Improved Approximation for Single-Sink Buy-at-Bulk[⋆]

Fabrizio Grandoni[1] and Giuseppe F. Italiano[2]

[1] Dipartimento di Informatica, Università di Roma "La Sapienza", Via Salaria 113, 00198 Roma, Italy. grandoni@di.uniroma1.it
[2] Dipartimento di Informatica, Sistemi e Produzione, Università di Roma "Tor Vergata", Via del Politecnico 1, 00133 Roma, Italy. italiano@disp.uniroma2.it

Abstract. In the single-sink buy-at-bulk network design problem we are given a subset of source nodes in a weighted undirected graph: each source node wishes to send a given amount of flow to a sink node. Moreover, a set of cable types is given, each characterized by a cost per unit length and by a capacity: the ratio cost/capacity decreases from small to large cables by economies of scale. The problem is to install cables on edges at minimum cost, such that the flow from each source to the sink can be routed simultaneously. The approximation ratio of this NP-hard problem was gradually reduced from $O(\log^2 n)$ to 65.49 by a long series of papers. In this paper, we design a better 24.92 approximation algorithm for this problem.

1 Introduction

Consider the problem of connecting different sites with an optical network. We know the distance and the traffic demand between each pair of sites. We are allowed to install optical cables, chosen from a limited set of available cable types: each cable type is characterized by a capacity and by a cost per unit length. By *economies of scale*, the cost per unit capacity decreases from small to large cables. The same kind of problem arises in several other applications, where optical cables are replaced by, e.g., pipes, trucks, and so on.

The essence of the mentioned problems is captured by *Multi-Sink Buy-at-Bulk* (MSBB) network design. In the MSBB we are given an n-node undirected graph $G = (V, E)$, with nonnegative edge lengths $c(e)$, $e \in E$. We distinguish a subset $P = \{(s_1, r_1), (s_2, r_2), \ldots, (s_p, r_p)\}$ of source-sink pairs: source s_i wishes to send $d(s_i)$ units of flow (the *demand* of s_i) to sink r_i. In order to support such flow, we are allowed to install *cables* on edges. There are k different *cable types* $1, 2, \ldots, k$. Cables of type i have capacity μ_i and cost σ_i per unit length (that is, the cost of installing one such cable on edge e is $c(e)\sigma_i$). The cables satisfy

⋆ This work has been partially supported by the Sixth Framework Programme of the EU under Contract Number 507613 (Network of Excellence "EuroNGI: Designing and Engineering of the Next Generation Internet") and by MIUR, under Project ALGO-NEXT.

the economies of scale principle: the cost $\delta_i = \sigma_i/\mu_i$ per unit capacity and unit length decreases from small to large cables. The aim is to find a minimum-cost installation of cables such that the flow between each source-sink pair can be routed simultaneously. In this paper we are concerned with the *Single-Sink* version of this problem (SSBB), where all the sources s_i send their flow to a unique sink r. The problem remains NP-hard also in this case (e.g., by reduction from the Steiner tree problem). The SSBB problem has been extensively studied in the literature. Meyerson, Munagala, and Plotkin [18] gave a $O(\log n)$ approximation. Garg et al. [9] described a $O(k)$ approximation, where k is the number of cable types. The first constant approximation is due to Guha, Meyerson, and Munagala [10]: the approximation ratio of their algorithm is roughly 2000. This approximation was reduced to 216 by Talwar [19], and later to 76.8 by Gupta, Kumar, and Roughgarden [12,15]. Using the same approach as Gupta et al., finally Jothi and Raghavachari [16] reduced the approximation factor to 65.49.

The contribution of this paper is a better approximation bound of 24.92 for the SSBB problem. Our improved bound is surprisingly obtained by a simple variant of the algorithm of Gupta et al. [12,15], combined with a more careful analysis. The algorithm by Gupta et al. works in two phases. First there is a *preprocessing phase*, where costs are rounded up and capacities are rounded down to the closest power of two. Part of the new cable types obtained in this way are "redundant" according to the new costs and capacities, and thus they can be safely discarded. Let $i(1)$, $i(2)$, $\ldots$, $i(k')$ be the remaining cable types, in increasing order of capacity. The second phase consists of a sequence of suitably defined *aggregation rounds*. In each round the demand is aggregated into a smaller, randomly selected subset of nodes, until all the demand is routed to the sink. In round t, only cables of type $i(t)$ and $i(t+1)$ are used. The initial rounding of this algorithm is responsible for a factor 4 in the approximation. Thus, it seems natural to wonder whether it is possible to improve substantially the approximation factor by replacing the first phase with a more sophisticated choice of the cable types to be used in the second phase (while preserving the same basic aggregation scheme). In this paper we present a simple, non-trivial cable selection rule which, in combination with a more careful analysis, reduces to 24.92 the approximation ratio for SSBB.

