



ELSEVIER

European Journal of Operational Research 121 (2000) 16–31

EUROPEAN
JOURNAL
OF OPERATIONAL
RESEARCH

www.elsevier.com/locate/orms

Theory and Methodology

Shortest path problems with partial information: Models and algorithms for detecting dominance

Luis C. Dias ^{a,b,*}, João N. Clímaco ^{a,b}

^a *INESC, Rua Antero de Quental, 199, 3000 Coimbra, Portugal*

^b *Faculdade de Economia da Universidade de Coimbra, Av. Dias da Silva, 165, 3000 Coimbra, Portugal*

Received 22 May 1998; accepted 13 October 1998

Abstract

In this work we focus on partial information models for the well-known shortest path problem, where we consider multiple instances of values for the parameters that determine the cost of each arc. This allows coping with the uncertainty about the future, the imprecision of data, the arbitrariness of some options, the evolving values of the decision makers (DMs) and/or the multiplicity of DMs (in group decision making). This paper proves some results and presents detailed algorithms to identify the set of non dominated paths, a concept from decision theory under partial information. We first address problems with a finite set of instances, then problems with a general (eventually not discrete) set of instances and finally we study a particular case of the latter, which complies with a condition that may hold in some situations. To deal with these partial information problems we propose a new use for existing multicriteria algorithms based on the ranking of shortest paths. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Network programming; Partial information; Multicriteria analysis

1. Introduction

Decision aid based on mathematical methods requires choosing a model and obtaining solutions using algorithms. These models, and sometimes the behaviour of the algorithms, are characterized by a number r of parameters of a more or less subjective nature. Setting the parameter values is often a difficult task for the decision makers (DMs), since these values may be subject to sources of uncertainty, imprecision and inaccurate determination (see also Roy and Bouyssou, 1989; French, 1995):

- the value of each parameter may depend on the future state of the world, or it may result from aggregating several aspects with an impact on the parameter (there may hence be some arbitrariness in constructing parts of the model), or it may result from a measuring instrument or from a statistic measure (which usually involve some imprecision);

* Corresponding author. Tel.: +351 39 32689; fax: +351 39 24692; e-mail: ldias@inescc.pt

- the DMs may wish to avoid (or delay) the costs of knowing a parameter with more exactitude;
- some parameters have no real existence (because they are inherent to the decision aid method); furthermore, they may reflect the values of the DMs, which they may find difficult to express and that may change over time;
- there may be several DMs, in a group decision setting, who may not entirely agree on the values that each parameter should take.

In these situations we may have imprecise, incomplete (lack of), contradictory (among different sources) or controversial (among several DMs) information on the value of the parameters. We consider a shortest path problem where the DMs are not willing to fix a value for each of the r parameters that influence the costs of each of the arcs. Instead, they are able to specify a set of acceptable instances (r -tuples) of parameter values. These instances may form a finite set or a subset contained in \mathbb{R}^r (defined by constraints). We will refer to the resulting problem as one of decision making under multiple instances or partial information.

This paper builds on some thoughts on how multicriteria analysis is related to the shortest path problem with multiple instances (Dias and Clímaco, 1998). The next section presents partial information models, where multiple instances of the model are considered, each with different parameter values. In Section 3 we review some concepts and formulate the problem we are addressing. We will say that a path is dominated whenever there is a different path that is always better than or equal to the first for every instance of the model (without being equal to the first under all of them). Sections 4 and 5 present detailed algorithms for identifying non dominated paths in problems of real-world dimension, adapting multicriteria shortest path algorithms for this new purpose. The subsequent step of choosing among the non dominated paths is not addressed. We conclude with a summary and some ideas for future research.

2. Partial information models

DMs often use an “approximate” model or a probabilistic model when certain and precise values for the parameters are missing. The first approach, where a “most likely” value is assigned to each parameter, is complemented by sensitivity analysis, which determines how much can the input values change without leading to a (even if slightly) different solution. The second approach requires probability distributions (including correlation among parameters) for the unknown values and leads to the choice of a path based on its expected cost, and perhaps its variance. However, both approaches have the drawback of prematurely focusing on a single solution, which is as arbitrary as the “most likely” values or the probabilities chosen. The DMs will then tend to disregard other potentially interesting solutions. Moreover, often only one parameter is varied at a time when performing sensitivity analysis and correlations are neglected when performing simulations, which does not account for eventual interactions and interdependencies among the parameters.

A different idea is to admit multiple instances of the model, each defined by an acceptable combination of values for the parameters. The “robustness analysis” approach, where decision aid focuses on identifying conclusions that are valid for every instance of the model (see recent frameworks in Vincke (1997) and Roy (1997)), is then appropriate. A related framework is that of decision theory under partial information (good reviews are included in Hazen (1986), Weber (1987) and Rios Insua and French (1991)). This theory addresses the general problem of choosing among a set of actions, whose value depends on r parameters. There are multiple instances of values for these parameters, constrained to a subset of \mathbb{R}^r . In that context, a solution is said to be “dominated” whenever there is a different solution that is always better or equal than the first, regardless of the instance considered (without being equal to the first under all the instances). If a path p_0 is dominated, then there exists a path p_1 such that

the conclusion “ p_1 is not worse than p_0 ” is robust and there is at least one instance where p_1 is better than p_0 .

Consider a graph containing several nodes connected by oriented arcs. Each arc is associated with a cost (e.g. money, distance, duration), which is incurred when selecting it to be part of a path between two given nodes. Our goal is to find a path between two given nodes such that its cost is minimum (the shortest path). We consider that the model has r parameters (e.g. price of fuel, discount rate, outcome of labour negotiations) that characterize the cost of each arc and there are multiple acceptable instances (r -tuples) of values for these parameters. Throughout we will use the following notation:

$G(N, A)$	is a directed graph
$N = 1, \dots, n$	is the set of nodes and
A	is the set of arcs: $(i, j) \in A$ iff there is a directed arc from i to j
r	is the number of parameters that affect the cost associated with each arc
T	represents the set of the acceptable instances of r parameter values
$t \in T$	is an instance (r -tuple) of values for the parameters
$c_{ij}(t)$	is the cost of arc $(i, j) \in A$ for the instance $t \in T$
p	will usually denote a path, i.e. a sequence of nodes and arcs $\{1, (1, x), x, (x, y), y, \dots, z, (z, n), n\}$ from nodes 1 to n
P	represents the set of all paths from node 1 to node n
$v(p, t) = \sum_{(i,j) \in p} c_{ij}(t)$	is the cost of path p for the instance $t \in T$

Let us also define the following limits for the cost of each arc $(i, j) \in A$:

$$m_{ij} = \min\{c_{ij}(t) : t \in T\} \quad \text{and} \quad M_{ij} = \max\{c_{ij}(t) : t \in T\}.$$

This allows us to define upper and lower bounds for the cost of each path $p \in P$

$$v^m(p) = \sum_{(i,j) \in p} m_{ij} \quad \text{and} \quad v^M(p) = \sum_{(i,j) \in p} M_{ij}.$$

We consider that the set T may contain a finite number of instances (discrete T) or a region contained in \mathbb{R}^r (defined by mathematical constraints). These will be addressed in Sections 4 and 5, respectively. The latter type of model is more demanding in terms of computation, hence we will separately deal with a particular case where computation may be reduced (Section 5.2). This particular case consists of models that comply with the following condition:

Condition 1. Let the set of the $c_{ij}(\cdot)$ be such that

$$\forall p \in P, \exists t^m, t^M \in T : v(p, t^m) = v^m(p) \text{ and } v(p, t^M) = v^M(p).$$

This condition holds if for every path $p \in P$ there exists an instance such that its cost equals $v^m(p)$ and there is another instance such that its cost equals $v^M(p)$. This definition is equivalent to the following:

$$\forall p \in P, \bigcap_{(i,j) \in p} T^m(i, j) \neq \emptyset \quad \text{and} \quad \bigcap_{(i,j) \in p} T^M(i, j) \neq \emptyset,$$

where $T^m(i, j) = \{t \in T : c_{ij}(t) = m_{ij}\}$ (set of parameter r -tuples that minimise the cost of arc (i, j)) and $T^M(i, j) = \{t \in T : c_{ij}(t) = M_{ij}\}$ (set of parameter r -tuples that maximise the cost of arc (i, j)).

