Available online at www.sciencedirect.com

ScienceDirect

European Journal of Operational Research 179 (2007) 1194–1207

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/ejor

A perturbation-based heuristic for the capacitated multisource Weber problem ☆

Z.M. Zainuddin [a], S. Salhi [b,*]

[a] *Mathematics Department, Universiti Teknologi Malaysia, Skudia, Johor, Malaysia*
[b] *Centre for Heuristic Optimisation, Kent Business School, University of Kent, Canterbury, UK*

Received 15 January 2004; accepted 15 September 2005
Available online 31 March 2006

Abstract

This paper proposes a perturbation-based heuristic for the capacitated multisource Weber problem. This procedure is based on an effective use of borderline customers. Several implementations are considered and the two most appropriate are then computationally enhanced by using a reduced neighbourhood when solving the transportation problem. Computational results are presented using data sets from the literature, originally used for the uncapacitated case, with encouraging results.

Keywords: Capacitated; Location–allocation; Continuous space; Heuristics

1. Introduction

The continuous capacitated location–allocation problem with a fixed number of open facilities each with a constant capacity, which is also known as the capacitated multisource Weber problem, may be stated as follows: given the location of each fixed point (customer point), the demand at each fixed point, the transportation cost for the area of interest, the number of facilities to open, and the capacity of each of these facilities, the aim is to determine the location of each facility, and the allocation of customers to these open facilities (if more than one facilities are to be opened). Given

Parameters

n	the number of fixed points (or customer points)
w_j	demand or weight of customer j $(j = 1,\ldots,n)$
$a_j = (a_j^1, a_j^2)$	location of customer j where $a_j \in \Re^2$, $(j = 1,\ldots,n)$
M	the number of facilities to be located
b	fixed capacity of a facility where $b \in N$

Decision variables

$X_i = (X_i^1, X_i^2)$ coordinates of facility i where $X_i \in \Re^2$

☆ This research was conducted when both authors were at the University of Birmingham, UK.

[*] Corresponding author.
E-mail address: S.Salhi@kent.ac.uk (S. Salhi).

0377-2217/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2005.09.050

x_{ij} quantity assigned from facility i to customer j, $i = 1, \ldots, M$, $j = 1, \ldots, n$

The problem can be formulated as follows:

$$\text{Minimise} \sum_{i=1}^{M} \sum_{j=1}^{n} x_{ij} d(\boldsymbol{X}_i, \boldsymbol{a}_j) \tag{1}$$

subject to

$$\sum_{j=1}^{n} x_{ij} \leqslant b \quad \forall i = 1, \ldots, M, \tag{2}$$

$$\sum_{i=1}^{M} x_{ij} = w_j \quad \forall j = 1, \ldots, n, \tag{3}$$

$$x_{ij} \geqslant 0 \quad \forall i = 1, \ldots, M; \;\; j = 1, \ldots, n, \tag{4}$$

where $d(\boldsymbol{X}_i, \boldsymbol{a}_j)$ represents the Euclidean distance between facility i and customer j.

(1) denotes the objective function which is the total transportation cost, (2) ensures that capacity constraints of the facilities are not violated, (3) guarantees that the demand of every customer is satisfied and (4) refers to non-negativity of the decision variables x_{ij}.

It can be noted that once the set of open facilities has been decided upon (e.g., if we fix the open facilities in the formulation), the resulting problem reduces to the usual Transportation Problem (TP) which can be solved optimally in polynomial time. In short, the problem is to find the best facility configuration.

In this study the value of b is set to $\left\lceil \frac{\sum_{j=1}^{n} w_j}{M} \right\rceil$ where $\lceil x \rceil$ is the smallest integer larger than or equal to x. Note that if $b > \frac{\sum_{j=1}^{n} w_j}{M}$ we introduce a dummy customer with a 0 transportation cost and a demand equals to the remaining demand, e.g., $b - \frac{\sum_{j=1}^{n} w_j}{M}$. This customer is used only when solving the TP, but not at the location and the allocation stages.

Most of the work in the literature on the capacitated facility location concentrates on the discrete problem and the methods mainly used include dual-ascent based [11], cross decomposition method [13], constructive-type heuristic [10,7] and Lagrangian relaxation heuristics [2,1].

Other related work on the continuous location problem include Eben-Chaine et al. [8] who studied the case of capacitated facility location on a line, and Brimberg and Mladenovic [4], Brimberg et al. [3] and recently by Salhi and Gamal [12] who investigated the multisource Weber problem. To our knowledge, it is only Cooper [6] in the 1970s who attempted the capacitated continuous case. He presented exact and approximate methods for solving the transportation-location problem. The heuristic method described in this work is a modification of the alternating transportation-location method introduced in [6]. Here, the location method and the usual TP are alternately applied until there is no epsilon improvement in cost. We shall describe Cooper's method [6] as this will be used as the foundation for our perturbation-based heuristic.

1.1. Cooper's alternating transportation-location heuristic (ATL)

Firstly, M facilities are randomly chosen from the fixed points. Then, the TP using these M open facilities is solved to find the allocation for the capacitated problem. For each of the M independent set of allocations, containing n_i fixed points where $i = 1, \ldots, M$ and $\sum_{i=1}^{M} n_i \geqslant n$, the new location of the facilities is found using the iterative procedure based on the Weiszfeld Equation which is given below:

$$X_i^{1^{(k)}} = \frac{\sum_{j_i=1}^{n_i} \frac{w_{j_i} a_{j_i}^1}{d\left(X_i^{(k-1)}, a_{j_i}\right)}}{\sum_{j_i=1}^{n_i} \frac{w_{j_i}}{d\left(X_i^{(k-1)}, a_{j_i}\right)}} \quad \text{and}$$

$$X_i^{2^{(k)}} = \frac{\sum_{j_i=1}^{n_i} \frac{w_{j_i} a_{j_i}^2}{d\left(X_i^{(k-1)}, a_{j_i}\right)}}{\sum_{j_i=1}^{n_i} \frac{w_{j_i}}{d\left(X_i^{(k-1)}, a_{j_i}\right)}}, \tag{5}$$

where the superscript k denotes the iteration number and w_{j_i} represents all or a fraction of the jth customer demand that is assigned to facility i. Obviously $w_{j_i} \leqslant w_j$ as some customers may have their demand split because of the solution of the TP and hence some customers can be used more than once in Eq. (5) with their appropriate demand adjusted accordingly.

The location problem and the TP are alternately solved until there is no epsilon improvement in cost.

According to [6], ATL yields a convergent monotone nonincreasing sequence of values for the objective function. However, there is no guarantee that it will converge to the global minimum but the result, when not optimal, is found empirically to lie within ~10%, and usually within 2–3%, of the optimal solution when tested on small instances.

