Jafar Fathali; Mehdi Zaferanieh
Stochastic queue core problem with an efficient length on a tree network

Kybernetika, Vol. 61 (2025), No. 2, 185–201

Persistent URL: `http://dml.cz/dmlcz/152987`

KYBERNETIKA — VOLUME *61* (2025), NUMBER 2, PAGES 185–201

STOCHASTIC QUEUE CORE PROBLEM WITH AN EFFICIENT LENGTH ON A TREE NETWORK

Jafar Fathali and Mehdi Zaferanieh

In this paper, we consider a stochastic queue core (SQC) problem on a tree network, aiming to identify a path P, called the core, in an $M/G/1$ environment system. Let T be a tree network, the SQC problem on T involves finding a core P, with an optimal length, that minimizes the total weighted travel time from all vertices to the core as well as the average response time to the customer demands. We assume that a mobile server traverses the core to provide services to customers, while customers move to their nearest vertex on the core to receive service. Some general properties of the SQC problem on the tree network are represented. Then a polynomial time algorithm is proposed to solve this problem.

Keywords: location theory, core, $M/G/1$ queue

Classification: 60K25, 90Bxx

1. INTRODUCTION

The Facility Location Problem (FLP) is a significant area in resource optimization management. Generally, the FLP models focus on determining optimal facility locations and assigning customers to these facilities while a cost objective function is minimized. A basic classification of the FLP includes single and multi-facility scenarios, fixed costs FLP, capacitated FLP, and covering problems [3]. A comprehensive study on strategies of FLP and related mathematical models has been conducted by Owen and Daskin [28]. They considered some advanced cases where locating facilities depend not only on current conditions but also on some additional criteria such as environmental factors and customer preferences, leading to uncertain or robust FLP optimization.

The extended shapes of FLP on graphs date back to the 1980s when Hedetniemi et al. [19], Morgan and Slater [24] and Slater [31] considered the tree and path shaped facilities rather than isolated points. The problem of finding the core of a tree network involves determining a path so that the total weighted distances from all customers located at vertices to the nearest vertex on the path is minimized, see [27, 36]. Morgan and Slater [24] developed an efficient linear time algorithm to find the core of a tree. Becker et al. [7] and Alstrup et al. [4] presented an $O(n \log n^2)$ time algorithm to find a core whose length is at most l on a weighted tree network. A branch-and-cut algorithm to find the

DOI: 10.14736/kyb-2025-2-0185

core on general networks has been presented by Avella et al. [5]. An interested extension of the core problem is k-core decomposition to find the largest subgraphs of a network in which each node has at least k neighbors in the subgraph, see [21].

Zaferanieh and Fathali [36] investigated a semi-obnoxious case of the core problem on a tree network where some of the vertices are desirable with positive weights and the others are undesirable with negative weights. They proved that when the weight of the tree network is negative, the core of the tree should be merely a single vertex, and when the weight of the tree is zero, there is a core vertex and probably some paths that are cores. In the classic p-median problem, customers travel to the nearest available facility, and the goal is finding the location of p facilities such that the total travel cost for customers is minimized, see [11, 35]. The semi-obnoxious p-median problem extends this concept by incorporating vertices with positive or negative weights, see [15].

The p-median problem is closely related to core and FLP problems. However, unlike core models where server facilities are on the core, the p-median problem involves a fixed number of p facilities distributed across the network to serve customers, see [16]. Kariv and Hakimi [20] proved that the p-median problem is NP-hard on general networks. They showed the p-median problem would be solved in $O(p^2n^2)$ running time on the tree networks, where n is the number of vertices. Tamir [32] improved the time complexity to $O(pn^2)$, and for the case $p = 1$, Goldman [18] developed a linear time algorithm. Additionally, for the case $p = 2$, Gavish and Sridhar [17] proposed an $O(n \log n)$ time algorithm. Beyond the distance criterion to locating facility in the network, some significant and important criteria such as security or proximity to the public transportation or energy hubs can be considered in practical p-median models, see [1, 37].

Stochastic queue p-median problem ($SQpM$) is a variation of the classic p-median problem that incorporates queueing theory and stochastic process. It involves locating a given number of facilities on a network to minimize the response average time to random request for service which occur according to independent Poisson processes, see [11]. Early work on the $SQpM$ problem has been introduced by Berman et al. [10]. They analyzed the $SQpM$ for the location of a single server on a network operating as $M/G/1$ queue. Chiu et al. [14] specialized Berman et al. [10] results for the case of a tree network. The $SQpM$ has been extended to locating the optimal location of a facility with p servers by Batta and Berman [6]. Finding the optimal location of p facilities with cooperation and without cooperation between servers have been investigated by Berman and Mandowsky [12] and Berman et al. [11], respectively. Wang et al. [34] considered the problem of locating facilities that are modeled as $M/M/l$ queuing systems. In this case, customers travel to the closest facilities where a constraint limits the maximum expected waiting times in all facilities. Berman and Drezner [8] generalized this model, by allowing facilities to host more than one server. There are some other researches on the queuing location models in the literature, for example see [23, 33, 13] and [30]. To read about the other extensions of the $SQpM$ problem, we refer the reader to Berman et al. [9].

Here, we propose the SQC problem on the tree networks. Providing service on tree-shaped facilities is practical, since customers should receive service in the least amount of time possible, and consequently many unnecessary paths and edges essentially are neglected, and the resulted graph practically reduces to a tree or pseudo-tree shaped facility, see [26]. The goal is to find the core P with an efficient length that minimizes the total of the average response time together with the average travel time from customers and the manufacturing cost of the core. When a service demand is requested by a customer, they should go to the closest vertex of the core to receive the service in a first in first out ($FIFO$) queue discipline. Then in response to the customers' demands, the server (if available) travels along the core to provide the requested demand. Since demands are assumed to arrive according to a homogeneous Poisson process and the service time distribution is general, see [9], then the model is an $M/G/1$ queuing system. Also, the demands are placed in a $FIFO$ infinite capacity queue to await service. In the real life, this problem can be applied to find the route of a mobile hospital. Mobile hospital is a medical center or a small hospital with necessary medical equipment that transported to the desired location in a short time and start operating, see [25]. Another application is vehicle routing for relief logistics, which requires strategic route planning and optimization to deliver critical supplies (e. g., food, emergency medical aid, and other essentials) to disaster-affected areas [22]. Due to disruptions and severe time constraints, especially in the immediate aftermath of a crisis, it is often impossible to assist all affected individuals within the first few days [29]. The proposed SQC model can thus provide decision-making support with practical managerial implications in such scenarios.

