An Approximation Algorithm for the Balanced Capacitated Minimum Spanning Tree Problem

Haniyeh Fallah · Farzad Didehvar∗ · Farhad Rahmati
Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran, P. O. Box: 15875-4413

Received: date / Accepted: date

Abstract The capacitated minimum spanning tree problem (CMSTP), a well-known combinatorial optimization problem, holds a central place in telecommunication network design. This problem is to find a minimum cost spanning tree with an extra cardinality limitation on the orders of the subtrees incident to a certain root node. The balanced capacitated minimum spanning tree problem (BCMSTP) is a special case that aims to balance the orders of the subtrees. We show this problem is NP-hard and present two approximation algorithms, in this paper.

Considering the maximum order of the subtrees Q, we provide a \((3 - \frac{1}{Q})\)-approximation algorithm to find a balanced solution. We improve this result to a \((2.5 + \epsilon)\)-approximation algorithm (for every given \(\epsilon > 0\)), in the \(2d\)-Euclidean spaces. Also, we present a polynomial time approximation scheme (PTAS) for CMSTP.

Keywords Capacitated Minimum Spanning Tree Problem · Load balancing · Approximation Algorithms · PTAS

1 Introduction

The capacitated minimum spanning tree problem (CMSTP) is a fundamental problem in telecommunication network planning. In this problem, we are given an undirected graph with non-negative costs on its edges and non-negative weights on its nodes. Also, the given inputs are a root node \(r\) and a capacity constraint \(Q\). The goal is to find a minimum cost spanning tree rooted at \(r\) in which the sum of the vertex weights in each rooted subtree (that indicates its load) is at most \(Q\). In the absence of the capacity constraint, the problem is limited to find a minimum cost spanning tree. A special case is when all the node weights are equal and known as the homogeneous demand case. This is equivalent to the case where all the node weights are unit and usually referred to as CMSTP in the literature [1]. In this case, the problem reduces to find a rooted minimum spanning tree in which each of the subtrees incident to the root contains at most \(Q\) nodes. CMSTP considers the unit demands, investigated in this paper.

Many variations of CMSTP are formulated depending on the type of applications (see e.g. [2–8]). A variety of CMSTP considers additional constraints, like the balance of the nodes number in component subtrees (see [2,9]). Ali et al. [2] tackle the spanning trees and forests with a balanced nodes number in the component subtrees. Incel et al. [9] present a practical application for this problem in wireless sensor networks. This is the problem we consider in this article and refer to it as the balanced capacitated minimum spanning tree problem (BCMSTP).

We show BCMSTP is NP-hard and provide two approximation algorithms for this problem. An approximation algorithm runs in time polynomial with the input size and ensures the solution quality [10]. We use the approach of Vazirani [10] to define an approximation algorithm and approximation guarantee. For a minimization problem, an approximation algorithm achieves approximation ratio \(\rho\), if on every instance of the problem the ratio between the cost of the found solution by the algorithm and the cost of an optimal solution is at most \(\rho\).

Also, we provide a polynomial time approximation scheme (PTAS) for CMSTP. A PTAS is an approximation algorithm whose ratio is \((1 + \epsilon)\) for every constant \(\epsilon > 0\).

∗. Corresponding author. Tel.: +98-21-64545665
Email addresses: didehvar@aut.ac.ir (Farzad Didehvar); hfallah@aut.ac.ir (Haniyeh Fallah); frahmati@aut.ac.ir (Farhad Rahmati)
Here, we confine ourselves to Euclidean metrics and mention it as the Euclidean balanced capacitated minimum spanning tree problem (Euclidean BCMSTP). From the theoretical point of view, we interest to identify whether it is solvable more efficient than BCMSTP in general metrics.

BCMSTP has practicable applications in convergecast wireless sensor networks [9] that aim to collect data from a set of sensors toward a common sink such that the schedule length becomes a minimum. The load balancing in a CMSTP minimizes the largest load, so minimizes the schedule length.

In a different scenario, the root is a central processor, and the other nodes are terminals that request demands for traffic that must be sent to the central processor through the edges. The edge traffic is the number of traffic demands passing through it, so the maximum traffic is on the gates that are incident edges to the root. Balancing the loads minimizes the maximum traffic on the gates, so maximizes the network robustness to unpredicted growth of traffic demand. Actually, it reduces the likelihood of overfilling gates with the additional load. Such a situation may arise when new terminals move into the network, or if the connecting network is redefined when modifying pre-existing connections.

Generally, the applications of BCMSTP are in the same areas as the applications of CMSTP e.g. in telecommunications network design, the local access network design in computer communication networks, the design of the local loop systems between the branch offices and final users in telephone systems, facility location planning, transportation, logistics, and telecommunications companies that build fiber-optic based local access networks.

This problem is also prevalent in the design of bus routes in urban transportation systems where it is significant to fairly distribute services [11]. Here, the solution is a set of Hamiltonian paths connected to the root. From this perspective, BCMSTP is a variant of the open vehicle routing problem [12], which considers a fairness issue across its distribution services.

1.1 Our results

As per our knowledge, we did not find any literature which attempts to find an approximation algorithm for BCMSTP. We present the first \((3 - \frac{1}{d})\)-approximation algorithm to this problem, in general metrics. We improve this result to a \((2.5 + \epsilon)\)-approximation (for every given \(\epsilon > 0\), in 2-dimensional \((2d)\) Euclidean spaces. Also, we provide a PTAS for CMSTP in 2-dimensional \((2d)\) Euclidean spaces, when \(Q = o(\ln \ln n)\) (n is the number of nodes). This result solves the problem of obtaining a ratio any better than 2.9 [13] for this case. As far as we know, there is no previous attempt to derive a PTAS for CMSTP in \(m\)-dimensional Euclidean spaces for any fixed \(m \geq 2\).

1.2 Related works

BCMSTP pertains to the class of balanced combinatorial optimization problems that was first introduced by Martello et al. [14]. Many generalizations and variations of these problems were studied by several authors e.g. [2, 15–22].

For CMSTP with unit demands, Gavish and Altinkemer in [23] presented a factor \(4 - 1/(2^{\log Q} - 1)\) approximation algorithm. Then, Altinkemer and Gavish in [24] improved this ratio to \(3 - \frac{2}{Q}\) and 4 for the unit and non-unit demands, respectively. Later, Jothi and Raghavachari [13] improved the ratio from 4 to \(\gamma + 2\) in the non-unit demand case, where \(\gamma\) is the inverse Steiner ratio. The Steiner ratio is the topmost proportion of the costs for the minimum Steiner tree against the minimum spanning tree for the same instance. In graphs \(\gamma = 2\), and in Euclidean and rectilinear metrics it is \(2/\sqrt{3}\) and \(3/2\), respectively. Moreover, they obtained a 2.9-approximation for the unit demand CMSTP in the \(L_p\)-metric plane, and a 2-approximation for this problem in the general metrics, when \(Q \in \{3, 4\}\).

Exact solution methods for CMSTP (unit or non-unit demands) have been developed by Gavish [25] and Kershenbaum and Boorstyn [26]. There are several algorithms and mathematical formulations [27, 28, 4, 29, 30] available for solving this problem. Gamvros et al. [31] studied a multi-level capacitated minimum spanning tree (MLCMST) problem with unit demands on tree topology networks. In their paper, the authors assume that there are multiple types of a facility (maybe a cable) with different capacities and costs, can be used to install between two nodes to carry the traffic. In comparison to our problem, we assume that there is a single facility type with capacity of \(Q\). Also, we assume that the graph is complete and at most one facility could be installed between any two nodes.

