



Robust optimization models for the discrete time/cost trade-off problem

Öncü Hazır^{a,b,*}, Erdal Erel^a, Yavuz Günalay^c

^a Faculty of Business Administration, Bilkent University, 06800 Ankara, Turkey

^b Industrial Engineering Department, Çankaya University, 06530 Ankara, Turkey

^c Faculty of Economics and Administrative Sciences, Bahçeşehir University, 34353 Beşiktaş, İstanbul, Turkey

ARTICLE INFO

Article history:

Received 12 November 2009

Accepted 11 November 2010

Available online 24 November 2010

Keywords:

Project scheduling

Robust Optimization

Benders decomposition

Tabu search

ABSTRACT

Developing models and algorithms to generate robust project schedules that are less sensitive to disturbances are essential in today's highly competitive uncertain project environments. This paper addresses robust scheduling in project environments; specifically, we address the discrete time/cost trade-off problem (DTCTP). We formulate the robust DTCTP with three alternative optimization models in which interval uncertainty is assumed for the unknown cost parameters. We develop exact and heuristic algorithms to solve these robust optimization models. Furthermore, we compare the schedules that have been generated with these models on the basis of schedule robustness.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

In project management it is often possible, with some additional costs, to reduce the duration of some activities and thereby expedite the project completion. In this paper, we consider discrete time/cost relationships and address the discrete time/cost trade-off problem (DTCTP), which is a well-known multi-mode project scheduling problem with practical relevance. Two main versions of the problem have been studied in the literature; namely, the deadline problem (DTCTP-D), the budget problem (DTCTP-B). In DTCTP-D, given a set of time/cost pairs (modes) and a project deadline of δ , each activity is assigned to one of the possible modes so that the total cost is minimized. The budget problem minimizes the project duration while meeting a given budget, B . We address the deadline version and formally define as

Given a project with a set of n activities along with the corresponding precedence graph in the AoN (activity-on-node) representation, $G=(N, A)$, where N is the set of nodes, which includes n activities and two dummy nodes, 0 and $n+1$, that are used to indicate the project start and completion instants. $A \subseteq N \times N$ is the set of arcs, which represents the immediate precedence constraints among activities. Each activity j can be performed at one of the $|M_j|$ modes where each mode $m \in M_j$, is characterized by a processing time p_{jm} and a cost c_{jm} . A mixed integer-programming

model of the DTCTP-D can be stated as follows:

$$\text{Min} \sum_{j=1}^n \sum_{m \in M_j} c_{jm} x_{jm} \quad (1.0)$$

Subject to

$$\sum_{m \in M_j} x_{jm} = 1, \quad j = 1, \dots, n \quad (1.1)$$

$$C_j - C_i - \sum_{m \in M_j} p_{jm} x_{jm} \geq 0 \quad \forall (i, j) \in A \quad (1.2)$$

$$C_{n+1} \leq \delta \quad (1.3)$$

$$C_j \geq 0 \quad \forall j \in N \quad (1.4)$$

$$x_{jm} \in \{0, 1\} \quad \forall m \in M_j, \quad j = 1, \dots, n \quad (1.5)$$

C_j is the continuous decision variable that represents the completion time of activity j . The binary decision variable x_{jm} assigns a mode $m \in M_j$ to activity j (1.5). Total cost is minimized (1.0), while a unique mode should be assigned to each activity (1.1); precedence constraints should not be violated (1.2); and the deadline should be met (1.3).

De et al. (1997) have shown that the DTCTP is strongly NP-hard. In their survey paper (1995), they review the problem characteristics, as well as exact and approximate solution strategies. Demeulemeester et al. (1996, 1998) propose branch-and-bound to solve the problem exactly, Akkan et al. (2005), and Vanhoucke and Debels (2007) propose approximate algorithms. Additionally, Erenguc et al. (1993) use Benders decomposition to solve the DTCTP with discounted cash flows and Lova et al. (2009) propose a hybrid genetic algorithm to solve the resource constrained case.

* Corresponding author at: Industrial Engineering Department, Çankaya University, 06530 Ankara, Turkey. Tel.: +90 312 2844500 4013; fax: +90 312 2848043.

E-mail addresses: oncuh@bilkent.edu.tr, hazir@cankaya.edu.tr (Ö. Hazır), erel@bilkent.edu.tr (E. Erel), yavuz.gunalay@bahcesehir.edu.tr (Y. Günalay).

All of the studies cited above assume complete information and a deterministic environment; however, projects are often subject to various sources of uncertainty that threaten the accomplishment of project objectives. Therefore it is vital to develop effective robust scheduling algorithms. To minimize the effect of unexpected events on project performance, five fundamental scheduling approaches have been used: stochastic scheduling, fuzzy scheduling, sensitivity analysis, reactive scheduling, and robust (proactive) scheduling (Herroelen and Leus, 2005). In stochastic project scheduling, the activity durations are modeled as random variables and probability distributions are used. Fuzzy project scheduling uses fuzzy membership functions to model activity durations. The effects of parameter changes are investigated in sensitivity analysis. In reactive scheduling, the schedule is modified when a disruption occurs, whereas in robust scheduling anticipation of variability is incorporated into the schedule and schedules that are insensitive to disruptions are generated.

Herroelen and Leus (2005) divide schedule robustness into two groups: solution robustness (stability) and quality robustness. The *solution robustness* is defined as the insensitivity of the activity start times with respect to variations in the input data. On the other hand, *quality robustness* is defined as insensitivity of schedule performance such as project makespan 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 by disruptions. In this research, we address on quality robust project scheduling.

To construct solution robust project schedules, Herroelen and Leus (2003) propose some mathematical programming models. They develop an LP model and propose heuristics for the solution of robust scheduling. Their LP model allows a single activity disruption which increases the duration of one activity during the schedule execution. In addition, Van De Vonder et al. (2008) propose heuristics for solution robust scheduling and compare the performance of proposed heuristics using simulation. On the other hand, Lambrechts et al. (2008a) investigate the uncertainty in resource availabilities that may be caused by reasons such as machine failures, and they combine a proactive scheduling procedure with a reactive improvement procedure. Recently, Al-Fawzan and Haouari (2005), Chtourou and Haouari (2008), Kobylanski and Kuchta (2007), and Lambrechts et al. (2008b) have proposed predictive robustness measures for resource constrained networks.

