Available online at www.sciencedirect.com
ScienceDirect
International Journal of Project Management 27 (2009) 600–609
International Journal of Project Management
www.elsevier.com/locate/ijproman

A multi-mode resource-constrained discrete time–cost tradeoff problem and its genetic algorithm based solution

Peng Wuliang [a,*], Wang Chengen [b]

[a] School of Economics and Management, Shenyang Ligong University, Shenyang 110168, China
[b] School of Information Sciences and Engineering, Northeastern University, Shenyang 110004, China

Received 5 March 2008; received in revised form 14 October 2008; accepted 23 October 2008

Abstract

The discrete time–cost tradeoff problem (DTCTP) is an important subject in the project scheduling theory and applications. Due to the fact that the resources used in projects in modern enterprises mainly belong to renewable resources, e.g. manpower resources, the general DTCTP is extended to a new multi-mode resource-constrained DTCTP model (MRC-DTCTP). The multi-mode resource-constrained project scheduling problem (MRCPSP) was referred and the renewable resource constraints were added to the general DTCTP. By predefining the resource price, the renewable resources are related to the project costs, including direct cost and indirect cost. Every activity can be executed in the crashing way in which the project direct costs are used to shorten the activity duration. According to the characteristics of the MRC-DTCTP, an improved genetic algorithm for solving it was developed and its effectiveness was verified by compared with the exact algorithm. Finally an entire time–cost tradeoff curve for a project network was drawn through computing project deadline problem, and the advantages of the MRC-DTCTP were investigated based on computation results.
© 2008 Elsevier Ltd and IPMA. All rights reserved.

Keywords: Project management; Discrete time–cost tradeoff; Planning and scheduling; Genetic algorithm

1. Introduction

An increasing number of products and services are developed by project organizations. The reasons are: firstly, shorter product life cycles, greater variety of products, and increased customer responsiveness reduce the production volumes of identical products. Therefore, project management plays an extremely important role in modern enterprise management. There have been dozens of project optimization scheduling models in project management theory, commonly referred to as resource-constrained project scheduling problems (RCPSPs), which have been recently reviewed by Brucker et al. [1], Demeulemeester et al. [2] and Kolisch and Padman [3]. Among RCPSPs, the discrete time/cost tradeoff problem (DTCTP) is well-known problem where the duration of each activity is a discrete, no-increasing function of the amount of a single nonrenewable resource committed to it. The DTCTP, firstly introduced by Hindelang and Muth [4] have been received remarkable attention for several years. It is a NP hard problem [5] and difficult to be solved [6]. The earliest literatures on DTCTP have paid more attention to its solution algorithms, which are classified into exact algorithms and heuristic algorithms. Exact algorithms are based dynamic programming e.g. [4], enumeration algorithm [7], or branch and bound algorithms [8–10]. None of the exact algorithms are able to solve large and hard instances measured in terms of, say, the number of activities, the structure of the project networks as well as the number of modes per activity. It follows that the exact algorithms which are also formally efficient are all but futile in practical application and one should instead search for effective heuristic algorithms to solve general DTCTP. The work by Akkan [11] is an example of a heuristic based on Lagrangian relaxation applied to an activity-on-arc

* Corresponding author.
E-mail address: peng-wuliang@163.com (P. Wuliang).

0263-7863/$34.00 © 2008 Elsevier Ltd and IPMA. All rights reserved.
doi:10.1016/j.ijproman.2008.10.009

(AoA) networks. Liu et al. [12] presented a meta-heuristic algorithm for solving general DTCTP, where the genetic algorithm is used to search a precedence-feasible activity list to generate a near-optimal solution with serial scheduling scheme. As research efforts progressed and practical demands grew, researchers also begin to improve the mathematic model of DTCTP according to the various demands of production practice. Elmaghraby and Kamburowski [13] considered the rewards and punishments on the DTCTP objective function. The crashing duration of activities was introduced to DTCTP in [14], where the duration/cost of an activity is determined by the mode selection and the duration reduction (crashing) applied within the selected mode. With present value as the objective function, the DTCTP with cash discount was studied in [15]. In [16], the general DTCTP model was constrained by time-switch, which imposes specified starting times on the project activities and forces them to be inactive during specified time periods.

Although the DTCTP has been studied for nearly 30 years, there are still some drawbacks to meeting the demands of project management. Most of researches on DTCTP focus on the nonrenewable resources while the renewable resources have not been attached more importance. In modern enterprises, especially high-tech enterprises, a large number of projects are set up to achieve the product innovation, and the resources used in projects mostly are manpower resources, which belong to renewable resources. Therefore, the project costs mainly consist of indirect costs related to the occupation of renewable resources.

In this paper, we presented a multi-mode resource-constrained discrete time/cost tradeoff model (MRC-DTCTP) that combined the general DTCTP and MRCPSP based on recent researches. Several important characteristics of this model are described as follows: (1) the renewable resources are focused. The renewable resources used in a project are limited and the resources used in activities are constrained; (2) the project cost consists of both direct costs and indirect costs, and they are calculated according to the different usages of renewable resources; (3) each activity is executed not only in a selected mode, but also in a selected way, which determines if the activity is executed in the crashing way to get a shortened duration by more direct costs. An improved genetic algorithm was developed to solve MRC-DTCTP. Through compared with the exact algorithm, the effectiveness of the algorithm was verified. A complete time–cost tradeoff curve for a project instance was drawn by computing the project deadline problem, and the advantages of the MRC-DTCTP were discussed.

