

## Research Article

# A Continuous Deviation-Flow Location Problem for an Alternative-Fuel Refueling Station on a Tree-Like Transportation Network

Sang Jin Kweon,<sup>1</sup> Seong Wook Hwang,<sup>2</sup> and Jose A. Ventura<sup>3</sup>

<sup>1</sup>Heinz College, Carnegie Mellon University, Pittsburgh, PA 15213, USA

<sup>2</sup>Agency for Defense Development, Yuseong, Daejeon 34186, Republic of Korea

<sup>3</sup>The Harold and Inge Marcus Department of Industrial and Manufacturing Engineering, The Pennsylvania State University, University Park, PA 16802, USA

Correspondence should be addressed to Sang Jin Kweon; sangjink@andrew.cmu.edu

Received 15 June 2017; Revised 27 August 2017; Accepted 8 October 2017; Published 14 December 2017

Academic Editor: Edward Chung

Copyright © 2017 Sang Jin Kweon et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to the increasing popularity of alternative-fuel (AF) vehicles in the last two decades, several models and solution techniques have been recently published in the literature to solve AF refueling station location problems. These problems can be classified depending on the set of candidate sites: when a (finite) set of candidate sites is predetermined, the problem is called discrete; when stations can be located anywhere along the network, the problem is called continuous. Most researchers have focused on the discrete version of the problem, but solutions to the discrete version are suboptimal to its continuous counterpart. This study addresses the continuous version of the problem for an AF refueling station on a tree-type transportation network when a portion of drivers are willing to deviate from their preplanned simple paths to receive refueling service. A polynomial time solution approach is proposed to solve the problem. We first present a new algorithm that identifies all possible deviation options for each travel path. Then, an efficient algorithm is used to determine the set of optimal locations for the refueling station that maximizes the total traffic flow covered. A numerical example is solved to illustrate the proposed solution approach.

## 1. Introduction

The US national highway system significantly contributes to the country's economic development. In particular, it plays a major role in providing mobility for goods and services. Using highways, trucks carry the largest shares by value, tons, and ton-miles for shipments moving 750 or fewer miles in the US [1]. As concerns over climate change are increasingly becoming a global issue, logistics companies are interested in replacing their diesel trucks by alternative-fuel (AF) trucks [2]. However, to introduce AF trucks to logistics companies successfully, a well-designed refueling infrastructure in the highway system is a prerequisite. The rest of this section highlights the relevance of tree networks in transportation, the significance of the single station location problem for a new AF refueling infrastructure, and the continuous version of the problem and summarizes the main contributions of this research work.

*1.1. Tree Structure in Transportation Networks.* While the US national highway system consists of a set of circuit networks, if it is partitioned into local highway systems by operating authority, then many of them form trees or tree-like networks. Table 1 displays the local highway systems in the US along with their operating authorities that have tree or tree-like network structures. As shown in this table, 10 turnpikes and 8 portions of interstate highways within certain states constitute tree structures [3]. Furthermore, if we consolidate beltways and cycles within metro areas into single vertices, then 11 more portions of interstate highways within certain states can also be considered as tree-like structures [4, 5]. In addition, we can observe that toll roads and highways in other countries, such as Chile, Croatia (southern and northeast regions), Indonesia, Ireland (excluding the Dublin region), Malaysia, Norway (northern region), Philippines, Slovakia, Spain (excluding the Catalan region), and Serbia, to name a few, have tree or tree-like network structures. Note also that

TABLE I: Tree network structures in the US highways.

State	Road type*	Type of network structure	Component highways	Operating authority
AL	I	Tree-like, including Birmingham beltway	I-20, I-22, I-59, I-65, I-85	AL Department of Transportation (DOT)
AK	S	Tree in Southcentral area	AK-1, AK-3	AK DOT and Public Facilities
AZ	I	Tree-like, including Phoenix beltway	I-8, I-10, I-15 I-17, I-19, I-40	AZ DOT
AR	I	Tree-like, including Little Rock beltway	I-30, I-40, I-49, I-55, I-530, I-555	AR Highway and Transportation Department
CO	I	Tree-like, including Denver beltway	I-25, I-70, I-76	CO DOT
GA	I	Tree-like, including beltways in Atlanta, Augusta, Columbus, and Macon	I-16, I-20, I-59, I-75, I-85, I-185, I-285, I-675, I-985	GA DOT
ID	I	Tree	I-15, I-84, I-86, I-90	ID Transportation Department
IN	T	Tree	I-80, I-90	IN Finance Authority/ ITR Concession Company LLC
KS	T	Tree	I-35, I-70, I-335, I-470	KS Turnpike Authority
ME	T	Tree	I-95	ME Turnpike Authority
MA	T	Tree	I-90	MA DOT
MN	I	Tree-like, including Minneapolis beltway	I-35, I-90, I-94	MN DOT
MS	I	Tree	I-20, I-22, I-55, I-59	MS DOT
MT	I	Tree	I-15, I-90, I-94	MT DOT
NE	I	Tree-like, including Omaha beltway	I-76, I-80	NE Department of Roads
NV	I	Tree-like, including Las Vegas beltway	I-15, I-80, I-515	NV DOT
NH	T	Tree	I-93, I-95, NH-3, NH-16	NH DOT
NM	I	Tree	I-10, I-25, I-40	NM DOT
NY	T	Tree	I-87, I-90	NY State Thruway Authority
ND	I	Tree	I-29, I-94	ND DOT
OH	T	Tree	I-76, I-80, I-90	OH Turnpike Commission
OK	T	Tree	I-35, I-40, I-44, US-66, US-165, US-177, US-412, OK-1, OK-51, OK-66, OK-165	OK Turnpike Authority
OR	I	Tree-like, including Portland beltway	I-5, I-84	OR DOT
PA	T	Tree	I-70, I-76, I-276, I-376, I-476, PA-43, PA-60, PA-66, PA-576	PA Turnpike Commission
SD	I	Tree	I-29, I-90	SD DOT
UT	I	Tree-like, including Salt Lake City beltway	I-15, I-70, I-80, I-84	UT DOT
VT	I	Tree	I-89, I-91, I-93	VT Agency of Transportation
WV	T	Tree	I-64, I-77	WV Parkways Authority
WY	I	Tree	I-25, I-80, I-90	WY DOT

\* I: interstate highway, S: state highway, and T: turnpike.

road networks in sparsely settled areas are generally trees, since tree road networks are the cheapest to construct.

Trees are central to the structural understanding of networks and graphs and often occur with additional attributes such as roots and vertex-ordering. They have a wide range of applications, including data storage, searching, information processing, and facility location [6]. Because it is easier to get insights into tree network problems, numerous articles

in classical facility location problems on transportation networks without cycles are available in the literature ([7–9] and others).

*1.2. Setting Up an AF Refueling Station on Toll Roads.* Building an AF refueling station usually requires a substantial investment. For example, it takes \$1.8 million to \$4.7 million to construct a natural gas refueling station on the Pennsylvania

Turnpike without including the land value [10]. Due to the financial risk, it is unusual to build multiple AF refueling stations on the same road network simultaneously. Instead, an operating authority may be more inclined in setting up a single station first in a high traffic area and planning the gradual future construction of additional stations with the goal of maximizing the coverage of traffic flow. For example, the Pennsylvania Turnpike Commission opened the first refueling station at New Stanton Service Plaza in 2014 to serve compressed natural gas (CNG) vehicles and it currently remains as the only CNG station on the turnpike [11]. As a second example, the Oklahoma Turnpike Authority opened the first CNG station at Stroud Travel Plaza on the turnpike in 1991, which was renovated in 2014, and then 23 years later, the authority opened the second CNG station at McAlester Travel Plaza on the turnpike in 2014 [12–14]. Note that both of these turnpikes form tree networks.

Several articles have been published regarding the classical single facility location problem on tree networks ([7, 15–17] and others). Three versions of the problem exist depending on the main objective: the objective of the 1-median problem is to minimize the total travel distance from the new facility to existing facilities; in the 1-center problem, the objective is to minimize the maximum travel distance; and the covering problem finds the location of a facility that can reach (cover) the maximum number of existing facilities within a given travel distance. Francis et al. ([5], Chapter 7) describe motivating applications of location problems on trees, as well as properties and solution algorithms for the three versions of the problem. None of the algorithms developed to solve classical single facility location problems on tree networks, however, is suitable for locating an AF refueling station on tree-like road networks because of the differences in the objective function. In the location of an AF station, the objective of maximizing the total traffic flow covered by the station is more desirable by operating authorities. Furthermore, locating an AF refueling station requires a different strategy to deal with demand. Note that the classical single facility location problem regards demand as a weight at vertices. Instead, drivers (demand) on toll roads and highways usually stop by the refueling station on the way to their final destination. Considering that this refueling activity for a driver is a unit of demand, a path-based demand model, in which the demand on the travel path can be covered by a facility located on the path, is more appropriate for solving the AF refueling station location problem.

**1.3. Continuous Refueling Station Location Problem.** We can classify refueling station location problems into two types depending on the set of candidate sites: when a preliminary (finite) set of candidate sites is given, this problem is called *discrete*; when the station can be located anywhere along the network, the problem is called *continuous*. Ventura et al. [18] introduced the continuous version of the refueling station location problem. Since an optimal solution to the discrete version of the problem is always a feasible solution to the continuous version, an optimal solution to the continuous refueling station location problem can achieve a considerable

improvement in terms of demand coverage. The continuous version of the problem is, however, more challenging than its discrete counterpart due to the size of the search space.

**1.4. Contribution.** In this study, we consider the continuous version of an AF refueling station location problem on a tree network, where a portion of drivers have the option of selecting a deviation path if the refueling station is not located along their preplanned routes. We believe that this is the first research work on the continuous version of the problem where the deviation option is considered. This option is expected to be very common for drivers with AF vehicles during the introductory period of these vehicles to the market since AF refueling infrastructures are usually nonexistent or underdeveloped. The deviation option is also available to drivers of conventional fossil-fuel vehicles (powered by gasoline and diesel engines) when they cannot find a refueling station on their routes. Thus, considering deviation will make the problem more practical in deciding the potential location for the refueling station.

Compared with the original continuous refueling station location problem [18], the theoretical contribution of this study is twofold. First, we derive a number of properties to identify the deviation options for each origin/destination (O/D) pair considering the length of the simple path, the minimum fuel tank levels at the origin and destination, and the vehicle driving range. Then, we develop a novel algorithm that finds all deviation paths for each O/D pair in polynomial time. Using the original path in the tree network and the corresponding deviation options, we construct the subtree that contains all the potential station locations that cover the traffic flow of the corresponding O/D pair. Second, we derive a mathematical property that characterizes the set of optimal station locations that do not require Ventura et al. [18] assumption that the intersection vertices in all used simple paths must have positive traffic flow in at least one short trip. This property is essential in the development of the algorithm that finds the complete set of optimal station locations for the entire network.

By allowing vehicle deviation in the continuous refueling station location problem, the solution approach proposed in this article is capable of finding all possible alternative paths for any O/D pair and improving suboptimal solutions obtained by the existing models in the literature. To best of our knowledge, all the existing models that consider vehicle deviation solve discrete versions of the problem, and therefore, they choose locations from a predetermined set of candidate sites (vertices) and enable deviation only if such vertices can be reached within the vehicle driving range. Therefore, their solutions may not be optimal [19]. On the other hand, this study considers all possible deviation paths in the network regardless of the location of vertices on detours and the station can be located anywhere along any detour edge. Thus, the proposed solution approach can improve any possible suboptimality produced by the existing models and generate the entire set of optimal points.

To verify the performance of our proposed solution approach, a numerical example is used in Section 7.3 to compare the solution obtained by our method with those from

TABLE 2: Comparison among three methodologies for solving an AF refueling station location problem.

	Kim and Kuby [20]	Ventura et al. [18]	Proposed solution approach
Objective function	Maximizing the total traffic flow covered by the station	Maximizing the total traffic flow covered by the station	Maximizing the total traffic flow covered by the station
Set of candidate sites	(i) Predetermined (ii) Finite (vertices only)	(i) Transportation network (ii) Infinite (anywhere in the network)	(i) Transportation network (ii) Infinite (anywhere in the network)
Deviation	Yes	No	Yes
Deviation paths	Endpoints of deviation paths must be vertices	N/A	One endpoint of a deviation path must be a vertex and the other can be any point along the network
Main constraints	(i) Number of refueling stations (ii) Vehicle driving range (iii) Distance decay functions for traffic flow rate on deviation paths	(i) Setting up an initial refueling station (ii) Vehicle driving range (iii) Positive traffic flow rate for some short trips	(i) Setting up an initial refueling station (ii) Vehicle driving range (iii) Portion of drivers who select the deviation option
Global optimum	No	Yes	Yes
Computational complexity	NP-hard	Polynomial ( $O(n^4)$ )*	Polynomial ( $O(n^5)$ )*

\*  $n$  = number of vertices.

the Kim and Kuby [20] model and the Ventura et al. [18] algorithm. Table 2 provides a detailed comparison of these three methodologies. Note that these three methodologies have the same objective function, that is, maximizing the total traffic flow covered by the station, but different settings. Section 7.3 uses a numerical example to compare the quality of the solutions obtained by these three methods.

The rest of this article is organized as follows. Section 2 briefly reviews literature related to refueling station location problems. Section 3 includes the problem statement and derives a property related to the deviation option. Section 4 identifies a set of candidate points for potential locations of the refueling station. Section 5 determines optimal points from the set of candidate points for the refueling station that covers the maximum traffic flow when a portion of drivers select the deviation option. Section 6 builds the complete set of optimal solutions by identifying additional optimal points located in the interior of paths whose endpoints belong to the preliminary set of optimal points. Section 7 provides a numerical example to illustrate the proposed solution approach, examine the coupled effects of deviation portion and vehicle driving range on the set of optimal locations and the maximum traffic flow covered, and analyze its performance. Lastly, Section 8 presents a summary of this study and a list of topics for future research.

## 2. Literature Review

While classical facility location problems regard demand as a weight at vertices representing number of customers that need to receive service at the new facilities or number of trips to these facilities, refueling station location problems require a different strategy to deal with demand. Drivers

(demand) on toll roads and highways are able to refuel their vehicles if they pass by a refueling station on their preplanned paths. Considering that this refueling activity for a driver is a unit of demand, Hodgson [21] and Berman et al. [22] introduced the flow-capturing location model (FCLM) to find the facility locations with the objective of maximizing the traffic flow covered. Kuby and Lim [23] extended the FCLM and suggested the flow refueling location model (FRLM) that incorporates a driving range. Since the concept of this model is well suited for identifying optimal refueling station sites for AF vehicles, many subsequent studies have been published using the idea of the FRLM. In particular, Kim and Kuby [20] proposed the deviation-flow refueling location model (DFRLM), which is an extension of the FRLM considering driver deviation behavior when searching for potential sites for the refueling stations. Later, Kim and Kuby [24] suggested two heuristics to solve the DFRLM efficiently.

Recently, refueling station location models have been extended to more general forms considering specific applications, including a multiperiod planning formulation for the allocation of electric charging stations over time [25]; a two-stage stochastic station location model, where the first stage locates permanent stations and the second locates portable stations [26]; a new model that considers multiple deviation paths between each of the O/D pairs when searching for station locations [27]; novel station location formulations for symmetric transportation networks that consider both single- and dual-access candidate sites [28, 29]; a biobjective model to estimate the greenhouse gases emissions reduction as a function of the refueling infrastructure budget [10]; and models that consider capacitated stations and traffic deviations over multiple time periods [30], and traffic deviation considering route choice and demand uncertainty [31], under

the assumption that vehicles only require one refueling stop for intracity trips.

While most refueling station location problems and extensions assume that the station can be located only either on the vertices of the network or within a finite set of candidate sites, Ventura et al. [18] considered the continuous version of the problem for a single station, where the station can be located anywhere along a tree-type transportation network, and developed a polynomial time algorithm to find the set of optimal location points.

### 3. Problem Statement

Now, we address the continuous deviation-flow location problem on a tree network. Considering that a portion of drivers are willing to deviate from their paths, the objective of this problem is to determine the set of locations for an AF refueling station that maximizes the total traffic flow covered (in round trips per time unit) by the station. To simplify but accurately formulate and solve the problem, we consider the following assumptions. First, an uncapacitated refueling station can provide service to all vehicles driving through both directions of the road segment where it is located. It assures that all vehicles passing through the refueling station in either direction of the road segment can be refueled at the station. Second, all vehicles have the same fuel tank size, and fuel consumption is a linear function of the driving distance. This means that all vehicles have the same driving range per refueling, denoted as  $R$ , regardless of road conditions, climate, congestion, or any other variables. Third, vehicles perform a complete round trip between their origin and destination points on the network. A complete round trip in this assumption ensures that every vehicle goes back to its origin after arriving at its intended destination. A trip from origin to destination is called an original trip, and a trip from destination to origin is called a return trip. A round trip consists of an original trip and a return trip. Fourth, each vehicle enters and exits the network with a fuel tank that is at least half-full. This assumption was first suggested by Kuby and Lim [23] and since then has been followed by most of the researchers on this type of problems, considering that information about the actual fuel tank level of vehicles at origins and destinations is difficult to obtain or likely to be inaccurately estimated. The purpose of this assumption is to make sure that the same round trip can be repeated many times without running out of fuel along the path. This assumption assures that if a vehicle can access the last refueling station on the original trip and reach the destination along the path with at least a half-full tank, then this vehicle can also make the return trip from the destination starting with a half-full tank and reaching the same refueling station without running out of fuel. Similarly, if a vehicle at the last station on the return trip can reach the origin along the path with at least a half-full tank, then this vehicle can also make the original trip from the origin with a half-full tank and safely arrive at the same station. Fifth, the deviation option from a simple path is available to all drivers for their refueling service. However, only a certain portion  $\alpha$  of the drivers selects this option, where  $0 \leq \alpha \leq 1$ . This implies that

$\alpha \times 100\%$  of traffic flow deviates from their simple paths for refueling service. We call this flow the deviation-flow.

