



A TWO-PHASE METHOD FOR THE CAPACITATED FACILITY PROBLEM OF COMPACT CUSTOMER SUB-SETS

Jaroslav Janáček¹, Lýdia Gábrisová²

¹ Dept of Transportation Networks, University of Žilina, Univerzitná 1, 01026 Žilina, Slovak Republic

² Dept of Mathematical Methods, University of Žilina, Univerzitná 1, 01026 Žilina, Slovak Republic

E-mails: ¹jaroslav.janacek@fri.uniza.sk; ²lydia.gabrisova@fri.uniza.sk

Received 22 November 2008; accepted 3 November 2009

Abstract. The cost optimal design of the majority of distribution and servicing systems consists of decisions on a number and on the locations of facilities from which customers' demands are satisfied; however, there are severe difficulties in solving exact procedures because the underlying mathematical model is NP-hard. These decisions should respect some additional conditions as a limited capacity of located facilities. The objective is to minimize the overall costs of the system and to satisfy all customers' demands. In this paper, we enrich the set of constraints by a new requirement called sub-pool compactness. This property of customer subset influences the quality of vehicle routes subsequently formed in a sub-set of customers served by the same facility. This paper formulates the problem of the enriched capacitated facility location considering compactness condition, formalizes and studies the property of compactness and suggests the compound method solving this problem.

Keywords: facility, customer demands, capacitated location problem, Lagrangean relaxation, customer sub-set compactness.

1. Introduction

The mathematical programming approach to the capacitated facility location problem originates from the assumption that goods distribution is performed from a primary source via warehouses or other places of goods transshipment to the particular customers (Jovanović *et al.* 2009; Liu *et al.* 2009; Baublys 2007, 2008 and 2009; Brauers and Zavadskas 2008; Bagdonienė 2008; Lingaitienė 2008; Afandzadeh and Moayedfar 2008; Pocklad 2007; Ziari *et al.* 2007; Kabashkin 2007; Chakroborty and Wivedi 2002). A number and positions of warehouses (facilities) should be determined to make the total yearly cost of maintaining the located facilities and the yearly cost of customer demand satisfaction are minimal.

The problem is described by a finite set I of possible facility locations and a finite set J of customers whose demands should be satisfied. The associated costs and charges are connected to the particular elements of these sets or pairs of the set elements.

The fixed charge for the location of a facility at possible location $i \in I$ is denoted by f_i and includes all costs connected with keeping this facility at the location for

one year. This charge does not include items dependent on the amount of demands satisfied via this location.

The cost of j -th customer yearly demand satisfaction via the facility located at the place i is denoted by the coefficient c_{ij} that includes all transportation costs of goods transport from the primary source r to the facility location i and from this location to the customer j .

It is presumed that a facility may be placed only at some place from the above-introduced finite set I of possible locations. To model the decision on placing or not placing a facility at the location i , the variable $y_i \in \{0,1\}$ is introduced for each location i from the set I .

Let us assume that each customer $j \in J$ should be supplied with a yearly amount b_j of goods. To be able to express that a customer is assigned to a given facility location and that s/he is supplied via this location, another set of zero-one variables is established. The variable $z_{ij} \in \{0,1\}$ models the decision on assigning or not assigning the customer j to the facility location i . Zero-one variables y_i and z_{ij} will describe each solution of the following integer-programming problem. This zero-one range of decision variables is assumed in each model within this paper and that is why this associated obligatory constraint is mostly not repeated in the following models.

Let us denote by a_i the capacity of a facility located at i ; then, the complete model of the cost minimal capacitated facility location problem can be formulated as follows:

$$\text{Minimize } \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij}, \quad (1)$$

$$\text{Subject to } \sum_{i \in I} z_{ij} = 1 \quad \text{for } j \in J, \quad (2)$$