Related Work. Awerbuch and Azar [1] gave a $O(\log^2 n)$ approximation for MSBB, based on the tree embedding techniques by Bartal [2]. The improved tree embeddings in [3,5,8] lead to a $O(\log n)$ approximation. To the best of our knowledge, no constant approximation for MSBB is currently known. A problem closely related to MSBB is *Multi-sink Rent-Or-Buy* (MROB) network design [4,10,11,13,14,15,17]. As in MSBB, there is a set of source-sink pairs that wish to communicate. Now, instead of installing cables, we can either *buy* or *rent* edges: if we buy one edge, we pay a fixed cost c_{buy} per unit length, and we are then free to route an unbounded amount of flow on the bought edge. If we rent it, we pay a cost c_{rent} per unit length and unit flow along the edge. The current best approximation for MROB is 6.828 in the multi-sink case [4] and 3.55 in the single-sink case [15]. Another related problem (from the point of view of the techniques used to solve it) is *Virtual Private Network Design* (VPND) [6,7,15]. Here we have a set of terminals which wish to send flow to each other, but the traffic matrix is not know a priori: only upper bounds are given on the total amount of (unsplittable) flow that each terminal can send and receive. The aim is to find a minimum cost capacity reservation which supports every feasible traffic matrix. The current best approximation for VPND is 3.55 [7].

Preliminaries. For the sake of simplicity, in this extended abstract we assume that capacities, costs, and demands are non-negative integers. The same results can be extended to the case of real values. Let $1, 2, \ldots, k$ be the set of cable types, in increasing order of capacity: $\mu_1 \leq \mu_2, \ldots, \leq \mu_k$. Recall that $\delta_1 \geq \delta_2, \ldots, \geq \delta_k$ by economies of scale. Note that we can assume $\sigma_1 < \sigma_2, \ldots, < \sigma_k$. In fact, if $\sigma_i \geq \sigma_j$, for some $i < j$, we can eliminate the cable type i (without modifying the optimum). Following [15], and without loss of generality, we assume each node $v \in V$ has a demand $d(v)$, which is either zero or one. This can be achieved by duplicating nodes. The algorithm presented can be easily adapted so as to run in polynomial time even when the (original) demands are not polynomially bounded. The algorithm by Gupta et al. [15] is designed for capacities which are powers of two. Jothi and Raghavachari [16] designed a somehow complicated generalization of the algorithm in [15], in order to handle capacities which are powers of $(1 + \epsilon)$. Here we describe a simpler and more natural generalization of the algorithm in [15], which works for any value of the capacities. Our generalization is based on the following simple assumption: the sum of the demands $\sum_{v \in V} d(v)$ is a multiple of each capacity μ_i. This property can be enforced by adding *dummy demands* in the sink. By OPT we denote either the optimum solution or its actual value, where the meaning will be clear from the context. $OPT(s)$ is the cost paid by OPT to install cables of type s.

The remainder of this paper is organized as follows. In Section 2 we describe a generalization of the algorithm by Gupta et al., and analyze it under a generic cable selection paradigm. In Section 3 we introduce within this framework a more sophisticated cable selection rule, and prove that this yields the claimed 24.92 approximation bound for SSBB.

2 The Algorithm

One of the key steps in the approach of Gupta et al. [12,15] is aggregating demands over a tree in multiples of a given quantity. More precisely, consider a tree T and a given integer $U > 0$. Suppose each node v of T has integer weight $x(v) \in [0, U)$, and the sum of the weights is a multiple of U. They need to compute a flow moving weights along the tree such that: (1) The amount of flow along each edge is at most U, (2) The new weight $x'(v)$ at each node is either 0 or U, and (3) The expected weight at each node is preserved, that is: $Pr[x'(v) = U] = x(v)/U$. Gupta et al. give a randomized *aggregation algorithm* for this problem, which we describe next from a slightly different perspective. Replace each edge of T with two oppositely directed edges. Compute an Euler tour C' in the resulting directed graph T'. The same node v may appear several times in C': in that case assign the weight $x(v)$ to one of the occurrences of v, and zero to the others. Then replace each node with a path of length equal to its weight minus one (if the weight of a node is zero, remove the node and add one edge between its two neighbors). Now select a random subset of nodes in the resulting cycle $C = (w_0, w_1, \ldots, w_{|C|-1})$, such that the distance (number of hops) between any two consecutive selected nodes is U. This is possible since the total weight, which is equal to the total number of nodes of the cycle C, is a multiple of U by assumption. Eventually each node sends one unit of flow to the closest selected node in, say, clockwise direction. In particular, each selected node receives exactly $(U-1)$ units of flow. The flow along C naturally induces a flow in the original graph. It is worth to mention that the duplication of nodes is not really necessary, but it is introduced here for the sake of simplicity.

