Discrete Optimization

Robust scheduling and robustness measures for the discrete time/cost trade-off problem

Öncü Hazır [a,c,*], Mohamed Haouari [b,d], Erdal Erel [a]

[a] Faculty of Business Administration, Bilkent University, Ankara, Turkey
[b] Department of Industrial Engineering, Faculty of Engineering, Ozyegin University, Istanbul, Turkey
[c] Industrial Engineering Department, Çankaya University, Ankara, Turkey
[d] ROI, Tunisia Polytechnic School, University of 7th November at Carthage, Tunisia

ARTICLE INFO

ABSTRACT

Projects are often subject to various sources of uncertainties that have a negative impact on activity durations and costs. Therefore, it is crucial to develop effective approaches to generate robust project schedules that are less vulnerable to disruptions caused by uncontrollable factors. In this paper, we investigate the robust discrete time/cost trade-off problem, which is a multi-mode project scheduling problem with important practical relevance. We introduce surrogate measures that aim at providing an accurate estimate of the schedule robustness. The pertinence of each proposed measure is assessed through computational experiments. Using the insights revealed by the computational study, we propose a two-stage robust scheduling algorithm. Finally, we provide evidence that the proposed approach can be extended to solve a complex robust problem with tardiness penalties and earliness revenues.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

In project management, it is often possible to expedite the duration of some activities and therefore reduce the project duration with additional costs. This time/cost trade-off has been widely studied in the literature focusing on linear and continuous time/cost relationships. In this paper, we address the discrete version, namely the discrete time/cost trade-off problem (DTCTP), which is a multi-mode project scheduling problem having practical relevance. Project managers often allocate more resources to accelerate the activities and each resource allocation defines an execution mode. Thus, multiple alternatives usually exist to execute an activity. DTCTP utilizes only one single nonrenewable resource (money) and does not explicitly consider *renewable* resources (e.g. machines, equipment and staff), which are available at constant amounts in every instance of the planning period.

Formally, the DTCTP is defined as follows. Given a project with a set of n activities along with a corresponding precedence graph $G = (N,A)$, where N is the set of nodes that refer to the activities of the project, and $A \subset N \times N$ is the set of immediate precedence constraints on the activities. It is noteworthy that G also includes two dummy "start" and "end" nodes indexed by 0 and $n + 1$, respectively. Each activity j $(j = 1,\ldots,n)$ can be performed at one of the modes chosen from the set M_j. Each mode $m \in M_j$, is characterized by a processing time p_{jm} and a cost c_{jm}.

Two basic versions of the DTCTP have been defined in the literature so far: the deadline problem (DTCTP-D) and the budget problem (DTCTP-B). In the deadline problem, given a project deadline δ, one of the possible modes is assigned to each activity so that the makespan does not exceed δ and the total cost is minimized. The budget problem, on the contrary, minimizes the makespan while not exceeding a maximum preset budget B. Despite its practical relevance, the research on DTCTP is rather sparse due to its inherent computational complexity (it has been shown to be strongly NP-hard for general activity networks (De et al., 1997)). In their comprehensive review papers, De et al. (1995) and Weglarz et al. (2010) discuss the problem characteristics as well as exact and approximate solution strategies. We refer the readers to the papers of Demeulemeester et al. (1996, 1998) for exact algorithms and to Skutella (1998), Akkan et al. (2005), Vanhoucke and Debels (2007) and Hafızoğlu and Azizoğlu (2010) for approximate algorithms. Furthermore, Erengüç et al. (1993) apply Benders decomposition to solve the time/cost trade-off problem with discounted cash flows, which combines the DTCTP and the payment-scheduling problem.

* Corresponding author at: Industrial Engineering Department, Çankaya University, 06530 Ankara, Turkey. Tel.: +90 312 2844500; fax: +90 312 2848043.
  E-mail address: hazir@cankaya.edu.tr (Ö. Hazır).

The existing studies on DTCTP generally assume complete information and deterministic environment. However, in practice, projects are often subject to various sources of uncertainty that may arise from the work content, resource availabilities, project network, etc. A schedule that is optimal with respect to project duration or cost may largely be affected by these disruptions. Therefore, it is crucial to develop effective approaches to generate project schedules, which are less vulnerable to disruptions caused by these uncontrollable factors. To the best our knowledge, the only paper which addresses uncertainty on DTCTP is by Klerides and Hadjiconstantinou (2010); they used stochastic programming to model uncertain activity durations.

The contribution of our paper is threefold. First, we introduce a new version of the DTCTP under uncertainty with tardiness penalties and earliness revenues. Second, we propose some surrogate measures to evaluate schedule robustness. The quality of the proposed schedules is assessed through several performance measures. Finally, we develop a two-phase approach for generating robust schedules. The solution approach integrates an analytical tool to support the decision makers in budget allocation decisions and a robust scheduling algorithm. The developed scheduling algorithm addresses the crucial need to construct robust project schedules that are less vulnerable to disruptions caused by uncontrollable factors. Furthermore, it serves as a basis to develop decision support systems (DSS) to help project managers in planning under uncertain environments.

2. Discrete time/cost trade-off problem under uncertainty

Stochastic programming and robust optimization are two fundamental optimization approaches under uncertainty. *Stochastic programming* uses probabilistic models to describe uncertain data in terms of probability distributions. Typically, the average performance of the system is examined and expectation over the assumed probability distribution is taken. *Robust optimization* is a modeling approach to generate a plan that is insensitive to data uncertainty. Generally, the worst-case performance of the system is optimized and plans that perform well under worst-case scenarios are sought. Since it is worst-case oriented, it is a conservative methodology. When accurate distributional information is available, stochastic programming has the advantage of incorporating this available distributional information; however, stochastic programming models are usually computationally more demanding. If the decision-maker either does not have or cannot access this information, robust optimization is more appropriate.

First, we formulate a stochastic programming model for a new version of DTCTP to determine mode assignments such that the expected net profit is maximized. A penalty cost is incurred if the project finishes later than the specified due date. In case of early completion, we assume that the enterprise can make an additional profit by elongating the operating period. We denote the revenue rate and the tardiness penalty rate per unit time by α, and β, respectively. The activity durations are modeled as random variables. We introduce the random vector P and denote a particular realization with vector p. The corresponding stochastic programming problem involves the completion times as random variables. A mixed integer-programming model of the stochastic DTCTP is as follows:

$$\text{Max} \quad E[\alpha \max\{0, \delta - C_{n+1}(P)\} - \beta \max\{0, C_{n+1}(P) - \delta\}] \tag{1.0}$$

$$\text{Subject to} \quad \sum_{m \in M_j} x_{jm} = 1 \quad \forall j \in N, \tag{1.1}$$

$$C_j(P) - C_i(P) - \sum_{m \in M_j} P_{jm}\, x_{jm} \geqslant 0 \quad \forall (i,j) \in A, \tag{1.2}$$

$$\sum_{j \in N} \sum_{m \in M_j} c_{jm} x_{jm} \leqslant B, \tag{1.3}$$

$$C_j(P) \geqslant 0 \quad \forall j \in N \cup \{0\}, \tag{1.4}$$

$$x_{jm} \in \{0,1\} \quad \forall m \in M_j, \ \forall j \in N. \tag{1.5}$$

The continuous random variable $C_j(P)$ denotes the completion time of activity j. The binary decision variable x_{jm} assigns modes to the activities and is equal to 1 if mode m is chosen for activity j, and 0 otherwise (1.5). While maximizing the total expected profit (1.0), a unique mode should be assigned to each activity (1.1), and the precedence constraints should not be violated (1.2). Furthermore, the given budget, B, should be met (1.3). Defining the budget is a critical issue for project managers and we will address this issue in Section 4.2.