The local access network design problem (see [32-34]) is a related problem that also deals with multiple facility types. However, in this problem, the topology of the network is not a tree. A related problem dealing with multiple facility types is the single sink buy-at-bulk (SSBB) problem [35]. Hassin et al. [5] provided algorithms for the single facility SSBB problem which is a special variant of CMSTP known as the network loading
problem in the literature, and where multiple copies of the facility may be installed on a graph edge. Hassin et al. [5] prepared approximations of factor 2 for unit demand and 3 for non-unit demand problems. A variation of CMSTP with connectivity constraint has been considered by Jothi and Raghavachari [36].

Paper outline. The rest of the paper is structured as follows. Section 2 presents the notations, definitions, and important assumptions. In Section 3, we analyze the complexity of the problem. Section 4 discusses about the balanced solution. We address BCMSTP (general metrics) in Section 5 and provide a $(3 - \frac{1}{K})$-approximation algorithm. In Section 6, we address this problem in $2d$-Euclidean metric spaces. We prepare an improved $(2.5 + \epsilon)$-approximation for every given $\epsilon > 0$. Also, we present a PTAS for CMSTP. The Conclusion section summarizes the results. We give an integer programing model for the problem in the Appendix.

2 Notations, Assumptions, and Problem Definition

The balanced capacitated minimum spanning tree problem (BCMSTP) can be described as under. Assume $G = (V, E)$ is a complete graph where $V = \{r, v_1, v_2, ..., v_n\}$ is the set of nodes and $E$ is the set of edges. Here, $r \in V$ is the root node. Each edge $(i, j) \in E$ has the weight of $c_{ij} > 0$ that denotes its length/cost. The weights on the edges are symmetric (i.e. $c_{ij} = c_{ji}$, for each $(i, j) \in E$) and obey the triangle inequality. Each node $v \in V \setminus \{r\}$ has unit demand. We will interchangeably use the terms vertices, nodes, and terminals. Let $T_v$ be a subtree dangling from $v$, the cost $C(T_v)$ indicates the sum of the edge costs incident in $T_v$. Moreover, the load $d(T_v)$ is the number of nodes (except root) incident in $T_v$ that also indicates its order. We seek to find a minimum cost spanning tree of $G$, so that order of each subtree dangling from the root does not surpass the capacity $Q$, and orders of the subtrees are balanced. At least $\lceil n/Q \rceil$ number of subtrees is required.

The number of subtrees may be a fixed parameter $K$, given by the user. However, it would be a decision variable that must be determined. When $K$ is fixed, we seek to find exactly $K$ subtrees. In the rest of the paper, we assume $K$ is fixed, except where noted.

A solution $S = \{T_1, T_2, ..., T_K\}$ of BCMSTP corresponds to a capacitated partition $P_1, P_2, ..., P_K$ which satisfies the following relations:

$$P_i \neq \emptyset, \forall i; \quad \bigcup_{1 \leq i \leq K} P_i = V \setminus \{r\}; \quad P_i \cap P_j = \emptyset, i \neq j; \quad \sum_{v \in P_i} 1 \leq Q, i \in [K].$$

Indeed, for each $T_i \in S$, we have $P_i = \{v : v \neq r, v \in T_i\}$. We say the nodes in $P_i$ are allocated to the $i$th subtree.

To avoid ambiguity, we notice that the notation $T_v$ (where $v$ is a terminal node) denotes a subtree dangling from $v$, while the notation like $T_i$ (where the index $i$ is not a terminal node) is used to denote a subtree dangling from the root $r$.

Let $L_S = (d(T_1), d(T_2), ..., d(T_K))$ be the loads of the subtrees in $S$ that specifies the “allocated loads” of $S$. In the next section, we define a “balanced load allocation” and a “balanced solution”.

We suggest a mathematical programing model for BCMSTP in the Appendix. The model reveals that BCMSTP is indeed CMSTP with extra limitations for the balance condition. The problem is NP-hard (see Section 3) and finding an approximation algorithm does not seem to be an easy task [37,38]. We prepare two approximation algorithms. The first algorithm $A(3)$ relies on a tour partitioning heuristic and is valid in any metric space. The second algorithm $A(3)$ is applicable in $2d$-Euclidean spaces. It separates the nodes into two parts: in the interior part, it employs $3$, and in the exterior part, it finds an optimal solution. The main idea is to separate the nodes such that the optimal solution in the exterior part could be found in polynomial time with respect to $n$ (i.e. the number of nodes). We will employ the following basic definition.

Definition 1 Let $\Gamma$ be an approximation algorithm for a minimization problem that finds a solution $S_\Gamma$ with cost $C(S_\Gamma)$. Let $OPT$ be an optimal value. The relative error of $\Gamma$ is defined by the equation

$$e_\Gamma = \frac{C(S_\Gamma) - OPT}{OPT}.$$  

The approximation ratio of $\Gamma$ is $\rho(\Gamma) = \frac{C(S_\Gamma)}{OPT}$.

2.1 The Balance Criteria

The equity measure is an indicator that evaluates the fairness of an allocated load vector. One of the accepted formulations for the notion of fairness in the load balancing domain is the range fairness that calculates
the maximum distinction between the loads (see e.g. [39]). Another equity measure is the ratio fairness that calculates the maximum ratio between the loads [40, 21].

Both the range and the ratio criteria convince the weak Pigou-Dalton (PD) principle which is the widely accepted property of equity measures [41]. For an allocated load vector \(x\) and an equity function \(I(x)\), let the vector \(x'\) be organized as follows: \(x_j = x_j + \delta\) , \(x_i = x_i - \delta\) , \(x_h = x_h\) for all \(h \notin \{j, i\}\). The PD principle (weak version) declares that \(I(x') \leq I(x)\) when \(0 \leq \delta \leq x_j - x_i\) is chosen. In the strong PD principle, the inequality is strict (\(I(x') < I(x)\)) which means that the new allocation must be more equitable. The PD principle can be used in situations where sum of the allocation outcomes and their number are identical.

We use the range fairness to evaluate the balance of the allocated loads, though the ratio fairness could also be used. Given a real parameter \(\alpha \geq 0\) and the number of subtrees \(K\), a partition of the nodes of \(G\) into \(K\) parts \(\{P_i : i \in [K]\}\) is \(\alpha\)-balanced if and only if for each \(i, j \in [K]\):

\[
|Load(P_i) - Load(P_j)| \leq \alpha,
\]

where \(Load(P_i) = \sum_{v \in P_i} 1\) and \(|x|\) is the absolute value of \(x\). This condition results in the following inequality:

\[
\max\{Load(P_i) : i \in [K]\} - \min\{Load(P_i) : i \in [K]\} \leq \alpha. \tag{1}
\]

The inequality (1) is the "balanced condition". Evidently, an \(\alpha_1\)-balanced partition is indeed \(\alpha_2\)-balanced for each \(\alpha_2 \geq \alpha_1\). The balanced range of the set \(\{P_i : i \in [K]\}\) is the smallest \(\alpha\) satisfying the balanced condition. Hence, \(\max\{Load(P_i) : i \in [K]\} - \min\{Load(P_i) : i \in [K]\}\) is the balanced range of the set \(\{P_i : i \in [K]\}\).

**Definition 2** A solution \(S\) of BCMSTP with the related partition \(P\) is \(\alpha\)-balanced (for a given parameter \(\alpha \geq 0\)) if and only if \(P\) is \(\alpha\)-balanced. The solution \(S\) is balanced if the balanced range of \(P\) is the least possible among the balanced ranges of the other capacitated partitions into \(K\) parts.

