An Efficient Approach for Service System Design with Immobile Servers, Stochastic Demand, Congestion, and Consumer Choice

ABSTRACT

In this paper, we develop models and solution methodologies for the service system design problem. Service system design problems aim to determine the optimal number, location, and capacity level of service facilities as well as the assignment of consumers to facilities to optimize some service objectives. In this paper, we consider the problem of designing service systems to find a socially optimal solution by minimizing the overall cost to consumers and service providers with applications in healthcare and public sector. We consider settings where consumers and facilities are part of a congested network in which the consumer demand and service times are stochastic, and the capacity cost is a concave function of capacity levels. Furthermore, we only consider assignments in which consumers choose which facility to patronize. The problem of designing a service system in our setting can be modeled by a non-linear mixed-integer programming problem, for which an exact solution within a reasonable time is not readily available. We use Generalized Benders Decomposition, and a special-purpose Search and Cut method to develop two efficient solution methodologies. Through a realistic case study of designing a service system in the city of Toronto, we provide further insight into the effect of various model parameters on the efficiency of the proposed solution methodologies.

Keywords: Location Models; Service System Design; Mixed-Integer Nonlinear Programming; Stochastic Demand; Congestion

I. INTRODUCTION AND RELATED WORK

In many applications, service time provided by service facilities is the primary factor in determining the service quality, and excessive waiting time negatively impacts the consumer experience. On the one hand, lower waiting times may necessitate the allocation of many decentralized facilities that are easily accessible, as measured by proximity to the majority of the consumers. On the other hand, few centralized facilities may be necessary to reduce service cost. *Service System Design* problems are often formulated to strike a balance between the consumer service time and the service cost with applications in many areas such as healthcare and public sector. Service system design problems fall into a more general category of *Stochastic Location Models with Congestion* (SLMC), in which demand is stochastically generated by the consumers and service times are stochastic as well. The nature of this stochastic behavior often implies that the demand may not be served immediately. The partial fulfillment of the consumer demand results in either waiting or, ultimately, demand loss. SLMCs capture the trade-off between consumer waiting time and service cost by finding an optimal assignment of consumers and allocation of service capacities to the open facilities. Servers in a SLMCs can be considered mobile or immobile. When servers are mobile, the service provider travels to provide service to consumers. When the servers are immobile, consumers travel to facilities to access the service.

In this paper, we consider the problem of designing a service system in a network of facilities with immobile servers. For a recent literature review that includes a detailed summary of location analysis with immobile servers and congestion, see Berman, Krass, and Wang (2011), Berman and Krass

(2020), and the references therein. Our model assumptions fit in a variety of applications, ranging from locating private facilities such as retail stores, repair shops, or service centers to locating public facilities such as government offices, hospitals, and medical clinics, where the consumers choose which facility to patronize, and capacity in these facilities can be an aggregation of a variety of service resources available at the facilities. Another important application example is the location of preventive health care facilities such as clinics with mammograms, vaccination sites, and blood test centers. For more applications of service system design problems in healthcare services, we refer the reader to the works of Aboolian, Berman, and Verter (2016); Zhang et al. (2010); Vidyarthi and Kuzgunkaya (2015); and Dogan, Karatas, and Yakici (2020).

The service system design problems can be classified using the following three characteristics: (1) whether the consumers choose the facility, or it is dictated to them by the provider; (2) whether the objective is to minimize consumer access cost or service provider's operating cost (or both); and (3) whether the cost of acquiring service capacity is linear or concave in capacity level. Marianov and Serra (1998) and Marianov and Ríos (2000) were first to consider congestion in service system design problems. They used the covering location model with a constraint that would not allow the waiting time or the queue length to exceed an acceptable level. These two papers, along with Wang, Batta, and Rump (2002); Berman and Drezner (2007); and Aboolian, Berman, and Drezner (2009) focus on minimizing service access cost to consumers assuming a limited-service capacity requirement. Wang, Batta, and Rump (2002) also consider the case in which the objective is to minimize the service provider's cost, assuming they provide a certain level of service quality. There are other papers in which the system is designed with the provider benefits in mind. One example of such problems is Aboolian, Berman, and Krass (2012), which introduces a profit-maximizing service system design problem with demand elasticity with respect to travel and waiting time. Other similar papers with

demand elasticity are Zhang, Berman, and Verter (2009) and Aboolian, Berman, and Verter (2016), where the objective is to maximize consumer participation.

Designing *socially optimal* service systems has been the focus of the attention of many researchers for the past two decades. In socially optimal service system designs, the objective is to minimize the overall cost to consumers (cost incurred for traveling, waiting, and service time) and service provider's operating costs (fixed facility and variable capacity costs). Amiri (1997); Aboolian, Berman, and Drezner (2008); Elhedhli (2006); Castillo, Ingolfsson, and Sim (2009); Vidyarthi and Jayaswal (2014), and Elhedhli, Wang, and Saif (2018) are examples in which a socially optimal service system design problem is considered. With the exception of Aboolian, Berman, and Drezner (2008), all papers in socially optimal models consider a consumer assignment by the service provider. In this paper, we consider service system design problems with *consumer choice*. To model consumer choice in stochastic location models, many papers such as Wang, Batta, and Rump (2002); Berman and Drezner (2007), and Aboolian, Berman, and Drezner (2008) consider settings in which consumers choose the closest facility to access their services. Similarly, we consider proximity to service facilities as the main proxy for consumer choice.

Elhedhli (2006) and Aboolian, Berman, and Drezner (2008) are perhaps the closest works to this paper. Although Elhedhli (2006) considers a service system design problem in which the service capacity cost is a concave function of service capacity, it differs from this paper in that it assumes consumers are assigned to facilities by the service provider. On the other hand, although Aboolian, Berman, and Drezner (2008) consider a service system design problem with consumer choice, it differs from this paper since it assumes service capacity cost is a linear function of the service capacity and facilities are modeled as M/M/k queuing systems. In this paper, we consider the consumer choice problem, use a concave service capacity cost function, and model facilities as M/M/1

queuing systems. In terms of service capacity, we assume that the service system designer's objective is to determine a service rate rather than determining the number of servers for a fixed service rate. In other words, we assume that each facility act as an M/M/1 queuing system rather than an M/M/k queuing system. There are two reasons for choosing the M/M/1 model over the M/M/k. First, facilities could use several distinct capacities and servers, which may be hard to determine. For example, a medical clinic will often use nurses, doctors, operating rooms, X-ray machines, etc., all with different levels of capacity and a different number of servers. For such a system, it is more suitable to use an aggregate capacity service rate that represents the clinic's different service resources. Second, when the system utilization is reasonably high, an M/M/1 queue could be used as a good approximation for an M/M/k queue (Baron, Berman, and Krass, 2008).

The problem is formulated as a Mixed-Integer Nonlinear Program (MINLP). We propose three solution methodologies. The first methodology is a heuristic, which uses a modified descent approach in neighborhood search for a known location set. The second solution methodology is another heuristic, which is based on a Generalized Benders Decomposition (Floudas, Aggarwal, and Ciric 1989; Geoffrion 1972) of the main problem into various Linear Programming (LP) subproblems and Mixed-Integer Programming (MIP) master problems. Finally, we use a special-purpose algorithm that utilizes a *Search and Cut* approach – an adaptation of the search and cut method introduced in Aboolian, Cui, and Shen (2013) to the service system design problem – for finding an optimal solution efficiently. To study the performance of the proposed methodologies, we perform extensive numerical testing on a realistic situation of designing service systems for the city of Toronto, Canada.

The contribution of this paper to the literature of service system design problems is two-fold. First, we model the service system design problem by considering consumer choice and a concave service capacity cost function where each facility is modeled as a M/M/1 queuing system, that when

compared to a M/M/k queuing systems, can represent a broader range of service capacities, and as such can be applied to a broader set of problems. Second, we develop an efficient exact solution methodology. In particular, unlike other known solution methodologies developed for the service system design problem, the Search and Cut solution approach in this paper can solve many large-scale problems efficiently while not being sensitive to the number of capacity levels.

The rest of this paper is organized as follows: Section 2 discusses the problem formulation. In Section 3, we present two heuristic methodologies based on a neighborhood search and a generalized Benders decomposition approach, and an exact solution approach based on a search and cut methodology. Numerical testing and computational results are discussed in Section 4. We provide conclusions and suggestions for future research in Section 5.

II. PROBLEM FORMULATION

Consider a set of consumers indexed by $i \in M = 1,2,\cdots,m$, in which consumer i's demand can be modeled as a Poisson process with a mean rate of λ_i. Moreover, consider the set of candidate facility locations indexed by $j \in N = 1,2,\cdots,n$, and the set of capacity levels indexed by $k \in K = 1,2,\cdots,\kappa$, in which the mean service rate of a facility when allocated capacity level k is denoted by $\bar{\mu}_k$. Without loss of generality, we assume that the vector $(\bar{\mu}_k)_{k\in K}$ is an increasing vector in k. Let y_{ij} be the binary variable that takes value 1 if consumer i's demand is allocated to service facility j. Furthermore, let z_{jk} be the binary variable that takes value 1 if facility j is allocated capacity level k. Also, let x_j be the binary variable that takes value 1 if we plan to open a facility in location j and 0 otherwise. We note that the problem can be formulated without the introduction of variables x_j, as $x_j = \sum_{k\in K} z_{jk}$. However, we use variables x_j to simplify the problem formulation. Consumer demands are assumed to be independent Poisson in which each consumer assigns the entirety of its demand to

the closest service facility. Moreover, we assume that the service times are exponentially distributed. Hence, each service facility acts as an M/M/1 queue with a mean demand of $\Lambda_j = \sum_{i \in M} \lambda_i \, y_{ij}$, and a mean service rate of $\mu_j = \sum_{k \in K} \bar{\mu}_k \, z_{jk}$. The service access cost for a unit demand of consumer i from facility j is denoted by c_{ij}, which we assume to be a linear function of the travel distance of consumer i to facility j denoted by d_{ij}. So the closer the facility is to the consumers, the lower their access cost. Denote α as the average waiting cost per unit of time and f_j to indicate the fixed cost of opening a facility at site j. Furthermore, the capacity cost of the facility j is modeled via a concave function $\omega(\mu_j) = (\beta \mu_j)^{\phi}$, in which $0 \leq \phi < 1$ and $\beta > 0$. Given the above definitions, the Service System Design Problem (SSDP) can be formulated as the following cost minimization problem.

$$\min \quad Z_{\text{SSDP}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_{j \in N} f_j \, x_j + \sum_{i \in M} \sum_{j \in N} c_{ij} \, \lambda_i y_{ij} + \sum_{j \in N} \sum_{k \in K} \omega \, (\bar{\mu}_k) z_{jk} + \alpha \sum_{j \in N} \frac{\sum_{i \in M} \lambda_i \, y_{ij}}{\sum_{k \in K} \bar{\mu}_k \, z_{jk} - \sum_{i \in M} \lambda_i \, y_{ij}}, \qquad (1.1)$$

subject to:

$$y_{ij} \leq x_j \,, \ \forall i \in M \,, \forall j \in N \,, \qquad (1.2)$$

$$\sum_{j \in N} y_{ij} = 1 \,, \ \forall i \in M \,, \qquad (1.3)$$

$$\sum_{k \in K} z_{jk} = x_j \,, \ \forall j \in N \,, \qquad (1.4)$$

$$\sum_{i \in M} \lambda_i \, y_{ij} \leq \sum_{k \in K} \bar{\mu}_k \, z_{jk} \,, \ \forall j \in N \,, \qquad (1.5)$$

$$\sum_{j' \in N} c_{ij'} \, y_{ij'} \leq (c_{ij} - L) x_j + L \,, \ \forall i \in M \ \forall j \in N \,, \qquad (1.6)$$

$$y_{ij} \in 0{,}1 \,, \ \forall i \in M \,, \forall j \in N \,, \qquad (1.7)$$

$$z_{jk} \in 0{,}1 \,, \ \forall j \in N \,, \forall k \in K \,, \qquad (1.8)$$

$$x_j \in 0{,}1 \,, \ \forall j \in N \,, \qquad (1.9)$$

The objective function terms represent the acquisition costs, access cost, capacity cost, and the expected waiting cost, respectively. Constraints 1.2 allow the assignment of demand to open facilities

only. Constraints 1.3 guarantee that each consumer is assigned to a single open facility. Constraints 1.4 guarantee that each open facility is assigned just one capacity level. Constraints 1.5 ensure that the demand arrival rate to a facility does not exceed its service capacity rate. Constraints 1.6 ensure that each consumer chooses (is assigned to) the closest facility, in which L is a large enough number (e.g., L=$\max_{ij} c_{ij}$ for $i \in M$ and $j \in N$). Note that Constraints 1.6 achieve the closest assignment restriction because if a facility is opened at $j(x_j = 1)$, the assignment access cost of customers at node $i \in M$, given by $\sum_{j' \in N} c_{ij'} y_{ij'}$, is at most c_{ij}.

