

Research Article

Preprocessing the Discrete Time-Cost Tradeoff Problem with Generalized Precedence Relations

Hanying Wei,^{1,2} Zhixiong Su ,² and Yuan Zhang ,^{3,4}

¹Research Center of Management Science and Engineering, Jiangxi Normal University, Nanchang 330022, China

²Business Administration College, Nanchang Institute of Technology, Nanchang 330099, China

³School of Finance and Economics, Shenzhen Institute of Information Technology, Shenzhen 518172, China

⁴Institute of Information Economy, Shenzhen Institute of Information Technology, Shenzhen 518172, China

Correspondence should be addressed to Yuan Zhang; 63443848@qq.com

Received 3 July 2019; Revised 22 December 2019; Accepted 14 March 2020; Published 27 April 2020

Academic Editor: Przemyslaw Ignaciuk

Copyright © 2020 Hanying Wei et al. 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.

This study investigates the deadline of the discrete time-cost tradeoff problem (DTCTP-D) with generalized precedence relations (GPRs). This problem requires modes to be assigned to the activities of a project such that the total cost is minimized and the total completion time and the precedence constraints are satisfied. Anomalies under GPRs are irreconcilable with many current theories and methods. We propose a preprocessing technology, an equivalent simplification approach, which is an effective method for solving large-scale complex problems. We first study a way to deal with the anomalies under GPRs, such as the reduce (increase) in project completion as a consequence of prolonging (shortening) an activity, and discover relationships between time floats and path lengths. Then, based on the theories, we transform the simplification into a time float problem and design a polynomial algorithm. We perform the simplification and improve the efficiency of the solution by deleting redundant calculation objects.

1. Introduction

Project management is designed to manage or control company resources on a given activity, within time, cost, and other constraints. Very few projects are completed without incurring tradeoffs on these constraints [1]. A successful project manager needs to maintain a balance between them such that the outcome of the project is not compromised. However, keeping the balance and tradeoffs between these objectives is inherently difficult. The time-cost tradeoff problem (TCTP) is a special version that addresses both discrete and continuous time-cost relationships. The TCTP in this paper is limited to the discrete version, namely, the discrete time-cost tradeoff problem (DTCTP) [2]. The DTCTP considers a limited number of time-cost alternatives, so-called modes, for each activity, and the aim is to select a mode for each activity so that the desired project goals are reached. De et al. [3] proved that the DTCTP is a strong \mathcal{NP} -hard optimization problem for general activity networks.

Three versions of the DTCTP have been studied in the literature so far: the deadline problem (DTCTP-D), the budget problem (DTCTP-B), and the efficiency problem (DTCTP-E) [4]. In DTCTP-D, given a set of modes (time-cost pairs) for each activity, the objective is to select a mode for each activity so that the total cost is minimized while meeting a given project deadline. Conversely, the budget problem minimizes the project duration while not exceeding a given budget. DTCTP-E is the problem of constructing efficient time-cost solutions over the set of feasible project durations [4]. This paper concentrates on the deadline version.

The basic DTCTP-D deals with scheduling project activities' modes subject to finish-to-start precedence constraints with zero time lags (strict precedence relation). When this kind of temporal constraints are taken into account, an activity can start only as soon as all its predecessors have finished. However, in a project, it can be necessary to specify other kinds of temporal constraints besides the strict

precedence relation. Roy [5] and Elmaghraby [6] denoted such constraints as Generalized Precedence Relations (GPRs). GPRs mandate the start/finish times of a pair of activities in a project that must be separated by a minimum or maximum amount of time referred to as a time lag. In this paper, we study the DTCTP-D with GPRs.

For the DTCTP, the amount of schemes of the problem will increase exponentially as the scale of the problem increases, which causes the computational demand to increase greatly. Conversely, if the amount of activities is reduced, the number of schemes will decrease exponentially, which means that a large-scale DTCTP may be solved by using exact algorithms. For the TCTP-D, we summarize a feature that matching this idea of reducing activities. Fulkerson [7] first proposed an exact network flow computation for the continuous TCTP-D, and the procedure is to compress activities from their longest durations. Of special interest is when observing the optimal solutions obtained by Fulkerson algorithm [7] in a number of TCTP-D instances, we discover only a few activities with compressed durations and most others keeping the original longest durations. Then, we further observe the optimal solutions of DTCTP instances and find the similar result and feature, even for the DTCTP with GPRs. For example, in [8], the optimal solution of a numerical example shows that 24 activities in total 29 ones choose their longest modes; therefore, maybe, obtaining the optimal solution is equivalent to identifying the optimal modes of the other 5 activities. The result enlightens us that maybe only a few activities need to be considered and computed, and if so, the computation for the DTCTP-D could be reduced substantially. The key of computation reduction is to identify the few activities with compression requirements. In this study, we focus on the preprocessing technology to simplify the DTCTP-D with GPRs.

The paper is organized as follows. Related work is reviewed in Section 2. Section 3 contains a mathematical programming model formulation for the DTCTP-D with GPRs, and Section 4 contains the representations of GPRs and computations of activity time parameters under GPRs. Section 5 presents the equivalent simplification for the DTCTP-D with GPRs and the model that employs the characteristics of time parameters under GPRs. A report on in-depth computational studies is provided in Section 6. Finally, discussing and concluding remarks in Sections 7 and 8 finish the paper.

2. Related Works

The importance of the time-cost tradeoff problem was brought to the attention of the research community almost 50 years ago [7, 9, 10], and moreover the works for GPRs were started nearly 70 years ago [5, 6]; however, still, due to the inherent computational complexity, research in the DTCTP with GPRs is rather sparse. We view the literature studies on the DTCTP and GPRs, respectively.

2.1. DTCTP. The methods proposed for the DTCTP could be categorized into three areas: exact methods, heuristics,

and metaheuristics. The global optimum solution is the ultimate aim of the time-cost tradeoff, and exact algorithms are required to compute this solution. Generally, the exact algorithms rely on optimization models. For exactly describing the DTCTP, the mixed-integer programming optimization models are mainly applied [8, 11–15], and recently, Klanšek and Pšunder [8] presented a mixed-integer nonlinear programming optimization model for the nonlinear DTCTP.

The exact methods for the DTCTP mainly contain the branch-and-bound-based algorithms [16–19], cutting plane algorithm [20], dynamic programming algorithm [2, 21], and column generation method [22]. However, all three types of DTCTP are *NP*-hard in the strong sense [3]. Hence, the exact algorithms are limited to solve small and medium scale instances. In times earlier than 2005, instances with (depending on the structure of the network and the number of processing alternatives per activity) no more than 20–50 activities can be solved to optimality in reasonable amount of time [22]. In particular, Vanhoucke [23] concluded that problem with time-switch constraint instances with 20 or 30 activities, and four or more modes are very difficult to solve and have to rely on heuristic procedures. In the times, directly computing the DTCTP mainly relied on the branch-and-bound algorithms, and we could view the efficiencies in computational experiments. Demeulemeester et al. [18] solved instances up to 50 activities, while Vanhoucke [23] achieved optimal solutions for instances of the problem with time-switch constraints up to 30 activities with a fixed number (2 or 4) of modes. However, after 2005, authors could directly compute the optimal solutions of the DTCTP instances with far more than 50 activities. Hadjiconstantinou and Klerides [20] proposed a new path-based cutting plane approach for the DTCTP, which can provide optimal solutions of the DTCTP instances with more than 100 activities and many modes per activity within short computational time. Some authors studied other methods based on models. Szmerekovsky and Venkateshan [11] studied four integer programming formulations for the DTCTP and achieved optimal solutions for instances with up to 90 activities. Klanšek and Pšunder [8] further considered the nonlinear DTCTP and presented a mixed-integer nonlinear programming optimization model which yields the exact optimum solution of the nonlinear DTCTP. Due to the much more complexity, they used the approach to solve the instances up to 29 activities with 3 modes.

In order, for the optimal solutions of the larger-size DTCTP instances, many authors research the problems from other perspectives, and most of them focused on the decomposition. For example, the dynamic programming formulation for the problem may exploit the decomposition structure of the project network [2, 3, 21, 22, 24]. The decomposition that facilitates the computation is known as modular decomposition or substitution decomposition and has many applications in network and other combinatorial optimization problems. This is carried out by “transforming” such an indecomposable project network to a series-parallel network and then performing the “easy” calculations for the series-parallel case. De et al. [16], Erengüç et al. [25], and

Demeulemeester et al. [24] provided the early implementations of this approach. Demeulemeester et al. [24] reported on computational experience for networks with up to 45 activities, while Erengüç et al. [25] developed a Benders decomposition approach to optimality and tested on projects with up to 64 activities. These results are excellent in 1990s. Akkan et al. [22] used network decomposition to optimally solve a set of series of subnetworks of the DTCTP, and the results were reported for large and hard instances including project networks with up to 138 activities and 20 modes per activity. Hazır et al. [4] presented an exact method based on Benders decomposition for the DTCTP-D and were able to solve instances including up to 136 activities and 10 modes within 90 min.

In addition, a few authors considered the direct reduction from a large-size DTCTP to a small-size one [16, 22, 24]. The approach could be seen as a special decomposition but may have more powerful function once realized. Akkan et al. [22] discussed techniques which help to eliminate some modes of activities to simplify the DTCTP. The work is productive but there are still some bottlenecks to overcome. For example, their procedure needs to determine all paths which have a longest possible path length that exceeds the deadline, but it may be difficult to compute and identify the numerous paths in a large network. To the best of our knowledge, there is no computational experience to test the approach because of these defects. We are interested in the idea and try to overcome the difficulties, which is one work in this paper.

Although the decomposition/reduction technologies could assist the exact method to solve the larger-size DTCTP, for the problem with more than 200 activities and more modes per activity, we have to use heuristics and metaheuristics to compute near-optimal solutions. The current research on heuristics and metaheuristics is focused on the ant colony algorithm [26], genetic algorithm [12, 13, 15, 27], simulated annealing algorithm [28], particle swarm algorithm [29, 30], variable neighborhood search [31], tabu search [31], Network analysis algorithm [32], and memetic algorithm [33]. Some authors also presented other approximation methods, such as “natural” linear relaxation [34, 35], Lagrange relaxation [22], and bicriteria approximation algorithms [36]. The computational experiences have tested the efficiencies of heuristics. Vanhoucke and Debels [37] proposed a metaheuristic approach for the DTCTP with time/switch constraints, work continuity constraints, and net present value maximization, and their computational experiences generated project with up to 200 activities, with the number of modes generated randomly from the interval [1, 11, 38]. Kandil and El-Rayes [39], and Bettemir [40] solved large-sized DTCTP instances earlier using metaheuristics; however, their computations required long time periods. Kandil and El-Rayes [39] computed instances up to 720 activities in 21 h. Bettemir [40] tested one of the best results until now that achieved solutions of instances including 630 activities with a 2 percent deviation from the optimum in 73 min. Furthermore, some authors proposed more advanced heuristic algorithms. Sonmez and Bettemir [13] developed a hybrid genetic algorithm for the

DTCTP-D and tested it on problem instances with up to 630 activities, while Li et al. [41] proposed a biobjective hybrid genetic algorithm for the DTCTP-C and tested it on problem instances with up to 500 activities with fixed 8 modes or [9, 27] modes. Recently, Leyman et al. [42] employed an iterated local search metaheuristic for the DTCTP with multiple cash flow and payment models and tested it on problem instances with up to 100 activities with fixed 6 modes.

However, heuristics mainly focus on the local optimality and near-optimality and may not always converge to the global optimum, whereas metaheuristics require significant computation to converge to the global optimum. Vanhoucke and Debels [37] revealed that even a truncated version of the branch-and-bound procedure still outperforms their metaheuristic procedures. Some works could be used to evaluate such heuristics and approximations, such as the work on the lower bound. Akkan et al. [22] proved a stronger lower bound for the DTCTP using column generation techniques.

In practice, it is common that a project will most likely consist of hundreds of activities [43]. For the optimal solutions of large-size DTCTP instances, decomposition/reduction approaches may be better choices, and one target of this paper is a more effective technology (polynomial algorithm) of decomposition/reduction following our idea that identify the activities requiring compressions.

2.2. GPRs. After the original work of Roy [5] and Elmaghraby [6], the first systemic treatment of GPRs is due to Kerbosch and Schell [44]. The GPRs are extension of the traditional strict precedence relation between activities in a project, which include the minimal/maximal time lags. However, the representations and time parameters of the former are significantly different from the latter.

