

Research Article

Payment Scheduling Problem with Multiple Modes for Activities to Maximize the Net Present Value of a Project

Somayeh Ghasedi  and Behrouz Afshar-Nadjafi

Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

Correspondence should be addressed to Somayeh Ghasedi; somayeh.ghasedi84@gmail.com

Received 16 July 2020; Revised 1 February 2021; Accepted 12 March 2021; Published 25 March 2021

Academic Editor: Kevser Dincer

Copyright © 2021 Somayeh Ghasedi and Behrouz Afshar-Nadjafi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, the multimode payment scheduling project is studied, which aims to maximize the net present value (NPV) of a project by deciding on payments and the start times of its activities. The problem is formulated mathematically from the point of view of the selected contractor of the project. In the proposed model, a bonus and penalty structure is considered, in which the project's activities can be performed through different execution modes. Another interesting feature of the developed model is the possibility of switching between the execution's modes during the implementation of each activity, which can increase the NPV. Because of the proposed model's computational complexity, two metaheuristic algorithms are developed to tackle the underlying problem. In order to evaluate the performance of the developed algorithms, a set of 108 test instances are solved, and the computational results confirm the applicability of the solving methodologies.

1. Introduction

Net present value (NPV) is among the standard methods to evaluate the economic aspects of the projects. Efficient management of financial currents has a key role in improving the benefits of any decision makings including those made in project management. Balanced decisions in multimode case of resource-constrained project scheduling have influential importance in NPV based objectives. In this version of project scheduling problems, each activity has several scenarios for implementation with different duration, cost, and resources to be accomplished. In such cases, when an activity needs to be finished sooner, more money and resources should be spent. By contrast, if it is possible to postpone an activity before the project's deadline, the implementation cost can be reduced through reduction of the consuming resources. This flexibility in selection of the execution modes can be exploited to achieve maximum NPV.

A survey on the literature of the project scheduling problem shows that despite a great deal of attention to

minimizing the makespan of projects, the works on maximizing NPV of projects are rare. How to schedule a project to maximize the NPV has gone under study since 1970, where Russell tried to model the financial dimensions of project management to maximize a project's financial currents. In that model, the problem was formulated as a nonlinear programming on activity-on-arc (AOA) network. Grinold sought to maximize transactions' present value. It was found that a payment scheduling problem can be transformed into a linear program [1]. Russell employed present value maximization to optimize a linearly constrained nonlinear function and utilized a linear programming succession to solve the problem [2]. Russell studied project scheduling for net present value maximization based on resource limitations, cash inflows, and cash outflows. They incorporated 80 test instances to compare six heuristic scheduling strategies [3]. Kazaz and Sepil proposed a programming model to maximize the NPV of client's financial currents. They considered a progress-based model for the financial currents [4]. Payment scheduling problem was studied by Ulusoy and Cebeli for the first time from both the

employer and contractor's point of view. They presented a two-loop genetic algorithm in which the exterior loop denotes the employer and the interior one denotes the contractor. Having decided on the amount of payment in the interior loop, the contractor schedules the activities to maximize the project's NPV [5]. Dayanand and Padman studied a model to improve NPV from the viewpoints of both the employer and the contractor. They presented a two-step algorithm which determines a collection of payments at the first step to maximize the employer's NPV, followed by the second step in which the activities are rearranged to improve the contractor's NPV [6]. Mika et al. investigated multimode resource-constrained project scheduling by incorporating cash flow discounts. They sought to implement the net present value maximization of project's cash flows. Furthermore, they proposed tabu search and simulated annealing for solving the problem [7]. Elazouni and Metwally employed a genetic algorithm (GA) for finance-based scheduling to obtain maximized project profits via financing and indirect cost minimization [8]. He and Xu studied the project payment scheduling from the employer's viewpoint in which activities could be done separately and the goal was to maximize the employer's NPV [9]. He et al. presented the multimode project payment scheduling in which activities can be executed separately and the goal was to determine the start time of the activities and prepayments to maximize the employer's NPV at the project due date [10]. Ranjbar proposed an optimal solution procedure for the project scheduling problem with permitted tardiness and discrete time/resource tradeoffs under maximum net present value objective. They showed that the project costs could be diminished if the mode with the minimum duration of each activity has been selected [11]. Fathollahi and Najafi presented a model to deal with imposed uncertainties in projects with the aim of maximizing the project's NPV. They assumed some uncertain parameters as fuzzy numbers in order to show the efficiency of using GA and SA algorithms in solving the problems [12]. Leyman and Vanhoucke investigated two problems with the constraints on capital, which result in positive cash balance for the intended project. This study aimed to maximize the NPV by assigning a negative or positive cash flow to the activities [13]. Nasrabadi and Mirzazadeh developed a model using Markovian inflationary conditions to minimize the expected present value of execution costs. In their study, unlike most of the previous studies, the inflation rate and cost of items in each cycle were considered continuous-time variables under a Markov process. The numerical example and sensitivity analysis showed that their model could be adopted for the real-life projects as an applicable approach [14]. Leyman and Vanhoucke in another study investigated the single and multimode resource-constrained project scheduling problem. They tried to present a new scheduling technique and general framework with the help of a genetic algorithm aiming at enhancing the net present value of the project. To evaluate the influence of their approach, the proposed technique was tested on several datasets. The final results highlighted the far-reaching impact of the data's parameters on the project's NPV [15]. Hassanpour et al. compared the

performance of genetic algorithm (NSGA-II) and a new multiobjective imperialist algorithm called MOICA for solving a biobjective preemptive multimode resource-constrained project scheduling problem based on minimizing makespan and maximizing net present value. In the end, they proved the superiority of the proposed MOICA over NSGA-II in obtaining the more high quality solutions [16]. Farughi et al. considered the resources as renewable ones, unlike the past studies. They tried to maximize the net present value and project flexibility with the aim of minimizing the project's completion time. The multiobjective simulated annealing algorithm (MOSAA) was selected to solve the suggested model. They found that the large scale problems need to be solved by considering the efficient performance of the suggested algorithm for four presented benchmark problems [17]. Alavipour and Arditi proposed a model for financing cost minimization via several financing alternatives. They introduced a working schedule with typical activity durations. The advantages of their model included lower financing expenses, preventing liquidated damage and project time extension, and an optimized financing schedule with a reduced number of essential activities [18]. Alavipour and Arditi developed an integrated profit maximization framework for expected profit maximization through cost-time trade-off evaluation, activity start time adjustment, financing cost minimization, and work schedule extension minimization. Their model demonstrated its efficiency for contractors to maximize profits [19]. Liang et al. presented a model considering stochastic activity durations in robust resource-constrained max-NPV project problem. Therefore, they used a composite robust scheduling model together with simulated annealing and tabu search to achieve the satisfactory solutions. Their approach provided them with reducing the chance of payment plan disruptions besides contributing to the notable performance of project scheduling [20]. Delgoshaeia et al. proposed a dynamic backward approach to enhance the remaining resource usage while carrying out the project. In this regard, they could maximize the net present value (NPV) in multimode resource-constrained project scheduling problem considering activity splitting. Their approach showed a considerable effect and flexibility of the backward approach to be utilized in real-world projects [21]. Zarei et al. proposed a mathematical model for the project schedule to find an optimal solution with the aim of maximizing the net present value and minimizing the project time. GA, SA, and PSO algorithms were selected to solve the proposed model. The best option to perform activities would be considered, which led to a balance between time and cost. GA algorithm showed a better performance compared to the other two algorithms. However, SA was preferable in terms of using less computational time to solve the problem [22].

There are many works in the literature of project scheduling considering multiple modes for executing the activities. Hsu and Kim developed a priority rule heuristic algorithm for multimode resource investment problems [23]. Zhu et al. focused on a multimode problem with limited general resources [24]. Sabzehparvar and Seyed-Hosseini studied the model with several discrete modes with resource restrictions considering the prerequisite relationships to maximize and minimize the mode-dependent delays of the

activities [25]. In another study, they proposed a model for multimode resource investment problem with mode-dependent time lags [26]. He et al. contributed to multimode project payment scheduling for maximizing contractor net present value under project deadline constraints. They developed a tabu search and simulated annealing heuristic algorithms. The latter was found to be promising in solving the instances of the problem [10]. Peteghem and Vanhoucke developed a problem considering multiple execution modes for each activity of the project. They utilized a bipopulation genetic algorithm to solve this multimode resource-constrained project scheduling problem (MRCPSP). They used two types of populations as well as an extended serial schedule generation scheme in order to minimize the finish time of activity. The final results proved the positive influence of the mentioned elements on the performance of the genetic algorithm [27].

