An Economies-of-Scale Service System Design Problem

Pooya Hoseinpour

Department of Industrial Engineering & Management Systems, Amirkabir University of Technology
(Tehran Polytechnic), Iran
p.hoseinpour@aut.ac.ir

Abstract

This paper studies the design of a service system in the presence of economies-of-scale. The goal is to make decision on the number, location, service capacities of facilities as well as on the allocation of customers to the opened facilities in order to minimize the cost of whole system. The total cost includes opening and serving costs of facilities aggregated to the transportation and waiting costs of customers. To reflect the economies-of-scale in the modeling, a general opening cost function is supposed for each service facility with the characteristic of being concave and non-decreasing on its service capacity. A Lagrangian relaxation algorithm is developed for solving the problem in its general form. The algorithm decomposes the relaxed model into some homogeneous subproblems where each one can be optimally solved in polynomial time with no need for any optimization solver. Our computational experiment shows that the developed algorithm is both efficient and effective in solving the proposed problem.

Keywords: Economies-of-scale; congested facility location; M/M/1 queue; Service system; Lagrangian relaxation; Mixed integer nonlinear program.
1 Introduction

A service system is primarily consisting of two key components, namely, service facilities and customers, where the customers create demands for getting the service offered by service facilities. Both of the demand levels and the service times are stochastic by nature which inevitably results in congestion at service facilities. Congestion in a service facility can be seen as a key factor that has a severe impact on service quality. A service system design (SSD) problem tries to balance the service providing cost and service offering quality. This is one of the interesting problems in location science and also known as stochastic facility location problems with congestion. For a comprehensive review of this problem, the interested reader is referred to the surveys Boffey et al. (2007) and Berman and Krass (2015).

In a service system, the service facilities can be either immobile or mobile; in the first case, the customers are supposed to travel to the locations of service facilities for getting service; however, the service facilities travel to the customers’ locations in the second case. This work explores designing an immobile service system, where some potential locations are predetermined for the service facilities. One needs to make decision on both of the numbers and the location of service facilities. Moreover, the service capacity of each opened service facility is often assumed to be optimally decided. There are two approaches for incorporating this decision in the modeling; some works simply assume a finite set of capacity levels for service provision (see e.g., Elhedhli, 2006; Vidyarthi and Jayaswal, 2014; Aboolian et al., 2012; Ahmadi-Javid and Hoseinpour, 2017, 2018; Hoseinpour and Ahmadi-Javid, 2019), the others model it as a continuous real-valued non-negative decision variable (see e.g., Wang et al., 2004; Castillo et al., 2009; Hoseinpour and Ahmadi-Javid, 2016; Elhedhli et al., 2018; Ahmadi-Javid et al., 2018). We apply the second approach here in our modeling and will find a closed-form solution for the service capacity of each service facility in a general setting, which would directly be a function of the customers’ demands who are allocated to the service facility.

The allocation of customers to the opened service facilities are assumed to be either user-choice or direct-choice. In a user-choice allocation, each customer selects a service
facility for getting service, with the goal of maximizing her utility (see survey Berman and Krass, 2015); however, in a direct-choice allocation, the customers behave in a way that the allocations be optimal for the whole system, called socially optimal solution. In other words, the goal in the socially-optimal decision making, to which our model belongs, is to minimize the cost of the whole system which is an aggregated function of the cost terms originated from both components, i.e., service facilities and customers. From the facilities’ side, typically, facility opening and service costs are considered; whereas, accessing and congestion costs are added from the customers’ side. One can see that the first one shows the system’s owner cost and the second one reflects the service quality issue. See e.g., Wang et al. (2002, 2004); Elhedhli (2006); Aboolian et al. (2008); Berman and Drezner (2007); Kim (2013); Vidyarthi and Jayaswal (2014); Hoseinpour and Ahmadi-Javid (2016); Ahmadi-Javid and Hoseinpour (2017, 2018); Elhedhli et al. (2018); Ahmadi-Javid et al. (2018); Hoseinpour and Ahmadi-Javid (2019) for recent works who apply this approach.

There is also another approach for considering the service quality, where a threshold is taken for some measure of quality, such as the average number of customers waiting in the queue. See for example the recent works Rajagopalan and Yu (2001); Silva and Serra (2008); Baron et al. (2008); Aboolian et al. (2012).

In order to precisely decide the optimal number and service capacity of service facilities, it is highly suggested to consider the advantages of economies-of-scale in the modeling. This has been extensively considered in the classical facility location models (see e.g., Dasci and Verter, 2001; Dupont, 2008; Baumgartner et al., 2012; Saif and Elhedhli, 2016), but is rare in SSD problems. There is strong evidence in the reality for the existence of economies-of-scale in service industries; for example in the banking industry (Doukas and Switzer, 1991), and healthcare (Preyra and Pink, 2006).

The benefits of economies-of-scale have been previously considered in the design of a service system, where discrete service capacity levels are assumed for each service facility. Technically, considering economies-of-scale would be much easier if some service capacity levels are predefined since only some modifications in the parameters are needed to meet this assumption. Whereas, it makes the structure of the proposed models to
be too complicated when the service capacity of each service facility is a continuous variable. Because of this, all previous works who assumed continuous service capacity ignored the benefits of considering economies-of-scale and just modeled the opening cost at each service facility as a linear function of service capacity. The only exception is Elhedhli et al. (2018) who defines \( h(\mu) := \sqrt{\mu} \) as an economies-of-scale opening cost of the service facility that works with capacity \( \mu \). Our work extends their work and considers a general \( h(\mu) \) function, with characteristics of being concave and non-decreasing. The modeling approach is completely novel which enables us to easily handle the complexity of considering economies-of-scale service opening costs. For doing so, the opening cost function is first approximated with its piece-wise linearization and then a novel solution algorithm based on Lagrangian relation is constructed where the relaxed model can be decomposed to smaller subproblems. A polynomial-time exact algorithm is developed for optimally solving the relaxed submodels. Using the proposed algorithm for solving the submodels, the optimal solution of the model can be found directly with no need for any optimization solver, such as CPLEX.

The remainder of the paper is organized as follows. Section 2 mathematically states the problem which can be solved using a novel Lagrangian relaxation algorithm, which is developed in Section 3. Section 4 reports the computational results of considering different opening functions. Finally, Section 5 concludes the main findings with some directions for future research.