For the representations of GPRs, Kerbosch and Schell [44], IBM [45], Crandall [46], and Neumann and Schwindt [47] modelled projects with GPRs by activity-on-node (AoN) networks, while Elmaghraby and Kamburowski [48] further developed an improved activity-on-arc (AoA) network under GPRs. There are negative length arcs and cycles in the activity network under GPRs, which are unallowed in the representation of strict precedence relation, such as the Critical Path Method (CPM) network [10]. De Reyck and Herroelen [38] declared that the networks that include activities among which minimal and maximal time lags exist can generate infeasibilities in the project. However, with considering only minimal time lags between activities, CPM can easily be extended to GPRs, that is, no cycles may occur in the network with only case of minimal time lags between activities.

The time parameters of activities are computed based on the Precedence Diagramming Method (PDM), and the abovementioned authors working on the representations of GPRs also proposed algorithms for the time parameters under the GPRs. In addition, some authors considered the extension of general GPRs [49]. Leachman et al. [50], Kis et al. [51], and Kis [52, 53] considered the intensity of the activities using the “feeding precedence” constraints. Bianco

and Caramia [54–56] classified the intensity as %Completed-to-start, Start-to-%Completed, Finish-to-%Completed, and %Completed-to-Finish. Crandall [46], Moder et al. [57], Valls et al. [58], Hajdu [59], Kis [53], and Ponz-Tienda et al. [60] considered the splitting of activities and proposed and developed computations for the splitting parameters under GPRs. There are still other new approaches to face the problem of GPRs particularly the overlapped activities, such as Beeline Diagramming Method [61], Design Structure Matrix [62], Collaborative Product Development Process [63], and Concurrency Scheduling [64].

However, the PDM with GPRs presents anomalous effects that are counterintuitive about the consequences of lengthening or shortening a job [48, 65–67], changing the concept of the critical path itself. Crandall [46] and Elmaghraby and Kamburowski [48] analyzed the anomalous effect called “reverse criticality” that the reduction (increase) in project completion time is a consequence of prolonging (shortening) the duration of a critical activity. Recently, Qi and Su [68] and Su and Wei [69] discovered the phenomenon that time floats increase following consumptions. Su [70] further found that the activity floats can be consumed and enlarged imperceptibly even if the activity has not started and all activities do not consume their time floats. These anomalies challenge the foundations and principles of project scheduling with GPRs including the TCTP with GPRs.

To the best of our knowledge, the literature on the DTCTP with GPRs is nearly scarce. Sakellariopoulos and Chassiakos [71] paid early attention to the DTCTP with GPRs and formulated a linear/integer program. Son et al. [72] developed a mixed (continuous + discrete) time-cost tradeoff model considering GPRs to represent the project more realistically. Klanšek and Pšunder [8] considered the nonlinear DTCTP with GPRs and presented a mixed-integer nonlinear programming optimization model. In these literature studies, the GPRs constraints are added to the DTCTP model, but paid no special attention to the GPRs and characteristics. Obviously, the generalized DTCTP models have not concerned the time parameters of activities; therefore, they cannot reflect the influences of the anomalies of GPRs to the time parameters. Maybe, it is a reason that there are many literature studies on the DTCTP but few on the problem with GPRs. All the same, a few authors considered the anomalies when solving the TCTP with GPRs. For avoiding the influences of the anomalies, Elmaghraby and Kamburowski [48] proposed a method liking Fulkerson algorithm but executing activities from their shortest durations. Although they only considered the continuous TCTP with GPRs, the contribution enlightens us to handle the discrete version. However, the decomposition/reduction based on the project networks for the DTCTP is heavily reliant upon the time parameters of activities. Hence, it is hard to realize a decomposition/reduction for the DTCTP with GPRs until the anomalies of GPRs are handled enough.

Although few authors researched the DTCTP with GPRs specially, there are relative more works on the problem considering other objectives for trading-off, and most focused on the discrete time-cost-quality tradeoff problem

(DTCQTP) with GPRs. The related contributions also contain the modellings and algorithms. Most authors used mixed-integer mathematical formulations to model the DTCQTP with GPRs [73, 74], but different algorithms to solve the problem [73–75]. Amiri et al. [73] developed a modified nondominated sorting genetic algorithm-II, while Tavana et al. [74] proposed an efficient version of the ε -constraint method and a dynamic self-adaptive version of a multiobjective evolutionary computation. Khalili-Damghani et al. [75] compared the performance of several methods including the epsilon-constraint, multistart partial bound enumeration algorithm, and dynamic self-adaptive multiobjective particle swarm optimization, and the computational experiments showed that the last one outperforms the others. They discovered and analyzed the time windows of an activity generated by the GPRs; however, liking the DTCTP with GPRs, their works on the DTCQTP with GPRs still did not concerned the time parameters of activities and neglected the influences of the anomalies. A few authors made a try to introduce the time parameters of activities to the models and computations for improved solutions of the TCQTP. Afruzi et al. [76] proposed a model and algorithm for the DTCQTP. Although they considered the problem on with one-type precedence relation, the frame and process of the proposed model and algorithm are bounded by the earliest/latest-start/finish times of activities, and the computational outcomes showed the competitive effects. Maybe the work will be useful to develop the DTCQTP with GPRs. Mohammadipour and Sadjadi [77] further provided a tradeoff between “project total extra cost,” “project total risk enhancement,” and “project total quality reduction” to shorten the overall project duration with GPRs. In their model, they set the “earliest-start time” and “latest-finish time” of an activity as variables, which could reflect a degree of the influences of the anomalies of GPRs.

In order to obtain the optimal solutions of larger-size DTCTP with GPRs instances, in this paper, we prefer to the decomposition/reduction for the problem, that is, the equivalent simplification. The related literature studies of DTCTP and our previous works have shown that the time parameters of activities are very useful to the simplification effect. Therefore, we should provide measures to handle the anomalies under GPRs and develop a procedure for the equivalent simplification of the DTCTP with GPRs, keeping away from the anomalies.

3. Problem Formulation and Notation

Consider a project composed of different activities linked by precedence relations. Each activity i in the project can be executed in $k \in r_i$ different modes and each mode m_k has its own duration (d_i^k) and cost (c_i^k).

3.1. Generalized Precedence Relations. Real-life project scheduling problems involve several types of precedence relations among the project activities. GPRs include the following types (s_i and f_i denote the start and finish time of an activity i , respectively).

3.1.1. Finish-to-Start Type of Minimum Time Lag. This time lag, $FTS_{ij}^{\min}(\lambda_{ij})$, defines that the start time of an activity j occurs no earlier than λ_{ij} units after the finish time of an activity i ; that is,

$$f_i + \lambda_{ij} \leq s_j. \quad (1)$$

3.1.2. Finish-to-Finish Type of Minimum Time Lag. This time lag, $FTF_{ij}^{\min}(\lambda_{ij})$, defines that the finish time of an activity j occurs no earlier than λ_{ij} units after the finish time of an activity i ; that is,

$$f_i + \lambda_{ij} \leq f_j. \quad (2)$$

3.1.3. Start-to-Start Type of Minimum Time Lag. This time lag, $STS_{ij}^{\min}(\lambda_{ij})$, defines that the start time of an activity j occurs no earlier than λ_{ij} units after the start time of an activity i ; that is,

$$s_i + \lambda_{ij} \leq s_j. \quad (3)$$

3.1.4. Start-to-Finish Type of Minimum Time Lag. This time lag, $STF_{ij}^{\min}(\lambda_{ij})$, defines that the finish time of an activity j occurs no earlier than λ_{ij} units after the start time of an activity i ; that is,

$$s_i + \lambda_{ij} \leq f_j. \quad (4)$$

3.1.5. Finish-to-Start Type of Maximum Time Lag. This time lag, $FTS_{ij}^{\max}(\lambda_{ij})$, defines that the start time of an activity j occurs no later than λ_{ij} units after the finish time of an activity i ; that is,

$$f_i + \lambda_{ij} \geq s_j. \quad (5)$$

3.1.6. Finish-to-Finish Type of Maximum Time Lag. This time lag, $FTF_{ij}^{\max}(\lambda_{ij})$, defines that the finish time of an activity j occurs no later than λ_{ij} units after the finish time of an activity i ; that is,

$$f_i + \lambda_{ij} \geq f_j. \quad (6)$$

3.1.7. Start-to-Start Type of Maximum Time Lag. This time lag, $STS_{ij}^{\max}(\lambda_{ij})$, defines that the start time of an activity j occurs no later than λ_{ij} units after the start time of an activity i ; that is,

$$s_i + \lambda_{ij} \geq s_j. \quad (7)$$

3.1.8. Start-to-Finish Type of Maximum Time Lag. This time lag, $STF_{ij}^{\max}(\lambda_{ij})$, defines that the finish time of an activity j occurs no later than λ_{ij} units after the start time of an activity i ; that is,

$$s_i + \lambda_{ij} \geq f_j. \quad (8)$$

To consider GPRs, the set of precedence relations, E , is partitioned into eight subsets denoted $E_{STS^{\min}}$, $E_{STS^{\max}}$,

$E_{STF^{\min}}$, $E_{STF^{\max}}$, $E_{FTS^{\min}}$, $E_{FTS^{\max}}$, $E_{FTF^{\min}}$, and $E_{FTF^{\max}}$, respectively, for the sets of STS^{\min} , STS^{\max} , STF^{\min} , STF^{\max} , FTS^{\min} , FTS^{\max} , FTF^{\min} , and FTF^{\max} precedence relations.

3.2. Problem Formulation. We consider the deadline problem of DTCTP (DTCTP-D) with GPRs. The objective faced by a decision maker is to minimize the project total cost with a given due date T . The project total cost in the DTCTP-D mainly means the sum of direct activity costs in much of the literature [16, 74]. In general, the direct activity cost is higher when the activity duration is shorter. The problem is then to select a mode for each project activity and generate a set of nondominated solutions. The DTCTP-D with GPRs could be formulated based on the formulations presented by Taviana et al. [74]. Assume that there are n activities in the project:

$$\min \sum_{i=1}^n \sum_{k=1}^{r_i} \delta_i^k c_i^k, \quad (9)$$

s.t.

$$\sum_{k=1}^{r_i} \delta_i^k = 1, \quad i \in V, \quad (10)$$

$$s_i + \sum_{k=1}^{r_i} d_i^k \delta_i^k + \lambda_{ij} \leq s_j, \quad \forall i, j \in E_{FTS^{\min}}, \quad (11)$$

$$s_i + \sum_{k=1}^{r_i} d_i^k \delta_i^k + \lambda_{ij} \geq s_j, \quad \forall i, j \in E_{FTS^{\max}}, \quad (12)$$

$$s_i + \lambda_{ij} \leq s_j + \sum_{k=1}^{r_i} d_i^k \delta_i^k, \quad \forall i, j \in E_{STF^{\min}}, \quad (13)$$

$$s_i + \lambda_{ij} \geq s_j + \sum_{k=1}^{r_i} d_i^k \delta_i^k, \quad \forall i, j \in E_{STF^{\max}}, \quad (14)$$

$$s_i + \lambda_{ij} \leq s_j, \quad \forall i, j \in E_{STS^{\min}}, \quad (15)$$

$$s_i + \lambda_{ij} \geq s_j, \quad \forall i, j \in E_{STS^{\max}}, \quad (16)$$

$$s_i + \sum_{k=1}^{r_i} d_i^k \delta_i^k + \lambda_{ij} \leq s_j + \sum_{k=1}^{r_j} d_j^k \delta_j^k, \quad \forall i, j \in E_{FTF^{\min}}, \quad (17)$$

$$s_i + \sum_{k=1}^{r_i} d_i^k \delta_i^k + \lambda_{ij} \geq s_j + \sum_{k=1}^{r_j} d_j^k \delta_j^k, \quad \forall i, j \in E_{FTF^{\max}}, \quad (18)$$

$$s_i + \sum_{k=1}^{r_i} d_i^k \delta_i^k \leq T, \quad \forall i \in V, \quad (19)$$

$$s_i \geq 0, \quad \forall i \in V, \quad (20)$$

$$\delta_i^k \in \{0, 1\}, \quad \forall i, k. \quad (21)$$

Relation (9) is allocated to describe budget objective functions. Constraint (10) guarantees the selection of one and only one mode for each activity. Constraints (11)–(18) preserve the GPRs between project activities. Constraint (19) restricts the project deadline. Constraint (20) ensures that no activity can start earlier than zero. Constraint (21) represents the binary form of the decision variables.

