Improved Approximations for Buy-at-bulk and Shallow-Light k-Steiner Trees and $(k, 2)$-Subgraph

M. Reza khani[*] Mohammad R. Salavatipour[†]

Abstract

In this paper we give improved approximation algorithms for some network design problems. In the Bounded-Diameter or Shallow-Light k-Steiner tree problem (SLkST), we are given an undirected graph $G = (V, E)$ with terminals $T \subseteq V$ containing a root $r \in T$, a cost function $c : E \to \mathbb{R}^+$, a length function $\ell : E \to \mathbb{R}^+$, a bound $L > 0$ and an integer $k \geq 1$. The goal is to find a minimum c-cost r-rooted Steinter tree containing at least k terminals whose diameter under ℓ metric is at most L. The input to the Buy-at-Bulk k-Steiner tree problem (BBkST) is similar: graph $G = (V, E)$, terminals $T \subseteq V$, cost and length functions $c, \ell : E \to \mathbb{R}^+$, and an integer $k \geq 1$. The goal is to find a minimum total cost r-rooted Steiner tree H containing at least k terminals, where the cost of each edge e is $c(e) + \ell(e) \cdot f(e)$ where $f(e)$ denotes the number of terminals whose path to root in H contains edge e. We present a bicriteria $(O(\log^2 n), O(\log n))$-approximation for SLkST: the algorithm finds a k-Steiner tree of diameter at most $O(L \cdot \log n)$ whose cost is at most $O(\log^2 n \cdot \text{OPT}^*)$ where OPT^* is the cost of an LP relaxation of the problem. This improves on the algorithm of [25] (APPROX'06/Algorithmica'09) which had ratio $(O(\log^4 n), O(\log^2 n))$. Using this, we obtain an $O(\log^3 n)$-approximation for BBkST, which improves upon the $O(\log^4 n)$-approximation of [25]. We also consider the problem of finding a minimum cost 2-edge-connected subgraph with at least k vertices, which is introduced as the $(k, 2)$-subgraph problem in [32] (STOC'07/SICOMP09). This generalizes some well-studied classical problems such as the k-MST and the minimum cost 2-edge-connected subgraph problems. We give an $O(\log n)$-approximation algorithm for this problem which improves upon the $O(\log^2 n)$-approximation of [32]

1 Introduction

We consider some network design problems where in each one we are given an undirected graph $G = (V, E)$ with a terminal set $T \subseteq V$ (including a node $r \in T$ called root) and some cost functions defined on the edges, plus an integer $k \geq 1$. The goal is to find a subgraph satisfying certain properties with minimum cost which contains at least k terminals. Below, we describe each of these problems in details.

Bounded Diameter or Shallow-Light Steiner Tree and k-Steiner Tree: Suppose we are given an undirected graph $G = (V, E)$, a cost function $c : E \to \mathbb{R}^+$, a length function $\ell : E \to \mathbb{R}^+$, a subset $T \subseteq V$ called terminals which includes a root node r, and a positive bound L. The goal is to find a Steiner tree over terminals T and rooted at r such that the cost of the tree (under c metric)

[*] Department of Computing Science,University of Alberta, Edmonton, Alberta T6G 2E8, Canada. e-mail: khani@ualberta.ca

[†] Toyota Tech. Inst. at Chicago, and Department of Computing Science, University of Alberta, Edmonton, Alberta T6G 2E8, Canada. e-mail: and mreza@cs.ualberta.ca. Supported by NSERC and an Alberta Ingenuity New Faculty award.

is minimized while the diameter of the tree (under ℓ metric) is at most L. This problem is referred to as Bounded Diameter (BDST) or Shallow-Light Steiner Tree (SLST). In a slightly more general setting, in the input we also have an integer $k \geq 1$ and a feasible solution is an r-rooted Steiner tree containing at least k terminals. We refer to this as Shallow-Light k-Steiner Tree (SLkST).

Another closely related class of network design problems are buy-at-bulk network design problems defined below.

Buy-at-Bulk Steiner Tree (BBST) and k-Steiner Tree (BBkST): Suppose we are given an undirected graph $G = (V, E)$, a set of terminals $T \subseteq V$ including root r, a sub-additive monotone non-decreasing cost function $f_e : \mathbb{R}^+ \to \mathbb{R}^+$ for each edge e, and positive real demand values $\{\delta_i\}_i$, one for each $t_i \in T$. In the BBST problem the goal is to find an r-rooted Steiner tree to route the demands from terminals to root which minimizes the sum of cost of the edges, where the cost of each edge e is $f_e(\delta(e))$ where $\delta(e)$ is the total demand routed over edge e. This is also refered to as single-sink buy-at-bulk problem. Similar to SLkST, one can generalize the BBST problem by having an extra parameter $k \geq 1$ in the input and a feasible solution is an r-rooted Steiner tree which contains at least k terminals (instead of all of the terminals). This way, we obtain the Buy-at-Bulk k-Steiner Tree (BBkST) problem. It can be shown that the definition of buy-at-bulk problems given above is equivalent (with a small constant factor loss in approximation factor) to the following variation which is also called cost-distance. The input is the same except that instead of function f_e for every edge e, we have two metric functions on the edges: $c : E \to \mathbb{R}^+$ is called cost and $\ell : E \to \mathbb{R}^+$ is called length. The cost of a feasible solution H is defined as: $\sum_{e \in H} c(e) + \sum_i \delta_i \cdot L(t_i)$, where $L(t_i)$ is the length (w.r.t ℓ) of the r, t_i-path in H. It is easy to see that this formulation is a special case of buy-at-bulk since a linear function (defined based on c and ℓ) is also sub-additive. It turns out that an α-approximation for the cost-distance version implies a $(2\alpha + \epsilon)$-approximation algorithm for the buy-at-bulk version too (see [2, 14, 34]). For simplicity, we focus on the two cost function (cost+distance) formulation of buy-at-bulk from now on.

Network optimization problems with multiple cost functions, such as buy-at-bulk network design problems, have been studied extensively because of their applications. These problems can model, among others, situations where every edge e (link) can be either purchased at a fixed price $c(e)$ or rented at a price $r(e)$ per amount of flow (or load). The selected edges are required to provide certain bandwidth to satisfy certain demands between nodes of the graph. So if an edge is rented and there is a flow of $f(e)$ on that edge the cost for that edge will be $r(e) \cdot f(e)$ whereas if the edge is purchased, the cost will be $c(e)$ regardless of the flow. It can be shown that this problem and some other variations can be modeled using buy-at-bulk network design defined above (see [25]). Buy-at-bulk problems and their special cases have been studied through a long line of papers in the operation research and computer science communities after the problem was introduced by Salman et al. [37] (see e.g. [2, 3, 4, 11, 14, 21, 23, 24, 25, 30, 31, 34]).

