Scientia Iranica (2024) 31(19), 1809–1824

Sharif University of Technology
Scientia Iranica
Transactions A: Civil Engineering
https://scientiairanica.sharif.edu

Optimization-based scheduling of construction projects with generalized precedence relationships: A real-life case study

Saman Aminbakhsh[a*] and Ary Ahmed[b]

a. *Department of Civil Engineering, Atilim University, Ankara, 06830, Turkey.*
b. *Graduate School of Natural and Applied Sciences, Atilim University, Ankara 06830, Turkey.*

Received 1 December 2021; received in revised form 13 January 2023; accepted 29 August 2023

KEYWORDS
Optimization;
Time-cost trade-off problem;
Generalized precedence relationships;
Genetic algorithm;
Meta-heuristics.

Abstract. Concomitant reduction of cost and duration is recognized as one of the main aspects of construction planning. Expedition of project schedule naturally incurs extra costs due to implementation of more productive and/or high-price construction techniques. Meanwhile, a reduction in time is usually plausible only down to a certain limit, below which renders expeditions either technically or financially unviable. Thus, striking a reasonable balance between project cost and duration remains a desirable yet challenging task for which there has been a myriad of advancements and literature. Despite the many studies associated with this problem-referred to as Time-Cost Trade-off Problem (TCTP) it is observed that only a few exercise TCTPs with the generalized logical relationships. This observation holds despite the fact that generalized precedence relationships are imperative to introduce parallelism and to secure a realistic overlap among the activities. In this regard, a Simulated Annealing-based (SA-based) Genetic Algorithm (GA) as proposed herein, is specifically designed to provide the capability of exerting TCTPs with properly overlapped activities. Efficiency of this algorithm is tested over a range of problems and its performance is validated over a large-scale real-case construction project. Results of the hybridized GA indicate fast and robust convergence to high-quality solutions.

1. Introduction

Indisputably, timely completion, reasonable cost, and high level of stakeholder satisfaction constitute the main objectives of almost any project. In the present time, competition among construction companies mainly boils down to unique schedule proposals with reasonable profit amounts. It is a common desire for the project parties to finish the project as efficiently as possible, chiefly with the aim of remobilizing resources and gaining favorable financial advantages. As most of

*. *Corresponding author.*
E-mail addresses: saman.aminbakhsh@atilim.edu.tr (S. Aminbakhsh); aryh.faraj@yahoo.com (A. Ahmed)

To cite this article:
S. Aminbakhsh and A. Ahmed "Optimization-based scheduling of construction projects with generalized precedence relationships: A real-life case study", *Scientia Iranica*, (2024) **31**(19), pp. 1809–1824
https://doi.org/10.24200/sci.2023.59493.6275

the project activities can be executed using more than one construction method or more rapidly by allocating extra resources, more than one duration and cost combination known as alternative might be available for each activity. Those in charge of making decisions desire to expedite the project with minimum extra expenses by consuming the network slack times as well as finding the best range of alternatives for performing the project activities. This is leveraged through finding the best equilibrium between the direct and indirect costs of a project, since, a decrease in project duration would naturally lead to a decline in the project cost but only until reaching a particular point. Thereupon, the additional direct costs invested for acceleration will start to surpass the amount of indirect cost saved due to duration reduction.

Striking a proper balance between the project cost and duration is a complicated and computationally challenging problem and even moreso for large-scale real-life projects. Large number of alternatives coupled with large number of activities make finding an optimal schedule a very complex process [1]. This problem, dubbed as Time-Cost Trade-off Problem (TCTP), is well recognized in the extant literature and has been studied under different presumptions. The different types of time-cost relationships identified in the literature include linear [2–5], concave [6], convex [7], and discrete [8,9]. Several optimization techniques have been proposed towards addressing this well-established problem [4,10–14].

TCTP research area essentially emerged in and around 1960s with introduction of project network analysis techniques by Kelley and Walker [2], Fulkerson [3], and Kelley [15]. Ever since, with the advancements of computer science and technology, successive progress has been made in this domain and different methods have been introduced. Techniques proposed for solution of TCTP mainly include exact methods, heuristics, and meta-heuristics. Traditionally TCTP has been modelled by mathematical programming such as linear programming [15], dynamic programming [16], hybrid linear/integer programming [17], branch-and-bound algorithm [18], integer programming [19], mixed integer programming [20], mixed integer nonlinear programming [21] and network reduction approaches [22]. The literature on heuristics for DTCTP is limited to the methods proposed by Fondahl [23], Siemens [4], Goyal [5], Moselhi [24], Bettemir and Birgonul [25] and Sonmez et al. [26]. Meta-heuristics practiced for TCTP primarily include Genetic Algorithm (GA) [9,10], Ant Colony Optimization (ACO) [27], Particle Swarm Optimization (PSO) [28], and Symbiotic Organisms Search (SOS) [29].

It is noteworthy that despite a large body of the literature has hitherto been dedicated to development of optimization methods for TCTP, only a few can be extended to real-life applications. This largely stems from the fact that the majority of the existing methods lack the ability to efficiently tackle real-scale TCTPs with the generalized logical relationships. On the other hand, those capable of capturing the complex generalized logic are frequently experimented using small-scale problems and their practicality have not been tested on large-scale instances. In fact, the 14-activity problem of Klanšek and Pšunder [30], the 29-activity problem of Sakellaropoulos and Chassiakos [31], the 54-activity of Cajzek and Klanšek [32], and the 63-activity problem of Sonmez and Bettermir [11] are among the problems most widely used to demonstrate applicability of the methods proposed in the current literature.

Arguably, generalized logical relationships play a pivotal role in securing a realistic overlapping among the project activities and planners usually overlap activities by applying Finish-to-Start (FS), Start-to-Start (SS), Finish-to-Finish (FF) and sometimes Start-to-Finish (SF) types of relationship as well as by introducing the concept of lag time into the schedule. Nevertheless, the state-of-the-art studies concentrating on this subject matter are not many and have mainly exploited mixed integer/linear programming [20,31], mixed integer nonlinear programming [21,32], and meta-heuristics [11,33]. As aforesaid, most of the past research has focused on relatively small instances that barely reflect the complexity of actual projects. The largest problem with various activity relationships is employed by Sonmez and Bettemir [11] that includes 290 activities. Though, this problem is based on a small network and is generated by copying the base problem in serial several times and thus may not truly represent a complex project. Last but not least, among the few research studies that present meta-heuristics for projects with various activity dependencies, the majority do not report the CPU times and that the obtained solutions deviate from the optima by not so meager amounts.

This study aims at this research gap by contributing to both researchers and professionals by presenting an optimization algorithm capable of unraveling large-scale projects with any type of precedence relationship. To this end, this paper presents a practical Simulated Annealing-based (SA-based) GA that achieves high-quality solutions by requiring modest processing times. This paper also enables improved performance evaluation of state-of-the-art approaches by introducing a multi-mode real-case time-cost trade-off construction management problem. The results of the proposed method are compared with the solutions obtained by the existing approaches in terms of quality of the results and computation time. The quality of the results is determined by obtaining the optimal solutions from the literature. In the following sections the details of the methodology implemented for developing the proposed

GA-based approach as well as the procedure performed during computational experiments and performance evaluations are presented.

This paper is organized as follows. Section 2 outlines the optimization problem formulation. In Section 3, the proposed GA-based optimization procedure is described. Section 4 presents the numerical examples and the associated results. Finally, Section 5 summaries the concluding remarks on the present study.

2. Problem description

The objective of general TCTP is to minimize the total cost, comprised of direct and indirect costs of a project by determining the optimal combination of the available time-cost alternatives.

2.1. Assumptions

It is assumed that the indirect costs vary linearly with project duration. A discrete relationship is assumed between the time and cost of project activities. In addition, activity splitting is not allowed; that is, once started, an activity should be executed without interruptions.