Furthermore, we formulate the DTCTP-D with GPRs considering indirect project cost based on the formulations presented by Elmaghraby and Kamburowski [48]. In contrast to the direct activity durations, the indirect project cost generally decreases, following the compression of project duration. Let e indicate the indirect project cost per date, t_i indicate the time of realization of node (i) in the activity network under GPRs with beginning node (0) and end node ($2n + 1$) [48] (see Section 3, and assume there are n activities in the project), and P indicate the set of arcs representing time lags in the network. Because the duration of activity i is $d_i = t_{2i} - t_{2i-1}$ [48], the problem of finding the cheapest project schedule with a given deadline date T may be formulated as follows:

$$\min \sum_{i=1}^n \sum_{k=1}^{r_i} \delta_i^k c_i^k + e(t_{2n+1} - t_0), \quad (22)$$

s.t.

$$\sum_{k=1}^{r_i} \delta_i^k = 1, \quad i \in V, \quad (23)$$

$$t_{2i} - t_{2i-1} = \sum_{k=1}^{r_i} d_i^k \delta_i^k, \quad i \in V, \quad (24)$$

$$t_j - t_i \geq w_{ij}, \quad (i, j) \in P, \quad (25)$$

$$t_{2n+1} - t_0 \leq T, \quad (26)$$

$$t_i \geq 0, \quad \forall (i) \in N, \quad (27)$$

$$\delta_i^k \in \{0, 1\}, \quad \forall i, k. \quad (28)$$

Relation (22) is allocated to describing budget objective functions including the direct and indirect costs. Constraints (23), (27), and (28) are similar to constraints (10), (20), and (21), respectively. Constraints (24)–(26) represent constraints (11)–(19) in another way based on the activity network under GPRs. Constraint (24) links the realization times of nodes and activity durations. Constraint (25) preserves the time lags between project activities. Constraint (26) restricts the project deadline.

4. Activity Network under GPRs

4.1. Representations of GPRs. The activity network under GPRs proposed by Elmaghraby and Kamburowski [48] is the current standard representation of GPRs. This activity

network has the following features. (1) The beginning node is (0) and the end node is ($2n + 1$), where n indicates the amount of activities, and the nodes denote the project beginning and project end, respectively. (2) An activity i is represented as arcs ($2i - 1, 2i$) and ($2i, 2i - 1$) with lengths $w_{2i-1,2i} = d_i$ and $w_{2i,2i-1} = -d_i$, where d_i indicates the activity duration. (3) A minimum (maximum) time lag is depicted as a forward (reversed) arc with a length equal to the value (negative value) of the time lag. Figures 1 and 2 represent the above-mentioned precedence relations, respectively. If we let w_{ab} indicate the length of arc (a, b), then obviously $w_{2i,2j-1} = \lambda_{ij}$ in Figure 1(a), $w_{2i,2j} = \lambda_{ij}$ in Figure 1(b), and so on.

4.2. Time Parameters under GPRs. The time parameters in an activity network under GPRs include time parameters of nodes and activities. The time parameters of a node (i) contain the minimum realization time \underline{t}_i and maximum realization time \bar{t}_i . The time parameters of an activity j contain the earliest start time ES_j , the earliest finish time EF_j , the latest start time LS_j , the latest finish time LF_j , total float TF_j , free float FF_j , and safety float SF_j .

We present the computations for the time parameters of nodes. The computation of \underline{t}_i of node (i) represents the problem of finding the longest path tree rooted at (0), and \bar{t}_i of node (i) can be derived from the longest path tree rooted at ($2n + 1$) when the orientations of all arcs have been reversed. Here, \underline{t}_i and \bar{t}_i of node (i) could be computed based on the Bellman–Ford algorithm as follows (Algorithm 1).

Obviously, Steps 2 and 4 of Algorithm 1 avoid cycles ($i \rightarrow (j) \rightarrow (i)$) in the computations.

Here, the project completion time is \bar{t}_{2n+1} , and the time parameters ES_i , EF_i , LS_i , and LF_i of activity i are computed as follows [48]:

$$ES_i = \underline{t}_{2i-1}, \quad (29)$$

$$EF_i = \underline{t}_{2i}, \quad (30)$$

$$LS_i = \bar{t}_{2i-1}, \quad (31)$$

$$LF_i = \bar{t}_{2i}, \quad (32)$$

and the definitions and computations of time floats of activity i are as follows [48].

4.2.1. Total Float. The total float of activity i , marked as TF_i , is defined as

$$\begin{aligned} TF_i &= \bar{t}_{2i} - \underline{t}_{2i-1} - d_i \\ &= \bar{t}_{2i} - \underline{t}_{2i-1} - w_{2i-1,2i} \\ &= \bar{t}_{2i-1} - \underline{t}_{2i-1} \\ &= \bar{t}_{2i} - \underline{t}_{2i}. \end{aligned} \quad (33)$$

The total float is the maximum delay in the start of an activity without deferring the project completion time.

4.2.2. Free Float. The free float of activity i , marked as FF_i , is computed as

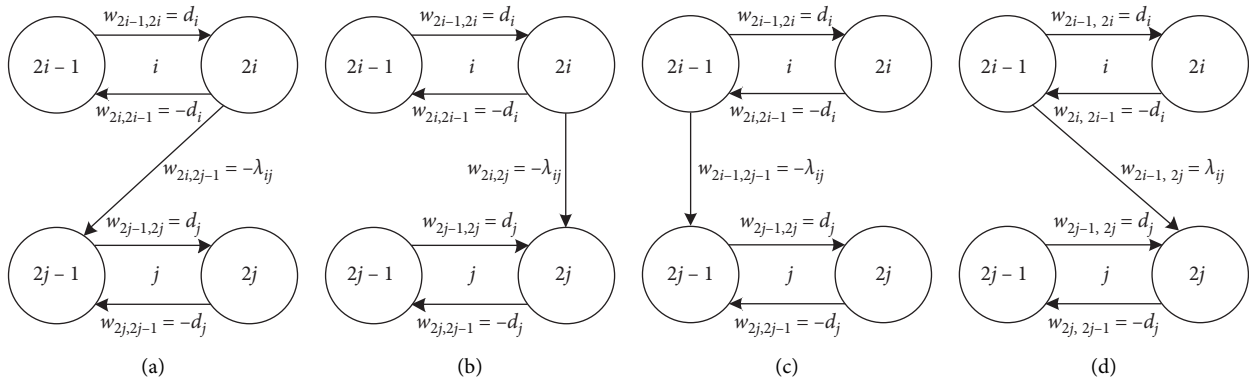


FIGURE 1: Representation of minimum time lags. (a) Representation of $FTS_{ij}^{min}(\lambda_{ij})$. (b) Representation of $FTF_{ij}^{min}(\lambda_{ij})$. (c) Representation of $STS_{ij}^{min}(\lambda_{ij})$. (d) Representation of $STF_{ij}^{min}(\lambda_{ij})$.

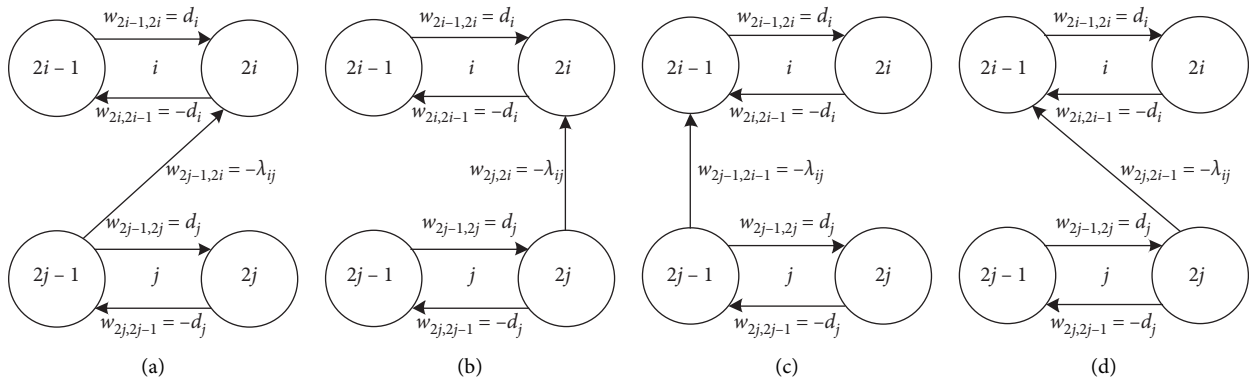


FIGURE 2: Representation of maximum time lags. (a) Representation of $FTS_{ij}^{max}(\lambda_{ij})$. (b) Representation of $FTF_{ij}^{max}(\lambda_{ij})$. (c) Representation of $STS_{ij}^{max}(\lambda_{ij})$. (d) Representation of $STF_{ij}^{max}(\lambda_{ij})$.

Step 1. For the beginning node (0), let

$$\underline{t}_0 = 0.$$

Step 2. For arc $(0, j)$ of node (0), let

$$D^{(1)}(0, j) = w_{0j}.$$

for $x = 2, 3, \dots$,

$$D^{(x)}(0, j) = \min_i \{D^{(x-1)}(0, i) + w_{ij}\},$$

and if $D^{(x)}(0, j) = D^{(x-1)}(0, i) + w_{ij}$, then do not consider arc (j, i) in the next computation. If $x = y$ and each node (j) meet

$$D^{(y)}(0, j) = D^{(y-1)}(0, j),$$

then let

$$\underline{t}_j = D^{(y)}(0, j).$$

Step 3. For the end node $(2n + 1)$, let

$$\bar{t}_{2n+1} = \underline{t}_{2n+1}.$$

Step 4. For arc $(j, 2n + 1)$ of node $(2n + 1)$, let

$$D^{(1)'}(j, 2n + 1) = \bar{t}_{2n+1} - w_{j,2n+1}.$$

for $x = 2, 3, \dots$,

$$D^{(x)'}(j, 2n + 1) = \max_k \{D^{(x-1)'}(k, 2n + 1) - w_{jk}\},$$

and if $D^{(x)'}(j, 2n + 1) = D^{(x-1)'}(k, 2n + 1) - w_{jk}$, then do not consider arc (k, j) in the next computation. If $x = y$ and each node (j) meets

$$D^{(y)'}(j, 2n + 1) = D^{(y-1)'}(j, 2n + 1),$$

then let

$$\bar{t}_j = D^{(y)'}(j, 2n + 1).$$

ALGORITHM 1: Bellman–Ford algorithm.

$$FF_i = \min_{u=2i-1, 2i} \min_{(u,v) \in P} \{\underline{t}_v - \underline{t}_u - w_{uv}\}, \quad (34)$$

and P is the set of precedence relations. The free float is the maximum delay in the start of an activity assuming all other activities are started at their earliest start times.

4.2.3. Safety Float. The safety float of activity i , marked as SF_i , is computed as

$$SF_i = \min_{u=2k-1, 2k} \min_{(u,v) \in P} \{\bar{t}_v - \bar{t}_u - w_{uv}\}. \quad (35)$$

The safety float is the maximum delay in the start of an activity based on the assumption that all other activities are started at their latest start times.

5. Simplifying DTCTP-D with GPRs

The work of Akkan et al. [22] shows that the process of preprocessing the DTCTP-D is inescapably associated with complex path problems in the activity networks. Nevertheless, compared with paths in a large network, activities are considerably fewer in number. The simplification could be a simple process when transforming the computations of paths into the computations of activity parameters.

5.1. Theorems. We discover a relationship between activity float and path lengths in the activity network under GPRs, which reflects the property of the network structure. This idea is similar to that of analytical geometry, which transforms a complex cone curve into a quadratic function. We summarize the relationship in the following theorem.

Theorem 1. For activity, i , its total float, TF_i , is equal to the difference between length of the critical path, μ^{∇} , and length of the longest path, μ_i^{∇} , passing it and from the beginning node (0) to the terminal node $(2n+1)$ of the network, that is,

$$TF_i = L(\mu^{\nabla}) - L(\mu_i^{\nabla}), \quad (36)$$

and $L(\mu)$ indicates the length of path μ that could contain cycles.

Proof. See Appendix A.