$$z_{ij} \leq y_i \quad \text{for } i \in I, j \in J, \quad (3)$$

$$\sum_{j \in J} b_j z_{ij} \leq a_i y_i \quad \text{for } i \in I, \quad (4)$$

$$y_i, z_{ij} \in \{0,1\} \quad \text{for } i \in I, j \in J. \quad (5)$$

In this integer programming model, the constraints (2) ensure that each customer demand must be satisfied from exactly one facility location and the constraints (3) force out the placement of a facility at the location i whenever any customer is assigned to this facility location. The constraints (4) ensure that the total supply provided via the facility location i does not exceed the given capacity a_i . Having omitted or relaxed the constraints (4), the problem (1)–(3), (5) is known as the uncapacitated facility location problem and can be effectively solved making the use of implementing the branch and bound method, see works by Erlenkotter (1978) and Körkel (1989). The computational behaviour of the technique was broadly examined and presented by Janáček and Buzna (2007 and 2008). It was shown that this approach was able to manage real-world problems. The capacitated problem (1)–(5) loses the integrality property of variables z_{ij} due to the capacity constraints and thus constitutes a very hard problem of exact solving when a real-sized instance of the problem is considered.

2. Compactness of Sub-Sets

Let customers be located at the nodes of the transportation network. Considering the partitioning of customers into a system of disjoint finite sub-sets, a question emerges: *What property of the finite sub-set may be denoted as compactness?* This term must be obviously taken relatively to the neighbouring sub-sets. One case that can be intuitively taken as incompactness is depicted in Fig. 1, where the black nodes of the graph represent the customers of one sub-set and the white nodes corresponds with the customers of the second sub-set.

Fig. 2 depicts the partitioning of the node set that could be denoted as compact. This intuitively defined property plays an important role in the subsequent service of customers located at the network nodes. The service of a sub-set is usually performed by the routes of vehicles starting and finishing at a depot associated with the serviced sub-set. The shapes of the serviced sub-sets considerably influence the effectiveness of vehicle routing algorithms used for the construction of vehicle routes, as shown by Matis (2008). As we shall deal mostly with the cases where each of the sub-sets is associated with one unique node (centre, median etc.), we can define compactness relative to these unique seed nodes, as in Janáček (2008).

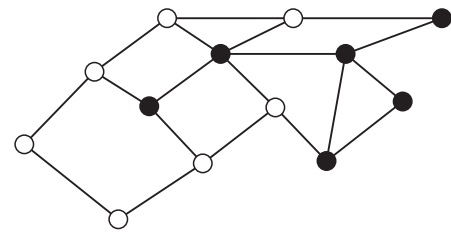


Fig. 1. Incompact partitioning

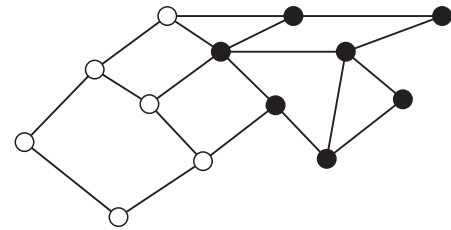


Fig. 2. Compact partitioning

Let us consider that partitioning is given by a system of sub-sets $S_p, i \in I_1$, where the set $I_1 \subseteq I$ includes the locations of facilities in an optimal solution of the capacitated location problem. Let each sub-set S_i is represented by a seed node $n(i)$. Let d_{kj} denote the distance between the nodes k and j of the underlying network.

We say that each sub-set of system $S_p, i \in I_1$ is compact; if the following condition holds for each pair $\langle S_p, S_k \rangle$ of sub-sets and for each pair $\langle j, s \rangle$ of nodes, where $j \in S_i$ and $s \in S_k$ (see Fig. 3):

$$d_{jn(i)} + d_{sn(k)} \leq d_{jn(k)} + d_{sn(i)}. \quad (6)$$

An algorithm that can verify the compactness of all the sub-sets of the given partitioning $S_i, i \in I_1$ of the set J of n customers can be written as follows:

```

for i:=1 to |I1|-1 do
  for k:=i+1 to |I1| do
    for j ∈ Si do
      for s ∈ Sk do
        if djn(i) + dsn(k) > djn(k) + dsn(i) then begin
          message ('Partitioning is incompact');
          break;
        end;
      end;
    end;
  end;
end;
    
```

