

RESEARCH ARTICLE

Discretization based heuristics for the capacitated multi-facility Weber problem with convex polyhedral barriers

M. Hakan Akyüz*

*Department of Industrial Engineering, Faculty of Engineering and Technology, Galatasaray University, Turkey
mhakyuz@gsu.edu.tr*

ARTICLE INFO

Article History:

Received 04 September 2016

Accepted 13 September 2017

Available 23 October 2017

Keywords:

Polyhedral barriers

Location-allocation

Mixed-integer linear programming

Discretization heuristics

Lagrangian relaxation

AMS Classification 2010:

49M25, 90C11, 90C59

ABSTRACT

The Capacitated Multi-facility Weber Problem (CMWP) tries to determine the location of I capacitated facilities in the plane and to satisfy demand of J customers so as to minimize the total transportation cost. The CMWP assumes that the facilities can be located anywhere on the plane and customers are directly connected to them. This study considers an extension of the CMWP where there exist convex polyhedral barriers blocking passage and locating facilities inside. As a result, the distances between facilities and customers have to be measured by taking into account the polyhedral barriers. The CMWP with convex polyhedral barriers (CMWP-B) is a non-convex problem that is difficult to solve. We propose specially tailored discretization based heuristic procedures. Since CMWP-B is novel to the literature, a new set of test problems is randomly generated. Then, the performance of the suggested methods are tested on the test instances. Our results imply that the suggested heuristics yield quite accurate and efficient solutions for the CMWP-B.



1. Introduction

The Capacitated Multi-facility Weber Problem (CMWP) tries to locate I facilities with capacity restrictions in the plane and to meet the demand of J customers while minimizing the total cost of transportation. The objective function of the CMWP is shown to be neither convex nor concave [1]. When capacity restrictions of the facilities are removed, the CMWP becomes the so called Multi-facility Weber Problem (MWP). Both of the MWP and CMWP are NP-hard as shown by Sherali and Nordai [2], and Meggido and Supowit [3], respectively. Moreover, they both generalize the Weber Problem (WP) that aims to find the optimal location of a single facility. The objective function of the WP is convex, and thus, easy to solve. However, this is not true for its generalizations of the MWP and CMWP as additional allocation decisions have to be made for them. One can partition the customer set into

non-intersecting subsets each of which is served by an uncapacitated facility for the MWP. This indicates that each customer is served from exactly one facility at the optimal solution of the MWP. On the other hand, customer demands may need to be met from multiple facilities for the CMWP when capacity restrictions are imposed. Nevertheless, all of the WP, MWP and CMWP assume that each facility can be freely located in the plane. Such an assumption can be misleading in practice since there may exist physical areas that obstruct to travel and to open facilities inside. For example, lakes, mountains, glaciers and forests are natural barrier areas which can prohibit both travelling and facility location inside. Indeed, traveling (or passage) is frequently blocked by existing work-shop (office room) areas in a manufacturing plant (in an office building). This work focuses on an extension of the CMWP where the existence of convex polyhedral barriers

*Corresponding Author

are considered as obstacles to pass inside and to locate facilities. Namely, the CMWP with convex polyhedral barriers (CMWP-B) is addressed. We suggest several discretization based heuristic approaches to efficiently solve the CMWP-B.

Mostly, the transportation cost between a facility and customer is a function of the distance and the amount of shipment between them. In particular, the distance is usually modelled with the Euclidean, squared Euclidean, rectilinear and ℓ_r (with $1 \leq r < \infty$) norms in location-allocation type problems such as the CMWP. Alpaydın et al. [4] and Brimberg et al. [5] provide excellent surveys that include techniques and functions to model distances. In this work, the distances between facilities and customers are measured by the so called “barrier distance” which uses the Euclidean distance, i.e. ℓ_2 , as its underlying distance measure. Observe that the Euclidean distance may not be feasible when polyhedral barriers exist between facilities and customers. Therefore, “barrier distance” is determined as the shortest path, which does not pass through the barrier regions where the edge costs are calculated with the Euclidean distance, between a facility and a customer. Clearly, the “barrier distances” require extra efforts to calculate the distances which makes the CMWP-B a more realistic and difficult problem to solve than the unrestricted CMWP. Nonetheless, the “barrier distance” still preserves the metric properties [6].

This work has the following contributions. For all we know, the CMWP-B is novel to the literature and we suggest a mathematical programming formulation for it. Next, we offer three discretization based (DB) heuristics which reduce the continuous location space into a discrete space using a discretization strategy. The first DB heuristic employs the solution of an approximating mixed-integer linear programming (MILP) formulation suggested for the CMWP-B. The second DB heuristic applies a Lagrangean Relaxation (LR) scheme on the suggested MILP formulation. The third DB heuristic performs a Tabu Search (TS) using neighborhoods defined over the discretized location space. All upper bounds are calculated with alternate location-allocation type heuristics for the DB heuristics. We generate new test instances for the CMWP-B where they inherit the data of the standard CMWP instances. For that purpose, convex polyhedral barriers are randomly constructed such that the feasibility of the CMWP-B instances is maintained. Lastly, we perform extensive computational experiments on randomly generated test instances derived from the standard CMWP instances.

The remainder of this study is organized as follows. In Section 2 a brief review of the relevant literature is presented. This is followed by the mathematical programming formulation of the CMWP-B in Section 3. Section 4 presents details of the suggested DB heuristics. Section 5 is where we discuss our computational findings. Lastly, the paper is concluded and future research directions are given in Section 6.

2. Literature Review

In this section, we present a short review of the relevant literature. Both the MWP and CMWP have attracted the attention of researchers starting with the seminal works by Cooper [1,7]. There exist several exact solution procedures suggested for the MWP ([8–11]) as well as heuristic approaches for the MWP ([7,12–18]). Furthermore, we can cite the works by Cooper [1], Sherali et al. [19] and Akyüz et al. [20] as examples of exact solution methods developed for the CMWP. On the other hand, we can mention Cooper’s [7] Alternate Location-Allocation (ALA) type heuristics [21,22], Discrete Approximation (DA) heuristics [23,24] and metaheuristics [25] developed for the CMWP. Akyüz et al. [26,27] also offer heuristic procedures for a multi-commodity extension of the CMWP. A survey on the location-allocation type problems can be found in the work by Brimberg et al. [28].

As an extension of the single facility WP, the WP-B arises when there exist barriers which prohibit to travel inside and to locate a facility. Katz and Cooper [29] introduce the WP-B having a single circular barrier region. The WP-B with rectilinear distances is considered in the works by Larson and Sadiq [30] and Batta et al. [31]. Aneja and Parlar [32] design an algorithm for the WP-B as well as for a variant of the WP-B where travelling is permitted within the barriers. Butt and Cavalier [33] address the WP-B where barriers are convex polygons and propose an algorithm which yields local optimum solution. Klamroth [34] derives a reduction result for the WP-B having convex polyhedral barriers. Then, an exact and a heuristic procedure is developed for the WP-B. McGarvey and Cavalier [35] develop a branch and bound approach that partitions the continuous location space to optimally solve the WP-B in the presence of polyhedral barriers. Bischoff and Klamroth [36] deal with the WP-B where the barriers are convex polyhedral sets. Their method benefits from the reduction result by Klamroth [34] and decomposes the WP-Bs into multiple subproblems (i.e. WPs) each can be solved over a convex and bounded region. The

authors [36] apply a genetic algorithm to decrease the number of subproblems to solve.

The MWP with convex polyhedral barriers (MWP-B) is first considered by Krau [10] which applies a column generation approach based on partitioning of the customer set. Bischoff et al. [37] suggest two ALA type heuristics that have similar allocation phases for the MWP-B. Their first ALA type heuristic alternately solves multiple WP-Bs and set partitioning problems resulting in an inefficient algorithm. The second ALA type heuristic suggested by Bischoff et al. [37] yields better results. In the location phase, the reduction result of [34] is extended to the multi-facility case. Unlike their first ALA type heuristic, this results in solving multiple WPs over convex restricted regions and increase the efficiency of the location phase for the second ALA type heuristic. In what follows, we give a formal definition of the CMWP-B.