Let  $T(V, E)$  be an undirected tree network consisting of a set  $V$  with  $n$  vertices and a set  $E$  with  $n - 1$  edges, where  $n \geq 2$ ; otherwise, the network is trivial. An edge  $(v_i, v_j) \in E$  is defined if  $v_i \in V$  and  $v_j \in V$  are directly connected. We also denote  $P(v_i, v_j)$  as the unique simple path between  $v_i$  and  $v_j$  for  $i < j$ , for all  $v_i, v_j \in V$ . Set  $L$  is defined as the set of all possible paths in  $T$ ; that is,  $L = \{P(v_i, v_j) \mid i < j, \text{ for all } v_i, v_j \in V\}$ . The average traffic flow along  $P(v_i, v_j)$  is denoted as  $f(v_i, v_j)$ . We note that  $f(v_i, v_j)$  refers to the average of traffic flow of the original trip from  $v_i$  to  $v_j$  and the return trip from  $v_j$  to  $v_i$  and is defined only for  $i < j$ . We use  $f(v_i, v_j)$  to represent the round trips per time unit between  $v_i$  and  $v_j$ . The length of  $P(v_i, v_j)$  is denoted as  $d(v_i, v_j)$ , and  $d(v_i, v_j) = d(v_j, v_i)$ . Similarly,  $P(v_i, x)$  denotes the unique simple path between  $v_i \in V$  and any point  $x \in T$ , and the length of this path is denoted as  $d(v_i, x) = d(x, v_i)$ .

To address the continuous deviation-flow location problem, we distinguish between the traffic flow and the deviation-flow on a path. The traffic flow between  $v_i$  and  $v_j$  is "covered" by a refueling station if the station is located at some point  $x$  along  $P(v_i, v_j)$  within a distance  $R/2$  from  $v_i$  and  $v_j$ . Then, the set of paths with positive traffic flow covered by a point  $x \in T$ , denoted as  $S(x)$ , is defined as follows:

$$S(x) = \left\{ P(v_i, v_j) \mid x \in P(v_i, v_j), f(v_i, v_j) > 0, d(v_i, x) \leq \frac{R}{2}, d(x, v_j) \leq \frac{R}{2}, i < j, \forall v_i, v_j \in V \right\}. \quad (1)$$

Alternatively, the deviation-flow originating in path  $P(v_i, v_j)$  is covered by the station if it is placed at a point  $x$  somewhere in the network within a distance  $R/2$  from  $v_i$  and  $v_j$ , but not along path  $P(v_i, v_j)$ . Then, the set of paths with positive deviation-flow covered by a point  $x \in T$ , denoted as  $S_D(x)$ , is defined as follows:

$$S_D(x) = \left\{ P(v_i, v_j) \mid x \in T \setminus P(v_i, v_j), \alpha \times f(v_i, v_j) > 0, d(v_i, x) \leq \frac{R}{2}, d(x, v_j) \leq \frac{R}{2}, i < j, \forall v_i, v_j \in V \right\}. \quad (2)$$

Now, the total traffic flow (in round trips per time unit), including  $f(v_i, v_j)$  and  $\alpha \times f(v_i, v_j)$ , for all  $v_i, v_j \in V$ , covered by a refueling station at  $x \in T$ , denoted as  $F(x)$ , is calculated as follows:

$$F(x) = \sum_{P(v_i, v_j) \in S(x)} f(v_i, v_j) + \sum_{P(v_i, v_j) \in S_D(x)} \alpha \times f(v_i, v_j). \quad (3)$$

Thus, an optimal point  $x^*$  that covers the maximum traffic flow (in round trips per time unit) in tree  $T$  can be obtained by comparing the values of  $F(x)$  for all  $x \in T$ ; that is,  $x^* = \arg \max\{F(x) \mid x \in T\}$ .



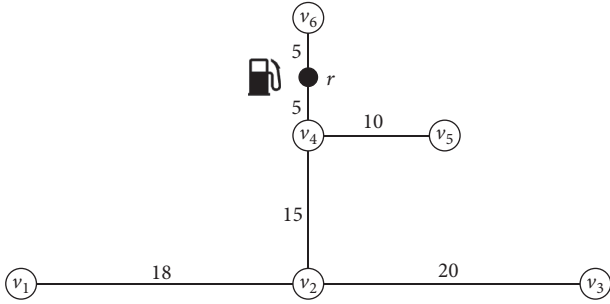


FIGURE 1: Simple deviation path.

When a refueling station is not located in a simple path, vehicles traveling along this path can deviate to a separate subpath for their refueling service. In a deviation path, this separate subpath is called a symmetric cycle, because the subpath begins at a vertex in the simple path, reaches the station location, and returns to the same vertex using the same subpath in the opposite direction. This closed walk includes at least one repeating vertex. Since we consider a refueling station to be located on the tree network, vehicles can take a symmetric cycle originating at a vertex located on their simple path and the entire path is called a simple deviation path. That is, each simple deviation path consists of a simple path and a symmetric cycle.

Figure 1 shows an example of an undirected tree network to help understand the concept of simple deviation path, as well as sets  $S(x)$  and  $S_D(x)$ . The numerical value next to each edge of the network indicates the distance between the two vertices or between a vertex and a point. In this example, let us assume  $R = 80$  and  $\alpha = 0.20$ . If drivers make a round trip between  $v_1$  and  $v_3$  with  $f(v_1, v_3) = 100$ , the simple path is  $P(v_1, v_3)$ , the vertex visitation sequence between  $v_1$  and  $v_3$  is  $\{v_1, v_2, v_3\}$ , and 100 round trips (per time unit) use that vertex visitation sequence. However, if a refueling station is located at point  $r$  in the middle of edge  $(v_4, v_6)$ , 20 round trips from  $f(v_1, v_3)$  deviate to a symmetric cycle with vertex visitation sequence  $\{v_2, v_4, r, v_4, v_2\}$ . Thus, the entire vertex visitation sequence for the simple deviation path between  $v_1$  and  $v_3$  is  $\{v_1, v_2, v_4, r, v_4, v_2, v_3\}$ . If there exists another traffic flow  $f(v_4, v_6) = 400$ , then  $S(r) = \{P(v_4, v_6)\}$  and  $S_D(r) = \{P(v_1, v_3)\}$ . Thus, the total traffic flow covered by a refueling station located at  $r$  is calculated as  $F(r) = \sum_{P(v_i, v_j) \in S(r)} f(v_i, v_j) + \sum_{P(v_i, v_j) \in S_D(r)} \alpha \times f(v_i, v_j) = f(v_4, v_6) + 0.20 \times f(v_1, v_3) = 420$  round trips (per time unit).

If there exists an edge in the network whose length is greater than  $R$ , no traffic flow going through the edge can be covered by a single refueling station due to the half-full tank assumption of vehicles at the origin and destination vertices in their round trips. Furthermore, no deviation-flow going through this edge can be covered by a station located in any symmetric cycle. Thus, we need to split the tree network into subtrees by removing all edges whose distance is greater than  $R$  and solve the station location problem in each subtree separately. Let  $\hat{E} = \{(v_i, v_j) \mid d(v_i, v_j) > R, \text{ for } (v_i, v_j) \in E\}$ . Then, we eliminate all edges in  $\hat{E}$  from  $T(V, E)$  to form a forest

$F(V_F, E_F)$  with  $V_F = V$  and  $E_F = E \setminus \hat{E}$ , such that  $F(V_F, E_F) = \bigcup_{q=1}^t T_q(V_q, E_q)$ , where  $T_q$  is the  $q$ th subtree that consists of a set of vertices  $V_q$  and a set of edges  $E_q$ , and  $t = |\hat{E}| + 1$  is the total number of subtrees  $T_q$  in forest  $F(V_F, E_F)$ .

Now, for subtree  $T_q$ ,  $q = 1, \dots, t$ , if there exists a symmetric cycle originating at a vertex located on a simple path in  $T_q$  that can reach some segments in the removed edges in  $\hat{E}$ , the corresponding deviation-flow can be covered by a point located on one of the segments in these edges. In this case, we need to construct an expanded subtree  $\hat{T}_q$  from  $T_q$  by adding these reachable edges into  $T_q$ . Thus, for each subtree  $T_q$ , we first need to identify all vertices in  $V_q$  that are endpoints of the edges in  $\hat{E}$ . Let  $v_k$  be one of the vertices in  $V_q$  that is adjacent to  $v_l \in V \setminus V_q$  by  $(v_k, v_l) \in \hat{E}$ . Then, the set of vertices  $v_k$  in  $V_q$  is defined as  $V'_q = V(\hat{E}) \cap V_q$ , where  $V(\hat{E})$  is the set of vertices in  $\hat{E}$ . After identifying  $V'_q$ , the set of edges in  $\hat{E}$  that are connected to  $v_k$  in  $V'_q$  is defined as  $E(v_k) = \{(v_k, v_l) \in \hat{E} \mid v_l \in V \setminus V_q\}$ . In addition, the set of vertices  $v_l \in V \setminus V_q$  that are adjacent to  $v_k$  in  $V'_q$  is defined as  $V(v_k) = V(E(v_k)) \setminus \{v_k\}$ , where  $V(E(v_k))$  is the set of vertices in  $E(v_k)$ . For  $v_k \in V'_q$ , if any positive deviation-flow  $\alpha \times f(v_i, v_j)$  from path  $P(v_i, v_j)$  in  $T_q$  can be covered by some point  $x$  located on one of the edges in  $E(v_k)$ , then we add the edge in  $E(v_k)$  into  $E_q$  and the corresponding end point (vertex) in  $V(v_k)$  into  $V_q$ . That is, for subtree  $T_q(V_q, E_q)$ ,  $q = 1, \dots, t$ , we first initialize set  $\bar{V}_q$  as an empty set, where  $\bar{V}_q$  is used to collect all  $v_k \in V'_q$  such that  $\alpha \times f(v_i, v_j)$ ,  $v_i, v_j \in V_q$ , can be covered by some point  $x$  located on one of the edges in  $E(v_k)$ . Next, for  $v_k \in V'_q$ , if we find at least one  $\alpha \times f(v_i, v_j) > 0$ , for  $P(v_i, v_j) \in L$  in  $T_q$ , such that  $\max\{d(v_i, v_k), d(v_j, v_k)\} < R/2$ , then we add  $v_k$  into  $\bar{V}_q$ . Finally, we construct expanded subtree  $\hat{T}_q(\bar{V}_q, \hat{E}_q)$ , such that  $\hat{V}_q = V_q \cup \{\bigcup_{v_k \in \bar{V}_q} V(v_k)\}$  and  $\hat{E}_q = E_q \cup \{\bigcup_{v_k \in \bar{V}_q} E(v_k)\}$ . Based on the expanded subtree  $\hat{T}_q$  constructed above for each subtree  $T_q$ , we can derive the following property.

*Property 1* (expanded subtree property). Let  $x^*$  denote an optimal point to the continuous deviation-flow location problem for an AF refueling station on  $T$ . Regarding the number of expanded subtrees from  $T$  and the location of  $x^*$  as well as the value of  $F(x^*)$ , we consider the following two cases:

- If  $f(v_i, v_j) = 0$ , for all  $P(v_i, v_j) \in L$  such that  $d(v_i, v_j) \leq R$ , then the number of expanded subtrees  $t \leq n$ ,  $x^*$  can be located anywhere in  $T$ , and  $F(x^*) = 0$ .
- Otherwise, the number of expanded subtrees  $t < n$ , and  $x^*$  has to be placed on an expanded subtree  $\hat{T}_q$  that contains some path  $P(v_i, v_j) \in L$  such that  $f(v_i, v_j) > 0$  and  $d(v_i, v_j) \leq R$ . In this case,  $F(x^*) > 0$ .

*Proof.* In case (a), if  $d(v_i, v_j) > R$ , for all  $(v_i, v_j) \in E$ , then  $\hat{E} = E$ . This implies that  $n$  expanded subtrees  $\hat{T}_q$  are created, where each  $\hat{T}_q$  is composed of an isolated vertex. Otherwise, at least

an edge  $(v_i, v_j) \in E$  such that  $d(v_i, v_j) \leq R$  exists. Thus, there exists an expanded subtree  $\hat{T}_q$  that contains edge  $(v_i, v_j) \in \hat{E}_q$ . This implies that  $\hat{T}_q$  includes at least two vertices; that is,  $v_i, v_j \in \hat{V}_q$ ; thus, the number of expanded subtrees  $t < n$ . Besides, since  $f(v_i, v_j) = 0$ , for all  $P(v_i, v_j) \in L$  such that  $d(v_i, v_j) \leq R$ ,  $S(x) = S_D(x) = \emptyset$ , and  $F(x) = 0$ , for all  $x \in T$ . Thus, for any point  $x \in T$ ,  $F(x) = 0$ , which means that  $x^*$  can be located anywhere in  $T$  and  $F(x^*) = 0$ .

In case (b), there exists at most one path  $P(v_i, v_j) \in L$ , such that  $f(v_i, v_j) > 0$  and  $d(v_i, v_j) \leq R$ . Thus, there exists an expanded subtree  $\hat{T}_q$  that contains  $P(v_i, v_j)$  and at least two vertices  $v_i, v_j \in \hat{V}_q$ . This means that the number of expanded subtrees  $t < n$ . If the refueling station is located at a point  $x$  at an isolated vertex in  $\hat{T}_q$ , then  $S(x) = S_D(x) = \emptyset$ , and  $F(x) = 0$ . If the station is located at a point  $x$  along a symmetric cycle of  $P(v_i, v_j)$ , then  $P(v_i, v_j) \in S_D(x)$ ; that is,  $S_D(x) \neq \emptyset$ , and  $F(x) \geq \alpha \times f(v_i, v_j) \geq 0$ . If the station is in the middle point  $x$  of  $P(v_i, v_j)$ , then  $P(v_i, v_j) \in S(x)$ , which indicates that  $S(x) \neq \emptyset$ , and therefore,  $F(x) \geq f(v_i, v_j) > 0$ . Since  $x^*$  is an optimal point,  $F(x^*) \geq F(x) \geq f(v_i, v_j) \geq \alpha \times f(v_i, v_j) \geq 0$ . Hence,  $x^*$  has to be placed on an expanded subtree  $\hat{T}_q$  that contains a path  $P(v_i, v_j) \in L$  such that  $f(v_i, v_j) > 0$  and  $d(v_i, v_j) \leq R$ . Then,  $F(x^*) > 0$ .  $\square$

Note that, for any expanded subtree  $\hat{T}_q(\hat{V}_q, \hat{E}_q)$ ,  $q = 1, \dots, t$ , a path  $P(v_i, v_j) \in \hat{T}_q$  such that  $d(v_i, v_j) > R$  cannot be covered by a single station; in addition, a path  $P(v_i, v_j) \in \hat{T}_q$  without flow, that is,  $f(v_i, v_j) = 0$ , does not need to be considered to maximize traffic flow coverage. Thus, the set of paths that need to be considered for coverage in  $\hat{T}_q$  is defined as  $\hat{L}_q = \{P(v_i, v_j) \in \hat{T}_q \mid d(v_i, v_j) \leq R, f(v_i, v_j) > 0, \text{ and } i < j, \text{ for all } v_i, v_j \in \hat{V}_q\}$ . Then, by definition of  $S(x)$ , for  $x \in \hat{T}_q$ ,  $x$  covers any path  $P(v_i, v_j)$ , for  $v_i, v_j \in \hat{V}_q$ , such that  $d(v_i, v_j) \leq R$  and  $f(v_i, v_j) > 0$ . Thus,  $S(x) \subseteq \hat{L}_q$ . Similarly, by definition of  $S_D(x)$ , for  $x \in \hat{T}_q \setminus P(v_i, v_j)$ ,  $x$  covers every path  $P(v_i, v_j)$ , for  $v_i, v_j \in \hat{V}_q$ , such that  $d(v_i, v_j) \leq R$  and  $\alpha \times f(v_i, v_j) > 0$ . Therefore,  $S_D(x) \subseteq \hat{L}_q$  as well.

#### 4. Set of Candidate Points

In this section, we establish a set of candidate points for each expanded subtree  $\hat{T}_q$  to locate an AF refueling station on a tree network when a portion of drivers are willing to deviate to be able to refuel their vehicles. Then, we prove that this set includes at least one optimal location for the continuous deviation-flow location problem for an AF refueling station. In addition, if multiple optimal locations exist, these candidate points can be used to generate the entire set of optimal locations.

The rest of this section is organized as follows. For path  $P(v_i, v_j) \in \hat{L}_q$ ,  $q = 1, \dots, t$ , Section 4.1 defines a segment that includes all potential locations for an AF refueling station to cover the traffic flow in the path. Based on each segment identified in Section 4.1, Section 4.2 searches for

points where a symmetric cycle can start. Given any point found in Section 4.2, Section 4.3 introduces an algorithm that identifies the farthest point for each symmetric cycle. By integrating the theoretical background regarding the deviation option suggested in Sections 4.1 to 4.3, Section 4.4 determines the set of candidate points when  $\alpha \times 100\%$  of drivers select the deviation option. Lastly, Section 4.5 proves that this set contains at least one optimal location.

**4.1. Refueling Segment.** In order to obtain the set of candidate points, the first step we need to do is to identify the segment for each simple path  $P(v_i, v_j) \in \hat{L}_q$  that contains all station locations that cover the round trips in the path when the deviation option is not considered. This segment is called the refueling segment of path  $P(v_i, v_j)$  and is denoted as  $RS(v_i, v_j)$ . Since vehicles have a driving range  $R$ , based on Assumption (ii), drivers have to refuel the vehicles within a distance of  $R/2$  from the origin and destination vertices during their round trips by Assumption (iv). Thus,  $RS(v_i, v_j)$  must contain all the points that are within  $R/2$  from  $v_i$  and  $v_j$ . That is,

$$RS(v_i, v_j) = \left\{ x \in P(v_i, v_j) \mid d(v_i, x) \leq \frac{R}{2}, d(x, v_j) \leq \frac{R}{2} \right\}. \quad (4)$$

Depending on the value of  $d(v_i, v_j)$ ,  $RS(v_i, v_j)$  includes one or two endpoints, denoted as  $w_{i,j}^k$ , for  $k = 1, 2$ . If  $d(v_i, v_j) < R$ ,  $RS(v_i, v_j)$  has two different endpoints,  $w_{i,j}^1$  and  $w_{i,j}^2$ ; otherwise,  $w_{i,j}^1 = w_{i,j}^2$ . Then, the set of endpoints of  $RS(v_i, v_j)$ , denoted as  $EP(v_i, v_j)$ , is defined as follows:

$$EP(v_i, v_j) = \left\{ w_{i,j}^k \mid w_{i,j}^k, \text{ for } k = 1, 2, \text{ are endpoints of } RS(v_i, v_j) \right\}. \quad (5)$$

In the example of the round trip between  $v_1$  and  $v_3$  in Figure 1,  $RS(v_1, v_3)$  is identical to  $P(v_1, v_3)$  since  $d(v_1, v_3) < R/2$ . Thus, the two endpoints of  $RS(v_1, v_3)$ ,  $w_{1,3}^1$  and  $w_{1,3}^2$ , are located exactly at  $v_1$  and  $v_3$ ; that is,  $EP(v_1, v_3) = \{v_1, v_3\}$ . Since  $RS(v_1, v_3) = P(v_1, v_3)$ , a station located at any point  $x \in P(v_1, v_3)$  can cover round trips between  $v_1$  and  $v_3$ .

