TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Approximating multi-objective time-dependent optimization problems

Citation for published version (APA):
Dabia, S., Talbi, E-G., Woensel, van, T., & Kok, de, A. G. (2011). *Approximating multi-objective time-dependent optimization problems*. (BETA publicatie : working papers; Vol. 362). Technische Universiteit Eindhoven.

Document status and date:
Published: 01/01/2011

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 25. Sep. 2023

Research School for Operations
Management and Logistics

Approximating Multi-Objective Time-Dependent Optimization Problems

Said Dabia, El-Ghazali Talbi, Tom van Woensel, Ton de Kok

Beta Working Paper series 362

BETA publicatie	WP 362 (working paper)
ISBN	
ISSN	
NUR	982
Eindhoven	November 2011

INFORMS JOURNAL ON COMPUTING
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0899-1499 | EISSN 1526-5528 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000
© 0000 INFORMS

Approximating Multi-Objective Time-Dependent Optimization Problems

Said Dabia
Eindhoven University of Technology, School of Industrial Engineering, The Netherlands, s.dabia@tue.nl

El-Ghazali Talbi
University of Lille, INRIA, CNRS, France, El-ghazali.Talbi@lifl.fr

Tom Van Woensel, Ton De Kok
Eindhoven University of Technology, School of Industrial Engineering, The Netherlands,
{t.v.woensel@tue.nl, a.g.d.kok@tue.nl}

In many practical situations, decisions are multi-objective in nature. Furthermore, costs and profits are time-dependent, *i.e.* depending upon the time a decision is taken, different costs and profits are incurred. In this paper, we propose a generic approach to deal with multi-objective time-dependent optimization problems (MOTDP). The aim is to determine the set of Pareto solutions that capture the interactions between the different objectives. Due, to the complexity of MOTDP, an efficient approximation based on dynamic programming is developed. The approximation has a provable worst case performance guarantee. Even though the approximate Pareto set consists of less solutions, it represents a good coverage of the true set of Pareto solutions. Numerical results are presented showing the value of the approximation.

Key words: Multi-objective optimization; Time-dependent costs; Approximation; dynamic programming; ϵ-dominance
History:

1. Introduction

Many optimization problems encountered in practice are multi-objective in nature, *i.e.* different objectives are conflicting and equally important. Many times, it is not desirable to drop some of them or to optimize them in a composite single objective or hierarchical manner. For instance, while designing a product, many criteria are taken into account: *e.g.* the product's reliability should be maximized, while the cost and the environmental impact should be minimized. Obviously, the amount of examples that can be formulated is infinite.

Contrary to single-objective optimization problems where the optimal solution is a single candidate, the aim of Multi-Objective optimization Problems (in short, MOPs) is the determination of the set of points representing the compromise solutions between the different conflicting objectives. This set of points is defined as the set of Pareto solutions or the Pareto front. A solution is Pareto, if it is not possible to improve an objective without deteriorating at least another one. In this line of thought, decision makers are presented with the entire Pareto front (rather than a single solution) such that they can select their preferred solution depending on their specific situation. Although the roots of multi-objective optimization go back to the nineteenth century in the work of Edgeworth (1881) and Pareto (1896), in the literature, most optimization problems dealt with are mono-objective. In fact, multi-objective cost functions are usually reduced to a composite single objective cost function by using a (weighted) sum of the various objectives. It is argued that solutions obtained as such are only a small subset of the entire set of Pareto solutions, and therefore could lead to suboptimal or infeasible managerial decisions (Ehrgott (2005), Miettinen (1999) and Talbi (2009)).

When dealing with Multi-Objective optimization Problems, a number of complicating factors influence both the quality and easiness of obtaining the Pareto fronts. First, the number of Pareto solutions increases with the size of the problem, mainly with the number of objectives. Therefore, multi-objective decision making becomes very challenging. In fact, most multi-objective problems are NP-hard. Hence, it is computationally very expensive to compute the complete Pareto front. Furthermore, it is not straightforward to select a solution from the Pareto fronts. Consequently, many researchers direct their efforts on approximating the Pareto front in the hope of reducing the complexity of the applied algorithms and producing good approximations (*i.e.* approximate Pareto fronts) of the Pareto front. Approximate Pareto fronts contain less solutions, which facilitate the selection of a final solution. However, a good approximate Pareto front should have enough solutions to properly represent the real Pareto front. Second, in many practical settings, the cost parameters change over time which makes scheduling problems harder. For instance, in a traffic network, travel costs are a function of travel times that change depending on the time of the day a vehicle is traveling.

In this paper, an approximation algorithm based on dynamic programming is proposed for *Multi-objective Time-Dependent Optimization Problems* (MOTDPs). The multi-dimensional profit space is partitioned into intervals with exponentially increasing size. Each interval defines a cluster of solutions that are considered to be *very close* to each other. From each cluster only one solution is kept and the dynamic programming is adapted to the partitioned profit space. In this way, in each iteration of the dynamic programming only a polynomial number of solutions is processed. The approximation has a provable performance guarantee. Moreover, it is easy for a decision maker to select a solution as he is provided with a limited number of solutions. Even with less solutions, the resulting approximate Pareto front still properly covers the real Pareto front in the sense that each optimal Pareto solution is represented by at least one approximate Pareto solution. The evaluation of the proposed approach is demonstrated on a time-dependent capacitated traveling salesman problem with time windows, though the approximation presented is applicable for a generic MOTDP. In fact, the proposed approximation could be applied to a variety of well-known optimization problems for which a dynamic programming formulation is possible (*e.g.* knapsack problems, shortest path problems, traveling salesman problems, variants of vehicle routing problems, sequence alignment problems, ...).

The contributions of this paper are summarized as follows. A generic approximation is proposed which can be applied to structured optimization problem that can be solved by means of dynamic programming. The approximation generates an approximate Pareto front with less solutions (a polynomial number of solutions). However, the approximate Pareto front represents a very good coverage of the real Pareto front. Additionally, the approximation's worst case performance guarantee is provable. Hence, it is easier for the decision to select a solution for which he has a good feeling about its quality. Furthermore, the approximation is very flexible in the sense that the decision maker can choose different precision levels for the different objectives. In fact, the decision maker might be willing to tolerate more error for objectives that are less sensible. Finally, we are dealing with a realistic MOP for which costs are dynamic. In fact, in most real-life situations costs are time-dependent. To our knowledge, this is the first time a multi-objective time-dependent optimization problem is approximated.

The paper is organized as follows. Section 2 reviews the literature relevant to MOPs. Section 3 is devoted to the introduction of the main concepts related to MOTDPs. Section 4 describes a generic MOTDP and the assumptions made to guarantee the non-dominance principle. In Section 5, an approximation of the Pareto front is developed and the main results of the paper are derived. In section 6, the main results are validated based on a specific traveling salesman problem with time windows. Finally, Section 7 concludes with a summary of the main results.

2. Literature Review

As in mono-objective optimization, MOPs can be divided into two categories: those whose solutions are encoded with *real-valued* variables, also known as *continuous MOPs*, and those where the solutions are encoded using *discrete* variables, known as multi-objective combinatorial optimization problems *(MOCO)*. In the class of continuous MOP, an infinite number of Pareto solutions composes the Pareto front whereas in combinatorial MOPs, both the feasible set S and the Pareto front are finite. Most heuristics for solving MOPs are designed to deal with continuous MOPs using, for instance, multi-objective simplex (Zeleny (1982) and Steuer (1986)). In the last decade, there is also a growing interest in solving combinatorial MOPs. However, in most of the cases, they are bi-objective optimization problems. Furthermore, there is a lack of test instances for real-life combinatorial MOPs, especially problems with many objectives (Ishibuchi et al. (2008)), uncertainty (Liefooghe et al. (2007)) and dynamicity (Farina et al. (2004)).

The study of computational complexity classes for MOCO started with the work of Serafini (1986), and Papadimitriou and Yannakakis (2002). They made a connection between the results obtained in mono-objective combinatorial optimization and the multi-objective field for several canonical problems. Serafini (1986) depicted nine possible definitions for MOCO problems and established reductions between them in order to facilitate obtaining complexity results. He showed that the following definition (denoted as V8 in his article) can be considered as a standard reference version to measure the computational complexity of MOCO problems. The definition can also be seen as the decision problem associated with a MOCO problem.

Definition 1 (Generic definition of MOCO by Serafini (1986)). . Given $z \in Z^n$, does there exist $x \in X$ such that $f_i(x) \leq z$?

A NP-hard mono-objective problem implies a NP-hard character to its multi-objective extensions. In the multi-objective case the NP-hard class appears for the majority of problems. For some of them derived from mono-objective NP-hard problems, it is easy to prove also their NP-hardness. This is a reason why the study must focus on determining a specific stronger class as NP-hard in the strong sense or results about the NP-completeness. For example, NP-completeness is proved for shortest path problems, assignment problems and minimum maximal matching by Serafini (1986); for the minimum weight spanning tree by Camerini and Vercellis (1984); and for the max-linear spanning tree by Hamacher and Ruhe (1994).

