Yugoslav Journal of Operations Research
28 (2018), Number 4, 453–473
DOI: https://doi.org/10.2298/YJOR170110021I

THE CAPACITATED SINGLE-SOURCE P-CENTER PROBLEM IN THE PRESENCE OF FIXED COST AND MULTILEVEL CAPACITIES USING VNS AND AGGREGATION TECHNIQUE

Chandra A. IRAWAN
Nottingham University Business School China Trent Building Room 250, 199 Taikang East Road Ningbo 315100, China
chandra.irawan@nottingham.edu.cn

Kusmaningrum SOEMADI
Jurusan Teknik Industri Institut Teknologi Nasional, Jl. PKH. Mustafa no 23, Bandung 40134
kusmaningrum@yahoo.com

Received: January 2017 / Accepted: July 2018

Abstract: In this study, the discrete p-center problem with the presence of multilevel capacities and fixed (opening) cost of a facility under a limited budget is investigated. A mathematical model of the problem is produced, where we seek the location of open facilities, their corresponding capacities, and the allocation of the customers to the open facilities in order to minimise the maximum distance between customers and their assigned facilities. Two matheuristic approaches are also proposed to deal with larger instances. The first approach is a hybridisation of a clustering-based technique, an exact method, while the second one is based on Variable Neighbourhood Search (VNS). Computational experiments show that the proposed methods produce interesting and competitive results on newly and randomly generated datasets.

Keywords: Capacitated p-center Problem, Mathematical Model, Matheuristic, VNS.

MSC: 90B85, 90C26.

1. INTRODUCTION

The discrete p-center problem deals with finding the location of p facilities among M potential sites and assigning customers to these facilities in order to minimise the maximum distance between customers and their nearest facility. This problem is also known as the minimax location problem where the minimax criterion aims to minimise the adverse effects of worst-case scenarios in providing service to the customers. This problem is considered as a single source type-problem where a customer demand is satisfied only from one facility. The application of this problem includes the location of facilities in emergency services such as ambulance, fire, and police stations. In the capacitated version, each customer's demand and capacity of a potential facility are known. The problem is NP-hard given its relaxed version, the uncapacitated one, to be known as NP-hard (Kariv and Hakimi [21]), too.

The uncapacitated p-center problem was first proposed by Hakimi [13] who investigated an absolute 1-center problem on a graph. Bar-Ilan *et* al. [3] introduced the capacitated version of the p-center problem to address the problem of locating centers in distributed communication networks. According to the authors, the problem is referred to as the Balanced p-Centre Problem as its aim is to obtain a workload balance among centers. For more description on the p-center problem, see the recent work by Irawan *et* al. [19].

In many real case applications, the fixed cost of opening facilities is usually considered when determining the best location for the opening facilities. The fixed cost of a facility may be dependent on its location and the capacity of the facility. For example, the cost of land in a bigger city is higher than in a smaller city. The effect of fixed cost is successfully introduced in related location problem such as the multi-source case, see Brimberg and Salhi [6]. As the fixed cost of opening facilities is also related to the capacity of the facilities, decision makers need to determine an optimal capacity for each opening facility given their limited budget. This paper attempts to address this logistical problem by developing a new mathematical model and solution methods. Moreover, the capacity of each facility is a decision variable that needs to be optimally chosen from a list of possible capacities.

The main contributions of this paper are as follows:

- propose a new mathematical model for the discrete capacitated p-center problem with the presence of fixed cost and multilevel capacities,
- propose effective matheuristic approaches based on clustering method and VNS algorithm,
- generate a new dataset for the new problem and produce optimal and best solutions for benchmarking purposes.

The paper is organised as follows. In the next section, mathematical models for the classical uncapacitated and capacitated p-center problems along with the new problem considering the presence of fixed cost and multilevel capacities are presented. The section thereafter describes the proposed matheuristic methods. Section 5 provides the computational results and analysis. In the last section, conclusions and some highlights of future research avenues are given.

2. LITERATURE REVIEW

In this section, a review on the capacitated p-center problem is presented. The capacitated p-center problem on tree networks was studied by Jaeger and Goldberg [20]. The authors reveal that the problem can be addressed in polynomial time when the facility capacities are identical. Khuller and Sussmann [22] investigated the capacitated p-center problem where each facility can be assigned to at most L customers. A polynomial approximation algorithm is introduced for solving the problem. Scaparra et al. [28] studied the application of very large neighbourhood search techniques for solving the capacitated vertex p-center problem. The authors characterize a local search neighbourhood in terms of path and cyclic exchanges of customers among facilities. They also use principles borrowed from network optimization theory to efficiently detect cost-decreasing solutions in such a neighbourhood. In addition, the multi-exchange methodology with a relocation mechanism is designed to perform facility location adjustments. Őzsoy and Pinar [24] proposed an exact algorithm for solving the capacitated vertex p-center problem. They built a simple and practical exact algorithm that iteratively sets a maximum distance value, within which it tries to assign all the clients.

Albareda-Sambola et al. [1] investigated two auxiliary problems related to the capacitated vertex p-center problem. They proposed two different Lagrangean duals based on each of these auxiliary problems. Two different strategies for solving exactly the problems, based on binary search and sequential search, are also introduced. Cygan et al. [10] investigated a generalization of the capacitated p-center problem where a constant factor approximation algorithm is introduced to address it, and an LP rounding technique is used. Recently, a metaheuristic was proposed by Quevedo-Orozco and Ríos-Mercado ([25], [26]), where a greedy randomized adaptive procedure with biased sampling in its construction phase, and iterated greedy with a variable neighbourhood descent in its local search phase are incorporated. An et al. [2] proposed a simple algorithm that incorporated the standard LP relaxation. The algorithm first reduces the problem into special tree instances, and then the best-possible algorithm is implemented to solve such instances.

