The Determination of the Path with Minimum-Cost Norm Value*

José M. P. Paixão
Faculdade de Ciências, Universidade de Lisboa, Lisbon, Portugal

Ernesto de Queirós Vieira Martins,† Mário S. Rosa, and José Luis E. Santos
Departamento de Matemática, Faculdade de Ciências e Tecnologia, Universidade de Coimbra, Apartado 3008, 3001-454 Coimbra, Portugal

The multiobjective shortest path problem (MSPP) consists of finding the best nondominated path linking two specified nodes in a network where $k > 1$ different criteria are considered for the arc costs. Solving the MSPP implies determining all the nondominated paths; there may be many such paths, making the selection process a very hard task to accomplish. A traditional way of dealing with such a difficulty is using a utility function where the parameters are aggregated with different weights. In this work, an alternative utility function based on the norm value associated with each path is considered. Two algorithms for solving the minimum-cost norm path problem are then proposed and some computational results are presented. © 2003 Wiley Periodicals, Inc.

Keywords: dominance relation; multiobjective shortest path; ranking paths

1. INTRODUCTION

The shortest path problem (SPP) is a well-known network optimization problem where one aims to determine the path that minimizes the sum of the arc weights (distance, cost, time, . . .). The SPP was initially proposed in the late 1950s [1, 18] and, since then, has been studied by many researchers (see [9, 10] for extensive bibliographies on the SPP).

Sometimes, a multiobjective formulation is needed, in which several parameters are associated with each arc, allowing the possibility of incorporating various criteria. The objectives can be defined as cost, distance, time, reliability [12], accessibility [7], capacity [17], and others. For a review of the multiobjective shortest path problem (MSPP), the interested reader is referred to [6].

Therefore, one intends to determine a path that minimizes simultaneously all the criteria under consideration. Usually, there is a conflict among the different criteria and such an ideal solution does not exist. The resolution of the MSPP turns into finding *nondominated paths,* that is, paths for which there is no other path with better values for all the criteria. The well-known labeling algorithm used for solving the SPP can be easily extended to the MSPP, allowing the determination of the full set of nondominated paths [3, 4, 11, 12, 14, 20].

The MSPP may become hard to solve due to the ambiguity associated with the characterization of the optimal path. Hansen [12] proved that this problem is intractable because the number of nondominated solutions may increase exponentially with the number of nodes in the network. Taking into account this fact, several methods have been developed to select a "satisfactory" nondominated path for a specific instance. These methods include interactive procedures [5, 8], utility functions [13, 19], and approximation methods [21].

A usual expression for the utility function is a weighted sum of the criteria, but it only allows us to determine supported nondominated solutions for the MSPP, that is, the extreme points of the convex hull of the set of feasible solutions for the MSPP. In this work, we propose a different expression based on the Euclidean norm of the cost for the MSPP (which also allows searching for unsupported nondominated solutions). In fact, the ideal solution would min-

* **Dedication:** The paper is dedicated to Professor Ernesto Martins. His main work on multiobjective and ranking shortest paths problems constitute an important foundation for the results reported in the paper and still is a precious inspiration for our research.

† Deceased on November 8, 2000

Received October 2001; accepted March 2003

Correspondence to: J. L. E. Santos; e-mail: zeluis@mat.uc.pt

Contract grant sponsors: CMUC (Centro de Matemática da Universidade de Coimbra); CISUC (Centro de Informática e Sistemas da Universidade de Coimbra); CIO/UL (Centro de Investigação Operacional da Universidade de Lisboa)

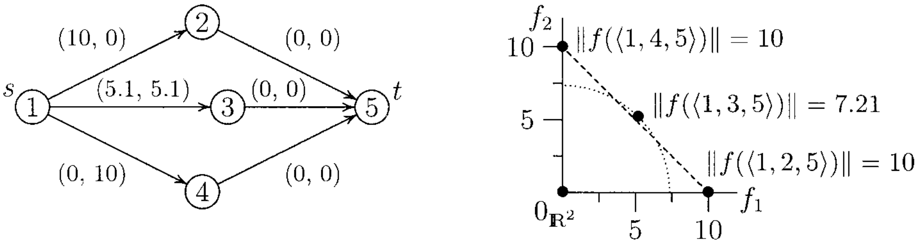
(10, 0)
(0, 0)
(5.1, 5.1)
(0, 0)
(0, 10)
(0, 0)
s
t
1
2
3
4
5
f_2
10
5
0$_{\mathbb{R}^2}$
5
10
f_1
$\|f(\langle 1,4,5\rangle)\| = 10$
$\|f(\langle 1,3,5\rangle)\| = 7.21$
$\|f(\langle 1,2,5\rangle)\| = 10$

FIG. 1. Bicriteria instance and path cost representation in $\mathbb{R}^2$.

imize each one of the criteria, and if that solution exists, it will be the one closest to $0_{\mathbb{R}^k}$ (the zero vector in $\mathbb{R}^k$), which means that it will be the solution with a minimum-cost norm value.

For random instances, it happens, in general, that there is not a significant difference between the solutions produced by the weighted sum and minimum norm procedures. Actually, they coincide very often. Nevertheless, instances may be identified where one may have a relevant difference between the solutions obtained by the two procedures. This is exhibited by the example presented in Figure 1 where (0, 10), (5.1, 5.1), and (10, 0) correspond to the values of the nondominated paths from node s to node t in the network. The weighted sum method will choose the first or the last solution, depending on the weights associated with each criterion. However, the solution with cost (5.1, 5.1) happens to be the one closest to $0_{\mathbb{R}^2}$ but it will never be chosen by this method, regardless of the weights.

In this paper, we will address the problem of finding the path with a minimum-cost norm value and present two algorithms for solving it. One of the algorithms is based on the determination of the nondominated paths for the MSPP and the other is based on a ranking shortest path procedure. An example is shown and some computational experience is reported.

2. THE MSPP AND RELATED PROBLEMS

Let $\mathcal{G} = (\mathcal{N}, \mathcal{A}, c)$ denote a network, where $\mathcal{N} = \{1, \ldots, n\}$ is the set of nodes and $\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of arcs; therefore, each arc is identified with a pair (i, j) with $i, j \in \mathcal{N}$. The cost function is of the form $c : \mathcal{A} \to (\mathbb{R}_0^+)^k$, with $c(i, j) = \mathbf{c_{i,j}} = (c_{i,j}^1, \ldots, c_{i,j}^k)$, where k is the number of parameters (criteria) associated with each arc.

A path p between two nodes $i, j \in \mathcal{N}$ is an alternating sequence of nodes and arcs of the form $p = \langle i = v_0, a_1, v_1, \ldots, a_l, v_l = j\rangle$, with $v_r \in \mathcal{N}$, $r = 0, \ldots, l$, and $a_r = (v_{r-1}, v_r) \in \mathcal{A}$, $r = 1, \ldots, l$. For simplicity, it can also be denoted by $p = \langle v_0, v_1, \ldots, v_l\rangle$ and, by convention, $p = \langle v_0\rangle$ is assumed when $l = 0$. The set of paths from i to j, $i, j \in \mathcal{N}$, is denoted by $\mathcal{P}_{i,j}$. The set of all paths that can be defined in the network is denoted by $\mathcal{P}_{\mathcal{G}}$, that is, $\mathcal{P}_{\mathcal{G}} = \cup_{i,j\in\mathcal{N}} \mathcal{P}_{i,j}$.

Now, let us define the concept of *concatenation* of two paths, which will be used throughout this paper:

Definition 1. *Let $p \in \mathcal{P}_{i,j}$ and $q \in \mathcal{P}_{j,l}$ be two paths in the network, that is, $p = \langle v_0, a_1, v_1, \ldots, a_{h_1}, v_{h_1}\rangle$, with $v_0 = i$ and $v_{h_1} = j$, and $q = \langle v'_0, a'_1, v'_1, \ldots, a'_{h_2}, v'_{h_2}\rangle$, with $v'_0 = j$ and $v'_{h_2} = l$. The concatenation of p and q is denoted by $p \diamond q$ and is defined by $p \diamond q = \langle v_0, a_1, v_1, \ldots, a_{h_1}, v_{h_1} = v'_0, a'_1, v'_1, \ldots, a'_{h_2}, v'_{h_2}\rangle$.*

It is important to mention that solving network optimization problems becomes an easy task whenever the following principle holds [2, 15]:

Optimality principle: *Every optimal path is formed from optimal subpaths.*

Note that if the optimality principle holds, then the labeling algorithm [3] guarantees the determination of an optimal solution.

Finally, let us point out that one looks for paths from an initial node s to a terminal node t of $\mathcal{G} = (\mathcal{N}, \mathcal{A}, c)$, and to simplify the notation, we will use $\mathcal{P}$ instead of $\mathcal{P}_{s,t}$ to denote the set of all paths from s to t.