Similarly to mono-objective optimization problems, MOPs can be solved by means of *exact* and *approximate* algorithms. In the literature, more attention has been devoted to bi-criteria optimization problems by using exact methods such as branch and bound algorithms (Sen et al. (1988), Ulungu and Teghem (1995), Visée et al. (1998), Sayin and Karabati (1999) and Lemesre et al. (2007a)), branch and cut (Jozefowiez et al. (2007)), *A* algorithm* (Stewart and White (1991), Mandow and Millan (1996)), and dynamic programming (White (1982) and Carraway et al. (1990)). Because of the complexity of MOPs, exact methods are only effective for problems with small instances and with no more than two criteria. However, there exists some new advances in this area, with several exact approaches proposed in the literature for bi-objective (Lemesre et al. (2006), Laumanns et al. (2004) and Lemesre et al. (2007b)) and multi-objective problems (Lemesre et al. (2006)). Approximate methods are mainly used to solve large-scale problems and when multiple criteria are involved. They can be divided into two classes: on the one hand algorithms that are only applicable to a specific problem. Such algorithms are developed based on some knowledge on the structure of the problem at hand. On the other hand, meta-heuristics which are of general purpose, in the sense that they can be applicable to a large variety of MOPs. A unifying view for analyzing, designing and implementing multi-objective meta-heuristics is provided in the book Talbi (2009). The main drawback of meta-heuristics is that they do not guarantee the performances related to the Pareto front. Moreover, the resulting approximate Pareto fronts might not properly cover the real Pareto front as they might contain very few solutions.

In the context of mono-objective optimization problems, an ϵ-approximation scheme is an algorithm that, for every instance of the problem, finds an approximate solution that is guaranteed to be within a constant factor from optimal. Two classes of approximation schemes are mainly considered: Polynomial Time Approximation Scheme (PTAS) and Fully Polynomial Time Approximation Scheme (FPTAS). For any $\epsilon > 0$, a PTAS runs in time polynomial in the size of the instance, while an FPTAS runs in time polynomial in the size of the instance and $1/\epsilon$. From a computational complexity point of view, FPTAS are the strongest approximation schemes with performance guarantee that can be obtained for optimization problems. The notion of approximation schemes can be generalized to the case of multi-objective optimization problems by considering, for each solution on the approximate Pareto front, worst case performance guarantees with regard to all criteria.

3. Definitions, Variables and Background

This section aims to give the most relevant definitions and notations used in the remainder of the paper.

Definition 2 (Multi-objective time-dependent optimization problem). A multi-objective time-dependent optimization problem (MOTDP) is defined as:

$$(MOTDP) = \begin{cases} vmin\ F(x,t) = (f_1(x,t), f_2(x,t), \ldots, f_n(x,t)) \\ s.t.\ (x,t) \in S \end{cases} \quad (1)$$

where n $(n \geq 2)$ is the number of objectives, $(x,t) = (x_1...,x_k,t)$ is the vector representing the decision variables depending upon the starting time t, and S represents the set of feasible solutions associated with equality and inequality constraints, and explicit bounds. $F(x,t) = (f_1(x,t), f_2(x,t)..., f_n(x,t))$ is the vector of objectives to be optimized. Note that the value of MOTDP also depends on the starting time t, which could also be considered as a decision variable.

The search space S represents the decision space or parameter space of the MOTDP. The space to which the objective vector belongs is called the objective space. F is a mapping from the decision space to the objective space which evaluates, for a starting time t, the quality of each solution $(x_1, \ldots, x_k)$ by assigning an objective vector $(y_1(t), \ldots, y_n(t))$. The objective vector represents the quality (or *fitness*) of the solution (Fig. 1). The decision maker is usually interested in the value of a solution on each criterion. Therefore, the analysis of MOPs is done in the objective space. The set $Y = F(S)$ represents the feasible points in the objective space, and $y(t) = F(x,t) = (y_1(t), y_2(t), ..., y_n(t))$, where $y_i(t) = f_i(x,t)$, is a point in the objective space.

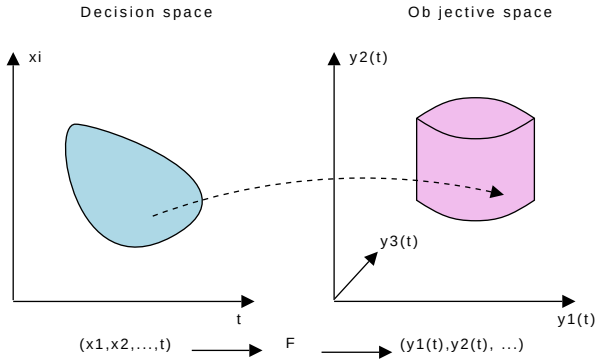

Figure 1 **Decision space and objective space in a time-dependent MOP.**

DEFINITION 3 (PARETO DOMINANCE). An objective vector $u(t) = (u_1(t), \ldots, u_n(t))$ is said to dominate $v(t') = (v_1(t'), \ldots, v_n(t'))$ (we write $u(t) \prec v(t')$) if and only if no component of $v(t')$ is smaller than the corresponding component of $u(t)$ and at least one component of $u(t)$ is strictly smaller, *i.e.*

$$\forall i \in \{1, \ldots, n\} : u_i(t) \leq v_i(t') \wedge \exists i \in \{1, \ldots, n\} : u_i(t) < v_i(t').$$

DEFINITION 4 (PARETO OPTIMALITY). A solution $(x^*, t^*) \in S$ is Pareto Optimal[1] if for every $(x,t) \in S$, $F(x,t)$ does not dominate $F(x^*, t^*)$, i.e. $F(x,t) \not\prec F(x^*, t^*)$.

A MOTDP involves the determination of a set of solutions known as the *Pareto optimal set*. The image of this set in the objective space is denoted as the *Pareto front*. We define the Pareto optimal set and Pareto front as follows:

DEFINITION 5 (PARETO OPTIMAL SET). For a given MOTDP (F, S), the Pareto optimal set is defined as $\mathcal{P}^* = \{(x,t) \in S / \nexists (x', t') \in S, F(x', t') \prec F(x,t)\}$.

DEFINITION 6 (PARETO FRONT). For a given MOTDP (F, S) and its Pareto optimal set $\mathcal{P}^*$, the Pareto front is defined as $\mathcal{PF}^* = \{F(x,t), (x,t) \in \mathcal{P}^*\}$.

The generation of the Pareto optimal set often turns out to be practically impossible or computationally too expensive. Therefore, good approximations of $\mathcal{PF}^*$ are desirable. We define the ϵ-Pareto concept as follows:

DEFINITION 7 (ϵ-DOMINANCE). An objective vector $u(t) = (u_1(t), \ldots, u_n(t))$ is said to ϵ-dominate $v(t') = (v_1(t'), \ldots, v_n(t'))$ (we write $u(t) \prec_\epsilon v(t')$) if and only if no component of $v(t')$ is dominated by the corresponding component of $u(t) - \epsilon$, *i.e.*

$$\forall i \in \{1, \ldots, n\} : u_i(t) - \epsilon \leq v_i(t') \wedge \exists i \in \{1, \ldots, n\} : u_i(t) - \epsilon < v_i(t').$$

DEFINITION 8 (ϵ-PARETO OPTIMALITY). A solution $(x^*, t^*) \in S$ is ϵ-Pareto Optimal if for every $(x,t) \in S$, $F(x,t)$ does not ϵ-dominate $F(x^*, t^*)$, *i.e.* $F(x,t) \not\prec_\epsilon F(x^*, t^*)$ (Fig.2).

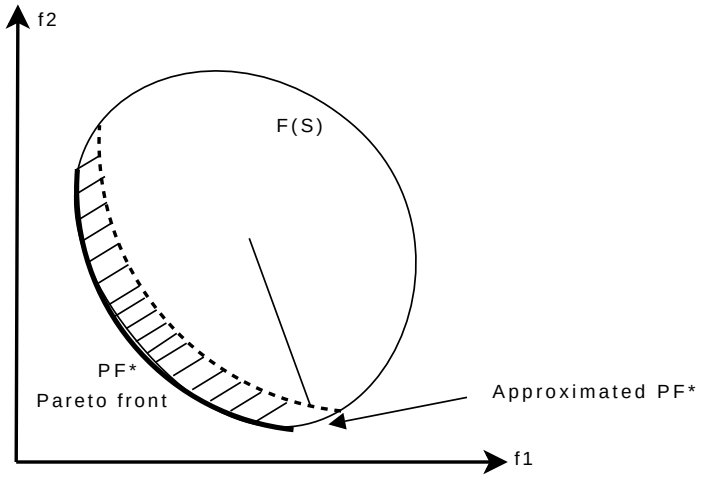

Figure 2 **ϵ-Pareto concept. Sets of Pareto and ϵ-Pareto solutions.**

[1] The Pareto optimal solutions are also known under the name of *acceptable solutions*, *efficient*, *not-dominated*, *non-inferior*.

