Advances in Steiner Trees
D.-Z. Du, J.M. Smith, and J.H. Rubinstein (Eds.) pp. 1–10
©1998 Kluwer Academic Publishers

Grade of Service Steiner Trees in Series-Parallel Networks

Charles J. Colbourn and Guoliang Xue
Department of Computer Science
University of Vermont, Burlington, VT 05405
E-mail: {colbourn, xue}@cs.uvm.edu

Abstract

The grade of service Steiner tree problem is to determine the minimum total cost of an assignment of a grade of service to each link in a network, so that between each pair of nodes there is a path whose minimum grade of service is at least as large as the grade required at each of the end nodes. This problem has important applications in communication networks and in transportation. It generalizes the Steiner tree problem, in which there are two grades of service. When the network has n nodes and there are r grades of service, an algorithm to determine the cost of a grade of service Steiner tree is given which runs in $O(r^3n)$ time on series-parallel networks.

1 The Problem

A communication network is often modeled by an undirected graph $G = (V, E)$ where V is the set of vertices and E is the set of edges. With each edge $e \in E$, there is an associated cost $c(e) \geq 0$. The traditional Steiner tree problem asks for a minimum cost subgraph of G which spans a given subset of vertices. Such problems find important applications in communication networks and have been studied by many researchers. We refer the readers to the survey papers [8, 13] and the book [9] for details.

In this traditional network model, there is only one cost function defined on the edges of the network and there are only two kinds of service requested by the vertices of the network: connected, or not (necessarily) connected. In real-life problems, the vertices may have more than two kinds of service request and the edges may have service-dependent cost functions. For example, all major universities are to be connected to the Internet via a $T3$-line, while other universities and colleges are to be connected to the Internet via a $T1$-line. Linking two points with a $T3$-line is more expensive than linking the two points with a $T1$-line, but the latter has nonzero cost and therefore is more expensive than not linking at all. As an example in transportation, all major cities are to be connected via interstate highways, while small cities and towns must be connected via local highways. Again, the cost of building an interstate highway is higher than the cost of building a local highway.

In this paper, we study the grade of service Steiner tree (GOSST) problem formally defined as follows.

INPUT:

1. A graph $G = (V, E)$ and an integer r.
2. A function $g : V \mapsto \{0, 1, \ldots, r-1\}$. The integer $g(x)$ for $x \in V$ is the *grade of service* required by vertex v.
3. A function $c : E \times \{0, \ldots, r-1\} \mapsto \mathbb{R} \cup \{\infty\}$ (the real numbers adjoined with the number infinity). The value $c(e, g)$ is the *cost* of providing grade of service at least g on the edge e. The function c satisfies $c(e, i) \geq c(e, i-1)$ for $1 \leq i < r$ and $c(e, 1) > c(e, 0) = 0$ for each edge $e \in E$.

PROBLEM: Determine the minimum total cost of an assignment, if one exists, of one grade of service to each edge, so that for every two vertices x and y in G, there is a path connecting x and y on which the minimum grade of service assigned is at least $\min\{g(x), g(y)\}$. If no such assignment exists, report ∞ as the solution.

When $r = 2$, the GOSST problem has only two grades of service. Grade 0 offers no service at all, while grade 1 offers full service. Hence the well known Steiner tree problem is a special case of the GOSST problem. Since the Steiner tree problem is NP-hard [3], the GOSST problem is also NP-hard. Therefore the existence of efficient optimal algorithms for solving either problem is very unlikely. In this paper, we are interested in solving the GOSST problem on a class of sparse networks known as series-parallel graphs which are subgraphs of 2-trees defined below.

Following [11], a 2-tree can be defined recursively as follows, and all 2-trees may be obtained in this way. A triangle is a 2-tree. Given a 2-tree and an edge $\{x, y\}$ of the 2-tree, we can add a new vertex z adjacent to both x and y; the result is a 2-tree. A series-parallel graph (also known as a partial 2-tree) is a subgraph of a 2-tree. With this definition, one can see that a 2-tree on n vertices has $2n - 3$ edges and $n - 2$ triangles.