The rest of the paper is structured as follows: in the next section, the modification on Cooper's ATL is presented. Section 3 describes our perturbation-based heuristic and section Section 4 presents a neighbourhood reduction for solving the TP. Section 5 provides our computational results and our findings as well as some research issues are given in the last section.

2. A modified Cooper's heuristic

In this section we present a scheme for generating initial solutions and implementations that consider the diversity of these solutions when addressing the capacitated problem. These ideas with a slight modification within Cooper's algorithm are then combined to form our first heuristic which we refer to as the modified Cooper's heuristic.

2.1. The generation of an initial solution

The first part of the heuristic is to generate an initial facility configuration. Instead of just starting with M randomly chosen points as in ATL, the initial facility configuration is found through solving heuristically the uncapacitated problem. This is used for two reasons (i) the solution found can be optimal or near optimal if found feasible, and (ii) this solution can be used as a lower bound especially if the solution is known to be optimal or very close to optimal as shown in the literature (see [3]). Our approach is based on Cooper's multi-start alternate algorithm (CMSA) [5]. For each starting configuration, the Cooper's alternate procedure of locate and allocate is carried out until there is less than epsilon improvement in cost, (say 0.0001). However, the solution found by this method is a local minimum. To increase the chance of getting a near optimal solution the method is repeated several times, say K, using different random starting locations. In other words, CMSA is the repeated use of Cooper's alternate method.

2.1.1. The Furthest Distance Rule

The obtention of the initial solution can be carried out either randomly or via quick greedy heuristics for the p median problem. In our preliminary testing (see [14, pp. 74–80]), based on the 50-customers problem from the literature (see [3]), we used the multi-start heuristic as performed in Cooper, the Furthest Distance Rule which we refer to as the FDR, and also a combination of FDR and the drop heuristic. As a compromise between solution quality and computational effort we have opted for the FDR as our quick heuristic for generating our initial facility configurations for the uncapacitated location problem. The reasoning behind the FDR is to generate reasonably quickly initial facility location points which are situated far apart. This rule is defined as

$$\sum_{i\in E_1} d(X_i, a_{j^*}) = \max_{j\in J} \sum_{i\in E_1} d(X_i, a_j), \tag{6}$$

where E_1 is the set of facility locations already chosen as initial points, J is the set of fixed points not chosen yet, and (j^*) is the new selected site using Eq. (6).

The first point is chosen randomly from the existing fixed points, then the remaining $M - 1$ points that are far apart are generated using Eq. (6). For simplicity we restrict our initial location to a fixed point though this could be generated randomly in the plane. The algorithm that uses this idea is referred to as the Furthest Distance Method (FDM for short) and its steps are given in Fig. 1.

2.2. Solving the capacitated location problem

In this section, we first discuss three implementations based on the solutions found by the FDM to solve the capacitated problem, and then we present the algorithm which we refer to as the modified Cooper's algorithm.

2.2.1. Multi-start alternate algorithm (MSA)

One way of solving the capacitated problem is by taking the best configuration (i.e., configuration with the minimum cost) out of the K runs for the uncapacitated problem to be the starting configuration for the capacitated problem. In other words, the capacitated problem is only solved once.

2.2.2. Single-start alternate algorithm (SSA)

It is observed that, the best cost for the capacitated problem does not necessarily originate from the initial solution that yields the best cost for the uncapacitated problem. Therefore, another way of solving the problem is by considering all the K configurations for the uncapacitated problem to be the initial starting location for the capacitated problem. In this case, the capacitated problem is solved K times.

2.2.3. Intermediate-start alternate algorithm (ISA)

In this method, the capacitated problem is solved by using a sample of configurations extracted from the K found initial configurations (say D

Stage 1: Furthest Distance Rule

Step 1 · Choose a customer point at random, say j_1. Set $E_1 = \{j_1\}$ and $k = 2$.

Step 2 · From E_1, apply the furthest distance rule to select a new location point, j^*_k, set $E_1 = E_1 \cup \{j^*_k\}$ and $k = k + 1$.

Repeat step 2 until $k > M$ then determine the set of customers served by each of the open facilities, say $A_i, \forall i \in E_1$.

Stage 2: Cooper's Scheme

Step 3 · Apply Cooper's alternate algorithm using E_1 and all $A_i, \forall i \in E_1$.

Step 4 · Repeat step 1 to step 4 for K times, and record the configuration that yields the cheapest cost.

Fig. 1. The Furthest Distance Method (FDM).

configurations, $D < K$) obtained when solving the uncapacitated problem. The scheme of selecting these D configurations is described below. Once these D configurations are chosen, we will then proceed to solve the capacitated problem for each value of these D scenarios. This scheme could be seen as a compromise between the MSA and the SSA. For instance, if $D = K$, this becomes the SSA whereas when $D = 1$ it is the MSA. It is obviously clear that when the value of D gets larger, the quality of the solution when solving the capacitated problem gets better or remains unchanged but such a gain in quality requires relatively more computing time.

In this paper, the diversity or the dissimilarity of the sample candidates is measured based on the cost. The K configurations are arranged in ascending order of the cost. The least cost configuration (i.e., the top of the list) is always selected since it has the minimum cost. To choose the other $(D - 1)$ candidates, the gaps between two successive costs are calculated and used as a measure to differentiate between dissimilar configurations. In this approach we only consider the configurations with gaps larger than a prescribed gap ϵ which is defined as the average value of the gaps, i.e., $\epsilon = \frac{\sum_{t=1}^{K-1} G(t)}{K-1}$ where $G(t)$ represents the gap between the cost of

Step 1 Find S_d the initial starting configuration for the capacitated problem.

Step 2 Apply TP to the facilities found in S_d to find the new allocation for the capacitated problem.

Step 3 Find their new locations by using equations (5).

Step 4 Allocate the customers to their nearest facility and find the new location using equations (5).

Step 5 Apply TP to find the new allocation and its corresponding cost.

Step 6 Repeat steps 3, 4 and 5 until there is less than epsilon improvement in cost to obtain $cost(S_d)$.

Fig. 2. Alternating transportation-location–allocation-location (ATLAL) for a given d, $d = 1, \ldots, D$.

the $(t+1)$th and the tth configuration. Thus, in this scheme, the value of D is not necessarily constant.

2.3. The modified Cooper's algorithm

The capacitated problem is solved using a procedure modified from Cooper's ATL. This method which we refer to as the alternating transportation-location–allocation-location method (ATLAL) is similar to the ATL except that instead of alternating between the TP and the location problem, we add another step (see step 4 in Fig. 2) where after we get the new location of the facilities, we allocate the customers to their nearest facility, solve the location problem again and then the TP for the new allocation for the capacitated problem. The main steps of ATLAL are given in Fig. 2, for a given d, $d = 1,\ldots,D$ where $D = 1, K$ or $1 < D < K$. Let S_d be the dth configuration and $\text{cost}(S_d)$ its corresponding cost. If $d > 1$, we select the configuration yielding the overall least cost, $\text{cost}(S_d^*) = \min_{d=1,\ldots,D}\{\text{cost}(S_d)\}$.

