Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2015, Article ID 539374, 17 pages
http://dx.doi.org/10.1155/2015/539374

Research Article

A Float-Path Theory and Its Application to the Time-Cost Tradeoff Problem

Zhi-xiong Su,[1] Jian-xun Qi,[2] and Han-ying Wei[1]

[1] *Business Administration College, Nanchang Institute of Technology, Nanchang 330099, China*
[2] *School of Economics and Management, North China Electric Power University, Beijing 102206, China*

Correspondence should be addressed to Zhi-xiong Su; suzhixiongbaner@126.com

Received 11 September 2014; Accepted 2 February 2015

Academic Editor: Quanke Pan

Activity floats are vital for project scheduling, such as total floats which determine the maximum permissible delays of activities. Moreover, activity paths in activity networks present essences of many project scheduling problems; for example, the time-cost tradeoff is to shorten long paths at lower costs. We discovered relationships between activity floats and paths and established a float-path theory. The theory helps to compute path lengths using activity floats and analyze activity floats using paths, which helps to transmute a problem into the other simpler one. We discussed applications of the float-path theory and applied it to solve the time-cost tradeoff problem (TCTP), especially the nonlinear and discrete versions. We proposed a simplification from an angle of path as a preprocessing technique for the TCTP. The simplification is a difficult path problem, but we transformed it into a simple float problem using the float-path theory. We designed a polynomial algorithm for the simplification, and then the TCTP may be solved more efficiently.

1. Introduction

A project is defined by a set of "real" activities and a set of immediate precedence relations. Based on this, DuPont company and Kelley and Walker [1] created activity networks to represent projects, such as CPM networks. The temporal analysis of the activity network resulted in the definition of several parameters, not the least important among which are the earliest and latest start and finish-time of an activity, which gave rise to the concept of activity floats. There are four such floats—total float, free float, safety float, and interference float [2–5]. These floats play an important role in two issues of central concern to managers: resource allocation and activity scheduling, since floats gave a measure of the flexibility in scheduling the activities during the project execution without delaying the project completion time [5].

In this paper, we researched the activity floats from the view of length instead of time and discovered their new properties—there are close relationships between floats and paths in activity networks. For example, the total float of an activity is equal to the difference between length of the critical path and length of the longest path passing the activity. The properties cause the activity floats and paths to be represented by each other, and we summarized them as a float-path theory. Paths are vital research subjects in most types of networks, and the activity network is no exception. Activity paths present essences of many project scheduling problems. For example, the essence of the time-cost tradeoff is to compress long paths to a required length at the lowest cost. The classic Fulkerson [6] algorithm is to compress the critical path and the compression in each step is determined by free floats of noncritical activities. Solving different problems needs considering different types of paths, and it may be difficult to find some types of paths. The float-path theory provides a new approach for path problems, that is, replacing finding paths by computing activity floats, even analyzing paths when changing the network's structure (such as adding, removing, or flipping arcs).

In this paper, we considered the application of the float-path theory to a classic project scheduling problem—time-cost tradeoff problem (TCTP), especially the nonlinear continuous and discrete versions. The continuous TCTP was proposed by Kelley [7] and contains linear and nonlinear TCTP. The nonlinear TCTP is more realistic but difficult than the linear one, and the nonlinear time-cost curve is a main research objective. The more general form of the time-cost curve was analyzed by Panagiotakoplos [8] and Moder et al. [9] following the earlier studies of the concave time-cost curve by Falk and Horowitz [10] and convex time-cost curve studies of others [11–15]. Their main ideas for the nonlinear TCTP were to approximate the nonlinear function by using a piecewise linear function and then solve the problem by computing the linear function. Later some other authors further developed the approaches [4, 16] and proposed new ones, for example, decision support model [17]. In addition to researching the time-cost curve, several heuristic algorithms are used to solve the nonlinear TCTP [18–23]. Discrete TCTP (DTCTP), the most difficult of the TCTPs, was proposed by Panagiotakoplos [8] and Harvey and Patterson [24]. De et al. [25] proved that the problem was strong nondeterministic polynomial-time hard (NP-hard). Many scholars designed heuristic algorithms for the problem. Skutella [26] first designed an approximation algorithm for the DTCTP, and some recent heuristic algorithms for the problem have been reported [27–34]. Akkan et al. [27], in particular, proposed a simplification for the problem, and their main approach was to compute path lengths, especially the shorter paths in the network. But the workload of directly computing all path lengths in a network is incredible, and the effect of this simplification may be not perfect. In addition, several exact methodologies are used to solve the problem with small-scale, including mathematical programming [35], dynamic programming [36, 37], branch and bound procedure [38], and robustness measures [23, 39].

We considered the large-scale nonlinear continuous TCTP and DTCTP from the idea of equivalent simplification. Simplification has a much larger benefit for the problems. For the nonlinear TCTP, narrowing the duration interval of an activity allows a more detailed approximation of the piecewise linear function to its nonlinear function. The measure can improve the effect and accuracy of solving the problem. The effectiveness of simplification will be more prominent to the DTCTP. The number of schemes of the DTCTP will increase exponentially as the scale of the problem increases, which causes the increase of the computation greatly. Conversely, in case of reducing activities, the number of schemes will decrease exponentially. Hence the simplification may cause a large-scale DTCTP to be solved using exact algorithms.

The float-path theory applies to the simplification. For instance, a target of the simplification is to remove paths shorter than the required length. There are numerous paths but fewer activities in a large-scale network. We can use the float-path theory to replace the computing path lengths by computing activity floats and remove numerous redundant paths by removing several activities. This is similar to the effect of the analytic geometry, which substitutes quadratic functions for complex cone curve. Based on the float-path theory, we designed a polynomial algorithm to simplify the TCTPs. The preprocessing technique may help to solve the TCTP and other project scheduling problems.

2. Time Parameter in a CPM Network

2.1. Time Parameter of Node. In a CPM network (activity-on-arc representation), let nodes (1) and (n) be the start and terminal nodes of the network, and let d_{ij} denote the duration of the activity (i, j), and ET_i and LT_i are the earliest and latest times of the node (i); then let $j = 2, 3, \ldots, n$,

$$\begin{aligned}\mathrm{ET}_1 &= 0,\\ \mathrm{ET}_j &= \max_i \left\{\mathrm{ET}_i + d_{ij}\right\}\end{aligned} \tag{1}$$

and let $j = n-1, n-2, \ldots, 1$,

$$\begin{aligned}\mathrm{LT}_n &= \mathrm{ET}_n,\\ \mathrm{LT}_j &= \min_k \left\{\mathrm{LT}_k - d_{jk}\right\}.\end{aligned} \tag{2}$$

Length $L(\mu^{\nabla})$ of the critical path, μ^{∇}, in the network is

$$L\left(\mu^{\nabla}\right) = \mathrm{ET}_n = \mathrm{LT}_n. \tag{3}$$

2.2. Activity Float. The activity floats mainly include total, free, safety, and interference floats and are described below in relation to an activity (i, j).

Total Float. The total float (TF_{ij}) is defined as

$$\mathrm{TF}_{ij} = \mathrm{LT}_j - \mathrm{ET}_i - d_{ij}. \tag{4}$$

It denotes the maximum permissible delay of the activity (i, j) without delaying the project duration.

Free Float. The free float (FF_{ij}) is computed as

$$\mathrm{FF}_{ij} = \mathrm{ET}_j - \mathrm{ET}_i - d_{ij}. \tag{5}$$

It denotes the maximum permissible delay of the activity (i, j) when all its succeeding activities start as early as possible and all its preceding activities finish as early as possible.

Safety Float. The safety float (SF_{ij}) is defined by

$$\mathrm{SF}_{ij} = \mathrm{LT}_j - \mathrm{LT}_i - d_{ij}. \tag{6}$$

It denotes the maximum permissible delay of the activity (i, j) when all its preceding activities start as late as possible and all its succeeding activities finish as late as possible.

Interference Float. The interference float (IF_{ij}) is computed as

$$\mathrm{TF}_{ij} = \mathrm{ET}_j - \mathrm{LT}_i - d_{ij}. \tag{7}$$

If the interference float is positive, then it describes the maximum permissible delay when all its succeeding activities start as early as possible and all its preceding activities finish as late as possible. In case it is negative, the IF is the minimum required shortening of the duration of the activity to allow all its succeeding activities to finish as early as possible and all its preceding activities to finish as late as possible.

3. Float-Path Theory in a CPM Network

Float-path theory reflects the property of the CPM network structure, and it mainly contains the relationships between activity floats and path lengths. We summarize them in the following theorems.

Theorem 1. *Suppose $\mu_{i\to j}$ is a path from the node (i) to node (j); then its length, $L(\mu_{i\to j})$, is*

$$\begin{aligned} L(\mu_{i\to j}) &= ET_j - ET_i - \sum_{(e,f)\in\mu_{i\to j}} FF_{ef} \\ &= LT_j - LT_i - \sum_{(e,f)\in\mu_{i\to j}} SF_{ef} \\ &= LT_j - ET_i - TF_{uv} - \sum_{(e,f)\in\mu_{i\to u}} FF_{ef} \\ &\quad - \sum_{(r,s)\in\mu_{v\to j}} SF_{rs} \\ &= LT_j - ET_i - TF_k - \sum_{(e,f)\in\mu_{i\to k}} FF_{ef} \\ &\quad - \sum_{(r,s)\in\mu_{k\to j}} SF_{rs}, \end{aligned} \tag{8}$$

where $(u,v)\in\mu_{i\to j}$ and $(k)\in\mu_{i\to j}$.

Proof. See Appendix A. □

Theorem 2. *For any node, (i), its earliest time, ET_i, is equal to length of the longest path, $\mu^{\nabla}_{1\to i}$, from the start node (1) of the network; that is,*

$$ET_i = L(\mu^{\nabla}_{1\to i}), \tag{9}$$

and its latest time, LT_i, is equal to the difference between length of the critical path, μ^{∇}, and length of the longest path, $\mu^{\nabla}_{i\to n}$, from it to the end node (n) of the network; that is,

$$LT_i = L(\mu^{\nabla}) - L(\mu^{\nabla}_{i\to n}). \tag{10}$$

Proof. See Appendix B. □

Corollary 3. *For any node, (i), the free float, FF_{ef}, of any activity (e,f) on the longest path $\mu^{\nabla}_{1\to i}$ from the start node (1) of the network to it is equal to 0; that is,*

$$FF_{ef} = 0, \quad (e,f)\in\mu^{\nabla}_{1\to i}, \tag{11}$$

and the safety float, SF_{rs}, of any activity (r,s) on the longest path $\mu^{\nabla}_{i\to n}$ from it to the end node (n) of the network is also equal to 0; that is,

$$SF_{rs} = 0, \quad (r,s)\in\mu^{\nabla}_{i\to n}. \tag{12}$$

Proof. See Appendix C. □

Theorem 4. *For any activity, (i,j), its total float, TF_{ij}, is equal to the difference between length of the critical path, μ^{∇}, and length of the longest path, μ^{∇}_{ij}, passing it; that is,*

$$TF_{ij} = L(\mu^{\nabla}) - L(\mu^{\nabla}_{ij}). \tag{13}$$

Proof. See Appendix D. □

Theorem 5. *For any activity, (i,j), its free float, FF_{ij}, is equal to the difference between length of the longest path, μ^{∇}_{j}, passing its end node (j) and length of the longest path, μ^{∇}_{ij}, passing it; that is,*

$$FF_{ij} = L(\mu^{\nabla}_{j}) - L(\mu^{\nabla}_{ij}). \tag{14}$$

Proof. See Appendix E. □

Theorem 6. *For any activity, (i,j), its safety float, SF_{ij}, is equal to the difference between length of the longest path, μ^{∇}_{i}, passing its start node (i) and length of the longest path, μ^{∇}_{ij}, passing it; that is,*

$$SF_{ij} = L(\mu^{\nabla}_{i}) - L(\mu^{\nabla}_{ij}). \tag{15}$$

Proof. It is similar to the proof of Theorem 5. □

Theorem 7. *For any activity, (i,j), its interference float, IF_{ij}, can be represented as follows: add an arc (j,i) with length $d_{ji} = d_{ij}$, and IF_{ij} is equal to the value of that length of the longest path, μ^{∇}_{ji}, passing the arc (j,i) once minus length of the critical path μ^{∇} of the old network, and minus double duration d_{ij} of the activity (i,j); that is,*

$$IF_{ij} = L(\mu^{\nabla}_{ji}) - L(\mu^{\nabla}) - 2d_{ij}. \tag{16}$$

If $FF_{ij} > 0$ and $SF_{ij} > 0$, the equation is still correct after flipping the arc (i,j).