III. SOLUTION METHODOLOGIES

SSDP is a MINLP for which finding an exact solution within a reasonable time is not easily available. In this section, we introduce a lower bound and several solution methodologies for SSDP. All of these solution methodologies are based on exploitation of SSDP's special structure. In particular, we demonstrate that solving SSDP can be simplified to the problem of finding a set of open facilities that minimizes the system's overall cost. To achieve this, we show that when a set of open facilities is determined, the optimal capacity allocations and the consumer assignments can be found efficiently. As such, we start this section by formalizing the idea of finding the optimal capacity allocation for a set of open facilities, and proceed to the introduction of a lower bound as well as approximate and exact solution methodologies for SSDP afterward.

3.1. Optimal Capacity Allocation for a Set of Open Facilities

Given a set of open facilities in SSDP, since consumers are to be assigned to their closest facility, the problem simplifies to finding the optimal capacity at each of the open facilities. To show this, let $S \subseteq N$ be a given set of open facilities in SSDP. Also, let $E_j(S)$ denote the set of consumers closest to

facility $j \in S$, in which $\bigcup_{j\in S} E_j\,(S) = M$, and $E_j(S) \cap E_{j'}(S) = \emptyset$, for $j \neq j' \in S$. Note that if consumer i's closest facility in set S is not unique, we consider the smallest indexed facility. Additionally, let $\Lambda_j(S)$ be the demand rate at facility $j \in S$, such that $\Lambda_j(S) = \sum_{i\in E_j(S)} \lambda_i$. Moreover, let $\bar{k}_j$ be the smallest capacity level for each facility determined as follows.

$$\bar{k}_j \;=\; \arg\min_{k\,\in\,K} \{k : \Lambda_j(S) \;\leq\; \bar{\mu}_k\} \text{ for } j \;\in\; S. \qquad (2)$$

Consider $\mu_j(S) \in \{\bar{\mu}_k : k \in \{\bar{k}_j, \bar{k}_j + 1, \cdots, \kappa\}\}$, a scalar denoting a feasible capacity rate assigned to the facility $j \in S$. Then the objective function of SSDP can be rewritten as follows:

$$\min_{(\mu_j(S))_{j\in S}} F\big(S, \mu(S)\big) \triangleq \sum_{j\in S} f_j + \sum_{j\in S} \sum_{i\in E_j(S)} c_{ij}\,\lambda_i + \sum_{j\in S} \omega\left(\mu_j(S)\right) + \alpha \sum_{j\in S} \frac{\Lambda_j(S)}{\mu_j(S) - \Lambda_j(S)}. \qquad (3)$$

Note that given S, since the first two terms of (3) are constant, the problem takes the following simpler objective:

$$\min_{(\mu_j(S))_{j\in S}} \bar{F}\big(S, \mu(S)\big) \triangleq \sum_{j\in S} \omega\left(\mu_j(S)\right) + \alpha \sum_{j\in S} \frac{\Lambda_j(S)}{\mu_j(S) - \Lambda_j(S)}. \qquad (4)$$

The optimal solution of (4), denoted by $\mu^\star(S) \triangleq \left(\mu_j^\star(S)\right)_{j\in S}$, can be found efficiently via complete enumeration of κ available capacity levels at each of the open facilities, even for large values of κ. This is because finding the capacity rate resulting in the least capacity and waiting time cost can be simplified to the problem of finding the minimum element of a vector, which has a polynomial complexity time.

Next, we formalize the idea of finding an optimal solution, given a set of open facilities, in Algorithm 1.

ALGORITHM 1: OPTIMAL CAPACITY ALLOCATION FOR A GIVEN SET OF OPEN FACILITIES

Input: S

Output: $\mu^{\star}(S), \bar{F}\left(S, \mu^{\star}(S)\right), F\left(S, \mu^{\star}(S)\right)$

1. $\forall j \in S$ let $E_j(S) = i : j = \text{argmin}_{j \in S} c_{ij}$, $i \in M$, and $\Lambda_j(S) = \sum_{i \in E_j(S)} \lambda_i$

2. For $j \in S$,

   I. Let $\bar{k}_j = arg\ min_{k \in K}\{k : \Lambda_j(S) \leq \bar{\mu}_k\}$

   II. $k_j^{\star} = \arg m\, in_{k \in \{\bar{k}_j, \bar{k}_j+1, \ldots, \kappa\}}\{k : \omega(\bar{\mu}_k) + \alpha \frac{\Lambda_j(S)}{\bar{\mu}_k - \Lambda_j(S)}\}$

   III. Set $\mu_j^{\star}(S) = \bar{\mu}_{k_j^{\star}}$

3. Set $\mu^{\star}(S) = \left(\mu_j^{\star}(S)\right)_{j \in S}$, $\bar{F}\left(S, \mu^{\star}(S)\right) = \sum_{j \in S} \omega\left(\mu_j^{\star}(S)\right) + \alpha \sum_{j \in S} \frac{\Lambda_j(S)}{\mu_j^{\star}(S) - \Lambda_j(S)}$,

   $F\left(S, \mu^{\star}(S)\right) = \sum_{j \in S} f_j + \sum_{j \in S} \sum_{i \in E_j(S)} c_{ij}\, \lambda_i + \bar{F}\left(S, \mu^{\star}(S)\right)$.

Given the above arguments, it becomes clear that the main decision in SSDP is simply to find the set of open facilities that minimizes the system's overall cost. We refer to such set as *optimal set of facility locations*.

3.2. A Lower Bound for SSDP

Let Z_{SSDP}^{*} and S_{SSDP}^{*} be the optimal value of the objective function and an optimal set of facility locations for SSDP, respectively. To obtain a lower bound on Z_{SSDP}^{*}, we start by finding a lower bound on the sum of capacity and waiting time cost, $\bar{F}\left(S, \mu(S)\right)$ as defined in (4). To do so, for a given number of open facilities, say $l \in N$, we solve the following non-linear optimization problem which we call Minimum Capacity and Waiting Cost Problem (MCWCP).

$$\min \quad \mathcal{C}_l(\mathbf{g}, \mathbf{m}) = \sum_{\eta=1}^{l} \omega\left(m_\eta\right) + \alpha \sum_{\eta=1}^{l} \frac{g_\eta}{m_\eta - g_\eta}, \qquad (5.1)$$

subject to:

$$g_\eta \le m_\eta\,, \forall \eta \in \{1,2,\cdots,l\}, \qquad (5.2)$$

$$m_\eta \le \bar{\mu}_K\,, \forall \eta \in \{1,2,\cdots,l\}, \qquad (5.3)$$

$$\sum_{\eta=1}^{l} g_\eta = \Lambda, \qquad (5.4)$$

$$g_\eta, m_\eta \ge 0\,, \forall \eta \in \{1,2,\cdots,l\}, \qquad (5.5)$$

in which $\Lambda = \sum_{i\in M} \lambda_i$, and $l \in N$. Also m_η and g_η are the decision variables corresponding to demand and capacity levels at facility $\eta \in \{1,2,\dots,l\}$, respectively. We note that the continuous variables of $\mathbf{g}$ and $\mathbf{m}$, resemble the discrete parameters of $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, respectively. For a given l, let $\mathcal{C}_l^\star \triangleq \mathcal{C}_l^\star(\mathbf{g}^\star, \mathbf{m}^\star) = \sum_{\eta=1}^{l} \omega\left(m_\eta^\star\right) + \alpha \sum_{\eta=1}^{l} \frac{g_\eta^\star}{m_\eta^\star - g_\eta^\star}$ be the optimal value of MCWCP. Note that, since MCWCP finds an optimal capacity and demand allocation to the open facilities without restricting the allocations of specific consumer nodes to specific facilities or requiring a discrete set of capacity rates, its value is a lower bound on the sum of capacity and waiting time cost.

For a given S, note that $\Lambda_j(S) = \sum_{i\in E_j(S)} \lambda_i$ for $j \in S$, $\mu^\star(S) = \left(\mu_j^\star(S)\right)_{j\in S}$, and $\bar{F}\left(S, \mu^\star(S)\right)$ can be found using Algorithm 1. Let r_η be the $\boldsymbol{\eta}$th smallest facility index in S such that $r_\eta \in S$ for $\eta \in 1,2,\dots,|S|$, in which $|S|$ is the cardinality of set S. Since $\left(m_\eta, g_\eta\right) = \left(\Lambda_{r_\eta}(S), \mu_{r_\eta}^\star(S)\right)$, for $\eta \in 1,2,\dots,|S|$, is a feasible solution of MCWCP, then for any S, $\mathcal{C}_{|S|}^\star \le \bar{F}\left(S, \mu^\star(S)\right)$. Therefore, we conclude:

$$\mathcal{C}_{|S_{\text{SSDP}}^*|}^\star \le \bar{F}\left(S_{\text{SSDP}}^*, \mu^\star(S_{\text{SSDP}}^*)\right). \qquad (6)$$

We now define the following MIP which is a modified version of the Uncapacitated Facility Location Problem (MUFLP):

$$\min \quad Z_{\text{MUFLP}}(\mathbf{x},\mathbf{y},\mathbf{v}) = \sum_{j\in N} f_j\, x_j + \sum_{i\in M}\sum_{j\in N} c_{ij}\,\lambda_{ij} y_{ij} + \sum_{l=1}^{n} C_l^{\star}\, v_l, \qquad (7.1)$$

subject to:

$$y_{ij} \le x_j\,,\ \forall i \in M\ \forall j \in N\,, \qquad (7.2)$$

$$\sum_{j\in N} y_{ij} = 1\,,\ \forall i \in M\,, \qquad (7.3)$$

$$\sum_{j'\in N} c_{ij'}\, y_{ij'} \le \big(c_{ij} - L\big)x_j + L\,,\ \forall i \in M\ ,\forall j \in N\,, \qquad (7.4)$$

$$\sum_{l=1}^{n} v_l = 1\,, \qquad (7.5)$$

$$\sum_{j\in N} x_j = \sum_{l=1}^{n} l \times v_l\,, \qquad (7.6)$$

$$y_{ij} \le x_j\,,\ \forall i \in M\,,\forall j \in N\,, \qquad (7.7)$$

$$y_{ij} \in \{0,1\},\ \forall i \in M\ ,\forall j \in N\,, \qquad (7.8)$$

$$x_j \in \{0,1\},\ \forall j \in N\,, \qquad (7.9)$$

$$v_l \in \{0,1\},\ \forall l \in \{1,2,\cdots,n\}, \qquad (7.10)$$

in which v_l is a binary variable that is one if we open l facilities and zero otherwise. Constraint 7.6 ensures that if $\sum_{j\in N} x_j = l$, then $v_l = 1$. Constraints 7.4 ensure consumers are assigned to the closest facility. Also note that Constraint 7.5 and Constraint 7.6 do not restrict the feasible region for a general Uncapacitated Facility Location Problem (UFLP). They simply decide the value for the last term in the objective function of MUFLP.

To see how the optimal solution of the MUFLP provides a lower bound for SSDP, consider the following result.

Lemma 1. Let $Z^{\star}_{\text{MUFLP}}$, and $S^{\star}_{\text{MUFLP}}$ be the optimal value of the MUFLP and the set of open facilities corresponding to the optimal solution of MUFLP, respectively. Then $Z^{\star}_{\text{MUFLP}} \le Z^{\star}_{\text{SSDP}}$.

A proof is included in Section 6.1 in the Appendix.