2.1. The MSPP

This problem is defined on a network $\mathcal{G} = (\mathcal{N}, \mathcal{A}, c)$ with a vector objective function $f(p) = (f_1(p), \ldots, f_k(p))$, where $f_l(p) = \Sigma_{(i,j)\in p}\, c_{i,j}^l$, for $l \in \{1, \ldots, k\}$. One aims to find the path from s to t that minimizes (simultaneously) all the components of f over the set $\mathcal{P}$. However, as mentioned above, obtaining such a solution is very unlikely and the most common situation is to seek a set of nondominated or efficient paths formally defined as follows:

Definition 2. *Let $\mathbf{a} = (a_1, \ldots, a_k)$ and $\mathbf{b} = (b_1, \ldots, b_k)$ be two elements of $\mathbb{R}^k$. Then, $\mathbf{a} \leq_{\mathbb{R}^k} \mathbf{b}$ if and only if $a_i \leq b_i$, for all $i \in \{1, \ldots, k\}$. If at least one of the inequalities is strict (i.e., if $\mathbf{a} \neq \mathbf{b}$), then we say that* **a** *dominates* **b** *or that* **b** *is dominated by* **a** *and we write $\mathbf{a} <_{\mathbb{R}^k} \mathbf{b}$.*

Definition 3. *Let p be a path from i to j. We say that p is a dominated path if and only if $f(q) <_{\mathbb{R}^k} f(p)$ for some $q \in \mathcal{P}_{i,j}$. Otherwise, p is a nondominated or efficient path.*

The set of nondominated paths from i to j is denoted by $\bar{\mathcal{D}}_{i,j}$ and only by $\bar{\mathcal{D}}$ when they are defined between s and t. These paths are the optimal solutions for the problem

"$\min_{\leq_{\mathbb{R}^k}}\{f(q) : q \in \mathcal{P}\}$," where a generalized definition of minimum is used, since, in this case, $\leq_{\mathbb{R}^k}$ is a partial order relation.

Definition 4. *Let A be a nonempty subset of $\mathbb{R}^k$ and R an order relation defined on A. An element m of A is a minimum of A using the relation R ($m = \min_R\{v : v \in A\}$) if and only if there is no $v \in A$ such that $v \neq m$ and $v\ R\ m$.*

The MSPP satisfies the optimality principle which can be stated, in this case, in the following way: Every nondominated path is formed from nondominated subpaths. Thus, using a labeling algorithm [3, 11, 12, 14, 20], the nondominated solutions are easily obtained. Once these solutions have been obtained, we need to choose which ones satisfy the requirements of the specific instance. In this way, a labeling algorithm can be used to do the initial selection of paths from s to t, making easier the task of selecting a final solution.

The determination of all nondominated paths becomes a very hard task to accomplish when the number of such paths is too high (although each nondominated path is easily determined). Assuming that the determination of all the nondominated paths is possible, one still faces the problem of selecting the "best" final solution.

2.2. The Path with Minimum-cost Norm Value

Since we intend to minimize each one of the components of the objective function, a natural selection for the "best" solution will be picking the one which is "closer" to $0_{\mathbb{R}^k}$, that is, the path with a minimum-cost norm value.

The problem of finding such a path is defined on a network $\mathcal{G} = (\mathcal{N}, \mathcal{A}, c)$ by considering the objective function $f_{norm} : \mathcal{P}_{\mathcal{G}} \to \mathbb{R}$, $f_{norm}(p) = \|f(p)\|$, where f is the objective function of the MSPP. Thus, one seeks the path $p^*_{norm} \in \mathcal{P}$ such that

$$f_{norm}(p^*_{norm}) = \|f(p^*_{norm})\| = \min\{\|f(q)\| : q \in \mathcal{P}\}. \quad \text{(minNorm)}$$

The following lemma allows us to reduce the solution space for the (minNorm) problem:

Lemma 1. *Let p be a path of $\mathcal{P}_{i,j}$. Then, $\|f(p)\| = \min\{\|f(q)\| : q \in \mathcal{P}_{i,j}\}$ if and only if $p \in \bar{\mathcal{D}}_{i,j}$ and $\|f(p)\| = \min\{\|f(q)\| : q \in \bar{\mathcal{D}}_{i,j}\}$.*

Proof. Since $\bar{\mathcal{D}}_{i,j} \subseteq \mathcal{P}_{i,j}$, we only need to prove that the path with a minimum norm in $\mathcal{P}_{i,j}$ is a nondominated path. Thus, let p be a path of $\mathcal{P}_{i,j}$ such that $\|f(p)\| = \min\{\|f(q)\| : q \in \mathcal{P}_{i,j}\}$ and assume that $p \notin \bar{\mathcal{D}}_{i,j}$. In this case, there is a path $q \in \mathcal{P}_{i,j}$ such that $f(q) <_{\mathbb{R}^k} f(p)$, that is, $f_i(q) \leq f_i(p)$, $i = 1, \ldots, k$, where at least one of these inequalities is strict. So, it is obvious that $\|f(q)\| < \|f(p)\|$, which contradicts our assumption on p. Therefore, p must be a nondominated path. ■

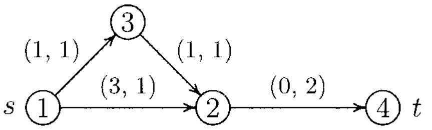

FIG. 2. A (minNorm) instance where the optimality principle is not verified.

In general, the (minNorm) problem does not satisfy the optimality principle, namely, the path with a minimum-cost norm value from s to $t(p^*_{norm})$ may not be formed from minimum-cost norm subpaths from s to each node $j \in p^*_{norm}$. This is illustrated by the example shown in Figure 2, where $\langle 1, 2, 4\rangle$ is the optimal path with a minimum-cost norm value from 1 to 4. However, the subpath $\langle 1, 2\rangle$ is not optimal from 1 to 2 since $\|f(\langle 1, 2\rangle)\| = \sqrt{10} > 2\sqrt{2} = \|f(\langle 1, 3, 2\rangle)\|$.

We can say, in this example, that the order between the values of the path $\langle 1, 2\rangle$ and $\langle 1, 3, 2\rangle$ was reversed by concatenation with $\langle 2, 4\rangle$. As we will show next, this can occur only for subpaths for which there is no dominance relationship.

Lemma 2. *Let $\mathbf{a}$ and $\mathbf{b}$ be two elements of $(\mathbb{R}_0^+)^k$ such that $\|\mathbf{a}\| < \|\mathbf{b}\|$. Then, $\mathbf{a} \leq_{\mathbb{R}^k} \mathbf{b}$ if and only if $\|\mathbf{a} + \mathbf{c}\| < \|\mathbf{b} + \mathbf{c}\|$, $\forall \mathbf{c} \in (\mathbb{R}_0^+)^k$.*

Proof. First, let us note that $\|\mathbf{a}\| < \|\mathbf{b}\|$ implies that there is at least one $i \in \{1, \ldots, k\}$ such that $a_i < b_i$. Then, if $\mathbf{a} \leq_{\mathbb{R}^k} \mathbf{b}$, it is obvious that $\mathbf{a} <_{\mathbb{R}^k} \mathbf{b}$ and for all $\mathbf{c} \in (\mathbb{R}_0^+)^k$, $\mathbf{a} + \mathbf{c} <_{\mathbb{R}^k} \mathbf{b} + \mathbf{c}$. Therefore, $\|\mathbf{a} + \mathbf{c}\| < \|\mathbf{b} + \mathbf{c}\|$.

On the other hand, if $\mathbf{a} \not\leq_{\mathbb{R}^k} \mathbf{b}$, there is (at least) one $j \in \{1, \ldots, k\}$ such that $a_j > b_j$, and this allows us to define the following vector $\mathbf{c}$ of $(\mathbb{R}_0^+)^k$: $c_l = 0$, $\forall l \in \{1, \ldots, k\}\backslash\{j\}$, and

$$c_j = 1 + \frac{\|\mathbf{b}\|^2 - \|\mathbf{a}\|^2}{2(a_j - b_j)}.$$

Thus,

$$c_j > \frac{\|\mathbf{b}\|^2 - \|\mathbf{a}\|^2}{2(a_j - b_j)},$$

that is, $2c_j(a_j - b_j) > \|\mathbf{b}\|^2 - \|\mathbf{a}\|^2$ and so $\sum_{l=1}^{k} 2c_l(a_l - b_l) > \|\mathbf{b}\|^2 - \|\mathbf{a}\|^2$. We conclude that $\|\mathbf{a} + \mathbf{c}\| > \|\mathbf{b} + \mathbf{c}\|$. ■

From this lemma, one obtains the following:

Corollary 1. *Let $\mathbf{a}$ and $\mathbf{b}$ be two elements of $(\mathbb{R}_0^+)^k$ such that $\mathbf{a} \not\leq_{\mathbb{R}^k} \mathbf{b}$ and $\|\mathbf{a}\| < \|\mathbf{b}\|$. Then, there is some $\mathbf{c} \in (\mathbb{R}_0^+)^k$ such that $\|\mathbf{a} + \mathbf{c}\| > \|\mathbf{b} + \mathbf{c}\|$.*

TABLE 1. Comparison of optimal values of the (minnorm) and (minsum) problems.

