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);
• 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 Vincze (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:

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

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

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

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., \( c_{ij}(t) = c_{ij}(x_1, x_2, \ldots, x_r) \), such that if any \( x_k \) increases \((1 \leq k \leq r)\), while the other parameters remain unchanged, then \( c_{ij}(t) \) also increases. Each parameter \( x_k \) might represent fuel cost, labour cost, interest rate, etc. Furthermore, there is an instance where all the \( x_1, x_2, \ldots, x_r \) are minimum and another instance where all the \( x_1, x_2, \ldots, x_r \) 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 \( x_1, x_2, \ldots, x_r \) 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 \( \lambda_{ij} \) 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 (\( \Delta_A \)-dominance)
  \[ p \Delta_A q \iff v(p(t)) \leq v(q(t)), \quad \forall t, t' \in T \quad \text{and} \quad \exists t, t' \in T : v(p(t)) < v(q(t')). \]

- Usual dominance or Bernoulli dominance (\( \Delta \)-dominance)
  \[ p \Delta q \iff v(q(t)) \leq v(q(t)), \quad \forall t \in T \quad \text{and} \quad \exists t \in T : v(p(t)) < v(q(t)). \]

Absolute dominance (\( \Delta_A \)-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 \( \Delta_A \)-dominates is strictly better than the other. In contrast, \( \Delta \)-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 \( \Delta \)-dominates is strictly better than the one that is \( \Delta \)-dominated. The costs of the two paths are now computed using a common instance.

Whenever \( p \Delta q \) (even if \( p \) does not \( \Delta_A \)-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 \( \Delta \)-dominated paths, instead of considering the stronger \( \Delta_A \)-dominance as a rejection rule. As a result there will be fewer paths to consider (those remaining in the set of non \( \Delta \)-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 \( \text{Opt}(p) = \{ t \in T : v(p(t)) \leq v(q(t)), \forall q \in P - \{p\} \} \) 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
Without loss of generality, paths are ranked according to their cost under instance
not usually exist, the choice of the best path depends on the DMs’ personal preferences, values, attitudes
and beliefs.

et al. (1991) can be used instead. Both algorithms are based on a procedure that computes a ranking of the
pessimistic and optimistic scenarios), each with different costs for some arcs.

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.

Table 1

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_1$</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>$p_2$</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>$p_3$</td>
<td>20</td>
<td>10</td>
</tr>
</tbody>
</table>

the two parameter combinations ($j = 1, 2$). We can observe that $p_1$, $p_2$ and $p_3$ are non $\Delta$-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 $\Delta$-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 $T = \{t_1, t_2, \ldots, t_k\}$, 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 $\Delta$-dominated. Indeed,
the concept of $\Delta$-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 algo-
rithm is presented in Appendix A (Algorithm 1). It can be outlined as follows:

Let $P_N$ contain the set of non $\Delta$-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 $\Delta$-dominated or non $\Delta$-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 $\Delta$-domi-
nated. 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 $\Delta$-dominated.

Case 3. $v(p_j, t_1) < f_1$ and $v(p_j, t_2) = f_2$. Paths is $S$ are $\Delta$-dominated by $p_j$. Hence, $S \leftarrow \{p_1\}$.

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


A. L. Dias, J. N. Clímaco
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 \( u \hat{=} 110 \) accordingly. The algorithm would continue as shown in Table 2. The set of the non \( \Delta \)-dominated paths would be \( \{p_1, p_2, p_3, p_5\} \).

