July), "New computational results on the discrete time/cost trade-off problem in project networks." *Journal of the operational research society,* [Online], 49(11) , pp.1153–1163.Available: https://doi.org/10.1057/palgrave.jors.2600634 [Dec. 20,2017].

[11] O. Moselhi,(1993)"Schedule compression using the direct stiffness method." *Canadian Journal of Civil Engineering* [Online], 20(1), pp.65-72. Available: https://doi.org/10.1139/l93-007[Feb. 1993].

[12] S.T .Ng , Y.Zhang, (2007, Nov.), "Optimizing Construction Time and Cost Using Ant Colony Optimization Approach." *Journal of construction engineering and management,* [Online],134(9),pp.721–728.Available: https://doi.org/10.1061/(ASCE)0733-9364(2008)134:9(721) [Sep. 01,2008]

[13] H. Zhang and F. Xing, (2010, July) "Fuzzy-multi-objective particle swarm optimization for time–cost–quality tradeoff in construction." *Automation in Construction,* [Online], 19(8), pp. 1067–1075.Available: https://doi.org/10.1016/j.autcon.2010.07.014 [Aug. 30,2010].

[14] D. Arditi, P. Sikangwan, O.B. Tokdemir, (2002, Feb.), "Scheduling system for high rise building construction" *Construction Management & Economics,* [Online], 20(7), pp. 353–364. Available: https://doi.org/10.1080/01446190210131647[Oct. 21, 2010].

[15] E. Bjarnason, “Critical Success Factors for Planning , Scheduling and Control in Design and Construction” M.A. thesis, University of Reykjavík, Iceland,2015.

[16] Z. Hejducki, (2003, Oct.), “Scheduling model of construction activity with time couplings” *Journal of Civil Engineering and Management,* [Online], 9(4), pp. 284–291. Available: https://doi.org/10.1080/13923730.2003.10531341[Jul. 26, 2012].

[17] Y.A. Olawale, M. Sun, (2010, Feb.), “Cost and time control of construction projects: Inhibiting factors and mitigating measures in practice” *Construction Management & Economics*, [Online], 28(5), pp. 509–526. Available: https://doi.org/10.1080/01446191003674519[June 15, 2010].

[18] H. Kerzner, *Project management : a systems approach to planning, scheduling, and controlling (11th edition).* Hoboken, NJ: John Wiley & Sons, Inc.,2013,pp.597-506.

[19] V. Sireesha, N.R. Shankar, (2010), “A New Approach to find Total Float time and Critical Path in a fuzzy Project Network” *International Journal of Engineering science and technology,* [Online], 2(4), pp. 600–609. Available:https://pdfs.semanticscholar.org/1a24/1dc35f8682c56b8416ec3157700ff4aaa8f3.pdf [2010].

[20] P.D. Galloway, (2006), “Survey of the construction industry relative to the use of CPM scheduling for construction projects” *Journal of Civil Engineering and Management ,* [Online],132(7), pp. 697–71. Available:https://doi.org/10.1061/(ASCE)0733-9364(2006)132:7(697)[July , 01,2006].

[21] M.J. Liberatore, B. Pollack-Johnson, and C.A. Smith, (2000, Aug.), “Project management in construction: Software use and research directions” *Journal of Civil Engineering and Management,* [Online],127(2), pp.101–107. Available: https://doi.org/10.1061/(ASCE)0733-9364(2001)127:2(101) [April 01, 2001].

[22] T. Hegazy, W. Menesi, (2010, March), “Critical Path Segments Scheduling Technique” *Journal of Civil Engineering and Management,* [Online], 136(10),pp. 1078–1085. Availabe https://doi.org/10.1061/(asce)co.1943-7862.0000212 [Sep. 15, 2010].

[23] Y. Wong, (1964, Oct.), “Critical Path Analysis for New Product Planning. Direct.” *Journal of Marketing,* [Online], 28(4), pp. 53–59. Available: https://www.jstor.org/stable/1249571 [Oct. 1964].