3.3. An Approximate Approach and An Upper Bound for SSDP

We note that when the unit costs for capacity and waiting time in SSDP equals zero ($\alpha = \beta = 0$), SSDP reduces to UFLP. Since UFLP is an NP-hard problem, SSDP is also an NP-hard problem. Thus, it is difficult to obtain good solutions for SSDP within a reasonable time. As such, developing a heuristic to find quality approximate solutions for SSDP becomes necessary. The heuristic presented here is based on the solution to MUFLP discussed in Section 3.2. Given Lemma 1, the optimal value of MUFLP is a lower bound to SSDP. We note that any feasible location set for MUFLP (including $S^{\star}_{\text{MUFLP}}$) is also a feasible solution for SSDP. This is true since in both problems the consumers are assigned to their closest facility. Therefore, the objective value of the SSDP for the optimal location set found in MUFLP provides an upper bound for SSDP such that

$$Z^{\star}_{\text{SSDP}} \leq \quad F\left(S^{\star}_{\text{MUFLP}}, \mu^{\star}(S^{\star}_{\text{MUFLP}})\right). \qquad (9)$$

We note that $F\left(S^{\star}_{\text{MUFLP}}, \mu^{\star}(S^{\star}_{\text{MUFLP}})\right)$ can be found using Algorithm 1.

Here, we use a *modified descent approach* in neighborhood search for $S^{\star}_{\text{MUFLP}}$ – the location set corresponding to the optimal solution of MUFLP – to find an approximate solution, and an improved upper bound, for SSDP. For each location set in the neighborhood we look for improvement in the objective value, which can be determined efficiently using Algorithm 1.

A modified descent approach can be applied as follows. We first define the distance-2 neighborhood of a set S. For example, S' is in neighborhood of S if the number of nonoverlapping elements in the two sets does not exceed 2. In particular, the distance-2 neighborhood of S includes S' with

- one additional facility,

- one facility removed from it (when $|S| > 1$),
- two additional facilities,
- two facilities removed from it (when $|S| > 2$), and
- one facility added and another facility removed from it.

Once the neighborhood is well defined, the modified descent approach is straightforward: Use a starting subset S; evaluate the change in the value of the objective function for all the subsets in the neighborhood, using Algorithm 1; if an improved subset exists in the neighborhood, we repeat the search for all the improved subsets in the neighborhood. The above steps are repeated using the new subsets until no improved subset exists in neighborhood searches. The set with the best objective is the approximate solution. We note that the descent algorithm and its modification can be expanded to arbitrary distance neighborhoods. See Aboolian, Cui, and Shen (2013) for an example.

The modified descent approach introduced here is an adaptation of the descent approach introduced in Aboolian, Cui, and Shen (2013). Whereas the descent approach considers the best location set when repeating the neighborhood search, the modified descent approach considers all improved location sets. Thus, when compared to the descent approach, the modified version here generally searches a wider range of location sets. As we discuss in Section 4, the wider neighborhood search of the modified descent approach results in improved upper and lower bounds, fewer iterations of the search and cut methodology, and ultimately better efficiency.

Algorithm 2 describes the modified descent approach to find the set of all improvements in the objective value, along with an approximate solution of SSDP given an initial set of open facilities.

ALGORITHM 2: MODIFIED DESCENT ALGORITHM FOR A GIVEN SET OF OPEN FACILITIES

Input: S

Output: $\mathcal{A}(S), S_H(S), F\big(S_H, \mu^\star(S_H)\big), \mathcal{G}(S)$

1. Find $F\big(S, \mu^\star(S)\big)$ using Algorithm 1 and set $b = F\big(S, \mu^\star(S)\big)$. Let $\mathcal{A}(S) \triangleq \mathcal{A} = S$, $\mathcal{G}(S) \triangleq \mathcal{G} = S$, $S_H(S) \triangleq S_H = S$, and $\mathcal{S} = S_H$.
2. Let S' be the first element of $\mathcal{S}$
   I) Let $\mathcal{B}$ be the set of all neighbors of S' with
      - one additional facility,
      - one facility removed from it (when $|S'| > 1$),
      - two additional facilities,
      - two facilities removed from it (when $|S'| > 2$), and
      - one facility added and another facility removed from it,

   where $|S'|$ is the cardinality of S'.
   II) Set $\mathcal{G} = \mathcal{G} \cup \mathcal{B}$:
   III) For $B \in \mathcal{B}$:
      a) Find $F\big(B, \mu^\star(B)\big)$ and $F\big(S', \mu^\star(S')\big)$ using Algorithm1.
      b) If $F\big(B, \mu^\star(B)\big) < F\big(S', \mu^\star(S')\big)$ then add B to the set of locations that we should search their neighborhood: $\mathcal{S} = \mathcal{S} \cup B$, and the set of cuts: $\mathcal{A} = \mathcal{A} \cup B$.
         b.1) If $F\big(B, \mu^\star(B)\big) < b$ then update the best objective: $b = F\big(B, \mu^\star(B)\big)$, and the location set with best objective: $S_H = B$.
   IV) Remove the set S' from the set of locations we should search their neighborhood: $\mathcal{S} = \mathcal{S} - S'$
3. If $\mathcal{S}$ is empty, STOP and return $\mathcal{A}(S)$, $S_H(S), F\big(S_H, \mu^\star(S_H)\big), \mathcal{G}(S)$. Else, go to **Step 2**.

If we use $S^\star_{\text{MUFLP}}$ as the starting set of open facilities in Algorithm 2, we will obtain a feasible solution of SSDP, hence a new and improved upper bound on the optimal value of SSDP, such that

$$Z^\star_{\text{SSDP}} \leq F\left(S^H(S^\star_{\text{MUFLP}}),)\mu^\star\big(S^H(S^\star_{\text{MUFLP}})\big)\right).$$

in which, $S^H(S^\star_{\text{MUFLP}})$ and $F\left(S^H(S^\star_{\text{MUFLP}}),)\mu^\star\left(S^H(S^\star_{\text{MUFLP}})\right)\right)$ are found using Algorithm 2.

3.4. Search and Cut Approach for SSDP

The exact approach presented here is based on successive improvements on upper bound and lower bounds of SSDP. We call this exact method *Search and Cut*, since the improvement on upper bound is based on increasing the *"Search"* area to include new neighborhoods, and the improvement on lower bound is achieved by solving the original MUFLP with added *"Cuts"* to exclude the area that has already been searched. In particular, we first find an initial lower and upper bound for SSDP by solving MUFLP and applying Algorithm 2 when $S = S^\star_{\text{MUFLP}}$. In the next step, we find an improved lower bound for $Z^\star_{\text{SSDP}}$ by solving the MUFLP with some added cuts that shrink the feasible region by removing all the subsets which have already been searched and evaluated in the neighborhood search in Algorithm 2. After solving the MUFLP with added cuts, we use the location set produced by the optimal solution as the starting set in a new neighborhood search to improve the upper bound using Algorithm 2. To complete this step, for every starting location set used in Algorithm 2 denoted by $\mathcal{A} \triangleq \mathcal{A}(S)$, we need to add a cut to exclude location sets that are in the neighborhood of the starting set. Consider the following condition.

$$\sum_{j\in B} x_j - \sum_{j\in B^c} x_j \leq |B| - 3 : \forall B \in \mathcal{A}, \qquad (10)$$

in which $B^c = N - B$, and $|B|$ is the cardinality of B. Let $\bar{x} = \left(\bar{x}_j\right)_{j\in N}$ be an assignment which determines a set of open facilities $\bar{S}$, that is $\bar{S} = j: \bar{x}_j = 1, j \in N$. Thus, as noted by Aboolian, Cui, and Shen (2013), if $\bar{S}$ belongs to a neighborhood of $S^\star_{\text{MUFLP}}$, as defined earlier, then $\bar{x} = \left(\bar{x}_j\right)_{j\in N}$ does not satisfy (10). To see why consider a location set B. For any location set B' in the distance-2

neighborhood of B, the right-hand-side of (10) will equate to $|B| - 2$ or larger: if B' is obtained by adding one additional facility to B, then the right-hand-side of (10) is $|B| - 1$; if B' is obtained by removing one additional facility from B, then the right-hand-side of (10) is $|B| - 1$; if B' is obtained by adding two additional facility to B, then the right-hand-side of (10) is $|B| - 2$. Therefore, constraints (10) excludes the location sets in the distance-2 neighborhood of B without restricting any other location set outside of the neighborhood. For a detailed discussion on the general case of the k −distance neighborhood search, we refer the reader to the work of Aboolian, Cui, and Shen (2013).

Subsequently, we consider problem MUFLP(r).

$$[\text{MUFLP}(r)] \quad min \quad Z_{\text{MUFLP(r)}}(x, y, v) = \sum_{j \in N} f_j x_j + \sum_{i \in M} \sum_{j \in N} c_{ij}\, \lambda_{ij} v_{ij} + \sum_{l=1}^{n} \mathcal{C}_{\ell}\, v_l \qquad (11.1)$$

$$\text{subject to} \qquad \sum_{j \in S} x_j - \sum_{j \in S^c} x_j \leq |S| - 1 : \forall S \in \mathcal{A}_r, \qquad (11.2)$$

$$(7.2) – (7.10)$$

Here, $\mathcal{A}_r$ is the set of all starting subsets that have been used in neighborhood search in Algorithm 2 in steps 1 through $r - 1$. Therefore, to solve MUFLP(r) we first need to solve MUFLP(t), $t = 1, 2, \dots, r - 1$. We note that $\mathcal{A}_1 =$ and MUFLP(1) is the MUFLP by definition. Also note that Constraints 11.2 exclude all location sets that have already been evaluated in Algorithm 2, without excluding location sets that have not yet been evaluated.

Let $Z^{\star}_{\text{MUFLP}(r)}$, and $S^{\star}_{\text{MUFLP}(r)}$ be the optimal value of the MUFLP(r) and the set of open facilities corresponding to the optimal solution of MUFLP(r), respectively. Define $\mathcal{D}_r \subseteq \mathcal{P}(N)$ to be the set of all location sets in the feasible region of MUFLP(r), in which $\mathcal{P}(N)$ is the power set of N. Then $\mathcal{D}_1 = \mathcal{P}(N)$, is all possible location sets. Additionally, we have $\mathcal{D}_r = \mathcal{D}_{r-1} - \mathcal{G}\left(S^{\star}_{\text{MUFLP}(r)}\right)$, and $\mathcal{A}_r =$

$\mathcal{A}_{r-1} \cup \mathcal{A}\left(S^{\star}_{\text{MUFLP}(r)}\right)$, where $\mathcal{G}\left(S^{\star}_{\text{MUFLP}(r)}\right)$ and $\mathcal{A}\left(S^{\star}_{\text{MUFLP}(r)}\right)$ are found using Algorithm 2 when $S = S^{\star}_{\text{MUFLP}(r)}$.

Denote $S^{H}\left(S^{\star}_{\text{MUFLP}(r)}\right)$ and $F\left(S^{H}\left(S^{\star}_{\text{MUFLP}(r)}\right)\right), \mu^{\star}\left(S^{H}\left(S^{\star}_{\text{MUFLP}(r)}\right)\right)\Big)$ as the location set solution and the objective function value of SSDP given that location set, both found using Algorithm 2 when $S = S^{\star}_{\text{MUFLP}(r)}$.

Let $U(r)$ be the improved upper bound found after solving MUFLP(r), that is

$$\text{U(r)} = \min\left\{\text{U(r}-1), F\left(S^{H}\left(S^{\star}_{\text{MUFLP}(r)}\right)\right), \mu^{\star}\left(S^{H}\left(S^{\star}_{\text{MUFLP}(r)}\right)\right)\right\}. \quad (12)$$

It can be simply verified that $U(t)$ is non-increasing in t.

Similarly, let $L(r)$ be the improved lower bound found after solving MUFLP(r). Thus, we get

$$\text{L(r)} = \text{Z}^{\star}_{\text{MUFLP(r)}}. \quad (13)$$

Given the formulation of MUFLP(r), it is also easy to verify that $L(t)$ is non-decreasing in t.

The Search and Cut Algorithm, which formalizes this successive improvement for further iterations until the optimal solution is found, is described as follows:

ALGORITHM 3: THE SEARCH AND CUT ALGORITHM FOR SOLVING SSDP

Input:
Output: $Z^{\star}_{\text{SSDP}}, S^{\star}_{\text{SSDP}}$

1. Set $t = 1$, UB $= \infty$, $\mathcal{A}_t = \emptyset$ and $\mathcal{D}_t = \mathcal{P}(N)$.

2. Solve MUFLP(t) and set LB $= Z^{\star}_{\text{MUFLP}(t)}$, $S = S^{\star}_{\text{MUFLP}(t)}$.

3. If $\mathcal{D}_t = \emptyset$ or LB $>$ UB, go to Step **4**, else proceed.

   I) Apply Algorithm 2 using S to find $\mathcal{A} = \mathcal{A}(S)$, $\mathcal{G} = \mathcal{G}(S)$, $S^H = S^H(S)$, and $F\left(S^H, \mu^{\star}(S^H)\right)$.