In the proposed SQC problem, the server moves only along the core instead of the entire network, due to cost constraints. The literature review on this problem is not extensive, where both server and customers travel to meet the customers' demands. Moshtagh et al. [27] considered the SQC problem on a tree network, where the edges of the tree are modeled with the $M/G/c/c$ state-dependent queues. Moshtagh et al. [26], according to the structure of these problems, represented some algorithms to find the core to evacuate the vehicles in an urban area. In this paper, we assume the edges of the tree network are modeled as operating general links rather than focusing only on roadway links. Hence, the server and customers can travel along the edges of the tree network with the given average values for their speeds.

The remainder of this paper is organized as follows. In Section 2, some frequently used notations and definitions are provided. Section 3 contains some properties and a sufficient condition to determining the SQC on the tree networks. In Section 3, an $O(n^3)$ time algorithm is proposed to solve the SQC problem. Illustration examples are provided in Section 4.

2. PROBLEM FORMULATION

Let $T = (V, E)$ be a tree network, where V is the set of vertices, $|V| = n$, and E is the set of edges. The customers' demands arise from the vertices of the tree network. Each vertex v_i generates an independent Poisson distributed stream of demands at an average rate of λ_i. Let P be the core of T and $T_u,\ u \in P$ be a sub-tree of T, containing all vertices $v \in T$, such that $d(u, v) \leq d(u_i, v)$ for all vertices $u_i \in P$. Table 1 contains a glossary of frequently used terms in this research.

Since the server travels along the path to respond to the requested demands, we should estimate the position of the server at different times. The probability that the server is located in a typical vertex $p \in P$ is indicated by $prob_P(\hat{u} = p)$ and may be

P	The core of the tree $T = (V, E)$
λ_i	The average rate of arrival demands of vertex i
$\lambda = \sum_{i=1}^{n} \lambda_i$	System-wide average rate of arrival demands
$w_i = \frac{\lambda_i}{\lambda}$	Fraction of demands originating in vertex i
$w_{T_i} = \sum_{v_j \in T_i} w_j$	The weight of branch T_i
p_i	ith vertex of the core P
$d(v_i, v_j)$	the length of the shortest path between vertices v_i and v_j
$d(P, v) = \min_{u \in P} d(u, v)$	The length of the shortest path between core P and vertex v
G_i	Non-travel related service time at demand i
$s_i(P)$	Service time of demand i
$TR_i(P)$	Total response time of demand i
$Q_i(P)$	The delay time that demand i wait in the queue
$\overline{G}$	The average non-travel service time
$\overline{d}_1(P)$	The total expected distance that customers travel to reach the core P
$\overline{d}_2(P)$	The total expected distance that service provider travels along the core P
$\overline{S}(P)$	The expected service time to serve a demand by the server on the core P
$\overline{S^2}(P)$	The second moment of service time to a demand for the server on the core P
$\overline{TR}(P)$	The total expected response time to a demand for the server on the core P
$\overline{Q}(P)$	The average waiting time of customers in the queue.

Tab. 1: Some frequently used notations.

stated as follows:

$$prob_p(\hat{u} = p) = \frac{\sum_{v_i \in T_p} \lambda_i}{\sum_{v_i \in T} \lambda_i} = \sum_{v_i \in T_p} w_i = w_{T_p}, \tag{1}$$

where $\hat{u}$ is a stochastic variable, representing the server's location on the core P. Therefore, the mean distance from the server to serve a customer demand in vertex $p \in P$ is readily computed by the following expression:

$$\overline{d}_P(\hat{u}, p) = \sum_{p_i \in P} prob_P(\hat{u} = p_i) d(p_i, p). \tag{2}$$

When a demand is put forward at a node, first, it should be transmitted to the closest vertex of the core. Therefore, the total expected distance that customers travel to reach the core to receive service is considered as follows

$$\overline{d}_1(P) = \sum_{v_i \in T} w_i d(P, v_i). \tag{3}$$

Also, the total expected distance that the server travels to serve customers' demands is achieved by the following expression:

$$\overline{d}_2(P) = \sum_{v_i \in T} w_i \overline{d}_P(\hat{u}, p(i)), \tag{4}$$

where $p(i)$ is the closest vertex on the core P to the vertex v_i. Since server behavior and customer demand are independent, then the structure of their movements are independent, as well. Now, without loss of generality, it is assumed that the average speed of a server is $v_s > 0$. The total expected time for a server to traverse along the core P to serve the customers' demands is achieved by the following expression:

$$\overline{T}_2(P) = \frac{1}{v_s}\overline{d}_2(P). \tag{5}$$

Note that when the core P reduces to 1-median, the server is just located at a vertex.

On the other hand, the total expected time for customers to travel on the edges of the tree network is given as follows:

$$\overline{T}_1(P) = \frac{1}{v_c}\overline{d}_1(P), \tag{6}$$

where v_c is the typical moving speed of the customers. By these definitions, the expected service time including the waiting time for the server and the total expected response time to reply to customers would be given by equations (7) and (8), respectively,

$$\overline{S}(P) = \overline{T}_2(P) + \overline{G} = \sum_{v_i \in T} w_i s_i(P) \tag{7}$$

$$\overline{TR}(P) = \beta(\overline{Q}(P) + \overline{T}_2(P)) + \overline{G} + (1-\beta)\overline{T}_1(P), \tag{8}$$

where β is a balancing parameters and can be chosen in the range $[0,1]$, and $s_i(P)$ is the total service time associated with a serviced demand from node i, represented by the following expression:

$$s_i(P) = \frac{1}{v_s}\overline{d}_P(\hat{u}, p(i)) + G_i. \tag{9}$$

Indeed, objective function (8) is a weighted combination of two distinct objective functions $\overline{Q}(P) + \overline{T}_2(P)$ and $\overline{T}_1(P)$. Thus, β is a parameter assigned to these objective functions to prioritize one over another. In the case $\beta = 0$, the problem is equivalent to the median problem, and if $\beta = 1$, the problem will be the same as finding the core of a tree. In other cases where $\beta \in (0,1)$, the value of β affects the quality of service provided by each objective function.