We propose the following corollary by extending the respective from activities to all arcs including those in representations of activities and precedence relations in the activity network under GPRs. \square

Corollary 1. For an arc, (i, j) , define

$$TF_{ij} = \bar{t}_j - \underline{t}_i - w_{ij}, \quad (37)$$

and it is equal to the difference between the length of the critical path, μ^{∇} , and the length of the longest path, μ_{ij}^{∇} , passing it and from the beginning node (0) to the terminal node $(2n+1)$ of the network, that is,

$$TF_{ij} = L(\mu^{\nabla}) - L(\mu_{ij}^{\nabla}). \quad (38)$$

Theorem 1 and Corollary 1 help to solve many path problems by transforming the computations of path lengths into the computations of time floats. We apply the theory to simplify DTCTP-D with GPRs. In an activity network, the project duration is equal to the length of the critical path; therefore, in solving the TCTP-D, the solution is to make all the paths shorter than the project deadline T at a minimum total cost.

5.2. Algorithm. We analyze features of the DTCTP-D with GPRs and mainly dispose off the related anomalies to overcome the difficulty of the simplification.

Consider the DTCTP-D with GPRs presented in Section 2.2 and the corresponding activity network under GPRs for the project with GPRs. For the problem mainly considering direct activity costs, the project cost increases following the compression of project duration so that we may not need to consider the case of project duration shorter than the project deadline T ($t_{2n+1} < T$). For the problem considering both direct activity costs and indirect project cost, the project cost may be reduced following the compression of project duration until the balance date of the direct and indirect costs (minimum sum of the direct and indirect project costs). If the balance date exceeds T , the project cost will increase following the compression of project duration from T , and we may not need to consider the case $t_{2n+1} < T$, which is similar to the problem only considering direct activity costs. However, if the balance date is earlier than T , we should consider the case $t_{2n+1} < T$ for the minimum project cost and could further compress the project duration after computing T with the minimum sum of all direct activity costs.

Therefore, we study the simplification of the DTCTP-D with GPRs from the perspectives of direct activity costs and project deadline T . The problem could be simplified as follows (Algorithm 2).

Proof. See Appendix B.

The complexity of Algorithm 2 is $O(mn)$, where m indicates the quantity of arcs (see Appendix C). The optimal solution of the DTCTP-D with GPRs can be solved using the simplified network: (1) the optimal mode of each activity in the simplified network can be obtained by using the proposed algorithms; (2) for each deleted activity in Step 4, its duration is unrelated to T so that the optimal mode is the cheapest of the remaining modes. \square

6. Illustration

6.1. Illustrative Example

6.1.1. Example 1. We consider the activity network under GPRs exhibited in the literature of Elmaghraby and Kamburowski [48] as an illustration, as in Figure 3. The figure represents a project with GPRs including 8 activities and 19 precedence relations between activities, and Table 1 lists 5 duration-cost modes of each activity. If the project must be finished within 14 ($T = 14$) at a minimum total cost, the problem is a DTCTP-D with GPRs. There are 48

Step 1. Let $w_{2i-1,2i} = \min_{i,k}\{d_i^k\}$ and $w_{2i,2i-1} = -\max_{i,k}\{d_i^k\}$ of each activity i , and compute \underline{t}_a and \bar{t}_a of each node (a) using Algorithm 1.

Step 2. Delete mode m_k of activity i that $d_i^k > \bar{t}_{2i} - \underline{t}_{2i-1} + T - \bar{t}_{2n+1}$ or $d_i^k < \underline{t}_{2i} - \bar{t}_{2i-1} - T + \bar{t}_{2n+1}$.

Step 3. Let $w_{2i-1,2i} = \max_{i,k}\{d_i^{k*}\}$ and $w_{2i,2i-1} = -\min_{i,k}\{d_i^{k*}\}$ and d_i^{k*} indicate durations of remaining modes of each activity i , and compute \underline{t}_a and \bar{t}_a of each node (a) using Algorithm 1.

Step 4. Compute TF_{ab} of each arc (a, b), and delete arc (a, b) whose $TF_{ab} > \bar{t}_{2n+1} - T$.

Step 5. For modes m_k of remaining activity i meeting $d_i^{k*} < \bar{t}_{2i} - \underline{t}_{2i-1} - \bar{t}_{2n+1} + T$ and $d_i^{k*} > \underline{t}_{2i} - \bar{t}_{2i-1} + \bar{t}_{2n+1} - T$, preserve the cheapest one and delete the others.

ALGORITHM 2: Preprocessing DTCTP-D algorithm.

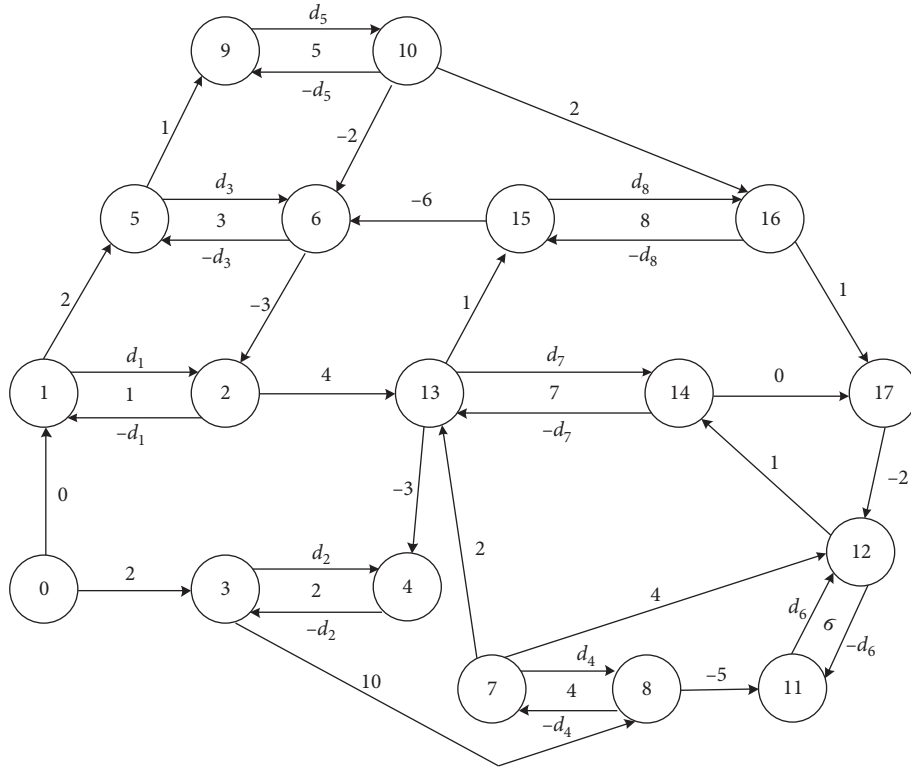


FIGURE 3: Example of an activity network under GPRs.

TABLE 1: Remaining modes after Step 2 of Algorithm 2.

Activity number	Mode m_1		Mode m_2		Mode m_3		Mode m_4		Mode m_5	
	d	c	d	c	d	c	d	c	d	c
1	1	1000	2	950						
2	5	5000	5	5500	6	4500	6	4200	7	3500
3	2	100	3	80						
4							6	250	7	200
5	3	10000	4	7000						
6	4	800	5	780						
7	6	2500								
8	1	150	2	140						

variables and 76 constraints in the problem formulation in Section 2.2, and the amount of schemes of the problem is $5^8 = 390625$.

For greater efficiency, we prefer to simplify the DTCTP-D with GPRs. There are few approaches to simplifying the problem in the literature, and a recent simplification method

for the deadline problem of classic DTCTP-D was presented by Akkan et al. [22]. However, the abovementioned practical problem with deadline $T = 14$ shown as in Figure 3 and Table 1 will be nonsimplified using the Akkan algorithm. For example, the first step of eliminating “long modes” is to compute the earliest start times and the latest finish times of

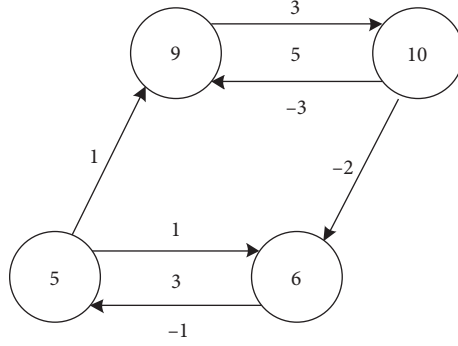


FIGURE 4: Positive cycle caused by the Akkan algorithm.

TABLE 2: Modes of each activity.

Activity number	Mode m_1		Mode m_2		Mode m_3		Mode m_4		Mode m_5	
	d	c	d	c	d	c	d	c	d	c
1	1	1000	2	950	8	500	10	450	11	420
2	5	5000	5	5500	6	4500	6	4200	7	3500
3	2	100	3	80	9	50	10	45	12	35
4	1	500	3	300	4	280	6	250	7	200
5	3	10000	4	7000	9	5000	10	4800	11	4500
6	4	800	5	780	6	760	8	730	10	700
7	6	2500	8	2300	9	2200	10	2000	13	1500
8	1	150	2	140	5	100	6	80	10	40

all activities using the shortest modes; however, the shortest mode selection for each activity is unfeasible because the execution results in positive cycles in the activity network under GPRs, such as the cycle in Figure 4.

Our algorithm in Section 4.2 is applicable to simplify the DTCTP-D with GPRs as follows.

Step 1. Table 2 shows that $d_i^1 \leq d_i^2 \leq d_i^3 \leq d_i^4 \leq d_i^5$ for each activity i . Let $w_{2i-1,2i} = d_i^1$ and $w_{2i,2i-1} = -d_i^5$ of each activity i ; compute \underline{t}_a and \bar{t}_a of each node (a) using Algorithm 1, as in Figure 5.

Step 2. For activity 1, $\bar{t}_2 - \underline{t}_1 + 14 - \bar{t}_{17} = 7$ and $d_1^3, d_1^4, d_1^5 > 7$; hence, delete modes m_3, m_4 , and m_5 . For activity 3, $\bar{t}_6 - \underline{t}_5 + 14 - \bar{t}_{17} = 8$ and $d_3^3, d_3^4, d_3^5 > 8$; hence, delete modes m_3, m_4 , and m_5 . For activity 4, $\underline{t}_8 - \bar{t}_7 - T + \bar{t}_{2n+1} = 6$ and $d_4^1, d_4^2, d_4^3 < 6$; hence, delete modes m_1, m_2 , and m_3 . Table 1 shows the remaining modes of each activity.

Step 3. Let $w_{2i-1,2i} = \max\{d_i^{k*}\}$ and $w_{2i,2i-1} = -\min\{d_i^{k*}\}$ of each activity i . For example, according to Table 1, for activity 1, $w_{1,2} = \max\{d_1^1, d_1^2\} = 2$ and $w_{2,1} = -\min\{d_1^1, d_1^2\} = -1$; for activity 2, $w_{3,4} = \max\{d_2^1, d_2^2, d_2^3, d_2^4, d_2^5\} = 7$ and $w_{4,3} = -\min\{d_2^1, d_2^2, d_2^3, d_2^4, d_2^5\} = -5$; and so on. Compute \bar{t}_a of each node (a) using Algorithm 1, as in Figure 6.

Step 4. According to Figure 6 and $T = 14$, $\bar{t}_{17} - T = 0$; hence, delete arc (a, b) with $TF_{ab} > 0$. Delete arc $(0, 1)$

because of $TF_{0,1} = \bar{t}_1 - \underline{t}_0 - w_{0,1} = 2 > 0$; delete arc $(1, 2)$ because of $TF_{1,2} = \bar{t}_2 - \underline{t}_1 - w_{1,2} = 2$; and so on, as in Figure 7. Analyzing Figure 7, the durations of activities 2 and 6 are unrelated to the project duration; the two activities could also be deleted, as in Figure 8.

Step 5. According to Figure 8 and $T = 14$, $\underline{t}_8 - \bar{t}_7 + \bar{t}_{17} - 14 = 6$ and $d_4^5 = 7 \geq 6$ for mode m_5 of activity 4. Mode m_5 is the only and obviously the cheapest mode meeting $d_4^k = 7 \geq 6$ in the remaining modes of activity 4; therefore, mode m_5 should remain. Table 3 shows remaining modes of remaining activities.