**4.2. Cycle Starting Vertex.** As a next step to build the set of candidate points for  $\hat{T}_q$ ,  $q = 1, \dots, t$ , considering the deviation option, in this subsection, we determine a vertex at which a symmetric cycle begins its deviation from a simple path. This vertex is called a cycle starting vertex and denoted as  $v_s$ . A cycle starting vertex is the only common point (vertex) between the simple path and the symmetric cycle.

In order to identify a cycle starting vertex of a path  $P(v_i, v_j)$ , we examine the minimum remaining travel distance of vehicles at a vertex within  $P(v_i, v_j)$ , as well as the degree of this vertex. First, the minimum remaining travel distance of vehicles at a cycle starting vertex  $v_s$  is denoted as  $\delta(v_i, v_j; v_s)$  and computed as follows:

$$\delta(v_i, v_j; v_s) = \frac{R}{2} - \max \{d(v_i, v_s), d(v_j, v_s)\}. \quad (6)$$

Intuitively,  $\delta(v_i, v_j; v_s)$  measures the minimum remaining travel distance at  $v_s$  that vehicles can drive up when they enter the network either at  $v_i$  in the original trip or at  $v_j$  in the return trip.  $\delta(v_i, v_j; v_s)$  is calculated by subtracting the maximum distance between  $d(v_i, v_s)$  and  $d(v_j, v_s)$  from  $R/2$  because any point within a symmetric cycle must be reachable from both  $v_i$  and  $v_j$  in order for vehicles to make a complete round trip between  $v_i$  and  $v_j$  in the network. In order for vehicles in path  $P(v_i, v_j) \in \widehat{L}_q$  to start a symmetric cycle originating at  $v_s$ , the value of  $\delta(v_i, v_j; v_s)$  must be positive; otherwise, the deviation option is not available at  $v_s$ . The value of  $\delta(v_i, v_j; v_s)$  depends on the location of  $v_s$  and the value of  $d(v_i, v_j)$ . In case of  $v_s \in P(v_i, v_j) \setminus \text{RS}(v_i, v_j)$ ,  $\delta(v_i, v_j; v_s) < 0$ , for any value of  $d(v_i, v_j)$ ; thus, no deviation is available. In case of  $v_s \in \text{RS}(v_i, v_j)$ , if  $R/2 \leq d(v_i, v_j) \leq R$ , then  $\delta(v_i, v_j; v_s) > 0$  for  $v_s$  in the interior of  $\text{RS}(v_i, v_j)$ , and  $\delta(v_i, v_j; v_s) = 0$  for  $v_s$  at the endpoint of  $\text{RS}(v_i, v_j)$ . If  $d(v_i, v_j) < R/2$ , then  $\delta(v_i, v_j; v_s) > 0$  for  $v_s \in V(\text{RS}(v_i, v_j))$ , where  $V(\text{RS}(v_i, v_j))$  is the set of vertices in  $\text{RS}(v_i, v_j)$ .

Next, the degree of a cycle starting vertex  $v_s$  is denoted as  $\text{deg}(v_s)$ . If  $v_s$  of a path  $P(v_i, v_j)$  is placed at either origin  $v_i$  or destination  $v_j$ , that is,  $v_s \in \{v_i, v_j\}$ , then  $v_s$  can be a cycle starting vertex if and only if  $\text{deg}(v_s) \geq 2$ . This implies that  $v_s$  has at least one adjacent edge that does not belong to path  $P(v_i, v_j)$  and a portion of this edge or the entire edge can form a subpath for a symmetric cycle. If  $v_s$  is placed at neither origin  $v_i$  nor destination  $v_j$ , that is,  $v_s \notin \{v_i, v_j\}$ , then  $v_s$  can only be a cycle starting vertex if  $\text{deg}(v_s) \geq 3$ . That is, besides the two subpaths from  $v_s$  to  $v_j$  in the original trip and from  $v_s$  to  $v_i$  in the return trip, at least one more separate subpath connected to  $v_s$  exists to initiate a symmetric cycle.

Since there can exist multiple cycle starting vertices for a path  $P(v_i, v_j) \in \widehat{L}_q$ , we define  $\text{CSV}(v_i, v_j)$  as the set of cycle starting vertices in path  $P(v_i, v_j)$ . Based on the observations of a cycle starting vertex discussed above,  $\text{CSV}(v_i, v_j)$  is determined as follows:

$$\begin{aligned} \text{CSV}(v_i, v_j) &= \{v_s \in V(\text{RS}(v_i, v_j)) \mid \delta(v_i, v_j; v_s) \\ &> 0; \text{deg}(v_s) \geq 2 \text{ if } v_s \in \{v_i, v_j\}, \text{ or } \text{deg}(v_s) \\ &\geq 3 \text{ if } v_s \notin \{v_i, v_j\}\}. \end{aligned} \quad (7)$$

Now, let us recall that  $\text{RS}(v_1, v_3) = P(v_1, v_3)$  in the example of the round trip between  $v_1$  and  $v_3$  in Figure 1. Thus,  $V(\text{RS}(v_1, v_3)) = \{v_1, v_2, v_3\}$ , and by using Expression (6),  $\delta(v_1, v_3; v_1) = 2$ ,  $\delta(v_1, v_3; v_2) = 20$ , and  $\delta(v_1, v_3; v_3) = 2$ . However,  $v_1$  and  $v_3$  cannot be cycle starting vertices of  $P(v_1, v_3)$  because  $\text{deg}(v_1) = \text{deg}(v_3) = 1$ . In contrast,  $v_2$  is a cycle starting vertex of  $P(v_1, v_3)$  because  $\text{deg}(v_2) = 3$ . Thus,  $\text{CSV}(v_1, v_3) = \{v_2\}$ .

**4.3. Cycle Returning Point.** In this subsection, for each symmetric cycle originating at  $v_s \in \text{CSV}(v_i, v_j)$ , we identify the farthest point vehicles can reach before returning to  $v_s$ . This point is called a cycle returning point and denoted as  $r$ . Given that the purpose of deviation from a simple path is to refuel vehicles, the cycle returning point is regarded as the farthest feasible site for the refueling station; that is, drivers that are

willing to deviate can travel up to the cycle returning point, refuel their vehicles at the refueling station, return to their simple path, and continue to their destination.

The cycle returning point has distinct characteristics compared to the cycle starting vertex. First, the cycle returning point is defined for  $0 < \alpha \leq 1$ , while the cycle starting vertex is defined for  $0 \leq \alpha \leq 1$ . Next, the cycle returning point belongs to the symmetric cycle but does not belong to the simple path, while the cycle starting vertex is the only common point between the simple path and the symmetric cycle. In addition, the cycle returning point can be a vertex or any point on an edge, whereas the cycle starting vertex is always a vertex literally.

Multiple symmetric cycles can begin at the same cycle starting vertex according to the network structure. This implies that one cycle starting vertex may lead to multiple cycle returning points. Let  $\text{CRP}(v_i, v_j; v_s)$  be the set of all cycle returning points  $r$  arising from a given  $v_s \in \text{CSV}(v_i, v_j)$ . Then,  $|\text{CRP}(v_i, v_j; v_s)|$  is the number of cycle returning points corresponding to a given  $v_s \in \text{CSV}(v_i, v_j)$ , or equivalently, the number of symmetric cycles originating at a given  $v_s \in \text{CSV}(v_i, v_j)$ . In addition, the number of all cycle returning points of path  $P(v_i, v_j)$  can be computed as  $\sum_{v_s \in \text{CSV}(v_i, v_j)} |\text{CRP}(v_i, v_j; v_s)|$ .

The location of a cycle returning point  $r \in \text{CRP}(v_i, v_j; v_s)$  is determined by comparing the value of  $\delta(v_i, v_j; v_s)$  to the length of the separate subpath originating at  $v_s$ . If the value of  $\delta(v_i, v_j; v_s)$  is less than the length of the separate subpath, then the cycle returning point  $r$  is located at a distance  $\delta(v_i, v_j; v_s)$  from  $v_s$  since drivers at cycle starting vertex  $v_s$  can travel up to this distance from  $v_s$  before refueling their vehicles; otherwise, the cycle returning point  $r$  is located at the leaf (end vertex) of the separate subpath because that vertex is the farthest point from  $v_s$  in the symmetric cycle originating at  $v_s$ .

From the observation above regarding the location of a cycle returning point, we propose a novel algorithm, called the Cycle Returning Point Algorithm, to identify the locations of all cycle returning points  $r \in \text{CRP}(v_i, v_j; v_s)$  for a given cycle starting vertex  $v_s \in \text{CSV}(v_i, v_j)$ . This algorithm systematically explores the edges along separate subpaths originating at  $v_s$  and computes the minimum remaining travel distance at each reachable vertex.

When exploring an edge  $(v_u, v_r)$  along a separate subpath such that  $d(v_i, v_u) < d(v_i, v_r)$  or  $d(v_j, v_u) < d(v_j, v_r)$ , we call  $v_u$  the parent of  $v_r$  and  $v_r$  the child of  $v_u$ . To establish a parent and child relationship between vertices in the course of scanning the edges, we let  $\text{PARENT}(v_r)$  be the set of parents of  $v_r$  and  $\text{CHILDREN}(v_u)$  the set of children of  $v_u$ . In a tree network, every child  $v_r$  has a single parent  $v_u$ , that is,  $\text{PARENT}(v_r) = \{v_u\}$ . In contrast, every parent  $v_u$  can have several children  $v_r$  according to the tree network structure. Thus, set  $\text{CHILDREN}(v_u)$  is determined as follows:

$$\begin{aligned} \text{CHILDREN}(v_u) &= \begin{cases} N(v_u) \setminus V(P(v_i, v_j)), & \text{if } v_u = v_s, \\ N(v_u) \setminus \text{PARENT}(v_u), & \text{otherwise,} \end{cases} \end{aligned} \quad (8)$$



where  $N(v_u)$  is the set of vertices adjacent to  $v_u$ , or equivalently, the neighborhood of  $v_u$ , and  $V(P(v_i, v_j))$  is the set of vertices within  $P(v_i, v_j)$ . In Expression (8), if  $v_u = v_s$ ,  $\text{CHILDREN}(v_u)$  is obtained by removing the vertices within  $P(v_i, v_j)$  from the vertices adjacent to  $v_u$ ; otherwise,  $\text{CHILDREN}(v_u)$  is generated by eliminating the parent of  $v_u$  from the vertices adjacent to  $v_u$ .

The algorithm iterates to explore each edge  $(v_u, v_r)$  along separate subpaths originating at a given vertex  $v_s \in \text{CSV}(v_i, v_j)$  until all cycle returning points  $r \in \text{CRP}(v_i, v_j; v_s)$  are identified. At a particular iteration exploring an edge  $(v_u, v_r)$ , the minimum remaining travel distance of vehicles at  $v_u$ , denoted as  $\delta(v_i, v_j; v_u)$ , is known. That is, if  $v_u = v_s$ , the value of  $\delta(v_i, v_j; v_u)$  is given by Expression (6); otherwise, this value is given from the previous iteration. Then, we can compute the minimum remaining travel distance of vehicles at  $v_r$  as  $\delta(v_i, v_j; v_r) = \delta(v_i, v_j; v_u) - d(v_u, v_r)$ . If  $\delta(v_i, v_j; v_r) < 0$ , a cycle returning point  $r$  is located along edge  $(v_u, v_r)$  such that  $d(v_u, r) = \delta(v_i, v_j; v_u)$ . If  $\delta(v_i, v_j; v_r) = 0$  or if  $\delta(v_i, v_j; v_r) > 0$  and  $\text{deg}(v_r) = 1$ , then  $r$  is placed exactly at  $v_r$ . If  $\delta(v_i, v_j; v_r) > 0$  and  $\text{deg}(v_r) \geq 2$ , then the drivers that are willing to deviate can travel further along the separate subpath; in this case, the algorithm assigns a child  $v_r$  of the current iteration to a new parent  $v_u$  at the next iteration and determines a new child  $v_r$  using Expression (8) to proceed the next iteration.

In order to keep track of a child  $v_r$  from a prior iteration that may become a parent in a subsequent iteration on the search for cycle returning points, this algorithm uses a (infinite) first-in, first-out queue  $Q$  consisting of  $v_r$  such that  $\delta(v_i, v_j; v_r) > 0$  and  $\text{deg}(v_r) \geq 2$ . Two functions are used to manage  $Q$ . The first function named  $\text{enqueue}(Q; v_r)$  places  $v_r$  at the tail of queue  $Q$ . Conversely, the second function named  $\text{dequeue}(Q)$  selects the vertex at the head of queue  $Q$  and eliminates it from  $Q$ . The details of the Cycle Returning Point Algorithm are provided below.

*Algorithm 2* (Cycle Returning Point Algorithm).

*Step 1.* For a given  $v_s \in \text{CSV}(v_i, v_j)$ , initialize  $\text{CRP}(v_i, v_j; v_s) = \emptyset$ . If  $\alpha = 0$ , terminate the algorithm; otherwise, compute  $\delta(v_i, v_j; v_s)$  using Expression (6).

*Step 2.* Initialize  $Q = \emptyset$ .

*Step 3.*  $\text{Enqueue}(Q; v_s)$ .

*Step 4.* Repeat the following substeps as long as there remain vertices in  $Q$ :

*Sub-Step 4.1.*  $v_u = \text{dequeue}(Q)$ .

*Sub-Step 4.2.* Determine  $\text{CHILDREN}(v_u)$  using Expression (8).

*Sub-Step 4.3.* For each  $v_r \in \text{CHILDREN}(v_u)$ , perform the following steps:

*Sub-Step 4.3.1.* Compute  $\delta(v_i, v_j; v_r) = \delta(v_i, v_j; v_u) - d(v_u, v_r)$ .

*Sub-Step 4.3.2.* According to the sign of  $\delta(v_i, v_j; v_r)$ , perform one of the following procedures:

- (a) If  $\delta(v_i, v_j; v_r) < 0$ :  $r$  is located along edge  $(v_u, v_r)$ , such that  $d(v_u, r) = \delta(v_i, v_j; v_u)$ . Also, add  $r$  into  $\text{CRP}(v_i, v_j; v_s)$ .
- (b) If  $\delta(v_i, v_j; v_r) = 0$  or if  $\delta(v_i, v_j; v_r) > 0$  and  $\text{deg}(v_r) = 1$ :  $r$  is located exactly at  $v_r$ . Also, add  $r$  into  $\text{CRP}(v_i, v_j; v_s)$ .
- (c) If  $\delta(v_i, v_j; v_r) > 0$  and  $\text{deg}(v_r) \geq 2$ :  $\text{enqueue}(Q; v_r)$  and set  $\text{PARENT}(v_r) = \{v_u\}$ .