In problem formulation (8), $\overline{G} = \frac{\sum_{i=1}^{n} G_i}{n}$ refers to the average non-travel service time, which is a positive constant value. If we also add the manufacturing costs to the proposed problem, then the objective function of the SQC problem becomes minimizing the following objective function,

$$F(P) = \alpha_1|P| + \alpha_2\overline{TR}(P) = \alpha_1|P| + \alpha_2\left(\beta(\overline{Q}(P) + \overline{T}_2(P)) + \overline{G} + (1-\beta)\overline{T}_1(P)\right) \tag{10}$$

where $\alpha_1 \geq 0$ is the manufacturing costs per unit of the path and $\alpha_2 \geq 0$ is the price one must pay for each unit of time. The term $\overline{Q}(P)$ is the average waiting time on the path P for customers in the $M/G/1$ queue to be serviced and is obtained by the Pollaczek-Khinchine formula, as follows:

$$\overline{Q}(P) = \begin{cases} \frac{\lambda\overline{S^2}(P)}{2(1-\lambda\overline{S}(P))} & if \quad 1-\lambda\overline{S}(P) > 0 \\ \infty & \text{otherwise,} \end{cases} \tag{11}$$

with $\overline{S^2}(P)$ being the second moment of the service time on the path P is computed as follows:

$$\overline{S^2}(P) = \sum_{v_i \in T} w_i s_i^2(P). \tag{12}$$

Note that the proposed $M/G/1$ queue is stable due to the constraint $1 - \lambda\overline{S}(P) > 0$ that guarantees that the average service rate is higher than the average arrival rate, preventing the queue from growing indefinitely and ensuring a stationary distribution for the system, see [2].

3. PROPERTIES OF THE STOCHASTIC QUEUE CORE PROBLEM

In this section, we investigate some properties of the objective function (8). Assume vertex u is an adjacent vertex to one of the end vertices of the path P. In Lemma 3.1, a recursion relation is given to find the first and second moment service time on the path P and thereupon on the path $P' = P \cup \{u\}$. Let $w(T) = \sum_{v_i \in T} w_i$.

Lemma 3.1. Let P be a path on the tree T and u be a vertex adjacent to one of its end vertices. Set $P' = P \cup \{u\}$, then

$$\overline{S}(P') = \overline{S}(P) + \frac{2}{v_s} w_{T_u}(w(T) - w_{T_u})d(P,u) \tag{13}$$

$$\overline{S^2}(P') = \overline{S^2}(P) + \frac{2}{v_s} w_{T_u} d(P,u)\overline{S}(P) + \frac{1}{v_s^2} w_{T_u}(w(T) - w_{T_u})w(T)d^2(P,u). \tag{14}$$

P r o o f.

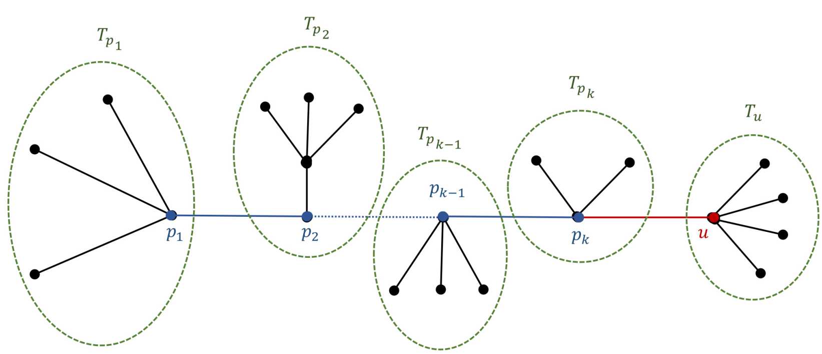

Fig. 1: The branches of the path $P : p_1 - p_k$.

Consider Figure 1, and let T_{p_i} be the branch associated with the vertex p_i, $i = 1, \ldots, k$, where k is the number of vertices of the path P. Also let T_u be the branch associated with the vertex u. The following equalities hold:

$$\begin{aligned} s_1(P') &= s_1(P) + \frac{1}{v_s} w_{T_u} d(P,u), \\ s_2(P') &= s_2(P) + \frac{1}{v_s} w_{T_u} d(P,u), \\ &\vdots \\ s_k(P') &= s_k(P) + \frac{1}{v_s} w_{T_u} d(P,u), \\ s_{k+1}(P') &= s_k(P) + \frac{1}{v_s} (w_{T_{p_1}} + \cdots + w_{T_{p_k}}) d(P,u). \end{aligned} \tag{15}$$

Multiplying the above equalities by $w_{T_{p_i}}$, and summing them up, we obtain the following equality:

$$\begin{aligned} \overline{S}(P') &= \overline{S}(P) + \frac{2}{v_s} w_{T_u} (w_{T_{p_1}} + \cdots + w_{T_{p_k}}) d(P,u) \\ &= \overline{S}(P) + \frac{2}{v_s} w_{T_u} (w(T) - w_{T_u}) d(P,u). \end{aligned} \tag{16}$$

Moreover, squaring both sides of equalities (15), multiplying them by $w_{T_{p_i}}$ and summing them up results in the following equality:

$$\begin{aligned} \overline{S^2}(P') &= \overline{S^2}(P) + \frac{2}{v_s} w_{T_u} d(P,u) \overline{S}(P) \\ &+ \frac{1}{{v_s}^2} w_{T_u} (w_{T_{p_1}} + \cdots + w_{T_{p_k}})(w_{T_{p_1}} + \cdots + w_{T_{p-k}} + w_{T_u}) d^2(P,u) \\ &= \overline{S^2}(P) + \frac{2}{v_s} w_{T_u} d(P,u) \overline{S}(P) + \frac{1}{{v_s}^2} w_{T_u} (w(T) - w_{T_u}) w(T) d^2(P,u). \end{aligned}$$

□

As a result, the queuing delay on the path $P' = P \cup \{u\}$ can significantly be computed, see Corollary 3.2.

