FACULTEIT ECONOMIE
EN BEDRIJFSKUNDE

Tweekerkenstraat 2
B-9000 GENT
Tel. : 32 - (0)9 – 264.34.61
Fax. : 32 - (0)9 – 264.35.92

WORKING PAPER

An electromagnetic time/cost trade-off optimization in project scheduling

Mario Vanhoucke [1]

April 2007

2007/457

[1] *Faculty of Economics and Business Administration, Ghent University, Gent, Belgium*
Operations & Technology Management Centre, Vlerick Leuven Gent Management School, Gent, Belgium
mario.vanhoucke@ugent.be

D/2007/7012/28

ABSTRACT

Abstract. Time/cost trade-offs have been extensively studied in the literature since the development of the critical path method. Recently, the discrete version of the problem formulation has been extended to various practical assumptions, and solved with both exact and heuristic optimization procedures. In this paper, we present a electromagnetic meta-heuristic (EM) algorithm for the discrete time/cost trade-off problem under four different assumptions. We extend the standard electromagnetic meta-heuristic with problem specific features and investigate the influence of various EM specific parameters on the solution quality. We test the new meta-heuristic on a benchmark set from the literature and present extensive computational results.

Keywords: discrete time/cost trade-off problem; work continuity; time/switch constraints; net present value; electromagnetism

1 Introduction

Time/cost trade-offs in projects have been the subject of research since the development of the critical path method, and has led to problem descriptions under various assumptions. While the early endeavours mainly focused on a linear non-increasing relation between activity duration and cost (Kelley and Walker (1959), Fulkerson (1961), Kelley (1961), Ford and Fulkerson (1962), Siemens (1971) and Elmaghraby and Salem (1984)), researchers gradually extended this basic problem type to concave (Falk and Horowitz (1972)), convex (Lamberson and Hocking (1970), Kapur (1973), Siemens and Gooding (1975), Elmaghraby and Salem (1982)) or discrete time/cost relations (Crowston and Thompson (1967), Crowston (1970), Robinson (1975) Billstein and Radermacher (1977), Wiest and Levy (1977), Hindelang and Muth (1979), Patterson and Harvey (1979), Elmaghraby and Kamburowski (1992), De et al. (1995, 1997), Demeulemeester et al. (1996, 1998), Skutella (1998) and Akkan et al. (2005)).

The specific problem addressed in this paper is the discrete time/cost trade-off problem and involves the selection of a set of execution modes (i.e. time/cost pairs for each activity) in order to achieve a certain objective. The objective of this problem description can be threefold. The so-called *deadline* problem involves the scheduling of project activities in order to minimize the total cost of the project while meeting a given deadline. The *budget* problem aims at minimizing the project duration without exceeding a given budget. A third objective involves the generation of the complete efficient time/cost profile over the set of feasible project durations. Due to its practical relevance, the discrete time/cost trade-off problem has been the main subject of research, leading to various solution approaches for various variants of the problem. In this paper, we study the discrete time/cost trade-off problem (DTCTP) under four different assumptions, and develop a meta-heuristic solution approach for the deadline version of the problem based on the principles of electromagnetic optimization (Birbil and Fang (2003)).

The outline of this paper is as follows. In section 2, we briefly present four versions of the discrete time/cost trade-off problem and give an overview of previous research efforts in the literature. Section 3 explains the building blocks of our electromagnetic meta-heuristic for the four versions of the problem in detail. In section 4, we illustrate our novel approach on a problem example. Section 5 reports computational experience on a benchmark dataset. Section 6 draws overall conclusions and highlights future research avenues.

2 Problem description

We assume that a project is represented by an activity-on-the-arc network $G=(N,A)$ where the set of nodes, N, represents network events and the set of arcs, A, represents the activities of the project. The nodes of the network are numbered from the single start node 1 to the single end node n. The duration $d_{ij}(k)$ of an activity $(i, j) \in A$ is a discrete, non-increasing function of the amount of a single non-renewable resource (money, $c_{ij}(k)$) allocated to it. The tuple $(d_{ij}(k), c_{ij}(k))$ is referred to as a *mode*, and we assume that each activity has M_{ij} *modes* with $d_{ij}(1) < d_{ij}(2) < \ldots < d_{ij}(M_{ij})$ and $c_{ij}(1) > c_{ij}(2) > \ldots > c_{ij}(M_{ij})$. We assume that the project is the subject of a pre-specified project deadline δ_n. A solution can be represented by a selected set of modes $(d_{ij}(k), c_{ij}(k))$ (with $k \in \{1, \ldots, M_{ij}\}$) for each activity (i, j) such that a certain objective is optimized. In the current paper, we study four versions of the discrete time/cost trade-off problem which differs in their objective function or their problem characteristics. In the next sub-section, we briefly review the specific differences between the four versions of the DTCTP. Section 2.1 briefly discusses the relevance of the four problem types in research environments and in practice.

2.1 Problem descriptions

Table 1 summarizes the numerous research efforts from literature and presents the classification codes of the four DTCTP variants according to the classification scheme of Herroelen et al. (1999).

Table 1. A literature overview for the discrete time/cost trade-off problem and its extensions

Problem type	Classification code[1]	Literature: Exact	Literature: Heuristic
DTCTP	$1,T \mid cpm, \delta_n, disc, mu \mid av$	[1],[2],[5],[6],[7], [8],[9],[10],[11] [22],[27],[29],[40]	[1],[36]
DTCTP-*tsc*	$1,T \mid tsc, cpm, \delta_n, disc, mu \mid av$	[34],[39]	[36]
DTCTP-*wc*	$1,T \mid cpm, \delta_n, disc, mu \mid av$	[28],[36]	[36]
DTCTP-*npv*	$1,T \mid cpm, \delta_n, disc, mu \mid npv$	[15]	[36]

[1] Following the classification scheme of Herroelen et al. (1999)

The basic *discrete time/cost trade-off problem* (DTCTP) involves the scheduling of project activities by selecting a mode *k* for each activity (*i*, *j*) in order to minimize the total cost $\sum_{(i,j)\in A} c_{ij}(k)$ of the project.

The *discrete time/cost trade-off problem with time-switch constraints* (DTCTP-*tsc*) is very similar to the DTCTP, but assumes that activities are forced to start in a specific time interval and are down in some specified rest interval. *Time-switch constraints* have been introduced by Yang and Chen (2000) as a logical extension of the analyses and achievements of Chen et al. (1997), and have been incorporated in the DTCTP as a special type of constraints in which each activity follows one of three possible work/rest patterns: Firstly, if an activity follows a *day*-pattern it can only be executed during day time, from Monday till Friday. Secondly, an activity follows a *d&n*-pattern if it can be executed during the day or night, from Monday till Friday. Finally, a *dnw*-pattern means that the corresponding activity can be in execution every day or night and also during the weekend.

The *discrete time/cost trade-off problem with work continuity constraints* (DTCTP-*wc*) also minimizes the total cost of the schedule, that consists of the sum of both the direct activity costs (resulting from the selection of a mode for each activity) and work continuity cost for each activity group $A' \subset A$. *Work continuity constraints* have been defined by El-Rayes and Moselhi (1998) in order to model the timely movement of project resources and hence to maintain continuity of work. The work continuity cost represents the cost of the use of resources during the execution of activity group *A'*. This cost can be minimized by minimizing the time-span between the first activity (start of use of resources) and last activity (release of resources) of the activity group *A'*. Vanhoucke and Debels (2007) have shown that the DTCTP-*wc* can be easily transformed into the basic DTCTP by adding two extra arcs per work continuity resource group.