2. Description of MRC-DTCTP

We consider a project which consists of n activities (jobs, tasks). Due to technological requirements precedence relations between some of the activities enforce that an activity $j, j = 2,\ldots,n$, may not be started before all its predecessors P_j, are finished. The structure of the project is depicted by an activity-on-node (AON) network $G = (V, E)$ where the nodes and arcs represent the activities V and precedence relations E, respectively. Activity 1 is the only start activity and activity n is the only finish activity. Inheriting the model of MRCPSP, each activity j may be executed in one of M_j modes. The activities may not be preempted and a mode once selected may not change, i.e., activity j once started in mode m has to be completed in the mode without interruption. There are K renewable resource types where the resource type $k \in K$ has the top bound R_k, and all the resources are monopolized in the project. Performing activity j in mode m takes d_{jm} periods and is supported by a set R_{jm} of renewable constrained resources of which a resources type k has resource requirement r_{jmk}. Once the project is initiated, the activity cost occurs. Just like DTCTP, MRC-DTCTP could be used to find the minimum project cost while meeting a given deadline (the deadline problem) or minimize the project duration meeting a given budget constraint (the budget problem). Also it might be of interest to compute the entire time–cost curve.

2.1. Calculation method of project cost

Project cost occurs when the renewable resources are occupied or nonrenewable resources are consumed. In knowledge intensive industry, e.g. manufacturing industry or software industry, the resources used in projects mainly consist of manpower resources, which belong to renewable resources. In general, the cost occurring from manpower resources (i.e. project member) can be correspondingly divided into two parts: one is salary paid by enterprise periodically, such as monthly pay and weekly pay; the other is paid by project leader based on his performance in the project. We define the former part of cost as project indirect cost since it paid by functional department not project, and the later part of cost as project direct cost since it is dominated by project leader directly. For project leader, the project indirect cost is renewable, and the project direct cost is nonrenewable for its total amount is limited for the project. It is accordant with the practice of the project cases in which activities might be executed in the crashing way to shorten the activity duration by the direct cost. For example, project leaders command some activities to be performed overtime and provide executants of the activities payment for additional work done outside of regular working hours.

According to above definitions, the standard-time pay of project member, usually called salary, belongs to project indirect cost, while the overtime time pay, paid by project leader directly to project member, belongs to project direct cost. For simplification, it is assumed that the project direct cost consists of only overtime pay occurring when activities are executed in crashing way. It is in accordance with the practice in the project mainly used manpower resources, especially the projects in knowledge intensive industry. As fore-mentioned, the project cost consists of the indirect cost and the direct cost. The former occurs with the occupation of renewable resources and the latter occurs with the consumption of nonrenewable resources. However, in MRC-DTCTP, both the direct cost and the indirect cost are calculated based on the renewable resource. The project indirect cost is related to the cost per unit and the occupation time of the renewable resources. As the renewable resources are monopolized in projects, their indirect costs paid by functional departments are constant in the whole project duration. That is, although a project member has lower or higher workload in the project, there will not be any differences in his income paid by functional department. Therefore, for a renewable resource $k \in K$, its indirect cost $uc_k = f_n \cdot u_k$, where f_n is the finish time of the end activity and u_k is the cost per unit time of k.

Moreover, k has two ways to be used: normal way and crashing way. There is no direct cost occurring when activities are performed in the normal way, and the direct cost only occurs when activities are performed in the crashing way to shorten the activity duration. The direct cost occurring on k can be calculated according to the formula $ec_{ik} = ed_{im} \cdot c_k$, where ec_{ik} is the direct cost when activity i uses resource k in the crashing way, ed_{im} is the crashing duration and c_k is the indirect cost per unit time of k when k is used in crashing way (illustrated by overtime cost). Therefore the project cost in MRC-DTCTP can be formulated as follows:

$$F_c = \sum_{i \in V} \sum_{m \in M_i} \left(x_{im} \cdot y_i \cdot \sum_{k \in K_{im}} (ed_{im} \cdot c_k) \right) + \sum_{k \in K} (u_k \cdot f_n). \tag{1}$$

The first term in this formula is the project direct cost, where x_{im} is a decision variable:

$$x_{im} = \begin{cases} 1, & \text{if activity } i \text{ is performed in mode } m, \\ 0, & \text{otherwise.} \end{cases}$$

y_i is the other decision variable:

$$y_i = \begin{cases} 1, & \text{if activity } i \text{ is performed in crashing way,} \\ 0, & \text{otherwise.} \end{cases}$$

K_{im} is the resource types set when the activity is performed in mode m.

The second term is the project indirect cost. As mentioned above, all the renewable resources are monopolized in the project, and then the project indirect cost from each renewable resource k has only related to its cost per unit time u_k and project make-span F_t, which is valued f_n, that is:

$$F_t = f_n. \tag{2}$$

2.2. Mathematical model of MRC-DTCTP

According to the fore-mentioned calculation method of project cost, the model of MRC-DTCTP is built based on MRCPSP and DTCTP. Just like that of DTCTP, there are three sub-problems of the MRC-DTCTP: (1) the deadline problem, for a given project deadline, and an assignment of processing times to activities with project make-span within the deadline that minimizes the total cost; (2) the budget problem, for a given non-negative budget, and an assignment of processing times to activities with total cost within the budget which minimizes the makespan; and (3) the time–cost curve problem, it is to construct the complete and efficient time–cost profile over the set of feasible project durations. This complete curve can be found by means of a horizon-varying approach, which involves the iterative solution of the deadline problem over the feasible project durations. As the time–cost curve problem should be calculated on either the deadline problem or the budget problem, we define the models of the deadline problem and the budget problem, respectively. The model of the deadline problem of MRC-DTCTP is formulated as follows:

$$\min \quad F_c, \tag{3}$$

s.t.

