Solving the bi-objective capacitated p-median problem with multilevel capacities using compromise programing and VNS

Chandra Ade Irawan[a*] , Arif Imran[b] , Martino Luis[c]

[a] *Nottingham University Business School China, University of Nottingham Ningbo China, Ningbo, China*

[b] *Department of Industrial Engineering, Institut Teknologi Nasional, Bandung 40124, Indonesia*

[c] *Othman Yeop Abdullah Graduate School of Business, Universiti Utara Malaysia*

Email :chandra.irawan@nottingham.edu.cn (Chandra Ade Irawan) ; arifimr@yahoo.com (Arif Imran) ; martino@uum.edu.my (Martino Luis)

Abstract

A bi-objective optimisation using a compromise programming approach is proposed for the capacitated *p*-median problem in the presence of the fixed cost of opening facility and several possible capacities that can be used by potential facilities. As the sum of distances between customers and their facilities and the total fixed cost for opening facilities are important aspects, the model is proposed to deal with those conflicting objectives. We develop a mathematical model using integer linear programming (ILP) to determine the optimal location of open facilities with their optimal capacity. Two approaches are designed to deal with the bi-objective capacitated *p*-median problem, namely compromise programming with an exact method and with a variable neighbourhood search based matheuristic. New sets of generated instances are used to evaluate the performance of the proposed approaches. The computational experiments show that the proposed approaches produce interesting results.

Keywords: the capacitated *p*-median problem, bi-objective, compromise programming, VNS

---

1. Introduction

The aim of the *p*-median problem (PMP) is to seek the location of *p* facilities among *m* discrete potential sites in such a way to minimise the sum of the distances between customers and their associated facilities. The PMP was originally formulated by ReVelle and Swain (1970). This problem is also known as the minisum location problem which is categorised as NP-hard (Kariv and Hakimi 1979). In the capacitated version of the *p*-median problem (CPMP), each customer

[*] Corresponding Author : chandra.irawan@nottingham.edu.cn

has a fixed demand where each potential facility has a known capacity. Each facility must serve the demand of its customers without violating its capacity. This capacity constraint significantly multiplies the complexity of the problem. Therefore, CPMP falls into NP-hard problems (Garey and Johnson 1979).

In many real case applications, when finding the best location for the facilities, the fixed cost for opening facilities is usually taken into account. The fixed cost of a potential facility may be dependent on its location and its capacity. One possible decision that has to be made is to determine the optimal location of open facilities and their corresponding capacity in order to minimise the sum of distances between facilities and their associated customers. As there is always a limited budget available, the decision makers should consider the total fixed cost for opening facilities. In the literature these two objectives are usually combined though both objectives may appear conflicting. Beheshtifar and Alimoahmmadi (2015) integrated geographical information system analysis with a genetic algorithm to locate optimum sites of new clinics using multiobjective criteria. Rath et al. (2016) proposed a two-stage bi-objective stochastic programming models to find depot locations in disaster relief operations. In their works, two objectives were used, i.e., monetary and humanitarian objectives. When unlimited budget is available, the open facilities will use a large capacity to ensure that customers will be served by their nearest facilities. In this case, the problem may be considered as the uncapacitated *p*-median problem.

In this paper, we investigate the capacitated *p*-median problem in the presence of two conflicting objectives. To the best of our knowledge, there is no paper in the literature studying such a problem.

The main contributions of this paper are as follows:

- to develop a new mathematical model for the bi-objective capacitated *p*-median problem using compromise programming method,
- to propose an effective variable neighbourhood search (VNS) to solve the bi-objective CPMP.

The paper is organised as follows. Section 2 provides a brief review of the past efforts at the capacitated *p*-median problem. In Section 3, mathematical models for the classical CPMP along with the new bi-objective CPMP are presented. A description of compromise programming method for solving the bi-objective CPMP is given in Section 4. The proposed

VNS to solve the problem is described in Section 5. Section 6 presents computational results using generated dataset. A summary of our findings and some avenues for future research are also provided in the last section.

2. Literature Review

The earliest works on the CPMP were done by Mulvey and Beck (1984) who designed two algorithms to tackle capacitated clustering problems and Pirkul (1987) who used the Lagrangian relaxation technique to solve communication networks deployment problems. Osman and Christofides (1994) integrated simulated annealing and tabu search to deal with the CPMP. Maniezzo et al. (1998) studied the CPMP by proposing a bionomic algorithm and an effective local search.

Baldacci et al. (2002) dealt with the CPMP using a set partitioning formulation technique. The proposed technique was tested on benchmark instances from the literature and also on new sets of instances which the authors generated by considering bounds on the cluster cardinality and incompatibilities between entities. Lorena and Senne (2004) applied a column-generation method to solve the CPMP by incorporating the Lagrangean/surrogate relaxation to determine new bounds and new productive columns through a modified knapsack subproblem. Ahmadi and Osman (2005) integrated the Greedy Random Adaptive Search Procedure and the Adaptive Memory Programming (AMP) to create a greedy random adaptive memory search method in tackling the CPMP.

Scheuerer and Wendolsky (2006) addressed the CPMP by proposing a scatter search heuristic. Several new best found solutions were obtained when the proposed scatter search heuristic was tested on benchmark instances from the literature. Díaz and Fernàndez (2006) hybridised a scatter search and path relinking algorithm for solving the CPMP. Fleszar and Hindi (2008) designed an effective variable neighborhood search to deal with the CPMP. Chaves et al. (2007) suggested a new hybrid heuristic known as clustering search for the CPMP. Boccia et al. (2007) developed a cutting plane algorithm based on Fenchel cuts to reduce considerably the integrality gap in solving hard CPMP instances.

Landa-Torres et al. (2012) put forward two new evolutionary algorithms based on genetic algorithms and harmony search approach. A grouping encoding procedure is introduced within both algorithms to guide the search and a unique local search based on swapping approach is applied to improve the solutions. Their experiments show that their results outperform the published evolutionary techniques. Yaghini et al. (2013a) hybridised a cutting-plane neighbourhood structure and tabu search to solve the CPMP. In the neighbourhood structure, three strategies were developed to choose an open median to be closed. In the following research, Yaghini et al. (2013b) proposed an efficient heuristic by integrating the local branching and relaxation induced neighbourhood search methods to deal with the CPMP.

Stefanello et al. (2015) developed a three stage matheuristic algorithm known as the Iterated Reduction Matheuristic Algorithm (IRMA) to tackle the CPMP. El Amrani et al. (2016) studied the CPMP by introducing a budget constraint into the problem. Three techniques were proposed to solve the problem i.e., a branch and cut algorithm, greatest customer demand first, and large neighbourhood search.