In other words, given any arc (i, j) from path $p \in P$, the worst (best) possible instance for (i, j) is also the worst (best) instance for all other arcs from path p . Although restrictive, this condition applies to many decision situations. We provide next two illustrative examples.

Example 2.1. Consider that the parameter r -tuples represent factors that influence the cost of an arc (i, j) , i.e., $c_{ij}(t) = c_{ij}(\alpha_1, \alpha_2, \dots, \alpha_r)$, such that if any α_k increases ($1 \leq k \leq r$), while the other parameters remain unchanged, then $c_{ij}(t)$ also increases. Each parameter α_k might represent fuel cost, labour cost, interest rate, etc. Furthermore, there is an instance where all the $\alpha_1, \alpha_2, \dots, \alpha_r$ are minimum and another instance where all the $\alpha_1, \alpha_2, \dots, \alpha_r$ are maximum (e.g. low inflation/high inflation scenarios). Such type of model would only violate Condition 1 if there existed an interdependency among the cost factors $\alpha_1, \alpha_2, \dots, \alpha_r$ such that an instance where these were all minimum (or maximum) would be inconsistent.

Example 2.2. Consider that the parameter r -tuples comprise one different parameter per arc in the graph, which defines the cost of that arc. In this model, let the value of the parameter associated with arc (i, j) be constrained to an interval bounded by λ_{ij} and v_{ij} , with $\lambda_{ij} \leq v_{ij}$. Then, $T = \prod_{(i,j) \in A} [\lambda_{ij}, v_{ij}] = \prod_{(i,j) \in A} \{c \in \mathbb{R}: \lambda_{ij} \leq c \leq v_{ij}\}$, where the operator \prod refers to a Cartesian product. This model allows the DMs to work with interval values for the parameters.

If the model has an infinite number of instances and does not comply with Condition 1, then the approach of Section 5.2 is no longer valid, but adapting a bicriteria shortest path algorithm may still be useful to reduce the amount of computation, as shown in Section 5.1.

3. Concepts of dominance and optimality

We recall in this section some concepts from decision theory under partial information, using the notation introduced above. In the following let $p, q \in P$.

- Absolute dominance (Δ_A -dominance)

$$p \Delta_A q \text{ iff } v(p, t) \leq v(q, t'), \quad \forall t, t' \in T \quad \text{and} \quad \exists t, t' \in T: v(p, t) < v(q, t').$$

- Usual dominance or Bernoulli dominance (Δ -dominance)

$$p \Delta q \text{ iff } v(q, t) \leq v(q, t), \quad \forall t \in T \quad \text{and} \quad \exists t \in T: v(p, t) < v(q, t).$$

Absolute dominance (Δ_A -dominance) occurs when a path is never worse than another, even if two different instances were used to evaluate the costs of the two paths, and there is at least a pair of instances such that the path that Δ_A -dominates is strictly better than the other. In contrast, Δ -dominance occurs when a path is never worse than the other for any given instance, and there is at least an instance such that the path that Δ -dominates is strictly better than the one that is Δ -dominated. The costs of the two paths are now computed using a common instance.

Whenever $p \Delta q$ (even if p does not Δ_A -dominate q), we deem that q is not worth for further consideration, assuming that the paths are always compared under common (although varied) circumstances. Hence we can exclude Δ -dominated paths, instead of considering the stronger Δ_A -dominance as a rejection rule. As a result there will be fewer paths to consider (those remaining in the set of non Δ -dominated paths). Consider now the following definitions:

- Optimality: p is optimal iff $v(p, t) \leq v(q, t), \forall q \in P - \{p\}, t \in T$.
- Potential optimality: p is potentially optimal iff $\exists t \in T: \forall q \in P - \{p\}, v(p, t) \leq v(q, t)$.

Let $\text{Opt}(p) \leq \{t \in T: v(p, t) \leq v(q, t), \forall q \in P - \{p\}\}$ denote the set of r -tuples for which p is a shortest path. Then, p is optimal iff $\text{Opt}(p) = T$ and p is potentially optimal iff $\text{Opt}(p) \neq \emptyset$.

Since the set of optimal paths is usually empty, the DMs do not have an evident best choice. Then, one could argue that the best solution should be searched among those potentially optimal. However, notice the following example: there are three paths – p_1, p_2 and p_3 – and two possible combinations of parameters (r -tuples) – t_1 and t_2 . Table 1 presents the cost $v(p_i, t_j)$ of each path ($i=1,2,3$) under each of

Table 1

	t_1	t_2
p_1	10	20
p_2	11	11
p_3	20	10

the two parameter combinations ($j=1,2$). We can observe that p_1 , p_2 and p_3 are non Δ -dominated, but only p_1 and p_3 are potentially optimal. However, few DMs would accept any method that did not consider p_2 .

Hence, we can state the problem we are addressing as follows: to find the set of non Δ -dominated paths, to present this set to the DMs and to let them choose the most preferred path. Since an optimal path will not usually exist, the choice of the best path depends on the DMs' personal preferences, values, attitudes and beliefs.

4. Discrete sets of parameter r -tuples

Suppose $T = \{t_1, t_2, \dots, t_K\}$, i.e. T is finite and discrete. We next show that already existent algorithms for the multicriteria shortest path problem can be used to determine which paths are non Δ -dominated. Indeed, the concept of Δ -dominance regarding the set of K instances is analogous to the well-known concept of dominance regarding K criteria, as if the cost of each path under each instance corresponded to a different criterion of evaluation.

Let us first consider the case where $T = \{t_1, t_2\}$. For instance, there may be two DMs, each one providing a "personal" set of values for the costs, or there may exist two possible scenarios for the future (e.g. pessimistic and optimistic scenarios), each with different costs for some arcs.

If the graph $G(N, A)$ contains cycles, the algorithm by Clímaco and Martins (1982) for bicriteria shortest path problems can be adapted. In case there are no cycles in the graph, the faster algorithm by Azevedo et al. (1991) can be used instead. Both algorithms are based on a procedure that computes a ranking of the paths from P , considering one of the criteria.

As an example, let us adapt the more general algorithm of Clímaco and Martins (1982) to our purpose. Without loss of generality, paths are ranked according to their cost under instance t_2 . The detailed algorithm is presented in Appendix A (Algorithm 1). It can be outlined as follows:

Let P_N contain the set of non Δ -dominated paths; initially, $P_N \leftarrow \emptyset$.