Farley [6] demonstrated that 2-trees are isolated failure immune (IFI) networks. Wald and Colbourn [11] showed that a minimum IFI network is a 2-tree. This fact made 2-trees an important class of fault tolerant networks. Wald and Colbourn also showed that the Steiner tree problem on a series-parallel graph is tractable, presenting a linear time algorithm.

In this paper, we show that the GOSST problem on a series-parallel graph is also tractable. Using the partial 2-tree representation of series-parallel graphs, we present an algorithm for solving the GOSST problem on series-parallel graphs whose running time is linear in n, the number of vertices of G, and polynomial in r, the number of grades of service.

The literature on the GOSST problem is relatively recent and most research has focused on the cases where $r = 2$ or 3. Current *et al.* [2] formulated this problem as an integer linear programming problem and developed a heuristic algorithm which employs a K shortest paths algorithm and a minimum spanning tree algorithm. Duin and Volgenant [5] proposed two heuristic algorithms for the special case where $r = 3$. Balakrishnan, Magnanti and Mirchandani [1] proposed an approximation algorithm with performance ratio of $\frac{4}{3}$ for the special case where $r = 3$ and that $c(e, 3)/c(e, 2)$ is a constant for all edges. For general values of r, Mirchandani [10] proposed an approximation algorithm whose performance ratio is $r\rho + 1$, where ρ is the best performance ratio of a Steiner tree heuristic. Related work can also be found in [4, 14].

The grade-of-service Steiner tree problem is also closely related to the design of survivable networks. In that case, again various grades of service are required, but multiple paths can be employed to achieve the grade. See [7] for an excellent survey of this topic, and [12] for an efficient algorithm on series-parallel networks in the biconnected case.

2 The Algorithm

First, to solve GOSST on series-parallel graphs, it suffices to solve it for 2-trees. To see this, use the Wald-Colbourn method [11] to embed a series-parallel graph in a 2-tree in $O(n)$ time. Whenever an edge e is added, simply set $c(e, 0) = 0$ and $c(e, i) = \infty$ for $1 \leq i < r$. The cost of the GOSST does not change.

Our basic strategy is standard. We reduce the 2-tree by suppressing degree two nodes, and by merging parallel edges, until only a single edge remains. At every point, an edge serves to represent the subgraph thus far reduced onto it, and a number of measures are associated with each edge.

To facilitate the presentation and implementation, we arbitrarily assign a direction to each edge $e = \{x, y\}$, replacing it with directed arcs $e = (x, y)$ and $\overline{e} = (y, x)$. This is a notational convenience only, to permit us to refer to the first vertex x and the second vertex y of edge e.

With every arc e, a total of $2r^2$ measures are maintained. Since an n-vertex 2-tree has $2n - 3$ edges and hence $4n - 6$ arcs are represented, $(4n - 6)2r^2$ measures are to be stored. The $2r^2$ measures for an arc e have the following representation and meaning. The measures for arc $e = (x, y)$ are of the form $(\alpha, \beta, \kappa, e)$ where $\alpha, \beta \in \{0, 1, \ldots, r - 1\}$ and $\kappa \in \{C, D\}$. The interpretation of each measure depends upon the subgraph S_e thus far reduced onto the arc e. In each case, α indicates the grade of service expected at x (when the cost is finite, this is always at least $g(x)$, but may be higher if x is assumed to be needed to connect a vertex within S_e to a vertex not in S_e). Similarly, β indicates the grade of service expected at y. Depending upon the selection of κ, various assignments are permitted to the arcs of S_e in which each arc and its reversal receive the same grade. Let S_e be the graph thus far reduced onto the arc $e = (x, y)$. If $\kappa = C$, let T_e be S_e; if $\kappa = D$, let T_e be S_e together with the *auxiliary arc* (x, y), whose cost for grade of service $\mathtt{min}\{\alpha, \beta\}$ is 0. Then we define a *κ-acceptable assignment* to be an assignment of grades of service to the edges of S_e for which

1. For every two vertices w, z of S_e, there is a path from w to z in T_e whose minimum grade of service is at least $\mathtt{min}\{g(w), g(z)\}$; and

2. There is a path from x to y in T_e whose minimum grade of service is at least $\mathtt{min}\{\alpha, \beta\}$.

3. If w is a vertex of S_e, then there is a path in T_e from w to x whose minimum grade of service is at least $\mathtt{min}\{\alpha, g(w)\}$, and a path from w to y whose minimum grade of service is at least $\mathtt{min}\{\beta, g(w)\}$.