3. Capacitated Multi-Facility Weber Problem with Convex Polyhedral Barriers

Let I , J and P denote the number of facilities, the number of customers and the number of polyhedral barriers, respectively. The coordinates of customer j is shown by $\mathbf{a}_j = (a_{j1}, a_{j2})^T$ and its demand is denoted by q_j . The parameter s_i stands for the capacity of facility i . There are two decisions to be made for the CMWP-B: location decisions and allocation decisions. Then, the unknown location of facility i is represented as $\mathbf{x}_i = (x_{i1}, x_{i2})^T$. The unknown amount of flow between facility i and customer j is denoted by f_{ij} . c_{ij} is the cost of transportation per unit flow between customer j and facility i per unit distance. A polyhedral barrier p is indicated with set \mathcal{B}_p and the union of barriers are stated with the set \mathcal{B} , i.e. $\mathcal{B} = \bigcup_{p=1}^P \mathcal{B}_p$. Now, the feasible region to locate facilities is given with the set \mathcal{X} that can be defined as $\mathcal{X} = \mathbb{E}^2 \setminus \mathcal{B}$ where \mathbb{E}^2 is the two dimensional Euclidean space. The Euclidean norm, i.e. $\|\mathbf{x}_i - \mathbf{a}_j\|_2 = [(x_{i1} - a_{j1})^2 + (x_{i2} - a_{j2})^2]^{1/2}$, is used as the underlying distance measure to calculate the “barrier distance” between facility i and customer j . Note that barrier distance is represented with $d_{\mathcal{B}}(\mathbf{x}_i, \mathbf{a}_j)$ between facility i and customer j . The details about its calculation will be discussed later on. Now a mathematical programming formulation of the CMWP-B is given as follows.

CMWP-B:

$$\min Z = \sum_{i=1}^I \sum_{j=1}^J c_{ij} f_{ij} d_{\mathcal{B}}(\mathbf{x}_i, \mathbf{a}_j) \quad (1)$$

$$\text{s.t.} \quad \sum_{j=1}^J f_{ij} = s_i \quad i = 1, \dots, I, \quad (2)$$

$$\sum_{i=1}^I f_{ij} = q_j \quad j = 1, \dots, J, \quad (3)$$

$$f_{ij} \geq 0 \quad i = 1, \dots, I; j = 1, \dots, J, \quad (4)$$

$$\mathbf{x}_i \in \mathcal{X} \quad i = 1, \dots, I. \quad (5)$$

The objective function (1) is to minimize the sum of total transportation cost between facilities and customers. Notice that constraints (2)-(4) are the constraints of the well-known Transportation Problem (TP). Constraints (2) state that the total flow sent from a facility i is equal to its capacity s_i . Constraints (3) imply that the demand q_j of customer j is exactly met. Constraints (4) ensure the nonnegativity of flows. Constraints (5) guarantee that facilities are placed within the feasible region \mathcal{X} . It is possible to further restrict the set of feasible region \mathcal{X} . Wendell and Hurter [38] show that an optimal solution of the WP can be found within the convex hull of customers. The result by Wendell and Hurter [38] can be directly generalized to the WP-B as well as to the CMWP-B in the case where barrier regions stay within the convex hull of customer locations. In this case, it is enough to consider the region within convex hull of customers that is outside of the polyhedral barriers. However, unlike the CMWP, the resulting feasible region to locate facilities is non-convex for the CMWP-B. Moreover, distances between facilities and customers are measured with barrier distances $d_{\mathcal{B}}(\mathbf{x}_i, \mathbf{a}_j)$ for the CMWP-B. Observe that, the CMWP employs the Euclidean distance between each facility and customer. However, barrier distance is calculated as the shortest path distance between any two points when there exist barriers. The CMWP-B can be reduced to the CMWP when all barriers are removed. The optimal objective value of the CMWP is a lower bound on the optimal objective value of the CMWP-B since barrier distance between any two points is larger than or equal to their Euclidean distances. That is to say, the CMWP-B is more difficult to solve than the CMWP which is NP-hard. This motivates the implementation of efficient heuristic methods for the CMWP-B.

The calculation of the barrier distance $d_{\mathcal{B}}(\cdot)$ requires additional efforts. Notice that convex polyhedral barriers can be represented with their extreme points. Let $\mathbf{e}_{pb} = (e_{pb1}, e_{pb2})^T$ be the coordinates of their extreme points for each polyhedral barrier $p = 1, \dots, P$ with corresponding extreme points $b = 1, \dots, B_p$. Here, B_p gives the number of extreme points defining the polyhedral barrier p represented with the set \mathcal{B}_p . It should be emphasized that considering only convex polyhedral barriers does not cause a loss of generality. McGarvey and Cavalier [35] have shown that optimal solution of the WP-B can be found outside of the convex hull of barriers which does not contain a customer location. As the convex hull of any barrier (convex or non-convex) can be approximated with a convex polyhedron, barriers are assumed to be convex polyhedrons. The barrier distance between two points \mathbf{a} and \mathbf{e} is defined as the distance of the shortest feasible path from \mathbf{a} to \mathbf{e} . To ensure that a path is feasible, it should not coincide with the interior of polyhedral barriers. The concept of visibility graph, e.g. Ghosh [39], is used to ensure that a path is feasible. Two points are said to be visible to each other if and only if they can be connected with a line which does not pass through the interior of polyhedral barriers. Nodes of the visibility graph are customer locations and the extreme points of polyhedral barriers. An edge is defined in the visibility graph if and only if two points, say customer \mathbf{a} and extreme point \mathbf{e} are visible to each other. Edge costs are calculated with the Euclidean distance $d(\mathbf{a}, \mathbf{e}) = [(a_1 - e_1)^2 + (a_2 - e_2)^2]^{1/2}$. Clearly, when two nodes are not visible to each other, there is no corresponding edge in the visibility graph. Finally, the barrier distance between two points (nodes) $d_{\mathcal{B}}(\mathbf{a}, \mathbf{e})$ is equal to the cost of the shortest path from \mathbf{a} to \mathbf{e} on the visibility graph. Shortest paths can be obtained by a shortest path algorithm such as the Dijkstra's algorithm [40].

The discussion above only considers the barrier distances among the nodes existing in the visibility graph. For an arbitrary point \mathbf{x} within the set of feasible locations, i.e. $\mathbf{x} \in \mathcal{X}$, the calculation of barrier distances is slightly different. One may consider re-constructing the visibility graph from scratch such that \mathbf{x} is also added into the node set. Fortunately, it is sufficient to determine only the set of nodes visible from \mathbf{x} , say $\mathcal{V}_{\mathbf{x}}$, in the visibility graph. Then, the barrier distances from the point \mathbf{x} to any customer location \mathbf{a} is measured using the formula:

$$d_{\mathcal{B}}(\mathbf{x}, \mathbf{a}) = \min_{\mathbf{h} \in \mathcal{V}_{\mathbf{x}}} \{d(\mathbf{x}, \mathbf{h}) + SPD(\mathbf{h}, \mathbf{a})\}, \quad (6)$$

where \mathbf{h} stands for the customer locations and/or extreme points of barriers that are visible from the point \mathbf{x} . $SPD(\mathbf{h}, \mathbf{a})$ represents the shortest path distance between points $\mathbf{h} \in \mathcal{V}_{\mathbf{x}}$ and \mathbf{a} in the visibility graph. The barrier distance function $d_{\mathcal{B}}(\mathbf{x}, \mathbf{a})$ defines a metric on \mathcal{X} satisfying the following properties of a metric: positivity, definiteness, symmetry and triangle inequality. Klamroth [6] is an excellent reference to resort for more details on the single facility location problems with barriers and their properties. In the left-hand side of Figure 1, an example with four customers and single tetragon barrier is given. In the right-hand side of Figure 1, the corresponding visibility graph is illustrated.

4. Discretization Based (DB) Heuristics

The CMWP-B reduces to solving I WP-B's when the amount of shipments are known. This results in a two dimensional minimum location problem in continuous space. When all barriers are removed, the resulting problem is a classical WP and using the results by Wendell and Hurter [38] and Hansen et al. [41], an optimal solution can be found within the convex hull of customer locations. Clearly, the single WP is easy to solve because it is a convex programming problem. That is to say, it can be solved by the Weiszfeld's [42] algorithm or one of its generalizations ([43, 44]). On the other hand, the objective function of the CMWP-B is non-convex and the problem itself is not easy to solve on continuous space. This sparks a discretization strategy to solve the CMWP-B using a discretized location space. It is conceivable that the CMWP-B can be transformed into a MILP problem formulation when facility locations are chosen from a set of candidate locations. Indeed, the resulting MILP problem formulation may yield the optimal solution of the CMWP-B when the set of candidate locations to place facilities includes the optimal facility locations. Obviously, it is not possible to determine the optimal facility locations in advance when constructing the set of candidate facility locations. However, solving a MILP problem considering a set of candidate facility locations produces an approximate solution for the CMWP-B. For that purpose, a systematic way of choosing candidate facility locations is of high importance and may provide good solutions for the CMWP-B. In fact, such a strategy is formerly offered by Hansen et al. [14] and Aras et al. [23] for the MWP and CMWP, respectively. They use customer locations as the set of candidate facility locations and