Another major line of research in network design problems has focused on problems with connectivity requirements where one has another parameter k, and the goal is to find a subgraph satisfying the connectivity requirements with a lower bound k on the total number of vertices. The most well-studied problem in this class is the minimum k-spanning tree problem, a.k.a. k-MST. The approximation factor for this problem was improved from $\sqrt{k}$ [35] to 2 [20] in a series of papers. A very natural common generalization of both the k-MST problem and the minimum cost λ-edge-connected spanning subgraph problem is the (k, λ)-subgraph problem inroduced in [32]. In this paper we focus on the case of $\lambda = 2$:

$(k, 2)$**-Subgraph Problem:** In the (k, λ)-subgraph problem, we are given a (multi)graph $G = (V, E)$ with a cost function $c : E \to \mathbb{R}^+$, and a positive integer k. The goal is to find a minimum cost λ-edge-connected subgraph containing at least k vertices.

We should point out that the cost function c is arbitrary (i.e. does not necessarily satisfy the triangle inequality). Furthermore, we are not allowed to take more copies of an edge that present in the graph. In particular, if G is a simple graph the solution must be simple too. The (k, λ)-subgraph problem contains some classical problems as special cases. For example, $(k, 1)$-subgraph problem is the k-Minimum Spanning Tree problem (k-MST) and $(|V|, \lambda)$-subgraph is simply asking for a minimum cost λ-edge-connected spanning subgraph. It was proved in [32] that the minimum densest k-subgraph problem has a poly-logarithmic reduction to the (k, λ)-subgraph problem. Since the densest k-subgraph has proved to be an extremely difficult problem (the best approximation algorithm for it has ratio $O(n^{\frac{1}{4}})$ [9]), this shows that for general λ, (k, λ)-subgraph problem is a very hard problem too.

Related Work: In the multi-commodity buy-at-bulk problem (which is a generalization of BBST) we are given p source-sink pairs of terminals $\{s_i, t_i\}_{i=1}^{p}$ each with a demand δ_i. A subset of edges E' is feasible if for every $1 \leq i \leq p$ there is a s_i, t_i-path in $G' = (V, E')$. The goal is to minimize $\sum_{e \in E'} c(e) + \sum_i \delta_i \cdot \text{dist}_{G'}(s_i, t_i)$ where the distance is with respect to length function ℓ. In the uniform version of buy-at-bulk all the values along the edges are the same, i.e. $c(e) = c(e')$ and $\ell(e) = \ell(e')$, for all $e, e' \in E$ (we refer to the version we defined as non-uniform). The uniform multi-commodity buy-at-bulk has an $O(\log n)$-approximation [4, 6, 19]. There are constant-factor approximation for the single-sink uniform case and some other special cases [21, 23, 24, 31]. Meyerson et al. [34] gave a randomized $O(\log n)$-approximation for the (non-uniform) BBST and this was derandomized in [15] using an LP formulation. For the (non-uniform) multi-commodity version [12] gave the first polylogarithmic approximation with ratio $O(\log^4 n)$. In [30] this was improved to $O(\log^3 n)$ if all the demands are polynomial in n. Some generalizations of these problems to higher connectivity are considered in [3, 22]. For hardness of approximation, Andrews [1] showed that unless NP $\subseteq$ ZPTIME $(n^{\text{polylog}\, n})$ the buy-at-bulk multicommodity problem has no $O(\log^{1/2-\epsilon} n)$-approximation algorithm for any $\epsilon > 0$. For the BBST [17] showed that the problem cannot be approximated better than $\Omega(\log \log n)$ unless NP $\subseteq$ DTIME$(n^{\log \log \log n})$.

The BBkST and SLkST problems generalize some classic problems such as Steiner tree and k-MST. The k-MST problem [5, 10, 20] is the special case of SLkST when $L = \infty$ and also the bounded diameter spanning tree problem [27] is the special case when costs are zero. Also, the SLST problem studied in [33] is a special case of SLkST with $k = |T|$. Even the $k = |T|$ special case is NP-hard and also NP-hard to approximate within a factor better than $c \log n$ for some universal constant c [7]. An (α, β)-bicriteria approximation algorithm for SLST or SLkST is an algorithm which finds a Steiner tree H (which has all the terminals in SLST or at least k terminals in SLkST) whose diameter (under ℓ metric) is at most βL and whose cost is at most α times an optimum solution with diameter bound L. For $k = |T|$ an $(O(\log n), O(\log n))$-approximation algorithm is given in [33] for SLST.

For the $(k, 2)$-subgraph problem, an $O(\log n \cdot \log k)$-approximation was presented in [32]. For the more general problem of requiring the k-subgraph to be 2-node-connected an $O(\log n \cdot \log k)$-approximation was presented in [16]. These are the best known approximation algorithms for the $(k, 2)$-subgraph problem. In [22] using a different approach an $O(\log^3 n)$-approximation was given. For metric cost functions, [36] presented an $O(1)$-approximation for (k, λ)-subgraph (the constant is very large though).

Our results: Our first result is an improved bicriteria approximation for SLkST.

Theorem 1 *There is a polynomial time $(O(\log^2 n), O(\log n))$-approximation for SLkST. More specifically, the algorithm finds a k-Steiner tree of diameter at most $O(L \cdot \log n)$ whose cost is at most $O(\text{OPT}^* \cdot \log^2 n)$ where OPT^* is the cost of an LP relaxation of the problem.*

To prove this theorem we combine ideas from all of [8, 14, 15, 30]. We first show that the algorithm of Marathe et al. [33] for SLST actually finds a solution with diameter at most $O(L \cdot \log|T|)$ whose cost is at most $O(\text{OPT}^* \cdot \log|T|)$, where OPT^* is the cost of a natural LP-relaxation, so we give a stronger bound (based on an LP relaxation) for the cost of their algorithm. This is based on ideas of [15] which gives a deterministic version of algorithm of [34] for BBST. Then we use an idea in [30] to write an LP for SLkST and use a trick in [8] for rounding this LP (the problem considered in [8] is completely unrelated to SLkST, namely k-ATSP tour problem). The only previous result for SLkST was [25] which had ratio $(O(\log^4 n), O(\log^2 n))$. This was obtained by applying the following theorem iteratively:

Theorem 2 *[25] There is a polynomial time algorithm that given an instance of the SLkST problem with diameter bound L returns a $\frac{k}{8}$-Steiner tree with diameter at most $O(\log n \cdot L)$ and cost at most $O(\log^3 n \cdot \text{OPT})$, where OPT is the cost of an optimum shallow-light k-Steiner tree with diameter bound L.*

Then a set-cover type analysis yields an $(O(\log^4 n), O(\log^2 n))$-approximation for SL$k$ST. We should point out that this theorem was the main ingredient in a greedy type $O(\log^4 n)$-approximation for multi-commodity buy-at-bulk in [12, 14] as well. In [25], the following lemma was also proved:

Lemma 1 *[25] Suppose we are given an approximation algorithm for the SLkST problem which returns a solution with at least $\frac{k}{8}$ terminals and has diameter at most $\alpha \cdot L$ and cost at most $\beta \cdot \text{OPT}$. Then we can obtain an approximation algorithm for the BBkST problem that such that given an instance of BBkST in which all demands $\delta_i = 1$ and a given parameter $M \geq \text{OPT}$ (where OPT is the optimum cost of the BBkST instance) returns a solution of cost at most $O((\alpha + \beta)\log k \cdot M)$.*

The corollary of this lemma and Theorem 2 is an $O(\log^4 n)$-approximation for the BBkST for unit demand instances; this can also be extended to an $O(\log^3 n \cdot \log D)$-approximation for general demands where $D = \sum_t \delta_t$. Using Theorem 1 and Lemma 1 we obtain:

Corollary 1 *There is an $O(\log^2 n \cdot \log D)$-approximation for BBkST, where D is the sum of demands.*

This improves the result of [25] for BBkST by a $\log n$ factor. Finally, we improve the result of [32] for the $(k, 2)$-subgraph problem:

Theorem 3 *There is an $O(\log n)$-approximation for the $(k, 2)$-subgraph problem.*

This is based on rounding an LP relxation of the problem similar to the one presented in [32]. Again we use the trick of [8] to round this LP and use the ideas of [32] to prune a partial solution.

2 Shallow-Light Steiner Trees

In this section we prove Theorem 1. In order to prove this we first show that the algorithm of [33] in fact bounds the integrality gap of the SLST problem too. Recall that the instance of SLST consists of a graph $G = (V, E)$ with costs $c(e)$, lengths $\ell(e)$, terminal set $T \subseteq V$ including a node r. The goal is to find a Steiner tree H over T with minimum $\sum_{e \in H} c(e)$ such that the diameter w.r.t. ℓ function is at most L. First, let us briefly explain the algorithm of [33] for SLST. Denote the given instance of SLST by $\mathcal{I}$ and define graph F over terminals as below. For every pair of terminals

$u, v \in T$, let $b(u, v)$ be the (approximate) lowest c-cost path between them whose length (under ℓ) is no more than L (there is an FPTAS for computing the value of $b(u, v)$ [26]); let the weight of edge between (u, v) in F be cost of $b(u, v)$. It is a simple exercise to show that in the optimum solution of $\mathcal{I}$, we can pair the terminals (except possibly one if the number of them is odd) in such a way that the unique paths connecting the pairs in the optimum are all edge-disjoint. Therefore, the total cost of these paths is at most the value of optimum solution, denoted by OPT, and the length of each of them is at most L. So, if we consider a minimum cost maximum matching in F, the cost of this matching is at most $(1 + \epsilon)$OPT. We find a minimum cost maximum matching in F and let say terminals $\{u_i, v_i\}_i$ are paired. We pick one of the two (arbitrarily), say u_i and remove v_i from the terminal set; let this new instance be $\mathcal{I}'$. Clearly the cost of optimum solution on $\mathcal{I}'$, denoted by OPT′, is at most OPT (as the original solution is still feasible). Also, for any solution of $\mathcal{I}'$, we can add the paths defined by $b(u_i, v_i)$ to connect v_i to u_i. This gives a solution to instance $\mathcal{I}$ of cost at most OPT′ + $(1 + \epsilon)$OPT and the diameter increases by at most L. We can do this repeatedly for $O(\log |T|)$ iterations until $|T| = 1$, since each time the number of terminals drops by a constant factor.

Remark: A similar algorithm was desgined in [34] to obtain an $O(\log n)$-approximation for BBST problem. Then an LP-based algorithm was presented by Chekuri et al. [15] to derandomize the algorithm of [34] for BBST.

We use the same approach as in [15] to bound the integrality gap of SLST. This LP is a flow-based LP (like those used in [14, 15]). We use the idea of [30] which only considers bounded lengths flow paths. For each terminal $t \in T$ let $\mathcal{P}_t$ be the set of all paths of length at most L from t to r in G. We assume that the terminals are at distinct nodes (we can enforce this by attaching some dummy nodes with edge cost and length equal to zero to the original nodes). Therefore, $\mathcal{P}_t$ and $\mathcal{P}_{t'}$ are disjoint. For every edge e we have an indicator variable x_e which indicates whether edge e belongs to the tree H or not. For each path $p \in \bigcup_t \mathcal{P}_t$, $f(p)$ indicates whether path p is used to connect a terminal to the root.

$$\begin{array}{llrcllr}
\textbf{LP-SLST} & \min & \sum_e c(e) \cdot x_e & & & & \\
 & \text{s.t.} & \sum_{p \in P_t | e \in p} f(p) & \leq & x_e & \forall e \in E, \quad t \in T & (1) \\
 & & \sum_{p \in \mathcal{P}_t} f(p) & \geq & 1 & t \in T & (2) \\
 & & x_e, f(p) & \geq & 0 & \forall e \in E, \quad p \in \cup_t \mathcal{P}_t & (3)
\end{array}$$

Define graph F over terminals T as above, i.e. the weight of edge $e = (u, v) \in F$ for two terminals $u, v \in T$ will be the cost of $(1 + \epsilon)$-approximate minimum c-cost u, v-path of length at most L computed using algorithm of [26]. Let (x^*, f^*) be an optimal solution to LP-SLST with cost OPT*. We show that the cost of algorithm of [33] is at most $O(\text{OPT}^* \cdot \log |T|)$ while the diameter is at most $O(L \cdot \log |T|)$. The proof of the following lemma is similar to that of Lemma 2.1 in [15] and is omitted here.