[24] W.S. Han , A.M. Yusof, S. Ismail, and N.C. Aun, (2011, Sep.), “Reviewing the notions of construction project success.” *International Journal of Business and Management,* [Online],7(1), pp. 90–101. Available: https://doi.org/10.5539/ijbm.v7n1p90[Jan. 01,2012].

[25] E. Howsawi, D. Eager, R. Bagia, (2014, May), “The four-level project success framework : application and assessment.” *Organizational Project Management,* [Online],1(1) pp.1-14.Avialablehttps://doi.org/10.5130/opm.v1i1.3865 [May 13, 2014].

[26] F. Sigur, “Critical Success Factors in Project Management : An ethical perspective.” M.A. thesis, University of Iceland, Iceland,2009.

[27] A. de Wit (1988, Jan.), “Measurement of project success” *International journal of project management,* [Online]. 6(3), pp.164-170.Available: https://doi.org/10.1016/0263-7863(88)90043-9 [April. 13, 2002].

[28] T. Cooke-Davies, (2002, April), “The “real” success factors on projects. ” *International journal of project management,*[Online]. 20(3), pp.185-190.Available: https://doi.org/10.1016/S0263-7863(01)00067-9[April 01,2002]

[29] G.A. Silva, B.N.F. Warnakulasooriya, and B. Arachchige, (2016, Dec.), “Criteria for Construction Project Success : A Literature Review” *13th International Conference on Business Management,* [Online].Available: https://doi.org/10.2139/ssrn.2910305[Feb. 02, 2017].

[30] J.R. Turner(2005, June), “The project manager’s leadership style as a success factor on projects: A literature review.” *Project management journal,* [Online], 36(2), pp. 49–61.Available: https://doi.org/10.1177/875697280503600206 [June 01, 2005].

[31] N. Azhar, , R.U. Farooqui, and S. Ahmed, (2008, Aug.), “Cost overrun factors in construction industry of Pakistan.” *First International Conference on Construction In Developing Countries, Advancing and Integrating Construction Education, Research & Practice,* [Online], pp. 499–508. Available: http://www.neduet.edu.pk/Civil/ICCIDC-I/Complete Proceedings.rar#page=510[Aug. 05, 2008].

[32] D. Lock, *Project management(6th edition).* Aldershot , England: Gower publishing,1996,pp.522-522.

[33] D.W. Chan, M.M. Kumaraswamy,(2001, Oct.), “Compressing construction durations : lessons learned from Hong Kong building projects.” *International journal of project management,* [Online], 20(1) ,pp. 23-35. Available: https://doi.org/10.1016/S0263-7863(00)00032-6 [Jan. , 2002].

[34] R.N. Nkado,(1994, July), “Construction time-influencing factors : the contractor ‘ s perspective” *Construction Management and Economics,* [Online], 13(1), pp. 81-89. Available: https://doi.org/10.1080/01446199500000009 [July 28, 2006].

[35] D. Lock, *Project management(9th edition).* Cherry Street , USA: Gower publishing , 2007 , pp. 520-545.

[36] J.B. Ebbesen, , A.J. Hope,(2013, March), “Re-imagining the Iron Triangle : Embedding Sustainability into Project Constraints” *PM World Journal,* [Online], 2(3) , pp. 1–13. Available: http://nrl.northumbria.ac.uk/id/eprint/11311 [March 04, 2013].

[37] R.L. Kliem, I.S. Ludin, and K.L. Robertson, *Project management methodology: A practical guide for the next millenium.* Broken Sound Parkway, NW: CRC Press ,1997, pp.11-272.

[38] L. Liu, S.A. Burns, and C.W. Feng, (1995, Dec.), "Construction time-cost trade-off analysis using LP/IP hybrid method." *Journal of construction engineering and management* [Online], 121(4), pp.446–454.Available: https://doi.org/10.1061/(ASCE)0733-9364(1995)121:4(446) [Dec. 01, 1995].