Let us now consider the general case for a discrete \( T \), where \( T = \{t_1, t_2, \ldots, t_K\} \). 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 \( \Delta \)-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 \( \Delta \)-dominated considering the set of sample points but non \( \Delta \)-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 \( p \Delta \)-dominates \( q \) for a given pair of paths \( p, q \in P \) as follows:

![Fig. 1. Problem with two instances for costs.](image-url)
(Exhaustive test) let $d^*$ be the optimal value of
\[
\begin{align*}
\max & \quad d(t) = v(p, t) - v(q, t) \\
\text{s.t.} & \quad t \in T; \\
\text{then} & \quad p \Delta q \iff d^* < 0 \text{ or } [d^* = 0 \text{ and } \exists t \in T : v(p, t) < v(q, t)].
\end{align*}
\]

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 $\Delta$-dominated and those that are surely non $\Delta$-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
\[
\begin{align*}
v^m(p) & \leq v(p, t) \leq v^M(p), \quad \forall t \in T, p \in P. \tag{1}
\end{align*}
\]

Let us now define the following binary relations for any pair of paths $(p, q) \in P^2$:
\[
\begin{align*}
p & \ll q \iff v^M(p) < v^m(q), \\
p & < q \iff [v^m(p) = v^m(q) \text{ and } v^M(p) < v^M(q)] \text{ or } [v^m(p) < v^m(q) \text{ and } v^M(p) = v^M(q)], \\
p & \prec q \iff v^m(p) \lessgtr v^m(q) \text{ and } v^M(p) \lessgtr v^M(q).
\end{align*}
\]

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

**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, $v(p, t) \leq v^M(p) < v^m(q) = v(q, t), \forall t \in T \Rightarrow v(p, t) < v(q, t), \forall t \in T \Rightarrow p \Delta q$. \(\Box\)

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
\[
\begin{align*}
&v(1, 2) = 20, \quad v(2, 3) = 5, \quad v(3, 5) = 0, \quad v(1, 4) = 5, \quad v(4, 5) = 3, \quad v(1, 4) = 15, \\
&v(1, 2) = 20, \quad v(2, 3) = 10, \quad v(3, 5) = 20.
\end{align*}
\]

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
\[
\begin{align*}
p \text{ is } \ll \text{-dominated iff } \exists q \in P : q \ll p.
\end{align*}
\]

<table>
<thead>
<tr>
<th>$(i, j)$</th>
<th>$c_{ij}(t_1)$</th>
<th>$c_{ij}(t_2)$</th>
<th>$c_{ij}(t_3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,2)</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>(2,3)</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>(3,5)</td>
<td>20</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>(1,4)</td>
<td>5</td>
<td>30</td>
<td>15</td>
</tr>
<tr>
<td>(4,5)</td>
<td>25</td>
<td>4</td>
<td>20</td>
</tr>
</tbody>
</table>
Among the non \( \ll \)-dominated paths we will find all the non \( \Delta \)-dominated paths along with some \( \Delta \)-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 Cl\'maco 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 \( \Delta \)-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 \( v^m(q) \ll v^m(p) \) \( \Rightarrow \) \( (p \Delta q) \).

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

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

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

**Proposition 3.** Let \( p,q \in P \) and Condition 1 be satisfied. Then \( p \Delta q \Rightarrow p \ll q \).

**Proof.** We will show that \( \sim (p \ll q) \Rightarrow \sim (p \Delta q) \). 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 \( \Delta \)-dominate \( q \). \( \square \)

**Definition.** \( p \) is \( \ll \)-dominated iff \( \exists q \in P : q \ll p \).

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

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

One cannot prove that \( p \ll q \Rightarrow p \Delta q, p \Delta q \Rightarrow p \ll q \) nor \( p \Delta q \Leftarrow p \ll q \). 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,
Proposition 3: for any shortest path (there are no more paths in the graph) or

$$p \in \mathcal{P}_L$$

In iteration 1, it finds the shortest path

and let

$$L \leftarrow L \cup P_j.$$

Notice that

$$p < q \Rightarrow p \Delta q$$

is false because

$$p_2 < p_3$$

but

$$p_2 \Delta p_3 \in (p_3, p_1)$$

is better under

Notice that

$$p \Rightarrow q$$

is false because

$$p_1 \Delta p_2$$

but

$$p_1 \leq p_2.$$
the expression depending on two parameters concerning the near future. Parameter costs, which may grow up to 5% above the current level. The cost of an arc (in the near future) is given by any path in $f_{vM}$ respectively.

