UPGRADING BOTTLENECK CONSTRAINED FORESTS
(Extended Abstract)

S. O. KRUMKE, M. V. MARATHE, H. NOLTEMEIER, S. S. RAVI,
AND H.-C. WIRTH

1. Introduction

Network upgrade problems investigate cases where the cost of implementing a computed network from scratch exceeds the cost of modifying an already installed network. There are two main models for network upgrading problems: the *edge upgrading model* [Ber92, Phi93, KN$^+$96] and the *node upgrading model* [PS95, KM$^+$97], where in the latter one upgrading a node reduces the delay on all edges incident with the upgraded node.

In this paper, we concentrate on the node upgrading problem Upgrading-CFP: the goal is to find a minimum cost set of vertices, such that after the upgrade of those vertices the resulting graph contains a constrained forest (as specified by a proper function) of bottleneck weight at most a given threshold. This problem generalizes a number of network design problems, such as node weighted Steiner trees.

We provide the first approximation algorithm for the above problem. The performance guarantee of our algorithm is $2\ln(\sqrt{e}/2\cdot|K|)$, where K is the set of terminals. This matches the $\ln |K|$ lower bound provided by the hardness result within a constant factor.

2. Preliminaries

The *node based upgrading model* used in this paper generalizes the model from [PS95] and was introduced in [KM$^+$97]. Let $G = (V, E)$ be a connected undirected graph with $n := |V|$ vertices and $m := |E|$ edges. For each edge $e \in E$, we are given three integers $d_0(e) \geq d_1(e) \geq d_2(e) \geq 0$. The value $d_i(e)$ represents the *delay* of the edge e if exactly i of its endpoints are upgraded.

For each node $v \in V$ the value $c(v)$ specifies the upgrading cost of node v. The cost of upgrading all the nodes in $W \subseteq V$ equals $c(W) := \sum_{v\in W} c(v)$. By d_W we denote the edge weight function resulting from the upgrade of node set W. The *bottleneck graph* $G_{d_W \leq D}$ of G contains the same set of vertices as G but only those edges e of weight $d_W(e) \leq D$.

Constrained forest problems were introduced by Goemans and Williamson in [GW95]. For a certain family of cuts in a graph G, one searches to find a minimum cost subgraph intersecting all the cuts in the family. A function $f\colon 2^V \to \{0, 1\}$ is used to define the family of cuts: $f(U) = 1$ if and only if the cut $\delta(U)$ is in the family. Here, $\delta(U)$ denotes the set of edges that have exactly one endpoint in U. The function f is termed *proper* if it satisfies the following two conditions:

(i) $f(U) = f(V - U)$ for all $U \subseteq V$; and
(ii) If $A \cap B = \emptyset$, then $f(A) = f(B) = 0$ implies that $f(A \cup B) = 0$.

Any subset F of the edges such that $|F \cap \delta(U)| \geq f(U)$ for all $\emptyset \neq U \subset V$ is termed a *constrained forest* with respect to f. The *set of terminals* specified by f is $K := \{ v \in V : f(\{v\}) = 1 \}$.

Proper functions can be used to re-formulate many interesting problems as constrained forest problems, e.g. the minimum spanning tree problem, the Steiner tree problem, and the shortest s-t-path problem. For instance, the proper function f, defined by $f(U) := 1$, if $0 < |U \cap K| < |K|$, and zero otherwise, is a proper function for which any (inclusion-wise) minimal constrained forest is a Steiner tree spanning the terminals in K.

Definition 1 (Problem UPGRADING-CFP). Given a graph $G = (V, E)$ with edge weights $d_0 \geq d_1 \geq d_2$, node weights c as before, a bound D and a proper function f, find a minimum cost set $W \subseteq V$ of nodes such that the bottleneck graph $G_{d_W \leq D}$ contains a constrained forest (with respect to f).

It is easy to show that UPGRADING-CFP generalizes the problem of finding a node-weighted Steiner tree of minimum cost. We also show that UPGRADING-CFP is at least as hard to approximate as MINSETCOVER. Using recent non-approximability results [Fei96] we obtain:

Theorem 2. *Let $\varepsilon > 0$ be arbitrary. Unless $\mathsf{NP} \subseteq \mathsf{DTIME}(N^{\mathcal{O}(\log\log N)})$, there is no approximation algorithm for* UPGRADING-CFP *(with proper function f) with performance $(1 - \varepsilon) \ln |K|$, where K is the set of terminals. This result continues to hold even if $c(v) = 1$ for all vertices $v \in V$.*

3. ALGORITHM AND PERFORMANCE GUARANTEE

We give a brief overview of our algorithm. The set W of upgraded nodes is initially empty. Our algorithm maintains the connected components of the current bottleneck graph $G_{d_W \leq D}$. A component C is called *active*, if $f(C) = 1$. In each iteration the algorithm merges at least two active components by upgrading nodes in the network. It terminates when no active components remain. The basic rule which nodes to upgrade in an iteration is to select a set that gives the best improvement ratio. This ratio is measured by the quotient of the upgrade cost of the vertices and the decrease of the number of active components.