In a different vein from the studies cited above, we assume interval uncertainty and make use of robust optimization to generate robust project schedules. *Robust optimization* is a modeling approach to generate a plan that performs well even in the worst-case scenarios. Robust optimization has been applied to some combinatorial optimization problems such as the shortest path problem during the last decade (Bertsimas and Sim, 2003). However, it has been implemented in only a few project scheduling problems as discussed below.

Valls et al. (1998) examine a special resource constrained project scheduling problem (RCPS) in which the activities might be interrupted for an uncertain period. Yamashita et al. (2007) address the resource availability cost problem (RACP). They propose two alternative models: the first model minimizes the maximum regret function, whereas the second one is a mean risk model that minimizes weighted sum of the mean and variance of the costs. Both Valls et al. (1998) and Yamashita et al. (2007) follow a scenario-based approach, where a scenario represents a realization of the duration of the activities. Alternatively, Cohen et al. (2007) use interval uncertainty in their recent robust scheduling study that addresses the effects of uncertainty on the continuous time-cost trade-off problem. They model the robust problem using the ARC methodology of Ben-Tal et al. (2004); some of the variables are determined before the realization of the uncertain parameters

(non-adjustable variables), while the other variables could be determined after the realization (adjustable variables).

We propose three robust optimization models in which uncertainty is modeled via intervals for the DTCTP-D. Our research differs from the previous studies in the literature regarding both the problem addressed and uncertainty modeling approach followed. Our models address the uncertainty in activity costs. In practice, fluctuations in the exchange rates, factor prices or resource usages result in cost variability. These fluctuations threaten the accomplishment of project cost objectives and it is essential to develop systematic methods to generate robust project schedules, which are less sensitive to uncertainty. We develop exact and heuristic algorithms to solve these robust models and compare the schedules that have been generated with these models on the basis of schedule robustness. The main contribution is the incorporation of uncertainty into a practically relevant project scheduling problem and the development of problem specific solution approaches.

In the next section, we formulate the robust DTCTP-D using alternative robust optimization models. We propose exact and heuristic algorithms to solve these robust models in Section 3. Finally in Section 4, we present some computational results and compare the robustness of the schedules generated with these models using some robustness metrics.

2. Robust discrete time/cost trade-off problem with interval data

In many real-life projects a tardiness penalty or an opportunity cost is incurred for each additional time unit the project is late. The cost includes explicit monetary charges, foregone revenue, lost profits, or goodwill losses. Due to these potential costs and early completion benefits, organizations seek on-time completion and aggressively monitor actual progress of these activities. The model proposed in this section addresses project environments in which timely completion of project activities is crucial. A frequently encountered practice that favors early completions is Build-Operate-Transfer (BOT) projects.

BOT model has been widely accepted in both developed and developing countries as it functions as an alternative financing mechanism in undertaking large investment projects.

In BOT projects, a public service or an infrastructure investment is performed and operated for a specific period by a private enterprise, and then transfers the right to operate to a public authority and therefore increase its profit. The operating period is usually long, often more than 10 years, so that the investment could be paid off. To give an example, a firm constructs a private toll road, and operates it for some time and then transfers the right to operate to public. If the firm could complete construction earlier, it can extend the operating period and therefore increase the profit. Whereas, if a delay occurs in the project completion time, the firm both pays a penalty cost to the regulatory authority and also loses its potential revenues for the delay period. Therefore the disadvantages of being late are usually much more than the advantages of early completion.

When deviations from the baseline plan are observed and are judged to threaten the completion of these activities on time, project managers usually allocate extra resources such as additional workers or extra machinery to these activities. These additional allocations create fluctuations in the amount of resources allocated to each activity and result in cost uncertainty. In addition, fluctuations in the exchange rates and factor prices may also cause uncertainty in costs. All these factors seriously affect the profitability of the projects. From this point of view, protection

against deviations in total cost becomes the key concern of project managers.

We focus on cost uncertainty in this research. We assume that each mode is associated with a fixed duration p_{jm} and an interval of possible costs $[\underline{c}_{jm}, \overline{c}_{jm}]$, $\underline{c}_{jm} \leq c_{jm} \leq \overline{c}_{jm}$ (in which j indexes the activities and m indexes the alternative durations, and equality between the lower and upper bounds on cost indicates certainty at a fixed value $\underline{c}_{jm} = c_{jm} = \overline{c}_{jm}$).

The traditional minmax (absolute robustness) criterion focuses on the worst-case solution, which corresponds to the scenario where the cost of each activity, c_{jm} is replaced by \overline{c}_{jm} , the upper bound of the corresponding interval. Optimization with respect to absolute robustness criterion is equivalent to the classic DTCTP-D with c_{jm} set to its upper bound, \overline{c}_{jm} . However, this approach of robustness is extremely pessimistic and most likely is unrealistic.

A more realistic approach would be modeling the uncertainty only over a subset of the scenarios space. One recent application of this restriction idea is the robust discrete optimization approach proposed by Bertsimas and Sim (2003). They assume that only a subset of the uncertain parameters is allowed to deviate from their estimates; in other words, only γ of activity cost parameters (out of a total of n) involve random behavior. If $\gamma = 0$, the influence of the cost deviations is ignored and the problem with nominal cost values is obtained. In contrast, if $\gamma = n$, maximal cost deviations are considered and the problem becomes a minmax optimization problem. Therefore, γ can be regarded as a parameter that reflects the pessimism level of the decision maker. High values of this parameter indicate a risk-averse decision making behavior. This restricted uncertainty approach has recently been applied to define robust optimal policies of an integrated production planning problem (Wei et al., 2010).