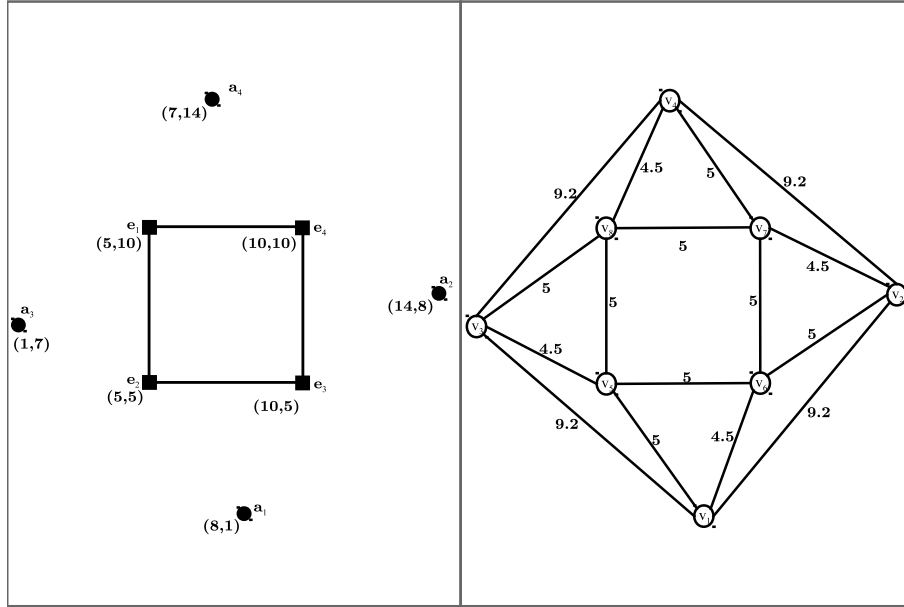


Figure 1. An illustrative example with 4 customers, a single tetragon barrier and the corresponding visibility graph.

solve approximating MILPs. Hansen et al. [14] solve an approximating p-median problem and obtain highly accurate solutions for the MWP. Similar results also hold for the CMWP from the work by Aras et al. [23]. The discretization strategy that we pursue in this study is analogous to the previous ones [14,23]. In what follows, we first present the approximating MILP problem formulation and the first DB (DB-I) heuristic. This part also elaborates how we tackle with the difficulties imposed by the barrier distances for the single WP-B. Then, the efficiency of DB-I heuristic is improved by a LR scheme. This heuristic is denominated as DB-II. Lastly, a Tabu Search (TS) algorithm is employed to further improve the efficiency of the DA-I heuristic. This heuristic is called as DB-III heuristic.

4.1. Approximating MILP Problem Formulation: DB-I Heuristic

Let $k = 1, \dots, K$ denote candidate facility locations with known coordinates given as $\mathbf{v}_k = (v_{k1}, v_{k2})^T$. The decision variables w_{ijk} represent the amount of flow between facility i located at candidate point k and customer j . Binary variables u_{ik} take a value of 1 if and only if facility i is opened at candidate point k and 0 otherwise. c_{ijk} is the corresponding transportation cost for flow w_{ijk} . Specifically, it is calculated as $c_{ijk} = c_{ij}d_B(\mathbf{v}_k, \mathbf{a}_j)$. Now, an approximating MILP problem formulation of the CMWP-B can be stated as follows.

DA:

$$\min Z_{DA} = \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K c_{ijk} w_{ijk} \tag{7}$$

$$\text{s.t.} \quad \sum_{j=1}^J w_{ijk} = s_i u_{ik} \quad i = 1, \dots, I; \tag{8}$$

$$k = 1, \dots, K,$$

$$\sum_{i=1}^I \sum_{k=1}^K w_{ijk} = q_j \quad j = 1, \dots, J, \tag{9}$$

$$\sum_{k=1}^K u_{ik} = 1 \quad i = 1, \dots, I, \tag{10}$$

$$w_{ijk} \geq 0 \quad i = 1, \dots, I; j = 1, \dots, J; \tag{11}$$

$$k = 1, \dots, K,$$

$$u_{ik} \in \{0, 1\} \quad i = 1, \dots, I; \tag{12}$$

$$k = 1, \dots, K.$$

Here, constraints (8), (9) and (11) are analogous to the TP constraints (2)-(4) for the approximating MILP problem formulation of the CMWP-B. Constraints (10) ensure that every facility is opened at exactly one candidate facility location.

Before giving the details of the DB-I heuristic, we first introduce an improvement heuristic that is employed within the DB heuristics in the following. This improvement heuristic is an adaptation of the famous Alternate Location-Allocation (ALA) heuristic that is offered for the MWP by Cooper [7]. It consists of alternately solving two

subproblems: allocation and location subproblems. These subproblems respectively arise when the facility locations and allocations between facility and customers are fixed. ALA heuristic repeats alternating until the objective function value does not change from one iteration to another. This also indicates that facility locations and allocation values do not significantly change and the objective value becomes stable. It can be observed that initialized from given facility locations, the CMWP-B reduces to solving a TP, which is the “allocation” subproblem, in order to determine allocations, i.e. flow between facilities and customers. TP can be solved straightforwardly using linear programming solvers. On the other hand, with a given feasible shipment plan, the CMWP-B can be decomposed into I single facility WP-Bs so called the “location” subproblems. Solving the resulting location subproblems, i.e. the WP-Bs, is not trivial and our approach for solving the location subproblems is summarized in the following. Recall that the WP-B is a non-convex problem and several solution approaches can be used to solve ([32, 33, 35]). Instead of an exact solution procedure, which can be prohibitive to use within an efficient heuristic approach, we prefer to make use of the Weiszfeld’s [42] procedure that solves the unrestricted WP optimally. Weiszfeld’s algorithm employs the gradient direction to update facility locations from one iteration to the other and eventually converges to the optimal facility locations. This procedure works well on a convex problem like the WP. However, there is no guarantee of optimality and it cannot be directly used for the non-convex WP-B. Nevertheless, observe that, when Weiszfeld’s algorithm ends up with a solution \mathbf{x}^* that is within the feasible location space outside of the polyhedral barriers, i.e. $\mathbf{x}^* \in \mathcal{X}$, then it is also optimal for the WP-B. Otherwise, when Weiszfeld’s algorithm terminates with a solution \mathbf{x}^* that is within a polyhedral barrier p , then, the facility location can be approximated by choosing the closest point $\hat{\mathbf{x}}$ on the border of the barrier p . In other words, our solution approach finds a solution for the location subproblems that yields approximate (or hopefully optimal) facility locations. This procedure is illustrated with Figure 2. This improvement heuristic solves the location and allocation subproblems as described until the objective value remains the same from one iteration to another. This heuristic is named as ALA with barriers (ALAB) heuristic. ALAB heuristic is frequently resorted within our DB heuristics.

The DB-I heuristic works as follows. First, the DA formulation is solved using the set of candidate facility locations consisting of customer locations. The DA formulation yields the facility locations which are used to initialize the improvement heuristic. Second, the ALAB heuristic is used to enhance the solution initialized with the facility locations obtained in the first phase. Then, the best solution found is reported as the outcome of the DB-I heuristic. Note that applying the ALAB heuristic in the second phase of the DB-I heuristic usually improves the initial solution obtained from the DA formulation.

4.2. A Lagrangean Relaxation (LR) Scheme: DB-II Heuristic

The DA formulation can be intractable for large problems and thus computationally very expensive to solve exactly. Therefore, it may be better to use an approximate solution approach. To this end, a LR scheme and subgradient optimization is employed to compute good feasible solutions for the DA. The demand constraints (9) are relaxed with Lagrangean multipliers μ_j to obtain the Lagrangean subproblem of DA, namely the LDA.