Figure 8 shows that only two activities and five arcs are enough for problem formulation. Table 3 shows that the optimal solution is determined by the remaining two modes of activity 4 and one mode of activity 7, and obviously the optimal modes of activities 4 and 7 are modes m_5 and m_1 , respectively. The optimal mode of each deleted activity in Step 4 is determined by the cheapest mode. Therefore, Table 4 shows the optimal solution of the DTCTP-D with GPRs that the project total cost is 15150 according to Table 2. After the simplification, the optimal solution of the problem only needs to be chosen from two feasible solutions, which is much simpler than the initial issue and tests the simplification.

6.1.2. Example 2. The illustrative example 1 in Section 6.1.1 shows that the instance with 8 activities and 5 modes per

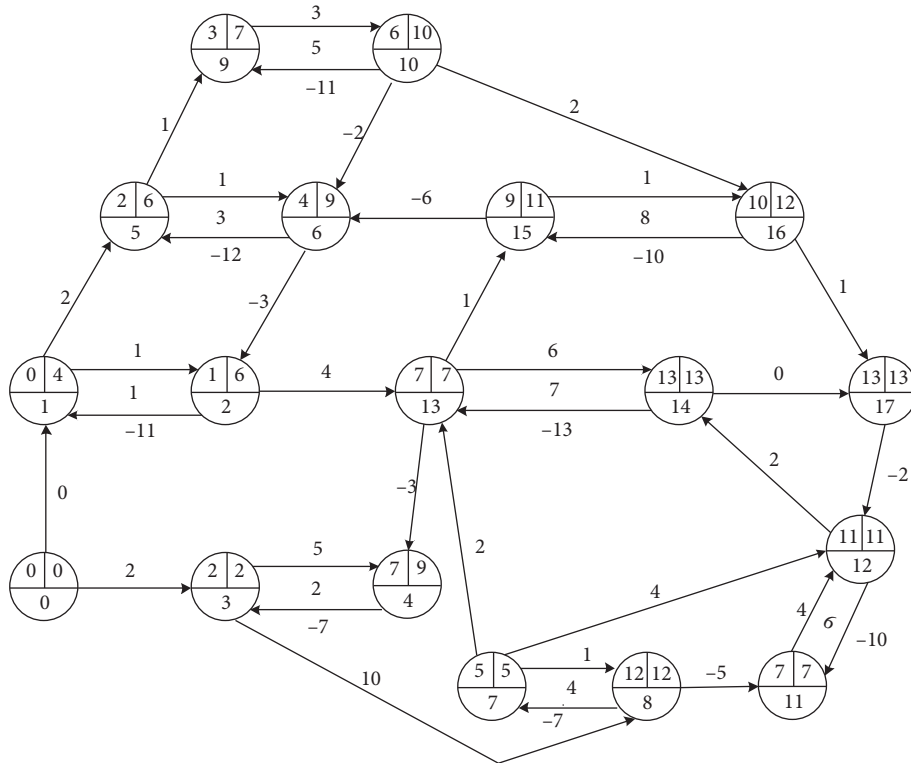


FIGURE 5: Time parameters of nodes computed using Step 1.

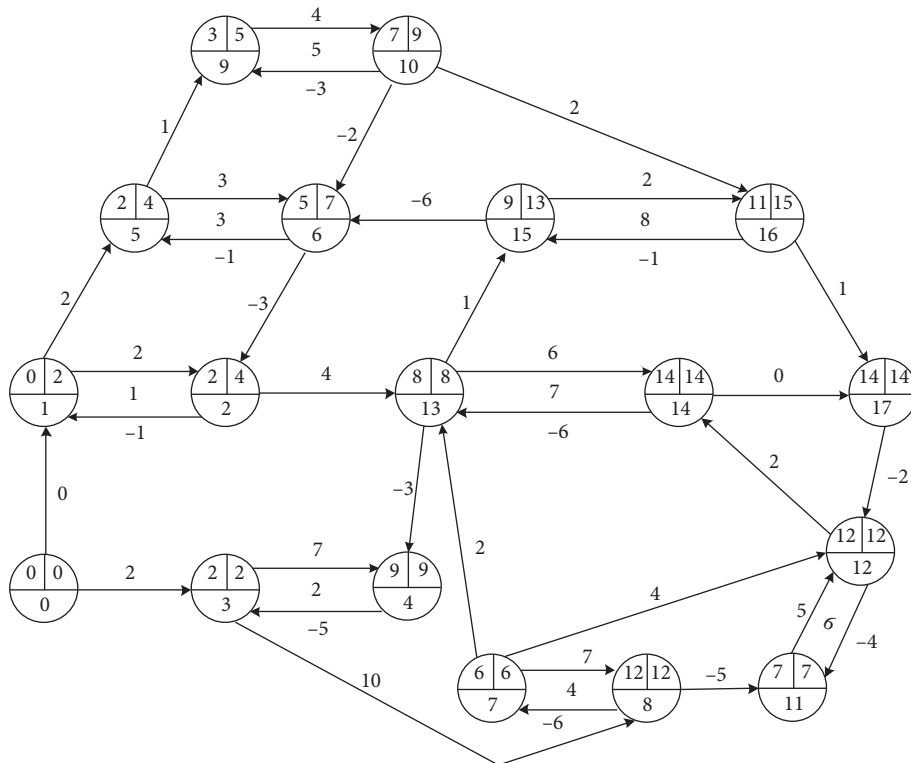


FIGURE 6: Time parameters of nodes computed using Step 3.

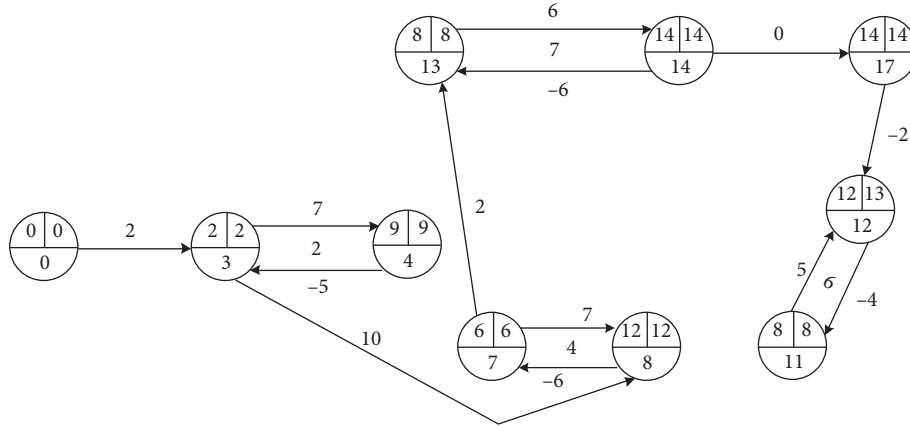


FIGURE 7: Simplified network using Step 4.

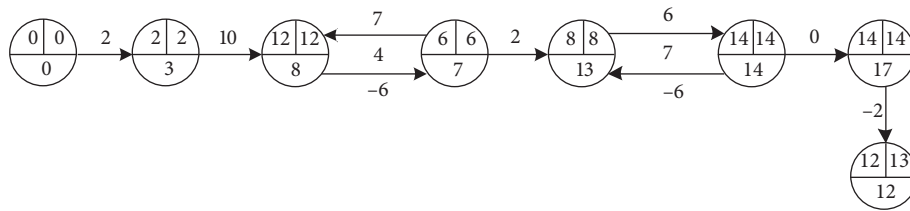


FIGURE 8: Further simplified network.

TABLE 3: Remaining activities and modes after Step 5.

Activity number	Mode m_1		Mode m_4		Mode m_5	
	d	c	d	c	d	c
4			6	250	7	200
7	6	2500				

TABLE 4: Optimal mode of each activity.

Activity number	Optimal mode
1	Mode m_2
2	Mode m_5
3	Mode m_2
4	Mode m_5
5	Mode m_2
6	Mode m_2
7	Mode m_1
8	Mode m_2

activity could be simplified as an instance only with 2 activities and up to 3 modes per activity using the proposed preprocessing technology, and the optimal solution of the original instance could be computed based on the simplified instance.

In this section, we present a medium-size example of nonlinear DTCTP with GPRs from the literature [8, 71], and set the problem as a DTCTP-D by letting the project deadline be 83 (days) which is the optimum project duration in [8]. The example project consists of 29 activities. The precedence relationships and the lag/lead times between

succeeding activities are presented in Table 5. While the lag times are defined with positive numbers, the lead times are determined with negative numbers. The alternative cost-duration options for each activity are presented in Table 6.

Applying our algorithm in Section 4.2 to simplify the problem instance, we could delete some activities from Table 6, that is, the optimal modes of these deleted activities for the problem are their longest modes. In addition, considering each activity with only one mode, obviously its optimal mode is the unique one, and we also could delete it from Table 6, as shown in Table 7. Therefore, in order to obtain the optimal solution of the problem instance, we only need to make decisions for the activity modes in Table 7. Comparing with Table 6, there are 14 activities in Table 7 that are 15 activities less than Table 6, particularly, most of the deleted activities have 3 modes per activity. The simplification reduces the total amount of optional solutions (schedules) from $3^{17} * 2^6 \approx 8 * 10^9$ to $3^9 * 2^5 \approx 6 * 10^5$. Furthermore, we also delete 9 precedence relations (lag/lead times) from Table 5 for more downsizing of the integer model of the instance.

6.2. Computational Experiments. We generate several random large-size instances of the DTCTP with GPRs to measure the performance of our preprocessing technology, and further test the effects of the simplification under different project deadlines for each instance. We first used RanGen2 [78, 79] to generate 20 random large-size instances (500 activities per instance), and model the parameters as a uniform probability density function in order to make a balanced representation of the instances' properties. Then, we apply the procedure proposed by Khalili-Damghani et al. [75] and Tareghian and

TABLE 5: Precedence relationships and the lag/lead times of the example project activities.

Activity number	Succeeding activity	Precedence relationship	Lag/lead time (day)
1	2	Finish-to-Start	-3
	3	Finish-to-Start	0
	6	Finish-to-Start	0
2	3	Finish-to-Start	0
	7	Finish-to-Start	0
3	4	Finish-to-Start	0
	8	Finish-to-Start	0
4	5	Start-to-Start	1
	9	Finish-to-Start	0
5	10	Finish-to-Start	0
	11	Finish-to-Start	0
6	7	Finish-to-Start	-1
7	8	Finish-to-Start	0
8	9	Finish-to-Start	0
9	10	Finish-to-Finish	1
10	11	Finish-to-Start	0
11	12	Finish-to-Start	0
12	13	Start-to-Start	2
	15	Finish-to-Start	-4
13	14	Start-to-Start	2
14	15	Finish-to-Start	-2
15	16	Finish-to-Start	-6
	17	Start-to-Start	4
	18	Finish-to-Start	4
	19	Finish-to-Start	0
16	—	—	—
17	20	Start-to-Start	3
18	—	—	—
19	—	—	—
20	21	Start-to-Start	2
21	22	Start-to-Start	2
22	23	Finish-to-Start	-9
23	24	Start-to-Start	6
	25	Finish-to-Start	-4
24	—	—	—
25	26	Start-to-Start	4
26	27	Finish-to-Start	0
27	28	Finish-to-Start	-3
28	29	Finish-to-Start	0

Taheri [80] to assign the GPRs, time lags, and execution modes to the project activities. Table 8 illustrates the parameter setting and GPRs and execution mode generations.

For these generated instances, our first phase is to compute the maximum makespan of each project using the network planning technology. In the second phase, we set different deadlines for each project and call the difference between the maximum makespan and a deadline of a project as a required compression of the project. We set a parameter as “Compression Ratio” that indicates the ration of the required compression and maximum makespan of a project. Our third phase is to let the compression ratio, respectively, be 5%, 10%, and 15% for each instance, and simplify the instances using the preprocessing technology. Figure 9 shows the details of the computational results (only focuses on the reductions of activities), and Table 9 lists several statistical results of the experiments.

Figure 9 and Table 9 illustrate that most instances could be shrunk as small- or medium-size ones by the

preprocessing technology. After the simplifications, the optimal solutions of the initial instances could be obtained based on their simplified ones. We could use exact algorithms to compute the optimal solutions of the small-size simplified instances or use approximate or heuristic algorithms to compute much better near-optimal solutions for the medium-size simplified instances.