**Theorem 3** (complexity of the Cycle Returning Point Algorithm). *The complexity of the Cycle Returning Point Algorithm is  $O(n)$ , where  $n = |V|$ .*

*Proof.* Step 1 includes initialization of  $\text{CRP}(v_i, v_j; v_s)$ , checking the value of  $\alpha$ , and computation of  $\delta(v_i, v_j; v_s)$ , which takes  $O(1)$ . Initialization of  $Q$  in Step 2 and enqueueing  $v_s$  to  $Q$  in Step 3 also take  $O(1)$ . Step 4 consists of three substeps to identify all  $r \in \text{CRP}(v_i, v_j; v_s)$  for a given  $v_s \in \text{CSV}(v_i, v_j)$ . Sub-Steps 4.1 and 4.2 dequeue  $v_u$  from  $Q$  and generate set  $\text{CHILDREN}(v_u)$ . Sub-Step 4.3 computes  $\delta(v_i, v_j; v_r)$  for  $v_r \in \text{CHILDREN}(v_u)$ , and then according to the sign of  $\delta(v_i, v_j; v_r)$ , it locates a cycle returning point within edge  $(v_u, v_r)$  or enqueues  $v_r$  to  $Q$  for the next iteration. Step 4 iterates until queue  $Q$  is empty. Since every vertex can be enqueued and dequeued at most once for a given  $v_s \in \text{CSV}(v_i, v_j)$ , the operations in Step 4 can be applied to every vertex at most once; thus, the total time devoted to the operations in Step 4 is  $O(n)$ . Therefore, the computational complexity of the Cycle Returning Point Algorithm is  $O(n)$ .  $\square$

Now, we apply the Cycle Returning Point Algorithm to the example in Figure 1 to determine all cycle returning points of  $P(v_1, v_3)$ ; that is, we need to identify  $\text{CRP}(v_1, v_3; v_s)$ , for all  $v_s \in \text{CSV}(v_1, v_3)$ . Recall that  $R = 80$ ,  $\alpha = 0.20$ , and  $v_s \in \text{CSV}(v_1, v_3) = \{v_2\}$ . In Step 1 of the algorithm, for a given cycle starting vertex  $v_2$ , we set  $\text{CRP}(v_1, v_3; v_2) = \emptyset$  and, by Expression (6), compute  $\delta(v_1, v_3; v_2) = 40 - \max\{d(v_1, v_2), d(v_3, v_2)\} = 20$ . In Step 2, we set  $Q = \emptyset$ . In Step 3, we place  $v_2$  at the tail of queue  $Q$ ; that is,  $Q = \{v_2\}$ . Now, we repeat Step 4 until finding all cycle returning points  $r \in \text{CRP}(v_1, v_3; v_2)$ . In Sub-Step 4.1, we set  $v_u = v_2$  and eliminate  $v_2$  from  $Q$ . In Sub-Step 4.2, since  $v_u = v_2 = v_s$ , by Expression (8),  $\text{CHILDREN}(v_2) = N(v_2) \setminus V(P(v_1, v_3)) = \{v_1, v_3, v_4\} \setminus \{v_1, v_2, v_3\} = \{v_4\}$ . For  $v_r = v_4$ , we perform Sub-Step 4.3. In Sub-Step 4.3.1,  $\delta(v_1, v_3; v_4) = \delta(v_1, v_3; v_2) - d(v_2, v_4) = 5$ . Since  $\delta(v_1, v_3; v_4) > 0$  and  $\text{deg}(v_4) = 3$ , we select Procedure (c) in Sub-Step 4.3.2; that is,  $Q = \{v_4\}$  and  $\text{PARENT}(v_4) = \{v_2\}$ . Since  $Q \neq \emptyset$ , we repeat Step 4. Now, in Sub-Step 4.1,  $v_u = v_4$  and  $v_4$  is removed from  $Q$ . In Sub-Step 4.2, by Expression (8),  $\text{CHILDREN}(v_4) = N(v_4) \setminus \text{PARENT}(v_4) = \{v_2, v_5, v_6\} \setminus \{v_2\} = \{v_5, v_6\}$ . For  $v_r \in \{v_5, v_6\}$ , we implement Sub-Step 4.3. For  $v_r = v_5$ , in Sub-Step 4.3.1,  $\delta(v_1, v_3; v_5) = \delta(v_1, v_3; v_4) - d(v_4, v_5) = -5$ . As  $\delta(v_1, v_3; v_5) < 0$ , we select Procedure (a) in Sub-Step 4.3.2; that is, we locate  $r_1$  along edge  $(v_4, v_5)$ , such that

$d(v_4, r_1) = \delta(v_1, v_3; v_4) = 5$ , and add  $r_1$  to  $\text{CRP}(v_1, v_3; v_2)$ , where  $r_1$  indicates the first cycle returning point  $r \in \text{CRP}(v_1, v_3; v_2)$ . By repeating the same process for  $v_r = v_6$ , we locate  $r_2$  along edge  $(v_4, v_6)$ , such that  $d(v_4, r_2) = 5$ , and add  $r_2$  to  $\text{CRP}(v_1, v_3; v_2)$ , where  $r_2$  refers to the second  $r \in \text{CRP}(v_1, v_3; v_2)$ . Now,  $Q = \emptyset$ , so we end the algorithm with  $\text{CRP}(v_1, v_3; v_2) = \{r_1, r_2\}$ .

**4.4. Refueling Subtree.** In this subsection, we determine the refueling subtree for each path  $P(v_i, v_j) \in \widehat{L}_q$  that includes all possible locations for an AF refueling station when a portion of drivers select the deviation option. Then, we identify candidate points in this subtree.

First, let us define symmetric cycle  $\text{SC}(v_i, v_j; v_s, r)$  as the segment consisting of all station locations that cover the deviation-flow of path  $P(v_i, v_j)$  originating at cycle starting vertex  $v_s \in \text{CSV}(v_i, v_j)$  and ending at cycle returning point  $r \in \text{CRP}(v_i, v_j; v_s)$ ; that is,  $\text{SC}(v_i, v_j; v_s, r) = P(v_s, r)$ . Note that any point  $x$  in  $\text{SC}(v_i, v_j; v_s, r) \setminus \{v_s\}$  covers deviation-flow  $\alpha \times f(v_i, v_j)$ , while cycle starting vertex  $v_s$  covers traffic flow  $f(v_i, v_j)$  since  $v_s \in \text{SC}(v_i, v_j; v_s, r) \cap \text{RS}(v_i, v_j)$ .

Now, for any path  $P(v_i, v_j) \in \widehat{L}_q$ , we can construct the refueling subtree, denoted as  $\text{RST}(v_i, v_j)$ , which consists of refueling segment  $\text{RS}(v_i, v_j)$  and all symmetric cycles  $\text{SC}(v_i, v_j; v_s, r)$  of  $P(v_i, v_j)$ . This refueling subtree contains all potential locations for an AF refueling station when a portion of drivers are willing to deviate from the preplanned route  $P(v_i, v_j)$  to be able to refuel their vehicles. That is,

$$\begin{aligned} \text{RST}(v_i, v_j) &= \left\{ \bigcup_{v_s \in \text{CSV}(v_i, v_j)} \left\{ \bigcup_{r \in \text{CRP}(v_i, v_j; v_s)} \text{SC}(v_i, v_j; v_s, r) \right\} \right\} \quad (9) \\ &\cup \text{RS}(v_i, v_j). \end{aligned}$$

In  $\text{RST}(v_i, v_j)$ , the cardinality of  $\text{RS}(v_i, v_j)$  is one, but the cardinality of  $\text{SC}(v_i, v_j; v_s, r)$  is  $\sum_{v_s \in \text{CSV}(v_i, v_j)} |\text{CRP}(v_i, v_j; v_s)|$ , because the number of symmetric cycles in  $\text{RST}(v_i, v_j)$  is equal to the number of cycle returning points  $r \in \text{CRP}(v_i, v_j; v_s)$ , for  $v_s \in \text{CSV}(v_i, v_j)$ .

The amount of traffic flow in  $P(v_i, v_j)$  covered by  $x \in \text{RST}(v_i, v_j)$  depends on the value of  $\alpha$ . If  $\alpha = 1$ , any point  $x$  in  $\text{RST}(v_i, v_j)$  covers  $f(v_i, v_j)$ . If  $0 < \alpha < 1$ , any point  $x$  in  $\text{RST}(v_i, v_j) \setminus \text{RS}(v_i, v_j)$  covers  $\alpha \times f(v_i, v_j)$ , while any point  $x$  in  $\text{RS}(v_i, v_j)$  covers  $f(v_i, v_j)$ . If  $\alpha = 0$ , then  $\text{CRP}(v_i, v_j; v_s) = \emptyset$  by the Cycle Returning Point Algorithm; thus,  $\text{RST}(v_i, v_j) = \text{RS}(v_i, v_j)$ , and any point  $x$  in  $\text{RST}(v_i, v_j)$  covers  $f(v_i, v_j)$ .

The endpoints of  $\text{RS}(v_i, v_j)$  indicate the boundary points defining the segment containing all station locations that can cover the positive traffic flow in path  $P(v_i, v_j)$ . Similarly, the endpoints of  $\text{SC}(v_i, v_j; v_s, r)$ , which are  $v_s$  and  $r$ , represent the boundary points for the segment encompassing all station locations that can cover the positive deviation-flow in path  $P(v_i, v_j)$  originating at cycle starting vertex  $v_s \in \text{CSV}(v_i, v_j)$  and ending at cycle returning point  $r \in \text{CRP}(v_i, v_j; v_s)$ . In  $\text{RST}(v_i, v_j)$ , the endpoints of  $\text{RS}(v_i, v_j)$  and all the endpoints

of  $\text{SC}(v_i, v_j; v_s, r)$  are called the candidate points because these points indicate the boundaries for potential station locations that cover path  $P(v_i, v_j)$  with positive traffic flow or deviation-flow. The set of candidate points of  $\text{RST}(v_i, v_j)$ , denoted as  $\text{CP}(\text{RST}(v_i, v_j))$ , is defined as follows:

$$\begin{aligned} \text{CP}(\text{RST}(v_i, v_j)) &= \left\{ \bigcup_{v_s \in \text{CSV}(v_i, v_j)} \text{CRP}(v_i, v_j; v_s) \right\} \quad (10) \\ &\cup \text{CSV}(v_i, v_j) \cup \text{EP}(v_i, v_j), \end{aligned}$$

where  $\text{CSV}(v_i, v_j) \cup \{\bigcup_{v_s \in \text{CSV}(v_i, v_j)} \text{CRP}(v_i, v_j; v_s)\}$  is the set of endpoints of all symmetric cycles and  $\text{EP}(v_i, v_j)$  is the set of endpoints of the refueling segment of  $P(v_i, v_j)$ . The  $k$ th candidate point of  $\text{RST}(v_i, v_j)$  is denoted as  $c_{i,j}^k$ ,  $k = 1, \dots, \sum_{v_s \in \text{CSV}(v_i, v_j)} |\text{CRP}(v_i, v_j; v_s)| + |\text{CSV}(v_i, v_j)| + |\text{EP}(v_i, v_j)|$ .

In Expression (10), note that  $\text{CRP}(v_i, v_j; v_s) = \emptyset$  when there is no deviation option ( $\alpha = 0$ ), while  $\text{CSV}(v_i, v_j)$  and  $\text{EP}(v_i, v_j)$  are defined for  $0 \leq \alpha \leq 1$ . Note that the vertices in  $\text{CSV}(v_i, v_j)$  still need to be considered although the deviation option is not available, that is, for  $\alpha = 0$ , because some of these points are intersection vertices that may cover traffic flows from multiple paths [32]. Since an intersection vertex within a refueling segment has the same characteristics of a cycle starting vertex, we construct set  $\text{CSV}(v_i, v_j)$  for  $\alpha = 0$  to identify this type of intersection vertices even though the deviation option is not available.

By using Expression (10), the set of candidate points in expanded subtree  $\widehat{T}_q$ ,  $q = 1, \dots, t$ , denoted as  $\text{CP}_q$ , can be defined as follows:

$$\text{CP}_q = \bigcup_{P(v_i, v_j) \in \widehat{L}_q} \text{CP}(\text{RST}(v_i, v_j)). \quad (11)$$

Now, we determine the refueling subtree for path  $P(v_1, v_3)$  in Figure 1. Recall that  $\text{CSV}(v_1, v_3) = \{v_2\}$  and  $\text{CRP}(v_1, v_3; v_2) = \{r_1, r_2\}$ . Then, symmetric cycles of  $P(v_1, v_3)$  are  $\text{SC}(v_1, v_3; v_2, r_1)$  and  $\text{SC}(v_1, v_3; v_2, r_2)$ . Thus,  $\text{RST}(v_1, v_3) = \text{SC}(v_1, v_3; v_2, r_1) \cup \text{SC}(v_1, v_3; v_2, r_2) \cup \text{RS}(v_1, v_3)$ . If we assume  $\widehat{L}_q = \{P(v_1, v_3)\}$ , since  $\alpha = 0.20$ , then any point  $x \in \text{RST}(v_1, v_3) \setminus \text{RS}(v_1, v_3)$  covers 20% of  $f(v_1, v_3)$ , and any point  $x \in \text{RS}(v_1, v_3)$  covers  $f(v_1, v_3)$ . In addition, by Expression (11),  $\text{CP}_q = \text{CRP}(v_1, v_3; v_2) \cup \text{CSV}(v_1, v_3) \cup \text{EP}(v_1, v_3) = \{r_1, r_2\} \cup \{v_2\} \cup \{v_1, v_3\}$ .

**4.5. Candidate Point Optimality Theorem.** In this subsection, we prove that the set of candidate points,  $\text{CP} = \bigcup_{q=1}^t \text{CP}_q$ , contains at least one optimal location to the problem.

**Lemma 4.** Consider a point  $x \in \widehat{T}_q$ , for some  $q = 1, \dots, t$ , such that  $x \notin \text{CP}_q$ . Then,

- if  $x \notin \bigcup_{P(v_i, v_j) \in \widehat{L}_q} \text{RST}(v_i, v_j)$ , then  $S(x) = S_D(x) = \emptyset$  and  $F(x) = 0$ ;
- if  $x \in \bigcup_{P(v_i, v_j) \in \widehat{L}_q} \text{RST}(v_i, v_j)$ , then there exists a candidate point  $c \in \text{CP}_q$  such that  $S(x) \subseteq S(c)$ ,  $S_D(x) \subseteq S_D(c) \cup S(c)$ , and  $F(x) \leq F(c)$ .

*Proof.* In case (a), by definition of refueling subtree,  $x$  does not cover any path in  $\widehat{L}_q$ . Thus, the result holds. In case (b),  $x$  is in the interior of all the refueling subtrees determined by all paths in  $S(x)$  and  $S_D(x)$ . Thus,  $x$  has at least one candidate point on each side. Let  $c_1$  and  $c_2$  be the two closest candidate points on each side of  $x$ . Because no candidate point is located between  $x$  and  $c_i$ ,  $i = 1, 2$ , the paths with positive traffic flow covered by  $x$  are the same paths with positive traffic flow covered by both  $c_1$  and  $c_2$ ; that is,  $S(x) = S(c_1) \cap S(c_2)$ , and thus,  $S(x) \subseteq S(c_i)$ ,  $i = 1, 2$ . On the other hand, any path with positive deviation-flow covered by  $x$  must either be a path with positive deviation-flow covered by both  $c_1$  and  $c_2$  or be a path with positive traffic flow covered by one of the two candidate points ( $c_1$  and  $c_2$ ) and with positive deviation-flow covered by the other candidate point; that is,  $S_D(x) = \{S_D(c_1) \cap S_D(c_2)\} \cup \{S(c_1) \cap S_D(c_2)\} \cup \{S_D(c_1) \cap S(c_2)\}$ . This implies  $S_D(x) \subseteq S_D(c_i) \cup S(c_i)$ ,  $i = 1, 2$ . Thus, for  $0 \leq \alpha \leq 1$ ,  $\sum_{P(v_i, v_j) \in S(x)} f(v_i, v_j) + \sum_{P(v_i, v_j) \in S_D(x)} \alpha \times f(v_i, v_j) \leq \sum_{P(v_i, v_j) \in S(c_i)} f(v_i, v_j) + \sum_{P(v_i, v_j) \in S_D(c_i)} \alpha \times f(v_i, v_j)$ , or equivalently,  $F(x) \leq F(c_i)$ ,  $i = 1, 2$ .  $\square$

**Theorem 5** (Candidate Point Optimality Theorem). *For the continuous deviation-flow location problem for an AF refueling station on tree  $T$ , there exists at least one optimal location  $x^* \in CP$ .*

*Proof (by contradiction).* Suppose that none of the candidate points in CP is an optimal solution. Let  $x^*$  be an optimal point that covers  $F(x^*)$  round trips per time unit. Since  $x^*$  is an optimal location, by case (a) of Lemma 4,  $x^*$  must belong to  $\bigcup_{P(v_i, v_j) \in \widehat{L}_q} RST(v_i, v_j)$ , for some  $q = 1, \dots, t$ . If  $x^* \in \bigcup_{P(v_i, v_j) \in \widehat{L}_q} RST(v_i, v_j)$ , then by case (b) of Lemma 4, there exists a candidate point  $c \in CP_q$  such that  $F(x^*) \leq F(c)$ . This contradicts the initial hypothesis that none of the candidate points in CP is an optimal solution. Thus, there exists at least one candidate point in CP that is an optimal location to the problem.  $\square$

## 5. Deviation-Flow Optimal Candidate Point Algorithm

Based on the Candidate Point Optimality Theorem, in this section the Deviation-Flow Optimal Candidate Point Algorithm is proposed to find the optimal set of candidate points for an AF refueling station that covers the maximum traffic flow (in round trips per time unit) when a portion of drivers select the deviation option. The central idea of this algorithm can be summarized in three steps. Step 1 constructs expanded subtrees  $\widehat{T}_q$ , for  $q = 1, \dots, t$ , from the original tree network  $T$ , as discussed in Section 3. Step 2 determines the set of local maximum candidate points for each  $\widehat{T}_q$ . As shown in Theorem 5, at least one local optimal candidate point is global optimal. Step 3 finds the set of global maximum candidate points of  $T$  by comparing the local maximum candidate points.

This algorithm is a generalization of the Single Refueling Point Algorithm that finds the set of optimal locations for

a refueling station when vehicle deviation is not available [18]. A distinct feature of the proposed algorithm is that it generates the deviation options for all O/D pairs with positive flows when  $0 < \alpha \leq 1$ , which usually produces a larger set of candidate points and consequently a different set of optimal locations. In addition, by changing the value of  $\alpha$  between 0 and 1 in the algorithm, we can analyze the effect of the portion of drivers choosing the deviation option on the optimal set of station locations and traffic flow covered. The steps of the algorithm are specified below.

*Algorithm 6* (Deviation-Flow Optimal Candidate Point Algorithm).

*Step 1.* Construct expanded subtrees  $\widehat{T}_q(\widehat{V}_q, \widehat{E}_q)$ , for  $q = 1, \dots, t$ , from original tree network  $T(V, E)$ .

*Sub-Step 1.1.* Establish set  $\widehat{E} = \{(v_k, v_l) \in E \mid d(v_k, v_l) > R\}$  and eliminate  $(v_k, v_l) \in \widehat{E}$  from  $E$ .

*Sub-Step 1.2.* Determine forest  $F(V_F, E_F)$  that consists of subtrees  $T_q(V_q, E_q)$ , for  $q = 1, \dots, t$ , such that  $V_F = \bigcup_{q=1}^t V_q = V$  and  $E_F = \bigcup_{q=1}^t E_q = E \setminus \widehat{E}$ , where  $t = |\widehat{E}| + 1$ .

*Sub-Step 1.3.* For subtree  $T_q(V_q, E_q)$ ,  $q = 1, \dots, t$ , determine expanded subtree  $\widehat{T}_q(\widehat{V}_q, \widehat{E}_q)$ .

*Sub-Step 1.3.1.* Initialize  $\overline{V}_q = \emptyset$  and generate  $V'_q = V(\widehat{E}) \cap V_q$ .

*Sub-Step 1.3.2.* For  $v_k \in V'_q$ , if there exists at least one  $\alpha \times f(v_i, v_j) > 0$ , for  $P(v_i, v_j) \in L$  in  $T_q$ , such that  $\max\{d(v_i, v_k), d(v_j, v_k)\} < R/2$ , then add  $v_k$  into  $\overline{V}_q$ .