   II) Set $t = t + 1$, $\mathcal{A}_t = \mathcal{A}_{t-1} \cup \mathcal{A}$, and $\mathcal{D}_t = \mathcal{D}_{t-1} - \mathcal{G}$.

   III) If UB $> F\left(S^H, \mu^{\star}(S^H)\right)$, then UB $= F\left(S^H, \mu^{\star}(S^H)\right)$ and $S^*_t = S^H$.

   IV) Go to **Step 2**.

4. Set $S^{\star}_{\text{SSDP}} = S^*_t$, $Z^{\star}_{\text{SSDP}}$ = UB, and STOP.

The algorithm terminates in Step 2 either when every possible location set has been evaluated, or when the value of most updated lower bound exceeds the value of the most updated upper bound. It is easy to conclude that when every possible location set has been evaluated then the Search and Cut algorithm already obtains the optimal solution.

Given the intricacies of the search and cut method, we include a diagram illustrating the complete methodology in Figure 1.

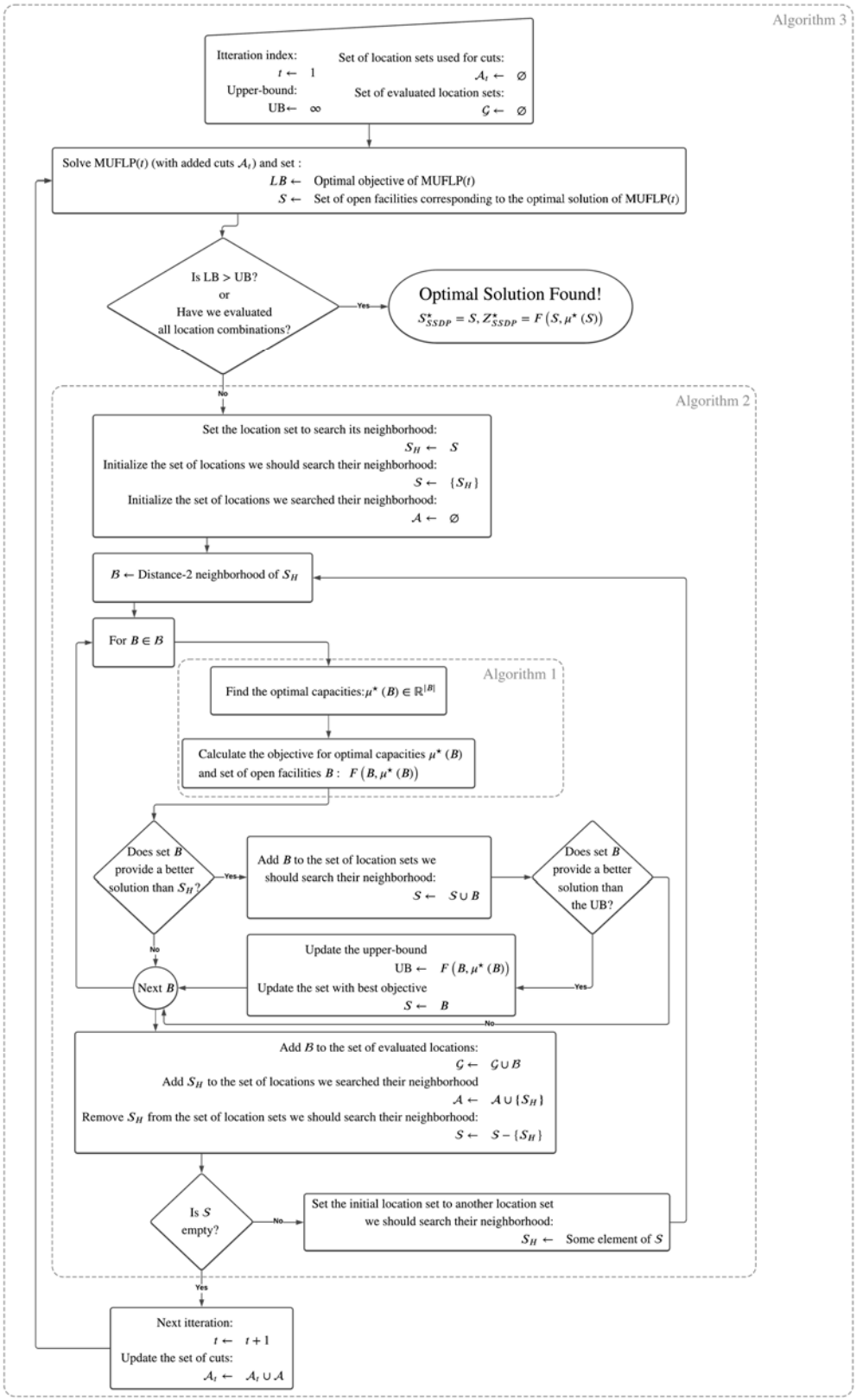

FIGURE 1. A FLOWCHART ILLUSTRATING THE COMPLETE SEARCH AND CUT METHODOLOGY.

Before we prove the exactness of the approach when lower bound exceeds the upper bound, we consider problem SSDP(r).

$$[\text{SSDP}(r)] \quad min \quad Z_{\text{SSDP(r)}}(x,y,z) = \sum_{j\in N} f_j x_j + \sum_{i\in M}\sum_{j\in N} c_{ij}\lambda_{ij} v_{ij} + \sum_{l=1}^{n} \omega(\bar{\mu}_k) z_{jk} + \alpha \sum_{j\in N} \frac{\sum_{i\in M}\lambda_i y_{ij}}{\sum_{k\in K}\bar{\mu}_k z_{jk} - \sum_{i\in M}\lambda_i y_{ij}} \qquad (14.1)$$

$$\text{subject to} \quad \sum_{j\in S} x_j - \sum_{j\in S^c} x_j \le |S| - 1 : \forall S \in \mathcal{A}_r, \qquad (14.2)$$

$$(1.2)\text{–}(1.9).$$

Recall that $\mathcal{A}_r$ is the set of all starting subsets that have been used in neighborhood search in Algorithm 2 in steps 1 through $r-1$. Thus, to solve SSDP(r) we first need to solve SSDP(t), t=1, 2,..., $r-1$. We note that $\mathcal{A}_1 =$ and SSDP(1) is the original SSDP by definition. Recall $\mathcal{D}_r$, which is the set of all location sets in the feasible region of MUFLP(r). We note that $\mathcal{D}_r$ is also all location sets in the feasible region of SSDP(r), that is MUFLP(r) and SSDP(r) have the same feasible region.

Let $Z^{\star}_{\text{SSDP}(r)}$ and $S^{\star}_{\text{SSDP}(r)}$ be the optimal value of the SSDP(r) and the set of open facilities corresponding to the optimal solution of SSDP(r), respectively. We note that, from the formulation of SSDP(r), it is easy to verify that $Z^{\star}_{\text{SSDP}(t)}$ is non-decreasing in t. We also note that since MUFLP(r) and SSDP(r) have the same feasible region, then using the same arguments to prove Lemma 1, we conclude

$$Z^{\star}_{\text{MUFLP}(r)} \le \quad Z^{\star}_{\text{SSDP}(r)} \text{ for } r \ge 1. \qquad (15)$$

The exactness of the Search and Cut approach in Algorithm 3 and the fact that it will converge to an optimal solution in finite number of steps is based on the following result:

Theorem 1 (Exactness of The Search and Cut Method) Algorithm 3 terminates and finds an optimal solution for SSDP in p steps when $\mathcal{D}_p = \emptyset$ or LB $= Z^{\star}_{\text{MUFLP}(p)} > F\left(S_p^{\star}, \mu^{\star}(S_p^{\star})\right) =$ UB, and then $Z^{\star}_{\text{SSDP}} = F\left(S_p^{\star}, \mu^{\star}(S_p^{\star})\right)$.

A proof is included in Section 6.1 in the Appendix.

3.5. Generalized Benders Decomposition for SSDP

Being a MINLP, SSDP is amenable for solution by Generalized Benders Decomposition (Geoffrion, 1972; Floudas, Aggarwal, and Ciric, 1989). Consider the SSDP once more, and let $S \subseteq N$ be a set of open facilities, and let x^S be the corresponding binary solution in which

$$x^S = \left(x_j^S : j \in 1,2,\cdots,N, x_j^S = 1 \text{ if } j \in S, 0 \text{ otherwise}\right) \cdot$$

Given x^S, y^S can be determined such that all the consumers are assigned to their closest facility. That is

$$y^S = \left(y_{ij}^S : i \in M, j \in N, y_{ij}^S = 1 \text{ if } j \in S \text{ and } j \text{ is the closest facility to } i, 0 \text{ otherwise}\right).$$

For a given S, having fixed variable x^S to $\bar{x}^S$ and y^S to $\bar{y}^S$, SSDP reduces to the series of subproblems:

$$\left[\text{SP}_j(S)\right] \min \quad Z(\mathbf{z}) = \sum_{k \in K} \left\{ \omega(\bar{\mu}_k) + \alpha \frac{\sum_{i \in M} \lambda_i \, \bar{y}_{ij}^S}{\bar{\mu}_k - \sum_{i \in M} \lambda_i \, \bar{y}_{ij}^S} \right\} z_{jk}$$

subject to:

$$\sum_{k \in K} z_{jk} = \bar{x}_j^S \,,$$

$$\sum_{k \in K} \bar{\mu}_k \, z_{jk} \geq \sum_{i \in M} \lambda_i \, \bar{y}_{ij}^S \,,$$

$$z_{jk} \in 0,1 \,, \ \forall k \in K \,.$$

Recall that $E_j(S)$ is the set of consumers closest to facility $j \in S$, and $\Lambda_j(S)$ is the demand rate at facility j given S, such that $\Lambda_j(S) = \sum_{i \in E_j(S)} \lambda_i$. Moreover, recall that $\bar{k}_j$, is the smallest capacity level for each facility as defined in (2). Thus $\mathrm{SP}_j(S)$ can be rewritten as the following simple mixed-integer linear program.

$$
\begin{aligned}
\left[\mathrm{SP}_j(S)\right] \min \quad & Z(\mathbf{z}) = \sum_{k=\bar{k}_j}^{\kappa} \left\{ \omega(\bar{\mu}_k) + \alpha \frac{\sum_{i \in M} \lambda_i \bar{y}_{ij}^S}{\bar{\mu}_k - \sum_{i \in M} \lambda_i \bar{y}_{ij}^S} \right\} z_{jk} \\
\text{subject to:} \quad & \\
& \sum_{k=\bar{k}_j}^{\kappa} z_{jk} = \bar{x}_j^S , \\
& z_{jk} \in \{0,1\}, \ \forall k \in \{\bar{k}_j, \bar{k}_j + 1, \dots, \kappa\} .
\end{aligned}
$$

Furthermore, without altering the optimal objective of $\mathrm{SP}_j(S)$, the binary requirement on z_{jk} can be relaxed.

$$
\begin{aligned}
\left[\mathrm{SP}_j(S)\right] \min \quad & Z(\mathbf{z}) = \sum_{k=\bar{k}_j}^{\kappa} \left\{ \omega(\bar{\mu}_k) + \alpha \frac{\sum_{i \in M} \lambda_i \bar{y}_{ij}^S}{\bar{\mu}_k - \sum_{i \in M} \lambda_i \bar{y}_{ij}^S} \right\} z_{jk} \\
\text{subject to:} \quad & \\
& \sum_{k=\bar{k}_j}^{\kappa} z_{jk} = \bar{x}_j^S , \\
& z_{jk} \geq 0, \ \forall k \in \{\bar{k}_j, \bar{k}_j + 1, \dots, \kappa\} .
\end{aligned}
$$

In light of the almost identical problem of determining the optimal capacities for a set of open facilities in the search and cut methodology and the generalized Benders decomposition, it is worth discussing why the relaxed LP cannot be exploited in both approaches. We note that while the degenerate cases violating the integrality condition of z_{jk}s do not change the optimal objective of $\mathrm{SP}_j(S)$ – which is what the generalized Benders decomposition uses in the subproblem duals – the

modified descent approach as well as the search and cut method require the exact choice of z_{jk} repeatedly throughout Algorithm 1 and Algorithm 2.

