CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Minimum Cost Path Problems with Relays

Gilbert Laporte
Marta M.B. Pascoal

May 2009

CIRRELT-2009-21

Bureaux de Montréal:
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone: 514 343-7575
Télécopie: 514 343-7121

Bureaux de Québec:
Université Laval
Pavillon Palasis-Prince, local 2642
Québec (Québec)
Canada G1K 7P4
Téléphone: 418 656-2073
Télécopie: 418 656-2624

www.cirrelt.ca

Université Laval · UQÀM · HEC Montréal · École Polytechnique Montréal · Université de Montréal

Minimum Cost Path Problems with Relays

Gilbert Laporte[1,*], Marta M.B. Pascoal[2]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Canada Research Chair in Distribution Management, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[2] Departamento de Matematica da Universidade de Coimbra, Apartado 3008, EC Universidade, 3001-454 Coimbra, Portugal and Institute for Systems and Computers Engineering, Coimbra (INESCC), rua Antero de Quental, 199, 3000 – 033 Coimbra, Portugal

Abstract. The *Minimum Cost Path Problem with Relays* (MCPPR) consists of finding a minimum cost path from a source to a destination, along which relay nodes are located at a certain cost, subject to a weight constraint. This paper first models the MCPPR as a particular bicriteria path problem involving an aggregated function of the path and relay costs, as well as a weight function. A variant of this problem that takes into account all three functions separately is then considered. Formulating the MCPPR as a part of a bicriteria path problem allows the development of labeling algorithms in which the bound on the weight of paths controls the number of node labels. The algorithm for this constrained single objective function version of the problem has a time complexity of $O(Wm + Wn \log \max \{W, n\})$, where n is the number of nodes, m the number of arcs and W the weight upper bound. Computational results on random instances with up to 10 000 nodes and 100 000 arcs are reported.

Keywords. Relays, shortest path problem, bicriteria optimization, labeling algorithms

Acknowledgements. This research was partially funded by the Natural Sciences and Engineering Research Council of Canada under grant 39682-05. Part of this work was developed during a visit of Marta Pascoal to the CIRRELT, Montréal, Canada. Her work was partially funded by the FCT Portuguese Foundation of Science and Technology (Fundação para a Ciência e a Tecnologia) under grant SFRH/BSAB/830/2008.

---

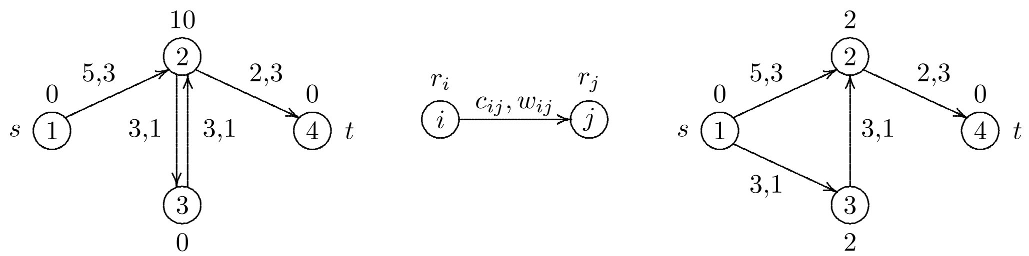
* Corresponding author: Gilbert.Laporte@cirrelt.ca

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2009

1 Introduction

The purpose of this paper is to describe exact algorithms for the single criterion and the bicriteria *Minimum Cost Path Problem with Relays* (MCPPR). This problem is defined on a directed network $(\mathcal{N}, \mathcal{A})$ with a set $\mathcal{N} = \{1, \dots, n\}$ of nodes and a set $\mathcal{A} = \{1, \dots, m\}$ of arcs. Two nodes are distinguished in $\mathcal{N}$: an origin s and a destination t. A path from s to t in $(\mathcal{N}, \mathcal{A})$ is represented by a sequence $p = \langle v_1, v_2, \dots, v_\ell \rangle$, where $v_1 = s$, $v_\ell = t$, $v_i \in \mathcal{N}$, $i = 1, \dots, \ell$, and $(v_i, v_{i+1}) \in \mathcal{A}$, $i = 1, \dots, \ell - 1$. For simplicity we write $i \in p$ if i is a node in the sequence p, and $(i, j) \in p$ if i immediately precedes j in p. We denote by $\mathcal{P}$ the set of paths from s to t in $(\mathcal{N}, \mathcal{A})$. Given $i, j \in p$, the subpath of p between nodes i and j is denoted by $\sigma_p(i, j)$. If a path ends at a node where another one starts, then their concatenation is denoted by $\diamond$.

With each arc $(i, j) \in \mathcal{A}$ are associated a cost $c_{ij} \in \mathbb{R}_0^+$ and a weight $w_{ij} \in \mathbb{R}_0^+$. Also, using a network node $i \in \mathcal{N}$ as a relay generates a fixed cost $r_i \in \mathbb{R}_0^+$, and it is assumed that s and t are relays of cost 0. The MCPPR aims to determine a least cost path and the locations of relays under a weight constraint. We will write $\widehat{i}$ to distinguish a relay node $i \in \mathcal{N} - \{s, t\}$ of a path from the remaining nodes that are not relays. As an example, in the network $(\mathcal{N}_1, \mathcal{A}_1)$ of Figure 1, $q = \langle 1, 2, 4 \rangle$ and $q' = \langle 1, \widehat{2}, 4 \rangle$ are paths from 1 to 4: path q contains no relay besides s and t, whereas q' has a relay at node 2.