LDA($\boldsymbol{\mu}$):

$$\min Z_{LDA}(\boldsymbol{\mu}) = \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K (c_{ijk} - \mu_j) w_{ijk} + \sum_{j=1}^J \mu_j q_j \quad (13)$$

$$\begin{aligned} \text{s.t.} \quad & (8), (10), (11), (12), \\ & w_{ijk} \leq \min\{s_i, q_j\} \quad i = 1, \dots, I; \\ & j = 1, \dots, J; k = 1, \dots, K. \quad (14) \end{aligned}$$

Observe that constraints (14) are introduced as basic upper bounds on the flow quantities to the LDA formulation. Clearly, these constraints are redundant for the DA formulation. However, they may significantly improve the optimal objective value, say $Z_{LDA}^*(\boldsymbol{\mu})$, of the Lagrangean subproblem LDA. The last term of the objective function (13) is constant and LDA($\boldsymbol{\mu}$) can be further decomposed over the facilities. That is to say, solving the following subproblems for each facility $i = 1, \dots, I$ is equivalent to solving the Lagrangean subproblem LDA($\boldsymbol{\mu}$)

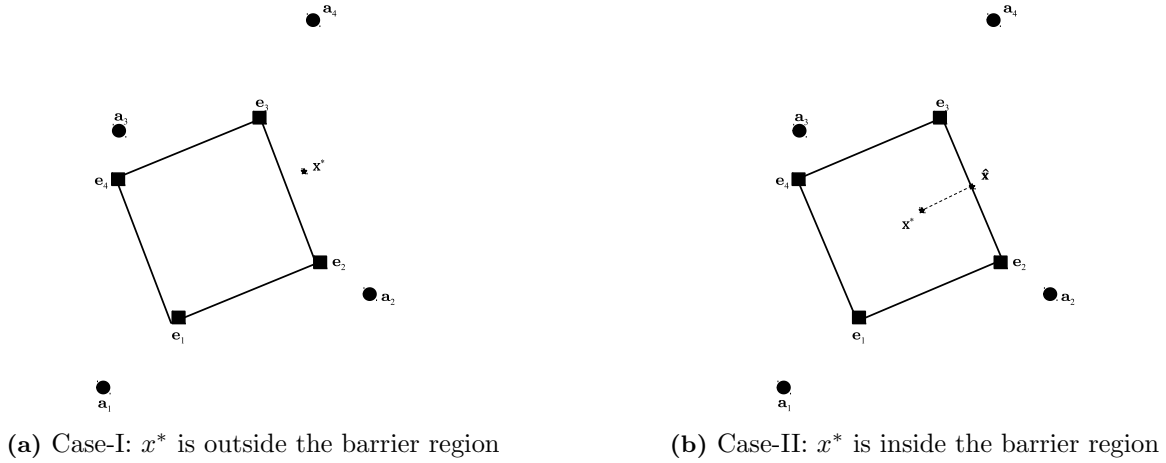


Figure 2. Two possible cases for the output of a Weiszfeld-like procedure.

$LDA_i(\boldsymbol{\mu})$:

$$\min Z_{LDA_i}(\boldsymbol{\mu}) = \sum_{j=1}^J \sum_{k=1}^K \bar{c}_{ijk} w_{ijk} \quad (15)$$

$$\text{s.t.} \quad \sum_{j=1}^J w_{ijk} = s_i u_{ik} \quad k = 1, \dots, K, \quad (16)$$

$$\sum_{k=1}^K u_{ik} = 1, \quad (17)$$

$$w_{ijk} \leq \min\{s_i, q_j\} \quad j = 1, \dots, J; \quad k = 1, \dots, K, \quad (18)$$

$$w_{ijk} \geq 0 \quad j = 1, \dots, J; \quad k = 1, \dots, K, \quad (19)$$

$$u_{ik} \in \{0, 1\} \quad k = 1, \dots, K. \quad (20)$$

where the unit costs \bar{c}_{ijk} are determined as $\bar{c}_{ijk} = (c_{ijk} - \mu_j)$ with a given Lagrange multiplier vector $\boldsymbol{\mu}$. The resulting subproblem $LDA_i(\boldsymbol{\mu})$ can be solved by an inspection procedure for each candidate facility location k . Note that, when facility i is placed at a candidate location k , then $LDA_i(\boldsymbol{\mu})$ further reduces to the following problem

$LDA_{ik}(\boldsymbol{\mu})$:

$$\min Z_{LDA_{ik}}(\boldsymbol{\mu}) = \sum_{j=1}^J \bar{c}_{ijk} w_{ijk} \quad (21)$$

$$\text{s.t.} \quad \sum_{j=1}^J w_{ijk} = s_i \quad (22)$$

$$w_{ijk} \leq \min\{s_i, q_j\} \quad j = 1, \dots, J, \quad (23)$$

$$w_{ijk} \geq 0 \quad j = 1, \dots, J. \quad (24)$$

$LDA_{ik}(\boldsymbol{\mu})$ is a continuous bounded knapsack problem that can be optimally solved in polynomial time [45]. This approach requires sorting of the cost coefficients for each candidate location

k . Hence, we need to run a sorting procedure K times. The least cost candidate point k^* is chosen as the optimal location of facility i , and thus for the subproblem $LDA_i(\boldsymbol{\mu})$. Now, the optimal value of $LDA_i(\boldsymbol{\mu})$ is calculated as $Z_{LDA_i}^*(\boldsymbol{\mu}) = \min_k \{Z_{LDA_{ik}}^*(\boldsymbol{\mu})\}$. It is conceivable that the values of binary variables are set as $u_{ik^*} = 1$ for the least cost candidate location k^* and $u_{ik} = 0$ for $k = 1, \dots, K$ and $k \neq k^*$. Once all subproblems $LDA_i(\boldsymbol{\mu})$ are solved, for a given Lagrange multiplier vector $\boldsymbol{\mu}$, the optimal value of the Lagrangean subproblem $LDA(\boldsymbol{\mu})$ is determined as $Z_{LDA}^*(\boldsymbol{\mu}) = \sum_{i=1}^I Z_{LDA_i}^*(\boldsymbol{\mu}) + \sum_{j=1}^J \mu_j q_j$. Notice that $Z_{LDA}^*(\boldsymbol{\mu})$ constitutes a lower bound on the optimal value of the DA for any Lagrange multiplier vector $\boldsymbol{\mu}$. The best lower bound can be obtained by solving the Lagrangean dual problem $\max_{\boldsymbol{\mu}} \{Z_{LDA}^*(\boldsymbol{\mu})\}$ that is accomplished using subgradient algorithm by Held et al. [46]. For the sake of brevity, we do not give details of the subgradient algorithm. The subgradient algorithm calculates upper bounds during its run, and hence, feasible solutions for the CMWP-B. For that purpose, the ALAB heuristic is used. The values of binary variables u_{ik} are used to determine the locations of each facility for each solution of the resulting Lagrangean subproblem $LDA(\boldsymbol{\mu})$ with given multiplier values $\boldsymbol{\mu}$. Lastly, the DB-II heuristic reports the best feasible solution obtained from the subgradient algorithm.

4.3. A Tabu Search Algorithm: DB-III Heuristic

Inspired with the promising results obtained by using the customer locations as the set of candidate facility location, the DB-III heuristic consists of applying a TS algorithm on the set of candidate facility locations. A simple neighborhood search

structure can be defined by exchanging the location of a facility i' at a candidate location k' with another one in order to move from one feasible solution to another one. This yields a feasible solution for the CMWP-B. Further, feasible solutions can be improved by applying a ALAB heuristic.

The suggested TS algorithm basically exchanges the location of a facility with another candidate location at each iteration. The selected candidate location is declared as tabu for that facility and its status can not be revoked during the tabu tenure. A greedy strategy of selecting the closest candidate location is applied to move from the current feasible solution to a neighbor solution. To increase the intensity of the neighbor search the closest N neighbor feasible solutions can be checked and the best feasible solution is picked as the new feasible solution. The TS algorithm searches over the candidate location set and reports the best solution found. For further details on TS we refer to the work by Glover and Laguna [47].

The TS algorithm uses a tabu list \mathcal{T} that keeps record of candidate facility locations which are declared as tabu for a duration (tabu tenure) say α iterations for each facility i . Each facility i is associated with a set of candidate locations which are declared as tabu. A tabu declared candidate location k for a facility i can not be in the current solution until the tabu tenure record, represented as $T(i, k)$, decreases to zero. Let t denote the current iteration number and θ stand for the maximum number of iterations completed by the TS, namely, the iteration limit of the TS. At each iteration, a facility i^* is chosen and a non-tabu candidate location k^* with $T(i^*, k^*) = 0$ is declared as tabu for i^* . The facility i^* is determined with respect to their facility index in the order from smallest facility index to largest facility index. In other words, if facility i is selected at iteration t then facility $i+1$ is the next facility to be selected at iteration $t+1$. Note that, when $i^* = I$ for the selected facility, the facility to be declared as tabu is the one with index number 1 at the next tabu iteration. Once tabu facility i^* is set at iteration t , the candidate point k^* is searched over the closest neighbors of the current candidate location, say k' , of facility i^* . Clearly, all facilities other than i^* maintain their facility locations determined at previous iteration $t-1$ and the new feasible solution only changes the location of facility i^* at iteration t . Let $k'_{(1)}$ be the closest candidate location to k' , $k'_{(2)}$ denote the second closest candidate location to k' , and so forth. The upper bound obtained at tabu iteration t and the best upper

bound found so far are respectively represented with Z_{UB}^t and Z_{Tabu}^{best} . The upper bound Z_{UB}^t is determined as the lowest upper bound value obtained by exchanging k' with the candidate locations of the set $\{k'_{(1)}, k'_{(2)}, \dots, k'_{(N)}\}$ one by one from the closest ($k'_{(1)}$) to farthest ($k'_{(N)}$) neighbor candidate location. Here, N is the number of neighbor solutions checked (or the width of the neighborhood) and $k'_{(N)}$ is the N^{th} closest candidate location to k' . This strategy is followed as long as Z_{UB}^t does not improve the best upper bound Z_{Tabu}^{best} . When an improvement is obtained, the neighborhood search is stopped. The best upper bound Z_{Tabu}^{best} is updated whenever the improvement occurs and the TS proceeds to the next iteration $t+1$ with the next facility. For example, when the feasible solution obtained by exchanging k' with the closest candidate location $k'_{(1)}$ yields a better upper bound than the best upper bound Z_{Tabu}^{best} , then Z_{UB}^t is calculated with candidate location $k'_{(1)}$ for facility i^* and Z_{Tabu}^{best} is updated. In addition, the current tabu solution is updated so that facility i^* is located on the candidate location $k'_{(1)}$. Similarly, such an approach is followed until N^{th} closest candidate location whenever an improvement is achieved. If there is no improvement for the best upper bound Z_{Tabu}^{best} after the exchange of N^{th} closest candidate location $k'_{(N)}$, then facility i^* is located at the candidate location which gives the lowest objective value among these N neighbor candidate locations.