We are now ready to describe our `SSBB` algorithm. We initially select a subset of cable types $i(1), i(2), \ldots, i(k')$ in increasing order of capacity, where we require that $i(1) = 1$ and $i(k') = k$ (that is, the first and last cable types are always selected). The selection rule will be described in Section 3. Note that there is no initial rounding. Then there is a sequence of rounds. In each round the demand is aggregated in a smaller and smaller randomly selected subset of nodes, until it is eventually routed to the sink. For ease of presentation, we distinguish the *initial* and *final* rounds from the remaining *intermediate* rounds. Let D_0 be the nodes with unit input demand. In the *initial round* we compute a ρ_{st}-approximate Steiner tree T_0 over $\{r\} \cup D_0$, and we apply the aggregation algorithm to T_0 with capacity $U = \mu_1$ and weights $x(v) = d(v)$ for each node v of T_0 (this is possible since by assumption the sum of the demands is a multiple of μ_1). The corresponding flow is supported by installing cables of type 1 (at most one on each edge of T_0). At the end of the round the demand at each node is either zero or μ_1. Now consider an *intermediate round* t, $t \in \{1, 2, \ldots, k'-1\}$. By induction on the number of rounds, the initial demand $d_t(v)$ of node v is either zero or $\mu_{i(t)}$, while its final demand $d_{t+1}(v)$ is either zero or $\mu_{i(t+1)}$. The round consists of three steps. Initially the demand is collected at a random subset of aggregation points. Then a Steiner tree is computed on the aggregation points, and the demand is aggregated along such tree with the aggregation algorithm. Eventually the aggregated demand is redistributed back to the source nodes. Only cables of type $i(t)$ and $i(t+1)$ are used in this process. We now describe the steps in more details. Let D_t denote the set of nodes with $d_t(v) = \mu_{i(t)}$.

Collection step : Each node in D_t is marked with probability $\sigma_{i(t)}/\sigma_{i(t+1)}$. Let D'_t be the set of marked nodes. Each node sends its demand to the closest node in $\{r\} \cup D'_t$ along a shortest path, using cables of type $i(t)$. Let $d'_t(w)$ be the new demand collected at each $w \in \{r\} \cup D'_t$.

Aggregation step : Compute a ρ_{st}-approximate Steiner tree T_t on $\{r\} \cup D'_t$. Apply the aggregation algorithm to T_t with $U = \mu_{i(t+1)}$ and weight $x(w) = d'_t(w)$ (mod $\mu_{i(t+1)}$) for each terminal node w (this is possible since the sum of the $d'_t(w)$, and hence of the $x(w)$, is a multiple of $\mu_{i(t+1)}$). The corresponding flow is supported by installing cables of type $i(t+1)$ (at most one for each edge of T_t). Let $d''_t(w)$ be the new demand aggregated at each node w.

Redistribution step : For each node $w \in \{r\} \cup D'_t$, consider the subset of nodes $D_t(w) \subseteq D_t$ that sent their demand to w during the collection step (including w itself, if $w \neq r$). Uniformly select a random subset $\widetilde{D}_t(w)$ of $D_t(w)$ of cardinality $d''_t(w)/\mu_{i(t+1)}$. Send $\mu_{i(t+1)}$ units of flow back from w to each node in $\widetilde{D}_t(v)$ along shortest paths, installing cables of type $i(t+1)$.

Note that no demand is routed to the sink during the initial and intermediate rounds. The algorithm ends with a *final round*, where all the demands are sent to the sink along shortest paths, using cables of type $i(k') = k$. A generalization of the analysis given in [12,15] yields the following result, whose proof is omitted here for lack of space.