$$\sum_{m \in M_i} x_{im} = 1, \quad i \in V, \tag{4}$$

$$f_j - \sum_{m \in M_j} x_{jm} \cdot d_{jm} \geqslant f_i,$$

$$\forall (i,j) \in E, d_{jm} \in \{nd_{jm}, ed_{jm}\}, \tag{5}$$

$$\sum_{i \in A_t} \sum_{m \in M_i} (x_{im} \cdot r_{imk}) \leqslant R_k, \quad k = 1, \ldots, K$$

$$A_t = \{i | f_i - d_i < t \leqslant f_i\}, \tag{6}$$

$$F_t \leqslant T_{\max}. \tag{7}$$

In the formulation, the objective function (4) minimizes the project cost, which is calculated by formula (1). Constraint set (4) requires every activity to be performed in only one mode. Constraint set (5) represents the precedence relationships, where d_{jm} is the duration of an activity i when activity i is performed in mode m. The value of d_{jm} may be either the normal duration nd_{jm} in normal way or the crashing duration ed_{jm} in crashing way. Constraint set (6) indicates that for each time period $[t-1, t]$ and for each resource type k, the renewable resource amounts required by the activities in progress (i.e., A_t) cannot exceed the resource availability, where r_{imk} is units of resource k required by activity i if it is executed in mode m. Finally, the constraint (7) forces the project duration not to be longer than the upper bound for project deadline $T_{\max}$.

The model of budget problem which shares constraints Eqs. (5) and (6) from the model of deadline problem is

$$\min \quad F_t \tag{8}$$

s.t.

$$F_c \leqslant C_{\max}. \tag{9}$$

The objective function (8) minimizes the project duration, which is calculated by formula (2). Constraint (9) represents that the project cost is limited within the upper bound for project budget $C_{\max}$.

3. Solution algorithm

There are several intelligent algorithms used to solve hard optimization problems, such as genetic algorithm, simulated annealing, particle swarm optimization, and ant colony algorithm. As an intelligent optimization algorithm, the genetic algorithm is simple and mature, and has excellent performance among all those intelligent optimization algorithms for solving RCPSPs [17]. Nevertheless, up to now, the researches on employing the intelligent algorithms to solve DTCTP are still very few. The genetic algorithm presented by Liu et al. [12] for solving general DTCTP is the unique research on this field. In this paper, we developed an improved genetic algorithm to solve MRC-DTCTP, where some other genetic algorithms for RCPSPs and general DTCTP presented recently were referred.

Genetic algorithms are based on the genetic process of biological organisms. Over many generations, natural populations evolve according to the principles of natural selection, i.e. survival of the fittest. By mimicking this process, genetic algorithms, if suitably encoded, are able to evolve solutions to real world problems. Before a genetic algorithm can be run, an encoding (or representation) for the problem must be devised. A fitness function, which assigns a figure of merit to each encoded solution, is also required. During the run, parents are selected for reproduction and recombined to generate offspring. The operators of the genetic algorithms are explained in the remainder of this section.

3.1. Priority-based encoding

As the MRC-DTCTP is an extension of MRCPSP and DTCTP, the information reflected in genetic representation, i.e., chromosome should contain three aspects: (1) the priority values of activities that determine which activity is scheduled firstly when resource conflicts occur; (2) the modes to be selected for activities; and (3) the executing ways that determine whether activities are performed in the crashing way. For the priority values of activities, we adopt the priority list in the genetic representation introduced in [18]. Referring to the mode list introduced in [19], we adopt a second list called mode list to encode the modes selected for all the activities and a third list called way list to encode the executing way of all activities. Fig. 1 is an example of the structure of a chromosome.

A position of the three lists denotes an activity ID. In the priority list, the value $v(i)$ denotes the priority value associated with the activity i. $v(i)$ is an integer exclusively within [1, n] (n: number of total activities in a project), and the larger the integer, the higher the priority. $m(i)$ in the mode list is valued by one of the modes of the activity i. The value of $w(i)$ in the way list valued either 0 if the activity i is performed in normal way or 1 if i performed in crashing way.

1	2	3	4	5	Position: activityies ID
4	5	2	3	1	The priority values
1	3	2	3	1	The selected modes
0	1	1	0	0	The selected ways

Fig. 1. The structure of a chromosome.

3.2. Decoding based on serial schedule scheme

In this paper, the serial schedule scheme is used to generate project plan from the chromosome. In MRC-DTCTP, not only the resource constraints, but the time constraints or cost constraints must be considered. Moreover, the selected modes and selected ways of activities which are encoded in chromosome should be scheduled. Therefore the traditional serial schedule scheme can not serve to MRC-DTCTP directly. In this paper, a revised serial schedule scheme is presented for decoding the three lists in the chromosome given in the previous section to generate a project plan of MRC-DTCTP.

In the decoding based on serial schedule scheme by priority-based encoding, one activity in project from the decision set is selected with the priority rule, in which the activity with highest priority value is selected firstly, and scheduled at the earliest precedence. We select an activity obeyed feasible finish time and resource through the decoding of revised serial scheme, and the selected activity is removed from the decision set and put into the scheduled set with a selected mode and a selected way and the corresponding duration determined by both the selected mode and selected way. This, in turn, may replace a number of activities into the decision set, since all their predecessors are scheduled. The algorithm terminates at stage number n, when all activities are in the scheduled set. The procedure to generate initial schedules through revised serial schedule scheme is shown as Fig. 2.