7. Discussion

The following discussion is addressed on the advantages and the limitations of the proposed preprocessing technology for the DTCTP with GPRs. Klanšek and Pšunder [8] discussed a limitation of their MINLP optimization model for the nonlinear DTCTP that the optimization model comprehends deterministic time and cost estimates. It is a limitation of our work too. Generally, a precise estimation of durations and direct costs of project activities is a very difficult task that it significantly depends upon the crew size, the level of

TABLE 6: Modes of each activity.

Activity number	Mode m_1		Mode m_2		Mode m_3	
	d	c	d	c	d	c
1	5	2030	4	2300	—	—
2	8	1020	7	1280	6	1510
3	8	1700	7	1850	6	2090
4	4	590	3	730	—	—
5	2	90	—	—	—	—
6	4	910	3	1100	—	—
7	2	250	—	—	—	—
8	7	1490	6	1650	5	1830
9	4	520	3	750	—	—
10	2	90	—	—	—	—
11	1	50	—	—	—	—
12	8	3260	7	3580	6	3710
13	5	1140	4	1400	3	1720
14	4	300	3	450	—	—
15	8	1020	6	1300	6	1430
16	9	790	8	900	6	1180
17	13	3340	12	3750	11	4060
18	9	470	8	650	7	830
19	6	460	5	600	4	810
20	6	1280	5	1430	—	—
21	14	1090	12	1320	10	1560
22	14	900	11	1140	9	1400
23	14	2220	12	2510	11	2690
24	3	230	—	—	—	—
25	6	1590	5	1790	4	1990
26	10	2630	9	2930	8	3240
27	8	2060	7	2450	6	2660
28	10	320	9	440	8	610
29	1	50	—	—	—	—

TABLE 7: Modes of each activity after preprocessing (simplification).

Activity number	Mode m_1		Mode m_2		Mode m_3	
	d	c	d	c	d	c
1	5	2030	4	2300	—	—
2	8	1020	7	1280	6	1510
3	8	1700	7	1850	6	2090
4	4	590	3	730	—	—
6	4	910	3	1100	—	—
8	7	1490	6	1650	5	1830
9	4	520	3	750	—	—
12	8	3260	7	3580	6	3710
14	4	300	3	450	—	—
22	14	900	11	1140	9	1400
23	14	2220	12	2510	11	2690
26	10	2630	9	2930	8	3240
27	8	2060	7	2450	6	2660
28	10	320	9	440	8	610

qualifications and experience of the workers, the quality of management and supervision, and the level of resource utilization. However, the deterministic approach of project parameters does not consider uncertainty, and in cases of risk associated, the stochastic consideration is necessary to estimate the project execution more realistically.

TABLE 8: Parameter setting, and GPRs and execution mode generations.

Number of activities	500
Number of modes	Fixed at 4
Activity durations	Randomly selected from the interval [10, 100]
Activity normal costs	Randomly selected from the interval [1000, 5000]
Precedence relations	1 = FTF, 2 = FTS, 3 = STF, 4 = STS; randomly selected from the interval [1, 4]
Time lags	1 = Positive, 2 = Negative; randomly selected from the interval [0, 10]

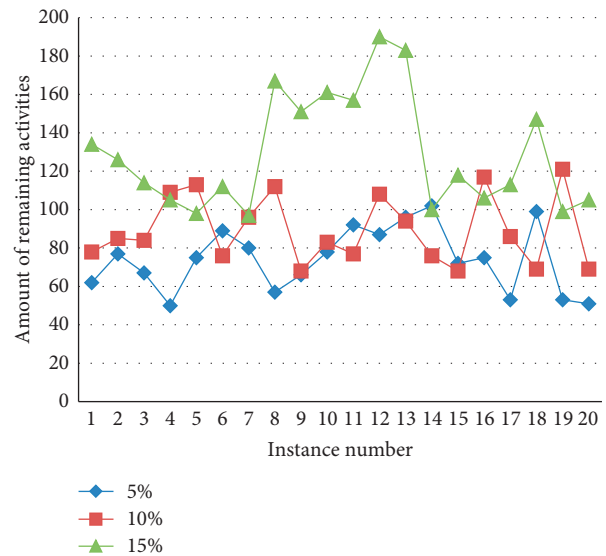


FIGURE 9: Details of experimental results.

The proposed preprocessing technology for the DTCTP with GPRs shows certain advantages over previous works. Namely, it yields the exact optimum solutions of the large-size problem instances, while the exact algorithms optimize small-size instances and the heuristic algorithms approximately optimize large-size instances. The preprocessing technology is adapt to linear and nonlinear DTCTP with GPRs, including the traditional DTCTP. However, we have to face another limitation that the preprocessing technology may not keep widely powerful for many cases. On the one hand, Table 9 shows that the effect of simplification may be weakened by increasing the compression ratio of project duration. It means that the preprocessing (simplification) technology has strong function to handle a large-size instance in case of a small difference between the deadline and makespan of project. However, for a much earlier project deadline, the preprocessing technology may be weak or even lose efficacy. On the other hand, the time required to compute the optimal solution of a specific DTCTP with GPRs using the preprocessing technology increases with the size of the optimization problem. The equivalent simplification cannot change the nature of complexity of the DTCTP with GPRs, and apparently, failure to ensure that the preprocessing technology could be used to simplify any much larger-size instance to an enough small-size one whose

TABLE 9: Statistical experimental results.

Compression ratio (%)	Amount of remaining activities after preprocessing (simplifying) the instances					
	Average		Minimum		Maximum	
5	74	19 with 2 modes	50	8 with 2 modes	102	10 with 2 modes
		10 with 3 modes		7 with 3 modes		14 with 3 modes
		45 with 4 modes		35 with 4 modes		78 with 4 modes
10	89	7 with 2 modes	68	7 with 2 modes	121	15 with 2 modes
		11 with 3 modes		14 with 3 modes		17 with 3 modes
		71 with 4 modes		47 with 4 modes		89 with 4 modes
15	129	11 with 2 modes	97	5 with 2 modes	193	12 with 2 modes
		24 with 3 modes		23 with 3 modes		30 with 3 modes
		94 with 4 modes		69 with 4 modes		151 with 4 modes

optimum solution could be computed using exact algorithm. Our further task is to handle these limitations and try to develop more effective approaches for the DTCTP with GPRs.

8. Conclusion

For large-scale project scheduling problems, direct computations of optimal solutions are not feasible even when the most advanced computers are utilized. In these situations, equivalent simplification is a more effective approach. Much of the literature presents simplification for the DTCTP. However, there are few works on simplifying the DTCTP with GPRs, and the current preprocessing technologies for the classic DTCTP are inapplicable to GPRs because of anomalies in their foundations. In this study, we focused on substantially simplifying the deadline version of DTCTP (DTCTP-D) with GPRs while coping with the related anomalies. The simplification includes identifying and deleting unfeasible and redundant solutions of the problem.

We studied the simplification process based on the activity network under GPRs, which is actually a complex path problem. We discovered relationships between time floats and paths. Furthermore, we solved the path problem using floats and proposed a polynomial algorithm for the simplification of the DTCTP-D with GPRs. This approach is analogous to analytical geometry, which transforms a complex cone curve into an algebraic quadratic function and solves the geometric problem using an algebraic method. We can transform many other complex path problems into simple float problems with the relationships.

In the future, we will develop the methods further to obtain better solutions for the DTCTP-D with GPRs based on the simplification and the other contributions of this paper. In particular, we will attempt to detect hybrid and dynamic approaches associated with the rules proposed in this work and improve the proposed heuristics and propose new algorithms for the DTCTP-D. Based on these works, we will also preprocess the other versions of DTCTP with GPRs, such as the budget problem (DTCTP-B) and efficiency problem (DTCTP-E).

Appendix

A. Proof of Theorem 1

For activity i , in an activity network under GPRs, according to equation (33), $TF_i = \bar{t}_{2i-1} - \underline{t}_{2i-1}$. According to the

computation of time parameters in Section 3.2, the relationships between \underline{t}_j and \bar{t}_j of node (j) and paths are as follows:

$$\underline{t}_j = L(\mu_0^\nabla \rightarrow j), \quad (\text{A.1})$$

$$\bar{t}_j = L(\mu^\nabla) - L(\mu_j^\nabla \rightarrow 2n+1), \quad (\text{A.2})$$

where $L(\mu_a^\nabla \rightarrow b)$ denotes the longest path from node (a) to (b). Substitute equations (A.1) and (A.2) into equation (33):

$$\begin{aligned} TF_i &= \bar{t}_{2i-1} - \underline{t}_{2i-1} \\ &= L(\mu^\nabla) - L(\mu_{2i-1}^\nabla \rightarrow 2n+1) - L(\mu_0^\nabla \rightarrow 2i-1) \\ &= L(\mu^\nabla) - (L(\mu_0^\nabla \rightarrow 2i-1) + L(\mu_{2i-1}^\nabla \rightarrow 2n+1)) \\ &= L(\mu^\nabla) - L(\mu_i^\nabla). \end{aligned} \quad (\text{A.3})$$

Therefore, equation (36) is proved. This completes the proof.

B. Proof of Algorithm 2

(1) Delete Unfeasible Modes of Activities.

Using Step 1, let $w_{2i-1,2i} = \min\{d_i^k\}$ and $w_{2i,2i-1} = -\max\{d_i^k\}$ of each activity i such that each path in the activity network under GPRs has its minimum length, and mark the network as G . Now, for arc $(2i-1, 2i)$ of activity i , if a path $\mu_{2i-1,2i}$ (e.g., the longest path $\mu_{2i-1,2i}^\nabla$ from the beginning node (0) to the terminal node $(2n+1)$ of G passing arc $(2i-1, 2i)$ with length d_i^k is longer than the project deadline T , and it means that the project duration will be longer than T if activity i selects its mode m_k with duration d_i^k no matter which modes other activities select. Therefore, mode m_k of activity i is unfeasible and could be deleted.

Compute \underline{t}_a and \bar{t}_a of each node (a) in network G using Algorithm 1. According to equations (29) and (30), if activity i selects mode m_k ,

$$\begin{aligned} L(\mu_{2i-1,2i}^\nabla) &= L(\mu_0^\nabla \rightarrow 2i-1) + w_{2i-1,2i} + L(\mu_{2i}^\nabla \rightarrow 2n+1) \\ &= \underline{t}_{2i-1} + d_i^k + \bar{t}_{2n+1} - \bar{t}_{2i}, \end{aligned} \quad (\text{B.1})$$

and $\mu_a^\nabla \rightarrow b$ indicates the longest path from node (a) to (b). If $L(\mu_{2i-1,2i}^\nabla) > T$, then

$$d_i^k > \bar{t}_{2i} - \underline{t}_{2i-1} + T - \bar{t}_{2n+1}. \quad (\text{B.2})$$

Therefore, using Step 2, if $d_i^k > \bar{t}_{2i} - \underline{t}_{2i-1} + T - \bar{t}_{2n+1}$, duration d_i^k results in the project duration longer than T and mode m_k could be deleted. In addition,

$$\begin{aligned} L(\mu_{2i,2i-1}^\nabla) &= L(\mu_{0 \rightarrow 2i}^\nabla) + w_{2i,2i-1} + L(\mu_{2i-1 \rightarrow 2n+1}^\nabla) \\ &= \underline{t}_{2i} - d_i^k + \bar{t}_{2n+1} - \bar{t}_{2i-1}. \end{aligned} \quad (\text{B.3})$$

If $L(\mu_{2i-1,2i}^\nabla) > T$, then

$$d_i^k < \underline{t}_{2i} - \bar{t}_{2i-1} - T + \bar{t}_{2n+1}. \quad (\text{B.4})$$

Therefore, if $d_i^k < \underline{t}_{2i} - \bar{t}_{2i-1} - T + \bar{t}_{2n+1}$, duration d_i^k results in the project duration longer than T and mode m_k could be removed.

(2) Delete Redundant Arcs and Activities.

After deleting the unfeasible modes of each activity, let $w_{2i-1,2i} = \max_{i,k}\{d_i^{k*}\}$ and $w_{2i,2i-1} = -\min_{i,k}\{d_i^{k*}\}$ using Step 3, and d_i^{k*} indicates durations of remaining modes m_k^* of each activity i . Now mark the network as G' . Here, G' means that each activity selects its remaining longest duration so that each acyclic path has its maximum length. In G' , if the longest path μ_{ab}^∇ passing arc (a, b) and from the beginning node (0) to the terminal node $(2n + 1)$ of the network is shorter than T , all paths passing arc (a, b) are certainly shorter than T that they are independent of the DTCTP-D. Therefore, if arc (a, b) represents a precedence relation between activities, the arc is redundant and could be deleted; if arc (a, b) represents an activity, each mode of the activity has no effect on T so that the optimal mode of the activity is the cheapest, and the activity is redundant for the problem formulation and could be deleted.