In the literature, there were attempts to hybridise heuristic and mathematical programming approaches, known as matheuristics. This technique has been successfully used to solve hard combinatorial problems, see for instance, the works of Büdenbender et al. (2000), Taillard and Voss (2002), Talbi (2002), Dumitrescu and Stützle (2003), Puchinger and Raidl (2005), Fernandes and Lourenço (2006), Jourdan et al. (2009), Fanjul-Peyro and Ruiz (2011), and Stefanello et al. (2015). A comprehensive review on matheuristics is provided by Maniezzo et al. (2010) and Salhi (2017). As this technique has been proven to be effective and efficient to solve NP-hard problems, in this research, we propose a matheuristic technique to solve the bi-objective CPMP.

3. Problem Formulation

In this section we first present the mathematical model of the capacitated *p*-median problem (CPMP) followed by the proposed model for the bi-objective capacitated *p*- median problem with the presence of fixed cost and multilevel capacities (bi-objective CPMP).

The capacitated p-median problem (CPMP)

The following notations are used to describe the sets, parameters and decision variables of the CPMP.

Sets

I set of customers $(i \in I = \{1,...,n\}, n = |I|)$

J set of potential sites $(j \in J = \{1,...,m\}, m = |J|)$

Parameters

d_{ij} the distance between customer $i \in I$ and facility $j \in J$

w_i the demand of customer $i \in I$

b_j the capacity of a facility located on site $j \in J$

p the number of open facilities

Decision variables

$$Y_j = \begin{cases} 1, & \text{if a facility is located at site } j \in J \\ 0, & \text{otherwise} \end{cases}$$

$$X_{ij} = \begin{cases} 1, & \text{if customer } i \in I \text{ is assigned to facility } j \in J \\ 0, & \text{otherwise} \end{cases}$$

The CPMP can be mathematically formulated as an Integer Linear Programming (ILP) as follows:

$$\text{Minimise} \quad \sum_{i \in I} \sum_{j \in J} w_i \cdot d_{ij} \cdot X_{ij} \tag{1}$$

Subject to

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

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

$$\sum_{i \in I} w_i \cdot X_{ij} \le b_j \cdot Y_j, \qquad \forall j \in J \tag{4}$$

$$X_{ij} - Y_j \le 0, \qquad \forall i \in I, j \in J \tag{5}$$

$$Y_j \in \{0,1\} \qquad \forall j \in J \tag{6}$$

$$X_{ij} \in \{0,1\} \qquad \forall i \in I, j \in J \tag{7}$$

The objective function (1) aims to minimise the sum weighted distance between open facilities and their associated customers which we refer to as “total distances” in this paper. Constraints

(2) ensure that each customer is assigned to exactly one facility. Constraint (3) imposes that there must be p open facilities. Constraints (4) state that the sum of the demands of the customers assigned to each facility does not exceed its capacity. Constraints (5) prevent the assignment of customers to unopened facilities. Constraints (6) and (7) state the integrality conditions of the decision variables.

The bi-objective capacitated p-median problem (bi-objective CPMP)

In this subsection, the mathematical model of the bi-objective CPMP is presented where the presence of fixed cost and multilevel capacities are taken into account. In the new model, the capacity of open facilities is treated as a decision variable. Each potential facility has a set of possible capacities and the fixed cost for opening a facility is dependent on the facility location and the capacity used by the facility. The notations used for sets, parameters and decision variables in the proposed model are similar to the ones presented in the previous model with some additions described as follows:

Set

R_j the set of capacity designs for facility $j \in J$.

Parameters

$\hat{f}_{jr}$ the fixed cost of potential facility j using capacity r ($r \in R_j$, $j \in J$)

$\hat{b}_{jr}$ the amount of customers' demand that can be served by potential facility j using capacity r ($r \in R_j$, $j \in J$)

Decision Variables

$$X_{ij} = \begin{cases} 1, \text{ if customer } i \text{ is assigned to facility } j \\ 0, \text{otherwise} \end{cases}$$

$$\hat{Y}_{jr} = \begin{cases} 1, \text{ if a facility is located at site } j \text{ using capacity design } r; \\ 0, \text{otherwise} \end{cases}$$

The bi-objective CPMP is much harder to solve than the classical model as the proposed model optimises both facilities' location and their corresponding capacity. The problem can be modelled as a bi-objective optimisation model as follows:

Minimise

$$Z_d = \sum_{i \in I} \sum_{j \in J} w_i \cdot d_{ij} \cdot X_{ij} \tag{8}$$

$$Z_c = \sum_{j \in J} \sum_{r \in R_j} \hat{f}_{jr} \cdot \hat{Y}_{jr} \tag{9}$$

Subject to

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

$$\sum_{k \in R_j} \hat{Y}_{jr} \le 1, \qquad \forall j \in J \tag{11}$$

$$\sum_{j \in J} \sum_{r \in R_j} \hat{Y}_{jr} = p \tag{12}$$

$$\sum_{i \in I} X_{ij} \cdot w_i \le \sum_{r \in R_j} \left(\hat{b}_{jr} \cdot \hat{Y}_{jr} \right), \qquad \forall j \in J \tag{13}$$

$$X_{ij} - \sum_{r \in R_j} \hat{Y}_{jr} \le 0, \qquad \forall i \in I, j \in J \tag{14}$$

$$\hat{Y}_{jr} \in \{0,1\} \qquad \forall j \in J, r \in R_j \tag{15}$$

$$X_{ij} \in \{0,1\} \qquad \forall i \in I, j \in J \tag{16}$$

The proposed model considers two objectives which contradict each other. The first objective function (8) is the same as the previous model, whereas the second one (9) is to minimise the total fixed cost for opening facilities. Note that in Dumitrescu and Stützle (2003) and El Amrani *et al.* (2016) both objectives are added to make one single objective function. Here we treat the two objectives separately within a bi-objective methodology. Constraints (10) ensure that each customer must be satisfied by one facility, whereas Constraints (11) guarantee that each open facility uses one capacity only. Constraint (12) states that the number of open facilities is set to *p*. Constraints (13) indicate that the capacity constraints for each facility. Constraints (14) ensure that each customer can only be served by an open facility. Constraints (15) and (16) define the integrality conditions of the decision variables.

4. Compromise programming for the bi-objective CPMP

Multi-objective problems can be solved by several methods including Pareto efficient set generation, compromise programming and goal programming. In this paper, we use Compromise Programming (CP) to tackle the capacitated *p*-median problem in the presence of two conflicting objectives. CP was officially introduced by Yu (1973) to solve group decision problems and Zeleny (1974) to tackle multiple attribute decision analysis. Romero et al. (1998) showed that CP works well for bi-objective problems. As indicated by Romero and Rehman (1989), this technique aims to choose a solution from the set of efficient solutions based on a reasonable assumption that any decision maker seeks a solution as close as possible to the ideal point. A brief explanation on how the CP works can be found in Gan et al. (1996).

In this method, a distance function is applied to measure the closeness between a solution and the ideal point where a group of L_p metrics is generally used. The general formulation of a CP approach is stated as follows:

$$Min\ L_p = \left(\sum_{i=1}^{n} \left| \alpha_i \cdot \frac{Z_i(x) - Z_i^*}{Z_{i*} - Z_i^*} \right|^p \right)^{\frac{1}{p}} \tag{17}$$

where

p	the distance measure with p in range $[1,\infty]$,
n	the number of objectives,
Z_i^*	the ideal solution of objective *i*,
Z_{i*}	the anti-ideal solution of objective *i*,
$Z_i(x)$	the compromise solution that minimises L_p ,
α_i	the weight/importance of objective *i* relative to the other objectives.