[39] I. Choudhury, S.S. Rajan, (2003), "Time-cost relationship for residential construction in Texas" *CIB REPORT,* [Online],284, pp.73.Available: https://itc.scix.net/pdfs/w78-2003-73.content.pdf [2003].

[40] C.W. Feng, L. Liu, , S.A. Burns, (1998, Oct.), "Stochastic construction time-cost trade-off analysis." *Journal of Computing in Civil Engineering,* [Online], 14(2) , pp.117–126. Available: https://doi.org/10.1061/(ASCE)0887-3801(2000)14:2(117) [April 01, 2000].

[41] M. Vanhoucke, D. Debels, (2007, Aug.), "The discrete time / cost trade-off problem : extensions and heuristic procedures." *Journal of Scheduling,* [Online], 10(4-5), pp. 311–326. Available: https://doi.org/10.1007/s10951-007-0031-y[Aug. 07, 2007].

[42] R. Reda, R.I. Carr, (1989, Sep.), "Time-cost trade-off among related activities." *Journal of Construction Engineering and Management,* [Online], 115(3), pp. 475–486.Available: https://doi.org/10.1061/(ASCE)0733-9364(1989)115:3(475) [Sep. 01, 1989].

[43] F. Smarandache,: *Collected Papers , V Florentin Smarandache*. clos du Parnasse 1000, Brussels: EuropaNova asbl, 2014,pp.260-334.

[44] E.I. John, A.S. Abdullateef, and A.O. Abdulganiyu, (2015), "A Study of Time and Cost Relationship of Private Building Projects in Abuja." *International Journal of Construction Engineering and Management,* [Online], 4(1) , pp.26–34.Available: https://doi.org/10.5923/j.ijcem.20150401.03 [2015].

[45] T. Hegazy, (1999), "Optimization of construction time–cost trade-off analysis using genetic algorithms." *Canadian Journal of Civil Engineering,* [Online],26(6),pp.685-697.Available: https://doi.org/10.1139/l99-031[1999].

[46] D. Barkovic, J. Jukic, and I. Blazevic, (2017, Jan.), "Cost analysis and project duration shortening in network planning technique with Pert/cost method." *Under the auspices of the President of the Republic of Croatia,* [Online], pp.36-50.Available: https://bib.irb.hr/datoteka/879480.349724720-Interdisciplinary-Management-Research-XIII.pdf#page=37 [Jan. 01,2017].

[47] E. Eshtehardian, A. Afshar, R. Abbasnia, (2009, Feb.), "Fuzzy-based MOGA approach to stochastic time – cost trade-off problem." *Automation in construction* ,[Online], 18(5), pp.692–701:Available: https://doi.org/10.1016/j.autcon.2009.02.001[Aug. 01,2009].

[48] J. Zhou, P.E. Love, X. Wang, K.L. Teo, Z. Irani, (2012, Nov.), "A review of methods and algorithms for optimizing construction scheduling." *Journal of the Operational Research Society,* [Online], 64(80), pp.1091–1105.Available: https://doi.org/10.1057/jors.2012.174 [Dec. 21,2017].

[49] J. E. Kelley, (1961, June), "Critical-path planning and scheduling: Mathematical basis." *Operations research* [Online], 9(3), pp.296-320.Available: https://doi.org/10.1287/opre.9.3.296 [June 01,1961].

[50] C. Hendrickson, C.T. Hendrickson, T. Au, *Project management for construction: Fundamental concepts for owners, engineers, architects, and builders* H. Cross, Engineering and Ivory Towers, New York: Chris Hendrickson.1989.

[51] W.L Myere, L.R Shaffer (1963), "Extensions of the critical path method through the application of integer programming" *Civil Engineering and Construction,* [Online], (2).