Proof. See Appendix F. □

The float-path theory helps to solve many path problems by transforming the computations of path lengths into the computations of activity floats. For example, according to Theorem 2 and Corollary 3, we can find the longest path from the start node (1) to any node and the longest path from any node to the end node (n); according to Theorem 4 and Corollary 3, we can find the longest path passing any activity; and according to Theorem 7, we can analyze paths when changing the network's structure (e.g., adding or flipping arcs).

4. Application of the Float-Path Theory to TCTP

4.1. The TCTP. Assume in a CPM network G that d_{ij} denotes the duration of each activity (i,j) and $l_{ij} \le d_{ij} \le u_{ij}$, where $f_{ij}(d_{ij})$ denotes the cost of duration (d_{ij}) and is a monotonic decreasing function, and t_i denotes the time of each node

(i) and T denotes project duration that $T = t_n - t_1$. The continuous TCTP can be formulated as follows (λ denotes the required project duration):

$$\begin{aligned} \min \quad & \sum_{(i,j)\in G} f_{ij}\left(d_{ij}\right) \\ \text{s.t.} \quad & t_j - t_i \geq d_{ij} \\ & l_{ij} \leq d_{ij} \leq u_{ij} \\ & t_n - t_1 \leq \lambda. \end{aligned} \tag{17}$$

And assume that d_{ijm} and c_{ijm}, respectively, denote the duration and cost of mode m of each activity (i, j), and if $d_{ijm_1} \geq d_{ijm_2}$ then $c_{ijm_1} \leq c_{ijm_2}$ for all $m \in M_{ij}$. The DTCTP can be formulated as follows:

$$\begin{aligned} \min \quad & \sum_{(i,j)\in G} \sum_{m\in M_{ij}} c_{ijm} x_{ijm} \\ \text{s.t.} \quad & \sum_{m\in M_{ij}} x_{ijm} = 1, \quad (i,j) \in G \\ & t_j - t_i - \sum_{m\in M_{ij}} d_{ijm} x_{ijm} \geq 0 \\ & t_n - t_1 \leq \lambda \\ & t_i \geq 0, \quad (i) \in G \\ & x_{ijm} \in \{0,1\}, \quad (i,j) \in G, \quad m \in M_{ij}. \end{aligned} \tag{18}$$

In an activity network, the project duration is equal to the length of the critical path; therefore in solving the TCTP the solution is to make all the paths no longer than the required length λ at a minimum total cost.

4.2. Principles of Equivalent Simplification of the TCTP. We first consider reducing the scale of the problem and propose Proposition 8 of the equivalent simplification in combination with the CPM network.

Proposition 8. *If all activities choose their longest durations and a path is no longer than λ, then in all cases the path will be no longer than λ so that can be eliminated from our consideration in solving the TCTP. Therefore, the path can be removed. However, because removing a path may cause the removal of other paths, keeping paths longer than λ is essential when removing the shorter ones.*

However, it is difficult to directly implement the proposition, and we simplify it to Proposition 9.

Proposition 9. *If all activities choose their longest durations and the longest path passing one activity (i, j) is no longer than λ, then in all cases all paths passing the activity will be no longer than λ. Thus, the activity (i, j) is a redundant activity and can be removed, and its optimal duration is the cheapest duration.*

Furthermore, we consider simplifying parameters of each activity and propose Propositions 10~12 of the equivalent simplification.

Proposition 10. *If activity (i, j) chooses a duration d_{ij} and the longest path passing the activity is longer than λ, even if all the other activities choose their shortest durations, then it will in all cases make the project duration longer than λ, and d_{ij} is not the optimal solution. On this basis, d_{ij} is a redundant duration of the activity (i, j) which is called redundant-long duration and can be removed.*

Proposition 11. *If one activity chooses a duration, the longest path passing the activity will be no longer than λ, even if all other activities choose their longest durations; therefore, in all cases the duration is unable to make project duration longer than λ, and one names it as short duration.*

Proposition 12. *For all the short durations of an activity (i, j), the total cost will be lower when choosing the cheapest duration d_{ij} compared to other choices; therefore, the minimum total cost will not be realized when using the short durations other than d_{ij}, and the other durations are not optimal solutions. Thence, these short durations are redundant durations of the activity and can be removed. One names them as redundant-short durations.*

4.3. Equivalent Simplification of TCTP Using the Float-Path Theory. Theorems 13~15 describe how to determine the redundant objectives.

Theorem 13. *Let each activity (i, j) choose the longest duration $d_{ij} = u_{ij}$; then compute its total float TF_{ij} and the project duration T. If $TF_{ij} \geq T - \lambda$, then the activity (i, j) is a redundant activity.*

Proof. See Appendix G. □

Theorem 14. *Let each activity (i, j) choose its shortest duration $d_{ij} = l_{ij}$; then add an assistant activity $(1, n)$ with duration $d_{1n} = \lambda$, and compute the total float TF_{ij}. If $LT_j - ET_i < u_{ij}$, then each duration d_{ij} meeting $d_{ij} > LT_j - ET_i$ is a redundant-long duration of the activity (i, j).*

Proof. See Appendix H. □

Theorem 15. *Let each activity (i, j) choose its longest duration $d_{ij} = u_{ij}$, and compute its total float TF_{ij} and project duration T. If $TF_{ij} < T - \lambda$ and $l_{ij} \leq LT_j - ET_i - T + \lambda$, then each duration d_{ij} meeting $d_{ij} \leq LT_j - ET_i - T + \lambda$ is a short duration of the activity (i, j). Except for the cheapest short duration for an activity, the others are redundant-short durations.*

Proof. See Appendix I. □

4.4. Algorithm of Equivalent Simplification of the TCTP. Suppose there are n nodes in a corresponding CPM network for a given project, indicated by G, with the start and terminal nodes marked as (1) and (n); then we can simplify the TCTP of the project as follows.

Step 1. Add assistant activity $(1, n)$ with duration $d_{1n} = u_{1n} = l_{1n} = \lambda$.

Step 2. Let $d_{ij} = l_{ij}$ of each activity (i, j), and compute time parameters.

Step 3. Remove duration of activity (i, j) that is longer than $\mathrm{LT}_j - \mathrm{ET}_i$.

Step 4. Let $d_{ij} = \min\{\mathrm{LT}_j - \mathrm{ET}_i, u_{ij}\}$ of each activity (i, j), and compute time parameters.

Step 5. Remove activity (i, j) whose $\mathrm{TF}_{ij} \geq \mathrm{ET}_n - \lambda$.

Step 6. For duration of each remaining activity (i, j) which is no longer than $\mathrm{LT}_j - \mathrm{ET}_i - \mathrm{ET}_n + \lambda$, preserve the longest one and remove the others.

Proof. See Appendix J. □

The complexity of the algorithm is $O(m)$, where m indicates quantity of activities (see Appendix K). The effectiveness of the algorithm depends on the comparisons among the length of the critical path, the length of the longest path passing each activity, and the required length λ. For example, when most paths are much shorter than the critical path (e.g., most activities have large total floats), or the difference between λ and the length of the critical path is small, the simplification may be more effective because it may remove more activities.

After the equivalent simplification of the network G, the remaining network is marked as G^*. The optimal solution of the TCTP can be solved using the network G^*: (1) the optimal duration of each activity in the network G^* can be obtained by using existing algorithms, such as the algorithms in References; (2) for each redundant activity (i, j) in Steps 5 and 6 of the simplification, its duration is unrelated to λ so that the optimal duration is the current longest (cheapest) one b_{ij}^* for the minimum total cost.

5. Illustration

5.1. Equivalent Simplification of the Nonlinear Continuous TCTP.

The detailed information for a project with 116 activities is shown in Table 1 (the cost unit is dollar and time unit is day). The start-time of any activity is no earlier than the finish-time of its immediate predecessor activity, and the cost function of each activity's duration is nonlinear and monotonically decreasing. How do we simplify the problem and improve the effect and accuracy of the solution if the demand is to shorten the project duration to 520 ($\lambda = 520$) at a minimum cost?

We use our algorithm to simplify the nonlinear time-cost tradeoff problem as follows.

Step 1. Represent the project as a CPM network based on Table 1 (let $d_{ij} = l_{ij}$), and assist activity $(1, 52)$ with duration $d_{1,52} = 520$.

Step 2. Let $d_{ij} = l_{ij}$ of each activity (i, j), and compute time parameters, as in Figure 1.

Table 1: The information of each activity.

Activity code	Duration interval	Immediate successor	Cost function $f(x)$
a_1	[60, 70]	b_1, b_2	$-x^3 + 60x^2 + 5000x$
a_2	[55, 60]	b_3, b_4, b_5	$x^2 - 140x + 5000$
a_3	[45, 50]	b_6, b_7, b_8	$-x^3 + 45x^2 + 8000$
a_4	[50, 55]	b_9, b_{10}, b_{11}	$-x^2 + 60x$
a_5	[70, 100]	b_{12}, b_{13}, b_{14}	$-x^3 + 70x^2 + 4400x$
a_6	[80, 120]	b_{15}, b_{16}	$x^2 - 260x + 16900$
b_1	[70, 75]	e_1, e_2	$x^2 - 180x + 8100$
b_2	[30, 35]	c_1, c_2	$-x^3 + 30x^2 + 400x$
b_3	[40, 45]	c_1, c_2	$x^2 - 120x + 3600$
b_4	[70, 75]	e_3, e_4, e_5	$-x^2 + 90x$
b_5	[25, 30]	c_3, c_4	$-x^3 + 25x^2 + 4500x$
b_6	[30, 33]	c_3, c_4	$-x^3 + 30x^2 + 400x$
b_7	[65, 70]	e_6, e_7, e_8	$x^2 - 160x + 6400$
b_8	[30, 33]	c_5, c_6	$-x^3 + 25x^2 + 12250$
b_9	[25, 28]	c_5, c_6	$-x^3 + 25x^2 + 4500$
b_{10}	[55, 60]	e_9, e_{10}, e_{11}	$x^2 - 160x + 6400$
b_{11}	[30, 33]	c_7, c_8	$-x^3 + 30x^2 + 50000$
b_{12}	[20, 23]	c_7, c_8	$-x^3 + 20x^2 + 9000$
b_{13}	[80, 95]	e_{12}, e_{13}, e_{14}	$x^2 - 200x + 10000$
b_{14}	[40, 80]	c_9, c_{10}	$-x^3 + 40x^2 + 4500x$
b_{15}	[30, 60]	c_9, c_{10}	$-x^3 + 25x^2 + 2450x$
b_{16}	[80, 115]	e_{15}, e_{16}	$-x^2 + 140x - 100$
c_1	[35, 40]	e_1, e_2	$-x^3 + 35x^2 + 750x$
c_2	[25, 30]	e_3, e_4, e_5	$-x^3 + 20x^2 + 1500x$
c_3	[30, 35]	e_3, e_4, e_5	$-x^3 + 30x^2 + 16000$
c_4	[25, 27]	e_6, e_7, e_8	$-x^3 + 20x^2 + 9000$
c_5	[35, 37]	e_6, e_7, e_8	$-x^3 + 35x^2 + 200x$
c_6	[30, 32]	e_9, e_{10}, e_{11}	$-x^3 + 30x^2 + 400x$
c_7	[20, 22]	e_9, e_{10}, e_{11}	$-x^3 + 15x^2 + 450x$
c_8	[30, 32]	e_{12}, e_{13}, e_{14}	$x^2 - 100x + 2500$
c_9	[40, 42]	e_{12}, e_{13}, e_{14}	$2x^2 - 480x + 7200$
c_{10}	[45, 85]	e_{15}, e_{16}	$-x^2 + 100x + 1100$
e_1	[65, 70]	g_1, g_2	$2x^2 - 320x + 12800$
e_2	[30, 35]	f_1, f_2	$-x^3 + 30x^2 + 400x$
e_3	[35, 37]	f_1, f_2	$-3x^3 + 90x^2 + 3000x$
e_4	[60, 65]	g_3, g_4, g_5	$x^2 - 140x + 4900$
e_5	[30, 32]	f_3, f_4	$x^4 - 80x^3 + 1600x^2$
e_6	[25, 27]	f_3, f_4	$x^4 - 60x^3 + 900x^2$
e_7	[55, 60]	g_6, g_7, g_8	$-x^3 + 55x^2 + 1050x$
e_8	[20, 22]	f_5, f_6	$-x^3 + 600x + 150$
e_9	[25, 27]	f_5, f_6	$2x^4 - 120x^3 + 900x^2$
e_{10}	[50, 55]	g_9, g_{10}, g_{11}	$-x^3 + 50x^2 + 600x$
e_{11}	[20, 22]	f_7, f_8	$x^4 - 60x^3 + 900x^2$
e_{12}	[25, 27]	f_7, f_8	$-x^3 + 25x^2 + 600x$
e_{13}	[20, 25]	g_{12}, g_{13}, g_{14}	$x^4 - 56x^3 + 784x^2$
e_{14}	[45, 90]	f_9, f_{10}	$2x^2 - 400x + 20000$
e_{15}	[40, 75]	f_9, f_{10}	$3x^2 - 480x + 19200$

Table 1: Continued.