\section{Problem Modeling}

This section mathematically models the problem, i.e., the economies-of-scale SSD problem. The goal is to find the optimal value of decision variables to minimize the whole cost of the service system. Assume \( I \) shows the set of potential service facility locations and \( J \) be the set of all customers. Let \( h_i(\mu_i) \), the economies-of-scale opening cost function for \( i \in I \), is defined as

\[ h_i(\mu_i) := f_i + c_i g_i(\mu_i), \]  

\[ (1) \]
where $f_i$ is the fixed cost of opening service facility $i \in I$ per time unit, $c_i$ is the operating cost in service facility $i \in I$ per time unit, and $g_i(.)$ is an arbitrary concave non-decreasing function. Let $s_i$ denotes the service cost per time unit at service facility $i \in I$ and $w_i$ shows the waiting cost of each customer in service facility $i \in I$ per time unit. $a_{ij}$ shows the access cost of customer $j \in J$ per time unit if she is assigned to the service facility $i \in I$. Finally, $\lambda_j$ denotes the demand rate of customer $j \in J$. The goal is to find the optimal value of the following decision variables:

- $\mu_i$ is a non-negative decision variable shows the service rate in service facility $i \in I$,
- $x_i \in \{0,1\}$ is a binary decision variable that is one if service facility at location $i \in I$ is opened to provide service,
- $y_{ij} \in \{0,1\}$ is one if customer $j \in J$ is assigned to service facility $i \in I$.

The queue of each opened service facility $i \in I$ is assumed to be an M/M/1 queue system and the arrival rate of customers is shown by $\Lambda_i$, which is equal to

$$\Lambda_i = \sum_j \lambda_j y_{ij}, \quad i \in I, \quad (2)$$

and serving each customer, independently, takes a random time that follows from an exponential distribution with the expected value of $1/\mu_i$, and the discipline of the queue is FCFS, first come first severed. Although we limit this work to M/M/1 queue in each service facility, which is widely used in the literature, the proposed methodology can easily be applied to SSD problems with other queue models in the service facilities.

**Proposition 1.** (Shortle et al., 2018) The steady-state condition in the M/M/1 queue system of service facility $i \in I$ is $\rho_i < 1$, where $\rho_i := \Lambda_i/\mu_i$ and $\Lambda_i$ is defined by Eq.(2).

**Proposition 2.** (Shortle et al., 2018) The average waiting time of each customer in the M/M/1 queue system of service facility $i \in I$ is $W_i^q = \frac{1}{\mu_i - \Lambda_i}$, where $\Lambda_i$ is defined by Eq.(2).

The mathematical program for our discussed problem, the economies-of-scale SSD, can be proposed as follows:

(EoS-SSD)
\[
\text{minimize} \quad \sum_{i \in I} \left\{ h_i(\mu_i)x_i + \sum_{j \in J} s_i \lambda_j y_{ij} + \sum_{j \in J} a_{ij} \lambda_j y_{ij} + w_i \Lambda_i W_i^q \right\} 
\]

subject to

\[
\begin{align*}
\sum_{i \in I} y_{ij} & \leq x_i \quad i \in I, \ j \in J \\
\sum_{i \in I} y_{ij} & = 1 \quad j \in J \\
\mu_i - \Lambda_i & \geq 0 \quad i \in I \\
\mu_i & \geq 0 \quad i \in I \\
x_i & \in \{0, 1\} \quad i \in I \\
y_{ij} & \in \{0, 1\} \quad i \in I, \ j \in J,
\end{align*}
\]

where the first term in the objective function (3a) is the total opening cost of service facilities in the service system and the function \(h_i(.)\) is defined by Eq.(1). The second term calculates the total serving cost of customers, the third term is customers accessing cost, and the last term is customer waiting cost. In the last term of the objective function, the total value of customers waiting cost can be calculated multiplying \(\Lambda_i\), from Eq.(2), by average waiting cost of each customer calculated in Proposition 2. One can say that the first two terms are facility-side cost, and the last two terms reflect customer’s service quality; by aggregating them the model seeks the socially optimal solution. Constraint set (3b) indicates that there is no assignment for the non-opened service facility. Constraint set (3c) ensures that each customer has been assigned to exactly one service facility for placing her demand. Constraint set (3d) is the steady-state condition obtained in Proposition 1. Constraint set (3e) shows that the service rate of each service facility should be a non-negative real value. Finally, Constraint sets (3f) and (3g) show binary location-allocation decision variables.

The EoS-SSD model, i.e., Model (3), is too complex to be solved for an arbitrary opening cost function. In order to deal with this, we first approximate \(g_i(\mu)\), for \(i \in I\), by its piece-wise linear function \(\hat{g}_i(\mu_i)\), which can be defined as

\[
\hat{g}_i(\mu_i) := \min_{k \in K} \{ \hat{g}_i^k(\mu_i) \},
\]

where \(\hat{g}_i^k(\mu_i)\) is the linear function that is tangent to \(g_i(\mu_i)\) in point \(\mu^k\) and can be defined
Figure 1: piece-wise linearization of function $g_i(\mu_i)$, $i \in I$, which is a general concave non-decreasing function as

$$\hat{g}_i^k(\mu_i) := g_i(\mu_k) + g_i'(\mu_k)(\mu_i - \mu_k),$$

for $k \in K$. See Figure 1 where function $\hat{g}_i(\mu_i)$ is approximated by $\min_{k \in K} \{\hat{g}_i^k(\mu_i)\}$.

Algorithm 1.

Step 1. Set $k \leftarrow 1$, and $b_0 = \nu$.

Step 2. Knowing $b_{k-1}$, find $\mu_k$ by solving following equation

$$(1 + \epsilon)g_i(b_{k-1}) = g_i(\mu_k) + g_i'(\mu_k)(b_{k-1} - \mu_k),$$

where breakpoint $b_{k-1}$ satisfies in line $\hat{g}_i^k(\mu_i)$ and $\hat{g}_i^k(b_{k-1}) = (1 + \epsilon)g_i(b_{k-1})$.

Step 3. Knowing $\mu_k$, find $b_k$ by solving following equation

$$(1 + \epsilon)g_i(b_k) = g_i(\mu_k) + g_i'(\mu_k)(b_k - \mu_k),$$

where breakpoint $b_k$ satisfies in line $\hat{g}_i^k(\mu_i)$ and $\hat{g}_i^k(b_k) = (1 + \epsilon)g_i(b_k)$.

Step 4. If $b_k \leq \bar{\nu}$ set $k \leftarrow k + 1$ and go Step 2.
One can find the finite set of breakpoints $b_k$, and tangent points $\mu_k$, for $k \in K$, such that $0 \leq \frac{g_i(\mu_i) - g_i(b_k)}{g_i(b_k)} \leq \epsilon$, for an arbitrary error threshold of $\epsilon > 0$. For this purpose, Algorithm 1 is developed inspiring from Elhedhli (2005). In Algorithm 1, Steps 2 and 3 iteratively find tangent points and breakpoints, respectively. Moreover, $\upsilon$ and $\bar{\upsilon}$ are determined numerically considering the type of function $g_i(.)$.