[52] J.H. Patterson, W.D. Huber, (1974, Feb), "A horizon-varying, zero-one approach to project scheduling." *Management Science* , [Online],20(6), pp. 990–998.Available: https://doi.org/10.1287/mnsc.20.6.990 [Feb. 01,1974].

[53] D. R. Robinson, (1975, Oct.)" A dynamic programming solution to cost-time tradeoff for CPM." *Management Science* , [Online], 22(2), pp. 158–166.Avialable: https://doi.org/10.1287/mnsc.22.2.158 [Oct. 02,1975].

[54] S. E. Elmaghraby, (1993, Jan.) “Resource allocation via dynamic programming in activity networks,” *European Journal of Operational Research,* [Online], 64(2), pp. 199–215.Available: https://doi.org/10.1016/0377-2217(93)90177-O [Jan. 13,2011].

[55] P. De, E.J Dunne, J. B. Ghosh, , C.E. Wells, (1995, March)“The discrete time-cost tradeoff problem revisited,” *European journal of operational research,* [Online],81(2), pp.225-238.Available: https://doi.org/10.1016/0377-2217(94)00187-H[Jan. 13,2011].

[56] S.M. Alavipour, “Project scheduling using optimized financing.” Doctor of Philosophy thesis, Illinois Institute of Technology, USA, 2017.

[57] W. Prager,(1962, July) “A structural method of computing project cost polygons.” *Management Science,* [Online], 9(3),pp.394-404.Available: https://doi.org/10.1287/mnsc.9.3.394[April 01,1963]

[58] R. Sonmez, S. Aminbakhsh, and T. Atan,(2020, Feb.) “Activity Uncrashing Heuristic with Noncritical Activity Rescheduling Method for the Discrete Time-Cost Trade-Off Problem,” *Journal of Construction Engineering and Management,* [Online], 146(8), pp. 1–14.Available: https://doi.org/10.1061/(ASCE)CO.1943-7862.0001870[May 18,2020].

[59] S.S. Leu, A.T. Chen, C.H. Yang,(1999, May), “A GA-based fuzzy optimal model for construction time-cost trade-off.” *International Journal of Project Management,* [Online], 19(1),pp. 47–58. Available: https://doi.org/10.1016/S0263-7863(99)00035-6[Oct. 27,2000]

[60] K. P. Anagnostopoulos , L. Kotsikas,(2010, Sep.) “Experimental evaluation of simulated annealing algorithms for the time–cost trade-off problem.” *Applied Mathematics and Computation* , [Online],217(1), pp. 260–270.Available: https://doi.org/10.1016/j.amc.2010.05.056 [May, 23,2010].

[61] I. Aydogdu “Optimum design of 3-d irregular steel frames using ant colony optimization and harmony search algorithms” Doctor of Philosophy thesis, Middle East Technical University , Turkey, 2010.

[62] Y. Sawaragi, H. Nakayma, and T.Tanino . *Theory of Multiobjective Optimization*. Florida, USA: Academic Press INC. ALL , 1985 , pp.1-2.

[63] J. Moussourakis, C. Haksever, (2009, July), "Project Compression with Nonlinear Cost Functions." *Journal of Construction Engineering and Management* , [Online],136(2),pp. 251–259:Available: https://doi.org/10.1061/(ASCE)CO.1943-7862.0000123[Feb. 01 ,2010].

[64] A. Sapkauskiene and S. Leitoniene,(2010, March) "The concept of time-based competition in the context of management theory." *Engineering Economics,* [Online],21(2).Available:http://www.inzeko.ktu.lt/index.php/EE/article/view/11700 [March 18, 2010].

[65] H.Yang, Y.Chen,(1997, Sep.), "Finding the critical path in an activity network with time-switch constraints." *European Journal of Operational Research,* [Online], 120(3),pp.603–613.Available: https://doi.org/10.1016/S0377-2217(98)00390-7 [Jan. 13, 2011].