*Sub-Step 1.3.3.* Construct expanded subtree  $\widehat{T}_q(\widehat{V}_q, \widehat{E}_q)$  such that  $\widehat{V}_q = V_q \cup \{v_k \in \overline{V}_q \mid v_k \in V_q\}$  and  $\widehat{E}_q = E_q \cup \{(v_k, v_l) \in E \mid v_l \in V \setminus V_q, V(v_k) = V(E(v_k)) \setminus \{v_k\}, \text{ and } V(E(v_k)) \text{ is the set of vertices in } E(v_k)\}$ .

*Step 2.* For each expanded subtree  $\widehat{T}_q$ ,  $q = 1, \dots, t$ , with  $|\widehat{V}_q| > 1$ , identify the set of local optimal candidate points, denoted as  $CP_q^*$ , in  $CP_q$  (note that, for  $\widehat{T}_q$  with  $|\widehat{V}_q| = 1$ , set  $CP_q^* = \{x_q^*\}$  and  $F(x_q^*) = 0$ , where  $x_q^*$  is located at the only vertex in  $\widehat{T}_q$ ):

*Sub-Step 2.1.* Construct  $\widehat{L}_q = \{P(v_i, v_j) \mid d(v_i, v_j) \leq R, f(v_i, v_j) > 0, \text{ and } i < j, \text{ for all } v_i, v_j \in \widehat{V}_q\}$ .

*Sub-Step 2.2.* Determine the set of candidate points,  $CP_q$ , using Expression (11).

*Sub-Step 2.3.* For all  $c \in CP_q$ , construct  $S(c) = \{P(v_i, v_j) \in \widehat{L}_q \mid c \in P(v_i, v_j), f(v_i, v_j) > 0, d(v_i, c) \leq R/2, \text{ and } d(c, v_j) \leq R/2\}$  and  $S_D(c) = \{P(v_i, v_j) \in \widehat{L}_q \mid c \in T_q \setminus P(v_i, v_j), \alpha \times f(v_i, v_j) > 0, d(v_i, c) \leq R/2, \text{ and } d(c, v_j) \leq R/2\}$ .



*Sub-Step 2.4.* Calculate  $F(c) = \sum_{P(v_i, v_j) \in S(c)} f(v_i, v_j) + \sum_{P(v_i, v_j) \in S_D(c)} \alpha \times f(v_i, v_j)$ , for all  $c \in CP_q$ , then determine the local optimal set of candidate points,  $CP_q^*$ , in  $CP_q$ :

$$CP_q^* = \arg \max \{F(c) \mid c \in CP_q\}. \quad (12)$$

*Step 3.* Find the global optimal set of candidate points, denoted as  $CP^*$ , for the original tree  $T$  and the corresponding maximum total traffic flow in round trips per time unit ( $F^*$ ):

$$\begin{aligned} CP^* &= \arg \max \{F(x^*) \mid x^* \in CP_q^*, q = 1, \dots, t\}, \\ F^* &= F(x^*), \quad \text{for } x^* \in CP^*. \end{aligned} \quad (13)$$

**Theorem 7** (complexity of the Deviation-Flow Optimal Candidate Point Algorithm). *The Deviation-Flow Optimal Candidate Point Algorithm runs in  $O(n^5)$  time, where  $n = |V|$ .*

*Proof.* Step 1 of the algorithm contains three substeps to generate expanded subtrees  $\hat{T}_q(\hat{V}_q, \hat{E}_q)$ , for  $q = 1, \dots, t$ , from original tree  $T(V, E)$ . Sub-Step 1.1 takes  $O(n)$  to find and remove edges  $(v_k, v_l) \in \hat{E}$  from  $E$ . To construct forest  $F(V_F, E_F)$  containing subtrees  $T_q(V_q, E_q)$ , for  $q = 1, \dots, t$ , in Sub-Step 1.2, we start with a forest where each vertex forms its own tree. Then, for each edge  $(v_i, v_k) \in E \setminus \hat{E}$ , two trees connected by an edge  $(v_i, v_k)$  are merged into a single tree. We repeat this iteration for all edges in  $E \setminus \hat{E}$ , thus we obtain the final set of subtrees in  $O(n)$  time. To build expanded subtree  $\hat{T}_q$  from  $T_q$ ,  $q = 1, \dots, t$ , Sub-Step 1.3 takes  $O(n^3)$  to identify  $V'_q$  and compute  $\max\{d(v_i, v_k), d(v_j, v_k)\}$  for  $v_k \in V'_q$  and  $P(v_i, v_j) \in L$  in  $T_q$ . Step 2 consists of four substeps to find the set of local optimal candidate points  $CP_q^*$  for each  $\hat{T}_q$ . Sub-Step 2.1 takes  $O(n^2)$  to measure  $d(v_i, v_j)$ , for all  $v_i, v_j \in \hat{V}_q$ , and build set  $\hat{L}_q$ ,  $q = 1, \dots, t$ . To determine  $CP_q$ ,  $q = 1, \dots, t$ , Sub-Step 2.2 first builds  $RS(v_i, v_j)$  and  $EP(v_i, v_j)$  for  $P(v_i, v_j) \in \hat{L}_q$ , which takes  $O(n^2)$ . Next, for each  $RS(v_i, v_j)$ , Sub-Step 2.2 identifies  $CSV(v_i, v_j)$ , which takes  $O(n)$ . Lastly, for each  $v_s \in CSV(v_i, v_j)$ , Sub-Step 2.2 finds  $CRP(v_i, v_j; v_s)$ , which takes  $O(n)$  by Theorem 3. Thus, the total time devoted to Sub-Step 2.2 is  $O(n^4)$ . Since we can find  $O(n)$  candidate points for each path  $P(v_i, v_j) \in \hat{L}_q$ , and furthermore, there exist  $O(n^2)$  paths in each  $\hat{L}_q$ , the number of candidate points in  $CP_q$  is  $O(n^3)$ . Therefore, Sub-Step 2.3 takes  $O(n^5)$  to generate the sets of paths with positive traffic flow and deviation-flow covered by all candidate points  $c \in CP_q$ ,  $q = 1, \dots, t$ . Similarly, Sub-Step 2.4 takes  $O(n^5)$  to compute the total traffic flow covered by all candidate points, compare the total traffic flows covered by candidate points in  $\hat{T}_q$ ,  $q = 1, \dots, t$ , and determine the local optimal set of candidate points for each  $\hat{T}_q$ . Step 3 takes  $O(n^2)$  to compare the local optimal solutions and determine the global optimal set of candidate points in  $T$ . The algorithm performs each step once. Thus, the total time devoted to operations in the algorithm is  $O(n^5)$ .  $\square$

## 6. Convex Combination Property

This section consists of two subsections to identify additional optimal points that can be generated as convex combinations of certain pairs of optimal candidate points. First, a condition is identified in Section 6.1 under which any point on the line segment between two consecutive optimal candidate points is also optimal. Then, based on this condition, the complete set of optimal solutions for the problem is determined in Section 6.2.

*6.1. Optimality Conditions for a Convex Combination of Consecutive Optimal Candidate Points.* In this subsection, given any pair of consecutive optimal candidate points,  $c_1^*, c_2^* \in CP^*$ , we identify conditions under which any point that can be written as a convex combination of  $c_1^*$  and  $c_2^*$  is also optimal.

To simplify the proof process to derive the conditions, we first partition  $S(c_1^*)$ ,  $S_D(c_1^*)$ ,  $S(c_2^*)$ , and  $S_D(c_2^*)$  into newly defined subsets as shown below. Since there exists no candidate point between  $c_1^*$  and  $c_2^*$ ,  $S(c_1^*)$  is partitioned into  $S_1(c_1^*)$  and  $S_{1,2}(c_1^*)$ , where  $S_1(c_1^*)$  denotes the subset of paths with positive traffic flow covered by  $c_1^*$  but not covered by  $c_2^*$ , and  $S_{1,2}(c_1^*)$  denotes the subset of paths with positive traffic flow covered by both  $c_1^*$  and  $c_2^*$ . We also partition  $S_D(c_1^*)$  into  $S_{D_1}(c_1^*)$  and  $S_{D_{1,2}}(c_1^*)$ , where  $S_{D_1}(c_1^*)$  denotes the subset of paths with positive deviation-flow covered by  $c_1^*$  but not covered by  $c_2^*$ , and  $S_{D_{1,2}}(c_1^*)$  denotes the subset of paths with positive deviation-flow covered by both  $c_1^*$  and  $c_2^*$ .  $S_{D_1}(c_1^*)$  can also be partitioned into  $S_{D_1 \setminus 2}(c_1^*)$  and  $S_{D_1}(c_1^*) \setminus S_{D_1 \setminus 2}(c_1^*)$ , where  $S_{D_1 \setminus 2}(c_1^*)$  denotes the subset of paths with positive deviation-flow covered by  $c_1^*$  and positive traffic flow not covered by  $c_2^*$ , and  $S_{D_1}(c_1^*) \setminus S_{D_1 \setminus 2}(c_1^*)$  denotes the subset of paths with positive traffic flow covered by  $c_2^*$  and positive deviation-flow covered by  $c_1^*$ . Similarly, we can partition  $S(c_2^*)$  and  $S_D(c_2^*)$ , such that  $S(c_2^*) = S_2(c_2^*) \cup S_{1,2}(c_2^*)$ ,  $S_D(c_2^*) = S_{D_2}(c_2^*) \cup S_{D_{1,2}}(c_2^*)$ , and  $S_{D_2}(c_2^*) = S_{D_2 \setminus 1}(c_2^*) \cup \{S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)\}$ . Then, by using these terms,  $F(c_1^*)$  and  $F(c_2^*)$  can be written as follows:

$$\begin{aligned} F(c_1^*) &= \sum_{P(v_i, v_j) \in S_1(c_1^*)} f(v_i, v_j) + \sum_{P(v_i, v_j) \in S_{1,2}(c_1^*)} f(v_i, v_j) \\ &+ \alpha \left( \sum_{P(v_i, v_j) \in S_{D_1 \setminus 2}(c_1^*)} f(v_i, v_j) \right. \\ &+ \sum_{P(v_i, v_j) \in S_{D_1}(c_1^*) \setminus S_{D_1 \setminus 2}(c_1^*)} f(v_i, v_j) \\ &\left. + \sum_{P(v_i, v_j) \in S_{D_{1,2}}(c_1^*)} f(v_i, v_j) \right), \\ F(c_2^*) &= \sum_{P(v_i, v_j) \in S_2(c_2^*)} f(v_i, v_j) + \sum_{P(v_i, v_j) \in S_{1,2}(c_2^*)} f(v_i, v_j) \end{aligned}$$



$$\begin{aligned}
& + \alpha \left( \sum_{P(v_i, v_j) \in S_{D_2}(c_1^*)} f(v_i, v_j) \right. \\
& + \sum_{P(v_i, v_j) \in S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)} f(v_i, v_j) \\
& \left. + \sum_{P(v_i, v_j) \in S_{D_1,2}(c_2^*)} f(v_i, v_j) \right). \tag{14}
\end{aligned}$$

Note also that, by definition,  $S_{1,2}(c_1^*) = S_{1,2}(c_2^*)$  and  $S_{D_1,2}(c_1^*) = S_{D_1,2}(c_2^*)$ . Similarly,  $\{S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)\} \subseteq S_1(c_1^*)$  and  $\{S_{D_1}(c_1^*) \setminus S_{D_1,2}(c_1^*)\} \subseteq S_2(c_2^*)$ .

The following theorem establishes optimality conditions for points that can be written as convex combinations of consecutive optimal candidate points.

**Theorem 8** (optimality conditions for the line segment joining consecutive optimal candidate points). *Given a pair of consecutive optimal candidate points,  $c_1^*, c_2^* \in CP^*$ , let  $\ell(c_1^*, c_2^*)$  denote the line segment joining  $c_1^*$  and  $c_2^*$ ; that is,  $\ell(c_1^*, c_2^*) = \{x \mid x = \beta c_1^* + (1 - \beta)c_2^*, 0 \leq \beta \leq 1\}$ . Then, any point  $x \in \ell(c_1^*, c_2^*)$  is also optimal if and only if one of the following conditions is satisfied:*

- (a)  $S(c_1^*) = S(c_2^*)$ , for  $\alpha = 0$ .
- (b)  $S(c_1^*) = S(c_2^*)$  and  $S_D(c_1^*) = S_D(c_2^*)$ , for  $0 < \alpha < 1$ .
- (c)  $S(c_1^*) \cup S_D(c_1^*) = S(c_2^*) \cup S_D(c_2^*)$ , for  $\alpha = 1$ .

*Proof.* ( $\Rightarrow$ ) Assume that any point  $x \in \ell(c_1^*, c_2^*)$  is also optimal. This implies  $F(c_1^*) = F(x) = F(c_2^*)$ , or equivalently,  $F(c_1^*) - F(x) = F(c_2^*) - F(x) = 0$ . Since  $c_1^*$  and  $c_2^*$  are the two closest candidate points on each side of  $x$ ,  $S(x) = S_{1,2}(c_1^*)$  and  $S_D(x) = S_{D_1,2}(c_1^*) \cup \{S_{D_1}(c_1^*) \setminus S_{D_1,2}(c_1^*)\} \cup \{S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)\}$ . Thus,  $F(x)$  can be written as follows:

$$\begin{aligned}
F(x) & = \sum_{P(v_i, v_j) \in S_{1,2}(c_1^*)} f(v_i, v_j) \\
& + \alpha \left( \sum_{P(v_i, v_j) \in S_{D_1,2}(c_1^*)} f(v_i, v_j) \right. \\
& + \sum_{P(v_i, v_j) \in S_{D_1}(c_1^*) \setminus S_{D_1,2}(c_1^*)} f(v_i, v_j) \\
& \left. + \sum_{P(v_i, v_j) \in S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)} f(v_i, v_j) \right). \tag{15}
\end{aligned}$$

We recall that  $S_{1,2}(c_1^*) = S_{1,2}(c_2^*)$  and  $S_{D_1,2}(c_1^*) = S_{D_1,2}(c_2^*)$ . Thus,  $F(c_1^*) - F(x)$  and  $F(c_2^*) - F(x)$  can be expressed as follows:

$$\begin{aligned}
F(c_1^*) - F(x) & = \sum_{P(v_i, v_j) \in S_1(c_1^*)} f(v_i, v_j) \\
& + \sum_{P(v_i, v_j) \in S_{D_1,2}(c_1^*)} \alpha \times f(v_i, v_j) \\
& - \sum_{P(v_i, v_j) \in S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)} \alpha \\
& \times f(v_i, v_j), \\
F(c_2^*) - F(x) & = \sum_{P(v_i, v_j) \in S_2(c_2^*)} f(v_i, v_j) \\
& + \sum_{P(v_i, v_j) \in S_{D_2 \setminus 1}(c_2^*)} \alpha \times f(v_i, v_j) \\
& - \sum_{P(v_i, v_j) \in S_{D_1}(c_1^*) \setminus S_{D_1,2}(c_1^*)} \alpha \\
& \times f(v_i, v_j). \tag{16}
\end{aligned}$$

In case (a), since  $\alpha = 0$ ,  $S_D(c_1^*) = S_D(c_2^*) = \emptyset$ . Then, the equalities  $F(c_1^*) - F(x) = F(c_2^*) - F(x) = 0$  can only be satisfied if  $S_1(c_1^*) = S_2(c_2^*) = \emptyset$ . This means that  $S(c_1^*) = S_{1,2}(c_1^*) = S(c_2^*)$ .

In case (b),  $0 < \alpha < 1$ . Recall that  $\{S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)\} \subseteq S_1(c_1^*)$  and  $\{S_{D_1}(c_1^*) \setminus S_{D_1,2}(c_1^*)\} \subseteq S_2(c_2^*)$ . This means that  $\sum_{P(v_i, v_j) \in S_1(c_1^*)} f(v_i, v_j) - \sum_{P(v_i, v_j) \in \{S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)\}} \alpha \times f(v_i, v_j) > 0$  in  $F(c_1^*) - F(x)$  and  $\sum_{P(v_i, v_j) \in S_2(c_2^*)} f(v_i, v_j) - \sum_{P(v_i, v_j) \in \{S_{D_1}(c_1^*) \setminus S_{D_1,2}(c_1^*)\}} \alpha \times f(v_i, v_j) > 0$  in  $F(c_2^*) - F(x)$  if  $S_1(c_1^*)$  and  $S_2(c_2^*)$  are not empty. Thus, to satisfy equalities  $F(c_1^*) - F(x) = F(c_2^*) - F(x) = 0$ , it is necessary to have  $S_1(c_1^*) = S_2(c_2^*) = S_{D_1,2}(c_1^*) = S_{D_2 \setminus 1}(c_2^*) = \emptyset$ . Note also that if  $S_1(c_1^*) = S_2(c_2^*) = \emptyset$ , then  $\{S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)\} = \{S_{D_1}(c_1^*) \setminus S_{D_1,2}(c_1^*)\} = \emptyset$  because  $\{S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)\} \subseteq S_1(c_1^*)$  and  $\{S_{D_1}(c_1^*) \setminus S_{D_1,2}(c_1^*)\} \subseteq S_2(c_2^*)$ . Therefore,  $S(c_1^*) = S_{1,2}(c_1^*) = S(c_2^*)$  and  $S_D(c_1^*) = S_{D_1,2}(c_1^*) = S_D(c_2^*)$ .

In case (c), since  $\alpha = 1$  and  $\{S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)\} \subseteq S_1(c_1^*)$ , to satisfy  $F(c_1^*) - F(x) = 0$ , it is necessary to have  $S_{D_1,2}(c_1^*) = \emptyset$  and  $\{S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)\} = S_1(c_1^*)$ . Similarly,  $F(c_2^*) - F(x) = 0$  can only be satisfied if  $S_{D_2 \setminus 1}(c_2^*) = \emptyset$  and  $\{S_{D_1}(c_1^*) \setminus S_{D_1,2}(c_1^*)\} = S_2(c_2^*)$ . These results also imply that  $S(c_1^*) \cup S_D(c_1^*) = S_1(c_1^*) \cup S_{1,2}(c_1^*) \cup S_{D_1,2}(c_1^*) \cup S_2(c_2^*)$  and  $S(c_2^*) \cup S_D(c_2^*) = S_2(c_2^*) \cup S_{1,2}(c_2^*) \cup S_{D_1,2}(c_2^*) \cup S_1(c_1^*)$ . Since  $S_{1,2}(c_1^*) = S_{1,2}(c_2^*)$  and  $S_{D_1,2}(c_1^*) = S_{D_1,2}(c_2^*)$ , we can finally conclude that  $S(c_1^*) \cup S_D(c_1^*) = S(c_2^*) \cup S_D(c_2^*)$ .