Corollary 3.2. Let P be a path on the tree network T and $P' = P \cup \{u\}$, where vertex u is adjacent to one of the end vertices of the path P, then

$$\overline{Q}(P') = \frac{\lambda \left(\overline{S^2}(P) + \frac{2}{v_s} w_{T_u} d(P,u) \overline{S}(P) + \frac{1}{{v_s}^2} w_{T_u} (w(T) - w_{T_u}) w(T) d^2(P,u) \right)}{2 \left(1 - \lambda \left(\overline{S}(P) + \frac{2}{v_s} w_{T_u} (w(T) - w_{T_u}) d(P,u) \right) \right)}. \tag{17}$$

The following property holds by the definition of $\overline{T}_1(P)$, Lemma 3.1 and Corollary 3.2.

Corollary 3.3. Let P and P' be two paths on the tree T such that $P \subset P'$, then the following equalities hold:

1. $\overline{T}_1(P') \leq \overline{T}_1(P)$, $\overline{T}_2(P) \leq \overline{T}_2(P')$

2. $\overline{S}(P) \leq \overline{S}(P')$, $\overline{S^2}(P) \leq \overline{S^2}(P')$

3. $\overline{Q}(P) < \overline{Q}(P')$.

Next, a condition is imposed to compare the objective function values on the path P and $P' = P \cup \{u\}$.

Lemma 3.4. Let P be a path and u is a vertex adjacent to one of its end vertices. Let $P' = P \cup \{u\}$. Then $F(P') > F(P)$ provided that $d(P,u) > 0$ and

$$\alpha_1 + \alpha_2 w_{T_u}\left(\frac{2}{v_s}\beta(w(T) - w_{T_u}) - \frac{1}{v_c} + \frac{\beta}{v_c}\right) > 0. \tag{18}$$

Proof. Let vertices of the path P be $p_1, \dots, p_k$ where $d(P,u) = d(p_k,u)$, see Figure 1. Then, vertices of the path P' are $p_1, p_2, \dots, p_k, p_{k+1} = u$. The objective function of the path P' would be formulated as follows:

$$\begin{aligned} F(P') &= \alpha_1|P'| + \alpha_2\left(\beta\left(\overline{Q}(P') + \overline{T}_2(P')\right) + \overline{G} + (1-\beta)\overline{T}_1(P')\right) \\ &= \alpha_1|P'| + \alpha_2\beta\left(\overline{Q}(P') + \overline{S}(P') - \overline{G}\right) + \alpha_2\overline{G} + \alpha_2(1-\beta)\left(\frac{1}{v_c}\sum_{i=1}^{k+1}\sum_{v_j\in T_i} w_j d(p_i,v_j)\right). \end{aligned} \tag{19}$$

Using Lemma 3.1,

$$\begin{aligned} F(P') &= \alpha_1(|P| + d(P,u)) + \alpha_2\beta(\overline{Q}(P') + \overline{S}(P) + \frac{2}{v_s}w_{T_u}(w(T) - w_{T_u})d(P,u) - \overline{G}) \\ &+ \alpha_2\overline{G} + \alpha_2(1-\beta)\left(\frac{1}{v_c}\sum_{i=1}^{k}\sum_{v_j\in T_i} w_j d(p_i,v_j) + \frac{1}{v_c}\sum_{v_j\in T_u} w_j d(u,v_j)\right) \\ &= \alpha_1(|P| + d(P,u)) + \alpha_2\beta\left(\overline{Q}(P') + \overline{S}(P) + \frac{2}{v_s}w_{T_u}(w(T) - w_{T_u})d(P,u) - \overline{G}\right) \\ &+ \alpha_2\overline{G} + \alpha_2(1-\beta)\frac{1}{v_c}\left(\sum_{i=1}^{k}\sum_{v_j\in T_i} w_j d(p_i,v_j) + \sum_{v_j\in T_u} w_j(d(p_k,v_j)) - w_{T_u}d(P,u)\right) \\ &= \alpha_1|P| + \alpha_2\beta\left(\overline{Q}(P') + \overline{S}(P) - \overline{G}\right) + \alpha_2\overline{G} + \alpha_2(1-\beta)\left(\frac{1}{v_c}\overline{T}_1(P) - w_{T_u}d(P,u)\right) \\ &+ \alpha_2\frac{2\beta}{v_s}w_{T_u}(w(T) \\ &- w_{T_u})d(P,u) + \alpha_1 d(P,u). \end{aligned} \tag{20}$$

Therefore, the following expression holds

$$\begin{aligned} F(P') &= F(P) + \alpha_1 d(P,u) + \alpha_2\beta(\overline{Q}(P') - \overline{Q}(P)) \\ &+ \frac{2\alpha_2\beta}{v_s} w_{T_u}(w(T) - w_{T_u})d(P,u) - (1-\beta)\frac{\alpha_2}{v_c} w_{T_u} d(P,u) \\ &= F(P) + (\alpha_1 + \frac{2\alpha_2\beta}{v_s} w_{T_u}(w(T) - w_{T_u}) - (1-\beta)\frac{\alpha_2}{v_c}(w_{T_u}))d(P,u) \\ &+ \alpha_2\beta(\overline{Q}(P') - \overline{Q}(P)). \end{aligned} \tag{21}$$

By Corollary 3.3 $\overline{Q}(P') > \overline{Q}(P)$. Also, $d(P,u) > 0$ is a positive value. Hence, if the inequality

$$\alpha_1 + \alpha_2 w_{T_u}\left(\frac{2}{v_s}\beta(w(T) - w_{T_u}) - \frac{1}{v_c} + \frac{\beta}{v_c}\right) > 0,$$

holds, we conclude the $F(P') > F(P)$. □

Note that inequality (18) in Lemma 3.4 is a sufficient but not necessary condition for fulfilling $F(P') > F(P)$ when $P' = P \cup \{u\}$, as demonstrated in Example 4.1. Nevertheless, through the application of this criterion and verifying the condition $F(P') < F(P)$, we introduce an algorithm to solve the *SQC* problem. These ideas yield the *SQC* algorithm (Algorithm 1).