The *discrete time/cost trade-off problem with net present value optimization* (DTCTP-*npv*) involves the scheduling of project activities in order to maximize the net present value of the project subject to precedence relations. In addition to the cost $c_{ij}(k)$ of mode *k* of activity (*i*, *j*), we assume that positive cash flows are associated to the project events (nodes). We use $C_j^+ \geq 0$ to denote the positive payment received at the realization of event *j*. Note that we assume that, without loss of generality, each cash outflow $c_{ij}(k)$ occurs at the completion of each activity. This is a reasonable assumption, since it is always possible to calculate a terminal value of each activity's cash flow upon completion by compounding the associated cash flow to the end of the activity as follows: $c_{ij}(k) = \sum_{t=1}^{d_{ij}(k)} c_{ij}^t(k) e^{\alpha(d_{ij}(k)-t)}$, where α represents the discount rate, $d_{ij}(k)$ the duration of activity (*i*, *j*) at mode *k* and $c_{ij}^t(k)$ the value of the known and deterministic cash outflow (i.e. the cost) of activity *(i, j)* at mode k in period t of its execution. Consequently, the net cash flow of each event j equals $cf_j(k) = C_j^+ - \sum_{(i,j)\in \overline{A_j}} c_{ij}(k)$, with $\overline{A_j}$ the set of all incoming arcs of event j.

2.2 Practical relevance

The development of a meta-heuristic algorithm is based on and inspired by various real-life projects where time/cost trade-offs are a matter of degree. Most of these applications have been described elsewhere such that a detailed description needs not to be repeated here. The use of *time/switch constraints* is straightforward and boils down to the presence of rest and work periods in daily work schedules. Although the specific choice of three work/rest patterns does not exclude more general problem descriptions, it is based on a practical construction project in the field of a water purification company in Belgium (Europe). In this project, a number of filtering machines have to be installed to purify water towers and make use of one or more filtering bags. The more filtering bags are used at the same time, the lower the duration of the particular job but the larger the execution cost. Some of these machines can work without human intervention (a *dnw*-pattern) or, in other cases, with a human intervention once and a while, such as control operations (since these activities only require one person once and a while, they follow a *d&n*-pattern). Of course, certain activities of the project require a whole team and can therefore only be executed during the day (*day*-pattern activities) (source: Vanhoucke et al. (2002)). The practical relevance of *work continuity constraints* have been extensively described in literature. In Vanhoucke (2006), a literature overview and various practical applications of work continuity constraints in project scheduling have been given, among which a huge and complex tunnel construction project in the Netherlands, Europe (see www.westerscheldetunnel.be). Optimization of work continuity could lead to enormous cost savings in the schedule for a large freezing machine needed to bore the links between the two lanes of the tunnel. The use and practical value of *net present value* optimization has been extensively described in Herroelen et al. (1997). Vanhoucke and Demeulemeester (2003) have illustrated the optimization of the net present value in a capacity expansion and construction project at a water purification company in Belgium, Europe.

3 Electromagnetic optimization

In this section, the electromagnetic meta-heuristic procedure to solve the four variants of the DTCTP is explained in detail. The EM approach follows the same generic approach as the original EM algorithm of Birbil and Fang (2003) and can be displayed in pseudo-code as follows:

```
Algorithm EM DTCTP-extensions
  Create initial population
```

```
while stop criterion not met
  compute forces
  apply forces
  local search
endwhile
```

In the remainder of this section, we explain the specific sub-routines of our EM algorithm for the four different versions of the DTCTP.

Create initial population: the algorithm creates an initial population containing *popsize* population elements. Each population element, or a so-called solution point $\mathbf{x}^t$ (t = 1, …, *popsize*) represents a vector of activity modes that need to be transformed into a project schedule. The EM algorithm randomly assigns values to each element x_{ij}^t $((i, j) \in A)$ of vector $\mathbf{x}^t$ between 1 and M_{ij}, and transforms the resulting vector into a project schedule using the following scheduling generation schemes: The *DTCTP* and the DTCTP-*wc* can be efficiently scheduled by determining the earliest completion time of each activity, using the traditional forward pass critical path calculations (problem $cpm|C_{\max}$). The *DTCTP-tsc* can be efficiently scheduled by the adapted forward pass critical path calculation method of Yang and Chen (2000) (problem $cpm,tsc|C_{\max}$). The *DTCTP-npv* reduces to the well-known max-*npv* problem (problem $cpm|npv$), which can be efficiently solved by the recursive search procedure of Vanhoucke et al. (2001). In order to use this activity-on-the-node (AoN) procedure, the AoA project network needs to be considered as an AoN network: each event *j* is then an activity with zero duration and a cash flow $cf_j(k)$ and the arcs represent the precedence relations with time-lags $l_{ij} = d_{ij}(k)$.

Compute forces: This sub-routine calculates charges for each solution point as well as a total force exerted on each solution point by all other solution points, following the principles of Coulomb's law. The charge of each solution point $\mathbf{x}^t$ depends on its objective function value $ov(\mathbf{x}^t)$ in relation to the objection function value of the current best point $\mathbf{x}^{best}$ in the population, with better objective function values resulting in higher charges. The charge q^t of solution point $\mathbf{x}^t$ is determined according to equation [1]. Note that the differences in objective functions are measured as absolute values in order to cope with both minimization and maximization problems. The formula uses |A| as the number of arcs in the project network.

$$q^t = \exp\left(-|A| \frac{\left|ov(\mathbf{x}^t) - ov(\mathbf{x}^{best})\right|}{\sum_{l=1}^{popsize} \left|ov(\mathbf{x}^l) - ov(\mathbf{x}^{best})\right|} \right) \qquad [1]$$

Next, the algorithm calculates a set of force vectors $\mathbf{F}^t$ (t = 1, …, *popsize*) that are exerted on the corresponding solution point x^t, as follows:

$$\mathbf{F}^t = \sum_{\substack{l=1 \\ l \neq t}}^{popsize} \begin{cases} (\mathbf{x}^l - \mathbf{x}^t)\left(\frac{q^t q^l}{\left\|\mathbf{x}^l - \mathbf{x}^t\right\|^2}\right) & \text{If } (\mathrm{ov}(\mathbf{x}^l) < \mathrm{ov}(\mathbf{x}^t) \text{ (DTCTP, DTCTP-}tsc\text{ and the DTCTP-}wc\text{)} \\ & \text{If } (\mathrm{ov}(\mathbf{x}^l) > \mathrm{ov}(\mathbf{x}^t) \text{ (DTCTP-}npv\text{)} \\ (\mathbf{x}^t - \mathbf{x}^l)\left(\frac{q^t q^l}{\left\|\mathbf{x}^l - \mathbf{x}^t\right\|^2}\right) & \text{If } (\mathrm{ov}(\mathbf{x}^l) \geq \mathrm{ov}(\mathbf{x}^t) \text{ (DTCTP, DTCTP-}tsc\text{ and the DTCTP-}wc\text{)} \\ & \text{If } (\mathrm{ov}(\mathbf{x}^l) \leq \mathrm{ov}(\mathbf{x}^t) \text{ (DTCTP-}npv\text{)} \end{cases} \quad [2]$$

The general philosophy of Coulomb's law is that the total force exerted on a solution point by all other solution points is inversely proportional to the distance between the solution points and directly proportional to the product of their charges, as shown in equation [2]. The equation has been modelled such that a point with a relatively good objective function value will attract the other one, whereas a point with the inferior objective value repels the other. The distance between two solution points is measured by the sum of the absolute values of the component-wise differences between the mode number of identical activities. This distance measure is normalized (denoted by the symbol || ||) by dividing it with the maximum of all distances between each pair of solution points, in order to lie in the interval [0, 1].