The representation of activity i in G' may be a positive cycle in G' because of $w_{2i-1,2i} + w_{2i,2i-1} = \max_{i,k}\{d_i^{k*}\} - \min_{i,k}\{d_i^{k*}\} \geq 0$, which must be avoided to compute the longest path for \underline{t}_a and \bar{t}_a of each node G' . Algorithm 1 could avoid the cycles and it applies to the computations of \underline{t}_a and \bar{t}_a in G' . According to equation (37),

$$\begin{aligned} L(\mu_{ab}^\nabla) &= L(\mu^\nabla) - \text{TF}_{ab} \\ &= \bar{t}_{2n+1} - \text{TF}_{ab}. \end{aligned} \quad (\text{B.5})$$

If $L(\mu_{ab}^\nabla) < T$, then $\text{TF}_{ab} > \bar{t}_{2n+1} - T$. Therefore, using Step 4, if $\text{TF}_{ab} > \bar{t}_{2n+1} - T$, arc (a, b) could be deleted.

(3) Delete Redundant Modes of Nonredundant Activities.

Mark the remaining network of G' as G'' , and Steps 3 and 4 mean that each acyclic path in G'' still has its maximum length. For the remaining modes of activities in G'' , if activity i selects mode m_k^* and both of the longest acyclic paths $L(\mu_{2i-1,2i}^\nabla)$ and $L(\mu_{2i,2i-1}^\nabla)$ passing arcs $(2i - 1, 2i)$ and arc $(2i, 2i - 1)$, respectively, are shorter than T , then duration d_i^{k*} of mode m_k^* is feasible but independent of T . Therefore, for such modes of an activity, we only need to consider the cheapest for the minimum project total cost, and the other more expensive ones are redundant and could be deleted.

According to equations (A.4) and (A.6), if activity i selects mode m_k^* , and $L(\mu_{2i-1,2i}^\nabla) < T$ and $L(\mu_{2i,2i-1}^\nabla) < T$, then

$$\underline{t}_{2i} - \bar{t}_{2i-1} - T + \bar{t}_{2n+1} < d_i^{k*} < \bar{t}_{2i} - \underline{t}_{2i-1} + T - \bar{t}_{2n+1}. \quad (\text{B.6})$$

Therefore, for all modes m_k^* of activity i meeting $d_i^{k*} < \bar{t}_{2i} - \underline{t}_{2i-1} + T - \bar{t}_{2n+1}$ and $d_i^{k*} > \bar{t}_{2i} - \bar{t}_{2i-1} - T + \bar{t}_{2n+1}$, except for the cheapest one in k^* , the others are redundant and could be deleted. This completes the proof.

C. Complexity of Algorithm 2

Suppose there are n activities (namely, $2n + 2$ nodes) and m arcs in the activity network under GPRs. In Steps 1 and 3, time parameters \underline{t}_a and \bar{t}_a of each node (a) need to be computed using Algorithm 1 based on the Bellman–Ford algorithm; therefore, the complexity is $O(mn)$. In Steps 2 and 5, the extreme value of the unfeasible and redundant modes of each activity needs to be computed to delete unfeasible and redundant modes; hence, the complexity is $O(n)$. In Step 4, TF_{ij} of each arc (a, b) needs to be computed and compared with $\bar{t}_{2n+1} - T$; thus, the complexity is $O(m)$. In summary, the complexity of Algorithm 2 is $O(mn)$.

The following symbols are used in this paper:

Notations

c_i^k :	Cost of an activity i in a mode m_k
d_i :	Duration of an activity i
d_i^k :	Duration of an activity i in a mode m_k
$D^{(x)}(i, j)$:	Iterated symbol in the Bellman–Ford algorithm
e :	Indirect project cost per date
E :	Set of time lags, such as $E_{\text{STG}^{\min}}$, including all start-to-start minimum time lags between activities in project
ES_i :	The earliest start time of an activity i
EF_i :	The earliest finish time of an activity i
f_i :	Finish time of an activity i
FF_i :	Free float of an activity i
$\text{FTF}_{ij}^{\max}(\lambda)$:	Finish-to-finish maximum time lag λ between activities i and j
$\text{FTF}_{ij}^{\min}(\lambda)$:	Finish-to-finish minimum time lag λ between activities i and j
$\text{FTS}_{ij}^{\max}(\lambda)$:	Finish-to-start maximum time lag λ between activities i and j
$\text{FTS}_{ij}^{\min}(\lambda)$:	Finish-to-start minimum time lag λ between activities i and j
G :	Activity network under GPRs
i :	Activity with number i
(i) :	Node with number i in an activity network
(i, j) :	Arc linking nodes (i) and (j) in an activity network
$L(\mu)$:	Length of a path μ in an activity network
LS_i :	The latest start time of an activity i
LF_i :	The latest finish time of an activity i
m_i :	Mode with number i of an activity
n :	Number of project activities
N :	Set of all nodes in an activity network
P :	Set of arcs representing time lags in an activity network

s_i :	Start time of an activity i
SF_i :	Safety float of an activity i
$STF_{ij}^{\max}(\lambda)$:	Start-to-finish maximum time lag λ between activities i and j
$STF_{ij}^{\min}(\lambda)$:	Start-to-finish minimum time lag λ between activities i and j
$STS_{ij}^{\max}(\lambda)$:	Start-to-start maximum time lag λ between activities i and j
$STS_{ij}^{\min}(\lambda)$:	Start-to-start minimum time lag λ between activities i and j
r_i :	Number of execution modes for an activity i
T :	Project deadline
t_i :	Realization time of a node (i)
\underline{t}_i :	Minimum realization time of a node (i) in an activity network
\bar{t}_i :	Maximum realization time of a node (i) in an activity network
TF_i :	Total float of an activity i
TF_{ij} :	Total float of an arc (i, j) (an extended concept of TF_i)
V :	Set of all activities in project
w_{ij} :	Length of an arc (i, j) in an activity network
x, y :	Counters in the Bellman–Ford algorithm
δ_i^k :	0-1 binary decision variable that equals 1 if an activity i executes in a mode m_k and is 0 otherwise
λ_{ij} :	Value of a time lag between activities i and j
μ :	Path in an activity network
μ_{ij} :	Path passing an arc (i, j) in an activity network
μ^{∇} :	Critical path (the longest path) in an activity network
μ_i^{∇} :	The longest path passing an activity i in an activity network
μ_{ij}^{∇} :	The longest path passing an arc (i, j) in an activity network
$\mu_{i \rightarrow j}^{\nabla}$:	The longest path from a node (i) to a node (j) in an activity network.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to acknowledge the Natural Science Foundation of China (Grant no. 71961020), Jiangxi Provincial Universities Humanities and Social Sciences Foundation of China (Grant no. GL162030), and Jiangxi Provincial Association of Social Sciences Foundation of China (Grant number 17GL32). The authors would like to acknowledge Dr Guoping Ji's works of literature review, writing assistance, and proofreading for the manuscript; he

participated in writing and technical editing of the manuscript.

References

- [1] H. Kerzner, *Project Management: A Systems Approach to Planning, Scheduling and Control*, John Wiley & Sons, Hoboken, NJ, USA, 2009.
- [2] T. J. Hindelang and J. F. Muth, "A dynamic programming algorithm for decision CPM networks," *Operations Research*, vol. 27, no. 2, pp. 225–241, 1979.
- [3] P. De, E. J. Dunne, J. B. Ghosh, and C. E. Wells, "Complexity of the discrete time-cost tradeoff problem for project networks," *Operations Research*, vol. 45, no. 2, pp. 302–306, 1997.
- [4] Ö. Hazır, M. Haouari, and E. Erel, "Discrete time/cost trade-off problem: a decomposition-based solution algorithm for the budget version," *Computers and Operations Research*, vol. 37, no. 4, pp. 649–655, 2010.
- [5] B. Roy, "Graphes et ordonnancements," *Revue Francaise Recherche Operation*, vol. 25, pp. 323–326, 1962.
- [6] S. E. Elmaghraby, "An algebra for the analysis of generalized activity networks," *Management Science*, vol. 10, no. 3, pp. 494–514, 1964.
- [7] D. R. Fulkerson, "A network flow computation for project cost curves," *Management Science*, vol. 7, no. 2, pp. 167–178, 1961.
- [8] U. Klanšek and M. Pšunder, "MINLP optimization model for the nonlinear discrete time-cost trade-off problem," *Advances in Engineering Software*, vol. 48, pp. 6–16, 2012.
- [9] J. E. Kelley, "Critical-path planning and scheduling: mathematical basis," *Operations Research*, vol. 9, no. 3, pp. 296–320, 1961.
- [10] J. E. Kelley and M. R. Walker, *Critical Path Planning and Scheduling: An Introduction*, Mauchly Associates, Ambler, PA, USA, 1959.
- [11] J. G. Szmerkovsky and P. Venkateshan, "An integer programming formulation for the project scheduling problem with irregular time-cost tradeoffs," *Computers & Operations Research*, vol. 39, no. 7, pp. 1402–1410, 2012.
- [12] H. Nikoosarm, F. H. Lotfi, J. Jassbi, and M. R. Shahriari, "A new mathematical model for time cost trade-off problem with budget limitation based on time value of money," *Applied Mathematical Sciences*, vol. 4, no. 63, pp. 3107–3119, 2010.
- [13] R. Sonmez and Ö. Bettemir, "A hybrid genetic algorithm for the discrete time-cost trade-off problem," *Expert Systems with Applications*, vol. 39, no. 13, pp. 11428–11434, 2012.
- [14] O. Kwon, S. Lee, and J. Son, "Advanced time-cost trade-off model using mixed integer programming," *Korean Journal of Construction Engineering and Management*, vol. 16, no. 6, pp. 53–62, 2015.
- [15] M. Shahriari, "Multi-objective optimization of discrete time-cost tradeoff problem in project networks using non-dominated sorting genetic algorithm," *Journal of Industrial Engineering International*, vol. 12, no. 2, pp. 159–169, 2016.
- [16] P. De, E. James Dunne, J. B. Ghosh, and C. E. Wells, "The discrete time-cost tradeoff problem revisited," *European Journal of Operational Research*, vol. 81, no. 2, pp. 225–238, 1995.
- [17] G. Değirmenci and M. Azizoğlu, "Branch and bound based solution algorithms for the budget constrained discrete time/cost trade-off problem," *Journal of the Operational Research Society*, vol. 64, no. 10, pp. 1474–1484, 2013.
- [18] E. Demeulemeester, B. De Reyck, B. Foubert, W. Herroelen, and a. M. Vanhoucke, "New computational results on the discrete time/cost trade-off problem in project networks,"