Let f_1 and f_2 contain the value under t_1 and t_2 , respectively, of the last path considered.

Let S be a working set containing paths that can be either Δ -dominated or non Δ -dominated.

Initiation: An upper bound u is set to the cost under t_2 of the path with lowest cost under t_1 . Clímaco and Martins (1982) show (in the multicriteria context) that any path p such that $v(p, t_2) > u$ must be Δ -dominated. If the r -tuple t_2 represents a pessimistic scenario, then the DMs may decrease u to the highest cost that they will tolerate.

Iteration 1: $p_1 \leftarrow$ shortest path under t_2 ; $S \leftarrow \{p_1\}$; $f_1 \leftarrow v(p_1, t_1)$; $f_2 \leftarrow v(p_1, t_2)$.

Iteration j : $p_j \leftarrow$ j th shortest path under t_2 ; IF $v(p_j, t_2) > u$ THEN stop;

OTHERWISE one of four cases may occur:

Case 1. [$v(p_j, t_1) = f_1$ and $v(p_j, t_2) = f_2$]. Path p_j joins other candidates, i.e. $S \leftarrow S \cup \{p_j\}$

Case 2. [$v(p_j, t_1) > f_1$] since $v(p_j, t_2) = f_2$, p_j must be Δ -dominated.

Case 3. [$v(p_j, t_1) < f_1$ and $v(p_j, t_2) = f_2$]. Paths in S are Δ -dominated by p_j . Hence, $S \leftarrow \{p_j\}$.

Case 4. [$v(p_j, t_1) < f_1$ and $v(p_j, t_2) > f_2$]. All paths in S are non Δ -dominated. Hence, $P_N \leftarrow P_N \cup S$ and $S \leftarrow \{p_j\}$.

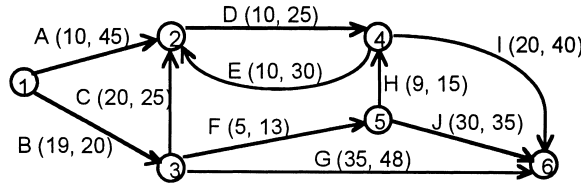


Fig. 1. Problem with two instances for costs.

Update f_1 and f_2 . Proceed to iteration $j + 1$ if there are more paths to examine.

This algorithm performs only a few instructions before moving on to the next path. It can therefore be quite fast, depending on the algorithm chosen for ranking the paths (e.g. see Azevedo et al. (1993) and Martins and Santos (1996) for efficient algorithms).

As an example, consider the graph depicted in Fig. 1, where we are interested in finding a shortest path from nodes 1–6. Each arc is associated with a label indicating its cost under t_1 and t_2 . Although in this example the cost under t_1 is never higher than that under t_2 , the algorithm does not require this condition. Step 1 finds the t_1 -shortest path, which is ADI, and sets $u = 110$ accordingly. The algorithm would continue as shown in Table 2. The set of the non Δ -dominated paths would be $\{p_1, p_2, p_3, p_5\}$.

Let us now consider the general case for a discrete T , where $T = \{t_1, t_2, \dots, t_K\}$. This may correspond to a group decision situation with K DMs. Since it could be difficult to find a consensus around exact or even approximate values for each parameter, it would be interesting to consider T as the union of the combinations of values suggested by them. Then, any Δ -dominated path could be unanimously discarded. A discrete T may also appear if we consider K scenarios for the future, each one implying different costs for some arcs. It may also happen that T was not discrete at first, but the DMs are satisfied by probing what happens for a finite number of sample r -tuples (“points”) of T , although noting that a path may be Δ -dominated considering the set of sample points but non Δ -dominated considering the original set.

If the graph $G(N, A)$ contains cycles, the algorithm by Clímaco and Martins (1980) for the general multicriteria shortest path problem can be adapted. However, this algorithm, which is an extension of the algorithm presented above, is slow for large problems. In case there are no cycles in the graph, the (faster) algorithm by Azevedo et al. (1991) can be used instead.

5. General sets of parameter r -tuples

5.1. General case

Let us now consider the possibility of T being any subset of R^r , either discrete or continuous (delimited by constraints provided by the DMs). In these cases we can test whether p Δ -dominates q for a given pair of paths $p, q \in P$ as follows:

Table 2

j	Path p_j	$v(p_j, t_1)$	$v(p_j, t_2)$	Remarks
1	BFJ	54	68	$P_N \leftarrow \emptyset; S \leftarrow \{p_1\}; f_1 \leftarrow 54; f_2 \leftarrow 68$
2	BG	54	68	Case 1 $\Rightarrow S \leftarrow \{p_1, p_2\}$
3	BFHI	53	88	Case 4 $\Rightarrow f_1 \leftarrow 53; f_2 \leftarrow 88; P_N \leftarrow \{p_1, p_2\}; S \leftarrow \{p_3\}$
4	BCDI	69	110	Case 2 \Rightarrow (do nothing: move to next path)
5	ADI	40	110	Case 4 $\Rightarrow f_1 \leftarrow 40; f_2 \leftarrow 110; P_N \leftarrow P_N \cup S = \{p_1, p_2, p_3\}; S \leftarrow \{p_5\}$
6	BFHEDI	73	143	$v(p_6, t_2) > u \Rightarrow$ STOP. $P_N \leftarrow P_N \cup S = \{p_1, p_2, p_3, p_5\}$

(Exhaustive test) let d^* be the optimal value of

$$\max d(t) = v(p, t) - v(q, t)$$
s.t. $t \in T$;
then $p \Delta q$ iff $d^* < 0$ or [$d^* = 0$ and $\exists t \in T: v(p, t) < v(q, t)$].

If T is delimited by constraints, this test amounts to a mathematical program, which may be costly to compute. Therefore, it may be worthwhile to spend some computational effort in trying to identify the paths that are surely Δ -dominated and those that are surely non Δ -dominated. Then, we would only perform the exhaustive test to see if the remaining paths are dominated or not. We only require a simpler test for the effort to be worthwhile.

Let us first note that, by definition (recall Section 2), $v^M(\cdot)$ and $v^m(\cdot)$ are such that

$$v^m(p) \leq v(p, t) \leq v^M(p), \quad \forall t \in T, p \in P. \quad (1)$$

Let us now define the following binary relations for any pair of paths $(p, q) \in P^2$:

$$p \ll q \text{ iff } v^M(p) < v^m(q),$$

$$p < q \text{ iff } [v^m(p) = v^m(q) \text{ and } v^M(p) < v^M(q)] \text{ or } [v^m(p) < v^m(q) \text{ and } v^M(p) = v^M(q)],$$

$$p < \approx q \text{ iff } v^m(p) \leq v^m(q) \text{ and } v^M(p) \leq v^M(q).$$

It is easy to see that \ll and $<$ are irreflexive and transitive relations and that $< \approx$ is a reflexive transitive relation. Note also that $p \ll q \Rightarrow p < q \Rightarrow p < \approx q$.

Proposition 1. Let $p, q \in P$. Then $p \ll q \Rightarrow p \Delta q$.