Lemma 2 *The graph F contains a matching M of size at least $|T|/3$ whose cost is at most* $(1 + \epsilon)$OPT*.

Suppose we have a matching M as above with cost C_M. For every pair of terminals u_i, v_i matched by M pick one of the two as the hub for connecting both of them to r and remove the other one from T. Let OPT′ be the LP cost of the new instance. The current solution (x^*, f^*) is still feasible for the new instance; therefore OPT′ ≤ OPT*. Also, the cost of routing all terminals that were deleted to their hubs is at most $C_M \leq (1 + \epsilon)$OPT*. Doing this iteratively, an easy inductive argument (using the fact that the number of terminals drops by a constant factor at each iteration.) shows that we obtain a solution whose cost is at most $O(\log|T|\cdot\textsc{opt}^*)$ and the diameter of the solution is at most $O(L\cdot\log|T|)$.

Now we prove Theorem 1. Our algorithm is based on rounding a natural LP relaxation of the problem. Before presenting the LP we explain how we preprocess the input. We first guess a value $\textsc{opt}'$ such that $\textsc{opt}\le\textsc{opt}'\le 2\textsc{opt}$. We do a binary search between zero and the largest possible value of $\textsc{opt}$ (e.g. $\sum_{e\in E}c(e)$). The solution returned by the algorithm satisfies the bounds if $\textsc{opt}'\ge\textsc{opt}$. If the algorithm fails we adjust our guess. We define $V'\subseteq V$ to be the set of vertices v such that v has a path p to r with $c(p)\le\textsc{opt}'$ and length at most L. Clearly, every vertex of any optimum solution must belong to V'. We can safely delete all the vertices of $V\setminus V'$; so let G be the new graph after pre-processing. The following LP is similar to LP-SLST, except that we have an indicator variable y_t for every terminal.

$$\begin{array}{llrcllr}
\textbf{LP-SL}k\textbf{ST} & \min & \sum_e c(e)\cdot x_e & & & & \\
 & \text{s.t.} & \sum_{p\in P_t|e\in p} f(p) & \le & x_e & \forall e\in E,\quad t\in T & (4)\\
 & & \sum_{p\in\mathcal{P}_t} f(p) & \ge & y_t & t\in T & (5)\\
 & & \sum_{t\in T} y_t & \ge & k & & (6)\\
 & & y_t & \le & 1 & t\in T & (7)\\
 & & x_e, f(p) & \ge & 0 & \forall e\in E,\quad p\in\cup_t\mathcal{P}_t &
\end{array}$$

If we replace y_t in the 2nd constraint with 1 and drop constraints (6) and (7) (and remove y_t variables) then we obtain the LP-SLST. Our rounding algorithm is similar to those in [14, 8] for two completely different problems (density version of Buy-at-Bulk Steiner tree in [14] and k-ATSP tour in [8]). Since we need to solve this LP let's briefly say that although LP-SLkST has an exponential number of variables, one can obtain an optimum feasible solution if one can give a separation oracle for the dual. It is easy to verify that a shortest-path algorithm gives a separation oracle for the dual LP. Suppose that (x^*,y^*,f^*) is an optimum feasible solution to LP-SLkST with value $\textsc{opt}^*$. Our first step is to convert (x^*,y^*,f^*) to an approximate solution in which y_t values are of the form 2^{-i}, $0\le i\le\lceil 3\log n\rceil$. Lemmas 3 and 5 are analogous of Lemma 9 and Theorem 10 in [8].

Lemma 3 *There is a feasible solution (x',y',f') to LP-SLkST of cost at most $4\textsc{opt}^*$ such that each y'_t is equal to 2^{-i} for some $0\le i\le\lceil 3\log n\rceil$.*

Proof. Let (x^*,y^*,f^*) be an optimal feasible solution to LP-SLkST. We set $x'_e=4x^*_e$ for all $e\in E$ and $f'(p)=\min(4f^*(p),1)$ for all $t\in T$ and $p\in\mathcal{P}_t$. For each $t\in T$ and i such that $1/2^i\le y^*_t<1/2^{i-1}$, if $i>\lceil 3\log(n)\rceil$ set $y'_t=0$; otherwise, $y'_t=\min(1,1/2^{i-2})$. It is easy to see that cost of (x',y',f') is at most $4\textsc{opt}^*$. Also, the first constraint is satisfied. The second constraint is also satisfied since it is clearly satisfied if $f'(p)=4f^*(p)$ for all $p\in\mathcal{P}_t$, and if this is not the case then at least one $f'(p)=1$ which at least as big as y'_t since $y'_t\le 1$. So it only remains is to show that the last constraint is satisfied.

Let Y_0 be the set of terminals t for which $y^*_t>0$ but $y'_t=0$. These are the only terminals whose y value has decreased. Note that for each $t\in Y_0$: $y^*_t\le 1/n^3$; so $\sum_{t\in Y_0}y^*_t\le 1/n^2$. Let Y_1 be the set of terminals t with $y'_t=1$. If $|Y_1|\ge k$, then the last constraint clearly holds. Otherwise, $|Y_1|\le k-1$ which implies that $\sum_{t\notin Y_1}y^*_t\ge 1$ must be true; therefore $\sum_{t\notin Y_1\cup Y_0}y^*_t\ge 1-1/n^2\ge 1/n^2\ge\sum_{t\in Y_0}y^*_t$. Also, note that for each vertex $t\notin Y_0\cup Y_1$: $y'_t\ge 2y^*_t$. Thus, the amount $\sum_{t\in Y_0}y^*_t$ that is decreased in y' is compensated for by $\sum_{t\notin Y_0\cup Y_1}y'_t$ therefore the last constraint holds too. ■

Let T_i be the set of terminals with $y'_t = 2^{-i}$ and $k_i = |T_i|$, for $0 \leq i \leq \lceil 3\log n \rceil$. Note that $\sum_{i=0}^{\lceil 3\log n\rceil} 2^{-i} \cdot k_i \geq k$. Consider the instance of SLST defined over $T_i \cup \{r\}$. First observe that we can obtain a feasible solution (x'', f'') to LP-SLST over this instance of SLST of cost at most $2^{i+2} \cdot \text{OPT}^*$ in the following way: define $x''_e = 2^i \cdot x'_e$ for each edge $e \in E$ and $f''(p) = 2^i \cdot f'(p)$ for each $t \in T_i$ and path $p \in \mathcal{P}_t$. The cost of this solution is $O(2^{i+2} \cdot \text{OPT}^*)$ since $x''_e = 2^{i+2} \cdot x^*_e$. Now since we proved the integrality gap of LP-SLST is $O(\log n)$, we obtain the following:

Lemma 4 *For each T_i, we can find a Steiner tree over $T_i \cup \{r\}$, rooted at r of total cost $O(2^{i+2} \cdot \text{OPT}^* \cdot \log n)$ and diameter $O(L \cdot \log n)$.*

Next we prove the following lemma.