( $\Leftarrow$ ) In case (a), we assume  $S(c_1^*) = S(c_2^*)$ . This implies that  $S_1(c_1^*) = S_2(c_2^*) = \emptyset$ . Also, since  $\alpha = 0$ ,  $S_D(c_1^*) = S_D(c_2^*) = \emptyset$ . Thus,  $F(c_1^*) - F(x) = F(c_2^*) - F(x) = 0$ .

In case (b), we assume that  $S(c_1^*) = S(c_2^*)$  and  $S_D(c_1^*) = S_D(c_2^*)$ . This implies that  $S_1(c_1^*) = S_2(c_2^*) = \emptyset$  and  $S_{D_1,2}(c_1^*) = S_{D_2 \setminus 1}(c_2^*) = \emptyset$ . Note also that  $\{S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)\} = \{S_{D_1}(c_1^*) \setminus S_{D_1,2}(c_1^*)\} = \emptyset$  because  $\{S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)\} \subseteq S_1(c_1^*)$ ,

$\{S_{D_1}(c_1^*) \setminus S_{D_1 \setminus 2}(c_1^*)\} \subseteq S_2(c_2^*)$ , and  $S_1(c_1^*) = S_2(c_2^*) = \emptyset$ . Thus,  $F(c_1^*) - F(x) = F(c_2^*) - F(x) = 0$ .

In case (c), we assume that  $S(c_1^*) \cup S_D(c_1^*) = S(c_2^*) \cup S_D(c_2^*)$ . This implies that  $S_1(c_1^*) \cup S_{D_1 \setminus 2}(c_1^*) \cup \{S_{D_1}(c_1^*) \setminus S_{D_1 \setminus 2}(c_1^*)\} = S_2(c_2^*) \cup S_{D_2 \setminus 1}(c_2^*) \cup \{S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)\}$ . Since  $\{S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)\} \subseteq S_1(c_1^*)$  and  $\{S_{D_1}(c_1^*) \setminus S_{D_1 \setminus 2}(c_1^*)\} \subseteq S_2(c_2^*)$ , to satisfy  $S_1(c_1^*) \cup S_{D_1 \setminus 2}(c_1^*) \cup \{S_{D_1}(c_1^*) \setminus S_{D_1 \setminus 2}(c_1^*)\} = S_2(c_2^*) \cup S_{D_2 \setminus 1}(c_2^*) \cup \{S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)\}$ , it is necessary to have  $S_{D_1 \setminus 2}(c_1^*) = S_{D_2 \setminus 1}(c_2^*) = \emptyset$ ,  $S_1(c_1^*) = \{S_{D_2}(c_2^*) \setminus S_{D_2 \setminus 1}(c_2^*)\}$ , and  $S_2(c_2^*) = \{S_{D_1}(c_1^*) \setminus S_{D_1 \setminus 2}(c_1^*)\}$ . As a result,  $F(c_1^*) - F(x) = F(c_2^*) - F(x) = 0$ .

Since the conditions in cases (a), (b), and (c) lead to  $F(c_1^*) = F(x) = F(c_2^*)$ , and furthermore  $c_1^*, c_2^* \in \text{CP}^*$ , any point  $x \in \ell(c_1^*, c_2^*)$  is optimal.  $\square$

**6.2. Complete Set of Optimal Solutions.** In this subsection, we determine the complete set of optimal solutions, denoted as  $\text{CS}^*$ , for the continuous deviation-flow location problem for an AF refueling station.

As an intermediate step toward determining  $\text{CS}^*$ , we first construct the set pairs of consecutive candidate points in  $\text{CP}^*$  that satisfy the conditions stated in Theorem 8, denoted as set  $\overline{\text{CP}}^*$ . Then, by Theorem 8, the line segments defined by these pairs of candidate points in  $\overline{\text{CP}}^*$  are also optimal. Consequently, the complete set of optimal solutions, denoted as  $\text{CS}^*$ , can be defined as follows:

$$\text{CS}^* = \left\{ \bigcup_{(c_1^*, c_2^*) \in \overline{\text{CP}}^*} \ell(c_1^*, c_2^*) \right\} \cup \{\text{CP}^* \setminus V(\overline{\text{CP}}^*)\}, \quad (17)$$

where  $V(\overline{\text{CP}}^*)$  is the set of optimal candidate points in  $\overline{\text{CP}}^*$ . The optimality of set  $\text{CS}^*$  is proved in the following theorem.

**Theorem 9** (optimality of the complete set of optimal solutions). *Set  $\text{CS}^*$ , defined in Expression (17), contains all optimal solutions for the problem.*

*Proof (by contradiction).* Suppose that there exists an optimal point  $x^*$  that does not belong to  $\text{CS}^*$ . Because  $\text{CS}^*$  includes all optimal candidate points in  $\text{CP}^*$  and  $x^* \notin \text{CS}^*$ ,  $x^*$  should be a point in  $T$  that is not a candidate point. First, suppose that  $x^* \notin \bigcup_{P(v_i, v_j) \in \tilde{L}_q} \text{RST}(v_i, v_j)$ . Then, by case (a) of Lemma 4,  $F(x^*) = 0$ . This contradicts the hypothesis that  $x^*$  is optimal. Thus,  $x^*$  must belong to  $\bigcup_{P(v_i, v_j) \in \tilde{L}_q} \text{RST}(v_i, v_j)$ . Since  $x^* \in \bigcup_{P(v_i, v_j) \in \tilde{L}_q} \text{RST}(v_i, v_j)$ , by case (b) of Lemma 4, there exist two closest candidate points,  $c_1$  and  $c_2$ , on each side of  $x^*$ , such that  $S(x^*) \subseteq S(c_i)$ ,  $S_D(x^*) \subseteq S_D(c_i) \cup S(c_i)$ , and  $F(x^*) \leq F(c_i)$ ,  $i = 1, 2$ . Because  $x^*$  is optimal,  $c_1$  and  $c_2$  are also optimal, and therefore, Theorem 8 holds for the convex combination of  $c_1$  and  $c_2$ . This means that  $x^* \in \ell(c_1, c_2) \in \text{CS}^*$ . This contradicts the initial hypothesis that there exists an optimal point  $x^*$  that does not belong to  $\text{CS}^*$ . Thus,  $\text{CS}^*$  contains all the optimal solutions.  $\square$

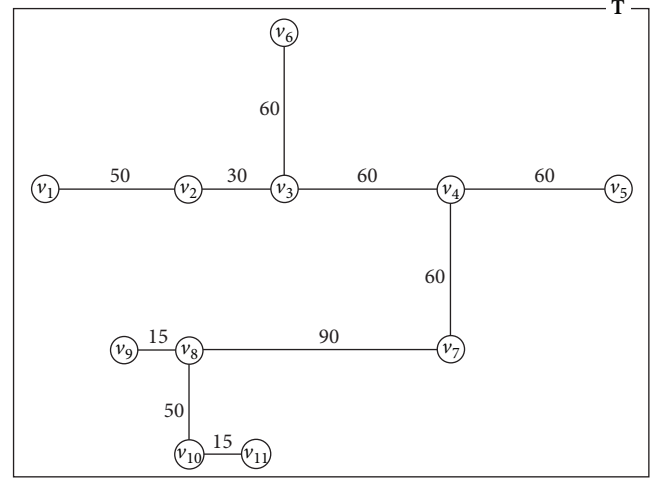


FIGURE 2: Structure of undirected tree  $T$ .

## 7. Numerical Example and Analysis of Results

In this section, a numerical example is provided to illustrate the proposed solution approach and verify its performance. Figure 2 shows the undirected tree  $T(V, E)$  with  $n = 11$  vertices, and Table 3 provides the average traffic flows  $f(v_i, v_j)$ , for all  $v_i, v_j \in V$  such that  $i < j$ . Note that although we use a small-size network to show the step-by-step application of the proposed procedure to find the optimal set of station locations, the procedure can also be applied for the large-scale problems due to its polynomial time complexity. In addition, since the actual portion of drivers who select the deviation portion  $\alpha$  is difficult to find in practice, in this example, we change  $\alpha$  from 0 to 1 to analyze its effect on the optimal set of location sites.

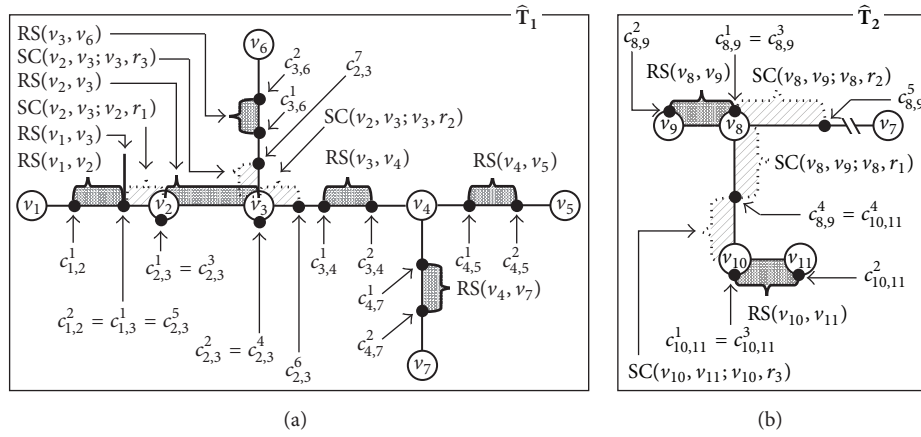
The remainder of this section is organized as follows. In Section 7.1, for a given value of the vehicle driving range  $R$ , we apply the Deviation-Flow Optimal Candidate Point Algorithm to solve the numerical example. In Section 7.2, we further examine the coupled effects of deviation portion and vehicle driving range on the set of optimal locations and the maximum traffic flow covered. In Section 7.3, we compare the results of the proposed solution approach with the existing models to verify its performance.

**7.1. Numerical Example for a Given Vehicle Driving Range.** In this subsection, we fix the driving range  $R = 80$  and solve the numerical example to illustrate the Deviation-Flow Optimal Candidate Point Algorithm in detail.

To construct the expanded subtrees from  $T$  in Step 1, Sub-Step 1.1 finds  $d(v_7, v_8) > R$ ; thus,  $\hat{E} = \{(v_7, v_8)\}$ . By removing  $(v_7, v_8)$  from  $T$ , Sub-Step 1.2 forms forest  $F(V_F, E_F)$  containing two subtrees  $T_1(V_1, E_1)$  and  $T_2(V_2, E_2)$ , where  $V_1 = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ ,  $E_1 = \{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_5), (v_3, v_6), (v_4, v_7)\}$ ,  $V_2 = \{v_8, v_9, v_{10}, v_{11}\}$ , and  $E_2 = \{(v_8, v_9), (v_8, v_{10}), (v_{10}, v_{11})\}$ . For  $T_1$ , Sub-Step 1.3 sets  $\bar{V}_1 = \emptyset$  and generates  $V_1' = V(\hat{E}) \cap V_1 = \{v_7\}$ . For  $v_k \in V_1'$ , even if  $\alpha > 0$ , there is no  $\alpha \times f(v_i, v_j) > 0$ , for  $v_i, v_j \in V_1$ , such that  $\max\{d(v_i, v_k), d(v_j, v_k)\} < R/2$ . Thus,  $\bar{V}_1$  remains

TABLE 3: Traffic flows  $f(v_i, v_j)$  (in round trips per time unit) corresponding to all simple paths  $P(v_i, v_j)$  in  $T$ .

	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6$	$v_7$	$v_8$	$v_9$	$v_{10}$	$v_{11}$
$v_1$	—	80	40	30	20	0	10	8	5	3	0
$v_2$	—	—	70	60	30	15	20	17	14	9	1
$v_3$	—	—	—	90	60	50	10	9	6	5	0
$v_4$	—	—	—	—	100	30	155	100	80	50	15
$v_5$	—	—	—	—	—	10	20	16	14	15	12
$v_6$	—	—	—	—	—	—	6	5	1	3	0
$v_7$	—	—	—	—	—	—	—	90	85	82	80
$v_8$	—	—	—	—	—	—	—	—	140	0	0
$v_9$	—	—	—	—	—	—	—	—	—	0	0
$v_{10}$	—	—	—	—	—	—	—	—	—	—	80
$v_{11}$	—	—	—	—	—	—	—	—	—	—	—

FIGURE 3: Refueling segments, symmetric cycles, and candidate points in  $\hat{T}_1$  and  $\hat{T}_2$ .

empty, and expanded subtree  $\hat{T}_1(\hat{V}_1, \hat{E}_1) = T_1(V_1, E_1)$ . For  $T_2$ , Sub-Step 1.3 sets  $\bar{V}_2 = \emptyset$  and forms  $V_2' = V(\hat{E}) \cap V_2 = \{v_8\}$ . For  $v_k \in V_2'$ , if  $\alpha > 0$ , there exists  $\alpha \times f(v_8, v_9) > 0$ , such that  $\max\{d(v_8, v_k), d(v_9, v_k)\} = 15 < R/2$ . Thus,  $v_8$  is added to  $\bar{V}_2$ , and expanded subtree  $\hat{T}_2(\bar{V}_2, \hat{E}_2)$  is such that  $\hat{V}_2 = V_2 \cup V(v_8)$  and  $\hat{E}_2 = E_2 \cup E(v_8)$ , where  $E(v_8) = \{(v_7, v_8)\}$  and  $V(v_8) = V(E(v_8)) \setminus \{v_8\} = \{v_7\}$ .

Now, for expanded subtree  $\hat{T}_q, q = 1, 2$ , Step 2 determines the set of local optimal candidate points. For  $\hat{T}_1$ , Sub-Step 2.1 builds set  $\hat{L}_1$  that consists of  $P(v_i, v_j)$  in  $\hat{T}_1$  such that  $d(v_i, v_j) \leq 80$  and  $f(v_i, v_j) > 0$ ; that is,  $\hat{L}_1 = \{P(v_1, v_2), P(v_1, v_3), P(v_2, v_3), P(v_3, v_4), P(v_3, v_6), P(v_4, v_5), P(v_4, v_7)\}$ . By using Expression (II), Sub-Step 2.2 establishes  $CP_1$ , which refers to the set of candidate points of the refueling subtrees corresponding to the paths in  $\hat{L}_1$ . For  $P(v_2, v_3)$ , since  $CSV(v_2, v_3) = \{v_2, v_3\}$ ,  $\alpha \times 100\%$  of traffic flow in  $P(v_2, v_3)$  can select the deviation option that starts from  $v_2$  or  $v_3$ . By applying the Cycle Returning Point Algorithm to  $v_2$  and  $v_3$ , three cycle returning points,  $r_1, r_2$ , and  $r_3$ , are obtained such that  $r_1 \in CRP(v_2, v_3; v_2)$  and  $r_2, r_3 \in CRP(v_2, v_3; v_3)$ . Thus,  $RST(v_2, v_3)$ , consisting of  $RS(v_2, v_3)$ ,  $SC(v_2, v_3; v_2, r_1)$ ,  $SC(v_2, v_3; v_3, r_2)$ , and  $SC(v_2, v_3; v_3, r_3)$ , has seven candidate points  $c_{2,3}^k$ , for  $k = 1, \dots, 7$ , such that  $c_{2,3}^1, c_{2,3}^2 \in EP(v_2, v_3)$ ,

$c_{2,3}^3, c_{2,3}^4 \in CSV(v_2, v_3)$ ,  $c_{2,3}^5 \in CRP(v_2, v_3; v_2)$ , and  $c_{2,3}^6, c_{2,3}^7 \in CRP(v_2, v_3; v_3)$ . Note that there are no cycle starting vertices in any path  $P(v_i, v_j) \in \hat{L}_1 \setminus \{P(v_2, v_3)\}$ ; that is, the deviation option is not available. Thus,  $RST(v_i, v_j) = RS(v_i, v_j)$ , and each  $RST(v_i, v_j)$  has one or two candidate points depending on the value of  $d(v_i, v_j)$ . Then,  $CP_1$  includes eighteen candidate points, that is,  $CP_1 = \{c_{1,2}^1, c_{1,2}^2, c_{1,3}^1, c_{1,3}^2, c_{2,3}^1, c_{2,3}^2, c_{2,3}^3, c_{2,3}^4, c_{2,3}^5, c_{2,3}^6, c_{2,3}^7, c_{3,4}^1, c_{3,4}^2, c_{3,6}^1, c_{3,6}^2, c_{4,5}^1, c_{4,5}^2, c_{4,7}^1, c_{4,7}^2\}$ . Figure 3(a) illustrates the locations of  $RS(v_i, v_j)$ ,  $SC(v_i, v_j; v_s, r)$ , and the corresponding  $c \in CP_1$  in  $\hat{T}_1$ , for  $P(v_i, v_j) \in \hat{L}_1, v_s \in CSV(v_i, v_j), r \in CRP(v_i, v_j; v_s)$ . For  $c \in CP_1$ , Sub-Step 2.3 constructs  $S(c)$  and  $S_D(c)$  to identify all positive traffic flows covered by  $c$ . The left and middle sides of Table 4 summarize the results. Based on the paths covered by each candidate point  $c \in CP_1$ , Sub-Step 2.4 first calculates  $F(c)$ , for  $c \in CP_1$ . The right side of this table shows the calculation processes. Sub-Step 2.4 next compares  $F(c)$ , for  $c \in CP_1$ , and determines the local optimal set of candidate points  $CP_1^*$  in  $\hat{T}_1$ , for  $0 \leq \alpha \leq 1$ , as follows:

$$CP_1^* = \arg \max \{F(c_{1,2}^1), F(c_{1,2}^2), F(c_{1,3}^1), F(c_{1,3}^2), F(c_{2,3}^1), F(c_{2,3}^2), F(c_{2,3}^3), F(c_{2,3}^4), F(c_{2,3}^5), F(c_{2,3}^6), F(c_{2,3}^7)\},$$

TABLE 4: Sets of paths and the traffic flows (in round trips per time unit) covered by  $c \in \text{CP}_1$  in  $\hat{T}_1$ .