2.1. Model 1

In this section, a MIP model for DTCTP-D using Bertsimas and Sim's approach is presented. We assume that at most $0 \leq \gamma \leq n$ activities have cost values at their upper bounds and the remaining $n - \gamma$ coefficients are set to the nominal values, c_{jm} . Nominal values are the most likely cost values assigned to each activity by the project manager. Nominal value of an activity lies between the upper and the lower bounds of the activity cost. We also define the maximum deviation of an activity cost as d_{jm} , i.e., $d_{jm} = \text{Max} \{ \overline{c}_{jm} - c_{jm}, c_{jm} - \underline{c}_{jm} \}$. Therefore, d_{jm} is the maximum (cost) error that the management could commit (for activity j) during the planning stage. The restricted uncertainty model, which will be called Model 1, could be expressed by the following nonlinear formulation:

$$\text{Min} \left\{ \sum_{j=1}^n \sum_{m \in M_j} c_{jm} x_{jm} + \text{Max}_{S \subset N, |S| \leq \gamma} \left\{ \sum_{j \in S} \sum_{m \in M_j} d_{jm} x_{jm} \right\} \right\} \quad (2)$$

Subject to
(1.1), (1.2), ..., (1.5)

In this model, S is a proper subset of N , such that activities in set S have cost values at their upper bounds. The cardinality of the set is bounded by the parameter γ , i.e. $|S| \leq \gamma$ and this parameter reflects the level of pessimism. Before proposing a decomposition based solution algorithm, we reformulate (2) as

$$f(\gamma) = \text{Min} \left\{ \sum_{j=1}^n \sum_{m \in M_j} c_{jm} x_{jm} + \text{Max} \left\{ \sum_{j=1}^n \sum_{m \in M_j} d_{jm} x_{jm} u_j : \sum_{j=1}^n u_j \leq \gamma, u \in B^n \right\} : x \in X^D \right\} \quad (3)$$

In this formulation, X^D denotes the feasibility set of the DTCTP-D, which is defined by constraints (1.1), (1.2), ..., (1.5). The inner

maximization problem in (3) is used to define the members of set S . When activity modes are fixed, this problem becomes a linear problem. This property will be used in the decomposition algorithm. In Eq. (3), the set of coefficients, which are subject to uncertainty, are determined by the n -dimensional binary vector u ; i.e., B^n represents the n -dimensional binary space. The indicator variable, u_j is set equal to one, if and only if the corresponding activity is assumed to deviate from its nominal value.

Model 1 assumes that all the activities are equally likely to deviate from their nominal values and we choose γ of such activities and set them at their upper bounds to generate a robust model. In real life, criticalities of the activities are crucial as well. Therefore we propose two additional novel models that integrate the criticality of project networks.

2.2. Criticality-based uncertainty models

In projects cost and time are interdependent as they both depend on the amount of the resource allocation. Activities having large slacks (i.e., non-critical activities) provide flexibility in resource allocation; hence, it is possible to delay their starting times or to elongate the durations via lowering the amount of resource allocations. Due to these flexibilities, these activities involve less risk to achieve the cost targets when compared to the critical activities. In case of disruptions, managers usually allocate more resources to critical activities or in managerial terms "crash these activities" and this incurs extra cost.

The conventional measure of an activity's criticality is the total slack, which is the amount of time by which the completion time of the activity can exceed its earliest completion time without delaying the project completion time. It is a measure of the insensitivity of schedule performance with respect to activity delays. The activities that have no slacks are defined to be critical activities.

In real-life projects, it would make much more sense to evaluate the activity slacks with respect to activity's duration, since the higher the ratio of slack to activity time the higher its capability to compensate for a delay. The reason is that as the activity durations increase the probability of a larger number of disruptions to be observed while the activity is being performed, increases. Thus, we use the slack/duration ratio to assess criticality of activities and define the activities that have slack values less than $100\xi\%$ of the activity duration as *potentially critical activities*, i.e. $CR = \{j = 1, \dots, n : TS_j/p_j \leq \xi\}$ where TS_j and p_j refer to the total slack and duration of activity j , respectively. ξ will be called slack duration threshold (SDT) hereafter. In this study, we set the SDT to 25%, i.e. $\xi = 0.25$. This new definition enlarges the criticality set.

2.2.1. Model 2

Model 1 is unrealistically pessimistic as the activity slacks are disregarded and the worst-case costs are allocated to activities with ample slacks. To eliminate this over-pessimism, the activities with cost values at the upper bounds are chosen from the critical ones in the criticality-based robust model. Given the mode assignments, only γ controls the pessimism level in model 1; however, in the new model the critical activity set and γ control the pessimism level. The following model represents the criticality-based approach.

$$f(\gamma) = \text{Min} \left\{ \sum_{j=1}^n \sum_{m \in M_j} c_{jm} x_{jm} + \text{Max} \left\{ \sum_{j=1}^n \sum_{m \in M_j} d_{jm} x_{jm} u_j : \sum_{j \in CR} u_j \leq \gamma, u \in B^n \right\} : x \in X^D \right\} \quad (4)$$

Note that the only difference between (3) and (4) is the set of activities which can have cost values at the upper bounds. CR refers to the set of potentially critical activities in (3). In this new approach, criticality definition becomes crucial. For our problem,

slacks are defined with respect to a specific mode assignment. As the mode assignments change, the slack distribution among the activities also changes. To illustrate the differences between the models, we use the simple network in Fig. 1, which is adapted from the example of De et al. (1995). The project has a deadline of $\delta=6$. Each activity has two mode alternatives characterized by the triplet, $(p_{jm}, c_{jm}, \overline{c_{jm}})$ shown above the nodes.

Table 1 depicts the objective function values of the optimistic model ($\gamma=0$) and two robust models (with $\gamma=1, 2$). The rows of the table show the feasible mode combinations for the activities. Optimal solutions are marked with “*” and given a mode combination, the critical activities are underlined. Note that activity slacks depend on mode assignments.

Table 1 illustrates that γ and γ_{CR} , which control the level of pessimism, are effective on the choice of activity modes. Comparing the models, Model 2 is less pessimistic than Model 1 as it considers less risk premium in costs. Next, we propose an alternative approach, which lies in between Models 1 and 2 regarding the level of conservatism.