The algorithm divides a schedule procedure into n stages, and only one activity is scheduled at a stage, which is denoted by integer h. At the initial stage, activity 1 is scheduled. PS_h is the partial scheduled plan at stage h. E_h is the feasible set of activities that meet the precedence relations at stage h. j^* is an activity that has the highest priority value in E_h. d_{j^*} is the duration of j^*, it is valued either ed_{j^*m} if the j^* is performed in the crashing way or nd_{j^*m} if the j^* performed in the normal way. ES_{j^*} is the earliest start time of activity j^* and valued by the maximum finish time of all the precedence activities of j^*. s_{j^*} is the start time of j^* and valued the earliest precedence and resource-feasible start time. f_{j^*} is the finish time of j^*. Add j^* to PS_h, stage h is over and the next stage $h + 1$ begins until all the activities are scheduled in PS_h.

3.3. Fitness computation

In the deadline problem, due to the constraint of the upper bound for project deadline T_{max}, not all the project

Procedure : decoding based on revised serial schedule method

$h := 1, PS_h := \{1\};$

WHILE $|PS_h| < J$ DO

BEGIN

$E_h := \{j \mid j \notin PS_h, P_j \subseteq PS_h\};$

$j^* \leftarrow j \mid v(j) = \max_{i \in E_h}\{v(i)\};$

$$d_{j^*} := \begin{cases} ed_{j^*m}, & x_{j^*m} = 1, y_{j^*} = 1 \\ nd_{j^*m}, & x_{j^*m} = 1, y_{j^*} = 0 \end{cases};$$

$ES_{j^*} = \max\{f_i \mid i \in P_{j^*}\};$

$s_{j^*} := \min\{t \mid ES_{j^*} \le t, r_{j^*k} \le vr_{kt}, vr_{kt} = a_k - \sum_{j \in A_t} r_{jk}, \tau = t, t+1, t+2, \ldots, t+d_{j^*}-1, k = 1,2,\ldots,K\};$

$f_{j^*} := s_{j^*} + d_{j^*};$

$PS_{h+1} := PS_h \cup \{j^*\};$

$h := h+1;$

END

Fig. 2. Revised serial schedule scheme.

plans generated from the chromosome meet the constraint (7). At the same time, in the budget problem, not all the chromosomes meet the constraint (9). All those chromosomes that can not meet the constraint (7) or the constraint (9) are regarded as non-feasible individuals and must be specially treated when the fitness of the individuals is computed. In this paper, a punishment mechanism in fitness computation is employed to punish the non-feasible individuals to ensure that the non-feasible individuals have a smaller fitness than the feasible ones. For the deadline problem, the revised objective function of an individual I is defined as

$$F'_c(I) = (1 - W_c) \cdot F_c(I), \tag{10}$$

where $F_c(I)$ is calculated as formulated in (1), W_c is a positive penalty-factor that is calculated as

$$W_c = \begin{cases} \frac{T_I - T_{\max}}{\overline{T} - T_{\max}}, & T_I > T_{\max} \\ 0, & T_I \leqslant T_{\max} \end{cases}, \tag{11}$$

where T_I is the project duration of the individual I, $\overline{T}$ is the max project duration of all the project plans generated up to the previous iteration.

For the budget problem, the revised objective function of an individual I is defined as

$$F'_t(I) = (1 - W_t) \cdot F_t(I), \tag{12}$$

where $F_c(I)$ is calculated as formulated in (8), W_t is the other positive penalty-factor that is calculated as

$$W_t = \begin{cases} \frac{C_I - C_{\max}}{\overline{C} - C_{\max}}, & C_I > C_{\max} \\ 0, & C_I \leqslant C_{\max} \end{cases}, \tag{13}$$

where C_I is the project cost duration of the individual I, $\overline{C}$ is the max project cost of all the project plans generated up to the previous iteration.

As the deadline problem and the budget problem all belong to the minimization problem, the fitness function defined in (10) and (12) must be converted as

$$f_c(I) = \overline{C} - F'_c(I), \tag{14}$$

$$f_t(I) = \overline{T} - F'_t(I). \tag{15}$$

(14) is the fitness function of the deadline problem, and (15) is the fitness function of the budget problem.

3.4. Selection operator

We have considered several variants of the selection operator (see, e.g., Michalewicz [20]), all of which follow a survival-of-the-fittest strategy. Let POP denote the size of population, the ranking method sorts the individuals with respect to their fitness values and selects the POP best ones while the remaining ones are deleted from the population. The proportional selection derives fitness based probabilities for the individuals in order to decide which individuals are selected for the next generation. Finally, in the tournament selection, a number of individuals compete for survival. These competitions, in which the least fit individual is removed from the population, are repeated until POP individuals are left. Based on the results of preliminary computational studies, we have chosen select operation composed of proportion selection & the best individual preservation. Then the selection probability of an individual is formulated as

$$p_c(I) = \frac{f_c(I)}{\sum_{j=1}^{POP} f_c(j)}, \tag{16}$$

$$p_t(I) = \frac{f_t(I)}{\sum_{j=1}^{POP} f_t(j)}. \tag{17}$$

(16) is used for the deadline problem while (17) for the budget problem.