Example 5.1. Consider again the graph in Fig. 1 and the following situation. Let costs be uncertain, 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.

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

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

Case 4. $v^m(p_i) > f_m$ and $v^M(p_i) < f_M$, which means that paths in $L$ are non $\Delta$-dominated. Check if any path in $L \Delta$-dominates any path in $P^\omega$ or $P^\omega$. Let $P_N \leftarrow P_N \cup L \cup P^\omega \cup \{p \in P^\omega : v^m(p) < v^m(p_i)\}$. Let $P^\omega \leftarrow \{p \in P^\omega : v^m(p) = v^m(p_i)\}$ and then set $P^\omega \leftarrow \emptyset$. 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 $<\ll$-dominated. Check if any path in $P_N \cup P^\omega \cup P^\omega \Delta$-dominates path $p_j$. If none does, check if $p_j \Delta$-dominates any path in $P^\omega$ and let $P^\omega \leftarrow P^\omega \cup \{p\}$.

The minimum number of tests to perform when using Algorithm 3 is zero, in the case where all paths are non $<\ll$-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).

Fig. 2. Worst case (a), usual case (b) and best case (c) examples for algorithm.
higher than \( u \) is \( \Delta \)-dominated. The algorithm (see Table 6) avoids many exhaustive tests, exploiting
Propositions 2a and b as well as the transitivity of the \( \Delta \)-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 \( \Delta \)-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 \( v_M \), which occurs for the 10th path. Then, it would examine one path at a time in non
decreasing order of their \( v_m \). Suppose that paths are numbered in the figure according to the order they
would appear (when several paths have equal \( v_m \), 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 pa-
rameters (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 consider-
aton.

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 ex-
ample, 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>
<thead>
<tr>
<th>Path</th>
<th>( v_m(p_i) )</th>
<th>( v^M(p_i) )</th>
<th>Case</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADI</td>
<td>40</td>
<td>110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BFII</td>
<td>53</td>
<td>88</td>
<td>4</td>
<td>( P_N = {p_1}; f_m \leftarrow 40; f_M \leftarrow 110; P_N = \emptyset; P^\circ = \emptyset; P^\prime = \emptyset. )</td>
</tr>
<tr>
<td>BFJ</td>
<td>54</td>
<td>68</td>
<td>4</td>
<td>( P_N \leftarrow P_N \cup L = {p_1}; L \leftarrow {p_2}; f_m \leftarrow 53; f_M \leftarrow 88. )</td>
</tr>
<tr>
<td>BG</td>
<td>54</td>
<td>68</td>
<td>1</td>
<td>( P_N \leftarrow P_N \cup L = {p_1, p_2}; L \leftarrow {p_3}; f_m \leftarrow 54; f_M \leftarrow 68. )</td>
</tr>
<tr>
<td>ADEDI</td>
<td>60</td>
<td>165</td>
<td>5</td>
<td>Clearly, ( p_i \Delta ) ( p_5 ) is discarded.</td>
</tr>
<tr>
<td>BCDI</td>
<td>69</td>
<td>110</td>
<td></td>
<td>( v^M(p_i) &gt; u \Rightarrow \text{exit outer loop}; )</td>
</tr>
</tbody>
</table>