Substituting (4) in the objective function (3a) results

$$\min_{x_i, \mu_i, y_{ij}} \sum_{i \in I} \min_{k \in K} \{v^{i,k}(x_i, \mu_i, y_{ij})\},$$

where

$$v^{i,k}(x_i, \mu_i, y_{ij}) := \left(f_i + c_i g_i(\mu_k) - c_i \mu_k^i g_i'(\mu_k)\right) x_i + c_i g_i'(\mu_k) \mu_i$$

$$+ \sum_{j \in J} s_i \lambda_j y_{ij} + \sum_{j \in J} a_{ij} \lambda_j y_{ij} + w_i \mu_i - \sum_{j \in J} \lambda_j y_{ij},$$

(6)

for $i \in I$, and $k \in K$. In next section, we will develop a novel Lagrangian relaxation algorithm to solve Model (5), (3b)-(3g) efficiently.

### 3 The Lagrangian Relaxation Algorithm

In this section, a Lagrangian relaxation algorithm is developed to efficiently solve the proposed Model (5), (3b)-(3g). An interested reader is referred to, e.g., Fisher (2004); Geoffrion (2010) for more details on Lagrangian relaxation method. In abstract, in a Lagrangian relaxation algorithm, first the hard constraints, by relaxing which the model will be much easier to solve, are determined and dualized by Lagrangian multipliers. For EoS-SSD model, the hard constraints (3c) are relaxed with Lagrangian multipliers $u_j, j \in J$ and aggregated to the objective function, (3a), which yields the relaxed Model R-EoS-SSD as follows:

(R-EoS-SSD)
\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in I} \left\{ \min_{k \in K} \left\{ v^{i,k}(x_i, \mu_i, y_{ij}) \right\} \right\} + \sum_{j \in J} u_j \left( 1 - \sum_{i \in I} y_{ij} \right) \\
\text{subject to} & \quad (3b), (3d) - (3g),
\end{align*}
\]

where \( v^{i,k}(x_i, \mu_i, y_{ij}) \) is defined by (6).

For any values of Lagrangian multipliers \( u_j \), for \( j \in J \), the optimal value of relaxed problem R-EoS-SSD provides a lower bound for the optimal value of the original problem EoS-SSD. The goal is to find the tightest Lagrangian lower bound. For doing so, the Lagrangian dual problem is defined as

\[
(D-EoS-SSD) \quad \text{maximum} \quad L(u) \quad \text{subject to} \quad u \in \mathbb{R}^{|J|},
\]

where \( L(u) \) is the optimal objective function value of R-EoS-SSD, i.e., (7a).

This problem is normally solved using the well-known subgradient algorithm, which is an iterative algorithm. Let \( u^t \) be the values of Lagrangian multipliers in iteration \( t \). The subgradient algorithm starts by setting initial values for the Lagrangian multipliers, shown as \( u^0 \), and updates them iteratively. In each iteration \( t \), the relaxed problem is optimally solved and provides a lower bound solution \( Lb^t \). As the solution might not be feasible to the original problem, a heuristic algorithm is then used to construct a feasible solution, which will be an upper bound solution for the original problem, shown as \( Ub^t \).

The Lagrangian multipliers for next iteration will be updated as follow:

\[
u^{t+1} \leftarrow u^t + \frac{\alpha^t \left( \min_{s=0,\ldots,t} \{ Ub^s \} - Lb^t \right)}{||v^t||} v^t,
\]

where \( v^t_j = 1 - \sum_{i \in I} y_{ij} \) for \( j \in J \). \( \alpha^t \) is a control parameter starts with initial \( \alpha^t \in (0, 2) \) and is reducing when there is no improvements for long time; see Geoffrion (2010). The algorithm stops when the lower bound and upper bound solutions are close enough to each other, i.e.,

\[
\min_{s=0,\ldots,t} \{ Ub^s \} - Lb^t \leq \epsilon, \quad (10)
\]
where \( e > 0 \) is a user-defined error bound.

In the following, Subsection 3.1 provides a methodology to polynomially solve the R-EoS-SSD in each iteration of Lagrangian relaxation algorithm. Subsection 3.2 presents a heuristic algorithm that constructs a feasible solution from the solution obtained by solving R-EoS-SSD.

3.1 Lower bound Solution

Let set the values of Lagrangian multipliers as \( u^t \) in iteration \( t \). The relaxed Model (7) is separable to \(|I|\) subproblem, called R-EoS-SSD\((i),t\), as follows:

\[
\begin{align*}
\text{minimize} & \quad \min_{\mu_i, y_{ij}} \left\{ v^{i,k}(1, \mu_i, y_{ij}) - \sum_{j \in J} u^t_j y_{ij} \right\} \\
\text{subject to} & \quad \mu_i - \sum_{j \in J} \lambda_j y_{ij} \geq 0 \\
& \quad \mu_i \geq 0 \\
& \quad y_{ij} \in \{0, 1\} \quad j \in J.
\end{align*}
\] (11a)

After solving R-EoS-SSD\((i),t\), for \( i \in I \), if the objective value is negative, then set \( x_i = 1 \) and open the service facility \( i \in I \); otherwise set \( x_i = 0 \) and close the service facility with objective value of zero. Theorem 1 finds the optimal solution of problem R-EoS-SSD\((i),t\), for \( i \in I \), in polynomial time. See Proposition 3 for more detail.

**Theorem 1.** The optimal solution of problem R-EoS-SSD\((i),t\), for \( i \in I \) in iteration \( t \), is

\[
\begin{align*}
y_{ij}^* &= \arg\min \{ z^{i,k(i)} \} \\
\mu_i^* &= \sum_{j \in J} \lambda_j y_{ij}^* + \sqrt{w_i \sum_{j \in J} \lambda_j y_{ij}^* c_i g^i(\mu_{k(i)})},
\end{align*}
\]

where \( k(i) = \arg\min_{k \in K} \{ z^{i,k} \} \) and

\[
z^{i,k} - p^{i,k} = \min_{y_j} \left( \sum_{j \in J} \left( q^{i,t,k}_j - u^{t}_j \right) y_j \right) + \sqrt{\sum_{j \in J} \left( r^{i,t,k}_j \right) y_j} \quad (12a)
\]
subject to \( y_j \in \{0, 1\} \quad j \in J \), \( (12b) \)

for \( i \in I \) and \( k \in K \), where

\[
\begin{align*}
p^{i,k} &= f_i + c_i g_i(\mu^k) - c_i g'_i(\mu^k)\mu^k \\
r^{i,k}_j &= 4w_i c_i g'_i(\mu^k)\lambda_j \\
q^{i,k}_j &= s_i \lambda_j + a_{ij} \lambda_j + c_i g'_i(\mu^k)\lambda_j.
\end{align*}
\]