When the ratio measure is used, the partition \(\{P_i : i \in [K]\}\) is said to be \(\beta\)-balanced (for a given parameter \(0 \leq \beta \leq 1\)) if for each \(i, j \in [K]\):

\[
1 - \beta \leq \frac{Load(P_i)}{Load(P_j)} \leq 1 + \beta,
\]

which simply results in the following inequality:

\[
\frac{\max\{Load(P_i) : i \in [K]\}}{\min\{Load(P_i) : i \in [K]\}} \leq 1 + \beta. \tag{2}
\]

Let \(\alpha\) \((0 \leq \alpha \leq Q)\) be a given load deviation and let \(\beta \leq \alpha/Q\) is chosen. A \(\beta\)-balanced set \(\{P_i : i \in [K]\}\) is also \(\alpha\)-balanced: we see \(-\frac{\alpha}{Q} \leq -\beta \leq \frac{Load(P_i)}{Load(P_j)} - 1 \leq \beta \leq \frac{Q}{Load(P_j)}\) \((Load(P_i) - Load(P_j)) \leq \alpha\). If \(Load(P_i) \geq Load(P_j)\) we have:

\[-\alpha \leq 0 \leq Load(P_i) - Load(P_j) \leq \frac{Q}{Load(P_j)} (Load(P_i) - Load(P_j)) \leq \alpha.\]

If \(Load(P_i) \leq Load(P_j)\), we have:

\[-\alpha \leq \frac{Q}{Load(P_j)} (Load(P_i) - Load(P_j)) \leq Load(P_i) - Load(P_j) \leq 0 \leq \alpha.\]

The role of \(\alpha\) and \(\beta\) (in definition 2) exchanges, when the ratio measure is used.

In the rest of the paper, \(S\) denotes a set of solutions for the considered problem, \(S^*\) denotes an optimal solution, \(L\) is the load of the subtrees, \(L^*\) is the load of an optimal solution, \(T_i\) denotes a subtree, \(C\) denotes the traveling cost, and \(K\) indicates the number of subtrees.

### 3 Complexity Analysis

We show BCMSTP is NP-hard for \(Q \geq 3\) and any given \(\alpha \geq 0\). We prove the NP-hardness by a reduction from CMSTP, which itself is NP-hard [42, 43]. The complexity of CMSTP depends on the capacity \(Q\). This problem is solvable in polynomial time if \(Q = 2\) [42]. Also, it is solvable in polynomial time, if vertices have 0, 1 demands and \(Q = 1\) [42]. Although, it is NP-hard if vertices have 0, 1 demands, \(Q = 2\) and all edge weights are 0 or 1 [42]. Also, it remains NP-hard for any \(Q \geq 3\) [42]. Evenmore, its geometric version, in which the metric space on the edges is the Euclidean metric, remains NP-hard [43]. Camerini et al. [44, 45] showed that many variants of this problem have the same complexity.
To show the NP-hardness of BCMSTP, we show its decision version is NP-complete [42]. Assume \( I_\alpha \) for \( \alpha \geq 0 \) is a decision problem that decides whether a feasible solution exists for the considered problem whose cost is at most a given bound \( D \). Theorem 1 shows that \( I_\alpha \) is NP-complete for each \( \alpha \geq 0 \) and \( Q \geq 3 \). The results are true when the ratio measure (Inequality (2) with \( 0 \leq \beta \leq 1 \)) is used instead of the range measure.

**Theorem 1** For every \( \alpha \geq 0 \) and \( Q \geq 3 \), \( I_\alpha \) is NP-complete.

**Proof** We show NP-completeness of \( I_\alpha \) by a reduction from CMSTP. As we mentioned in the last section, CMSTP is NP-hard for \( Q \geq 3 \). Let \( I' \) be an instance of this problem (with capacity constraint \( Q \), cost \( c_{ij} > 0 \) on each edge \((i, j)\), number of terminals \( n \), number of subtrees \( K' = \left\lceil \frac{n}{Q} \right\rceil \), and a bound \( D ) \) which decides whether a feasible solution exists whose cost is at most \( D \). Clearly, \( I' \) is NP-complete.

Let \( \Delta = \max\{c_{ij} : i, j \in \{0, 1, ..., n\}\} \), we define an instance \( I_\alpha \) of BCMSTP with capacity constraint \( Q \), number of terminals \( n' = K'Q \), number of subtrees \( K' \), cost \( c'_{ij} = c_{ij} \) on each edge \((i, j)\) where \( i, j \in \{0, 1, ..., n\} \) and cost \( c'_{ij} = c'_{ji} = \Delta \) on each edge where \( i \in \{n + 1, ..., n'\} \), and a bound \( D + \Delta(n' - n) \). Indeed, we add \( n' - n \) auxiliary nodes to the set of terminals in \( I' \) that are at distance \( \Delta \) from each other and the others. Evidently, the weights on the edges obey the triangle inequality.

Let \( x = \{T_1, T_2, ..., T_K\} \) be a solution for \( I' \) with loads \( L_1, L_2, ..., L_K \), whose cost is smaller than \( D \). We add \( Q - L_i \) nodes to the \( i \)-th subtree. Since the augmented nodes are at distance \( \Delta \) from each other and the others, the augmented length is \( \Delta(n' - n) \). So, we obtain a solution for \( I_\alpha \). Now, let \( x \) be a solution for \( I_\alpha \) whose cost is smaller than \( D + \Delta(n' - n) \). We remove the augmented nodes, so we obtain a feasible solution for \( I' \) whose cost is smaller than \( D \). This proves the NP-completeness of \( I_\alpha \), for each \( \alpha \geq 0 \).

4 Load Balancing

In BCMSTP, we require a set of capacitated subtrees dangling from the root. So, a solution has two portions: the number of nodes that each subtree receives and the choice of them. The number of the allocated nodes should be balanced. A solution is balanced if its balanced range is the least possible among the balanced ranges of all feasible solutions (see definition 2). We construct an algorithm to find balanced allocation of the nodes to the subtrees. We show this allocation, in some sense, is the fairest. First, we define the fairest solution.

We describe a partial order (identified with \( \prec \)) on the class of the allocated load vectors. For the sorted (in non-decreasing order) load vectors \( l' = (l'_1, l'_2, ..., l'_K) \) and \( l = (l_1, l_2, ..., l_K) \), we assume \( R_l = (l'_K - l'_1, ..., l'_2 - l'_1, 0) \), \( R_l = (l_K - l_1, ..., l_2 - l_1, 0) \). We say \( l' \prec l \), iff \( R_l \) is smaller than \( R_l \) in lexicographical order, that is, \( R_l = R_l \) or there is an index \( j \) for which \( l'_j < l_j \) and \( l'_j - l'_1 = l_i - l_1 \) for all \( i < j \). If \( l' \prec l \) and \( l \prec l' \), the vectors \( l \) and \( l' \) are called to be equivalent. Thus, the equivalence classes contain a total order. The minimal equivalence class under \( \prec \) contains the fairest allocations.

**Definition 3** A solution \( S \) for BCMSTP with the allocated loads \( L_S \) is the fairest solution if and only if \( L_S \) is the fairest load vector.

Definition 3 reveals that the fairest solution is indeed balanced since its balanced range is the least possible among the balanced ranges of all the feasible solutions. However, in general, the balanced solution is not the fairest. Here, we restate the algorithm proposed by the authors in [40] to find balanced loads (see Algorithm 1).