2.2. Formulation

The proposed method is provided with the capability of unraveling problems that incorporate a particular desired completion time. In such problems, an upper bound is set for the duration of the project to reflect the completion deadline stipulated in the contract. Project parties often contractually agree to make/receive a certain payment on a daily basis either as a delay penalty in the case of delays, or as a bonus payment in the event of early completion. Resultantly, logical formulation of TCTP is extended to include the incentive payment as well as delay penalty and is implemented as follows:

$$\text{minimize } C = \sum_{i=1}^{n} dc_i + TC_d + \Delta T(x_{bon}C_{bon} + x_{pen}C_{pen}), \tag{1}$$

subject to:

$$T = \text{LF}_n, \tag{2}$$

$$\Delta T = T - T_d, \tag{3}$$

$$\begin{cases} \text{if} \quad \Delta T \leq 0 & x_{bon} = 1 \quad \text{and} \quad x_{pen} = 0 \\ \text{Otherwise,} & x_{bon} = 0 \quad \text{and} \quad x_{pen} = 1 \end{cases} \tag{4}$$

$$x_{bon}, x_{pen} \in \{0, 1\} \qquad \text{and} \qquad x_{bon} + x_{pen} = 1, \tag{5}$$

where; C = total project cost; dc_i = direct cost of ith activity; n = total number of activities including the start and finish milestones; T = total project duration; C_d = amount of daily indirect cost; T_d = desired completion time; C_{bon} = daily bonus payment; C_{pen} = daily delay penalty; x_{bon} and x_{pen} = cost component selection variables; and LF_n = late finish time of the last (nth) activity, i.e., finish milestone. It should be noted that Eq. (1) and thus the method presented in this study involves no steps to clamp the total penalty to a maximum amount.

The proposed method determines the activity times including LF_n used in Eq. (2) according to the following conditions:

$$\text{ES}_1 = 1, \tag{6}$$

$$Decision \begin{cases} \text{if} \quad rel = \text{FS} \\ \quad \text{ES}_{s(a)} = \max\{\text{EF}_i + l_{s(a)}\}; \\ \quad \text{LF}_{p(b)} = \min\{\text{LS}_i - l_{p(b)}\} \\ \\ \text{else if} \quad rel = \text{SS} \\ \quad \text{ES}_{s(a)} = \max\{\text{ES}_i + l_{s(a)}\}; \\ \quad \text{LF}_{p(b)} = \min\{\text{LS}_i + d_{p(b)} - l_{p(b)}\} \\ \\ \text{else if} \quad rel = \text{SF} \\ \quad \text{ES}_{s(a)} = \max\{\text{ES}_i - d_{s(a)} + l_{s(a)}\}; \\ \quad \text{LF}_{p(b)} = \min\{\text{LF}_i + d_{p(b)} - l_{p(b)}\} \\ \\ \text{else} \\ \quad \text{ES}_{s(a)} = \max\{\text{EF}_i - d_{s(a)} + l_{s(a)}\}; \\ \quad \text{LF}_{p(b)} = \min\{\text{LF}_i - l_{p(b)}\} \end{cases} \tag{7}$$

subject to:

$$\text{EF}_i = \text{ES}_i + d_i, \tag{8}$$

$$\text{LF}_n = \text{EF}_n, \tag{9}$$

$$\text{LS}_i = \text{LF}_i - d_i, \tag{10}$$

$$\forall i = \{1, ..., n\}; \quad \forall s(a) \in S(i); \quad \forall a = \{1, ..., |S(i)|\};$$

$$\forall p(b) \in P(i); \quad \forall b = \{1, ..., |P(i)|\}, \tag{11}$$

where; FS = Finish-to-Start relationship; SS=Start-to-Start relationship; SF = Start-to-Finish relationship; ES_1=early start of the first activity; $\text{ES}_{s(a)}$= early start of ath successor; ES_i=early start of ith activity; EF_i=early finish of ith activity; $\text{LF}_{p(b)}$=late finish of bth predecessor; LF_i=late finish of ith activity; LS_i=late start of ith activity; $l_s(a)$=lag time of ath successor; $l_{p(b)}$=lag time of bth predecessor; $d_{s(a)}$= duration of ath successor; $d_{p(b)}$ = duration of bth predecessor; d_i = duration of ith activity; s(a) = ath successor; $S(i)$ = set of immediate successors for i; p(b) = bth predecessor; and $P(i)$ = set of immediate predecessors for i.

3. SA-based Hybrid Genetic Algorithm (HGA) for discrete TCTP

In this section a background on GA ensues the description of the hybrid GA-based approach proposed for discrete TCTPs.

3.1. Genetic Algorithm (GA)

GA has been the subject of the well-established research domain of optimization ever since it was originally proposed by Holland [34] and has often been specifically tweaked for addressing various problems. For instance, Namazian et al. [35] combined goal programming and GA for project selection and scheduling; Akhbari [36] embedded GA to four other meta-heuristics to solve resource-constrained project scheduling problem; Erden et al. [37] presented a PSO with GA operators for solving integrated process planning dynamic scheduling and due date assignment problem; and Esmailnezhad and Saidi-Mehrabad [38] used GA for stochastic supply chain scheduling.

GA is the precursor of the nature inspired optimization algorithms that set a trend for later advancements. It was inspired by the natural evolution and the process of natural selection. Akin to more recent evolutionary algorithms, selection, crossover, and mutation are the main operators of GA. Chromosomes are used to portray solutions to problems by holding information about each design/decision variables, referred to as genes. GA evaluates the fitness of each individual in the population using a fitness (objective) function. For improving poor solutions, the best solutions are chosen randomly with a selection (e.g., roulette wheel) mechanism.

Albayrak and Ozdemir [39] by making a comprehensive review on meta-heuristic methods for TCTPs indicated that GA has kept its preference and continues to be practiced by applying some modifications and/or hybridization techniques. Numerous research studies implementing GAs for TCTPs suggest GA to be a potent and efficient procedure [9,13,40]; however, number of studies on large-scale TCTPs with general relationships is very limited if not null. Aiming at this point, this paper presents a GA-based method for the large-scale TCTPs that incorporate diverse activity relations. In the proposed GA approach, chromosomes will form the potential solutions to the problem while each gene will define the selected time-cost alternative for the corresponding activity in the chromosome sequence. Chromosomes of the initial population will be generated following a random scheme, among which the candidates to survive are identified using the roulette-wheel method.

3.2. GA for discrete TCTP

In real-life projects the time-cost relations are seldom continuous linear. Rather, execution alternatives are available in a discrete space. In addition, by utilizing discretization, any time-cost function can be estimated [41]. Due to these very reasons, GA-based approach presented in this paper is designed to unravel the discrete version of TCTP.

Initially, the members of the first population are generated by random assignment of time-cost alternatives. Logical sequencing of the activities for the generated chromosomes is established according to Eqs. (6)–(11). Thereby, the objective function for each chromosome is evaluated based on the obtained schedule and by using Eqs. (1)–(5). Thereafter, fitness is evaluated for the candidate solutions according to Eqs. (12)–(14):

$$M = \max\{C_1, ..., C_N\}, \tag{12}$$

$$f_j = M - C_j, \tag{13}$$

$$\forall j = \{1, ..., N\}, \tag{14}$$

where; C_j = objective function value (i.e., total project cost) of jth chromosome; M= largest objective function value among the population; and f_j = fitness function value of jth chromosome. As depicted before, the roulette wheel selection method is implemented to give a greater chance to the survival of the fitter individuals, i.e., those with lower objective function values using Eq. (15):

$$P_j = \frac{f_j}{\sum_{j=1}^{N} f_j}, \tag{15}$$

where; P_j = probability of selection of jth chromosome based on the roulette wheel selection technique. Expected count given in Eqs. (16) and (17) is evaluated which secures discarding the chromosomes with lower probability values and duplicating those with higher probabilities.

$$f_{avg.} = \frac{\sum_{j=1}^{N} f_j}{N}, \tag{16}$$

$$EC_j = \frac{f_j}{f_{avg.}}, \tag{17}$$

where; $f_{avg.}$ = the average fitness function value of the population; and EC_j = the expected count function of jth chromosome. Hereafter, reproduction starts by selecting the chromosomes with high fitness values and using expected counts for discarding low-fit chromosomes.