Apply forces: A new population is generated by moving each population element into a direction dictated by the forces. The imposed force is normalized, by dividing it by the maximal force over all dimensions (i.e. the total number of arcs) for the population element, and therefore only identifies the direction of the move. The magnitude of the move is determined by a randomly selected parameter λ generated from a uniform distribution from the interval [0, 1] (in analogy with Birbil and Fang (2003)) and by the number of modes for each vector element x_{ij}^t ((i, j) $\in A$) of vector $\mathbf{x}^t$.

$$x_{ij}^t = \begin{cases} x_{ij}^t + \lambda \dfrac{F_{ij}^t}{F_{\max}^t}(M_{ij} - x_{ij}^t) & \text{if } F_{ij}^t > 0 \\ x_{ij}^t + \lambda \dfrac{F_{ij}^t}{F_{\max}^t}(x_{ij}^t - 1) & \text{if } F_{ij}^t \leq 0 \end{cases} \quad [3]$$

Since the x_{ij}^t vector elements are discrete numbers, while the EM move assumes a fractional solution space, the calculated x_{ij}^t are round up (fractional value above 0.5) or round down (fractional value below or equal to 0.5) to prevent that some mode numbers are seldom or never chosen (e.g. without the rounding up mechanism, the mode number M_{ij} will only be chosen (i.e. $x_{ij}^t = M_{ij}$**)** when $\boldsymbol{\lambda} = 1$ and $F_{ij}^t = F_{\max}^t$).

Local search: The generation of new solution points is followed by a local search procedure which explores the immediate (Euclidian) neighbourhood of individual points. The EM procedure of the current paper makes use of two sequential heuristic procedures after the generation of each new population, as follows:

Local search 1 – repair function: Any infeasible solution point $\mathbf{x}^t$ for which the project duration exceeds the project deadline is the subject to a heuristic repair function, which transforms infeasible solution points into feasible ones. The repair function iteratively crashes activity durations until the project duration is smaller than the pre-specified project deadline. The project activities can be selected according to various heuristic rules:

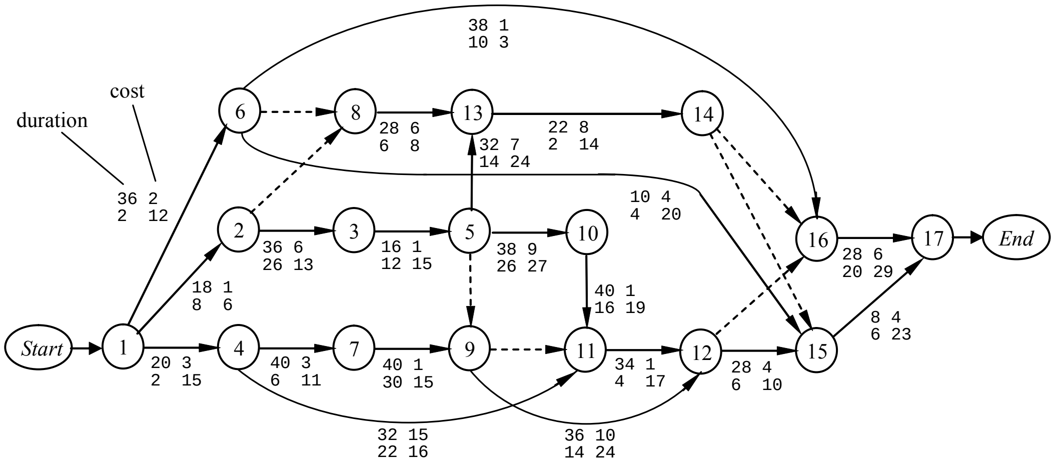
- Random selection of activity/mode combinations (RAN): the repair method iteratively selects project activities at random and crashes its duration to its neighbourhood mode until a feasible solution is obtained.
- Lowest cost per time unit (LCT): all project activities are ranked according to its cost increase per time unit when crashing the activity duration to its neighbourhood mode. The repair method selects the activities with the lowest cost increase first, and updates the cost increase ranking each time an activity duration has been crashed,.
- Lowest absolute cost difference (LAC): this method is similar to the LCT method, but ranks all project activities according to its lowest absolute cost difference the activity's current and neighbourhood mode.

Local search 2 – improvement search: Feasible solution points can possibly be improved by increasing activity durations (and hence, decreasing the activity cost), resulting in an improved objective function. A truncated recursive search procedure randomly ranks all activity/mode combinations and searches for improved schedules based on a truncated dynamic programming heuristic of Debels and Vanhoucke (2007). This heuristic search enumerates a subset of all possible combinations to increase activity durations following the ranking of the randomly generated activity/mode list, and is truncated after a very small pre-defined number of backtracking steps. In the current manuscript, the search is truncated after five backtracking steps, to guarantee a very fast and efficient improvement search.

Stop criterion: The length of a search is determined by a pre-defined stop criterion, which is a function of the number of iterations and the size of the population. More precisely, the length of the search is defined as the product of the population size and the maximum number of iterations (i.e. *popsize* * *iter*), which serves as a measure for the estimated number of generated schedules during the complete search. Indeed, since the electromagnetic heuristic completely replaces all population elements at each iteration run, this product serves as a reliable estimate for the total number of generated schedules. This approach also allows the fine-tuning of the population size for identical stop criteria (i.e. varying the *popsize* and the *iter* parameters while keeping their product at a constant level).

4 Example

In this section, we show the difference between the four problem types on an example network taken from Vanhoucke et al. (2002) and illustrate the philosophy of the electromagnetic meta-heuristic on a small example schedule population of the project. Each arc has two activity modes, which are displayed near the arcs. Dummy arcs are displayed as dashed arcs. We assume that the project deadline δ_n equals 82 days.



Additional information for the DTCTP-*tsc*

Arc (*i*,*j*)	Pattern	Arc (*i*,*j*)	Pattern	Arc (*i*,*j*)	Pattern
(1,2)	*dnw*	(5,10)	*dnw*	(10,11)	*d&n*
(1,4)	*dnw*	(5,13)	*dnw*	(11,12)	*d&n*
(1,6)	*d&n*	(6,8)	*dummy*	(12,15)	*day*
(2,3)	*d&n*	(6,15)	*d&n*	(12,16)	*dummy*
(2,8)	*dummy*	(6,16)	*dnw*	(13,14)	*day*
(3,5)	*dnw*	(7,9)	*day*	(14,15)	*dummy*
(4,7)	*dnw*	(8,13)	*day*	(14,16)	*dummy*
(4,11)	*dnw*	(9,11)	*dummy*	(15,17)	*day*
(5,9)	*dummy*	(9,12)	*d&n*	(16,17)	*dnw*

Additional information for the DTCTP-*wc*

Activity group A' = {(3,5), (5,10), (9,12), (10,11), (11,12)} is subject to work continuity constraints with a cost 6 per time unit of use.