Lemma 5 *For every $0 \leq i \leq \lceil 3\log n \rceil$ and given a Steiner tree H_i over T_i with total cost $O(2^{i+2} \cdot \text{OPT}^* \log n)$ and diameter $O(L \cdot \log n)$ we can find a Steiner tree H'_i rooted at some $r_i \in T_i$ containing at least $\lceil k_i/2^i \rceil$ terminals of T_i of cost at most $O(\text{OPT}^* \cdot \log n)$ and diameter at most $O(L \cdot \log n)$.*

For now, let us assume this lemma and see how to complete the proof. Suppose that H'_i is the Steiner tree promised by Lemma 5 which contains $\lceil k_i/2^i \rceil$ terminals of T_i and is rooted at at a node r'_i. Let p_i be the minimum cost path from r'_i to r with length at most L (note that because of the pre-processing we did, such path p_i exists). Let $H''_i = H'_i \cup p_i$ and let $H = \bigcup_i H''_i$. Observe that H contains at least $\sum_{i=0}^{\lceil 3\log n\rceil} 2^{-i} \cdot k_i \geq k$ terminals. Also, the total cost of H is at most $\sum_{i=0}^{\lceil 3\log n\rceil} c(H''_i) \leq O(\text{OPT}^* \cdot \log^2 n)$. Since the diameter of each H''_i is at most $O(L \cdot \log n)$ (because diameter of H'_i is at most $O(L \cdot \log n)$ and we added a path p_i of length at most L to H'_i) and since all of H''_i's share the root r, the diameter of H is at most $O(L \cdot \log n)$ as well. This completes the proof of Theorem 1.

So it only remains to prove Lemma 5. If we are given the Steiner tree H_i over T_i we use the following lemma with $\beta = \lceil k_i/2^i \rceil$ to edge-decompose H_i into $F_1, \ldots, F_d$ such that the number of terminals of each F_i is in $[\beta, 3\beta)$. It follows that $d = \theta(2^i)$ and so by an averaging argument, at least one of F_i's has cost $O(\text{OPT}^* \cdot \log n)$. The proof of the following lemma is implicit in [18] and is explicitly proved in [29].

Lemma 6 *Given a rooted tree F containing a set of k terminals and given an integer $1 \leq \beta \leq k$ we can edge-decompose F into trees $F_1, \ldots, F_d$ with the number of terminals of each F_i in $[\beta, 3\beta)$, $1 \leq i \leq d$.*

3 $O(\log n)$-approximation $(k, 2)$-subgraph problem

In this section we prove Theorem 3. In fact (similar to the algorithm in [32]) our algorithm works for a slightly more general case in which along with the weighted graph $G = (V, E)$ and integer k we are also given a set of terminals $T \subseteq V$ and the goal is to find a minimum cost 2-edge-connected subgraph that contains at least k terminals. Since our algorithm is based on that of [32], let us briefly explain how their algorithm works. The algorithm of [32] is for the rooted version of the problem, in which we are given an extra parameter $r \in V$ in the input and the solution must contain root r. Since one can try every possible vertex as the root, we can reduce the un-rooted version to the rooted version as well. A partial solution is a 2-edge-connected subgraph containing the root and the density of a partial solution is the ratio of the cost of the edges over the number of terminals it contains. The algorithm of [32] was based on finding a good density partial solution iteratively until the number of terminals is at least k. They presented an $O(\log n)$-approximation for finding good density partial solutions using an LP rounding procedure and since one has to repeat the procedure until the number of terminals covered is at least k, a simple set-cover type analysis shows the final approximation ratio would be $O(\log n \cdot \log k)$. One has to be careful as in an iteration where we are looking to cover k' terminals (for some $k' \leq k$) it is possible to find a partial solution with much larger than k' terminals (and so the combined solution has much larger than k terminals). In that case the algorithm has to be able to prune the partial solution to obtain a good density solution with about k' terminals. Lau et al. [32] present an algorithm for this pruning step which we will use too.

Our algorithm will round a LP relaxation directly instead of iteratively finding good density partial solutions. This is similar to the overall structure of the algorithm we presented for the SLkST. Note that, it is sufficient to find a solution in which every terminal has two edge-disjoint paths to r. Similar to [32] first we preprocess the graph by deleting the vertices that cannot be part of any optimum solution. For that for every vertex v we find two edge-disjoint paths between v and r of minimum total cost, let us denote it by $d_2(v,r)$. For this we can use a minimum cost flow algorithm between v and r [38]. Suppose we know have guessed a value $\textsc{opt}'$ such that $\textsc{opt} \leq \textsc{opt}' \leq 2\textsc{opt}$, where $\textsc{opt}$ is the value of optimum solution. Clearly every vertex v with $d_2(v,r) > \textsc{opt}'$ cannot be part of any optimum solution and can be safely deleted. We work with this pruned version of graph G. Our algorithm is guided by the solution of an LP relaxation of the problem. Consider the following LP relaxation which is similar to what proposed by Lau *et al.*[32].

$$\begin{array}{llrcll}
\textbf{LP-k2EC} & \min & \sum_e c(e)\cdot x_e & & & \\
& \text{s.t.} & x(\delta(U)) & \geq & 2y_v & U \subseteq V-\{r\}, v\in U & (8)\\
& & x(\delta(U)) - x_{e'} & \geq & y_v & U \subseteq V-\{r\}, v\in U, e'\in\delta(U) & (9)\\
& & \sum_{v\in T} y_v & \geq & k & & (10)\\
& & y_r & = & 1 & & (11)\\
& & y_t & \leq & 1 & & (12)\\
& & x_e, y_v & \geq & 0 & \forall e\in E, \quad v\in T &
\end{array}$$

There are two types of indicator variables, x_e for each $e \in E$ and y_v for each $v \in T$; for every subset $U \subseteq V$, $\delta(U)$ is the set of edges across the cut $(U, V-U)$. Constraints (8) and (9) guarantee 2-edge-connectivity to the root. Our algorithm solves this LP and then uses the solution to find an integral solution of cost at most $O(\log(n))$ apart from the optimal value, in order to do that we merge ideas from [8] and [32]. As argued in [32] this LP is a relaxation of the $(k,2)$-subgraph problem and we can find an optimum solution of this LP. We run the following algorithm whose detailed steps are explained below.