Of the fundamental operations for genetic reproduction, crossover operates between two randomly selected chromosomes. Crossover swaps genes between the selected individuals to produce a better offspring. Generally, crossover operation is not applied to all

pairs of selected individuals. The selection of pairs is made by assigning a random number, R_c, between [0,1] to each chromosome and then comparing it with the preset crossover probability value, P_c, which typically ranges between 0.60 and 1.0. Crossover operator kicks in only if R_c becomes smaller than or equal to P_c. In case the crossover is not applied, parents duplicate for producing offspring. In this step, another measure, Eq. (18), is taken to guarantee that only the worst chromosomes in the population get replaced by fitter reproduced chromosomes in the crossover operation. As a result of this controller, chromosomes produced through crossover will be discarded if they yield lower fitness values; otherwise, they will replace their parents.

$$\text{Crossover operation} \begin{cases} \text{if} \quad f_{j0} \leq f_{jc} & \text{Accept} \\ \text{Otherwse,} & \text{Reject} \end{cases} \tag{18}$$

where; f_{j0} = fitness function value of jth chromosome before crossover operation; and f_{jc} = fitness function value of jth chromosome after crossover operation. Subsequent to reproduction through crossover operation, the population might shrink due to low-fit chromosomes being discarded. In this study, population size will be calibrated after the crossover operator according to Eqs. (19) and (20).

$$\Delta N = N_{Parent} - N_{Offspring}, \tag{19}$$

Population size calibration

$$\begin{cases} \text{if} \quad \Delta N = 0 & \text{Dismiss} \\ \text{Otherwse,} & \text{Calibrate} \end{cases} \tag{20}$$

With respect to condition (20), chromosomes reproduced in the crossover operation will be used to make up for any deficiencies in the population size. For calibration, first, all the chromosomes-reproduced prior to and after crossover operation-will be sorted in descending order with respect to their fitness values and then the first N_{Parent} number of chromosomes will be preserved.

Following crossover, the mutation operation is performed on a gene-by-gene basis to promote the diversity of the population. The mutation rate, P_m, is considered to adjust the probability of a change in the values of genes in a chromosome. The mutation operation process is done by assigning a random number, R_m, between [0,1] to each gene in chromosomes and comparing these randomly assigned values with the predetermined mutation probability value. Should R_m of a gene be smaller than or equal to P_m, it 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 GA to suboptimal solutions. The value for this parameter can be achieved according to Eq. (21)-multiplicative inverse of the number of activities-while clamping the obtained results to the feasible range [0.005, 0.05] as recommended by Srinivas and Patnaik [42].

$$P_m = 1/n, \tag{21}$$

where; n = number of genes in a chromosome (i.e., number of activities).

3.3. SA-based Hybrid Genetic Algorithm (HGA)

As discussed earlier, GA is shown to successfully perform in many engineering fields as it has the potential to help decision-makers locate optimal solutions for complex large-scale problems. 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 and mutation operators. On the other hand, GAs might lose direction as their operators are susceptible to discarding useful solutions. Immature early convergence of GA mainly stems from this very characteristic.

It has also been observed that utilization of sole-GAs for optimization purposes may require high iteration counts for achieving adequate results. In this respect, some complementary computation methods can be applied to increase the convergence capabilities of GA for escaping the local optima which ultimately will lead to shortened computation time as well. In a pioneering study hereof, Bettemir [43] presented hybridized GAs with improved convergence capabilities for TCTP. Hybrid GAs were developed by integrating Simulated Annealing (SA) and Quantum SA (QSA) into GA which were shown to exhibit better performance while requiring less computation time. In a more recent study, Sonmez and Bettemir [11] performed optimization of discrete TCTP using an improved hybrid strategy –named HA –in which GA was combined with both SA and QSA. SA is a probabilistic meta-heuristic algorithm inspired by the cooling schedule of alloys subjected to tempering. In higher temperatures, molecules can move freely in any direction; as the alloy cools down, the movement of the molecules gets restricted depending on the temperature [44,45]. By taking advantage of the cooling schedule and local selection concept of SA method, an interaction can be established between SA and GA reproduction mechanism. A naturally parallel evaluation structure can be obtained by merging the foregoing algorithms. This will help shorten the computation time since SA will reject mutations leading to inferior

Table 1. Parameter configuration of HGA.

Parameter	Description	Factor level		Selected value	
		Low	High	Small-scale	Medium- to Large-scale
N	Population size	5n	3,000	5n	3,000
$t_{\max}$	Iteration number	$1.75n$	500	$1.75n$	500
β	Boltzmann constant	1.0	1.2	1.0	1.2
P_c	Crossover rate	0.8	1.0	0.8	0.8
P_m	Mutation rate	$1/n$ 'or' 0.005	$1/n$ 'or' 0.05	$1/n$	0.005

solutions as the algorithm progresses toward larger iteration numbers [11,46]. Such a hybrid approach maintains the generality of GA and can easily be used to solve various optimization problems [47]. In light of these points, the principles of SA are benefitted herein to develop a hybrid GA for solving multi-mode discrete TCTPs.

As a result, the GA-based hybrid strategy presented in this paper – hereafter referred to as HGA – accepts mutations leading to better offspring chromosomes. Acceptance or rejection of bad mutations, on the other hand, is regulated as per the condition given in Eqs. (22) and (23):

$$\Delta f_j = f_{jm} - f_{j0}, \tag{22}$$

Mutation operation

$$\begin{cases} \text{if} \quad R_\beta(j) \le e^{\frac{(\Delta f_j)\ t}{\beta\ f_{j0}}} & \text{Accept} \\ \text{Otherwise,} & \text{Reject} \end{cases} \tag{23}$$

where; f_{j0} = fitness function value of jth chromosome before mutation operation; f_{jm} = fitness function value of jth chromosome after mutation operation; R_β(j)= a random number between 0 and 1 assigned to jth chromosome; β = Boltzmann constant used to set the rate of cooling; and t = current iteration number. A flowchart is shown in Figure 1 to further elaborate the optimization procedure of HGA. As seen in the flowchart, per each chromosome the schedule process of HGA determines the activity times as well as the total project duration, T, according to the method discussed in Section 2.2.

4. Computational experiments

Computational experiments are conducted to validate the performance of the proposed HGA method using benchmark instances. A real-life construction project with over four hundred activities, with generalized precedence relationships is also used to demonstrate the real-life applicability and effectiveness of the developed method. The proposed strategy is coded in C# programming language and compiled using Microsoft Visual Studio 2019. All of the tests are carried out on a computer with an Intel Core i7-8550U CPU @ 1.80 GHz. Due to the inherent randomness of the proposed HGA model, it is executed ten consecutive times for each problem and the Average Percent Deviation (APD) from the global optimum is reported in percentages.

4.1. Parameter configuration of HGA

It is well documented that the meta-heuristics are sensitive to parameter settings [10,11,13,48–50] and HGA is not an exception to this. Therefore, pilot experiments that include different combinations of low and high levels are conducted to tune the parameters by finding the best compromise between the solution quality and the processing time. Population size (N) between $5n$ to 3,000 is recommended to be used with regard to the complexity of the network and the number of activities. It is to be noted that for the larger and more complex problems higher values might be preferred at the cost of added CPU time. Similarly, iteration number ($t_{\max}$) ranging from $1.75n$ to 500 is experienced to yield acceptable solutions; naturally though, larger problems suggest using larger values for this parameter. Table 1 summarizes the recommended ranges of values for all the parameters.

As given in Table 1, since medium- to large-scale problems have significantly larger solution spaces than small-scale problems, larger values are assigned for the population size and iteration number parameters. Similarly, due to the larger dimension of the solution space, increasing Boltzmann constant by 20% is experienced to prevent HGA from getting stuck into the local optima as higher values for this parameter can improve exploration capabilities by promoting the selection of bad mutations.