Additional information for the DTCTP-*npv*

Event *j*	C_j^+	Event *j*	C_j^+	Event *j*	C_j^+
1	0	7	10	13	6
2	10	8	12	14	5
3	10	9	11	15	20
4	1	10	14	16	15
5	3	11	10	17	20
6	15	12	3		

Figure 1. An AoA project network example (source network: Vanhoucke et al. (2002))

We assume, without loss of generality, that the duration of each activity mode has been expressed in work periods of 12 hours. The DTCTP versions without time-switch constraints all assume that activities can be executed during all days of the week (no weekends and no holidays), i.e. all days are working days of 12 hours. The DTCTP-*tsc*, on the contrary, allows night execution (for the *d&n* and *dnw* patterns), and hence, can be used to execute two work periods of an activity. On the other hand, the *day* and *d&n* patterns exclude the possibility to execute an activity during the weekend.

Table 2. Solutions for the example project of figure 1

	Activity cost	Work continuity A' duration	Work continuity A' cost	Net present value	Project duration
DTCTP	**127**	[17,68]	306	13.09	82 days
DTCTP-*wc*	153	[27,58]	**186**	13.83	76 days
DTCTP-*npv*	153	[22,72]	300	**18.39**	82 days
DTCTP-*tsc*	**199**	[23,64]	246	-6.36	82 days

Table 2 reports the total activity cost, the work continuity cost, the net present value, the duration of the activities of the work continuity group and the total project duration for four optimal schedules of the example project. The table shows that the objective values of the problem types are indeed optimized (as indicated in bold). The optimal DTCTP and DTCTP-*tsc* schedules have the lowest total activity cost, the DTCTP-*wc* schedule has the lowest total cost (153 + 186) while the DTCTP-*npv* schedule has the highest net present value. Detailed results can be found in appendix 1. Note that the objective function values of the DTCTP-*tsc* are separated from the rest of the table, since they cannot be compared with the values of the other problem formulations. The DTCTP-*tsc* incorporates additional time-switch constraints which are not taken into account by the three other problem types.

Figure 2 shows an example translated from our C++ code of an electromagnetic move for the DTCTP on a population of three solution elements $\mathbf{x}^1$, $\mathbf{x}^2$ and $\mathbf{x}^3$ with a total activity cost of 159, 138 and 141, respectively. The charges, the (normalized) distance matrix and the forces are displayed in the figure. The new solution point $\mathbf{x}^4$ is generated by performing the resulting move on a subset of the activity set (these activities are indicated in grey, e.g. activity (5, 10) is not part of the move) of solution point $\mathbf{x}^3$. Computational results of section 4 reveal that this outperforms the ‘complete’ move on all activities. The general electromagnetic philosophy is conceptually displayed in the figure. Since $\mathbf{x}^3$ lies far from $\mathbf{x}^1$ and has a better objective function, the resulting move is directed away from $\mathbf{x}^1$ with a small magnitude. The opposite is true for $\mathbf{x}^3$ versus $\mathbf{x}^2$. The new solution point $\mathbf{x}^4$ has a project duration larger than the project deadline, and hence, the algorithm randomly decreases some activity durations (displayed by the grey local search area around $\mathbf{x}^4$). The improvement step increases the duration of activity (13,14) within its available slack, leading to the optimal solution for the DTCTP.

Non-dummy activities	Solution points x^1	x^2	x^3	Forces F^3	$\|F^3\|$	New solution point x^4 M[1]	R[2]	I[3]
(1,2)	1	1	2	-0.037	-1.000	**1**	1	1
(1,4)	2	2	2	0.000	0.000	2	2	2
(1,6)	2	2	2	0.000	0.000	**2**	2	2
(2,3)	2	2	2	0.000	0.000	**2**	**1**	1
(3,5)	1	2	2	0.000	0.000	2	2	2
(4,7)	2	2	2	0.000	0.000	**2**	2	2
(4,11)	2	2	2	0.000	0.000	2	**1**	2
(5,10)	2	1	2	-0.037	-1.000	2	2	2
(5,13)	2	2	2	0.000	0.000	2	2	2
(6,15)	2	2	2	0.000	0.000	2	2	2
(6,16)	2	2	2	0.000	0.000	**2**	**1**	2
(7,9)	2	2	2	0.000	0.000	2	2	2
(8,13)	2	2	2	0.000	0.000	**2**	**1**	2
(9,12)	2	2	1	0.037	1.000	**2**	2	2
(10,11)	1	2	1	0.037	1.000	**2**	2	2
(11,12)	2	1	1	0.000	0.000	1	1	1
(12,15)	1	1	2	-0.037	-1.000	**1**	1	1
(13,14)	2	2	2	0.000	0.000	**2**	**1**	**2**
(15,17)	2	2	2	0.000	0.000	2	2	2
(16,17)	1	2	2	0.000	0.000	2	2	2
Activity cost:	159	138	141			120	138	127
Duration:	82	82	82			85	82	82

[1] Move: only executed on a sub-part of the activity set (see computational results), indicated in gray (activity (5,10) was not part of the move). λ has been randomly set to 0.9

[2] Repair function: randomly decreases 5 activity durations

[3] Improvement method: increases the duration of activity (13,14) within the available slack

Distances

	x^1	x^2	x^3
x^1	0	5	6
x^2	5	0	5
x^3	6	5	0

Normalized distances

	x^1	x^2	x^3
x^1	0	0.83	1
x^2	0.83	0	0.83
x^3	1	0.83	0

Charges: $q^1 = 0$, $q^2 = 1$ and $q^3 = 0.308$,

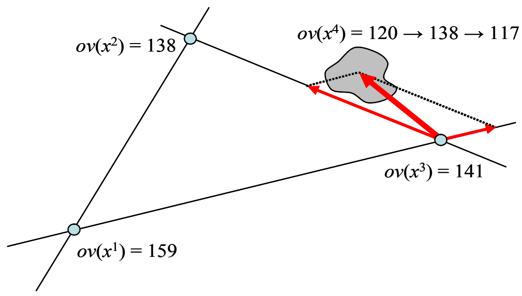


Figure 2. A conceptual representation of the electromagnetic charges and forces calculation

4 Computational results

We have coded the electromagnetic meta-heuristic procedure in Visual C++ version 6.0 to run on a Toshiba personal computer with a Pentium IV 2 GHz processor under Windows XP. In order to evaluate the quality of the heuristic solutions, we compare them with exact solutions for all four problem types as well as another meta-heuristic procedure of Vanhoucke and Debels (2007) which is able to cope with the four versions of the problem under study. The DTCTP and the DTCTP-*wc* instances will be solved to optimality by the procedure of Demeulemeester et al. (1998). The DTCTP-*tsc* instances will be solved by the exact procedure of Vanhoucke (2005). The exact procedure for the DTCTP-*npv* has been linked with the industrial LINDO optimization library version 5.3 (Schrage, 1995) in order to rely on an adapted version of the procedure of Erengüç et al. (1993). The testset used is an extended set from Demeulemeester et al. (1998) and has been previously used by Vanhoucke and Debels (2007). Table 3 summarizes the parameter settings for the different problem instances.