In this study, as the problem is bi-objective, we set the value of p to 1 and ∞. This will allow the calculation of all intermediate compromise set points. When $p = 1$, Equation (17) takes the following form:

$$Min\ L_1 = Min \sum_{i=1}^{n} \alpha_i \cdot \frac{\left| Z_i(x) - Z_i^* \right|}{\left| Z_{i*} - Z_i^* \right|} \tag{18}$$

whereas if $p = \infty$, the objective function (17) aims to minimise the maximum deviation (π) as follows:

$$Min\ L_\infty = Min\ \pi \tag{19}$$

$$s.t.\quad \alpha_i \cdot \frac{\left|Z_i(x) - Z_i^*\right|}{\left|Z_{i*} - Z_i^*\right|} \le \pi\ ,\forall i = 1,\ldots,n \tag{20}$$

Figure 1 shows the main steps of the CP for solving the bi-objective CPMP which consists of three stages. The first stage is to find the anti-ideal solution for each objective. Here, for each objective, the maximising problem is used instead of minimising.

When solving maximising total fixed cost problem (Equation 9), we may only consider Constraints 11, 12 and 16 as the assignment of customers to their facilities may not be required. It is common that a facility with a larger capacity has a larger fixed cost. When solving the maximising total fixed cost problem, the optimiser will select the facilities with large fixed cost and large capacity. This will guarantee that the customers can be served by the open facilities. The maximising problem is relatively easy to solve by an exact method. The second stage is to obtain the ideal solution by optimising each objective (total distances and total fixed cost) separately subject to constraints 10 to 16. In this study, this problem is solved by an exact method (CPLEX).

In Stage 3, the objective is to seek the solutions that minimise L_1 and L_∞ as the compromise solutions are bounded by L_1 and L_∞. In this study, the ILPs for L_1 and L_∞ will be addressed using the exact method (CPLEX) and the proposed VNS based matheuristic. We propose the VNS based matheuristic for solving minimising L_1 and L_∞ problems as these problems are very hard to solve especially for relatively large problems.

Stage 1

a. Using CPLEX, solve maximising (instead of minimising) total distances problem (Equation 8) subject to constraints 10 to 16. Let Z_{d*} be the anti-ideal total distances.

b. Using CPLEX, solve maximising (instead of minimising) total fixed cost problem (Equation 9) subject to constraints 11, 12 and 15. Let Z_{c*} be the anti-ideal total fixed cost.

Stage 2

a. Using CPLEX, solve minimising sum distances problem (Equation 8) subject to constraints 10 to 16. Let Z_d^* be the ideal total distances.

b. Using CPLEX, solve minimising total fixed cost problem (Equation 9) subject to constraints 10 to 16. Let Z_c^* be the ideal total fixed cost.

Stage 3

a. Using CPLEX/VNS, solve minimising L_1 problem subject to constraints 10 to 16 where

$$L_1 = \frac{\alpha \cdot ((\sum_{i \in I} \sum_{j \in J} w_i \cdot d_{ij} \cdot X_{ij}) - Z_d^*)}{Z_{d*} - Z_d^*} + \frac{(1-\alpha) \cdot ((\sum_{j \in J} \sum_{r \in R_j} \hat{f}_{jr} \cdot \hat{Y}_{jr}) - Z_c^*)}{Z_{c*} - Z_c^*} \tag{21}$$

and α is the weight (parameter) of the first objective (total distances). Let Z_d^1 denote the total distances obtained and Z_c^1 the total fixed cost.

b. Using CPLEX/VNS, solve minimising L_∞ problem where

$$L_\infty = \pi \tag{22}$$

subject to constraints 10 to 16 with additional constraints as follow:

$$\frac{\alpha \cdot ((\sum_{i \in I} \sum_{j \in J} w_i \cdot d_{ij} \cdot X_{ij}) - Z_d^*)}{Z_{d*} - Z_d^*} \le \pi \tag{23}$$

$$\frac{(1-\alpha) \cdot ((\sum_{j \in J} \sum_{r \in R_j} \hat{f}_{jr} \cdot \hat{Y}_{jr}) - Z_c^*)}{Z_{c*} - Z_c^*} \le \pi \tag{24}$$

Let Z_d^∞ be the total distances obtained and Z_c^∞ the total fixed cost.

c. Compromise solutions are bounded by L_1 and L_∞.

Fig. 1. The procedure of CP for solving the bi-objective CPMP

Figure 2 explains compromise solutions for the bi-objective CPMP. Points A and B are the ideal solutions for minimising total fixed cost and minimising total distances problems respectively. Point E is the anti-ideal or nadir point. All compromise solutions are bounded by Points C and D. The decision maker will pick from within this solution set based on their individual preferences.

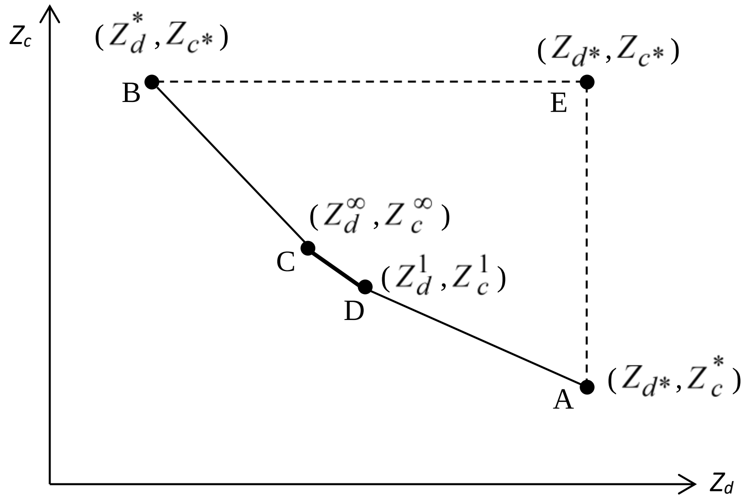

Fig. 2. Compromise solutions in the bi-objective CPMP

5. The VNS based matheuristic for solving L_1 and L_∞ problems

Brimberg and Mladenović (1996) introduced a powerful metaheuristic method called variable neighbourhood search (VNS) for solving continuous location-allocation problems. Hansen and Mladenović (1997) first formally formulated this metaheuristic to solve the *p*-median problem. VNS and its extensions have been successfully implemented to solve various optimisation problems, such as vehicle routing problems, batching problems, polyphonic sheet music, among others (Vidović et al., 2017; Kammoun et al., 2016; Menéndez et al., 2017; Balliauw et al., 2017). Hansen and Mladenović (2001) and Hansen *et al*. (2010) provided VNS implementations and variants of VNS. VNS comprises local search and neighbourhood search. The local search seeks local optimality while the neighbourhood search aims to escape from these local optima by systematically using a larger neighbourhood if no improvement is found and then reverts back to the smaller one otherwise. In the VNS, the smallest neighbourhood is the one that is closest to the current solution, whereas the largest one farthest from the current solution (Hansen and Mladenovic, 1997). Figure 3 presents the main steps of the procedure of our VNS based matheuristic.