Other variants on the capacitated p-center problem include Bashiri et al. [4], along with Chechik and Peleg [7]. A fuzzy capacitated p-hub center model was introduced by Bashiri et al. [4]. It aims to minimise maximum travel time in networks by locating p hubs from a set of candidate hub locations, allocating demand, and supply nodes to hubs. A genetic algorithm solution is proposed to address such problem. Chechik and Peleg [7] investigated the fault-tolerant capacitated k-center problem where one or more service facilities might fail simultaneously. Two variants of the problem are taken into account. The first problem is called the α-fault-tolerant problem, where after the failure of some facilities, all customers are allowed to be reassigned to alternate facilities. The second one is referred to as the α-fault-tolerant conservative, where once a facility fails, only the customers assigned to this facility before its failure are allowed to be reassigned to other facilities.

3. PROBLEM FORMULATION

In this section we first present the mathematical model of the capacitated single source p-center problem, followed by the new problem, where the presence of multilevel capacities and fixed cost are considered.

3.1. The capacitated single source p-center problem

In the capacitated version of the p-center problem (CPCP), each customer i has a known demand (w_i) and each potential facility j has a known capacity (b_j). Total customer demand allocated to each facility j does not exceed its capacity while imposing each customer demand to be entirely assigned to one facility (single-source location problem). In this case, due to the capacity constraint, a customer is not necessarily served by its closest facility. The following notations are used to describe the sets, parameters, and decision variables of the problem.

Sets

- I: a set of demand points/customers with i as its index and $n = |I|$
- J: a set of potential sites with j as its index and $M = |J|$

Parameters

- d_{ij}: the distance between customer $i \in I$ and potential site $j \in J$ (Euclidian distance will be used in this study)
- p: the number of open facilities
- w_i: the demand of customer $i \in I$
- b_j: the capacity of facility located at potential site $j \in J$

Decision Variables

- r: the maximum distance between customers and their facilities

- $Y_{ij} = \begin{cases} 1 & \text{if customer } i \in I \text{ is assigned to facility } j \in J, \\ 0 & \text{otherwise} \end{cases}$
- $X_j = \begin{cases} 1 & \text{if an open facility is located at site } j \in J, \\ 0 & \text{otherwise} \end{cases}$

In Scaparra *et* al. [28], the capacitated single source p-center problem is modelled as a Mixed Integer Linear Programming (MILP):

$$\min r \tag{1}$$

Subject to:

$$\sum_{j \in J} Y_{ij} = 1, \qquad \forall i \in I \tag{2}$$

$$\sum_{j \in J} X_j = p \tag{3}$$

$$Y_{ij} - X_j \leq 0, \qquad \forall i \in I, j \in J \tag{4}$$

$$r \geq \sum_{j \in J} (d_{ij} \cdot Y_{ij}), \qquad \forall i \in I \tag{5}$$

$$\sum_{i \in I} (w_i \cdot Y_{ij}) \leq b_j \cdot X_j, \qquad \forall j \in J \tag{6}$$

$$X_j \in \{0, 1\}, \qquad \forall j \in J \tag{7}$$

$$Y_{ij} \in \{0, 1\}, \qquad \forall i \in I, j \in J \tag{8}$$

The objective function (1) is to minimise the maximum distance between a customer and his associated facility. Constraints (2) ensure that each customer i is assigned to exactly one open facility. Constraint (3) guarantees the number of open facilities to be exactly p whereas Constraints (4) ensure that customer i can only be assigned to an open facility. Constraints (5) impose the variable r to be the minimum of longest customer-facility distance. Constraints (6) impose the total demand of customers allocated to a facility not to exceed its capacity. Constraints (7) and (8) indicate that X and Y are binary decision variables.

3.2. *The capacitated single source p-center problem with fixed cost and multilevel capacities*

The mathematical model of the single source capacitated p-center problem with fixed cost and multilevel capacities (CPCPFC) is presented in this subsection. In this study, the model treats the capacity of the chosen facility as a decision variable. In other words, each potential facility has a set of possible facility designs $(K_j, j \in J)$, where a facility design defines the capacity of a potential facility $(\hat{b}_{jk}, j \in J, k \in K_j)$. As the fixed cost of an open facility is based on the chosen capacity and its location, let f_{jk} $(j \in J, k \in K_j)$ be the fixed cost of each potential facility j when using capacity design k. The model also takes into account the available budget $\hat{c}$ for opening the selected facilities meaning that the total fixed cost cannot exceed the available budget. In this model, the notations used for sets, parameters and decision variables are similar to the ones presented in subsection 3.1 with some additions described as follows:

Set

- K_j: a set of capacity designs for facility $j \in J$

Parameters

- f_{jk}: fixed cost of potential facility $j \in J$ when using capacity design $k \in K_j$
- $\hat{b}_{jk}$: capacity of facility $j \in J$ using design $k \in K_j$
- $\hat{c}$: maximum (available) budget for opening p facilities

Decision Variables

- r: the maximum distance between customers and their facilities
- $Y_{ij} = \begin{cases} 1 & \text{if customer } i \in I \text{ is assigned to facility } j \in J, \\ 0 & \text{otherwise} \end{cases}$
- $\hat{X}_{jk} = \begin{cases} 1 & \text{if an open facility is located at site } j \in J \text{ when using design } k \in K_j, \\ 0 & \text{otherwise} \end{cases}$

The CPCPFC problem is much harder to solve than the CPCP problem as it determines the location of the open facilities but also the capacity for each open facility. The problem can be modelled as a MILP as follows:

$$\min r \tag{9}$$

Subject to:

$$\sum_{j \in J} Y_{ij} = 1, \qquad \forall i \in I \tag{10}$$

$$\sum_{k\in K_j} \hat{X}_{jk} \le 1, \qquad \forall j \in J \tag{11}$$

$$\sum_{j\in J}\sum_{k\in K_j} \hat{X}_{jk} = p \tag{12}$$

$$\sum_{i\in I} (w_i \cdot Y_{ij}) \le \sum_{k\in K_j} \left(\hat{b}_{jk} \cdot \hat{X}_{jk}\right), \qquad \forall j \in J \tag{13}$$

$$Y_{ij} - \sum_{k\in K_j} \hat{X}_{jk} \le 0, \qquad \forall i \in I, j \in J \tag{14}$$

$$r \ge \sum_{j\in J} (d_{ij} \cdot Y_{ij}), \qquad \forall i \in I \tag{15}$$

$$\sum_{j\in J}\sum_{k\in K_j} \left(f_{jk} \cdot \hat{X}_{jk}\right) \le \hat{c} \tag{16}$$

$$\hat{X}_{jk} \in \{0,1\}, \qquad \forall j \in J, k \in K_j \tag{17}$$

$$Y_{ij} \in \{0,1\}, \qquad \forall i \in I, j \in J \tag{18}$$

The objective function of this model is the same as in the previous model, whereas Constraints (10) ensure that each customer is served by one facility. Constraints (11) ensure that each open facility uses only one capacity design. Constraint (12) guarantees the number of open facilities to be exactly p. Constraints (13) state capacity constraints of the facilities. Constraints (14) guarantee that each customer can only be assigned to one open facility. Constraints (15) define the maximum distance between customers and their facilities. Constraint (16) ensures that the total fixed cost does not exceed the available budget. Constraints (17) and (18) define the binary nature of decision variables $\hat{X}$ and Y.