[66] M. Vanhoucke, (2003, March), "New computational results for the discrete time / cost trade-off problem with time-switch constraints." *European Journal of Operational Research,* [Online],165(2), pp.359–374. Available: https://doi.org/10.1016/j.ejor.2004.04.007 [Jan. 13,2011].

[67] M. Vanhoucke, E. Demeulemeester, W. Herroelen, (2001, Dec.), "Discrete time / cost trade-offs in project scheduling with time-switch constraints." *Journal of the Operational Research Society,* [Online],53(7), pp.741–751.Available: https://doi.org/10.1057/palgrave.jors.2601351 [Dec. 21,2017].

[68] A.P. Chassiakos, A.M. Asce, , S.P. Sakellaropoulos, (2004, Dec.), "Time-Cost Optimization of Construction Projects with Generalized Activity Constraints." *Journal of Construction Engineering and Management,* [Online],131(10), pp.1115–1124. Available: https://doi.org/10.1061/(ASCE)0733-9364(2005)131:10(1115) [Oct. 01,2005]

[69] R. Sonmez, Ö.H. Bettemir, (2012, April), “A hybrid genetic algorithm for the discrete time–cost trade-off problem.” *Expert Systems with Applications,* [Online],39(13), pp.11428–11434.Available: https://doi.org/10.1016/j.eswa.2012.04.019 [Apirl 10,2012]

[70] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* MIT press. USA, 1992.

[71] D.X. Zheng , S.T. Ng, M.M.Kumaraswamy, (2002, Dec.), “Applying a Genetic Algorithm-Based Multiobjective Approach for Time-Cost Optimization.” *Journal of Construction Engineering and management,* [Online],130(2),pp.168–176.Available: https://doi.org/10.1061/(ASCE)0733-9364(2004)130:2(168)[March 15,2004]

[72] D.X. Zheng , S.T. Ng, M.M.Kumaraswamy, (2004, Jan.), “Applying Pareto Ranking and Niche Formation to Genetic Algorithm-Based Multiobjective Time – Cost Optimization.” *Journal of Construction Engineering and Management,* [Online],131(1),pp. 81–92.Available:https://doi.org/10.1061/(ASCE)0733-9364(2005)131:1(81)[Jan. 01,2005]

[73] E. Eshtehardian, A. Afshar, R. Abbasnia, (2008, March), “Time–cost optimization: using GA and fuzzy sets theory for uncertainties in cost.” *Construction Management and Economics*,[Online],26(7),pp.37–41.Available: https://doi.org/10.1080/01446190802036128[June 28,2010]

[74] L. A. Zadeh,(1965, June) “Fuzzy sets. ” *Information and control,* [Online],8(3), pp.338-353.Available: https://doi.org/10.1016/S0019-9958(65)90241-X [Nov. 29,2004]

[75] Ö.H. Bettemir “Optimization of time-cost-resource trade-off problems in project scheduling using meta-heuristic algorithms.”, PhD. thesis, Middle east technical university,Turkey, 2009

[76] G. Albayrak and Ö. İlker,(2017, Feb.) “A state of art review on metaheuristic methods in time-cost trade-off problems.” *International Journal of Structural and Civil Engineering,* [Online], 6(1), pp. 30–34, 2017.Available: http://www.ijscer.com/uploadfile/2017/0216/20170216021710303.pdf[Feb. 01,2017].

[77] M. Dorigo, A. Colorni, V. Maniezzo, (1991) “Distributed optimization by ant colonies.” *Proceeding from the First European Conference on Artificial Life* [1994].

[78] A. Afshar , F. Sharifi, (2009, March)“Non-dominated archiving multi-colony ant algorithm for multi-objective optimization: Application to multi-purpose reservoir operation.” *Engineering Optimization,* [Online],41(4), pp.313-325.Available: https://doi.org/10.1080/03052150802460414[March 30,2009].