Azimi et al. studied multimode resource-constrained project scheduling with the aim of maximization of the net present value and minimization of makespan simultaneously. To solve this problem, they used an evolutionary algorithm named multiobjective particle swarm optimization (MOPSO) [28]. A mixed-integer formulation and a solution procedure based on simulated annealing were proposed by Afshar-Nadjafi for the multimode resource availability cost problem [29]. Qi et al. extended a modified version of particle swarm optimization including forward activity list for the multimode resource availability cost problem [30].

Multimodal genetic algorithms (MMGAs) have been used by Pérez et al. to optimize the resource-constrained multiproject scheduling problem (RCMPSP) as an alternative approach instead of heuristics algorithms based on priority rules. This approach enabled managers to use the capability of this algorithm to select between suboptimal solutions [31]. Nemati-Lafmejani et al. developed a novel biobjective optimization model to deal with the multimode resource-constrained project scheduling problem (MRCPSP) together with the contractor selection (CS) problem. The final goal of their study was to minimize the total project cost and minimize project makespan. Therefore, 30 test problems of various sizes were solved with the help of two metaheuristic algorithms called a nondominated sorting genetic algorithm (NSGA-II) and multiobjective particle swarm optimization algorithm (MOPSO). The most worth-mentioning conclusion to be drawn was that the makespan of the project was curtailed by increasing the number of contractors, which could provide decision-makers with a wider range of flexible and applicable approaches [32]. Ehsanifar et al. suggested a model with the ability to perform activities under different executive modes. Their model aimed to minimize project completion time and maximize the cash flow of the project. They used simulating annealing and multiobjective particle swarm optimization (MOPSO) for solving the proposed model in which SA showed better performance in obtaining the solutions [33].

Having a review of the project scheduling literature, we found many works on project scheduling problems with possibility of preemption for the activities. Chiu-Chi compared and evaluated the benefits and downsides of traditional project management and theory-of-constraints (TOC)

in project management. They also introduced an augmented resource-constrained TOC project scheduling framework [34]. Józefowska et al. performed an advantage/disadvantage comparison of traditional and TOC project management and proposed a TOC project scheduling method under resource constraints [35]. Afshar-Nadjafi et al. presented multimode resource-constrained project scheduling problem (P-MRCPSP) to minimize the project's makespan subject to mode changeability after preemption. He used simulated annealing (SA) algorithm to obtain a globally optimum solution. He concluded from the results of 480 test instances of the problem that mode change would be so influential in improving the optimal makespan of the intended project [36]. Delgoshaei et al. developed a method to address the long delay problem caused by resource overallocation in the project completion date. This being the case, they adopted a new forward programming heuristic to consider positive cash flow priority and activity ID rules simultaneously. Their model proved that the makespan would be increased by modifying the overallocated schedules [37]. Afshar-Nadjafi in another study developed the preemptive multimode resource-constrained project scheduling problem (P-MRCPSP), where mode changeability is allowed following preemption, to minimize makespan of the project. He used simulated annealing (SA) algorithm to deal with this problem and applied statistical method in order to determine the effect of the presented method on the problem. It turned out that mode change could be influential in reaching the optimal makespan of the project [38].

As mentioned, in the past decades, there have been many studies on different management problems. From the point of view of activities, project management has several discrete modes based on prerequisite relations, the single or multimode executive conditions of activities, and the possibility of interruption of activities, and from the viewpoint of the objective function, the goal of a problem can comprise different goals for optimization. In the present study, it is decided to have the possibility of mode change during execution of the project.

Herein, we assume that each activity can be split intentionally for resolving the resources' conflicts. In addition, the project manager can take advantage of the multimodeness of the activities. More specifically, the planner can relax the uniqueness of the execution mode, which permits an activity to be restarted after preemption with a different mode than the initial mode. This may result in a better solution than the classic models because it relaxes some constraints, which in turn widens the solution space. In terms of activity preemption, it needs to be highlighted that although splitting the activities may culminate in a longer project's makespan, it can increase the NPV of the project, which is the objective function considered in this research.

The remainder of this paper is structured as follows. Section 2 describes the optimization model. The solution procedures are presented in Section 3. The computational results are analyzed and reported in Section 4. Finally, Section 5 concludes the paper.

2. Model Description

From the contractor's viewpoint, the project payment scheduling problem with a multimode execution structure for maximization of NPV happens at regular time intervals. Project activities management with one specific due date happens while considering the probability of interruption of the activities and changing the modes.

We assume that the activities are numbered from 1 to I . Each activity has different execution modes (1, 2, ..., i). The project is shown as an AON network in which the node set denotes the activities and the arc set represents the finish-to-start prerequisite restrictions between the activities. At each time instant, each activity is done according to one of its own execution modes. There is a limitation for each resource per time unit (day), which will result in delay if its usage is exceeded; in other words, there would be preemption in case of resource shortages. If an activity is interrupted the execution mode of that activity can be changed to resolve the resource balance. At the end of each time period, the accomplished percent of each performed activity at the mentioned period is calculated, and then the payment amount has resulted through cost, benefit, and the payment proportion.

Payments are done at regular periods and there exists one payment at the reward and punishment system in which there is a due date for each activity. If the activity is done sooner, there is a reward for the days the activity is finished sooner than scheduled and, of course, there is a fine for any delay. The total cost can be achieved by calculating the employer's payment to the contractor and also the benefit and cost of each daily activity. In NPV calculation, as the total costs and payments are evaluated in comparison to the first day of the project, money loses its value day by day; that is why the interest rate is used to find out the final value of cash currents.

2.1. Assumptions

- (1) Project network is an AON which is a directed digraph with multiple execution modes for each activity.
- (2) Payments are done periodically and the last payment should be done while finishing the project.
- (3) The project should be completed before its due date.

- (4) The contract expenditure includes both the contractor's total expenditure and his benefit.
- (5) If there is preemption in activities, there can be a change in execution mode.
- (6) Each activity in each day of the project can be executed by just one mode of the execution modes.
- (7) All the parameters in each problem are deterministic.
- (8) There is a due date for each activity.
- (9) There is a certain amount of job to be accomplished for each activity.
- (10) There is a reward and punishment system for completion of each activity.
- (11) All the resources are renewable.

2.2. Parameters

w_i : Workload of activity i

m_i : Executive mode of activity i

$w_i m_i$: Performed amount of job by activity i if it is carried out according to mode m_i (in each day).

h_i : Due date of activity i

K : Number of payments

p_k : Amount of payment k

e_i : Bonus of activity i if it is carried out sooner than the due date.

e'_i : Penalty cost of activity i if it is carried out after the due date.