**Proof.** Proof of Theorem 1. By changing the order of minimizations in problem R-EoS-SSD\(^{(i),t}\), i.e., (11), its optimal solution can be found by solving \(|K|\) similar problems as follows:

\[
\begin{align*}
\text{minimize}_{\mu, y_{ij}} & \quad v^{i,k}(x_i, \mu_i, y_{ij}) - \sum_{j \in J} u^t_j y_{ij} \quad (13a) \\
\text{subject to} & \quad (11b) - (11d),
\end{align*}
\]

and choosing the optimal solution of the one with minimum objective value. In order to solve (13) one can first find the optimal value of \( \mu_i \) by considering the following problem:

\[
\begin{align*}
\text{minimize}_{\mu_i} & \quad c_i g'_i(\mu^k)\mu_i + w_i \frac{\Lambda_i}{\mu_i - \Lambda_i} \quad (14a) \\
\text{subject to} & \quad \mu_i \geq \Lambda_i. \quad (14b)
\end{align*}
\]

The program (14) is convex respect to \( \mu_i \); therefore, by setting its differentiation equal zero, one can find the optimal value of service capacity as follows

\[
\mu^*_i = \Lambda_i + \sqrt{\frac{w_i \Lambda_i}{c_i g'_i(\mu^k)}}. \quad (15)
\]

which clearly satisfies Constraint (14b). Substituting the optimal value of service capacity (15) into objective function (13a) which completes the proof. \( \square \)

Model (12), defined in Theorem 1, is a pure integer program and one can optimally solve it using the proposed Algorithm 2 in the following.
Algorithm 2.

Step 1. For each \( j \in J \), if \( q_{j}^{i,t,k} - u_{j}^{t} \geq 0 \), put \( j \) in set \( J_{1} \) and if \( q_{j}^{i,t,k} - u_{j}^{t} < 0 \) and \( r_{j}^{i,k} = 0 \), put it in set \( J_{2} \); otherwise put it in set \( J_{3} \).

Step 2. For customer \( j \in J_{1} \), set \( y_{ij} = 0 \) and for customer \( j \in J_{2} \) set \( y_{ij} = 1 \).

Step 3. Sort customers \( j \in J_{3} \) in non-decreasing order of \( \{ q_{j}^{i,t,k} - u_{j}^{t} r_{j}^{i,k} \} \).

Step 4. Define \( J_{3}^{(m)} \subseteq J_{3} \) such that contain first \( m \) customers in \( J_{3} \), and calculate

\[
Q_{m} := \sum_{j \in J_{3}^{(m)}} \left( q_{j}^{i,t,k} - u_{j}^{t} \right) + \sqrt{\sum_{j \in J_{3}^{(m)}} r_{j}^{i,k}}
\]

for \( m \in J_{3} \). Find \( m^{*} := \arg\min_{m \in J_{3}} Q_{m} \).

Step 5. For \( j \in J_{3} \) that \( j \leq m^{*} \) set \( y_{ij} = 1 \); for \( j \in J_{3} \) that \( j > m^{*} \) set \( y_{ij} = 0 \).

Algorithm 2 is driven directly from Theorem 2.

Theorem 2. (Daskin et al., 2002) Consider problem

\[
\text{minimize} \quad h(y) = \sum_{j \in J} d_{j} y_{j} + \sqrt{\sum_{j \in J} b_{j} y_{j}}
\]

\text{subject to} \quad y_{j} \in \{0, 1\} \quad j \in J

with \( d_{j} \in \mathbb{R}, b_{j} \in \mathbb{R}^{+} \) for \( j \in J \).

For each \( j \in J \), if \( d_{j} \geq 0 \), then \( y_{j} = 0 \); if \( d_{j} < 0 \) and \( b_{j} = 0 \), then \( y_{j} = 1 \); otherwise for ordered \( j \) as

\[
\left\{ \frac{d_{1}}{b_{1}} \right\} \leq \cdots \leq \left\{ \frac{d_{|J|}}{b_{|J|}} \right\},
\]

in set \( \hat{J} \subseteq J \), if \( y_{s} = 1 \), then \( y_{l} = 1 \) for all \( l \leq s \).

Proposition 3. The complexity of Algorithm 2 is \( O(|I| \times |K| \times |J| \log |J|) \).

Proof. Proof of Proposition 3. The complexity of algorithm for finding minimum of \( n \) numbers is \( O(n) \) and the complexity of an efficient sorting algorithm of \( n \) numbers is \( O(n \log n) \).
Based on Proposition (3), one can conclude that Theorem (1) finds the optimal solution of problem \( R\text{-EoS-SSD}^{(i)} \) in low-order polynomial time.

### 3.2 Upper bound Solution

The solution provided by the relaxed problem, i.e, R-EoS-SSD, might be infeasible to the original problem. For example, some customers may be assigned to more than one service facility and some customers may not be assigned to any opened service facility. To construct a feasible solution from the solution obtained from the relaxed problem, one can use the following simple heuristic algorithm:

---

**Algorithm 3.**

**Step 1.** Open service facilities same as the values of \( x_i \), for \( i \in I \), obtained by solving the relaxed model R-EoS-SSD.

**Step 2.** Do following substeps:

**Step 2.1.** For each customer \( j \in J \) who is assigned to at least one service facility, based on the values of \( y_{ij} \) obtained from solving the relaxed model R-EoS-SSD, select service facility \( i \) such that \( y_{ij} = 1 \) and has minimum value of \( \left\{ q_{i,k}^{j} \right\} \).

**Step 2.2.** For each customer \( j \in J \) who is assigned to no service facility, based on the values of \( y_{ij} \) obtained from solving the relaxed model R-EoS-SSD, select service facility \( i \) such that \( x_i = 1 \) and has minimum value of \( \left\{ q_{i,k}^{j} \right\} \).

**Step 3.** Close an open service facility if no customer is assigned to it in Step 2.

---

In Step 1, as the same service facilities as the ones opened in lower bound solution with the same determined capacity are selected to be opened. Step 2 assigns the customers to the opened service facilities. For this purpose, first, for the customers who are assigned to at least one service facility based on the lower bound solution, the service facility with
less increase in the objective function is determined in Step 2.1. For the customers who have already no assignments, the service facility among all opened service facilities is selected which increases in the objective function less. Finally in Step 3 if there is an opened service facility with no assignments, it will be closed.