Proof. $p \ll q \Rightarrow v^M(p) < v^m(q)$ (by definition). From Eq. (1) we guarantee that $v(p, t) \leq v^M(p)$ and $v^m(q) \leq v(q, t), \forall t \in T$. Hence, $v(p, t) \leq v^M(p) < v^m(q) = v(q, t), \forall t \in T \Rightarrow v(p, t) < v(q, t), \forall t \in T \Rightarrow p \Delta q$. \square

A similar result for a relation weaker than \ll cannot be proven for the general case. To see this, consider the following counter-example. Suppose we are interested in a path from nodes 1–5 and are considering path $p = \{1, (1,2), 2, (2,3), 3, (3,5), 5\}$ and path $q = \{1, (1,4), 4, (4,5), 5\}$. Let $T = \{t_1, t_2, t_3\}$ and consider the costs shown in Table 3. Then,

$$\begin{aligned} v(p, t_1) &= 20, & v(p, t_2) &= 30, & v(p, t_3) &= 30, & v^m(p) &= 10, & v^M(p) &= 60, \\ v(q, t_1) &= 30, & v(q, t_2) &= 34, & v(q, t_3) &= 35, & v^m(q) &= 9, & v^M(q) &= 55. \end{aligned}$$

We conclude that for this example $p \Delta q$ and not $(p \ll q)$. In fact, $q \ll p$! Therefore, we cannot affirm that $p < q \Rightarrow p \Delta q$ nor that $p \approx q \Rightarrow p \Delta q$.

In the absence of further restrictions on the problem, the approach suggested by these conclusions is to adapt bicriteria shortest path algorithms to consider \ll -dominance, where

$$p \text{ is } \ll\text{-dominated iff } \exists q \in P: q \ll p.$$

Table 3

(i,j)	$c_{ij}(t_1)$	$c_{ij}(t_2)$	$c_{ij}(t_3)$
(1,2)	0	0	20
(2,3)	0	20	0
(3,5)	20	10	10
(1,4)	5	30	15
(4,5)	25	4	20

Among the non \ll -dominated paths we will find all the non Δ -dominated paths along with some Δ -dominated paths.

An algorithm to find non dominated paths could compare each path in P (assume that there are n_p paths) with all the other, which requires up to $n_p(n_p - 1)$ exhaustive tests. However, thanks to Proposition 1, this can be improved if we use an algorithm based on the one that Clímaco and Martins (1982) have built for the bicriteria shortest path problem. Algorithm 2, presented in Appendix A, requires a shortest paths ranking algorithm, instead of an algorithm that provides all paths in no particular order.

This second algorithm starts by finding the shortest path according to the $v^M(\cdot)$ cost and sets the upper bound u to that minimum $v^M(\cdot)$ cost. Afterwards, paths are considered in ascending order of their $v^m(\cdot)$ cost until the upper bound is reached. For any path p , $v^m(p) > u$ means that p is \ll -dominated and, by Proposition 1, p is Δ -dominated. Given $n'_p = n_p$ paths with a $v^m(\cdot)$ cost less than (or equal to) u , the number of exhaustive tests to perform does not exceed $n'_p(n'_p - 1)$. This can mean a significant improvement if n'_p is much less than n_p .

5.2. A particular case

If Condition 1 holds (see Section 2), then the approach suggested in Section 5.1 may be extended to reduce further the number of exhaustive tests to perform. We start by proving some results.

Proposition 2a. *Let $p, q \in P$ and Condition 1 be satisfied. Then $v^m(q) < v^m(p) \Rightarrow \sim (p \Delta q)$.*

Proof. From Condition 1 we know that $\{t \in T: v(q, t) = v^m(q)\}$ is not empty. Let $t_q^m \in \{t \in T: v(q, t) = v^m(q)\}$ (a best-case instance for q). Then, $v^m(p) > v^m(q) \iff v^M(q) > v(q, t_q^m)$. Since $v(p, t_q^m) \geq v^m(p)$, we conclude that $v(p, t_q^m) > v(q, t_q^m) \Rightarrow \sim (p \Delta q)$. \square

Proposition 2b. *Let $p, q \in P$ and Condition 1 be satisfied. Then $v^M(q) < v^M(p) \Rightarrow \sim (p \Delta q)$.*

Proof. From Condition 1 we know that $\{t \in T: v(p, t) = v^M(p)\}$ is not empty. Let $t_p^M \in \{t \in T: v(p, t) = v^M(p)\}$ (a worst-case instance for p). Then, $v^M(p) > v^M(q) \iff v(p, t_p^M) > v^M(q)$. Since $v^M(q) = v(q, t_p^M)$, we conclude that $v(p, t_p^M) > v(q, t_p^M) \Rightarrow \sim (p \Delta q)$. \square

Proposition 3. *Let $p, q \in P$ and Condition 1 be satisfied. Then $p \Delta q \Rightarrow p < \approx q$.*

Proof. We will show that $\sim (p < \approx q) \Rightarrow \sim (p \Delta q)$. Since either $v^m(p) > v^m(q)$ is true or $v^M(p) > v^M(q)$ is true, we conclude from Propositions 2a and b that $\sim (p \Delta q)$, i.e., p cannot Δ -dominate q . \square

Definition. p is $< \approx$ -dominated iff $\exists q \in P: q < \approx p$.

Proposition 4 (A corollary from Proposition 3). *Let $p \in P$ and Condition 1 be satisfied. Then, p is non $< \approx$ -dominated $\Rightarrow p$ is non Δ -dominated.*

Proof. From Proposition 3, $\exists q \in P: q \Delta p \Rightarrow \exists q \in P: q < \approx p$. Hence, $\sim (\exists q \in P: q < \approx p) \Rightarrow \sim (\exists q \in P: q \Delta p)$, i.e. p is non $< \approx$ -dominated $\Rightarrow p$ is non Δ -dominated. \square

One cannot prove that $p < \approx q \Rightarrow p \Delta q$, $p \Delta q \Rightarrow p < q$ nor $p \Delta q \Leftarrow p < q$. To see this, let us consider the example in Table 4, where we are considering paths from nodes 1–5 – $p_1 = \{1, (1, 2), 2, (2, 5), 5\}$, $p_2 = \{1, (1, 3), 3, (3, 5), 5\}$ and $p_3 = \{1, (1, 4), 4, (4, 5), 5\}$ – and instances $T = \{t_1, t_2, t_3\}$. Then,

Table 4

(i,j)	$c_{ij}(t_1)$	$c_{ij}(t_2)$	$c_{ij}(t_3)$
(1,2)	5	10	15
(2,5)	5	10	15
(1,3)	5	15	15
(3,5)	5	15	15
(1,4)	10	12	20
(4,5)	10	12	20

$$\begin{aligned}
v(p_1, t_1) &= 10, & v(p_1, t_2) &= 20, & v(p_1, t_3) &= 30, & v^m(p_1) &= 10, & v^M(p_1) &= 30, \\
v(p_2, t_1) &= 10, & v(p_2, t_2) &= 30, & v(p_2, t_3) &= 30, & v^m(p_2) &= 10, & v^M(p_2) &= 30, \\
v(p_3, t_1) &= 20, & v(p_3, t_2) &= 24, & v(p_3, t_3) &= 40, & v^m(p_3) &= 20, & v^M(p_3) &= 40.
\end{aligned}$$

Notice that $p \llapprox q \Rightarrow p \Delta q$ is false because $p_2 \llapprox p_3$ but $\sim(p_2 \Delta p_3)$ (p_3 is better under t_2). The same pair of paths shows that $p < q \Rightarrow p \Delta q$ is also false. It can only be proven that

$$p < q \Rightarrow \sim(q \llapprox p) \Rightarrow \sim(q \Delta p).$$

Finally, notice that $p \Delta q \Rightarrow p < q$ is false because $p_1 \Delta p_2$ but $\sim(p_1 < p_2)$.