In the first step, the parameters required in the proposed matheuristic method are defined. This includes the number of iterations (*T*) for solving the aggregated problems, the number of aggregated potential facilities (μ), the maximum computational time ($cpu_{\max}$), the maximum computational time for CPLEX to solve the aggregated problems (τ and τ') and the parameter that indicates the largest neighbourhood from incumbent solutions used by the proposed VNS ($k_{\max}$).

The second step is an aggregation approach where an iterative process is conducted. Aggregation technique is usually used for solving location problems in the present/presence of a large number of demand points (customers). It may be impossible and time consuming to solve optimally the large location problems. The main idea behind the aggregation is to reduce the number of customers or potential facilities to be small enough so an optimiser can be used. Here, the problem is partitioned into smaller problems and can be solved within a reasonable amount of computing time. However, this aggregation may reduce the accuracy of the model as this aggregation introduces error in the data used by location models and models output. The aggregation approach has shown to be promising when solving large *p*-median (Irawan *et al.*, 2014; Irawan and Salhi, 2015a) and *p*-centre problems (Irawan *et al.*, 2016). A review on the aggregation method for large facility location problems is provided by Irawan and Salhi (2015b).

Procedure VNS-based

1. Define T, μ, $cpu_{\max}$, τ , τ' and $k_{\max}$. Set $z = \infty$.
2. Do the following steps T times:
   a. Aggregate m to μ potential facility sites using random approach and by including the facility locations in the incumbent solution (S).
   b. Solve the aggregated L_1 and L_∞ problems using the exact method (CPLEX) within τ seconds. Let z' be its objective function value with S' and U' as vectors of the obtained facility configuration and their corresponding capacity design respectively.
   a. If $z' < z$ then set $z = z'$, $S \leftarrow S'$ and $U \leftarrow U'$.
3. Update $z' = z$, $S' \leftarrow S$, and $U' \leftarrow U$
4. Set k = 1
5. Shaking Procedure.
   Do the following step k times:
   a. Choose randomly a potential facility, say facility $\hat{j}$ $(\hat{j} \notin S')$. Pick randomly the capacity for facility $\hat{j}$, say capacity $\hat{r}$.
   b. Remove a facility (the nearest form facility $\hat{j}$) from the current solution and insert facility $\hat{j}$ into current solution S' and update $U'_{\hat{j}} = \hat{r}$.
   c. Calculate the total capacity of facilities. If it is less than the total demand then go back to Step 5(a). Calculate the total fixed cost for opening facilities (Z'_c) based on S' and U'
   d. Implement CPLEX to solve the assignment problem (GAP) within τ' seconds based on S' and U' so the total distance (Z'_d) is obtained. Calculate z' (for L_1 or L_∞ problems) based on Z'_c and Z'_d .
6. Local search
   Implement the proposed local search (presented in Figure 4) with z' , S' , U' , Z'_c and Z'_d as inputs and outputs.
7. Move or Not
   If $z' < z$ then
   Update k = 1 along with $z = z'$, $Z_d = Z'_d$, $Z_c = Z'_c$, $S \leftarrow S'$, and $U \leftarrow U'$
   Else
   Update k = k + 1 along with $z' = z$, $Z'_d = Z_d$, $Z'_c = Z_c$, $S' \leftarrow S$, and $U' \leftarrow U$
8. If computing time is greater than $cpu_{\max}$ then go to Step 10
9. If $k \le k_{\max}$ go back to Step 5
10. Return z, Z_c , Z_d , S , and U .

Fig. 3. The procedure of the proposed VNS-based matheuristic

In this paper, the second step aims to generate a relatively good initial solution using aggregation approach. This step incorporates potential facility sites aggregation and the use of the exact method (CPLEX). Firstly, we select randomly μ potential sites out of m sites. When selecting the aggregated potential facility sites, the aggregation includes the facility sites found in the previous iteration (the best solution). In the first iteration, for the minimising L_1 problem the best solution is from the solution produced by Stage 2b of Figure 1 whereas for the minimising L_∞ problem it is from the solution generated by minimising L_1 problem (using VSN). The aggregated problem (minimising L_1 or L_∞) consisting of n customers and μ (instead of m) potential facility sites is then solved by CPLEX within τ seconds. The sets of facility sites and their corresponding capacity design are denoted by S and U respectively. The obtained solution 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 which is the VNS algorithm.

In the proposed VNS, the shaking process (Step 5) is conducted by inserting a randomly selected facility, say facility $\hat{j}$ $(\hat{j} \notin S')$, and removing a facility (the nearest facility form facility $\hat{j}$) from the current solution. Note that, the capacity for facility $\hat{j}$ is also randomly selected, say capacity $\hat{r}$ $(\hat{r} \in R_{\hat{j}})$. The total capacity of selected facilities in the current solution must be greater than the total customer demand. As the capacity of each selected facility is known, the total fixed cost of opening facilities (Z_c) can be calculated. To calculate the objective function value (z) of the new solution (for minimising L_1 or L_∞), the total distance (Z_d) must be determined. This can be achieved by solving the generalised assignment problem (GAP) which can be solved by CPLEX. In this study, we limit the computational time for CPLEX to solve the GAP to τ' seconds. The mathematical formulation of the GAP is expressed in Equation (25) – (28).

$$Minimise \quad Z_d = \sum_{j \in S} \sum_{i \in I} \left(X_{ij} \cdot d_{ij} \cdot w_i \right) \tag{25}$$

Subject to

$$\sum_{j \in S} X_{ij} = 1, \qquad \forall i \in I \tag{26}$$

$$\sum_{i \in I} X_{ij} \cdot w_i \le \hat{b}_{jr}, \qquad \forall j \in S \tag{27}$$

$$X_{ij} \in \{0,1\} \qquad \forall i \in I, j \in S \tag{28}$$

The GAP is still relatively difficult to solve due to the binary nature of the decision variable (X_{ij}). In Constraints (27), capacity $\hat{b}_{jr}$ is fixed as index *r* is element of set *U*. The shaking procedure is repeated *k* times.

In Step 6, the proposed local search is put forward to improve the quality of solution by finding the local optima. The description of our proposed local search is provided in next subsection. In Step 7 of the algorithm (Move or Not), if local search is not able to improve the solution, a larger neighbourhood is systematically used otherwise the smallest one will be used. This can be performed by updating the value of *k* where $k = k_{\max}$ indicates the largest neighbourhood while $k = 1$ represents the smallest one. In the VNS, the smallest neighbourhood is the one that is closest to the current solution, whereas the largest one farthest from the current solution (Hansen and Mladenovic, 1997).