We would like to note that the introduction of this additional step (step 4) no longer results in a convergent sequence as the monotonic property can be lost from one cycle (step 2 to step 5) to another. However, this shake up is embedded purposely to provide flexibility in exploring more than one local minimum by being able to escape from regions of the previously found local minima.

3. A perturbation-based scheme

A post optimisation procedure that attempts to improve the currently found solution by ATLAL for each S_d, $d = 1,\ldots,D$ is proposed. In this approach, the locations of the facilities found with the ATLAL heuristic are perturbed by taking into account the clustering of the borderline customers. These customers are defined as those which lie in between their nearest facility and their second nearest facility. In other words, the distance between the customers and their nearest and second nearest facilities is more or less the same. The formation of these clusters is defined in the next subsection. The point candidates (customers) of these clusters are temporarily forced to be assigned to their nearest facilities while we solve the TP. This task is performed by temporarily removing these customers from the system when we are solving the TP and then re-introducing them back when we solve the location problem. This restriction is imposed in order to make the locations of their 'best' facilities nearer to these customers. When using these new locations, it is likely that some of these customers will be allocated to their nearest facility as in the uncapacitated case. This scheme is repeated starting with the recent best configuration for the capacitated problem until there is no epsilon reduction in cost or when there is no borderline customers that can be served by their second best facility. The remainder of this section covers the different mechanisms used within this perturbation-based procedure.

3.1. The creation of the clusters

The construction of the clusters is performed as follows:

3.1.1. Borderline customers

We identify the set of borderline customers, B, irrespective of the capacity of the facilities as

$$B = \left\{ i \in \{1,\ldots,n\} \quad \text{s.t. } \rho_i = \frac{d(F_{1,i}i)}{d(F_{2,i,}i)} \geqslant \rho_{\max} \right\},$$

where $d(F_{1,i}, i)$ is the distance of customer i to its nearest facility, $F_{1,i}$, $d(F_{2,i}, i)$ is the distance of customer i to its second nearest facility, $F_{2,i}$, and $\rho_{\max}$ is the cut off point.

The choice of the value $\rho_{\max}$ is important. If the value is too small (near 0), too many points will be in the set B and if it is too large (near 1), the number of points in B will be too small. In this work, the value of $\rho_{\max}$ is found dynamically as shown below where the initial value of $\rho_{\max}$ is set to 0.8.

3.1.2. Assignment of customers

The set of customers that were re-assigned not to their nearest facility, due to the TP, is defined by B_1 as

$$B_1 = \{i \in B \text{ and } i \text{ is not completely allocated to } F_{1,i}\} = \{j_h\}_{h=1,\ldots,H},$$

where H denotes the number of elements in B_1 (i.e., $|B_1|$).

Note that B_1 may include borderline customers, that are not necessarily served by the second, third,…, best facility. Note also if the weight of a customer is not unity, this customer may be served by more than one facility. In this case, even if a fraction of the weight of some facility is served by its 'best' facility, it is still considered as a candidate in B_1. The value of $|B_1|$ plays also an important role in defining the centre of the clusters. More explanation on this issue will be given in Section 3.1.3 below.

(a) Case $B_1 = \{\}$ (i.e., all the borderline customers are completely served by their 'best' facility).
- If $\rho_{\max} > \rho_{\min}$ (the minimum cut off, say 0.6), do the following steps:

```
Do while $\rho_{\max} > \rho_{{}_{\max}\min}$ and $B_1 = \{\}$,
    set $\rho_{\max} = \rho_{\max} - 0.1$ and reconstruct $B_1$
Enddo
```

- If $\rho_{\max} = \rho_{\min}$ (i.e., all the borderline customers are still served by their 'best' facility), do the following;

```
*if* (*the perturbation scheme is applied for the first time*) *then*
    take the configuration found for the capacitated problem using ATLAL without the perturbation scheme.
*else*
    take the best configuration found from the previous application of the perturbation scheme as the best solution.
*endif*
```

(b) Case $|B_1| > 0$ (i.e., not all borderline customers are completely served by their best facility).
Set $\rho_{\max} = \rho_{\max} - 0.1$, reconstruct B_1, and let $L = |B_1|$.

```
if ($L > H$) take the new value of $\rho_{\max}$ and the new set $B_1$.
*else* (i.e., $L = H$) take the previous value of $\rho_{\max}$ and the previous set $B_1$ since there is no change in the number of candidates of $B_1$ even with $\rho_{\max}$ decreased.
*endif*
```

The last value of $\rho_{\max}$ found is then used as our cut off point for generating borderline customers. Note that $L \geqslant H$.

3.1.3. Formation of the clusters

After B_1 has been identified, we proceed with the formation of the clusters. The maximum number of clusters is taken to be k_0, which is set to M in our study. We first find the centres of the clusters then assign the customers to these clusters.

(a) The obtention of the centres of the clusters
- Find X_1 the centre of the cluster C_1 such that $X_1 \in B_1$. The first centre is chosen as the customer of B_1 with the smallest value of ρ_i.
- Apply the Furthest Distance Rule as given by Eq. (6) based on B_1 to get the other $k_0 - 1$ centres, $X_k \in B_1$, $k = 2, \ldots, k_0$.
The idea of using the Furthest Distance Rule in finding the centres is that we want the centres to be as far away from each other as possible. This is because, if the centres are too close to each other, they may attract one another in the process of clustering the points.
- Construct a forbidden region
Note that, when applying the Furthest Distance Rule we may get a point which is close to one of the points we have already selected previously. To avoid this, we impose a forbidden region around the current centre/s. The concept of making previously visited solutions forbidden for future exploration is one of the key factors in tabu search meta-heuristic methodology. In this work, for simplicity, we define such a region by a circle centred at the current centre with a radius to be defined below. In other words, only the points that are outside the already constructed circle(s) are potential points for cluster centers. Therefore, the number of clusters may be less than or equal to k_0, say k_1. A similar idea was also used by Gamal and Salhi [9] when solving the multi-source Weber problem. The radius of the forbidden region is defined as follows. Initially, the customers situated within a certain radius around the centre are found. As there might be some other points that lie close to these already chosen customers but happen to be just marginally outside the cluster, the neighbouring customers of these chosen points need also to be included in the cluster. In the following, for simplicity of notation, we consider the first cluster as an example since the same formulae applies to all the k clusters. Let $d(X_1, FX_1)$ be the Euclidean distance between the first centre X_1 and the facility that serves it (FX_1) and set the radius of the forbidden region (r) to $\mathbf{r} = \epsilon_{\max} + \hat{\epsilon}$ where $\epsilon_{\max}$ is the initial radius set to $\frac{d(X_1, FX_1)}{2}$. In other words, customers situated within this radius of the centre will be assigned to this cluster.
$\hat{\epsilon}$ is the radius of the neighbourhood of the initial cluster candidates which we set to

$\frac{d(X_1,FX_1)}{4}$. This flexibility is introduced to allow those customers very close to those already assigned based on $\epsilon_{\max}$ to be included.