α : Modification rate

θ : Payment proportion

ε : Benefit margin

D : Project due date

D' : Project completion time

T' : Payment intervals

T_0 : Payment time of day zero (at the beginning of the project in which we have no payment)

The model:

$$\begin{aligned} \text{Max NPV} = & \sum_{K=1}^{[D'/T']} P_K e^{-\alpha T_K} + P_K e^{-\alpha D'} - \left(\sum_{K=1}^{([D'/T']-1)} e^{-\alpha T_K} + \sum_{i=1}^I \sum_{m_i=1}^M \sum_{t=T_{k-1}}^{T_k} C_{it} \cdot X_{im,t} + e^{-\alpha D'} \sum_{i=1}^I \sum_{m_i=1}^M \sum_{t=[D'/T'] \cdot T' + 1}^{D'} C_{it} \cdot X_{im,t} \right) \\ & + \sum_{i=1}^I e_i \sum_{t=EF_i}^{h_i} O_{it} e^{-\alpha(h_i-t)} - \sum_{i=1}^I e'_i \sum_{t=h_i+1}^{LF_i} O_{it} e^{-\alpha(t-h_i)}, \end{aligned} \quad (1)$$

subject to

$$X_{im,t} + X_{im,t \leq 1}; \quad i = 1, \dots, I, \quad (2)$$

$$\sum_{t=EF_i}^{LF_i} O_{it} \cdot t \leq D', \quad (3)$$

$$\sum_{EF_i}^{LF_i} O_{it} = 1; \quad i = 1, \dots, I \quad (4)$$

$$\sum_{K=1}^{D'/T} P_K = P_{MAX}; \quad k = 1, \dots, K, \quad (5)$$

$$P_K \leq \theta(1 + \varepsilon) \sum_{i=1}^I \left(\frac{W_{iT_k} - W_{iT_{k-1}}}{W_i} \right) * C_i, \quad k = 1, \dots, \frac{D'}{T}, \quad (6)$$

$$\sum_{i=1}^I \sum_{m_i=1}^M R_{i,m_i} - X_{im,t} \leq \sum_{L=1}^L R_L; \quad t = 1, \dots, D', L = 1, k, \dots, L, \quad (7)$$

$$\sum_{ES_i}^{LF_i} \sum_{m_i} W_{im_i} X_{im,t} \geq W_i, \quad i = 1, \dots, I, \quad (8)$$

$$\sum_{t=1}^{D'} \sum_{m_i=1}^M W_{im_i} \cdot X_{im,t} \geq W \sum_{m_j} X_j M_j; \quad (t+1) \in [ES_i + 1, LF_j - 1], \quad (9)$$

$$C_i = \sum_{O}^{D'} \sum_{m_i} \sum_{L=1}^L R_{iL} \cdot X_{im,t} \cdot C_L; \quad i = 1, \dots, I, \quad (10)$$

$$W_{iT_k} = \sum_{t=T_{k-1}}^{T_k} \sum_{m_i=1}^M W_{im_i} \cdot X_{im,t}; \quad k = 1, \dots, K, i = 1, \dots, I, \quad (11)$$

$$C_{it} = \sum_{m_i} \sum_{L=1}^L R_{iL} m_i \cdot X_{im,t} \cdot C_L; \quad t = 0, \dots, D', i = 1, \dots, I. \quad (12)$$

T_k : Payment time of k

R_{i,m_i} : Amount of L resource needed for activity i if it is carried out in mode m_i (in each day).

C_{it} : Cost of the necessary resources to carry out activity i in t day

C_L : Specific cost of each resource L

C_i : Total cost of the necessary resources to carry out activity i .

R_L : Renewable limit of resource L in each day. There is a limit for each resource during the whole project which cannot be exceeded.

W_{iT_k} : Performed amount of job by activity i at the end of the payment time of k

$O^i_t = \begin{cases} 1, & \text{if activity } i \text{ is finished at time } t = 1, \\ 0, & \text{otherwise.} \end{cases}$

EF^i : Earliest finish time of the activity i

LF^i : Latest finish time of the activity i

ES^i : Earliest start time of the activity i .

2.3. Objective Function and Constraints. Based on the abovementioned formulation, the optimization model of max NPV with a bonus penalty structure can be constructed from the perspectives of the contractor. In this model, the objective function represents the contractor's NPV of the project. The NPV is the subtraction of all discounted costs associated with the project from the employer's discounted payments to a contractor while considering the bonus and penalty structure of activities. The total costs include the costs of used resources in a project. Also, if an activity is done sooner, there is a bonus, and if it is finished later, there is a penalty for it. To calculate NPV of the contractor, the penalties are subtracted from the employer's payments and then the bonuses are added to that.

Equation (2) shows that the project activities are executed just in one of their executive modes per day and each activity cannot be done with more than one mode within a day. Equation (3) shows that the project is done before its deadline; D . Equation (4) calculates the finish point of each activity. Equation (5) represents the summit of payments. Equation (6) shows the value of each payment. Equation (7) guarantees that using each resource in the whole project has a specific restriction. Equation (8) specifies that each activity has work content and in order to fulfill one activity, it should reach its job content. Equation (9) denotes the prerequisite relationships in a network. Equation (10) shows the total cost of each activity in the project. Equation (11) denotes the performed job content in each activity at the end of each time period. Equation (12) represents the total cost of each activity in a time period.

3. Heuristic Solution Procedures

Genetic algorithm (GA) has been used frequently to solve scheduling problems due to its effectiveness and efficiency in exploring solution space for complex problems. GA can work on a collection of feasible solutions at any given time. This is why it can go through a bigger space in a shorter time.

The algorithm is started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population (generation). This is motivated by the hope that the new population will be better than the old one by inheriting good features from the parent population. Solutions that are selected to form new solutions (offspring) are selected

according to their fitness—the more suitable they are, the more chances they have to reproduce.

Simulated annealing is a method for finding a good (not necessarily perfect) solution to an optimization problem. If you are in a situation where you want to maximize or minimize some function, your problem can likely be tackled with simulated annealing. This algorithm is among the most powerful algorithms to face the combinatorial optimization problems and this is why it is selected to solve our model.

3.1. Solving the Problem with GA Algorithm

3.1.1. Choosing Chromosome. The number of project activities is shown by I and the vector length of the chromosome is assumed to be $2I$. Each chromosome is divided into two divisions with length I and the members of the first part of the chromosome represent the activities and the members of the second part show the number of executive modes.

3.1.2. Generating Initial Population of Chromosomes. Producing chromosomes of the initial population or generation is randomly performed to start the algorithm.

Consider matrix i . “ i ” shows the number of activity. Activities are shown while considering their prerequisite relationships in a matrix. If “ i ” is the prerequisite of j , matrix space will be filled. Now, we act as the following to choose the primary sequence of the activities.

Step 1:

All the activities should be put on a list called activity list.

Step 2:

Consider a collection called S-list which includes activities which can be performed based on precedence relations. This collection is empty at first at the start of the algorithm.

Step 3:

We consider another collection named M-list, which includes all the activities which can be at any stage and can be executed.

Finally, we call the arranged list of activities a sequence. Then, the activities are emitted from the activity list, while considering the prerequisite relationships, and after that, they are added to M-list. Through producing a random number, the activities in M-list are emitted and then put in the sequence list. In this method, the sequence of the activities is in a way that we can have a proper sequence and, therefore, chromosomes. A detailed structure of the proposed algorithms is depicted in Figures 1 and 2.

3.1.3. Fitness. Herein we evaluate the fitness value of each chromosome in the population. With the use of the fitness function's value, we can calculate each chromosome's probability and its cumulative probability to find the best chromosome. After the production of one generation of the

initial population, it is time to have the genetic operator of crossover and mutation. Before the crossover and mutation operator, evolution operations are performed on chromosomes. First, chromosomes are put in an ascending order based on the fitness value. A percent of the new generation population comprises the best chromosomes of the previous generation. Having arranged the chromosomes, the best chromosomes are directly transferred to the new generation based on the fitness value. The chromosomes transferred through evolution are transferred based on the defined evolution probability ($1 - P_c - P_m$). The achieved answers out of the elite orientation are used in order to have the crossover and mutation operators.

3.1.4. New Population. Create a new population by repeating following steps until the new population is completed.

3.1.5. Selection. Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected). Selecting parent chromosomes for the crossover is done with roulette wheel. We calculate the cumulative function probability on value for each chromosome and then make a random number between zero and one to have roulette wheel. The generated random number is adapted to the cumulative probabilities of selected chromosome and then we choose the parent chromosome adapted to the random number to have the crossover operator. The other parent chromosome should be chosen the same way, too. It should be added that if the chosen adapted number is put between two cumulative probabilities by roulette wheel, the higher cumulative probability is chosen. Parents are called P_1 and P_2 .

3.1.6. Applying the Operators. In the above algorithm, the chromosomes include two parts. The first part is the sequence of activities and the second part shows the executive modes. The crossover operator is applied to the first part of the chromosome and the mutation operator is applied to the second part of it, and they both produce children as the new answers.