Algorithm 1 The *SQC* algorithm

Require: The weighted tree $T = (V, E)$, and parameters λ_i, $1 \le i \le n$, $\alpha_1, \alpha_2, v_s, v_c$ and β.

Ensure: The *SQC* denoted by P^*.

1: Set $k = 1$.
2: Consider each vertex $v_i \in V$ as a path and put it in A_1.
3: For each $P \in A_1$ compute $F(P)$.
4: Set $P^* = argmin_{P \in A_1} F(P)$ and $F^* = F(P^*)$.
5: IF $A_k \neq \emptyset$ then for each path $P \in A_k$ do the following steps else stop:

a. For each vertex u adjacent to the one of end vertices of P such that

$$\alpha_1 + \alpha_2 w_{T_u}\left(\frac{2}{v_s}\beta(w(T) - w_{T_u}) - \frac{1}{v_c} + \frac{\beta}{v_c}\right) \le 0, \tag{22}$$

do the following steps:

i. Set $P' = P \cup \{u\}$
ii. If $P' \notin A_{k+1}$ then compute $F(P')$ using (21). Else go to step (a).
iii. If $F(P') < F(P)$ then add P' to A_{k+1}.
iv. If $F(P') < F^*$ then set $P^* = P'$ and $F^* = F(P')$.

6: Set $k = k + 1$ and go to Step 5.

The algorithm starts with computing the value of objective function for all vertices in T. Then, in iteration k, some paths with $k+1$ vertices are constructed by adding a vertex to the paths from the previous iteration, and the paths whose objective function value is improved are considered in the next iteration. Note that, this algorithm also could be applied to the problem whose goal is finding the best path with a specific number of edges. For example, if the best path with r edges is desired, the best path in $(r-1)$th iteration is the solution.

3.1. Complexity of the SQC algorithm

When $P = \{v_i\}$ then $F(P)$ can be computed in $O(n)$ time. Thus Step (3) can be computed in $O(n^2)$ time. Step (5) is the main step of the algorithm. Let $P' = P \cup \{u\}$ and the value of $F(P)$ is given, then $F(P')$ can be computed using (21), Lemma 3.1 and Corollary 3.2 in a linear time. In the worst case the algorithm counts all paths in a tree. Since there is just one path between any two vertices in a tree, thus the number of all paths is equal to the number of selecting 2 from n vertices, which is $O(n^2)$. Therefore, the time complexity of the algorithm in the worst case is $O(n^3)$. Note that although the time complexity is $O(n^3)$, however, in many cases the condition (22) does not hold and the algorithm terminates in a fewer time.

4. NUMERICAL EXAMPLES

In this section, we provide two examples to illustrate the details of the SQC algorithm. In the first example, a tree network with 12 vertices is examined and details of calculating the expected customers' travel times, expected server's travel time and the SQC objective function are given. In the second example, the iterations of running the SQC algorithm on a tree network with 9 vertices are presented in the associated tables.

Example 4.1. Consider the tree network shown in Figure 2. The numbers written next to the edges are their lengths. The demand rates, weights, and service times of the vertices are inserted in Table 2. For the sake of simplicity, consider the cruising speeds are $v_s = v_c = 1$ $(\frac{Km}{hour})$, $\alpha_2 = 1$ and α_1 and β take different values. We have $\lambda = 0.34$ $(customers/hour)$, $\overline{G} = 0.01$ $(hour)$, and $w(T) = 1$.

vertices	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}
λ_i	0.02	0.03	0.01	0.04	0.03	0.03	0.05	0.02	0.03	0.05	0.02	0.01
$w_i = \frac{\lambda_i}{\lambda}$	0.0588	0.0882	0.0294	0.1176	0.0882	0.0882	0.1471	0.0588	0.0882	0.1471	0.0588	0.0294
G_i (hour)	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01

Tab. 2: The demand rates and service times of vertices in Example 4.1.

First, consider the path $P = \{v_5, v_{12}\}$. The two sub-trees T_5 and T_{12} are shown in Figure 3. To compute $F(P)$, we should calculate $\bar{T}_1(P)$ and $\bar{T}_2(P)$ which are depended on $\bar{d}_1(P)$ and $\bar{d}_2(P)$, respectively. Using (3), we obtain

$$\bar{d}_1(P) = \sum_{v_i \in T_5} w_i d(v_5, v_i) + \sum_{v_i \in T_{12}} w_i d(v_{12}, v_i) = 2.1026.$$

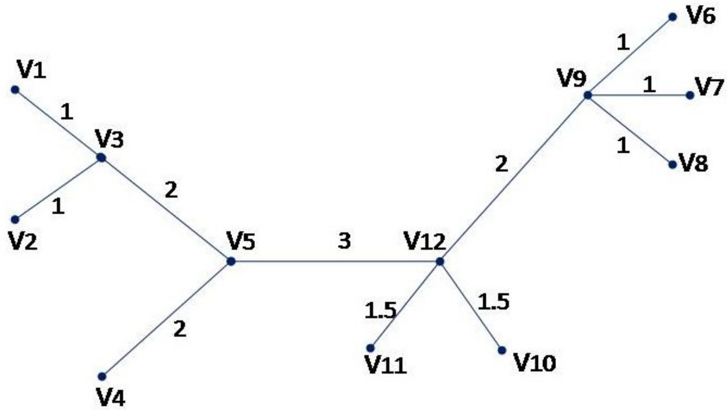

Fig. 2: Tree for Example 4.1.