4 Numerical Testing

The proposed Lagrangian relaxation algorithm in Section 3 has been implemented in C++ to solve the SSD problem with no need for any commercial solver. The tests are done on a machine with an Intel Core i5 1.8GHz CPU and 8GB RAM. The same test problems as in Elhedhli (2006) are picked and modified. The test problems are initially proposed by Holmberg et al. (1999) for a class of facility location problems and later some of them are later modified and used by Elhedhli (2006) for designing an SSD problem. In our problem, the service capacities of facilities are decision variables. Thus, the test problems whose parameters are similar to others, with the exception of their facility capacities, are excluded. 27 test problems are finally selected for the current experiment where the number of service facilities and customers varies from 10 to 30 and 50 to 150, respectively. The parameters of the test problems are modified as follows. The serving cost of a customer in service facilities, $s_i$, $i \in I$, are randomly generated from the interval $[1, 5]$. The waiting cost of each customer per day, $w_i$, $i \in I$, is randomly generated from the interval $[50, 300]$. The fixed cost of opening service facility, $f_i$, $i \in I$, the demand rates of customers, i.e., $\lambda_j$, $j \in J$, and the accessing cost of customers to their service facilities, i.e., $a_{ij}$, $i \in I$, $j \in J$, are exactly collected from the original test problems.

The proposed solution method enables us to consider any arbitrary function with the general setting of being concave and non-decreasing for defining the opening cost of service facilities, i.e., $h_i(\mu_i)$, $i \in I$. Here, three different functions have been defined for opening cost of service facilities; namely, linear, square-root, and fractional functions. The first one, the linear function is defined as

$$h_i(\mu_i) = f_i + c_i \mu_i \quad i \in I,$$
which was traditionally used for mathematically showing opening cost of service facilities; see for examples Castillo et al. (2009); Hoseinpour and Ahmadi-Javid (2016); Ahmadi-Javid et al. (2018). The second one, the square-root function, is defined as

\[ h_i(\mu_i) = f_i + c_i \sqrt{\mu_i} \quad i \in I. \]

This function has been recently considered by Elhedhli et al. (2018). However, their proposed approach for modeling can only handle square function. The third one is defined here as a concave fractional function as follows:

\[ h_i(\mu_i) = f_i + \frac{c_i \mu_i}{\mu_i + 1} \quad i \in I. \]

Here, we fix the operating cost of each service facility, i.e., \( c_i \) as 1, 10, and 100 for linear, square-root, and fractional opening costs, respectively.

Using Algorithm 1 when \( g(\mu_i) = \sqrt{\mu_i} \), one can use

\[ \mu_k \leftarrow \left( (1 + \epsilon) \sqrt{b_{k-1}} + \sqrt{(\epsilon^2 + 2\epsilon) b_{k-1}} \right)^2 \]

to update \( \mu_k \) in Step 2 and use

\[ b_k \leftarrow \left( (1 + \epsilon) \mu_k + \sqrt{(\epsilon^2 + 2\epsilon) \mu_k} \right)^2 \]

to find \( b_k \) in Step 3. When \( g(\mu_i) = \frac{\mu_i}{\mu_i + 1} \), one can use

\[ \mu_k \leftarrow \frac{(1 + b) \sqrt{eb_{k-1}} + (1 + \epsilon) b_{k-1}}{1 - \epsilon b_{k-1}} \]

to update \( \mu_k \) in Step 2 and use

\[ b_k \leftarrow \frac{1}{2} \left( 2\mu_k + \epsilon(1 + \mu_k)^2 + (1 + \mu_k) \sqrt{\epsilon^2(1 + \mu_k)^2 + 4\epsilon \mu_k} \right) \]

to find \( b_k \) in Step 3.

All instances are solved using the developed Lagrangian relaxation algorithm with user-defined tolerance of \( \epsilon = 0.01 \) and maximum 10000 iterations of the running algorithm. The algorithm will stop when at least one of these criteria be satisfied. Tables 1-3 respectively presents the computational results of solving the generated instances.
considering the three above-mentioned functions for facility opening cost, namely, linear, square-root, and fractional functions. To have a fair comparison, we fixed α₀, the initial value of the control parameter in Eq. 9 equal 0.01 and it is halved if there is no significant improvement in lower bounds for 10 consecutive iterations.

The columns of each table respectively show the instance name, picked from Holmberg et al. (1999), number of potential service facilities, number of customers, total cost of designing the service system, percentage of each cost component from the total cost, i.e., the opening, serving, accessing, and waiting costs, number of the iterations the algorithm runs for reaching the user-defined tolerance or the predefined limit for maximum number of iterations, the tolerance when the algorithm stops running, the CPU time of running the algorithm, number of opened service facilities, and the average service capacity in the opened facilities. As shown, the proposed algorithm is very successful in solving the instances, where most instances reach the defined tolerance in less than 24 seconds. It is also seen that more iterations are required for the algorithm to reach the user-defined tolerance in the case that the economies-of-scale opening functions are assumed. Moreover, the CPU time is almost higher for square-root and fractional functions. This happens since some breakdown points have been calculated for piece-wise linearization of the opening function and the algorithm spends much longer time for finding the lower bound solution in each iteration of the solution algorithm.

By switching from linear opening cost to the ones benefits from the economies-of-scale, i.e., square-root or fractional opening cost, the total cost slightly decreases, see Figure 2. Although the total cost might not change significantly in the presence of economies of scale, the system opens fewer service facilities with higher service capacities on average. This causes the waiting costs to decrease in the whole system, which has a higher intensity in fractional function rather than the square-root one; see Figure 3 for better comparison per instance.
5 Conclusions

In this paper, we study an economies-of-scale design of a service system where the opening cost of service facilities are assumed to be any arbitrary function, with characteristics of being concave and non-decreasing. The goal is to optimally decide the number, location, and service capacity of opened facilities as well as the allocation of customers to the facilities. The problem is modeled as a mixed-integer nonlinear program that is too complex to be solved for any kind of opening function. Therefore, it is first approximated by its piece-wise linear functions and then a Lagrangian decomposition algorithm is developed, in which the relaxed model is decomposed into smaller subproblems. A polynomial-time exact algorithm is developed for solving each submodel where the optimal service capacity of each facility is drawn as a closed-form expression. The proposed solution algorithm is successful in solving the instances of the problem; however, considering economies-of-scale increases the solving time in comparison to the traditional model in which a simple linear opening function is considered. Our computational results show that in the presence of economies-of-scale, fewer service facilities are opened with higher service capacity. This decreases the average waiting time of customers and subsequently increases the service quality.
Figure 3: Comparison on waiting cost percentage considering different opening cost function.