When $\kappa = D$, the second condition is met automatically by the presence of the auxiliary arc in T_e. Then $(\alpha, \beta, \kappa, e)$ represents the minimum cost of a κ-acceptable assignment to the arcs of S_e, and takes on the value ∞ when there is no such acceptable assignment.

The letters 'C' and 'D' represent *connected* and *disconnected*, and serve to remind us whether we have already required the ends of the edge to be connected. We also compute the value $H(e)$, the maximum grade of service required on a vertex of S_e *other than* x and y. This value is needed to determine when a vertex in S_e requires a grade of service greater than that provided by x or y. If a vertex in the remainder of the graph also expects such a higher grade of service, we shall be unable to provide it.

Arc e and its reverse $\overline{e}$ are both retained, but $(\alpha, \beta, \kappa, e)$ and $(\beta, \alpha, \kappa, \overline{e})$ are the same, so we actually only need to store one of the sets of measures.

If we can calculate these measures, we can solve GOSST. To see this, suppose that we have reduced the entire graph onto the arc $e = (x, y)$. Then, by the definition of the measures, we need only examine $(g(x), g(y), C, e)$.

So it remains to establish methods to initialize and update these measures. Initially, we want each arc $e = (x, y)$ to have just the single edge $\{x, y\}$ in S_e. Hence we initialize as shown in Table 1.

Next we describe how to suppress a vertex z of degree two. Supposing that $e' = (x, z)$ and $e'' = (z, y)$ are the two arcs incident at z, we replace these by the arc $e = (x, y)$ and compute measures for e. In the expressions given in Table 2, when a minimum is taken over an empty set, its value is ∞. When a measure is referenced whose expected grade of service is outside the permissible range, its value is taken to be ∞. To understand the reductions, observe that whenever

$$\begin{aligned}
(\alpha,\beta,C,e) &= \begin{cases} \infty \text{ if } \alpha < g(x) \text{ or } \beta < g(y) \\ c(e,\mathtt{min}\{\alpha,\beta\}) \text{ otherwise} \end{cases} \\
(\alpha,\beta,D,e) &= \begin{cases} \infty \text{ if } \alpha < g(x) \text{ or } \beta < g(y) \\ 0 \text{ otherwise} \end{cases} \\
H(e) &= 0
\end{aligned}$$

Table 1: Initialization

$$\begin{aligned}
(\alpha,\beta,C,e) &= \mathtt{min}\{(\alpha+1,\beta,C,e),(\alpha,\beta+1,C,e), \\
&\quad \mathtt{min}\{(\alpha,\gamma,C,e')+(\gamma,\beta,C,e''): \\
&\qquad \gamma \geq \mathtt{min}\{\alpha,\beta\}, \\
&\qquad \mathtt{max}\{\mathtt{min}\{\alpha,H(e'')\},\mathtt{min}\{\beta,H(e')\}\} \leq \gamma, \\
&\qquad ((H(e') \leq \mathtt{max}\{\alpha,\gamma\}) \text{ or } (H(e'') \leq \mathtt{max}\{\gamma,\beta\}))\}\} \\
(\alpha,\beta,D,e) &= \mathtt{min}\{(\alpha,\beta,C,e), \\
&\quad \mathtt{min}\{(\alpha,\gamma,C,e')+(\gamma,\beta,D,e''): \\
&\qquad \alpha \geq \mathtt{min}\{\gamma,\beta\},\ \gamma \geq g(z), \\
&\qquad ((H(e') \leq \mathtt{max}\{\alpha,\gamma\}) \text{ or } (H(e'') \leq \mathtt{max}\{\gamma,\beta\}))\}, \\
&\quad \mathtt{min}\{(\alpha,\gamma,D,e')+(\gamma,\beta,C,e''): \\
&\qquad \beta \geq \mathtt{min}\{\alpha,\gamma\},\ \gamma \geq g(z), \\
&\qquad ((H(e') \leq \mathtt{max}\{\alpha,\gamma\}) \text{ or } (H(e'') \leq \mathtt{max}\{\gamma,\beta\}))\}\} \\
H(e) &= \mathtt{max}\{H(e'),g(z),H(e'')\}
\end{aligned}$$