(a) Network $(\mathcal{N}_1, \mathcal{A}_1)$ (b) Network $(\mathcal{N}_2, \mathcal{A}_2)$

Figure 1: Two instances of the MCPPR

The objective function $f(p)$ is the sum of the path cost

$$c(p) = \sum_{(i,j) \in p} c_{ij},$$

and of the cost of the selected relays,

$$r(p) = \sum_{\widehat{i} \in p} r_i.$$

In addition, the weight of any subpath between two consecutive relays cannot exceed a given upper bound W. The weight of a path q from x to y, whose only relay nodes are x and y is given by

$$w(q) = \sum_{(i,j) \in q} w_{ij}.$$

Letting $R(p) = \{\widehat{s}_1 = s, \dots, \widehat{s}_k = t\}$ be the set of relay nodes of a path $p \in \mathcal{P}$, the function w can be extended as $\tilde{w}(p) = w(\sigma_p(\widehat{s}_{k-1}, t))$. In the paths above we have $R(q) = \emptyset$, $R(q') = \{2\}$, and $f(q) = 7$, $\tilde{w}(q) = 6$, $f(q') = 17$, $\tilde{w}(q') = \langle \widehat{2}, 4 \rangle = 3$.

The function f aggregates c and r. For instance, still in $(\mathcal{N}_1, \mathcal{A}_1)$ and considering $W = 5$, $q'' = \langle 1, 2, \widehat{3}, 2, 4\rangle$ is the optimal solution of the MCPPR, however q' is also feasible, and better than q'' in terms of cost ($c(q'') = 13 > c(q') = 7$) but worse in terms of relay cost ($r(q'') = 0 < r(q') = 10$). Motivated by situations like this, we will consider a variant of the MCPPR which consists of minimizing c and r, while looking for feasible paths in terms of weight. This will be called the *Bicriteria Minimum Cost Path Problem with Relays* (BMCPPR).

The MCPPR arises in the context of a telecommunications network design problem. It has been introduced and studied in [4, 5], and applied to several contexts in [2, 7, 6]. In [5] three methods are proposed to find a minimum cost path with relays. The most efficient has a complexity of $\mathcal{O}(Wnm \log W)$. In this paper we develop a more efficient algorithm. We first address the original version of the MCPPR and then study the variant that considers arc and relay costs separately.

The remainder of the paper is organized as follows. Section 2 is devoted to the MCPPR. After the problem is defined, two labeling algorithms to find the optimal path are described and the theoretical results that support them are presented. In Section 3 the MCPPR objective function is decomposed into two: the BMCPPR is defined and a method for dealing with this problem is developed. Section 4 presents results of computational experiments on random instances for the problems studied in Sections 2 and 3. Conclusions follow in Section 5.

2 The minimum cost path problem with relays

The purpose of the MCPPR is to determine a feasible path between s and t, i.e., a path that satisfies the weight constraint and has a minimum objective function value. In other words, denoting by

$$\overline{\mathcal{P}} = \{p \in \mathcal{P} : \ w(\sigma_p(\widehat{s}_{i-1}, \widehat{s}_i)) \leq W \ (i = 2, \ldots, k \text{ and } R(p) = \{\widehat{s}_1, \ldots, \widehat{s}_k\})\}$$

the set of feasible paths in $\mathcal{P}$, the MCPPR aims to find a path from s to t in $(\mathcal{N}, \mathcal{A})$ satisfying

$$\min_{p \in \overline{\mathcal{P}}} \{f(p)\}. \tag{1}$$

This problem can be seen as a more general case of the weight constrained shortest path problem. However, in the MCPPR the relay location affects both the weight constraint and the relay costs involved in the objective function itself. One of the differences with respect to the shortest path problem is that an optimal path can have loops if it contains a relay node. For instance, when $W = 5$ the minimum cost path with relays in the network in Figure 1a is $q'' = \langle 1, 2, \widehat{3}, 2, 4\rangle$. Another difference is the fact that optimal paths may contain subpaths that are not optimal, which means that labeling algorithms supported by Bellman's principle of optimality [1] cannot be applied directly to the MCPPR. The only optimal solution for the MCPPR in network $(\mathcal{N}_2, \mathcal{A}_2)$ of Figure 1b is path $\langle 1, 3, 2, 4\rangle$. However $\langle 1, 2\rangle$ is feasible and has objective value 5, but it is not optimal from 1 to 3 because $\langle 1, 3, 2\rangle$ is also feasible but has an objective value 6.