4.2. Test instances

Well-studied benchmark instances containing various combinations of lag/lead times and logical relationships are used for experimental tests. In addition, data of a real-scale construction project is used to introduce a

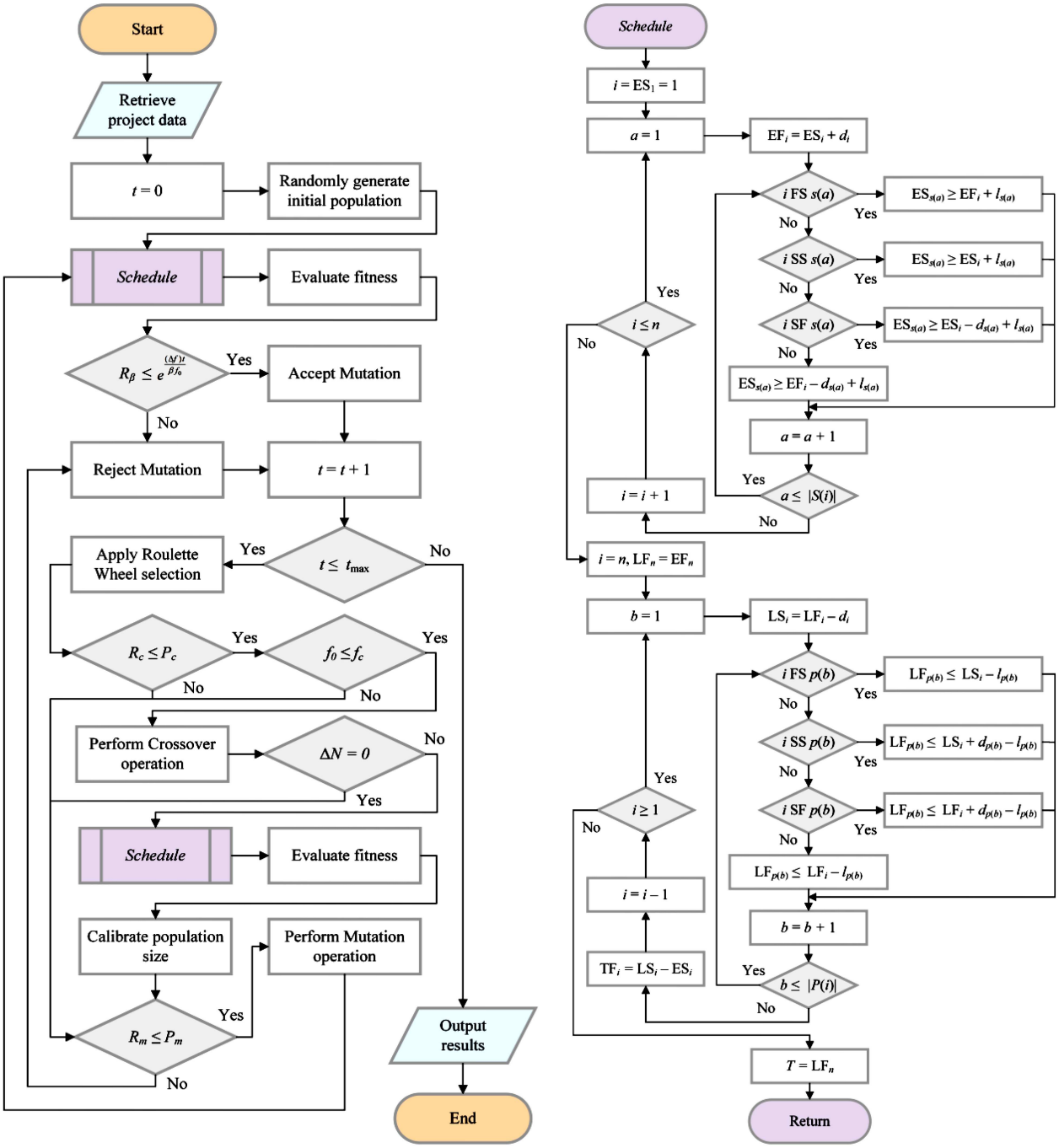

Figure 1. Flowchart of the proposed HGA.

new test example and is fed into HGA. For comparison purposes, optimal solutions of the benchmark instances are also obtained from the literature. Performance of HGA is evaluated not only based on APD from the optima but also with respect to its convergence speed over every instance used.

4.2.1. Small-scale instances

The first set of experimental tests are carried out using a well-known benchmark problem that consists of 29 activities with up to three time-cost alternatives. This problem was originally introduced by Chassiakos and Sakellaropoulos [51] as a highway upgrading project. Later, Sonmez and Bettemir [11] slightly modified this problem by relaxing the external constraints that limited the activity execution times. In the present work, the modified version of Sonmez and Bettemir [11] has been adopted. Activity table outlining the precedence relationships as well as the activity time-cost alternatives of the 29-activity problems is given in Table 2. Figure 2 demonstrates the Activity-on-Node (AoN) diagram of this project and the critical path

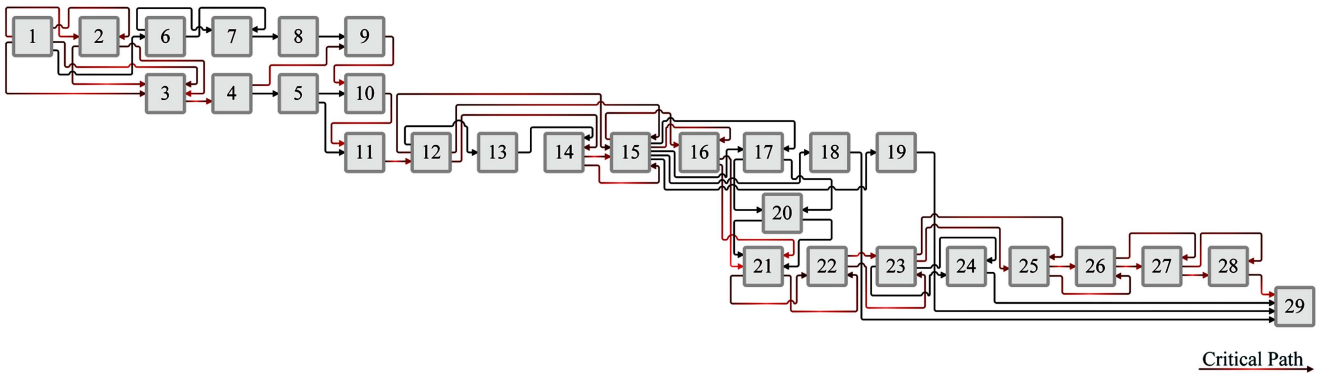

Figure 2. Activity-on-Node (AoN) diagram of the 29-activity problem.

Table 2. Activity table for 29-activity problem [11].

Activity	Predecessor (s)	Mode-1		Mode-2		Mode-3	
		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	–	–	–	–

Table 3. Total direct cost for 447-activity case project.

Major project phases	Total direct cost ($)	Start date	Finish date
Construction works	5,853,926	12/04/2012	08/02/2014
Electrical works	1,728,333	20/04/2012	12/04/2014
Mechanical works	2,265,613	17/02/2013	12/04/2014
Finishing works	7,583,803	30/08/2012	17/05/2014
Other work items	67,368		
Total	**17,499,043**	**12/04/2012**	**17/05/2014**

associated with the normal schedule, i.e., when 'Mode 1' is selected for all the activities. Different time-cost combinations of this problem allow for creation of 8.3×10^9 particular schedules. The 29-activity instance is studied under two different conditions known as 29A and 29B. For 29A problem the indirect cost is assumed as €1,200/day, while, for the 29B problem, in addition to the same rate of daily indirect cost, delay penalty of €1,500/day and incentive payment at a rate of €500/day applies for a maximum desired completion duration of 240 days.

4.2.2. Medium- to large-scale instances

The 29-activity problem described in Subsection 4.2.1 was used by Sonmez and Bettemir [11] as the basis for generation of two larger instances, namely, 290A and 290B problems. Sonmez and Bettemir [11] have constructed the 290-activity network by cloning AoN diagram presented in Figure 2 nine times and relating the ten projects using FS type of relationships. As a result, 290A and 290B problems have been constructed based on problems 29A and 29B, respectively. 290-activity problems are studied using the same assumptions mentioned in subsection 4.2.1 while a desired completion duration of 2,400 days is set for 290B problem. In addition to the problems obtained from the literature, an actual construction project has been also used to demonstrate the potential of the proposed method. As mentioned earlier, only a few studies have exercised problems incorporating the generalized logical relationships with realistic overlapping among the activities. Notwithstanding the limited number of test problems in this domain, this stu57dy takes on the task of exerting an optimization method capable of tackling an actual construction project with properly overlapped activities. Based on what has been discussed above, an actual construction project with over four hundred activities and generalized precedence relationships is modelled and used as one of the test instances.

The project under consideration has been officially handed over to the client. Project scope contained six identical 25-storey residential buildings each with a total area of 25,000 m^2; though, only one of the repeated buildings is used during the analysis since they all share the same schedule and are associated with the same sets of execution modes. The project data have been retrieved from the prime contractor and processed into a useful form. The collected data include information related to all the phases from site preparation to the final hand-over of the building. The type of project data compiled as well as their sources include: (i) network logic from the baseline project schedule, (ii) activity time-cost modes from the bids of the contractors and quotations of the vendors, (iii) agreed completion time and the associated rate of delay penalty from the contract agreement, (iv) actual total project cost from the priced Bill of Quantities of the awarded tenderer, and (v) actual total project duration from the certificate of final completion.