Figure 1 shows an outline of our algorithm. The calculation of the *minimum upgrading cost* to obtain a path from a vertex v to a connected component C can be accomplished by two single source shortest paths computations on an auxiliary graph, where the length of a path is defined to be the node costs of the nodes on the path excluding the source vertex. It can then be shown that the algorithm can be implemented to run in time $\mathcal{O}(|K|\, n^2 \log n)$, where K is the set of terminals.

Algorithm Upgrade

1. Start with an empty set of upgraded nodes $W := \emptyset$
2. **while** $G_{d_W \le D}$ contains an active component
   (a) Compute for each node v and each active component C the minimum upgrading cost to obtain a path of bottleneck delay at most D from v to C.
   (b) Select a node $v \in V$ and a subset $C_1, \dots, C_l$ of active components with minimum quotient cost
   $$q(v) = \frac{\text{cost of upgrading the nodes on the paths}}{l}$$
   (c) Upgrade the nodes identified in the previous step and add them to W.

Figure 1. Node upgrading for constrained forests.

The proof of the performance guarantee uses the notion of a spider covering, which extends the definitions given in [KR93].

Definition 3 (Spider, Spider Covering). A *spider* is a tree with at most one node of degree greater than two. Given a collection $M := \{M_1, \dots, M_p\}$ of disjoint node sets, where each M_i induces a connected subgraph of G, a *spider covering of M in G* is a collection of node disjoint nontrivial spiders such that: (1) each set M_i contains a foot or a center of a spider, and (2) if a set M_i contains a foot, then M_i does not contain any other foot or center.

We show that such a spider covering always exists. This enables us to decompose optimal solutions and show that in each iteration a good choice of vertices to upgrade is possible.

Lemma 4. *Let v be a node chosen in Step 2b of Algorithm* Upgrade *and let $c(U)$ denote the total cost of the nodes added to the solution set W in this iteration. Let there be p active clusters before v is chosen and assume that in this iteration r clusters are merged. Then: $c(U)/r \le \mathrm{OPT}/p$. Here OPT denotes the total upgrade cost in an optimum solution.*

The proof of the last lemma is accomplished by considering a spider covering of the active components. The cost of a spider is defined as the cost of the vertices from the optimum solution which have not been upgraded yet. Our arguments show that one center of a spider in the covering has quotient cost at most OPT/p. Since our algorithm chooses a node with minimum quotient cost this then shows the claim.

Theorem 5. *Algorithm* Upgrade *always finds a solution of cost at most $2\ln(\sqrt{e}/2 \cdot |K|)$ times the optimum upgrading cost.*

The proof of the last theorem uses a potential function counting the number of active components. By using Lemma 4 in conjunction with the potential function argument, we establish the performance of the algorithm as stated.

References

[Ber92] O. Berman, *Improving the location of minisum facilities through network modification*, Annals of Operations Research **40** (1992), 1–16.

[Fei96] U. Feige, *A threshold of* $\ln n$ *for approximating set cover*, Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC'96), 1996, pp. 314–318.

[KM+97] S. O. Krumke, M. V. Marathe, H. Noltemeier, R. Ravi, S. S. Ravi, R. Sundaram, and H. C. Wirth, *Improving spanning trees by upgrading nodes*, Proceedings of the 24nd International Colloquium on Automata, Languages and Programming (ICALP'97), Lecture Notes in Computer Science, vol. 1256, 1997, pp. 281–291.

[KN+96] S. O. Krumke, H. Noltemeier, S. S. Ravi, M. V. Marathe, and K. U. Drangmeister, *Modifying networks to obtain low cost trees*, Proceedings of the 22nd International Workshop on Graph-Theoretic Concepts in Computer Science, Cadenabbia, Italy, Lecture Notes in Computer Science, vol. 1197, June 1996, pp. 293–307.

[KR93] P. Klein and R. Ravi, *A nearly best-possible approximation for node-weighted Steiner trees*, Proceedings of the 3rd MPS conference on Integer Programming and Combinatorial Optimization, 1993, pp. 323–332.

[Phi93] C. Phillips, *The network inhibition problem*, Proceedings of the 25th Annual ACM Symposium on the Theory of Computing (STOC'93), May 1993, pp. 288–293.

[PS95] D. Paik and S. Sahni, *Network upgrading problems*, Networks **26** (1995), 45–58.

Sven O. Krumke, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Department Optimization, Takustr. 7, 14195 Berlin-Dahlem, Germany.
E-mail address: `krumke@zib.de`

Madhav V. Marathe, Los Alamos National Laboratory, P.O.Box 1663, MS B265, Los Alamos NM 87545, USA
E-mail address: `madhav@c3.lanl.gov`

Hartmut Noltemeier, Department of Computer Science, University of Würzburg, Am Hubland, 97074 Würzburg, Germany.
E-mail address: `noltemei@informatik.uni-wuerzburg.de`

S. S. Ravi, Department of Computer Science, University at Albany – SUNY, Albany, NY 12222, USA.
E-mail address: `ravi@cs.albany.edu`

Hans-Christoph Wirth, Department of Computer Science, University of Würzburg, Am Hubland, 97074 Würzburg, Germany
E-mail address: `wirth@informatik.uni-wuerzburg.de`