Assume each terminal corresponds to an object (with a volume of 1) in the set \( O = \{o_1, ..., o_n\} \) and each subtree corresponds to a bin (with capacity \( Q \)) in the set \( M = \{M_1, ..., M_K\} \). We desire to assign each object to a bin, and the found solution \( L^* = (L^*_1, ..., L^*_K) \) determines the number of objects each bin receipts. Indeed, \( L^* \) is the allocated loads to the \( K \) subtrees. An assignment is a function \( L : O \rightarrow M \) so that \( L \) assigns each object \( o_j \) to a bin in \( M \). In an assignment \( L \), the degree of each bin is the number of objects assigned to it.

**Algorithm 1** Load balancing algorithm

\begin{verbatim}
input : A set of similar objects \( O = \{o_1, ..., o_n\} \) that should be assigned to the set of bins \( M = \{M_1, ..., M_K\} \).
output : Optimum assignment \( L^* \).

Start
Assign \( \left\lceil \frac{n}{Q} \right\rceil \) objects to every bin.
Allocate an object to each bin in the set \( \{M_{K-(n-\left\lceil \frac{n}{Q} \right\rceil)+1}, ..., M_{K-1}, M_K\} \).
Return degrees of the bins: \( L^* = (L^*_1, ..., L^*_K) \).

End
\end{verbatim}
Theorem 2

S

The proof is the same to the proof of Lemma 1 in [40].

Proof

In the rest of the paper, the sequence \( L_1^* \leq L_2^* \leq \ldots \leq L_K^* \) denotes the balanced loads. Two possible cases are: \( L_K^* = L_1^* + 1 \), or \( L_K^* = L_1^* \). We suppose \( L_K^* \geq 3 \) and search for a set of subtrees with balanced loads.

5 BCMSTP equipped with general metrics

Our algorithm for BCMSTP relies on a method called route first-cluster second [46, 24]. We propose a special partitioning procedure that obtains the fairest solution, in this case. Algorithm 2 presents this procedure in detail, which we refer to it as \( \mathbb{3} \).

Algorithm 2 \( \mathbb{3} \)

\[ \text{input : BCMSTP with balanced loads } L^*. \]
\[ \text{output : } K \text{ balanced rooted subtrees.} \]

\[
\begin{align*}
\text{Start} \\
\text{Find a rooted minimum spanning tree } \text{MST}(V) \text{ of the graph } G = (V, E). \text{Let } \tau := (r, v_1, v_2, \ldots, v_n, r) \text{ be an Eulerian tour of } \text{MST}(V). \\
\text{for } i = 1 \text{ to } L_1^* \text{ do} \\
\quad \text{Start at } v_i \text{ and identify the forthcoming subtrees: } \\
\quad S_i = (T_i^1 = (r, v_i, v_{i+1}, \ldots, v_i+L_{K_i}^*-1)), T_i^2 = (r, v_i+L_{K_i}^*, \ldots, v_i+L_{K_i}^*-1), \ldots \\
\text{Find the total cost } C(S_i) = \sum_{j=1}^K C(T_j^i). \\
\text{end for} \\
\text{Restore the solution } S_p := \{ T_1^p, T_2^p, \ldots, T_K^p \}, 1 \leq p \leq L_1^* \text{ having the smallest total cost } C(S_p). \\
\text{End}
\end{align*}
\]

Let \( C_{\text{BCMSTP}} \) be the cost of an optimal solution for BCMSTP. A lower bound for \( C_{\text{BCMSTP}} \) is given in Lemma 2. We remind that \( c_{rv} \) is the edge cost between \( r \) and \( v \).

Lemma 2 \( C_{\text{BCMSTP}} \geq \sum_{v \in V} \frac{c_{rv}}{L_K} \).

Proof

Suppose \( S^* := \{ T_1^*, T_2^*, \ldots, T_K^* \} \) is an optimal solution for BCMSTP and \( T_i^* \in S^* \) is a subtree with cost \( C(T_i^*) \) and let \( c_{rv}^{\text{max}} = \max \{ c_{rv} : v \in T_i^* \} \). Since \( d(T_i^*) = \sum_{v \in (\bar{v}, r) \in T_i^*} 1 \leq L_K^* \), we have:
\[
C(T_i^*) \geq c_{rv}^{\text{max}} = \frac{\sum_{v \in (\bar{v}, r) \in T_i^*} 1}{L_K} c_{rv}^{\text{max}} \geq \frac{\sum_{v \in (\bar{v}, r) \in T_i^*} 1}{L_K} c_{rv} \geq \sum_{v \in T_i^*} \frac{c_{rv}}{L_K}.
\]

Summing over all the subtrees in \( S^* \), we obtain:
\[
C_{\text{BCMSTP}} = \sum_{i=1}^K C(T_i^*) \geq \sum_{v \in V} \frac{c_{rv}}{L_K}.
\]

\( \square \)

Theorem 2

The total traveling cost of \( S_p \) satisfies \( C(S_p) \leq (3 - \frac{1}{L_1^*})C_{\text{BCMSTP}}. \)

Proof

Assume \( C(\tau) \) is the cost of the Eulerian tour \( \tau \) of the minimum spanning tree. We observe that each vertex \( v \in V \setminus \{ r \} \) emerges at most once as the initial node of a subtree in all the iterations. Thus, every edge \( (r, v) \) emerges at most once during the iterations. When \( v \in \tau \) emerges as the initial node of a subtree, the edge \( (u, v) \in \tau \) does not appear in that solution. If \( v \in \tau \) is not the initial node of any subtree, \( (u, v) \in \tau \) appears in all the iterations. An upper bound for the total cost of the found solutions is:
\[
\sum_{j=1}^{L_1^*} \sum_{i=1}^K C(T_i^j) \leq (L_1^* - 1)C(\tau) + \sum_{v \in V} c_{rv}.
\]

The right side of Inequality (3) is an upper bound for the total cost since we have added \( c_{rv} + c_{uv} - c_{uv} \) to the right-hand side when the edge \( (u, v) \in \tau \) emerges in all the iterations. We see by the triangle inequality that
Since $S_p$ is the cheapest among the others, we obtain:

$$L_1^* C(S_p) \leq \sum_{j=1}^{L_1^*} \sum_{i=1}^{K} C(T_i^j) \leq (L_1^* - 1) C(\tau) + \sum_{v \in V} c_{rv}.$$

Suppose $L_K^* = L_1^*$. Thus,

$$C(S_p) \leq \sum_{v \in V} c_{rv} L_1^* \leq (2 - \frac{1}{L_1^*}) C_{BCMSTP} = (3 - \frac{1}{L_1^*}) C_{BCMSTP}.$$

The third inequality has resulted from Lemma 2.

Now, suppose $L_K^* = L_1^* + 1$.

$$C(S_p) \leq (1 - \frac{1}{L_1^*}) C(\tau) + \sum_{v \in V} c_{rv} L_1^* \leq (2 - \frac{1}{L_1^*}) C_{BCMSTP} + L_1^* \left( \sum_{v \in V} c_{rv} \right) \leq (3 - \frac{1}{L_1^*}) C_{BCMSTP}.$$

where we have used the Lemma 2.

We see $3 - \frac{1}{L_1^*} \leq 3 - \frac{1}{Q}$ since $L_1^* \leq Q$. So, the proposed algorithm provides $3 - \frac{1}{Q}$ factor of approximation for BCMSTP. In the following, we consider BCMSTP in the 2d–Euclidean metric spaces and produce a better approximation algorithm.