The proposed local search

The proposed local search is designed based on the interchange heuristic using a first improvement strategy. The main steps of the proposed local search are presented in Figure 4. The algorithm aims to seek a facility location site along with its capacity to be swapped with a facility site used in the current solution. The swap will be done if improvement occurs. First, the average distance ($\hat{d}$) between facilities in the current solution (*S*) is calculated. This distance will be a criterion whether a facility site in current solution can be swapped with a potential facility site or not. If the distance between these two facility sites is greater than $\hat{d}$ then the swap process will not be conducted. This is to save the computational time where a facility in current solution can be only swapped with the one near to this facility. Salhi (2017) also pointed out that the use and design of neighbourhood reduction is found to be promising.

In Step 2c of Figure 4, each facility of a potential facility to be inserted in the solution is evaluated. First, the total capacity of the facilities in the new solution must be large enough to serve customers demand. The total fixed cost (Z_c) is then determined. The total distance (Z_d) is obtained by solving the GAP using CPLEX within τ' seconds. The objective function value z (for L_1 or L_∞ problems) can be calculated based on Z_c and Z_d. The best capacity for the potential facility is the one that yields the smallest objective function value. In Step 2d, the swap will be performed if an improvement occurs. The local search process will be restarted from the beginning once an improvement is made.

Procedure LocalSearch (z, Z_c, Z_d, S, and U **)**

1. Calculate the distance criteria $\hat{d}$ (the average distance between facilities in current solution)
2. For each potential facility $\hat{j} \in J$, $\hat{j} \notin S$, do the following:

    For each facility $j \in S$ (current solution) do the following procedure:

    a. If $d_{j,\hat{j}} > \hat{d}$ then continue (skip following steps under loop *j*)

    b. Set $z' \leftarrow \infty$, $S' \leftarrow S$, and $U' \leftarrow U$

    c. For each capacity $r \in R_{\hat{j}}$ do the following steps:

    - Set $S'' \leftarrow S'$ and $U'' \leftarrow U'$. Update $U''_{\hat{j}} = r$
    - Calculate the total capacity of facilities. If it is less than the total demand then continue (skip the following steps under loop *r*).
    - Calculate the total fixed cost (Z''_c) based on S'' and U''.
    - Solve the GAP using CPLEX within τ' seconds based on S'' and U'' so the total distance (Z''_d) is obtained. Calculate z'' (for L_1 or L_∞ problems) based on Z''_c and Z''_d.
    - If $z'' < z'$ and the solution is feasible, do the followings:
        - Update $z' = z''$, $Z'_d = Z''_d$, $Z'_c = Z''_c$, $S' \leftarrow S''$, and $U' \leftarrow U''$

    End for *r*

    d. If $z' < z$ do the followings:

    - Update $z = z'$, $Z_d = Z'_d$, $Z_c = Z'_c$, $S \leftarrow S'$, and $U \leftarrow U'$
    - Go to Step 1

    End for *j*

    End for $\hat{j}$
3. Return z, Z_c, Z_d, S, and U.

Fig. 4. The main steps of the proposed local search

6. Computational Study

We carried out extensive experiments to examine the performance of the proposed solution method. This was coded in C++ .Net 2012 where the IBM ILOG CPLEX version 12.63 Concert Library was also used to solve the problems with an exact method. The computational experiments were conducted on a PC with an Intel Core i5 CPU @ 3.20GHz processor, 8.00 GB of RAM. To the best of our knowledge, there is no benchmark dataset available in the literature for the proposed problem; hence we constructed four new datasets with $n = 150$ to 600 with an increment of 150. In this experiment, the potential facility locations are located in the customer sites i.e. $|J| = n$. Each potential facility has three possible capacities $(|R_j| = 3, j \in J)$ where the values of $\hat{b}_{jr}$ (the amount of customers' demand that can be served by potential facility *j* using capacity *r*) and $\hat{f}_{jr}$ (the fixed cost of potential facility *j* using capacity *r*) are randomly generated. The demand of each customer is also randomly generated in the range of [1, 10]. The value of *p* varies from 10 to 30 with an increment of 5.

Table 1 presents the computational results in obtaining ideal and anti-ideal solutions (Stages 1 and 2 of Figure 1) using the exact method (CPLEX 12.63). The table shows that all instances can be solved optimally by CPLEX within a relatively short computational time. The maximising total cost problem is very easy to solve as there is no assignment task in this problem. On the other hand, the minimising total distance is relatively hard to solve. In general, when the value of *n* increases, the computational time needed to solve the problems also grows significantly.

The compromise solutions are obtained by solving minimising L_1 and L_∞ problems. In this experiment, we set the weight of the first objective (total distances) to 0.5. This means that the weight of the second objective (total fixed cost) is also 0.5. These problems are very hard to solve by the exact method. Therefore, we propose the VNS based matheuristic. To evaluate the performance of our proposed matheuristic, 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 2 hours for each problem (minimising L_1 and L_∞ problems). The upper bound value obtained from CPLEX for each problem is treated as an objective function value of the exact method. In order to assess the performance of the proposed methods, the deviations (Dev) between the *z* value obtained by our proposed matheuristic and the best *z** (the best objective function value that can be produced by either the exact method or the proposed matheuristic) are calculated using the following formula:

$$Dev = \frac{z_p - z^*}{z^*} \tag{29}$$

where z_p refers to the objective function value of the feasible solution obtained by either the exact method or the proposed solution methods.

Table 1
Computational results in obtaining ideal and anti-ideal solutions

		Total Distance				Total Cost			
n	p	Minimising Problem		Maximising Problem		Minimising Problem		Maximising Problem	
		Z_d^*	CPU (s)	Z_{d*}	CPU (s)	Z_c^*	CPU (s)	Z_{c*}	CPU (s)
150	10	13,776	1.14	134,449	0.88	60,933	0.08	155,658	0.00
	15	10,472	1.09	134,450	0.97	62,793	3.10	232,702	0.02
	20	8,245	1.15	134,450	1.14	63,476	0.08	309,432	0.01
	25	6,717	0.88	134,449	0.85	65,739	8.40	385,850	0.01
	30	5,729	1.04	134,449	0.83	67,776	489.21	461,486	0.01
300	10	52,956	15.98	480,801	6.68	107,644	0.32	277,821	0.01
	15	41,853	11.48	480,801	6.60	108,993	0.17	415,476	0.02
	20	35,117	12.17	480,801	6.33	112,528	1,076.34	553,131	0.02
	25	30,463	9.92	480,801	6.13	114,053	3.11	690,508	0.00
	30	26,727	9.22	480,801	6.30	115,754	0.06	826,773	0.00
450	10	124,994	107.87	1,138,028	40.38	170,684	0.77	441,000	0.02
	15	98,484	97.02	1,138,028	40.65	172,262	0.11	660,618	0.02
	20	83,386	76.15	1,138,028	40.45	177,711	24.92	878,913	0.02
	25	72,860	73.59	1,138,028	40.80	179,928	0.13	1,097,208	0.02
	30	64,792	69.43	1,138,028	40.95	182,434	0.06	1,314,180	0.02
600	10	232,953	827.97	2,017,681	91.84	227,974	0.17	589,500	0.00
	15	186,715	534.98	2,017,681	92.52	230,117	0.12	883,070	0.02
	20	156,587	425.35	2,017,681	92.10	237,031	5.38	1,174,870	0.00
	25	135,376	258.96	2,017,681	92.42	239,307	0.16	1,466,670	0.00
	30	121,149	198.60	2,017,681	92.95	241,902	0.05	1,758,470	0.00
Average			136.70		35.09		80.64		0.01