4. A Dynamic Programming formulation for the MOTDP

We consider a generic Multi-Objective Time-Dependent Optimization problem (MOTDP) related to a set of items H. Practically, H represents jobs to be processed on a machine, or customers to be served by a truck or project tasks to be executed. The MOTDP involves the joint minimization of n objectives. We assume that each instance of MOTDP naturally decomposes into $|H|$ arrays of vectors $\left\{X^1, X^2, ..., X^{|H|}\right\}$. For every $k \in [\![1, |H|]\!]$, the array X^k consists of the vectors $\left\{I_1^k, I_2^k, ..., I_{|X^k|}^k\right\}$ such that for all $k \in [\![1, |H|]\!]$ and $p \in [\![1, |X^k|]\!]$, I_p^k is an instance of MOTDP. By considering $(x^\alpha)_{\alpha \in \mathbb{N}}$ to be a set of mappings from H to H, I_p^k is represented by the vector $[x^\alpha(1), x^\alpha(2), ..., x^\alpha(k)]$ for some $\alpha \in \mathbb{N}$. Note that $|X^k|$ depends on MOTDP and not on any instance of it. I_p^k is then a schedule consisting of at most k items, and X^k the set of schedules with at most k items. To each item $i \in H$ a time-dependent vector costs $c^i(t) = [c_1^i(t), c_2^i(t), ..., c_n^i(t)]$ is associated. The cost $c_r^i(t)$ is related to the r^{th} objective and is incurred when item i is processed at time t. We consider the first objective to be the total processing time of the items in H. Consequently, $c_1^i(t)$ is the processing time of item i. Without loss of generality we assume that MOTDP is bounded in all objectives and let B denote the vector bounds such that B_r, $r \in [\![1, n]\!]$, is the upper bound corresponding to the r^{th} objective. Table1 summarizes the notations used in the paper.

Table 1 **Notation used in this paper.**

Variable	Description
H	: A set of items
t	: Time. Time origin is always taken to be 0
t_0	: Starting time, $t_0 \geq 0$
S	: The set of feasible solutions
t^i	: starting time of processing item i
t_X	: Processing time of the subset of items $X \subseteq S$
$\|X\|$	: Size of the set X
$[y_r]_{r=1}^{r=n}$	: The vector (or array of vectors) $y_1, y_2, ..., y_n$
$\lfloor x \rfloor$	: Nearest integer smaller or equal to the real number x
$\lceil x \rceil$	: Nearest integer larger or equal to the real number x
$[\![a, b]\!]$	: The interval of integer numbers between a and b (a and b are also integers).
$\mathbb{N}$	: The set of natural numbers
$\mathbb{R}$	: The set of real numbers
ϵ	: The approximation worst case precision

The MOTDP is solved by a Dynamic Program (DP). DP goes through $|H|$ iterations such that, in the k^{th} iteration it processes the input X^k and generates X^{k+1}, the input to the next iteration. For simplicity and without loss of generality, we assume that the starting time t_0 is given. Therefore, when unnecessary the time index is omitted from the notation. The value of DP is then denoted $F(X^k)$ instead of $F(X^k, t_0)$, and is represented by the array of vectors $\left\{f^k(I_1^k), f^k(I_2^k), ..., f^k(I_{|X^k|}^k)\right\}$ where f^k is a mapping from X^k to $\mathbb{R}^n$ such that:

$$f^k(I_p^k) = \left[f_r^k(I_p^k)\right]_{r=1}^{r=n} \quad \text{and} \quad f_r^k(I_p^k) = \sum_{i=1}^{k} c_r^{x^\alpha(i)}(t^{x^\alpha(i)}) \tag{2}$$

$t^{x^\alpha(i)}$ is calculated in an iterative way as follows:

$$t^{x^\alpha(1)} = t_0 \quad \text{and} \quad t^{x^\alpha(i+1)} = t^{x^\alpha(i)} + c_1^{x^\alpha(i)}(t^{x^\alpha(i)}) \tag{3}$$

The structure of DP is assuming the generation of Pareto solutions in all iterations. However, because of time-dependency, the generation of the Pareto front is not guaranteed. In fact, time-dependency makes it possible to generate non Pareto solutions from Pareto ones. This problem

can be solved by imposing some structure on the vector costs. This structure is illustrated in the following assumption:

ASSUMPTION 1 **(FIFO)**. *For every item i, and times t_1 and t_2, $t_1 \leq t_2$ implies that:*

$$t_1 + c_1^i(t_1) \leq t_2 + c_1^i(t_2) \tag{4}$$
$$c_r^i(t_1) \leq c_r^i(t_2), \quad \textit{for all} \quad r \in [\![2, n]\!] \tag{5}$$

Remember that the first objective reflects the total processing time of the items in H. Assumption 1 thus means that for every item $i \in H$ the processing time adheres to the well-known FIFO assumption (*i.e.* tasks cannot overtake each other). Furthermore, all other objectives costs are monotonically increasing in time.

The recursion for DP is formulated as follows:

$$F(X^k) = \underset{\substack{1 \leq p \leq |X^{k-1}| \\ i \in H \setminus I_p^{k-1}}}{vmin} \left\{ \left[f_r\left(I_p^{k-1}\right) + c_r^i\left(t_0 + t_{I_p^{k-1}}\right) \right]_{r=1}^{r=n}, \ (I_p^{k-1}, t_0) \in S \right\} \tag{6}$$

In most cases $|X^k|$ is exponential in the size of MOTDP. Consequently, the running time of DP is exponential too. Therefore, the determination of the optimal Pareto front for most MOTDP is very time consuming. In the next section, an approximate dynamic program (DP^ϵ) is presented.

5. Approximation based on Dynamic Programming, DP^ϵ

In order to reduce the complexity of DP, we impose extra structure during its execution. In each iteration k, we trim the generated set $F(X^k)$ of Pareto solutions by eliminating the solutions that are very close to each other. The trimmed set $F(\widetilde{X}^k)$ is then used in the dynamic program to compute the untrimmed set $F(X^{k+1})$. The idea of adding this type of structure to the execution of algorithms was first introduced by Ibarra and Kim (1975). Sahni (1976) and Woeginger (2005) applied it to a variety of single-objective and time-independent scheduling problems.

5.1. Setting up DP^ϵ

The new approximate dynamic program recursion is formulated as follows:

$$F(X^k) = \underset{\substack{1 \leq p \leq |\widetilde{X}^{k-1}| \\ i \in H \setminus \widetilde{I}_p^{k-1}}}{vmin} \left\{ \left[f_r\left(\widetilde{I}_p^{k-1}\right) + c_r^i\left(t_0 + t_{\widetilde{I}_p^{k-1}}\right) \right]_{r=1}^{r=n}, \ (\widetilde{I}_p^{k-1}, t_0) \in S \right\} \tag{7}$$

Formally, the set $F(X^k)$ can be represented by geometric points in the polyhedron $[0, B_1] \times [0, B_2] \times ... \times [0, B_n]$. The polyhedron is cut into multiple boxes of exponentially increasing size. Points contained by the same box are considered to be very close to each other. In each box, only one point is retained. Figure 3 illustrates the trimming of the set of Pareto solutions in case $n = 2$. Obviously, the size of boxes depends on the precision vector $\epsilon = [\epsilon_r]_{r=1}^{r=n}$. Smaller precisions result in smaller boxes. Moreover, boxes with an exponentially increasing size result in a sort of logarithmic scale with a polynomial number of boxes (polynomial in the input size). Therefore, after trimming, only a polynomial number of solutions is kept. Note that if the boxes' size increases linearly, the number of solutions kept will still be exponential.

The cuts on the axis corresponding to the r^{th} objective are executed at the coordinates $\Delta_r^{m_r}, m_r = 1, ..., L_r$, such that:

$$\Delta_r = 1 + \frac{\epsilon_r}{2|H|} \tag{8}$$

Where $\epsilon = [\epsilon_r]_{r=1}^{r=n}$ is a vector of real numbers between 0 and 1 representing the approximation's precision vector. ϵ_r is the precision related to the r^{th} objective.