Table 5

<table>
<thead>
<tr>
<th>Arc ((i,j))</th>
<th>( B(i,j) )</th>
<th>( k_s(i,j) )</th>
<th>( k_S(i,j) )</th>
<th>( m_i )</th>
<th>( M_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A )</td>
<td>10.115</td>
<td>2.3</td>
<td>695.4</td>
<td>10</td>
<td>45</td>
</tr>
<tr>
<td>( B )</td>
<td>19.205</td>
<td>4.1</td>
<td>11.8</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>( C )</td>
<td>20.29</td>
<td>5.8</td>
<td>88.4</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td>( D )</td>
<td>10.16</td>
<td>3.2</td>
<td>293.6</td>
<td>10</td>
<td>25</td>
</tr>
<tr>
<td>( E )</td>
<td>10.1</td>
<td>2</td>
<td>396</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>( F )</td>
<td>5.105</td>
<td>2.1</td>
<td>155.8</td>
<td>5</td>
<td>13</td>
</tr>
<tr>
<td>( G )</td>
<td>35.375</td>
<td>7.5</td>
<td>245</td>
<td>35</td>
<td>48</td>
</tr>
<tr>
<td>( H )</td>
<td>9.135</td>
<td>2.7</td>
<td>114.6</td>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>( I )</td>
<td>20.255</td>
<td>5.1</td>
<td>389.8</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>( J )</td>
<td>30.365</td>
<td>7.3</td>
<td>85.4</td>
<td>30</td>
<td>35</td>
</tr>
</tbody>
</table>
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.

---

**Table 7**

<table>
<thead>
<tr>
<th>$j$</th>
<th>Case</th>
<th>$P_N$</th>
<th>$L$</th>
<th>$P^r$</th>
<th>$P^a$</th>
<th>$f_M$</th>
<th>$f_M$</th>
<th>Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$-$</td>
<td>$\emptyset$</td>
<td>$p_1$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
<td>$v^r(p_1)$</td>
<td>$v^a(p_1)$</td>
<td>None</td>
</tr>
<tr>
<td>2</td>
<td>$2$</td>
<td>$\supseteq$</td>
<td>$p_2$</td>
<td>$p_1$</td>
<td>$\supseteq$</td>
<td>$v^r(p_2)$</td>
<td>$v^a(p_1)$</td>
<td>None</td>
</tr>
<tr>
<td>3</td>
<td>$4$</td>
<td>$p_1, p_2$</td>
<td>$p_3$</td>
<td>$\emptyset$</td>
<td>$\supseteq$</td>
<td>$v^r(p_1)$</td>
<td>$v^a(p_1)$</td>
<td>$p_2 \Delta p_1$</td>
</tr>
<tr>
<td>4</td>
<td>$4$</td>
<td>$p_1, p_2, p_3$</td>
<td>$p_4$</td>
<td>$\supseteq$</td>
<td>$\supseteq$</td>
<td>$v^r(p_4)$</td>
<td>$v^a(p_1)$</td>
<td>None</td>
</tr>
<tr>
<td>5</td>
<td>$1$</td>
<td>$\supseteq$</td>
<td>$p_4, p_5$</td>
<td>$\supseteq$</td>
<td>$\supseteq$</td>
<td>$\supseteq$</td>
<td>$\supseteq$</td>
<td>$p_4 \Delta p_5$ ; $p_1 \Delta p_5$</td>
</tr>
<tr>
<td>6</td>
<td>$3$</td>
<td>$\supseteq$</td>
<td>$p_6$</td>
<td>$\supseteq$</td>
<td>$\supseteq$</td>
<td>$\supseteq$</td>
<td>$\supseteq$</td>
<td>None</td>
</tr>
<tr>
<td>7</td>
<td>$1$</td>
<td>$\supseteq$</td>
<td>$p_4, p_5, p_7$</td>
<td>$\supseteq$</td>
<td>$\supseteq$</td>
<td>$\supseteq$</td>
<td>$\supseteq$</td>
<td>$p_4 \Delta p_7$ ; $p_4 \Delta p_7$ ; $p_4 \Delta p_7$ ; $p_4 \Delta p_7$</td>
</tr>
<tr>
<td>8</td>
<td>$4$</td>
<td>$p_1, p_2, p_3, p_4, p_5, p_6, p_7$</td>
<td>$p_8$</td>
<td>$\emptyset$</td>
<td>$\supseteq$</td>
<td>$v^r(p_8)$</td>
<td>$v^a(p_1)$</td>
<td>$p_8 \Delta p_8$ ; $p_8 \Delta p_8$ ; $p_8 \Delta p_8$</td>
</tr>
<tr>
<td>9</td>
<td>$5$</td>
<td>$\supseteq$</td>
<td>$\supseteq$</td>
<td>$p_9$</td>
<td>$\supseteq$</td>
<td>$\supseteq$</td>
<td>$\supseteq$</td>
<td>$p_9 \Delta p_9$ ; $p_9 \Delta p_9$ ; $p_9 \Delta p_9$</td>
</tr>
<tr>
<td>10</td>
<td>$4$</td>
<td>$p_1, p_2, p_3, p_4,$</td>
<td>$p_{10}$</td>
<td>$\supseteq$</td>
<td>$\supseteq$</td>
<td>$v^r(p_{10})$</td>
<td>$v^a(p_{10})$</td>
<td>None</td>
</tr>
<tr>
<td>11</td>
<td>$-$</td>
<td>$p_1, p_2, p_3, p_4,$</td>
<td>$p_5, p_6, p_7, p_8, p_9, p_{10}$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>None</td>
</tr>
</tbody>
</table>
Appendix A