When an activity mode is selected, the activity durations and costs are defined. As the activity durations depend on the activity modes, considering the precedence relations (1.2), the entire duration distribution function is dependent on the chosen mode. It is noteworthy that the completion times of the activities and hence the project makespan are random variables.

The proposed problem is widely encountered in scheduling Build–Operate–Transfer (BOT) projects. BOT model describes the situation in which a public service or an infrastructure investment is made and operated for a specific period by a private enterprise, and then transferred to a public institution. One application area is the private toll roads. A private enterprise first constructs the road and operates it for some time, such as 10–20 years, and then transfers the right to operate these roads to the public. The enterprise can elongate the operating period by completing construction earlier and increase the profit.

Projects are temporary and unique; they have a finite duration and distinguishing characteristics. In many real-life projects, managers either do not have or cannot access on probability information. Therefore, in this research, we use robust optimization to generate robust project schedules that are minimally affected from disruptions in activity durations. Herroelen and Leus (2005) divide schedule robustness into two groups: solution robustness (stability) and quality robustness. They define *solution robustness* as the insensitivity of the activity start times to variations in the input data, and *quality robustness* as the insensitivity of schedule performance (such as project makespan or cost) with respect to disruptions. Quality robust scheduling aims to construct schedules in such a way that the value of the performance measure is affected as little as possible from disruptions.

The most popular approach of project management aiming quality robustness is the critical chain project scheduling (CCPS) that has been introduced by Goldratt (1997) who applied the theory of constraints (TOC) to project management. CCPS suggests inserting *buffers*, which are protection mechanisms against uncertainty in the duration of activities, into the schedule. Safety factors are eliminated from individual activities and aggregated at the end as a project buffer. Aggressive time estimates, usually lower than the nominal durations, are used in building the baseline schedule, and in this way the project personnel is forced to increase productivity. Nominal values are

the most likely duration values assigned to each activity by the project manager. Aggressive estimates may lead to quality problems, since these buffers aim to prevent the project from exceeding the deadline.

We refer the reader to Herroelen and Leus (2001), for an experimental evaluation of CCPS, and to Tukel et al. (2006) for analysis of several buffer sizing methods. Van de Vonder et al. (2005, 2006) examined the trade-off between the quality robustness and solution robustness. In a more recent study, Van de Vonder et al. (2008) developed robust scheduling heuristics and compared their performance. In these papers, the schedule robustness is usually evaluated using simulation and usually measured with the probability that the project is completed on time.

3. Measuring robustness

Developing quantitative metrics that provide a good estimate of schedule robustness is essential for building robust scheduling algorithms. The baseline schedules are execution plans prepared prior to the project execution. The schedules that are created by using these robustness measures could absorb unanticipated disruptions. Existing robust scheduling studies generally employ either direct measures, which are derived from realized performances, or heuristic approaches, which utilize simple surrogate measures. We refer the readers to Gören and Sabuncuoğlu (2008) for a more detailed discussion of the robustness measures in machine scheduling.

There are some difficulties of using the optimization problem (1.0)–(1.5) directly. In order to calculate the expected profit, we need to define the possible disruption scenarios with their probability information. First, disruption scenarios cannot be easily defined or determined a priori, estimating the probability distributions accurately is usually difficult, and furthermore there may be too many disruption scenarios to be considered. Second, computational burden of optimizing a direct measure in a real-life project environment can be quite high due to the fact that analytical determination of the effect of a disruption on other activities and project completion, especially in multi-disruption case, is difficult to model in complex networks as project networks consist of multiple paths that typically intersect. Hence, a reasonable approach to increase the computational efficiency is to use good surrogate measures and determine an algorithm optimizing the surrogate measure.

In this paper, we concentrate on slack-based measures to assess schedule robustness. We use the nominal activity durations to find out the activity slacks. They are the most likely values assigned to each activity by the project manager. Two types of slacks are widely used in project management literature: total slack and free slack. Total slack (TS) is the amount of time by which the completion time of an activity can exceed its earliest completion time without delaying the project completion time. Free slack (FS) is the amount of time by which the completion time of an activity could be delayed without affecting the earliest start time of its immediate successors in the project. The total slack concept is closely related to quality robustness, whereas the free slack is related to stability of a schedule. Since our primary goal is to generate quality robust DTCTP schedules, we restrict our attention to total slack-based surrogate measures.

In the literature, there are only a few studies that propose measures to assess the robustness of project schedules. Commonly, they suggest the use of surrogate measures for the resource constrained project scheduling problem. Al-Fawzan and Haouari (2005) use the sum of free slacks as a surrogate metric for measuring the robustness of a schedule. Kobylański and Kuchta (2007) discuss a limitation of this measure and propose using the minimum of all free slacks or the minimum of free slack/duration ratios. However, focusing on the minimum values has the weakness that two schedules with the same minimum values could have different slack patterns and the measures proposed by Kobylański and Kuchta (2007) fail to differentiate between these schedules. On the other hand, Lambrechts et al. (2008) introduce a free slack utility function and use this function in their tabu search algorithm to generate robust schedules. Recently, Chtourou and Haouari (2008) propose twelve predictive indicators for resource constrained networks.

3.1. Proposed measures

We consider a profit-based objective function in Model (1.0)–(1.5) that leads to early project completion. The schedules with less critical or potentially critical paths or activities have tendency to absorb interruptions and avoid delaying the project completion. We also notice that the criticality of an activity is defined by its slack. Therefore, one could reasonably expect that more profits could be gained and tardiness penalties might be mitigated through minimizing the delays in makespan by integrating larger slacks in the schedule. However, not only the magnitude but the placement of slacks in the schedule and the relationship with activity durations is also important. In the sequel, we make a thorough analysis of activity slacks considering these factors and introduce slack-based measures and compare them against each other through simulation.

3.1.1. Average slack (RM_1)
The average slack, or equivalently, the sum of the slacks has been commonly used to assess schedule robustness in scheduling literature. Experimental studies of Leon et al. (1994) and Mehta and Uzsoy (1998) on job shop scheduling reveal that there is a high correlation between robustness and the average slack value.