(b) The generation of the clusters

Let B_2 be the set of all customers served by other facilities than their best one and note that B_2 is not necessarily a subset of B_1. Fig. 3 shows how r, $\epsilon_{\max}$ and $\hat{\epsilon}$ are defined for a given cluster k, and also illustrates the elements in B, B_1, B_2 and the cluster C_k. For each centre, those customers in B_2 which fall within the radius $\epsilon_{\max}$ are checked. If the cluster is empty (this means that there are no other candidate besides the centre), then the next centre is checked and so on. If some customers are obtained, the cluster's candidates are assigned as follows. Firstly, those customers which fall within the radius ϵ are assigned to the cluster, initially $\epsilon = \epsilon_{\min}$, see Fig. 4 for details. If the cluster is empty, the value of e is increased by a fixed amount (say $\beta = 0.1$) up to $\epsilon_{\max}$ until customer(s) are assigned to this cluster. Then, the neighbouring customers which lie outside the ϵ radius but within $\epsilon_{\min}$ radius of the currently chosen customers are also chosen. To obtain the cluster's candidates for a certain cluster, say cluster k, the scheme given in Fig. 4 can be followed.

Note that if some customers are served by their 'best' facilities, even though they are located close to one another, as in a cluster, they will not be considered to form a cluster.

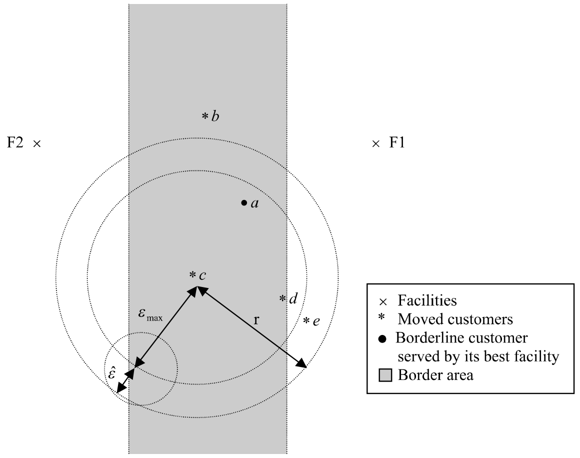

Fig. 3. Formation of a given cluster k, $k = 1, \ldots, k_1$ with radius r and centre c.

3.2. Temporary removal of clusters

In this subsection, we restrict the clusters to remain assigned to their nearest facility while we solve the allocation problem. Note that as the total demand of those customers in a given cluster is relatively smaller compared to the capacity of the facility as given by the value of b, the assignment of a given cluster to its nearest facility is therefore feasible. In the case where the facilities happen to have different capacities, the proposed relaxation scheme needs to be modified to cater for such a situation. In this scheme we temporarily omit the customers of the clusters when we are solving the TP. By doing this, we are forcing the customers to remain served by their nearest facilities until the location of the facilities become unchanged from one iteration to the next. However, if by assigning a cluster point to its nearest facility violates the supply constraint of that facility, the point will be omitted from the cluster. This is repeated for all the clusters obtained. The main steps are summarised in Fig. 5.

We present two variants for handling these clusters when solving the TP with full capacity. The issue here is to avoid the snowball effect where the location and allocation of one facility will affect the location of other facilities and the allocation of their customers. The first variant is based on temporarily removing all clusters one at time whereas the second concentrates on temporarily removing only those clusters that are likely to have an effect on the total cost.

3.2.1. Removal of all clusters

The location and their allocation problems with full capacity are solved alternately for all the clusters until less than epsilon improvement in cost is found. Obviously, this will require relatively a longer computing time since there are k_1 full TPs to be solved at each iteration.

3.2.2. Removal of some clusters

An empirical study is conducted to see the impact of the change after solving the TP without those customers belonging to the clusters (step 3 of Fig. 5), $(\delta L)_k$ where $k = 1, \ldots, k_1$ to the final solution found by the full TP when using the final configuration. This change in cost $(\delta L)_k$ is then sorted in descending order. It is worth noting that in some

```
Stage 1: Initial cluster candidates
Step 1  Check the set of customers positioned around the centre $X_k$,
    $N(X_k) = \{l_h, l_h \in B_2$ such that $d(l_h, X_k) \leq \epsilon_{max}\}$.
    If $N(X_k) = \{\}$, then set the candidate of $C_k = X_k$ and stop.
    Else set $\epsilon_{min} = \frac{\epsilon_{max}}{2}$, $\beta = 0$ and $L = \{\emptyset\}$.
Step 2  Let $\epsilon = \epsilon_{min}(1 + \beta)$. If $\epsilon \leq \epsilon_{max}$, for each $l_h \in N(X_k)$ do
    if $d(l_h, X_k) < \epsilon$ then $L = L \cup \{l_h\}$.
Step 3  If $L \neq \{\}$, set the cluster $C_k = L$, else set $\beta = \beta + 0.10$ and go to step 2.
Stage 2: Neighbourhood of the initial cluster candidates
Step 4  For each $l_h \in L$, choose the neighbouring points, $J_h$ of the point $l_h$ as
    $J_h = \{j, j \in B_2 - (B_2 \cap C_k)$ s.t. $d(l_h, j) < \hat{\epsilon}\}$ and $C_k = C_k \cup \{J_h\}$.
```

Fig. 4. The selection of the kth cluster candidates, $k = 1, \ldots, k_1$.

Step 1 Remove the members of one cluster at a time from their current allocation and assign them to their nearest facilities. In the case of split facilities (i.e. part of the demand of a given customer is served by some facility and the remaining by others), all the weight from all the next best facilities that serve the customers will be removed and added to their nearest facility. If part of the weight is served by the second best facility, the respective weight will be transfered to the nearest facility. If by assigning a cluster's candidate to its nearest facility violates the supply constraint, the candidate will be omitted from the cluster.

Step 2 For each facility and its assigned customers determine its new location using Weiszfeld Equation (5).

Step 3 Solve the TP by omitting these customers. In other words, the capacity of the cluster's candidates nearest facilities is reduced by the amount of their weights when solving the TP.