2.2.2. Model 3

This model also accounts for the cost deviations in the non-critical activity set, but unlike the criticality model (Model 2), the set of possible deviations is not limited with the set of critical activities; the critical activities have priority over non-critical ones for cost deviations. Given the mode combinations, γ activities which have cost values at their upper bounds are first chosen from the critical activity set. If the cardinality of the critical activity set is less than γ , the remaining units are chosen from the non-critical activity set. As we consider the worst-case scenario, the activities

with higher deviation values are always given the priority to calculate the maximum deviation amount from the given mode combination.

3. Solution algorithms

3.1. Benders reformulation of Model 1

In this section, we show how to solve Model 1 using a Benders decomposition algorithm.

Proposition 1. Model 1 could be formulated as follows:

$$\begin{aligned} & \text{Min } \sum_{j=1}^n \sum_{m \in M_j} c_{jm} x_{jm} + z \\ & \text{Subject to} \\ & z - \sum_{j=1}^n \sum_{m \in M_j} u_j^k d_{jm} x_{jm} \geq 0, \quad k = 1, \dots, K \\ & \sum_{j=1}^n \sum_{m \in M_j} w_j^s p_{jm} x_{jm} \leq \delta, \quad s = 1, \dots, S \\ & \sum_{m \in M_j} x_{jm} = 1, \quad j = 1, \dots, n \\ & x_{jm} \in \{0, 1\} \quad \forall m \in M_j, \quad j = 1, \dots, n \\ & z \geq 0 \end{aligned} \tag{5}$$

where $u^k = (u_1^k, \dots, u_n^k)$ for $k = 1, \dots, K$ are the extreme points of the polytope $U = \{u \in R^n : \sum_{j \in N} u_j \leq \gamma, 0 \leq u_j \leq 1, j = 1, \dots, n\}$. S refers to the total number of paths between node 0 and $n+1$ in $G(N, A)$

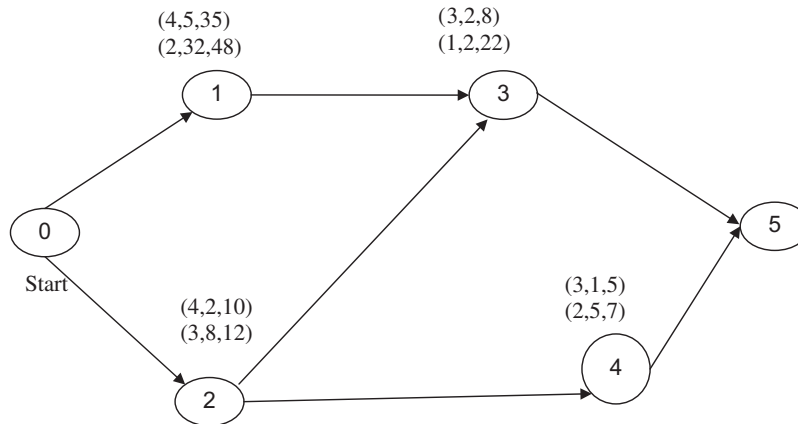


Fig. 1. The example network (robust problem).

Table 1 Comparison of robust models.

Activity (j)				C _s	f(γ=0)	d _{jm} = $\overline{c_{jm}} - c_{jm}$				Robust objective: f(γ)			
1	2	3	4			j=1	j=2	j=3	j=4	γ=1	γ _{CR} =1	γ=2	γ _{CR} =2
Feasible mode combinations													
2	2	2	2	5	68	8	2	10	1	78	68	86	68
2	2	2	1	6	65	8	<u>2</u>	10	<u>2</u>	75	67	83	69
2	2	1	2	6	61	8	<u>2</u>	<u>2</u>	1	69	63	71	65
2	2	1	1	6	58	8	<u>2</u>	<u>2</u>	<u>2</u>	66	60	68*	62
1	2	2	2	5	48	<u>15</u>	2	10	1	63	63	73	63
1	2	2	1	6	45	<u>15</u>	<u>2</u>	10	<u>2</u>	60	60	70	62*
1	1	2	2	6	44*	<u>15</u>	<u>4</u>	10	<u>1</u>	59*	59*	69	63
2	1	2	2	6	64	8	<u>4</u>	10	<u>1</u>	74	68	82	69

and $w_j^s, j = 1, \dots, n$ is the elements of the node-path incidence vector, w_j^s of the path $s, s = 1, \dots, S$; i.e., $w_j^s = \begin{cases} 1 & \text{if node } j \text{ belongs to path } s \\ 0 & \text{otherwise} \end{cases}$

Proof. See Appendix B.

Enumerating all the extreme points and paths is burdensome, so we use a relaxation approach and generate the constraints as needed. We propose the following Benders decomposition algorithm to solve the problem exactly.

Solution Algorithm:

Introduce an additional index t to the notation to denote the values at iteration t .

1. Start with an initial solution,

$$\bar{x}^1 \in X^0 = \{(x_{11}, \dots, x_{1|M_1|}, \dots, x_{n1}, \dots, x_{n|M_n|}) : x_{jm} \in B, \forall m \in M_j, j = 1, \dots, n; \sum_{m \in M_j} x_{jm} = 1, j = 1, \dots, n\}$$

Set $z^0 = -\infty, t = 1$.

2. Solve $SP_1^t(\bar{x}^t) : \eta^t = \text{Max}_{s=1, \dots, S} \left\{ \sum_{j=1}^n \sum_{m \in M_j} w_j^s p_{jm} x_{jm}^t \right\}$ and

Solve $SP_2^t(\bar{x}^t)$:

$$\psi^t = \sum_{j=1}^n \sum_{m \in M_j} c_{jm} \bar{x}_{jm}^t + \text{Max} \left\{ \sum_{j=1}^n \sum_{m \in M_j} d_{jm} \bar{x}_{jm}^t u_j : \sum_{j=1}^n u_j \leq \gamma, 0 \leq u_j \leq 1, j = 1, \dots, n \right\}$$

Let u^t be the optimal solution.