Network				p^*_{norm}		p^*_{sum}		%	
Type	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{A}\rvert$	k	f_{norm}	f_{sum}	f_{norm}	f_{sum}	(a)	(b)
Random	1000	5000	2	2704	3732	2708	3725	**0.17**	0.20
Random	1000	5000	5	4817	10,453	4824	10,443	0.15	0.10
Random	1000	5000	10	7014	21,402	7021	21,374	0.10	0.13
Grid	900	3480	2	25,696	36,204	25,795	36,097	**0.38**	0.29
Grid	900	3480	5	50,206	111,940	50,256	111,840	0.10	0.09
Grid	900	3480	10	77,472	244,497	77,626	244,332	0.20	0.07

(a) $100 \times [(f_{norm}(p^*_{sum}) - f_{norm}(p^*_{norm}))/(f_{norm}(p^*_{sum}))]$; (b) $100 \times [(f_{sum}(p^*_{norm}) - f_{sum}(p^*_{sum}))/(f_{sum}(p^*_{norm}))]$.

Finally, let us note that we are assuming that the units among the objectives are equivalent. If that is not the case, the cost for the arcs can be normalized as follows: $\tilde{c}^l_{i,j} = c^l_{i,j}/\beta_l$, where $\beta_l = \min\{f_l(p) : p \in \mathcal{P}\}$.

2.3. The Path with Minimum-cost Sum Value

We will be seen later, the algorithms described in Section 3 for solving the (minNorm) problem determine, as a first step, a path $p^*_{sum} \in \mathcal{P}$ such that

$$f_{sum}(p^*_{sum}) = \sum_{l=1}^{k} f_l(p^*_{sum})$$

$$= \min\{f_{sum}(q) : q \in \mathcal{P}\}. \quad \text{(minSum)}$$

As shown next, such a solution can be easily obtained by solving an SPP in $\mathbb{R}$ (this satisfies the optimality principle unless there is a cycle with negative cost).

Theorem 1. *Let $\mathcal{G}' = (\mathcal{N}, \mathcal{A}, c')$ be a network where $c'_{i,j} = \sum_{l=1}^{k} c^l_{i,j}$, $\forall (i, j) \in \mathcal{A}$ and $f'(p) = \sum_{(i,j)\in p} c'_{i,j}$, $p \in \mathcal{P}_{\mathcal{G}'}$. Then, the (minSum) problem on $\mathcal{G}$ and $\min\{f'(p) : p \in \mathcal{P}'\}$, where $\mathcal{P}'$ is the set of paths from s to t in $\mathcal{G}'$, are equivalent.*

Proof. Since the underlying graph of the networks $\mathcal{G}$ and $\mathcal{G}'$ is the same, then both networks have the same set of paths, that is, $\mathcal{P}_{\mathcal{G}} = \mathcal{P}_{\mathcal{G}'}$ (in particular, $\mathcal{P} = \mathcal{P}'$). Moreover, since for $p \in \mathcal{P}_{\mathcal{G}}$, $f_{sum}(p) = \sum_{l=1}^{k} \sum_{(i,j)\in p} c^l_{i,j} = \sum_{(i,j)\in p} c'_{i,j} = f'(p)$, we conclude that both problems are equivalent. ■

The value $\|f(p^*_{sum})\|$ is a good approximation to $\|f(p^*_{norm})\|$ and the path p^*_{sum} may be considered as a good initial solution for the (minNorm) problem. In Table 1, we present some empirical results averaged over 50 instances of each type. They show that, for random instances, p^*_{norm} and p^*_{sum} do not differ much from each other, in terms of the corresponding values. In fact, the last two columns of the table show that the gap between the values of f_{norm} and f_{sum} for p^*_{norm} and p^*_{sum} is less than 0.5%.

Nevertheless, theoretically, the value of $\|f(p^*_{sum})\|$ may be significantly different from $\|f(p^*_{norm})\|$. The following theorem provides variation bounds for $\|f(p^*_{norm})\|$ in terms of $f(p^*_{sum})$.

Theorem 2. *Let p^*_{norm} and p^*_{sum} be the optimal solution for, respectively, the (minNorm) and (minSum) problems. Then, $\sum_{i=1}^{k} f_i(p^*_{sum})/\sqrt{k} \le \|f(p^*_{norm})\| \le \|f(p^*_{sum})\|$.*

Proof. The inequality $\|f(p^*_{norm})\| \le \|f(p^*_{sum})\|$ is obvious since p^*_{norm} is the optimal solution for (minNorm).

Now, let us consider the following problem:

$$\min\{\|f(q)\| : q \in \mathcal{P} \text{ and } f_{sum}(q) = \beta\}. \qquad (\text{P}1_\beta)$$

By definition, p^*_{norm} is an optimal solution for $(\text{P}1_\beta)$ with $\beta = f_{sum}(p^*_{norm})$ and we can get a lower bound for $(\text{P}1_\beta)$ by solving the following problem:

$$\min \sqrt{\sum_{i=1}^{k} x_i^2} \qquad (\text{P}1_\beta)$$

$$\text{s.t.} \sum_{i=1}^{k} x_i = \beta. \qquad (\text{P}2_\beta)$$

Using Lagrange multipliers, the optimal solution for $(\text{P}2_\beta)$ is $x_i = \beta/k$ $(i = 1, \ldots, k)$ with optimal value $\sum_{i=1}^{k} x_i/\sqrt{k} = \beta/\sqrt{k}$. Since the minimum value of β in $(\text{P}1_\beta)$ is $\beta = f_{sum}(p^*_{sum})$, then $f_{sum}(p^*_{sum})/\sqrt{k} \le \|f(p^*_{norm})\|$. ■

Corollary 2. *$\|f(p^*_{norm})\|$ is at most $\sqrt{k}$ times smaller than $\|f(p^*_{sum})\|$.*

Proof. From the previous theorem, $\|f(p^*_{norm})\| \ge \sum_{i=1}^{k} f_i(p^*_{sum})/\sqrt{k}$. Since we assume that $f_i(q) \ge 0$, $\forall q \in \mathcal{P}$, $\forall i = 1, \ldots, k$, then $\|f(p^*_{sum})\|^2 = \sum_{i=1}^{k} f_i^2(p^*_{sum}) \le (\sum_{i=1}^{k} f_i(p^*_{sum}))^2$, and this proves the corollary. ■

Therefore, the best theoretical improvement on $\|f(p^*_{sum})\|$ is

$$100\left(\frac{\|f(p^*_{sum})\| - \|f(p^*_{norm})\|}{\|f(p^*_{sum})\|}\right)\% = 100\left(1 - \frac{1}{\sqrt{k}}\right)\%.$$

This value increases with the number of objectives, being approximately 30% in the biobjective case ($k = 2$).

3. ALGORITHMS

In this section, we describe the two algorithms proposed in this work. Although both start with the determination of p^*_{sum}, they use different strategies for solving the (minNorm) problem. One is based on the labeling algorithm for the MSPP and the other one on a procedure to rank paths from s to t.

3.1. Labeling Algorithm

The labeling algorithm for the MSPP determines all the nondominated paths. Thus, taking into account Lemma 1 ($p^*_{norm} \in \bar{\mathscr{D}}$), one possible algorithm for the (minNorm) problem consists of the following two main steps:

1. Determine all the nondominated paths from s to t ($\bar{\mathscr{D}}$).
2. Choose from $\bar{\mathscr{D}}$ the path p^*_{norm} with minimum cost.

This process avoids the direct determination of p^*_{norm}, which is harder to obtain because the (minNorm) problem does not satisfy the optimality principle, but it turns out to be inefficient when the number of nondominated paths is large. The complexity of the labeling algorithm depends on the number of nondominated paths, which is difficult to estimate. Henig [13] proposed the expression $\#\bar{\mathscr{D}} = \log(\#\mathscr{P})$ as a rough estimate of that number in the biobjective case. However, Hansen [12] showed that this number may grow exponentially with the number of nodes.

The algorithm proposed in this section is an adaptation of the labeling algorithm where the result stated by Corollary 1 is used to avoid the exhaustive determination of nondominated solutions. Recall that, by Corollary 1, if $\mathbf{a}$ and $\mathbf{b}$ are two nondominated labels at a specific node with $\|\mathbf{a}\| < \|\mathbf{b}\|$, then there exists $\mathbf{c} \in (\mathbb{R}^+_0)^k$ such that $\|\mathbf{a} + \mathbf{c}\| > \|\mathbf{b} + \mathbf{c}\|$. However, that value for $\mathbf{c}$ may lead to a situation where $\|\mathbf{b} + \mathbf{c}\|$ is greater than an upper bound on the optimum value given by the norm of a known path from s to t. In this case, $\mathbf{b}$ is no longer considered in the searching procedure. In short, the algorithm has the following scheme:

1. Determine the path p^*_{sum} and let $\alpha = \|f(p^*_{sum})\|$.
2. Apply the labeling algorithm for the MSPP with an optimality condition which allows us to keep a new efficient label for a specific node, namely, suppose that $\mathbf{a}$ is a label of the node j already found and $\mathbf{b}$ is a new nondominated label of j created by the labeling algorithm. Then, $\mathbf{b}$ must be kept if there is $\mathbf{c} \in (\mathbb{R}^+_0)^k$ such that $\|\mathbf{a} + \mathbf{c}\| > \|\mathbf{b} + \mathbf{c}\|$ and $\|\mathbf{b} + \mathbf{c}\| < \alpha$.