These results suggest the partition of the set P into three subsets: paths that are surely Δ -dominated (which can be discarded), paths surely non Δ -dominated (which should be kept) and paths that are contenders for non Δ -dominance. A new algorithm can hence be used whenever Condition 1 holds, to:

- mark all non \llapprox -dominated paths as surely non Δ -dominated (according to Proposition 4);
- keep \llapprox -dominated paths as potentially non Δ -dominated;
- exclude \ll -dominated paths as surely Δ -dominated (according to Proposition 1).

We must only use the exhaustive test to discover whether \llapprox -dominated (but non \ll -dominated) paths are Δ -dominated or not. The number of (exhaustive) tests to perform can be further reduced according to Proposition 3: for any $p, q \in P$, if p does not \llapprox -dominate q we will know that p also does not Δ -dominate q .

The algorithm that we outline next scans the first n'_p shortest paths in a ranking considering the lower bound for the cost $v^m(\cdot)$. It stops as soon as this lower bound exceeds a cost u equal to the lowest upper bound $v^M(\cdot)$. A detailed algorithm is presented in Appendix A (Algorithm 3). Consider the following sets and variables:

Let P_N contain the set of non Δ -dominated paths;

Let L contain the last paths considered that are candidate to be non Δ -dominated;

Let f_m and f_M contain the costs, considering $v^m(\cdot)$ and $v^M(\cdot)$, respectively, of the paths in L ;

Let $P^=$ contain the paths that are \llapprox -dominated but may be non Δ -dominated, with $v^m(\cdot) = f_m$;

Let $P^>$ contain the paths that are \llapprox -dominated but may be non Δ -dominated, with $v^m(\cdot) > f_m$.

In the beginning, all the sets are empty. The algorithm sets the upper bound u to the minimum $v^M(\cdot)$ cost. In iteration 1, it finds the shortest path p_1 considering $v^m(\cdot)$ and places it in the set L , setting f_m and f_M to the values $v^m(p_1)$ and $v^M(p_1)$, respectively.

In iteration j ($j > 1$), the algorithm finds the j th shortest path p_j considering $v^m(\cdot)$. If either there is no j th shortest path (there are no more paths in the graph) or $v^m(p_j)$ exceeds the bound u , then the algorithm stops. In this case, the paths in sets P_N and L must be non Δ -dominated, whereas paths in $P^=$ and $P^>$ are either non Δ -dominated or dominated by paths in L (some exhaustive tests have to be performed). Otherwise, one of the following cases applies:

Case 1. $v^m(p_j) = f_m$ and $v^M(p_j) = f_M$, which means that path $p_j \llapprox$ -dominates the paths in L and paths in $L \llapprox$ -dominate p_j . Check if any path in $L \Delta$ -dominates p_j . If none does, check if $p_j \Delta$ -dominates any path in L and let $L \leftarrow L \cup p_j$.

Case 2. $v^m(p_j) = f_m$ and $v^M(p_j) < f_M$, which means that path p_j $<\approx$ -dominates the paths in L . Check if any path in L Δ -dominates any path in $P^=$. Let $P^= \leftarrow P^= \cup L$. Let $L \leftarrow p_j$ and set $f_m \leftarrow v^m(p_j)$.

Case 3. $v^m(p_j) = f_m$ and $v^M(p_j) > f_M$, which means that paths in L $<\approx$ -dominate p_j . Check if any path in $P_N \cup P^=$ Δ -dominates path p_j . If none does, check if p_j Δ -dominates any path in $P^=$ and let $P^= \leftarrow P^= \cup p_j$.

Case 4. $v^m(p_j) > f_m$ and $v^M(p_j) < f_M$, which means that paths in L are non Δ -dominated. Check if any path in L Δ -dominates any path in $P^=$ or $P^>$. Let $P_N \leftarrow P_N \cup L \cup P^= \cup \{p \in P^> : v^m(p) < v^m(p_j)\}$. Let $P^= \leftarrow \{p \in P^> : v^m(p) = v^m(p_j)\}$ and then set $P^> \leftarrow \emptyset$. Let $L \leftarrow p_j$ and update f_m and f_M to $v^m(p_j)$ and $v^M(p_j)$, respectively.

Case 5. $v^m(p_j) > f_m$ and $v^M(p_j) \geq f_M$, which means that p_j is $<\approx$ -dominated. Check if any path in $P_N \cup P^= \cup P^>$ Δ -dominates path p_j . If none does, check if p_j Δ -dominates any path in $P^>$ and let $P^> \leftarrow P^> \cup \{p_j\}$.

The minimum number of tests to perform when using Algorithm 3 is zero, in the case where all paths are non $<\approx$ -dominated as in Fig. 2(a). The maximum number of tests to perform is $n'_p (n'_p - 1)$, where n'_p is the number of paths such that its $v^m(\cdot)$ cost is less than or equal to the lowest $v^M(\cdot)$ cost. This is the number of tests performed by the algorithm in Section 5.1. This worst case occurs when all paths have equal $v^m(\cdot)$ and $v^M(\cdot)$ costs, as in Fig. 2(c). Most of the times one will be in an intermediate situation, as depicted in Fig. 2(b). The next two examples illustrate the execution of this algorithm.

Example 5.1. Consider again the graph in Fig. 1 and the following situation. Let costs be uncertain, depending on two parameters concerning the near future. Parameter α_1 represents the evolution of fuel costs, which may vary up to $\pm 5\%$ around the current level. Parameter α_2 represents the evolution of labour costs, which may grow up to 5% above the current level. The cost of an arc (in the near future) is given by the expression

$$c_{ij}(\alpha_1, \alpha_2) = B(i, j) + k_1(i, j)\alpha_1 + k_2(i, j)\alpha_2, \quad \alpha_1 \in [-0.05, 0.05] \quad \text{and} \quad \alpha_2 \in [0, 0.05],$$

where $B(i, j)$, $k_1(i, j)$ and $k_2(i, j)$ are constants associated with each arc. Table 5 presents the value of these constants, as well as the minimum and maximum possible costs for each arc.