4. THE PROPOSED MATHEURISTICS

The discrete capacitated p-center problem with the presence of fixed cost and multilevel capacities (the CPCPFC problem) is very hard to solve using an exact method (CPLEX) especially when the size of the problem is relatively large. To overcome this weakness, matheuristic approaches are proposed in this study. For more information on heuristic search and metaheuristic, see Salhi [27]. Here, we propose two solution methods for solving the problem where the first one is referred to as the adaptive matheuristic method (AMM) while the second is a VNS-based matheuristic. For simplicity, we consider all potential facility sites as customer sites (i.e. $|J| = n$).

4.1. The adaptive matheuristic method (AMM)

One way to tackle the large problem is to simplify it by adapting a common approach to aggregate/decompose the problem into a smaller one so it can be addressed within a reasonable amount of computing time (Francis et al., [12]). Irawan and Salhi [18] provide a review on aggregation techniques for large facility location problems. Here, AMM that incorporates aggregation technique, an exact method, and a local search is proposed. The main steps of this approach are depicted in Algorithm 1.

Algorithm 1: The adaptive matheuristic method (AMM)

Initialisation: Define T and m. Set $z = \infty$, $S = \emptyset$ and $U = \emptyset$.

Step 1: Construct p clusters of potential facility sites using Coopers algorithm that aims to partition n potential facility sites into p clusters.

Step 2: Do the following steps T times:

a. Aggregate n to m potential facility sites by including the facility locations with their corresponding capacity design in the incumbent solution (S and U).
b. Solve the aggregated problem (model 9 - 18) using the exact method (CPLEX). Let z' be its objective function value with S' and U' as the obtained facility configuration and their corresponding capacity design respectively. i' and j' are also obtained where $z' = d_{i'j'}$.
c. If $z' < z$, then set $z = z', i^* = i'$ and $j^* = j'$ along with $S \leftarrow S'$ and $U \leftarrow U'$.

Step 3:

a. Apply the proposed local search (given in Algorithm 4) with z, i^*, j^*, S, and U obtained from previous stage as inputs and outputs.
b. Take z, S and U as the best objective functions generated along with their facility configuration and the corresponding capacity design.

In the first step, a clustering process is conducted to construct p clusters/groups of potential facility sites. The clustering procedure is used to aggregate n potential facility sites into m potential sites, with $m << n$. By using this method, we aim to select relatively good potential facility sites and make sure that the selected potential facilities are not located only in one specific area. We use coopers method [8] to cluster n potential facility sites into p groups.

The second step is an iterative process that incorporates potential facility sites aggregation and the use of the exact method (CPLEX). When selecting the potential facility sites, the aggregation includes the facility sites with their corresponding capacity design in the incumbent solution. This set of facility sites and their corresponding capacity designs are denoted by S and U, respectively. The resulting aggregated problem with n customers and m potential facility sites is then solved optimally by CPLEX. The model will find customer i^* who has the longest distance from its location to its facility (j^*). It can be written as $z = d_{i^*j^*}$. The obtained solution (the location of open facilities and their capacity design) is then fed to the next iteration as part of the set of the aggregated potential sites. The process is repeated T times and the best solution from this step will be fed to the next step.

In the final step (Step 3), a local search is proposed to solve the original problem (without aggregation) starting from the best solution obtained in the previous step. The description of the clustering method, the aggregation technique, and the proposed local search are presented in the following subsections.

A. The clustering method

To cluster potential facility sites, we implement the well-known alternate location-allocation (ALA) method, initially introduced by Cooper [8], to address the classical location-allocation problem. The main idea of ALA is that the location-allocation problem is alternately applied until no epsilon (ε) improvement in total cost is found. The objective function of this method is to minimise the total distance (minisum) which is not the same as the p-center problem (minimax). However, this is used for an approximation to get some clusters. The main steps of ALA are presented in Algorithm 2.

Initially, p potential facilities are randomly selected as the cluster centers $(C_c(x_c, y_c), \forall c = 1, \ldots, p)$, and then all potential facilities are assigned to their nearest cluster center. The total distance between potential facilities and their cluster center (f) is also calculated. Let N_c, as the set of potential facilities belong to cluster c with $d(C_c, a_j)$, be the Euclidean distance between center c and potential facility j and $a_j = \left(x_j^a, y_j^a\right)$ be the location of potential facility j. The Weiszfeld equations (19) is iteratively carried out to get the new location of these p cluster centers $(C_c(x_c, y_c), \forall c = 1, \ldots, p)$.

$$\hat{x}_c = \frac{\sum_{j \in N_c} \frac{x_j^a}{d(C_c, a_j)}}{\sum_{j \in N_c} \frac{1}{d(C_c, a_j)}}, \quad \hat{y}_c = \frac{\sum_{j \in N_c} \frac{y_j^a}{d(C_c, a_j)}}{\sum_{j \in N_c} \frac{1}{d(C_c, a_j)}} \tag{19}$$

The potential facilities are then assigned to the new cluster centers $(\hat{C}_c, \forall c = 1, \ldots, p)$ resulting new total distance ($\hat{f}$) and new allocation ($\hat{N}_c$). This process is repeated until there is no further changes in total distance, within some tolerance (ε), in two successive iterations. As the initial cluster centers affect the quality of the p constructed clusters, we implement a multi-start approach to tackle this problem. In other words, the ALA process is repeated $\hat{T}$ times. We take the cluster configuration that gives the smallest total distance between the cluster centers and their potential facilities (f^*) .

