Diameter-Constrained Trees for General Nonlinear Cost Flow Networks*

Dalila B. M. M. Fontes, *Faculdade de Economia da Universidade do Porto*
Rua Dr. Roberto Frias, 4200-464 Porto, Portugal
Tel: 351-225 571 100, fax: 351-225 505 050 Email: fontes@fep.up.pt

Keywords: Dynamic programming, network flows, diameter constrained trees, general nonlinear costs.

Abstract

In this work we propose a dynamic programming approach to the problem of finding a diameter-constrained minimum cost spanning tree on a flow network. The tree to be found must be rooted at the single source vertex and span all other vertices designated by demand vertices, of the given flow network. Furthermore, a constraint on the number of arcs to be used on the longest path is imposed. Since all demand vertices have a nonnegative flow requirement we not only need to find which arcs are to be used, but also the amount of flow that must be routed through each arc. The total cost, which is to be minimized, is given by nonlinear cost functions of the arc flows. The cost functions considered can be of any type or form, they may be neither convex nor concave and need not be differentiable or continuous. The only requirements are separability and additivity. Computational experiments were performed using randomly generated test problems and the results reported, for small and medium size problems, indicate the effectiveness and efficiency of the method. Moreover the method is independent of the type of cost function considered and also of the number of nonlinear arc costs.

1. Introduction

We consider a problem which is an extension of the classical Minimum Spanning Tree problem (MST). As in the MST problem we want to find a minimum cost tree, rooted at the single source, spanning all other vertices in a given network. However, we consider that all vertices, except for the source vertex, have an integer nonnegative flow requirement and thus we must also find the flow that must be routed along each used arc. A nonlinear flow dependent cost function is associated with each arc. Furthermore, we also consider a bound on the diameter of the tree, which is the maximum number of arcs permitted on any of its paths.

Network flow problems arise frequently in several application areas [10]: transportation, communication, network design and distribution, production and inventory planning, facility location, scheduling and air traffic control. The diameter constraint is important to guarantee a specified level of service, for example guarantee a prescribed level of reliability to potential arc or vertex failure (see e.g. [15]) or to avoid excessive delay of sending a message since this delay is roughly proportional to the number of arcs the message has to traverse (see e.g. [4]).

The problem we address here is NP-*Hard*, which is not surprisingly since the problem of finding a diameter constrained minimum spanning trees is HP-*Hard* for $D \geq 4$ [8] and the problem of finding optimal trees for concave minimum cost network flow problems is also HP-*Hard*, even for the simplest version [11].

Some authors have looked at the diameter constraint versions of classical MST and Steiner tree problems, see for example [9] and the references therein. Many other authors have looked at Minimum Cost Network Flow Problems (MCNFPs): for a recent discussion on general concave MCNFPs, see for example [3, 5] for approximate methods and [6] for exact methods. General nonlinear MCNFPs have arc costs that are neither convex nor concave thus no convexity or concavity properties can be explored in the determination of an optimal solution. This type of problems is also known as indefinite or discontinuous MCNFPs. To the best of

*Research supported by FCT Project POCI/EGE/61823/2004

our knowledge, no optimization methods have been reported in the literature for general nonlinear MCNFPs. Existing literature considers only specific types of indefinite cost functions, namely staircase [2, 13, 14] and sawtooth [14] cost functions.

In this work we go further and consider the problem of finding a diameter constrained tree defined on a flow network at minimum total cost. We consider general nonlinear cost functions, which can be of any type and take any form, as long as they are separable and additive. In addition, the dynamic programming model we propose can also handle arc capacity constraints, i.e. restrictions on the flow that can be routed through each arc. The dynamic programming model and solution algorithm given here are an extension of the work previously developed for single-source concave MCNFPs [7].

The computational results have shown the method to be rather robust, since its performance does not depend of the type of cost functions considered neither depends of the number of nonlinear arc costs. Moreover, the computational results are also basically unaffected by the tightness of the diameter constraints.

2. Problem description and formulation

As explained before, our objective is to find a diameter constrained minimum cost tree spanning all vertices of a given network having general nonlinear arc costs subject to a bound D on the diameter of the tree. The arc flows must be such that all demand vertices are satisfied.