Condition 1 is satisfied: parameter 2-tuple $(\alpha_1, \alpha_2) = (-0.05, 0)$ is a best case for all arcs and $(\alpha_1, \alpha_2) = (0.05, 0.05)$ is a worst case for all arcs. The algorithm starts by finding a path that minimises $v^M(\cdot)$. In this case it finds paths BFJ or BG, where $v^M(\cdot) = 68$. It then sets the upper bound $u = 68$: any path with a $v^m(\cdot)$

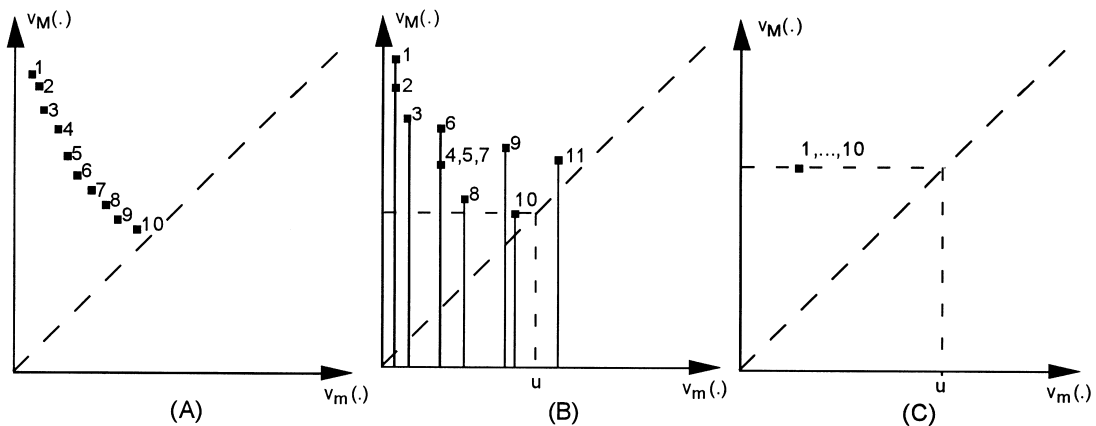


Fig. 2. Worst case (a), usual case (b) and best case (c) examples for algorithm.

Table 5

Arc (i,j)	$B(i,j)$	$k_1(i,j)$	$k_2(i,j)$	m_{ij}	M_{ij}
<i>A</i>	10.115	2.3	695.4	10	45
<i>B</i>	19.205	4.1	11.8	19	20
<i>C</i>	20.29	5.8	88.4	20	25
<i>D</i>	10.16	3.2	293.6	10	25
<i>E</i>	10.1	2	396	10	30
<i>F</i>	5.105	2.1	155.8	5	13
<i>G</i>	35.375	7.5	245	35	48
<i>H</i>	9.135	2.7	114.6	9	15
<i>I</i>	20.255	5.1	389.8	20	40
<i>J</i>	30.365	7.3	85.4	30	35

higher than u is Δ -dominated. The algorithm (see Table 6) avoids many exhaustive tests, exploiting Propositions 2a and b as well as the transitivity of the Δ -dominance relation.

Example 5.2. Consider the situation depicted in Fig. 2(b). We next show how the algorithm would operate if all paths (except the eleventh) were non Δ -dominated. Notice that this implies more tests than if some paths were discarded along its execution. First, the algorithm would determine the upper bound u as the minimum $v^M(\cdot)$, which occurs for the 10th path. Then, it would examine one path at a time in non decreasing order of their $v^m(\cdot)$. Suppose that paths are numbered in the figure according to the order they would appear (when several paths have equal $v^m(\cdot)$, their order of appearance is undetermined a priori). The algorithm would then continue as shown in Table 7. At most 14 exhaustive tests would have to be performed, compared to the $10(10-1)=90$ tests performed by the algorithm of Section 5.1.

6. Summary and future research

We addressed the shortest path problem with multiple instances for the parameters that determine the cost of each arc. We considered models where the DMs define a set of multiple acceptable values of parameters (model instances). Within this framework we will not usually find a path that is optimal under all circumstances. We suggest as a first step to exclude only the dominated paths (in the sense that there is a better path for every acceptable instance) while keeping all the non dominated ones for further consideration.

We have shown that already existent algorithms that deal with to the multicriteria shortest path problems can be adapted when the DMs only consider a discrete and finite set of K instances. As an example, we provided an algorithm for the case where $K=2$, which can be extended for $K > 2$ but at greater computational cost. When considering the case where the set of instances is defined by constraints, the test for non dominance may require an amount of computation that is not negligible. For example, it can

Table 6

j	Path p_j	$v^m(p_j)$	$v^M(p_j)$	Case	Remarks
1	ADI	40	110	–	$L \leftarrow \{p_1\}; f_m \leftarrow 40; f_M \leftarrow 110; P_N \leftarrow \emptyset; P^= \leftarrow \emptyset; P^> \leftarrow \emptyset.$
2	BFHI	53	88	4	$P_N \leftarrow P_N \cup L = \{p_1\}; L \leftarrow \{p_2\}; f_m \leftarrow 53; f_M \leftarrow 88.$
3	BFJ	54	68	4	$P_N \leftarrow P_N \cup L = \{p_1, p_2\}; L \leftarrow \{p_3\}; f_m \leftarrow 54; f_M \leftarrow 68.$
4	BG	54	68	1	not $p_3 \Delta p_4$ and not $p_4 \Delta p_3; L \leftarrow \{p_3, p_4\}.$
5	ADEDI	60	165	5	Clearly, $p_1 \Delta p_5 \Rightarrow \{p_5\}$ is discarded.
6	BCDI	69	110	–	$v^m(p_6) > u \Rightarrow$ exit outer loop; $P_N \leftarrow P_N \cup L \cup P^= \cup P^> = \{p_1, p_2, p_4, p_3\}.$

Table 7

<i>j</i>	Case	P_N	L	$P^=$	$P^>$	f_m	f_M	Tests
1	–	\emptyset	p_1	\emptyset	\emptyset	$v^m(p_1)$	$v^M(p_1)$	None
2	2	\gg	p_2	p_1	\gg	$v^m(p_2)$	$v^M(p_2)$	None
3	4	p_1, p_2	p_3	\emptyset	\gg	$v^m(p_3)$	$v^M(p_3)$	$p_2 \Delta p_1?$
4	4	p_1, p_2, p_3	p_4	\gg	\gg	$v^m(p_4)$	$v^M(p_4)$	None
5	1	\gg	p_4, p_5	\gg	\gg	\gg	\gg	$p_4 \Delta p_5?, p_5 \Delta p_4?$
6	3	\gg	\gg	p_6	\gg	\gg	\gg	None
7	1	\gg	p_4, p_5, p_7	\gg	\gg	\gg	\gg	$p_4 \Delta p_7?, p_5 \Delta p_7?, p_7 \delta p_5?, p_7 \Delta p_4?$
8	4	$p_1, p_2, p_3, p_4, p_5, p_6, p_7$	p_8	\emptyset	\gg	$v^m(p_8)$	$v^M(p_8)$	$p_4 \Delta p_6?, p_5 \Delta p_6?, p_7 \Delta p_6?$
9	5	\gg	\gg	\gg	p_9	\gg	\gg	$p_4 \Delta p_9?, p_5 \Delta p_9?, p_7 \Delta p_9?$
10	4	$p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9$	p_{10}	\gg	\emptyset	$v^m(p_{10})$	$v^M(p_{10})$	$p_8 \Delta p_9?$
11	–	$p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}$	–	–	–			None

correspond to a mathematical program. To lower the number of tests to perform, we adapted a bicriteria shortest path algorithm. Afterwards, after proving some propositions, we have shown that the number of tests can be further reduced if the problem complies with a condition that holds in some real-world situations. This allowed us to build a specific algorithm for these problems.