3.1.2. Weighted slack
Minimizing the average of slacks assumes that the contribution of slacks to robustness is the same for each activity. However, some of the activities are more likely to delay the project completion; hence, larger slacks should be allocated to these activities. Chtourou and Haouari (2008) propose to use the number of immediate successors as the weights since the activities having larger number of successors are more likely to affect the project makespan. We adapt this measure using total slacks:

$$RM_2 = \sum_{i=1}^{n} NIS_i \times TS_i. \tag{2}$$

In this formulation, NIS_i refers to the number of immediate successors of activity i. Moreover, we suggest to use the total number of all successors, NS_i ($i = 1,\ldots,n$):

$$RM_3 = \sum_{i=1}^{n} NS_i \times TS_i. \tag{3}$$

3.1.3. Slack utility function

The average or weighted slack approach assumes the same return for every unit of slack assigned. This approach might unnecessarily inflate the slacks for some activities. One alternative approach is to use a function that has diminishing returns per extra unit of slack. For the resource constrained project networks, Lambrechts et al. (2008) use a free slack-based utility function. We use TS and assume equal weights for each activity, we propose the following measure:

$$RM_4 = \sum_{i=1}^{n} NS_i \sum_{j=1}^{TS_i} e^{-j}. \tag{4}$$

In practice, it would make more sense to evaluate the activity slacks with respect to the activity's duration, since the higher the slack/duration ratio (SDR), the higher is its capacity to prevent a delay. The reason for this is that as the activity durations increase, the probability of observing longer delays increases. Thus, as an alternative approach, we propose to use SDR to assess robustness as follows:

$$RM_5 = \sum_{i=1}^{n} NS_i \sum_{j=1}^{\lceil SDR_i \rceil} e^{-j}, \quad \text{where } SDR_i = \frac{TS_i}{p_i}. \tag{5}$$

Chtourou and Haouari (2008) define a threshold level and assume zero return when the slacks allocated is more than the threshold. We adapt this measure as follows:

$$RM_6 = \sum_{i=1}^{n} Min\{TS_i, frac \times p_i\}. \tag{6}$$

The parameter $frac$ refers to the expected percentage increase in activity duration. We set this parameter to 20% in our experiments.

3.1.4. Dispersion of slacks

In addition to the magnitude of activity slacks, their dispersion over the activities might be important to evaluate schedule robustness. Low variability of activity slacks is expected to be beneficial as it evenly distributes the delay risk among activities. As a dispersion measure, we propose using the coefficient of variation (CV) and formulate as:

$$RM_7 = \frac{\sqrt{Var(SDR)}}{\overline{SDR}}, \quad \text{where } \overline{SDR} = \frac{\sum_{i=1}^{n} SDR_i}{n}. \tag{7}$$

Since high slack quantities and small variability among activities are preferred regarding the robustness, the schedules with smaller CV are deemed to be more robust.

3.1.5. Percentage of potentially critical activities (RM_8)

Traditionally in the literature, the activities with zero total slack are defined to be critical activities. We use the SDR as a criterion to identify the criticality of activities and define the activities that have slack values less than $100\xi\%$ of the activity duration as potentially critical activities (PCA), i.e. $PCA = \{j: TS_j/p_j \leqslant \xi\}$. In our implementation, we set the slack duration threshold to 25%, i.e. $\xi = 0.25$. Since delays in PCA are likely to result in delays in the project completion time, schedules with fewer critical activities are preferred with respect to the robustness. We employ the ratio of the cardinality of PCA to the total number of activities as RM_8.

3.1.6. Project buffer size

Project buffers are inserted at the end of projects to provide protection against possible delays. The critical chain project scheduling (CCPS) emphasizes the importance of buffer management and proposes to insert project buffers. Schedules with larger project buffers are preferred regarding robustness, but it may also deteriorate the project cost. We use the project buffer size as percentage of project deadline,

$$RM_9 = 100 \frac{\delta - C_{n+1}}{\delta}, \tag{8}$$

C_{n+1} denotes to the earliest project completion time. Project buffer size used in this measure is, in fact, the total slack of activity $n + 1$, which is the amount of time by which the completion time of the project can be delayed without exceeding the project deadline.

Each measure has its own characteristic. For instance, RM_2 and RM_3 integrate project specific information regarding network characteristics so that the performance of the measure improves. As the number of successors increases, it is more likely that the disruptions in these activities might create a domino effect and increase the completion times of other activities, and it is more likely to have due date violations, which are penalized by the objective function in (1.0). Our experimental study also shows that the correlation increases as the number of successors is integrated (RM_2 and RM_3). Measures from RM_4 to RM_8 utilize functions of slacks as alternative measures. For instance, RM_8 considers all the activities that are on paths that are likely to be critical after some possible disruptions.

Unlike the other measures, RM_9 results in aggregating the safety factor at the end as a project buffer instead of inserting slacks in between individual activities. Total variability of activity durations is reduced by risk pooling. Therefore by maximizing this measure, we anticipate that the expected project completion time is minimized and (1.0) is maximized.

3.2. Experimental analysis

We use *Monte Carlo Simulation* to generate a random set of realizations of activity durations to test robustness measures. Given a baseline schedule for a benchmark project and realizations of activity durations, we evaluate the effect of disruptions by using some performance measures. These measures are functions of the difference between the given project deadline and the realized completion time.

Having simulated the projects, the robustness measure that has the highest correlation with some performance measure (*PM*) is selected as the best metric to represent robustness. In our implementation, we use two performance measures:

1. PM_1: The proportion of replicates in which the project ends before the deadline.
2. PM_2: Average delay in the project completion time as percentage of the project deadline (i.e., $100\frac{C_{n+1}-\delta}{\delta}$ for delayed projects). For each problem instance, we generate 21 schedules using three scheduling policies:
   (1) *Optimal DTCTP-D:* An initial baseline schedule is generated assuming no disruptions and solving the DTCTP-D exactly. We solve small- and medium-sized DTCTP-D and DTCTP-B instances, which have less than 100 activities, efficiently by using optimization MIP solver CPLEX 9.1. However, for optimally solving large-scale instances, we recommend to use the tailored Benders decomposition proposed by Hazir et al. (2010). After assigning the modes, corresponding earliest start schedule (ESS) is found by using CPM calculations.
   (2) *Project buffer insertion:* Buffer insertion policy assumes a smaller deadline, i.e., $\delta' = \delta(1-\tau)$, $0 < \tau < 1$. In this way, the solution of the DTCTP-D with a smaller deadline, δ', inserts a project buffer at the end of the schedule. The project buffer size is proportional to the project deadline. Modes are assigned to activities by solving the DTCTP-D exactly with δ' and then corresponding ESS is determined. We use $\tau \in \{0.01, 0.02, \ldots, 0.10\}$ in our experiments.
   (3) *Safety time insertion:* This policy introduces a safety time proportional to nominal durations, i.e., $p'_i = p_i(1 + \Delta)$, $\forall i \in N$. Thus, activity durations are augmented and modes are assigned to the activities by solving the DTCTP-D exactly, assuming the augmented durations, p'_i. However, the so-called roadrunner mentality is used to generate the schedule, i.e., the non-gating tasks (activities with non-dummy predecessors) are started as soon as possible and safety times are ignored in this schedule. We use $\Delta \in \{0.01, 0.02, \ldots, 0.10\}$ in our experiments.