A variant of (1) is obtained if the weight is considered as an objective function

$$\min_{p \in \overline{\mathcal{P}}} \{f(p)\}, \quad \min_{p \in \overline{\mathcal{P}}} \{\tilde{w}(p)\}. \tag{2}$$

If f and $\tilde{w}$ are not correlated then (2) may have no optimal solution. Instead, the set of feasible non-dominated paths can be defined as the set of feasible solutions for which there is no other solution that improves one of the objectives without worsening the other. Given two feasible paths p_1, p_2 between the same pair of nodes, p_1 dominates p_2 ($p_{1D}p_2$) if and only if $f(p_1) \leq f(p_2)$, $\tilde{w}(p_1) \leq \tilde{w}(p_2)$, and at least one of the inequalities is strict. Then it is also said that $(f(p_1), \tilde{w}(p_1))$ dominates $(f(p_2), \tilde{w}(p_2))$, which is denoted by $(f(p_1), \tilde{w}(p_1))_D(f(p_2), \tilde{w}(p_2))$. A feasible path p is non-dominated if and only if it is not dominated by any other.

Next we show, first that there is a non-dominated solution of (2) which is also an optimal solution of MCPPR, and second that this bicriteria problem can be solved by labeling procedures. In fact, even though the function $\tilde{w}$ is not additive, a variant of the principle of optimality can be proved if the procedure is restricted to feasible paths, and thus a labeling algorithm can be used.

Theorem 1. *There exists an optimal solution of (1) that is a non-dominated solution of (2).*

Proof. Let p^* be a minimum cost path with relays from s to t, so that p^* satisfies the weight constraint and $f(p)$ is the minimum value for paths in $\overline{\mathcal{P}}$. By contradiction, assume p^* is dominated by another path $p \in \overline{\mathcal{P}}$, i.e.,

1. $f(p^*) \geq f(p)$ and $\tilde{w}(p^*) > \tilde{w}(p)$, or

2. $f(p^*) > f(p)$ and $\tilde{w}(p^*) \geq \tilde{w}(p)$.

As p is feasible and p^* is a minimum cost feasible path in terms of f, 2. yields a contradiction. On the other hand 1. implies that $f(p^*) = f(p)$. Therefore p is also an optimal solution. If it is non-dominated the proof is concluded; otherwise the argument can be repeated, so there is a non-dominated path that is the optimal solution. $\square$

According to Theorem 1, finding all solutions to (2) provides a minimum cost path with relays as well. Besides, this theorem implies that a path is obtained for every non-dominated pair of objective values, then one of them is an optimal solution to (1). There are two non-dominated solutions of problem (2) in the network of Figure 2 with $W = 5$, $p_1 = \langle 1, 2, \widehat{4}, 5\rangle$ and $p_2 = \langle 1, 3, 2, \widehat{4}, 5\rangle$, and with objective values $(f(p_1), \tilde{w}(p_1)) = (2, 3)$ and $(f(p_2), \tilde{w}(p_2)) = (2, 3)$. However, $\langle 1, 3, 2\rangle_D\langle 1, 2\rangle$. This means that finding all non-dominated solutions to (2) would require storing dominated labels. Nevertheless, despite the constraint and the fact that f is not a standard function, the MCPPR can be solved by finding only the non-dominated pairs of objective values, which can be achieved by using labeling procedures, as stated in Theorem 2.

Theorem 2. *For any non-dominated pairs of objective values of (2) there is a path formed by non-dominated subpaths only.*

Proof. Let p^* be a non-dominated path in $\overline{\mathcal{P}}$ and let i be one of its nodes. Assume that $\sigma_{p^*}(s, i)$ is dominated by another feasible path from s to i, q, thus

1. $f(\sigma_{p^*}(s, i)) \geq f(q)$ and $\tilde{w}(\sigma_{p^*}(s, i)) > w(q)$, or

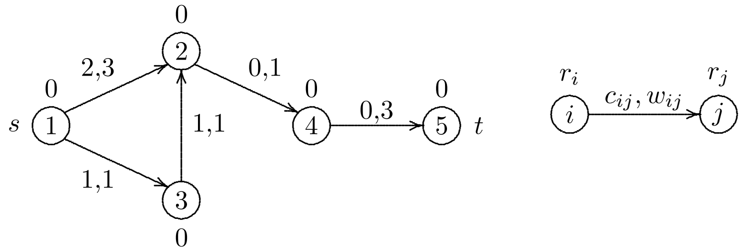


Figure 2: Network $(\mathcal{N}, \mathcal{A})$

2. $f(\sigma_{p^*}(s,i)) > f(q)$ and $\tilde{w}(\sigma_{p^*}(s,i)) \geq \tilde{w}(q)$.

Let $q^* = q \diamond \sigma_{p^*}(i,t)$ be the path that results from replacing $\sigma_{p^*}(s,i)$ by q in p^*. Then $\tilde{w}(q^*) \geq \tilde{w}(p^*)$ and, in case 1.

$$f(q^*) = f(q) + f(\sigma_{p^*}(i,t)) \leq f(p^*)$$

holds, while in case 2. the inequality is strict.

If condition 2. is satisfied, p^* is dominated by q^*. As for condition 1. either the same conclusion is valid, or else p^* and $q \diamond \sigma_{p^*}(i,t)$ are equivalent, which means there is a non-dominated path in $\overline{\mathcal{P}}$ equivalent to p^* and formed only by non-dominated subpaths. □

By this result a set of paths with all non-dominated pairs of objective values of (2) can be found using a labeling algorithm that associates a network node with several nodes in different paths starting in s. Each of these nodes, for instance x, is associated with a label with the form $l_x = [\pi_x^f, \pi_x^w, \xi_x, \beta_x]$, where

- π_x^f denotes the path f value (that depends on the arc costs and node relays),
- π_x^w denotes its weight,
- ξ_x is the node that precedes x in the path, and
- β_x is the network node that corresponds to x.