Activity table covering the general precedence relationships of 447 activities and their time-cost modes can be found in Ahmed [52] which is not repeated here for the sake of brevity. Per the baseline schedule, for some activities, lag time or lead time concepts have also been used to secure the time to be delayed or to be advanced, respectively. Time-cost modes comprise the actual time and cost spent on executing the tasks as well as the bids offered by alternative subcontractors. Not all the activities include multiple time-cost modes due to direct execution/procurement by the employer. It should also be noted that, for all the activities the first time-cost mode represents the actual time and cost spent on execution of the corresponding activity.

Table 3 summarizes the major phases of case project and the total direct costs associated with them. Start and finish dates of the major phases are also provided in Table 3. As can be seen from the table, the actual direct costs add up to $17,499,043. Start and finish dates of the project implies a total project duration of 766 calendar days. Though, a total duration of 684 working days is determined according to the calendar assigned to the project. This duration in working days will form the basis for experiments. Project indirect costs, on the other hand, are often assessed for the entire project rather than individual activities. For this purpose, a single rate is determined for the daily indirect cost by uniformly distributing

Table 4. Total indirect cost for 447-activity case project.

Item	Indirect cost
Engineering staff	$295,550
Company lawyer	$126,500
Administration	$182,275
Laboratory staff	$65,550
Electricity, water, gasoline, etc.	$142,715
Total	**$812,590**

the total indirect cost over the project makespan. The total indirect cost for this project is calculated by collecting all the indirect costs spent on principal items as listed in Table 4. This figure is then used to determine the rate of the daily indirect cost as: $812,590/684=$1,188. In addition, as per the contract clauses, a delay penalty of $50,000/day applies if the project completion duration exceeds 684 days. As a result, a total sum of $18,311,635 was spent on the case project since it was completed within the desired duration of 684 days.

4.3. Results and performance evaluation

In this section a series of computational experiments performed via the proposed HGA algorithm for optimization of five challenging instances, namely, 29A, 29B, 290A, 290B, and a 447-activity problem is elaborated. For 29-activity-based problems, the optimal costs are drawn from the literature and the quality of the results located by HGA are compared with the solutions of state-of-the-art methods. These methods include GASA (GA with SA), GMASA (Genetic Memetic Algorithm with SA), and GASAVNS (GA with SA and Variable Neighborhood Search) of Bettemir [43], as well as Hybrid Algorithm (HA) of Sonmez and Bettemir [11]. GASA is a GA in which the acceptance of mutation is under the control of SA. GMASA, on the other hand, combines GA, SA, and Memetic Algorithm (MA) to improve the local search capabilities of the GA. Memetic algorithm provides GMASA with a more systematic approach for performing the mutation process through mutating the genes in a sequential manner rather than a completely random gene selection. GASAVNS contrasts with the methods mentioned above as it merges GA and Variable Neighborhood Search (VNS) to improve the convergence capabilities of GA. Instead of following a trajectory, VNS aids GA to systematically change its search neighborhood and to explore increasingly distant regions in the solution space. Lastly, HA uses SA to decide whether to accept or reject a bad mutation. Furthermore, this method incorporates a mutation acceptance criterion analogous to quantum fluctuations. Unlike SA, QSA uses tunneling field strength rather than temperature to enhance the local search characteristics of GA. Due to its practical importance, computational time requirements of the noted time-cost optimization methods are also included in the performance evaluations.

4.3.1. Small-scale instances

29A and 29B time-cost trade-off problems are optimized using HGA and the results obtained are summarized in Table 5. Figures 3 and 4 illustrate the minimum total cost achieved by HGA per each iteration for problems 29A and 29B, respectively. It can be observed

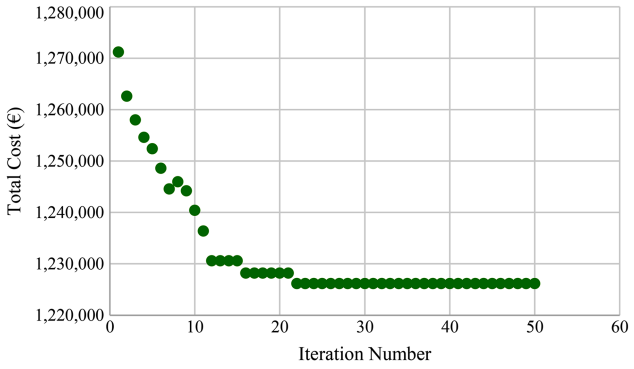

Figure 3. Total cost versus iteration number for 29A problem.

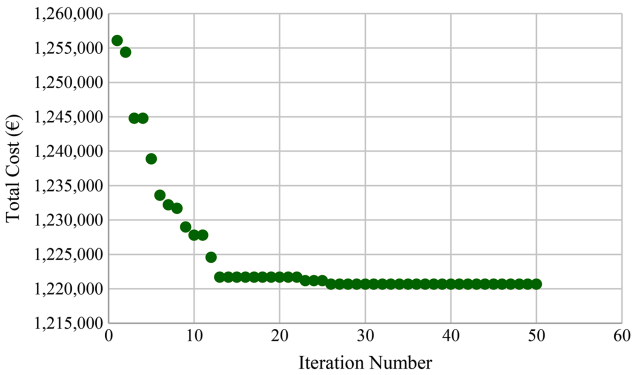

Figure 4. Total cost versus iteration number for 29B problem.

Table 5. Performance comparison over 29-activity problems.

Instance	GASA*		GMASA*		GASAVNS*		HA**		HGA (this study)	
	APD (%)	CPU time (S)	APD (%)	CPU time (S)	APD (%)	CPU time (S)	APD (%)	CPU time (S)	APD (%)	CPU time (S)
29A	0.00	5.00	0.00	8.00	0.09	8.00	0.00	2.00	0.00	0.22
29B	0.00	5.00	0.00	8.00	0.03	8.00	0.00	2.00	0.00	0.22

* Bettemir [43]; ** Sonmez and Bettemir [11].

from the figures that at the initial stages, HGA has taken huge leaps toward better solutions as it is encouraged to perform spatially more extensive explorations at this stage; afterwards, HGA has improved the solutions rather gradually as it is designed to increase exploitation towards the final rounds of iterations. Due to the stochastic nature of the proposed HGA, it is executed ten consecutive times for each problem and the APD from the global optima is measured. The exact solutions for 29A and 29B problems are taken from Sonmez and Bettemir [11] as €1,226,200 for 236 days and €1,220,700 for 221 days, respectively. Using the exact solutions, performance of HGA over 29-activity problems is compared with GASA, GMASA, and GASAVNS of Bettemir [43] and HA proposed by Sonmez and Bettemir [11]. As given in Table 5, the proposed method, similar to GASA and GMASA methods, locates the global optimum solutions for 29A and 29B instances in all of the ten experimental trials. As seen in Table 5, HGA outperforms GASAVNS with respect to the average solution quality across ten consecutive runs as GASAVNS has APD percentages of 0.09 and 0.03 for 29A and 29B instances, respectively. In addition, HGA proves to be more efficient than the existing state-of-the-art SA-based GAs, since, it converges to global optima within 0.22 seconds for both the problem variants; whilst, it takes 5 to 8 seconds for the methods presented in Bettemir [43] and 2 seconds for Sonmez and Bettemir's [11] approach to solve the same problems. SA component is found to significantly improve the convergence capabilities of HGA by locating higher quality solutions with lower APD values. In fact, the inclusion of SA helps GA find the global optimum for both variants of 29-activity problem and thus reduces APD from 0.20% and 0.16% to nil for 29A and 29B problems, respectively.