Lemma 1. *The* SSBB *algorithm above computes a solution of cost* $APX \leq \sum_{s=1}^{k} apx(s)\, OPT(s)$ *where*

$$apx(s) := 1+\rho_{st}\frac{\sigma_{i(1)}}{\sigma_s}+\sum_{t=1}^{k'-1}\left(\left(2+2\frac{\delta_{i(t+1)}}{\delta_{i(t)}}\right)\left(1-\frac{\sigma_{i(t)}}{\sigma_{i(t+1)}}\right)+\rho_{st}\right)\min\left\{\frac{\sigma_{i(t+1)}}{\sigma_s},\,\frac{\delta_{i(t)}}{\delta_s}\right\}. \tag{1}$$

3 An Improved Cable-Selection Rule

Let $i(1)$, $i(2)$, ..., $i(k')$ be the cable types, in increasing order of capacity, left after the first phase of the algorithm by Gupta et al. Such cables have the property that the σ's double and the δ's halve from one cable to the next one:

$$\forall\, t \in \{1,2,\ldots,k'-1\}, \quad \sigma_{i(t+1)} \geq 2\sigma_{i(t)} \quad \text{and} \quad \delta_{i(t+1)} \leq \delta_{i(t)}/2.$$

Recall that all the remaining cables are used in the second phase. Thus, by Lemma 1, for every s,

$$apx(s) \leq 1+(2+1+\rho_{st})\left(\sum_{i\geq 0}\frac{1}{2^i}+\sum_{j\geq 0}\frac{1}{2^j}\right) = 1+(3+\rho_{st})4.$$

Unfortunately, the initial rounding introduces an extra factor 4 in the approximation, thus leading to an overall $4(1+(3+\rho_{st})4) < 76.8$ approximation.

It is then natural to wonder whether it is possible to keep $apx(s)$ small, while avoiding rounding, by means of a more sophisticated cable selection rule. An intuitive approach could be selecting cables (in the original problem) in the following way: for a given selected cable type $i(t)$, starting from $i(1) = 1$, $i(t+1)$ is the smallest cable type such that $\sigma_{i(t+1)} \geq 2\sigma_{i(t)}$ and $\delta_{i(t+1)} \leq \delta_{i(t)}/2$. This way, we maintain the good *scaling* properties of σ's and δ's of selected cables. In particular, for any selected cable type $s = i(t')$, $apx(s) \leq 1+(3+\rho_{st})4$. Unluckily, this approach does not work for discarded cable types s, $i(t') < s < i(t'+1)$: in fact, in this case the *intermediate* term $(\min\{\sigma_{i(t'+1)}/\sigma_s,\, \delta_{i(t')}/\delta_s\})$ can be arbitrarily large. What can we do then? There is a surprisingly simple approach to tackle this problem. The idea is to slightly relax the scaling properties of the σ's: instead of requiring that $\sigma_{i(t+1)} \geq 2\sigma_{i(t)}$, we only require that $\sigma_{i(t+1)+1} \geq 2\sigma_{i(t)}$. More precisely, we use the following cable selection rule:

Improved cable selection rule: *Let* $i(1) = 1$. *Given* $i(t)$, $1 < i(t) < k$, $i(t+1)$ *is the smallest index such that*

$$\sigma_{i(t+1)+1} \geq 2\sigma_{i(t)} \quad \textit{and} \quad \delta_{i(t+1)} \leq \delta_{i(t)}/2.$$

If such index does not exist, $i(t+1) = i(k') = k$.

Observe that the δ's halve at each selected cable (excluding possibly the last one), and the σ's double every other selected cable:

$$\forall\, t \in \{1, 2, \ldots, k'-2\}, \quad \delta_{i(t+1)} \leq \delta_{i(t)}/2 \quad \text{and} \quad \sigma_{i(t+2)} \geq \sigma_{i(t+1)+1} \geq 2\sigma_{i(t)}. \quad (2)$$

With this cable-selection policy we obtain $apx(s) \leq 1 + (3 + \rho_{st})7 < 32.85$ for every cable type s, including discarded ones. This is also a feasible bound on the overall approximation ratio since we avoided the initial rounding. This analysis can be refined by exploiting the telescopic sum hidden in Equation (1). This refinement improves to $4(1 + (3 + 2\rho_{st}) + (3 + \rho_{st})2) < 64.8$ the approximation bound of the algorithm by Gupta et al., and yields a better $16 + 7\rho_{st} < 26.85$ approximation bound if we use our approach.