One can consider other kinds of queuing systems in service facilities, e.g., queue with general service time such as M/G/1, as a direction for future research. Moreover, one can consider a profit-maximizing service system where the facilities can use the benefits of economies-of-scale to serve the customers, serving who are not traditionally profitable for the system.
Table 1: Computational results when facility opening cost is a linear function of service capacity

| Instance | # Service Facilities | # Customers | Total Cost | Opening Cost | Serving Cost | Accessing Cost | Waiting Cost | # Iterations | CPU time (ms) | Error Tolerance | # Open Facilities | Average Service Capacity |
|----------|----------------------|-------------|------------|--------------|--------------|----------------|--------------|--------------|----------------|----------------|------------------|------------------------|
| P1       | 10                   | 50          | 25027.4    | 23%          | 19%          | 48%            | 16%          | 9            | 0              | 0.005           | 1                | 318                    |
| P2       | 10                   | 50          | 23077.4    | 19%          | 20%          | 30%            | 16%          | 11           | 0              | 0.002           | 1                | 318                    |
| P3       | 10                   | 50          | 25877.4    | 25%          | 19%          | 46%            | 16%          | 8            | 0              | 0.007           | 1                | 318                    |
| P4       | 10                   | 50          | 27877.4    | 31%          | 17%          | 43%            | 9%           | 4            | 0              | 0.000           | 1                | 318                    |
| P13      | 20                   | 50          | 28450.6    | 25%          | 17%          | 34%            | 25%          | 11           | 0              | 0.003           | 1                | 700                    |
| P14      | 20                   | 50          | 26858.0    | 25%          | 18%          | 32%            | 25%          | 8            | 0              | 0.010           | 1                | 755                    |
| P15      | 20                   | 50          | 28958.0    | 30%          | 16%          | 30%            | 23%          | 7            | 0              | 0.011           | 1                | 755                    |
| P16      | 20                   | 50          | 30058.0    | 35%          | 15%          | 28%            | 22%          | 5            | 0              | 0.010           | 1                | 755                    |
| P25      | 30                   | 150         | 40259.5    | 30%          | 32%          | 27%            | 11%          | 89           | 125            | 0.010           | 2                | 309                    |
| P26      | 30                   | 150         | 37264.3    | 24%          | 34%          | 30%            | 12%          | 85           | 110            | 0.010           | 2                | 308                    |
| P27      | 30                   | 150         | 41264.3    | 32%          | 31%          | 27%            | 11%          | 80           | 109            | 0.009           | 2                | 308                    |
| P28      | 30                   | 150         | 45264.3    | 38%          | 28%          | 24%            | 16%          | 77           | 172            | 0.008           | 2                | 308                    |
| P41      | 10                   | 90          | 17009.8    | 33%          | 13%          | 37%            | 18%          | 39           | 16             | 0.010           | 1                | 500                    |
| P42      | 20                   | 80          | 15807.8    | 34%          | 12%          | 35%            | 18%          | 5            | 0              | 0.001           | 1                | 473                    |
| P43      | 30                   | 70          | 14311.3    | 37%          | 12%          | 30%            | 20%          | 7            | 0              | 0.000           | 1                | 452                    |
| P44      | 10                   | 90          | 20486.1    | 28%          | 13%          | 43%            | 16%          | 35           | 0              | 0.010           | 1                | 550                    |
| P45      | 20                   | 80          | 18250.5    | 31%          | 13%          | 40%            | 17%          | 27           | 15             | 0.010           | 1                | 520                    |
| P46      | 30                   | 70          | 16990.9    | 32%          | 12%          | 30%            | 17%          | 17           | 16             | 0.010           | 1                | 489                    |
| P47      | 10                   | 90          | 18871.2    | 31%          | 14%          | 41%            | 14%          | 55           | 16             | 0.010           | 1                | 459                    |
| P48      | 20                   | 80          | 17403.5    | 33%          | 14%          | 38%            | 15%          | 53           | 16             | 0.010           | 1                | 525                    |
| P49      | 30                   | 70          | 16791.7    | 33%          | 12%          | 40%            | 15%          | 60           | 32             | 0.010           | 1                | 490                    |
| P50      | 10                   | 100         | 21212.7    | 27%          | 11%          | 46%            | 15%          | 8            | 0              | 0.007           | 1                | 530                    |
| P51      | 10                   | 100         | 26095.9    | 28%          | 12%          | 45%            | 16%          | 64           | 32             | 0.010           | 1                | 530                    |
| P52      | 10                   | 100         | 26288.9    | 23%          | 11%          | 54%            | 13%          | 16           | 0              | 0.008           | 1                | 578                    |
| P53      | 20                   | 100         | 26177.2    | 37%          | 14%          | 34%            | 15%          | 290          | 156            | 0.010           | 3                | 239                    |
| P54      | 10                   | 100         | 22792.3    | 26%          | 13%          | 47%            | 13%          | 59           | 16             | 0.010           | 1                | 592                    |
| P55      | 20                   | 100         | 18356.9    | 33%          | 16%          | 34%            | 17%          | 39           | 15             | 0.010           | 1                | 592                    |
| Instance | # Service Facilities | # Customers | Total Cost | Opening Cost | Serving Cost | Accessing Cost | Waiting Cost | # Iterations | CPU time (ms) | Error Tolerance | # Open Facilities | Average Service Capacity |
|----------|----------------------|-------------|------------|--------------|--------------|---------------|--------------|--------------|---------------|----------------|-----------------|--------------------------|
| P1       | 10                   | 50          | 23421.1    | 19%          | 21%          | 53%           | 7%           | 13           | 312           | 0.010          | 1               | 13631                     |
| P2       | 10                   | 50          | 21305.2    | 21%          | 11%          | 50%           | 8%           | 16           | 265           | 0.010          | 1               | 2282                     |
| P3       | 10                   | 50          | 23305.2    | 28%          | 10%          | 54%           | 8%           | 8            | 250           | 0.002          | 1               | 2282                     |
| P4       | 10                   | 50          | 25305.2    | 34%          | 10%          | 50%           | 7%           | 5            | 250           | 0.002          | 1               | 2282                     |
| P13      | 20                   | 50          | 23642.3    | 21%          | 20%          | 48%           | 11%          | 17           | 1110          | 0.010          | 1               | 3053                     |
| P14      | 20                   | 50          | 22099.7    | 21%          | 21%          | 46%           | 11%          | 12           | 1062          | 0.009          | 1               | 2915                     |
| P15      | 20                   | 50          | 24309.7    | 28%          | 20%          | 43%           | 10%          | 8            | 1047          | 0.001          | 1               | 2915                     |
| P16      | 20                   | 50          | 26399.7    | 33%          | 18%          | 30%           | 10%          | 6            | 1062          | 0.002          | 1               | 2915                     |
| P25      | 30                   | 150         | 37604.0    | 29%          | 34%          | 31%           | 6%           | 107          | 23125         | 0.010          | 3               | 735                      |
| P26      | 30                   | 150         | 34624.0    | 23%          | 37%          | 34%           | 6%           | 117          | 23500         | 0.010          | 2               | 886                      |
| P27      | 30                   | 150         | 38624.0    | 31%          | 33%          | 30%           | 6%           | 94           | 2812          | 0.010          | 2               | 886                      |
| P28      | 30                   | 150         | 42624.0    | 37%          | 30%          | 27%           | 5%           | 84           | 23579         | 0.010          | 2               | 886                      |
| P41      | 10                   | 90          | 14756.0    | 30%          | 15%          | 47%           | 9%           | 7            | 813           | 0.001          | 1               | 1720                     |
| P42      | 20                   | 80          | 13692.6    | 32%          | 14%          | 43%           | 9%           | 69           | 2922          | 0.010          | 1               | 1627                     |
| P43      | 30                   | 70          | 12275.7    | 35%          | 15%          | 40%           | 10%          | 82           | 4484          | 0.010          | 1               | 1558                     |
| P44      | 10                   | 90          | 17958.6    | 25%          | 15%          | 53%           | 8%           | 8            | 813           | 0.002          | 1               | 1890                     |
| P45      | 20                   | 80          | 15920.2    | 28%          | 15%          | 49%           | 8%           | 54           | 2797          | 0.010          | 1               | 1786                     |
| P46      | 30                   | 70          | 14834.5    | 29%          | 14%          | 48%           | 8%           | 75           | 4453          | 0.010          | 1               | 1684                     |
| P47      | 10                   | 90          | 16588.4    | 27%          | 16%          | 51%           | 6%           | 29           | 890           | 0.010          | 1               | 2130                     |
| P48      | 20                   | 80          | 15173.1    | 20%          | 16%          | 48%           | 8%           | 7            | 2547          | 0.006          | 1               | 1804                     |
| P49      | 30                   | 70          | 14749.8    | 30%          | 14%          | 49%           | 7%           | 81           | 4469          | 0.010          | 1               | 1686                     |
| P50      | 10                   | 100         | 18774.0    | 24%          | 13%          | 56%           | 7%           | 67           | 1250          | 0.010          | 1               | 1823                     |
| P51      | 10                   | 100         | 18257.2    | 24%          | 13%          | 55%           | 8%           | 85           | 4469          | 0.010          | 1               | 1823                     |
| P52      | 10                   | 100         | 23603.2    | 19%          | 12%          | 63%           | 6%           | 73           | 1266          | 0.010          | 1               | 2241                     |
| P53      | 20                   | 100         | 23365.8    | 35%          | 16%          | 41%           | 8%           | 321          | 7000          | 0.010          | 2               | 1137                     |
| P54      | 10                   | 100         | 20205.9    | 23%          | 15%          | 57%           | 6%           | 19           | 1094          | 0.010          | 1               | 2296                     |
| P55      | 20                   | 100         | 15748.3    | 29%          | 19%          | 44%           | 8%           | 53           | 4266          | 0.010          | 1               | 2296                     |
Table 3: Computational results when facility opening cost is a fractional function of service capacity