The condition referred to in the second step is described in the following theorem:

Theorem 3. *Let $\Pi^*_i(j)$, $i \in \{1, \ldots, k\}$, $j \in \mathscr{N}$, be the cost of the shortest path from j to t when just the i-th cost component of the arc is considered. Let $\mathbf{a}$ and $\mathbf{b}$ be two nondominated labels of a node $j \in \mathscr{N}$ such that $\|\mathbf{a} + \Pi^*(j)\| < \|\mathbf{b} + \Pi^*(j)\|$ and let $I = \{i \in \{1, \ldots, k\} : b_i < a_i\}$. Then, an optimal solution of the minimum-cost norm path problem can be found from the label $\mathbf{b}$ of the node j if there is any $S \subseteq I$ such that $\|\mathbf{b} + \mathbf{c}\| \leq \alpha$, where the components of $\mathbf{c}$ are defined by*

$$c_i = \Pi^*_i(j),\ \textit{if}\ i \notin S; \quad c_i = \lambda(a_i - b_i) - b_i,\ \textit{if}\ i \in S$$

with

$$\lambda = \frac{\|\mathbf{b} + \Pi^*(j)\|^2 - \|\mathbf{a} + \Pi^*(j)\|^2 + 2\sum_{l \in S}(a_l - b_l) \times (b_l + \Pi^*_l(j))}{2\sum_{l \in S}(a_l - b_l)^2}$$

satisfying

$$\lambda \geq \max_{l \in S}\left\{\frac{b_l + \Pi^*_l(j)}{a_l - b_l}\right\}.$$

Proof. To simplify the proof of this theorem, let us note that the path which minimizes $\|f(q)\|$, $q \in \mathscr{P}$, also minimizes $\|f(q)\|^2 = \sum_{l=1}^k (f_l(q))^2$. Moreover, since $\mathbf{b}$ is a label of j, then every label $\mathbf{b} + \mathbf{c}$ of a path from s to t satisfies $\Pi^*(j) \leq_{\mathbb{R}^k} \mathbf{c}$. Then, $c_i = \Pi^*_i(j) + x_i^2$, $i = 1, \ldots, k$, with $\mathbf{x} \in \mathbb{R}^k$. Let be $\mathbf{a}' = \mathbf{a} + \Pi^*(j)$, $\mathbf{b}' = \mathbf{b} + \Pi^*(j)$, and $\mathbf{x}'$ with $x'_i = x_i^2$, $i = 1, \ldots, k$; $\mathbf{a} + \mathbf{c}$ and $\mathbf{b} + \mathbf{c}$ will be written as $\mathbf{a}' + \mathbf{x}'$ and $\mathbf{b}' + \mathbf{x}'$, respectively. In this form, we want to know if

$$\min_{\mathbf{x} \in \mathbb{R}^k} \{\|\mathbf{b}' + \mathbf{x}'\|^2 : \|\mathbf{b}' + \mathbf{x}'\|^2 \leq \|\mathbf{a}' + \mathbf{x}'\|^2\} \leq \alpha^2, \qquad (1)$$

where $\mathbf{x}'$ depends on $\mathbf{x}$. The optimal solution of problem (1) can be found using Lagrange multipliers. The restriction $\|\mathbf{b}' + \mathbf{x}'\|^2 \leq \|\mathbf{a}' + \mathbf{x}'\|^2$ needs to be written in the form $h(\cdot) = 0$, that is, $h(\mathbf{x}, y) = \|\mathbf{b}' + \mathbf{x}'\|^2 + y^2 - \|\mathbf{a}' + \mathbf{x}'\|^2 = 0$, for $y \in \mathbb{R}$. The optimal solution of this problem is obtained from the solutions of the following equations:

$$\frac{\partial}{\partial x_i}(\|\mathbf{b}' + \mathbf{x}'\|^2 + \lambda h(\mathbf{x}, y)) = 4x_i(b'_i + x_i^2) + 4\lambda x_i(b_i - a_i) = 0,\ i \in \{1, \ldots, k\} \qquad (2)$$

$$\frac{\partial}{\partial y}(\|\mathbf{b}' + \mathbf{x}'\|^2 + \lambda h(\mathbf{x}, y)) = 2\lambda y = 0 \qquad (3)$$

$$\frac{\partial}{\partial \lambda}(\|\mathbf{b}' + \mathbf{x}'\|^2 + \lambda h(\mathbf{x}, y)) = h(\mathbf{x}, y) = 0. \qquad (4)$$

From Eq. (3), we obtain $\lambda = 0$ or $y = 0$. If $\lambda = 0$, using Eq. (2), we get $x_i = 0$ or $x_i^2 = -b'_i$. As $-b'_i \leq 0$; then, $x_i = 0$. However, $\mathbf{x} = 0_{\mathbb{R}^k}$ does not satisfy Eq. (4). Thus, $\lambda \neq 0$ and then $y = 0$.

From Eq. (2), the conditions on the components of $\mathbf{x}$ are $x_i = 0$ or $x_i^2 = -b'_i - \lambda(b_i - a_i)$, $i = 1, \ldots, k$.

Equation (4) now gives $h(\mathbf{x}, 0) = \|\mathbf{b}'\|^2 - \|\mathbf{a}'\|^2 + 2 \sum_{l=1}^{k} (b_l - a_l)x_l^2 = 0$. Since $h(0_{\mathbb{R}^k}, 0) = \|\mathbf{b}'\|^2 - \|\mathbf{a}'\|^2 > 0$, then $\mathbf{x} = 0_{\mathbb{R}^k}$ does not satisfy this equation. We need to know which components of $\mathbf{x}$ would be different from zero in order to decrease the value of $h(\mathbf{x}, 0)$. Since the value of x_l^2 can only increase, we are only interested in the components x_l such that $b_l - a_l < 0$. Thus, let I be the set $\{i \in \{1, \ldots, k\} : b_i < a_i\}$. From what we have explained, $x_i = 0$, $\forall i \notin I$ and $x_i = 0$ or $x_i^2 = \lambda(a_i - b_i) - b'_i$, $\forall i \in I$. Now, we consider a subset of indices $S \subseteq I$ and let x_i^2 be $\lambda(a_i - b_i) - b'_i$ only for $i \in S$. To determine the value of λ, we substitute the expression for x_i^2 into Eq. (4) to obtain $h(\mathbf{x}, 0) = \|\mathbf{b}'\|^2 - \|\mathbf{a}'\|^2 - 2 \sum_{l \in S} (a_l - b_l)(\lambda(a_l - b_l) - b'_l) = 0$, that is,

$$\lambda = \frac{\|\mathbf{b}'\|^2 - \|\mathbf{a}'\|^2 + 2 \sum\limits_{l \in S} (a_l - b_l)b'_l}{2 \sum\limits_{l \in S} (a_l - b_l)^2} .$$

However, λ is feasible if $x_i^2 = \lambda(a_i - b_i) - b'_i \geq 0$, $i \in S$ or, in other words, if $\lambda \geq b'_i/(a_i - b_i)$, $\forall i \in S$. Therefore, $\min_{\mathbf{x} \in \mathbb{R}^k} \{\|\mathbf{b}' + \mathbf{x}'\|^2 : h(\mathbf{x}, 0) = 0\}$ is obtained at one of these arrangements, so it is enough to verify if one of them satisfies $\|\mathbf{b}' + \mathbf{x}'\| \leq \alpha$. ∎

The condition given by this theorem depends on the labels **a** and **b**. In this work, we suggest keeping, for each node $j \in \mathcal{N}$, the path with the minimum-cost norm among all the paths from s to j already determined. The label of this path will be the label **a** of the node j, decreasing, in this way, the probability of keeping new labels for this node. The complexity of this algorithm is, in the worst case, the same as that of the labeling algorithm. Theorem 3 allows us to avoid determining some nondominated labels, but it is difficult to estimate how many labels can be discarded. Algorithm 1 details the process described in this subsection.

3.2. Ranking Algorithm

Now, we propose another algorithm, based on a ranking path procedure, to determine the minimum-cost norm path. The paths are ranked by a nondecreasing order of $f_{sum}(p)$, $p \in \mathcal{P}$. In this way, paths of the (minSum) problem are ranked until we can guarantee that the optimal solution of the (minNorm) problem has been found; see Algorithm 2. As shown in [16], the complexity of this algorithm is $\mathcal{O}(m \log(n) + Kn)$, where K is the number of paths ranked (unknown in advance).