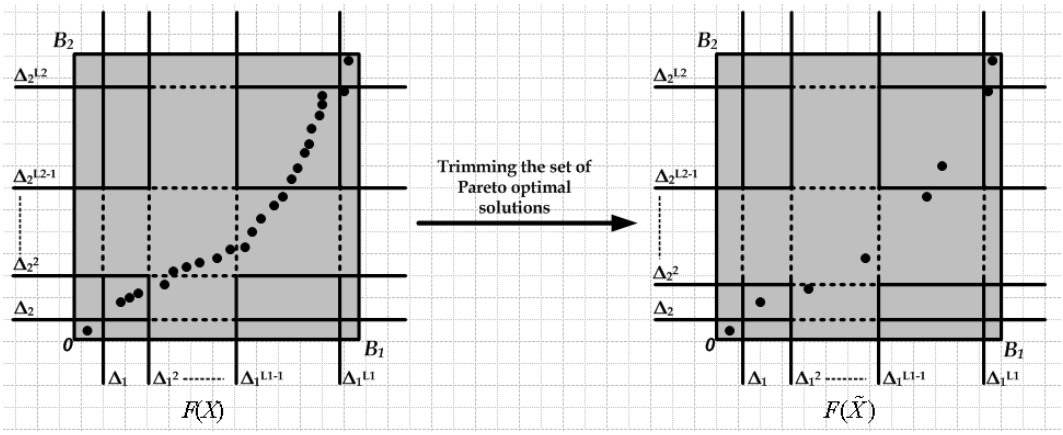

Figure 3 **The reduction of the set of Pareto-optimal solutions in case of two objectives.**

Note that different precisions can be assigned to the different objectives. Objectives that are less sensitive to errors (*e.g.* with a flat cost structure) could be assigned larger precision values. Furthermore, the form of the vector $\Delta = [\Delta_r]_{r=1}^{r=n}$ given by equation (8) is justified by two reasons. First, the values of the vector $\Delta = [\Delta_r]_{r=1}^{r=n}$ will have values very close to 1. Hence, two solutions in the same box are indeed very close to each other. Second, we know how the sequence $\left(1+\frac{x}{a}\right)^a$ behaves when a goes to infinity.

The values of L_r are chosen such that $\Delta_r^{L_r} \leq B_r$. We choose:

$$L_r = \left\lceil \frac{lnB_r}{ln\Delta_r} \right\rceil \leq \left\lceil \left(1 + \frac{2|H|}{\epsilon_r}\right) lnB_r \right\rceil \tag{9}$$

The trimmed set of Pareto solutions contains at most $\prod_{r=1}^{n} L_r$ solutions. We can compute the complexity of DP^ϵ as being proportional to:

$$\begin{aligned} \sum_{k=1}^{|H|} \left| F\left(\widetilde{X}^k\right) \right| &= O\left(|H| \prod_{r=1}^{n} L_r \right) \\ &= O\left(|H|^{n+1} \prod_{r=1}^{n} \left(\frac{lnB_r}{\epsilon_r} \right) \right) \end{aligned} \tag{10}$$

Given all elements of the arrays X^k are binary coded, Equation (9) and (10) shows that the running time of DP^ϵ is polynomial in the input size and in the vector $\frac{1}{\epsilon} = \left[\frac{1}{\epsilon_r}\right]_{r=1}^{r=n}$. Moreover, the running time is still exponential in the number of objectives. However, in most cases the number of objectives is limited. Obviously there is a trade-off between the values of the elements of the vector ϵ and the running time of DP^ϵ. In fact, for small values of ϵ more solutions are kept during the execution of DP^ϵ as the boxes illustrated in Figure 3 are smaller. Therefore, more data is processed which results in an increase of the running time. Intuitively, the quality of the approximation depends on ε. In fact, because of the trimming action, DP^ϵ generates incorrect data. However, less data is trimmed for small values of ϵ which limits the error caused by the trimming action.

5.2. Worst Case Performance of DP^ϵ

We show that the worst case performance guarantee is such that every solution generated by DP is at most a constant factor ϵ from that of a DP^ϵ solution. For the sake of a clear presentation, the proofs needed in this section are available in the Appendix. Henceforth, we assume the following structure regarding the vector costs:

ASSUMPTION 2. *For every item $i \in S$ and real number $1 \leq \alpha \leq 2$, it holds that for every time t and $r \in [\![1,n]\!]$:*

$$c_r^i(\alpha t) \leq \alpha c_r^i(t) \tag{11}$$

Assumption 2 means that processing item i at a later time αt, instead of time t, will not multiply costs by more than a coefficient α. In other words, sudden increases in costs are not allowed (which seems reasonable from a real-life point of view). For instance, the costs functions $c_r^i(t) = t$, $c_r^i(t) = \sqrt{t}$ and $c_r^i(t) = ln(t)$ satisfy assumption (2). Note that the time origin is taken to be 0. We show that if assumption 2 is satisfied, the following lemma holds:

LEMMA 1. *For all $p \in [\![1, |X^k|]\!]$, there exists $q \in [\![1, |\widetilde{X}^k|]\!]$ such that for all $k \in [\![1, |H|]\!]$ and $r \in [\![2,n]\!]$ (given assumptions 1 and 2):*

$$t_{\widetilde{I}_q^k} \leq \Delta_1^k t_{I_p^k} + (\Delta_1^k - 1)t_0 \quad and \quad f_r(\widetilde{I}_q^k) \leq max(\Delta_1^k, \Delta_r^k) f_r(I_p^k) \tag{12}$$

Lemma 1 presents an important result as upper bounds on the quality of the approximation are proved. Furthermore, it shows that the approximate Pareto front covers well the real Pareto front. In fact, every Pareto solution is closely approximated by at least one solution from the approximate Pareto front. In the following theorem, we further demonstrate that the upper bounds depends on the vector ϵ and on the starting time t_0:

THEOREM 1. *For all $p \in [\![1, |X^k|]\!]$, there exists $q \in [\![1, |\widetilde{X}^k|]\!]$ such that for all $k \in [\![1, |H|]\!]$ and $r \in [\![2,n]\!]$ (Given assumptions 1 and 2):*

$$t_{\widetilde{I}_q^k} \leq (1+\epsilon_1) t_{I_p^k} + \epsilon_1 t_0 \quad and \quad f_r(\widetilde{I}_q^k) \leq (1 + max(\epsilon_1, \epsilon_r)) f_r(I_p^k) \tag{13}$$

Theorem 1 also shows that there is a trade-off between the quality of approximation and the starting time. In fact, a later starting time ($t_0 > 0$) might result in reduced processing time (e.g. congestion might be avoided). However, a starting time $t_0 \neq 0$ triggers an additional small error $\epsilon_1 t_0$ in the first objective. Moreover, upper bounds of the other objectives are affected by ϵ_1. This is not a surprise since costs are time-dependent and the starting time of processing item i is affected by ϵ_1. Furthermore, in the special case where $t_0 = 0$, $max(\epsilon_1, \epsilon_r) = \epsilon_r$ for all $r \in [\![2,n]\!]$, and the number of objectives n is fixed, DP^ϵ belongs to the family of FPTAS algorithms.

6. Computational results

To validate the material presented in the previous sections, a time-dependent capacitated traveling salesman problem with time windows and multiple tours is considered. The described problem is relevant in the situation where a vehicle is required to fill up ATMs located at different places from a central bank. For security reasons, it is not allowed to carry a large amount of money. Consequently, the vehicle is forced to make several short tours during its operating period (*e.g.* a working day) going back and forth to the central bank. Similarly, in the case of food home delivery, tours are relatively short as products are perishable (*e.g.* should remain warm) and thus need to be delivered as soon as possible to their destinations (Azi et al. 2007).

A single vehicle performs several tours to serve a set $H = \{1, 2, ..., N\}$ of geographically dispersed customers. The vehicle has a finite capacity Q and is only available for a limited amount of time T. Moreover, the tours' duration is restricted (*e.g.* due to quality or security issues), to last not more than t^{lim} time units. The same single vehicle could of course do multiple tours, if time allows for this. Because of road congestion, travel times are time-dependent: depending on the departure time t^i at customer i, a different travel time $tt_{ij}(t^i)$ to customer j is incurred. Furthermore, all customers need to get delivered in their specific hard time windows. Service at customer i, with duration s_i, can only be started after its opening time t_i^l and no delivery is allowed after its closing

time t_j^u. We consider a situation with tight time windows, *i.e.* the size of time windows is small compared to travel time.

We consider a multi-objective cost function: simultaneously minimizing the total time traveled, including service time and waiting times at customers due to time windows, and maximizing the total demand fulfilled. Therefore, the vector costs is such that:

$$[c_{ij}(t^i)] = \begin{bmatrix} tt^*_{ij}(t^i) = tt_{ij}(t^i) + max(0, t_j^l - t_i - tt_{ij}(t^i)) + s_j \\ d_j \end{bmatrix} \tag{14}$$

in which d_j is the stationary demand of customer j.

For our numerical study, the customers' location as well as the related demand are taken from the Solomon's instances (Solomon (1987)). Congestion is taken into account, by assuming that the speed on each link is time-dependent and derive the travel time profile by using the relation $tt_{ij}(t) = \frac{d_{ij}}{v_{ij}(t)}$ where d_{ij}, distance between customers i and j, is computed based on Solomon's data sets. $v_{ij}(t)$ is the time-dependent speed by which the vehicle traverse the link between customers i and j. The resulting travel times satisfy both assumptions 1 and 2. The congested speed is $v_c = 30km/hr$, and the free speed is $v_f = 70km/hr$. Furthermore, we take $t^{lim} = \{500, 2000\}$, $Q = \{100, 200\}$ and $\epsilon = \{0.01, 0.05, 0.1, 0.3\}$. We consider instances with 100 and 300 customers and a planning horizon $T = 6000$ minutes.