(1) **Crossover.** At first, it is applied to the first part of parent chromosomes with probability of (P_c) and then the mutation operator is applied to the second part of chromosomes with the probability of (P_m). The crossover operator combines the features of the two parent chromosomes to produce two children who have the qualities of their parents. The crossover operators are one of the most important genetic algorithm operators and performing the algorithm correctly is really dependent on the proper designing of the crossover operator. Chromosomes are chosen in pairs and each pair changes into an intersect operator with the probability of (P_c) which results in the production of two children. In this algorithm, one-point crossover is used to choose parent chromosome and the roulette wheel is used for the crossover.

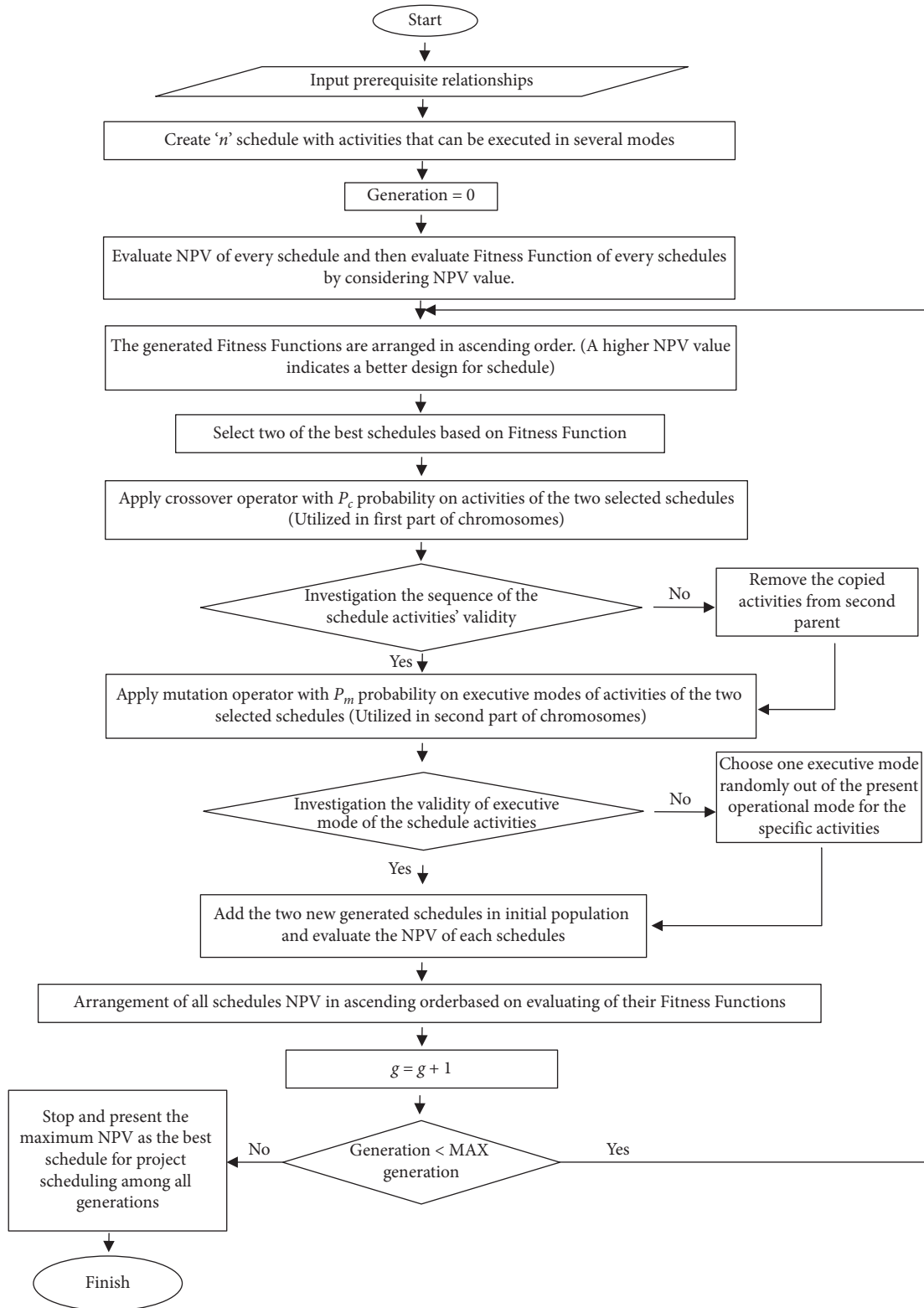


FIGURE 1: Conceptual flowchart of optimization process by GA.

Firstly, in order to avoid the production of improper children, the first part of parent chromosomes, point 1 to cutting point, is copied in the child's chromosome. Now, we search all the copied activities of the first two parents in the

second parent and omit them in the second parent. Then, the remaining activities of the second parent are copied in the first child in a succession similar to that of the second parent from cutting point to I .

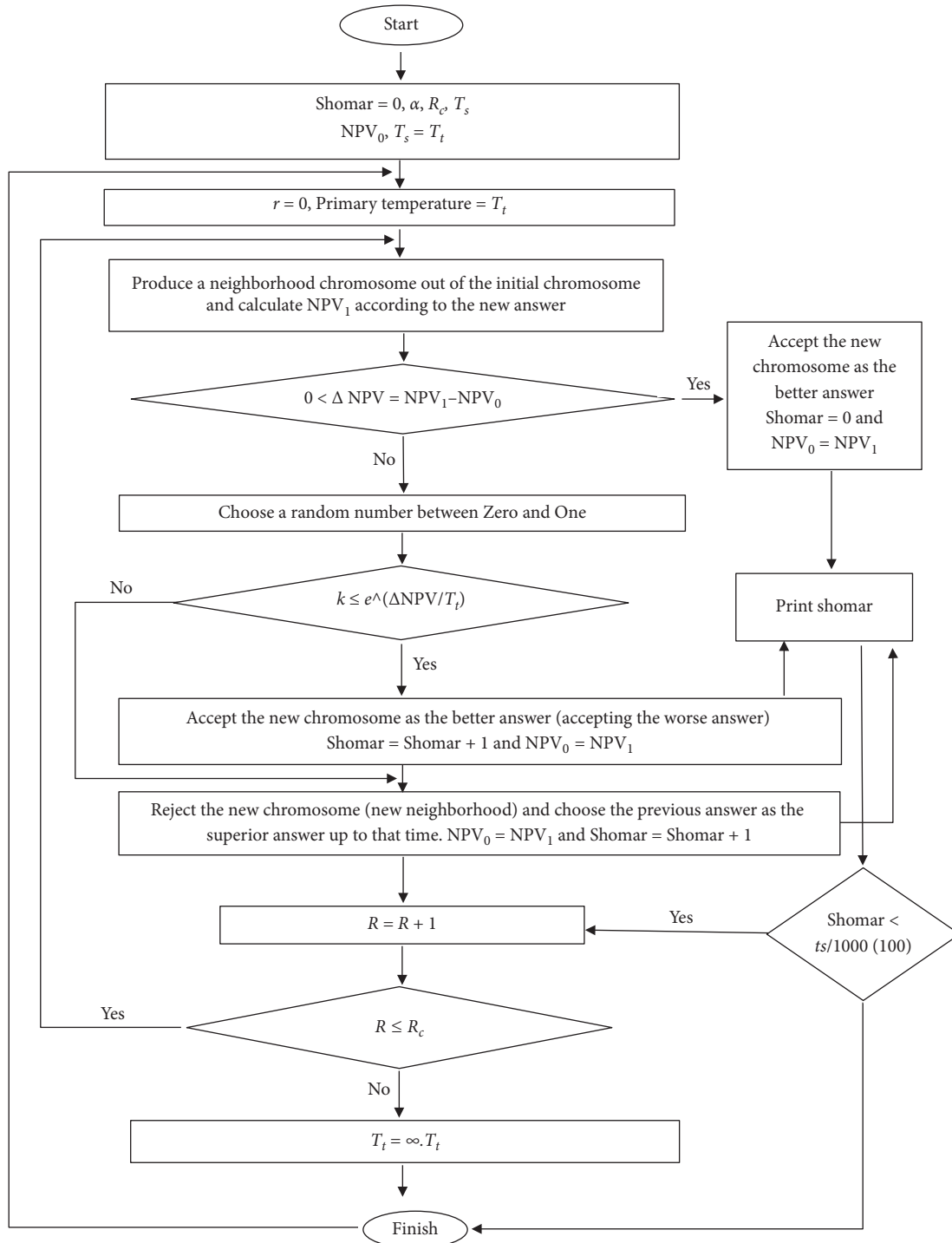


FIGURE 2: Conceptual flowchart of optimization process by SA.