Algorithm 2: ALA heuristic with multi-start

Step 1: Define ε and ($\hat{T}$.

Step 2: Set $f^* = \infty$ along with $C_c^* = \emptyset$ and $N_c^* = \emptyset \forall = c = 1, \ldots, p$.

Step 3: Do the following steps $\hat{T}$ times:

a. Select p initial starting locations at random from potential facilities locations as cluster centers ($C_c(x_c, y_c), \forall c = 1, \ldots, p$).
b. Assign the potential facilities to the nearest cluster center ($C_c, \forall c = 1, \ldots, p$). Calculate $f = \sum_{c=1}^{p} \sum_{j \in N_c} d(C_c, a_j)$.
c. Determine new cluster centers locations ($\hat{C}_c, \forall c = 1, \ldots, p$) by using the following procedure:
   i. Set $c = 1$.
   ii. Calculate $\delta = \sum_{j \in N_c} d(C_c, a_j)$.
   iii. Repeat the following steps:
      - Calculate $\hat{C}_c$ using Weiszfelds equations (19).
      - Calculate $\hat{\delta} = \sum_{j \in N_c} d(\hat{C}_c, a_j)$.
      - If $(\delta - \hat{\delta}) < \varepsilon$, then go to Step 3b(iv). Otherwise, set $\delta = \hat{\delta}$ and $C_c = \hat{C}_c$.
   iv. If $c = p$, then go to Step 3d, otherwise update $c = c + 1$ and go to Step 3c(ii).
d. Assign the potential facilities to the new nearest cluster center ($\hat{C}_c, \forall c = 1, \ldots, p$) with $\hat{f}$ and ($\hat{N}_c, \forall c = 1, \ldots, p$) as the outputs.
e. If $\hat{f} < f$, then update $f = \hat{f}$ along with $C_c = \hat{C}_c$ and $N_c = \hat{N}_c$ then go back to Step 3c(i).
f. Assign the potential facilities to the new nearest cluster center ($\hat{C}_c, \forall c = 1, \ldots, p$) with $\hat{f}$ and ($\hat{N}_c, \forall c = 1, \ldots, p$) as the outputs.
g. If $\hat{f} < f$, then update $f = \hat{f}$ along with $C_c = \hat{C}_c$ and $N_c = \hat{N}_c, \forall c = 1, \ldots, p$ then go back to Step 3c(i).
h. If $f < f^*$, then update $f^* = f$ along with $C_c^* = C_c$ and $N_c^* = N_c, \forall c = 1, \ldots, p$.

Step 4: Return $N_c^* = N_c, \forall c = 1, \ldots, p$.

B. The aggregation method

This subsection presents the description of the procedure to aggregate n potential facility sites into m sites used in Step 2a of Algorithm 1. Our aggregation method is an enhancement of the method proposed by Irawan et al. [17]. The set of the m aggregated potential facility sites includes the followings:

- The best facility locations configuration (S) obtained from previous iterations along with their corresponding capacity (U).
- $(m-p)$ pseudo randomly generated points.

First, the method includes the incumbent facility locations (S) as part of the aggregated potential facility sites. The use of these facility locations (S) will increase the probability of obtaining a good solution. Moreover, the capacity for these facility locations (S) in the aggregated problem is set to be fixed, based on the capacity configuration (U) obtained from previous iterations. On the other hand, there is a set of possible capacity designs (K_j) for the remaining aggregated points. By doing this, the computing time to solve the aggregated problem using CPLEX is reduced. Second, the cluster-based method, which is briefly described below, is applied to generate the subsequent $(m-p)$ aggregated points. This scheme overcomes the weaknesses of a simple random process when dealing with clustered customers. In addition, it ensures that the aggregated potential facility sites are not too close to each other and will spread in the study area. The main steps of the method are formally given in Algorithm 3.

Algorithm 3: The cluster-based method to aggregate potential facilities

Step 1: Set $A = S$.

Step 2: Calculate the total demand of customers (the location of potential facility sites) for each cluster c as D_c, where $D_c = \sum_{i \in N'_c} w_i, \forall c = 1, \dots, p$ with N'_c is the set of customers belonging to cluster c.

Step 3: Determine the cluster probability distribution, say $P_c = D_c / \sum_{i \in I} w_i, \forall c = 1, \dots, p$.

Step 4: Determine the customer (a potential facility site) probability distribution, say $P'_{ci} = w_i / D_c, \forall c = 1, \dots, p; i \in N'_c$.

Step 5: for $j = (p+1)$ to m do the following steps:

a. Generate randomly $\beta \in \{0, 1\}$.
b. Choose cluster $\tilde{c}$ st. $\tilde{c} = F^{-1}_{(c)}(\beta)$ with $F_{(c)} = \sum_{a=1}^{c} P_a$.
c. Generate randomly $\alpha \in \{0, 1\}$.
d. Select customer $\tilde{i}$ st. $\tilde{i} = F^{-1}_{(\tilde{c}i)}(\alpha)$ with $F_{(\tilde{c}i)} = \sum_{a=1}^{i} P'_{\tilde{c}a}$.
e. If $\tilde{i} \in A$, then go back to Step 5a, otherwise $A = A \bigcup \tilde{i}$.

Step 6: Return A.

Initially, all potential facilities are clustered using the ALA heuristic. Let A be a set of aggregated potential facilities which consists of the facility configuration obtained from the previous iteration. The total demand of customers (which are the location of potential facility sites) for each cluster, D_c, is calculated and its corresponding probability distribution (P_c) is determined. The probability distribution of each potential facility (located in customer site) (P'_{ci}) corresponding to its cluster is also calculated. A cluster is selected in a pseudo random manner, based on the cumulative probability distribution. Here, a random number $\beta \in \{0, 1\}$ is generated and then, a cluster is chosen with $\tilde{c}$ st. $\tilde{c} = F^{-1}_{(c)}(\beta)$ with $F_{(c)} = \sum_{a=1}^{c} P_a$. See Figure 1 for an illustration in the case $\tilde{c} = 3$. A potential facility/customer (say $\tilde{i}$) is chosen pseudo randomly in cluster $\tilde{c}$ using the same technique as for choosing cluster $\tilde{c}$. The location of customer $\tilde{i}$ is then selected into the set of aggregated potential facility sites (A). This procedure is reiterated until $|A| = m$.