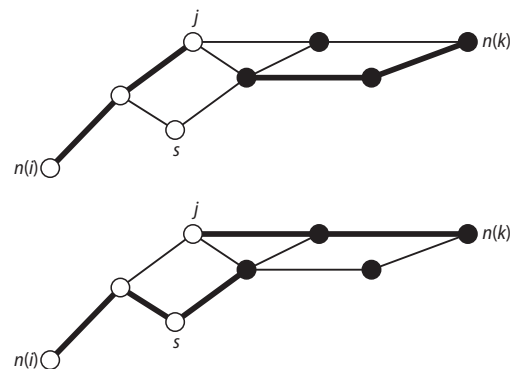


Fig. 3. Illustration of the left-hand side (in the 1-th graph) and right-hand side (in the 2-th graph) of inequality (6)

For n customers and $m = |I_1|$ the sub-sets and a uniform distribution of customers over the sub-sets, this process of verification needs $O(n^4 \cdot m^2)$ operations, i.e. the evaluations of the logical expression.

Now, a question *How to reduce computational effort making the use of the special properties of the sub-sets?* arises.

Theorem 1 (The separation theorem)

Let
$$\max\{d_{jn(i)}: j \in S_i\} + \max\{d_{sn(k)}: s \in S_k\} \leq d_{n(i)n(k)} \quad (7)$$

then inequality $d_{jn(i)} + d_{sn(k)} \leq d_{jn(k)} + d_{sn(i)}$ holds for each $j \in S_i$ and $s \in S_k$.

Proof:

By triangle inequality, we have $d_{n(i)n(k)} \leq d_{jn(i)} + d_{jn(k)}$ and $d_{n(i)n(k)} \leq d_{sn(i)} + d_{sn(k)}$. From the assumption of the theorem, it follows that $d_{jn(i)} + d_{sn(k)} \leq d_{n(i)n(k)}$.

Having multiplied this inequality by 2, we get $2 \cdot d_{jn(i)} + 2 \cdot d_{sn(k)} \leq 2 \cdot d_{n(i)n(k)}$.

The right-hand- side is less than $d_{jn(i)} + d_{jn(k)} + d_{jn(i)} + d_{sn(k)}$.

Thus, $d_{jn(i)} + d_{sn(k)} \leq d_{jn(k)} + d_{sn(i)}$.

Corollary: Compactness need not be verified for pairs $\langle i, k \rangle$, for which (7) holds.

Theorem 2

The sub-sets formed by all customers assigned to the same facility location in an optimal solution of the uncapacitated location problem (1)–(3), (5) are compact under an assumption that the associated cost c_{ij} of customer demand satisfaction is computed in accordance to the formula $c_{ij} = (e_1 d_{ri} + e_0 d_{ij} + g_i) b_j$, where:

- e_1, e_0 denote the unit costs of demand for one unit of transport along with a unit of distance,
- g_i denotes the cost of a transshipment unit,
- d_{ri} denotes the distance between the primary source r and the transshipment location i ,
- d_{ij} denotes the distance between the transshipment location i and the customer j ,
- b_j denotes demand for the customer j .

The coefficients e_1, e_0, b_j are positive real numbers.

Proof:

Let us consider that an optimal solution $\langle y^*, z^* \rangle$ of the uncapacitated location problem (1)–(3), (5) has been found.

The individual sub-sets are formed for each occupied location i , for which $y_i^* = 1$.

The sub-set S_i is formed by those customers $j \in J$, for which $z_{ij}^* = 1$ holds.

It is obvious that $c_{ij} \leq c_{kj}$ follows from $z_{ij}^* = 1$ for each $k \in I$ with $y_k^* = 1$.

Let us consider an arbitrary pair i, k of locations, for which $y_i^* = 1$ and $y_k^* = 1$ hold.

Let us take an arbitrary pair of customers j, s , for which $z_{ij}^* = 1$ and $z_{ks}^* = 1$, thus $c_{ij} \leq c_{kj}$ and $c_{ks} \leq c_{is}$.