Algorithm 1

Let $P_N$ contain the set of non $\Delta$-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 $\Delta$-dominated or non $\Delta$-dominated;

\[
P_N \leftarrow \emptyset;
\]

\[
p^* \leftarrow \text{shortest path under } t_1;
\]

\[
u \leftarrow v(p^*, t_2); \quad /\!* \text{first estimate for the upper bound} */\]

\[
p_1 \leftarrow \text{shortest path under } t_2;
\]

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;

\text{REPEAT} /* outer loop */

\text{REPEAT} /* inner loop */

\[j \leftarrow j + 1;\]

\[p_j \leftarrow j\text{th shortest path under } t_2; \quad /\!* p_j \text{ becomes NULL if there are no more paths} */\]

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 $P_t$ contain the set of paths that may be non $\Delta$-dominated;

\[
p^* \leftarrow \text{shortest path considering } v^M(\cdot);
\]

\[
u \leftarrow v^M(p^*); \quad /\!* \text{upper bound: any path } q \text{ such that } v^m(q) > u \text{ must be } \Delta\text{-dominated} */\]

\[
p_1 \leftarrow \text{shortest path considering } v^m(\cdot);
\]

\[
P_\Delta \leftarrow \{p_1\};
\]

\[j \leftarrow 2;\]

\[
p_2 \leftarrow 2\text{nd shortest path considering } v^m(\cdot); \quad /\!* p_2 \text{ becomes NULL if there are no more paths} */\]

WHILE $p_2$ is not NULL and $v^m(p_2) \leq u$ DO

\text{dominated} \leftarrow False;

LOOP for all paths $p_i$ in $P_t$
IF exhaustive_test($p_i \Delta p_i$) = False THEN
  IF exhaustive_test($p_j \Delta p_i$) = True THEN $P_j \leftarrow P_j - \{p_i\}$
  END IF
ELSE /* $p_j$ is $\Delta$-dominated */
  dominated $\leftarrow$ True;
  exit loop
END IF
END LOOP;
IF not dominated THEN
  $P_i \leftarrow P_i \cup \{p_j\}$
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 $\Delta$-dominated paths; Let $L$ contain the last paths considered that are candidate to be non $\Delta$-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 $\preceq$-dominated but may be non $\Delta$-dominated, with $v^m(\cdot) = f_m$; Let $P^{>}$ contain the set of paths that are $\preceq$-dominated but may be non $\Delta$-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($p \Delta q$) = True THEN RETURN True; /* exits function */
    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 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 */
\[ p_1 \leftarrow \text{shortest path considering } v^m(\cdot); \]

\[ L \leftarrow \{p_1\}; f_m \leftarrow v^m(p_1); f_M \leftarrow v^M(p_1); \]

\[ P_N \leftarrow \emptyset; P^+=\emptyset; P^> \leftarrow \emptyset; \]

\[ j \leftarrow 2; p_2 \leftarrow \text{2nd shortest path considering } v^m(\cdot); \]