### 6 BCMSTP equipped with 2d–Euclidean metrics

First, we study CMSTP in the plane and provide a PTAS for this problem. Then, we use a similar technique to provide a factor $2 + \frac{1}{L_1^*} + \epsilon$ approximation algorithm for BCMSTP for every given $\epsilon > 0$. To prove the performance ratios of the algorithms, we need to find an upper bound for the minimum spanning tree (MST) in the plane. This is presented in the next section.

#### 6.1 Approximation of MST in $\mathbb{R}^2$

We use a technique similar to that of given in [47] to find an upper bound for the MST. Let $C(MST(V))$ be the cost of the minimum spanning tree of a complete graph defined on the set $V$, $c_{rv}^{\text{max}} = \max \{c_{rv} : v \in V\}$, and $\bar{c}_{rv} = \sum_{v \in V} c_{rv} / n$, we have the following theorem.

**Theorem 3** $C(MST(V)) \leq 2 \sqrt{\pi n c_{rv}^{\text{max}} \bar{c}_{rv}}$.

**Proof** We partition the circle of radius $c_{rv}^{\text{max}}$ into $4h$ equal sectors. We use the boundaries of the sectors to construct two star-shaped trees partitioning the circle (see Fig. 1). We convert each tree into a spanning tree by a double connection of minimal length from each point; the sum of these double connections is less than $2(2\pi/4h)c_{rv} = \pi c_{rv}/h$ due to $a \leq c_{rv}, \sin \theta \leq c_{rv}\theta$ (see Fig. 1). Hence, the sum of the costs of the trees is less than:

$$\pi n \bar{c}_{rv} \frac{1}{h} + 4hc_{rv}^{\text{max}}$$

and we conclude that

$$C(MST(V)) \leq \pi n \bar{c}_{rv} \frac{1}{2h} + 2hc_{rv}^{\text{max}}.$$ 

By taking $h$ equal to the value minimizing the right-hand side and rounding up i.e. $h = \left\lceil \sqrt{\pi n \bar{c}_{rv}/4c_{rv}^{\text{max}}} \right\rceil$, we obtain the desired result.

$\Box$
6.2 PTAS for CMSTP equipped with 2d–Euclidean metrics

We use the tour partitioning heuristic proposed in [24] (see Lemma 3) that is represented below by $\Gamma$.

**Lemma 3** There exists an approximation algorithm for CMSTP with performance ratio of $3 - \frac{2}{Q}$.

**Proof** See the proof in [24].

Using the algorithm $\Gamma$, we construct PTAS for the 2d–Euclidean CMSTP and refer to it as $A(\Gamma)$. The main idea of the algorithm is to separate the nodes into two parts: in the interior part, it employs $\Gamma$ and in the exterior part, it finds an optimal solution. The idea is to separate the nodes in a way that the optimal solution in the exterior part could be found in polynomial time with respect to $n$ (number of nodes). Khachay et al. [48] used a similar technique to find PTAS for the capacitated vehicle routing problem in Euclidean space.

Let $S_{CMST}^e := \{T_1^{CMST}, T_2^{CMST}, ..., T_k^{CMST}\}$ be a set of optimal subtrees for CMSTP with optimal cost $C(S_{CMST}^e)$. Haimovich and Rrinnooy Kan [47] proved that $n \bar{C}_{rV} = \frac{1}{Q}C_{rV}$ is a lower bound for the optimal cost: $\frac{n}{Q} \bar{C}_{rV} \leq C(S_{CMST}^e)$. Altinkemer and Gavish [24] developed an upper bound of $2 C(MST(V)) + \frac{n}{Q} \bar{C}_{rV}$ for the optimal cost and derived $(3 - \frac{2}{Q})$–approximation ratio using this upper bound. Based on the results of the authors in [24, 47], we derive the following inequalities:

$$\frac{n}{Q} \bar{C}_{rV} \leq C(S_{CMST}^e) = \sum_{i=1}^{k} C(T_i^{CMST}) \leq 2C(MST(V)) + \frac{n}{Q} \bar{C}_{rV}, \quad (4)$$

The algorithm $A(\Gamma)$ is presented below (see Algorithm 3).

**Algorithm 3 A(\Gamma)**

**Input**: An instance of CMSTP.

**Output**: A feasible solution for CMSTP.

**Start**

1. Enumerate terminals by decreasing their distance from the root $c_{v_1} \geq c_{v_2} \geq ... \geq c_{v_n}$.

2. Take the set $V(l) = \{v_1, v_2, ..., v_{l-1}\}$ of outside nodes. We will later determine the value of $l$, and will show that it is independent of $n$.

3. Apply $\Gamma$ to the set of inside nodes $V \backslash V(l)$. Denote the obtained solution by $S_{CMST}(V \backslash V(l))$.

4. Find the optimal solution for CMSTP defined on the set of outside nodes $V(l)$ and the same root. Denote the found solution by $S_{CMST}(V;l)$. Return $S_{CMST}(V) := S_{CMST}(V \backslash V(l)) \cup S_{CMST}(V;l)$.

**End**

**Theorem 4** The algorithm $A(\Gamma)$ is a PTAS for the Euclidean CMSTP.

**Proof** We show that for any $\epsilon > 0$, the relative error $e^{A(\Gamma)}(V)$ satisfies $e^{A(\Gamma)}(V) \leq \epsilon$. First, we find an upper bound for $e^{A(\Gamma)}(V)$.

Let $I$ be a given instance of CMSTP with an optimal solution $S_{CMST}^o(V)$. Consider the circle with radius $c_l = c_{v_1}$ centered at the root (we will later determine $l$). We connect the nodes in $V(l)$ incident to the edges between $V(l)$ and $V \backslash V(l)$ directly to the root (see Fig. 2). Let $S_{CMST}^e(V(l)), S_{CMST}^e(V \backslash V(l))$ be the optimal solutions of the problems defined on the sets $V(l)$ and $V \backslash V(l)$, respectively, and let $C(S_{CMST}^e(V(l))), C(S_{CMST}^e(V \backslash V(l)))$ be their respective costs. We have

$$C(S_{CMST}^e(V(l))) + C(S_{CMST}^e(V \backslash V(l))) \leq C(S_{CMST}^o(V)) + (l - 1)c_l.$$

From the $A(\Gamma)$ algorithm,

$$C(S_{CMST}(V)) = C(S_{CMST}(V(l))) + C(S_{CMST}(V \backslash V(l))).$$

Since $C(S_{CMST}(V \backslash V(l))) \geq \sum_{i=1}^{n} c_{v_i}$ (see Inequality (4)), we get

$$C(S_{CMST}(V)) \leq C(S_{CMST}(V(l))) + C(S_{CMST}(V \backslash V(l))) + \left(C(S_{CMST}(V \backslash V(l))) - \sum_{i=1}^{n} c_{v_i} \frac{1}{Q}\right).$$
Thus, we have
\[ e^{A(P)}(V) = \frac{C(S_{CMST}(V)) - C(S^*_{CMST}(V))}{C(S_{CMST}(V))} \]
\leq \frac{C(S^*_{CMST}(V)) + C(S^*_{CMST}(V\setminus l)) + C(S_{CMST}(V\setminus V(l))) - \frac{\sum_{l=1}^{n} c_{rv_i}}{Q} - C(S^*_{CMST}(V))}{C(S^*_{CMST}(V))} \]
\leq \frac{(l-1)q + C(S_{CMST}(V\setminus V(l))) - \frac{\sum_{l=1}^{n} c_{rv_i}}{Q}}{C(S^*_{CMST}(V))}.