4.3.2. Medium- to large- scale instances

Time-cost optimization of 290A and 290B problems is performed using HGA and the optimization results are tabulated in Table 6. In Figures 5 and 6 the minimum total costs found by HGA are plotted against the corresponding iteration numbers for problems 290A and 290B, respectively. Raising Boltzmann constant value to 1.2 has limited the elimination of harmful mutations particularly towards the final iterations. This, in turn, has facilitated exploration of some unvisited portions

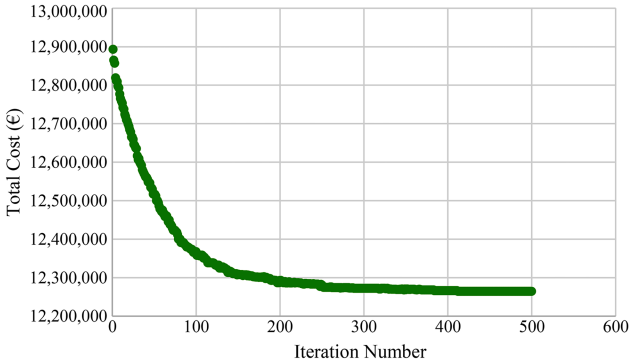

Figure 5. Total cost versus iteration number for 290A problem.

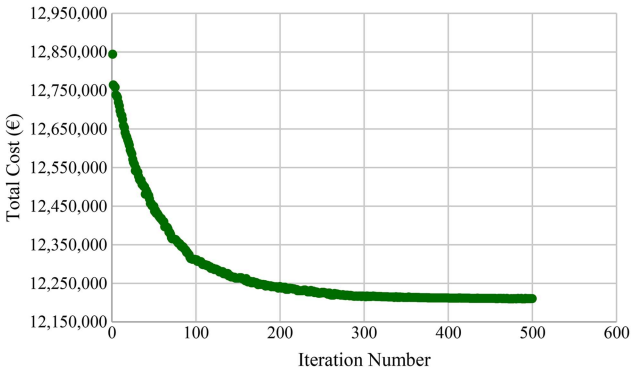

Figure 6. Total cost versus iteration number for 290B problem.

of the search domain and has helped HGA to untrap itself from the local optimum points. For each exercised instance, the solutions clustered along the convergence curve roughly reveal an approximation of the time-cost curve. Due to the inherent randomness of the proposed HGA model, APD from the exact solution is measured based on ten consecutive runs. For the ten times duplicated versions of 29-activity problem, i.e., 290A and 290B, the exact solutions are drawn from Sonmez and Bettemir [11] as €12,262,000 for 2,360 days and €12,207,000 for 2,210 days, respectively. Performance of HGA over 290-activity problems is evaluated against GASA, GMASA, and GASAVNS presented by Bettemir [43] and HA of Sonmez and Bettemir [11]. The results provided in Table 6 continue to support robustness of HGA since it is shown to be able to find high-quality solutions for 290A and 290B with extremely slim deviations of 0.01% and 0.03% from the optima,

Table 6. Performance comparison over 290-activity problems.

Instance	GASA*		GMASA*		GASAVNS*		HA**		HGA (this study)	
	APD (%)	CPU time (S)	APD (%)	CPU time (S)	APD (%)	CPU time (S)	APD (%)	CPU time (S)	APD (%)	CPU time (S)
290A	5.15	-	4.98	–	4.91	–	0.74	–	0.01	12.40
290B	4.99	-	4.75	–	4.68	–	0.43	–	0.03	11.86

* Bettemir [43]; ** Sonmez and Bettemir [11].

respectively. HGA is experimented to find significantly better solutions than the ones reported in Bettemir [43] with larger APDs ranging from 4.68% to 5.15%. In the same fashion, HGA excels the existing state-of-the-art method of Sonmez and Bettemir [11] with respect to solution quality as HA obtains solutions with larger APD percentages of 0.74 and 0.43 for 290A and 290B problems, respectively. Moreover, it takes less than 13 minutes for HGA to unravel the 290-activity problems with the parameter setting discussed in section 4.1. Though, results of GASA, GMASA, GASAVNS, and HA provide no grounds for comparing the convergence speeds since processing times for these problems have not been reported. 13 minutes can nevertheless be considered as an acceptable CPU time for tackling relatively large problems with complicated logics.

The results summarized in Tables 5 and 6 reveal that the HGA method proposed in this study is able to locate better solutions than the ones reported in the existing state-of-the-art methods. Among the previous methods, GMASA, GASAVNS, and HA engage various techniques to improve the performance of SA-based GAs; whereas, GASA takes a similar approach to HGA for regulating the acceptance of mutation by solely integrating SA to GA. Despite the similarities between GASA and HGA, they administer different GA selection and reproduction strategies. As given in Eqs. (12) to (17), HGA uses expected count and roulette wheel as the reproduction and selection strategy while in GASA, only (elitist) roulette wheel selection is implemented. It is important to note that the expected count is used to guide the algorithm in selection of potentially more useful chromosomes for further processing in the mating and reproduction process. Superior performance of HGA is chiefly attributed to the employed selection and reproduction strategies as well as rigorous tuning of the parameters. Since, as discussed in Section 4.1, it is well documented that the meta-heuristics (including GA and its variants) are sensitive to parameter settings. Therefore, defining adequate parameter values by taking into account the complexity of the problems have contributed to the improved performance of the presented SA-based GA. Inclusion of the SA component is experienced to contribute to a drop from 0.22% to 0.01% in APD value for 290A problem and a drop from 0.24% to 0.03% in APD for 290B variant.

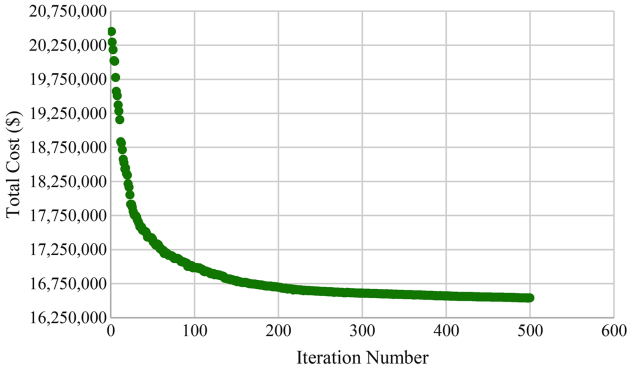

Figure 7. Total cost versus iteration number for 447-activity case project.

Ultimately, HGA is experimented for time-cost optimization of the 447-activity case project and the results of the analysis are provided in Table 7. Figure 7 demonstrates the minimum total cost obtained by HGA per each iteration for this project. As can be seen from the figure, more exploitation and less exploration is promoted toward the latter stages of the optimization. This pattern can mainly be attributed to the cooling schedule of the integrated SA method.

As can be seen from Table 7, HGA requires an average processing time of 16 minutes to tackle the 447-activity case project with the parameter configuration given in Table 1. The average performance of HGA over ten runs reveals that in all the experiments it has been able to locate a solution with a total cost of \$16,548,248 for a total duration of 684 days. HGA is shown to be capable of providing a solution with a total cost less than the actual budget of \$18,311,635 spent on the project. In other terms, due to the \$1,763,387 cost difference, a considerable saving of almost 10% would have been possible should an optimization method such as HGA have been used by the decision makers. In like manner, by applying the same technique to all the six identical 25-storey residential buildings, the owner could have saved north of \$10.5 million for a project with an estimated budget of \$110 million. Table 8 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 every 447 activities.

As discussed, HGA is capable of reducing the total cost of the case project approximately by 10% with a total duration of 684 days, equaling the actual

Table 7. Results achieved by HGA for 447-activity case project.

No. of runs	Conditions				Best solution		Avg. CPU time (Min)
	C_d (\$)	C_{bon} (\$)	C_{pen} (\$)	T_d (Day)	Total cost (\$)	Total duration (Day)	
10	1,188	0	50,000	684	16,539,149	684	16.00

Table 8. Chromosome structure of the best solution for 447-activity case project.

Selected time-cost modes for activities 1 to 447
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,2,
1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,1,1,1, 1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,3,3,3,3,3,3,3,3,3,3,3,3,
3,3,3,3,3,3,3,3,3,3,3,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,3,1,2,2,2,2,2,2,2,2,2,
2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,3,3,3,3,3,2,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1,2,1,3,
2,1,1,1,2,2,1,1,1,3,2,1,1,1,1,1,1,3,3,3,3,3,1,2,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1,1

makespan of the project. In other terms, minimization of cost is facilitated by clever selection of modes and in turn consuming the total slack time of the network, at the cost of decreased schedule flexibility [53]. More specifically, a portion of the schedule flexibility is consumed by reducing the overall slack time by 2.79% (i.e., 2,665 days) in the optimal solution. HGA manages to capture the optimum cost typically by determining the best combination of time-cost alternatives. HGA is also found to best the performance of plain GA by providing better results with lower deviations from the best solution located (i.e., 684 days and $16,548,248). APD values of 0.98% and 0.46% are calculated for results of GA and HGA, respectively. Therefore, it can be concluded that SA contributes to at least halving the deviations for this problem.