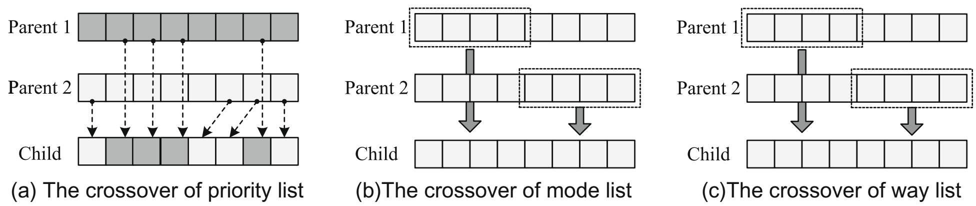

Fig. 3. Example of crossover.

3.5. Crossover operator

There are three lists in the proposed encoding and all of them must be crossed in crossover operator respectively, as shown in Fig. 2. The nature of the priority list can be viewed as a kind of permutation representation. A number of crossover operators have been investigated for permutation representation. The position-based crossover operator proposed by Syswerda [21] was adopted for the crossover of priority list. Essentially, the child takes some genes from one parent at random and fills vacuum position with genes from the other parent by a left-to-right scan, as shown in (a) of Fig. 2. For the mode list, the one-point crossover is used. We draw a random integer r with $0 \leqslant r \leqslant n$ as crossover-point, the positions $i = 1, 2, \ldots r$ in the child are taken genes from one parent, that is $m_d(i) = m_f(i)$, where $m_d(i)$ is the value of the position i in the child and $m_f(i)$ is the value of the position i in the father, as shown in (b) of Fig. 2. Meanwhile, the positions $i = r + 1, \ldots n$ in the child filled with the genes from the other parent, that is $m_d(i) = m_m(i)$, where $m_m(i)$ is the value of the position i in the mother. The handling method of the third list in proposed encoding, i.e., the executing way list, is same as that of the mode list, but the random crossover-point need to be regenerated, as shown in (c) of Fig. 3.

3.6. Mutation operator

We considered the three lists in the proposed encoding respectively. For the way list, a position is selected at random and its value is changed from 1 to 0 if the original value is 1 or from 0 to 1 if the original value is 0. For the mode list, the value at a selected random position, which has the same meaning as that of the way list, need to transformed to an other random integer within the range $[1, M_j]$. The handling method of priority list is somewhat complex due to its permutation nature. If the values of odd positions in the list are changed, the duplicate values in the list occur and the structure of chromosome is disrupted. We take the following steps to achieve the mutation of a chromosome: (1) generate two random positions within the range $[1, n]$ and they are not equal with each other; (2) swap their priority values of the two selected positions.

4. Computational experiment

The proposed algorithm has been coded in JAVA language. The experiment has been performed on a PC (1CPU, Intel 2.0 GHz, 512 MB RAM, 80 GB Hard Disk) under the Windows XP operating system. As MRC-DTCTP has several new features other than classical RCPSPs and general DTCTP, there is no benchmark instance for it. In this paper, we modified the MRCPSP instances in the well-known PSPLIB [22] as the test instances of MRC-DTCTP. The MRCPSP instances are modified as following five steps:

(1) The direct cost per unit time and the indirect cost per unit time are set for each renewable resource. For simplification purpose, we assume that the indirect cost per unit time of all the resources is in a fix proportion ψ to the direct cost, that is $\psi = c_k/u_k$.
(2) All the activity can be performed in the crashing way. For simplification purpose we assume that the duration of a activity performed in the crashing way is in a fix proportion ξ to the duration of the activity performed in the normal way, that is $\xi = ed_i/nd_i$.
(3) If there are the same resource types in more than one mode in an activity, only the mode that has the longest duration is reserved while the other modes are removed.
(4) All the activity durations are lengthened for convenient to be crashed, and the new duration value is set the triplication of the original value.
(5) The nonrenewable resources in original instances are removed.

To verify the effectiveness and efficiency of the proposed algorithm, an exact algorithm based on branch and bound, which is introduced in Appendix 1, is developed. The computation results of the improved genetic algorithm are compared with that of the exact algorithm.

The experiment results are compared with respect to the following measures:

Avdev average relative deviation from optimality (in%).
#best number of instances for which the algorithm finds the best solution from the number of all the instances (in%).
Cpu1 average computation time of an instance for which the proposed genetic algorithm is used.
Cpu2 average computation time of an instance for which the branch and bound based algorithm is used.

Table 1
The effectiveness of different mutation probabilities.

Mutation probability	#Best (%)	Avdev (%)	Cpu_1 (s)	Cpu_2 (s)
0.01	96.00	5.2	33.22	156.59
0.03	97.23	4.5	33.13	
0.05	97.11	3.1	33.23	
0.07	97.02	3.9	33.21	

Table 2
The values of parameters in experiment instance.

ξ	ψ	R_1	u_1	R_2	u_2
0.6	3.5	9	2	4	1

Table 3
The activities of project instance.