The dual of $\mathrm{SP}_j(S)$ is:

$$\begin{aligned}
\left[\mathrm{DSP}_j(S)\right] \max \quad & Y(\boldsymbol{\zeta}_j) = \zeta_j \bar{x}_j^S \\
\text{subject to:} \quad & \\
& \zeta_j \leq \omega(\bar{\mu}_k) + \alpha \frac{\Lambda_j(S)}{\bar{\mu}_k - \Lambda_j(S)}\,,\ \forall k \in \{\bar{k}_j, \bar{k}_j + 1, \dots, \kappa\}\,.
\end{aligned}$$

The Generalized Benders master problem is:

$$\begin{aligned}
\min\ [\mathrm{MP}(q)] \quad & \mathrm{Z}(\mathbf{x}, \mathbf{y}, \gamma) = \gamma, \qquad (17.1) \\
\text{subject to:} \quad & \\
& \gamma - \left(\sum_{j \in N} f_j\, x_j + \sum_{i \in M} \sum_{j \in N} c_{ij}\, \lambda_{ij} y_{ij} + \sum_{j \in N} \zeta_{j,t}^{\star}\, x_j \right) \geq 0\,,\ \forall t \in \{1, 2, \dots, \mathrm{q} - 1\}, \qquad (17.2) \\
& \sum_{j \in N} y_{ij} = 1\,,\ \forall i \in M\,, \qquad (17.3) \\
& \sum_{j' \in N} c_{ij'}\, y_{ij'} \leq (c_{ij} - L) x_j + L\,,\ \forall i \in M\,, \forall j \in N\,, \qquad (17.4) \\
& y_{ij} \leq x_j\,,\ \forall i \in M\,, \forall j \in N\,, \qquad (17.5) \\
& y_{ij} \in \{0, 1\}\,,\ \forall i \in M\,, \forall j \in N\,, \qquad (17.6) \\
& x_j \in \{0, 1\}\,,\ \forall j \in N\,, \qquad (17.7) \\
& \gamma_j \geq 0\,,\ \forall j \in N\,, \qquad (17.8)
\end{aligned}$$

in which $\zeta_{j,t}^{\star}$ is the optimal solution of $\mathrm{DSP}_j(S_t)$, and S_t is the optimal location set of $\mathrm{MP}(t)$ for $t \in 1,2,\cdots,q-1$. We note that $\zeta_{j,0}^{\star} = 0, j \in N$. We also note that if $j \notin S_t$, then $\zeta_{j,t}^{\star} = 0$. Constraints 17.2 are the Benders cuts that are typically added iteratively, each time $\mathrm{DSP}_j(S_t)$s are solved. Given the above formulation $\mathrm{MP}(1)$ reduces to the traditional UFLP. A solution of SSDP can be found by successively solving the sub-problems and the master problem, with added cuts. A necessary condition for finite convergence is to have the objective function and the feasible set of SSDP to be

convex in $\mathbf{z}$, once $\mathbf{x}$ and $\mathbf{y}$ are fixed, and the feasible set of possible realization of $\mathbf{z}$ values to be convex, which is unfortunately not the case in this problem as $\mathbf{z}$ is binary, for a detailed discussion see Cai et al. (2001). The Generalized Benders Decomposition algorithm, however, may still be used as a heuristic (Geoffrion, 1972). In all case in which we were able to use Generalized Benders Decomposition to solve the SSDP within the time limit set in the numerical experiments, it generated an optimal solution. The optimally of the instances were verified using the exact method introduced in Section 3.4. The steps of the proposed Generalized Benders Decomposition are detailed in Algorithm 4.

ALGORITHM 4: SSDP SOLUTION VIA GENERALIZED BENDERS DECOMPOSITION

Input: ϵ
Output: $S^{\star}_{\text{SSDP}}, Z^{\star}_{\text{SSDP}}$ (the set of open facilities and the objective value corresponding to the Generalized Benders Decomposition solution of SSDP, respectively)

1. LB $= -\infty$, UB $= +\infty$.
2. Set $t = 1, \zeta^{\star}_{j,0} = 0, j \in N$, and solve MP(1). Let $S_1 = S^{\star}_{\text{MP}(1)}$ be the set of open facilities corresponding to the optimal solution of MP(1).
3. For $j \in S_t$:
   I) let $E_j(S_t) = i : j = \text{argmin}_{j \in S_t} c_{ij}$, $i \in M$, and $\Lambda_j(S_t) = \sum_{i \in E_j(S_t)} \lambda_i$.
   II) Let $\bar{k}_j = \arg\min_{k \in K} \{k : \Lambda_j(S) \leq \bar{\mu}_k\}$ for $j \in S$.
   III) Solve DSP$_j(S_t)$ and let $\zeta^{\star}_{j,t}$ be the optimal solution of the DSP$_j(S_t)$
4. If UB $> \sum_{j \in S_t} f_j + \sum_{j \in S_t} \sum_{i \in M} c_{ij}\, \lambda_i \bar{y}_{ij} + \sum_{j \in S_t} \zeta^{\star}_{j,t}$, then: Set UB $= \sum_{j \in S_t} f_j + \sum_{j \in S_t} \sum_{i \in M} c_{ij}\, \lambda_i \bar{y}_{ij} + \sum_{j \in S_t} \zeta^{\star}_{j,t}$, and $S^{\star} = S_t$.
5. Set $t = t + 1$.
6. Add Benders cut $\gamma - \left(\sum_{j \in N} f_j\, x_j + \sum_{i \in M} \sum_{j \in N} c_{ij}\, \lambda_{ij} y_{ij} + \sum_{j \in N} \zeta^{\star}_{j,t}\, x_j\right) \geq 0$.
7. Solve MP(t), let $Z^{\star}_{\text{MP}(t)}$ and $S_t = S^{\star}_{\text{MP}(t)}$ be the optimal objective, and the set of facilities corresponding to the optimal solution, and set LB $= Z^{\star}_{\text{MP}(t)}$.
8. If UB $-$ LB $\leq \epsilon$, set $S^{B}_{\text{SSDP}} = S^{\star}, Z^{B}_{\text{SSDP}} =$ UB and STOP. Else go to Step 4.

As we discuss in Section 4, the generalized Benders decomposition provides an efficient solution for SSDP when the number of candidate facilities are small (relative to the number of consumers). When the number of facilities is large, however, the Generalized Benders method may not provide solutions within a reasonable amount of time.

We also note that generalized Benders decomposition, as well as other solution approaches for solving SSDP discussed in the literature are highly sensitive to the value of κ – the number of capacity levels. In contrast, the modified descent approach and the search and cut methodology – discussed in Section 3.3 and 3.4, respectively – which repeatedly use Algorithm1 are not as sensitive to an increase in κ. This is because finding the capacity rates resulting in the least capacity and waiting time cost in Algorithm1 has a polynomial time complexity.

IV. NUMERICAL TESTING

We tested the two solution methodologies of the generalized Benders decomposition and search and cut on a hypothetical problem of locating a set of service facilities in the city of Toronto, Canada. We assume that the new service will be targeted to a demographic, who makes up roughly 1% of the Canadian population and require service ten times a year. We assume the maximum probability of participation to be 95% (i.e., this is the probability of participation by the member of the target demographic). We assume that each service facility will be open 52 weeks per year, 5 days per week, and 10 hours per day. The waiting cost is assumed to be α =\$20 per hour. The travelling cost is θ dollars per hour which might be lower (if the travel is on the way to work), higher (if we also consider the transportation cost on the top of lost time) or the same as waiting cost. Therefore we consider $\theta \in \{0.5\alpha, \alpha, 2\alpha\}$. Furthermore, we set $c_{ij} = \theta \frac{d_{ij}}{40}$, in which 40 miles/hour is the average speed in the City of Toronto. We assume 10 levels of available capacity for each facility. Each service facility will

house a certain capacity, with a maximum capacity of 120% of the overall demand rate. That is $\bar{\mu}_{10} \leq 1.2\sum_{i\in M}\lambda_i = 1.2\Lambda$. In particular $\bar{\mu}_k \in \{\frac{1.2\Lambda}{10}, \frac{1.2\Lambda}{9}, \cdots, \frac{1.2\Lambda}{2}, 1.2\Lambda\}, \forall k \in \{1,2,\cdots,10\}$, in which $\bar{\mu}_k$ is a scalar denoting the k^{th} level of capacity rate. We assume that the hourly operating cost for the first capacity unit is β. The cost for each additional unit will decrease. We set $(\beta\mu)^{\phi}$ to be the cost to obtain μ units of capacity, where $\phi < 1$. We consider three values of $\phi \in \{0.5,0.75,0.99\}$. Moreover, we consider various values of β in which $\beta = \{0.1\alpha, 0.5\alpha, \alpha, 2\alpha\}$, and 10α. The city of Toronto is divided into 96 regions called Forward Sortation Areas (FSAs), see Fig. 2.

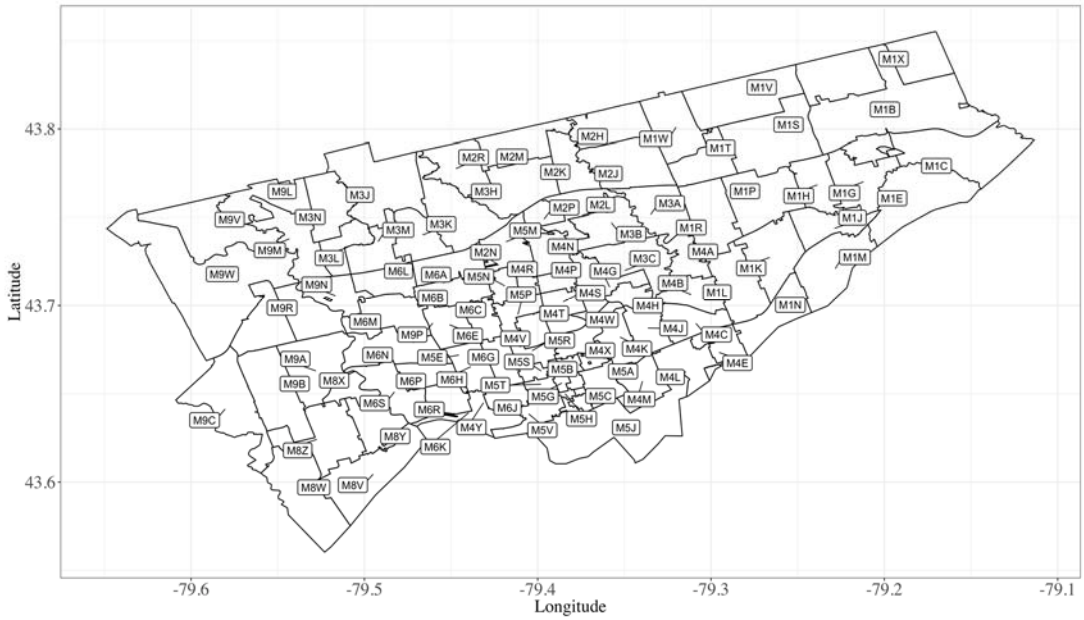

FIGURE 2. CITY OF TORONTO'S POSTAL CODE BOUNDARIES AT THE FSA. THE FSA IS CONTAINED WITHIN THE FIRST THREE CHARACTERS OF THE POSTAL CODE AND IS USED TO IDENTIFY AN URBAN OR RURAL GEOGRAPHICAL AREA.

These regions are roughly equivalent to the 5-digit ZIP codes in the US. Euclidean travel distance between the region centroids is assumed. Demographic data at the FSA level is available from Statistics Canada (Statistics Canada, 2016). Finally, for fixed location costs we use the current

commercial real estate lease prices per square foot (averaged for each FSA) provided by the Toronto Real Estate Board (Toronto Real State Board, 2019), and we assume a 2000 square foot size for each facility. We consider three cases with $m = 96$, and $n = 10,24,$ and 96 and solve problems for different combinations of $\beta, \theta,$ and ϕ (45 in total). In each problem, we solved the corresponding SSDP using both algorithms and find the optimal location, optimal cost, CPU time and optimal assignments.