Table 3. Parameter settings for our computational tests

Dataset from [x][1]	Extensions			
	DTCTP	DTCTP-*tsc* [2]	DTCTP-*wc* [3]	DTCTP-*npv* [4]
The number of activities ranges from 10, 20, 30, 40 to 50 activities, the number of modes is fixed at 2, 4 or 6 modes or is randomly chosen between [1,3], [1,7] or [1,11] and the project deadline lies between the minimal and maximal project duration in steps of 0% (minimum), 25%, 50%, 75% and 100% (maximum)	no extensions	Each activity belongs to an work/rest pattern, ranging from [0,0,100], [0,33,66], [0,66,33], [0,100,0], [33,0,66], [33,33,33], [33,66,0], [66,0,33], [66,33, 0] to [100,0,0] (where [x,y,z] indicates [%day pattern,%day and night pattern,%day, %night and weekend pattern])	We define three sizes of activity groups subject to work continuity constraints containing 25%, 50% of 75% of the original activities. The work continuity cost has been defined as low, in-between or high as, respectively, 75%, 100% and 150% of the average total activity cost of the corresponding activity group.	Each event node has a certain cash inflow value, which is a function of the total cost of all incoming activities, ranging between the minimal value and the maximal value in in steps of 0% (minimum cost), 25%, 50%, 75% and 100% (maximum cost)

[1] Each setting contains 30 problem instances, resulting in 30*5*6*5 = 4,500 problem instances

[2] The extended set contains 4,500 * 10 = 45,000 problem instances

[3] The extended set contains 4,500 * 3 * 3 = 40,500 problem instances

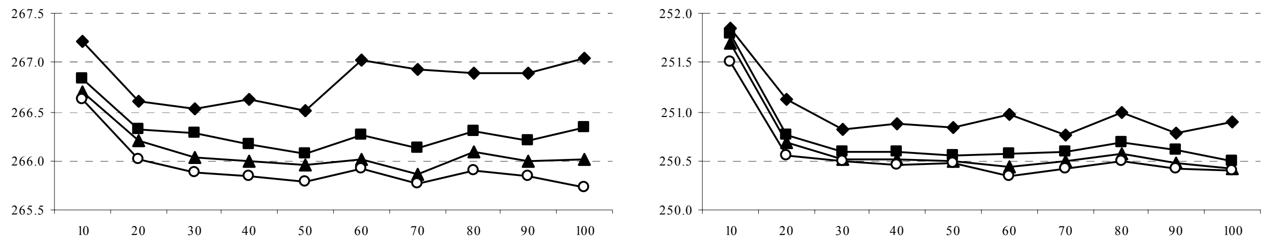
[4] The extended set contains 4,500 * 5 = 22,500 problem instances

We test the quality of our meta-heuristic procedure in two ways. In section 4.1, we test the influence of the various EM building blocks on the solution quality of the problem instances. In this section, we restrict our tests on a subset of table 3, where we have selected only 50-activity networks with the number of modes randomly selected between 1 and 11. In section 4.2, we compare our generated solutions with both exact and heuristic solutions as described earlier and test our algorithms on the complete testset of table 3.

4.1 Electromagnetic heuristic performance

This section reports results on the influence of the length of the electromagnetic search as well as the contribution of the repair function and the improvement method on the solution quality of the four DTCTP versions. Figure 3 displays the contribution of the stop criterion on the solution quality. The figure compares four equal stop criteria values while varying the population size and the number of iterations for each stop criterion. The population size varies between 10 and 100 in steps of 10, keeping the *popsize* * *iter* product constant at a level of 1,000, 2,500, 5,000 or 10,000.

DTCTP DTCTP-*tsc*



DTCTP-*wc* DTCPTP-*npv*

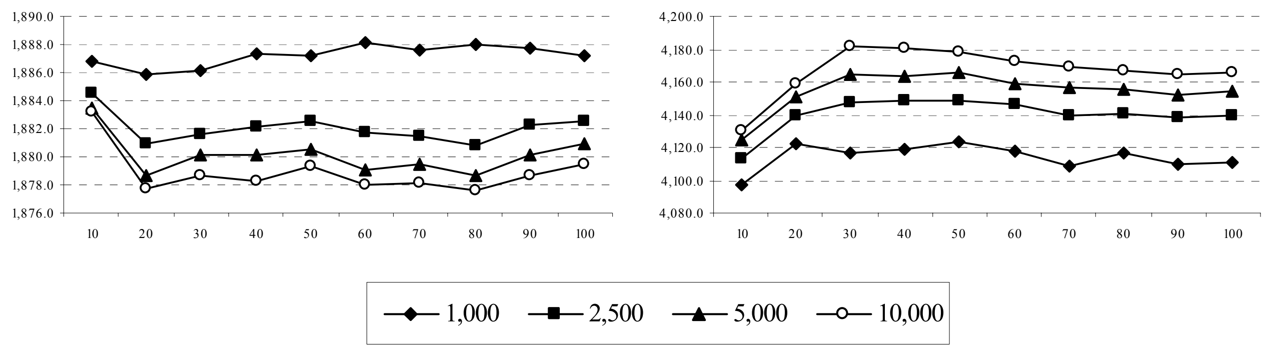


Figure 3. Average solution quality under four stop criterion values

All figures show that a longer search, expressed as increasing *popsize* * *iter* values and displayed as separate lines in each figure, leads to improved solutions, at the expense of a higher CPU time (see table 4).

The division between the population size and the number of iterations for equal *popsize* * *iter* values is less intuitively clear. The figures reveal an improving start trend for increasing *popsize* values, with an optimum at 70, 60, 20 and 30 for the different DTCTP versions. However, table 4 shows that an increase in the population size (and a resulting decrease in the number of iterations) goes hand in hand with a larger CPU time requirement. This observation can mainly be explained by the electromagnetic calculations of distances between solutions, charges and forces. Consider, as an example, a *popsize* * *iter* value of 1,000. A population size of 10 and a number of iterations equal to 100 means that 10 * 10 = 100 distances, forces and charges need to be calculated per iteration, resulting in 100 * 100 calculations during the complete run. However, when the population size equals 100 and the number of iterations equals 10, 100 * 100 distances, charges and forces need to be calculated at each iteration, leading to 100,000 calculations during the complete run. We also observed that a higher population size leads to more repairs, which also contributes to the higher computational burden.

Due to this observed trade-off between the solution quality and the computational burden, we have selected a population size of 30 in the remainder of this manuscript for all DTCTP versions, regardless of the number of iterations.

Table 4. CPU times (in seconds) for the DTCTP graph of figure 3

	popsize									
popsize* * *iter	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
1000	0.415	0.523	0.621	0.729	0.861	0.974	1.145	1.301	1.509	1.702
2500	0.938	1.214	1.397	1.597	1.812	2.027	2.256	2.477	2.781	3.024
5000	1.757	2.345	2.650	2.998	3.350	3.689	4.045	4.428	4.846	5.199
10000	3.310	4.557	5.136	5.796	6.429	7.040	7.617	8.264	8.965	9.619

The contribution of the improvement method and the repair function is analyzed in table 5 under a stop criterion of 999 generated schedules (a *popsize* of 30 and a number of iterations of 33). The table displays the solution quality measured as the average relative deviation from the best found project cost (for the DTCTP, DTCTP-*tsc* and DTCTP-*wc*) or from the best found net present value (DTCTP-*npv*) in each cell (a cell with 0% denotes the best found solution method).