3. If $\eta^t > \delta$ then
Find the longest path and its incidence vector \bar{w}^t and set $\bar{u}^t = u^t$.

$$X^t = X^{t-1} \cap \left\{ x \in X^0 : \sum_{j=1}^n \sum_{m \in M_j} \bar{w}_j^t p_{jm} x_{jm} \leq \delta \right\}$$

Else

If $(\psi^t > \varphi^{t-1})$

$$X^t = X^{t-1} \cap \{x \in X^0 : z \geq \sum_{j=1}^n \sum_{m \in M_j} \bar{u}_j^t d_{jm} x_{jm}\}$$

Else

Stop and report \bar{x}^t as the optimal solution.

End if

End if

4. Solve the relaxed master problem, MP^t :

$$\varphi^t = \text{Min} \left\{ \sum_{j=1}^n \sum_{m \in M_j} c_{jm} x_{jm} + z : x \in X^t \right\}$$

Let x^t be the optimal solution.

5. $t = t + 1, \bar{x}^t = x^{t-1}$.
6. Return to Step 2.

Note that solving the first subproblem, namely $SP_1(\bar{x})$ is equivalent to determining the length of the critical path (C_{n+1}) with respect to the given mode assignment. If $C_{n+1} \leq \delta$, the problem is feasible, otherwise it is infeasible. In addition to the feasibility cuts, optimality cuts are inserted (step 3) after solving an additional LP, $SP_2(\bar{x})$ (step 2). A greedy algorithm provides a solution that is

close to optimum, easily. This algorithm orders $\sum_{m \in M_j} d_{jm} \bar{x}_{jm}$ values and this order identifies γ of the activities that affect the objective function the most.

To solve Model 1, we use Benders decomposition, which is known to exhibit slow convergence. Therefore, we include several features to accelerate its convergence and solve large scale problem instances optimally. For a detailed description of the algorithmic enhancements, we refer the readers to Hazır et al. (2010a). Given the complex structure of Models 2 and 3 due to criticality requirement, we use Tabu Search (TS) to solve the models and obtain good quality approximate solutions.

3.2. Tabu search and parameter settings

TS is a local-search improvement heuristic proven to be effective to solve many difficult combinatorial optimization problems (Hazır et al., 2008). It has a penalty mechanism to avoid getting trapped at local optima by forbidding or penalizing moves that cause cycling among solution points previously visited. These forbidden moves are called “tabu”. The short term memory keeps track of move attributes that have changed during the recent past and these attributes become tabu for a specific number of iterations. Under some conditions called *aspiration criterion*, tabu status of a move can be overridden. Two strategies are commonly used to obtain good solutions: diversification to direct the search into less visited regions of the search space, and intensification to fully explore a region.

Local search-based algorithms may not result in high quality solutions for the DTCTP-D, since it is not simple to identify a feasible solution in the neighborhood of the current solution; classical move operators do not guarantee feasibility. To overcome this shortcoming, we apply the features proposed by Kulturel-Konak et al. (2004), which are specially designed to solve constrained optimization problems. Their algorithm uses an adaptive penalty function which encourages the search to proceed through a portion of the infeasible region, namely the “near feasibility threshold (NFT)”. The generated solutions are penalized according to their distances to the feasibility region. The details of penalty structure and our implementation are given in Appendix C.

We represent each solution by a mode assignment vector. Infeasible mode assignments are allowed with some penalties. The algorithm starts the exploration in the infeasible region with the least cost solution. By using a cost-based fitness function that is composed of total project cost and an adaptive penalty cost, it keeps searching for feasible and efficient directions until the stopping criterion is satisfied. To calculate the total project cost, given the mode combination, we first calculate the slack values and define the set of critical activities by using classical Critical Path Method iterations. Given the slack values, we select γ activities by giving priority to the activities with higher deviation values.

We examine the entire neighborhood with single mode decreasing and increasing moves. To find the best parameters, some test problems of varying problem sizes are solved for a wide range of system parameters. In the test runs, we test a tabu list of size 5, 7, and 10. The best solutions are achieved with a tabu list of size 7. We define the aspiration criterion so that tabu status of a move can be overridden if it leads to a solution better than the incumbent solution. The stopping criterion is set to 10,000 iterations after observing that it is sufficient to obtain convergence for the test instances. In order to direct the search into less-visited regions of the search space and escape from local optima, we use a simple diversification strategy. If the incumbent solution is not updated for 1000 iterations, the algorithm restarts with a randomly generated neighbor of the initial solution; the tabu list is initialized and the move values are recalculated according to the new solution.

4. Computational experiments and results

4.1. Experimentation

We use a subset of the random instances generated by Akkan et al. (2005) to test the proposed measures and algorithms. Their test bed is generated for the deterministic problem and mainly the network structure parameters, the number of modes per activity and the tightness of the deadline characterize a problem instance. It includes large sized projects having 85 to 136 activities.

We have employed the instances generated by Akkan et al. (2005), as it includes large instances that are far beyond the problem sizes reported in the literature. Among their instances, we have chosen the test-bed 1; it includes the largest projects. In addition, in order to define the robust problem, two additional parameters, namely robustness level (γ) and uncertainty factor (ψ), are required. The uncertainty factor represents the rate by which the variables d_{jm} are allowed to change around c_{jm} , i.e. $d_{jm} = \psi c_{jm}$.

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. Assessing the distance of a given network from being series–parallel is important for this study, because DTCTP with series–parallel graphs could be solved quickly (Demeulemeester et al., 1996). The second complexity measure, CNC was suggested 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, first 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 calculated. Then, the deadline is set as follows:

$$\delta = T_{min} + \theta(T_{max} - T_{min}) \quad \text{with } 0 \leq \theta \leq 1 \quad (5)$$

Concave (ccv), convex (cvx), and neither concave nor convex functions (hyb) are used to generate the costs. We have generated the uncertainty factor, ψ , using uniform distribution in interval $[0.1, 1]$. When CNC=5, the networks have 85, when CNC=8 they have 136 activities. In the computational study, as we have not observed a significant effect of CI on computational effort in our

pretests, we have set CI=13. This is a result in line with Akkan et al. (2005)'s finding for approximate solutions.

All the algorithms are implemented in C programming language on a Sun UltraSPARC 12 × 400 MHz workstation with 3 GB RAM. Optimization software CPLEX 9.1 is used to solve the linear and integer programs.