Selection of the candidate location k^* of facility i^* to move to another neighbor solution requires further attention. The tabu status of candidate locations terminates after a while and it is likely that the same candidate location is selected repetitively. To avoid such a case, a tabu frequency list (FL) keeps record of the total number of times a candidate point k is declared tabu for a facility i within the TS algorithm. $FL(i, k)$ stands for the FL, and initially, $FL(i, k) = 0$ for all $i = 1, \dots, I$; $k = 1, \dots, K$. When a candidate location k for facility i^* is tested within the neighborhood search described at an iteration, $FL(i^*, k)$ is increased by one. The candidate location k which has the highest $FL(i^*, k)$ value is excluded from the neighborhood search in the next iteration. Then the set of neighbor candidate locations does not contain a tabu declared candidate location or the most frequent candidate location for i^* . Therefore, this strategy favors diversification of the solutions during the TS. The upper bound Z_{UB}^t obtained at iteration t is associated with a solution vector (k^1, k^2, \dots, k^I) which represents the candidate locations of each facility in the solution. Here, k^1

Algorithm 1: Tabu Search Algorithm

Step 1. (*Initialization*): Find initial upper bound Z_{UB}^0 and its associated solution vector of candidate locations $(k^1, k^2, \dots, k^I)^0$. Set $Z_{Tabu}^{best} = Z_{UB}^0$ and iteration counter $t = 1$. Set $FL(i, k) = 0$ for $i = 1, \dots, I; k = 1, \dots, K$. Set $T(i, k^i) = \alpha$ and $FL(i, k^i) = 1$ for each facility $i = 1, \dots, I$.

Step 2. For facility $i = 1, \dots, I$,

(i) determine the neighborhood set \mathcal{K}^i as $\mathcal{K}^i = \{k_{(1)}^i, k_{(2)}^i, \dots, k_{(N)}^i\}$ so that $T(i, k_{(n)}^i) = 0$ for $n = 1, \dots, N$ and $FL(i, k_{(n)}^i) < \max\{FL(i, k)\}$ for $n = 1, \dots, N; k = 1, \dots, K$.

(ii) for neighbor $n = 1, \dots, N$, set $(k^1, k^2, \dots, k^i, \dots, k^I)^t = (k^1, k^2, \dots, k^i = k_{(n)}^i, \dots, k^I)^{(t-1)}$ and find its associated objective value as $Z_{UB}^{k_{(n)}^i}$.

(iii) if $Z_{UB}^{k_{(n)}^i} < Z_{Tabu}^{best}$ then update $Z_{Tabu}^{best} = Z_{UB}^{k_{(n)}^i}$, $Z_{UB}^t = Z_{UB}^{k_{(n)}^i}$. Set $FL(i, k_{(n)}^i) = FL(i, k_{(n)}^i) + 1$ and $T(i, k_{(n)}^i) = \alpha$. Go to Step 4.

Step 3. If Z_{Tabu}^{best} does not improve, find k^* such that $k^* = \underset{k \in \mathcal{K}^i}{\operatorname{argmin}}\{Z_{UB}^{k^i}\}$,

$(k^1, k^2, \dots, k^i, \dots, k^I)^t = (k^1, k^2, \dots, k^i = k^*, \dots, k^I)^{(t-1)}$. Set $FL(i, k_{(n)}^i) = FL(i, k_{(n)}^i) + 1$ for $n = 1, \dots, N$ and $T(i, k^*) = \alpha$.

Step 4. Set $t = t + 1$ and decrease each tabu tenure value $T(i, k^i) > 0$ by one. If $t = \theta$ or Z_{Tabu}^{best} does not improve for 30 consecutive iterations STOP and report Z_{Tabu}^{best} , otherwise go to Step 2.

is the candidate location of the first facility, k^2 is the candidate location of the second facility and so forth, in the solution at iteration t . A formal outline of the suggested TS algorithm is given in Algorithm 1.

5. Computational Experiments

In this section, first the test bed used in this work is given. Second, the results obtained with our DB heuristics are presented for the CMWP-B. A Dell Precision T5810 workstation with Intel(R) Xeon(R) E5-1650v3 processor of 3.50 GHz and 64 GB RAM operating within Microsoft Windows 7 Pro 64-bit environment is employed as our computing platform. The callable library of Gurobi 5.6.3 with default settings is used to solve MILP and LP formulations presented and all codes are written in C++ programming language.

5.1. Test Bed

We have performed our computational experiments on randomly generated test instances that are produced using standard CMWP test instances from the literature. Standard test instances are taken from the works by Sherali et al. [19] and Boyacı [48]. Recall that the CMWP instances do not have barriers forbidding location and travel. However, the CMWP-B can inherit facility capacities, customers' demand and location information from the CMWP instances. We have generated convex polyhedral barriers for

each CMWP instance by considering the following two issues. First, we ensure that barrier regions do not contain any customer location inside. Second, interior areas of barriers should not be coinciding with others. Then, these barriers are integrated with the data of the CMWP instances resulting in CMWP-B test instances.

The optimal objective value of the CMWP constitutes a lower bound on the CMWP-B. We have employed the best known objective values of standard CMWP test instances as benchmark values to make a comparison among the performance of the suggested DB heuristics. These benchmark values representing the best known objective values of the CMWP instances are reported in Table 1. The first column indicates the names of the instances. Original instance numbers are used as stated in the work by Sherali et al. [19]. We add a prefix "S" before the corresponding instance number. For example, the CMWP instance number 10 in the work by Sherali et al. [19] is shown as S10. 11 CMWP test instances from Sherali et al. [19] are considered. Analogously, a total of 40 CMWP test instances from Boyacı [48], which have non-unique cost values, are represented with a prefix "B" followed by an instance number. The second column stands for the size of the instance by giving the number of facilities and the number of customers in parenthesis, respectively. The instances which have the same number of facilities and customers are distinguished by adding a suffix letter starting from "a" to "e" in accordance with the

original denomination presented in the reference works. The last column includes the best known objective values that are taken from the reference works ([19, 27, 48]). As a remark, these benchmark values are not necessarily optimal for the CMWP as the latter two studies present heuristic outcomes.

To construct barriers for each standard CMWP test instance given, we have randomly generated different number of extreme points and convex polygons. P polygons are chosen from the set $P \in \{1, 3, 5, 10\}$ for the CMWP-B test instances. When $P = 1$, that is a single polygon, the total number of extreme points, denoted as B , of the polygon is determined within the set $\mathcal{H} = \{2, 3, 4, 5, 6, 7, 8, 9\}$. For $P > 1$, when there are more than one polygon, B is calculated by multiplying the elements of \mathcal{H} with the number of polygons P . For example, for $P = 3$, B is chosen such that $B \in \mathcal{H} = \{6, 9, 12, 15, 18, 21, 24, 27\}$. This makes a total of $4 \times 8 = 32$ combinations of CMWP-B test instances which are generated for each source instance of the CMWP. Further, two different strategies are followed to generate instances that are grouped as regular and random instances. Regular instances contains polygons with the same number of extreme points. For example, an instance is called regular when all barriers are triangles. On the other hand, random instances contains polygons each of which do not necessarily have the same number of extreme points as long as their total does not exceed B . Shortly, each of regular and random instance groups for the CMWP-B contain $4 \times 8 = 32$ test instances. This makes $11 \times 32 \times 2 = 704$ and $40 \times 32 \times 2 = 1920$ CMWP-B test instances that are constructed using existing CMWP test instances Sherali et al. [19] and Boyacı [48] instances, respectively.