From Table 1, we conclude that the optimal cost norm path values of (minNorm) and of (minSum) are "closer" to each other, in the general case. So, we should need to rank relatively few paths in the (minSum) problem to obtain the desired solution. As suggested by Figure 3, the cost of p^*_{sum} produces a deep restriction in the region where the cost of the path p^*_{norm} can be found. Theorem 4 establishes a relation between the optimal paths p^*_{sum} and p^*_{norm} and gives a stopping condition to finish ranking the paths in the (minSum) problem.

Algorithm 1: Labeling algorithm for the (minNorm) problem

```
Input:    a network $(\mathcal{N}, \mathcal{A}, c)$ and $s, t \in \mathcal{N}$
Output:   an optimal path $p^*$ for the (minNorm) problem
          {$X$: set of labels that the algorithm
          needs to analyze}
          {$node(x)$: node that corresponds to the
          label $x$}
          {$path(x)$: path of $\mathcal{G}$ that originates the
          label $x$}
Step 1:   {initialization}
          $count \leftarrow 1$; $X \leftarrow \{count\}$; $node(count) \leftarrow s$; $path(count)$
          $\leftarrow \langle s \rangle$
          $minNorm(s) \leftarrow 0$; $rotMin(s) \leftarrow count$; $\alpha \leftarrow \|f(p^*_{sum})\|$
          $minNorm(i) \leftarrow \infty$, $\forall i \in \mathcal{N} \backslash \{s\}$
Step 2:   {shortest value per node and component}
          $\Pi^*_l(i) \leftarrow$ cost of the shortest path from $i$ to $t$
                  using only the $l$-th component, $\forall i \in \mathcal{N}$ and $\forall l$
                  $\in \{1, \ldots, k\}$
Step 3:   {adaptation of labeling algorithm}
          if $(X = \emptyset)$
          then goto step 8
          else $x \leftarrow$ first label of $X$ using the FIFO rule
            $X \leftarrow X \backslash \{x\}$; $i \leftarrow node(x)$; $A_i \leftarrow \{(i, j) : (i, j) \in \mathcal{A}\}$
Step 4:     {for all arcs leaving node $i$}
            if $(A_i = \emptyset)$
            then goto step 3
            else $(i, j) \leftarrow$ an element of $A_i$
              $A_i \leftarrow A_i \backslash \{(i, j)\}$; $q \leftarrow path(x) \diamond \langle i, (i, j), j \rangle$
Step 5:       {bounding test}
              if $(\|f(q) + \Pi^*(j)\| \geq \alpha)$
              then goto step 3
              else if $(\|f(q)\| < minNorm(j))$
                then goto step 6
                else goto step 7;
Step 6:       {new $minNorm(j)$ value}
              $count \leftarrow count + 1$; $X \leftarrow X \cup$
              $\{count\}$; $node(count) \leftarrow j$
              $path(count) \leftarrow q$; $minNorm(j) \leftarrow$
              $\|f(q)\|$; $rotMin(j) \leftarrow count$
              Remove the labels of $j$ dominated by $f(q)$
              goto step 4;
Step 7:       {optimality condition}
              if $f(q)$ does not satisfy the condition stated in Theorem 3
                  using the label $rotMin(j)$
              then goto step 4
              else $count \leftarrow count + 1$; $X \leftarrow X \cup$
              $\{count\}$; $node(count) \leftarrow j$
                $path(count) \leftarrow q$; Remove the labels of $j$
                dominated by $f(q)$
                goto step 4;
Step 8:   {optimal solution of the (minNorm)
          problem}
          $p^* \leftarrow path(rotMin(t))$
```

Algorithm 2: Ranking algorithm for the (minNorm) problem

```
Input:   a network $(\mathcal{N}, \mathcal{A}, c)$ and $s, t \in \mathcal{N}$
Output:  an optimal path $p^*$ for the (minNorm) problem
Step 1:  {initialization}
         $c'_{i,j} \leftarrow \sum_{l=1}^{k} c^l_{i,j}, \forall (i, j) \in \mathcal{A}$
         $p \leftarrow$ shortest path from $s$ to $t$ in $(\mathcal{N}, \mathcal{A}, c')$ $\{p \leftarrow p^*_{sum}\}$
         $p^* \leftarrow p;\ \alpha \leftarrow \|f(p^*)\|$
Step 2:  {stopping condition}
         if $f_{sum}(p) \geq \sqrt{k}\alpha$
         then STOP
         else $p \leftarrow$ next shortest path from $s$ to $t$ in $(\mathcal{N}, \mathcal{A}, c')$
           if $(\|f(p)\| < \alpha)$
           then $p^* \leftarrow p;\ \alpha \leftarrow \|f(p^*)\|$;
           goto step 2
```

Theorem 4. *Let p^*_{sum} and p^*_{norm} be the optimal paths of the (minSum) and (minNorm) problems, respectively. Then, $\sum_{i=1}^{k} f_i(p^*_{norm}) \leq \sqrt{k}\|f(p^*_{sum})\|$, that is, $f_{sum}(p^*_{norm}) \leq \sqrt{k} f_{norm}(p^*_{sum})$.*

Proof. First, let us consider the following problem:

$$\max\{f_{sum}(q) : q \in \mathcal{P} \text{ and } \|f(q)\| = \beta\}. \qquad (\mathrm{P3}_\beta)$$

By definition, $f_{sum}(p^*_{norm})$ is less than or equal to the value of an optimal solution for $(\mathrm{P3}_\beta)$ with $\beta = \|f(p^*_{norm})\|$. We can get an upper bound for $(\mathrm{P3}_\beta)$ by solving the following problem:

$$\max \sum_{i=1}^{k} x_i$$

$$\text{s.t.} \sum_{i=1}^{k} x_i^2 = \beta^2. \qquad (P4_\beta)$$

Using again Lagrange multipliers, the optimal solution is $x_i = \beta/\sqrt{k}$ $(i = 1, \ldots, k)$, and $\sum_{i=1}^{k} x_i = \sqrt{k}\beta$ is the optimal value. Then, an upper bound on $f_{sum}(p^*_{norm})$ is $\sqrt{k}\|f(p^*_{norm})\|$. As $\|f(p^*_{norm})\| \leq \|f(p^*_{sum})\|$, we conclude the proof of this theorem. ∎

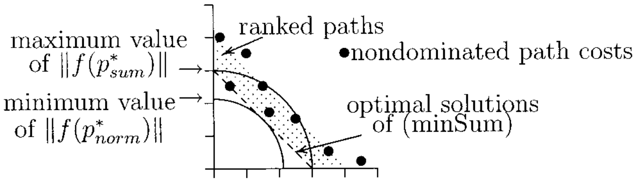
maximum value of $\|f(p^*_{sum})\|$
minimum value of $\|f(p^*_{norm})\|$
ranked paths
•nondominated path costs
optimal solutions of (minSum)

FIG. 3. Relative positions of $f(p^*_{sum})$ and $f(p^*_{norm})$.

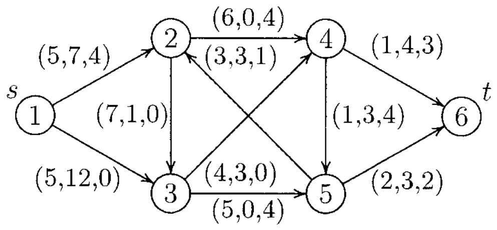

FIG. 4. Instance used to illustrate the algorithms.

4. EXAMPLE

In this section, the example given in Figure 4 (with $k = 3$) is used to illustrate the algorithms described in the previous section.

4.1. Computing $\bar{\mathcal{D}}$

As mentioned before, the (minNorm) problem can be solved by determining the full set of nondominated paths from s to t and then selecting the one with the least-cost norm. Computing $\bar{\mathcal{D}}$ is easily done by using, for example, a label-correcting algorithm [12], building up a tree with the paths starting at s and leading to the other nodes of the network.

Figure 5 shows how this procedure works for the given example, generating a tree where the dotted branches mean that the label of the corresponding path from s to, say, j is dominated by some other label already computed for a path linking s to the same node j. For this example, one finds four nondominated paths, with ⟨1, 2, 4, 6⟩ being the one with the minimum-cost norm while ⟨1, 3, 4, 6⟩ is the optimal solution for the (minSum) problem. Let us point out that, for this small example, a total of 14 labels were kept after applying the dominance test.

4.2. Algorithm 1

This algorithm corresponds to an enhancement of the labeling algorithm by incorporating the bounding test and the optimality condition stated in Theorem 3. For that, one needs to compute, as a first step, the values $\Pi^*_l(i)$, $i = 1, 2, \ldots, 6$, $l = 1, 2, 3$, shown in Table 2. Also, p^*_{sum} is determined and $\alpha = \|f(p^*_{sum})\|$ is used as an initial upper bound on the optimal value for (minNorm).