Let $G = (W, A)$ denote a directed network with a set W of $n + 1$ vertices (the source vertex and n demand vertices) and with a set A of m directed arcs. Vertices 1 to n have associated a nonnegative integer demand r_i, which must be satisfied. The total cost, to be minimized, is given by the summation of all costs incurred by both using an arc and routing flow through it, since each arc $(i, j) \in A$ has associated a general nonlinear and nonnegative cost function g_{ij}. The cost of sending r units of flow through an arc, say (i, j) is given by a monotonously increasing function $g_{ij}(r)$ which satisfies $g_{ij}(0) = 0$. (The flow that can be routed through each arc (i, j) may have upper u_{ij} and lower l_{ij} limits.) The diameter bound D forces all paths of the minimum cost tree to have no more than D arcs.

For such a problem the state variable is defined as a triplet (S, x, k) where S is the set of vertices to be supplied and hence spanned, x is the vertex acting as a source and k is the maximum number of arcs in any path. Therefore, at this state we want to find the k diameter constrained tree rooted at vertex x that supplies all vertices in set S at minimum total cost. Define $f(S, x, k)$ to be the minimum cost of such a tree.

At each state, the set of vertices S is to be partitioned into two subsets, $\{S', \bar{S}'\}$ where $S' \subseteq S \setminus \{x\}$ and $\bar{S}'$ is the complement of S' in the set S, that is $\bar{S}' = S - S'$. Then an immediate decision on a vertex to act as a source for set S' (receiving the necessary commodity from vertex x) is made. Therefore, three costs are incurred: one associated with supplying set $\bar{S}' = S - S'$ from vertex x using at most k arcs $f(S - S', x, k)$, another associated with supplying set S' from the chosen vertex, say z using at most $k-1$ arcs $f(S', x, k-1)$ (since arc (x, z) has already been used), and finally a cost associated with making the flow required by the vertices in S', say r, available at vertex z $g_{xz}(r)$.

Since we must make two decisions, one regarding the subset S' and another regarding the vertex z that will be supplying vertices in set S', and both must be done in order to minimize costs, then our formulation will have two minimizations, as follows:

$$f(S, x, k) = \min_{S' \subseteq S \setminus \{x\}} \left[f\left(S - S', x, k\right) + \min_{\substack{z \in S' \\ l_{xz} \leq r \leq u_{xz}}} \left[f(S', z, k-1) + g_{xz}(r) \right] \right]. \quad (1)$$

An illustration, for $k = 4$, is given in Figure 1.

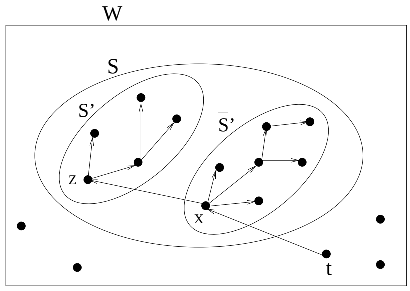

Figure 1: Possible directed trees with diameter bound $D = 4$.

Recursion (1) applies for all $S \subseteq W$ and all $x \in S$. Hence, the cost of an optimal tree supplying all demand vertices in set W from the source vertex t with diameter bound D, is given by $f(W, t, D)$, if one exists.

$$f^* \equiv f(W,t,D) = \min_{S' \subseteq W \setminus \{t\}} \left[f\left(W - S', t, D\right) + \min_{\substack{z \in S' \\ l_{tz} \leq \sum_{i \in S'} r_i \leq u_{tz}}} \left[f(S', z, D-1) + g_{tz}\left(\sum_{i \in S'} r_i\right)\right]\right] \quad (2)$$

Initial conditions are provided by

$$f\left(S, x, k\right) = \begin{cases} 0, & \text{if } S = \{x\} \text{ for all } k \\ \infty, & \text{if } S \neq \{x\} \text{ for all } k. \end{cases} \quad (3)$$

The dynamic programming algorithm

In an initial procedure we label all states as not computed and then initialize states as given by equation (3). The optimal tree $f(W, t, D)$ is obtained by calling the recursive function *Compute*(W, t, D).