- Journal of the Operational Research Society*, vol. 49, no. 11, pp. 1153–1163, 1998.
- [19] M. Vanhoucke, E. Demeulemeester, and W. Herroelen, “Discrete time/cost trade-offs in project scheduling with time-switch constraints,” *Journal of the Operational Research Society*, vol. 53, no. 7, pp. 741–751, 2002.
- [20] E. Hadjicostantinou and E. Klerides, “A new path-based cutting plane approach for the discrete time-cost tradeoff problem,” *Computational Management Science*, vol. 7, no. 3, pp. 313–336, 2010.
- [21] D. R. Robinson, “A dynamic programming solution to cost-time tradeoff for CPM,” *Management Science*, vol. 22, no. 2, pp. 158–166, 1975.
- [22] C. Akkan, A. Drexl, and A. Kimms, “Network decomposition-based benchmark results for the discrete time-cost tradeoff problem,” *European Journal of Operational Research*, vol. 165, no. 2, pp. 339–358, 2005.
- [23] M. Vanhoucke, “New computational results for the discrete time/cost trade-off problem with time-switch constraints,” *European Journal of Operational Research*, vol. 165, no. 2, pp. 359–374, 2005.
- [24] E. L. Demeulemeester, W. S. Herroelen, and S. E. Elmaghraby, “Optimal procedures for the discrete time/cost trade-off problem in project networks,” *European Journal of Operational Research*, vol. 88, no. 1, pp. 50–68, 1996.
- [25] S. S. Erengüç, S. Tufekci, and C. J. Zappe, “Solving time/cost trade-off problems with discounted cash flows using generalized Benders decomposition,” *Naval Research Logistics*, vol. 40, no. 1, pp. 25–50, 1993.
- [26] H. Mokhtari, R. Baradaran Kazemzadeh, and A. Salmasnia, “Time-cost tradeoff analysis in project management: an ant system approach,” *IEEE Transactions on Engineering Management*, vol. 58, no. 1, pp. 36–43, 2011.
- [27] W. David, O. N. Jorge, and M. Forrest, “Utility of genetic algorithms for solving large-scale construction time-cost trade-off problems,” *Journal of Computing in Civil Engineering*, vol. 32, no. 1, Article ID 04017072, 2018.
- [28] K. P. Anagnostopoulos and L. Kotsikas, “Experimental evaluation of simulated annealing algorithms for the time-cost trade-off problem,” *Applied Mathematics and Computation*, vol. 217, no. 1, pp. 260–270, 2010.
- [29] S. Aminbakhsh and R. Sonmez, “Discrete particle swarm optimization method for the large-scale discrete time-cost trade-off problem,” *Expert Systems with Applications*, vol. 51, pp. 177–185, 2016.
- [30] S. Aminbakhsh and R. Sonmez, “Pareto front particle swarm optimizer for discrete time-cost trade-off problem,” *Journal of Computing in Civil Engineering*, vol. 31, no. 1, Article ID 04016040, 2017.
- [31] Z. He, H. He, R. Liu, and N. Wang, “Variable neighbourhood search and tabu search for a discrete time/cost trade-off problem to minimize the maximal cash flow gap,” *Computers & Operations Research*, vol. 78, pp. 564–577, 2017.
- [32] Ö. Bettemir and M. Talat Birgönül, “Network analysis algorithm for the solution of discrete time-cost trade-off problem,” *KSCSE Journal of Civil Engineering*, vol. 21, no. 4, pp. 1047–1058, 2017.
- [33] D. A. Wood, “Gas and oil project time-cost-quality tradeoff: integrated stochastic and fuzzy multi-objective optimization applying a memetic, nondominated, sorting algorithm,” *Journal of Natural Gas Science and Engineering*, vol. 45, pp. 143–164, 2017.
- [34] M. Skutella, “Approximation algorithms for the discrete time-cost tradeoff problem,” *Mathematics of Operations Research*, vol. 23, no. 4, pp. 909–929, 1998.
- [35] A. B. Hafizoğlu and M. Azizoğlu, “Linear programming based approaches for the discrete time/cost trade-off problem in project networks,” *Journal of the Operational Research Society*, vol. 61, no. 4, pp. 676–685, 2010.
- [36] V. G. Deĭneko and G. J. Woeginger, “Hardness of approximation of the discrete time-cost tradeoff problem,” *Operations Research Letters*, vol. 29, no. 5, pp. 207–210, 2001.
- [37] M. Vanhoucke and D. Debels, “The discrete time/cost trade-off problem: extensions and heuristic procedures,” *Journal of Scheduling*, vol. 10, no. 4-5, pp. 311–326, 2007.
- [38] B. De Reyck and W. Herroelen, “A branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations,” *European Journal of Operational Research*, vol. 111, no. 1, pp. 152–174, 1998.
- [39] A. Kandil and K. El-Rayes, “Parallel genetic algorithms for optimizing resource utilization in large-scale construction projects,” *Journal of Construction Engineering and Management*, vol. 132, no. 5, pp. 491–498, 2006.
- [40] Ö. H. Bettemir, *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.
- [41] H. Li, Z. Xu, and W. Wei, “Bi-objective scheduling optimization for discrete time/cost trade-off in projects,” *Sustainability*, vol. 10, no. 8, p. 2802, 2018.
- [42] P. Leyman, N. Van Driessche, M. Vanhoucke, and P. De Causmaecker, “The impact of solution representations on heuristic net present value optimization in discrete time/cost trade-off project scheduling with multiple cash flow and payment models,” *Computers & Operations Research*, vol. 103, pp. 184–197, 2019.
- [43] J. D. Wiest, “A heuristic model for scheduling large projects with limited resources,” *Management Science*, vol. 13, no. 6, 1967.
- [44] J. Kerbosch and H. Schell, “Network planning by the extended metra potential method (EMPM),” Reports KS - 1.1, Department of Industrial Engineering, Group Operational Research, University of Technology Eindhoven, Eindhoven, Netherlands, 1975.
- [45] IBM, *Project Management System, Application Description Manual (H20-0210)*, IBM, New York, NY, USA, 1968.
- [46] K. Crandall, “Project planning with precedence lead/lag factors,” *Project Management Quarterly*, vol. 4, pp. 18–27, 1973.
- [47] K. Neumann and C. Schwindt, “Activity-on-node networks with minimal and maximal time lags and their application to make-to-order production,” *OR Spektrum*, vol. 19, no. 3, pp. 205–217, 1997.
- [48] S. E. Elmaghraby and J. Kamburowski, “The analysis of activity networks under generalized precedence relations (GPRs),” *Management Science*, vol. 38, no. 9, pp. 1245–1263, 1992.
- [49] M. Hajdu, “Survey of precedence relationships: classification and algorithms,” *Automation in Construction*, vol. 95, pp. 245–259, 2018.
- [50] R. C. Leachman, A. Dtnecerler, and S. Kim, “Resource-Constrained scheduling of projects with variable-intensity activities,” *IIE Transactions*, vol. 22, no. 1, pp. 31–40, 1990.

- [51] T. Kis, G. Erdős, and A. Márkus, "A project-oriented decision support system for production planning in make-to-order manufacturing," *ERCIM News*, vol. 58, pp. 66-67, 2004.
- [52] T. Kis, "A branch-and-cut algorithm for scheduling of projects with variable-intensity activities," *Mathematical Programming*, vol. 103, no. 3, pp. 515-539, 2005.
- [53] T. Kis, "RCPS with variable intensity activities and feeding precedence constraints, in perspectives in modern project scheduling," *International Series in Operations Research & Management Science*, vol. 92, pp. 105-129, Springer, New York, NY, USA, 2006.
- [54] L. Bianco and M. Caramia, "An exact algorithm to minimize the makespan in project scheduling with scarce resources and feeding precedence relations," Report RR-03.09, University of Rome Tor Vergata, Roma, Italy, 2009.
- [55] L. Bianco and M. Caramia, "Minimizing the completion time of a project under resource constraints and feeding precedence relations: a Lagrangian relaxation based lower bound," *4OR: A Quarterly Journal of Operations Research*, vol. 9, no. 4, pp. 371-389, 2011.
- [56] L. Bianco and M. Caramia, "Minimizing the completion time of a project under resource constraints and feeding precedence relations: an exact algorithm," *4OR: A Quarterly Journal of Operations Research*, vol. 10, no. 4, pp. 1-17, 2012.
- [57] J. Moder, C. Philips, and E. Davis, *Project Management with CPM, PERT and Precedence Diagramming*, Van Nostrand Reinhold, New York, NY, USA, 3rd edition, 1983.
- [58] V. Valls, R. Mart, and P. Lino, "A heuristic algorithm for project scheduling with splitting allowed," *Journal of Heuristics*, vol. 2, no. 1, pp. 87-104, 1996.
- [59] M. Hajdu, *Network Scheduling Techniques for Construction Project Management*, Springer, Dordrecht, Netherlands, 1997.
- [60] J. L. Ponz-Tienda, E. Pellicer, J. Benlloch-Marco, and C. Andrés-Romano, "The fuzzy project scheduling problem with minimal generalized precedence relations," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, no. 11, pp. 872-891, 2015.
- [61] S.-G. Kim, "CPM schedule summarizing function of the beeline diagramming method," *Journal of Asian Architecture and Building Engineering*, vol. 11, no. 2, pp. 367-374, 2012.
- [62] I. M. Srouf, M.-A. U. Abdul-Malak, A. A. Yassine, and M. Ramadan, "A methodology for scheduling overlapped design activities based on dependency information," *Automation in Construction*, vol. 29, pp. 1-11, 2013.
- [63] X. Zhang, Y. Li, S. Zhang, and C. M. Schlick, "Modelling and simulation of the task scheduling behavior in collaborative product development process," *Integrated Computer-Aided Engineering*, vol. 20, no. 1, pp. 31-44, 2013.
- [64] T.-K. Lim, C.-Y. Yi, D.-E. Lee, and D. Arditi, "Concurrent construction scheduling simulation algorithm," *Computer-Aided Civil and Infrastructure Engineering*, vol. 29, no. 6, pp. 449-463, 2014.
- [65] J. D. Wiest, "Precedence diagramming method: some unusual characteristics and their implications for project managers," *Journal of Operations Management*, vol. 1, no. 3, pp. 121-130, 1981.
- [66] J. Herbert and R. Deckro, "Combining contemporary and traditional project management tools to resolve a project scheduling problem," *Computers & Operations Research*, vol. 38, pp. 21-32, 2011.
- [67] V. Valls and P. Lino, "Criticality analysis in activity-on-node networks with minimal time lags," *Annals of Operations Research*, vol. 102, no. 1/4, pp. 17-37, 2001.
- [68] J. X. Qi and Z. X. Su, "Analysis of an anomaly: the increase in time float following consumption," *The Scientific World Journal*, vol. 2014, p. 415870, 2014.
- [69] Z. X. Su and H. Y. Wei, "New quantization approach for the anomaly: the increase in time float following consumption," *Mathematical Problems in Engineering*, vol. 2018, Article ID 8069202, 11 pages, 2018.
- [70] Z. Su, "Invisible consumptions and enlargements of activity floats under generalized precedence relations," *Soft Computing*, vol. 23, no. 21, pp. 10837-10852, 2019.
- [71] S. Sakellariopoulos and A. P. Chassiakos, "Project time-cost analysis under generalised precedence relations," *Advances in Engineering Software*, vol. 35, no. 10-11, pp. 715-724, 2004.
- [72] J. Son, T. Hong, and S. Lee, "A mixed (continuous + discrete) time-cost trade-off model considering four different relationships with lag time," *KSCE Journal of Civil Engineering*, vol. 17, no. 2, pp. 281-291, 2013.
- [73] M. Amiri, A. R. Abtahi, and K. K. Damghani, "Solving a generalised precedence multi-objective multi-mode time-cost-quality trade-off project scheduling problem using a modified NSGA-II algorithm," *International Journal of Services and Operations Management*, vol. 14, no. 3, pp. 355-372, 2013.
- [74] M. Tavana, A.-R. Abtahi, and K. Khalili-Damghani, "A new multi-objective multi-mode model for solving preemptive time-cost-quality trade-off project scheduling problems," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1830-1846, 2014.
- [75] K. Khalili-Damghani, M. Tavana, A.-R. Abtahi, and F. J. Santos Arteaga, "Solving multi-mode time-cost-quality trade-off problems under generalized precedence relations," *Optimization Methods and Software*, vol. 30, no. 5, pp. 965-1001, 2015.
- [76] E. N. Afruzi, A. A. Najafi, E. Roghanian, and M. Mazinani, "A Multi-Objective Imperialist Competitive Algorithm for solving discrete time, cost and quality trade-off problems with mode-identity and resource-constrained situations," *Computers & Operations Research*, vol. 50, pp. 80-96, 2014.
- [77] F. Mohammadipour and S. J. Sadjadi, "Project cost-quality-risk tradeoff analysis in a time-constrained problem," *Computers & Industrial Engineering*, vol. 95, pp. 111-121, 2016.
- [78] E. Demeulemeester, M. Vanhoucke, and W. Herroelen, "RanGen: a random network generator for activity-on-the-node networks," *Journal of Scheduling*, vol. 6, no. 1, pp. 17-38, 2003.
- [79] M. Vanhoucke, J. Coelho, D. Debels, B. Maenhout, and L. V. Tavares, "An evaluation of the adequacy of project network generators with systematically sampled networks," *European Journal of Operational Research*, vol. 187, no. 2, pp. 511-524, 2008.
- [80] H. R. Tareghian and S. H. Taheri, "A solution procedure for the discrete time, cost and quality tradeoff problem using electromagnetic scatter search," *Applied Mathematics and Computation*, vol. 190, no. 2, pp. 1136-1145, 2007.