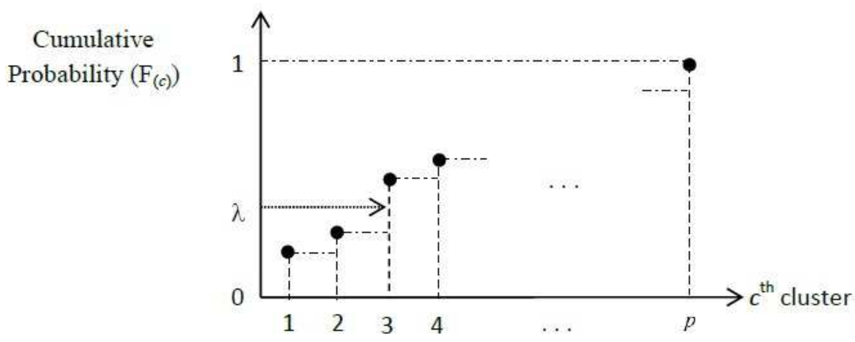

Figure 1: The illustration of selecting cluster

C. The proposed local search

The proposed local search is a hybridisation of the vertex substitution heuristic and the exact method. We enhance the vertex substitution heuristic used by Mladenovic *et* al. [23] by incorporating the exact method to allocate customers to the open facilities and to determine the capacity design used by an open facility. The exact method is embedded into the local search to optimally solve the customer allocation problem whenever the locations of open facilities are fixed. In case the location of open facilities is known, the problem reduces to an assignment problem. The model will also find the optimal capacity design for each opening facility. We refer to this problem as the CPCPFC-FL. Let $S(S \subset J)$ be a set of facility locations in the solution with $|S| = p$. The formulation of the CPCPFC-FL is similar to the one of the CPCPFC expressed as Equations (9) - (18) with minor changes as follows:

- The set of potential facility sites (J) is replaced by the set S.

- Constraints (11) are replaced by Constraints (20) as follows:

$$\sum_{k \in K_j} \hat{X}_j k = 1, \qquad \forall j \in J \tag{20}$$

- Constraints (12) are removed.

The main steps of the proposed local search are presented in Algorithm 4, which is based on the interchange heuristic using a first improvement strategy (the swapping process is conducted once there is an improvement). The algorithm aims to find a potential facility site (located in customer i) to be swapped with a facility used in the current solution. Moreover, the capacity of the facility candidate is determined. The swap will be done if improvement occurs. First, the facility (in S) that serves each customer i ($\rho(i), \forall i \in I$) needs to be found. In Step 3 of Algorithm 4, as we aim to reduce the maximum distance between a customer and his facility, we only consider the potential facilities whose distance between their locations and customer i^* (the customer that has the maximum distance to its facility (facility j^*)) is less than $z = d_{i^*j^*}$, and which are to be swapped with a facility in the current solution.

In Steps 3b of Algorithm 4, a potential facility located at customer i is inserted into the solution, whereas the facility in the current solution that serves customer i is removed. Let $H(j)$ be a set of capacity designs for each facility j. Here, there are several possible capacity designs for the new facility (located at customer i site) which is based on set K_i. On the other hand, the capacity for the existing facilities in the current solution is set to be fixed, based on the capacity configuration (U) in the incumbent solution. This will reduce the computing time to solve the CPCPFC-FL using CPLEX. Note that in Step 3b, the CPCPFC-FL using S' and H may be infeasible due to capacity or budget constraints. In Step 3d, the swap will be conducted if there is an improvement. Step 4 is an iterative step where the local search will stop if there is no improvement after all possible swaps based on incumbent solution have been done.

Algorithm 4: The proposed local search

Step 1: Find the facility (in S) that serves customer i, i.e. find array $\rho(i), \forall i \in I$.

Step 2: Set $\theta = 0$ (θ is the saving occurred from swapping).

Step 3: For $i = 1$ to n, $i \notin S$, do the following:

If $d_{i^*i} < z$, then do the following procedure:

a. Set $S'(j) \leftarrow S(j)$ and $H(j) \leftarrow U(j)$ $\forall j = 1, \dots, p$.
b. Set $S'(\rho(i)) \leftarrow i$ and $H(\rho(i)) \leftarrow K_i$.

c. Solve the CPCPFC-FL optimally using CPLEX with S' and H as inputs. Let z' be its objective function value and U be a set of the capacity designs for each open facility (S'). The model also obtain i' and j' where $z' = d_{i'j'}$.

d. Set $\theta = z - z'$.

e. If $\theta > 0$, and the solution is feasible, do the followings:

- Update $z = z'$, $i^* = i'$, $j^* = j'$, $S \leftarrow S'$ and $U \leftarrow U'$.
- Update array $\rho(i), \forall i \in I$.
- Go to Step 4.

End If

Step 4: If $\theta \leq 0$, then stop, otherwise go to Step 2.

Step 5: Return z, i^*, j^*, S and U.

4.2. VNS-based Matheuristic approach

Variable Neighbourhood Search (VNS) is a powerful metaheuristic method introduced by Brimberg and Mladenovic [5] for solving continuous location-allocation problems. This metaheuristic was formally formulated by Hansen and Mladenovic [14], who applied it to solve the p-median problem. VNS implementations and variants of VNS are provided in Hansen and Mladenovic [15] and Hansen et al. [16]. VNS consists of local search and neighbourhood search. The local search finds local optimality, whereas the neighbourhood search aims to escape from these local optima by systematically using a larger neighbourhood if there is no improvement, and then reverts back to the smaller one otherwise. In the VNS, the smallest neighbourhood is the one closest to the current solution, and the largest is the one furthest from the current solution (Hansen and Mladenovic, [14]). In this subsection, we propose a VNS-based method and present its main steps in Algorithm 5..