Table 2: Suppressing a Degree Two Vertex

$$\begin{aligned}
(\alpha,\beta,C,e) &= \texttt{min}\{(\alpha+1,\beta,C,e),(\alpha,\beta+1,C,e),\\
&\quad \texttt{min}\{(\alpha,\beta,C,e')+(\alpha,\beta,D,e''),(\alpha,\beta,D,e')+(\alpha,\beta,C,e''):\\
&\qquad \texttt{max}\{\alpha,\beta\}\geq\texttt{min}\{H(e'),H(e'')\}\}\}\\
(\alpha,\beta,D,e) &= \texttt{min}\{(\alpha,\beta,C,e),\\
&\quad (\alpha,\beta,D,e')+(\alpha,\beta,D,e'')\text{ if }\texttt{max}\{\alpha,\beta\}\geq\texttt{min}\{H(e'),H(e'')\}\}\\
H(e) &= \texttt{max}\{H(e'),H(e'')\}
\end{aligned}$$

Table 3: Reducing Multiple Edges

$H(e') > \texttt{max}\{\alpha,\gamma\}$ and $H(e'') > \texttt{max}\{\gamma,\beta\}$, there is a vertex in $S_{e'}$ and a vertex in $S_{e''}$ which cannot establish a path of the required grade of service. Other conditions ensure that x can reach the vertex of maximum grade in $S_{e''}$, and that y can reach the vertex of maximum grade in $S_{e'}$.

Now we describe the reduction of two arcs e' and e'', both of the form (x,y), and their replacement by the single arc e again of the form (x,y). Table 3 gives the corresponding computations.

Obviously updating any particular measure takes O(r) time assuming that arithmetic operations take constant time. It follows that the entire graph can be reduced in O(r^3n) time, and hence we have proved the following theorem.

Theorem 2.1 *The GOSST problem can be solved in O(r^3n) time on an n-vertex series-parallel graph.* □

3 An Example

A sample series-parallel network is illustrated in Figure 1. Every vertex is labeled with a capital letter followed by an integer in parentheses which gives the degree of service request of that vertex. For example, vertex A has a service request of grade 2 while vertex C has a service request of grade 0. An edge is labeled with a positive integer followed by the letter g. For example, the edge $\{A,B\}$ is labeled with $2g$ which means that the cost of providing a service of grade g on this edge has a cost of $2\times g$. The given network is only a partial 2-tree. Adding the dotted edge connecting vertices A and U makes it a 2-tree. We use this example to illustrate the reduction steps of our algorithm.

We write arcs as pairs of vertices, using AB to denote the arc (A,B), for example. Primes and double primes are used to indicate new measures on the arc after a reduction is completed.

The algorithm is carried out in the following way. After the initialization step using the rules in Table 1, we perform five pairs of reductions. The first consists

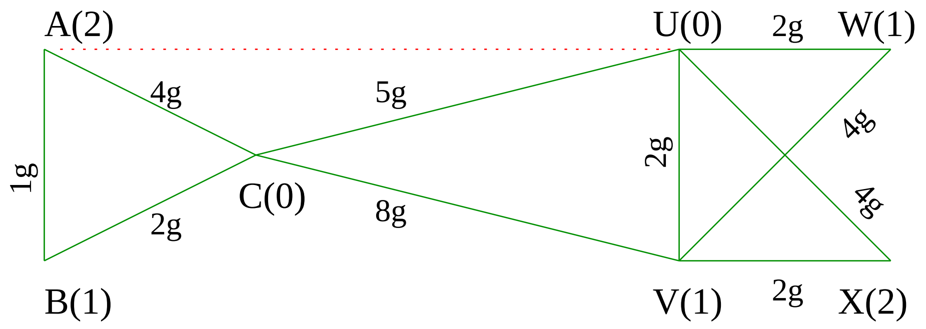

Figure 1: A series-parallel network: The number in parenthesis at each vertex represents the grade of service requested by the vertex. For example, the grade of service request at vertex U is 0 while that at vertex V is 1. The edge label represents the edge cost function. For example, the cost to provide grade-g service on edge $\{U, X\}$ is $4 \times g$ for $g = 0, 1, 2$.