(2) *Mutation*. With a mutation, probability mutates new offspring at each locus (position in chromosome). The chromosomes in the above problem have two I parts. Therefore, in order to finally produce the children, the operator should be applied to the second part of chromosome. The mutation operator is the selective operator applied to the second part of chromosome; roulette wheel is used to choose chromosomes to have a mutation. The

second parts of chromosomes are the modes to execute the activities. Later, the mutation operator is applied to the second part of a chromosome.

$I+1$ spaces relate to the executive mode of the first activity, $I+2$ is for the operational mode of the second activity, and $I+1$ is related to the executive mode of activity, all of which have the mutation operation, too. The mutation operator is applied to the modes, and if we face the

infeasibility chromosome, we will randomly choose one executive mode out of the present operational mode for the specific activities.

3.1.7. Accepting. Place new offspring in a new population. Each time the algorithm produces a new generation of answers and then achieves the new answers through applying the elitism operators of crossover and mutation.

3.1.8. Replace. Use the newly generated population for a further run of the algorithm. The new population will be better than the old one, which is selected for a further run of the algorithm.

3.1.9. Test. If the end condition is satisfied, stop and return the best solution in current population. The needed condition to end the genetic algorithm in the above problem is considered to be the number of generations. Each generation has better answers compared to the previous one because each time the best is transferred to the next generations or we can produce a child through combining the best chromosomes of parents. When the condition to end the algorithm is met, the optimum answer is achieved which is the best succession of activities with the executive modes at the beginning of the project to have the highest amount of NPV for the contractor and to solve the management problem.

3.1.10. Loop. If the condition is not fulfilled, go to step 3.1.5.

3.2. Solving the Problem with SA Algorithm. To choose chromosome in this method is similar to GA algorithm. Project activities equal I and the vector length of the chromosome is $2I$. Each chromosome is divided into I parts. The members of part one are the activities and the members of part two show the number of executive modes for each activity. First, we enter the primary parameters and then perform them while considering the condition of stopping the algorithm during the described steps.

- SA algorithm parameters
- Starting temperature (T_s)
- Iterations at each temperature (R_c)
- Predetermined cooling rate ($0 < a < 1$)
- SA algorithm stop condition

We defined the condition of algorithm stop ($t_s/1000$) which is a counter as the function of temperature. First, we produce a proper answer under the initial temperature and then calculate the value of objective function according to the primary chromosome.

SA algorithm operation steps are as follows.

Step 1:

Primary temperature (T_s), the number of repetitions in each temperature (R_c), predetermined cooling rate ($0 < a < 1$), and (Shomar = 0) as a counter are entered.

Step 2:

Now we determine the present temperature ($T_t = T_s$) and the amount of objective function (NPV_0).

Step 3:

Now we determine the present temperature as T_t , and the present repetitive number $R=0$. Later, we enter Shomar and the primary answer (initial chromosome) and the amount of objective function (NPV_0).

Step 4:

We produce a neighborhood chromosome out of the initial chromosome and calculate NPV_1 according to the new answer.

Step 5:

If $\Delta NPV = NPV_1 - NPV_0 < 0$, we accept the new chromosome as the better answer, Shomar = 0 and $NPV_0 = NPV_1$; now go to 6. If not, we choose a random number between Zero and One.

If $R \leq e^{\Delta NPV/T_t}$, we accept the new chromosome as the better answer (accepting the worse answer), Shomar = Shomar + 1 and $NPV_0 = NPV_1$; go to 6. If not, we reject the new chromosome (new neighborhood) and choose the previous answer as the superior answer up to that time. $NPV_0 = NPV_1$ and Shomar = Shomar + 1 and now go to 6.

Step 6:

Print Shomar and if Shomar $< t_s/1000$, go to 7; if not go to 9.

Step 7:

Now $R = R + 1$. If $R \leq R_c$, go to step 4. If not, go to step 8.

Step 8:

$T_t = a \cdot T_t$; go to step 3.

Step 9:

Print Shomar and Render the achieved results and offer the best answer of NPV.

After frequent and defined repetitions which are a function of temperature, if no better condition is achieved for NPV, we stop the algorithm and then the best answer up to that time is given as the optimum answer.

Chromosome's state of being proper is another important consideration. If one activity is a prerequisite for the next activity, it cannot be a proper candidate to produce new neighborhood ones, because if it is exchanged by each of the next activities which are not the prerequisite, it will be called improper in its next first activity due to being a prerequisite.

The flowcharts of optimization process by SA and GA algorithms are shown in Figures 1 and 2.

4. Computational Results

The function of the two metaheuristic algorithms (GA and SA) is compared and studied through different tests. To do such tests, we require defining some proper problems. Obviously, setting the algorithm parameters has a direct

TABLE 1: Taguchi's orthogonal arrays matrix (GA).

Run	A	B	C
1	A(1)	B(1)	C(1)
2	A(1)	B(2)	C(2)
3	A(1)	B(3)	C(3)
4	A(2)	B(1)	C(2)
5	A(2)	B(2)	C(3)
6	A(2)	B(3)	C(1)
7	A(3)	B(1)	C(3)
8	A(3)	B(2)	C(1)
9	A(3)	B(3)	C(2)

TABLE 2: Operators, parameters, and their levels (GA).

Factor	Symbol	Level	Type
Temp	A	Small problems	$B(1) = 10, B(2) = 20, B(3) = 30$
		Medium problems	$B(1) = 40, B(2) = 50, B(3) = 60$
		Large problems	$B(1) = 80, B(2) = 100, B(3) = 120$
Repeat	B	Small problems	$C(1) = 0.6, C(2) = 0.7, C(3) = 0.8$
		Medium problems	$C(1) = 0.7, C(2) = 0.8, C(3) = 0.9$
		Large problems	$C(1) = 0.7, C(2) = 0.8, C(3) = 0.9$
α	C	Small problems	$C(1) = 0.05, C(2) = 0.07, C(3) = 0.09$
		Medium problems	$B(1) = 0.05, B(2) = 0.07, B(3) = 0.09$
		Large problems	$B(1) = 0.05, B(2) = 0.07, B(3) = 0.09$

relation with its efficiency; this is why Taguchi method is used to regulate the parameters.

4.1. Assumptions and Setting of GA Parameters. In this algorithm, the initial population is randomly produced as big as the pop size. Random move and crossover are used as operators. The algorithm is stopped after a certain repetition (generation). Therefore, the algorithm has four parameters called probability crossing, probability mutation, population size (pop size), and generation number (generation). In Tables 1–5, the operators and parameters along with their levels are given.

4.2. Assumptions and Arranging or Setting SA Parameters. In this algorithm, we consider a hypothetical value for temperature based on previous studies (Initial TEMP = 20000). The algorithm is stopped after a certain repetition ($T/1000$) and it has three parameters called initial temperature (TEMP), repetition in each temperature (Repeat), and temperature decrease coefficient (α). In Tables 6–10, the operators and parameters along with their levels are given.

In this study, the performance of the proposed methods is evaluated by two indicators, which are NPV and solving time. Comparing the results of doing tests on solution algorithms, the tables and graphs show the average function evaluation of GA and SA algorithm in three groups called small, medium, and large groups (each problem is performed five times).

To compare the two presented algorithms, 108 sample problems were produced and each of the mentioned problems was solved using genetic and simulated annealing algorithms. The sample problems were given in three groups

TABLE 3: Optimum levels of GA parameters in small size instances.

Factor	Symbol	Optimal level
Pop size	B	A(2) = 20
(Pc)	B	B(2) = 0.7
p_m	B	C(2) = 0.07

TABLE 4: Optimum levels of GA parameters in medium size instances.

Factor	Symbol	Optimal level
Pop size	B	A(2) = 50
(Pc)	B	B(2) = 0.8
p_m	B	C(2) = 0.07

TABLE 5: Optimum levels of GA parameters in large size instances.