Computational experiments on four standard benchmarks and a case project demonstrate that the suggested method is robust and competitive, since it either locates new best-known solutions or repeats the existing optimal solutions for the well-studied TCTPs with generalized logic in a reasonable processing time.

5. Conclusions

Striking a proper balance between the project cost and duration is a complicated and computationally challenging problem and even moreso for large-scale real-life projects. A large body of the literature has hitherto been dedicated to development of optimization methods for this problem, referred to as Time-Cost Trade-off Problem (TCTP). In spite of this, only a few of the existing methods can be extended to real-life applications. This largely stems from the fact that the majority of existing methods lack the ability to tackle TCTPs with the generalized logic. Generalized precedence relationships, meanwhile, play a pivotal role in introducing a certain degree of parallelism and securing a realistic overlapping among the project activities. Aiming at this research gap, this study contributes to both researchers and professionals by presenting an optimization algorithm capable of unraveling large-scale projects with any type of precedence relationship. To this end, a Hybrid Genetic Algorithm (HGA) is proposed since classical Genetic Algorithm (GAs) tend to require large number of iterations to achieve adequate results and that they are more susceptible to getting trapped into local optima.

The capabilities of HGA is experimented over a wide range of problems. The performance of the proposed method is compared with the existing state-of-the-art approaches on four standard benchmarks. The results demonstrate that the suggested method is robust and competitive, since it either locates new best-known solutions or repeats the existing optimal solutions for the well-studied TCTPs with generalized logic in a reasonable processing time. HGA proves to be more efficient than the existing methods since it converges to the global optima faster. In addition to the test instances available in the literature, an actual construction project including over four hundred activities and generalized precedence relationships has been also used to demonstrate the real-life applicability and effectiveness of the proposed method. Selection of the execution modes identified by HGA is shown to contribute to a considerable saving compared with the experience-based decisions made by the experts. It is realized that for the actual construction project, HGA is able to capture a solution with a much lower total cost than what has actually been spent on the project. It is concluded that a saving of almost 10% would have been possible should an optimization model such as HGA have been used by the decision makers earlier, especially 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.

References

1. Kaveh, A., Rajabi, F., and Mirvalad, S. "Many-objective optimization for construction project scheduling using non-dominated sorting differential evolution algorithm based on reference points", *Scientia Iranica*, **28**(6), pp. 3112–3128 (2021). https://doi.org/10.24200/sci.2021.58952.5988
2. Kelley, J.E. and Walker, M.R. "Critical-path planning and scheduling", *Proceedings of Eastern Joint Computer Conference*, Association for Computing Machinery, New York, NY, USA, pp. 160–173 (1959). https://doi.org/10.1145/1460299.1460318
3. Fulkerson, D.R. "A network flow computation for project cost curves", *Management Science, INFORMS*, **7**(2), pp. 167–178 (1961). https://doi.org/10.1287/mnsc.7.2.167
4. Siemens, N. "A simple CPM time-cost tradeoff algorithm", *Management Science, INFORMS*, **17**(6), pp. B354–B363 (1971). https://doi.org/10.1287/mnsc.17.6.B354
5. Goyal, S.K. "A note on a simple CPM time-cost trade-off algorithm", *Management Science, INFORMS*, **21**(6), pp. 718–722 (1975). https://doi.org/10.1287/mnsc.21.6.718
6. Falk, J.E. and Horowitz, J.L. "Critical path problems with concave cost-time curves", *Management Science, INFORMS*, **19**(4), pp. 446–455 (1972). https://doi.org/10.1287/mnsc.19.4.446
7. Foldes, S. and Soumis, F. "PERT and crashing revisited - mathematical generalizations", *European Journal of Operational Research*, **64**(2), pp. 286–294 (1993). https://doi.org/10.1016/0377-2217(93)90183-N
8. Skutella, M. "Approximation algorithms for the discrete time-cost tradeoff problem", *Mathematics of Operations Research*, **23**(4), pp. 909–929 (1998). https://doi.org/10.1287/moor.23.4.909
9. Zheng, D.X.M., Ng, S.T., and Kumaraswamy, M.M. "Applying a genetic algorithm-based multiobjective approach for time-cost optimization", *Journal of Construction Engineering and Management*, **130**(2), pp. 168–176 (2004). https://doi.org/10.1061/(ASCE)0733-9364(2004) 130:2(168)
10. Feng, C.W., Liu, L., and Burns, S.A. "Using genetic algorithms to solve construction time-cost trade-off problems", *Journal of Computing in Civil Engineering*, **11**(3), pp. 184–189 (1997). https://doi.org/10.1061/(ASCE)0887-3801 (1997)11:3(184)
11. Sonmez, R. and Bettemir, O.H. "A hybrid genetic algorithm for the discrete time–cost trade-off problem", *Expert Systems with Applications*, **39**(13), pp. 11428–11434 (2012). https://doi.org/10.1016/j.eswa.2012.04.019
12. Aminbakhsh, S. and Sonmez, R. "Pareto front particle swarm optimizer for discrete time-cost trade-off problem", *Journal of Computing in Civil Engineering*, **31**(1), 04016040 (2017). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000606
13. Agdas, D., Warne, D.J., Osio-Norgaard, J., and Masters, F.J. "Utility of genetic algorithms for solving large-scale construction time-cost trade-off problems", *Journal of Computing in Civil Engineering*, **32**(1), 04017072 (2018). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000718
14. Chen, L., Zhang, J., and Peng, W. "Research on the hierarchical discrete time-cost trade-off problem for program", *Journal of Construction Engineering and Management*, **148**(7), 04022039 (2022). https://doi.org/10.1061/(ASCE)CO.1943-7862.0002293
15. Kelley, J.E. "Critical-path planning and scheduling - mathematical basis", *Operations Research*, **9**(3), pp. 296–320 (1961). https://doi.org/10.1287/opre.9.3.296
16. Butcher, W.S. "Dynamic programming for project cost-time curves", *Journal of the Construction Division, ASCE*, **93**, pp. 59–74 (1967). https://doi.org/10.1061/JCCEAZ.0000191
17. Liu, L., Burns, S. A., and Feng, C. W. "Construction time-cost trade-off analysis using LP/IP hybrid method", *Journal of Construction Engineering and Management*, **121**(4), pp. 446–454 (1995). https://doi.org/10.1061/(ASCE)0733-9364(1995) 121:4(446)
18. Demeulemeester, E. L., De Reyck, B., Foubert, B., Herroelen, W. S., and Vanhoucke, M. "New computational results on the discrete time/cost trade-off problem in project networks", *Journal of the Operational Research Society*, **49**(11), pp. 1153–1163 (1998). https://doi.org/10.1057/palgrave.jors.2600634
19. Nasiri, S., and Lu, M. "Streamlined project time-cost tradeoff optimization methodology: algorithm, automation, and application", *Automation in Construction*, **133**, 104002 (2022). https://doi.org/10.1016/j.autcon.2021.104002
20. Son, J., Hong, T., and Lee, S. "A mixed (continuous + discrete) time-cost trade-off model considering four different relationships with lag time", *KSCE Journal of Civil Engineering*, **17**(2), pp. 281–291 (2013). https://doi.org/10.1007/s12205-013-1506-3
21. Klanšek, U., and Pšunder, M. "MINLP optimization model for the nonlinear discrete time–cost trade-off problem", *Advances in Engineering Software*, **48**, pp. 6–16 (2012). https://doi.org/10.1016/j.advengsoft.2012.01.006
22. Van Eynde, R. and Vanhoucke, M. "A reduction tree approach for the discrete time/cost trade-off problem", *Computers and Operations Research*, **143**, 105750 (2022). https://doi.org/10.1016/j.cor.2022.105750