From Inequality (4) and Theorem 3, we have:
\[ C(S^*_{CMST}(V)) \geq \frac{n}{Q} C_{rv}, \]
and
\[ C(S_{CMST}(V\setminus V(l))) \leq 2C(MST(V\setminus V(l))) + \frac{n}{Q} C_{rv}(V\setminus V(l)) \leq 4\sqrt{\pi n c_{max} C_{rv}} + \sum_{l=1}^{n} c_{rv_i}. \]

Thus, we obtain
\[ e^{A(P)}(V) \leq \frac{(l-1)q + C(S_{CMST}(V\setminus V(l))) - \frac{\sum_{l=1}^{n} c_{rv_i}}{Q}}{\frac{n}{Q} C_{rv}} \]
\leq Q(l-1)q + 4\sqrt{\pi n c_{max} C_{rv}} + \sum_{l=1}^{n} c_{rv_i},
\leq Q(l-1)q + 4\sqrt{\pi} \sqrt{\frac{c_l}{\sum_{l=1}^{n} c_{rv_i}}}.

(5)

Now, we choose \(l\) such that Inequality (5) is less than \(\epsilon\). We see for large values of \(l\), the right side of Inequality (5) is smaller than \(\epsilon\) since the algorithm finds an optimal solution in the exterior part. However, its running time is exponential concerning \(l\). We need to choose \(l\) in a way that CMSTP defined on the set of outside nodes could be solved in polynomial time concerning \(n\). Suppose that \(l\) is selected to be the smallest number (from 1 to \(n\)) for which Inequality (5) is less than \(\epsilon\), for some fixed \(\epsilon > 0\) or, in the other direction, without loss of generality, assume \(l\) is selected to be the largest number (from 1 to \(n\)) for which Inequality (5) is larger than \(\epsilon\). We obtain an upper bound on \(l\) independent of \(n\). To do this, we put \(s_h = \sqrt{\frac{\sum_{l=1}^{n} c_{rv_i}}{2}}\), \(A = Q\), \(2B = 4Q\sqrt{\pi}\) and investigate the lower bound of inequality solutions:
\[ Ah s_h^2 + 2B s_h - \epsilon > 0, \quad (h = 1, ..., l-1). \]

(6)

Hence, \(s_h\) must be larger than the positive root of the quadratic equation defined by the left-hand side of Inequality (6) for \(h = 1, ..., l-1:\)
\[ s_h > \frac{-2B + \sqrt{4B^2 + 4Ah\epsilon}}{2Ah}, \quad (h = 1, ..., l-1). \]

Thus,
\[ s_h^2 > \left(\frac{-2B + \sqrt{4B^2 + 4Ah\epsilon}}{2Ah}\right)^2, \quad (h = 1, ..., l-1) \]
\[ s_h^2 = \frac{c_l}{\sum_{l=1}^{n} c_{rv_i}} = \left(\frac{-2B + \sqrt{4B^2 + 4Ah\epsilon}}{2Ah}\right)^2 \]
\[ \geq \frac{\epsilon}{Ah} + \frac{B^2}{A^2 h^2} - \frac{4B\sqrt{4B^2 + 4Ah\epsilon}}{4A^2 h^2} \]
\[ \geq \frac{\epsilon}{Ah} - \frac{4B\sqrt{\pi}}{2A^3 h^3}, \quad (h = 1, ..., l-1) \]

Consequently,
\[ 1 \geq \sum_{h=1}^{l-1} s_h^2 \geq \sum_{h=1}^{l-1} \frac{c_{rv_i}}{\sum_{l=1}^{n} c_{rv_i}} \geq \epsilon \sum_{h=1}^{l-1} \frac{1}{Ah} - \frac{4B\sqrt{\pi}}{2} \sum_{h=1}^{l-1} \frac{1}{\sqrt{A^3 h^3}}. \]
Note that
\[
\sum_{h=1}^{l-1} \frac{1}{h} > \int_1^{l-1} \frac{1}{z} \, dz = \ln(l - 1),
\]
and
\[
\sum_{h=1}^{l-1} \frac{1}{\sqrt{A^3 h^3}} < \frac{1}{A^{3/2}} \int_1^{l} \frac{1}{z^{3/2}} \, dz < \frac{2}{A^{3/2}}.
\]
We conclude that
\[
\frac{\varepsilon}{A} \ln(l - 1) - \frac{AB\sqrt{\pi}}{A^{3/2}} < 1,
\]
i.e.
\[
l < \epsilon^4 \left(1 + \frac{4B\sqrt{\pi}}{A^{3/2}}\right) + 1.
\]
It follows that the computational effort done on seeking an optimal set of subtrees for the \(l - 1\) outside nodes does not rely on \(n\). Moreover, the other steps of the algorithm can be done in polynomial time. Thus, we prove that \(A(I)\) is PTAS for CMSTP. Its running time relies on algorithm solving MST.

Since \(A\) and \(B\) are \(\Theta(Q)\), the algorithm \(A(I)\) is PTAS for CMSTP and \(Q = o(\ln \ln n)\). Indeed, the running time of the algorithm is exponential with respect to \(l\). Since \(l\) is exponential with respect to \(Q\), the running time will be polynomial (concerning \(n\)) for \(Q = o(\ln \ln n)\).

6.3 BCMSTP equipped with 2d–Euclidean metrics

In this section, we use a similar technique to find a factor \(2 + \frac{1}{l^3} + \epsilon\) approximation algorithm for BCMSTP for every given \(\epsilon > 0\). We use the algorithm given in Section 5 and represent it by 3. As we have seen, 3 provides a solution \(S_p\) with an approximation factor of \(3 - \frac{1}{l^3}\). We see that
\[
C(S_p) = \sum_{i=1}^{i=K} C(T_i^H) \leq 2C(MST(V)) + C_{rv} = 2C(MST(V)) + \frac{\sum_{i=1}^{i=K} c_{rv}}{L_1} = 2C(MST(V)) + \frac{n}{L_1} \tilde{C}_{rv}.
\]
where \(C_{rv} = \sum_{i=1}^{i=K} c_{rv}\), and \(n\) is the number of terminal nodes. Furthermore, it follows from Lemma 2 that each optimal solution \(S^*_V = \{T_1^*, T_2^*, ..., T_k^*\}\) of BCMSTP satisfies
\[
C(S^*_V) = \sum_{i=1}^{i=K} C(T_i^*) \geq \frac{\sum_{i=1}^{i=K} c_{rv}}{L_k^*} = \frac{n}{L_k^*} \tilde{C}_{rv}.
\]
Thus, \(S^*_V\) satisfies the following relation:
\[
\frac{n}{L_k^*} \tilde{C}_{rv} \leq C(S^*_V) \leq 2C(MST(V)) + \frac{n}{L_1} \tilde{C}_{rv}.
\]
These inequalities are valid in any metric space, especially in Euclidean metrics. Algorithm 4 presents our algorithm which is referred to below by \(A(3)\).

**Theorem 5** The algorithm \(A(3)\) achieves \(2 + \frac{1}{l^3} + \epsilon\) factor of approximation for the Euclidean BCMSTP.

**Proof** To prove the theorem, we show for any \(\epsilon > 0\), the relative error \(e^{A(3)}(V)\) of \(A(3)\) satisfies the inequality \(e^{A(3)}(V) \leq 1 + \frac{1}{l^3} + \epsilon\). First, we find an upper bound for \(e^{A(3)}(V)\).

We provide a solution for BCMSTP on the set \(V(l') \cup \{r\}\) using the loads \(L_k^*, L_{k-1}^*, ..., L_1^*\), and also obtain a solution for this problem on the set \(V \setminus (l') \cup \{r\}\) using the loads \(L_1^*, L_2^*, ..., L_{k-1}^*\). Let \(I\) be an instance of BCMSTP and \(S^*_V\) be an optimal solution. Consider the circle with radius \(c_i = c_{rv}\) centered at the root (we will later determine \(l\)).

