A New Algorithm Based on Ranking Paths Procedure for the Multiobjective Shortest Path Problem

Ernesto Martins[†] José Paixão[*] Mário Rosa[†] José Luis Santos[†]

[*]DEIO, University of Lisbon
Edifício C2, Campo Grande
1749-016 Lisbon, Portugal
jpaixao@fc.ul.pt

[†]Department of Mathematics, University of Coimbra
Apartado 3008
3001-454 Coimbra
{mrosa,zeluis}@mat.uc.pt

1 Introduction

This paper deals with the multiobjective shortest path problem (MSPP), defined over a network where several parameters are associated with each arc and one intends to find a path between two specific vertices (s and t), which minimizes a function of the arc parameters. This is an extension of the classical shortest path problem (SPP) where only one parameter (weight) is considered for each arc and one aims to determine a s-t that minimizes the sum of the weights associated to the arcs in the path (see [3, 9] for extensive bibliographies on the SPP).

Vincke, [10], proposed the multiobjective formulation 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, accessibility, capacity, and others. For a review of the MSPP, the interested reader is referred to Current and Marsh, [2] and Ehrgott and Gandibleux, [4].

Therefore, one intends to determine a s-t 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 (ND) paths, that is, paths for which there is no other path with better values for all the criteria.

The MSPP is known as being a hard problem to solve and Hansen, [6], proved that it may exist an exponential number of ND solutions in the worst case.

2 Algorithms for the MSPP

The large number of applications of the MSPP led to the development of various strategies to obtain the optimal solutions. The existing approaches can be classified in two classes. The first one includes algorithms that determine a single ND path. This is the case of a global optimization problem where a utility function is defined. Interactive procedures, where the user is guiding the search into the criteria space, are also considered in this group. In the second class of approaches for the MSPP, the full set of ND paths is required. The well-known labeling algorithm and the procedures based on ranking paths proposed by Clímaco and Martins, [1], are considered in this group. Mote, [8], suggested the determination of the full set of ND paths in two stages. At the first one, all the supported ND paths are computed. They are always ND ones and can be easily determined because they correspond to the optimal solutions of linear programming relaxation of the MSPP. At the second stage, the remainder ND paths are found by local search using label techniques.

The simplicity of the labeling algorithm makes it a very efficient technique with a large application in this area. Briefly, this algorithm develops a search tree where the root corresponds to the initial node s. The general step consists of selecting a label for a node i in the search tree and expanding it by the arcs outgoing from i. Using the Optimality Principle (OP), some branches can be ignored. In fact, the OP asserts that if p is a ND path, then every subpath of p will be a ND one. So, if a branch q from s to i in the search tree is dominated by another branch w already computed (between the same pair of vertices), then the tree will never be expanded from q. Therefore a dominance test is defined which can be applied either to the new generated label and/or to the label selected to expand the tree. In this way, the labeling algorithm can be viewed as an improvement of the exhaustive search procedure incorporating the dominance test. The labels that are not eliminated by this test are considered as temporary ND ones. Some of them may be turned into dominated when new labels are added to the search tree.

There are two well-known forms for the labeling algorithm. They differ in the way one selects the label from which the search tree is expanded. The first one (label correcting) was proposed by Vincke [10] and follows a FIFO rule to select the label. In this way, the search tree is expanded through levels (wide search). However, the labels with respect to node t (that is, the s-t paths) only appear after developing great part of the search tree. The second one (label setting) was proposed by Hansen, [6], where one looks for a label with a higher probability of being ND at the end of the algorithm. In the present work, a lexicographic shortest label technique is used leading to a kind of deep search procedure for generating the set of ND paths.

3 The new algorithm

The new algorithm proposed in this work can be seen as a variant of the label setting algorithm where the lexicographic shortest s-t path is selected and the search tree is then developed from the nodes involved in the path. Hence, this algorithm computes sequentially deviation shortest paths, [7], until all optimal solutions are obtained. Consequently, it is a full deep search and finds ND s-t paths at a very early stage.

In this algorithm, we use the MPS procedure, [7], to rank lexicographic shortest s-t paths. This procedure is an improvement of Eppstein's algorithm, [5]. It initially computes the lexicographic shortest tree rooted at t ($\mathcal{T}_t^\star$) and the reduced cost $\bar{c}_{i,j}$ of an arc (i,j) with respect to $\mathcal{T}_t^\star$. The main idea of the MPS procedure is the use of a set X of candidates for the next lexicographic shortest path. Its elements can be written as:

$$q^p_{v_i,j} = \langle v_0, \ldots, v_j \rangle \diamondsuit \langle v_i, j \rangle \diamondsuit \mathcal{T}_t^\star(j), i \in \{0, \ldots, r\}$$

where (v_i, j) does not belongs to $\mathcal{T}_t^\star$, $\mathcal{T}_t^\star(j)$ is the path of $\mathcal{T}_t^\star$ from j to t and $p = \langle v_0, \ldots, v_r \rangle$ is one of the lexicographic shortest s-t paths already determined. In this way, the lexicographic shortest element of X will be the next lexicographic shortest s-t paths of the network.