The search tree starts from node 1, labeled (0, 0, 0), generating the labels (5, 7, 4) and (5, 12, 0) for nodes 2 and 3, respectively. Following a FIFO rule, one looks now for an expansion of the path ⟨1, 2⟩ creating a new label (12, 8, 4) for node 3 and labeling, for the first time, node 4 with (11, 7, 8).

Note that, so far, the bounding test has not been effective for any of the created labels. However, since there are two labels for node 3, the optimality condition of Theorem 3 can be applied taking $\mathbf{a} = f(\langle 1, 3\rangle) = (5, 12, 0)$ and $\mathbf{b} = f(\langle 1, 2, 3\rangle) = (12, 8, 4)$. Since $\|\mathbf{a} + \Pi^*(3)\| = \|(10, 15, 3)\| =$

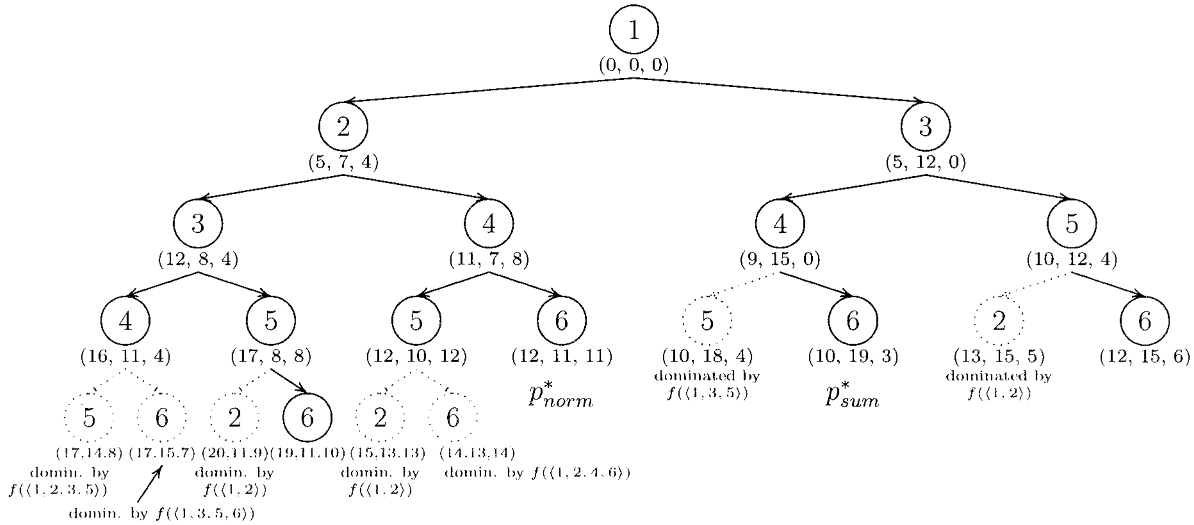

FIG. 5. Tree produced by the label-correcting algorithm.

$\sqrt{334} < \sqrt{459} = \|(17, 11, 7)\| = \|\mathbf{b} + \Pi^*(3)\|$, one may check if p^*_{norm} can be obtained from the path $\langle 1, 2, 3 \rangle$. According to Theorem 3, the array (5, 18.625, 3) is the smallest value for $\mathbf{c}$ such that $\|\mathbf{b} + \mathbf{c}\| \leq \|\mathbf{a} + \mathbf{c}\|$. However, $\|\mathbf{b} + \mathbf{c}\| = \|(17, 26.625, 7)\| = \sqrt{1046.89}$ will then be greater than α, which discards the label as a potential element for the construction of p^*_{norm}.

The next label to be chosen is $f(\langle 1, 3 \rangle) = (5, 12, 0)$, which generates the labels (9, 15, 0) for node 4 and (10, 12, 4) for node 5. The former label is canceled by the bounding test and the algorithm proceeds by looking for an expansion of the path $\langle 1, 2, 4 \rangle$. However, to illustrate the calculations of the optimality condition, we can verify that label $\mathbf{b} = f(\langle 1, 3, 4 \rangle) = (9, 15, 0)$ does not satisfy it when compared with label $\mathbf{a} = f(\langle 1, 2, 4 \rangle) = (11, 7, 8)$. In fact, none of the three vectors obtained from Theorem 3, $\mathbf{c}' = (22, 4, 3)$, $\mathbf{c}'' = (1, 4, 8.25)$, and $\mathbf{c}''' = (-6.471, 4, 10.12)$, satisfy the optimality condition.

The full search tree built by the algorithm for the example is depicted in Figure 6. One can see that a total of six labels were created and another five labels were computed but ignored after performing the tests.

4.3. Algorithm 2

In this case, the algorithm corresponds to using the optimality condition stated by Theorem 4 within the framework of the ranking algorithm presented in [16]. More precisely, the paths are ranked by increasing order of the corresponding f_{sum} value until one gets a path $p \in \mathcal{P}$ such that $f_{sum}(p) \geq \sqrt{k}\alpha$, where α is the best value of f_{norm} over the set of the computed paths.

In our example, the path $q_1 = p^*_{sum} = \langle 1, 3, 4, 6 \rangle$ is the first one to be produced by the algorithm with $f_{sum}(q_1) = 32$ and $\alpha = f_{norm}(q_1) = \sqrt{470}$. From q_1, the algorithm generates the following deviation paths: $q_2 = \langle 1, 2, 4, 6 \rangle$, $q_3 = \langle 1, 3, 5, 6 \rangle$, and $q_4 = \langle 1, 3, 4, 5, 6 \rangle$. Among these, q_3 ($f_{sum}(q_3) = 33$) is selected as the second shortest path and α is updated to the value $f_{norm}(q_3) = \sqrt{405}$.

Then, $q_5 = \langle 1, 3, 5, 2, 4, 6 \rangle$ is a new deviation path derived from q_3 and the third shortest path is obtained from the set $\{q_2, q_4, q_5\}$, that is, q_2 ($f_{sum}(q_2) = 34$) with $f_{norm}(q_2) = \sqrt{386}$, leading to a new updating of α. The ranking procedure generates two deviation paths from q_2: $q_6 = \langle 1, 2, 3, 4, 6 \rangle$ and $q_7 = \langle 1, 2, 4, 5, 6 \rangle$.

Now, the path $q_6 = \langle 1, 2, 3, 4, 6 \rangle$ turns out to be the fourth shortest path with $f_{sum}(q_6) = 39 > \sqrt{3}\sqrt{386}$, meaning that the algorithm stops with $p^*_{norm} = q_2$.

In this example, four paths were ranked and another three deviation paths were identified, with the ranking condition value being updated twice (see Fig. 7).

TABLE 2. Cost of the shortest path from i, $i \in \mathcal{N}$, to $t = 6$ for each criterion.

i	1	2	3	4	5	6
$\Pi^*_1(i)$	10	7	5	1	2	0
$\Pi^*_2(i)$	11	4	3	4	3	0
$\Pi^*_3(i)$	3	3	3	3	2	0

5. COMPUTATIONAL RESULTS

The algorithms were computationally tested on a set of instances generated over two types of networks and considering three different distributions for the costs assigned to the arcs. The computational experiments were carried out at CISUC (University of Coimbra), using a 266-MHz Pentium II biprocessor with 256 Mb of RAM.

The two types of networks considered for test instances are the following:

- Random: A Hamiltonian cycle is first randomly generated to guarantee connectivity of the graph. Then, the remain-

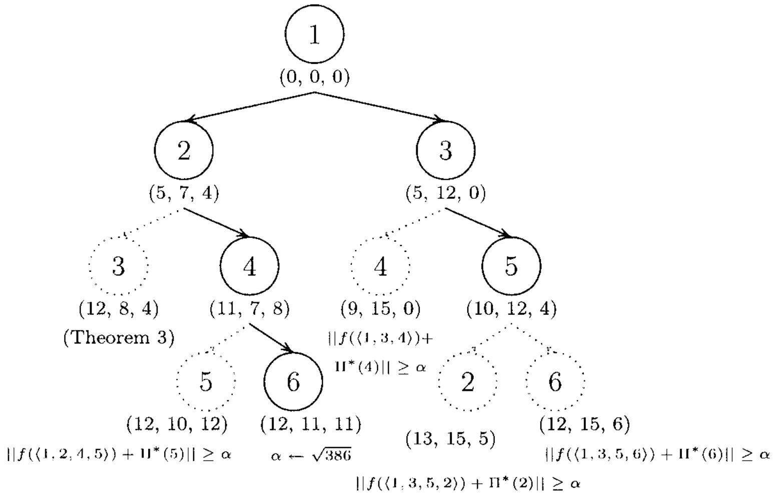
1
(0, 0, 0)
2
(5, 7, 4)
3
(5, 12, 0)
3
(12, 8, 4)
(Theorem 3)
4
(11, 7, 8)
4
(9, 15, 0)
$||f(\langle 1,3,4\rangle)+$
$\Pi^*(4)|| \geq \alpha$
5
(10, 12, 4)
5
(12, 10, 12)
$||f(\langle 1,2,4,5\rangle) + \Pi^*(5)|| \geq \alpha$
6
(12, 11, 11)
$\alpha \leftarrow \sqrt{386}$
2
(13, 15, 5)
$||f(\langle 1,3,5,2\rangle) + \Pi^*(2)|| \geq \alpha$
6
(12, 15, 6)
$||f(\langle 1,3,5,6\rangle) + \Pi^*(6)|| \geq \alpha$

FIG. 6. Tree produced by Algorithm 1.

ing $m - n$ arcs are randomly generated with multiple arcs being excluded ($n = |\mathcal{N}|$ and $m = |\mathcal{A}|$).