of suppressing vertex X using the rules in Table 2. A new edge connecting U and V is created as the result of the suppression of vertex X. This new edge is then reduced along with the arc UV using the rules in Table 3, to form UV'. The two steps in eliminating the vertex are tabulated in the two columns headed by UV' in Table 5. The next four pairs of reductions are carried out with the suppression of the vertices W, V, U, C, in the given order, to form arcs UV'', CU', AC' and AB', respectively. The values of the $2r^2(2n-3)$ measures and their updates are shown in Tables 4-5. The $(\alpha\beta\kappa, PQ)$ entry gives the value of $(\alpha, \beta, \kappa, (P, Q))$. For example, the $(00C, VX)$ entry is ∞ which means $(0, 0, C, (V, X)) = \infty$ and the $(21C, AC)$ entry is 4 which means $(2, 1, C, (A, C)) = 4$. The cost of the GOSST for the given example can be read from Table 5; it is 26.

	AB	CB	AC	AU	UC	VC	UV	UW	WV	UX	XV
00C	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	∞
01C	∞	0	∞	∞	0	∞	0	0	∞	∞	∞
02C	∞	0	∞	∞	0	∞	0	0	∞	0	∞
10C	∞	∞	∞	∞	0	0	∞	∞	∞	∞	∞
11C	∞	2	∞	∞	5	8	2	2	4	∞	∞
12C	∞	2	∞	∞	5	8	2	2	4	4	∞
20C	∞	∞	0	∞	0	0	∞	∞	∞	∞	∞
21C	1	2	4	∞	5	8	2	2	4	∞	2
22C	2	4	8	∞	10	16	4	4	8	8	4

Table 4: Initial Values of C Measures.

	UV'		UV''		UC'		AC'		AB'	
H	2	2	1	2	2	2	2	2	2	2
00C	∞	∞	∞	∞	6	6	∞	∞	∞	∞
00D	∞	∞	∞	∞	6	6	∞	∞	∞	∞
01C	2	2	4	6	14	11	∞	∞	∞	∞
01D	2	2	4	6	14	11	∞	∞	∞	∞
02C	4	4	4	8	24	20	∞	∞	∞	∞
02D	4	4	4	8	24	20	∞	∞	∞	∞
10C	∞	∞	∞	∞	6	6	∞	∞	∞	∞
10D	∞	∞	∞	∞	6	6	∞	∞	∞	∞
11C	6	4	6	6	14	11	∞	∞	∞	∞
11D	2	2	2	4	6	6	∞	∞	∞	∞
12C	8	6	6	8	24	20	∞	∞	∞	∞
12D	4	4	2	6	22	20	∞	∞	∞	∞
20C	∞	∞	∞	∞	10	10	∞	28	∞	∞
20D	∞	∞	∞	∞	10	10	∞	28	∞	∞
21C	10	8	6	10	18	15	∞	28	30	26
21D	8	8	2	10	10	10	∞	28	28	26
22C	12	8	6	10	26	20	∞	28	32	26
22D	4	4	2	6	10	10	20	20	24	24

Table 5: Computed Values of Measures after Reductions.

This method for computing the value of the GOSST is a dynamic programming algorithm. Whenever an updated value of a measure is computed, we may set pointers to the items which resulted in this new value (the one which attains the minimum). Once the optimal value of the GOSST is computed, we may find the optimal tree by tracing out the pointers which we set during the computation. For our example, the optimal tree is illustrated in Figure 2. Vertices B, C, U and V all receive service of grade 2, which is higher than being requested.