To compare the different Pareto fronts generated, we introduce two measures defined by Zitzler et al. (2000): the two-set coverage metric and the average distance metric. The choice of the metrics is motivated by their intuitive explanation. The *two-set coverage* metric is defined as:

$$C(X \succ Y) = \frac{|\{y \in Y; \exists x \in X : x \text{ dominates } y\}|}{|Y|} \tag{15}$$

in which X and Y are two Pareto curves. The two-set coverage metric calculates the fraction of solutions in Y that are dominated by a solution in X.

The *average distance* metric is defined as:

$$M(X, Y) = \frac{1}{|X|} \sum_{x \in X} min\{\|x - y\|; y \in Y\} \tag{16}$$

The metric M reflects how distant two Pareto curves are from each other.

The algorithms DP and DP^ϵ are implemented on a Intel(R) Core(TM)2 CPU, 2.13 GHz, 3 GB of RAM computer, in a Matlab R2008b environment. All instances and software are available from the authors upon request.

6.1. Comparing DP and DP^ϵ

To illustrate the different Pareto fronts, Figure 4 depicts these corresponding to an instance with a relatively bad accuracy, namely the **R** instance with the input parameters $t^{lim} = 500$ and $Q = 100$. As expected, smaller values of the worst case precision ϵ result in a better ϵ-Pareto front, *i.e.* closer to the real one. Furthermore, the deviation of a ϵ-Pareto front from the optimal one increases in later iterations of DP (The size of boxes in Figure 3 increases). However, the deviation stays clearly within the worst case precision ϵ.

Based on the results given in Table 2 and 3, we conclude that the accuracy of DP^ϵ is excellent. For all instances, on average no more than 15% of the solutions generated by $DP^{0.01}$ are dominated by any of DP's solutions. Moreover, on average less than 32% of the $DP^{0.05}$ solutions and 35% of the $DP^{0.1}$ solutions are dominated by a DP solution. Furthermore, the distance of the 0.01-Pareto front from the optimal one is negligible (less than 2 in most cases). Moreover, even the distance of the 0.05-Pareto front and the 0.1-Pareto front from the optimal one can be consider as very small. We also observe that the fraction of DP^ϵ solutions dominated by a DP solution increases slightly when randomness is added to the location of customers.

Table 2 CPU and number of solutions generated.

Inst	t^{lim}	Q	DP		$DP^{0.01}$		$DP^{0.05}$		$DP^{0.1}$		$DP^{0.3}$	
			CPU	**Nb sol**	**CPU**	**Nb sol**	**CPU**	**Nb sol**	**CPU**	**Nb sol**	**CPU**	**Nb sol**
C100	**500**	**100**	1079	97	1091	97	947	68	775	46	640	23
		200	1182	97	1184	97	1080	62	814	44	670	23
	2000	**100**	1217	101	1232	102	1054	65	863	46	596	22
		200	1245	102	1270	102	1087	68	871	47	610	21
R100	**500**	**100**	1370	122	1376	119	1136	77	925	53	643	22
		200	1521	128	1453	122	1167	77	947	54	654	22
	2000	**100**	1762	143	1737	130	1273	74	999	53	596	22
		200	1555	136	1552	127	1210	76	944	53	638	22
RC100	**500**	**100**	1200	99	1195	97	935	55	748	42	541	19
		200	1300	103	1283	100	1023	56	831	40	587	19
	2000	**100**	1270	105	1261	98	969	56	757	40	510	20
		200	1335	107	1328	99	1036	60	809	40	575	19
C300	**500**	**100**	-	-	28770	88	25385	69	22877	46	15438	20
		200	-	-	24085	88	22208	67	20290	47	15545	19
	2000	**100**	-	-	-	-	-	-	21341	50	13810	24
		200	-	-	-	-	-	-	21895	51	15013	24
R300	**500**	**100**	-	-	32276	134	26670	87	21933	53	15814	23
		200	-	-	32862	129	29446	86	23005	57	16566	23
	2000	**100**	-	-	-	-	-	-	-	-	13628	21
		200	-	-	-	-	-	-	-	-	14152	24
RC300	**500**	**100**	-	-	-	-	27917	88	23646	48	16910	21
		200	-	-	-	-	28927	71	24002	48	17432	19
	2000	**100**	-	-	-	-	-	-	21600	48	15075	23
		200	-	-	-	-	-	-	23021	49	16954	23

Table 3 The metrics C and M.

Inst	t^{lim}	Q	$C(DP^{0.01})$	$M(DP^{0.01})$	$C(DP^{0.05})$	$M(DP^{0.05})$	$C(DP^{0.1})$	$M(DP^{0.1})$	$C(DP^{0.3})$	$M(DP^{0.3})$
C100	**500**	**100**	0.01	0.04	0.34	29.62	0.37	76.17	0.30	358.36
		200	0.01	0.04	0.29	28.89	0.30	90.69	0.28	401.62
	2000	**100**	0.05	0.10	0.32	21.7	0.39	69.91	0.32	321.84
		200	0.04	0.59	0.34	21.88	0.38	60.85	0.35	315.69
R100	**500**	**100**	0.43	4.61	0.40	21.81	0.38	45.23	0.42	276.37
		200	0.45	1.43	0.39	16.29	0.42	45.77	0.39	266.63
	2000	**100**	0.45	0.67	0.39	27.47	0.43	54.63	0.41	281.85
		200	0.22	0.67	0.45	27.47	0.28	54.63	0.42	285.70
RC100	**500**	**100**	0.06	0.65	0.30	27.47	0.33	54.63	0.30	286.38
		200	0.17	0.88	0.30	24.81	0.25	59.38	0.21	290.64
	2000	**100**	0.08	1.40	0.34	24.27	0.30	62.48	0.24	321.86
		200	0.11	0.73	0.20	18.85	0.30	62.48	0.33	315.71

6.2. Impact of the Worst Case Precision ϵ for DP^{ϵ}

The complexity of the DP^{ϵ} algorithm increases with $\frac{1}{\epsilon}$. Hence, choosing smaller worst case precisions results in higher computation times. Table 2 shows the impact of ϵ on the computation times of DP^{ϵ}. We observe that with an $\epsilon = 0.01$, there is a small increase of about 0.1% in computation times with regard to DP. In fact, for small values of ϵ not many solutions are deleted during the execution of DP^{ϵ} and therefore it is not compensated for the trimming time. Furthermore, for $\epsilon = 0.05$ and $\epsilon = 0.1$, CPU times are remarkably low, respectively 18% and 35% on average with regard to DP.