To model the activity durations in the simulation, we use a lognormal distribution with mean equal to the baseline duration and $CV = 0.5$. Several other project scheduling studies also suggest the use of this distribution (see Tavares et al. (1998), Herroelen and Leus (2001) and Tukel et al. (2006)). We refer the readers to Tavares et al. (1998) for a discussion of the advantages of lognormal distribution. To determine the number of replications required, we use the sequential procedure proposed by Law and Kelton (2000, Chapter 9). This procedure inserts new replications one by one and determines the length of the simulation so that 95% confidence level for the mean of the performance measures is constructed. In this procedure, a relative error of 5%, which specifies a bound on the percentage error of the point estimate of the sample means, is used. The following algorithm is used to test the robustness measures using simulation.

1. *Schedule generation*: Given the Scheduling Policy (SP), generate an initial baseline schedule. Then, calculate RM_i ($i = 1, \ldots, 9$) of each schedule.
2. *Monte Carlo Simulation*:
   a. Generate the processing time of each activity using the activity time distribution (in other words, the activity durations are perturbed while executing the schedule in the simulation run).
   b. Generate the early start schedule by using the randomly generated durations and classical CPM. Compute and record the project completion time.
   c. Repeat *Step 2 Nr* times (*Nr*: number of replications).
3. *Correlation computation*: Calculate the PM_j, ($j = 1, 2$). Find out and report the correlation between RM_i and PM_j ($i = 1, \ldots, 9; j = 1, 2$).

3.3. Computational results

We use a subset of the random instances generated by Akkan et al. (2005) to test the proposed measures and algorithms. Mainly three factors determine the difficulty to solve a particular problem instance: network structure, the number of modes per activity and the tightness of the deadline. Similarly, two parameters define the network structure: complexity index (CI) and the coefficient of network complexity (CNC). CI is a measure developed by Bein et al. (1992) to assess how far the given network is from being series-parallel. It is defined to be the minimum number of node reductions required to reduce a given two terminal directed acyclic graph into a single-arc graph, when used together with series and parallel reductions. The second complexity measure, CNC is developed by Pascoe (1966) and defined to be the ratio of the number of arcs to the number of nodes.

The number of modes per activity is randomly generated with discrete uniform distribution using interval $U[2, 10]$. To compute the deadline for each instance, the minimum possible project duration, T_{min} (length of the critical path with shortest modes), and the maximum possible project duration, T_{max} (length of the critical path with longest modes), are first calculated. Then, the deadline is set as follows:

$$\delta = T_{min} + \theta(T_{max} - T_{min}) \quad \text{with } 0 < \theta < 1. \tag{9}$$

To generate the cost figures, three cost functions are used: Concave (ccv), convex (cvx), and neither concave nor convex cost functions (hyb). Table 1 summarizes the parameters of the test beds.

CNC and CI are the determinants of the number of activities in a project. This experimental setting involves projects with 85–136 activities. In the computational study, for each setting three different instances are solved; hence, each measure is tested with 36 problems. Since we do not observe a significant effect of CI on computational effort in our experiments, we set CI = 13. This is a result that supports Akkan et al.'s experimental finding for approximate solutions of DTCTP-D. Furthermore, the *hardest* problem instances correspond to $\theta = 0.15$; we use this value in our experiments.

To evaluate the relationship between *RM* and *PM*, we report the coefficient of determination (R^2) and the significance levels. Table 2 illustrates the average of R^2 over all problem instances that have the same network complexity figures. Looking at Table 2, we see that the buffer size (RM_9) has the largest R^2 value regardless of the network complexity. We also illustrate the high correlation between RM_9 and PM_1 in Fig. 1. Furthermore, measures RM_1, RM_2, RM_3, RM_5, and RM_8 also have high correlations with the *PM*. For all the measures, the R^2 is found to be insensitive to the changes in CNC. Also, the proposed transformation of the slack utility function dramatically increases the correlations (RM_5 vs RM_4).

Table 1
Experimental setting.

Parameters	Level(s)
CI	13
CNC	5, 6, 7, 8
Number of modes	$U[2,10]$
Deadline parameter (θ)	0.15
Cost function (CF)	ccv, cvx, hyb

Table 2
Correlation between robustness and performance measures (R^2).

|	CNC = 5		CNC = 6		CNC = 7		CNC = 8	
	PM_1	PM_2	PM_1	PM_2	PM_1	PM_2	PM_1	PM_2
RM_1	0.9104	0.8717	0.9486	0.9301	0.9611	0.9446	0.9812	0.9579
RM_2	0.9468	0.9191	0.9681	0.9503	0.9542	0.9432	0.9778	0.9634
RM_3	0.9538	0.9251	0.9701	0.9552	0.9582	0.9460	0.9754	0.9627
RM_4	0.2239	0.2541	0.1928	0.2194	0.2318	0.2454	0.2284	0.2537
RM_5	0.8529	0.8376	0.9349	0.9239	0.9131	0.8972	0.9402	0.9299
RM_6	0.6411	0.6626	0.5867	0.6270	0.5347	0.5378	0.5814	0.5364
RM_7	0.3789	0.3803	0.4016	0.3952	0.6597	0.6610	0.6765	0.6711
RM_8	0.8524	0.8381	0.8726	0.8608	0.8253	0.8164	0.7500	0.7470
RM_9	0.9603	0.9462	0.9703	0.9612	0.9707	0.9588	0.9802	0.9653
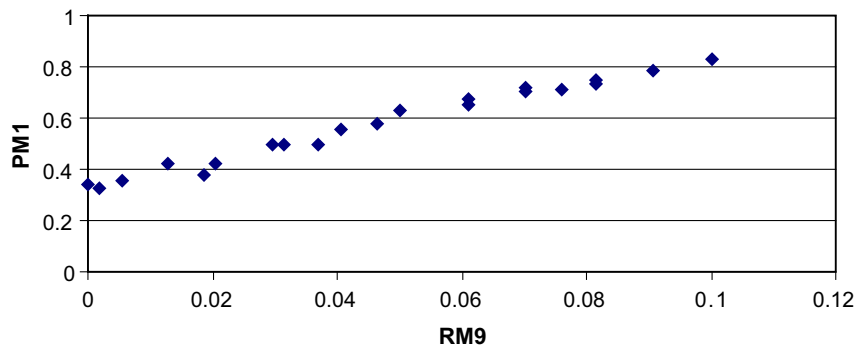
Fig. 1. Scatter plot of PM_1 and RM_9.

In addition, we compare the best four robustness measures among each other. In Table 3, we report the significance of the mean differences of the R^2 values over 36 problem instances; the t-test with 95% confidence interval, and the corresponding p values. For example, the difference between RM_9 and RM_1 is found to be significant for both performance measures. This table illustrates that RM_9 and RM_3 are good estimates of schedule robustness. They have significant differences in R^2 values when compared to the other robustness measures. To maximize robustness, both measures could be optimized either as a single criterion or multi-criteria.

Table 3
Individual 95% confidence intervals for all pairwise comparisons.