Step 4 If there is no change in the location of all those affected facilities, stop else re-introduce those removed customers to their respective facilities and go to step 2 except if M calls were already performed then stop.

Fig. 5. Temporary removal of a cluster.

instances, the final solution is found to be better even though the respective change in cost $(\delta L)_k$ is lower. Therefore, the difference between the highest change in cost $(\delta L)_1$ and the change in cost $(\delta L)_k$ that yields the best solution is calculated for 2–25 open facilities for the 50-fixed points problem and the 287-fixed points problem. These data sets which are usually used for the multisource Weber problem are taken from the literature (see [3]). From this limited preliminary experiment, we observed that it is not necessary to solve the full TPs for all the clusters but only to concentrate on those clusters having a change in cost $|(\delta L)_k| \leqslant 4|(\delta L)_1|$. Here, at iteration, the number of full TPs solved is k_2 where $k_2 \ll k_1$. This simple but powerful reduction scheme will then be used in our future testing.

In this approach, we explore another configuration besides the k_2 configurations already obtained. We restrict to just one more only to provide more flexibility while limiting the additional computational burden as this exercise is performed for each value of d ($d = 1, \ldots, D$) and at each iteration. It may be useful to investigate the effect of exploring more than one additional configuration in future. The idea is inspired from Genetic Algorithms where a new solution is constructed based on combining two existing configurations. Here, only the clusters with positive $(\delta L)_k$ are considered for combination. The rational behind this is that if we combine two or more clusters having positive $(\delta L)_k$ values, we may generate a new solution with a higher positive value. The new configuration will take the location of the affected facilities of all the clusters involved. For instance, in Fig. 6 where we have two clusters and four facilities to be opened, after temporary removing the cluster candidates, the location of facilities 2 and 3 are changed in cluster 1 and the location of facilities 1 and 4 are changed in cluster 2. Therefore, the new configuration will take the location of facilities 2 and 3 from cluster 1 and the location of facilities 1 and 4 from cluster 2. However, if there are shared affected facilities, the location of the facilities i, $i = 1, \ldots, M$ with more total demand will be taken. For example, in Fig. 7, the location of facility 3 is changed in both clusters 1 and 2. But, since the demand to facility 3 at the location in cluster 1 is more, or in other words, facility 3 has more customers if it is situated as in cluster 1, therefore, the new configuration will take the location of facility 3 from cluster 1. At each iteration, the number of full TPs to be solved in this method is then $k_3 = k_2 + 1$.

The selected configuration is the one that yields the least cost after solving the full TPs in both cluster methods.

As the solution obtained might be a local minimum, the perturbation scheme is then applied with the recently chosen configuration as the starting locations. However, before doing this, the change between the current cost of the capacitated problem after applying the perturbation scheme, $\text{cap}^*(S_d)$, and the cost of the capacitated problem without the perturbation scheme, $\text{cap}(S_d)$, is evaluated. Let $(\delta G)_d$ denote this change.

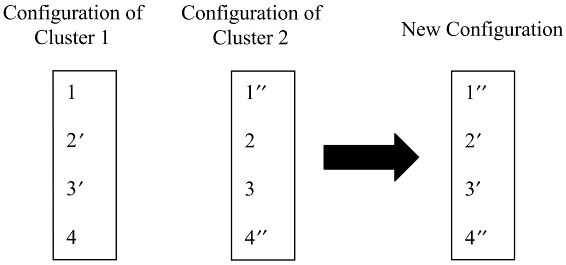

Fig. 6. No shared affected facilities.

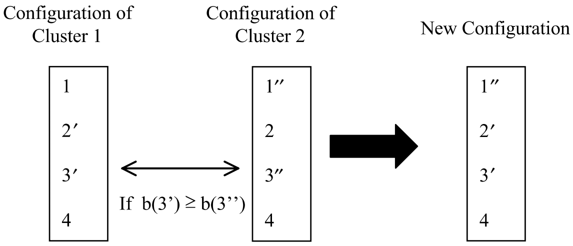

Fig. 7. Shared affected facilities.

- If $(\delta G)_d$ is positive, we start the perturbation scheme again with the recently obtained configuration.
  The process of creating the clusters, forcing them to stay at their nearest facilities and solving the location and allocation problem is repeated until there is less than ϵ reduction in cost. It can be observed that after a few iterations, though the number of clusters and the candidates of the clusters remain the same, we still continue with this process until no more improvement in cost. We adopt this strategy since the location may have changed without affecting any change in the allocation.
- If δG_d is negative, we record $\text{cap}(S_d)$ as our solution and stop.

4. Effect of neighbourhood reduction

It can be shown that a large amount of the total cpu time is consumed in solving the large number of TPs. Note that though the TP is solved in polynomial time, the use of such a procedure so many times renders the whole exercise computationally unattractive. There are a few ways on how to overcome this drawback. In this study, at each iteration, when solving the TP, we concentrate on a smaller portion of the original problem by considering a subset of facilities only. A smaller neighbourhood is then defined and the TP is solved for the facilities involved and their respective customers only. The main steps of the procedure are given in Fig. 8 and an illustration is provided in Fig. 9. In other words, for each cluster, we determine those facilities close to it and the assigned customers and then solve the TP based on this smaller subset of facilities.

Step 1 Identify the clusters' candidates nearest facility, F_{1,C_k} and their current allocated facility for the capacitated problem, FC_k.

Step 2 Compute the distance from the centre of the cluster to its current allocated facility, $R = d(X_k, FC_k)$.

Step 3 Find other facilities, F_k which lie within the circle with radius $\Phi = \frac{3}{2}R$.

Step 4 Identify respective customers currently allocated to the set of facilities involved $H^* = \{F_{1,C_k}, FC_k, F_k\}$, say $A(H^*)$.

Fig. 8. Definition of a smaller neighbourhood ($|H^*| < M, |A(H^*)| < n$).

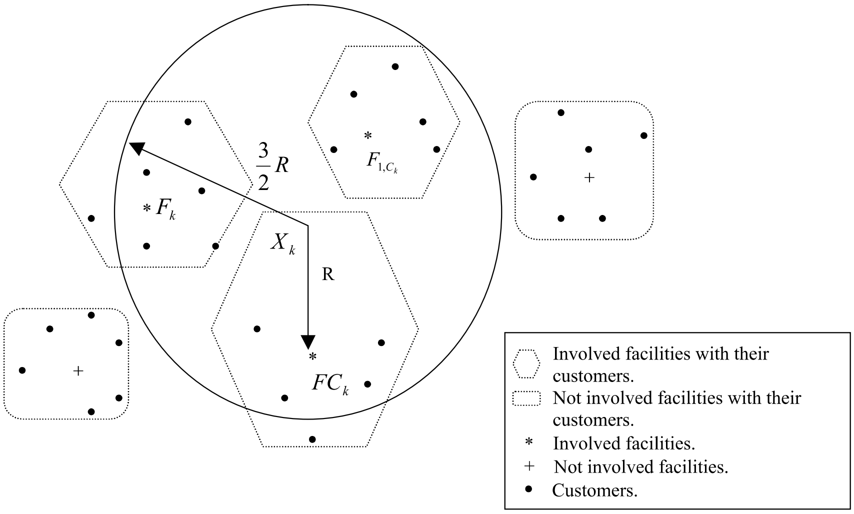
$\frac{3}{2}R$
$*F_k$
$*F_{1,C_k}$
X_k
R
$*FC_k$
Involved facilities with their customers.
Not involved facilities with their customers.
* Involved facilities.
+ Not involved facilities.
● Customers.