Activity code	Duration interval	Immediate successor	Cost function $f(x)$
e_{16}	$[85, 130]$	g_{15}, g_{16}	$2x^2 - 600x + 45000$
f_1	$[30, 35]$	g_1, g_2	$-x^3 + 30x^2 + 675x$
f_2	$[25, 28]$	g_3, g_4, g_5	$2x^4 - 120x^3 + 1800x^2$
f_3	$[30, 33]$	g_3, g_4, g_5	$-2x^3 + 60x^2 + 700x$
f_4	$[30, 33]$	g_6, g_7, g_8	$x^4 - 80x^3 + 1600x^2$
f_5	$[35, 38]$	g_6, g_7, g_8	$-3x^3 + 105x^2 + 600x$
f_6	$[25, 28]$	g_9, g_{10}, g_{11}	$x^4 - 60x^3 + 900x^2$
f_7	$[25, 28]$	g_9, g_{10}, g_{11}	$2x^4 - 160x^3 + 4000x^2$
f_8	$[30, 33]$	g_{12}, g_{13}, g_{14}	$2x^4 - 200x^3 + 5000x^2$
f_9	$[35, 35]$	g_{12}, g_{13}, g_{14}	300000
f_{10}	$[40, 75]$	g_{15}, g_{16}	$3x^2 - 540x + 24300$
g_1	$[70, 75]$	p_1, p_2	$2x^2 - 340x + 14450$
g_2	$[35, 40]$	h_1, h_2	$-3x^3 + 105x^2 + 2250x$
g_3	$[30, 43]$	h_1, h_2	$x^2 - 100x + 2500$
g_4	$[65, 70]$	p_3, p_4, p_5	$-x^3 + 65x^2 + 1200x$
g_5	$[25, 28]$	h_3, h_4	$3x^4 - 180x^3 + 2700x^2$
g_6	$[25, 28]$	h_3, h_4	$2x^4 - 160x^3 + 3200x^2$
g_7	$[60, 65]$	p_6, p_7, p_8	$x^2 - 140x + 4900$
g_8	$[30, 33]$	h_5, h_6	$3x^4 - 210x^3 + 3675x^2$
g_9	$[25, 28]$	h_5, h_6	$2x^4 - 200x^3 + 5000x^2$
g_{10}	$[60, 65]$	p_9, p_{10}, p_{11}	$3x^2 - 280x + 9800$
g_{11}	$[15, 18]$	h_7, h_8	$-x^5 + 15x^4 + 2000x^2$
g_{12}	$[30, 30]$	h_7, h_8	60000
g_{13}	$[20, 25]$	p_{12}, p_{13}, p_{14}	$5x^4 - 3000x^3 + 4500x^2$
g_{14}	$[30, 35]$	h_9, h_{10}	$-2x^3 + 60x^2 + 800x$
g_{15}	$[35, 65]$	h_9, h_{10}	$x^2 - 140x + 4900$
g_{16}	$[80, 125]$	p_{15}, p_{16}	$x^2 - 280x + 19600$
h_1	$[30, 35]$	p_1, p_2	$-x^3 + 30x^2 + 16000$
h_2	$[25, 27]$	p_3, p_4, p_5	$4x^4 - 280x^3 + 4900x^2$
h_3	$[30, 32]$	p_3, p_4, p_5	$-2x^3 + 60x^2 + 12250$
h_4	$[30, 32]$	p_6, p_7, p_8	$-3x^3 + 90x^2 + 525x$
h_5	$[25, 27]$	p_6, p_7, p_8	$3x^4 - 180x^3 + 2700$
h_6	$[35, 37]$	p_9, p_{10}, p_{11}	$-2x^3 + 70x^2 + 16000$
h_7	$[25, 27]$	p_9, p_{10}, p_{11}	$x^4 - 80x^3 + 64000x$
h_8	$[25, 25]$	p_{12}, p_{13}, p_{14}	80000
h_9	$[40, 60]$	p_{12}, p_{13}, p_{14}	$x^2 - 130x + 4225$
h_{10}	$[40, 80]$	p_{15}, p_{16}	$x^2 - 170x + 7225$
p_1	$[60, 65]$	v_1	$3x^2 - 450x + 16875$
p_2	$[30, 35]$	r_1, r_2	$-x^3 + 30x^2 + 16000$
p_3	$[35, 37]$	r_1, r_2	$-2x^3 + 70x^2 + 800x$
p_4	$[65, 70]$	v_2	$2x^2 - 360x + 16200$
p_5	$[30, 32]$	r_3, r_4	$3x^4 - 240x^3 + 4800x^2$
p_6	$[25, 27]$	r_3, r_4	$-x^5 + 25x^4 + 300x^3$
p_7	$[50, 55]$	v_3	$-2x^3 + 100x^2 + 72000$
p_8	$[15, 17]$	r_5, r_6	$-x^5 + 225x^3 + 2000x^2$
p_9	$[20, 22]$	r_5, r_6	$2x^4 - 100x^3 + 1250x^2$
p_{10}	$[55, 60]$	v_4	$-x^3 + 55x^2 + 42250$
p_{11}	$[15, 17]$	r_7, r_8	$-2x^5 + 450x^3 + 4000x^2$
p_{12}	$[20, 22]$	r_7, r_8	$3x^4 - 150x^3 + 2250x^2$
p_{13}	$[8, 10]$	v_5	$3x^6 - 90x^5 + 675x^4$
p_{14}	$[30, 35]$	r_9, r_{10}	$-2x^3 + 60x^2 + 800x$
p_{15}	$[35, 65]$	r_9, r_{10}	$x^2 - 160x + 6400$
p_{16}	$[95, 140]$	v_6	$x^2 - 300x + 22500$
r_1	$[25, 30]$	v_1	$-x^3 + 25x^2 + 24000$
r_2	$[35, 38]$	v_2	$-2x^3 + 70x^2 + 900x$
r_3	$[30, 33]$	v_2	$-4x^3 + 120x^2 + 1600x$
r_4	$[25, 28]$	v_3	$-2x^3 + 50x^2 + 50000$
r_5	$[30, 33]$	v_3	$-6x^3 + 180x^2 + 2400x$
r_6	$[25, 28]$	v_4	$-3x^3 + 75x^2 + 75000$
r_7	$[30, 33]$	v_4	$-4x^3 + 120x^2 + 1600x$
r_8	$[25, 28]$	v_5	$-4x^3 + 100x^2 + 2200x$
r_9	$[40, 45]$	v_5	$2x^4 - 240x^3 + 432000x$
r_{10}	$[55, 90]$	v_6	$x^2 - 200x + 10000$
v_1	$[75, 80]$	—	$2x^2 - 360x + 19200$
v_2	$[50, 55]$	—	$-3x^3 + 150x^2 + 108000$
v_3	$[55, 60]$	—	$2x^2 - 400x + 20000$
v_4	$[40, 45]$	—	$0.5x^4 - 60x^3 + 108000$
v_5	$[85, 90]$	—	$2x^2 - 440x + 24200$
v_6	$[90, 150]$	—	$x^2 - 400x + 40000$

Step 3. For activity a_5, remove duration longer than $\mathrm{LT}_6 - \mathrm{ET}_1 = 85$; that is, remove duration interval $[85, 100]$, and let $u_{a_5} := 85$; similarly, remove duration interval $[90, 120]$ of activity a_6, duration interval $[55, 80]$ of activity b_{14}, duration interval $[45, 60]$ of activity b_{15}, duration interval $[90, 115]$ of activity b_{16}, duration interval $[60, 85]$ of activity c_{10}, duration interval $[65, 90]$ of activity e_{14}, duration interval $[55, 75]$ of activity e_{15}, duration interval $[95, 130]$ of activity e_{16}, duration interval $[55, 75]$ of activity f_{10}, duration interval $[50, 65]$ of activity g_{15}, duration interval $[90, 125]$ of activity g_{16}, duration interval $[55, 80]$ of activity h_{10}, duration interval $[50, 65]$ of activity p_{15}, duration interval $[105, 140]$ of activity p_{16}, duration interval $[70, 90]$ of activity r_{10}, and duration interval $[100, 150]$ of activity v_6.

Step 4. Let $d_{ij} = u_{ij}$, and compute time parameters, as in Figure 2.

Step 5. Remove activities whose $\mathrm{TF}_{ij} \geq \mathrm{ET}_{52} - \lambda = 635 - 520 = 115$, as in Figure 3.

Step 6. There are no durations shorter than $\mathrm{LT}_j - \mathrm{ET}_i - \mathrm{ET}_{52} + \lambda$ of each remaining activity (i, j) in Figure 3; therefore we could not remove them.

After the equivalent simplification, a network G^* is acquired (Figure 4). Obviously, network G^* is much simpler than the old network (Figure 1), and duration intervals of many remaining activities are significantly narrowed.

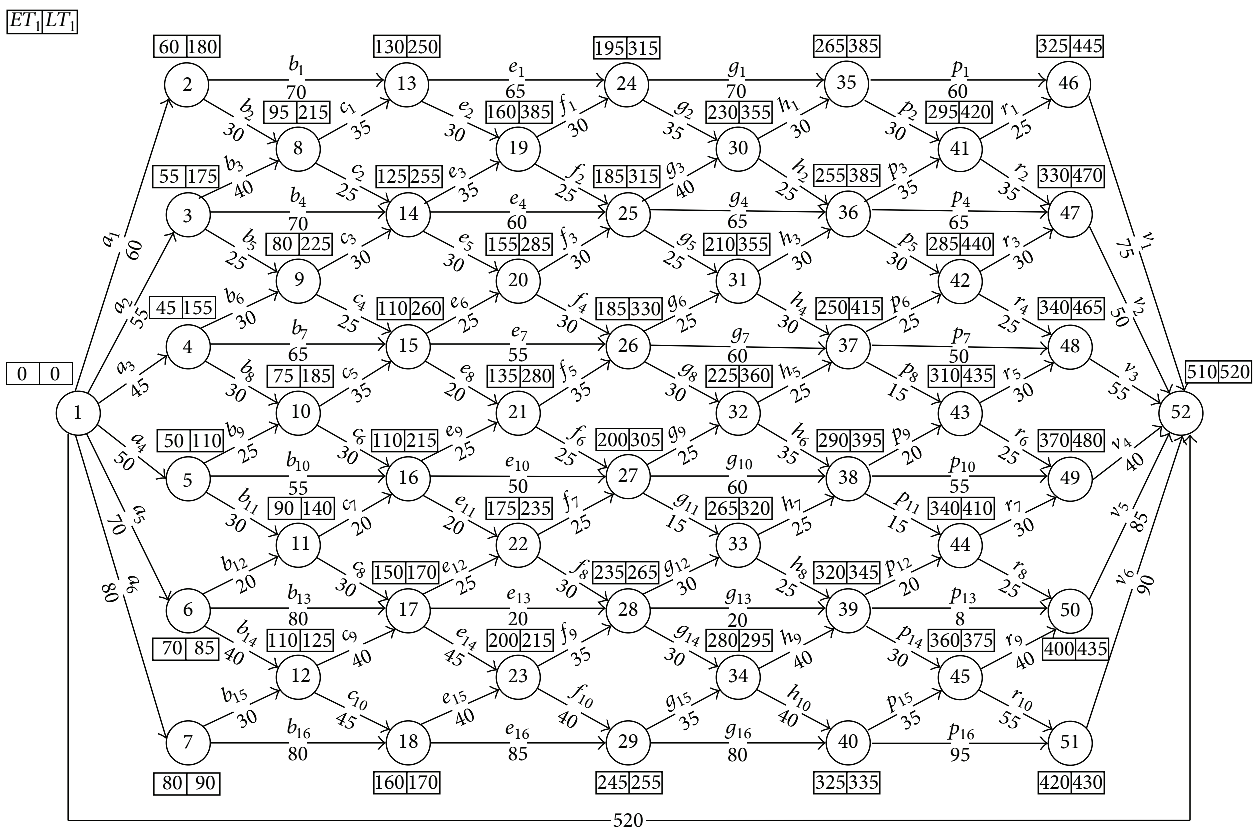

Figure 1: Time parameters for all activities with their shortest durations.