Table 5. Influence of the repair function and the improvement method on the solution quality

<table>
<tr><th colspan="2" rowspan="2">Improvement</th><th colspan="2">DTCTP</th><th colspan="2">DTCTP-tsc</th><th colspan="2">DTCTP-wc</th><th colspan="2">DTCTP-npv</th></tr>
<tr><th>no</th><th>yes</th><th>no</th><th>yes</th><th>no</th><th>yes</th><th>no</th><th>yes</th></tr>
<tr><td rowspan="4">repair</td><td>no</td><td rowspan="4">215.90%</td><td>1.43%</td><td rowspan="4">226.20%</td><td>1.14%</td><td rowspan="4">162.57%</td><td>4.05%</td><td rowspan="4">126.24%</td><td>5.28%</td></tr>
<tr><td>RAN</td><td>0.93%</td><td>0.66%</td><td>1.60%</td><td>1.95%</td></tr>
<tr><td>LCT</td><td>0.10%</td><td>0.10%</td><td>0.20%</td><td>0.00%</td></tr>
<tr><td>LAC</td><td>0.00%</td><td>0.00%</td><td>0.00%</td><td>0.11%</td></tr>
</table>

The table shows the indispensable contribution of the improvement method on the solution quality. The results without improvement (columns with label "no") are very poor, and the repair function has no influence whatsoever on the solution quality. This can be explained by the specific implementation of the repair function, which decreases activity durations (randomly or controlled) to construct deadline feasible schedules. When this repair function is not immediately followed by the improvement method, very poor feasible solutions will be created. The contribution of the repair function in combination with the improvement method (columns with label "yes") is best for the LAC heuristic to minimize overall project costs and for the LCT heuristic to maximize the net present value.

Although not displayed in a table or a figure, we have found the best results when the forces (eq. [3]) are operated on a sub-part of the schedule (see the example of figure 2). More precisely, forces are applied on each activity with a probability of only 50%.

4.2 Benchmark results

In this section, we report computational results on our benchmark set (table 3) and compare the solutions of each DTCTP version with an exact and a meta-heuristic solution. The exact solutions are found by the solution approaches of Demeulemeester et al. (1998) (for the DTCTP and the DTCTP-*wc*), Vanhoucke (2005) (DTCTP-*tsc*) and Erengüç et al. (1993) (DTCTP-*npv*). We also compare the obtained solutions with the heuristic solutions of Vanhoucke and Debels (2007), who presented, to the best of our knowledge, the only heuristic solution procedure available in the open literature that can solve the four versions of the DTCTP.

Solutions obtained by exact procedures: The results for the exact solution procedure are obtained by allowing a maximal time limit of 1 minute. After that, the procedure stops and the solution is reported. In doing so, the obtained solution can be classified in one of the following categories: optimal solution, feasible (but not necessarily optimal) solution or infeasible solution (i.e. no solution found). The columns with label "*% opt*" are used to denote the percentage of problem instances for which an optimal solution has been found. The columns labelled with "*% limit*" display the percentage of problem instances for which a feasible (but not guaranteed to be optimal) solution has been found within the pre-specified time limit of 1 minute. This means that the procedure already has found one or more feasible solutions, but it is truncated after the pre-specified time. The columns with "*% infeas*" show the percentage of problem instances for which no feasible solution has been found within the pre-specified time limit of 1 minute. Each problem instance belongs to one of these three categories, which are used for comparison purposes with the heuristic procedures. The column labelled "*Avg CPU*" contains the average CPU time needed to solve the problem instances.

Meta-heuristic solutions: The results found by the heuristic procedure are compared with the results of one of the three categories. The instances for which an exact solution has been found (i.e. "% *opt*") are used to compare them with the heuristic solutions as follows. The column labelled with "*% opt=*" displays the percentage of problem instances for which the heuristic solution has found the optimal solution (only for the problem instances of column "% *opt*"). The column indicated with "*avg opt*" gives the average percentage of deviation from the optimal solution (only for the problem instances of column "% *opt*"). The problem instances with a feasible, though not necessarily an optimal solution (i.e. "*% limit*") are analyzed as follows. On the one hand, the column labelled with "*% limit+*" displays the percentage of problem instances for which the heuristic solution is better than the feasible solution found by the exact procedure (only for the problem instances of column "% *limit*") On the other hand, the column labelled with "*% limit-*" indicates the percentage of problem instances for which the heuristic solution is worse than the feasible solution found by the exact procedure (only for the problem instances of column "*% limit*"). The remaining fraction is then the percentage of problem instances with a solution equal to the feasible solution found. Furthermore, the columns labelled with "*avg limit+*" display the average percentage of deviation (improvement) of the heuristic solution (only for the problem instances of column "*% limit*") while the columns with label "*avg limit-*" refer to the average percentage of deviation (deterioration) of the heuristic solution (only for the problem instances of column "*% limit*").

Stop criterion: The computational tests haven been performed under various stop criteria. All runs performed by exact solution approaches have been truncated after 60 seconds CPU time. The meta-heuristic solution approaches make use of two classes of stop criteria: The first class of test runs has been truncated after a pre-specified number of generated schedules (rows 'schedule limit'), as discussed previously. More precisely, the electromagnetic procedure has been truncated when the product *popsize* * *iter* has reached the 1,000 or 5,000 threshold. These solutions will be compared with the meta-heuristic solution procedure of Vanhoucke and Debels (2007) truncated after 1,000 and 5,000 generated schedules. The second class of test runs have been truncated after a pre-specified time limit of 0.1 and 0.5 seconds (rows 'time limit').

Table 6. Computational comparative results

	Exact	DTCTP [9]				DTCTP-*tsc* [34]				DTCTP-*wc* [9]				DTCTP-*npv* [15]			
time limit	*% opt*	100%				89.76%				100%				49.70%			
	% limit	0%				8.08%				0%				50.30%			
	% infeas	0%				2.16%				0%				-			
	Avg CPU	0.081				7.906				0.105				32.948			
	Heuristic	**[36a]**	**[a]**	**[36b]**	**[b]**	**[36a]**	**[a]**	**[36b]**	**[b]**	**[36a]**	**[a]**	**[36b]**	**[b]**	**[36a]**	**[a]**	**[36b]**	**[b]**
schedule limit	*% opt=*	79.09%	94.20%	86.22%	96.31%	86.80%	97.65%	92.37%	98.46%	58.70%	84.13%	69.25%	89.11%	90.79%	91.68%	91.92%	92.84%
	avg opt	0.50%	0.049%	0.217%	0.028%	0.263%	0.022%	0.117%	0.014%	98.600%	0.118%	0.504%	0.070%	2.536%	1.503%	2.199%	1.453%
	% limit+	-	-	-	-	29.40%	56.22%	46.31%	59.71%	-	-	-	-	80.73%	83.52%	83.53%	84.65%
	avg limit+	-	-	-	-	0.350%	0.739%	0.580%	0.794%	-	-	-	-	107.124%	115.840%	113.074%	116.864%
	% limit-	-	-	-	-	50.80%	11.00%	25.17%	6.24%	-	-	-	-	19.16%	16.33%	16.37%	15.19%
	avg limit-	-	-	-	-	1.203%	0.088%	0.348%	0.047%	-	-	-	-	6.570%	1.436%	3.637%	1.224%
	Avg CPU	0.049	0.195	0.244	0.824	0.079	0.273	0.388	1.158	0.075	0.331	0.363	1.465	0.570	0.606	2.869	2.281
	Heuristic	**[36c]**	**[c]**	**[36d]**	**[d]**	**[36c]**	**[c]**	**[36d]**	**[d]**	**[36c]**	**[c]**	**[36d]**	**[d]**	**[36c]**	**[c]**	**[36d]**	**[d]**
time limit	*% opt=*	80.51%	85.09%	86.67%	93.98%	86.74%	90.10%	92.51%	97.54%	60.86%	67.21%	71.10%	83.17%	85.62%	84.85%	90.43%	91.01%
	avg opt	0.856%	0.607%	0.239%	0.073%	0.380%	0.305%	0.115%	0.023%	1.788%	1.067%	0.482%	0.150%	28.523%	4.504%	11.916%	2.563%
	% limit+	-	-		-	14.91%	18.45%	38.78%	51.95%	-	-	-	-	64.83%	72.69%	76.97%	80.69%
	avg limit+	-	-		-	0.172%	0.205%	0.482%	0.670%	-	-	-	-	82.311%	97.491%	101.604%	111.301%
	% limit-	-	-		-	73.79%	67.05%	36.30%	15.92%	-	-	-	-	35.11%	27.17%	22.90%	19.17%
	avg limit-	-	-		-	5.171%	3.359%	0.653%	0.160%	-	-	-	-	60.275%	10.961%	25.139%	5.285%
	Avg CPU	0.1	0.1	0.5	0.5	0.1	0.1	0.5	0.5	0.1	0.1	0.5	0.5	0.1	0.1	0.5	0.5