In the proposed VNS based matheuristic, we set parameters T = 10, μ = min(50,4p), $cpu_{\max} = 40p$, $\tau = 2p$, $\tau' = 0.5$ and $k_{\max} = 10$. Those parameters were chosen based on our preliminary experiments. The values of T, μ and τ affect the quality of the initial solution produced. The higher the values of T, μ and τ, the higher is the chance of getting a better initial solution. However, the computing time increases with increasing values of T, μ and τ.

Tables 2 and 3 present computational results in obtaining compromise solutions using the exact method and the VNS based matheuristic respectively. In the tables, the first four columns refer to the number of customers/potential facilities, the number of open facilities, the best L_1 and the best L_∞. The next two blocks of five columns each refer to the objective function value (L_1 / L_∞), Deviation (Dev), the total distance (Z_d^1 / Z_d^∞), the total cost (Z_c^1 / Z_c^∞), and the CPU time in seconds. The **bold** face in the tables refers to the optimal solutions. According to Table 2, within 2 hours, CPLEX was able to guarantee optimality only for one instance in the minimising L_1 problem and two instances in the minimising L_∞ problem.

Based on the average deviation in the tables, the VNS based matheuristic performs better than the exact method in obtaining the compromise solutions, especially for the large problems. For the minimising L_1 problem, the proposed method yields an average deviation of 0.1075 whereas the exact method produces 0.2670. For the minimising L_∞ problem, the proposed method and the exact method produce deviation of 0.0250 and 3.6985 respectively. In summary, the VNS based matheuristic is found to be the best performer for obtaining the compromise solutions as it produced the smallest deviation. Moreover, the VNS based matheuristic runs much faster than the exact method as the exact method is approximately eight time times longer compared to the VNS based matheuristic.

Based on the results from Tables 2 and 3, it can be noted that the solutions obtained by either CP with exact method or with VNS based matheuristic for L_1 and L_∞ are quite close to each other. As all the compromise solutions are bounded by L_1 and L_∞, they are not much different from one another. Prior justification for choosing a solution on the compromise set bounded by L_1 and L_∞ is needed. For example, if the decision maker thinks that total cost is more important, then the solution generated by minimising L_1 is a desirable choice.

Table 2

Computational results in obtaining compromise solutions using the exact method

n	p	*The best* L_1	*The best* L_∞	Exact Method									
				Minimising L_1 problem					Minimising L_∞ problem				
				L_1	Dev	Z_d^1	Z_c^1	CPU (s)	L_∞	Dev	Z_d^∞	Z_c^∞	CPU (s)
150	10	0.01612	0.01187	**0.01612**	0.0000	17,162	61,329	425	**0.01187**	0.0000	16,641	63,178	2,734
	15	0.01590	0.00921	0.01590	0.0000	13,291	64,334	7,212	**0.00921**	0.0000	12,756	65,915	5,756
	20	0.01513	0.00880	0.01513	0.0000	11,026	65,498	7,202	0.00880	0.0000	10,445	67,803	7,232
	25	0.01085	0.00561	0.01102	0.0152	8,023	69,519	7,236	0.00561	0.0000	8,147	69,332	7,202
	30	0.01055	0.00566	0.01055	0.0000	7,461	70,784	7,220	0.00582	0.0280	7,225	72,357	7,205
300	10	0.02606	0.01637	0.02606	0.0000	71,793	109,020	7,200	0.01637	0.0000	66,967	112,723	7,200
	15	0.01728	0.01093	0.01728	0.0000	53,334	111,571	7,200	0.01093	0.0000	51,453	115,688	7,200
	20	0.01287	0.00757	0.01287	0.0000	43,965	115,121	7,206	0.00828	0.0946	42,501	119,648	7,200
	25	0.00769	0.00455	0.00769	0.0000	34,167	118,184	7,207	0.00485	0.0671	34,833	119,383	7,203
	30	0.00700	0.00473	0.00700	0.0000	30,847	119,256	7,206	0.00473	0.0000	31,020	122,482	7,200
450	10	0.02252	0.01666	0.02513	0.1157	171,638	171,823	7,201	0.01968	0.1814	164,873	177,982	7,201
	15	0.01768	0.01391	0.01768	0.0000	131,638	173,956	7,203	0.01832	0.3177	136,581	188,943	7,201
	20	0.01109	0.00842	0.01109	0.0000	103,180	180,100	7,201	0.00842	0.0000	99,953	189,516	7,201
	25	0.00880	0.00398	0.00880	0.0000	80,846	189,204	7,201	0.00467	0.1726	82,691	188,498	7,201
	30	0.00751	0.00417	0.00751	0.0000	70,208	193,718	7,201	0.00417	0.0000	73,750	191,814	7,201
600	10	0.02687	0.01735	0.03942	0.4672	323,577	238,119	7,201	0.20002	10.5259	946,927	371,025	7,201
	15	0.01926	0.01512	0.04697	1.4384	264,625	263,677	7,201	0.04620	2.0559	338,509	290,444	7,201
	20	0.01386	0.00790	0.01933	0.3946	206,178	248,292	7,202	0.02240	1.8336	238,084	279,044	7,201
	25	0.00764	0.00405	0.02415	2.1603	148,522	290,006	7,201	0.02526	5.2307	176,247	301,324	7,201
	30	0.00805	0.00426	0.01407	0.7479	146,317	264,453	7,201	0.23190	53.4615	1,000,744	771,965	7,202
Average					0.2670			6,866		3.6985			6,907