In the rest of this section we show that Algorithm K2EC finds a 2-edge-connected subgraph of value $O(\log(n)\cdot\textsc{opt})$ for the $(k,2)$-subgraph problem. First we provide the details of the steps of the algorithm. Assume we sort all the vertices v according to their $d_2(v,r)$ value and let L be the kth smallest value. It is easy to see that $L \leq \textsc{opt} \leq k.L$. So we can start with L as our guess for $\textsc{opt}'$; if the algorithm fails to return a feasible solution of cost at most $O(\textsc{opt}'\cdot\log n)$ then we double our guess $\textsc{opt}'$ and run the algorithm again. Note that in $O(\log k)$ many steps will have a guessed value $\textsc{opt}'$ with $\textsc{opt} \leq \textsc{opt}' \leq 2\textsc{opt}$ and therefore all the vertices that are deleted surely cannot be part of an optimum solution. Let (x^*, y^*) be an optimum feasible solution to LP-k2EC with value $\textsc{opt}^*$. For Step 5 of K2EC we round y values of the LP following the schema in [8]. The proof of following lemma is very similar to Lemma 3 and appears in Appendix A

```
$(k,2)$-Subgraph Algorithm (k2EC)

    Input: Graph $G = (V, E)$, terminal set $T \subseteq V$ with root $r$, and integer $k \geq 1$
    Output: a 2-edge-connected subgraph containing at least $k$ terminals including $r$
    1. Guess a value of $\textsc{opt}'$ for optimum solution and run the following algorithm.
    2. $U \leftarrow r$
    3. Start from original graph $G$ and remove all the vertices with $d_2(v,r) > \textsc{opt}'$
    4. Solve LP-K2EC and let its solution be $(x^*, y^*)$
    5. Obtain $(x', y')$ from $(x^*, y^*)$ according to Lemma 7
    6. Let $T_i$ be the set of terminals $v$ with $y'_v = 2^{-i}$ plus the root, for $0 \leq i \leq \lceil 3\log(n) \rceil$
    7. Find a 2-edge-connected subgraph $H_i$ over $T_i \cup \{r\}$ with cost $O(2^i \cdot \textsc{opt}^*)$
    8. From $H_i$, find a 2-edge-connected subgraph $H'_i$ containing $r$ and at least $\lceil |T_i|/2^i \rceil$ and
         at most $2\lceil |T_i|/2^i \rceil$ vertices of $T_i$ of cost at most $O(\textsc{opt}^*)$ and add it to $U$;
         if failed for any $i$ then double the guess for $\textsc{opt}'$ and start from Step 2.
    9. Return $U$.
```

Figure 1: Algorithm K2EC

Lemma 7 *There is a feasible solution (x', y') to LP-K2EC of cost at most $4\textsc{opt}^*$ such that all nonzero entries of y' belong to $\{2^{-i} | 0 \leq i \leq \lceil 3\log(n) \rceil\}$.*

Let T_i be the set of terminals with $y'_t = 2^{-i}$ and $k_i = |T_i|$, for $0 \leq i \leq \lceil 3\log n \rceil$. Note that $\sum_{i=0}^{\lceil 3\log n \rceil} 2^{-i} \cdot k_i \geq k$. Consider an instance of classical survivable network design problem over terminals in $T_i \cup \{r\}$ with connectivity requirement 2 from every node in T_i to root. In the following lemma we show that we can compute a 2-edge-connected subgraph H_i over $T_i \cup \{r\}$ of cost at most $O(2^i \cdot \textsc{opt}^*)$. This describes how to perform Step 7. The proof of this lemma is similar to Lemma 5.2 in [32].

Lemma 8 *In Step 7, For each $0 \leq i \leq \lceil 3\log n \rceil$, we can find a 2-edge-connected subgraph H_i of cost at most $2^{i+3} \cdot \textsc{opt}^*$ containing terminals $T_i \cup \{r\}$.*

Proof. In order to bound the cost of 2-edge-connected subgraph over $T_i \cup \{r\}$ we use the following natural LP for the special case of survivable network design problem in which all connectivity requirements are 2:

$$\begin{array}{llrcll} \textbf{LP-2EC} & \min & \sum_e c(e) \cdot x_e & & & \\ & \text{s.t.} & x(\delta(U)) & \geq & 2 \quad U \subseteq V - \{r\}, U \cap T_i \neq \emptyset & (11) \\ & & 1 \geq x_e & \geq & 0 \quad \forall e \in E & \end{array}$$

Jain [28] proved that the integrality gap of this LP is at most 2. Here, we show that after scaling (x', y'), we can find a feasible solution of LP-2EC over terminals $T_i \cup \{r\}$ of value at most $2^{i+2} \cdot \textsc{opt}^*$. Using Jain's algorithm, we can then obtain an integer solution, i.e. a 2-edge-connected subgraph over $T_i \cup \{r\}$ of cost at most $2^{i+3} \cdot \textsc{opt}^*$, which completes the proof of lemma.

Consider (x', y') obtained by Lemma 7 and define $\hat{x}_e = \min(1, 2^i \cdot x'_e)$. We will show that $\hat{x}$ is a feasible solution for LP-2EC, which clearly has cost at most $2^{i+2} \cdot \textsc{opt}^*$ since $2^i \cdot x'_e = 2^{i+2} \cdot x^*_e$.