After substitution, we obtain:

$$(e_1 d_{ri} + e_0 d_{ij} + g_i) b_j \leq (e_1 d_{rk} + e_0 d_{kj} + g_k) b_j \text{ and } (e_1 d_{rk} + e_0 d_{ks} + g_k) b_s \leq (e_1 d_{ri} + e_0 d_{is} + g_i) b_s.$$

These two inequalities can be rewritten as

$$e_1 d_{ri} + e_0 d_{ij} + g_i \leq e_1 d_{rk} + e_0 d_{kj} + g_k \text{ and } e_1 d_{rk} + e_0 d_{ks} + g_k \leq e_1 d_{ri} + e_0 d_{is} + g_i.$$

The summation of these inequalities gives

$$e_1 d_{ri} + e_0 d_{ij} + e_1 d_{rk} + e_0 d_{ks} + g_k + g_i \leq e_1 d_{rk} + e_0 d_{kj} + e_1 d_{ri} + e_0 d_{is} + g_i + g_k \text{ and consequently } d_{ij} + d_{ks} \leq d_{kj} + d_{is}.$$

This proves the compactness of partitioning induced by the optimal solution $\langle y^*, z^* \rangle$.

3. Lagrangean Relaxation

The Lagrangean relaxation of the problem (1)–(5) considered in this case consists of removing constraints (4) from the model and of embedding a measure of their dissatisfaction into an objective function, see work by Beasley (1993). The measure of capacity constraint dissatisfaction is calculated as the difference between capacity and demand modelled by an expression on the left-hand side of capacity constraint. This overload of facility is weighed by the nonnegative weight called the Lagrangean multiplier representing a penalty paid for breaking constraints by one unit of capacity.

Having denoted u_i , the Lagrangean multiplier of capacity constraint for i , we can transform the objective function (1) to the form:

$$\sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} + \sum_{i \in I} u_i \left(\sum_{j \in J} b_j z_{ij} - a_i y_i \right) = \sum_{i \in I} (f_i - a_i u_i) y_i + \sum_{i \in I} \sum_{j \in J} (c_{ij} + b_j u_i) z_{ij}. \quad (8)$$

If all Lagrangean multipliers in this objective function are fixed as real nonnegative values, then the problem of minimizing (8) subject to (2), (3), (5) takes a form of the uncapacitated facility location problem.

An optimal solution to this relaxed problem provides us with a lower bound of the optimal solution and with an infeasible solution of the original problem. This infeasibility breaks only capacity constraints.

Nevertheless, this relaxation makes the model too loose and results in a bad lower bound. This defect can be partially removed introducing one strengthening constraint derived from the system of capacity constraints. It is obvious that at least such a number of facilities must be located to cover all customer demands. If $a = \max\{a_i: i \in I\}$ and B is the sum of all customer demands, then the minimal number p of the necessary located facilities is $p = B / a$ and the strengthening constraint can be constructed as:

$$\sum_{i \in I} y_i \geq p. \quad (9)$$

To keep the form of the uncapacitated location problem, this constraint can be also relaxed as the other capacity constraints using nonnegative Lagrangean multiplier v (Janáček and Buzna 2008). After this reformulation we get:

$$\text{Minimize} \sum_{i \in I} (f_i - a_i u_i - v) y_i + \sum_{i \in I} \sum_{j \in J} (c_{ij} + b_j u_i) z_{ij} + p v \quad (10)$$

$$\text{Subject to} \sum_{i \in I} z_{ij} = 1 \quad \text{for } j \in J \quad (11)$$

$$z_{ij} \leq y_i \quad \text{for } i \in I, j \in J \quad (12)$$

$$y_i, z_{ij} \in \{0, 1\} \quad \text{for } i \in I, j \in J \quad (13)$$

To obtain a solution meeting the strengthening constraint, the problem (10)–(13) is repeatedly solved for the fixed values of Lagrangean multipliers u_i and for the changing values of the multiplier v . To determine such value of v , for which the constraint (9) is satisfied as equality, the dichotomy scheme is used.

It is true that this algorithm does not ensure constraint satisfaction (9) in general; however, we did not meet any real instance of the solved problem in which the algorithm failed.

We start this process with determining the lower and upper bound of the sought value v^* . It is obvious that if the value of v increases, a number of located facilities drop to one.

The following sub-section shows how a suitable vector \mathbf{u} of multipliers can be found for the given value v^* .