Table 3
Computational results in obtaining compromise solutions using the VNS based matheuristic

n	p	*The best known* L_1	*The best known* L_∞	VNS based matheuristic									
				Minimising L_1 problem					Minimising L_∞ problem				
				L_1	Dev	Z_d^1	Z_c^1	CPU (s)	L_∞	Dev	Z_d^∞	Z_c^∞	CPU (s)
150	10	**0.01612**	0.01187	0.01612	0.0000	17,162	61,329	400	**0.01187**	0.0000	16,641	63,178	400
	15	0.01590	0.00921	0.01590	0.0000	13,291	64,334	601	0.00944	0.0252	12,811	66,002	600
	20	0.01513	0.00880	0.01625	0.0742	11,139	65,831	804	0.00884	0.0051	10,439	67,825	801
	25	0.01085	0.00561	0.01085	0.0000	8,346	68,604	1,001	0.00565	0.0067	8,160	69,329	1,002
	30	0.01055	0.00566	0.01072	0.0158	7,414	71,059	1,201	0.00566	0.0000	7,185	72,182	1,202
300	10	0.02606	0.01637	0.02641	0.0137	70,884	109,503	401	0.01692	0.0331	67,432	113,354	401
	15	0.01728	0.01093	0.02042	0.1817	55,759	111,803	602	0.01154	0.0556	51,983	116,068	603
	20	0.01287	0.00757	0.01440	0.1192	43,690	116,745	810	0.00757	0.0000	41,863	119,174	808
	25	0.00769	0.00455	0.00940	0.2220	34,450	119,791	1,011	0.00455	0.0000	34,503	119,295	1,010
	30	0.00700	0.00473	0.00941	0.3437	30,372	123,421	1,211	0.00500	0.0566	31,249	122,863	1,210
450	10	0.02252	0.01666	0.02252	0.0000	163,045	172,707	401	0.01666	0.0000	158,749	179,679	410
	15	0.01768	0.01391	0.02490	0.4081	143,739	175,318	610	0.01391	0.0000	127,395	183,753	610
	20	0.01109	0.00842	0.01401	0.2635	104,028	183,634	810	0.01086	0.2898	106,287	192,893	811
	25	0.00880	0.00398	0.00928	0.0535	84,152	187,220	1,010	0.00398	0.0000	81,347	186,943	1,010
	30	0.00751	0.00417	0.01092	0.4546	75,835	195,510	1,211	0.00429	0.0272	73,919	192,138	1,211
600	10	0.02687	0.01735	0.02687	0.0000	304,040	233,000	404	0.01735	0.0000	293,850	240,522	401
	15	0.01926	0.01512	0.01926	0.0000	238,138	236,936	610	0.01512	0.0000	242,071	249,276	611
	20	0.01386	0.00790	0.01386	0.0000	191,500	245,432	810	0.00790	0.0000	186,010	251,307	810
	25	0.00764	0.00405	0.00764	0.0000	150,117	248,450	1,010	0.00405	0.0000	150,640	248,915	1,011
	30	0.00805	0.00426	0.00805	0.0000	133,826	256,181	1,211	0.00426	0.0000	137,015	254,817	1,210
Average					0.1075			806		0.0250			807

7. Conclusion

This paper investigates the bi-objective capacitated *p*-median problem using compromise programming approach. A new problem is studied by considering the fixed cost of opening facility and several possible capacities that can be used by potential facilities. An optimisation model is developed to deal with two conflicting objectives, namely the total distances and the total fixed cost. A mathematical model using integer linear programming (ILP) is put forward to find the optimal location of open facilities with their optimal capacity. As the exact method experiences difficulties in finding the compromise solutions, a VNS based matheuristic is proposed. The proposed solution method incorporates an aggregation technique, the exact method, and the VNS algorithm. The proposed approach was assessed using newly generated datasets. The solutions of the proposed VNS based matheuristic were compared with the ones obtained by the exact method executed within a limited computing time. Based on the computational results, the VNS based matheuristic performs very well as it produced small deviations within a short computational time.

The following research directions may be worthy of investigation in the future. We classify them into two categories, namely problem-based and approach-based issues. In the problem-based case, this bi-objective problem can also be applied for/to the location-routing problem (LRP) where the problem is to determine the location of facilities, assigning customers to them and determining vehicle routes. Nagy and Salhi (2007) provides an excellent review on the LRP. From an approach-based point of view, the solution technique could include other heuristic frameworks, such as hybridisation heuristic search techniques, see Salhi (2017) for more comprehensive classifications on heuristics/ meta-heuristics approaches.

References

1. Ahmadi, S., Osman, I.H., 2005. Greedy random adaptive memory programming search for the capacitated clustering problem. *European Journal of Operational Research* 162, 1, 30–44.
2. Baldacci, R., Hadjiconstantinou, E., Maniezzo, V., Mingozzi, A., 2002. A new method for solving capacitated location problems based on a set partitioning approach. *Computers & Operations Research* 29, 4, 365–386.

3. Balliauw, M., Herremans, D., Palhazi Cuervo, D., Sörensen, K., 2017. A variable neighborhood search algorithm to generate piano fingerings for polyphonic sheet music. *International Transactions in Operational Research* 24, 3, 509-535.
4. Beheshtifar, S., Alimoahmmadi, A., 2015. A multiobjective optimization approach for location-allocation of clinics. *International Transactions in Operational Research* 22, 2, 313-328.
5. Boccia, M., Sforza, A., Sterle, C., Vasilyev, I., 2007. A cut and branch approach for the capacitated *p*-median problem based on Fenchel cutting planes. *Journal of Mathematical Modelling and Algorithms* 7, 1, 43–58.
6. Brimberg, J., Mladenovic, N., 1996. A variable neighbourhood algorithm for solving the continuous location-allocation problem. *Studies of Locational Analysis* 10, 1-12.
7. Büdenbender, K., Grünert, T., Sebastian, H.-J., 2000. A hybrid tabu search/branch-and-bound algorithm for the direct flight network design problem. *Transportation Science* 34, 4, 364–380.
8. Chaves, A., Assis Correa, F., Lorena, L., 2007. Clustering search heuristic for the capacitated p-median problem. In Corchado, E., Corchado, J., Abraham, A. (eds) *Innovations in Hybrid Intelligent Systems*, Vol. 44 of Advances in Soft Computing. Springer, Berlin/Heidelberg, pp. 136–143.
9. Díaz, J.A., Fernàndez, E., 2006. Hybrid scatter search and path relinking for the capacitated p-median problem. *European Journal of Operational Research* 169, 2, 570–585.
10. Dumitrescu, I., Stützle, T., 2003. Combinations of local search and exact algorithms. In Cagnoni, S., Johnson, C., Cardalda, J.,Marchiori, E.,Corne,D.,Meyer, J.-A., Gottlieb, J.,Middendorf, M., Guillot, A.,Raidl,G., Hart,E. (eds) *Applications of EvolutionaryComputing*. Vol. 2611 of Lecture Notes in Computer Science. Springer, Berlin/Heidelberg, pp. 211–223.
11. El Amrani, M., Bernadada, Y., Gendron, B., 2016. Generalization of capacitated *p*-median location problem: modeling and resolution. 2016 3rd International Conference on Logistics Operations Management (GOL), 1-6.
12. Fanjul-Peyro, L., Ruiz, R., 2011. Size-reduction heuristics for the unrelated parallel machines scheduling problem. *Computers & Operations Research* 38, 1, 301–309.
13. Fernandes, S., Lourenço, H.R., 2006. Optimised search heuristics: a mapping of procedures and combinatorial optimisation problems. Technical Report, Universidade do Algarve, Faro, Portugal.

14. Fleszar, K., Hindi, K., 2008. An effective VNS for the capacitated p-median problem. *European Journal of Operational Research* 191, 3, 612–622.