Step 3 aims to construct a relatively good initial solution. T facility configurations are generated where each facility configuration consists of p open facilities (S'). Each facility configuration is then evaluated by solving the CPCPFC-FL using an exact method (CPLEX). We take the facility configuration that yields the smallest objective function value. In Step 3a, when selecting p facility sites, each cluster contributes an open facility site, which is chosen randomly based on cumulative probability distribution described in Step 5c - 5d of Algorithm 3. The best facility configuration is then to be fed into the next step, which is the VNS algorithm.

Algorithm 5: The VNS-based matheuristic

Step 1: Define T and k_{max}. Set $z = \infty$.

Step 2: Apply Step 1 of Algorithm 1 to construct p clusters of potential facility sites.

Step 3: Do the following steps T times:

a. Select p facilities (S') from potential facility sites (J) based on cluster-based method given in Section 3.1. Determine the possible capacity designs of the p open facilities along with their corresponding fixed (opening) costs.
b. Implement CPLEX to solve the CPCPFC-FL optimally using selected open facilities (S') and their corresponding possible capacity designs and fixed costs as inputs. The outputs of the model are as follows: z', U', i' and j'.
c. If ($z' < z$), then update $z = z'$, $i^* = i'$, $j^* = j'$, $S \leftarrow S'$ and $U \leftarrow U'$.

Step 4: Update $z' = z$, $i' = i^*$, $j' = j^*$, $S' \leftarrow S$ and $U' \leftarrow U$.

Step 5: Set $k = 1$.

Step 6: Shaking procedure (Do the following step k times:)

a. Choose randomly a potential facility located at customer $\hat{\imath}$ site, say facility $\hat{\jmath}$ ($\hat{\jmath} \notin S'$), where $d_{i'\hat{\jmath}} < z'$.
b. Remove the facility that serves customer $\hat{\imath}$ from the current solution and insert facility $\hat{\jmath}$ into current solution (S').
c. Implement CPLEX to the CPCPFC-FL optimally using selected open facilities (S') and their corresponding possible capacity designs and fixed costs as inputs. Decision variables z', U', i', and j' are updated.

Step 7: Local search.

Implement the proposed local search (presented in Algorithm 4) with z', U', i' and j' as inputs and outputs.

Step 8: Move or not.

If ($z' < z$), then update $k = 1$ along with $z = z'$, $i^* = i'$, $j^* = j'$, $S \leftarrow S'$ and $U \leftarrow U'$.

Else update $k = k + 1$ along with $z' = z$, $i' = i^*$, $j' = j^*$, $S' \leftarrow S$ and $U' \leftarrow U$.

Step 9: If $k \leq k_{max}$, go back to Step 6.

Step 10: If computing time is less than cpu_{max}, then go back to Step 5.

Step 11: Return z, i^*, j^*, S, and U.

The shaking process (Step 6) in the proposed VNS is conducted by inserting a facility, say facility $\hat{j}$ ($\hat{j} \in S'$), located at a customer site randomly selected (say customer $\hat{i}$) and removing the facility that serves customer $\hat{i}$ from the current solution. Note that the distance value between facility $\hat{j}$ and customer i' (the customer who has the maximum distance to its facility) must be less than the current objective function value or $d_{i'\hat{j}} < z'$. The new facility configuration along with its possible capacity designs and its corresponding fixed costs is then evaluated by solving the CPCPFC-FL. This procedure is repeated k times.

In the local search (Step 7), the proposed algorithm presented in Algorithm 4 is implemented to improve the quality of solution by finding the local optima. In the move or not move step, if there is no improvement, a larger neighbourhood is systematically used, otherwise the search returns to the smallest neighbourhood. This can be conducted by updating k value, where $k = k_{max}$ represents the largest neighbourhood while $k = 1$ indicates the smallest one.

5. COMPUTATIONAL EXPERIMENTS

In this section, the computational experiments are presented. We carried out extensive experiments to examine the performance of the proposed heuristic approaches. These were coded in C++ .Net 2012 where we also used the IBM ILOG CPLEX version 12.6 Concert Library. The tests were run on a PC with an Intel Core i5 CPU @ 3.20GHz processor, 8.00 GB of RAM. As there is no data available in the literature for the proposed problem, we therefore generated a new dataset where $n = 50$ to 1000. The demand of each customer is randomly generated between 1 and 10. In this study, the potential facility locations are located in the customer sites i.e. $|J| = n$. Here, we set the number of possible capacities for all potential facility sites to 3 $|K_j| = 3$, $\forall j = 1, \ldots, n$. We generate randomly the capacity of each potential facility for each design $\hat{b}_{jk}$ and the fixed cost of each potential facility for using each capacity design (f_{jk}). The maximum (available) budget for opening p facilities ($\hat{c}$) is also generated, based on the average of fixed cost of all potential facility sites and the number of open facilities (p).

To evaluate the performance of our proposed solution approaches, we compare the solutions of the proposed method with solutions of the exact method (using CPLEX). Here, we limit the computing time of CPLEX to 3 hours so lower bound (LB), upper bound (UB), and %Gap are obtained. The performance of the proposed approaches will also be measured by %Dev between the Z value obtained by our proposed approaches and the best known Z (Z^*). %Dev is calculated as follows:

$$\%Dev = \frac{Z_p - Z^*}{Z^*} \cdot 100 \tag{21}$$

where Z_p refers to the objective function value of a feasible solution obtained by either the exact method (UB) or the proposed solution methods.

In the experimental study, we set parameters $\varepsilon = 0.0001$ and $\hat{T} = 5$ for clustering the potential facility sites. For the adaptive heuristic method (AMM), we set parameters $m = min\left(max(4p, 30), min(60, 0.75n)\right)$ and $T = 10$, whereas for VNS we set the parameters $T = 10$ and $k_{max} = 10$. In addition, parameter cpu_{max} for VNS is set based on the computational time required by AMM to address the problem. By doing this, we can compare the performance between two proposed methods within the same computational time. Those parameters were selected based on our preliminary experiments. Using these settings, an acceptable performance, in terms of both quality of the solution and the computational effort, is obtained.