| Instance | # Service Facilities | # Customers | Total Cost | Opening Cost | Serving Cost | Accessing Cost | Waiting Cost | # Iterations | CPU time (ms) | Error Tolerance | # Open Facilities | Average Service Capacity |
|----------|----------------------|-------------|------------|--------------|--------------|----------------|--------------|--------------|---------------|----------------|----------------|------------------------|
| P1       | 10                   | 50          | 26011.7    | 28%          | 19%          | 49%            | 4%           | 10000        | 5032          | 0.027           | 1               | 5746                   |
| P2       | 10                   | 50          | 23711.6    | 31%          | 10%          | 54%            | 5%           | 275          | 172           | 0.10            | 1               | 5746                   |
| P3       | 10                   | 50          | 25711.6    | 36%          | 9%           | 50%            | 5%           | 172          | 230           | 0.10            | 1               | 5746                   |
| P4       | 10                   | 50          | 27711.6    | 41%          | 9%           | 46%            | 4%           | 210          | 156           | 0.10            | 1               | 5746                   |
| P13      | 20                   | 50          | 25723.7    | 29%          | 18%          | 45%            | 7%           | 10000        | 9297          | 0.13            | 1               | 7676                   |
| P14      | 20                   | 50          | 24532.3    | 30%          | 19%          | 44%            | 7%           | 10000        | 9300          | 0.031           | 2               | 4761                   |
| P15      | 20                   | 50          | 26332.3    | 35%          | 18%          | 40%            | 6%           | 10000        | 9641          | 0.029           | 2               | 4761                   |
| P16      | 20                   | 50          | 28532.3    | 40%          | 17%          | 38%            | 6%           | 10000        | 10297         | 0.036           | 2               | 4761                   |
| P25      | 30                   | 150         | 43399.6    | 40%          | 29%          | 28%            | 3%           | 154          | 5968          | 0.10            | 2               | 2965                   |
| P26      | 30                   | 150         | 40419.6    | 35%          | 31%          | 30%            | 3%           | 156          | 5875          | 0.10            | 2               | 2965                   |
| P27      | 30                   | 150         | 44419.6    | 41%          | 29%          | 28%            | 2%           | 155          | 5938          | 0.10            | 2               | 2965                   |
| P28      | 30                   | 150         | 48419.6    | 46%          | 26%          | 25%            | 2%           | 163          | 6032          | 0.10            | 2               | 2965                   |
| P41      | 10                   | 90          | 17397.2    | 42%          | 12%          | 41%            | 4%           | 235          | 106           | 0.10            | 1               | 4894                   |
| P42      | 20                   | 80          | 16369.2    | 45%          | 12%          | 39%            | 4%           | 260          | 1047          | 0.10            | 1               | 4894                   |
| P43      | 30                   | 70          | 14972.5    | 49%          | 12%          | 35%            | 5%           | 278          | 1547          | 0.10            | 1               | 4894                   |
| P44      | 10                   | 90          | 20529.8    | 36%          | 13%          | 48%            | 4%           | 218          | 359           | 0.10            | 1               | 4894                   |
| P45      | 20                   | 80          | 183419.6   | 30%          | 13%          | 44%            | 4%           | 353          | 1157          | 0.10            | 1               | 4894                   |
| P46      | 30                   | 70          | 17500.7    | 42%          | 12%          | 42%            | 4%           | 250          | 1500          | 0.10            | 1               | 4894                   |
| P47      | 10                   | 90          | 19178.4    | 38%          | 14%          | 45%            | 4%           | 93           | 265           | 0.10            | 2               | 3005                   |
| P48      | 20                   | 80          | 17820.9    | 41%          | 13%          | 42%            | 4%           | 120          | 812           | 0.10            | 1               | 5133                   |
| P49      | 30                   | 70          | 17445.8    | 42%          | 12%          | 43%            | 4%           | 181          | 1359          | 0.10            | 2               | 4006                   |
| P50      | 10                   | 100         | 21387.9    | 34%          | 11%          | 51%            | 4%           | 224          | 469           | 0.10            | 1               | 5185                   |
| P51      | 10                   | 100         | 20851.0    | 33%          | 12%          | 49%            | 4%           | 248          | 1531          | 0.10            | 1               | 5185                   |
| P52      | 10                   | 100         | 26165.4    | 28%          | 11%          | 58%            | 3%           | 234          | 469           | 0.10            | 1               | 5638                   |
| P53      | 20                   | 100         | 26193.1    | 28%          | 11%          | 58%            | 3%           | 10000        | 23641         | 0.13            | 2               | 2943                   |
| P54      | 10                   | 100         | 22734.9    | 32%          | 13%          | 51%            | 3%           | 96           | 328           | 0.10            | 1               | 5776                   |
| P55      | 20                   | 100         | 18299.5    | 40%          | 16%          | 39%            | 4%           | 122          | 1203          | 0.10            | 1               | 5776                   |
References