To provide a solution for BCMSTP on the set \(V(l') \cup \{r\}\), perform in steps as follows:

1. Remove the edges between the nodes in \(V(l) = \{v_1, v_2, ..., v_l\}\) and \(V \setminus (l)\).
Algorithm 4 A(3)

input : An instance of BCMSTP.
output : A feasible solution for BCMSTP.

Start

Enumerate terminals by decreasing their distance from the root

\[ c_{rv_1} \geq c_{rv_2} \geq \ldots \geq c_{rv_n}. \]

Take the set \( V'(l') = \{v_1, v_2, \ldots, v_l, v_{l+1}, \ldots, v_{l'}, v_{l'+1}, \ldots, v_p\} \), \( l' < 3l \), of outside nodes. We will later determine the values of \( l \) and \( l' \), and will show that they are independent of \( n \).

Apply 3 to the set of inside nodes \( V \setminus V'(l') \). Use the loads \( L_1^*, L_2^*, \ldots, L_{l'-1}^* \), where \( 1 \leq k \leq K \) and \( \sum_{i=1}^{k-1} L_i^* = n-l' \). Denote the solution by \( S_{V \setminus V'(l')} \).

Find the optimal solution for BCMSTP defined on the set of outside nodes \( V'(l') \) and the same root. Use the loads \( L_K^*, L_{K-1}^*, \ldots, L_1^* \) for the subtrees. Denote the solution by \( S_{V'(l')}^* \).

Return \( S_V = S_{V \setminus V'(l')} \cup S_{V'(l')}^* \).

End

2. \( S_{V'(l')}^* = \emptyset \). If the load/order of a subtree \( T' \) (in \( V(l) \)) is \( L_1^* \), take \( S_{V'(l')} = S_{V'(l')}^* \cup T' \). Without loss of generality, we assume loads of the subtrees in \( V(l) \) are at most \( L_1^* \).

3. Connect the nodes in \( V(l) \) incident to the removed edges (i.e. the edges between \( V(l) \) and \( V \setminus V(l) \)) to the \( i^{th} \) node in the set \( V'(l') = \{v_{i+1}, v_{i+2}, \ldots, v_{l'}, v_{l'+1}, \ldots, v_p\} \). Some of the nodes in \( V'(l') \) may not be connected to any subtree in \( V(l) \) (see Fig. 3b). We see \( c_{rv_i} \leq c_{rv_{i+1}} + c_{rv_j} + c_l \) for \( v_i \in \{v_1, \ldots, v_l\} \) and \( v_j \in \{v_{i+1}, v_{i+2}, \ldots, v_{l'}\} \). Since \( c_{rv_i} \leq c_{rv_{i+1}} + c_l \) (see Fig. 3c), we get \( c_{v_i,v_j} \leq \sum c_{v_i,v_j} + 2c_l \).

4. Connect the nodes in \( V'(l') \) to each other as it is shown in Fig. 3b. We see \( c_{v_i,v_h} \leq 2c_l \) for \( g,h \in \{l+1, l+2, \ldots, l'\} \).

5. Find an Eulerian tour \( \tau_{V'(l')} \) spanning the vertices in the found tree by doubling and shortening the edges. Note that, there are smaller than \( L_1^* \) nodes of \( V(l) \) between two consecutive nodes of \( V'(l') \) in \( \tau_{V'(l')} \), since loads of the subtrees in \( V(l) \) are at most \( L_1^* \).

6. Let \( (v'_1, v'_2, \ldots, v'_{l'}) \) be an order of the nodes in \( \tau_{V'(l')} \), we obtain subtrees with loads \( L_{K-l}^*, L_{K-1}^*, \ldots, L_1^* \) as follows:

\[
T_{V(l')}^{1} := \{v'_1, v'_2, \ldots, v'_{L_{K-l}^*}\}, \quad T_{V(l')}^{2} := \{v'_{L_{K-l}^*+1}, v'_{L_{K-l}^*+2}, \ldots, v'_{L_{K-l}^*+L_{K-1}^*}\}, \quad \ldots, \quad T_{V(l')}^{V'(l')} := \{v'_{l'-L_{1}^*+1}, \ldots, v'_{l'}\}.
\]

7. Connect each subtree \( T_{V(l')}^{i} \) to the root using one of the nodes in \( V'(l') \) and incident on it. This can be done, since there are smaller than \( L_1^* \) nodes (of \( V(l) \)) between two consecutive nodes of \( V'(l') \) in \( \tau_{V'(l')} \).

Thus, we obtain a solution \( S_{V'(l')}^* \) of the problem on the set \( V(l') \) whose cost is at most \( 2C(S^*(V))_{V'(l')} + 9lc_l \).

To provide a solution for the problem on the set \( V \setminus V'(l') \), remove the edges between \( V'(l') \) and \( V \setminus V(l') \).

We see there are at most \( K \) subtrees danging from the root and inside the circle with radius \( c_l \) whose loads are \( L_1, L_2, \ldots, L_K \). Without loss of generality, we assume \( L_1 \geq L_2 \geq \ldots \geq L_K \). To construct a solution for the problem on \( V \setminus V'(l') \), we perform as follows. We choose the \( k-1 \) largest subtrees and connect the nodes of the other trees to the \( i^{th} \) subtree (1 \( \leq i \leq k-1 \)) until its load becomes \( L_i^* \) (see Fig. 4). The sum of the length of these edges is at most \( 6l_2 \), since \( l' < 3l \). Let \( S_{V \setminus V'(l')}^* \) be the optimal solutions for the problems defined on the sets \( V(l'), V \setminus V(l') \), respectively and let \( C(S_{V \setminus V'(l')}^*), C(S_{V \setminus V'(l')}^*) \) be their respective costs. We have

\[
C(S_{V \setminus V'(l')}^*) + C(S_{V \setminus V'(l')}^*) \leq 2C(S_{V}^*) + 15lc_l.
\]

Let \( C(S_V) \), \( C(S_{V \setminus V'(l')} \) be the costs of the solutions \( S_V, S_{V \setminus V'(l')} \), respectively. By construction, for any A(3)-based approximation algorithm,

\[
C(S_V) = C(S_{V'}^*) + C(S_{V \setminus V'(l')}^*).
\]

Since \( C(S_{V \setminus V'(l')}^*) \geq \sum_{i=t+1}^{n} \frac{c_{rv_i}}{L_K} \) (from Lemma 2), we get

\[
C(S_V) \leq C(S_{V'}^*) + C(S_{V \setminus V'(l')}^*) + \left( C(S_{V \setminus V'(l')}^*) - \sum_{i=t+1}^{n} \frac{c_{rv_i}}{L_K} \right).
\]
Thus, we have
\[ e^{A(3)}(V) = \frac{C(S_V) - C(S_V')} {C(S_V')} \leq \frac{C(S_V') + C(S_V \setminus V')} {C(S_V')} - \sum_{e = 1}^{n} \frac{c_{rv_e}} {L_k} - C(S_V') \]
\[ \leq \frac{2C(S_V') + 15l_{cl} + C(S_V \setminus V')} {C(S_V')} - \sum_{e = 1}^{n} \frac{c_{rv_e}} {L_k} \]
\[ \leq \frac{C(S_V') + 15l_{cl} + C(S_V \setminus V')} {C(S_V')} - \sum_{e = 1}^{n} \frac{c_{rv_e}} {L_k} \].