Given $\beta_x = i \in \mathcal{N}$ and $(i,j) \in \mathcal{A}$, if $j \neq t$ then two new nodes, y, z, are labeled with

$$l_y = [\pi_x^f + c_{ij} + r_j, 0, x, j] \text{ and } l_z = [\pi_x^f + c_{ij}, \pi_x^w + w_{ij}, x, j], \tag{3}$$

where the first case corresponds to locating a relay at node j. When $j = t$ only the second case has to be considered. The insertion of a new label should be accompanied by a dominance test that compares it with labels previously set for the same node. If the new label is dominated by one of the others, it is discarded; otherwise it is stored as a potential non-dominated label, and any label that is dominated by this one can be deleted. Similar labeling algorithms for bicriteria shortest path problems (thus different objective functions) can be found in [3, 8, 10, 11], among others.

When solving problem (1) some modifications can be introduced to a general labeling algorithm because the goal now is to determine a single optimal solution, which has to be feasible and to have minimum objective value f. As W is one of the problem's input, then there is at most one label of weight $0, \ldots, W$ associated with each node, where $w_i = 0$ means that i is used as a relay node. Also a label l_z is only created if it corresponds to a feasible path, that is, if $\pi_x^w + w_{ij} \leq W$. This strategy may require non-dominated labels to be stored. For instance, the paths $\langle 1, 2 \rangle$ and $\langle 1, \widehat{2} \rangle$ in network $(\mathcal{N}_1, \mathcal{A}_1)$ shown in Figure 1a correspond to labels $l_a = [5, 3, 1, 2]$ and $l_b = [5, 0, 1, 2]$, respectively. Both have different weights but $(\pi_b^f, \pi_b^w)_D(\pi_a^f, \pi_a^w)$. Storing such labels simplifies the dominance test as shown hereafter. Let M be a $(W+1) \times n$ matrix, where M_{di} stores the index of the label with the best objective value for each feasible weight $d = 0, \ldots, W+1$ and node $i \in \mathcal{N}$. Given a node that produces two new labels as in (3), l_y is inserted in the set of labels if and only if $\pi_x^r + c_{ij} + r_j > \pi_{M_{0j}}^r$, and l_z is inserted if and only if $\pi_x^r + c_{ij} > \pi_{M_{dj}}^r$, with $d = \pi_x^w + w_{ij}$. Each label replaces the previous labels at M_{0j} and M_{dj}, if they exist. This test can be easily implemented and, as non-dominated labels can be stored in different matrix positions they are no longer deleted. This improves the theoretical complexity order of the algorithm by comparison to a bicriteria shortest path labeling algorithm.

The pseudo-code of Algorithm 1 summarizes the procedure described for the MCPPR. The working variable X represents a set with the labels that have not been scanned.

Algorithm 1. ***Determination of the minimum cost path with relays***

For $d \in \{0, \ldots, W\}$ Do
  For $j \in \mathcal{N}$ Do $M_{dj} \leftarrow 0$
$nX \leftarrow 1$; $l_{nX} \leftarrow [0, 0, -, s]$; $X \leftarrow \{1\}$
While $X \neq \emptyset$ Do
  $x \leftarrow$ node in X; $X \longleftarrow X - \{x\}$; $i \leftarrow \beta_x$
  For $j \in \mathcal{N}$ such that $(i, j) \in \mathcal{A}$ Do
    If $(i \neq t)$ Then
      If $(M_{0j} = 0)$ or $(M_{0j} \neq 0$ and $\pi_x^f + c_{ij} + r_j < \pi_{M_{0j}}^f)$ Then
        $nX \leftarrow nX + 1$; $l_{nX} \leftarrow [\pi_x^f + c_{ij} + r_j, 0, x, j]$; $M_{0j} \leftarrow nX$
        $X \leftarrow X \cup \{nX\}$
      EndIf
    $d \leftarrow \pi_x^w + w_{ij}$
    If $d \leq W$ Then
      If $(M_{dj} = 0)$ or $(M_{dj} \neq 0$ and $\pi_x^f + c_{ij} > \pi_{M_{dj}}^f)$ Then
        $nX \leftarrow nX + 1$; $l_{nX} \leftarrow [\pi_x^f + c_{ij}, d, x, j]$; $M_{dj} \leftarrow nX$
        $X \leftarrow X \cup \{nX\}$
      EndIf
  EndFor
EndWhile

Theorem 3 proves that if the labels to be scanned are selected by non-decreasing order of f then Algorithm 1 is label setting and thus the nodes chosen in X have the correct label. As a consequence the algorithm can halt as soon as a node that corresponds to t is chosen; otherwise, if it continues until $X = \emptyset$ then the labels stored in $M_{\bullet t}$ correspond to the optimal paths from s to t

for every feasible weight.