Aboolian, R., Berman, O., and Drezner, Z. (2008). Location and allocation of service units on a congested network. *IIE Transactions*, 40(4):422–433.

Aboolian, R., Berman, O., and Krass, D. (2012). Profit maximizing distributed service system design with congestion and elastic demand. *Transportation Science*, 46(2):247–261.

Ahmadi-Javid, A., Berman, O., and Hoseinpour, P. (2018). Location and capacity planning of facilities with general service-time distributions using conic optimization. *arXiv preprint arXiv:1809.00080*.

Ahmadi-Javid, A. and Hoseinpour, P. (2017). Convexification of queueing formulas by mixed-integer second-order cone programming: An application to a discrete location problem with congestion. *arXiv preprint arXiv:1710.05794*.

Ahmadi-Javid, A. and Hoseinpour, P. (2018). Service system design for managing interruption risks: A backup-service risk-mitigation strategy. *European Journal of Operational Research*.

Baron, O., Berman, O., and Krass, D. (2008). Facility location with stochastic demand and constraints on waiting time. *Manufacturing & Service Operations Management*, 10(3):484–505.

Baumgartner, K., Fuetterer, A., and Thonemann, U. W. (2012). Supply chain design considering economies of scale and transport frequencies. *European journal of operational research*, 218(3):789–800.

Berman, O. and Drezner, Z. (2007). The multiple server location problem. *Journal of the Operational Research Society*, 58(1):91–99.

Berman, O. and Krass, D. (2015). Stochastic location models with congestion. In *Location science*, pages 443–486. Springer.
Boffey, B., Galvao, R., and Espejo, L. (2007). A review of congestion models in the location of facilities with immobile servers. *European Journal of Operational Research, 178*(3):643–662.

Castillo, I., Ingolfsson, A., and Sim, T. (2009). Social optimal location of facilities with fixed servers, stochastic demand, and congestion. *Production and Operations Management, 18*(6):721–736.

Dasci, A. and Verter, V. (2001). The plant location and technology acquisition problem. *IIE transactions, 33*(11):963–974.

Daskin, M. S., Coullard, C. R., and Shen, Z.-J. M. (2002). An inventory-location model: Formulation, solution algorithm and computational results. *Annals of operations research, 110*(1-4):83–106.

Doukas, J. and Switzer, L. N. (1991). Economies of scale and scope in canadian branch banking. *Journal of International Financial Markets, Institutions & Money, 1*(2):61–84.

Dupont, L. (2008). Branch and bound algorithm for a facility location problem with concave site dependent costs. *International journal of production economics, 112*(1):245–254.

Elhedhli, S. (2005). Exact solution of a class of nonlinear knapsack problems. *Operations research letters, 33*(6):615–624.

Elhedhli, S. (2006). Service system design with immobile servers, stochastic demand, and congestion. *Manufacturing & Service Operations Management, 8*(1):92–97.

Elhedhli, S., Wang, Y., and Saif, A. (2018). Service system design with immobile servers, stochastic demand and concave-cost capacity selection. *Computers & Operations Research, 94*:65–75.

Fisher, M. L. (2004). The lagrangian relaxation method for solving integer programming problems. *Management science, 50*(12 supplement):1861–1871.
Geoffrion, A. M. (2010). Lagrangian relaxation for integer programming. In *50 Years of Integer Programming 1958-2008*, pages 243–281. Springer.

Holmberg, K., Rönqvist, M., and Yuan, D. (1999). An exact algorithm for the capacitated facility location problems with single sourcing. *European Journal of Operational Research*, 113(3):544–559.

Hoseinpour, P. and Ahmadi-Javid, A. (2016). A profit-maximization location-capacity model for designing a service system with risk of service interruptions. *Transportation Research Part E: Logistics and Transportation Review*, 96:113–134.

Hoseinpour, P. and Ahmadi-Javid, A. (2019). Designing service system networks with interruption risks. *International Transactions in Operational Research*, 26(6):2561–2578.

Kim, S. (2013). A column generation heuristic for congested facility location problem with clearing functions. *Journal of the Operational Research Society*, 64(12):1780–1789.

Preyra, C. and Pink, G. (2006). Scale and scope efficiencies through hospital consolidations. *Journal of health economics*, 25(6):1049–1068.

Rajagopalan, S. and Yu, H.-L. (2001). Capacity planning with congestion effects. *European Journal of Operational Research*, 134(2):365–377.

Saif, A. and Elhedhli, S. (2016). A lagrangian heuristic for concave cost facility location problems: the plant location and technology acquisition problem. *Optimization Letters*, 10(5):1087–1100.

Shortle, J. F., Thompson, J. M., Gross, D., and Harris, C. M. (2018). *Fundamentals of queueing theory*, volume 399. John Wiley & Sons.

Silva, F. and Serra, D. (2008). Locating emergency services with different priorities: the priority queuing covering location problem. *Journal of the Operational Research Society*, 59(9):1229–1238.
Vidyarthi, N. and Jayaswal, S. (2014). Efficient solution of a class of location–allocation problems with stochastic demand and congestion. *Computers & Operations Research*, 48:20–30.

Wang, Q., Batta, R., and Rump, C. M. (2002). Algorithms for a facility location problem with stochastic customer demand and immobile servers. *Annals of operations Research*, 111(1-4):17–34.

Wang, Q., Batta, R., and Rump, C. M. (2004). Facility location models for immobile servers with stochastic demand. *Naval Research Logistics (NRL)*, 51(1):137–152.