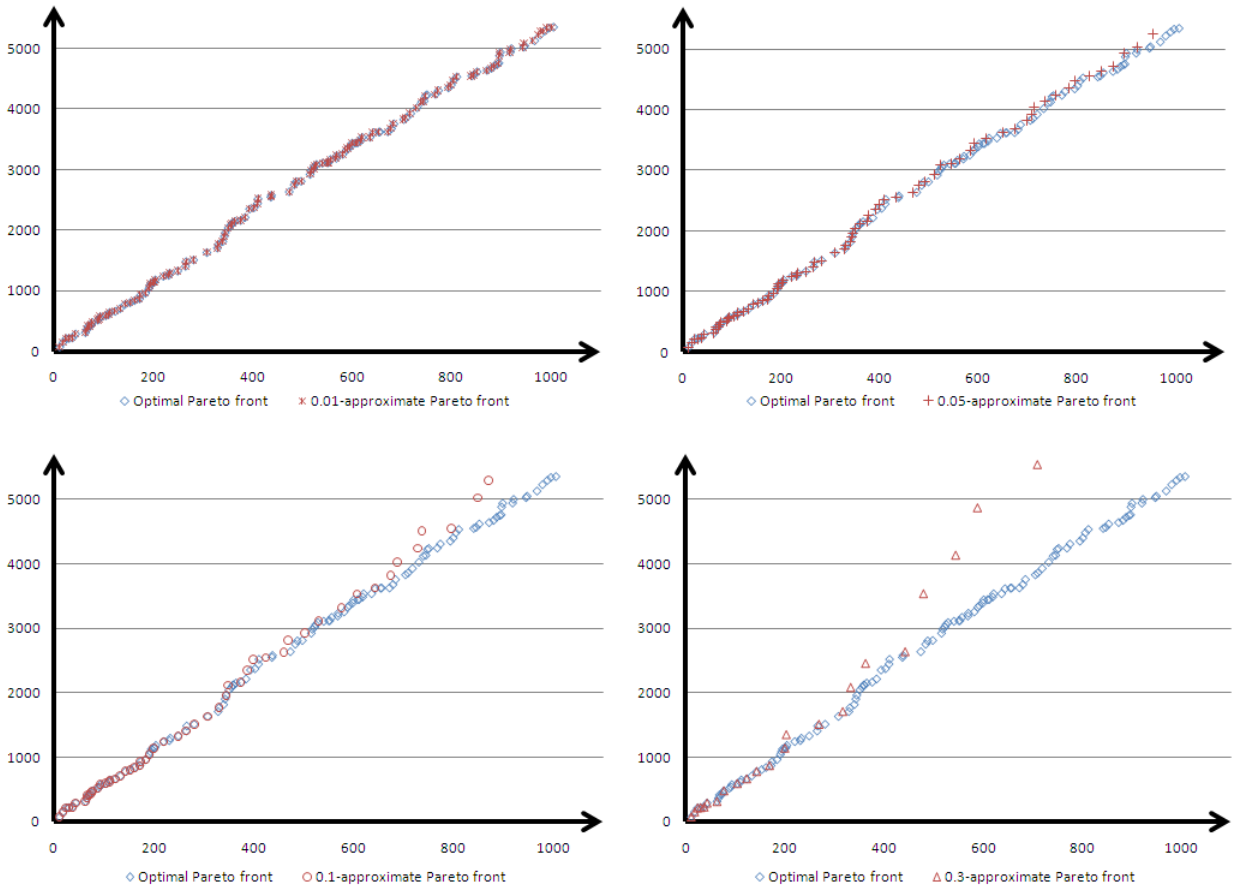

Figure 4 ϵ-**Pareto fronts.**

7. Conclusions

Multi-objective optimization problems are very challenging. They are at least as complex as their mono-objective version. Furthermore, cost and profits are hardly time-independent making the analysis and solution of multi-objective optimization problems even harder. Most existing algorithms fail to perform well in terms of both computation times and solutions quality. While exact algorithms can only deal with small problems, heuristics produce weak Pareto fronts that badly cover real Pareto fronts. In this paper, we propose a generic and flexible framework to deal with multi-objective time-dependent optimization problems. An efficient approximation based on dynamic programming is developed that generates good quality approximate Pareto fronts. Although they contain less solutions, the approximate Pareto fronts cover well the real Pareto fronts. The quality of the solutions can be decided on as the precision on each objective is an input to the algorithm and can be tuned by the decision maker. However, small precisions require more computation time. Moreover, per objective a different precision can be set. Therefore, larger errors might be allowed for less sensible objectives (*e.g.* with a flat cost structure). The approximation is tested on a time-dependent capacitated traveling salesman problem with time windows and multiple tours. Reasonably large instance with 300 customers are solved. The approximation reduces computation times considerably. Furthermore, it generates approximate Pareto fronts with few solutions, which makes the selection of a solution easier.

Acknowledgments
The research of Said Dabia has been funded by TRANSUMO, project number 10004927.

Appendix. Proofs

Proof of lemma 1: Note that if α and $\widetilde{\alpha}$ are two vectors in the same box, then:

$$\frac{\alpha_r}{\Delta_r} \leq \widetilde{\alpha}_r \leq \Delta_r \alpha_r \quad \text{for all} \quad r \in [\![1,n]\!] \tag{17}$$

Since Δ_r is very close to 1, α and $\widetilde{\alpha}$ are indeed very close to each other.
To prove lemma 1, we use induction on $k \in [\![1,|H|]\!]$.
From (17), we conclude that lemma 1 is true for $k=1$. Let us assume lemma 1 is true for $k-1$ and prove it for k.
Let $I_p^k \in \{X^k\}$ for some $p \in [\![1,|X^k|]\!]$. Per definition of the set X^k, I_p^k is feasible. Hence, there exists a feasible solution $I_u^{k-1} \in X^{k-1}$ for some $u \in [\![1,|X^{k-1}|]\!]$, and some item $i \in H$ such that $I_p^k = I_u^{k-1} \bigcup \{i\}$ and for all $r \in [\![2,n]\!]$:

$$t_{I_p^k} = t_{I_u^{k-1}} + c_1^i(t_0 + t_{I_u^{k-1}}) \quad \text{and} \quad f_r(I_p^k) = f_r(I_u^{k-1}) + c_r^i(t_0 + t_{I_u^{k-1}}) \tag{18}$$

On the other hand, because of the induction assumption, there exists $\widetilde{I}_v^{k-1} \in \widetilde{X}^{k-1}$ for some $v \in [\![1,|\widetilde{X}^{k-1}|]\!]$ such that for all $r \in [\![2,n]\!]$:

$$t_{\widetilde{I}_v^{k-1}} \leq \Delta_1^{k-1} t_{I_u^{k-1}} + (\Delta_1^{k-1} - 1)t_0 \quad \text{and} \quad f_r(\widetilde{I}_v^{k-1}) \leq max(\Delta_1^{k-1}, \Delta_r^{k-1}) f_r(I_u^{k-1}) \tag{19}$$

Furthermore, DP^ϵ generates the vector $[f_r(\widetilde{I}_v^{k-1}) + c_r^i(t_0 + t_{\widetilde{I}_v^{k-1}})]_{r=1}^{r=n}$ in the kth step. This vector point might be removed after trimming. However some vector $[f_r(\widetilde{I}_q^k)]_{r=1}^{r=n}$, located in the same box, should be left. From (17), we obtain for all $r \in [\![2,n]\!]$:

$$t_{\widetilde{I}_q^k} \leq \Delta_1(t_{\widetilde{I}_v^{k-1}} + c_1^i(t_0 + t_{\widetilde{I}_v^{k-1}})) \quad \text{and} \quad f_r(\widetilde{I}_q^k) \leq \Delta_r(f_r(\widetilde{I}_v^{k-1}) + c_r^i(t_0 + t_{\widetilde{I}_v^{k-1}})) \tag{20}$$

Because of the FIFO and the induction assumptions, we have:

$$t_0 + t_{\widetilde{I}_q^k} \leq \Delta_1(\Delta_1^{k-1}(t_0 + t_{I_u^{k-1}}) + c_1^i(\Delta_1^{k-1}(t_0 + t_{I_u^{k-1}}))) \tag{21}$$

and for all $r \in [\![2,n]\!]$:

$$f_r(\widetilde{I}_q^k) \leq \Delta_r(\Delta_r^{k-1} f_r(I_u^{k-1}) + c_r^i(\Delta_1^{k-1}(t_0 + t_{I_u^{k-1}}))) \tag{22}$$

Using assumption 2, we obtain:

$$t_0 + t_{\widetilde{I}_q^k} \leq \Delta_1^k(t_{I_u^{k-1}} + c_1^i(t_0 + t_{I_u^{k-1}})) + \Delta_1^k t_0 \tag{23}$$

and for all $r \in [\![2,n]\!]$:

$$f_r(\widetilde{I}_q^k) \leq \Delta_r(\Delta_r^{k-1} f_r(I_u^{k-1}) + \Delta_1^{k-1} c_r^i(t_0 + t_{I_u^{k-1}})) \tag{24}$$

Hence, for all $r \in [\![2,n]\!]$:

$$t_{\widetilde{I}_q^k} \leq \Delta_1^k t_{I_p^k} + (\Delta_1^k - 1)t_0 \quad \text{and} \quad f_r(\widetilde{I}_q^k) \leq max(\Delta_1^k, \Delta_r^k) f_r(I_p^k) \tag{25}$$

Proof of theorem 1: DP^ε and DP have at most $|H|$ iterations. Let us assume they converge after k iterations ($k \leq |H|$). According to lemma 1, For all $p \in [\![1,|X^k|]\!]$, there exists $q \in [\![1,|\widetilde{X}^k|]\!]$ such that for all $r \in [\![2,n]\!]$:

$$t_{\widetilde{I}_q^k} \leq \Delta_1^k t_{I_p^k} + (\Delta_1^k - 1)t_0 \quad \text{and} \quad f_r(\widetilde{I}_q^k) \leq max(\Delta_1^k, \Delta_r^k) f_r(I_p^k) \tag{26}$$

The sequences $\left(1 + \frac{\epsilon_r}{2|H|}\right)^{|H|}$ are increasing in $|H|$ and converges to $e^{\frac{\epsilon_r}{2}}$. Hence, for every $|H| \geq 1$:

$$\left(1 + \frac{\epsilon_r}{2|H|}\right)^{|H|} \leq e^{\frac{\epsilon_r}{2}} \tag{27}$$

Furthermore, for $0 < \epsilon_r < 1$ we have:

$$e^{\frac{\epsilon_r}{2}} \leq 1 + \epsilon_r \tag{28}$$

Therefore, we have:

$$t_{\widetilde{I}_q^k} \leq (1+\epsilon_1) t_{I_p^k} + \epsilon_1 t_0 \quad \text{and} \quad f_r(\widetilde{I}_q^k) \leq (1 + max(\epsilon_1, \epsilon_r)) f_r(I_p^k) \tag{29}$$

References

Azi, N., M. Gendreau, J. Y. Potvin. 2007. An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. *European Journal of Operational Research* **178** 755–766.

Camerini, P.M., C. Vercellis. 1984. The matroidal knapsack: a class of (often) well-solvable problems. *Operations Research Letters* **3** 157–162.

Carraway, R. L., T. L. Morin, H. Moskowitz. 1990. Generalized dynamic programming for multicriteria optimization. *European Journal of Operational Research* **44** 95–104.

Edgeworth, F. Y. 1881. Mathematical psychics: An essay on the application of mathematics to the moral sciences. *C. Kegan Paul and Co., London*.