Solved by [9] Demeulemeester et al. (1998), [34] Vanhoucke (2005), [15] Erengüç et al. (1993)

[36a] Solved by Vanhoucke and Debels (2007) truncated after 1,000 generated schedules

[a] Solved by electromagnetic procedure of the current paper (popsize = 30 and iter = 1000)

[36b] Solved by Vanhoucke and Debels (2007) truncated after 5,000 generated schedules

[b] Solved by electromagnetic procedure of the current paper (popsize = 30 and iter = 5000)

[36c] Solved by Vanhoucke and Debels (2007) truncated after 0.1 seconds

[c] Solved by electromagnetic procedure of the current paper truncated after 0.1 seconds (popsize = 30)

[36d] Solved by Vanhoucke and Debels (2007) truncated after 0.5 seconds

[d] Solved by electromagnetic procedure of the current paper truncated after 0.5 seconds (popsize = 30)

The results of table 6 can be summarized as follows. Since there is no major difference between the optimal procedures for both problem types (apart from extra arcs), their results are discussed together. The table shows that the exact branch-and-bound procedure of Demeulemeester et al. (1998) can solve all problem instances for the *DTCTP* and the *DTCTP-wc* to optimality within the time limit of 60 seconds. Although the electromagnetic procedure is able to generate high-quality solutions within a small time fraction, it has already been concluded by Vanhoucke and Debels (2007) that it is not beneficial to rely on this meta-heuristic procedure to solve instances of this size. These authors have shown that even for tests on larger instances (with up to 200 activities and 50 modes) there is no need to use meta-heuristic procedures. The main reason is that the specific approach used for the exact branch-and-bound algorithm of Demeulemeester et al. (1998) results very quickly in truncated (heuristic) solutions that are very close to the optimal solution. Moreover, thanks to the use of an efficient lower bound calculation of Ford and Fulkerson (1962), many nodes can be evaluated in the branch-and-bound tree within a limited amount of CPU-time, and hence, the meta-heuristic procedure has no computational advantage on that aspect. Note that, for obvious reasons, the columns "*% limit*" and "*% infeas*", the columns "*% limit̃*" and "*% limit+*", and their corresponding columns "*avg limit̃*" and "*avg limit+*" are empty.

The table shows that meta-heuristic algorithms are good alternatives to exact algorithm for both the DTCTP-*tsc* and the DTCTP-*npv*. The procedure of Vanhoucke and Debels (2007) and the newly developed electromagnetic procedure outperform the exact algorithms, both in terms of CPU time and solution quality. The "schedule limit" results also show that the electromagnetic search outperforms the Vanhoucke and Debels (2007) procedure, but at a higher CPU time expense (due to the calculation of charges, forces and distances). For this reason, we have tested and compared their results within a time limit of 0.1 and 0.5 seconds. These results show that the electromagnetic procedure performs best.

5 Conclusions

This paper studied four variants for the well-known discrete time/cost trade-off problem, and developed an electromagnetic meta-heuristic to solve the problem types. The heuristic relies on the law of Coulomb and iteratively calculate charges and forces on population elements following the principles of Birbil and Fang (2003). The generation of a schedule has been extended by a dual local search method. The first local search method repairs infeasible solutions by crashing project activities while the second local search randomly increases activity durations of feasible project solutions within the available activity slack.

The computational results are promising and show that solutions are comparable and often better than a previously developed procedure of Vanhoucke and Debels (2007). Lower CPU time and higher solution quality have been obtained by running tests on a large dataset truncated after a pre-specified number of generated solutions of after a certain CPU time limit.

Our future research intensions lie in the development of dedicated solution approaches for the DTCTP-*tsc* and DTCTP-*npv*. While the meta-heuristic procedures in the current paper are rather general search procedures that can deal with all four problem types, dedicated algorithms that exploit problem specific information should allow to test and compare results on larger problem instances for which the exact algorithm fail to provide a feasible solution.

6 References

[1] C. Akkan, A. Drexl and A. Kimms, “Network decomposition-based benchmark results for the discrete time-cost trade-off problem”, European Journal of Operational Research, Vol. 165, pp. 339-358, 2005.

[2] N. Billstein and F.J. Radermacher, “Time-cost optimization,” Methods of Operations Research, Vol. 27, pp. 274-294, 1977.

[3] I. Birbil, and S.C. Fang, “An electro-magnetism-like mechanism for global optimization,” Journal of Global Optimization, Vol. 25, pp. 263-282, 2003.

[4] Y.L. Chen, D. Rinks and K. Tang, “Critical path in an activity network with time constraints,” European Journal of Operational Research, Vol. 100, pp. 122-133, 1997.

[5] W. Crowston, “Network reduction and solution,” Operations Research Quarterly, Vol. 21, pp. 435-450, 1970.

[6] W. Crowston and G.L. Thompson, "Decision CPM: A method for simultaneous planning, scheduling and control of projects," Operations Research, Vol. 15, pp. 407-426, 1967.

[7] P. De, E.J. Dunne, J.B. Ghosh and C.E. Wells, "The discrete time/cost trade-off problem revisited," European Journal of Operational Research, Vol. 81, pp. 225-238, 1995.

[8] P. De, E.J. Dunne, J.B. Ghosh and C.E. Wells, "Complexity of the discrete time/cost trade-off problem for project networks," Operations Research, Vol. 45, pp. 302-306, 1997.

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

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

[11] S.E. Elmaghraby and J. Kamburowski, "The analysis of activity networks under generalized precedence relations," Management Science, Vol. 38, pp. 1245-1263, 1992.

[12] S.E. Elmaghraby and A. Salem, "Optimal project compression under quadratic cost functions", Applications of management science, Vol. 2, pp 1-39, 1982.

[13] S.E. Elmaghraby and A. Salem, "Optimal linear approximation in project compression," IIE Transactions, Vol. 16, pp. 339-347, 1984.

[14] K. El-Rayes and O. Moselhi, "Resource-driven scheduling of repetitive activities," Construction Management and Economics, Vol. 16, pp. 433-446, 1998.

[15] S.S. Erengüç, S. Tufekci and C.J. Zappe, "Solving time/cost trade-off problems with discounted cash flows using generalized Benders decomposition," Naval Research Logistics, Vol. 40, pp. 25-50, 1993.

[16] J.E. Falk and J.L. Horowitz, "Critical path problems with concave cost-time curves," Management Science, Vol. 19, pp. 446-455, 1972.

[17] L.R. Ford and D.R. Fulkerson, Flows in networks, Princeton University Press: New Jersey, 1962.

[18] D.R. Fulkerson, "A network flow computation for project cost curves," Management Science, Vol. 7, pp. 167-178, 1961.

[19] S.K. Goyal, "A note on the paper: A simple CPM time/cost trade-off algorithm," Management Science, Vol. 21, pp. 718-722, 1975.