23. Fondahl, J.M. "A non-computer approach to the critical path method for the construction industry", *Technical Report*, **9**, Construction Institute, Department of Civil Engineering, Stanford University, California (1961). https://hdl.handle.net/2027/mdp.39015000453970
24. Moselhi, O. "Schedule compression using the direct stiffness method", *Canadian Journal of Civil Engineering*, **20**(1), pp. 65–72 (1993). https://doi.org/10.1139/l93-007
25. Bettemir, O.H. and Birgonul, M.T. "Network analysis algorithm for the solution of discrete time-cost trade-off problem", *KSCE Journal of Civil Engineering*, **21**(2), pp. 1047–1058 (2017). https://doi.org/10.1007/s12205-016-1615-x
26. Sonmez, R., Aminbakhsh, S., and Atan, T. "Activity uncrashing heuristic with noncritical activity rescheduling method for the discrete time-cost trade-off problem", *Journal of Construction Engineering and Management*, **146**(8), 04020084 (2020). https://doi.org/10.1061/(ASCE)CO.1943-7862.0001870
27. Xiong, Y. and Kuang, Y.P. "Applying an ant colony optimization algorithm-based multiobjective approach for time-cost trade-off", *Journal of Construction Engineering and Management*, **134**(2), pp. 153–156 (2008). https://doi.org/10.1061/(ASCE)0733-9364(2008)134:2(153)
28. Yang, I.T. "Performing complex project crashing analysis with aid of particle swarm optimization algorithm", *International Journal of Project Management*, **25**(6), pp. 637–646 (2007). https://doi.org/10.1016/j.ijproman.2006.11.001
29. Liu, D., Li, H., Wang, H., et al. "Discrete symbiotic organisms search method for solving large-scale time-cost trade-off problem in construction scheduling", *Expert Systems With Applications*, **148**, 113230 (2020). https://doi.org/10.1016/j.eswa.2020.113230
30. Klanšek, U. and Pšunder, M. "Cost optimal project scheduling", *Organizacija*, **41**(4), pp. 153–158 (2008). https://doi.org/10.2478/v10051-008-0017-3
31. Sakellaropoulos, S. and Chassiakos, A.P. "Project time-cost analysis under generalised precedence relations", *Advances in Engineering Software*, **35**(10), pp. 715–724 (2004). https://doi.org/10.1016/j.advengsoft.2004.03.017
32. Cajzek, R. and Klanšek, U. "Cost optimization of project schedules under constrained resources and alternative production processes by mixed-integer nonlinear programming", *Engineering, Construction and Architectural Management*, **26**(10), pp. 2474–2508 (2019). https://doi.org/10.1108/ECAM-01-2019-0013
33. Tavana, M., Abtahi, A.R., and Khalili-damghani, K. "A new multi-objective multi-mode model for solving preemptive time-cost-quality trade-off project scheduling problems", *Expert Systems with Applications*, **41**(4), pp. 1830–1846 (2013). https://doi.org/10.1016/j.eswa.2013.08.081
34. Holland, J.H., *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, Cambridge, MA, USA: University of Michigan Press (1975). https://doi.org/10.7551/mitpress/1090.001.0001
35. Namazian, A., Haji Yakhchali, S., and Rabbani, M. "Integrated bi-objective project selection and scheduling using bayesian networks: a risk- based approach", *Scientia Iranica*, **26**(6), pp. 3695–3711 (2019). https://doi.org/10.24200/sci.2019.21387
36. Akhbari, M. "Integration of multi-mode resource-constrained project scheduling under bonus-penalty policies with material ordering under quantity discount scheme for minimizing project cost", *Scientia Iranica*, **29**(1), pp. 427–446 (2022). https://doi.org/10.24200/sci.2020.54286.3680
37. Erden, C., Demir, H.I., and Canpolat, O. "A modified integer and categorical pso algorithm for solving integrated process planning, dynamic scheduling and due date assignment problem", *Scientia Iranica*, **30**(2), pp. 738–756 (2023). https://doi.org/10.24200/sci.2021.55250.4130
38. Esmailnezhad, B. and Saidi-Mehrabad, M. "A two-stage stochastic supply chain scheduling problem with production in cellular manufacturing environment: a case study", *Scientia Iranica*, **30**(4), pp. 1399-1422 (2023). https://doi.org/10.24200/sci.2021.53506.3277
39. Albayrak, G. and Ozdemir, I. "A state of art review on metaheuristic methods in time-cost trade-off problems", *International Journal of Structural and Civil Engineering Research*, **6**(1), pp. 30–34 (2017). https://doi.org/10.18178/ijscer.6.1.30-34
40. El-Rayes, K. and Kandil, A. "Time-cost-quality trade-off analysis for highway construction", *Journal of Construction Engineering and Management*, **131**(4), pp. 477–486 (2005). https://doi.org/10.1061/(ASCE)0733-9364(2005)131:4(477)
41. Azaron, A., Perkgoz, C., and Sakawa, M. "A genetic algorithm approach for the time-cost trade-off in PERT networks", *Applied Mathematics and Computation*, **168**(2), pp. 1317–1339 (2005). https://doi.org/10.1016/j.amc.2004.10.021
42. Srinivas, M. and Patnaik, L.M. "Adaptive probabilities of crossover and mutation in genetic algorithms", *IEEE Transactions on Systems, Man, and Cybernetics*, **24**(4), pp. 656–667 (1994). https://doi.org/10.1109/21.286385
43. Bettemir, O.H. "Optimization of time-cost-resource trade-off problems in project scheduling using meta-heuristic algorithms", Ph.D. Thesis, Middle East Technical University, Ankara, Turkey (2009). https://hdl.handle.net/11511/18989
44. Adler, D. "Genetic algorithms and simulated annealing: a marriage proposal", *International Conference on Neural Networks*, IEEE, San Francisco, CA, USA,

pp. 1104–1109 (1993).
https://doi.org/10.1109/ICNN.1993.298712

45. Reeves, C.R. "A genetic algorithm for flowshop sequencing", *Computers and Operations Research*, **22**(1), pp. 5–13 (1995).
https://doi.org/10.1016/0305-0548(93)E0014-K

46. Tumuluru, J.S. and McCulloch, R.C. "A new hybrid genetic algorithm for optimizing the single and multivariate objective functions", *ASABE Annual International Meeting, ASABE, New Orleans Marriott*, New Orleans, LA, USA, 152188606 (2015).
https://doi.org/10.13031/aim.20152188606

47. Wang, L. and Zheng, D.Z. "An effective hybrid optimization strategy for job-shop scheduling problems", *Computers and Operations Research*, **28**(6), pp. 585–596 (2001).
https://doi.org/10.1016/S0305-0548(99)00137-9

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

49. Goncalves, J.F., Mendes, J.J.M., and Resende, M.G.C. "A genetic algorithm for the resource constrained multi-project scheduling problem", *European Journal of Operational Research*, **189**(3), pp. 1171–1190 (2008). https://doi.org/10.1016/j.ejor.2006.06.074

50. Elbeltagi, E., Hegazy, T., and Grierson, D. "Comparison among five evolutionary-based optimization algorithms", *Advanced Engineering Informatics*, **19**(1), pp. 43–53 (2005).
https://doi.org/10.1016/j.aei.2005.01.004

51. Chassiakos, A.P. and Sakellaropoulos, S.P. "Time-cost optimization of construction projects with generalized activity constraints", *Journal of Construction Engineering and Management*, **131**(10), pp. 1115–1124 (2005). https://doi.org/10.1061/(ASCE)0733-9364(2005)131:10(1115)

52. Ahmed, A. "Genetic-algorithm-based optimization approach for time-cost trade-off problem with generalized precedence relationships", M.Sc. Thesis, Atilim University, Ankara, Turkey (2020).
https://hdl.handle.net/20.500.14411/4708

53. Hegazy, T., and Abuwarda, Z. "Schedule flexibility and compression horizon: new key parameters for effective corrective actions", *Canadian Society for Civil Engineering Annual Conference*, CSCE2019 (2019).
https://legacy.csce.ca/elf/apps/CONFERENCEVIEWER/conferences/2019/pdfs/PaperPDFversion_145_0301042149.pdf

Biographies

Saman Aminbakhsh is an Assistant Professor of Construction Engineering and Management at the Department of Civil Engineering at Atilim University. He received his MSc (2013) and PhD (2018) in Construction Engineering and Management from Middle East Technical University (METU), Ankara, Turkey. His research interests mainly include high-dimensional optimization, metaheuristics, safety management, resource leveling, and blockchain technology.

Ary Ahmed received his MSc (2020) in the Construction Engineering and Management program of the Department of Civil Engineering, Atilim University, Ankara, Turkey. His research interests include construction project management and optimization.