Ehrgott, M. 2005. *Multicriteria Optimization*. Springer.

Farina, M., K. Deb, P. Amato. 2004. Dynamic multi-objective optimization problems: Test cases, approximations, and applications. *IEEE Trans. on Evolutionary Computation* **8** 425–442.

Hamacher, H. W., G. Ruhe. 1994. On spanning tree problems with multiple objectives. *Annals of Operations Research* **52** 209–230.

Ibarra, O. H., C. E. Kim. 1975. Fast approximation algorithms for the knapsack and sum of subset problems. *Journal of the ACM* **22** 463–468.

Ishibuchi, H., N. Tsukamoto, Y. Nojima. 2008. Behavior of evolutionary many-objective optimization. *Tenth International Conference on Computer Modeling and Simulation, UKSIM'2008*. 266–271.

Jozefowiez, N., F. Semet, E-G. Talbi. 2007. The bi-objective covering tour problem. *Computers and Operations Research* **34** 1929–1942.

Laumanns, M., L. Thiele, E. Zitzler. 2004. An adaptive scheme to generate the Pareto front based on the epsilon-constraint method. Tech. Rep. TIK-report 199, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich.

Lemesre, J., C. Dhaenens, E.-G. Talbi. 2006. Méthode parallèle par partitions: Passage d'une méthode exacte bi-objectif à une méthode exacte multi-objectif. *ROADEF'06 Proceedings*.

Lemesre, J., C. Dhaenens, E-G. Talbi. 2007a. An exact parallel method for a bi-objective permutation flowshop problem. *European Journal of Operational Research (EJOR)* **177** 1641–1655.

Lemesre, J., C. Dhaenens, E-G. Talbi. 2007b. Parallel partitioning method (PPM) : a new exact method to solve bi-objective problems. *Computers and Operational Research (COR)* **34** 2450–2462.

Liefooghe, A., M. Basseur, L. Jourdan, E-G. Talbi. 2007. Combinatorial optimization of stochastic multi-objective problems: An application to the flow-shop scheduling problem. *EMO'2007 Evolutionary Multi-criterion Optimization*, vol. LNCS 4403. Springer, 457–471.

Mandow, L., E. Millan. 1996. Goal programming and heuristic search. R. Caballero, F. Ruiz, R. Steuer, eds., *Second Int. Conf. on Multi-Objective Programming and Goal Programming MOPGP'96*. Springer-Verlag, Torremolinos, Spain, 48–56.

Miettinen, K. 1999. *Nonlinear multiobjective optimization*. Kluwer.

Papadimitriou, C. H., M. Yannakakis. 2002. On the approximability of trade-offs and optimal access of web services. *IEEE Symp. on Foundations of Computer Science* 86–92.

Pareto, V. 1896. Cours d'conomie politique. *Rouge, Lausanne, Switzerland*.

Sahni, K. S. 1976. Algorithms for scheduling independent tasks. *Journal of the ACM* **23** 116–127.

Sayin, S., S. Karabati. 1999. A bicriteria approach to the two-machine flow shop scheduling problem. *European Journal of Operational Research* **113** 435–449.

Sen, T., M. E. Raiszadeh, P. Dileepan. 1988. A branch and bound approach to the bicriterion scheduling problem involving total flowtime and range of lateness. *Management Science* **34** 254–260.

Serafini, P. 1986. Some considerations about computational complexity for multiobjective combinatorial problems. *In Jahn, J., Krabs, W. (Eds.), Recent Advances and Historical Development of Vector Optimization, LNMES. Springer-Verlag*, vol. 294. 222–232.

Solomon, M. M. 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research* **35** 254–265.

Steuer, R. 1986. *Multiple criteria optimization: Theory, computation and application*. Wiley, New York.

Stewart, B. S., C. C. White. 1991. Multiobjective A*. *Journal of the ACM* **38** 775–814.

Talbi, E-G. 2009. *Metaheuristics: from design to implementation*. Wiley.

Ulungu, E. L., J. Teghem. 1995. The two phase method: An efficient procedure to solve bi-objective combinatorial optimization problems. *Foundations of Computing and Decision Sciences*, vol. 20. 149–165.

Visée, M., J. Teghem, M. Pirlot, E. L. Ulungu. 1998. Two-phases method and branch and bound procedures to solve knapsack problem. *Journal of Global Optimization* **12** 139–155.

White, D. J. 1982. The set of efficient solutions for multiple-objectives shortest path problems. *Computers and Operations Research* **9** 101–107.

Woeginger, G. J. 2005. A comment on schedulling two parallel machines with capacity constraints. *Dicrete Optimization* **2** 269–275.

Zeleny, M. 1982. *Multiple criteria problem solving*. McGraw-Hill, New York.

Zitzler, E., K. Deb, L. Thiele. 2000. Comparaison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation* **8** 173–195.

Working Papers Beta 2009 - 2011

nr.	Year	Title	Author(s)
362	2011	Approximating Multi-Objective Time-Dependent Optimization Problems	Said Dabia, El-Ghazali Talbi, Tom Van Woensel, Ton de Kok
361	2011	Branch and Cut and Price for the Time Dependent Vehicle Routing Problem with Time Window	Said Dabia, Stefan Röpke, Tom Van Woensel, Ton de Kok
360	2011	Analysis of an Assemble-to-Order System with Different Review Periods	A.G. Karaarslan, G.P. Kiesmüller, A.G. de Kok
359	2011	Interval Availability Analysis of a Two-Echelon, Multi-Item System	Ahmad Al Hanbali, Matthieu van der Heijden
358	2011	Carbon-Optimal and Carbon-Neutral Supply Chains	Felipe Caro, Charles J. Corbett, Tarkan Tan, Rob Zuidwijk
357	2011	Generic Planning and Control of Automated Material Handling Systems: Practical Requirements Versus Existing Theory	Sameh Haneyah, Henk Zijm, Marco Schutten, Peter Schuur
356	2011	Last time buy decisions for products sold under warranty	M. van der Heijden, B. Iskandar
355	2011	Spatial concentration and location dynamics in logistics: the case of a Dutch provence	Frank P. van den Heuvel, Peter W. de Langen, Karel H. van Donselaar, Jan C. Fransoo
354	2011	Identification of Employment Concentration Areas	Frank P. van den Heuvel, Peter W. de Langen, Karel H. van Donselaar, Jan C. Fransoo
353	2011	BOMN 2.0 Execution Semantics Formalized as Graph Rewrite Rules: extended version	Pieter van Gorp, Remco Dijkman
352	2011	Resource pooling and cost allocation among independent service providers	Frank Karsten, Marco Slikker, Geert-Jan van Houtum
351	2011	A Framework for Business Innovation Directions	E. Lüftenegger, S. Angelov, P. Grefen
350	2011	The Road to a Business Process Architecture: An Overview of Approaches and their Use	Remco Dijkman, Irene Vanderfeesten, Hajo A. Reijers
349	2011	Effect of carbon emission regulations on transport mode selection under stochastic demand	K.M.R. Hoen, T. Tan, J.C. Fransoo G.J. van Houtum

348	2011	An improved MIP-based combinatorial approach for a multi-skill workforce scheduling problem	Murat Firat, Cor Hurkens
347	2011	An approximate approach for the joint problem of level of repair analysis and spare parts stocking	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
346	2011	Joint optimization of level of repair analysis and spare parts stocks	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
345	2011	Inventory control with manufacturing lead time flexibility	Ton G. de Kok
344	2011	Analysis of resource pooling games via a new extenstion of the Erlang loss function	Frank Karsten, Marco Slikker, Geert-Jan van Houtum
343	2011	Vehicle refueling with limited resources	Murat Firat, C.A.J. Hurkens, Gerhard J. Woeginger
342	2011	Optimal Inventory Policies with Non-stationary Supply Disruptions and Advance Supply Information	Bilge Atasoy, Refik Güllü, TarkanTan
341	2011	Redundancy Optimization for Critical Components in High-Availability Capital Goods	Kurtulus Baris Öner, Alan Scheller-Wolf Geert-Jan van Houtum
339	2010	Analysis of a two-echelon inventory system with two supply modes	Joachim Arts, Gudrun Kiesmüller
338	2010	Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh	Murat Firat, Gerhard J. Woeginger
335	2010	Attaining stability in multi-skill workforce scheduling	Murat Firat, Cor Hurkens
334	2010	Flexible Heuristics Miner (FHM)	A.J.M.M. Weijters, J.T.S. Ribeiro
333	2010	An exact approach for relating recovering surgical patient workload to the master surgical schedule	P.T. Vanberkel, R.J. Boucherie, E.W. Hans, J.L. Hurink, W.A.M. van Lent, W.H. van Harten
332	2010	Efficiency evaluation for pooling resources in health care	Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Nelly Litvak
		The Effect of Workload Constraints in Mathematical Programming Models for Production Planning	M.M. Jansen, A.G. de Kok, I.J.B.F. Adan