All problem instances were solved on a machine with 3.4 GHz 4-Core Intel Core i5 CPU, with 16 GB RAM running macOS Mojave. We set the time limit of 7200 seconds (2 hours) for each instance for both Generalized Benders Decomposition (with $\epsilon = 0.001$) and Search and Cut methodologies. If the algorithm failed to converge during this time, the relative gap is reported. All problem instances were solved using the python docplex package, version 2.9.141, except MCWCPs (5) which were solved using Wolfram Mathematica (version 11.1.1.0) and its FindMinimum function with the accuracy goal set to 10,000, which enforces the convergence criteria of $||x_k - x^\star|| \leq 10^{-10000}$ and $\nabla f(x_k) < 10^{-10000}$ for the FindMinimum function. MCWCPs were solved for an average of 0.5885, 1.9322, and 23.2393 seconds for the three parameters of $n = 10,\ 24,$ and 96, respectively.

When applying Generalized Benders Decomposition, for $n = 24$, only 15 (out of 45) instances of the converge within the time limit of 2 hours, and for the cases that do not converge, the average solution gap is approximately 5%. As such, we only report the Search and Cut method's results for the case of $n = 24$ and $n = 96$. A detailed summary of the test is presented in Table 1, Table 2, and Table 3 (see Table 4 for a reference of FSAs to numerical IDs) in Section 6.2 in the Appendix.

The numerical results provide further insight into various model parameters such as capacity cost, access cost, the degree of concavity of the capacity function, and the number of candidate facilities.

Capacity Cost: Parameter β has a non-linear relationship with CPU time, in which the initial increase in β decreases the CPU time, and further increase of β results in a spike of CPU time. Additionally, as β increases, number of open facilities, and total service capacity decrease; and the total average distance increases, see Fig. 3.

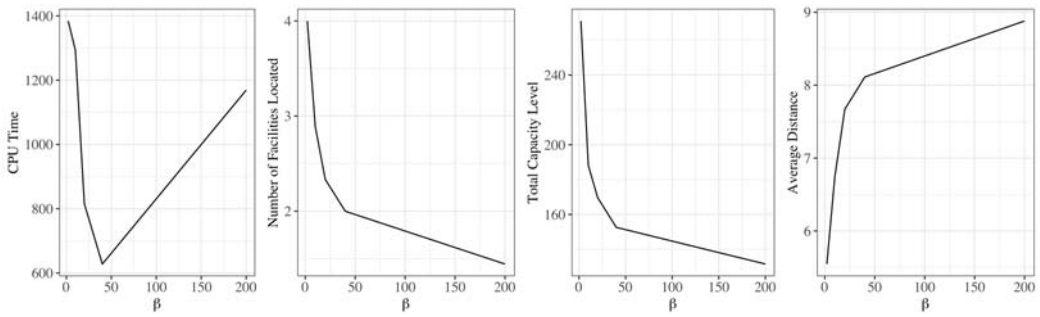

FIGURE 3. THE EFFECT OF PARAMETER β ON AVERAGE: CPU TIME, NUMBER OF FACILITIES LOCATED, TOTAL CAPACITY, AND DISTANCE TO FACILITIES LOCATED.

The CPU time results are mainly due to the fact that extreme capacity costs, too small or too large, increases the size of the neighborhood with potential improvement in the modified descent algorithm, Algorithm 2. As such, the search takes a longer time to find the best set of improvements. The effect of β on the number of open facilities, capacity levels, and average distances are more straightforward. Higher capacity cost means a lower number of facilities will be selected which results in larger travel distances and lower capacity level assignments.

Access Cost: When θ increases, the CPU time, number of open facilities, and total service capacity increases while the total average distances decrease, see Fig. 4.

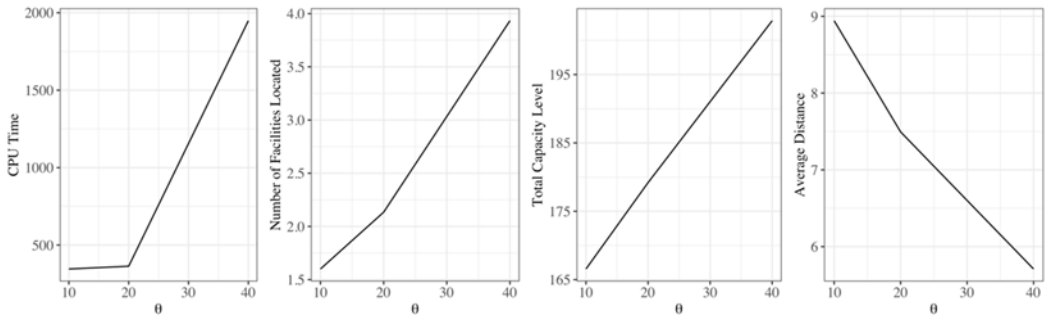

FIGURE 4. THE EFFECT OF PARAMETER θ ON AVERAGE: CPU TIME, NUMBER OF FACILITIES LOCATED, TOTAL CAPACITY, AND DISTANCE TO FACILITIES LOCATED.

The access cost has an opposite effect on the number of open facilities, capacity levels, and average distances, when compared to the capacity cost. Higher access cost means that more facilities should be utilizes, which in turn increases the capacity levels and decreases the distance traveled by consumers to receive services. The higher access cost also increases the size of the neighborhood with potential improvement in the modified descent algorithm, which results in longer time spent on the search for finding improved set of open facilities.

Capacity Concavity: As ϕ increases, the CPU time decreases initially and spikes when $\phi = 0.99$. Furthermore, as ϕ increases the number of open facilities and total service capacity decrease; and total average distance increases, see Fig. 5.

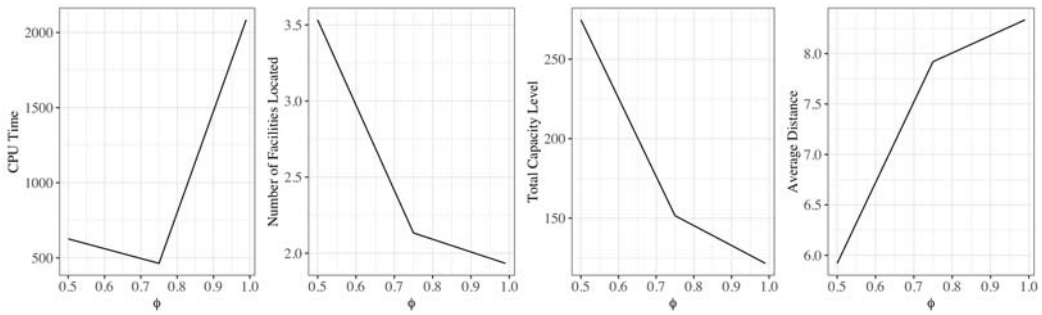

FIGURE 5. THE EFFECT OF PARAMETER ϕ ON AVERAGE: CPU TIME, NUMBER OF FACILITIES LOCATED, TOTAL CAPACITY, AND DISTANCE TO FACILITIES LOCATED.

Since most of the capacity levels assigned are greater than 1, as the degree of the concavity of the capacity increases, so does the capacity cost. Thus, the effects of the degree of the concavity of the capacity cost are akin to the effects of capacity cost for similar reasons.

Number of candidate facilities: As expected, the increase in the number of candidate facilities increases the complexity of the solutions measured by CPU time, see Figure 6.

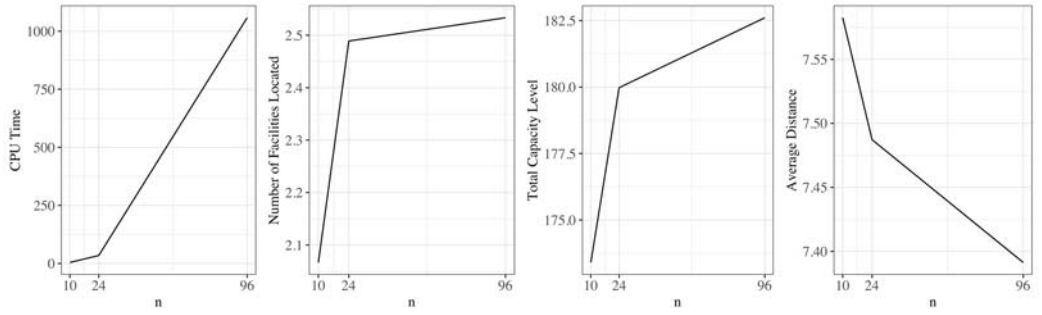

FIGURE 6. THE EFFECT OF PARAMETER n ON AVERAGE: CPU TIME, NUMBER OF FACILITIES LOCATED, TOTAL CAPACITY, AND DISTANCE TO FACILITIES LOCATED.

For the Search and Cut methodology, the average CPU time is increased by a factor of approximately 7.6 times (minimum factor of 4.56, maximum factor of 28.00) when increasing the number of candidate facilities from $n = 10$ to $n = 24$. The complexity is further increased by a factor of

approximately 27.2 times (minimum factor of 6.31, maximum factor of 143.93) when the number of candidate facilities increases from $n = 24$ to $n = 96$. The average improvement of the optimal costs are disproportionate to the increase of complexity, in which on average, the optimal costs are approximately 0.6% and 1.6% lower for the change from $n = 10$ to $n = 24$, and from $n = 24$ to $n = 96$ respectively.

V. CONCLUSION

In this paper, we discuss the service system design problem for congested facilities that behave as M/M/1 queuing systems where consumers choose which facility to use, and the problem is to locate facilities and allocate service capacity to minimize the overall cost of the social system. The problem is modeled as a MINLP. The nonlinearity of the objective function is the challenging part of developing an exact solution approach, but we exploit the structure of the model and provide two exact approaches. In the first approach, we apply Generalized Benders Decomposition of the problem into various LP subproblems and MIP master problems. In the second methodology, we use a novel special-purpose algorithm in which we define a MIP model to find the lower bound and a modified descent search approach to find an upper bound. At each step, we add a set of cuts of the solution space which has been recently searched and resolve the MIP to find an improved lower bound and continue the search of the solution space to improve the upper bound. We repeat these steps until the overall cost of the most updated lower bound becomes greater than the overall cost of the most updated upper bound. We then prove that when the overall cost of the most updated upper bound is less than the overall cost of the most updated lower bound, then the solution to the most updated upper bound is the optimal solution to the original problem. The Generalized Benders Decomposition approach tends to be efficient for small size problems (96 demand points, ten candidate facilities, and

ten capacity levels). However, its efficiency decreases as we increase the number of candidate facilities to 24. The Search and Cut approach seems to be more efficient for both small and large size problems. The Search and Cut approach finds optimal solutions in a reasonable time for problems with as large as 96 demand points, 96 candidate facilities, and ten capacity levels. We also want to emphasize that the Search and Cut approach with minor modifications could also be used to optimally solve the M/M/k case significantly better than the algorithm offered in Aboolian, Berman, and Drezner (2008). For future research, we would like to suggest developing exact methodologies capable of solving the following problems.

- The problems in which the facilities behave as M/G/1 queuing system, similar to Vidyarthi and Jayaswal (2014), but with consumer choice considerations.
- The problems in which the service capacity is continuous and can take any value, similar to Elhedhli, Wang, and Saif (2018), but with consumer choice considerations.

VI. APPENDIX

6.1. Proofs

Proof of Lemma 1.

$$\begin{aligned} Z^{\star}_{\text{SSDP}} &= F\left(S^{\star}_{\text{SSDP}}, \mu^{\star}(S^{\star}_{\text{SSDP}})\right) \\ &= \sum_{j \in S^{\star}_{\text{SSDP}}} f_j + \sum_{j \in S^{\star}_{\text{SSDP}}} \sum_{i \in E_j} c_{ij}\, \lambda_i + \bar{F}\left(S^{\star}_{\text{SSDP}}, \mu^{\star}(S^{\star}_{\text{SSDP}})\right) \\ &\geq \sum_{j \in S^{\star}_{\text{SSDP}}} f_j + \sum_{j \in S^{\star}_{\text{SSDP}}} \sum_{i \in E_j} c_{ij}\, \lambda_i + \mathcal{C}^{\star}_{|S^{*}_{\text{SSDP}}|}, \qquad (8) \end{aligned}$$

in which $F(\cdot)$, and $\bar{F}(\cdot)$ are as found in Algorithm 1, and the inequality is derived from (6). Furthermore, since $S^{\star}_{\text{SSDP}}$ is a feasible solution of MUFLP, we get

$$
\begin{aligned}
Z^{\star}_{\text{MUFLP}} = & \sum\nolimits_{j \in S^{\star}_{\text{MUFLP}}} f_j + \sum\nolimits_{j \in S^{\star}_{\text{MUFLP}}} \sum\nolimits_{i \in E_j} c_{ij}\, \lambda_i + \mathcal{C}^{\star}_{|S^{*}_{\text{MUFLP}}|} \\
\leq_{(8)} & \sum_{j \in S^{\star}_{\text{SSDP}}} f_j + \sum_{j \in S^{\star}_{\text{SSDP}}} \sum_{i \in E_j} c_{ij}\, \lambda_i + \mathcal{C}^{\star}_{|S^{*}_{\text{SSDP}}|} \\
\leq & \; Z^{\star}_{\text{SSDP}},
\end{aligned}
$$

which completes the proof.