Compute(S, x, k)

If state (S, x, k) has already been computed then return $f\left(S, x, k\right)$

Set $min = \infty$

For each $S' \subseteq S$ /* recall that a set is represented by an integer, therefore to consider all subsets it is enough to do a for $i = 1$ to $2^{|S|} - 1$ */

Call $Compute(S \setminus S', x, k)$

If $f\left(S \setminus S', x, k\right) \geq min$ then get another S'

For each $z \in S'$ /* here a cycle for $z = i$ to n followed by a bit test is performed */

If $(x, z) \notin A$ then get another z

$r = \sum_{i \in S'} r_i$

If $r > u_{xz}$ or $r < l_{xz}$ then get another z

If $f\left(S \setminus S', x, k\right) + g_{x,z}\left(r\right) \geq min$ then get another z

```
    Call Compute(S', z, k − 1)
    If f(S \ S', x, k) + g_{x,z}(r) + f(S', z, k − 1) ≥ min then get another z
    min = f(S \ S', x, k) + g_{x,z}(r) + f(S', z, k − 1)
    Store information on:
      subset=S', vertex=z, flow=r, and f(S, x) = min.
  End for
End for
Return f(S, x, k)
```

At the end of the procedure, if $f(W,t,D) = \infty$ then no tree network exists satisfying the diameter bound D and the flow limits; otherwise $f(W,t,D)$ gives the cost associated with an optimal diameter constrained tree. The solution structure, i.e. the arcs used and the amount of flow routed through these arcs, is obtained by a recursive routine that backtracks through the information stored (subset, vertex, and flow) during the computation of intermediate states.

The complexity of the DP algorithm, as expected, increases exponentially with problem size. On the other hand, the DP model performance does not deteriorates with the type, nature, or form of the cost functions used neither with the diameter bounds value.

3. Computational results

The algorithm presented in this paper was implemented in Fortran 90 and computationally evaluated by solving a set of randomly generated test problems. The problems data can be downloaded from the OR-Library [1] and a thorough description of the generation procedure is provided in [5].

Three different types of cost functions are considered: type G1 and type G2 are variations of the fixed-charge cost function where discontinuities other than at the origin are introduced and type G3, for which we consider that arc costs are initially concave and then convex having a discontinuity at the break point. The discontinuity point, $\bar{R}$ was set to 50% of the total demand R.

Types G1 and G2 correspond, respectively, to the so called staircase and sawtooth cost functions, see [12], in our case with two segments.

$$g_{ij}(r) = \begin{cases} 0, & \text{if } r = 0, \\ -a_{ij}r^2 + b_{ij}r + c_{ij} & \text{if } r \leq \bar{R}, \\ a_{ij}r^2 + b_{ij}r + c_{ij} + k & \text{otherwise,} \end{cases}$$

where $a_{ij} = 0$ for G1 and G2, $k = b_{ij}$ for G1, and $k = -b_{ij}$ for G2 and G3.

In tables 1 to 3 we summarize the results obtained for uncapacitated problems involving cost functions of types G1, G2, and G3 with the discontinuity point occurring at 50% of R, and considering three different odd bound values $D = 5$, 7, or 9. Only odd values have been considered since, according to literature, they are much harder to deal with then even values. We report on the average, maximum, and minimum computational time, in minutes, required to solve the problems. For each size, cost function type and D value we solve 30 problem instances.Thus, overall we have solved 450 problem instances for each bound value D.

The results reported show that the computational time increases rapidly with problem size. Nevertheless, as expected, the computational time is independent of the cost function type and also of the D values.

N	Type G1 aver.	max	min	Type G2 aver.	max	min	Type G3 aver.	max	min
10	0.00	0.02	0.00	0.00	0.02	0.00	0.00	0.02	0.00
12	0.03	0.05	0.02	0.03	0.05	0.02	0.03	0.03	0.02
15	0.76	1.17	0.53	0.75	1.12	0.50	0.75	1.17	0.52
17	8.08	12.13	4.40	8.03	11.95	4.35	7.97	12.40	4.45
19	68.09	124.58	32.00	53.94	107.40	19.17	71.02	125.40	31.98

Table 1: Computational performance for bound value $D = 5$.

N	Type G1 aver.	max	min	Type G2 aver.	max	min	Type G3 aver.	max	min
10	0.00	0.02	0.00	0.00	0.02	0.00	0.00	0.02	0.00
12	0.02	0.05	0.00	0.02	0.03	0.02	0.02	0.05	0.00
15	0.66	1.00	0.45	0.61	0.88	0.43	0.62	0.95	0.47
17	7.38	12.02	3.78	6.94	10.83	3.57	6.96	11.43	4.12
19	71.55	139.98	32.72	71.52	145.77	32.03	74.19	134.03	33.13

Table 2: Computational performance for bound value $D = 7$.