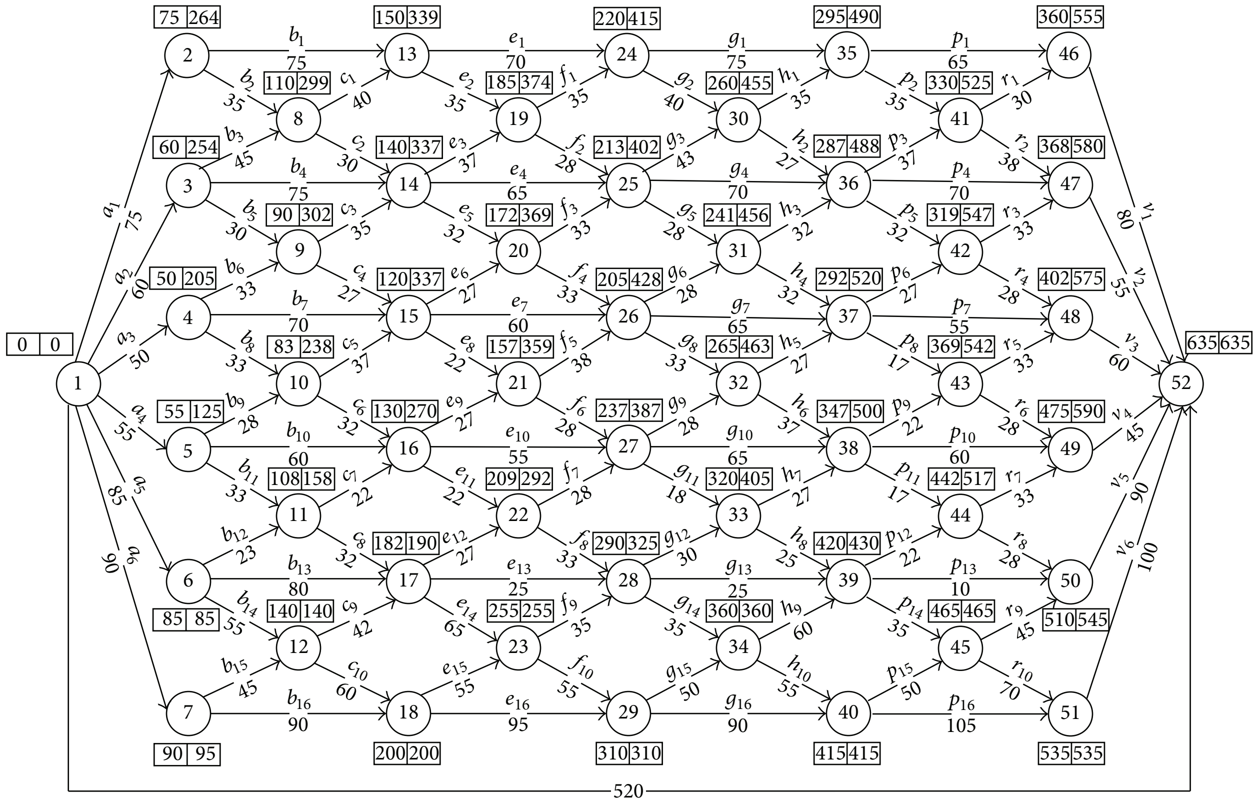

Figure 2: Time parameters for all activities with their remaining longest durations.

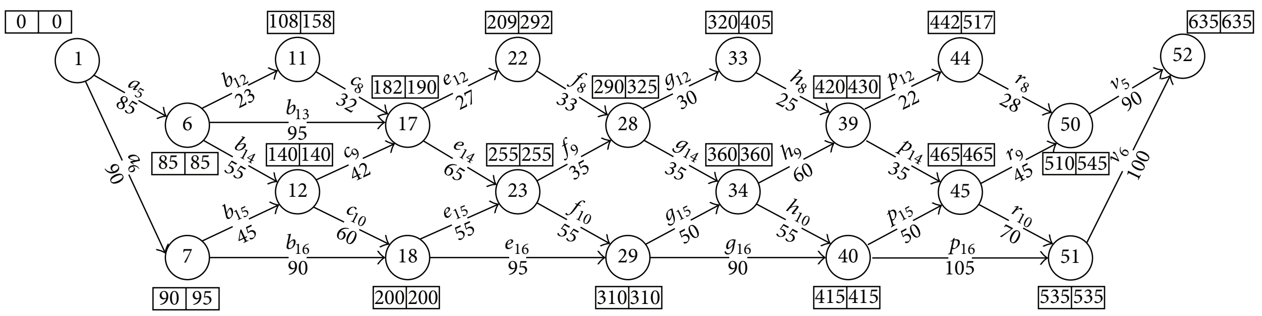

Figure 3: Network with nonredundant activities.

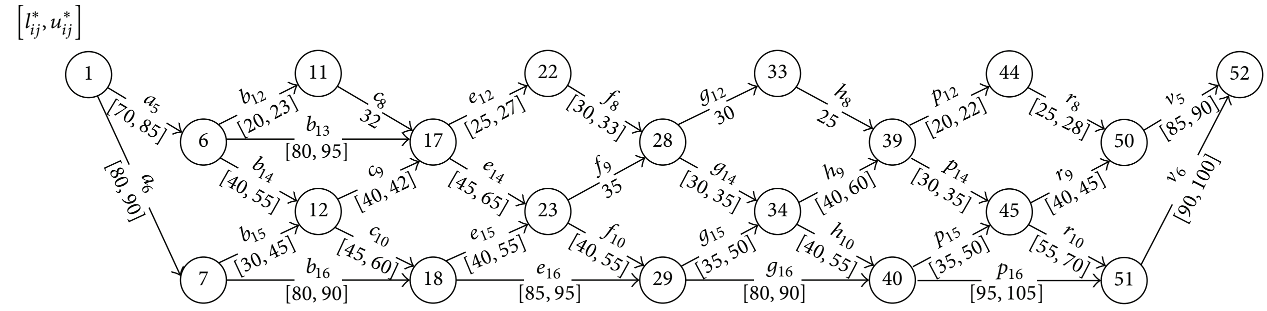

Figure 4: Network G^*.

Next we further solved the nonlinear TCTP that computed the optimal durations of activities using the mathematical model in Section 4.1 and intelligent algorithm software, such as LINDO. The experiment was performed on a PC (1 CPU, Intel 2.0 GHz, 1 G RAM) using the Windows XP operating system.

(i) We first solved the problem without simplification and obtained the project cost of 749,351,200. The computation time was ~30 min.

(ii) And then we solved the problem after the simplification and obtained the project cost of 749,106,282, which is 2,449,108 cheaper than the former. The computation time was ~2 sec, which is also much faster than the former.

This experiment further tests and verifies that the equivalent simplification could improve the effect and accuracy of the solution of the nonlinear continuous TCTP.

5.2. Equivalent Simplification of the DTCTP. The detailed information for a project with 88 activities is shown in Table 2 (similarly, the cost unit is dollar and time unit is day). The start-time of any activity is no earlier than the finish-time of its immediate predecessor activity, and each activity has three time-cost modes. If the project must be finished within 385 ($\lambda = 520$) at a minimum cost, then the amount of schemes is $3^{88} \approx 10^{42}$. What is the equivalent simplification to improve scheduling effect?

Step 1. Represent the project as an activity network based on Table 2 (let each activity choose mode 1), and assist activity $(1, 37)$ with duration $d_{1,37} = 385$.

Step 2. Let $d_{ij} = l_{ij}$ of each activity (i, j), and compute time parameters, as in Figure 5.

Step 3. For activity a_3, remove modes 2 and 3 whose durations are longer than $\mathrm{LT}_4 - \mathrm{ET}_1 = 48$; similarly, remove modes 2 and 3 of activity d_{13}, mode 3 of activity e_{13}, modes 2 and 3 of activity f_{13}, mode 3 of activity g_{12}, and modes 2 and 3 of activity h_5.

Step 4. Let each activity in Figure 5 choose the mode with the longest duration from its remaining modes, and compute time parameters, as in Figure 6.

Step 5. Remove activities whose $\mathrm{TF}_{ij} \geq \mathrm{ET}_{37} - \lambda = 392 - 385 = 7$, as in Figure 7.

Step 6. There are no durations shorter than $\mathrm{LT}_j - \mathrm{ET}_i - \mathrm{ET}_{37} + \lambda$ within the remaining modes of each activity in Figure 7; therefore we could not remove them.

Network G^* is obtained by the above simplification and shown in Figure 8, and brackets around each activity denote its choice of time-cost modes. It is much simpler than Figure 5, and the amount of schemes is only $3^9 \times 2^2 = 78732$. The optimal scheme is easily determined using any computer and algorithm.

6. Conclusion

A project can be represented by using an activity network, and the activity floats and activity paths in the network are important to solve many project scheduling problems. For example,

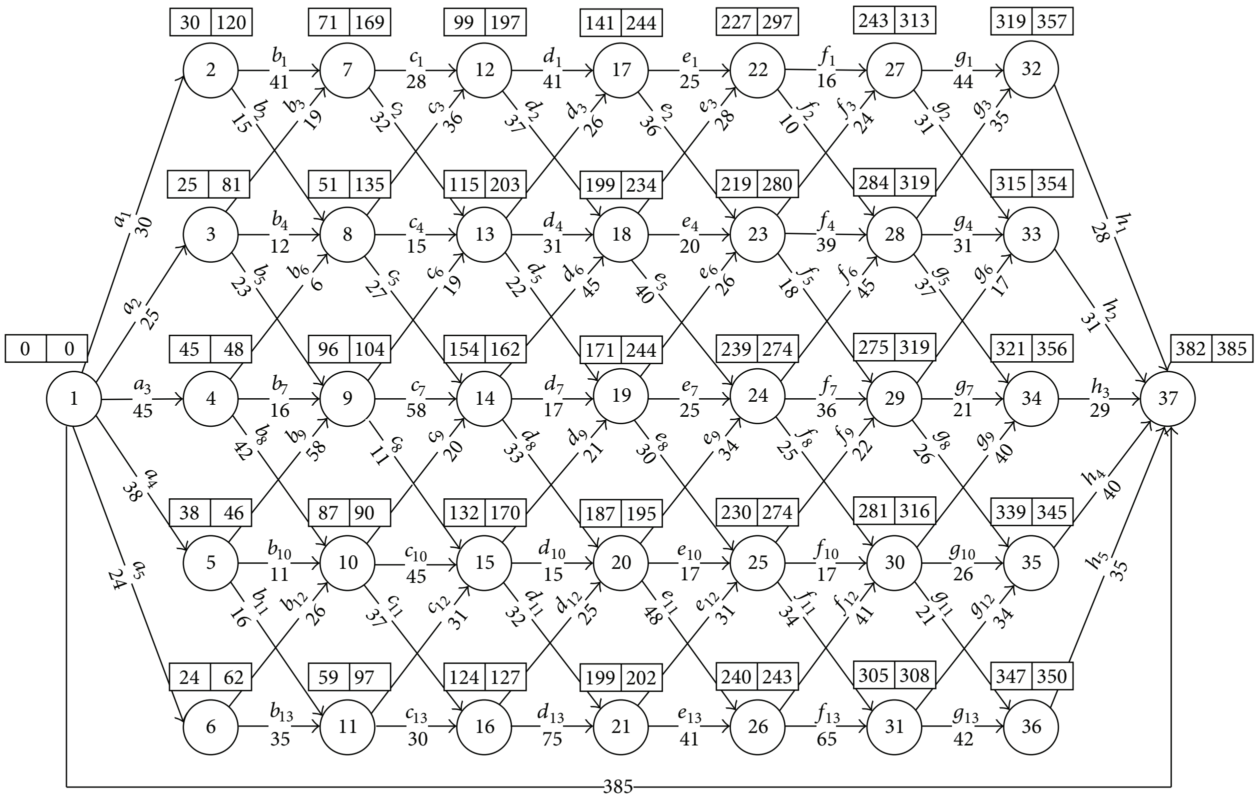
Figure 5: Time parameters for all activities with their shortest durations.