Moreover, by relation (1) we have

$$prob_P(\hat{u} = v_5) = w_{T_5} = 0.3822,$$

$$prob_P(\hat{u} = v_{12}) = w_{T_{12}} = 0.6178.$$

Using relation (2) yields,

$$\bar{d}_P(\hat{u}, v_5) = prob_P(\hat{u} = v_{12})d(v_{12}, v_5) = 1.8534,$$

$$\bar{d}_P(\hat{u}, v_{12}) = prob_P(\hat{u} = v_5)d(v_5, v_{12}) = 1.1466.$$

Therefore, the expected distance travel by server, using relation (4), is

$$\overline{d}_2(P) = \sum_{v_i \in T_5} w_i \bar{d}_P(\hat{u}, v_5) + \sum_{v_i \in T_{12}} w_i \bar{d}_P(\hat{u}, v_{12}) = 1.4167.$$

To calculate $\bar{Q}(P)$, using relation (9) we have

$$s_i(P) = \begin{cases} \bar{d}_P(\hat{u}, v_5) + G_i = 1.8634 & i = 1, \ldots, 5 \\ \bar{d}_P(\hat{u}, v_{12}) + G_i = 1.1566 & i = 6, \ldots, 12. \end{cases}$$

Thus,

$$\overline{S}(P) = \sum_{v_i \in T} w_i s_i(P) = 1.4267,$$

$$\overline{S}^2(P) = \sum_{v_i \in T} w_i s_i^2(P) = 2.1535.$$

Therefore,

$$\bar{Q}(P) = \frac{\lambda \overline{S^2}(P)}{2(1 - \lambda \overline{S}(P))} = 0.7110.$$

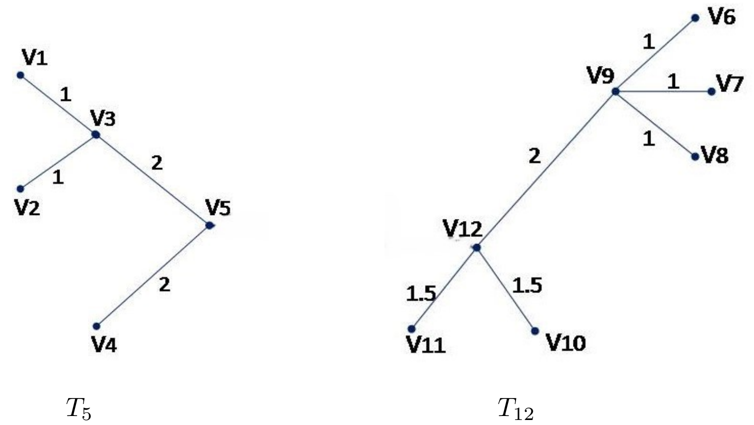

Fig. 3: Two sub-trees of tree in Figure 2.

				$F(P)$		
				$\alpha_1 = 0$	$\alpha_1 = 0.1$	$\alpha_1 = 0.5$
path	$\overline{T}_1$	$\overline{T}_2$	$\overline{Q}$	$\beta = 0$	$\beta = 0.1$	$\beta = 0.1$
$\{v_5\}$	3.9559	0.0000	0.0000	3.9659	3.5703	3.5703
$\{v_{12}\}$	3.2500	0.0000	0.0000	3.2600	2.9350	2.9350
$\{v_9\}$	3.7206	0.0000	0.0000	3.7306	3.3585	3.3585
$\{v_5, v_{12}\}$	2.1026	1.4167	0.7110	2.1129	2.3150	3.6155
$\{v_{12}, v_9\}$	2.4853	0.9446	0.2425	2.4953	2.5655	3.3655
$\{v_5, v_{12}, v_9\}$	1.3382	2.3616	5.0013	1.3482	2.4507	4.4507
$\{v_3, v_5, v_{12}\}$	1.7500	1.9983	2.4618	1.7600	2.5310	4.5310
$\{v_3, v_5, v_{12}, v_9\}$	0.9853	2.9429	∞	0.9953	∞	∞

Tab. 3: The objective functions of some paths in Example 4.1.

Then using (10) for the case $\alpha_1 = \beta = 0.1$, we obtain $F(P) = 2.3150$. Table 3 contains some paths and their corresponding $\bar{T}_1, \bar{T}_2, \bar{Q}$ and $F(P)$ for varying values of α_1 and β.

Now, consider the paths $P = \{v_5, v_{12}\}$ and $P' = P \cup \{v_9\}$ when $\alpha_1 = \beta = 0.1$. Then $w_{T_9} = 0.3824$ and

$$\alpha_1 + \alpha_2 w_{T_9}(\frac{2}{vs}\beta(w(T) - w_{T_9}) - \frac{1}{vc} + \frac{\beta}{vc}) = 0.1 - 0.2969 < 0.$$

Although, condition (18) of Lemma 3.4 is not satisfied for these two paths, the inequality $F(P) < F(P')$ holds. Therefore, this condition is not necessary. However, for these two paths in the case $\alpha_1 = 0.5$ and $\beta = 0.1$, the condition of Lemma 3.4 holds, i. e.

$$\alpha_1 + \alpha_2 w_{T_9}(\frac{2}{vs}\beta(w(T) - w_{T_9}) - \frac{1}{vc} + \frac{\beta}{vc}) = 0.5 - 0.2969 > 0, \tag{23}$$

and also $F(P) < F(P')$. Note that in the case $\alpha_1 = \beta = 0$ the problem is equivalent to find the core of tree.

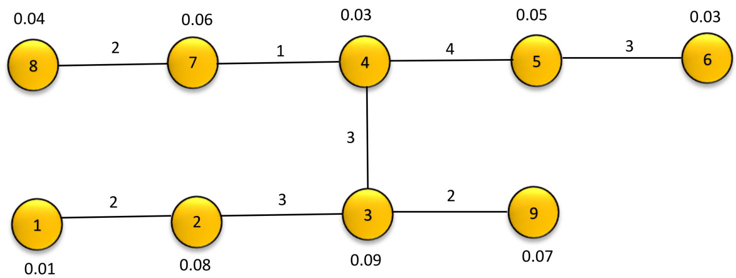

Fig. 4: A tree with 9 vertices.

P	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
$F(P)$	2.1528	1.7224	1.3115	1.3702	1.9571	2.5441	1.4974	1.8691	1.7615

Tab. 4: The value of objective functions of vertices the tree in Figure 4.