4.2. Computational results

We have investigated the effects of various complexity parameters on the solution efficiency of Model 1 and summarize the results in Table 2. For each Γ setting, 36 different project instances are solved. The results of the exact procedure are presented under the column labeled “Optimum”. In order to accelerate the solution, we solve the LP relaxations first and generate cuts from the fractional solutions. Then, integrality constraints are added and the algorithm is resolved. The average number of linear and integer master problems solved and the average CPU time in seconds to solve the instances are reported under the columns “LP Iter.,” “IP Iter.,” “CPU(s)”, respectively.

Moreover, we solve the MP iterations heuristically at a relative optimality tolerance level of 2% until feasibility is satisfied. This means that at each iteration branch-and-bound algorithm is truncated and the best feasible solution found is used. The final integer solution of each MP is guaranteed to be within 2% of the optimal value. This solution could be used as a reliable approximate solution and the results are presented under the column called “Truncated Solution”, within which, the percentage of problem instances that the optimal solution has been found, the average percentage deviation from the optimal solution, the maximum percentage deviation from the optimal solution and the average CPU time reduction with truncation are reported under the columns “Ins Opt (%)”, “Avg Dev (%)”, “Max Dev (%)”, “Dec CPU (%)”, respectively.

Table 2 reveals that CNC measure of the network and the pessimism level are effective on the computational effort given in CPU seconds. However, as CNC increases the number of nodes also increase in our experimental set, it is hard to conclude that higher CNC may lead to higher computational effort, but higher pessimism level does. When the CPU time for solving the deterministic problem and the robust problem are compared, we conclude that considerably higher computational effort is required when the notions of uncertainty and robustness are incorporated into the model. Finally, the truncation-based heuristic may be used as a solution alternative for large scale instances, as it is shown to be

Table 2
Summary of computational results of Model 1.

	Optimum		Truncated solution				
	LP Iter.	IP Iter.	CPU (s)	Ins Opt (%)	Avg Dev (%)	Max Dev (%)	Dec CPU (%)
Robust							
CNC							
5	16.32	19.23	1400.96	81.82	0.03	0.42	25.03
6	22.42	27.33	10,427.91	100	0.00	0.00	15.04
7	23.00	31.14	18,044.09	90.91	0.01	0.22	16.38
8	20.28	31.88	19,139.61	84.00	0.03	0.37	16.20
γ							
[0.25n]	19.42	26.84	8772.72	87.10	0.02	0.31	21.64
[0.5n]	20.40	27.73	10,789.31	87.10	0.02	0.37	18.07
[0.75n]	21.77	27.58	11,690.94	93.55	0.01	0.42	14.81
Deterministic							
CNC							
5	15.00	12.00	410.87	44.44	0.21	0.82	27.77
6	22.50	15.25	1935.93	100.00	0.00	0.00	21.42
7	22.00	17.17	7277.80	71.00	0.02	0.12	28.81
8	21.14	31.29	10,974.72	75.00	0.01	0.04	30.44

generating solutions that are very close to the optimal solution. In the following section, we compare the proposed robust project scheduling models.

4.3. Comparison of the proposed models

In this section, we give a brief review of the metrics to evaluate schedule robustness and perform computational experiments and evaluate the generated schedules.

4.3.1. Robustness measures

Existing robust scheduling studies generally address machine environments and often follow scenario-based approaches, where scenarios for job attributes are required to be defined. They basically employ two types of robustness measures: direct measures, which are derived from realized performances, and approximate measures, which utilize simple surrogate measures. Computational burden of optimizing direct measures is generally higher when compared to surrogate measures. We refer the readers to Sabuncuoğlu and Gören (2005), for a detailed examination of these measures.

In a recent study, Hazır et al. (2010b) introduced nine slack-based measures that could be used to evaluate schedule robustness. They compared these measures through simulation. Given a baseline schedule for a benchmark project and realizations of uncertain variables, the effect of disruptions on project performance is evaluated by the use of some performance measures. These measures reflect the performance of the solutions in a stochastic environment. One example is the average delay in the project completion time as percentage of the project deadline. Having simulated the projects, the robustness measure that has the highest correlation with the performance measures is selected as the best metric to represent robustness.

The following cost based measures are used to evaluate the robustness of the project schedule in this paper:

(a) Expected realized cost

The cost of performing each activity is represented with two parameters: the nominal cost and the worst-case cost. We assume that the nominal cost of a mode is equal to the expectation of the project manager over all the scenarios corresponding to possible alternatives in practice. Therefore, for a given schedule the summation of the nominal costs over all activities defines the expected realized cost of the project schedule. The schedule which has the minimal expected realized cost is chosen as the most robust schedule ($\gamma=0$).

(b) Worst-case cost

The upper bounds of the cost intervals define the worst-case costs, the maximal cost among all possible scenarios. Hence the summation of the upper bounds of the cost intervals over all

activities characterizes the worst-case realized cost of a schedule. The schedule that has the smallest worst-case cost (among all schedules) is chosen. This is a risk-averse approach and corresponds to the minimax objective in decision analysis ($\gamma=n$).

(c) Cost of the reference scenario

This measure concentrates on a specific scenario in which critical activities are realized at the worst-case costs with the remaining activities being realized at the nominal costs. The schedule that has the smallest cost with respect to this scenario is selected.

4.3.2. Computational analysis

Computational experiments are carried out to evaluate the performance of the models under different problem settings and the models are compared using the above-mentioned robustness measures. For comparison purposes we set pessimism level to $\gamma = \lfloor 0.25n \rfloor$. In the computational study, for each setting 3 different instances are solved, hence each model is tested with 36 problems. Table 3 compares the introduced models with 3 robustness measures. For each problem set and robustness measure, all three models are applied to find the best schedule that minimizes the corresponding worst-case cost. While comparing, Model 1 is taken as the reference model and percent differences between robustness measure of the reference model and the criticality-based models are reported. For each robustness measure, the average percentage deviation from the reference model and the percentage of problem instances that the model dominates the reference model are reported in the rows “% Dev”, “% Dom”, respectively. We also report the paired *t*-test confidence intervals (Conf. Int.) for percent differences between robustness measures of Model 1 and criticality-based models. 95% confidence level is used in these intervals. Due to large number of activities at each problem set, we assume that the anticipated cost of the project is approximately normal, and use *t*-test to identify the statistical significance of the differences. The last row expresses whether minimization or maximization of the measure is preferred in terms of robustness.