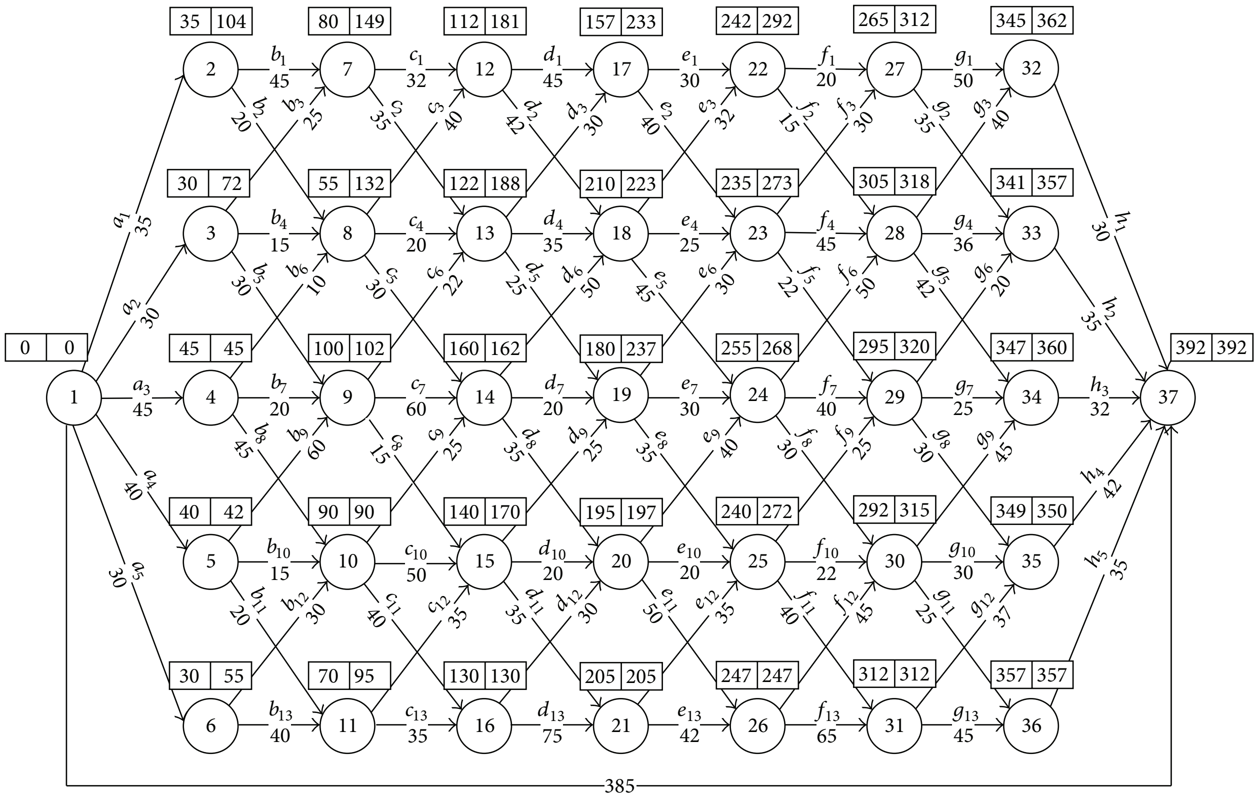
Figure 6: Time parameters for all activities with their remaining longest durations.

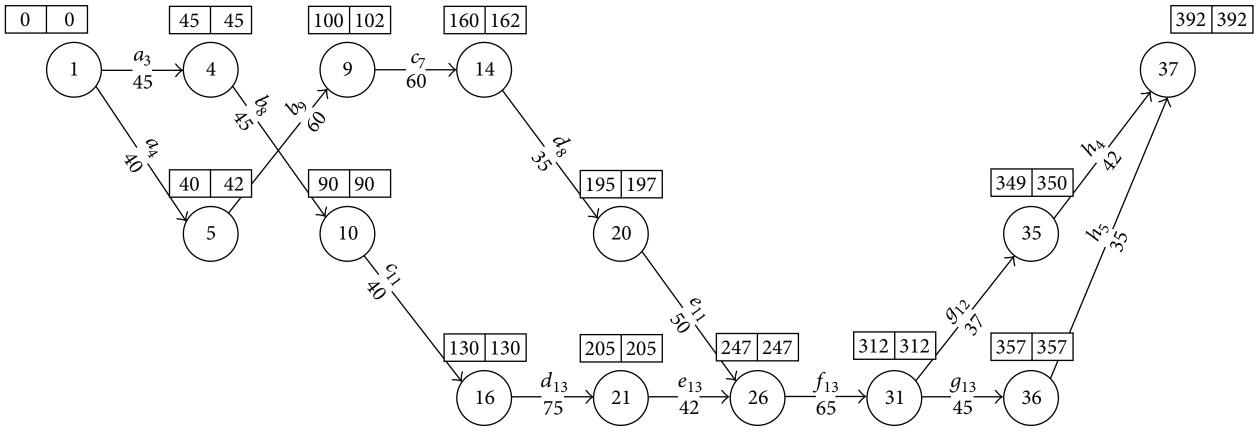
Figure 7: Network with nonredundant activities.

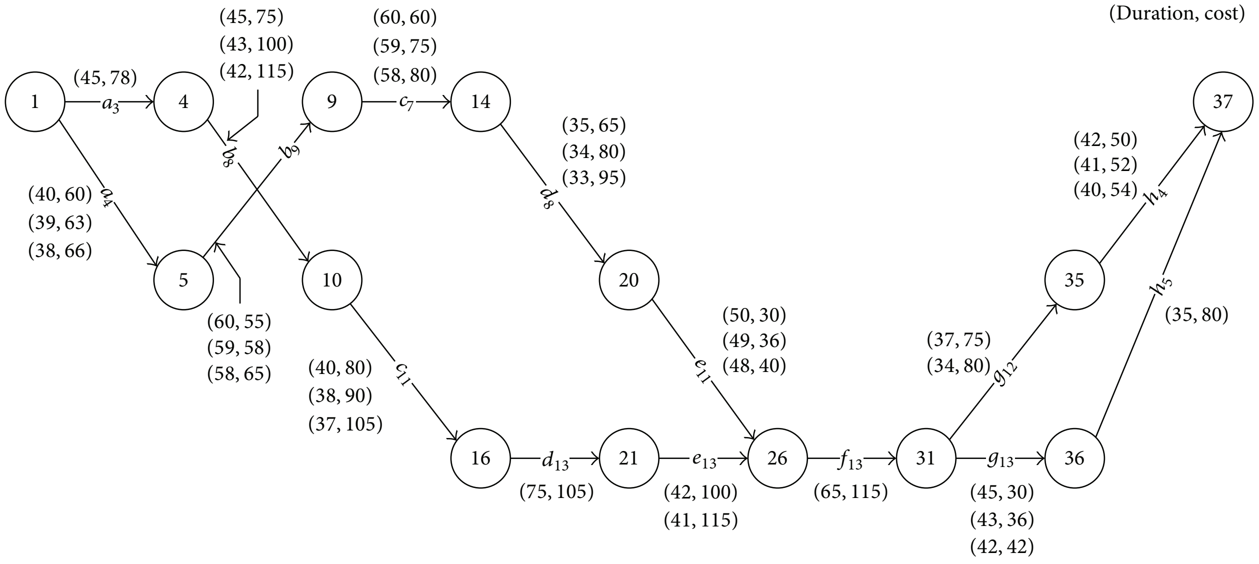
Figure 8: Network G^*.

for resource leveling, the range of adjusting each activity is determined by its float; and for time-cost tradeoff, a key step is to compress long paths to required lengths. In conventional thinking, activity floats and paths are separate, and all floats are easy to compute but many paths are difficult to find. However, we discovered that there are close relationships between activity floats and paths, and paths and their lengths can be represented by floats. For instance, the total float can be used to compute length of the longest path passing an activity; the free float and safety float can be used to find required paths between two nodes; and the interference float can be used to compute path lengths when changing the network's structure (adding or flipping arcs). Based on these, we established the float-path theory, which helps to simplify path problems.

We use the float-path theory to solve a classic project scheduling problem—the time-cost tradeoff problem, especially the nonlinear continuous and discrete (strong NP-hard) versions. Due to the difficulty, we first simplify the problem rather than designing algorithms to solve it directly. The simplification is to remove redundant activities and redundant durations (time-cost mode) of nonredundant activities, which need to compute enormous path lengths. We used the float-path theory to transform the simplification into computations of some activity floats and designed a polynomial algorithm. Complexity of the algorithm is $O(m)$ and m indicates the amount of activities. After the simplification, the old problem may be solved more effectively using current algorithms.

TABLE 2: Modes of each activity.

Activity code	Immediate successor	m_1		m_2		m_3	
		d	c	d	c	d	c
a_1	b_1, b_2	30	58	33	53	35	50
a_2	b_3, b_4, b_5	25	46	27	42	30	40
a_3	b_6, b_7, b_8	45	78	49	75	50	70
a_4	b_9, b_{10}, b_{11}	38	66	39	63	40	60
a_5	b_{12}, b_{13}	24	50	28	47	30	45
b_1	c_1, c_2	41	62	42	55	45	50
b_2	c_3, c_4, c_5	15	78	18	70	20	65
b_3	c_1, c_2	19	32	23	24	25	20
b_4	c_3, c_4, c_5	12	35	14	32	15	30
b_5	c_6, c_7, c_8	23	102	27	95	30	90
b_6	c_3, c_4, c_5	6	50	8	45	10	40
b_7	c_6, c_7, c_8	16	270	18	200	20	150
b_8	c_9, c_{10}, c_{11}	42	115	43	100	45	75
b_9	c_6, c_7, c_8	58	65	59	58	60	55
b_{10}	c_9, c_{10}, c_{11}	11	40	14	37	15	35
b_{11}	c_{12}, c_{13}	16	82	19	77	20	75
b_{12}	c_9, c_{10}, c_{11}	26	72	27	68	30	65
b_{13}	c_{12}, c_{13}	35	100	38	90	40	85
c_1	d_1, d_2	28	85	30	76	32	70
c_2	d_3, d_4, d_5	32	55	34	45	35	40
c_3	d_1, d_2	36	155	38	135	40	120
c_4	d_3, d_4, d_5	15	60	18	55	20	50
c_5	d_6, d_7, d_8	27	14	28	11	30	10
c_6	d_3, d_4, d_5	19	15	21	12	22	10
c_7	d_6, d_7, d_8	58	80	59	75	60	60
c_8	d_9, d_{10}, d_{11}	11	68	13	60	15	55
c_9	d_6, d_7, d_8	20	145	23	120	25	100
c_{10}	d_9, d_{10}, d_{11}	45	70	47	55	50	50
c_{11}	d_{12}, d_{13}	37	105	38	90	40	80
c_{12}	d_9, d_{10}, d_{11}	31	45	33	38	35	30
c_{13}	d_{12}, d_{13}	30	27	32	22	35	20
d_1	e_1, e_2	41	68	43	65	45	60
d_2	e_3, e_4, e_5	37	100	40	66	42	60
d_3	e_1, e_2	26	52	29	48	30	45
d_4	e_3, e_4, e_5	31	25	32	22	35	20
d_5	e_6, e_7, e_8	22	60	24	50	25	45
d_6	e_3, e_4, e_5	45	75	48	60	50	50
d_7	e_6, e_7, e_8	17	105	18	90	20	80
d_8	e_9, e_{10}, e_{11}	33	95	34	80	35	65
d_9	e_6, e_7, e_8	21	50	23	44	25	40
d_{10}	e_9, e_{10}, e_{11}	15	23	17	21	20	20
d_{11}	e_{12}, e_{13}	32	240	34	210	35	200
d_{12}	e_9, e_{10}, e_{11}	25	65	26	55	30	50
d_{13}	e_{12}, e_{13}	75	105	79	90	80	85
e_1	f_1, f_2	25	33	28	31	30	30

TABLE 2: Continued.

Activity code	Immediate successor	m_1		m_2		m_3	
		d	c	d	c	d	c
e_2	f_3, f_4, f_5	36	130	39	110	40	100
e_3	f_1, f_2	28	58	30	53	32	50
e_4	f_3, f_4, f_5	20	105	23	96	25	90
e_5	f_6, f_7, f_8	40	80	42	66	45	60
e_6	f_3, f_4, f_5	26	85	29	70	30	65
e_7	f_6, f_7, f_8	25	125	28	110	30	90
e_8	f_9, f_{10}, f_{11}	30	320	33	290	35	240
e_9	f_6, f_7, f_8	34	90	37	80	40	70
e_{10}	f_9, f_{10}, f_{11}	17	80	19	72	20	65
e_{11}	f_{12}, f_{13}	48	40	49	36	50	30
e_{12}	f_9, f_{10}, f_{11}	31	42	32	38	35	35
e_{13}	f_{12}, f_{13}	41	115	42	100	45	90
f_1	g_1, g_2	16	60	17	48	20	40
f_2	g_3, g_4, g_5	10	85	13	72	15	65
f_3	g_1, g_2	24	78	26	70	30	60
f_4	g_3, g_4, g_5	39	120	43	115	45	100
f_5	g_6, g_7, g_8	18	52	20	48	22	45
f_6	g_3, g_4, g_5	45	85	47	70	50	65
f_7	g_6, g_7, g_8	36	125	38	110	40	90
f_8	g_9, g_{10}, g_{11}	25	320	27	290	30	240
f_9	g_6, g_7, g_8	22	90	24	80	25	70
f_{10}	g_9, g_{10}, g_{11}	17	80	20	72	22	65
f_{11}	g_{12}, g_{13}	34	40	36	36	40	30
f_{12}	g_9, g_{10}, g_{11}	41	42	43	38	45	35
f_{13}	g_{12}, g_{13}	65	115	69	100	70	90
g_1	h_1	44	60	47	48	50	40
g_2	h_2	31	85	32	72	35	65
g_3	h_1	35	78	37	70	40	60
g_4	h_2	31	120	33	115	36	100
g_5	h_3	37	52	39	48	42	45
g_6	h_2	17	75	19	65	20	60
g_7	h_3	21	20	23	17	25	15
g_8	h_4	26	58	27	55	30	50
g_9	h_3	40	36	42	33	45	30
g_{10}	h_4	26	50	28	44	30	40
g_{11}	h_5	21	13	24	11	25	10
g_{12}	h_4	34	80	37	75	42	70
g_{13}	h_5	42	42	43	36	45	30
h_1	—	28	23	29	21	30	20
h_2	—	31	62	33	55	35	50
h_3	—	29	17	30	16	32	15
h_4	—	40	54	41	52	42	50
h_5	—	35	80	39	75	40	65

Appendices

A. Proof of Theorem 1

In a CPM network, the length, $L(\mu_{i\to j})$, of a path $\mu_{i\to j}$ from a node (i) to a node (i) is $\sum_{(e,f)\in\mu_{i\to j}} d_{ef}$.