	RM_3	RM_2	RM_1
PM_1			
RM_9	$(-0.318, 1.518)$	$(-0.212, 1.946)$	$(0.385, 3.626)^*$
	0.193	0.112	0.017
RM_3		$(-0.063, 0.596)$	$(0.106, 2.705)^*$
		0.109	0.035
RM_2			$(0.015, 2.263)^*$
			0.047
PM_2			
RM_9	$(-0.074, 2.052)$	$(0.148, 2.624)^*$	$(1.455, 4.912)^*$
	0.067	0.029	0.001
RM_3		$(0.059, 0.735)^*$	$(0.820, 3.569)^*$
		0.023	0.003
RM_2			$(0.650, 2.945)^*$
			0.003

* Statistically significant difference.

4. Robust scheduling of DTCTP

Using the insight obtained by the computational experiments, we generate the baseline schedule by maximizing the project buffer size (RM_9), the robustness measure that has the highest correlation with the performance measures, so that the schedule involves sufficient safety time to absorb unanticipated disruptions. However, while maximizing robustness, the project cost should also remain within acceptable limits.

4.1. A two-phase methodology to generate a robust schedule

4.1.1. Phase 1: Exact solution of the DTCTP-D
The input is a DTCTP-D instance having a preset project deadline δ. The objective value of the optimal solution sets a threshold budget value B_0 for the next phase. This is the minimum achievable cost under the assumption that each activity lasts as it is planned and the deadline constraint is satisfied. However, this generated schedule is not protected against disruptions. In the sequel, we refer to this schedule as the *non-protected schedule* and we will use it as a benchmark.

4.1.2. Phase 2: Exact solution of the DTCTP-B
Having set the budget to B_0, a baseline schedule is generated by optimally solving the budget problem (DTCTP-B). In so doing, the algorithm aims at inserting a maximal-size project buffer while controlling the project cost. The project buffer size is maximal given the budget constraint because DTCTP-B, which minimizes the makespan while not exceeding a preset budget, is solved exactly. Furthermore, given the mode assignments, ESS maximizes the sum of total slacks (RM_1).

We observe that increases in the budget lead to increases in the project buffer. Hence in order to improve schedule robustness, we set in Phase 2 an "inflated" budget $B = (1 + \eta)B_0$ (with $\eta > 0$). However, as this policy inflates the budget, it is crucial to address the trade-off between project cost and schedule robustness. For this purpose, an analytical model to set η in a most profitable way will be introduced in Section 4.2.

Interestingly, we observe that a slight increase in the project buffer usually results in a significant improvement in the performance measures. We test the significance of the difference between the performance measures of the protected schedules with $\eta = 0.02$ and the non-protected schedules ($\eta = 0$) as well as the corresponding confidence intervals with simulation. We use for each instance, a coefficient of variation (CV) of 0.25 and 0.5, respectively, to characterize small and moderate variability in the activity durations. The confidence intervals for difference of the performance measures (*protected vs non-protected*) are reported in Table 4. We see from this table that when the proposed two-phase approach is used, it is possible to increase the probability of completing the project on-time (PM_1) significantly with small budget amplifications.

In Fig. 2, we show the relationship between budget amplification and the RM_9. This figure depicts the behavior of the buffer size as a function of the percentage increase of the budget (i.e. $100\eta\%$). The averages over the entire project instances included in the aforementioned test bed are reported in this figure. It clearly demonstrates the strong correlation between budget increase and buffer size increase.

Table 4
The significance test for the differences (protected–non-protected).

Non-protected ($\eta = 0.00$)	Protected ($\eta = 0.02$)	
	PM_1	PM_2
CV = 0.25		
PM_1	$(18.570, 24.790)^*$	
PM2		$(-0.334, -0.229)^*$
CV = 0.5		
PM_1	$(2.841, 4.609)^*$	
PM2		$(-0.972, -0.576)^*$

* Indicates that test statistic is significant at 5% level.
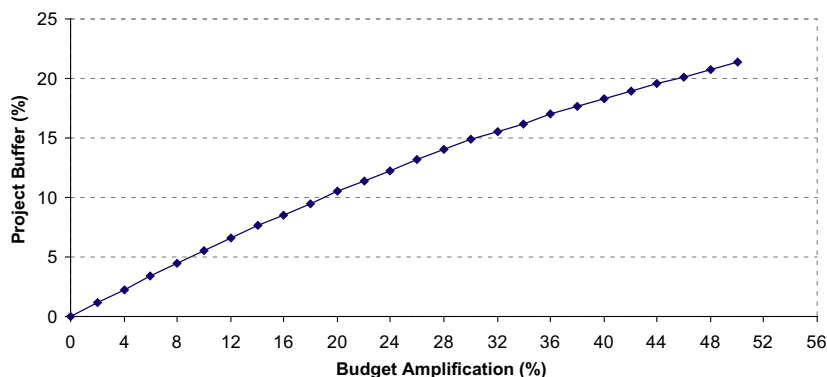

Fig. 2. Relationship between budget amplification and buffer size increase.

We carry out a simulation study to assess the effectiveness of the proposed two-phase approach as follows. First, we set the budget amplification rate η. Then, for each instance of the test bed, we randomly generate perturbed processing times and compute the project completion time. Finally, we calculate the average values of PM_1 and PM_2 over all instances. Fig. 3 illustrates the relationship between budget amplification and PM_1.

We see that when variability is low, the performance measures could be significantly improved with small budget increases. Indeed, when $CV = 0.25$, with a 6% budget increase, PM_1 increases from 44% to 89%. When variability in activity durations gets larger, much larger budget amplification is required to have the projects completed on time. We call the schedules obtained at the end of Phase 2 as *protected schedules*. Furthermore, the comparison of Figs. 2 and 3 provides evidence of the correlation between RM_9 and PM_1.

In addition to the above mentioned PM, we examine the average project completion time deviations from the project deadline as a function of η. This measure evaluates the project lateness as it considers both tardiness and earliness as well. The results are displayed in Fig. 4; when $CV = 0.25$ (0.5), a 3% (16%) budget increase is sufficient to make the expected lateness be zero.

We show that using the two-phase approach, the larger the budget is, the more robust is the derived schedule. However, a crucial issue is to decide on the budget to allocate and determine the corresponding activity modes so that an optimal trade-off between cost and robustness is achieved. In the next section, we propose a model, which is closely related to Model (1.0)–(1.4), and present the solution approach.