When worst-case robustness measure is considered, Model 1 dominates the criticality-based models. This finding is consistent with the argument that Model 1 is over-pessimistic. Model 1 is better in most of the problem instances when the expected realized cost of the models is compared. However, in this case differences among model performances are small. Examining Table 3 and comparing the optimal solutions of Models 1 and 2, Model 2 generates schedules that have expected costs 3.6% higher on the average. On the other hand, among the problem instances, in only 13.89% of the instances Model 2 generates schedules with less expected costs.

The parameter γ reflects the risk attitude of the decision makers. As the decision makers become more risk-averse, larger γ values

Table 3
Model comparison with robustness measures.

	Expected cost	Worst-case cost	Reference scenario
Model 2			
% Dev	3.60*	14.56*	-17.33*
Conf. int.	(2.51, 4.69)	(13.29, 15.84)	(-18.86, -15.81)
% Dom	13.89	0	100
Model 3			
% Dev	0.67	9.21*	-9.24*
Conf. int.	(-0.27, 1.61)	(7.96, 10.43)	(-11.41, -7.07)
% Dom	38.89	2.78	94.44
Optimization criterion	Min	Min	Min

* Indicates that % dev is significant at 5% level.

should be used so that schedules with lower worst-case costs could be generated. Note that Model 2 is not so sensitive to the changes in parameter γ when compared to other models. This is basically due to the criticality requirement. The second parameter that affects model characteristics is the slack/duration threshold (SDT). Lower SDT results in different reactions in criticality-based models. Since the risk premiums are incurred for only critical activities in Model 2, as the threshold decreases, the number of critical activities is reduced so that total risk premium decreases and majority of the activities take their nominal costs. However, in Model 3, a reduction in the threshold might increase the total risk premiums, as risks of non-critical activities are also incorporated.

5. Conclusions

In this paper, we have proposed three models to formulate the robust DTCTP. These models assume interval uncertainty for activity costs. It is assumed that the activity times are known with certainty when the mode of the activity is fixed and all the uncertainty is captured by the cost of the activity. In order to solve these models, we have developed both exact and approximate algorithms.

The first model is solved exactly by using Benders decomposition; the other two criticality-based models are rather complex and solved approximately by a tabu search algorithm. The main advantage of Model 1 is that it could be solved exactly. However, the limitation is that the activities with the same cost intervals are assumed to be equally uncertain and all activities are likely to have cost values at the upper bounds.

To evaluate the performance of the algorithms under various problem settings, we have conducted computational experiments. We have assessed the robustness of the schedules generated by the algorithms by using several cost based robustness measures. The models developed in this manuscript address the requirement to generate robust project schedules that are less sensitive to uncertainty. To the best of our knowledge, the models are the first implementation of robust optimization to the DTCTP. In that sense, results presented serve as a useful base to fill the research gap in developing robust project schedules for multi-mode project networks. In addition, they provide decision support to managers in project planning under uncertainty.

As a future extension of this research, robust optimization models could be formulated for Multi-Mode Resource Constrained Project Scheduling Problem (MRCPS), which allows the use of both renewable and nonrenewable resources. For this new generalized problem setting, in addition to the uncertainty costs, the uncertainty in activity durations or in resource requirements or in resource availabilities should also be addressed.

Appendix A. List of abbreviations and notation

A	set of arcs
BOT	Build-Operate-Transfer
CI	complexity index
Conf. Int.	confidence interval
c_{jm}	activity cost of activity j when processed at mode m
C_j	completion time of activity j
CR	set of potentially critical activities
δ	project due date
d_{jm}	the difference between the upper bound and nominal costs, $d_{jm} = \bar{c}_{jm} - c_{jm}$
DTCTP	the discrete time/cost trade-off problem
DTCTP-D	the deadline version of DTCTP

γ	the parameter that reflects the risk attitude of the decision makers, i.e. only γ of activity cost parameters are expected to incur at their upper bounds
M_j	set of executable modes for activity j
n	number of non-dummy activities
N	set of nodes
p_{jm}	activity processing time of activity j when processed at mode m
ψ	uncertainty factor, i.e. $d_{jm} = \psi \cdot c_{jm}$
S	set of activities that have cost values at their upper bounds
SDR	slack duration ratio
θ	the parameter to reflect tightness of deadline or restrictiveness of the budget
TS	total slack
u	binary variable to identify the activities with deviations that influence the objective most
ξ	slack/duration threshold
x_{jm}	binary variable showing whether mode m is assigned to activity j or not

Appendix B. Proof of Proposition 1.

We could reformulate (1) as

$$f(\gamma) = \text{Min} \left[\sum_{j=1}^n \sum_{m \in M_j} c_{jm} x_{jm} + g(x) : x \in X^D \right], \text{ where} \tag{6}$$

$$g(x) = \text{Max} \left\{ \sum_{j=1}^n \sum_{m \in M_j} d_{jm} x_{jm} u_j : \sum_{j=1}^n u_j \leq \gamma, \quad u \in B^n \right\} \tag{7}$$

Given a solution vector, $x \in X^D$, $g(x)$ is a knapsack problem of which LP relaxation has binary optimal solutions (see Theorem 1 of Bertsimas and Sim, 2003), hence $g(x)$ could be rewritten as

$$g(x) = \text{Max} \left\{ \sum_{j=1}^n \sum_{m \in M_j} d_{jm} x_{jm} u_j : \sum_{j=1}^n u_j \leq \gamma, \quad 0 \leq u_j \leq 1, \quad j = 1, \dots, n \right\} \tag{8}$$