Example 4.2. Consider the tree depicted in Figure 4. The numbers on the vertices show the demand rates of vertices. For this example, we set $\beta = 0.1$, $\alpha_1 = 0.1$, $\alpha_2 = 1$, $v_c = 4$, $v_s = 100$ and $G_i = 0.5$. First, $F(P)$ should be computed for all vertices belong to the set $A_1 = \{v_1, \dots, v_9\}$, see Table 4. Then condition (22) will be checked to make new paths. The results of the first iteration are given in Table 5. In this table the column with heading "cand. val." indicates the value of condition (22). To extend a path by a new vertex, we should check the "cand. val." values. When the "cand. val." is non-negative, absolutely, the objective function value increases. But, when the "cand. val." is negative, the objective function may decreases or increases.

path P	$F(P)$	P'	cond. val.	$F(P')$
v_1	2.1528	$v_1 - v_2$	-0.1201	1.9163
v_2	1.7224	$v_2 - v_3$	-0.0807	1.5646
v_3	1.3115	$v_3 - v_9$	0.0660	-
		$v_3 - v_4$	-0.0022	1.3084
v_4	1.3702	$v_4 - v_5$	0.0612	-
		$v_4 - v_7$	0.0514	-
v_5	1.9571	$v_5 - v_6$	0.0854	-
v_7	1.4974	$v_7 - v_8$	0.0806	-

Tab. 5: The results of the first iteration of the algorithm for Example 4.2.

Here, in the first iteration, we obtain $A_2 = \{v_1 - v_2,\ v_2 - v_3,\ v_3 - v_4\}$ as the improved paths compared to the nodes $\{v_1, v_2, v_3\}$. The results of the other iterations are shown in tables 6 to 8. The best path is $P^* = v_3 - v_4$ with $f(P^*) = 1.3084$.

path P	$F(P)$	P'	cond. val.	$F(P')$
$v_1 - v_2$	1.9163	$v_1 - v_2 - v_3$	-0.0807	1.6751
$v_2 - v_3$	1.5646	$v_2 - v_3 - v_4$	-0.0022	1.5584
		$v_2 - v_3 - v_9$	0.0660	-
$v_3 - v_4$	1.3084	$v_3 - v_4 - v_5$	0.0612	-
		$v_3 - v_4 - v_7$	0.0514	-
		$v_9 - v_3 - v_4$	0.0660	-

Tab. 6: The results of the second iteration of the algorithm for Example 4.2.

path P	$F(P)$	P'	cond. val.	$F(P')$
$v_1 - v_2 - v_3$	1.6751	$v_1 - v_2 - v_3 - v_4$	-0.0022	1.6695
		$v_1 - v_2 - v_3 - v_9$	0.0660	-
$v_2 - v_3 - v_4$	1.5646	$v_2 - v_3 - v_4 - v_5$	0.0612	-
		$v_2 - v_3 - v_4 - v_7$	0.0514	-

Tab. 7: The results of the third iteration of the algorithm for Example 4.2.

path P	$F(P)$	P'	cond. val.	$F(P')$
$v_1 - v_2 - v_3 - v_4$	1.6751	$v_1 - v_2 - v_3 - v_4 - v_5$	0.0612	-
		$v_1 - v_2 - v_3 - v_4 - v_7$	0.0514	-

Tab. 8: The results of the 4th iteration of the algorithm for Example 4.2.

5. SUMMARY AND CONCLUSION.

In this paper, we considered the SQC problem on tree networks, where the customers' demands should be transferred to the closest vertex of the core. Upon a customer request reaching the core, it is placed in the first-come, first-served queuing system. When the server is available, it proceeds toward the customer awaiting at the closest vertex of the core. A mathematical model has been developed, along with several properties, to explore the challenge of determining the core with the optimal length in the tree network. An $O(n^3)$ time algorithm has been proposed to solve the SQC problem. Some extensions of this work include considering variance of serving customers as a constraint, and using different queue discipline can be considered in the future works.

Disclosure statement. The authors report there are no competing interests to declare.

ACKNOWLEDGEMENTS

The authors thank the Editor and the Reviewers for their helpful comments, which certainly improved the original version of our manuscript.

(Received July 24, 2023)

REFERENCES

[1] M. Abareshi abd M. Zaferanieh: A bi-level capacitated p-median facility location problem with the most likely allocation solution. Transport. Res. Part B: Methodological *123* (2019), 1–20. DOI:10.1016/j.trb.2019.03.013

[2] H. Abouee-Mehrizi and O. Baron: State-dependent m/g/1 queueing systems. Queueing Systems *82* (2016), 121–148. DOI:10.1007/s11134-015-9461-y

[3] O. J. Adeleke and D. O. Olukanni: Facility location problems: models, techniques, and applications in waste management. Recycling *5* (2020), 10. DOI:10.3390/recycling5020010

[4] S. Alstrup, P. W. Lauridsen, P. Sommerlund, and M. Thorup: Finding cores of limited length. In: Algorithms and Data Structures: 5th International Workshop, WADS'97, Halifax 1997, Proceedings 5, Springer, pp. 45–54.

[5] P. Avella, M. Boccia, A. Sforza, snd I. Vasilev: A branch-and-cut algorithm for the median-path problem. Comput. Optim. Appl. *32* (2005), 215–230. DOI:10.1007/s10589-005-4800-2

[6] R. Batta and O. Berman: A location model for a facility operating as an m/g/k queue. Networks *19* (1989), 717–728. DOI:10.1002/net.3230190609

[7] R. I. Becker, Y. I. Chang, I. Lari, A. Scozzari, and G. Storchi: Finding the l-core of a tree. Discrete Appl. Math. *118* (2002), 25–42. DOI:10.1016/S0166-218X(01)00254-2

[8] 0. Berman and Z. Drezner: The multiple server location problem. J. Oper. Res. Soc. *58* (2007), 91–99. DOI:10.1007/s00020-007-1497-x

[9] 0. Berman, D. Krass, and J. Wang: Locating service facilities to reduce lost demand. IIE Trans. *38* (2006), 933–946. DOI:10.1080/07408170600856722

[10] 0. Berman, R. C. Larson, and S. S. Chiu: Optimal server location on a network operating as an m/g/1 queue. Oper. Ress *33* (1985), 746–771. DOI:10.1287/opre.33.4.746

[11] 0. Berman, R. C. Larson, and C. Parkan: The stochastic queue p-median problem. Transport. Sci. *21* (1987), 207–216. DOI:10.1287/trsc.21.3.207

[12] 0. Berman and R. R. Mandowsky: Location-allocation on congested networks. Europ. J. Oper. Ress *26* (1986), 238–250. DOI:10.1016/0377-2217(86)90185-2

[13] C. Chen, B. Yao, G. Chen, and Z. Tian: A queuing location allocation model for designing a capacitated bus garage system. Engrg. Optim. *54* (2022), 709–726. DOI:10.1080/0305215X.2021.1897800

[14] S. S. Chiu, O. Berman, and R. C. Larson: Locating a mobile server queueing facility on a tree network. Management Sci. *31* (1985), 764–772. DOI:10.1287/mnsc.31.6.764

[15] J. Fathali, M. Nazari, and K. Mahdvar: Semi-obnoxious backup 2-median problem on a tree. J. Appl. Res. Industr. Engrg. *8* (2021), 159–168.