We divide the results of the proposed methods into two tables. Table 1 presents the results on $n = 50$ to 500, whereas the results on $n = 750$ to 1000 are given in Table 2. According to the results shown in Table 1, the exact method found the optimal solution within a relatively short computing time for the small problems ($n \leq 125$). CPLEX experienced difficulties when solving the problem with $n > 125$. On average, the proposed methods (AMM and VNS) yield a better deviation than the exact method. Moreover, the proposed methods required less than 10% of the computational time needed by the exact method. The AMM performs slightly better than VNS, where the AMM and VNS produced an average deviation of 2.39% and 3.35% respectively. When $n = 50$, the AMM was able to obtain the optimal solutions for all instances except for $p = 5$. The table also reveals that for $n \leq 250$, the exact method produced a very small deviation, however the computational time required to solve the problem is huge especially when $n \geq 250$.

Table 2 shows the computational results for relatively large problems ($n \geq 750$). The table does not show the results from the exact method (CPLEX) as this method was not able to obtain the upper and lower bounds values within 3 hours. In this table we only compare the performance of the AMM and VNS. Here, VNS is found to be the best method as it produced the smallest deviation (0.1%). For all instances, VNS yields a smaller deviation than the AMM, except for $p = 25$. Figure 2 shows the optimal solution of the CPCPFC for $n = 50$ with $p = 5$ where a customer is assigned to the facility with the same colour.

			Exact Method					Matheuristics		
n	p	Best Known	Dev(%)	UB	LB	Gap(%)	CPU(s)	AMM (%Dev)	VNS (%Dev)	CPU(s)
50	5	16.03	0.00	16.03	16.03	0.00	11	1.54	1.54	49
50	10	10.77	0.00	10.77	10.77	0.00	2	0.00	0.00	10
50	15	9.22	0.00	9.22	9.22	0.00	6	0.00	6.83	13
50	20	8.54	0.00	8.54	8.54	0.00	5	0.00	20.50	5
125	5	37.01	0.00	37.01	37.01	0.00	73	1.41	0.00	23
125	10	25.32	0.00	25.32	25.32	0.00	207	1.16	1.16	20
125	15	21.26	0.00	21.26	21.26	0.00	277	2.73	4.12	63
125	20	17.46	0.00	17.46	17.46	0.00	226	0.00	4.49	77
125	25	16.03	0.00	16.03	16.03	0.00	221	0.00	2.31	157
250	5	78.00	0.00	78.00	78.00	0.00	940	2.41	1.68	48
250	10	52.20	1.84	53.16	35.66	32.91	10,800	0.00	7.17	82
250	15	40.80	0.48	41.00	32.06	21.80	10,800	7.50	0.00	161
250	20	35.74	0.00	35.74	35.73	0.01	6,243	5.34	1.09	620
250	25	31.62	0.00	31.62	31.14	1.51	10,800	1.39	1.98	1,532
500	5	157.84	251.79	555.24	108.47	80.46	10,801	3.57	0.00	473
500	10	104.96	42.99	150.08	70.05	53.33	10,801	5.41	0.00	525
500	15	83.60	149.06	208.22	54.71	73.72	10,802	5.89	0.00	1,505
500	20	75.89	663.06	579.12	46.24	92.02	10,809	7.01	0.00	872
500	25	68.12	714.87	555.07	40.52	92.70	10,801	0.00	10.74	233
Average			96.00				4,731	2.39	3.35	341

Table 1: Computational Results for $n = 50$ to 500

			Matheuristics		
n	p	Best Known	AMM(%Dev)	VNS(%Dev)	CPU(s)
750	5	232.13	1.11	0.00	665
750	10	163.17	2.41	0.00	1,581
750	15	129.77	5.08	0.00	6,566
750	20	113.32	7.95	0.00	2,650
750	25	105.12	0.00	0.46	797
1000	5	314.84	0.81	0.00	1,457
1000	10	215.23	2.21	0.00	4,223
1000	15	175.55	3.70	0.00	8,602
1000	20	155.31	4.67	0.00	4,742
1000	25	145.15	0.00	0.52	1,363
Average			2.79	0.10	3,265

Table 2: Computational Results on $n = 750$ to 1000

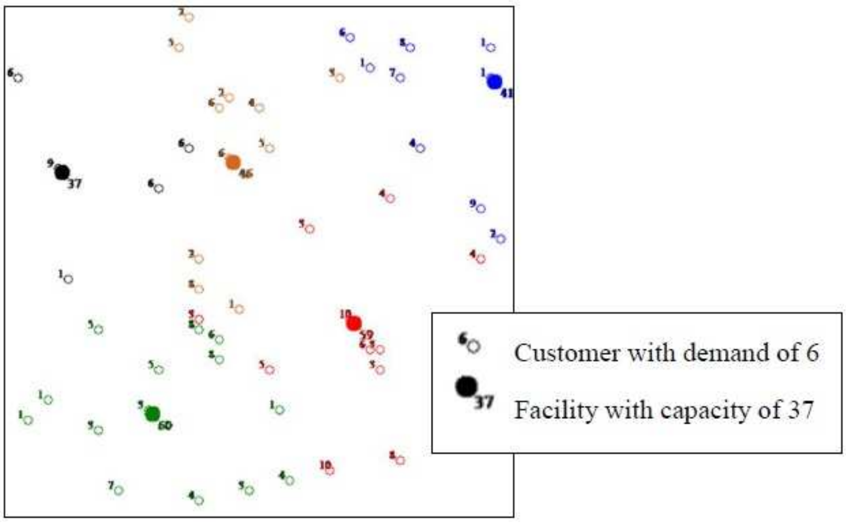

Figure 2: The optimal solution for $n = 50$ and $p = 5$

6. CONCLUSIONS and SUGGESTIONS

In this paper we investigate the single source p-center problem with the presence of limited budget available for opening facilities, considering several possible capacities for each potential facility. We refer to it as the capacitated single source

p-center problem with fixed cost and multilevel capacities (CPCPFC). A mathematical model of this problem is introduced and two solution methods for solving the problem are given. The first method, we refer to as the Adaptive Matheuristic Method (AMM), whereas the second one is a matheuristic technique based on Variable Neighbourhood Search (VNS). The proposed approaches were assessed on randomly generated datasets. For relatively small problems, the solutions of the proposed methods are compared with the solutions obtained by the exact method executed within a limited computing time (3 hour). According to the computational experiments, the proposed methods run very well and produced small deviations within a short computational time. In this case, the AMM performs better than VNS as it yields a smaller average deviation. For large problems ($n > 500$), the exact method was not able to obtain the lower and upper bounds. Therefore, we compared only the performance between the AMM and VNS. Based on the results, for relatively large problems, it is found that VNS performs slightly better than the AMM.