\[ \text{WHILE } p_j \neq \text{NULL and } v^m(p_j) \leq u \text{ DO} \]

\[ \text{WHILE } p_j \neq \text{NULL and } v^m(p_j) = f_m \text{ DO} \]

\[ \text{IF } v^M(p_j) = f_m \text{ THEN } \text{* CASE 1 *}/ \]

\[ \text{REPEAT} \]

\[ \text{IF not is_dominated } (p_j, L) \text{ THEN} \]

\[ L \leftarrow \text{delete_dominated } (L, \{p_j\}) \cup \{p_j\} \]

\[ \text{END IF;} \]

\[ j \leftarrow j + 1; p_j \leftarrow j\text{th shortest path considering } v^m(\cdot) \]

\[ \text{UNTIL } p_j = \text{NULL or } v^M(p_j) \neq f_M \text{ or } v^m(p_j) > f_m; \]

\[ \text{END IF;} \]

\[ \text{IF } p_j \neq \text{NULL and } v^m(p_j) = f_m \text{ and } v^M(p_j) < f_M \text{ THEN } \text{* CASE 2 *}/ \]

\[ P^+ \leftarrow \text{delete_dominated } (P^+, L) \cup L \]

\[ L \leftarrow \{p_j\}; f_m \leftarrow v^m(p_j) \]

\[ j \leftarrow j + 1; p_j \leftarrow j\text{th shortest path considering } v^m(\cdot) \]

\[ \text{END IF;} \]

\[ \text{IF } p_j \neq \text{NULL and } v^m(p_j) = f_m \text{ and } v^M(p_j) > f_M \text{ THEN } \text{/ CASE 3: */} \]

\[ \text{IF not is_dominated } (p_j, P_N \cup P^+) \text{ THEN} \]

\[ P^+ \leftarrow \text{delete_dominated } (P^+, \{p_j\}) \cup \{p_j\} \]

\[ \text{END IF;} \]

\[ j \leftarrow j + 1; p_j \leftarrow j\text{th shortest path considering } v^m(\cdot) \]

\[ \text{END IF;} \]

\[ \text{END WHILE;} \]

\[ \text{IF } p_j \neq \text{NULL and } v^m(p_j) \leq u \text{ THEN} \]

\[ \text{IF } v^M(p_j) < f_m \text{ THEN } \text{* CASE 4 *}/ \]

\[ \text{delete_dominated } (P^+, L); \text{delete_dominated } (P^>, L) \]

\[ P_N \leftarrow P_N \cup L \cup P^+; P^+ \leftarrow \emptyset; \]

\[ \text{LOOP for all paths } q \text{ in } P^> \]

\[ \text{IF } v^m(q) < v^m(p_j) \text{ THEN} P_N \leftarrow P_N \cup \{q\} \text{ ELSE } P^+ \leftarrow P^+ \cup \{q\} \]

\[ \text{END IF} \]

\[ \text{END LOOP;} \]

\[ P^+ \leftarrow \emptyset; \]

\[ L \leftarrow \{p_j\}; f_m \leftarrow v^m(p_j); f_M \leftarrow v^M(p_j) \]

\[ \text{ELSE } \text{/ CASE 5 */} \]

\[ \text{IF not is_dominated } (p_j, P_N \cup P^+ \cup P^>) \text{ THEN} P^> \leftarrow \text{delete_dominated } (P^>, \{p_j\}) \cup \{p_j\} \]

\[ \text{END IF;} \]

\[ j \leftarrow j + 1; p_j \leftarrow j\text{th shortest path considering } v^m(\cdot) \]

\[ \text{END IF;} \]

\[ \text{END WHILE;} \]

\[ \text{delete_dominated } (P^+, L); \text{delete_dominated } (P^>, L) \]

\[ P_N \leftarrow P_N \cup L \cup P^+ \cup P^>; \text{* will be the set of all non } \Delta \text{-dominated paths */} \]

\[ \text{END} \]
References