Theorem 1. *The algorithm of Section 2, combined with the improved cable selection rule, yields a* $16 + 7\rho_{st} < 26.85$ *approximation bound for* `SSBB`.

Proof. Let us restrict to the case $s \leq i(k'-1)$. The case $i(k'-1) < s \leq i(k')$ is analogous, and thus it is omitted from this extended abstract. We distinguish between selected and discarded cables.

a) Discarded cable s, $\mathbf{i(t') < s < i(t'+1) \leq i(k'-1)}$. By Lemma 1 and Equation (2), and observing that $\sigma_s \geq \sigma_{i(t')+1} \geq 2\sigma_{i(t'-1)}$ (for $t' > 1$), $apx(s)$ is bounded above by

$$1 + \rho_{st}\frac{\sigma_{i(1)}}{\sigma_s} + \sum_{t=1}^{t'-1}\left(3 + \rho_{st} - 3\frac{\sigma_{i(t)}}{\sigma_{i(t+1)}}\right)\frac{\sigma_{i(t+1)}}{\sigma_s}$$
$$+ \sum_{t=t'}^{k'-1}\left(2 + 2\frac{\delta_{i(t+1)}}{\delta_{i(t)}} + \rho_{st}\right)\min\left\{\frac{\sigma_{i(t+1)}}{\sigma_s}, \frac{\delta_{i(t)}}{\delta_s}\right\} \leq$$
$$1 + (3 + \rho_{st})\frac{\sigma_{i(t')}}{\sigma_s} + \rho_{st}\sum_{t=1}^{t'-2}\frac{\sigma_{i(t+1)}}{\sigma_s} + \sum_{t=t'}^{k'-1}\left(2 + 2\frac{\delta_{i(t+1)}}{\delta_{i(t)}} + \rho_{st}\right)\min\left\{\frac{\sigma_{i(t+1)}}{\sigma_s}, \frac{\delta_{i(t)}}{\delta_s}\right\} \leq$$
$$1 + (3 + \rho_{st}) + \rho_{st}\sum_{i=1}^{t'-2}\frac{1}{2^{\lceil i/2\rceil}} + \sum_{t=t'}^{k'-1}\left(2 + 2\frac{\delta_{i(t+1)}}{\delta_{i(t)}} + \rho_{st}\right)\min\left\{\frac{\sigma_{i(t+1)}}{\sigma_s}, \frac{\delta_{i(t)}}{\delta_s}\right\} \leq$$

$$4 + 3\,\rho_{st} + \sum_{t=t'}^{k'-1}\left(2 + 2\frac{\delta_{i(t+1)}}{\delta_{i(t)}} + \rho_{st}\right)\min\left\{\frac{\sigma_{i(t+1)}}{\sigma_s}, \frac{\delta_{i(t)}}{\delta_s}\right\}.$$

From Equation (2), and observing that $\delta_s \geq \delta_{i(t'+1)}$, we get that $apx(s)$ is upper bounded by

$$4 + 3\,\rho_{st} + (3+\rho_{st}) \min\left\{\frac{\sigma_{i(t'+1)}}{\sigma_s}, \frac{\delta_{i(t')}}{\delta_s}\right\} + (3+\rho_{st}) \sum_{t=t'+1}^{k'-2} \frac{\delta_{i(t)}}{\delta_s} + (4+\rho_{st}) \frac{\delta_{i(k'-1)}}{\delta_s} \leq$$

$$4 + 3\,\rho_{st} + (3+\rho_{st}) \min\left\{\frac{\sigma_{i(t'+1)}}{\sigma_s}, \frac{\delta_{i(t')}}{\delta_s}\right\} + (3+\rho_{st}) \sum_{j=0}^{k'-t'-3} \frac{1}{2^j} + (4+\rho_{st}) \frac{1}{2^{k'-t'-2}} \leq$$

$$4 + 3\,\rho_{st} + (3+\rho_{st}) \min\left\{\frac{\sigma_{i(t'+1)}}{\sigma_s}, \frac{\delta_{i(t')}}{\delta_s}\right\} + (3+\rho_{st}) \left(\sum_{j=0}^{k'-t'-3} \frac{1}{2^j} + \frac{1}{2^{k'-t'-3}}\right)$$

Thus we get

$$apx(s) \leq 10 + 5\,\rho_{st} + (3+\rho_{st}) \min\left\{\frac{\sigma_{i(t'+1)}}{\sigma_s}, \frac{\delta_{i(t')}}{\delta_s}\right\}. \tag{3}$$