This knapsack problem has a non-empty feasible set, for any given γ . Let U be the polytope that defines the feasible set, then U can be defined as a convex combination of K vertex, $u^k = (u_1^k, \dots, u_n^k)$ for $k = 1, \dots, K$ and one of them is the optimum solution of $g(x)$. Thus we have

$$g(x) = \text{Max}_{1 \leq k \leq K} \left\{ \sum_{j=1}^n \sum_{m \in M_j} d_{jm} x_{jm} u_j^k \right\} \tag{9}$$

Note that the polytope does not depend on the solution vector, x . Therefore the same set of extreme points could be used in solution of $g(x)$. If we set $z = g(x)$ and define a constraint for each extreme point of U , (6) could be reformulated as

$$\text{Min}_{x \in X^D, z} \left\{ \sum_{j=1}^n \sum_{m \in M_j} c_{jm} x_{jm} + z : z \geq \sum_{j=1}^n \sum_{m \in M_j} u_j^k d_{jm} x_{jm}, \quad k = 1, \dots, K \right\} \tag{10}$$

Similarly, the deadline constraint could be defined as the longest path in graph G should not be more than the time limit, δ . Combining the path constraints to satisfy deadline feasibility and (10), Model 1 given in Eq. (2) is reformulated as equation set (4). Note that we add the non-negativity constraint on z for the sake of algorithmic convenience; otherwise z could be unrestricted in sign.

Appendix C. Penalty function of the tabu search

In order to deal with the deadline constraint (1.3) and to facilitate generation of feasible solutions throughout the search process, we use the following adaptive penalty function proposed by Kulturel-Konak et al. (2004).

$$f_p(x) = f(x) + (f_{feas} - f_{all}) \left(\frac{d(x)}{NFT} \right) \quad (11)$$

The penalized and unpenalized objective function of a solution vector x are denoted by $f_p(x)$ and $f(x)$, respectively. Throughout the search, the objective functions of the best solution is recorded and the unpenalized objective function of the best solution found and best feasible solution found are denoted by f_{feas} and f_{all} , respectively. The distance of any solution vector x to the feasibility region is measured with the difference between the makespan of the solution and the deadline, which is denoted by $d(x)$. The distance is normalized and it scales itself by using an adaptive memory based parameter NFT . By using the short term memory of the tabu search algorithm and the value of the current move, NFT changes as follows during the search process

$$NFT_{j+1} = \begin{cases} NFT_j + \left(1 + \frac{R_j}{2}\right) & \text{if the current move is to a feasible solution} \\ NFT_j + \left(\frac{1+R_j}{2}\right) & \text{otherwise} \end{cases} \quad (12)$$

In (12), j is used as an index to denote the iteration number and a feasibility ratio of $R_j = F_j/T_j$ is defined, where T_j and F_j denote the size and the number of feasible solutions of the tabu list, respectively. As an initial value, we have used a percent of the deadline constraint, i.e. $NFT_0 = \delta * 0.1$.

References

- Akkan, C., Drexler, 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.
- Ben-Tal, A., Goryashko, A., Guslitzer, E., Nemirovski, A., 2004. Adjusting robust solutions of uncertain linear programs. *Mathematical Programming* 99, 351–376.
- Bertsimas, D., Sim, M., 2003. Robust discrete optimization and network flows. *Mathematical Programming* 98, 49–71.
- 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.
- Cohen, I., Golany, B., Shtub, A., 2007. The stochastic time/cost trade-off problem: a robust optimization approach. *Networks* 49, 175–188.
- De, P.E., Dunne, 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.E., Dunne, 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.
- Erenguc, 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.
- Hazir, Ö., Günalay, Y., Erel, E., 2008. Customer order scheduling problem: a comparative metaheuristics study. *International Journal of Advanced Manufacturing Technology* 37, 589–598.
- Hazir, Ö., Haouari, M., Erel, E., 2010a. Discrete time/cost trade-off problem: a decomposition based solution algorithm for the budget version. *Computers and Operations Research* 37, 649–655.
- Hazir, Ö., Haouari, M., Erel, E., 2010b. Robustness measures and a scheduling algorithm for discrete time/cost trade-off problem. *European Journal of Operational Research* 207, 633–643.
- Herroelen, W., Leus, R., 2005. Project scheduling under uncertainty—survey and research potentials. *European Journal of Operational Research* 165, 289–306.
- Herroelen, W., Leus, R., 2003. The construction of stable project baseline schedules. *European Journal of Operational Research* 156, 550–565.
- Kobylanski, 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.
- Kulturel-Konak, S., Norman, B.A., Coit, D.W., Smith, A.E., 2004. Exploiting tabu search memory in constrained problems. *INFORMS Journal on Computing* 16 (3), 241–254.
- Lambrechts, O., Demeulemeester, E., Herroelen, W., 2008a. Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities. *Journal of Scheduling* 11, 121–136.
- Lambrechts, O., Demeulemeester, E., Herroelen, W., 2008b. A tabu search procedure for developing robust predictive project schedules. *International Journal of Production Economics* 111, 493–508.
- Lova, A., Tormos, P., Cervantes, M., Barber, F., 2009. An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes. *International Journal of Production Economics* 117, 302–316.
- Pascoe, T.L., 1966. Allocation of resources—CPM. *Revue Française de Recherche Opérationnelle* 38, 31–38.
- Sabuncuoğlu, İ., Gören, S., 2005. A review of reactive scheduling research: proactive scheduling and new robustness and stability measures. Technical Report, IE/OR 2005-02, Department of Industrial Engineering, Bilkent University, Ankara.
- Valls, V., Laguna, M., Lino, P., Perez, A., Quintanilla, S., 1998. Project scheduling with stochastic activity interruptions. In: Werglarz, J. (Ed.), *Project Scheduling: Recent Models, Algorithms and Applications*. Kluwer Academic Publishers, Boston, pp. 333–353.
- 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.
- Wei, C., Li, Y., Cai, X., 2010. Robust optimal policies of production and inventory with uncertain returns and demand. *International Journal of Production Economics*, doi:10.1016/j.ijpe.2009.11.008.
- Yamashita, D.S., Armentano, V., Laguna, M., 2007. Robust optimization models for project scheduling with resource availability cost. *Journal of Scheduling* 10, 67–76.