5.2. Computational Results

We have employed the best known values of the CMWP as our benchmark values as given in Table 1 to compare the performance of our DB heuristics. The percent deviations of the heuristics are determined using the following formula.

$$100 \times \frac{Z_{UB} - Z_{CMWP}^{best}}{Z_{CMWP}^{best}} \quad (25)$$

Table 1. Benchmark values for the standard CMWP test instances from the literature.

Instance Name	Size (I, J)	Benchmark Value
S6	(3,9a)	221.40
S7	(3,9b)	871.62
S8	(4,8)	609.23
S9	(5,15)	8169.79
S10	(5,20a)	12846.87
S11	(5,20b)	1107.18
S12	(5,30)	23990.04
S15	(5,10)	2595.47
S16	(6,10)	7797.21
S18	(8,10)	1564.46
S20	(10,10)	7719.00
B1	(5,20)	22146.54 ^b
B2	(5,25)	38236.43 ^b
B3	(5,30)	42899.92 ^b
B4	(5,40)	104983.19 ^b
B5	(5,50)	71433.35 ^b
B6	(5,100a)	119564.85 ^b
B7	(5,250)	891589.16 ^b
B8	(6,20)	14693.26 ^b
B9	(6,25)	19036.28 ^b
B10	(6,30)	59665.36 ^b
B11	(7,20)	15075.29 ^b
B12	(7,25)	16086.30 ^b
B13	(7,30)	33627.21 ^b
B14	(8,20)	6499.80 ^b
B15	(8,25)	9486.23 ^b
B16	(8,30)	18081.73 ^b
B17	(9,20)	6551.37 ^b
B18	(9,25)	5983.27 ^b
B19	(9,30)	10643.15 ^b
B20	(10,20)	4702.16 ^b
B21	(10,25)	3950.70 ^b
B22	(10,30)	6863.79 ^b
B23	(10,40)	37834.75 ^b
B24	(10,50)	47125.8 ^b
B25	(10,100)	171126.36 ^b
B26	(20,40)	8240.14 ^a
B27	(20,50)	13535.76 ^b
B28	(20,250)	220033.88 ^b
B29	(25,250a)	208587.12 ^a
B30	(25,250b)	278454.58 ^a
B31	(25,250c)	259745.76 ^a
B32	(25,250d)	195125.60 ^a
B33	(25,250e)	289012.15 ^a
B34	(25,500a)	646653.93 ^a
B35	(25,500b)	512483.15 ^a
B36	(25,500c)	484237.33 ^a
B37	(25,500d)	643832.22 ^a
B38	(25,500e)	491602.33 ^a
B39	(30,100)	26185.03 ^b
B40	(50,250)	81055.87 ^b

^a Results from [27]

^b Results from [48]

where Z_{UB} is the upper bound value found by the suggested heuristic for the CMWP-B. Z_{CMWP}^{best} is the best known objective value for standard CMWP test instances as shown in Table 1. Clearly, the formula (25) yields an upper bound on the performance of the heuristics. The instances are divided into two groups as small and large instances depending on their number of facilities and customers. For example, large instances contain $I > 10$ facilities to locate. In addition, instances with $J \geq 100$ customers are also considered as large instances.

The upper bounds are determined with the ALAB heuristic as described in Section 4.1 for our DB heuristics. The DB-I heuristic employs the ALAB heuristic as an improvement step by using the facility locations obtained after the proposed MILP formulation is solved. For the DB-II heuristic, an initial upper bound is needed to increase the efficiency of the subgradient algorithm. Initially, facilities are assumed to have unlimited capacity and they can meet all customer demand. Then, facilities are assigned to the candidate locations from which they serve the customers with least total cost. Once the corresponding candidate locations are fixed as the starting facility locations, the ALAB heuristic is run initialized from those facility locations. Lastly, the resulting upper bound is used as the initial upper bound of the subgradient algorithm for the DB-II heuristic. Besides, the ALAB heuristic is systematically run (i.e. once in ten iterations) to update best upper bound found within the subgradient algorithm. Finally, the ALAB heuristic is applied on the best solution found by the TS algorithm within the DB-III heuristic.

Table 2 gives the outcomes of the DB heuristics on both random and regular CMWP-B test instances generated from the instances by Sherali et al. [19]. The first three columns explain instance properties. The first column gives the name of the standard CMWP instance as mentioned. (I, J) stands for the number of facilities and customers in the second column. The third column states the total number of barrier regions P in the instances. “% Dev.” and “CPU” denote the percent deviation calculated by the formula (25) and the CPU time of the corresponding DB heuristic in seconds, respectively. To be precise, the cells under these columns are the average of 8 test instances having different number of total extreme points from the set \mathcal{H} depending on the number of barriers P . We should emphasize that all instances in Table 2 are small instances. The last row indicates the average of each column. Starting with column 4, six columns are consecutively dedicated to the

outcomes of the DB heuristics for random and regular test instances. Best percent deviations at each row are marked in bold characters. On these instances, we observed that, DB-I heuristic outperforms both DB-II and DB-III heuristics in terms of accuracy. Indeed, the DB-I heuristic yields outstanding accuracy for both random and regular test instances. Using LR scheme as in the DB-II heuristic increases the efficiency and yields outcomes in almost half time of the DB-I heuristic. Additionally, the DB-II heuristic produces similar accuracy to the DB-I heuristic in 12 out of 44 cases of random instances. This value is 11 out of 44 cases when regular instances are considered.

The tabu tenure parameter α for the DB-III heuristic is set to $\alpha = \max\{I, 20\}$ in the light of our preliminary experiments. A maximum number of tabu iterations θ is set to 300. Another stopping condition of 30 non-improving tabu iterations is also imposed to avoid from unnecessary computations for the DB-III heuristic. A neighborhood search width denoted with N is empirically determined as $N = \min\{\lceil J/3 \rceil, 30\}$. Here, $\lceil a \rceil$ is the smallest integer value that is larger than or equal to a . These TS settings are employed in all of our computational experiments. Unfortunately, the performance of the DB-III heuristic is not promising on these small instances. Nevertheless, running time of the DB-III heuristic is shorter than the DB-I heuristic on the average for small instances.

Although the instances based on Sherali et al. [19] are small instances, their sizes are of limited variety. Therefore, Table 3 shows additional results obtained for random and regular CMWP-B instances (small instances) based on Boyacı [48] instances. Diversity of these instances is higher than the ones by Sherali et al. [19]. Table 3 can be read similar to Table 2 since they share the same structure. The results of Boyacı [48] instances are similar to Sherali et al. [19] instances and strengthen the success of the DB-I heuristic for small instances. The DB-I heuristic is the winner for both random and regular CMWP-B instances over all test instances in terms of accuracy. In particular, the DB-I heuristic yields percent deviations with a difference of 10% more than that of the closest approach, the DB-II heuristic, on the average for small instances. For these small instances, it becomes evident that the efficiency of the DB-I heuristic decreases drastically, i.e. 110.6 and 108.96 seconds on the average for random and regular instances, respectively. The performance

Table 2. The performance of the DB heuristics over random and regular CMWP-B instances based on Sherali et al. [19] instances.