15. Gan, J., Colletti, J.P., Kolison Jr, S.H., 1996. A Compromise Programming Approach to Integrated Natural Resource Management. In Society of American Foresters.

16. Garey, M.R., Johnson, D.S., 1979. Computers and intractability: a guide to the theory of NP-completeness. W.H. Freeman & Co., New York, NY.

17. Hansen, P., Mladenovic, N., 1997. Variable neighbourhood search for the *p*-median. *Location Science* 5, 4, 207-225.

18. Hansen, P., Mladenovic, N., 2001. Variable neighbourhood search: Principles and applications. *European Journal of Operational Research* 130, 3, 449-467.

19. Hansen, P., Mladenovic, N., Perez, J.A.M., 2010. Variable neighbourhood search: methods and applications. *Annals of Operations Research* 175, 1, 367-407.

20. Irawan C.A., Salhi S., Scaparra, M.P., 2014. An Adaptive Multiphase Approach for Large Unconditional and Conditional *p*-Median Problems. *European Journal of Operational Research* 237, 2, 590–605.

21. Irawan C. A., Salhi S., 2015a. Solving large *p*-median problems by a multistage hybrid approach using demand points aggregation and variable neighbourhood search. *Journal of Global Optimization*, 63, 3, 537–554.

22. Irawan C.A., Salhi S., 2015b. Aggregation and non-aggregation techniques for large facility location problems – A survey. *Yugoslav Journal of Operations Research* 25, 3, 1-11.

23. Irawan C. A., Salhi S., & Drezner, Z., 2016. Hybrid meta-heuristics with VNS and exact methods: application to large unconditional and conditional vertex p-centre problems. *Journal of Heuristics* 22, 4, 507–537.

24. Jourdan, L., Basseur, M., Talbi, E.-G., 2009. Hybridizing exact methods and metaheuristics: a taxonomy. *European Journal of Operational Research* 199, 3, 620–629.

25. Kammoun, M., Derbel, H., Ratli, M., Jarboui, B., 2017. An integration of mixed VND and VNS: the case of the multivehicle covering tour problem. *International Transactions in Operational Research* 24, 663–679.

26. Kariv, O., Hakimi, S.L., 1979. An algorithmic approach to network location problems I: the *p*-centers. *SIAM J. Appl. Math.* 37, 3, 513-538

27. Landa-Torres I., Del Ser J., Salcedo-Sanz S., Gil-Lopez S., Portilla-Figueras, J. A., and Alonso-Garrido O., 2012. A comparative study of two hybrid grouping evolutionary techniques for the capacitated *p*-median problem. *Computers & Operations Research* 39, 9, 2214–2222.

28. Lorena, L.A.N., Senne, E.L.F., 2004. A column generation approach to capacitated *p*-median problems. *Computers & Operations Research* 31, 6, 863–876.

29. Maniezzo, V., Mingozzi, A., Baldacci, R., 1998. A bionomic approach to the capacitated *p*-median problem. *Journal of Heuristics* 4, 3, 263–280.

30. Maniezzo, V., Stützle, T., Voß, S., 2010. Matheuristics: Hybridizing Metaheuristics and Mathematical Programming (1st edn). Springer, Berlin.

31. Menéndez, B., Pardo, E. G., Sánchez-Oro, J., & Duarte, A., 2017. Parallel variable neighborhood search for the min–max order batching problem. *International Transactions in Operational Research* 24, 3, 635-662.

32. Mulvey, J.M., Beck, M.P., 1984. Solving capacitated clustering problems. *European Journal of Operational Research* 18, 3, 339–348.

33. Nagy, G., Salhi, S., 2007. Location-routing: Issues, models and methods. *European Journal of Operational Research* 177, 649–672.

34. Osman, I.H., Christofides, N., 1994. Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions in Operational Research* 1, 3, 317–336.

35. Pirkul, H., 1987. Efficient algorithms for the capacitated concentrator location problem. *Computers & Operations Research* 14, 3, 197–208.

36. Puchinger, J., Raidl, G., 2005. Combining metaheuristics and exact algorithms in combinatorial optimization: a survey and classification. In Mira, J., Álvarez, J. (eds) *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach*. Vol. 3562 of Lecture Notes in Computer Science. Springer, Berlin, pp. 41–53.

37. Rath, S., Gendreau, M., Gutjahr, W. J., 2016. Bi-objective stochastic programming models for determining depot locations in disaster relief operations. *International Transactions in Operational Research* 2, 6, 997-1023.

38. ReVelle, C.S., Swain, R.W., 1970. Central facilities location. *Geographical Analysis* 2, 1, 30–42.

39. Romero, C., Rehman, T., 1989. Multiple Criteria Analysis for Agricultural Decisions. Elsevier Science Publishers, Amsterdam, the Netherland.

40. Romero, C., Tamiz, M., Jones, D.F., 1998. Goal programming, compromise programming and reference point method formulations: linkages and utility interpretation. *Journal of the Operational Research Society* 49, 9, 986-991.

41. Salhi, S., 2017. *Heuristic Search: The Emerging Science of Problem Solving*. Springer, Cham, Switzerland, DOI: 10.1007/978-3-319-49355-8.

42. Scheuerer, S., Wendolsky, R., 2006. A scatter search heuristic for the capacitated clustering problem. *European Journal of Operational Research* 169, 2, 533–547.

43. Stefanello, F., de Araújo, O. C. B., Müller, F. M., 2015. Matheuristics for the Capacitated *p*-Median Problem. *International Transactions in Operational Research* 22, 1, 49–167.

44. Taillard, E.D., Voss, S., 2002. POPMUSIC—partial optimization metaheuristic under special intensification conditions. In Ribeiro, C.C., Hansen, P. (eds) Essays and Surveys in Metaheuristics SE-27. Vol. 15 of Operations Research/Computer Science Interfaces Series. Springer, New York, pp. 613–629.

45. Talbi, E.-G., 2002. A taxonomy of hybrid metaheuristics. *Journal of Heuristics* 8, 5, 541–564.

46. Vidović, M., Popović, D., Ratković, B., Radivojević, G., 2017. Generalized mixed integer and VNS heuristic approach to solving the multisize containers drayage problem. *International Transactions in Operational Research* 24, 3, 583-614.

47. Yaghini, M., Momeni, M., Sarmadi, M., Ahadi, H. R., 2013a. An efficient heuristic algorithm for the capacitated *p*−median problem. *4OR: Quarterly Journal of Operations Research* 11, 3,229–248.

48. Yaghini, M., Karimi, M., Masoud, M., Rahbar, M., 2013b. A hybrid metaheuristic approach for the capacitated *p*-median problem. *Applied Soft Computing* 13, 9, 3922–3930.

49. Yu, P.L., 1973. A Class of Solutions for Group Decision Problems. *Management Science* 19, 8, 936-946.

50. Zeleny, M., 1974. A concept of compromise solutions and the method of the displaced ideal. *Computers and Operations Research* 1, 3-4, 479–496.