We next show that

$$\min\left\{\frac{\sigma_{i(t'+1)}}{\sigma_s}, \frac{\delta_{i(t')}}{\delta_s}\right\} \leq 2. \tag{4}$$

Let $j(t')$ be the smallest index such that $\delta_{j(t')}/\delta_{i(t')} \leq 1/2$. Consider the case $s < j(t')$. By the definition of $j(t')$, $\delta_{j(t')-1}/\delta_{i(t')} > 1/2$. Therefore

$$\min\left\{\frac{\sigma_{i(t'+1)}}{\sigma_s}, \frac{\delta_{i(t')}}{\delta_s}\right\} \leq \frac{\delta_{i(t')}}{\delta_s} \leq \frac{\delta_{i(t')}}{\delta_{j(t')-1}} \leq 2.$$

Consider now the case $s \geq j(t') > i(t')$. Observe that $\sigma_{i(t'+1)}/\sigma_{i(t')} < 2$. In fact otherwise we would have

$$\sigma_{i(t'+1)-1+1}/\sigma_{i(t')} \geq 2 \quad \text{and} \quad \delta_{i(t'+1)-1}/\delta_{i(t')} \leq \delta_{j(t')}/\delta_{i(t')} \leq 1/2.$$

Thus cable $i(t'+1)-1$ should be selected, which contradicts the fact that $i(t'+1)$ is the first cable selected after $i(t')$. As a consequence

$$\min\left\{\frac{\sigma_{i(t'+1)}}{\sigma_s}, \frac{\delta_{i(t')}}{\delta_s}\right\} \leq \frac{\sigma_{i(t'+1)}}{\sigma_s} \leq \frac{\sigma_{i(t'+1)}}{\sigma_{i(t')}} \leq 2.$$

From (3) and (4),

$$apx(s) \leq 10 + 5\,\rho_{st} + (3+\rho_{st})\,2 = 16 + 7\rho_{st}. \tag{5}$$

b) Selected cable s, $\mathbf{s = i(t') \leq i(k'-1)}$. By basically the same arguments as for the case of discarded cables,

$$\begin{aligned}
apx(s) &\leq 1 + (3+\rho_{st}) \frac{\sigma_{i(t')}}{\sigma_{i(t')}} + \rho_{st} \sum_{t=1}^{t'-2} \frac{\sigma_{i(t+1)}}{\sigma_{i(t')}} + (3+\rho_{st}) \sum_{t=t'}^{k'-2} \frac{\delta_{i(t)}}{\delta_{i(t')}} + (4+\rho_{st}) \frac{\delta_{i(k'-1)}}{\delta_{i(t')}} \\
&\leq 1 + (3+\rho_{st}) + \rho_{st} \sum_{i=1}^{t'-2} \frac{1}{2^{\lfloor i/2 \rfloor}} + (3+\rho_{st}) \left(\sum_{j=0}^{k'-t'-2} \frac{1}{2^j} + \frac{1}{2^{k'-t'-2}}\right) \\
&\leq 1 + (3+\rho_{st}) + 3\rho_{st} + (3+\rho_{st})\,2 \\
&= 10 + 6\,\rho_{st}.
\end{aligned} \tag{6}$$

By (5) and (6)

$$APX \le \sum_{s=1}^{k} apx(s)\, OPT(s) \le (16 + 7\rho_{st}) OPT.$$

□

Remark 1 *In order to prove* (4) *the "relaxed" condition* $\sigma_{i(t+1)+1} \ge 2\sigma_{i(t)}$ *is crucial. The "naive" condition* $\sigma_{i(t+1)} \ge 2\sigma_{i(t)}$ *would not work properly.*

3.1 Adapting the Scaling Factors

The approximation can be further reduced to 24.92 by using better scaling factors. Let $\alpha > 1$ and $\beta > 1$ be two real parameters to be fixed later. Consider the following generalization of the improved cable selection rule:

Generalized cable selection rule: *Let* $i(1) = 1$. *Given* $i(t)$, $1 < i(t) < k$, *index* $i(t+1)$ *is the smallest index such that*

$$\sigma_{i(t+1)+1} \ge \alpha\, \sigma_{i(t)} \quad \textit{and} \quad \delta_{i(t+1)} \le \delta_{i(t)}/\beta.$$

If such index does not exist, $i(t+1) = i(k') = k$.