Although we have not implemented the proposed algorithms in any machine, we can affirm that they allow to solve problems of real-world dimension in reasonable time as they can use a very fast shortest paths ranking algorithm and, for each path in the ranking, they only perform a few operations. For instance, the Martins and Santos (1996) algorithm is reported to rank half a million paths in a 10 000 node network in less than a second. Algorithms that rank loopless paths (in graphs that may contain cycles) would also be useful, since non dominated paths cannot have cycles. This problem is considered harder to solve, but recent advances are promising (e.g. see Martins et al., 1998). The homepage of Ernesto Q.V. Martins (<http://www.mat.uc.pt/~eqvm>) presents his papers along with computer codes and results.

Future research should focus on the second step of the problem, which consists of helping the DMs in making a choice when faced with a large number of non dominated paths. In what regards this step, much of the research concerning multicriteria problems may be valuable. Several issues that could be addressed are the following:

- search for optimal paths where the optimality condition is relaxed, as implied in the work of Vincke (1997);
- search for the path with lowest worst-case cost (when Condition 1 does not apply, or else the problem is trivial), as suggested by Kouvelis and Yu (1997);
- construction of functions that attribute a value for each path (for instance, the result of a balance between the potential benefit and potential regret implied by choosing that path);
- construction of binary relations among pairs of paths for posterior exploitation. For instance considering the “volume” of the parameter region that supports each of the paths;
- human–machine dialogue protocols in an interactive decision support system.

Acknowledgements

This work has been partially supported by PRAXIS XXI Technological Development and Scientific Research contract number 2/2.1/MAT/465/94.

Appendix A

Algorithm 1

Let P_N contain the set of non Δ -dominated paths; Let f_1 and f_2 contain the value under t_1 and t_2 , respectively, of the last path considered; Let S be a working set containing paths that can be either Δ -dominated or non Δ -dominated;

```

 $P_N \leftarrow \emptyset;$ 
 $p^* \leftarrow$  shortest path under  $t_1;$ 
 $u \leftarrow v(p^*, t_2);$  /* first estimate for the upper bound */
 $p_1 \leftarrow$  shortest path under  $t_2;$ 
IF  $v(p_1, t_1) = v(p^*, t_1)$  THEN  $u \leftarrow v(p_1, t_2)$  /* see if estimate for  $u$  can be improved */
END IF;
 $S \leftarrow \{p_1\}; f_1 \leftarrow v(p_1, t_1); f_2 \leftarrow v(p_1, t_2);$ 
 $j \leftarrow 1;$ 
REPEAT /* outer loop */
  REPEAT /* inner loop */
     $j \leftarrow j + 1;$ 
     $p_j \leftarrow$   $j$ th shortest path under  $t_2;$  /*  $p_j$  becomes NULL if there are no more paths */
    IF  $p_j$  is NULL THEN exit inner and outer loop
    END IF;
    IF  $v(p_j, t_1) = v(p^*, t_1)$  THEN  $u \leftarrow \min \{u, v(p_j, t_2)\}$  /* see if estimate for  $u$  can be improved */
    END IF;
    IF  $v(p_j, t_1) = f_1$  and  $v(p_j, t_2) = f_2$  THEN  $S \leftarrow S \cup \{p_j\}$  /* path joins other candidates in  $S$  */
    END IF;
  UNTIL  $v(p_j, t_1) < f_1$  or  $v(p_j, t_2) > f_2;$ 
  IF  $v(p_j, t_1) < f_1$  THEN
     $f_1 \leftarrow v(p_j, t_1);$ 
    IF  $v(p_j, t_2) > f_2$  THEN
       $f_2 \leftarrow v(p_j, t_2);$ 
       $P_N \leftarrow P_N \cup S$  /* the candidates in  $S$  were all non  $\Delta$ -dominated */
    END IF;
     $S \leftarrow \{p_j\}$  /* new sole candidate */
  END IF;
UNTIL  $v(p_j, t_2) > u;$ 
 $P_N \leftarrow P_N \cup S$  /* will contain the set of non  $\Delta$ -dominated paths */
END

```

Algorithm 2

```

Let  $P_?$  contain the set of paths that may be non  $\Delta$ -dominated;
 $p^* \leftarrow$  shortest path considering  $v^M(\cdot);$ 
 $u \leftarrow v^M(p^*);$  /* upper bound: any path  $q$  such that  $v^m(q) > u$  must be  $\Delta$ -dominated */
 $p_1 \leftarrow$  shortest path considering  $v^m(\cdot);$ 
 $P_? \Delta \leftarrow \{p_1\};$ 
 $j \leftarrow 2;$ 
 $p_2 \leftarrow$  2nd shortest path considering  $v^m(\cdot);$  /*  $p_2$  becomes NULL if there are no more paths */
WHILE  $p_j$  is not NULL and  $v^m(p_j) \leq u$  DO
  dominated  $\leftarrow$  False;
  LOOP for all paths  $p_i$  in  $P_?$ 

```

```

IF exhaustive_test( $p_i \Delta p_j$  ?) = False THEN
  IF exhaustive_test( $p_j \Delta p_i$  ?) = True THEN  $P_i \leftarrow P_i - \{p_i\}$ 
  END IF
ELSE /*  $p_j$  is  $\Delta$ -dominated */
  dominated  $\leftarrow$  True;
  exit loop
END IF
END LOOP;
IF not dominated THEN  $P_i \leftarrow P_i \cup \{p_i\}$ 
END IF;
 $j \leftarrow j + 1$ ;
 $p_j \leftarrow j$ th shortest path considering  $v^m(\cdot)$  /*  $p_j$  becomes NULL if there are no more paths */
END WHILE
END /*  $P_i$  now contains all the non  $\Delta$ -dominated paths */

```

Algorithm 3

Let P_N contain the set of non Δ -dominated paths; Let L contain the last paths considered that are candidate to be non Δ -dominated; Let f_m and f_M contain the value, considering $v^m(\cdot)$ and $v^M(\cdot)$, respectively, of the last paths considered in L ; Let $P^=$ contain the set of paths that are $<\approx$ -dominated but may be non Δ -dominated, with $v^m(\cdot) = f_m$; Let $P^>$ contain the set of paths that are $<\approx$ -dominated but may be non Δ -dominated, with $v^m(\cdot) > f_m$;

```

FUNCTION is_dominated ( $q$ , set_of_paths) RETURNS true or false;
/* returns true iff path  $q$  is  $\Delta$ -dominated by any path in set_of_paths */
BEGIN
  LOOP for all paths  $p$  in set_of_paths /* see if  $q$  is  $\Delta$ -dominated */
    IF  $v^m(p) \leq v^m(q)$  and  $v^M(p) \leq v^M(q)$  THEN /* this condition avoids unnecessary tests */
      IF exhaustive_test( $p \Delta q \Delta$ ) = True THEN RETURN True; /* exits function */
      END IF
    END IF
  END LOOP;
  RETURN False /* if loop did not end after testing true for some  $p$  in set_of_paths */
END;
FUNCTION delete_dominated (passive_set, active_set) RETURNS a set of paths;
/* returns the set of all paths in passive_set that are not  $\Delta$ -dominated by paths in active_set */
BEGIN
  LOOP for all paths ( $p, q$ ) such that  $p \in$  passive_set,  $q \in$  active_set
    IF  $v^m(p) \geq v^m(q)$  and  $v^M(p) \geq v^M(q)$  THEN /* this condition avoids unnecessary tests */
      IF exhaustive_test( $q \Delta p$  ?) = True THEN passive_set  $\leftarrow$  passive_set -  $\{p\}$ 
      END IF
    END IF
  END LOOP;
  RETURN passive_set
END;