4.2. Analytical study of budget allocation

Model (1.0)–(1.5) assumes that the budget is given and activity durations are random. Now, we investigate a more general where the budget to be allocated (or equivalently, the increase coefficient η) is a decision variable instead of being a parameter

$$\text{Maximize } h(\eta) = E[\alpha \text{Max}\{0, \delta - C_{n+1}(\eta)\} - \beta \text{Max}\{0, C_{n+1}(\eta) - \delta\}] - B_0(1 + \eta) \tag{10}$$

Subject to 1.1,1.2,1.4,1.5 and

$$\sum_{j \in N} \sum_{m \in M_j} c_{jm} x_{jm} \leqslant B_0(1 + \eta), \tag{1.3$'$}$$

$$\eta \geqslant 0.$$

In the above model, $h(\eta)$ and $C_{n+1}(\eta)$ refer to the net expected profit and project completion time, respectively. The objective function maximizes the net profit, which is equal to the difference between the revenue and the sum of the allocated budget ($B_0(1 + \eta)$) and penalty cost. The complexity of this latter model stems from the fact that calculating the expected project completion time ($C_{n+1}(\eta)$) exactly is very
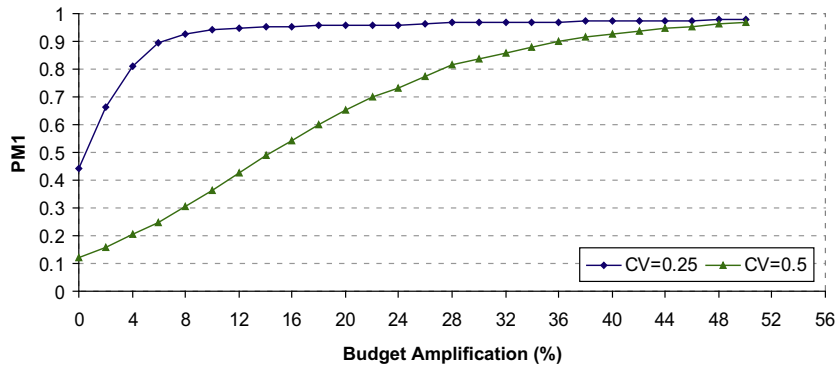

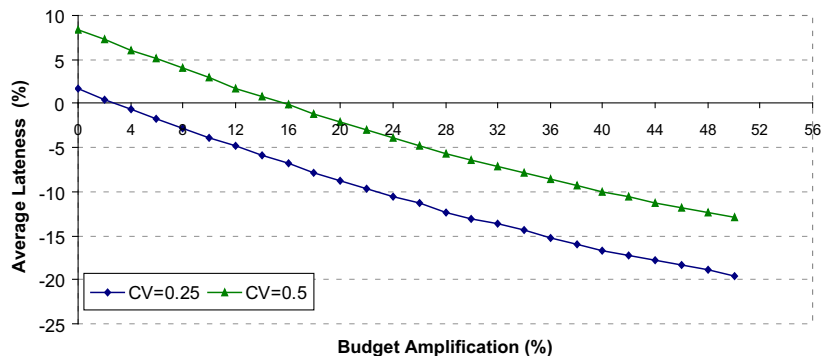**Fig. 3.** The relationship between budget amplification and PM_1.

Fig. 4. Relationship between budget amplification and average lateness.

difficult. However, a striking observation on Fig. 4 is that the average lateness (or equivalently, the average completion time) varies almost linearly with budget amplification. To evaluate this relationship, having checked the assumptions, we use a linear regression model to approximate the expected completion time. In this sequel, we make the following assumption.

A1: The expected project completion time is a *convex piecewise linear* function of the budget augmentation factor with k intervals. That is:

$$C_{n+1}(\eta) = a_0 + \sum_{j=1}^{i-1} b_{j-1}(\eta_j - \eta_{j-1}) + b_{i-1}(\eta - \eta_{i-1}) \quad \text{if } \eta_{i-1} \leqslant \eta \leqslant \eta_i < \eta_k, \quad i = 1,\ldots,k-1, \tag{11}$$

where $\eta_0 = 0$, a_i and b_i represent the intercepts and slopes of linear segments.

In practice, two to four segments would be sufficient to have a reasonable approximation of this time/cost relationship. Clearly, more linear segments could be used to yield more accurate approximation, however, it requires more computational effort as defining a segment requires simulating the project. The second assumption (A2) expresses that the budget amplifications have diminishing rate of return. Hence, as the budget increases, the marginal reduction in the project completion decreases.

A2: The slopes are assumed to be negative and increasing with respect to budget increases, i.e. $b_i < b_{i+1} < 0$, $i = 1,\ldots,k$.

The following proposition defines the structure of the optimal solution.

Proposition 1. *Under assumptions A1 and A2, the optimal budget amplification policy is either one of the breakpoints of the piecewise linear function or the critical point where the expected completion time is zero.*

Proof. See Appendix A. □

As an immediate result of Proposition 1, we propose the following algorithm to allocate the budget.

Algorithm

1. Initialization:
   (a) Define the minimum and maximum budgets, B_0, and B_{max}. These are the costs so that activities are performed in the least and the most costly modes, respectively.
   Define k intervals $[\eta_{i-1}, \eta_i]$ $(i = 1,\ldots,k)$, where $\eta_k = (B_{max} - B_0)/B_0$, $\eta_i = \frac{i}{k}\eta_k$ $(i = 1,\ldots,k)$ and $\eta_0 = 0$.
   (b) Given $B = B_0$, generate a schedule by solving the DTCTP-B exactly.
   (c) Invoke Monte Carlo Simulation (the activity durations are perturbed while executing the schedule in the simulation run), and find out the expected completion time, set $a_0 = C_{n+1}(0)$.
2. For $i = 1,\ldots,k$
   (a) Set the budget, $B = (1 + \eta_i)B_0$.
   (b) Given B, generate the schedule by solving the DTCTP-B exactly.
   (c) Invoke Monte Carlo Simulation and find the expected completion time, $C_{n+1}(\eta_i)$ Then, set $b_{i-1} = \frac{C_{n+1}(\eta_i) - C_{n+1}(\eta_{i-1})}{\eta_i - \eta_{i-1}}$, and $a_i = a_{i-1} + b_{i-1}(\eta_i - \eta_{i-1})$.
3. Having defined the parameters α, β, B_0, a_i, b_i, $i = 0,\ldots,k$, use Table 5 in the Appendix to find out the optimal policy.

The major advantage of the analytical model is that the computational effort to find the expected completion time is significantly reduced. We show that at most $k + 1$ budget settings, which are the breakpoints of the piecewise linear function, should be considered.

5. Conclusion

To address the crucial need to build robust project schedules that are less vulnerable to disruptions caused by uncontrollable factors, we have investigated the robust scheduling of a variant of the multi-mode discrete time/cost trade-off project scheduling problem. In this variant, the problem is to select a mode for each activity so that the project is completed within a preset deadline and the total cost is minimized. We describe and analyze the pertinence of several robustness measures. We provide empirical evidence that the project buffer size is the more appropriate robustness measure regardless of the network complexity. Based on this finding, we propose a two-step methodology for generating robust schedules. In the first phase of the methodology, we determine the minimum required budget. Next, in the second phase, this budget is slightly inflated by a specified amplification factor and then the buffer size is maximized. We provide strong empirical evidence that the budget amplification consistently improves the schedule robustness. Therefore, we address the important issue of determining the best trade-off between project cost and schedule robustness. To that aim, we propose an extended model involving both tardiness penalties and earliness revenues and we describe an appropriate solution strategy that requires a restricted number of simulations.

A promising research focus could be deriving robust schedules for the multi-mode resource constrained project scheduling problem. This problem might be viewed as a natural combination of the well-known resource constrained project scheduling problem and the time/cost trade-off problem. We expect that the results presented in this paper could provide a useful base for investigating this challenging problem.

Appendix 1

Proof of Proposition 1. The proof rests on investigating all the possible parameter settings and finding out the optimal budget allocation for each setting. Given a budget allocation, the project could be either early or late and each case is considered separately due to the structure of (11).