331	2010	Using pipeline information in a multi-echelon spare parts inventory system	Christian Howard, Ingrid Reijnen, Johan Marklund, Tarkan Tan
330	2010	Reducing costs of repairable spare parts supply systems via dynamic scheduling	H.G.H. Tiemessen, G.J. van Houtum
329	2010	Identification of Employment Concentration and Specialization Areas: Theory and Application	F.P. van den Heuvel, P.W. de Langen, K.H. van Donselaar, J.C. Fransoo
328	2010	A combinatorial approach to multi-skill workforce scheduling	Murat Firat, Cor Hurkens
327	2010	Stability in multi-skill workforce scheduling	Murat Firat, Cor Hurkens, Alexandre Laugier
326	2010	Maintenance spare parts planning and control: A framework for control and agenda for future research	M.A. Driessen, J.J. Arts, G.J. v. Houtum, W.D. Rustenburg, B. Huisman
325	2010	Near-optimal heuristics to set base stock levels in a two-echelon distribution network	R.J.I. Basten, G.J. van Houtum
324	2010	Inventory reduction in spare part networks by selective throughput time reduction	M.C. van der Heijden, E.M. Alvarez, J.M.J. Schutten
323	2010	The selective use of emergency shipments for service-contract differentiation	E.M. Alvarez, M.C. van der Heijden, W.H. Zijm
322	2010	Heuristics for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering in the Central Warehouse	B. Walrave, K. v. Oorschot, A.G.L. Romme
321	2010	Preventing or escaping the suppression mechanism: intervention conditions	Nico Dellaert, Jully Jeunet.
320	2010	Hospital admission planning to optimize major resources utilization under uncertainty	R. Seguel, R. Eshuis, P. Grefen.
319	2010	Minimal Protocol Adaptors for Interacting Services	Tom Van Woensel, Marshall L. Fisher, Jan C. Fransoo.
318	2010	Teaching Retail Operations in Business and Engineering Schools	Lydie P.M. Smets, Geert-Jan van Houtum, Fred Langerak.

317	2010	Design for Availability: Creating Value for Manufacturers and Customers	Pieter van Gorp, Rik Eshuis.
316	2010	Transforming Process Models: executable rewrite rules versus a formalized Java program	Bob Walrave, Kim E. van Oorschot, A. Georges L. Romme
315	2010	Getting trapped in the suppression of exploration: A simulation model	S. Dabia, T. van Woensel, A.G. de Kok
314	2010	A Dynamic Programming Approach to Multi-Objective Time-Dependent Capacitated Single Vehicle Routing Problems with Time Windows	
313	2010		
	2010		
312	2010	Tales of a So(u)rcerer: Optimal Sourcing Decisions Under Alternative Capacitated Suppliers and General Cost Structures	Osman Alp, Tarkan Tan
311	2010	In-store replenishment procedures for perishable inventory in a retail environment with handling costs and storage constraints	R.A.C.M. Broekmeulen, C.H.M. Bakx
310	2010	The state of the art of innovation-driven business models in the financial services industry	E. Lüftenegger, S. Angelov, E. van der Linden, P. Grefen
309	2010	Design of Complex Architectures Using a Three Dimension Approach: the CrossWork Case	R. Seguel, P. Grefen, R. Eshuis
308	2010	Effect of carbon emission regulations on transport mode selection in supply chains	K.M.R. Hoen, T. Tan, J.C. Fransoo, G.J. van Houtum
307	2010	Interaction between intelligent agent strategies for real-time transportation planning	Martijn Mes, Matthieu van der Heijden, Peter Schuur
306	2010	Internal Slackening Scoring Methods	Marco Slikker, Peter Borm, René van den Brink
305	2010	Vehicle Routing with Traffic Congestion and Drivers' Driving and Working Rules	A.L. Kok, E.W. Hans, J.M.J. Schutten, W.H.M. Zijm
304	2010	Practical extensions to the level of repair analysis	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
303	2010	Ocean Container Transport: An Underestimated and Critical Link in Global Supply Chain Performance	Jan C. Fransoo, Chung-Yee Lee
302	2010	Capacity reservation and utilization for a manufacturer with uncertain capacity and demand	Y. Boulaksil; J.C. Fransoo; T. Tan
300	2009	Spare parts inventory pooling games	F.J.P. Karsten; M. Slikker; G.J. van Houtum
299	2009	Capacity flexibility allocation in an outsourced	Y. Boulaksil, M. Grunow, J.C. Fransoo

		supply chain with reservation	
298	2010	An optimal approach for the joint problem of level of repair analysis and spare parts stocking	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
297	2009	Responding to the Lehman Wave: Sales Forecasting and Supply Management during the Credit Crisis	Robert Peels, Maximiliano Udenio, Jan C. Fransoo, Marcel Wolfs, Tom Hendrikx
296	2009	An exact approach for relating recovering surgical patient workload to the master surgical schedule	Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Wineke A.M. van Lent, Wim H. van Harten
295	2009	An iterative method for the simultaneous optimization of repair decisions and spare parts stocks	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
294	2009	Fujaba hits the Wall(-e)	Pieter van Gorp, Ruben Jubeh, Bernhard Grusie, Anne Keller
293	2009	Implementation of a Healthcare Process in Four Different Workflow Systems	R.S. Mans, W.M.P. van der Aalst, N.C. Russell, P.J.M. Bakker
292	2009	Business Process Model Repositories - Framework and Survey	Zhiqiang Yan, Remco Dijkman, Paul Grefen
291	2009	Efficient Optimization of the Dual-Index Policy Using Markov Chains	Joachim Arts, Marcel van Vuuren, Gudrun Kiesmuller
290	2009	Hierarchical Knowledge-Gradient for Sequential Sampling	Martijn R.K. Mes; Warren B. Powell; Peter I. Frazier
289	2009	Analyzing combined vehicle routing and break scheduling from a distributed decision making perspective	C.M. Meyer; A.L. Kok; H. Kopfer; J.M.J. Schutten
288	2009	Anticipation of lead time performance in Supply Chain Operations Planning	Michiel Jansen; Ton G. de Kok; Jan C. Fransoo
287	2009	Inventory Models with Lateral Transshipments: A Review	Colin Paterson; Gudrun Kiesmuller; Ruud Teunter; Kevin Glazebrook
286	2009	Efficiency evaluation for pooling resources in health care	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
285	2009	A Survey of Health Care Models that Encompass Multiple Departments	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
284	2009	Supporting Process Control in Business Collaborations	S. Angelov; K. Vidyasankar; J. Vonk; P. Grefen
283	2009	Inventory Control with Partial Batch Ordering	O. Alp; W.T. Huh; T. Tan
282	2009	Translating Safe Petri Nets to Statecharts in a Structure-Preserving Way	R. Eshuis
281	2009	The link between product data model and process model	J.J.C.L. Vogelaar; H.A. Reijers
280	2009	Inventory planning for spare parts networks with delivery time requirements	I.C. Reijnen; T. Tan; G.J. van Houtum
279	2009	Co-Evolution of Demand and Supply under Competition	B. Vermeulen; A.G. de Kok
			B. Vermeulen, A.G. de Kok

278	2010	Toward Meso-level Product-Market Network Indices for Strategic Product Selection and (Re)Design Guidelines over the Product Life-Cycle	R. Seguel, R. Eshuis, P. Grefen
277	2009	An Efficient Method to Construct Minimal Protocol Adaptors	
276	2009	Coordinating Supply Chains: a Bilevel Programming Approach	Ton G. de Kok, Gabriella Muratore
275	2009	Inventory redistribution for fashion products under demand parameter update	G.P. Kiesmuller, S. Minner
274	2009	Comparing Markov chains: Combining aggregation and precedence relations applied to sets of states	A. Busic, I.M.H. Vliegen, A. Scheller-Wolf
273	2009	Separate tools or tool kits: an exploratory study of engineers' preferences	I.M.H. Vliegen, P.A.M. Kleingeld, G.J. van Houtum
272	2009	An Exact Solution Procedure for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering	Engin Topan, Z. Pelin Bayindir, Tarkan Tan
271	2009	Distributed Decision Making in Combined Vehicle Routing and Break Scheduling	C.M. Meyer, H. Kopfer, A.L. Kok, M. Schutten
270	2009	Dynamic Programming Algorithm for the Vehicle Routing Problem with Time Windows and EC Social Legislation	A.L. Kok, C.M. Meyer, H. Kopfer, J.M.J. Schutten
269	2009	Similarity of Business Process Models: Metics and Evaluation	Remco Dijkman, Marlon Dumas, Boudewijn van Dongen, Reina Kaarik, Jan Mendling
267	2009	Vehicle routing under time-dependent travel times: the impact of congestion avoidance	A.L. Kok, E.W. Hans, J.M.J. Schutten
266	2009	Restricted dynamic programming: a flexible framework for solving realistic VRPs	J. Gromicho; J.J. van Hoorn; A.L. Kok; J.M.J. Schutten;

Working Papers published before 2009 see: http://beta.ieis.tue.nl