Theorem 3. *If the labeled nodes are selected by non-decreasing order of f, then those that have been selected have permanent labels.*

Proof. Suppose node x is chosen in X at a step of Algorithm 1 and assume its label l_x is not permanent, that is, another label for the same network node, l_y, feasible and such that $\pi_y^f < \pi_x^f$ will be chosen later on. There are two possibilities:

1. If $y \in X$ when x is selected, then at that moment $\pi_y^f < \pi_x^f$.
2. If y is inserted in X after x has been selected, then its label is obtained from a sequence of feasible labels $l_{y_1}, \ldots, l_{y_k}$ starting at some node y_1, which belongs to X at the time x is selected. Furthermore, as $c_{ij} \geq 0$, $(i, j) \in \mathcal{A}$, and $r_i \geq 0$, $i \in \mathcal{N}$, we have $\pi_{y_1}^f \leq \ldots \leq \pi_{y_k}^f \leq \pi_y^f$, therefore $\pi_{y_1}^f < \pi_x^f$.

Both cases lead to contradiction, as x should not have been chosen because x was not the node associated with the best label in X. □

As there are at most Wn labels, the total number of scanned arcs is $\mathcal{O}(Wm)$ and, if the label set is manipulated as a Fibonacci heap, each scan takes $\mathcal{O}(1)$ operations, namely comparisons and insertions. Besides, all the Wn labels have to be selected in the label set, which can be achieved in $\mathcal{O}(Wn \log(Wn))$ time. Therefore the worst-case time complexity bound of Algorithm 1 is $\mathcal{O}(Wm + Wn \log \max\{W, n\})$.

3 The bicriteria minimum cost path problem with relays

The objective function of the MCPPR aggregates two other: the path cost and the relay cost. Considering them separately can provide effective alternative solutions. The BMCPPR is thus a constrained bicriteria problem defined as

$$\min_{p \in \overline{\mathcal{P}}} \{c(p)\}, \quad \min_{p \in \overline{\mathcal{P}}} \{r(p)\}. \tag{4}$$

Its aim is to find non-dominated paths in $\overline{\mathcal{P}}$, that is, paths $p \in \overline{\mathcal{P}}$ such that there is no $q \in \overline{\mathcal{P}}$ with $c(p) \geq c(q)$, $r(p) \geq r(q)$ and at least one inequality is strict. Similarly to problem (1), the solutions of (4) can be found amongst the non-dominated solutions of the tricriteria problem

$$\min_{p \in \overline{\mathcal{P}}} \{c(p)\}, \quad \min_{p \in \overline{\mathcal{P}}} \{r(p)\}, \quad \min_{p \in \overline{\mathcal{P}}} \{\tilde{w}(p)\}. \tag{5}$$

Still, function $\tilde{w}$ is not additive and the principle of optimality is again invalid. As a result, a labeling method would have to store intermediate dominated labels. Alternatively Algorithm 1 can be extended by including in the node labels a new parameter that refers to the path cost. Given a node x associated with a path starting in s, the new labels have the form $l_x = [\pi_x^c, \pi_x^r, \pi_x^w, \xi_x, \beta_x]$, where

- π_x^c, denotes the path cost,

- π^r_x, denotes the cost of the relays in the path.

Taking x as the starting point, new labels for nodes y and z are updated as

$$l_y = [\pi^c_x + c_{ij}, \pi^r_x + r_j, 0, x, j] \text{ and } l_z = [\pi^c_x + c_{ij}, \pi^r_x, \pi^w_x + w_{ij}, x, j], \tag{6}$$

where the first case corresponds to using j as a relay node, if $j \neq t$. Similarly to the matrix M in Section 2, a $(W+1) \times n$ matrix L is defined. For every feasible weight $d = 0, \ldots, W+1$ and node i, L_{di} contains the paths from s to i with weight d, which are not dominated by any other of the same weight. Each position in L is usually associated with more than one path, unlike matrix M.

Given a node x that corresponds to $i \in \mathcal{N}$ and $(i, j) \in \mathcal{A}$, the labels in (6) are produced. Then, l_y is inserted in set L_{0j} if and only if it contains no other label that dominates it (i.e., another label l_a such that $\pi^c_a \leq \pi^c_y$, $\pi^r_a \leq \pi^r_y$ and at least one of the inequalities is strict), while l_z is inserted in $L_{\pi^w_x + w_{ij} j}$ under analogous conditions. The labels in L_{0j} and in L_{dj} that are dominated by l_y and by l_z, respectively, can be removed from these sets; otherwise $L_{\bullet j}$ will contain other labels besides the necessary ones. Finally, it is worth stressing that this approach in effect computes the set of non-dominated paths for every feasible weight $d = 0, \ldots, W$. At the end of the process all the BMCPPR solutions, and possibly some dominated ones which should be filtered, are in $L_{\bullet t}$.