- Grid: The nodes correspond to the points of a square mesh with each one of them being linked to its nearest four neighbors. The initial node, $s = 1$, is placed at the top-left corner while the terminal node, $t = n$, is located at the bottom right corner.

Recall from Table 1 that the optimal paths for f_{sum} and f_{norm} are very close to each other for instances with a uniform distribution for the arc costs, regardless the type of network. Hence, to try out instances with a larger gap between those optimal values, two other distributions were considered for the costs. In fact, as can be seen in Table 3, which shows the average results obtained with 50 instances of each type, the gap between the optimal paths for f_{sum} and f_{norm} significantly increases, for both types of networks, when "circular" or "cubic" distributions are used.

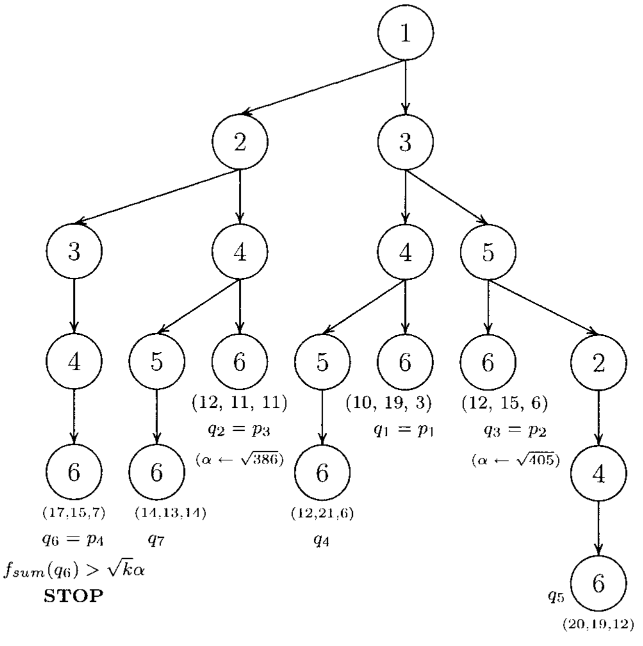

FIG. 7. Tree produced by Algorithm 2.

Thus, the following three cost distributions were studied:

- Uniform: The costs are randomly generated from the set $\{1, 2, \ldots, 1000\}$.
- Cubic: For each arc (i, j), all but one of the components of its cost are zero. The index of the nonzero element is randomly chosen from the set $\{1, 2, \ldots, k\}$ and the corresponding cost is made equal to an integer in the interval $[0.8 \times 1000/i, 1.2 \times 1000/i]$. This means that $\|\mathbf{c_{i,j}}\| \approx 1000/i$, $\forall(i, j) \in \mathcal{A}$, with a maximum deviation of plus or minus 20%. The initial and the terminal nodes are set equal to 1 and n, respectively.
- Circular: The cost components are randomly generated but satisfy the condition $\|\mathbf{c_{i,j}}\| = 1000/i$, $\forall(i, j) \in \mathcal{A}$. As in the previous case, $s = 1$ and $t = n$.

We note that the number of paths generated by the ranking procedure is also significantly increased when the "circular" and the "cubic" distributions are considered. This is shown in Table 4, where, in particular, the number of ranked paths for the grid networks with a "cubic" distribution becomes too high for computational purposes.

Tables 5–8 report computational results with the algorithms for the different test instances. All tables contain results based on the average CPU time (column *sec*) and the average memory (column *Mb*) required by the algorithms for solving 50 instances with the characteristics described in the respective table. Also, the number of created labels (column *labels*) is presented for the labeling algorithms, while the number of deviation paths determined (column *paths*) is exhibited for Algorithm 2—which corresponds, in such a way, to the number of labels generated by the labeling algorithms.

Table 5 compares the label-correcting algorithm with the algorithms proposed here. It shows a drastic reduction in the number of determined labels when using Algorithm 1. Also,

TABLE 3. Comparison of optimal value of the (minnorm) and (minsum) problems.

Network				p^*_{norm}		p^*_{sum}		%	
Type	n	m	k	f_{norm}	f_{sum}	f_{norm}	f_{sum}	(a)	(b)
				Circular					
Random	1000	5000	2	207	288	207	288	0.00	0.00
Random	1000	5000	5	201	397	206	393	2.62	0.96
Random	1000	5000	10	198	426	204	414	**2.78**	2.90
Grid	900	3480	2	3232	4516	3262	4488	0.94	0.61
Grid	900	3480	5	2826	5741	2955	5586	4.36	2.70
Grid	900	3480	10	2639	7489	2774	7407	**4.86**	1.09
				Cubic					
Random	1000	5000	2	900	946	909	921	**1.00**	2.70
Random	1000	5000	5	886	915	889	906	0.33	0.96
Random	1000	5000	10	891	918	896	909	0.59	1.05
Grid	**100**	360	2	1138	1431	1211	1365	6.05	4.61
Grid	**100**	360	5	1014	1441	1085	1359	6.57	5.64
Grid	**100**	360	10	965	1378	1043	1357	**7.55**	1.46

(a) $100 \times [(f_{norm}(p^*_{sum}) - f_{norm}(p^*_{norm}))/(f_{norm}(p^*_{sum}))]$; (b) $100 \times [(f_{sum}(p^*_{norm}) - f_{sum}(p^*_{sum}))/(f_{sum}(p^*_{norm}))]$.

with the exception of grid networks with a "cubic" cost distribution, one can see that the number of paths generated by Algorithm 2 is usually smaller than is the number of labels generated by the labeling algorithms. Also, Table 5 indicates that the networks with a "cubic" cost distribution are the hardest to solve, due mainly to the large number of nondominated paths.

Tables 6–8 refer to the computational experiments carried out with the new algorithms for different size instances, in terms of the number of nodes (n), the density (d), and the number of criteria (k).

Table 6 reports on the results obtained when the number of nodes increases for the two types of networks and the different cost distributions, keeping $d = 4$ and $k = 6$. From

TABLE 4. No. paths ranked in the (minsum) problem to obtain the optimal solution of the (minnorm) problem.

Type	n	m	k	Uniform	Circular	Cubic
Random	1000	5000	2	2	9	134,803
Random	1000	5000	5	2	102,736	111,328
Random	1000	5000	10	3	71,604	66,785
Grid	900	3480	2	4	120	$>10^6$
Grid	900	3480	5	5	44,029	$>10^6$
Grid	900	3480	10	102	19,080	$>10^6$

TABLE 5. The labeling algorithm versus the new algorithms.

		Labeling algorithm			Algorithm 1			Algorithm 2		
Cost distribution	$\#\bar{\mathcal{D}}$	Sec	Mb	Labels	Sec	Mb	Labels	Sec	Mb	Paths
				Random networks with 100 nodes, 400 arcs, and 6 criteria						
Uniform	12.84	0.00	0.64	1479.20	0.00	0.29	145.88	0.00	0.68	6.72
Circular	15.36	0.01	0.71	1697.32	0.00	0.29	150.50	0.00	0.27	18.58
Cubic	—	—	—	—	0.40	4.21	12,986.94	—	—	—
				Grid networks with 49 nodes, 168 arcs, and 6 criteria						
Uniform	328.08	0.01	0.87	2598.02	0.00	0.16	185.52	0.00	0.34	55.76
Circular	548.74	0.04	1.66	5185.34	0.00	0.17	209.84	0.01	0.47	902.36
Cubic	11,688.46	17.82	159.05	520,903.28	0.02	1.00	2917.18	105.03	79.01	215,421.1

TABLE 6. Algorithm performance with n ($d = 4$, $k = 6$).