Fig. 9. Illustration of smaller neighbourhood.

According to Fig. 8, we apply the TP based on $|H^*|$ facilities ($|H^*| < M$) and $A(H^*)$ the new set of allocated customers instead of n where $|A(H^*)| < n$. Note that the selection of neighbourhood is carried out using a distance-related criterion but other procedures such as the use of the Voronoi diagram are also possible. Though the latter scheme may obviously select more precisely those affected facilities, since the process of alternating between location–allocation requires several iterations, our approximation scheme in detecting the affected facilities is reasonably appropriate. This reduction scheme is embedded into our method when we are solving the TP for M times while temporary removing the cluster's candidates. The effect of such a reduction is demonstrated in our computational results in the next section.

5. Computational results

The proposed heuristics are written in *For*tran90 and run on Sun Enterprise Workstation 450 running Solaris 2.6. We used the four test problems given in the literature for the uncapacitated case, see [3]. These are the 50-fixed points, the 287-fixed points, 654-fixed points and the 1060-fixed points test problems. The weight of all customers is set to unity except for the 287-fixed point problem. The algorithms are applied to the test problems to solve for 2–25 open facilities for the 50-fixed points problem and 5–50 with an increment of 5 for the other three problems. To evaluate the performance of our heuristics, we present the computational results obtained when solving the problem using the ATL and the ATLAL.

As these instances do not include the capacity of the facilities, we generate the capacity of the facility as $b = \left\lceil \frac{\sum_{j=1}^{n} w_j}{M} \right\rceil$. The value of b is for some facilities either larger or smaller than the total demand of the allocated customers to the facilities for the case of the uncapacitated problem. Note that in some cases, the total supply of the facilities will exceed the total customers demand as we are using the smallest integer which is greater than $\frac{\sum_{j=1}^{n} w_j}{M}$. In this case, a dummy customer with a unit transportation cost of 0 and a demand equals to the remaining demand is added. This dummy customer is used only when solving the TP, but not at the location and allocation stages.

The values of the parameters K and ϵ are found empirically using limited experiments. These include $K = 50$ as required in Fig. 1, and $\epsilon = 0.0001$ as referred in several places throughout the text.

We compute the % deviation based on the cost found for the multisource Weber problem. As these solutions were already reported in the literature to be optimal or very close to optimal (see [3]), we can therefore use such solution costs as lower bounds in our experiments. These are the optimal solutions for the 50 and 287-fixed points problems and the best known solutions for the 654 and 1060-fixed points problems given in [3]. The deviation is then computed based on these lower bounds as follows:

$$\text{dev}(\%) = \frac{F_{\text{best}} - F_{\text{LB}}}{F_{\text{LB}}} \times 100,$$

where F_{best} is our overall best solution cost and F_{LB} refers to the lower bound or 'best' cost for the uncapacitated case. We also record the overall average deviation (OAD) for the instances for each of the four test problems.

We conducted two experiments. In the first one, our aim is to select, based on the smallest data set, the most appropriate cluster type method which will then be used in our second experiment. This variant is then used with the reduction neighbourhood when dealing with the larger data sets.

5.1. Experiment 1: The choice of the variant for the large capacitated problems

The decision is based on the solution quality (measured by the average deviation) and the average computing time for the SSA, MSA and ISA using the all-clusters method (A: single all cluster, B: multi all cluster and C: intermediate all cluster, respectively) without the neighbourhood reduction for the 50-fixed point problem. These three algorithms are applied to solve for 2–25 open facilities. The average deviation obtained through SSA was 10.20% using 23.67 seconds of computing time, 13.11% through MSA with 0.81 seconds and 10.21% through ISA with 9.10 seconds. The detailed results of all the test problems can be found in [14].

According to our limited experiments, ISA gives much lower average deviation than MSA but a fraction higher than SSA. In terms of computing time, ISA takes much shorter time than SSA but slightly longer than MSA. Taking into consideration both the solution quality and the computational time, we conclude that ISA is the most appropriate method and therefore for convenience we proceed with the use of the neighbourhood reduction procedure on this method only.

5.2. Experiment 2: The choice of the cluster type method

The results for each test problem are summarised in Table 1. Columns 1–3 give the number of customers, the number of facilities and the lower bound (LB) respectively. The rest four double columns represent the % deviation from the LB and the computing time in seconds needed for the ATL (existing method in [6]), ATLAL (the modified Cooper's ATL heuristic), and our two new variants respectively that are intermediate all cluster + reduction and enhanced Intermediate (ISA using the some-clusters method) + reduction. The computing time given here excludes the time for generating the initial starting locations for the uncapacitated problem as this is almost negligible. It could also be noted that as ATL is relatively much faster than the others, it may be useful to test this simple method starting from several initial solutions as this may improve the current solution found by the present implementation of ATL.

From our experiments, it can be seen that the neighbourhood reduction does give a significant improvement in the computing time especially for enhanced intermediate + reduction. For instance, for the 50-fixed point problem, the computing time is reduced by up to 58%. But for this procedure, the cost is slightly inferior compared to the procedure where the full TP is solved for every cluster in intermediate all cluster + reduction. However, for the