(1) According to (5),

$$d_{ef} = \mathrm{ET}_f - \mathrm{ET}_e - \mathrm{FF}_{ef}. \tag{A.1}$$

Assume $\mu_{i\to i} = (i) \to (a) \to (b) \to \cdots \to (g) \to (h) \to (j)$; then

$$\begin{aligned} L(\mu_{i\to j}) &= \sum_{(e,f)\in\mu_{i\to j}} d_{ef} \\ &= \sum_{(e,f)\in\mu_{i\to j}} (\mathrm{ET}_f - \mathrm{ET}_e - \mathrm{FF}_{ef}) \\ &= (\mathrm{ET}_a - \mathrm{ET}_i - \mathrm{FF}_{ia}) + (\mathrm{ET}_b - \mathrm{ET}_a - \mathrm{FF}_{ab}) \\ &\quad + \cdots + (\mathrm{ET}_h - \mathrm{ET}_g - \mathrm{FF}_{gh}) \\ &\quad + (\mathrm{ET}_j - \mathrm{ET}_h - \mathrm{FF}_{hj}) \\ &= \mathrm{ET}_j - \mathrm{ET}_i - (\mathrm{FF}_{ia} + \mathrm{FF}_{ab} + \cdots + \mathrm{FF}_{gh} + \mathrm{FF}_{hj}) \\ &= \mathrm{ET}_j - \mathrm{ET}_i - \sum_{(e,f)\in\mu_{i\to j}} \mathrm{FF}_{ef}. \end{aligned} \tag{A.2}$$

(2) According to (6),

$$d_{ef} = \mathrm{LT}_f - \mathrm{LT}_e - \mathrm{SF}_{ef}. \tag{A.3}$$

Assume $\mu_{i\to j} = (i) \to (a) \to (b) \to \cdots \to (g) \to (h) \to (j)$; then

$$\begin{aligned} L(\mu_{i\to j}) &= \sum_{(e,f)\in\mu_{i\to j}} d_{ef} \\ &= \sum_{(e,f)\in\mu_{i\to j}} (\mathrm{LT}_f - \mathrm{LT}_e - \mathrm{SF}_{ef}) \\ &= (\mathrm{LT}_a - \mathrm{LT}_i - \mathrm{SF}_{ia}) + (\mathrm{LT}_b - \mathrm{LT}_a - \mathrm{SF}_{ab}) \\ &\quad + \cdots + (\mathrm{LT}_h - \mathrm{LT}_g - \mathrm{SF}_{gh}) \\ &\quad + (\mathrm{LT}_j - \mathrm{LT}_h - \mathrm{SF}_{hj}) \\ &= \mathrm{LT}_j - \mathrm{LT}_i - (\mathrm{SF}_{ia} + \mathrm{SF}_{ab} + \cdots + \mathrm{SF}_{gh} + \mathrm{SF}_{hj}) \\ &= \mathrm{LT}_j - \mathrm{LT}_i - \sum_{(e,f)\in\mu_{i\to j}} \mathrm{SF}_{ef}. \end{aligned} \tag{A.4}$$

(3) Assume an activity $(u,v) \in \mu_{i\to j}$, and $\mu_{i\to j} = (i) \to (a) \to (b) \to \cdots \to (u) \to (v) \to \cdots \to (g) \to (h) \to (j)$. According to (4),

$$d_{uv} = \mathrm{LT}_v - \mathrm{ET}_u - \mathrm{TF}_{iv}; \tag{A.5}$$

then

$$\begin{aligned} L(\mu_{i\to j}) &= \sum_{(e,f)\in\mu_{i\to j}} d_{ef} \\ &= \sum_{(e,f)\in\mu_{i\to u}} (\mathrm{ET}_f - \mathrm{ET}_e - \mathrm{FF}_{ef}) \\ &\quad + (\mathrm{LT}_v - \mathrm{ET}_u - \mathrm{TF}_{uv}) \\ &\quad + \sum_{(e,f)\in\mu_{v\to j}} (\mathrm{LT}_s - \mathrm{LT}_r - \mathrm{SF}_{rs}) \\ &= \left(\mathrm{ET}_u - \mathrm{ET}_i - \sum_{(e,f)\in\mu_{i\to u}} \mathrm{FF}_{ef}\right) \\ &\quad + (\mathrm{LT}_v - \mathrm{ET}_u - \mathrm{TF}_{uv}) \\ &\quad + \left(\mathrm{LT}_j - \mathrm{LT}_v - \sum_{(r,s)\in\mu_{v\to j}} \mathrm{SF}_{rs}\right) \\ &= \mathrm{LT}_j - \mathrm{ET}_i - \mathrm{TF}_{uv} \\ &\quad - \sum_{(e,f)\in\mu_{i\to u}} \mathrm{FF}_{ef} - \sum_{(r,s)\in\mu_{v\to j}} \mathrm{SF}_{rs}. \end{aligned} \tag{A.6}$$

(4) Assume a node $(k) \in \mu_{i\to j}$, and $\mu_{i\to j} = (i) \to (a) \to (b) \to \cdots \to (k) \to \cdots \to (g) \to (h) \to (j)$; then

$$\begin{aligned} L(\mu_{i\to j}) &= \sum_{(e,f)\in\mu_{i\to j}} d_{ef} \\ &= \sum_{(e,f)\in\mu_{i\to k}} (\mathrm{ET}_f - \mathrm{ET}_e - \mathrm{FF}_{ef}) \\ &\quad + \sum_{(e,f)\in\mu_{k\to j}} (\mathrm{LT}_s - \mathrm{LT}_r - \mathrm{SF}_{rs}) \\ &= \left(\mathrm{ET}_k - \mathrm{ET}_i - \sum_{(e,f)\in\mu_{i\to k}} \mathrm{FF}_{ef}\right) \\ &\quad + \left(\mathrm{LT}_j - \mathrm{LT}_k - \sum_{(r,s)\in\mu_{k\to j}} \mathrm{SF}_{rs}\right) \\ &= \mathrm{LT}_j - \mathrm{ET}_i - \mathrm{TF}_k \\ &\quad - \sum_{(e,f)\in\mu_{i\to k}} \mathrm{FF}_{ef} - \sum_{(r,s)\in\mu_{k\to j}} \mathrm{SF}_{rs}. \end{aligned} \tag{A.7}$$

Therefore, (8) is proved. This completes the proof.

B. Proof of Theorem 2

For any node (i) in a CPM network, according to (1), we can summarize that

$$\mathrm{ET}_i = \max_h \{\mathrm{ET}_h + d_{hi}\}$$

$$\mathrm{ET}_h = \max_g \{\mathrm{ET}_g + d_{gh}\}$$

$$\begin{aligned} &\vdots \\ \mathrm{ET}_a &= \max_1 \{\mathrm{ET}_1 + d_{1a}\} \\ \mathrm{ET}_1 &= 0 \end{aligned} \tag{B.1}$$

and further deduce by iteration that

$$\begin{aligned} \mathrm{ET}_i &= \max_h \{\mathrm{ET}_h + d_{hi}\} \\ &= \max_h \left\{\max_g \{\mathrm{ET}_g + d_{gh}\} + d_{hi}\right\} \\ &\vdots \\ &= \max_h \left\{\max_g \left\{\cdots \left\{\max_1 \{\mathrm{ET}_1 + d_{1a}\} + d_{ab}\right\} \right.\right. \\ &\qquad \left.\left. + \cdots + d_{gh}\right\} + d_{hi}\right\} \\ &= \max_{1 \to i} \left\{d_{1a} + d_{ab} + \cdots + d_{gh} + d_{hi}\right\} \\ &= L\left(\mu^{\nabla}_{1 \to i}\right). \end{aligned} \tag{B.2}$$

And According to (2), we can summarize that

$$\begin{aligned} \mathrm{LT}_i &= \min_j \left\{\mathrm{LT}_j - d_{ij}\right\} \\ \mathrm{LT}_j &= \min_k \left\{\mathrm{LT}_k - d_{jk}\right\} \\ &\vdots \\ \mathrm{LT}_y &= \min_n \left\{\mathrm{LT}_n - d_{yn}\right\} \\ \mathrm{LT}_n &= L\left(\mu^{\nabla}\right) \end{aligned} \tag{B.3}$$

and further deduce by iteration that

$$\begin{aligned} \mathrm{LT}_i &= \min_j \left\{\mathrm{LT}_j - d_{ij}\right\} \\ &= \min_j \left\{\min_k \left\{\mathrm{LT}_k - d_{ji}\right\} - d_{ij}\right\} \\ &\vdots \\ &= \min_j \left\{\min_k \left\{\cdots \left\{\min_n \left\{\mathrm{LT}_n - d_{yn}\right\} - d_{xy}\right\} \right.\right. \\ &\qquad \left.\left. - \cdots - d_{jk}\right\} - d_{ij}\right\} \\ &= \mathrm{LT}_n - \max_{i \to n} \left\{d_{ij} + d_{jk} + \cdots + d_{xy} + d_{yn}\right\} \\ &= L\left(\mu^{\nabla}\right) - L\left(\mu^{\nabla}_{i \to n}\right). \end{aligned} \tag{B.4}$$

Therefore, (9) and (10) are proved. This completes the proof.

C. Proof of Corollary 3

For any node (i), according to (1) and (8),

$$\begin{aligned} L\left(\mu^{\nabla}_{1 \to i}\right) &= \mathrm{ET}_i - \mathrm{ET}_1 - \sum_{(e,f) \in \mu_{i \to j}} \mathrm{FF}_{ef} \\ &= \mathrm{ET}_i - \sum_{(e,f) \in \mu_{i \to j}} \mathrm{FF}_{ef}. \end{aligned} \tag{C.1}$$

And according to (9), $L(\mu^{\nabla}_{1 \to i}) = \mathrm{ET}_i$; then

$$\mathrm{ET}_i - \sum_{(e,f) \in \mu_{i \to j}} \mathrm{FF}_{ef} = \mathrm{ET}_i; \tag{C.2}$$

that is,

$$\sum_{(e,f) \in \mu_{i \to j}} \mathrm{FF}_{ef} = 0. \tag{C.3}$$

According to (1) and (5),

$$\mathrm{FF}_{ef} \geq 0; \tag{C.4}$$

therefore

$$\mathrm{FF}_{ef} = 0, \quad (e, f) \in \mu_{1 \to i}. \tag{C.5}$$

Equation (11) is proved.

According to (3) and (8),

$$\begin{aligned} L\left(\mu^{\nabla}_{i \to n}\right) &= \mathrm{LT}_n - \mathrm{LT}_i - \sum_{(r,s) \in \mu_{i \to n}} \mathrm{SF}_{rs} \\ &= L\left(\mu^{\nabla}\right) - \mathrm{LT}_i - \sum_{(r,s) \in \mu_{i \to n}} \mathrm{SF}_{rs}. \end{aligned} \tag{C.6}$$

And according to (10), $L(\mu^{\nabla}_{i \to n}) = L(\mu^{\nabla}) - \mathrm{LT}_i$; then

$$L\left(\mu^{\nabla}\right) - \mathrm{LT}_i - \sum_{(r,s) \in \mu_{i \to n}} \mathrm{SF}_{rs} = L\left(\mu^{\nabla}\right) - \mathrm{LT}_i; \tag{C.7}$$

that is,

$$\sum_{(r,s) \in \mu_{i \to n}} \mathrm{SF}_{rs} = 0. \tag{C.8}$$

According to (2) and (6),

$$\mathrm{SF}_{rs} \geq 0; \tag{C.9}$$

therefore

$$\mathrm{SF}_{rs} = 0, \quad (r, s) \in \mu_{i \to n}. \tag{C.10}$$

Equation (12) is proved. This completes the proof.