[20] W. Herroelen, E. Demeulemeester and B. De Reyck, "A classification scheme for project scheduling problems," in: J. Weglarz (Ed.), Handbook on Recent Advances in Project Scheduling, Kluwer Academic Publishers, Amsterdam, 1999, chapter 1, pp. 1-26.

[21] Herroelen, W., Demeulemeester, E. and Van Dommelen, P. (1997). Project network models with discounted cash flows: A guided tour through recent developments. *European Journal of Operational Research*, 100, 97-121.

[22] T.J. Hindelang and j.F. Muth, "A dynamic programming algorithm for decision CPM networks," Operations Research, Vol. 27, pp. 225-241, 1979.

[23] K.C. Kapur, “An algorithm for the project cost/duration analysis problem with quadratic and convex cost functions,” IIE Transactions, Vol. 5, pp. 314-322, 1973.

[24] J.E. Kelley, 1961, “Critical path planning and scheduling: Mathematical basis,” Operations Research, Vol. 9, pp. 296-320.

[25] J.E. Kelley and M.R. Walker, 1959, Critical path planning and scheduling: An introduction, Mauchly Associates, Ambler, PA.

[26] L.R. Lamberson and R.R. Hocking, “Optimum time compression in project scheduling,” Management Science, Vol. 16, pp. B-597-B-606, 1970.

[27] J.H. Patterson and R.T. Harvey, “An implicit enumeration algorithm for the time/cost trade-off problem in project network analysis,” Foundations of Control Engineering, Vol. 6, pp. 107-117, 1979.

[28] R.M. Reda, “RPM: repetitive project modelling,” Journal of Construction Engineering and Management, Vol. 116 (2), pp. 316-330, 1990.

[29] D.R. Robinson, “A dynamic programming solution to cost/time trade-off for CPM,” Management Science, Vol. 22, pp. 158-166, 1975.

[30] L. Schrage, LINDO: Optimization software for linear programming, LINDO Systems Inc.: Chicago, IL, 1995.

[31] N. Siemens, “A simple CPM time/cost trade-off algorithm,” Management Science, Vol. 17, pp. B-354-363, 1971.

[32] N. Siemens and C. Gooding, “Reducing project duration at minimum cost: a time/cost trade-off algorithm, OMEGA, Vol. 3, pp. 569-581, 1975.

[33] M. Skutella, “Approximation algorithms for the discrete time-cost tradeoff problem,” Mathematics of Operations Research, Vol. 23, pp. 909-929, 1998.

[34] M. Vanhoucke, 2005, “New computational results for the discrete time/cost trade-off problem with time-switch constraints,” European Journal of Operational Research, Vol. 165, pp. 359-374.

[35] Vanhoucke, M., 2006, “Work continuity constraints in project scheduling”, Journal of Construction Engineering and Management, 132, 14-25.

[36] M. Vanhoucke and D. Debels, “The discrete time/cost trade-off problem: extensions and heuristic procedures”, to appear in Journal of Scheduling, 2007.

[37] Vanhoucke, M. and Demeulemeester, E., 2003, “The Application of Project Scheduling Techniques in a Real-Life Environment”, Project Management Journal, 34, 30-42.

[38] M. Vanhoucke, E. Demeulemeester and W. Herroelen, “On maximizing the net present value of a project under renewable resource constraints,” Management Science, 47, pp. 1113-1121, 2001.

[39] M. Vanhoucke, E. Demeulemeester and W. Herroelen, 2002, “Discrete time/cost trade-offs in project scheduling with time-switch constraints,” Journal of the Operational Research Society, Vol. 53, pp. 741-751.

[40] J.D. Wiest and F.K. Levy, A management guide to PERT/CPM: with GERT/PDM/DCPM and other networks, Prentice-Hall, Inc: NJ, 1977.

[41] H.H. Yang and Y.L. Chen, 2000, “Finding the critical path in an activity network with time-switch constraints,” European Journal of Operational Research, Vol. 120, pp. 603-613.

7 Appendix

Table 7. Optimal mode selection and the corresponding activity cost

non-dummy arcs	input modes		Selected mode and corresponding activity cost							
			DTCTP		DTCTP-*tsc*		DTCTP-*wc*		DTCTP-*npv*	
	$k=1$	$k=2$	mode	cost	mode	cost	mode	cost	mode	cost
(1,2)	6	1	1	6	1	6	2	1	1	6
(1,4)	15	3	2	3	1	15	2	3	2	3
(1,6)	12	2	2	2	2	2	2	2	1	12
(2,3)	13	6	1	13	1	13	2	6	2	6
(2,8)	0	0	1	0	1	0	2	0	1	0
(3,5)	15	1	2	1	2	1	2	1	2	1
(4,7)	11	3	2	3	1	11	1	11	2	3
(4,11)	16	15	2	15	2	15	2	15	2	15
(5,9)	0	0	1	0	1	0	2	0	1	0
(5,10)	27	9	2	9	2	9	1	27	2	9
(5,13)	24	7	2	7	2	7	2	7	2	7
(6,8)	0	0	1	0	1	0	2	0	1	0
(6,15)	20	4	2	4	2	4	2	4	2	4
(6,16)	3	1	2	1	2	1	2	1	2	1
(7,9)	15	1	2	1	1	15	2	1	2	1
(8,13)	8	6	2	6	2	6	2	6	2	6
(9,11)	0	0	1	0	1	0	2	0	1	0
(9,12)	24	10	2	10	1	24	2	10	2	10
(10,11)	19	1	2	1	1	19	1	19	2	1
(11,12)	17	1	1	17	1	17	1	17	1	17
(12,15)	10	4	1	10	1	10	2	4	1	10
(12,16)	0	0	1	0	1	0	2	0	1	0
(13,14)	14	8	2	8	1	14	2	8	2	8
(14,15)	0	0	1	0	1	0	2	0	1	0
(14,16)	0	0	1	0	1	0	2	0	1	0
(15,17)	23	4	2	4	2	4	2	4	2	4
(16,17)	29	6	2	6	2	6	2	6	1	29
total activity cost				127		199		153		153

work continuity group A'

dummy activities

Table 8. The event realization moments and the net present value and work continuity cost

event nodes	CI	Net present value and work continuity information							
		DTCTP		DTCTP-*tsc*		DTCTP-*wc*		DTCTP-*npv*	
		RM	NC	RM	NC	RM	NC	RM	NC
1	0	0	0	0	0	0	0	0	0
2	10	4	4	4	4	9	9	4	4
3	10	17	-3	23	-3	27	4	22	4
4	1	10	-2	1	-14	10	-2	10	-2
5	3	25	2	31	2	35	2	30	2
6	15	18	13	24	13	18	13	1	3
7	10	30	7	4	-1	13	-1	30	7
8	12	18	12	24	12	18	12	4	12
9	11	50	10	46	-4	35	10	50	10
10	14	44	5	50	5	48	-13	49	5
11	10	64	-6	60	-24	56	-24	70	-6
12	3	68	-24	64	-38	58	-24	72	-24
13	6	41	-7	64	-7	51	-7	61	-7
14	5	52	-3	66	-9	62	-3	72	-3
15	20	71	6	72	6	72	12	75	6
16	15	68	14	66	14	62	14	72	14
17	20	82	10	82	10	76	10	82	-13
net present value			13.09		52.27		13.83		18.39
work continuity cost		306		246		186		300	

start event work continuity group

end event work continuity group

CI = cash inflow of each event (C_j^+)

RM = realization moment of each event

NC = net cash flow for each event ($cf_j(k)$)