Proof of Theorem 1. It is evident that if $\mathcal{D}_p = \emptyset$, then the neighborhood search must have exhausted all the location sets and found the optimal solution by complete enumeration. To see why $Z^{\star}_{\text{SSDP}} = F\left(S^{\star}_p, \mu^{\star}(S^{\star}_p)\right)$ when $Z^{\star}_{\text{MUFLP}(p)} > F\left(S^{\star}_p, \mu^{\star}(S^{\star}_p)\right)$, assume otherwise and let $Z^{\star}_{\text{SSDP}} < F\left(S^{\star}_p, \mu^{\star}(S^{\star}_p)\right)$, while $Z^{\star}_{\text{MUFLP}(p)} > F\left(S^{\star}_p, \mu^{\star}(S^{\star}_p)\right)$. Since $Z^{\star}_{\text{SSDP}} < F\left(S^{\star}_p, \mu^{\star}(S^{\star}_p)\right)$, then $Z^{\star}_{\text{SSDP}} < F\left(S^{\star}_t, \mu^{\star}(S^{\star}_t)\right), t \in \{1,2,\cdots,p-1\}$, and consequently $S^{\star}_{\text{SSDP}} \in \mathcal{D}_p$. Hence the following holds.

$$
Z^{\star}_{\text{MUFLP}(p)} \leq Z^{\star}_{\text{SSDP}(p)} = Z^{\star}_{\text{SSDP}} \leq F\left(S^{\star}_p, \mu^{\star}(S^{\star}_p)\right), \quad (16)
$$

in which the first inequality is derived from equation (15), and the equality is true since $S^{\star}_{\text{SSDP}}$ belongs to $\mathcal{D}_p$, the feasible region of SSDP(p). Finally, the second inequality is true since $S^{\star}_{\text{SSDP}} \in \mathcal{D}_p$ has yet to be evaluated in the neighborhood search in Algorithm 2 in search of an upper bound. As such $Z^{\star}_{\text{MUFLP}(p)} \leq F\left(S^{\star}_p, \mu^{\star}(S^{\star}_p)\right)$ and the claim is proved by contradiction. Finally, we note that although the search and cut approach is not a branch-and-bound method, but similar to branch-and-bound method, it exhausts all of the possible solutions to an optimal solution.

6.2. Detailed Numerical Results

TABLE 1: NUMERICAL TESTING OF THE PROBLEM FOR TORONTO FSA DATA USING THE SEARCH AND CUT, AND GENERALIZED BENDERS DECOMPOSITION FOR $n = 10$.

Instance			Solution and Optimal Cost		Solution Via GBD				Solution Via Search and Cut			
β	θ	ϕ	$S^{\star}_{\text{SSDP}}$	$Z^{\star}_{\text{SSDP}}$	CPU Time	Status	# of Iter.	# of Cuts	CPU Time	Status	# of Iter.	# of Cuts
2	10	0.5	[12, 69]	273.07	9.07	Optimal	7	6	5.06	Optimal	2	2
2	10	0.75	[12, 69]	357.11	21.90	Optimal	20	19	2.55	Optimal	1	1
2	10	0.99	[12, 69]	572.75	26.28	Optimal	24	23	2.53	Optimal	1	1
2	20	0.5	[12, 51, 69, 87]	430.40	21.58	Optimal	15	14	4.28	Optimal	1	1
2	20	0.75	[12, 69, 87]	536.82	41.43	Optimal	29	28	4.85	Optimal	1	2
2	20	0.99	[12, 69]	752.84	69.21	Optimal	49	48	2.62	Optimal	1	1
2	40	0.5	[12, 31, 64, 87]	687.03	17.99	Optimal	10	9	4.34	Optimal	1	1
2	40	0.75	[12, 31, 64, 87]	804.30	34.60	Optimal	21	20	4.33	Optimal	1	1
2	40	0.99	[12, 31, 64, 87]	1043.89	86.30	Optimal	48	47	7.98	Optimal	2	2
10	10	0.5	[12, 69]	311.12	12.71	Optimal	11	10	5.15	Optimal	2	2
10	10	0.75	[12, 69]	576.65	49.46	Optimal	37	36	3.92	Optimal	2	2
10	10	0.99	[69]	1488.12	150.74	Optimal	79	78	3.92	Optimal	2	2
10	20	0.5	[12, 69, 87]	486.54	32.60	Optimal	24	23	3.18	Optimal	1	1
10	20	0.75	[12, 69]	756.74	69.15	Optimal	47	46	5.23	Optimal	2	2
10	20	0.99	[12, 69]	1703.80	193.50	Optimal	94	93	4.13	Optimal	2	2
10	40	0.5	[12, 31, 64, 87]	752.33	27.92	Optimal	17	16	4.26	Optimal	1	1
10	40	0.75	[12, 31, 64, 87]	1089.52	103.71	Optimal	57	56	4.36	Optimal	1	1
10	40	0.99	[12, 69]	2063.98	209.52	Optimal	97	96	5.37	Optimal	2	2
20	10	0.5	[12, 69]	338.80	17.00	Optimal	15	14	5.08	Optimal	2	2
20	10	0.75	[69]	724.41	77.01	Optimal	51	50	3.85	Optimal	2	2
20	10	0.99	[69]	2509.02	159.09	Optimal	81	80	4.03	Optimal	2	2
20	20	0.5	[12, 69]	518.89	32.01	Optimal	24	23	5.26	Optimal	2	2
20	20	0.75	[12, 69]	944.09	80.47	Optimal	54	53	2.54	Optimal	1	1
20	20	0.99	[62]	2767.63	341.93	Optimal	136	135	3.96	Optimal	2	2
20	40	0.5	[12, 31, 64, 87]	796.47	33.27	Optimal	19	18	4.26	Optimal	1	1
20	40	0.75	[12, 69]	1304.27	126.82	Optimal	66	65	2.80	Optimal	1	1
20	40	0.99	[12, 69]	3159.26	391.35	Optimal	143	142	5.23	Optimal	2	2
40	10	0.5	[12, 69]	376.95	24.29	Optimal	22	21	2.51	Optimal	1	1
40	10	0.75	[69]	955.86	56.52	Optimal	40	39	3.86	Optimal	2	2
40	10	0.99	[69]	4455.84	161.70	Optimal	81	80	3.90	Optimal	2	2
40	20	0.5	[12, 69]	557.03	29.01	Optimal	23	22	2.73	Optimal	1	1
40	20	0.75	[62]	1214.47	114.39	Optimal	63	62	3.97	Optimal	2	2
40	20	0.99	[62]	4714.44	342.22	Optimal	132	131	3.97	Optimal	2	2
40	40	0.5	[12, 31, 64, 87]	854.58	36.75	Optimal	22	21	4.27	Optimal	1	1
40	40	0.75	[12, 69]	1619.01	172.17	Optimal	87	86	5.26	Optimal	2	2
40	40	0.99	[62]	5227.31	595.11	Optimal	185	184	10.02	Optimal	2	6
200	10	0.5	[12, 69]	527.50	35.31	Optimal	29	28	2.71	Optimal	1	1
200	10	0.75	[69]	2293.95	72.32	Optimal	47	46	3.87	Optimal	2	2
200	10	0.99	[69]	19826.4	223.20	Optimal	79	78	3.86	Optimal	2	2
200	20	0.5	[12, 69]	707.59	49.64	Optimal	39	38	2.55	Optimal	1	1
200	20	0.75	[62]	2552.55	119.90	Optimal	67	66	3.95	Optimal	2	2
200	20	0.99	[62]	20085	327.31	Optimal	127	126	5.93	Optimal	3	3
200	40	0.5	[12, 64, 87]	1067.41	84.45	Optimal	49	48	2.75	Optimal	1	1
200	40	0.75	[62]	3065.42	266.46	Optimal	88	87	3.96	Optimal	2	2
200	40	0.99	[62]	20597.9	505.60	Optimal	166	165	9.82	Optimal	5	5

TABLE 2: NUMERICAL TESTING OF THE PROBLEM FOR TORONTO FSA DATA USING THE SEARCH AND CUT FOR $n = 24$.

Instance			Solution via Search and Cut					
β	θ	ϕ	$S^{\star}_{\text{SSDP}}$	$Z^{\star}_{\text{SSDP}}$	CPU Time	Status	# of Iter.	# of Cuts
2	10	0.5	[12, 69]	273.07	32.61	Optimal	2	2
2	10	0.75	[12, 69]	357.11	14.96	Optimal	1	1
2	10	0.99	[12, 69]	572.75	15.30	Optimal	1	1
2	20	0.5	[3, 27, 41, 69, 87]	413.53	31.68	Optimal	1	1
2	20	0.75	[3, 27, 41, 87]	534.23	26.24	Optimal	1	1
2	20	0.99	[12, 69]	752.84	14.64	Optimal	1	1
2	40	0.5	[3, 7, 15, 27, 41, 64, 69, 87, 91]	633.48	121.48	Optimal	1	2
2	40	0.75	[3, 72, 15, 87, 27, 41, 91]	778.46	85.93	Optimal	1	2
2	40	0.99	[3, 27, 41, 72, 87, 91]	1030.28	130.29	Optimal	4	4
10	10	0.5	[12, 69]	311.12	26.94	Optimal	2	2
10	10	0.75	[12, 69]	576.65	22.97	Optimal	2	2
10	10	0.99	[27]	1483.42	22.51	Optimal	2	2
10	20	0.5	[3, 27, 41, 87]	482.40	26.44	Optimal	1	1
10	20	0.75	[12, 69]	756.74	26.95	Optimal	2	2
10	20	0.99	[12, 69]	1703.80	27.23	Optimal	2	2
10	40	0.5	[3, 27, 41, 72, 87, 91]	725.86	40.41	Optimal	1	1
10	40	0.75	[94, 3, 72, 27]	1085.29	73.33	Optimal	2	3
10	40	0.99	[12, 69]	2063.98	50.63	Optimal	3	3
20	10	0.5	[12, 69]	338.80	27.07	Optimal	2	2
20	10	0.75	[27]	719.71	22.66	Optimal	2	2
20	10	0.99	[27]	2504.32	22.45	Optimal	2	2
20	20	0.5	[12, 69]	518.89	32.69	Optimal	2	2
20	20	0.75	[12, 69]	944.09	14.33	Optimal	1	1
20	20	0.99	[49]	2767.22	22.44	Optimal	2	2
20	40	0.5	[3, 27, 41, 72, 87, 91]	782.23	40.46	Optimal	1	1
20	40	0.75	[12, 69]	1304.27	14.48	Optimal	1	1
20	40	0.99	[12, 69]	3159.26	45.26	Optimal	3	3
40	10	0.5	[12, 69]	376.95	14.26	Optimal	1	1
40	10	0.75	[27]	951.16	22.77	Optimal	2	2
40	10	0.99	[27]	4451.13	22.46	Optimal	2	2
40	20	0.5	[12, 69]	557.03	14.42	Optimal	1	1
40	20	0.75	[49]	1214.06	22.48	Optimal	2	2
40	20	0.99	[49]	4714.03	22.96	Optimal	2	2
40	40	0.5	[3, 27, 64, 87]	849.07	25.58	Optimal	1	1
40	40	0.75	[12, 78]	1611.87	38.56	Optimal	2	3
40	40	0.99	[12, 78]	5210.07	45.72	Optimal	2	4
200	10	0.5	[12, 69]	527.50	14.37	Optimal	1	1
200	10	0.75	[27]	2289.24	22.81	Optimal	2	2
200	10	0.99	[27]	19821.73	22.44	Optimal	2	2
200	20	0.5	[12, 69]	707.59	14.28	Optimal	1	1
200	20	0.75	[49]	2552.14	22.51	Optimal	2	2
200	20	0.99	[49]	20084.63	35.57	Optimal	3	3
200	40	0.5	[12, 31, 72]	1062.82	19.89	Optimal	1	1
200	40	0.75	[49]	3064.79	23.08	Optimal	2	2
200	40	0.99	[49]	20597.27	112.50	Optimal	9	9

TABLE 3: NUMERICAL TESTING OF THE PROBLEM FOR TORONTO FSA DATA USING THE SEARCH AND CUT FOR $n = 96$. IN CASE THE ALGORITHM DID NOT CONVERGE WITHIN THE TIME LIMIT, THE OPTIMAL VALUE AND THE CORRESPONDING OPTIMAL SET OF OPEN FACILITIES WERE FOUND BY SETTING A HIGHER TIME THRESHOLD.