D. Proof of Theorem 4

For any activity (i, j) in a CPM network, according to (4), $\mathrm{TF}_{ij} = \mathrm{LT}_j - \mathrm{ET}_i - d_{ij}$. Substitute (9) and (10) in the above equation; then

$$\begin{aligned}\mathrm{TF}_{ij} &= \mathrm{LT}_j - \mathrm{ET}_i - d_{ij}\\ &= L\left(\mu^{\nabla}\right) - L\left(\mu^{\nabla}_{j\to n}\right) - L\left(\mu^{\nabla}_{1\to i}\right) - d_{ij}\\ &= L\left(\mu^{\nabla}\right) - \left(L\left(\mu^{\nabla}_{j\to n}\right) + d_{ij} + L\left(\mu^{\nabla}_{1\to i}\right)\right)\\ &= L\left(\mu^{\nabla}\right) - L\left(\mu^{\nabla}_{ij}\right).\end{aligned} \tag{D.1}$$

Therefore, (13) is proved. This completes the proof.

E. Proof of Theorem 5

For any activity (i, j) in a CPM network, according to (5), $\mathrm{FF}_{ij} = \mathrm{ET}_j - \mathrm{ET}_i - d_{ij}$. Substitute (9) in the above equation; then

$$\begin{aligned}\mathrm{FF}_{ij} &= \mathrm{ET}_j - \mathrm{ET}_i - d_{ij}\\ &= L\left(\mu^{\nabla}_{1\to j}\right) - L\left(\mu^{\nabla}_{1\to i}\right) - d_{ij}\\ &= \left(L\left(\mu^{\nabla}_{1\to j}\right) + L\left(\mu^{\nabla}_{j\to n}\right)\right)\\ &\quad - \left(L\left(\mu^{\nabla}_{1\to i}\right) + d_{ij} + L\left(\mu^{\nabla}_{j\to n}\right)\right)\\ &= L\left(\mu^{\nabla}_{j}\right) - L\left(\mu^{\nabla}_{ij}\right).\end{aligned} \tag{E.1}$$

Therefore, (14) is proved. This completes the proof.

F. Proof of Theorem 7

For any activity (i, j) in a CPM network, according to (7), $\mathrm{IF}_{ij} = \mathrm{ET}_j - \mathrm{LT}_i - d_{ij}$. Substitute (9) and (10) in the above equation; then

$$\begin{aligned}\mathrm{IF}_{ij} &= \mathrm{ET}_j - \mathrm{LT}_i - d_{ij}\\ &= L\left(\mu^{\nabla}_{1\to j}\right) - \left(L\left(\mu^{\nabla}\right) - L\left(\mu^{\nabla}_{i\to n}\right)\right) - d_{ij}\\ &= L\left(\mu^{\nabla}_{1\to j}\right) + L\left(\mu^{\nabla}_{i\to n}\right) - L\left(\mu^{\nabla}\right) - d_{ij}.\end{aligned} \tag{F.1}$$

If we add an arc (j, i) with length $d_{ji} = d_{ij}$, then the longest path passing the arc (j, i) once is $\mu^{\nabla}_{ji} = \mu^{\nabla}_{1\to j} + (j, i) + \mu^{\nabla}_{i\to n}$, and its length is $L(\mu^{\nabla}_{ji}) = L(\mu^{\nabla}_{1\to j}) + d_{ij} + L(\mu^{\nabla}_{i\to n})$. Hence

$$\begin{aligned}\mathrm{IF}_{ij} &= L\left(\mu^{\nabla}_{1\to j}\right) + L\left(\mu^{\nabla}_{i\to n}\right) - L\left(\mu^{\nabla}\right) - d_{ij}\\ &= L\left(\mu^{\nabla}_{1\to j}\right) + d_{ij} + L\left(\mu^{\nabla}_{i\to n}\right) - L\left(\mu^{\nabla}\right) - 2d_{ij}\\ &= L\left(\mu^{\nabla}_{ji}\right) - L\left(\mu^{\nabla}\right) - 2d_{ij}\end{aligned} \tag{F.2}$$

and μ^{∇} indicates the critical path in the old network before adding the arc (j, i).

In addition, if $\mathrm{FF}_{ij} > 0$ and $\mathrm{SF}_{ij} > 0$, then according to Corollary 3, $(i, j) \notin \mu^{\nabla}_{1\to j}$ and $(i, j) \notin \mu^{\nabla}_{i\to n}$. Hence $\mu^{\nabla}_{1\to j}$ and $\mu^{\nabla}_{i\to n}$ will be existent when flipping the arc (i, j). After flipping the arc (i, j), the longest path μ^{∇}_{ji} passing the arc (i, j) will appear; that is, $\mu^{\nabla}_{ji} = \mu^{\nabla}_{1\to j} + (j, i) + \mu^{\nabla}_{i\to n}$, and its length is $L(\mu^{\nabla}_{ji}) = L(\mu^{\nabla}_{1\to j}) + d_{ij} + L(\mu^{\nabla}_{i\to n})$. Therefore,

$$\begin{aligned}\mathrm{IF}_{ij} &= L\left(\mu^{\nabla}_{1\to j}\right) + L\left(\mu^{\nabla}_{i\to n}\right) - L\left(\mu^{\nabla}\right) - d_{ij}\\ &= L\left(\mu^{\nabla}_{1\to j}\right) + d_{ij} + L\left(\mu^{\nabla}_{i\to n}\right) - L\left(\mu^{\nabla}\right) - 2d_{ij}\\ &= L\left(\mu^{\nabla}_{ji}\right) - L\left(\mu^{\nabla}\right) - 2d_{ij}\end{aligned} \tag{F.3}$$

and μ^{∇} indicates the critical path in the old network before flipping the arc (j, i). Equation (16) is proved. This completes the proof.

G. Proof of Theorem 13

Since each activity (i, j) has a duration $d_{ij} = u_{ij}$, all paths in the network get to their longest lengths. For the activity (i, j), according to Theorem 1,

$$\begin{aligned}\mathrm{TF}_{ij} - (T - \lambda) &= \left(L\left(\mu^{\nabla}\right) - L\left(\mu^{\nabla}_{ij}\right)\right) - (T - \lambda)\\ &= \left(L\left(\mu^{\nabla}\right) - L\left(\mu^{\nabla}_{ij}\right)\right) - \left(L\left(\mu^{\nabla}\right) - \lambda\right)\\ &= \lambda - L\left(\mu^{\nabla}_{ij}\right).\end{aligned} \tag{G.1}$$

If $\mathrm{TF}_{ij} \geq T - \lambda$, namely, $\mathrm{TF}_{ij} - (T - \lambda) \geq 0$, then $\lambda - L(\mu^{\nabla}_{ij}) \geq 0$. Therefore, the longest path μ^{∇}_{ij} passing the activity has a length $L(\mu^{\nabla}_{ij}) \leq \lambda$. According to Proposition 9, the activity (i, j) is redundant. This completes the proof.

H. Proof of Theorem 14

Since each activity (i, j) chooses duration $d_{ij} = l_{ij}$, all paths in the network get their shortest lengths. The project duration T is equal to the length of the critical path μ^{∇}, and the assistant activity $(1, n)$ with a duration $d_{1n} = \lambda$ is added; therefore $T = L(\mu^{\nabla}) = \lambda$. According to Theorem 1, $L(\mu^{\nabla}_{ij}) = L(\mu^{\nabla}) - \mathrm{TF}_{ij}$. If the duration d_{ij} of the activity (i, j) is prolonged by TF_{ij}, namely,

$$\begin{aligned}d_{ij} &= l_{ij} + \mathrm{TF}_{ij}\\ &= l_{ij} + \left(\mathrm{LT}_j - \mathrm{ET}_i - l_{ij}\right)\\ &= \mathrm{LT}_j - \mathrm{ET}_i,\end{aligned} \tag{H.1}$$

then μ^{∇}_{ij} also will be correspondingly prolonged by TF_{ij} and then equal to the same length as μ^{∇} and will be a new critical path. In this case the project duration remains λ.

But if d_{ij} is still prolonged, this will prolong the critical path, and the project duration will be longer than λ. Since all other activities choose their shortest durations, it is inevitable that $L(\mu^{\nabla}) = L(\mu_{ij}^{\nabla}) > \lambda$ once the activity (i, j) meets $d_{ij} > \mathrm{LT}_j - \mathrm{ET}_i$, and the project duration is longer than λ. According to Proposition 10, the duration $d_{ij} > \mathrm{LT}_j - \mathrm{ET}_i$ is a redundant-long duration of the activity (i, j). Obviously, the precondition of an existing redundant-long duration is $\mathrm{LT}_j - \mathrm{ET}_i < u_{ij}$. This completes the proof.

I. Proof of Theorem 15

Since each activity (i, j) chooses duration $d_{ij} = u_{ij}$, all paths in the network get their longest lengths. For an activity (i, j), according to Theorem 1,

$$\begin{aligned}\mathrm{TF}_{ij} - (T - \lambda) &= \left(L\left(\mu^{\nabla}\right) - L\left(\mu_{ij}^{\nabla}\right)\right) - (T - \lambda)\\ &= \left(L\left(\mu^{\nabla}\right) - L\left(\mu_{ij}^{\nabla}\right)\right) - \left(L\left(\mu^{\nabla}\right) - \lambda\right)\\ &= \lambda - L\left(\mu_{ij}^{\nabla}\right).\end{aligned} \tag{I.1}$$

If $\mathrm{TF}_{ij} < T - \lambda$, namely, $\mathrm{TF}_{ij} - (T - \lambda) < 0$, then $\lambda - L(\mu_{ij}^{\nabla}) < 0$. Therefore, the longest path μ_{ij}^{∇} passing the activity has a length $L(\mu_{ij}^{\nabla}) > \lambda$.

According to Proposition 11, if the duration of the activity (i, j) is shortened to a value that makes the longest path $\mu_{ij}'^{\nabla}$ shorter than λ, then it is a short duration of the activity. Therefore, if one wants to make $L(\mu_{ij}'^{\nabla}) \le \lambda$, the duration of the activity (i, j) needs to be shortened by at least $L(\mu_{ij}^{\nabla}) - \lambda$, which means that the duration $d_{ij} \le u_{ij} - (L(\mu_{ij}^{\nabla}) - \lambda)$ is a redundant-short duration of the activity (i, j). Obviously, the precondition of an existing short duration is $u_{ij} - (L(\mu_{ij}^{\nabla}) - \lambda) \ge l_{ij}$. According to Theorem 1, $L(\mu_{ij}^{\nabla}) = L(\mu^{\nabla}) - \mathrm{TF}_{ij} = T - \mathrm{TF}_{ij}$; thence

$$\begin{aligned}u_{ij} - \left(L\left(\mu_{ij}^{\nabla}\right) - \lambda\right) &= u_{ij} - \left(T - \mathrm{TF}_{ij} - \lambda\right)\\ &= u_{ij} - T + \left(\mathrm{LT}_j - \mathrm{ET}_i - u_{ij}\right) + \lambda\\ &= \mathrm{LT}_j - \mathrm{ET}_i - T + \lambda\end{aligned} \tag{I.2}$$

and additionally

$$u_{ij} - \left(L\left(\mu_{ij}^{\nabla}\right) - \lambda\right) \ge l_{ij} \Longrightarrow l_{ij} \le \mathrm{LT}_j - \mathrm{ET}_i - T + \lambda. \tag{I.3}$$

Thus, if $\mathrm{TF}_{ij} < T - \lambda$ and $l_{ij} \le \mathrm{LT}_j - \mathrm{ET}_i - T + \lambda$, the duration $d_{ij} \le \mathrm{LT}_j - \mathrm{ET}_i - T + \lambda$ is a short duration of the activity (i, j). According to Proposition 12, except for the cheapest duration, all the other short durations of the activity (i, j) are redundant-short durations. This completes the proof.

J. Proof of Algorithm in Section 4.4

(1) Remove Redundant-Long Duration of Each Activity. Using Step 1, the assistant activity $(1, n)$ with $d_{1n} = u_{1n} = l_{1n} = \lambda$ is added to ensure project duration $T = \mathrm{ET}_n \ge \lambda$ in the process of simplification. Using Step 2, the time parameters of each node and the project duration in the CPM network are computed when all activities choose their shortest durations. According to Proposition 10 and Theorem 15, for any activity (i, j), the duration $d_{ij} > \mathrm{LT}_j - \mathrm{ET}_i$ is its redundant-long duration. Therefore, Steps 1~3 can be used to remove redundant-long duration of each activity.