4. Conclusions

In this paper we have presented a DP methodology for finding diameter constrained trees that satisfy customer demands at minimum cost. The diameter constraint forces the paths in the trees to have no more than D arcs, where D is the diameter bound value. The cost functions considered may be neither differentiable nor continuous. Also, they might be neither convex nor concave having only to be separable and additive.

No other works have been founded in literature for diameter constrained trees that involve general nonlinear arc costs that are neither convex nor concave. Furthermore, the works found are on graphs and do not involve flow routing on the arcs and flow supplying to the customers.

A large number of randomly generated test problems of varying size and complexity was used to evaluate the algorithms performance. Overall, computational experiments were carried out on 450 problem instances for each of the three odd bound values considered. The results have shown the DP algorithm to be effective at solving such a problem for any type of cost function. Furthermore, the algorithm is also efficient, although only for small and medium size problem instances, since computational requirements grow rapidly with problem size.

References

[1] J. E. Beasley. Or-Library. http://www.brunel.ac.uk/depts/ma/research/jeb/info.html.

[2] C. T. Bornstein and R. Rust. Minimizing a sum of staircase functions under linear constraints. *Opti-*

N	Type G1 aver.	max	min	Type G2 aver.	max	min	Type G3 aver.	max	min
10	0.00	0.01	0.00	0.00	0.02	0.00	0.00	0.02	0.00
12	0.02	0.04	0.00	0.02	0.03	0.00	0.02	0.03	0.00
15	0.57	0.90	0.34	0.57	0.87	0.42	0.57	0.90	0.43
17	7.16	11.46	3.85	6.81	10.55	3.70	6.65	10.67	3.72
19	57.60	107.62	25.25	60.97	115.27	25.17	60.13	116.72	25.40

Table 3: Computational performance for bound value $D = 9$.

mization, 19:181–190, 1988.

[3] R. E. Burkard, H. Dollani, and P. H. Thach. Linear approximations in a dynamic programming approach for the uncapacitated single-source minimum concave cost network flow problem in acyclic networks. *Journal of Global Optimization*, 19:121–139, 2001.

[4] V. Chepoi and Y. Vaxes. Augmenting trees to meet biconnectivity and diameter constraints. *Algorithmica*, 33:243–262, 2002.

[5] D. B. M. M. Fontes, E. Hadjiconstantinou, and N. Christofides. Upper bounds for single source uncapacitated minimum concave-cost network flow problems. *Networks*, 41:221–228, 2003.

[6] D. B. M. M. Fontes, E. Hadjiconstantinou, and N. Christofides. A branch-and-bound algorithm for concave network flow problems. *Journal of Global Optimization*, 34:127–155, 2006.

[7] D. B. M. M. Fontes, E. Hadjiconstantinou, and N. Christofides. A dynamic programming approach for solving single-source uncapacitated concave minimum cost network flow problems. *European Journal of Operational Research*, 174:1205–1219, 2006.

[8] M. Garey and D. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. Freeman, San Francisco, 1979.

[9] L. Gouveia, T. L. Magnanti, and C. Requejo. A 2-path approach for odd-diameter-constrained minimum spanning and steiner trees. *NETWORKS*, 44(4):254–265, 2004.

[10] G. M. Guisewite. Network problems. In R. Horst and P. M. Pardalos, editors, *Handbook in Global Optimization*, pages 506–648. Kluwer Academic, 1994.

[11] G. M. Guisewite and P. M. Pardalos. Algorithms for the single-source uncapacitated minimum concave-cost network flow problem. *Journal of Global Optimization*, 3:245–265, 1991.

[12] D. Kim. Piecewise linear network flow problems. In C. A. Floudas and P. M. Pardalos, editors, *Encyclopedia of Optimization*. Kluwer Academic Publisher, 2003.

[13] D. Kim and P. M. Pardalos. A dynamic domain contraction algorithm for nonconvex piecewise linear network flow problems. *Journal of Global Optimization*, 17:225–234, 2000.

[14] B. W. Lamar. A method for solving network flow problems with general nonlinear arc costs. In D.-Z. Du and P. M. Pardalos, editors, *Network optimization Problems*. World Scientific, 1993.

[15] K. Woolston and S. Albin. The design of centralized networks with reliability and availability constraints. *Computers and Operations Research*, 15:207–217, 1988.