The solutions obtained with this labeling method applied to the BMCPPR with $W = 5$ in $(\mathcal{N}_1, \mathcal{A}_1)$ are represented as a tree in Figure 3. After all labels have been scanned the set of candidate solutions is given by $L_{\bullet 4} = \{\langle 1, 2, 3, \widehat{2}, 4\rangle, \langle 1, 2, \widehat{3}, 2, 4\rangle, \langle 1, \widehat{2}, 4\rangle, \langle 1, \widehat{2}, 3, 2, 4\rangle\}$. In this set $\langle 1, 2, 3, \widehat{2}, 4\rangle$, $\langle 1, 2, \widehat{3}, 2, 4\rangle$ and $\langle 1, \widehat{2}, 4\rangle$ are solutions of BMCPPR, while $\langle 1, \widehat{2}, 3, 2, 4\rangle$ is dominated by the first two paths and thus should be discarded in the end.

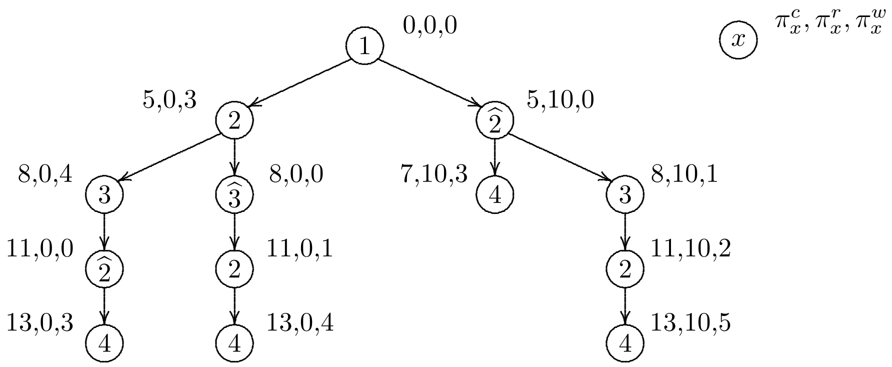


Figure 3: Tree of candidate solutions for the BMCPPR with $W = 5$ in $(\mathcal{N}_1, \mathcal{A}_1)$

The pseudo-code presented in Algorithm 2 outlines the method to solve the BMCPPR.

Algorithm 2. ***Determination of the non-dominated minimum cost paths with relays***

  `For` $d \in \{0, \ldots, W\}$ `Do`

    `For` $j \in \mathcal{N}$ `Do` $L_{dj} \leftarrow \emptyset$

$nX \leftarrow 1$; $l_{nX} \leftarrow [0, 0, 0, -, s]$; $X \leftarrow \{1\}$

`While` $X \neq \emptyset$ `Do`

    $x \leftarrow$ node in X; $X \longleftarrow X - \{x\}$; $i \leftarrow \beta_x$

    `For` $j \in \mathcal{N}$ such that $(i, j) \in \mathcal{A}$ `Do`

```
        If (i ≠ t) Then
            If (L_0j = ∅) or (L_0j ≠ ∅ and (π_x^c + c_ij, π_x^r + r_j) is not dominated in L_0j) Then
                nX ← nX + 1; l_nX ← [π_x^c + c_ij, π_x^r + r_j, 0, x, j]; L_0j ← L_0j ∪ {nX}
                X ← X ∪ {nX}
            EndIf
        d ← π_x^w + w_ij
        If d ≤ W Then
            If (L_dj = ∅) or (L_dj ≠ ∅ and (π_x^c + c_ij, π_x^r) is not dominated in L_dj) Then
                nX ← nX + 1; l_nX ← [π_x^c + c_ij, π_x^r, d, x, j]; L_dj ← L_dj ∪ {nX}
                X ← X ∪ {nX}
            EndIf
    EndFor
EndWhile
Select the non-dominated labels associated with t in L_dt, d = 0, ..., W
```

Algorithm 2 can be implemented either as a label correcting method, if the selection of nodes in X is arbitrary, or as a label setting method, if the node with the least lexicographic label in terms of c and r is chosen (attending to the fact that $r_i, c_{ij} \geq 0$, for $i \in \mathcal{N}$, $(i,j) \in \mathcal{A}$). In the second case the labels chosen in X are permanent, that is they are non-dominated in the correspondent set L_{di}, for some $d \in \{0, \dots, W\}$ and $i \in \mathcal{N}$, and this includes the BMCPPR solutions. The proof is analogous to that of Theorem 3 and is therefore omitted.

As mentioned earlier this approach solves a problem similar to the bicriteria shortest path problem for every feasible weight, which Hansen proved to have a number of non-dominated solutions that may grow exponentially [9], and thus the BMCPPR is NP-hard.

4 Computational results

Computational experiments were carried out to evaluate and to compare the empirical performance of the algorithms just described. The tests were run on a Pentium 4 at 3 GHz, with 512 Kb of cache memory and 1Gb of RAM, over SUSE Linux 10.3. The results presented in the following were obtained on ten different instances generated for each dimension of the data sets.

4.1 The minimum cost path problem with relays

Two versions of Algorithm 1 were coded in C to determine a minimum cost path with relays: a label correcting algorithm where the labels set is managed as a FIFO list, and a label setting algorithm which is interrupted when a label associated with t is selected.

The instances considered were random networks with $n = 500, 1\,000, 3\,000, 5\,000, 10\,000$, and dn arcs, for densities $d = 4, 5, 10$. Uniformly integer cost, relay and weight values generated in $[1, 100]$ and $W = 50, 110$ were considered. The plots in Figure 4 show the average running times of the two codes for ten different instances of each data set dimension.