Factor	Symbol	Optimal level
Pop size	B	A(2) = 100
(Pc)	B	B(2) = 0.8
p_m	B	C(2) = 0.07

called small, medium, and large. The achieved results are shown in Figures 3–8 through two algorithms.

As can be seen, the genetic algorithm has achieved much better answer in comparison to simulated annealing algorithm at 100 percent of the cases. Further as the time needed to solve problems by SA algorithm was so close to that of GA algorithm in most problems and even in some problems, solving them by SA algorithm was much more time-consuming, and the achieved answers through genetic algorithm seemed better and finally resulted in higher NPV for the

TABLE 6: Taguchi’s orthogonal arrays matrix (SA).

Run	A	B	C
1	A(1)	B(1)	C(1)
2	A(1)	B(2)	C(2)
3	A(1)	B(3)	C(3)
4	A(2)	B(1)	C(2)
5	A(2)	B(2)	C(3)
6	A(2)	B(3)	C(1)
7	A(3)	B(1)	C(3)
8	A(3)	B(2)	C(1)
9	A(3)	B(3)	C(2)

TABLE 7: Operators, parameters, and their levels (SA).

Level	Factor	Symbol	Type
Small	Temp	A	A(2) = 100000
	Repeat		B(2) = 15
	α		C(3) = 0.99
Medium	Temp	B	A(2) = 500000
	Repeat		B(2) = 30
	α		C(3) = 0.99
Large	Temp	C	A(2) = 1500000
	Repeat		B(2) = 50
	α		C(3) = 0.99

TABLE 8: Optimum levels of SA parameters in small size instances.

Factor	Symbol	Optimal level
Temp	A	A(2) = 100000
Repeat	B	B(2) = 15
α	C	C(3) = 0.99

TABLE 9: Optimum levels of SA parameters in medium size instances.

Factor	Symbol	Optimal level
Temp	B	A(2) = 500000
Repeat	B	B(2) = 30
α	C	C(3) = 0.99

TABLE 10: Optimum levels of SA parameters in large size instances.

Factor	Symbol	Optimal level
Temp	B	A(2) = 200000
Repeat	B	B(2) = 50
α	C	C(3) = 0.99

contractor. In general, it can be concluded that genetic algorithm has better answers than SA at the same given time to solve problems.

To show GA algorithm quality compared to SA algorithm in all problems, the 2-sample *t*-test with the certainty level of %95 ($\alpha = 0.05$) is given to compare NPV. The variance of samples was considered equal and the quality of the achieved answers in all the three groups, small, medium, and large, seemed better by GA

algorithm compared to SA, because the *P* value amount is larger than $\alpha = 0.05$ recognition level. Therefore, the “Zero” hypothesis is accepted and the opposite one is rejected.

In project scheduling problems, time-oriented and financial objectives have always been modeled as trade-off measures. In this paper, we tried to investigate the procedure, in which mode switching may increase projects profitability.

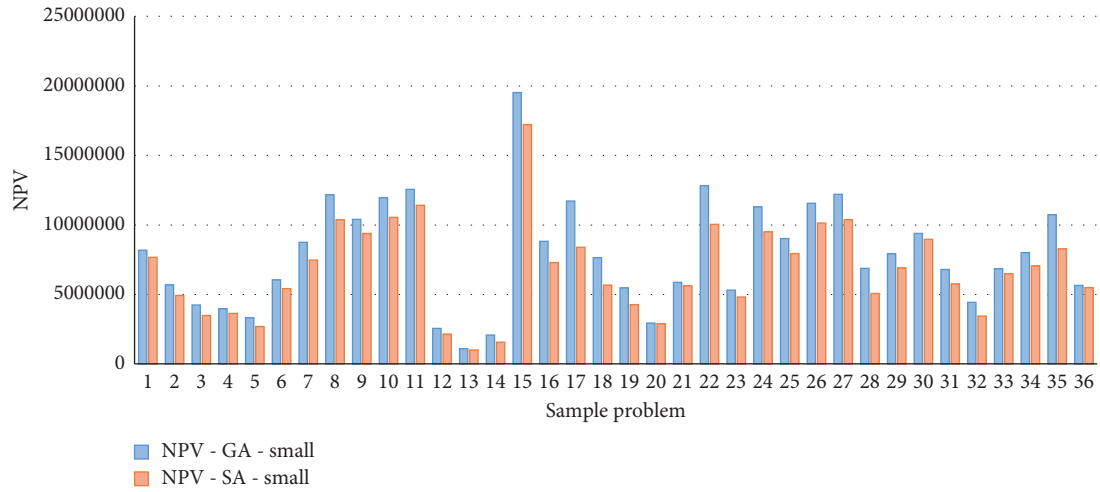


FIGURE 3: Comparison chart of net present value obtained by GA and SA algorithms in small problems.

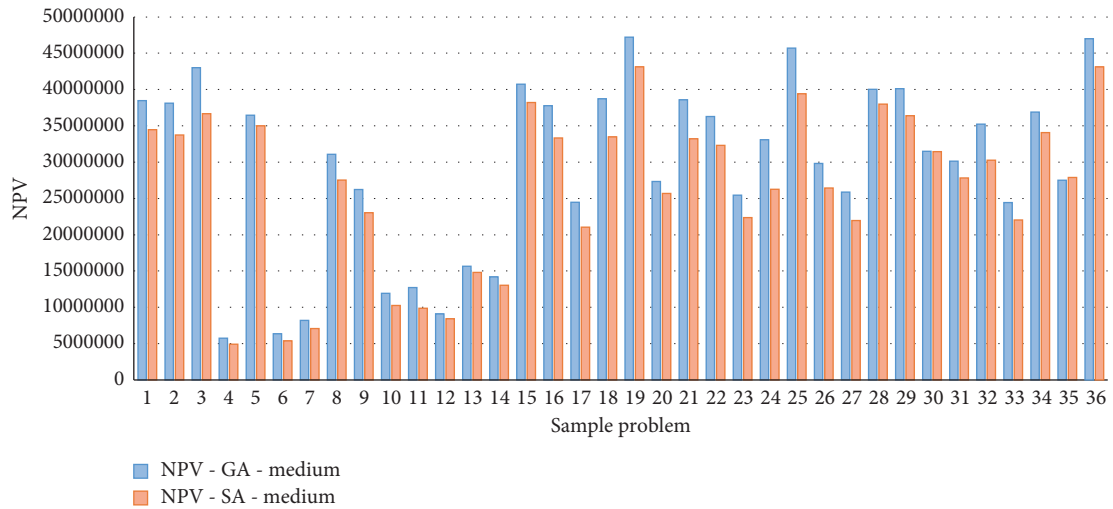


FIGURE 4: Comparison chart of net present value obtained by GA and SA algorithms in medium problems.

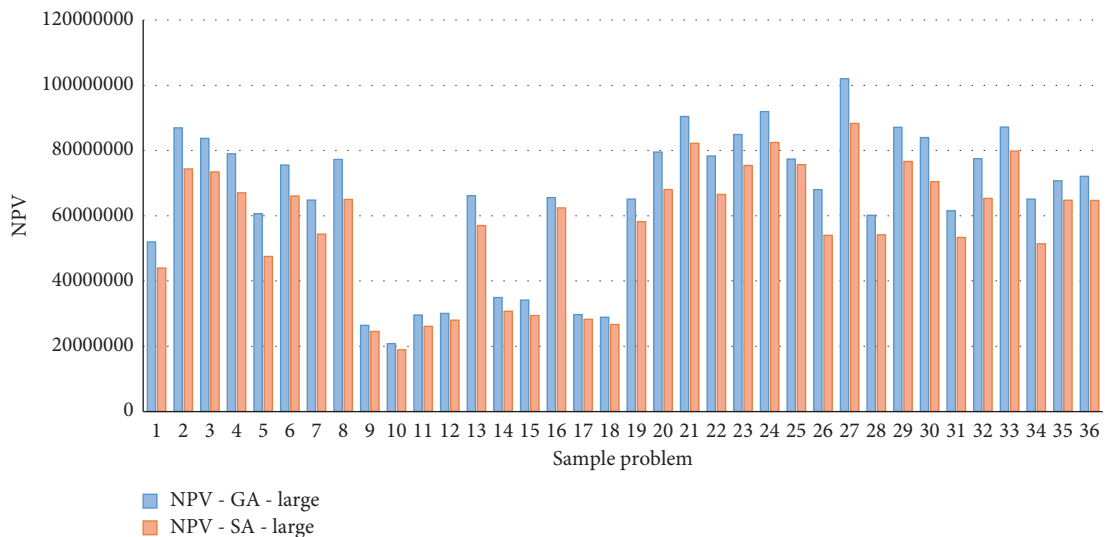


FIGURE 5: Comparison chart of net present value obtained by GA and SA algorithms in large problems.