$c$	$S(c)$	$S_D(c)$	$F(c)$
$c_{1,2}^1$	$S(c_{1,2}^1) = \{P(v_1, v_2)\}$	$S_D(c_{1,2}^1) = \emptyset$	$F(c_{1,2}^1) = f(v_1, v_2) = 80$
$c_{1,2}^2$	$S(c_{1,2}^2) = \{P(v_1, v_2), P(v_1, v_3)\}$	$S_D(c_{1,2}^2) = \{P(v_2, v_3)\}$	$F(c_{1,2}^2) = f(v_1, v_2) + f(v_1, v_3) + \alpha \{f(v_2, v_3)\} = 120 + 70\alpha$
$c_{1,3}^1$	$S(c_{1,3}^1) = \{P(v_1, v_2), P(v_1, v_3)\}$	$S_D(c_{1,3}^1) = \{P(v_2, v_3)\}$	$F(c_{1,3}^1) = f(v_1, v_2) + f(v_1, v_3) + \alpha \{f(v_2, v_3)\} = 120 + 70\alpha$
$c_{2,3}^1$	$S(c_{2,3}^1) = \{P(v_2, v_3)\}$	$S_D(c_{2,3}^1) = \emptyset$	$F(c_{2,3}^1) = f(v_2, v_3) = 70$
$c_{2,3}^2$	$S(c_{2,3}^2) = \{P(v_2, v_3)\}$	$S_D(c_{2,3}^2) = \emptyset$	$F(c_{2,3}^2) = f(v_2, v_3) = 70$
$c_{2,3}^3$	$S(c_{2,3}^3) = \{P(v_2, v_3)\}$	$S_D(c_{2,3}^3) = \emptyset$	$F(c_{2,3}^3) = f(v_2, v_3) = 70$
$c_{2,3}^4$	$S(c_{2,3}^4) = \{P(v_2, v_3)\}$	$S_D(c_{2,3}^4) = \emptyset$	$F(c_{2,3}^4) = f(v_2, v_3) = 70$
$c_{2,3}^5$	$S(c_{2,3}^5) = \{P(v_1, v_2), P(v_1, v_3)\}$	$S_D(c_{2,3}^5) = \{P(v_2, v_3)\}$	$F(c_{2,3}^5) = f(v_1, v_2) + f(v_1, v_3) + \alpha \{f(v_2, v_3)\} = 120 + 70\alpha$
$c_{2,3}^6$	$S(c_{2,3}^6) = \emptyset$	$S_D(c_{2,3}^6) = \{P(v_2, v_3)\}$	$F(c_{2,3}^6) = \alpha \times f(v_2, v_3) = 70\alpha$
$c_{2,3}^7$	$S(c_{2,3}^7) = \emptyset$	$S_D(c_{2,3}^7) = \{P(v_2, v_3)\}$	$F(c_{2,3}^7) = \alpha \times f(v_2, v_3) = 70\alpha$
$c_{3,4}^1$	$S(c_{3,4}^1) = \{P(v_3, v_4)\}$	$S_D(c_{3,4}^1) = \emptyset$	$F(c_{3,4}^1) = f(v_3, v_4) = 90$
$c_{3,4}^2$	$S(c_{3,4}^2) = \{P(v_3, v_4)\}$	$S_D(c_{3,4}^2) = \emptyset$	$F(c_{3,4}^2) = f(v_3, v_4) = 90$
$c_{3,6}^1$	$S(c_{3,6}^1) = \{P(v_3, v_6)\}$	$S_D(c_{3,6}^1) = \emptyset$	$F(c_{3,6}^1) = f(v_3, v_6) = 50$
$c_{3,6}^2$	$S(c_{3,6}^2) = \{P(v_3, v_6)\}$	$S_D(c_{3,6}^2) = \emptyset$	$F(c_{3,6}^2) = f(v_3, v_6) = 50$
$c_{4,5}^1$	$S(c_{4,5}^1) = \{P(v_4, v_5)\}$	$S_D(c_{4,5}^1) = \emptyset$	$F(c_{4,5}^1) = f(v_4, v_5) = 100$
$c_{4,5}^2$	$S(c_{4,5}^2) = \{P(v_4, v_5)\}$	$S_D(c_{4,5}^2) = \emptyset$	$F(c_{4,5}^2) = f(v_4, v_5) = 100$
$c_{4,7}^1$	$S(c_{4,7}^1) = \{P(v_4, v_7)\}$	$S_D(c_{4,7}^1) = \emptyset$	$F(c_{4,7}^1) = f(v_4, v_7) = 155$
$c_{4,7}^2$	$S(c_{4,7}^2) = \{P(v_4, v_7)\}$	$S_D(c_{4,7}^2) = \emptyset$	$F(c_{4,7}^2) = f(v_4, v_7) = 155$

TABLE 5: Sets of paths and the traffic flows (in round trips per time unit) covered by  $c \in \text{CP}_2$  in  $\hat{T}_2$ .

$c$	$S(c)$	$S_D(c)$	$F(c)$
$c_{8,9}^1$	$S(c_{8,9}^1) = \{P(v_8, v_9)\}$	$S_D(c_{8,9}^1) = \emptyset$	$F(c_{8,9}^1) = f(v_8, v_9) = 140$
$c_{8,9}^2$	$S(c_{8,9}^2) = \{P(v_8, v_9)\}$	$S_D(c_{8,9}^2) = \emptyset$	$F(c_{8,9}^2) = f(v_8, v_9) = 140$
$c_{8,9}^3$	$S(c_{8,9}^3) = \{P(v_8, v_9)\}$	$S_D(c_{8,9}^3) = \emptyset$	$F(c_{8,9}^3) = f(v_8, v_9) = 140$
$c_{8,9}^4$	$S(c_{8,9}^4) = \emptyset$	$S_D(c_{8,9}^4) = \{P(v_8, v_9), P(v_{10}, v_{11})\}$	$F(c_{8,9}^4) = \alpha \{f(v_8, v_9) + f(v_{10}, v_{11})\} = 220\alpha$
$c_{8,9}^5$	$S(c_{8,9}^5) = \emptyset$	$S_D(c_{8,9}^5) = P(v_8, v_9)$	$F(c_{8,9}^5) = \alpha \times f(v_8, v_9) = 140\alpha$
$c_{10,11}^1$	$S(c_{10,11}^1) = \{P(v_{10}, v_{11})\}$	$S_D(c_{10,11}^1) = \emptyset$	$F(c_{10,11}^1) = f(v_{10}, v_{11}) = 80$
$c_{10,11}^2$	$S(c_{10,11}^2) = \{P(v_{10}, v_{11})\}$	$S_D(c_{10,11}^2) = \emptyset$	$F(c_{10,11}^2) = f(v_{10}, v_{11}) = 80$
$c_{10,11}^3$	$S(c_{10,11}^3) = \{P(v_{10}, v_{11})\}$	$S_D(c_{10,11}^3) = \emptyset$	$F(c_{10,11}^3) = f(v_{10}, v_{11}) = 80$
$c_{10,11}^4$	$S(c_{10,11}^4) = \emptyset$	$S_D(c_{10,11}^4) = \{P(v_8, v_9), P(v_{10}, v_{11})\}$	$F(c_{10,11}^4) = \alpha \{f(v_8, v_9) + f(v_{10}, v_{11})\} = 220\alpha$

$$F(c_{3,4}^1), F(c_{3,4}^2), F(c_{3,6}^1), F(c_{3,6}^2), F(c_{4,5}^1), F(c_{4,5}^2),$$

$$F(c_{4,7}^1), F(c_{4,7}^2)\} = \arg \max \{80, 120 + 70\alpha, 120$$

$$+ 70\alpha, 70, 70, 70, 70, 120 + 70\alpha, 70\alpha, 70\alpha,$$

$$90, 90, 50, 50, 100, 100, 155, 155\} = \{c_{1,2}^2, c_{1,3}^1, c_{2,3}^5, c_{4,7}^1,$$

$$c_{4,7}^2\}.$$

(18)

This result implies that if  $\alpha \geq 0.5$ , then  $c_{1,2}^2$ ,  $c_{1,3}^1$ , and  $c_{2,3}^5$  are the local optimal locations to  $\hat{T}_1$ ; otherwise,  $c_{4,7}^1$  and  $c_{4,7}^2$  are the local optimal locations that cover the maximum traffic flow (in round trips per time unit) in  $\hat{T}_1$ .

Similarly, Step 2 is repeated for  $\hat{T}_2$ . Sub-Step 2.1 constructs  $\hat{L}_2 = \{P(v_8, v_9), P(v_{10}, v_{11})\}$ . Sub-Step 2.2 generates  $\text{EP}(v_8, v_9) = \{v_8, v_9\}$ ,  $\text{EP}(v_{10}, v_{11}) = \{v_{10}, v_{11}\}$ ,  $\text{CSV}(v_8, v_9) = \{v_8\}$ ,

$\text{CSV}(v_{10}, v_{11}) = \{v_{10}\}$ ,  $\text{CRP}(v_8, v_9; v_8) = \{r_1, r_2\}$ , and  $\text{CRP}(v_{10}, v_{11}; v_{10}) = \{r_3\}$ , where  $r_1$ ,  $r_2$ , and  $r_3$  are the three cycle returning points obtained by the Cycle Returning Point Algorithm. Then, by Expression (11),  $\text{CP}_2 = \{c_{8,9}^1, c_{8,9}^2, c_{8,9}^3, c_{8,9}^4, c_{8,9}^5, c_{10,11}^1, c_{10,11}^2, c_{10,11}^3, c_{10,11}^4\}$ . Figure 3(b) graphically shows the locations of  $\text{RS}(v_i, v_j)$ ,  $\text{SC}(v_i, v_j; v_s, r)$ , and the corresponding  $c \in \text{CP}_2$  in  $\hat{T}_2$ , for  $P(v_i, v_j) \in \hat{L}_2$ ,  $v_s \in \text{CSV}(v_i, v_j)$ ,  $r \in \text{CRP}(v_i, v_j; v_s)$ . For  $c \in \text{CP}_2$ , Sub-Step 2.3 determines  $S(c)$  and  $S_D(c)$ . They are summarized in the left and middle sides of Table 5. Next, Sub-Step 2.4 first computes  $F(c)$ , for  $c \in \text{CP}_2$ . The calculations are summarized in the right side of the table. Sub-Step 2.4 then compares  $F(c)$ , for  $c \in \text{CP}_2$ , and identifies the local optimal set of candidate points  $\text{CP}_2^*$  in  $\hat{T}_2$ , for  $0 \leq \alpha \leq 1$ , as follows:

$$\text{CP}_2^* = \arg \max \{F(c_{8,9}^1), F(c_{8,9}^2), F(c_{8,9}^3), F(c_{8,9}^4),$$

$$F(c_{8,9}^5), F(c_{10,11}^1), F(c_{10,11}^2), F(c_{10,11}^3), F(c_{10,11}^4)\}$$



$$= \arg \max \{140, 140, 140, 220\alpha, 140\alpha, 80, 80, 80, 220\alpha\} = \{c_{8,9}^1, c_{8,9}^2, c_{8,9}^3, c_{8,9}^4, c_{10,11}^4\}. \quad (19)$$

This means that if  $\alpha \geq 0.64$ , then  $c_{8,9}^4$  and  $c_{10,11}^4$  are local optimal candidate points in  $\hat{T}_2$ ; otherwise,  $c_{8,9}^1, c_{8,9}^2$ , and  $c_{8,9}^3$  become local optimal to  $\hat{T}_2$ .

Finally, by comparing  $F(x_1^*)$ , for  $x_1^* \in CP_1^*$ , and  $F(x_2^*)$ , for  $x_2^* \in CP_2^*$ , Step 3 derives the global optimal set of candidate points  $CP^*$  for the original tree  $T$  that covers the maximum traffic flow in round trips per time unit ( $F^*$ ):

$$\begin{aligned} CP^* &= \arg \max \{F(x_1^*), F(x_2^*) \mid x_1^* \in CP_1^*, x_2^* \in CP_2^*\} \\ &\in CP_2^* = \arg \max \{F(c_{1,2}^2), F(c_{1,3}^1), F(c_{2,3}^5), \\ &F(c_{4,7}^1), F(c_{4,7}^2), F(c_{8,9}^1), F(c_{8,9}^2), F(c_{8,9}^3), F(c_{8,9}^4), \\ &F(c_{10,11}^4)\} = \arg \max \{120 + 70\alpha, 120 + 70\alpha, 120 \\ &+ 70\alpha, 155, 155, 140, 140, 140, 220\alpha, 220\alpha\} \\ &= \{c_{1,2}^2, c_{1,3}^1, c_{2,3}^5, c_{4,7}^1, c_{4,7}^2, c_{8,9}^4, c_{10,11}^4\}. \end{aligned} \quad (20)$$

The set of optimal candidate points  $CP^*$  changes as the deviation portion  $\alpha$  varies between 0 and 1. When  $0 \leq \alpha < 0.5$ , the two candidate points in  $CP^* = \{c_{4,7}^1, c_{4,7}^2\}$  are optimal locations in tree  $T$  covering a traffic flow of 155 round trips per time unit. Since  $c_{4,7}^1$  and  $c_{4,7}^2$  are consecutive optimal candidate points satisfying the conditions of Theorem 8, all interior points between  $c_{4,7}^1$  and  $c_{4,7}^2$  are also optimal, and therefore, the complete set of optimal solutions  $CS^* = \ell(c_{4,7}^1, c_{4,7}^2)$ . If  $0.5 \leq \alpha < 0.8$ , the three candidate points in  $CP^* = \{c_{1,2}^2, c_{1,3}^1, c_{2,3}^5\}$  are optimal locations with a traffic flow coverage of 155 to 176 round trips per time unit depending on  $\alpha$ . Since these three optimal candidate points are located in the same spot,  $CS^* = \{c_{1,2}^2, c_{1,3}^1, c_{2,3}^5\}$ . Lastly, if  $0.8 \leq \alpha \leq 1$ , the two candidate points in  $CP^* = \{c_{8,9}^4, c_{10,11}^4\}$  are optimal solutions with a traffic flow coverage of 176 to 220 round trips per time unit. Since  $c_{8,9}^4$  and  $c_{10,11}^4$  are located in the same spot,  $CS^* = \{c_{8,9}^4, c_{10,11}^4\}$ . Figure 4 illustrates the change of the optimal solutions in tree  $T$  as the value of  $\alpha$  changes from 0 to 1.

**7.2. Sensitivity Analysis for Coupled Effects of Deviation Portion and Vehicle Driving Range.** In this subsection, we analyze the coupled effects of deviation portion and vehicle driving range on the set of optimal locations and the maximum traffic flow covered in tree  $T$ . The actual deviation portion  $\alpha$ , which refers to the portion of drivers willing to deviate from their preplanned paths for refueling service per time unit, is difficult to obtain or likely to be inaccurately estimated. Thus, the values of  $\alpha$  used in this analysis range from 0 to 1. The vehicle driving range  $R$ , which depends on the fuel consumption, is expected to increase substantially with improving technologies [33]. To reflect about this point, the values of  $R$  used in this subsection range from  $R = 40$  to  $R = 120$ , which are  $\pm 50\%$  of the value of  $R = 80$  used in the previous subsection, in increments of 20.

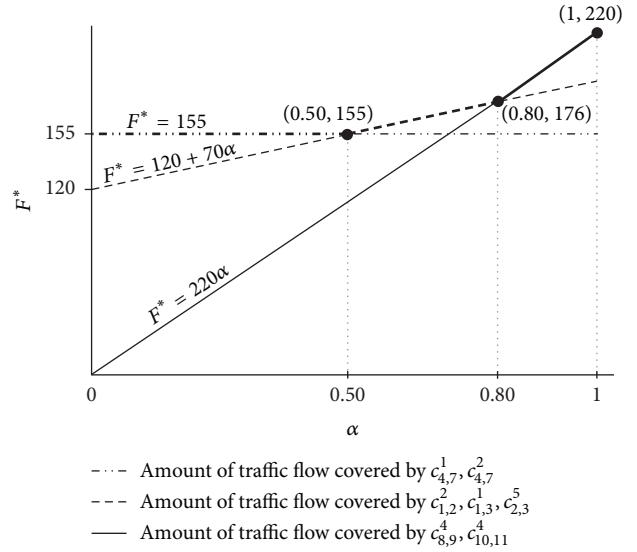


FIGURE 4: Change of the optimal solutions for tree  $T$  as deviation portion  $\alpha$  changes.

The first and last three columns of Table 6 show the complete set of optimal solutions  $CS^*$  and corresponding traffic flow covered  $F^*$  (in round trips per time unit) for five different vehicle driving ranges ( $R = 40, R = 60, R = 80, R = 100$ , and  $R = 120$ ) and deviation portions between 0 and 1 for the proposed solution approach. In addition, Figure 5 displays the trade-off between deviation portion  $\alpha$  and maximum traffic flow covered  $F^*$  for the five different vehicle driving ranges. From these results, we first notice that the maximum traffic flow covered  $F^*$  increases or at least stays the same as the deviation portion  $\alpha$  increases for a given vehicle driving range. For example, when the vehicle driving range is fixed to  $R = 80$ ,  $F^* = 155$  as  $\alpha$  increases from 0 to 0.5; then  $F^*$  increases from 155 to 220 round trips per time unit as  $\alpha$  increases from 0.5 to 1. Note also that the value of  $F^*$  increases or stays the same as the vehicle driving range  $R$  increases for a given value of  $\alpha$  between 0 and 1. For example, when the deviation portion is fixed to  $\alpha = 0.5$ ,  $F^* = 140$  for  $R = 40$ ;  $F^* = 155$  for  $R = 60$  and  $R = 80$ ;  $F^* = 190$  for  $R = 100$ ;  $F^* = 435$  for  $R = 120$ .