Activity ID	#Successors	#Modes	R_1	R_2	Duration	Crashing duration
1	2, 3, 4	1	0	0	0	
2	5, 6	1	5	0	27	17
		2	0	6	30	18
3	10, 11	1	7	0	3	2
		2	0	4	15	9
4	9	1	6	0	24	15
5	7, 8	1	2	0	18	11
		2	0	5	30	18
6	10, 11	1	0	8	12	8
		2	2	0	18	11
7	9, 10	1	0	7	18	11
		2	5	0	24	15
8	8	1	4	0	30	18
9	12	1	1	0	30	18
10	12	1	0	2	3	2
		2	4	0	27	17
11	12	1	0	1	30	18
12		1	0	0	0	0

As shown in Table 1, the experiment results suggest that the algorithm proposed in this paper is effective to solve the MRC-DTCTP, and the execution efficiency has been improved remarkably over that of the algorithm based on branch and bound. When the population size $POP > 30$, the difference of population sizes has little effect on the algorithm. However the different mutation probability has important influence on the effectiveness of the algorithm and an appropriate mutation probability must be selected for the proposed algorithm. On the basis of the experiment results, the mutation probability that takes a value 0.05 is better than others.

Now, taking the instance J102_2.mm (available in the PSPLIB: http://129.187.106.231/psplib/j10.mm.zip/ J102_2.mm) as an experiment instance, a complete time–cost tradeoff curve has been drawn based on the deadline problem to investigate the characteristics of MRC-DTCTP. It was realized by solving the deadline problem of MRC-DTCTP and computing the minimization values of project cost by assigning all possible values to $T_{\max}$. The values of parameters are set as Table 2, where R_k is the quality of resource type k, u_k is the indirect cost per unit time of resource type k. Based on aforementioned method, the instance J202_2.mm is changed into a new project instance of MRC-DTCTP as Table 3.

The time–cost tradeoff curve is drawn as Fig. 4, in which (a) is the efficient time–cost tradeoff curve and (b) is the time constraint-cost curve. It can be seen that there are only a finite number of efficient points obtained in the time–cost tradeoff curve and the time constraint-cost curve gradually drops. The computation results are detailedly listed in Table 4, where the data in the fourth column is activity lists representing the near-optimal solution. An activity list is a precedence-feasible list, in which both the precedence relations and the priority list in the chromosome of the genetic algorithm are considered, and the more close to the head of the list, the activity is scheduled earlier. As the selected modes and ways of activities should be given, each node in the activity list is a triple $[i, m(i), w(i)]$.

It is shown that MRC-DTCTP has no solution when the time constraint $T_{\max}$ takes a too small value. The reason for it is that the problem subjects to not only the time

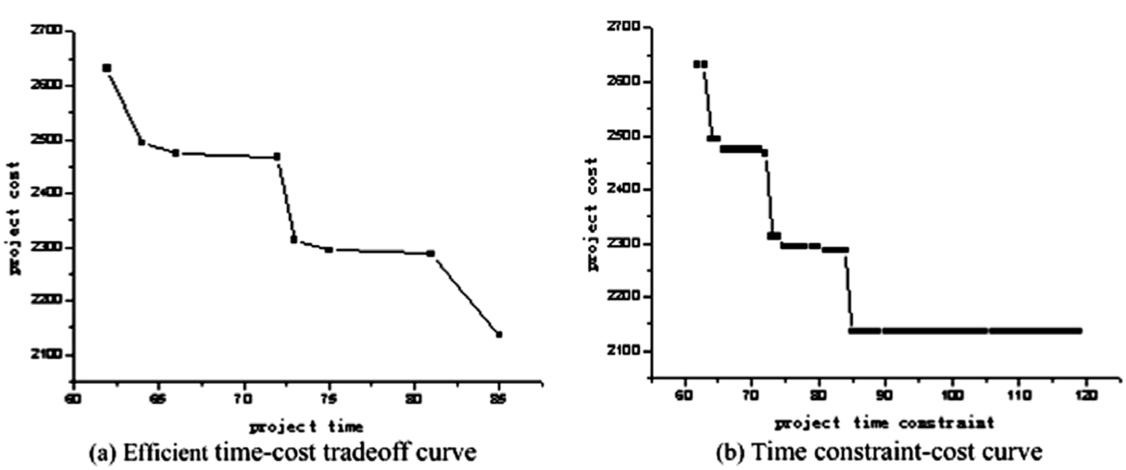

Fig. 4. Time–cost tradeoff curve.

Table 4
Detailed computation results of the time–cost curve.

Time constraint	Efficient time	Efficient cost	Activity list
–61	–	–	–
62–63	62	2631	[1,0,0][2,1,1][3,1,0][4,1,0][5,1,1][6,2,0][7,2,0][8,1,1][9,1,1][10,2,0][11,1,1][12,0,0]
64–65	64	2493	[1,0,0][2,2,1][3,1,0][4,1,0][5,1,1][6,2,0][7,2,0][8,1,1][9,1,1][10,2,0][11,1,1][12,0,0]
66–71	66	2474	[1,0,0][2,2,1][3,1,0][4,1,0][5,1,1][6,1,0][8,1,1][11,1,0][7,2,0][9,1,1][10,2,0][12,0,0]
72	72	2466	[1,0,0][2,2,1][3,1,0][4,1,0][5,1,0][6,1,0][8,1,1][11,1,0][7,2,0][9,1,1][10,2,0][12,0,0]
73–74	73	2313	[1,0,0][2,1,0][3,1,0][4,1,0][5,1,1][6,2,0][7,2,0][8,1,1][9,1,1][10,2,0][11,1,1][12,0,0]
75–80	75	2294	[1,0,0][2,1,0][3,1,0][4,1,0][5,1,1][6,1,0][8,1,1][11,1,0][7,2,0][9,1,1][10,2,0][12,0,0]
81–84	81	2286	[1,0,0][2,1,0][3,1,0][4,1,0][5,1,0][6,1,0][8,1,1][11,1,0][7,2,0][9,1,1][10,2,0][12,0,0]
85–	85	2136	[1,0,0][2,1,0][3,1,0][4,1,0][5,1,1][6,1,0][7,1,0][8,1,0][9,1,1][10,2,0][11,1,0][12,0,0]