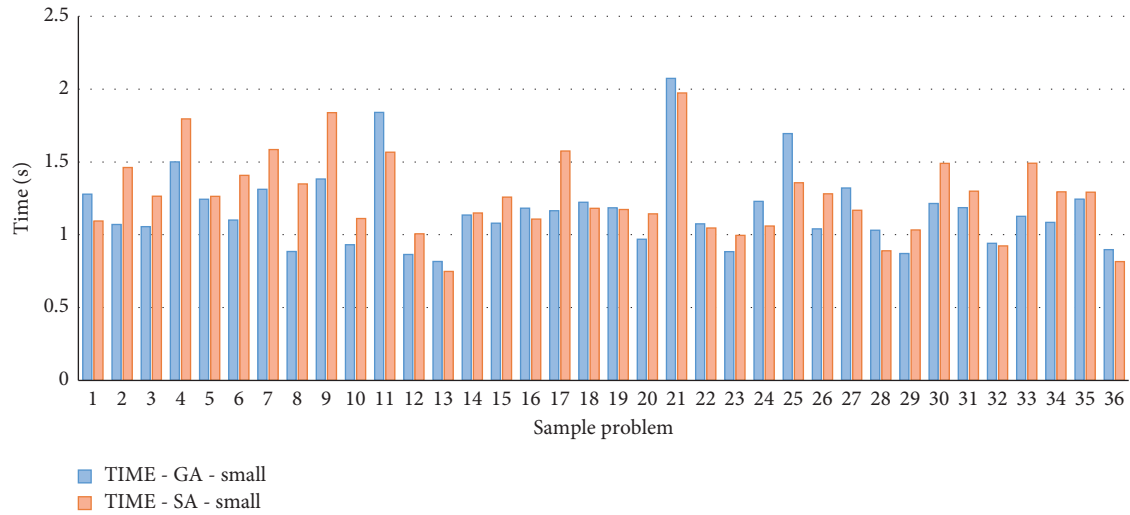


FIGURE 6: Comparison the outcomes of running time of small size algorithm between GA and SA algorithms.

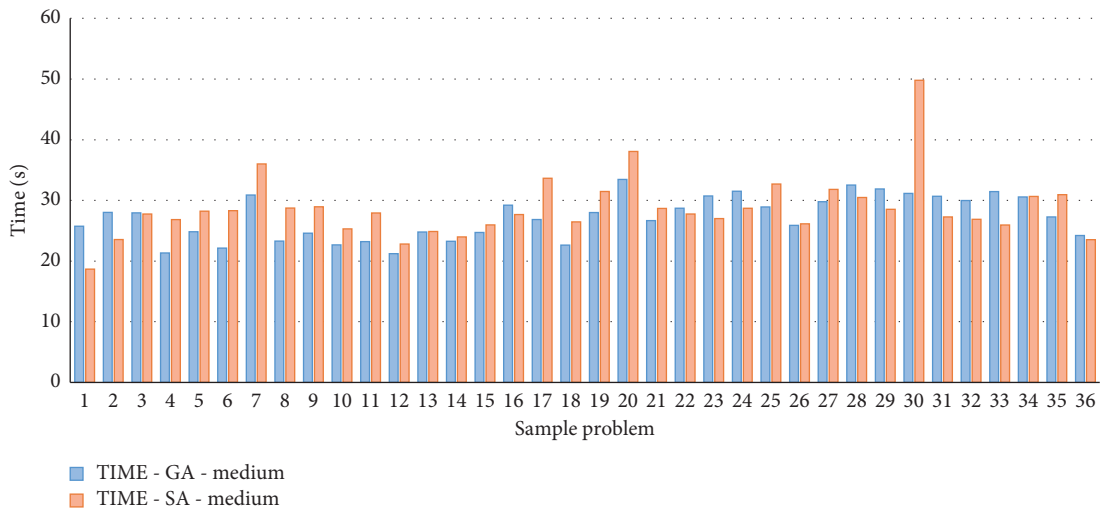


FIGURE 7: Comparison the outcomes of running time of medium size algorithm between GA and SA algorithms.

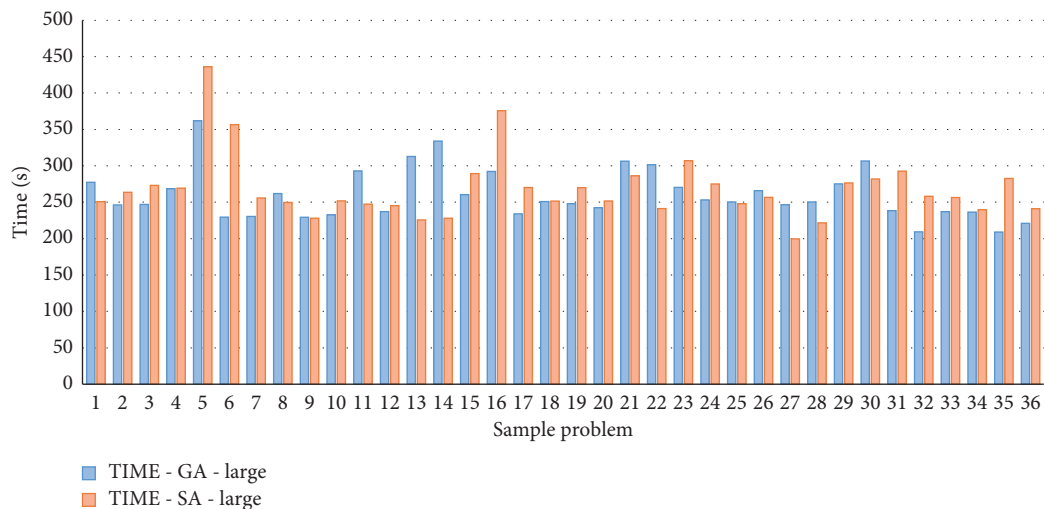


FIGURE 8: Comparison the outcomes of running time of large size algorithm between GA and SA algorithms.

5. Conclusion

This paper proposes the use of metaheuristic algorithm to solve project scheduling problems with activities along with the possibility of preemption and change mode with a bonus penalty structure. The problem has complexity in calculation time, and in other words, it is classified as NP-hard problems. In this research, genetic metaheuristic algorithm and simulated annealing algorithm were used for optimized scheduling. In order to compare the two presented algorithms, 108 sample problems were studied in this paper. The performance of the procedures was tested on an available test set, which shows genetic algorithm is more efficient than simulation annealing at the given time. For further research, it is recommended that studying the other metaheuristic algorithms can be considered to develop the desired model. Furthermore, there should be a model to discuss the problem from the viewpoints of both the contractor and employer. This is just the multiobjective model which can focus on maximizing NPV, resource restriction, and minimizing the project time.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request. It is worth mentioning that the used data are based on hypothetical values considering different real-world projects.

Conflicts of Interest

The authors declare that there were no conflicts of interest.