Table 1
The ATL, ATLAL, intermediate all cluster + reduction and enhanced intermediate + reduction results for solving the continuous capacitated problem

n	*M*	LB	Previous		Modified ATL		New techniques			
			ATL		ATLAL		Intermediate all cluster + reduction		Enhanced intermediate + reduction	
			Dev. (%)	Time (seconds)	Dev. (%)	Time (seconds)	Dev. (%)	Time (seconds)	Dev. (%)	Time (seconds)
50	2	135.52	0.84	0.01	0.99	0.38	0.83	0.56	0.83	0.54
	3	105.21	1.06	0.01	1.05	0.43	1.02	0.70	1.02	0.65
	4	84.15	14.65	0.02	3.23	0.40	2.76	0.67	2.76	0.66
	5	72.24	12.37	0.02	6.70	0.47	5.94	1.14	5.94	0.93
	6	60.97	0.83	0.02	0.85	0.53	0.85	0.76	0.85	0.75
	7	54.50	4.75	0.03	5.21	0.84	3.35	2.17	3.35	2.12
	8	49.94	5.33	0.04	2.76	0.72	2.76	1.06	2.76	1.04
	9	45.69	9.85	0.04	4.38	1.01	4.05	3.05	4.05	2.81
	10	41.69	18.58	0.02	17.50	1.03	15.72	4.21	15.72	3.76
	11	38.02	11.62	0.04	7.39	1.29	6.57	3.42	6.57	3.35
	12	35.06	16.10	0.05	1.98	1.20	1.98	1.40	1.98	1.38
	13	32.31	27.87	0.03	10.07	1.36	8.83	2.81	8.83	2.64
	14	29.66	9.69	0.07	7.20	1.44	6.62	2.86	6.62	2.79
	15	27.63	14.81	0.05	6.77	1.71	6.49	3.68	6.49	3.08
	16	25.74	15.88	0.06	6.83	1.65	5.93	4.41	5.93	4.35
	17	23.99	17.80	0.03	9.34	1.66	7.92	6.47	7.92	6.24
	18	22.29	22.14	0.04	7.47	1.61	6.89	6.85	6.89	6.35
	19	20.64	17.98	0.06	7.36	1.63	7.32	5.97	7.32	5.92
	20	19.36	13.97	0.05	17.59	1.94	17.43	9.56	17.43	9.88
	21	18.08	39.58	0.07	16.63	2.33	16.41	9.53	16.41	9.08
	22	16.82	34.70	0.08	13.90	2.08	13.60	6.21	13.60	6.18
	23	15.61	29.76	0.09	16.44	2.63	16.12	6.35	16.12	6.28
	24	14.44	20.82	0.10	15.35	2.18	15.35	5.92	15.35	5.88
	25	13.30	110.13	0.06	72.08	1.68	70.96	6.25	70.96	6.00
OAV			19.63	0.04	10.80	1.34	**10.24**	4.00	**10.24**	$\boxed{3.86}$
287	5	9715.63	11.61	0.60	7.95	8.09	7.90	56.67	7.90	56.15
	10	6705.04	32.19	1.07	28.28	15.78	25.77	539.11	25.77	467.42
	15	5224.70	45.88	1.34	40.81	26.77	33.82	1837.17	33.82	1583.15
	20	4148.84	48.62	3.57	53.20	30.70	38.99	2532.21	38.63	2451.81
	25	3348.71	63.17	3.68	68.02	37.71	44.37	3442.54	44.97	2589.46
	30	2716.91	65.63	5.51	92.72	59.84	54.23	6365.81	54.27	5369.35
	35	2238.18	107.55	7.41	112.02	74.73	69.27	8310.36	69.56	4815.19
	40	1900.84	149.34	6.51	148.19	89.40	84.96	10948.99	88.25	4155.36
	45	1630.31	293.58	8.98	167.56	125.38	94.82	12169.82	99.80	2987.60
	50	1402.58	383.08	8.73	267.65	183.21	115.94	19814.90	119.22	3593.01
OAV			120.06	4.74	98.64	65.16	**57.01**	6601.76	**58.22**	$\boxed{2806.85}$
654	5	209068.80	54.00	1.73	58.20	9.79	54.00	62.72	54.00	63.45
	10	115339.03	42.81	2.63	48.70	20.62	42.81	136.94	42.81	110.84
	15	80177.04	67.69	4.90	75.55	52.63	67.69	852.92	67.69	576.32
	20	63389.02	71.97	5.60	74.70	60.20	69.37	1086.59	69.37	788.59
	25	52209.51	66.45	15.42	53.99	117.95	47.52	3842.79	47.52	2382.54
	30	44705.19	83.41	15.14	91.79	66.81	86.77	852.18	86.78	275.71
	35	39257.27	95.64	15.67	82.28	193.61	78.13	5772.53	78.75	1604.92
	40	35704.41	49.78	18.61	45.90	189.50	44.25	4740.56	44.25	1547.38
	45	32306.97	80.47	27.99	61.79	226.92	59.23	5112.34	59.23	846.86
	50	29338.01	43.47	36.11	43.07	349.55	41.96	20333.38	42.09	3568.67
OAV			65.57	14.38	63.60	128.76	**59.17**	4279.29	**59.25**	$\boxed{1176.53}$

(*continued on next page*)

Table 1 (*continued*)

n	*M*	LB	Previous		Modified ATL		New techniques			
			ATL		ATLAL		Intermediate all cluster + reduction		Enhanced intermediate + reduction	
			Dev. (%)	Time (seconds)	Dev. (%)	Time (seconds)	Dev. (%)	Time (seconds)	Dev. (%)	Time (seconds)
1060	5	1851879.88	1.06	5.60	1.20	54.44	1.06	370.85	1.06	266.73
	10	1249564.75	3.14	11.24	3.76	110.14	3.11	1167.35	3.11	562.17
	15	980132.13	1.73	47.67	2.02	287.13	1.63	4063.94	1.63	1123.70
	20	828802.00	3.56	29.45	3.90	254.62	3.42	5116.90	3.42	2514.43
	25	722061.19	6.08	47.75	4.72	780.42	3.87	30551.64	4.18	12072.97
	30	638263.00	5.68	58.44	4.57	868.19	3.92	45191.83	3.96	6789.35
	35	577526.63	8.13	63.36	3.83	548.16	3.35	21713.60	3.37	5279.45
	40	529866.19	7.14	123.59	6.75	1139.13	6.02	47253.61	6.03	21112.48
	45	489650.00	10.95	87.97	9.24	1582.52	7.83	120072.31	7.89	34370.33
	50	453164.00	9.65	111.25	7.39	1678.53	5.87	130753.06	5.87	75512.33
OAV			5.71	58.63	4.74	730.33	**4.01**	40625.51	**4.05**	$\boxed{\textbf{15960.40}}$

Bold: Good solution quality; $\boxed{\textbf{Bold}}$: CPU best amongst the two new techniques.

50-fixed point problem, this method reduces the average deviation by up to 48% from the average of deviation given by ATL, 52% for the 287-fixed point problem, 10% for the 654-fixed point problem and 29% for the 1060-fixed point problem. It is observed that the one additional configuration in enhanced intermediate does not give much contribution to the final solution.

The summary of the performance of all the methods given in Table 1 when represented by the solution quality and the computing time is shown in Fig. 10.