constraints but also the resource constraints. If all the activities are performed in crashing way and the resource constraints are considered, the MRC-DTCTP can be transformed into MRCPSP. With the minimal project duration as the objective function, solving the MRCPSP of the above experiment instance, the object value is 62, which equals with the minimal one of efficient times in the time–cost curve. When the project duration is the minimal one in MRC-DTCTP, only those activities that effect project duration should be performed in crashing way, and to save cost, the other activities are still performed in normal way. Because the problem is discrete, the efficient time is not always equal to the time constraints. As it can be seen that when the time constraint $T_{\max} \geqslant 85$, it is meaningless to further relax the time constraints. The main reason is that both the direct cost and the indirect cost are considered in the MRC-DTCTP. It follows that lengthened project duration can cause the decreased project direct cost while the indirect project cost will be increased at the same time. Due to the renewable resource constraints in the MRC-DTCTP, upper and lower bounds of the project duration and project cost can not be calculated by CPM method directly. Through the time–cost curve, the minimal possible project duration and its corresponding cost can be determined. Based on the complete time–cost curve, the minimal possible project cost and its corresponding time can be determined in like manner too. From the time–cost curve, it is easy for the project managers to balance the project duration, project cost according to the given resources and the commands of project stakeholders.

5. Conclusion

The discrete time–cost tradeoff problem is a well-established project scheduling problem. It has attracted several researchers in the past decades. In this paper, aiming at the characteristics of knowledge intensive projects, we considered renewable resources, especially manpower resources in DTCTP, and the general DTCTP was extended to a new multi-mode resource-constrained DTCTP model, which inherits some features of the well-known MRCPSP. In MRC-DTCTP, time constraints, renewable resource constraints, cost constraints were considered. The relationships among project duration, renewable resources as well as the direct project cost and the indirect project cost are balanced in the model. According to the characteristics of the proposed problem, an improved genetic algorithm was presented to solve it. Through compared with the exact algorithm, the effectiveness and efficiency of the presented genetic algorithm were verified. An example of time–cost curve of MRC-DTCTP was drawn based on the deadline problem and the characteristics of the MRC-DTCTP were investigated based on it. The complete time–cost curve can effectively help the project managers to balance the project duration, project cost based on the given resources.

MRC-DTCTP is adapted to schedule the projects which mainly use the renewable resources. For the projects where the renewable resources and nonrenewable resources have the same importance, the method presented in this paper has only a reference meaning. Moreover, as all the renewable resources are monopolized and can't be shared with other projects, the multi-project management is hard to be scheduled with MRC-DTCTP. These are the main limitations of the method proposed in this paper.

Appendix 1. The algorithm based on branch and bound for solving MRC-DTCTP

The algorithm is developed based on the deadline problem of MRC-DTCTP. The branch and bound is used to search a precedence-feasible activity list that can be imported to serial schedule scheme to generate a optimal project plan. A node in the search tree is a partial activity list that responding to a partial project plan. Each element in activity list is an activity, which has its selected mode and selected way. Let G denote the activity set of a project and G' denote the partial precedence-feasible activity list on the node of the search tree, an element in G' can be denoted by a triple $[i, m(i), w(i)]$, where i is an activity ID, $m(i)$ is its selected mode, $w(i)$ is its selected way. In the precedence-feasible activity list, the more close to the head of the list, the more early the activity is scheduled.

Definition 1. For a partial precedence-feasible partial activity list G', considering the precedence relations, all of the activities that can be add to the tail of G' is defined as *feasible activity set M*. The necessary and sufficient condition for $j \in M$ is that $j \notin G'$ and $P_j \subset G'$.

Take an activity from the feasible activity set M and give it a designated mode and way, a new node of the search tree is created when the activity is added to G'. Let $\bar{G}'$ denote a activity set in which all the activity is in G but not in G'. The partial project plan, which satisfy all the constraint sets, can be generated from a partial precedence-feasible partial activity list G' by serial schedule scheme.

Definition 2. Considering all constraints of the problem, a partial project plan can be generated from G'. Based on the partial project plan, let the activities in $\bar{G}'$ performed in the crashing way and in the mode which has the shortest duration, all of them can be scheduled in the partial project plan by CPM(critical path method) if all constraints of the problem are neglected, and a *temporary complete project plan* can be generated.

Definition 3. The duration of the temporary complete project plan denoted is defined as *the shortest possible project duration* of G', denoted by $T^{'}$.

Definition 4. The direct cost occurs in G' is considered and that occurs in $\bar{G}'$ is neglected, the project cost of the *temporary complete project plan* is defined as the minimum possible project cost of G', denoted by $C^{'}$.

For the partial activity list G' of a node in the search tree, by moving activities $\bar{G}'$ to G' iteratively until all the activities is put in G', a complete activity list begins with G' can be achieved.

Theorem 1. For a G', if its shortest possible project duration $T' > T_{\max}$, the project duration T'' of the project plan generated from any complete activity list begins with G' must more than $T_{\max}$ when all the constraints of the problem are considered, that is $T'' > T_{\max}$.

Theorem 2. For a G', if its minimum possible project cost $C' \geqslant C_{\max}$, the project cost C'' of the project plan generated from any complete activity list begins with G' must more than $C_{\max}$ when all the constraints of the problem are considered, that is $C'' \geqslant C_{\max}$.