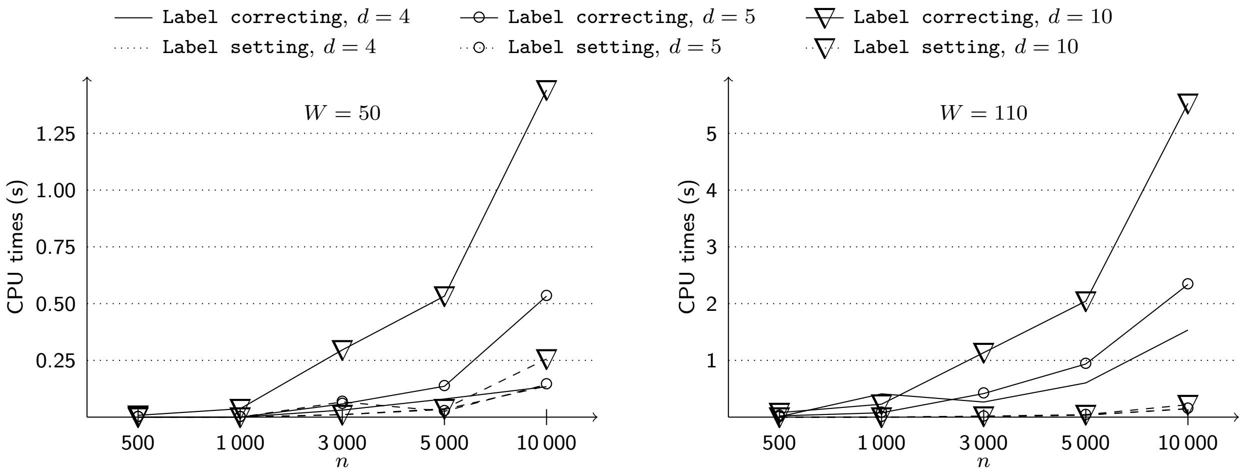


Figure 4: Average running times of Algorithm 1 versus n

For a fixed value of W the CPU times grow with instance size, namely with n and with d. The impact of d seems larger than that of n, as times increase faster with the first parameter than with the number of nodes for $W = 50$ and $W = 110$. The relative behavior of the algorithms was similar for both values of W. However, greater values imply a larger matrix and a higher number of labels will be necessary. Therefore the times are clearly greater when $W = 110$. The label setting method clearly outperformed label correcting for all problem dimensions. It should also be noted that its sensibility to the increase of the instance parameters was much smaller than for the label correcting version. This program solved the larger problems, $n = 10\,000$, $d = 10$ and $W = 110$, in an average time of 0.219 seconds.

4.2 The bicriteria minimum cost path problem with relays

As for the bicriteria minimum cost path with relays, two implementations of Algorithm 2 were coded in C and tested on a subset of the previous random networks with $n = 500, 1\,000, 3\,000, 5\,000$. The two variants correspond to a label correcting version where the set X is managed as a FIFO list, and a label setting version. No prunning of the labels within each set L_{di}, $d = 0, \ldots, W$, $i \in \mathcal{N}$, dominated by new ones was implemented. The results in the following are averages for ten instances of each dimension of the data set.

As expected in this case the number of paths produced, and thus the total number of labels generated was greater than for the MCPPR, which was also reflected in greater running times for both variants of Algorithm 2. The average numbers of solutions for each problem and the average number of labels generated by the programs are presented in Table 1 (“—” means the program did not run until the end). The label correcting method was less economic than the label setting in terms of the memory space it used. In most of the $n = 5\,000$ and $d = 10$ instances these problems did not run until the end due to memory overflow. In the remaining cases the number of generated labels was always greater with the first method. As a consequence the running times for the BMCPPR problem were greater than for the MCPPR.

The same remark can be made concerning the times growth with the number of nodes, which is

		# solutions		# labels			
				Label correcting		Label setting	
n	d	$W = 50$	$W = 110$	$W = 50$	$W = 110$	$W = 50$	$W = 110$
500	4	5	4	78120	574150	20608	106027
500	5	7	6	139090	994255	36306	176186
500	10	9	7	511630	2115965	119925	328788
1 000	4	7	7	206064	1616795	49113	265170
1 000	5	7	8	313224	2299274	73828	384668
1 000	10	8	5	1172799	4547079	242858	661032
3 000	4	10	8	880406	6693012	178322	854451
3 000	5	12	9	1319813	9698591	287023	1321826
3 000	10	10	8	4622109	19770964	1004046	2853506
5 000	4	9	8	1780350	13091331	334742	1739159
5 000	5	10	11	2595751	19226014	528745	2426392
5 000	10	15	8	8832978	—	1777294	4758894

Table 1: Average number of non-dominated paths and of labels generated by Algorithm 2

related to the number of paths that have to be computed and with the total number of generated labels. Figure 5 depicts the running times obtained by the variants on the test bed. Even though in general the label setting algorithms outperformed label correcting algorithms this was not always the case and, for the smaller instances with $W = 50$ the label correcting method was faster. This is due to the more demanding process of inserting a new label by lexicographic order in set X. However, as noted above, the number of labels was much higher for the label correcting versions, and thus its running times still grew faster as a function of instance size.