In order to make a distinction between these two cases, we find out the critical budget factor, η_c, so that $C_{n+1}(\eta_c) = \delta$. If it is not possible to complete on time with the maximum budget, i.e. $C_{n+1}(\eta_k)a_o + \sum_{j=1}^{k} b_{j-1}(\eta_j - \eta_{j-1}) < \delta$, then we set $\eta_c = \eta_k$. On the other hand, if the project is expected to be early without any budget increase i.e. $a_0 \leqslant \delta$, we set $\eta_c = 0$.

The project is expected to be late when $0 \leqslant \eta \leqslant \eta_c$. Hence, in this case

$$
\begin{aligned}
h(\eta) &= -B_0(1+\eta) - \beta\left(a_o + \sum_{j=1}^{i-1} b_{j-1}(\eta_j - \eta_{j-1}) + b_{i-1}(\eta - \eta_{i-1}) - \delta\right) \\
&= \beta\delta - \beta\left(a_o + \sum_{j=1}^{i-1} b_{j-1}(\eta_j - \eta_{j-1}) - b_{i-1}\eta_{i-1}\right) - B_0 - (B_0 + \beta b_{i-1})\eta.
\end{aligned}
\tag{12}
$$

On the other hand, if $0 < \eta_c < \eta$ then project is expected to be early. Thus,

$$
\begin{aligned}
h(\eta) &= \alpha\left(\delta - a_o - \sum_{j=1}^{i-1} b_{j-1}(\eta_j - \eta_{j-1}) - b_{i-1}(\eta - \eta_{i-1})\right) - B_0(1+\eta) \\
&= \alpha\delta - \alpha\left(a_o + \sum_{j=1}^{i-1} b_{j-1}(\eta_j - \eta_{j-1}) - b_{i-1}\eta_{i-1}\right) - B_0 - (B_0 + \alpha b_{i-1})\eta.
\end{aligned}
\tag{13}
$$

Two additional parameters are required to be defined: $d_1 = \text{Max}\{i : \eta_i < \eta_c, i = 0,\ldots,k\}$, $d_2 = \text{Min}\{i : \eta_{i+1} > \eta_c, i = 0,\ldots,k\}$. Three cases each containing three sub-cases could be identified such that each condition has specific optimality conditions:

Case 1: $\beta \leqslant \frac{-B_0}{b_0}$:

In this case, $\beta \leqslant \frac{-B_0}{b_i}$ $\forall i = 1,\ldots,k-1$, due to A2.

1a. $\alpha \leqslant \frac{-B_0}{b_{d_2}}$: due to A2, $\alpha \leqslant \frac{-B_0}{b_i}$ $\forall i = d_2,\ldots,k-1$.

In this case that no budget augmentation improves the profit. The optimal policy is $\eta^* = 0$.

1b. $\frac{-B_0}{b_{d_2}} < \alpha \leqslant \frac{-B_0}{b_{k-1}}$,

Define $f = \text{Min}\left\{i : \alpha < \frac{-B_0}{b_i}, \forall i = d_2+1,\ldots,k-1\right\}$. The optimal policy is $\eta^* = 0$ with profit: $h(0) = -\beta(a_o - \delta) - B_0$ or $\eta^* = \eta_f$ with $h(\eta_f) = \alpha\left(\delta - a_o - \sum_{j=1}^{f} b_{j-1}(\eta_j - \eta_{j-1})\right) - B_0(1+\eta_f)$.

The profits are compared and the comparison results in:

$$
\eta^* = \begin{cases} 0 & \text{if } (a_o - \delta)(\beta - \alpha) - \alpha\sum_{j=1}^{f} b_{j-1}(\eta_j - \eta_{j-1}) - B_0\eta_f \leqslant 0, \\ \overline{\eta_f} & o/w. \end{cases}
$$

1c. $\frac{-B_0}{b_{k-1}} < \alpha$: due to A2, $\frac{-B_0}{b_i} < \alpha$ $\forall i = d_2,\ldots,k-1$.

In this case, an optimal policy is either to set $\eta^* = 0$ with profit $h(0) = -\beta(a_o - \delta) - B_0$ or $\eta^* = \eta_k$ with profit $h(\eta_k) = \alpha(\delta - a_o - \sum_{j=1}^{k} b_{j-1}(\eta_j - \eta_{j-1})) - B_0(1+\eta_k)$. Comparison results in:

$$
\eta^* = \begin{cases} 0 & \text{if } (a_o - \delta)(\beta - \alpha) - \alpha\sum_{j=1}^{k} b_{j-1}(\eta_j - \eta_{j-1}) - B_0\eta_k \leqslant 0, \\ \overline{\eta_k} & o/w. \end{cases}
$$

Case 2. $\frac{-B_0}{b_0} < \beta \leqslant \frac{-B_0}{b_{d_1}}$

Define $e = \text{Min}\left\{i : \beta < \frac{-B_0}{b_i}, i = 1,\ldots,d_1\right\}$.

2a. $\alpha \leqslant \frac{-B_0}{b_{d_2}}$, $\eta^* = \eta_e$

2b. $\frac{-B_0}{b_{d_2}} < \alpha \leqslant \frac{-B_0}{b_{k-1}}$:

Compare $h(\eta_e) = -\beta(a_o + \sum_{j=1}^{e} b_{j-1}(\eta_j - \eta_{j-1}) - \delta) - B_0(1+\eta_e)$ with $h(\eta_f) = \alpha(\delta - a_o - \sum_{j=1}^{f} b_{j-1}(\eta_j - \eta_{j-1})) - B_0(1+\eta_f)$. Comparison results in

$$
\eta^* = \begin{cases} \eta_e & \text{if } (a_o - \delta)(\beta - \alpha) - \alpha\sum_{j=1}^{f} b_{j-1}(\eta_j - \eta_{j-1}) + \beta\sum_{j=1}^{e} b_{j-1}(\eta_j - \eta_{j-1}) - B_0(\eta_f - \eta_e) \leqslant 0, \\ \eta_f & o/w. \end{cases}
$$

2c. $\frac{-B_0}{b_{k-1}} < \alpha$

Now compare $h(\eta_e) = -\beta(a_o + \sum_{j=1}^{e} b_{j-1}(\eta_j - \eta_{j-1}) - \delta) - B_0(1+\eta_e)$ with $h(\eta_k) = \alpha(\delta - a_o - \sum_{j=1}^{k} b_{j-1}(\eta_j - \eta_{j-1})) - B_0(1+\eta_k)$. The optimal policy is:

Table 5
Budget amplification policies.