[16] J. Fathali and M. Zaferanieh: The balanced 2-median and 2-maxian problems on a tree. J. Combinat. Optim. *45* (2023), 69. DOI:10.1007/s10878-023-00997-9

[17] B. Gavish and S. Sridhar: Computing the 2-median on tree networks in $O(n \log n)$ time. Networks *26* (1995), 305–317. DOI:10.1016/0168-1605(94)00136-T

[18] A. J. Goldman: Optimal center location in simple networks. Transport. Sci. *5* (1971), 212–221. DOI:10.1287/trsc.5.2.212

[19] S. M. Hedetniemi, E. Cockayne, and S. Hedetniemi: Linear algorithms for finding the jordan center and path center of a tree. Transport. Sci. *15* (1981), 98–114. DOI:/10.1287/trsc.15.2.98

[20] 0. Kariv and S. L. Hakimi: An algorithmic approach to network location problems. i: The p-centers. SIAM J. Appl. Math. *37* (1979), 513–538. DOI:10.1137/0137040

[21] Y. X. Kong, G. Y. Shi, R. J. Wu, and Y. C. Zhang: k-core: Theories and applications. Physics Rep. *832* (2019), 1–32. DOI:10.1007/s40274-019-6058-4

[22] G. Kovacs and K. M. Spens: Humanitarian logistics in disaster relief operations. Int. J. Phys. Distribut. Logist. Management *37* (2007), 99–114. DOI:10.1108/09600030710734820

[23] M. Mohammadi, F. Jolai, and H. Rostami: An m/m/c queue model for hub covering location problem. Math. Computer Modell. *54* (2011), 2623–2638. DOI:10.1016/j.mcm.2011.06.038

[24] C. A. Morgan and P. J. Slater: A linear algorithm for a core of a tree. J. Algorithms *1* (1980), 247–258. DOI:10.1016/0196-6774(80)90012-7

[25] S. A. Morgan and N. H. Agee: Mobile healthcare. Frontiers Health Services Management *29* (2012), 3–10. DOI:10.1097/01974520-201210000-00002

[26] M. Moshtagh, J. Fathali, and J. M. Smith: The stochastic queue core problem, evacuation networks, and state-dependent queues. Europ. J. Oper. Res. *269* (2018), 730–748. DOI:10.1016/j.ejor.2018.02.026

[27] M. Moshtagh, J. Fathali, J. M. Smith, and N. Mahdavi-Amiri: Finding an optimal core on a tree network with m/g/c/c state-dependent queues. Math. Methods Oper. Res. *89* (2019), 115–142. DOI:10.1007/s00186-018-0651-3

[28] S. H. Owen and M. S. Daskin: Strategic facility location: A review. Europ. J. Oper. Res. *111* (1998), 423–447. DOI:10.1016/S0377-2217(98)00186-6

[29] L. Ozdamar, E. Ekinci, and B. Kucukyazici: Emergency logistics planning in natural disasters. Ann. Oper. Res. *129* (2004), 217–245. DOI:10.1023/b:anor.0000030690.27939.39

[30] P. Pourmohammadi, R. Tavakkoli-Moghaddam, Y. Rahimi, and C. Triki: Solving a hub location routing problem with a queue system under social responsibility by a fuzzy meta-heuristic algorithm. Ann. Oper. Res. *324* (2023), 1099–1128. DOI:10.1007/s10479-021-04299-3

[31] P. J. Slater: Locating central paths in a graph. Transport. Sci. *16* (1982), 1–18.

[32] A. Tamir: An $O(pn^2)$ algorithm for the p-median and related problems on tree graphs. Oper. Res. Lett. *19* (1996), 59–64. DOI:10.1016/0167-6377(96)00021-1

[33] R. Tavakkoli-Moghaddam, S. Vazifeh-Noshafagh, A. A., Taleizadeh, V. Hajipour, and A. Mahmoudi: Pricing and location decisions in multi-objective facility location problem with m/m/m/k queuing systems. Engrg. Optim. *49* (2017), 136–160. DOI:10.1080/0305215x.2016.1163630

[34] Q. Wang, R. Batta, and C. M. Rump: Algorithms for a facility location problem with stochastic customer demand and immobile servers. Ann. Oper. Res. *111* (2002), 17–34.

[35] M. Zaferanieh, M. Abareshi, and J. Fathali: The minimum information approach to the uncapacitated p-median facility location problem. Transport. Lett. *14* (2022), 307–316. DOI:10.1080/19427867.2020.1864595

[36] M. Zaferanieh and J, Fathali: Finding a core of a tree with pos/neg weight. Math. Methods Oper. Res. *76* (2012), 147–160. DOI10.1007/s00186-012-0394-5:

[37] M. Zaferanieh, M. Sadra, and T. Basirat: P-facility capacitated location problem with customer equilibrium decisions: a recreational case study in Mazandaran province. J. Modell. Management *19* (2024), 1883–1906. DOI:10.1002/ange.19060194415

Jafar Fathali, Faculty of Mathematical Sciences, Shahrood University of Technology, University Blvd., Shahrood. Iran.
e-mail: fathali@shahroodut.ac.ir

Mehdi Zaferanieh, Department of Mathematics, Hakim Sabzevari University, Tovhid town, Sabzevar. Iran.
e-mail: m.zaferanieh@sttu.ac.ir