[79] A. Afshar, A.K. Ziaraty, A. Kaveh, F. Sharifi, (2009, Feb.), “Nondominated Archiving Multicolony Ant Algorithm in Time–Cost Trade-Off Optimization. ” *Journal of Construction Engineering and Management,* [Online],135(7),pp.668–674. Available: https://doi.org/10.1061/(ASCE)0733-9364(2009)135:7(668)[June 15, 2009]

[80] Y. Zhang, and T. Ng, (2012, Sep.)“ An ant colony system based decision support system for construction time-cost optimization.” *Journal of Civil Engineering and Management,* [Online], 18(4) , pp. 580-589.Available: https://doi.org/10.3846/13923730.2012.704164[Sep. 11,2012].

[81] E. Elbeltagi, T. Hegazy, D. Grierson, (2005, May), “A modified shuffled frog-leaping optimization algorithm: applications to project management.” *Structure and Infrastructure Engineering,* [Online], 3(1),pp.53-60.Available: https://doi.org/10.1080/15732470500254535 [Feb. 16,2007].

[82] R. Eberhart, and J. Kennedy, (1995, Oct.), “A new optimizer using particle swarm theory.” *In MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science,* [Online], pp. 39-43.Available: 10.1109/MHS.1995.494215 [Oct. 06,1995].

[83] Y. Shi , R. Eberhart, (1998, May), "A modified particle swarm optimizer." In *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360),*[Online] pp. 69–73.Available: 10.1109/ICEC.1998.699146[May 01,1998].

[84] Y. Xiong and Y. Kuang, (2007, Aug.) "Applying an ant colony optimization algorithm-based multiobjective approach for time–cost trade-off." *Journal of Construction Engineering and Management,* [Online],134(2), pp.153-156.Available: https://doi.org/10.1061/(ASCE)0733-9364(2008)134:2(153)[Feb. 01,2008].

[85] S. Aminbakhsh, "Hybrid particle swarm optimization algorithm for obtaining Pareto front of discrete time-cost trade-off problem" M.A. thesis, Middle east technical university, Turkey, 2013.

[86] Lee, H.G., Yi, C.Y., Lee, D.E. and Arditi, D., (2015, June), "An advanced stochastic time-cost tradeoff analysis based on a CPM-guided genetic algorithm." *Computer-Aided Civil and Infrastructure Engineering*, [Online], 130(10), pp.824-842.Available: https://doi.org/10.1111/mice.12148 [June 15, 2015].

[87] S.G. Kim, (2012, Aug.), "CPM Schedule Summarizing Function of the Beeline Diagramming Method.", [Online], *Journal of Asian Architecture and Building Engineering,* 11(2), pp. 367–374. Available: https://doi.org/10.3130/jaabe.11.367[Aug. 20, 2012].

[88] J.K. Parker, A.R. Khoogar, D.E. Goldberg, (1989, May) "Inverse Kinematics of Redundant Robots using Genetic Algorithms." *International Conference on Robotics and Automation,* [Online], (1), pp. 271-276.Available: DOI**:** 10.1109/ROBOT.1989.100000 [Aug. 06,2002].

[89] M.P. Poland, C.D. Nugent, H. Wang, L. Chen, (2012), "Genetic algorithm and pure random search for exosensor distribution optimisation." *International Journal of Bio-Inspired Computation,* [Online], 4(6), pp.359-372.Avilable: https://doi.org/10.1504/IJBIC.2012.051408[Jan. 15,2013].

[90] A. Azaron, C. Perkgoz, and M. Sakawa, (2004, Dec.) “A genetic algorithm approach for the time-cost trade-off in PERT networks.” *Applied mathematics and computation,* [Online] 168(2), pp.1317-1339.Available: https://doi.org/10.1016/j.amc.2004.10.021 [Dec. 02,2004]

[91] S. Tong,(2013) “Roulette Wheel Selection Game Player” *Mathematics, Statistics, and Computer Science Honors Projects,* [Online],30. Available: https://digitalcommons.macalester.edu/cgi/viewcontent.cgi?article=1032&context=mathcs_honors [2013].