(2) Remove Redundant Activity. The time parameters of each node and the project duration are computed by using Step 4 when each activity (i, j) chooses its current longest duration $u_{ij}' = \min\{\mathrm{LT}_j - \mathrm{ET}_i, u_{ij}\}$. According to Proposition 10 and Theorem 13, if an activity (i, j) meets $\mathrm{TF}_{ij} \ge \mathrm{ET}_n - \lambda$, then it is redundant activity and can be removed. Therefore, Steps 4~5 can be used to identify and remove all redundant activities.

(3) Remove Redundant-Short Activity of Each Remaining Activity. According to Proposition 12 and Theorem 15, for a remaining activity (i, j), durations meeting $d_{ij} \le \mathrm{LT}_j - \mathrm{ET}_i - \mathrm{ET}_n + \lambda$ are its short durations, and except for the longest one, the others are redundant-short durations and can be removed. Therefore, Step 6 can be used to remove the redundant-short duration of the remaining activities.

(4) Consider Whether the Simplification Needs to Be Repeated. After removing the redundant-short duration, we need to consider whether new redundant duration appears for each remaining activity or whether it is the appearance of new redundant activity.

According to Step 6, the shortest duration l_{ij}' of each remaining activity (i, j) meets $l_{ij}' \le a_{ij}$, $a_{ij} = \max\{\mathrm{LT}_j - \mathrm{ET}_i - \mathrm{ET}_n + \lambda, l_{ij}\}$. We let the algorithm run again for the second round simplification.

(1) First, Consider the Redundant-Long Duration. According to Proposition 10, the redundant-long duration of an activity is determined by the shortest durations of the other activities on the longest path passing it. Thence, a new redundant-long duration of the activity may appear only when the shortest durations of these activities become longer.

Assume $l_{ij}' = a_{ij}$, and if $\mathrm{LT}_j - \mathrm{ET}_i - \mathrm{ET}_n + \lambda \le l_{ij}$, then $a_{ij} = l_{ij}$. It means that the shortest duration of the activity (i, j) does not change and cannot lead the other activities having a new redundant-long duration. But if $\mathrm{LT}_j - \mathrm{ET}_i - \mathrm{ET}_n + \lambda > l_{ij}$, then $a_{ij} = \mathrm{LT}_j - \mathrm{ET}_i - \mathrm{ET}_n + \lambda > l_{ij}$. This means that the shortest duration of the activity (i, j) becomes longer and may lead the other activities on the path passing it to have new redundant-long durations. But according to Proposition 11, now a_{ij} is the cheapest short duration of the activity (i, j). When the activity chooses the duration, the longest path passing it is still no longer than λ, even if all other activities choose their current longest durations. In other words, although the shortest duration of activity (i, j) becomes longer, any path passing it will be no longer than λ when the activity in all cases chooses the current shortest duration. According to Proposition 10, all activities on these paths will not have new redundant-long durations. Similarly, it is the same for other remaining activities. Therefore, new redundant-long duration will not appear in the second round simplification.

Similarly, when assuming $l_{ij}' < a_{ij}$, new redundant-long duration of any activity will also not appear.

(2) Then Consider Redundant Activity and Redundant-Short Duration. Each activity will not have a new redundant-long duration such that the longest duration of each remaining activity (i, j) is still u'_{ij} after repeating Steps 2~3. Therefore, after repeating Step 4, the values ET_i, LT_i, TF_{ij}, and $\mathrm{ET}_n - \lambda$ are the same as the corresponding ones in the first round simplification. Each remaining activity (i, j) is nonredundant in the first round simplification; thus $\mathrm{TF}_{ij} < \mathrm{ET}_n - \lambda$ is still met in the second round simplification. All remaining activities are still nonredundant activities.

Similarly, assume $l'_{ij} = a_{ij}$, for a remaining activity (i, j); its shortest and longest durations are $a_{ij} = \max\{\mathrm{LT}_j - \mathrm{ET}_i - \mathrm{ET}_n + \lambda, l_{ij}\}$ and $u'_{ij} = \min\{\mathrm{LT}_j - \mathrm{ET}_i, u_{ij}\}$, and its total floats TF_{ij} and $\mathrm{ET}_n - \lambda$ are unchanged. According to Step 6, $\mathrm{LT}_j - \mathrm{ET}_i - \mathrm{ET}_n + \lambda$ is unchanged and the extreme duration a'_{ij} of the activity is

$$\begin{aligned} a'_{ij} &= \max\left\{\mathrm{LT}_j - \mathrm{ET}_i - \mathrm{ET}_n + \lambda, a_{ij}\right\} \\ &= \max\left\{\mathrm{LT}_j - \mathrm{ET}_i - \mathrm{ET}_n + \lambda, \right. \\ &\qquad \left. \max\left\{\mathrm{LT}_j - \mathrm{ET}_i - \mathrm{ET}_n + \lambda, l_{ij}\right\}\right\} \\ &= \max\left\{\mathrm{LT}_j - \mathrm{ET}_i - \mathrm{ET}_n + \lambda, l_{ij}\right\} \\ &= a_{ij}. \end{aligned} \tag{J.1}$$

Thence, the extreme duration of each activity is unchanged and new redundant-short duration will not appear.

Similarly, when assuming $l'_{ij} < a_{ij}$, a new redundant-short duration of any activity will also not appear.

According to above analysis, a new redundant objective does not appear after the second round simplification. The network cannot be simplified further using the algorithm. Similarly, the result will be the same after more simplification rounds; therefore the algorithm needs to be run only once to remove all redundant objectives. This completes the proof.

K. Complexity of Algorithm in Section 4.4

Suppose there are m activities in the network. In Steps 2 and 4, time parameters ET_i and LT_i of each node need to be computed using the CPM algorithm and the complexity is $O(m)$. In Step 3, the extreme value of the redundant-long duration of each activity needs to be computed in order to remove redundant-long duration; thence the complexity is $O(m)$. In Step 5, the total float of each activity needs to be computed and compared with $\mathrm{ET}_n - \lambda$; thus the complexity is $O(m)$. In Step 6, the extreme duration of each remaining activity needs to be computed to determine redundant-short duration; thus the complexity is $O(m)$. Since the algorithm only needs to run once, its complexity is $O(m)$.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The paper is supported by the Natural Science Foundation of China (no. 71171079) and the Soft Science Research Base of Water Security and Sustainable Development of Jiangxi Province in China.

References

[1] J. E. Kelley and M. R. Walker, "Critical-path planning and scheduling," in *Proceedings of the Eastern Joint IRE-ACM Computer Conference*, pp. 160–173, December 1959.

[2] T. Warren, "Four oat measures for critical path scheduling," *Journal of Industrial Engineering*, vol. 10, pp. 19–23, 1969.

[3] A. Battersby, *Network Analysis for Planning and Scheduling*, John Wiley & Sons, New York, NY, USA, 1970.

[4] S. E. Elmaghraby and J. Kamburowski, "On project representation and activity oats," *Arabian Journal of Science and Engineering*, vol. 15, pp. 625–637, 1990.

[5] S. E. Elmaghraby, "Activity nets: a guided tour through some recent developments," *European Journal of Operational Research*, vol. 82, no. 3, pp. 383–408, 1995.

[6] D. R. Fulkerson, "A network flow computation for project cost curves," *Management Science*, vol. 7, no. 2, pp. 167–178, 1961.

[7] J. Kelley, "Critical-path planning and scheduling: mathematical basis," *Operations Research*, vol. 9, no. 3, pp. 296–320, 1961.

[8] D. Panagiotakoplos, "A CPM time-cost computational algorithm for arbitrary activity cost function," *INFOR*, vol. 15, no. 2, pp. 183–195, 1977.

[9] J. J. Moder, C. R. Phillips, and E. W. Davis, *Project Management with CPM, PERT and Precedence Diagramming*, Van Nostrand Reinhold, New York, NY, USA, 1983.

[10] J. E. Falk and J. L. Horowitz, "Critical path problem with concave cost-time curve," *Management Science*, vol. 19, no. 4, pp. 446–455, 1972.

[11] L. R. Lamberson and R. R. Hocking, "Optimum time compression in project scheduling," *Management Science*, vol. 16, no. 10, pp. 597–606, 1970.

[12] K. C. Kapur, "An algorithm for the project cost/duration analysis problem with quadratic and convex cost functions," *IIE Transactions*, vol. 5, no. 4, pp. 314–322, 1973.

[13] N. Siemens and C. Gooding, "Reducing project duration at minimum cost: a time-cost tradeoff algorithm," *Omega*, vol. 3, no. 5, pp. 569–581, 1975.

[14] S. E. Elmaghraby and A. Salem, "Optimal project compression under convex cost functions I: quadratic cost with continuous derivative," OR Technical Report 157, North Carolina State University, Raleigh, NC, USA, 1980.

[15] S. E. Elmaghraby and A. Salem, "Optimal project compression under convex cost functions II," OR Technical Report 158, North Carolina State University, Raleigh, NC, USA, 1980.

[16] 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.

[17] K. Choi and Y. H. Kwak, "Decision support model for incentives/disincentives time-cost tradeoff," *Automation in Construction*, vol. 21, no. 1, pp. 219–228, 2012.

[18] R. F. Deckro, J. E. Hebert, W. A. Verdini, P. H. Grimsrud, and S. Venkateshwar, "Nonlinear time/cost tradeoff models in project management," *Computers and Industrial Engineering*, vol. 28, no. 2, pp. 219–229, 1995.

[19] A. Azaron and R. Tavakkoli-Moghaddam, "Multi-objective time-cost trade-off in dynamic PERT networks using an interactive approach," *European Journal of Operational Research*, vol. 180, no. 3, pp. 1186–1200, 2007.

[20] N. Halman, C.-L. Li, and D. Simchi-Levi, "Fully polynomial-time approximation schemes for time-cost tradeoff problems in series-parallel project networks," *Operations Research Letters*, vol. 37, no. 4, pp. 239–244, 2009.

[21] D. S. Yamashita and R. Morabito, "A note on time/cost trade-off curve generation for project scheduling with multi-mode resource availability costs," *International Journal of Operational Research*, vol. 5, no. 4, pp. 429–444, 2009.

[22] B. K. Pathak and S. Srivastava, "Integrated ANN-HMH approach for nonlinear time-cost tradeoff problem," *International Journal of Computational Intelligence Systems*, vol. 7, no. 3, pp. 456–471, 2014.

[23] M. Li and G. D. Wu, "Robust optimization for time-cost tradeoff problem in construction projects," *Abstract and Applied Analysis*, vol. 2014, Article ID 926913, 7 pages, 2014.

[24] R. T. Harvey and J. H. Patterson, "An implicit enumeration algorithm for the time/cost tradeoff problem in project network analysis," *Foundations of Control Engineering*, vol. 4, no. 2, pp. 107–117, 1979.

[25] P. De, J. E. 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.

[26] M. Skutella, "Approximation algorithms for the discrete time-cost tradeoff problem," *Mathematics of Operations Research*, vol. 23, no. 4, pp. 909–929, 1998.

[27] 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.

[28] 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.

[29] 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.

[30] H. Mokhtari, R. B. 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.

[31] E. N. Afruzi, E. Roghanian, A. A. Najafi, and M. Mazinani, "A multi-mode resource-constrained discrete time-cost tradeoff problem solving using an adjusted fuzzy dominance genetic algorithm," *Scientia Iranica*, vol. 20, no. 3, pp. 931–944, 2013.

[32] S. Parveen and S. K. Saha, "A GA-based fuzzy optimal model for time-cost trade-off in a project with resources consideration," *International Journal of Industrial and Systems Engineering*, vol. 15, no. 2, pp. 153–170, 2013.

[33] N. Glišović, "Comparison of a fuzzy genetic and simulated annealing algorithm approach for project time-cost tradeo," *Journal of Applied Mathematics*, vol. 2014, Article ID 817921, 12 pages, 2014.

[34] M.-Y. Cheng and D.-H. Tran, "Two-phase differential evolution for the multiobjective optimization of time-cost tradeoffs in resource-constrained construction projects," *IEEE Transactions on Engineering Management*, vol. 61, no. 3, pp. 450–461, 2014.

[35] J. G. Szmerekovsky 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.

[36] P. De, E. J. 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.

[37] E. L. Demeulemeester and W. S. Herroelen, *Project Scheduling: A Research Handbook*, Kluwer Academic Publishers, London, UK, 2002.

[38] S. S. Erenguc, T. Ahn, and D. G. Conway, "The resource constrained project scheduling problem with multiple crashable modes: an exact solution method," *Naval Research Logistics*, vol. 48, no. 2, pp. 107–127, 2001.

[39] Ö. Hazır, M. Haouari, and E. Erel, "Robust scheduling and robustness measures for the discrete time/cost trade-off problem," *European Journal of Operational Research*, vol. 207, no. 2, pp. 633–643, 2010.