```

MAIN BODY

```

 $p^* \leftarrow$  shortest path considering  $v^M(\cdot)$ ; /* these instructions return NULL whenever there are no more paths */
 $u \leftarrow v^M(p^*)$ ; /* upper bound: any path  $q$  such that  $v^m(q) > u$  must be  $\Delta$ -dominated */

```

```

 $p_1 \leftarrow$  shortest path considering  $v^m(\cdot)$ ;
 $L \leftarrow \{p_1\}; f_m \leftarrow v^m(p_1); f_M \leftarrow v^M(p_1)$ ;
 $P_N \leftarrow \emptyset; P^= \leftarrow \emptyset; P^> \leftarrow \emptyset$ ;
 $j \leftarrow 2; p_2 \leftarrow$  2nd shortest path considering  $v^m(\cdot)$ ;
WHILE  $p_j \neq \text{NULL}$  and  $v^m(p_j) \leq u$  DO
  WHILE  $p_j \neq \text{NULL}$  and  $v^m(p_j) = f_m$  DO
    IF  $v^M(p_j) = f_M$  THEN /* CASE 1 */
      REPEAT
        IF not is_dominated ( $p_j, L$ ) THEN
           $L \leftarrow$  delete_dominated ( $L, \{p_j\}$ )  $\cup \{p_j\}$ 
        END IF;
         $j \leftarrow j + 1; p_j \leftarrow$   $j$ th shortest path considering  $v^m(\cdot)$ 
      UNTIL  $p_j = \text{NULL}$  or  $v^M(p_j) \neq f_M$  or  $v^m(p_j) > f_m$ ;
    END IF;
    IF  $p_j \neq \text{NULL}$  and  $v^m(p_j) = f_m$  and  $v^M(p_j) < f_M$  THEN /* CASE 2 */
       $P^= \leftarrow$  delete_dominated ( $P^=, L$ )  $\cup L$ 
       $L \leftarrow \{p_j\}; f_m \leftarrow v^M(p_j)$  /*  $f_m = v^m(p_j)$  */
       $j \leftarrow j + 1; p_j \leftarrow$   $j$ th shortest path considering  $v^m(\cdot)$ 
    END IF;
    IF  $p_j \neq \text{NULL}$  and  $v^m(p_j) = f_m$  and  $v^M(p_j) > f_M$  THEN /* CASE 3: */
      IF not is_dominated ( $p_j, P_N \cup P^=$ ) THEN  $P^= \leftarrow$  delete_dominated ( $P^=, \{p_j\}$ )  $\cup \{p_j\}$ 
      END IF;
       $j \leftarrow j + 1; p_j \leftarrow$   $j$ th shortest path considering  $v^m(\cdot)$ 
    END IF;
  END WHILE;
  IF  $p_j \neq \text{NULL}$  and  $v^m(p_j) \leq u$  THEN
    IF  $v^M(p_j) < f_M$  THEN /* CASE 4 */
      delete_dominated ( $P^=, L$ ); delete_dominated ( $P^>, L$ )
       $P_N \leftarrow P_N \cup L \cup P^=$ ;  $P^= \leftarrow \emptyset$ ;
      LOOP for all paths  $q$  in  $P^>$ 
        IF  $v^m(q) < v^m(p_j)$  THEN  $P_N \leftarrow P_N \cup \{q\}$  ELSE  $P^= \leftarrow P^= \cup \{q\}$ 
        END IF
      END LOOP;
       $P^> \leftarrow \emptyset$ ;
       $L \leftarrow \{p_j\}; f_m \leftarrow v^m(p_j); f_M \leftarrow v^M(p_j)$ 
    ELSE /* CASE 5 */
      IF not is_dominated ( $p_j, P_N \cup P^= \cup P^>$ ) THEN  $P^> \leftarrow$  delete_dominated ( $P^>, \{p_j\}$ )  $\cup \{p_j\}$ 
      END IF
    END IF
     $j \leftarrow j + 1; p_j \leftarrow$   $j$ th shortest path considering  $v^m(\cdot)$ 
  END IF;
END WHILE;
delete_dominated ( $P^=, L$ ); delete_dominated ( $P^>, L$ )
 $P_N \leftarrow P_N \cup L \cup P^= \cup P^>$ ; /* will be the set of all non  $\Delta$ -dominated paths */
END

```

References

- Azevedo, J.A., Martins, E.Q.V., 1991. An algorithm for the multiobjective shortest path problem on acyclic networks. *Investigação Operacional* 11 (1), 52–69.
- Azevedo, J.A., Costa, M.E.O.S., Madeira, J.J.E.R.S., Martins, E.Q.V., 1993. An algorithm for the ranking of shortest paths. *European Journal of Operational Research* 69, 97–106.
- Clímaco, J.C.N., Martins, E.Q.V., 1980. On the determination of the nondominated paths in a multiobjective network problem. *Proceedings of the V Symposium über Operations Research, Köln, in Methods in Operations Research, Vol. 40, Anton Hain, Königstein, 1981, pp. 255–258.*
- Clímaco, J.C.N., Martins, E.Q.V., 1982. A bicriterion shortest path algorithm. *European Journal of Operational Research* 11, 399–404.
- Dias, L.C., Clímaco, J.N., 1998. Some thoughts on the shortest path problem with partial information on costs. *Proceedings (CD-ROM) CESA'98 – Computational Engineering in Systems Applications, IMACS Multiconference (IMACS and IEEE), Nabeul-Hammamet, Tunisia, 1–4 April 1998, p. 6.*
- French, S., 1995. Uncertainty and imprecision: modelling and analysis. *Journal of the Operational Research Society* 46, 70–79.
- Hazen, G.B., 1986. Partial information, dominance and potential optimality in multiattribute utility theory. *Operations Research* 34 (2), 297–310.
- Kouvelis, P., Yu, G., 1997. *Robust Discrete Optimization and its Applications*. Kluwer Academic Publishers, Dordrecht.
- Martins, E.Q.V., Santos, J.L.E., 1996. A new shortest paths ranking algorithm. *Research report, Department of Mathematics, Universidade de Coimbra, Coimbra, Portugal.*
- Martins, E.Q.V., Pascoal, M.M.B., Santos, J.L.E., 1998. A new algorithm for ranking loopless paths. *Research report, Department of Mathematics, Universidade de Coimbra, Coimbra, Portugal.*
- Rios Insua, D., French, S., 1991. A framework for sensitivity analysis in discrete multi-objective decision-making. *European Journal of Operational Research* 54, 176–190.
- Roy, B., Bouyssou, D., 1989. Main sources of inaccurate determination, uncertainty and imprecision in decision models. *Mathematical and Computer Modelling* 12, 1245–1254.
- Roy, B., 1997. Un chaînon manquant en RO-AD: Les conclusions robustes. *Cahier du LAMSADE, No 144, Université de Paris Dauphine, Paris.*
- Vincke, Ph., 1997. Robust solutions and methods on decision aid. *Working paper.*
- Weber, M., 1987. Decision making with incomplete information. *European Journal of Operational Research* 28, 44–57.