Instance		Random CMWP-B Instances						Regular CMWP-B Instances						
Name	Size	DB-I		DB-II		DB-III		DB-I		DB-II		DB-III		
	(<i>I, J</i>)	<i>P</i>	% Dev.	CPU	% Dev.	CPU	% Dev.	CPU	% Dev.	CPU	% Dev.	CPU	% Dev.	CPU
S6	(3,9)	1	0.27	0.05	0.27	0.24	6.64	0.16	0.51	0.33	0.51	0.24	5.54	0.13
		3	0.48	0.16	0.48	0.29	24.44	0.16	0.36	0.39	0.37	0.24	17.07	0.13
		5	0.27	0.33	0.45	0.29	6.34	0.17	0.72	0.56	0.72	0.23	13.51	0.16
		10	0.27	0.23	0.28	0.35	1.96	0.20	0.20	0.22	0.26	0.30	7.03	0.19
S7	(3,9)	1	0.57	0.02	0.57	0.05	4.35	0.18	0.53	0.02	0.53	0.05	4.14	0.17
		3	1.89	0.03	1.89	0.06	6.95	0.18	1.29	0.02	1.29	0.04	6.03	0.20
		5	0.24	0.02	0.24	0.06	5.06	0.21	1.26	0.02	1.26	0.05	10.27	0.20
		10	0.34	0.03	0.34	0.05	4.42	0.20	1.04	0.04	1.04	0.05	7.47	0.23
S8	(4,8)	1	0.34	0.06	0.34	0.30	32.06	0.19	0.22	0.41	0.22	0.29	32.44	0.18
		3	0.69	0.47	0.69	0.29	33.77	0.17	0.09	0.52	0.09	0.28	35.85	0.18
		5	0.42	0.08	0.42	0.29	30.89	0.20	0.34	0.62	0.34	0.27	32.42	0.19
		10	0.33	0.32	0.33	0.32	30.78	0.25	0.72	0.42	0.72	0.28	29.74	0.23
S9	(5,15)	1	0.31	0.25	1.30	0.10	33.97	0.52	0.25	0.65	1.21	0.10	42.65	0.44
		3	0.46	0.65	1.25	0.10	35.38	0.54	0.66	1.01	2.63	0.15	29.94	0.48
		5	0.20	0.56	1.24	0.11	36.96	0.46	0.19	0.95	1.25	0.11	40.13	0.50
		10	0.47	0.74	1.46	0.10	42.03	0.49	0.54	1.25	1.47	0.10	44.45	0.48
S10	(5,20)	1	0.00	0.37	1.81	0.31	4.16	0.39	0.05	0.99	2.34	0.31	3.22	0.44
		3	0.26	0.75	1.36	0.31	4.60	0.39	0.10	1.29	2.43	0.32	4.32	0.40
		5	0.69	0.91	2.16	0.35	4.72	0.39	0.45	0.68	1.89	0.31	4.47	0.41
		10	1.11	0.69	3.23	0.34	5.30	0.47	0.86	0.63	2.96	0.31	5.54	0.45
S11	(5,20)	1	0.04	0.51	0.04	0.43	51.89	0.68	0.05	0.89	0.05	0.50	60.21	0.79
		3	0.05	0.95	0.83	0.47	56.74	0.70	0.43	0.90	0.43	0.50	63.36	0.80
		5	2.00	1.18	2.00	0.47	52.39	0.75	1.34	0.77	1.34	0.52	67.39	0.71
		10	0.15	0.72	0.15	0.49	53.14	1.08	0.40	0.90	0.40	0.52	75.35	0.77
S12	(5,30)	1	0.09	1.84	5.46	0.48	19.49	0.88	0.20	2.49	4.83	0.60	20.60	0.90
		3	0.16	2.09	5.90	0.50	17.59	0.96	0.07	2.43	5.16	0.56	18.85	0.84
		5	0.02	3.68	5.50	0.54	20.28	0.95	0.08	2.52	5.11	0.55	17.63	0.90
		10	0.11	2.13	6.72	0.55	14.22	1.04	0.28	2.59	4.87	0.58	20.82	1.00
S15	(5,10)	1	0.00	0.10	2.32	0.36	123.50	0.29	0.00	0.25	2.33	0.32	125.95	0.34
		3	0.00	0.09	2.33	0.30	124.69	0.37	0.00	0.29	2.38	0.32	127.35	0.42
		5	0.21	0.45	2.54	0.35	129.09	0.32	0.02	0.28	2.35	0.25	130.11	0.36
		10	0.02	0.27	2.35	0.37	123.87	0.42	0.08	0.24	2.41	0.36	123.51	0.42
S16	(6,10)	1	0.00	0.51	1.65	0.31	54.28	0.22	0.01	0.53	1.66	0.28	56.84	0.23
		3	0.02	0.12	1.67	0.28	50.54	0.24	0.22	0.90	1.87	0.32	58.82	0.28
		5	0.02	0.61	1.68	0.32	59.27	0.24	0.05	0.45	1.71	0.37	73.10	0.28
		10	0.21	0.47	1.70	0.38	45.93	0.26	0.55	0.47	2.20	0.40	54.90	0.28
S18	(8,10)	1	0.24	0.36	82.75	0.27	101.38	0.26	0.22	0.56	86.94	0.30	108.18	0.28
		3	0.14	0.29	67.41	0.31	100.41	0.27	0.22	0.78	80.52	0.29	117.53	0.30
		5	0.37	0.82	69.61	0.30	91.55	0.30	0.53	0.58	75.48	0.34	96.01	0.29
		10	0.11	0.57	48.99	0.32	102.53	0.34	0.08	0.67	46.85	0.30	115.58	0.34
S20	(10,10)	1	0.43	0.40	18.82	0.22	61.25	0.45	0.77	0.57	19.16	0.22	50.91	0.48
		3	0.32	0.22	18.62	0.22	50.76	0.58	0.62	1.09	19.04	0.24	56.04	0.61
		5	0.60	0.54	18.92	0.22	53.93	0.55	1.03	0.76	19.60	0.23	51.51	0.49
		10	0.33	0.47	18.73	0.26	46.57	0.57	0.07	0.67	18.57	0.24	37.40	0.56
Average			0.35	0.59	9.25	0.30	44.55	0.42	0.40	0.76	9.74	0.30	47.82	0.42

of the DB-II heuristic using LR scheme outperforms the DB-I heuristic in CPU times. Moreover, the DB-II heuristics yields the best results in 18 out of 88 cases for the random instances. Analogously, the DB-II heuristic finds the best results in 17 out of 88 cases for the regular instances. The DB-II heuristic is slightly more efficient than the DB-III heuristic with CPU times of 0.75 (0.74) and 0.98 (0.98) seconds on the average, respectively, for random (regular) instances. The accuracy of the DB-III heuristic is poor for small

instances. The efficiency of the DB-I heuristic deteriorates especially for the instances with more than 30 customers (or similarly 30 candidate facility locations). The DB-II and DB-III heuristics show different behavior than the DB-I heuristic does since they are not significantly affected from number of customers. Observe that increasing the number of facilities deteriorates the CPU time of all DB heuristics.

Table 4 presents the results of the large CMWP-B instances based on Boyacı [48] instances. The

structure of Table 4 is similar to Table 2 and Table 3. We have imposed an additional CPU time limit of 3600 seconds for large instances. The DB-I heuristic outperforms the others in 26 (28) out of 72 cases for random (regular) test instances. However, the efficiency of the DB-I heuristic significantly diminishes on large instances. Only instances B6 with 5 facilities and 100 customers can be solved within CPU time limits. Moreover, the DB-I heuristic can not produce a solution for 20 out of 72 cases that correspond to the instances with 25 facilities and 500 customers for both random and regular instances. On the other hand, our findings are outstanding for the DB-II heuristic on large instances. The DB-II heuristic yields the best outcome on 46 (42) out of 72 cases for large random (regular) test instances. Clearly, the computational requirements of the DB-II heuristic is quite reasonable for large instances when compared to the DB-I heuristic. Lastly, we have observed that the DB-III heuristic yields superior results in 2 out of 72 cases on regular instances. The DB-III performs even faster than the DB-II heuristic. Therefore, it can be used as a compromise for larger instances where both DB-I and DB-II heuristics require prohibitive solution times. Finally, we should point on the DB-III heuristic performs almost twice faster than the DB-II heuristic does on large instances in average.

6. Conclusion

In this work, we have focused on the capacitated multi-facility Weber problem with polyhedral barriers (CMWP-B). We have suggested a mathematical formulation for the CMWP-B and its discrete equivalent that is a MILP problem. Discretization based heuristic procedures for the CMWP-B have been proposed. We have carried out extensive computational experiments on randomly generated test instances that are based on standard CMWP instances. The proposed methods compute upper bounds for the optimal objective value of the CMWP-B.

The first discretization based heuristic solves the mixed integer linear programming formulation using the customer locations as the set of candidate facility locations. Its efficiency is improved by using a Lagrangean relaxation scheme and subgradient algorithm that resulted in the second discretization based heuristic. Lastly, the third discretization based heuristic employs a tabu search algorithm using a neighborhood search over customer locations. Among the discretization based

heuristics, the first one yields the highest accuracy. However, its performance is limited with relatively small instances. As a remedy, we have applied a Lagrangean relaxation scheme within the second discretization based heuristic which constitutes a compromise between accuracy and efficiency. The performance of the third discretization based heuristic is usually poor, however, its performance is promising for large instances. Implementing exact solution procedures can be a good direction of research for the CMWP-B in the future. Efforts to provide an effective branch and bound algorithm is of high importance for this type of location-allocation type problems. Last but not least, a probabilistic extension of the CMWP-B, where the customer locations and/or their demand quantities are stochastic, is a worthwhile open research area.

Acknowledgments

The work described in this paper is supported by a grant from Galatasaray University Scientific Research Project Grant No: 16.402.013 and the Turkish Scientific and Technological Research Council (TÜBİTAK) Research Grant No: 214M222.