To verify that $\hat{x}$ is feasible for LP-2EC, take any set $U \subseteq V - r$ with $U \cap T_i \neq \emptyset$ and the corresponding constraint (11) in LP-2EC: $x(\delta(U)) \geq 2$. This has the corresponding constraint (8) in LP-k2EC $x(\delta(U)) \geq 2y_v$ for each $v \in U - \{r\}$. Suppose we define $\hat{x}_e = \min\{1, 2^i \cdot x'_e\}$ and $\hat{y}_v = \min\{1, 2^i \cdot y'_v\}$. Note that for each $v \in T_i$: $\hat{y}_v = 1$. If all the edges $e \in \delta(U)$ have values $x'_e \leq y'_v$ then after scaling we will have $\hat{x}(\delta(U)) \geq 2$ because the left hand side of $x(\delta(U)) \geq 2y_v$ is grown at least as much as the RHS is scaled. If there is at least one edge $e' \in \delta(U)$ with $x'_{e'} > y'_v$ then because of constraints (9) in LP-k2EC and since (x', y') is feasible, we have $x'(\delta(U)) - x'_{e'} \geq y'_v$. Thus after the scaling we still have $\hat{x}(\delta(U)) - \hat{x}_{e'} \geq 1$ because again the LHS is grown at least as much as the RHS. Also $\hat{x}_{e'} = 1$ because $\hat{y}_v = 1$ and $x'_{e'} > y'_v$; so $\hat{x}(\delta(U)) \geq 2$. This shows constraints (11) in LP-2EC are satisfied and so there is a feasible solution to LP-2EC with terminal set $T_i \cup \{r\}$ with cost at most 2^iOPT*. ■

In the following we show how to find subgraph H'_i in Step 8, which is 2-edge-connected, has root r, and has cost $O($OPT$')$, assuming that OPT$' \geq$ OPT. Note that union of all H_i's ($0 \leq i \leq \lceil 3 \log n \rceil$) will be 2-edge-connected (since r is common in H'_i's), has at least k terminals, and has cost $O($OPT$' \cdot \log n)$. This will complete the proof of approximation ratio of the algorithm.

To show how to find a subgraph H'_i we use the same trick as in Section 5.1 of [32] for pruning a large good density solution to a smaller one. A nowhere-zero 6-flow in a directed graph $D = (V, A)$, is a function $f : A \rightarrow \mathbb{Z}_6$ such that we have flow conservation at every node (*i.e.* $f(\delta^{in}(v)) = f(\delta^{out}(v)))$ and no edge gets f value of zero. If there is an orientation of an undirected graph H in which a nowhere-zero 6-flow can be defined we say H has a nowhere-zero 6-flow. Seymour [39] proved that every 2-edge-connected graph has a nowhere-zero 6-flow which can also be found in polynomial time. We obtain a multigraph $D = (H_i, A)$ from H_i by placing $f(e)$ copies of e with the direction defined by the flow. From Lemma 8 and the fact that we have at most 6 copies of each edge, the cost of D can be at most $6 \times 2^{i+3} \cdot$ OPT*.

Note that D does not have directed cycle of length 2, therefore has an Eulerian Walk. Start from r and build an Eulerian walk and partition the walk into the segments $P_1, P_2, \ldots, P_\ell$ each of which includes $\lceil |T_i|/2^i \rceil$ terminals of H_i excepts possibly P_ℓ which can have between $\lceil |T_i|/2^i \rceil$ and $2\lceil |T_i|/2^i \rceil$ terminals. Thus, $\ell \geq \max(1, 2^{i-1})$ and so there is an index $1 \leq q \leq \ell$ such that the cost of path P_q is at most $6 \times 2^{i+2} \cdot$ OPT$^*/2^{i-1} = 48$OPT*. Let u, w be the endpoints of P_q and let Q^1_u and Q^2_u be the two edge-disjoint paths of $d_2(u, r)$ (in G) and Q^1_w and Q^2_w be the two edge-disjoint paths of $d_2(w, r)$ (again in G) of minimum total cost. Because of the preprocess step, the sum of costs of Q^1_u, Q^2_u, Q^1_w, and Q^2_w is at most 2OPT$'$. Let F_q be the simple graph in G defined by the edges of P_q and let $H'_i = F_q \cup Q^1_u \cup Q^2_u \cup Q^1_w \cup Q^2_w$. It follows that H'_i has cost at most 48OPT* + 2OPT$' \leq 50$OPT$'$. It only remains to show that H'_i is 2-edge-connected. By way of contradiction, suppose there is an edge e' such that $H'_i - e'$ has two components C_1 and C_2. Because of Q^1_u, Q^2_u, Q^1_w, and Q^2_w the two endpoints u and w are in the same component let say C_1. Since P_q is a directed walk from u to w and there is no cycle of size 2, there must be another edge $e'' \neq e'$ between C_1 and C_2 which goes in opposite direction of e', thus e' is not a cut edge.

References

[1] M. Andrews. Hardness of buy-at-bulk network design. *Proc. of IEEE FOCS*, 115–124, 2004.

[2] , M. Andrews and L. Zhang. Approximation algorithms for access network design, Algorithmica 32(2):197-215, 2002. Preliminary version in *Proc. of IEEE FOCS*, 1998.

[3] S. Antonakopoulos, C. Chekuri, B. Shepherd, and L. Zhang, Buy-at-bulk network design with protection, InProceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science FOCS 2007.

[4] , B. Awerbuch and Y. Azar, Buy-at-bulk network design, In Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS '97),

[5] B. Awerbuch, Y. Azar, A. Blum, and S. Vempala, New approximation guarantees for minimum-weight k-trees and prize-collecting salesmen, SIAM Journal on Computing 28(1):254-262, 1999.

[6] Y. Bartal. On approximating arbitrary metrics by tree metrics. *Proc. of ACM STOC*, 161–168, 1997.

[7] J. Bar-Ilan, G. Kortsarz, and D. Peleg, Generalized submodular cover problems and applications, Theoretical Computer Science Vol. 250, pages 179-200, 2001.

[8] M. Bateni and J. Chuzhoy, Approximation Algorithms for the Directed k-Tour and k-stroll Problems, In proceedings of Approx 2010

[9] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan, Detecting High Log-Densities – an $O(n^1/4)$-Approximation for Densest k-Subgraph In Proceedings of Symposium on the Theory of Computing (STOC) 2010.

[10] A. Blum, R. Ravi, and S. Vempala, A constant-factor approximation algorithm for the k-MST problem, In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (STOC '96),

[11] M. Charikar and A. Karagiozova, On non-uniform multicommodity buy-at-bulk network design, In Proceedings of the thirty-seventh annual ACM symposium on Theory of computing, STOC 2005.

[12] C. Chekuri, M. Hajiaghayi, G. Kortsarz, and M. Salavatipour. Approximation algorithms for non-uniform buy-at-bulk network design problems. *Proc. of IEEE FOCS*, 677–686, 2006.

[13] C. Chekuri, M. Hajiaghayi, G. Kortsarz, and M. Salavatipour. Approximation algorithms for node-weighted buy-at-bulk network design. *Proc. of ACM-SIAM SODA*, 1265–1274, 2007.