The two theorems are rather intuitive so the proofs of them are omitted. The algorithm generates lower bounds based on the above two theorems. For the deadline problem, the specific procedures of the algorithm are as follows:

(1) Let $C = 99{,}999$ is initial lower bound, add activity 1 to the feasible partial activity list. An element of feasible partial activity list must be a triple $z = [j, m(j), y(j)]$. Since activity 1 is dummy activity, [1,0,0] can be added to G' directly, the initial feasible partial activity list is $G' = \{[1,0,0]\}$ and the top node of search tree is created with the initial G'.

(2) A feasible activity set M is computed based on G'. As every activity should be selected a mode and a way, there are several possible elements z for each activity j. The possible elements of all the activities in M are denoted by a set Z, that is, $z \in Z$. Therefore, a node of the search tree has a Z for it.

(3) Select a $z \in Z$ and add it to the feasible partial activity list G', and a new node of search tree is created. The element z is marked selected.

(a) A complete activity list is formed if all the activities have been selected once. $Z = \varphi$. Importing the complete activity list to the serial schedule scheme, the complete project plan can be achieved and the project cost F_c can be computed. If $F_c < C$, C is valued by F_c and a new lower bound is created.

(b) If all the elements in Z have been marked selected, removing the last z from G' and returning to previous node, the step (3) is executed repeatedly (recursive algorithm is used here). If its previous node is the top node, step (4) is executed and the algorithm terminates.

(c) The shortest possible project duration T' and the minimum possible project cost $C^{'}$ are computed. If $T' > T_{\max}$ or $C' \geqslant$ C, the branch from the current nod of search tree is cut. The last element is removed from G' and the algorithm returns to step (3). If $T' \leqslant T_{\max}$ and $C^{'} < C$, the algorithm returns to step (2).

(4) The algorithm terminates. If $C = 99999$, there is no solution for the problem. Otherwise, the problem is solved and the value of C is the best objective function value.

References

[1] Brucker P, Drexl A, Mohring R, et al. Resource-constrained project scheduling: notation, classification, models, and methods. Eur J Operat Res 1999;112(1):3–41.
[2] Demeulemeester EL, Herroelen WS. Project scheduling: a research handbook. Boston: Kluwer Academic Publishers; 2002.
[3] Kolisch R, Padman R. An integrated survey of deterministic project scheduling. Omega 2001;29:249–72.
[4] Hindelang TJ, Muth JF. A dynamic programming algorithm for decision CPM networks. Operat Res 1979;27(2):225–41.
[5] Prabuddha DE, Dunne EJ, Ghosh JB, et al. Complexity of the discrete time–cost tradeoff problem for project networks. Operat Res 1997;45(2):302–6.
[6] Vladimir G Deineko, Woeginger Gerhard J. Hardness of approximation of the discrete time–cost tradeoff problem. Operat Res Lett 2001;29(5):207–10.
[7] Patterson JH, Harvey RT. An implicit enumeration algorithm for the time/cost tradeoff problem in project network analysis. Found Control Eng 1979;4(2):107–17.
[8] Demeulemeester E, Herroelen W, Elmaghraby SE. Optimal procedures for the discrete time/cost trade-off problem in project networks. Eur J Operat Res 1996;88(1):50–68.
[9] Demeulemeester E, De Reyck B, Foubert B, et al. New computational results on the discrete time/cost trade-off problem in project networks. J Operat Res Soc 1998;49(6):1153–63.
[10] Erenguc SS, Ahn T, Conway DG. The resource constrained project scheduling problem with multiple crashable modes: an exact solution method. Naval Res Log 2001;48(2):107–27.
[11] Akkan C. A Lagrangian heuristic for the discrete time–cost tradeoff problem for activity-on-arc project networks. Working Paper, Koc University, Istanbul; 1998.

[12] Liu SX, Wang MG, Tang LX, et al. Genetic algorithm for the discrete time/cost trade-off problem in project network. J Northeastern Univ [China] 2000;21(3):257–9.
[13] Elmaghraby SE, Kamburowski J. The analysis of activity network under generalized precedence relations. Manage Sci 1992;38(9): 1245–63.
[14] Ann T, Erenguc SS. The resource constrained project scheduling problem with multiple crashable modes: a heuristic procedure. Eur J Operat Res 1998;107(2):250–9.
[15] Erenguc SS, Tufekci S, Zappe CJ. Solving time/cost trade-off problem with discounted cash flows using generalized benders decomposition. Naval Res Log 1993;40(1):25–50.
[16] Vanhoucke M, Demeulemeester E, Herroelen W. Discrete time/cost trade-offs in project scheduling with time-switch constraints. J Operat Res Soc 2002;53(7):741–51.
[17] Kolisch R, Hartmann S. Experimental investigation of heuristics for resource-constrained project scheduling: an update. Eur J Operat Res 2006;174(1):23–37.
[18] Gen M, Cheng R. Genetic algorithm and engineering optimization. New York: John Wiley and Sons; 2000.
[19] Hartmann S. Project scheduling with multiple modes: a genetic algorithm. Ann Operat Res 2001;102(2):111–35.
[20] Michalewicz Z. Heuristic methods for evolutionary computation techniques. J Heuristics 1996;1:177–206.
[21] Davis L. Handbook of genetic algorithm. New York: Van Nostrand Reinhold; 1991.
[22] Kolisch R, Sprecher A. PSPLIB – a project scheduling library. Eur J Operat Res 1996;96:205–16.