References

- [1] Cooper, L. (1972). The transportation-location problem. *Oper. Res.*, 20, 94-108.
- [2] Sherali, H.D. and Nordai, F.L. (1988). NP-hard, capacitated, balanced p-median problems on a chain graph with a continuum of link demands. *Math. Oper. Res.*, 13, 32-49.
- [3] Meggido, N. and Supowit, K.J. (1984). On the complexity of some common geometric location problems. *SIAM J. Comput.*, 13, 182-196.
- [4] Alpaydm, E., Altınel, İ.K. and Aras, N. (1996). Parametric distance functions vs. nonparametric neural networks for estimating road travel distances. *Eur. J. Oper. Res.*, 93, 230-243.
- [5] Brimberg, J., Walker, J.H. and Love, R.F. (2007). Estimation of travel distances with the weighted l_p norm: some empirical results. *J. Transp. Geogr.*, 15, 62-72.
- [6] Klamroth, K. (2002). *Single-facility location problems with barriers*. Springer, Berlin.
- [7] Cooper, L. (1964). Heuristic methods for location-allocation problems. *SIAM Rev.*, 6, 37-53.
- [8] Kuenne, R.E. and Soland, R.M. (1972). Exact and approximate solutions to the multisource Weber problem. *Math. Program.*, 3, 193-209.
- [9] Rosing, K.E. (1992). An optimal method for solving the (generalized) multi-Weber problem. *Eur. J. Oper. Res.*, 58, 414-426.
- [10] Krau, S. (1997). *Extensions du problème de Weber*. Thesis (PhD). University of Montréal.
- [11] Righini, G. and Zaniboni, L. (2007). A branch-and-price algorithm for the multi-source Weber problem. *Int. J. Oper. Res.*, 2, 188-207.
- [12] Love, R.F. and Juel, H. (1982). Properties and solution methods for large location-allocation problems. *J. Oper. Res. Soc.*, 33, 443-452.
- [13] Bongartz, I., Calamai, P.H. and Conn, A.R. (1994). A projection method for the l_p norm location-allocation problems. *Math. Program.*, 66, 283-312.

- [14] Hansen, P., Mladenović, N. and Taillard, É. (1998). Heuristic solution of the multisource Weber problem as a p -median problem. *Oper. Res. Lett.*, 22, 55-62.
- [15] Gamal, M.D.H. and Salhi, S. (2003). A cellular heuristic for the multi-source Weber problem. *Comput. Oper. Res.*, 30, 1609-1624.
- [16] Brimberg, J., Hansen, P. and Mladenović, N. (2006). Decomposition strategies for large-scale continuous location-allocation problems. *IMA J. Manag. Math.*, 17, 307-316.
- [17] Brimberg, J., Drezner, Z., Mladenović, N. and Salhi, S. (2014). A new local search for continuous location problems. *Eur. J. Oper. Res.*, 232, 256-265.
- [18] Drezner, Z., Brimberg, J., Mladenović, N. and Salhi, S. (2016). New local searches for solving the multi-source Weber problem. *Ann. Oper. Res.*, 246, 181-203.
- [19] Sherali, H.D., Al-Loughani, I. and Subramanian, S. (2002). Global optimization procedures for the capacitated Euclidean and L_p distance multifacility location-allocation problem. *Oper. Res.*, 50, 433-448.
- [20] Akyüz, M.H., Altınel, İ.K. and Öncan, T. (2014). Location and allocation based branch and bound algorithms for the capacitated multi-facility Weber problem. *Ann. Oper. Res.*, 222, 45-71.
- [21] Zainuddin, Z.M. and Salhi, S. (2007). A perturbation-based heuristic for the capacitated multisource Weber problem. *Eur. J. Oper. Res.*, 179, 1194-1207.
- [22] Luis, M., Salhi, S. and Nagy, G. (2009). Region-rejection based heuristics for the capacitated multi-source Weber problem. *Comput. Oper. Res.*, 36, 2007-2017.
- [23] Aras, N., Altınel, İ.K. and Orbay, M. (2007). New heuristic methods for the capacitated multi-facility Weber problem. *Nav. Res. Log.*, 54, 21-32.
- [24] Boyacı, B., Altınel, İ.K. and Aras, N. (2013). Approximate solution methods for the capacitated multi-facility Weber problem. *IIE Trans.*, 45, 97-120.
- [25] Luis, M., Salhi, S. and Nagy, G. (2011). A guided reactive GRASP for the capacitated multi-source Weber problem. *Comput. Oper. Res.*, 38, 1014-1024.
- [26] Akyüz, M.H., Öncan, T. and Altınel, İ.K. (2012). Efficient approximate solution methods for the multi-commodity capacitated multi-facility Weber problem. *Comput. Oper. Res.*, 39, 225-237.
- [27] Akyüz, M.H., Öncan, T. and Altınel, İ.K. (2013). Beam search heuristics for the single and multi-commodity capacitated Multi-facility Weber Problems. *Comput. Oper. Res.*, 40, 3056-3068.
- [28] Brimberg, J., Hansen, P., Mladenović, N. and Salhi, S. (2008). A survey of solution methods for the continuous location-allocation problem. *Int. J. Oper. Res.*, 5, 1-12.
- [29] Katz, I.N. and Cooper, L. (1981). Facility location in the presence of forbidden regions, I: formulation and the case of Euclidean distance with one forbidden circle. *Eur. J. Oper. Res.*, 6, 166-173.
- [30] Larson, R.C. and Sadiq, G. (1983). Facility Locations with the Manhattan Metric in the Presence of Barriers to Travel. *Oper. Res.*, 31, 652-669.
- [31] Batta, R., Ghose, A. and Palekar, U.S., (1989). Locating Facilities on the Manhattan Metric with Arbitrarily Shaped Barriers and Convex Forbidden Regions. *Transport. Sci.*, 23, 26-36.
- [32] Aneja, Y.P. and Parlar, M. (1994). Technical Note—Algorithms for Weber Facility Location in the Presence of Forbidden Regions and/or Barriers to Travel. *Transport. Sci.*, 28, 70-76.
- [33] Butt, S.E. and Cavalier, T.M. (1996). An efficient algorithm for facility location in the presence of forbidden regions. *Eur. J. Oper. Res.*, 90, 56-70.
- [34] Klamroth, K. (2001). A reduction result for location problems with polyhedral barriers. *Eur. J. Oper. Res.*, 130, 486-497.
- [35] McGarvey, R.G. and Cavalier, T.M. (2003). A global optimal approach to facility location in the presence of forbidden regions. *Comput. Ind. Eng.*, 45, 1-15.
- [36] Bischoff, M. and Klamroth, K. (2007). An efficient solution method for Weber problems with barriers based on genetic algorithm. *Eur. J. Oper. Res.*, 177, 22-41.
- [37] Bischoff, M., Fleischmann, T. and Klamroth, K. (2009). The multi-facility location-allocation problem with polyhedral barriers. *Comput. Oper. Res.*, 36, 1376-1392.
- [38] Wendell, R.E. and Hurter, A.P. (1973). Location theory, dominance and convexity. *Oper. Res.* 21, 314-320.
- [39] Ghosh, S.K. (2007). *Visibility algorithms in the plane*. Cambridge University Press, New York.
- [40] Dijkstra, E.W. (1959). A note on two problems in connexion with graphs. *Numer. Math.*, 1, 269-271.
- [41] Hansen, P., Perreux, J. and Thisse, F. (1980). Location theory, dominance and convexity: some further results. *Oper. Res.*, 28, 1241-1250.
- [42] Weiszfeld, E. (1937). Sur le point le quel la somme des distances de n points donné est minimum. *Tôhoku Math. J.*, 43, 355-386.
- [43] Brimberg, J. and Love, R.F. (1993). Global convergence of a generalized iterative procedure for the minimum location problem with L_p distances. *Oper. Res.*, 41, 1153-1163.
- [44] Brimberg, J., Chen, R. and Chen, D. (1998). Accelerating convergence in the Fermat-Weber location problem. *Oper. Res.*, 22, 151-157.
- [45] Martello, S. and Toth P. (1990). *Knapsack problems: algorithms and computer implementations*. Wiley, New York.
- [46] Held, M., Wolfe, P. and Crowder H.P. (1974). Validation of subgradient optimization. *Math. Program.*, 6, 62-88.
- [47] Glover, F.W. and Laguna, M. (1997). *Tabu search*. Kluwer academic publishers, Boston
- [48] Boyacı, B. (2009). *Solving the capacitated multifacility Weber problem approximately*. Thesis (MSc). Boğaziçi University.

M. Hakan Akyüz received a PhD degree from the Department of Industrial Engineering at Boğaziçi University in 2011. He worked as a post-doctoral fellow at the Department of Industrial Engineering and Logistics Management at the Hong Kong University of Science and Technology afterwards. His research area includes applications of operations research in facility location, container terminal operations and maritime logistics. He is working as an Assistant Professor at the Department of Industrial Engineering at Galatasaray University since 2014.



in IJOCTA, so long as the original authors and source are credited. To see the complete license contents, please visit <http://creativecommons.org/licenses/by/4.0/>.