[14] , C. Chekuri, M. Hajiaghayi, G. Kortsarz, and M. Salavatipour. Approximation Algorithms for Non-Uniform Buy-at-Bulk Network Design SIAM J. on Computing, 39(5):1772–1798, 2009.

[15] C. Chekuri, S. Khanna and J. Naor. A Deterministic Approximation Algorithm for the Cost-Distance Problem Short paper in *Proc. of ACM-SIAM SODA*, 232–233, 2001.

[16] C. Chekuri and N. Korula Pruning 2-Connected Graphs, In Proceedings of FSTTCS, December 2008. Preliminary version with a different title in ArXiv.

[17] , J. Chuzhoy, A. Gupta, S. Naor, and A. Sinha, On the approximability of some network design problems, In Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms (SODA '05).

[18] G. Even, N. Garg, J. Konemann, R. Ravi, and A. Sinha Covering graphs using trees and stars, Operations Research Letters 32(4):309-315, 2004. Earlier version in Proceedings of APPROX 2003.

[19] , J. Fakcharoenphol, S. Rao, and K. Talwar, A tight bound on approximating arbitrary metrics by tree metrics, Journal of Computer and System Sciences 69(3):485-497, 2004.

[20] N. Garg, Saving an epsilon: a 2-approximation for the k-MST problem in graphs, In Proceedings of the thirty-seventh annual ACM symposium on Theory of computing (STOC), 396 - 402, 2005.

[21] S. Guha, A. Meyerson, and K. Munagala, A constant factor approximation for the single sink edge installation problems, In Proceedings of the thirty-third annual ACM symposium on Theory of computing (STOC '01).

[22] A. Gupta, R. Krishnaswamy, and R. Ravi, Tree Embeddings for Two-Edge-Connected Network Design In Proceedings of SODA 2010.

[23] 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 Proceedings of the 44rd Symposium on Foundations of Computer Science (FOCS '03).

[24] A. Gupta, A. Kumar, T. Roughgarden, Simpler and better approximation algorithms for network design, In Proceedings of the thirty-fifth ACM symposium on Theory of computing (STOC '03).

[25] M.T. Hajiaghayi, G. Kortsarz, and M.R. Salavatipour. Approximating buy-at-bulk and shallow-light k-steiner tree. Algorithmica 53(1): 89-103, 2009.

[26] R. Hassin, Approximation schemes for the restricted shortest path problem. Mathematics of Operations Research 17(1):36–42, 1992.

[27] R. Hassin and A. Levin, Minimum Restricted Diameter Spanning trees, In Proceedings of APPROX 2002.

[28] K. Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[29] M.R. Khani and M.R. Salavatipour, Approximation algorithms for min-max tree cover and bounded tree cover problems To appear in Proceedings of APPROX 2011.

[30] G. Kortsarz and Z. Nutov. Approximating some network design problems with vertex costs. *Proc. APPROX-RANDOM*, 231-243, 2009.

[31] A. Kumar, A. Gupta, and T. Roughgarden, A Constant-Factor Approximation Algorithm for the Multicommodity Rent-or-Buy Problem, In Proceedings of the 43rd Symposium on Foundations of Computer Science (FOCS '02).

[32] L. Lau, S. Naor, M.R. Salavatipour, and M. Singh , Survivable network design with degree or order constraints, SIAM J. on Computing 39(3): 1062-1087, 2009. Special issue for selected papers of STOC 2007.

[33] M. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. Rosenkrantz, H. B. Hunt. Bicriteria network design. *Journal of Algorithms*, 28(1):142–171, 1998.

[34] A. Meyerson, K. Munagala, and S. Plotkin. Cost-Distance: Two Metric Network Design. *SIAM J. on Computing*, 2648–1659, 2008. Preliminary version in *Proc. of IEEE FOCS*, 2000.

[35] R. Ravi, R. Sundaram, M.V. Marathe, D.J. Rosenkrants, and S.S. Ravi, Spanning trees short or small, SIAM Journal on Discrete Mathematics, 9(2):178-200, 1996.

[36] M.A. Safari and M.R. Salavatipour, A constant factor approximation for minimum λ-edge-connected k-subgraph with metric costs, To appear in SIDMA. Earlier version in APPROX 2006.

[37] F. S. Salman, J. Cheriyan, R. Ravi, and S. Subramanian, Approximating the Single-Sink Link-Installation Problem in Network Design, SIAM J. on Optimization 11(3):595-610, 2000. Preliminary version in *Proc. of ACM-SIAM SODA*, 619–628, 1997.

[38] A. Schrijver. *Combinatorial optimization: Polyhedra and Efficiency.* Springer-Verlag, Berlin, 2003.

[39] P.D. Seymour, In Graph Theory and related topics (Proc. Waterloo, 1977). Academic Press (1979), 341-355.

[40] V. Vazirani. *Approximation algorithms.* Springer, 1994.

A Proof of Lemma 7

We set $x'_e = min(4x^*_e, 1)$ for all $e \in E$ and for all $v \in T$, select i such that $2^{-i} \leq y_v < 2^{-i+1}$, then if $i > \lceil 3\log(n) \rceil$ set $y'_v = 0$; otherwise, $y'_v = min(1, 2^{-i+2})$. It is easy to see that cost of (x', y') is at most 4OPT^*; what remains is to show that (x', y') is a feasible solution to LP-K2EC. It is easy to see that Equations (8),(9),(11), and (12) are true for (x', y') as LHS is scaled at least as much as the RHS. Equation (10) is the only one to verify. As in the proof of Lemma 3, let Y_0 be the set of vertices v such that $y^*_v > 0$ but $y'_v = 0$. Note that $\sum_{v \in Y_0} y^*_v \leq 1/n^2$. These vertices are the only ones whose y value has decreased. Let Y_1 be the set of vertices v with $y'_v = 1$. If $|Y_1| \geq k$, then constraint (10) holds. Otherwise, $|Y_1| \leq k - 1$ which implies $\sum_{v \notin Y_1} y^*_v \geq 1$, and therefore $\sum_{v \notin Y_1 \cup Y_0} y^*_v \geq 1 - 1/n^2 \geq \sum_{v \in Y_0} y^*_v$. Note also for each vertex $v \notin Y_0 \cup Y_1$, we know that $y'_v \geq 2y^*_v$. Thus, the amount $\sum_{v \in Y_0} y^*_v$ is compensated for with $\sum_{v \notin Y_0 \cup Y_1} y'_v$; therefore constraint (10) continues to hold.