Since \( C(S_V') \geq \frac{n} {L_k} C_{rv}, \ C(S_V \setminus V') \leq 2C(MST(V)) + \sum_{e = 1}^{n} \frac{c_{rv_e}} {K^2}, \) and \( C(MST(V)) \leq 2\sqrt{\pi}c_{max}C_{rv}, \) we obtain
\[ e^{A(3)}(V) \leq 1 + \frac{15l_{cl}} {L_k} C_{rv} + \frac{C(S_V \setminus V')} {L_k} - \sum_{e = 1}^{n} \frac{c_{rv_e}} {L_k} \leq 1 + \frac{L_k^* 15l_{cl}} {\sum_{i = 1}^{n} c_{rv_i}} + L_k^*(4\sqrt{\pi}c_{max}C_{rv}) + \frac{1} {L_1^*} (7) \]
\[ = 1 + \frac{1} {L_1^*} + 4L_k^* \sqrt{\pi} \leq \frac{c_{l}} {\sum_{i = 1}^{n} c_{rv_i}} + 4L_k^* \sqrt{\pi} \sum_{i = 1}^{n} c_{rv_i}. \] (8)

Due to \( L_k^* = L_1^* + 1, \) Inequality (7) is hold. In the case that \( L_k^* = L_1^* \), the term \( \frac{1} {L_1^*} \) would be removed from Equation (8).

Now, we choose \( l \) in a way that (8) is less than \( 1 + \frac{1} {L_1^*} + \epsilon \). We obtain an upper bound on \( l \) independent of \( n \). To do this, we put \( s_h = \sqrt{\sum_{i = 1}^{n} c_{rv_i}} \), \( A = 15L_k^* \), \( 2B = 4L_k^* \sqrt{\pi} \) and investigate the lower bound of inequality solutions:
\[ Ahs_h^2 + 2Bs_h - \epsilon > 0, \quad (h = 1, \ldots, l) \].

A similar method as it is given in the proof of Theorem 4 shows that
\[ l < e^{\frac{1} {2}(1 + \frac{4n\epsilon}{A^2})}. \]

We choose the smallest \( l' \), \( 2l \leq l' < 3l \), so that \( \sum_{i = 1}^{K} l'^* = l', \ 1 \leq k \leq K \). It follows that finding an optimal set of subtrees for the \( l' \) outside nodes does not depend on \( n \). Moreover, the other steps of \( A(3) \) can be done in polynomial time concerning \( n \). We conclude the heuristic \( A(3) \) is a polynomial time approximation algorithm for BCMSTP.

\( \square \)

Since \( L_k \leq Q \) and \( A \) and \( B \) are \( \Theta(L_k^*) \), the algorithm \( A(3) \) is an approximation algorithm for BCMSTP and \( Q = o(ln ln n) \).

**Conclusion**

In this paper, the balanced capacitated minimum spanning tree problem (BCMSTP) is considered, and an effort is made to design two approximation algorithms. We proposed a factor \( 3 - \frac{1} {L_1^*} \) approximation algorithm that finds the fairest solution. In the Euclidean metrics, we provided an improved algorithm that achieves \( 2 + \frac{1} {L_1^*} + \epsilon \) factor of approximation. In addition to its applications in practice, BCMSTP on Euclidean metrics is compelling theoretically. Most of the geometric problems admit PTAS in Euclidean metrics, so an interesting question is whether the Euclidean BCMSTP has PTAS, which remains as an open problem. In this paper, we presented PTAS for the \( 2d \)-Euclidean CMSTP. Future work could be to improve this algorithm.

**Acknowledgements** The authors thank Amirkabir University of Technology for facilities.

**Conflict of interest** : The authors declare that they have no conflict of interest.


References
Appendix A

We present a mathematical programming model for BCMSTP. We presume the root node corresponds to the node 0 and \( N = \{1, 2, \ldots, n\} \) is the set of terminals. Moreover, we assume demand of the root is zero. Let \( x_{ij}^k \) be the binary variables on the edges \((i, j) \in E\) that decide whether the edge \((i, j)\) is presented in the \(k^{th}\) subtree. An edge \((i, j)\) is presented in the \(k^{th}\) subtree, iff \( x_{ij}^k = 1 \) and is not presented, otherwise. Let \( y_{ij}^k \) be the quantity that is carrying through \( i \) to \( j \) along the \(k^{th}\) subtree. The following is an integer programming formulation of BCMSTP:

\[
C_{opt} = \min \sum_{k=1}^{K} \sum_{i=0}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}^k
\]

\[
\sum_{k=1}^{K} \sum_{i=0}^{n} x_{ij}^k = 1 \quad j := 1, \ldots, n \quad (A.1)
\]

\[
\sum_{k=1}^{K} n y_{ij}^k = \sum_{k=1}^{K} n y_{ji}^k = 1 \quad j := 1, \ldots, n \quad (A.2)
\]

\[
\left| \sum_{j=1}^{n} y_{ij}^k - \sum_{j=1}^{n} y_{ij}^k \right| < \alpha \quad j := 1, \ldots, n, i := 1, \ldots, n, k := 1, \ldots, K \quad (A.3)
\]

\[
x_{ij}^k \leq y_{ij}^k \leq Q x_{ij}^k \quad j := 1, \ldots, n, i := 1, \ldots, n, k := 1, \ldots, K \quad (A.4)
\]

\[
\sum_{j=1}^{n} x_{0j}^k = 1 \quad k := 1, \ldots, K \quad (A.5)
\]

\[
y_{ij}^k \geq 0, x_{ij}^k \in \{0, 1\} \quad j := 1, \ldots, n, i := 1, \ldots, n, k := 1, \ldots, K \quad (A.6)
\]

The constraints of (A.1) ensure that each node \( j \in N \) is sourced by exactly one edge \((i, j)\) from some node \( i \in \{0\} \cup N \). Constraint set (A.2) implies that the cumulative flow going into every node \( j \) is one unit more than the cumulative flow coming out of that node. The loads of the subtrees should satisfy the balanced condition (1) (or (2)) for a parameter \( \alpha \) (or \( \beta \)), specified by the user. This is guaranteed by the constraints (A.3). Constraint set (A.4) implies that the flow on an activated (or used) edge will not exceed the capacity \( Q \). In this formulation, for a certain \( k \), there can be more than one subtree dangling from the root. In other words, what the model considers as a subtree dangling from the root would be the union of several such trees. Constraint set (A.5) is to fix this bug. Constraint set (A.6) implies that flows on all edges are nonnegative, and implies that an edge is either used or is not used. The cost of using an edge is fixed regardless of the volume of flow on the edge.
**Fig. 1** Approximation of minimum spanning tree (MST) in $\mathbb{R}^2$.

**Fig. 2** An optimal solution of the capacitated minimum spanning tree problem (CMSTP) defined on the set of nodes $V$ can be transformed into a solution for CMSTP defined on the set of nodes $V(l) \cup \{r\}$ and a solution for CMSTP defined on the set $V \setminus V(l)$ by deleting and adding some edges.

**Fig. 3** An optimal solution of the balanced capacitated minimum spanning tree problem (BCMSTP) defined on the set of nodes $V$ can be transformed into a solution for BCMSTP defined on the set of nodes $V(l') \cup \{r\}$.

**Fig. 4** An optimal solution of the balanced capacitated minimum spanning tree problem (BCMSTP) defined on the set of nodes $V$ can be transformed into a solution for BCMSTP defined on the set of nodes $V \setminus V(l')$. 

Fig. 1

Fig. 2

Fig. 3
Fig. 4