Instance			Solution via Search and Cut					
β	θ	ϕ	$S^{\star}_{\mathrm{SSDP}}$	$Z^{\star}_{\mathrm{SSDP}}$	CPU Time	Status	# of Iter.	# of Cuts
2	10	0.5	[14, 41, 77]	259.91	654.79	Optimal	2	2
2	10	0.75	[26, 77]	349.93	451.52	Optimal	2	2
2	10	0.99	[26, 77]	566.66	453.60	Optimal	2	2
2	20	0.5	[4, 27, 41, 77]	397.04	447.69	Optimal	1	1
2	20	0.75	[14, 41, 77]	514.16	339.81	Optimal	1	1
2	20	0.99	[14, 41, 77]	741.09	656.14	Optimal	2	2
2	40	0.5	[64, 4, 37, 84, 92, 29, 14]	594.83	1753.88	Optimal	1	2
2	40	0.75	[64, 4, 37, 84, 92, 29, 14]	739.14	722.91	Optimal	1	1
2	40	0.99	[64, 4, 37, 87, 29, 14]	992.60	6976.94	Optimal	8	10
10	10	0.5	[26, 77]	301.91	455.20	Optimal	2	2
10	10	0.75	[26, 77]	569.28	455.35	Optimal	2	2
10	10	0.99	[27]	1483.42	376.84	Optimal	2	2
10	20	0.5	[14, 41, 77]	463.21	341.29	Optimal	1	1
10	20	0.75	[26, 77]	751.81	459.20	Optimal	2	2
10	20	0.99	[26, 77]	1705.47	671.86	Optimal	3	3
10	40	0.5	[4, 14, 29, 37, 72, 87]	686.86	720.75	Optimal	1	1
10	40	0.75	[11, 29, 50, 87]	1054.76	870.02	Optimal	2	2
10	40	0.99	[64, 92, 27, 4]	2054.05	7287.83	Gap 0.03%	12	20
20	10	0.5	[26, 77]	330.42	454.79	Optimal	2	2
20	10	0.75	[27]	719.71	381.69	Optimal	2	2
20	10	0.99	[27]	2504.32	377.36	Optimal	2	2
20	20	0.5	[14, 41, 77]	500.50	342.60	Optimal	1	1
20	20	0.75	[26, 77]	941.61	462.40	Optimal	2	2
20	20	0.99	[48]	2765.23	379.25	Optimal	2	2
20	40	0.5	[4, 14, 29, 37, 72, 87]	743.43	726.93	Optimal	1	1
20	40	0.75	[14, 50, 77]	1285.17	659.89	Optimal	2	2
20	40	0.99	[12, 69]	3159.26	3552.17	Optimal	8	14
40	10	0.5	[26, 77]	369.60	455.50	Optimal	2	2
40	10	0.75	[27]	951.16	379.55	Optimal	2	2
40	10	0.99	[27]	4451.13	377.10	Optimal	2	2
40	20	0.5	[14, 41, 77]	550.37	340.82	Optimal	1	1
40	20	0.75	[48]	1212.08	379.44	Optimal	2	2
40	20	0.99	[48]	4712.05	380.55	Optimal	2	2
40	40	0.5	[64, 36, 87, 29, 13]	817.99	1116.19	Optimal	1	2
40	40	0.75	[12, 77]	1602.75	243.44	Optimal	1	1
40	40	0.99	[12, 78]	5210.07	1979.01	Optimal	6	9
200	10	0.5	[26, 77]	521.07	457.50	Optimal	2	2
200	10	0.75	[27]	2289.24	376.58	Optimal	2	2
200	10	0.99	[27]	19821.73	378.09	Optimal	2	2
200	20	0.5	[26, 77]	703.60	462.98	Optimal	2	2
200	20	0.75	[48]	2550.16	378.29	Optimal	2	2
200	20	0.99	[48]	20082.64	812.72	Optimal	4	4
200	40	0.5	[14, 50, 77]	1036.89	663.20	Optimal	2	2
200	40	0.75	[48]	3060.63	384.94	Optimal	2	2
200	40	0.99	[48]	20593.12	6606.29	Optimal	30	30

TABLE 4. REFERENCE TABLE OF FSA TO NUMERICAL IDS. * INDICATE THE FSA CORRESPONDING TO CANDIDATE FACILITY LOCATIONS CONSIDERED FOR THE CASE OF $n = 24, 96$, AND **INDICATE THE FSA CORRESPONDING TO CANDIDATE FACILITY LOCATIONS CONSIDERED FOR ALL THREE CASE OF $n = 10, 24$, AND 96.

FSA	ID	FSA	ID	FSA	ID	FSA	ID	FSA	ID	FSA	ID
M1B	1	M1X	17	M3M	33	M4V*	49	M5T	65	M6S	81
M1C	2	M2H	18	M3N	34	M4W	50	M5V*	66	M8V	82
M1E*	3	M2J	19	M4A	35	M4X**	51	M5W	67	M8W	83
M1G	4	M2K**	20	M4B	36	M4Y	52	M6A	68	M8X	84
M1H	5	M2L	21	M4C	37	M5A	53	M6B**	69	M8Y	85
M1J	6	M2M	22	M4E	38	M5B	54	M6C	70	M8Z**	86
M1K*	7	M2N*	23	M4G**	39	M5C	55	M6E	71	M9A**	87
M1L	8	M2P	24	M4H	40	M5E	56	M6G*	72	M9B	88
M1M	9	M2R	25	M4J*	41	M5G	57	M6H	73	M9C	89
M1N	10	M3A	26	M4L	42	M5H	58	M6J	74	M9L	90
M1P	11	M3B*	27	M4M*	43	M5J	59	M6K	75	M9M*	91
M1R**	12	M3C	28	M4N	44	M5M*	60	M6L	76	M9N	92
M1S	13	M3H	29	M4P	45	M5N	61	M6M	77	M9P	93
M1T	14	M3J	30	M4R	46	M5P**	62	M6N*	78	M9R*	94
M1V*	15	M3K**	31	M4S	47	M5R	63	M6P	79	M9V	95
M1W	16	M3L	32	M4T	48	M5S**	64	M6R	80	M9W	96

VII. REFERENCES

Aboolian, Robert, Oded Berman, and Zvi Drezner. 2008. "Location and Allocation of Service Units on a Congested Network." *IIE Transactions* 40 (4): 422–33.

———. 2009. "The Multiple Server Center Location Problem." *Annals of Operations Research* 167 (1): 337–52.

Aboolian, Robert, Oded Berman, and Dmitry Krass. 2012. "Profit Maximizing Distributed Service System Design with Congestion and Elastic Demand." *Transportation Science* 46 (2): 247–61.

Aboolian, Robert, Oded Berman, and Vedat Verter. 2016. "Maximal Accessibility Network Design in the Public Sector." *Transportation Science* 50 (1): 336–47.

Aboolian, Robert, Tingting Cui, and Zuo-Jun Max Shen. 2013. “An Efficient Approach for Solving Reliable Facility Location Models.” *INFORMS Journal on Computing* 25 (4): 720–29.

Amiri, Ali. 1997. “Solution Procedures for the Service System Design Problem.” *Computers & Operations Research* 24 (1): 49–60.

Baron, Opher, Oded Berman, and Dmitry Krass. 2008. “Facility Location with Stochastic Demand and Constraints on Waiting Time.” *Manufacturing & Service Operations Management* 10 (3): 484–505.

Berman, Oded, and Zvi Drezner. 2007. “The Multiple Server Location Problem.” *Journal of the Operational Research Society* 58 (1): 91–99.

Berman, Oded, and Dmitry Krass. 2020. “Stochastic Location Models with Congestion.” In *Location Science*, edited by Gilbert Laporte, Stefan Nickel, and Francisco Saldanha-da-Gama, 2nd ed., 477–535. Springer Cham Heidelberg New York Dordrecht London.

Berman, Oded, Dmitry Krass, and Jiamin Wang. 2011. “Stochastic Analysis in Location Research.” In *Foundations of Location Analysis*, 241–71. Springer.

Cai, Ximing, Daene C McKinney, Leon S Lasdon, and David W Watkins Jr. 2001. “Solving Large Nonconvex Water Resources Management Models Using Generalized Benders Decomposition.” *Operations Research* 49 (2): 235–45.

Castillo, Ignacio, Armann Ingolfsson, and Thaddeus Sim. 2009. “Social Optimal Location of Facilities with Fixed Servers, Stochastic Demand, and Congestion.” *Production and Operations Management* 18 (6): 721–36.

Dogan, Kerim, Mumtaz Karatas, and Ertan Yakici. 2020. “A Model for Locating Preventive Health Care Facilities.” *Central European Journal of Operations Research* 28 (3): 1091–1121.

Elhedhli, Samir. 2006. “Service System Design with Immobile Servers, Stochastic Demand, and Congestion.” *Manufacturing & Service Operations Management* 8 (1): 92–97.

Elhedhli, Samir, Yan Wang, and Ahmed Saif. 2018. “Service System Design with Immobile Servers, Stochastic Demand and Concave-Cost Capacity Selection.” *Computers & Operations Research* 94: 65–75.

Floudas, CA, A Aggarwal, and AR Ciric. 1989. “Global Optimum Search for Nonconvex Nlp and Minlp Problems.” *Computers & Chemical Engineering* 13 (10): 1117–32.

Geoffrion, Arthur M. 1972. “Generalized Benders Decomposition.” *Journal of Optimization Theory and Applications* 10 (4): 237–60.

Marianov, Vladimir, and Miguel Ríos. 2000. “A Probabilistic Quality of Service Constraint for a Location Model of Switches in Atm Communications Networks.” *Annals of Operations Research* 96 (1-4): 237–43.

Marianov, Vladimir, and Daniel Serra. 1998. “Probabilistic, Maximal Covering Location—Allocation Models Forcongested Systems.” *Journal of Regional Science* 38 (3): 401–24.

Statistics Canada. 2016. “Data Products, 2016 Census.” January 2016. https://www12.statcan.gc.ca/census-recensement/2016/dp-pd/index-eng.cfm.

Toronto Real State Board. 2019. “Commercial Realty Watch, First Quarter 2019.” May 2019. http://www.trebhome.com/index.php/market-news/commercial-report/commercial-report-archive.

Vidyarthi, Navneet, and Sachin Jayaswal. 2014. “Efficient Solution of a Class of Location–Allocation Problems with Stochastic Demand and Congestion.” *Computers & Operations Research* 48: 20–30.

Vidyarthi, Navneet, and Onur Kuzgunkaya. 2015. “The Impact of Directed Choice on the Design of Preventive Healthcare Facility Network Under Congestion.” *Health Care Management Science* 18 (4): 459–74.

Wang, Qian, Rajan Batta, and Christopher M Rump. 2002. “Algorithms for a Facility Location Problem with Stochastic Customer Demand and Immobile Servers.” *Annals of Operations Research* 111 (1-4): 17–34.

Zhang, Yue, Oded Berman, Patrice Marcotte, and Vedat Verter. 2010. “A Bilevel Model for Preventive Healthcare Facility Network Design with Congestion.” *IIE Transactions* 42 (12): 865–80.

Zhang, Yue, Oded Berman, and Vedat Verter. 2009. “Incorporating Congestion in Preventive Healthcare Facility Network Design.” *European Journal of Operational Research* 198 (3): 922–35.