This research could be worthwhile expanding and adapting to its counterpart problem namely the continuous capacitated p-center problem with the presence of fixed cost and multilevel capacities.

REFERENCES

[1] Albareda-Sambola, M., Daz, J.A., and Fernández, E., "Lagrangean duals and exact solution to the capacitated p-center problem", *European Journal of Operational Research*, 201 (2010) 71-81.

[2] An, H., Bhaskara, A., Chekuri, C., Gupta, S., Madan, V., and Svensson, O., "Centrality of trees for capacitated k-center", *Mathematical Programming*, 154 (2015) 29-53.

[3] Bar-Ilan, J., Kortsarz, G., and Peleg, D., "How to allocate network centers", *Journal of Algorithms*, 15 (1993) 385-415.

[4] Bashiri, M., Mirzaei, M., and Randall, M., "Modeling fuzzy capacitated p-hub center problem and a genetic algorithm solution", *Applied Mathematical Modelling*, 37 (2013) 3513-3525.

[5] Brimberg, J., and Mladenovic, N., "A variable neighbourhood algorithm for solving the continuous location-allocation problem", *Studies of Locational Analysis*, 10 (1996) 1-12.

[6] Brimberg, J., and Salhi, S., "A Continuous Location-Allocation Problem with Zone-Dependent Fixed Cost", *Annals of Operations Research*, 136 (2005) 99-115.

[7] Chechika, S., and Peleg, D., "The fault-tolerant capacitated k-center problem", *Theoretical Computer Science*, 566 (2015) 12-25.

[8] Cooper, L., "Heuristic Methods for Location-Allocation Problems", *SIAM Review*, 6 (1964) 37-53.

[9] Cooper, L., "The transportation-location problem", *Operations Research*, 20 (1972) 94-108.

[10] Cygan, M., Hajiaghayi, M., Khuller, S., "LP rounding for k-centers with non-uniform hard capacities", *In Proceeding 53rd IEEE Symposium on Foundations of Computer Science*, (FOCS) (2012) 273-282.

[11] Elshaikh, A., Salhi, S., and Nagy, G., "The continuous p-centre problem: An investigation into variable neighbourhood search with memory", *European Journal of Operational Research*, 241 (2015) 606-621.

[12] Francis, R.L., Lowe, T.J., Rayco, M.B., and Tamir, A., "Aggregation error for location models: survey and analysis", *Annals of Operations Research*, 167 (2009) 171-208.

[13] Hakimi, S.L., "Optimum locations of switching centers and the absolute centers and medians of a graph", *Operation Research*, 12 (1964) 450-459.

[14] Hansen, P., and Mladenovic, N., "Variable neighbourhood search for the p-median", *Location Science*, 5 (1997) 207-225.

[15] Hansen, P., and Mladenovic, N., "Variable neighbourhood search: Principles and applications", *European Journal of Operational Research*, 130 (2001) 449-467.

[16] Hansen, P., Mladenovic, N., and Perez, J.A.M., "Variable neighbourhood search: methods and applications", *Annals of Operations Research*, 175 (2010) 367-407.

[17] Irawan, C.A., Salhi, S., and Scaparra, M.P., "An Adaptive Multiphase Approach for Large Unconditional and Conditional p-Median Problems", *European Journal of Operational Research*, 237 (2014) 590-605.

[18] Irawan, C.A., and Salhi, S., "Aggregation and non-aggregation techniques for large facility location problems - A survey", *Yugoslav Journal of Operations Research*, 25 (2015) 1-11.

[19] Irawan, C.A., Salhi, S., and Drezner, D., " Hybrid meta-heuristics with VNS and exact methods: application to large unconditional and conditional vertex p-centre problems", *Journal of Heuristics*, 22 (2016) 507-537.

[20] Jaeger, M., and Goldberg, J., " A polynomial algorithm for the equal capacity p-center problem on trees", *Transportation Science*, 28 (1994) 167-175.

[21] Kariv, O., and Hakimi, S.L., "An algorithmic approach for the vertex p-center problem. Part I: The p-centers", *SIAM Journal on Applied Mathematics*, 37 (1979) 513-538.

[22] Khuller, S., and Sussmann, Y.J., "The capacitated k-center problem", *SIAM Journal on Discrete Mathematics*, 13 (2000) 403-418.

[23] Mladenovic, N., Labbe, M., and Hansen, P., " Solving the p-center problem with tabu search and variable neighbourhood search", *Networks*, 42 (2003) 48-64.

[24] Őzsoy, F.A., and Pnar, M.C., " An exact algorithm for the capacitated vertex p-center problem", *Computers & Operations Research*, 33 (2006) 1420-1436.

[25] Quevedo-Orozco, D.R., and Ríos-Mercado, R.Z., " A New Heuristic for the Capacitated Vertex p-Center Problem", in: C. Bielza, A. Salemrn, A. Alonso-Betanzos, J.I. Hidalgo, L. Martnez, A. Troncoso, et al., (eds.)*Advances in artificial intelligence. Lecture notes in artificial intelligence*, Heidelberg,Germany: Springer, 8109 (2013) 279-288.

[26] Quevedo-Orozco, D.R., and Ríos-Mercado, R.Z., " Improving the quality of heuristic solutions for the capacitated vertex p-center problem through iterated greedy local search with variable neighborhood descent", *Computers & Operations Research*, 62 (2015) 133-144.

[27] Salhi, S., *Heuristic Search: The Emerging Science of Problem Solving*, Springer, Cham, Switzerland, 2017.

[28] Scaparra, M.P., Pallottino, S., Scutellá, M.G., "Large-scale local search heuristics for the capacitated vertex p-center problem", *Networks*, 43 (2004) 241-255.