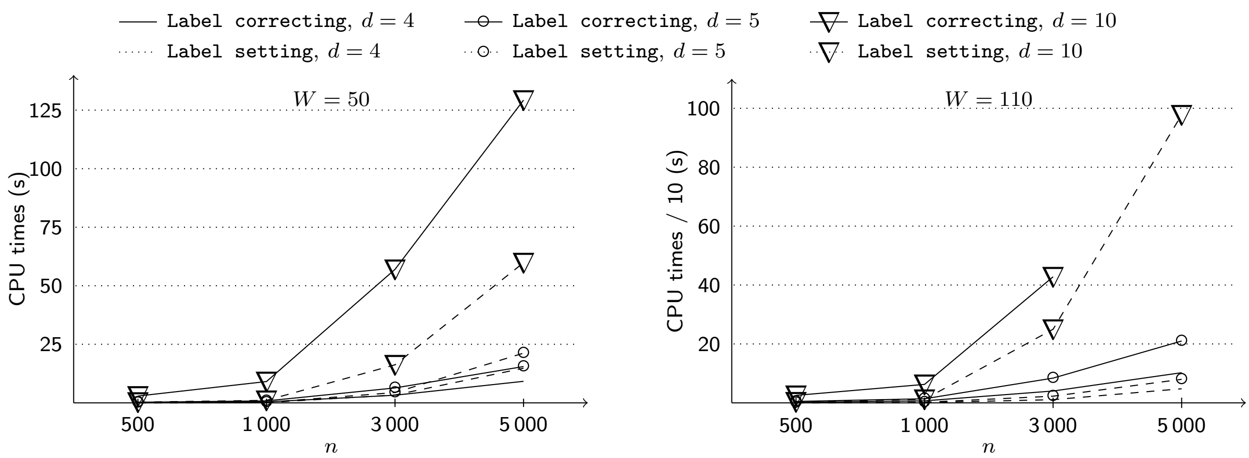


Figure 5: Average running times of Algorithm 2 versus n

5 Conclusions

This paper addressed the minimum cost path problem with relays. It introduced an algorithm with a time complexity of $\mathcal{O}(Wm + Wn \log \max\{W, n\})$ for the constrained problem with a single objective function. That method is based on a labeling algorithm aided by an auxiliary matrix that stores labels with different feasible weights. A second version of the problem, dealing with the same weight constraint but considering each path and relay costs separately, was also studied. The algorithm proposed for the MCPPR was extended and adapted for this problem. Computational results of label setting and label correcting forms for the two developed algorithms were presented. These results show that the MCPPR can be solved in networks with up to 10 000 nodes and 100 000 arcs in less than 0.256 seconds. As for the BMCPPR instances with 5 000 nodes and 50 000 the sets of non-dominated paths were computed in about 980 seconds.

Acknowledgments This research was partially funded by the Canadian Natural Sciences and Engineering Research Council under grant 39682-05. Part of this work was developed during a visit of Marta Pascoal to the CIRRELT, Montréal, Canada. Her work was partially funded by the FCT Portuguese Foundation of Science and Technology (Fundação para a Ciência e a Tecnologia) under grant SFRH/BSAB/830/2008.

References

[1] R. Bellman. On the theory of dynamic programming. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 38, pages 716–719. National Academy of Sciences, 1952.

[2] J. Bouliane and G. Laporte. Locating postal relay boxes using a set covering algorithm. *American Journal of Mathematical and Management Sciences*, 12:65–74, 1992.

[3] J. Brumbaugh-Smith and D. R. Shier. An empirical investigation of some bicriterion shortest path algorithms. *European Journal of Operational Research*, 43:216–224, 1989.

[4] E. A. Cabral. *Wide area telecommunication network design: problems and solution algorithms with application to the Alberta SuperNet*. Ph.D. thesis, School of Business, University of Alberta, 2005.

[5] E. A. Cabral, E. Erkut, G. Laporte, and S. A. Tjandra. The network design problem with relays, 2005. Unpublished manuscript.

[6] E. A. Cabral, E. Erkut, G. Laporte, and R. A. Patterson. Wide area telecommunication network design: Application to the Alberta SuperNet. *Journal of the Operational Research Society*, 59:1460–1470, 2008.

[7] G. Ghiani, G. Laporte, and F. Semet. The black and white travelling salesman problem. *Operations Research*, 54:366–378, 2006.

[8] F. Guerriero and R. Musmanno. Label correcting methods to solve multicriteria shortest path problems. *Journal of Optimization Theory and Applications*, 111:589–613, 2001.

[9] P. Hansen. Bicriterion path problems. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making: Theory and Applications, Lectures Notes in Economics and Mathematical Systems, 177*, pages 109–127. Springer-Verlag, Berlin, 1980.

[10] E. Martins. On a multicriteria shortest path problem. *European Journal of Operational Research*, 16:236–245, 1984.

[11] A. Skriver and K. Andersen. A label correcting approach for solving bicriterion shortest-path problems. *Computers & Operations Research*, 27:507–524, 2000.