4 Computational results

Computational results are also presented comparing the performance of the three procedures mentioned above. They are reported for a large set of instances and concern to CPU time, space of memory required, percentage of ND s-t paths calculated by iteration and average work executed (which indicates the average computational effort involved on the calculation of each ND path), showing that the new algorithm outperforms the previous ones in a very significant way.

In the computational results, two kind of test problems were solved:

- the bicriteria case: using 10000 nodes networks;
- the general case: considering up to 10 criteria. However, the number of nodes was reduced to 1000 and the number of arcs to 6000.

The next tables shows some results obtained in bicriteria networks. From them, one may conclude that the new algorithm (**newAlg**) outperforms the label one (**labelCor** and **labelSet**) in terms of the average work. The reverse situation occours for the CPU time. This means that, in the beginning, **newAlg** is faster than **labelCor** and **labelSet** to find ND paths. However, it becomes slower than label algorithm each time a new ND paths is computed.

arcs	20000	30000	60000	11000
newAlg	0.01	0.01	0.05	0.14
labelCor	0.00	0.01	0.02	0.07
labelSet	0.01	0.01	0.04	0.12

CPU time in 10000 nodes bicriteria networks

arcs	20000	30000	60000	11000
newAlg	1386.00	2635.30	5725.00	8835.80
labelCor	2949.00	4065.30	6982.50	10058.20
labelSet	2941.90	4062.10	6973.90	10044.50

Number of iterations in 10000 nodes bicriteria networks

arcs	20000	30000	60000	11000
newAlg	248.58	617.15	2254.18	4861.62
labelCor	1538.32	2878.08	5834.34	9804.88
labelSet	1342.09	2450.63	4725.62	8211.90

Average work in 10000 nodes bicriteria networks

In what concerns the multicriteria case (see the following tables), the conclusion is similar to the previous one.

criterias	2	3	5	10
newAlg	0.05	0.13	0.93	12.64
labelCor	0.02	0.06	0.56	10.17
labelSet	0.04	0.12	0.85	12.71

CPU time in 1000 nodes and 6000 arcs multicriteria networks

criterias	2	3	5	10
newAlg	5725.00	10179.30	28433.10	99638.40
labelCor	6982.50	13339.70	35296.50	121711.40
labelSet	6973.90	12850.40	35135.70	121708.50

Number of iterations in 1000 nodes and 6000 arcs multicriteria networks

criterias	2	3	5	10
newAlg	248.58	3875.60	11326.47	30998.70
labelCor	1538.32	10940.02	27835.87	102415.80
labelSet	1342.09	8407.68	23052.95	75210.52

Average work in 1000 nodes and 6000 arcs multicriteria networks

References

[1] J.N. Clímaco and E.Q. Martins, ”On the determination of the nondominated paths in a multiobjective network problem”, *Methods in Operations Research*, 40, 255–258 (1981).

[2] J. Current and M. Marsh, ”Multiple transportation network design and routing problems: taxonomy and annotation”, *European Journal of Operational Research*, 65, 4–19 (1993).

[3] N. Deo and C. Pang, ”Shortest path algorithms: taxonomy and annotation”, *Networks*, 14 (2), 275–323 (1984).

[4] M. Ehrgott and X. Gandibleux, *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Survey*, Kluwer Academic Publishers, Boston, 2002.

[5] D. Eppstein, ”Finding the *k* shortest paths”, *SIAM J. Computing*, 28(2), 652–673 (1998).

[6] P. Hansen, "Bicriterion path problems", in *Multiple Criteria Decision Making: Theory and Application*, editors: G. Fandel and T. Gal, Lectures Notes in Economics and Mathematical Systems, 177, 109-127, Springer Heidelberg, 1980.

[7] E.Q. Martins, M.M. Pascoal, and J.L. Santos, "Deviation algorithms for ranking shortest paths", *International Journal of Foundations of Computer Science*, 10 (3), 247–261 (1999).

[8] J. Mote, I. Murthy, and D.L. Olson, "A parametric approach to solving bicriterion shortest path problems", *European Journal of Operational Research*, 53, 81–92 (1991).

[9] S. Pallotino and M.G. Scutellà, "Shortest path algorithms in transportation models: classical and innovative aspects", in *Equilibrium and Advanced Transportation Modelling*, P. Marcotte and S. Nguyen (eds), 245-281, Kluwer, 1998.

[10] P. Vincke, "Problèmes multicritères", *Cahiers du Centre d'Études de Recherche Opérationelle*, 16, 425–439 (1974).