A proper choice of α and β leads to the following slightly refined approximation.

Theorem 2. *There is a* 24.92 *approximation algorithm for* `SSBB`.

Proof. Consider the algorithm of Section 2, with the generalized cable selection rule. For the sake of simplicity, let us assume $\beta \le 3.77$, from which

$$(4 + \rho_{st}) \le \left(2 + \frac{2}{\beta} + \rho_{st}\right) \frac{\beta}{\beta - 1}.$$

Consider first the case $i(t') < s < i(t'+1) \le i(k'-1)$. By basically the same arguments as in the proof of Theorem 1, either $\delta_{i(t')}/\delta_s < \beta$ or $\sigma_{i(t'+1)}/\sigma_s < \alpha$. In the first case

$$\begin{aligned} apx(s) \le\ & 1 + \left(2 + \frac{2}{\beta} + \rho_{st}\right) \frac{\sigma_{i(t')}}{\sigma_s} + \rho_{st} \sum_{t=1}^{t'-2} \frac{\sigma_{i(t+1)}}{\sigma_s} + \left(2 + \frac{2}{\beta} + \rho_{st}\right) \left(\beta + \sum_{t=t'+1}^{k'-2} \frac{\delta_{i(t)}}{\delta_s}\right) \\ & + (4 + \rho_{st}) \frac{\delta_{i(k'-1)}}{\delta_s} \\ \le\ & 1 + \left(2 + \frac{2}{\beta} + \rho_{st}\right) + \rho_{st} \sum_{i=1}^{t'-2} \frac{1}{\alpha^{\lceil i/2 \rceil}} + \left(2 + \frac{2}{\beta} + \rho_{st}\right) \beta \\ & + \left(2 + \frac{2}{\beta} + \rho_{st}\right) \sum_{j=0}^{k'-t'-3} \frac{1}{\beta^j} + \frac{4 + \rho_{st}}{\beta^{k'-t'-2}} \\ \le\ & 1 + \left(2 + \frac{2}{\beta} + \rho_{st}\right) + \frac{2\rho_{st}}{\alpha - 1} + \left(2 + \frac{2}{\beta} + \rho_{st}\right) \beta + \left(2 + \frac{2}{\beta} + \rho_{st}\right) \frac{\beta}{\beta - 1}. \end{aligned} \tag{7}$$

In the second case,

$$apx(s) \leq 1 + \left(2 + \frac{2}{\beta} + \rho_{st}\right) \frac{\sigma_{i(t'+1)}}{\sigma_s} + \rho_{st} \sum_{t=1}^{t'-1} \frac{\sigma_{i(t+1)}}{\sigma_s} + \left(2 + \frac{2}{\beta} + \rho_{st}\right) \sum_{t=t'+1}^{k'-2} \frac{\delta_{i(t)}}{\delta_s}$$
$$+ (4 + \rho_{st}) \frac{\delta_{i(k'-1)}}{\delta_s}$$
$$\leq 1 + \left(2 + \frac{2}{\beta} + \rho_{st}\right) \alpha + \rho_{st} \sum_{i=1}^{t'-1} \frac{1}{\alpha^{\lfloor i/2 \rfloor}} + \left(2 + \frac{2}{\beta} + \rho_{st}\right) \frac{\beta}{\beta - 1}$$
$$\leq 1 + \left(2 + \frac{2}{\beta} + \rho_{st}\right) \alpha + \rho_{st} \frac{\alpha + 1}{\alpha - 1} + \left(2 + \frac{2}{\beta} + \rho_{st}\right) \frac{\beta}{\beta - 1}. \quad (8)$$

For any selected cable type $s = i(t') \leq i(k' - 1)$,

$$apx(s) \leq 1 + \left(2 + \frac{2}{\beta} + \rho_{st}\right) \frac{\sigma_{i(t')}}{\sigma_{i(t')}} + \rho_{st} \sum_{t=1}^{t'-2} \frac{\sigma_{i(t+1)}}{\sigma_s} + \left(2 + \frac{2}{\beta} + \rho_{st}\right) \sum_{t=t'}^{k'-2} \frac{\delta_{i(t)}}{\delta_s}$$
$$+ (4 + \rho_{st}) \frac{\delta_{i(k'-1)}}{\delta_s}$$
$$\leq 1 + \left(2 + \frac{2}{\beta} + \rho_{st}\right) + \rho_{st} \sum_{i=1}^{t'-2} \frac{1}{\alpha^{\lfloor i/2 \rfloor}} + \left(2 + \frac{2}{\beta} + \rho_{st}\right) \sum_{j=0}^{k'-t'-2} \frac{1}{\beta^j} + \frac{4 + \rho_{st}}{\beta^{k'-t'-1}}$$
$$\leq 1 + \left(2 + \frac{2}{\beta} + \rho_{st}\right) + \rho_{st} \frac{\alpha + 1}{\alpha - 1} + \left(2 + \frac{2}{\beta} + \rho_{st}\right) \frac{\beta}{\beta - 1}. \quad (9)$$