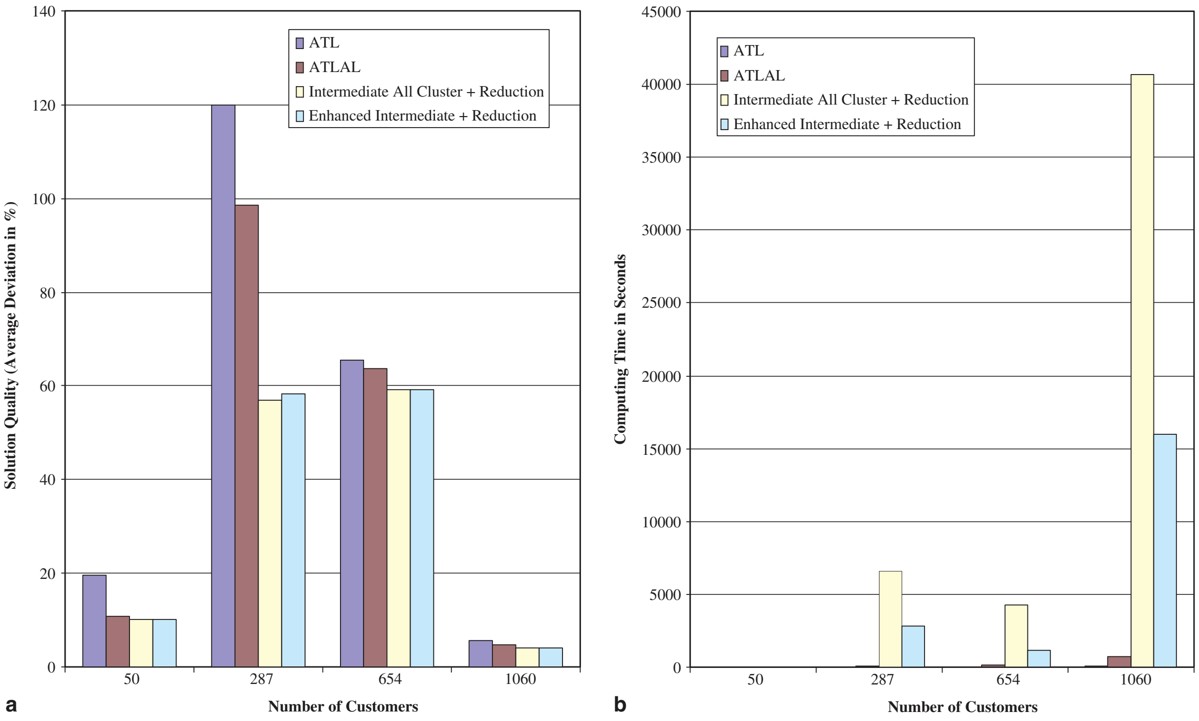

Fig. 10. A summary of solution quality and computational time for all the methods. (a) Solution quality (b) computational time.

6. Conclusion and possible research issues

A perturbation-based heuristic is proposed to solve the capacitated continuous location–allocation problem which appears to have been scarcely investigated in the past. The heuristic uses the Furthest Distance Rule method to generate the initial starting locations for the uncapacitated problem though other rules were also tested. The uncapacitated problem is then solved using different starting locations for K times but only a sample of configurations are chosen as the starting locations for the capacitated problem. The sample is selected using a diversity scheme based on cost. Initially, the capacitated problem is solved by the alternating transportation-location–allocation-location (ATLAL) heuristic and later, a perturbation scheme based on borderline customers is put forward to improve the obtained solution. A neighbourhood reduction technique is embedded into the perturbation scheme when solving the TPs with a considerable reduction in computational effort without a detriment in solution quality. Encouraging results are obtained when compared to the ATL and its enhanced version the ATLAL. These comparisons are based on the lower bounds which are taken as those optimal or near optimal solutions published for the uncapacitated case. Our obtained solutions could be used in future as benchmarks for those researchers interested in tackling this challenging continuous capacitated location problem.

The present work can be enhanced by considering a modification within the TP to further reduce the computing time. For instance, instead of starting the TP from the very beginning, we can use the current location and allocation as the basic feasible solution and continue the search to find the new optimal TP solution. A possible approach would be to develop also a suitable meta-heuristic to generate even better solutions. From a practical research viewpoint, it would also be interesting to tackle the capacitated continuous location problem with an unknown number of facilities by incorporating the facility fixed cost into the model. The fixed cost can be considered either constant (fixed charge) or throughput-dependent and/or zone-related. The authors are currently investigating some of the above issues.

Acknowledgements

The authors would like to thank the Malaysian Government for the sponsorship of the first author. We are also grateful to the referees for their constructive comments that improved both the content as well as the presentation of the paper.

References

[1] M.C. Agar, S. Salhi, Lagrangean heuristics applied to a variety of large capacitated plant location problems, Journal of The Operational Research Society 49 (1998) 1072–1084.

[2] J.E. Beasley, Lagrangean heuristic for location problems, European Journal of Operational Research 65 (1993) 383–399.

[3] J. Brimberg, P. Hansen, N. Mladenovic, E.D. Taillard, Improvements and comparison of heuristics for solving the uncapacitated multisource Weber problem, Operations Research 48 (3) (2000) 444–460.

[4] J. Brimberg, N. Mladenovic, A variable neighbourhood algorithm for solving the continuous location–allocation problem, Studies in Locational Analysis 10 (1996) 1–12.

[5] L. Cooper, Heuristic methods for location–allocation problems, SIAM Review 6 (1964) 37–53.

[6] L. Cooper, The transportation-location problem, Operations Research 20 (1972) 94–108.

[7] W. Domschke, A. Drexl, ADD-heuristics starting procedures for capacitated plant location models, European Journal of Operational Research 21 (1985) 47–53.

[8] M. Eben-Chain, A. Mehrez, G. Markovich, Capacitated location allocation problem on a line, Computers and Operations Research 29 (5) (2002) 459–470.

[9] M.D.H. Gamal, S. Salhi, Constructive heuristics for the uncapacitated continuous location–allocation problem, Journal of Operational Research Society 52 (2001) 821–829.

[10] S.K. Jacobsen, Heuristics for the capacitated plant location model, European Journal of Operational Research 12 (1983) 253–261.

[11] B. Khumawala, An efficient heuristic procedure for the capacitated warehouse location problem, Naval Research Logistics Quarterly 21 (1974) 609–623.

[12] S. Salhi, M.D.H. Gamal, A genetic algorithm-based approach for the uncapacitated continuous location–allocation problem, Annals of Operations Research 123 (2003) 203–222.

[13] T.J. Van Roy, A cross decomposition algorithm for capacitated facility location, Operations Research 34 (1986) 145–163.

[14] Z.M. Zainuddin, Constructive and tabu search heuristics for the capacitated continuous location–allocation problem, PhD thesis, School of Mathematics and Statistics, University of Birmingham, UK, 2004.