[92] D. Beasley, D.R. Bull, R.R. Martin, (1993), “An Overview of Genetic Algorithms : Part 1 , Fundamentals.” *University computing,* [Online],15(2),pp.55-59.Available: http://orca.cf.ac.uk/id/eprint/64436[June 04,2017].

[93] I. Babaoglu, O. Findik, E. Ülker, (2009, April), “A comparison of feature selection models utilizing binary particle swarm optimization and genetic algorithm in determining coronary artery disease using support vector machine.” *Expert Systems with Applications,* [Online], 37(4),pp. 3177–3183.Available: https://doi.org/10.1016/j.eswa.2009.09.064[Sep. 26,2009]

[94] M. Srinivas, L.M. Patnaik, (1994, April) “Adaptive probabilities of crossover and mutation in genetic algorithms” *IEEE Transactions on Systems, Man, and Cybernetics,*[Online], 24(4),pp. 656–667.Available: doi: 10.1109/21.286385. [April 1994].

[95] D. Adler,(1993, April) “Genetic algorithms and simulated annealing: A marriage proposal.” *IEEE International Conference on Neural Networks,* [Online], (2), pp. 1104–1109. Available: https://doi.org/10.1109/ICNN.1993.298712[Aug. 06,2002].

[96] C.R. Reeves,(1995), “A genetic algorithm for flowshop sequencing.” *Computers & operations research,* [Online], 22(1),pp. 5–13.Available: https://doi.org/10.1016/0305-0548(93)E0014-K[Jan. 20,2000]..

[97] J.S. Tumuluru, R. McCulloch, (2015, July), "A new hybrid genetic algorithm for optimizing the single and multivariate objective functions." *ASABE Annual International Meeting,* [Online], (3), pp.2233–2247.Available: https://doi.org/10.13031/aim.20152188606[July 26,2015].

[98] L.Wang, D.Z. Zheng, (1999, Sep.) "An effective hybrid optimization strategy for job-shop scheduling problems." *Computers & Operations Research,* [Online], 28(6), pp.585-596.Available: https://doi.org/10.1016/S0305-0548(99)00137-9 [Jan. 10,2001].

[99] H. Eskandar, A. Sadollah, ,A. Bahreininejad, M. Hamdi,(2012, July), "Water cycle algorithm - A novel metaheuristic optimization method for solving constrained engineering optimization problems." *Computers & Structures,* [Online], 110-111, pp.151–166.Available https://doi.org/10.1016/j.compstruc.2012.07.010[Aug. 26,2012].

[100] M. Safe, J. Carballido, I. Ponzoni, N. Brignole,(2004), "On Stopping Criteria for Genetic Algorithms." *In Brazilian Symposium on Artificial Intelligence,* [Online], 3171 ,pp. 405–413.Available: https://doi.org/10.1007/978-3-540-28645-5_41[2004].

[101] R. Kolisch, S. Hartmann,(2006, Oct.), "Experimental investigation of heuristics for resource-constrained project scheduling : An update." *European journal of operational research,* [Online], 174(1), pp.23–37.Available: https://doi.org/10.1016/j.ejor.2005.01.065[Jan. 13,2011].

[102] S. Aminbakhsh, "Heuristic and exact methods for the large-scale discrete time-cost trade-off problems." PhD. thesis, Middle east technical university, Turkey, 2018.

[103] J.F. Gonçalves, J.J. Mendes, M.G. Resende, (2008, Sep.), "A genetic algorithm for the resource constrained multi-project scheduling problem." *European journal of operational research,* [Online], 189(3), pp.1171–1190.Available: https://doi.org/10.1016/j.ejor.2006.06.074[Jan. 13,2011].

[104] M. A. Eirgash, “Pareto-front performance of multiobjective teaching learning based optimization algorithm on time-cost trade off optimization problems.” M.A. thesis, Karadeniz Technical university, Turkey, 2018