Figure 6 graphically presents the locations of the optimal candidate points listed in Table 6. It can be observed that when the maximum traffic flow covered  $F^*$  stays the same, the complete set of optimal solutions  $CS^*$  may expand as the deviation portion  $\alpha$  increases and the vehicle driving range  $R$  remains constant. For example, when  $R = 40$  and  $\alpha$  increases from 0 to 1, the maximum traffic flow covered  $F^*$  stays at 140 round trips per time unit, but  $CS^*$  is expanded from  $\ell(c_1, c_2)$ , for  $0 \leq \alpha < 1$ , to  $\ell(c_1, c_2) \cup \ell(c_1, c_3) \cup \ell(c_1, c_4)$ , for  $\alpha = 1$ . Similarly, while  $F^*$  remains constant,  $CS^*$  may enlarge as  $R$  increases and  $\alpha$  stays the same. For example, given that  $0 \leq \alpha < 0.5$ , as  $R$  increases from 60 to 80,  $F^*$  stays at 155 round trips per time unit, but  $CS^*$  is expanded from one candidate point  $\{c_5\}$  to line segment  $\ell(c_6, c_7)$  including  $\{c_5\}$ .

TABLE 6: Optimal locations and the corresponding traffic flows covered for each vehicle driving range (and deviation portion combination) for three models.

Vehicle driving range ( $R$ )	DFRLM [20]		Ventura et al. [18] algorithm		Proposed solution approach		
	Complete set of optimal solutions (vertices)	Traffic flow covered ( $F^*$ )	Complete set of optimal solutions ( $CS^*$ )	Traffic flow covered ( $F^*$ )	Deviation portion ( $\alpha$ )	Complete set of optimal solutions ( $CS^*$ )	Traffic flow covered ( $F^*$ )
40	$\{v_8, v_9\}$	140	$\ell(c_1, c_2)$	140	$0.00 \leq \alpha < 1.00$	$\ell(c_1, c_2)$	140
					$\alpha = 1$	$\ell(c_1, c_2) \cup \ell(c_1, c_3) \cup \ell(c_1, c_4)$	140
60	$\{v_8, v_9\}$	140	$\{c_5\}$	155	$0.00 \leq \alpha \leq 1.00$	$\{c_5\}$	155
80	$\{v_8, v_9\}$	140	$\ell(c_6, c_7)$	155	$0.00 \leq \alpha < 0.50$	$\ell(c_6, c_7)$	155
					$0.50 \leq \alpha < 0.80$	$\{c_8\}$	$120 + 70\alpha$
					$0.80 \leq \alpha \leq 1.00$	$\{c_{13}\}$	$220\alpha$
100	$\{v_2\}$	190	$\{c_{11}\}$	190	$0.00 \leq \alpha < 0.57$	$\{c_{11}\}$	190
					$0.57 \leq \alpha < 1.00$	$\{c_{12}\}$	$150 + 70\alpha$
					$\alpha = 1$	$\{c_{12}\} \cup \ell(c_9, c_{10})$	220
120	$\{v_4\}$	435	$\{c_{14}\}$	435	$0.00 \leq \alpha \leq 1.00$	$\{c_{14}\}$	435

Note. The locations of candidate points in the complete set of optimal solutions ( $CS^*$ ) are shown in Figure 6.

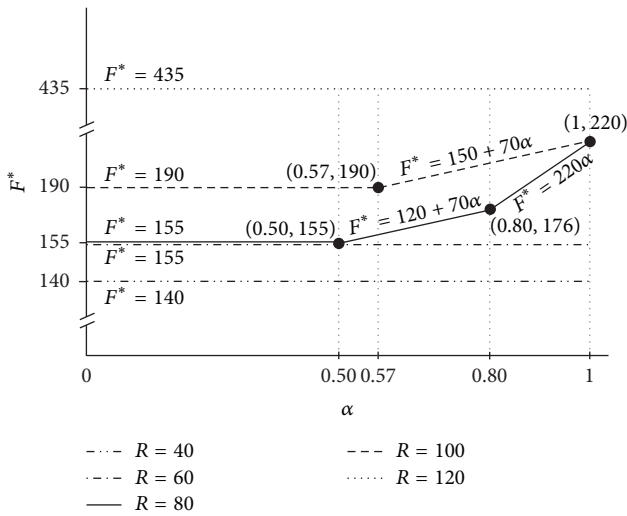


FIGURE 5: Trade-off between deviation portion and maximum traffic flow covered for the five different vehicle driving ranges ( $R = 40$ ,  $R = 60$ ,  $R = 80$ ,  $R = 100$ , and  $R = 120$ ).

7.3. Performance Analysis. In this subsection, the optimal solution of the proposed procedure is compared with those of the DFRLM [20] and the Ventura et al. [18] algorithm using the example in this section. The three approaches consider the same objective function, that is, maximizing the total traffic flow covered by the station, but have different assumptions. The main distinctions among these three models can be summarized briefly. First, the DFRLM, like most models published in the literature, assumes that the set of candidate sites is the set of vertices of the network. Therefore, the DFRLM is able to find deviation paths only if such vertices can be reached within the remaining driving

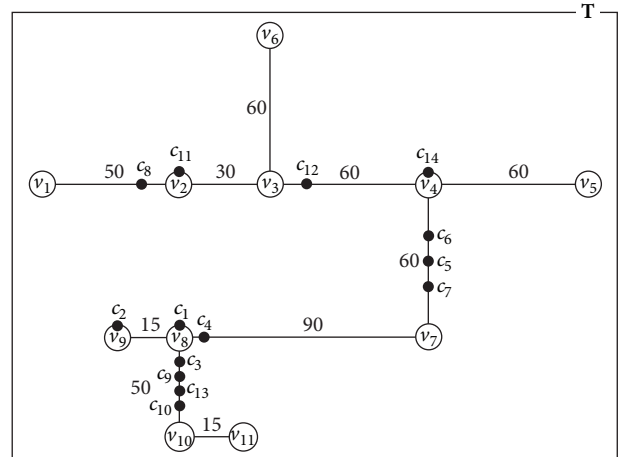


FIGURE 6: Locations of the optimal candidate points listed in Table 6.

range of vehicles. On the other hand, Ventura et al. [18] consider that the station can be located anywhere along the network, but vehicle deviations are not allowed. Lastly, the proposed approach considers the continuous version of the problem and allows vehicle deviations. Table 2 provides a detailed comparison of these three methodologies, including their computational complexity. In spite of the different computational requirements, since the size of the network in this example is small, the three solution approaches solve the example problem very quickly.

The first column of Table 6 shows the five values of  $R$  that are considered. The optimal locations and corresponding traffic flow covered  $F^*$  (in round trips per time unit) for the DFRLM are shown in columns 2 and 3 and for Ventura et al. [18] algorithm are provided in columns 4 and 5. The last

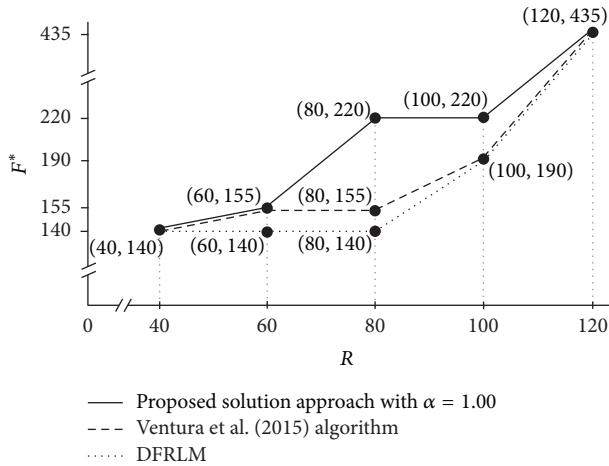


FIGURE 7: Coupled effects of vehicle driving range and deviation portion on the maximum traffic flow covered for the three models (DFRLM, Ventura et al. algorithm, and the proposed solution approach).

three columns include the results of the proposed solution approach as explained in Section 7.2. Figure 7 displays the coupled effects of vehicle driving range and deviation portion on the maximum traffic flow  $F^*$  for the three models: DFRLM, Ventura et al. algorithm, and proposed approach.

The comparison among the three studies in Table 6 and Figure 7 verifies that the proposed solution approach performs better than the other two studies in terms of maximum traffic flow covered and set of optimal locations. First, Figure 7 shows that the maximum traffic flow covered  $F^*$  of the proposed approach is always higher than or equal to those of the other two models for all values of  $R$ . For example, when  $R = 80$ , the proposed approach with  $\alpha = 1.00$  finds the optimal locations covering 220 round trips (per time unit), which is 57% and 42% higher than the round trips covered at the suboptimal locations found using the DFRLM and Ventura et al. algorithm, respectively. Next, Table 6 also demonstrates that when all three models find the same value of  $F^*$ , the proposed approach may be able to detect more optimal locations than the other two models. For example, when  $R = 40$ ,  $F^* = 140$  for all three models, but the set of optimal locations found by the proposed approach, which is  $CS^* = \ell(c_1, c_2) \cup \ell(c_1, c_3) \cup \ell(c_1, c_4)$ , includes the two optimal locations,  $\{v_8, v_9\}$ , found by the FRLM and the set of optimal locations,  $\ell(c_1, c_2)$ , found by the Ventura et al. algorithm.

An interesting point is that the DFRLM fails to find any deviation paths in this example, while the proposed approach finds six deviation paths from simple paths  $P(v_2, v_3)$ ,  $P(v_8, v_9)$ , and  $P(v_{10}, v_{11})$ , as shown in Figure 3. In this example, no vertices from these simple paths are reachable within the remaining driving range of vehicles; thus, the DFRLM cannot find any feasible deviation path. On the other hand, the proposed approach explores every point along detour edges in polynomial time, so it finds all possible deviation paths in the network regardless of the location of vertices on detours.

## 8. Conclusions and Future Research

This article has studied the continuous version of an AF refueling station location problem on a tree network considering that a portion of drivers have the option to deviate from their preplanned paths to be able to refuel their vehicles. The objective of the problem is to identify the set of optimal station locations that maximize the total traffic flow covered (in round trips per time unit). To achieve this goal, we have first generated a set of subtrees from the original tree network to reduce the problem size and consider the deviation option. Next, we have identified three sets of candidate points, including endpoints of refueling segments, cycle starting vertices, and cycle returning points, to locate the refueling station. We have proved that there exists at least one optimal solution in these sets of candidate points. Then, we have developed a polynomial time algorithm to determine the set of optimal candidate points that maximizes the total traffic flow covered. Lastly, we have derived conditions under which the line segments defined by certain pairs of consecutive optimal candidate points are also optimal. Our performance analysis demonstrates that the proposed solution approach can significantly improve the suboptimality of the solutions found by the existing refueling station location models. Note that although in this study we have used a small-size numerical example to illustrate and verify the proposed solution approach step by step, this procedure can also be applied for the large-scale problems due to its polynomial time complexity.

For future research we have identified three clear extensions to our solution approach. First, some of the assumptions we made in this article can be relaxed. For example, some vehicles may have different fuel tank levels at their origin and destination points, and fuel consumption may be different for some vehicles, which implies that they can have different driving ranges. We also suggest to extend the approach discussed in this article for tree networks to general networks having multiple paths between pairs of vertices. In addition, a multiobjective version of the refueling station location problem can also be considered to identify efficient locations that both maximize the total traffic flow covered by the stations and minimize construction and maintenance costs of the stations, assuming that these costs differ regionally.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

## References

- [1] M. Chambers, J. Goworowska, C. Rick, and J. Sedor, *Freight Facts and Figures*, Bureau of Transportation Statistics, U.S. Department of Transportation, 2015.
- [2] M. Wisniewski, Haunted by Past Oil Spikes, Commercial and Public Fleets Try Alternatives, *Chicago Tribune*, <http://www.chicagotribune.com/news/columnists/>, 2016.
- [3] Federal Highway Administration, Toll Facilities in the United States. U.S. Department of Transportation, <https://www.fhwa.dot.gov/policyinformation/tollpage/>, 2016.

- [4] Federal Highway Administration, National Highway System, U.S. Department of Transportation, [http://www.fhwa.dot.gov/planning/national\\_highway\\_system/](http://www.fhwa.dot.gov/planning/national_highway_system/), 2016.
- [5] R. L. Francis, L. F. McGinnis, and J. A. White, *Facility Layout and Location: An Analytical Approach*. Prentice-Hall, Prentice-Hall, Englewood Cliffs, NJ, USA, 1992.
- [6] D. B. West, *Introduction to Graph Theory*, Pearson Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2001.
- [7] A. W. Chan and R. L. Francis, "Round-trip location problem on a tree graph," *Transportation Science*, vol. 10, no. 1, pp. 35–51, 1976.
- [8] E. Erkut, R. L. Francis, and T. J. Lowe, "Locating two facilities on a tree subject to distance constraints," *Transportation Science*, vol. 22, no. 3, pp. 199–208, 1988.
- [9] B. C. Tansel, R. L. Francis, and T. J. Lowe, "Biobjective multi-facility minimax location problem on a tree network," *Transportation Science*, vol. 16, no. 4, pp. 407–429, 1982.
- [10] J. A. Ventura, S. J. Kweon, S. W. Hwang, M. Tormay, and C. Li, "Energy policy considerations in the design of an alternative-fuel refueling infrastructure to reduce GHG emissions on a transportation network," *Energy Policy*, vol. 111, pp. 427–439, 2017.
- [11] Pennsylvania Turnpike Commission, PA Turnpike, Sunoco Inc. Open Three CNG Pumps at New Stanton Plaza, Pennsylvania Turnpike Media and Public Relations, <https://www.paturnpike.com/press/2014/20141113115418.htm>, 2014.
- [12] J. Beaty, New McAlester Travel Plaza Opens on Turnpike. McAlester News-Capital. [http://www.mcalesternews.com/news/new-mcalester-travel-plaza-opens-on-turnpike/article\\_a4e025a2-87cc-11e4-bc94-ef82de09fb38.html](http://www.mcalesternews.com/news/new-mcalester-travel-plaza-opens-on-turnpike/article_a4e025a2-87cc-11e4-bc94-ef82de09fb38.html), 2014.
- [13] J. F. Marks, Upgrades Coming for Oklahoma Turnpike CNG Station, The Oklahoman, <http://newsok.com/article/5362464>, 2014.
- [14] United Press International, Natural Gas Sold for Turnpike Cars, United Press International Archives, <http://www.upi.com/Archives/1991/05/07/Natural-gas-sold-for-turnpike-cars/2878673588800/>, 1991.
- [15] G. Y. Handler, "Minimax location of a facility in an undirected tree graph," *Transportation Science*, vol. 7, no. 3, pp. 287–293, 1973.
- [16] O. Kariv and S. L. Hakimi, "An Algorithmic Approach to Network Location Problems. II: The p-medians," *SIAM Journal on Applied Mathematics*, vol. 37, no. 3, pp. 539–560, 1979.
- [17] P. J. Slater, "On locating a facility to service areas within a network," *Operations Research*, vol. 29, no. 3, pp. 523–531, 1981.
- [18] J. A. Ventura, S. W. Hwang, and S. J. Kweon, "A continuous network location problem for a single refueling station on a tree," *Computers & Operations Research*, vol. 62, pp. 257–265, 2015.
- [19] M. Kuby, S. Lim, and C. Upchurch, "Dispersion of nodes added to a network," *Geographical Analysis*, vol. 37, no. 4, pp. 383–409, 2005.
- [20] J.-G. Kim and M. Kuby, "The deviation-flow refueling location model for optimizing a network of refueling stations," *International Journal of Hydrogen Energy*, vol. 37, no. 6, pp. 5406–5420, 2012.
- [21] M. J. Hodgson, "A flow capturing location-allocation model," *Geographical Analysis*, vol. 22, no. 3, pp. 270–279, 1990.
- [22] O. Berman, R. C. Larson, and N. Fouska, "Optimal location of discretionary service facilities," *Transportation Science*, vol. 26, no. 3, pp. 201–211, 1992.
- [23] M. Kuby and S. Lim, "The flow-refueling location problem for alternative-fuel vehicles," *Socio-Economic Planning Sciences*, vol. 39, no. 2, pp. 125–145, 2005.
- [24] J.-G. Kim and M. Kuby, "A network transformation heuristic approach for the deviation flow refueling location model," *Computers & Operations Research*, vol. 40, no. 4, pp. 1122–1131, 2013.
- [25] S. H. Chung and C. Kwon, "Multi-period planning for electric car charging station locations: A case of Korean expressways," *European Journal of Operational Research*, vol. 242, no. 2, pp. 677–687, 2015.
- [26] M. Hosseini and S. A. MirHassani, "Refueling-station location problem under uncertainty," *Transportation Research Part E: Logistics and Transportation Review*, vol. 84, pp. 101–116, 2015.
- [27] Y. Huang, S. Li, and Z. S. Qian, "Optimal deployment of alternative fueling stations on transportation networks considering deviation paths," *Networks and Spatial Economics*, vol. 15, no. 1, pp. 183–204, 2015.
- [28] S. W. Hwang, S. J. Kweon, and J. A. Ventura, "Infrastructure development for alternative fuel vehicles on a highway road system," *Transportation Research Part E: Logistics and Transportation Review*, vol. 77, pp. 170–183, 2015.
- [29] S. W. Hwang, S. J. Kweon, and J. A. Ventura, "Locating alternative-fuel refueling stations on a multi-class vehicle transportation network," *European Journal of Operational Research*, vol. 261, no. 3, pp. 941–957, 2017.
- [30] M. Miralinaghi, B. B. Keskin, Y. Lou, and A. M. Roshandeh, "Capacitated refueling station location problem with traffic deviations over multiple time periods," *Networks and Spatial Economics*, vol. 17, no. 1, pp. 129–151, 2017.
- [31] M. Miralinaghi, Y. Lou, B. B. Keskin, A. Zarrinmehr, and R. Shabanpour, "Refueling station location problem with traffic deviation considering route choice and demand uncertainty," *International Journal of Hydrogen Energy*, vol. 42, no. 5, pp. 3335–3351, 2017.
- [32] S. J. Kweon, S. W. Hwang, and J. A. Ventura, "Intersection of a Continuous 1-Location Problem on a Tree Network," in *Proceedings of IIE Annual Conference*, pp. 1399–1408, 2015.
- [33] J. Park, Natural Gas Technology Improvements Changing Attitudes, Truckinginfo.com, <http://www.truckinginfo.com/article/story/2014/09/natural-gas.aspx>, 2014.





**Hindawi**

Submit your manuscripts at  
<https://www.hindawi.com>