In the case $i(k' - 1) < s \leq i(k') = k$ one obtains similarly:

$$apx(s) \leq 1 + \left(2 + \frac{2}{\beta} + \rho_{st}\right) + \frac{2\rho_{st}}{\alpha - 1} + (4 + \rho_{st})\beta, \quad (10)$$
$$apx(s) \leq 1 + (4 + \rho_{st})\alpha + \rho_{st} \frac{\alpha + 1}{\alpha - 1}, \quad (11)$$
$$apx(s) \leq 1 + (4 + \rho_{st}) + \rho_{st} \frac{\alpha + 1}{\alpha - 1}. \quad (12)$$

For a given choice of α and β, the approximation ratio is the maximum over (7)-(12). In particular, for $\alpha = \beta = 2$ we obtain the result of Theorem 1. The claim follows by imposing $\alpha = 3.1207$ and $\beta = 2.4764$. □

Acknowledgments A special thank to Jochen Könemann for carefully reading a preliminary version of this paper and for helpful discussions.

References

1. B. Awerbuch and Y. Azar. Buy-at-bulk network design. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 542–547, 1997.
2. Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 184–193, 1996.
3. Y. Bartal. On approximating arbitrary metrics by tree metrics. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 161–168, 1998.

4. L. Becchetti, J. Konemann, S. Leonardi, and M. Pal. Sharing the cost more efficiently: improved approximation for multicommodity rent-or-buy. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 375–384, 2005.
5. M. Charikar, A. Chekuri, A. Goel, and S. Guha. Approximating a finite metric by a small number of tree metrics. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 379–388, 1998.
6. F. Eisenbrand and F. Grandoni. An improved approximation algorithm for virtual private network design. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 928–932, 2005.
7. F. Eisenbrand, F. Grandoni, G. Oriolo, and M. Skutella. New approaches for virtual private network design. In *International Colloquium on Automata, Languages and Programming (ICALP)*, pages 1152–1162, 2005.
8. J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *ACM Symposium on the Theory of Computing (STOC)*, pages 448–455, 2003.
9. N. Garg, R. Khandekar, G. Konjevod, R. Ravi, F. Salman, and A. Sinha. On the integrality gap of a natural formulation of the single-sink buy-at-bulk network design problem. In *International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 170–184, 2001.
10. S. Guha, A. Meyerson, and K. Munagala. A constant factor approximation for the single sink edge installation problem. In *ACM Symposium on the Theory of Computing (STOC)*, pages 383–388, 2001.
11. A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network: a network design problem for multicommodity flow. In *ACM Symposium on the Theory of Computing (STOC)*, pages 389–398, 2001.
12. A. Gupta, A. Kumar, M. Pal, and T. Roughgarden. Approximation via cost-sharing: simpler and better approximation algorithms for network design. Manuscript.
13. A. Gupta, A. Kumar, M. Pal, and T. Roughgarden. Approximation via cost-sharing: a simple approximation algorithm for the multicommodity rent-or-buy problem. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 606–617, 2003.
14. A. Gupta, A. Kumar, and T. Roughgarden. A constant-factor approximation algorithm for the multicommodity. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 333–344, 2002.
15. A. Gupta, A. Kumar, and T. Roughgarden. Simpler and better approximation algorithms for network design. In *ACM Symposium on the Theory of Computing (STOC)*, pages 365–372, 2003.
16. R. Jothi and B. Raghavachari. Improved approximation algorithms for the single-sink buy-at-bulk network design problems. In *Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 336–348, 2004.
17. A. Kumar and C. Swamy. Primal-dual algorithms for the connected facility location problem. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 256–269, 2002.
18. A. Meyerson, K. Munagala, and S. Plotkin. Cost-distance: two metric network design. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 624–630, 2000.
19. K. Talwar. The single-sink buy-at-bulk LP has constant integrality gap. In *International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 475–486, 2002.