β	α		
	$\left[0, -\frac{B_0}{b_{d_2}}\right]$	$\left(-\frac{B_0}{b_{d_2}}, -\frac{B_0}{b_{k-1}}\right]$	$\left(-\frac{B_0}{b_{k-1}}, \infty\right)$
$\left[0, -\frac{B_0}{b_0}\right]$	$\eta^* = 0$	$\eta^* \in \{0, \eta_f\}$	$\eta^* \in \{0, \eta_k\}$
$\left(-\frac{B_0}{b_0}, -\frac{B_0}{b_{d_1}}\right]$	$\eta^* = \eta_e$	$\eta^* \in \{\eta_e, \eta_f\}$	$\eta^* \in \{\eta_e, \eta_f\}$
$\left(-\frac{B_0}{b_{d_1}}, \infty\right)$	$\eta^* = \eta_c$	$\eta^* = \eta_f$	$\eta^* = \eta_k$

$$\eta^* = \begin{cases} \eta_e & \text{if } (a_o - \delta)(\beta - \alpha) - \alpha \sum_{j=1}^{k} b_{j-1}(\eta_j - \eta_{j-1}) + \beta \sum_{j=1}^{e} b_{j-1}(\eta_j - \eta_{j-1}) - B_0(\eta_k - \eta_e) \leqslant 0, \\ \eta_k & \text{o/w.} \end{cases}$$

Case 3. $\frac{-B_0}{b_{d_1}} < \beta$: due to A2, $\beta \geqslant \frac{-B_0}{b_i}$ $\quad \forall i = 0, \ldots, d_1 - 1$.

3a. $\alpha \leqslant \frac{-B_0}{b_{d_2}}$, the optimal policy is $\eta^* = \eta_c$.

3b. $\frac{-B_0}{b_{d_2}} < \alpha \leqslant \frac{-B_0}{b_{k-1}}$, the optimal policy is $\eta^* = \eta_f$.

3c. $\alpha \geqslant \frac{-B_0}{b_{k-1}}$, the optimal policy is $\eta^* = \eta_k$.

Table 5 summarizes the different cases and the corresponding optimal budget augmentation policies. Considering all the possibilities summarized in Table 5, the optimal budget amplification policy is defined with either one of the breakpoints or the critical point which makes the expected completion time zero. □

References

Akkan, C., Drexl, A., Kimms, A., 2005. Network decomposition-based benchmark results for the discrete time–cost trade-off problem. European Journal of Operational Research 165, 339–358.

Al-Fawzan, M.A., Haouari, M., 2005. A bi-objective model for robust resource-constrained project scheduling. International Journal of Production Economics 96, 175–187.

Bein, W.W., Kamburowski, J., Stallmann, M.F.M., 1992. Optimal reduction of two-terminal directed acyclic graphs. SIAM Journal on Computing 21, 1112–1129.

Chtourou, H., Haouari, M., 2008. A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling. Computers and Industrial Engineering 55, 183–194.

De, P., Dunne, E.J., Ghosh, J.B., Wells, C.E., 1995. The discrete time/cost trade-off problem revisited. European Journal of Operational Research 81, 225–238.

De, P., Dunne, E.J., Ghosh, J.B., Wells, C.E., 1997. Complexity of the discrete time/cost trade-off problem for project networks. Operations Research 45, 302–306.

Demeulemeester, E., Herroelen, W., Elmaghraby, S.E., 1996. Optimal procedures for the discrete time/cost trade-off problem in project networks. European Journal of Operational Research 88, 50–68.

Demeulemeester, E., De Reyck, B., Foubert, B., Herroelen, W., Vanhoucke, M., 1998. New computational results for the discrete time/cost trade-off problem in project networks. Journal of the Operational Research Society 49, 1153–1163.

Erengüç, S.S., Tufekci, S., Zappe, C.J., 1993. Solving time/cost trade-off problems with discounted cash flows using generalized benders decomposition. Naval Research Logistics Quarterly 40, 25–50.

Goldratt, E.M., 1997. Critical Chain. The North River Press Publishing Corporation, Great Barrington, MA.

Gören, S., Sabuncuoğlu, I., 2008. Robustness and stability measures for scheduling: Single machine environment. IIE Transactions 40, 66–83.

Hafızoğlu, A.B., Azizoğlu, M., 2010. Linear programming based approaches for the discrete time/cost trade-off problem in project networks. Journal of the Operational Research Society 61 (4), 676–685.

Hazir, Ö, Haouari, M., Erel, E., 2010. Discrete time/cost trade-off problem: A decomposition based solution algorithm for the budget version. Computers and Operations Research 37 (4), 649–655.

Herroelen, W., Leus, R., 2001. On the merits and pitfalls of critical chain scheduling. Journal of Operations Management 19, 559–577.

Herroelen, W., Leus, R., 2005. Project scheduling under uncertainty – Survey and research potentials. European Journal of Operational Research 165, 289–306.

Klerides, E., Hadjiconstantinou, E., 2010. A decomposition-based stochastic programming approach for the project scheduling problem under time/cost trade-off settings and uncertain durations. Computers and Operations Research 37 (12), 2131–2140.

Kobylański, P., Kuchta, D., 2007. A note on the paper by M.A. Al-Fawzan and M. Haouari about a bi-objective problem for robust resource-constrained project scheduling. International Journal of Production Economics 107, 496–501.

Lambrechts, O., Demeulemeester, E., Herroelen, W., 2008. A tabu search procedure for developing robust predictive project schedules. International Journal of Production Economics 111, 493–508.

Law, A.M., Kelton, W.D., 2000. Simulation Modeling and Analysis, third ed. McGraw-Hill, New York.

Leon, V.J., Wu, S.D., Storer, R.H., 1994. Robustness measures and robust scheduling for job shops. IIE Transactions 26, 32–43.

Mehta, S.V., Uzsoy, R., 1998. Predictable scheduling of a job shop subject to breakdowns. IEEE Transactions on Robotics and Automation 14 (3), 365–378.

Pascoe, T.L., 1966. Allocation of resources – CPM. Revue Française de Recherche Opérationelle 38, 31–38.

Skutella, M., 1998. Approximation algorithms for the discrete time–cost trade-off problem. Mathematics of Operations Research 23, 195–203.

Tavares, L.V., Ferreira, J.A.A., Coelho, J.S., 1998. On the optimal management of project risk. European Journal of Operational Research 107, 451–469.

Tukel, O.I., Rom, O.W., Ekşioğlu, S.D., 2006. An investigation of buffer sizing techniques in critical chain scheduling. European Journal of Operational Research 172, 401–416.

Van de Vonder, S., Demeulemeester, E., Herroelen, W., Leus, R., 2005. The use of buffers in project management: The trade-off between stability and makespan. International Journal of Production Economics 97, 227–240.

Van de Vonder, S., Demeulemeester, E., Herroelen, W., Leus, R., 2006. The trade-off between stability and makespan in resource-constrained project scheduling. International Journal of Production Research 44, 215–236.

Van de Vonder, S., Demeulemeester, E., Herroelen, W., 2008. Proactive heuristic procedures for robust project scheduling: An experimental analysis. European Journal of Operational Research 189 (3), 723–733.

Vanhoucke, M., Debels, D., 2007. The discrete time/cost trade-off problem under various assumptions exact and heuristic procedures. Journal of Scheduling 10, 311–326.

Weglarz, J., Jozefowska, J., Mika, M., Waligora, G., 2010. Project scheduling with finite or infinite number of activity processing modes – A survey. European Journal of Operational Research. doi:10.1016/j.ejor.2010.03.037.