Random networks											
		10,000		20,000		30,000		40,000		50,000	
Cost distribution	n	Sec	Mb	Sec	Mb	Sec	Mb	Sec	Mb	Sec	Mb
Uniform	Algm 1	0.20	29.08	0.09	58.41	0.27	87.64	0.43	116.73	0.17	131.02
	Algm 2	0.01	18.79	0.03	36.85	0.04	54.88	0.08	72.93	0.22	81.85
Circular	Algm 1	0.06	30.90	0.08	61.83	0.31	92.95	0.42	125.01	0.37	144.19
	Algm 2	0.04	18.11	0.03	36.16	0.09	55.91	0.13	72.15	0.29	81.23
	n	100		200		300		400		500	
Cubic	Algm 1	0.49	4.21	1.07	15.08	1.51	68.69	—	—	—	—
	Algm 2	—	—	—	—	—	—	—	—	—	—
Square grid networks											
Cost distribution	n	200		400		600		800		1000	
Uniform	Algm 1	0.01	1.33	0.09	5.94	0.22	15.23	0.20	28.17	0.85	64.81
	Algm 2	0.00	0.65	0.01	1.13	0.04	1.61	0.07	2.07	0.02	2.55
	n	20		40		60		80		100	
Circular	Algm 1	0.00	0.06	0.00	0.11	0.00	0.20	0.00	0.28	0.01	0.43
	Algm 2	0.00	0.13	0.01	0.26	0.02	0.67	0.13	1.92	0.39	4.64
Cubic	Algm 1	0.00	0.07	0.01	0.28	0.15	2.24	0.54	12.30	—	—
	Algm 2	0.02	0.43	1.29	11.48	—	—	—	—	—	—

this table, it is clear that the two algorithms are capable of tackling very large size random networks when the costs for the arcs are generated with a uniform or a circular distribution. In fact, all those instances were solved in less than 1 second of CPU time, with a slight superiority seen for Algorithm 2. The same is observed for square grid networks with uniform costs but with a much smaller number of nodes.

Actually, n has to decrease even further when one considers a cubic distribution for both types of networks, to keep the same level of performance for the algorithms. In this case, Algorithm 2 failed for most of the instances and Algorithm 1 clearly becomes a more effective approach. The two algorithms show the same kind of performance when one increases the density (Table 7) or the number of criteria (Table 8), keeping a fixed number of nodes for the networks. However, with the exception of grid networks with a uniform cost distribution, Algorithm 2 appears to be more sensitive to changes in n, d, or k.

6. CONCLUSIONS

In this work, a utility function based on the Euclidean norm was proposed for the MSPP, leading to finding the path with a minimum-cost norm on a network, called the (minNorm) problem. Two algorithms for solving the (min-

TABLE 7. Algorithm performance with d ($k = 6$).

Random networks											
		2		4		6		8		10	
Cost distribution	d	Sec	Mb	Sec	Mb	Sec	Mb	Sec	Mb	Sec	Mb
Uniform	Algm 1	0.14	53.84	0.34	87.64	0.46	120.31	0.25	152.41	0.41	184.36
($n = 30{,}000$)	Algm 2	0.02	35.49	0.03	54.88	0.09	74.84	0.12	94.91	0.16	115.01
Circular	Algm 1	0.16	60.10	0.32	92.95	0.47	125.62	0.41	157.11	0.42	188.49
($n = 30{,}000$)	Algm 2	0.09	34.44	0.13	55.91	0.66	89.58	0.70	112.62	0.80	132.29
Cubic	Algm 1	1.11	66.59	8.21	300.70	8.17	319.25	—	—	—	—
(n = **300**)	Algm 2	—	—	—	—	—	—	—	—	—	—

TABLE 8. Algorithm performance with k ($d = 4$).

Random networks

		3		4		6		8		10	
Cost distribution	k	Sec	Mb	Sec	Mb	Sec	Mb	Sec	Mb	Sec	Mb
Uniform	Algm 1	0.11	59.91	0.10	69.07	0.21	87.64	0.23	106.36	0.40	125.14
(n = 30,000)	Algm 2	0.07	40.71	0.03	45.43	0.05	54.88	0.03	64.33	0.19	73.78
Circular	Algm 1	0.09	61.29	0.11	71.81	0.24	92.95	0.38	113.99	0.29	134.76
(n = 30,000)	Algm 2	0.11	40.35	0.08	44.93	0.09	55.51	0.16	64.04	1.43	93.33
Cubic	Algm 1	0.03	1.31	7.88	4.83	85.16	31.73	—	—	—	—
(n = **300**)	Algm 2	—	—	—	—	—	—	—	—	—	—

Square grid networks

		3		4		6		8		10	
Cost distribution	k	Sec	Mb	Sec	Mb	Sec	Mb	Sec	Mb	Sec	Mb
Uniform	Algm 1	0.01	2.08	0.04	3.38	0.07	8.03	0.24	17.28	0.64	35.01
(n = 500)	Algm 2	0.01	0.86	0.01	0.98	0.01	1.24	0.01	1.62	0.02	1.87
Circular	Algm 1	.001	0.09	.001	0.11	.001	0.17	.001	0.17	.001	0.20
(n = 500)	Algm 2	0.00	0.16	0.00	0.21	0.01	0.47	0.03	0.85	0.07	1.52
Cubic	Algm 1	0.00	8.86	0.00	10.36	0.07	13.36	0.97	16.36	10.82	19.36
(n = **50**)	Algm 2	0.06	0.95	3.70	5.26	—	—	—	—	—	—

Norm) problem were presented and computationally tested. One of the algorithms is based on the labeling algorithm for the MSPP with an initial estimate for the optimal value produced by the solution of the (minSum) problem, which consists of determining the path with a minimum total sum of the criteria values. This algorithm integrates an optimality condition that allows removing a very large number of labels generated in the procedure (in general, about 95%). The second algorithm described in this paper produces a ranking of the paths until a stopping condition occurs. The new algorithms were computationally tested for two types of networks (random and grid) and three different distributions for the arc costs (uniform, circular, and cubic). The two algorithms proved to be very efficient for networks with a uniform or a circular distribution for the costs. The new labeling algorithm is able to find, in less than 1 second of CPU time, the optimal solution for most of the "cubic cost" instances.

Acknowledgments

The authors would like to acknowledge the referees for their relevant comments. In addition, the authors are grateful to Prof. Douglas R. Shier for his help in improving the readability and the presentation of the manuscript.

REFERENCES

[1] R.E. Bellman, On a routing problem, Q Appl Math 16 (1958), 87–90.

[2] R.E. Bellman and S.E. Dreyfus, Applied dynamic programming, Princeton University Press, Princeton, NJ, 1962.

[3] J. Brumbaugh-Smith and D. Shier, An empirical investigation of some bicriterion shortest path algorithms, Eur J Oper Res 43 (1989), 216–224.

[4] J.N. Clímaco and E.Q. Martins, A bicriterion shortest path algorithm, Eur J Oper Res 11 (1982), 399–404.

[5] J.M. Coutinho-Rodrigues, J.N. Clímaco, and J.R. Current, An interactive biobjective shortest-path approach: Searching for unsupported nondominated solutions, Comput Oper Res 26 (1999), 789–798.

[6] J. Current and M. Marsh, Multiple transportation network design and routing problems: taxonomy and annotation, Eur J Oper Res 65 (1993), 4–19.

[7] J.R. Current, C.S. ReVelle, and J.L. Cohon, The median shortest path problem: A multiobjective approach to analyze cost vs. accessibility in the design of transportation networks, Transport Sci 21 (1987), 188–197.

[8] J.R. Current, C.S. ReVelle, and J.L. Cohon, An interactive approach to identify the best compromise solution for the two objective shortest path problems, Comput Oper Res 17 (1990), 187–198.

[9] N. Deo and C. Pang, Shortest path algorithms: Taxonomy and annotation, Networks 14 (1984), 275–323.

[10] B. Golden and T. Magnanti, Deterministic network optimization: A bibliography, Networks 7 (1977), 149–183.

[11] F. Guerriero and R. Musmanno, Label correcting methods to solve multicriteria shortest path problems, J Optimiz Theory App 111 (2001), 589–613.

[12] P. Hansen, "Bicriterion path problems," Multiple criteria decision making: Theory and application, G. Fandel and T.

Gal (Editors), Lectures Notes in Economics and Mathematical Systems 177, Springer, Heidelberg, 1980, pp. 109–127.

[13] M.I. Henig, The shortest path problem with two objective functions, Eur J Oper Res 25 (1986), 281–291.

[14] E.Q. Martins, On a multicriteria shortest path problem, Eur J Oper Res 16 (1984), 236–245.

[15] E.Q. Martins, M.M. Pascoal, D.M. Rasteiro, and J.L. Santos, The optimal path problem, Invest Oper 19 (1999), 43–60.

[16] E.Q. Martins, M.M. Pascoal, and J.L. Santos, Deviation algorithms for ranking shortest paths, Int J Found Comput Sci 10 (1999), 247–261.

[17] E.Q. Martins and J.L. Santos, An algorithm for the quickest path problem, Oper Res Lett 20 (1997), 195–198.

[18] E.F. Moore, The shortest path through a maze, Proc Int Symp Theory of Switching, Harvard University Press, Cambridge, MA, 1959, pp. 285–292.

[19] A.D. Vainshtein, The vector shortest path problem in the ℓ_∞-norm, Am Math Soc Translat 158 (1994), 207–215.

[20] P. Vincke, Problèmes multicritères, Cah Centre Etudes Rech Oper 16 (1974), 425–439.

[21] A. Warburton, Approximation of Pareto optima in multiple-objective shortest path problems, Oper Res 35 (1987), 70–79.