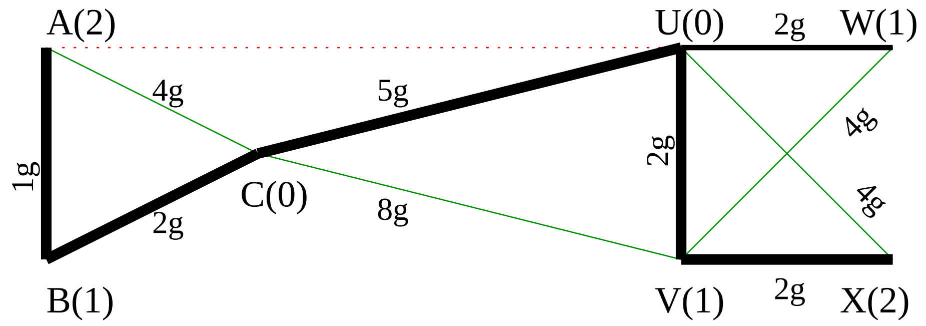

Figure 2: The optimal solution: The thickest edges have service of grade 2. The edge connecting U and W has service of grade 1. All other edges have service of grade 0.

4 Conclusions

In this paper, we have studied the grade of service Steiner tree problem on a series-parallel network and have shown that the optimal solution can be found in $O(r^3 n)$ time where n is the number of vertices in the network and r is the number of grades of service. Possible future research directions include generalizing the techniques used in this paper to other sparse networks and to problems where each link has an operational probability.

Acknowledgments

The research was supported in part by the U.S. Army Research Office under Grants DAAG55-98-1-0272 (Colbourn) and DAAH04-96-1-0233 (Xue). Research of G.L. Xue is also supported by the National Science Foundation grants ASC-9409285 and OSR-9350540.

References

[1] A. Balakrishnan, T.L. Magnanti, and P. Mirchandani, Modeling and heuristic worst-case performance analysis of the two-level network design problem, *Management Science*, Vol. 40(1994), pp. 846–867.

[2] J.R. Current, C.S. Revelle, and J.L. Cohon, The hierarchical network design problem, *European Journal of Operational Research*, Vol. 27(1986), pp. 57–66.

[3] M.R. Garey, R.L. Graham and D.S. Johnson, The complexity of computing Steiner minimal trees, *SIAM Journal on Applied Mathematics*, Vol. 32(1977), pp. 835–859.

[4] C. Duin and A. Volgenant, Reducing the hierarchical network design problem *European Journal of Operational Research*, Vol. 39(1989), pp. 332–344.

[5] C. Duin and T. Volgenant, The multi-weighted Steiner tree problem, *Annals of Operations Research*, Vol. 33(1991), pp. 451–469.

[6] A.M. Farley, Networks immune to isolated failures, *Networks*, Vol. 11(1981), pp. 255–268.

[7] M. Grötschel, C.L. Monma, and M. Stoer, Design of survivable networks, in: *Network Models* (M. Ball et al., eds.) Elsevier North-Holland, 1995, pp. 617-672.

[8] F. Hwang and D. Richards, Steiner tree problems, *Networks*, Vol. 22(1992), pp. 55–89.

[9] F.K. Hwang, D.S. Richards and Pawel Winter, *The Steiner tree problem*, *Annals of Discrete Mathematics*, Vol. 53, North-Holland, 1992.

[10] P. Mirchandani, The multi-tier tree problem, *INFORMS Journal on Computing*, Vol. 8(1996), pp. 202–218.

[11] J.A. Wald and C.J. Colbourn, Steiner trees, partial 2-trees, and minimum IFI networks, *Networks*, Vol. 13(1983), pp. 159–167.

[12] P. Winter, Generalized Steiner problem in series-parallel networks, *J. Algorithms*, Vol. 7(1986), pp. 549-566.

[13] P. Winter, The Steiner problem in graphs: a survey, *Networks*, Vol. 17(1987), pp. 129–167.

[14] G.L. Xue, Grade of service Steiner minimum trees in the Euclidean plane, preprint, 1998.