References

- [1] R. C. Grinold, "The payment scheduling problem," *Naval Research Logistics Quarterly*, vol. 19, no. 1, pp. 123–136, 1972.
- [2] A. H. Russell, "Cash flows in networks," *Management Science*, vol. 16, no. 5, pp. 357–373, 1970.
- [3] R. A. Russell, "A comparison of heuristics for scheduling projects with cash flows and resource restrictions," *Management Science*, vol. 32, no. 10, pp. 1291–1300, 1986.
- [4] B. Kazaz and C. Sepil, "Project scheduling with discounted cash flows and progress payments," *Journal of the Operational Research Society*, vol. 47, no. 10, pp. 1262–1272, 1996.
- [5] G. Ulusoy and S. Cebelli, "An equitable approach to the payment scheduling problem in project management," *European Journal of Operational Research*, vol. 127, no. 2, pp. 262–278, 2000.
- [6] N. Dayanand and R. Padman, "A two stage search heuristic for scheduling payments in projects," *Annals of Operations Research*, vol. 102, no. 1/4, pp. 197–220, 2001.
- [7] M. Mika, G. Waligóra, and J. Węglarz, "Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models," *European Journal of Operational Research*, vol. 164, no. 3, pp. 639–668, 2005.
- [8] A. M. Elazouini and F. G. Metwally, "Finance-based scheduling: tool to maximize project profit using improved genetic algorithms," *Journal of Construction Engineering and Management*, vol. 131, no. 4, pp. 400–412, 2005.
- [9] Z. He and Y. Xu, "Multi-mode project payment scheduling problems with bonus-penalty structure," *European Journal of Operational Research*, vol. 189, no. 3, pp. 1191–1207, 2008.
- [10] Z. He, N. Wang, T. Jia, and Y. Xu, "Simulated annealing and tabu search for multi-mode project payment scheduling," *European Journal of Operational Research*, vol. 198, no. 3, pp. 688–696, 2009.
- [11] M. Ranjbar, "An optimal NPV project scheduling with fixed work content and payment on milestones," *International Journal of Industrial Engineering & Production Research*, vol. 22, no. 3, pp. 181–186, 2011.
- [12] F. Fathollahi and A. A. Najafi, "A hybrid genetic algorithm to maximize net present value of project cash flows in resource-constrained project scheduling problem with fuzzy parameters," *Scientia Iranica*, vol. 23, no. 4, pp. 1893–1903, 2015.
- [13] P. Leyman and M. Vanhoucke, "Capital- and resource-constrained project scheduling with net present value optimization," *European Journal of Operational Research*, vol. 256, no. 3, pp. 757–776, 2016.
- [14] M. Nasrabadi and A. Mirzazadeh, "The inventory system management under uncertain conditions and time value of money," *International Journal of Supply and Operations Management*, vol. 3, no. 1, pp. 1192–1214, 2016.
- [15] P. Leyman and M. Vanhoucke, "Payment models and net present value optimization for resource-constrained project scheduling," *Computers & Industrial Engineering*, vol. 91, pp. 139–153, 2016.
- [16] J. Hassanpour, M. Ghodoosi, and Z. Sadat Hosseini, "Optimizing a Bi-objective preemptive multi-mode resource-constrained project scheduling problem: NSGA-II and MOICA algorithms," *Journal of Optimization in Industrial Engineering*, vol. 10, no. 21, pp. 79–92, 2017.
- [17] H. Farughi, A. Amiri, and F. Abdi, "Project scheduling with simultaneous optimization, time, net present value, and project flexibility for multimode activities with constrained renewable resources," *International Journal of Engineering*, vol. 31, no. 5, pp. 780–791, 2018.
- [18] S. R. Alavipour and D. Arditi, "Optimizing financing cost in construction projects with fixed project duration," *Journal of Construction Engineering and Management*, vol. 144, no. 4, Article ID 04018012, 2018.
- [19] S. M. R. Alavipour and D. Arditi, "Maximizing expected contractor profit using an integrated model," *Engineering, Construction and Architectural Management*, vol. 26, no. 1, pp. 118–138, 2019.
- [20] Y. Liang, N. Cui, T. Wang, and E. Demeulemeester, "Robust resource-constrained max-NPV project scheduling with stochastic activity duration," *OR Spectrum*, vol. 41, no. 1, pp. 219–254, 2019.
- [21] A. Delgoshaei, S. Esmaeili Hanjani, and A. Hossein Nasiri, "A genetic algorithm for scheduling multimode resource-constrained project problem in the presence of preemptive resources," *Journal of Project Management*, vol. 4, no. 3, pp. 195–212, 2019.
- [22] M. Zarei, R. Kamranrad, and Emami, "Project scheduling optimization for contractor's Net present value maximization using meta-heuristic algorithms: a case study," *Journal of Industrial Engineering and Management Studies*, vol. 7, no. 2, pp. 36–55, 2020.
- [23] C.-C. Hsu and D. S. Kim, "A new heuristic for the multi-mode resource investment problem," *Journal of the Operational Research Society*, vol. 56, no. 4, pp. 406–413, 2005.
- [24] G. Zhu, J. F. Bard, and G. Yu, "A branch-and-cut procedure for the multimode resource-constrained project-scheduling

- problem,” *Informs Journal on Computing*, vol. 18, no. 3, pp. 377–390, 2006.
- [25] M. Sabzehparvar and S. M. Seyed-Hosseini, “A Mathematical model for the multi-mode resource-constrained project scheduling problem with mode dependent time lags,” *The Journal of Supercomputing*, vol. 44, no. 3, pp. 257–273, 2007.
- [26] M. Sabzehparvar, S. M. Seyed-Hosseini, and S. Nouri, “A mathematical model for the multi-mode resource investment problem,” *Journal of Industrial Engineering International*, vol. 4, no. 7, pp. 25–32, 2008.
- [27] V. V. Peteghem and M. Vanhoucke, “A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem,” *European Journal of Operational Research*, vol. 201, no. 2, pp. 409–418, 2010.
- [28] F. Azimi, R. Aboutalebi, and A. A. Najafi, “Using multi-objective particle swarm optimization for Bi-objective multi-mode resource-constrained project scheduling problem,” *International Journal of Industrial and Manufacturing Engineering*, vol. 5, no. 6, 2011.
- [29] B. Afshar-Nadjafi, “Multi-mode resource availability cost problem with recruitment and release dates for resources,” *Applied Mathematical Modelling*, vol. 38, no. 21-22, pp. 5347–5355, 2014.
- [30] J.-J. Qi, Y.-J. Liu, P. Jiang, and B. Guo, “Schedule generation scheme for solving multi-mode resource availability cost problem by modified particle swarm optimization,” *Journal of Scheduling*, vol. 18, no. 3, pp. 285–298, 2015.
- [31] E. Pérez, M. Posada, and A. Lorenzana, “Taking advantage of solving the resource constrained multi-project scheduling problems using multi-modal genetic algorithms,” *Soft Computing*, vol. 20, no. 5, pp. 1879–1896, 2016.
- [32] R. Nemati-Lafmejani, H. Davari-Ardakani, and H. Najafzad, “Multi-mode resource constrained project scheduling and contractor selection: mathematical formulation and meta-heuristic algorithms,” *Applied Soft Computing*, vol. 81, 2019.
- [33] M. Ehsanifar, N. Hamta, and H. Soltani, “A Preemptive multimode resource constrained project scheduling model with cash flows,” *Journal of Industrial and Systems Engineering*, vol. 12, pp. 28–44, 2019.
- [34] W. Chiu-Chi, L. Ping-Hung, and T. Ying-Chin, “Resource-constrained project management using enhanced theory of constraint,” *International Journal of Project Management*, vol. 20, no. 7, pp. 561–567, 2002.
- [35] J. Józefowska, M. Mika, R. Różycki, G. Waligóra, and J. Węglarz, “A heuristic approach to allocating the continuous resource in discrete-continuous scheduling problems to minimize the makespan,” *Journal of Scheduling*, vol. 5, no. 6, pp. 487–499, 2002.
- [36] B. Afshar-Nadjafi, M. Yazdani, and M. Majlesi, “A hybrid of tabu search and simulated annealing algorithms for preemptive project scheduling problem,” in *Advances in Artificial Intelligence: From Theory to Practice. IEA/AIE*, S. Benferhat, K. Tabia, and M. Ali, Eds., vol. 10350, pp. 102–111, Springer, Heidelberg, Germany, 2017.
- [37] A. Delgoshaei, T. Rabczuk, A. Ali, and M. K. A. Ariffin, “An applicable method for modifying over-allocated multi-mode resource constraint schedules in the presence of preemptive resources,” *Annals of Operations Research*, vol. 259, no. 1-2, pp. 85–117, 2017.
- [38] B. Afshar-Nadjafi, “A solution procedure for preemptive multi-mode project scheduling problem with mode changeability to resumption,” *Applied Computing and Informatics*, vol. 14, no. 2, pp. 192–201, 2018.