3.1. Sub-gradient Method

The sub-gradient method used in our approach was designed to maximize a lower bound of the objective function value of the optimal solution of the original capacitated facility location problem, see Janáček and Gábrišová (2006). The method improves the lower bound by the steps in which the values of Lagrangean multipliers u_i for each $i \in I$ are adjusted. During this iterative process, the overloads of facilities with dissatisfied capacity constraints are penalized by the higher values of the associated Lagrangean multipliers, and hence the algorithm is forced to minimize the overloads. This process may lead either to complete capacity constraint satisfaction or, what is a more frequent case, to a slightly infeasible solution.

The sub-gradient method maximizes function $F(\mathbf{u})$ over all vector $\mathbf{u} = \langle u_1, u_2, \dots, u_{|I|} \rangle$, the components of which take arbitrary non-negative real values. The value of function $F(\mathbf{u})$ for the fixed values of $u_i, i \in I$ is defined as a minimal value of the expression (8) subject to constraints (9), (11), (12) and (13) and the assumption that variables y_i and z_{ij} take either the value of 1 or 0.

The problem of minimizing (8) subject to (9), (11), (12) and (13) is solved by the customized branch and bound algorithm applied on relaxation formulated as minimize (10) subject to (11), (12) and (13) for the fixed value of Lagrangean multiplier v . The suitable value of multiplier v is determined using the dichotomy algorithm repeating the branch and bound algorithm for various values of v and searches for such a value, for which the associated solution fulfils constraint (9) as equality.

The procedure minimizing $F(\mathbf{u})$ over a non-negative part of the Euclidean space of Lagrangean multipliers $u_i, i \in I$, starts from the initial point \mathbf{u}^0 which is usu-

ally the zero vector. The procedure subsequently forms a sequence of vectors $\mathbf{u}^0, \mathbf{u}^1, \mathbf{u}^2, \dots$, where $F(\mathbf{u}^0) < F(\mathbf{u}^1) < F(\mathbf{u}^2) < \dots$

A move from the current point \mathbf{u}^k to the next point \mathbf{u}^{k+1} is made along the direction, in which the function increases in the neighbourhood of the point \mathbf{u}^k . The steepest increase in $F(\mathbf{u})$ in the neighbourhood of the point \mathbf{u}^k is realized along the gradient of this function at the point \mathbf{u}^k . The components of the gradient can be computed as the values of the partial derivatives of (10) by individual multipliers u_i with a consequent substitution of values u_i^k . Hence, the i -th component takes the value of the associated located facility overload.

To follow the direction of the gradient, the move should be performed in accordance to the equality:

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \alpha \mathbf{grad} / \|\mathbf{grad}\|,$$

where i -th component of \mathbf{grad} is computed as $grad_i = \sum_{j \in J} b_j z_{ij} - a_i y_i$.

Unfortunately, if this formula is applied, the requirement for the non-negativity of multipliers may be broken. That is why the following formula is used in the procedure:

$$\mathbf{u}^{k+1} = \max\{\mathbf{0}, \mathbf{u}^k + \alpha \mathbf{grad} / \|\mathbf{grad}\|\}.$$

Therefore, the move is not performed exactly along the gradient but along the direction of the sub-gradient. The length of the step is given by parameter α , the value of which is chosen from interval $\langle \alpha_{min}, \alpha_{max} \rangle$. After each step, the values of $F(\mathbf{u}^k)$ and $F(\mathbf{u}^{k+1})$ are compared, and if $F(\mathbf{u}^k) \geq F(\mathbf{u}^{k+1})$ holds, then return to \mathbf{u}^k is done and the move is repeated with a lower value of α . The process terminates if parameter α reaches the value of α_{min} or if resulting improvement on the last step is less than the given value ϵ .

3.2. Sub-Set Compactness in the Obtained Solution

Theorem 3

Partitioning created by the sub-sets formed by all customers assigned to one facility in an optimal solution of the Lagrangean relaxation of the capacitated facility location problem is compact for arbitrary non-negative Lagrangean multipliers under the assumption that associated costs c_{ij} are computed in accordance to the formula $c_{ij} = (e_1 d_{ri} + e_0 d_{ij} + g_i) b_j$.

Proof:

Let $\langle \mathbf{y}^*, \mathbf{z}^* \rangle$ denote an optimal solution of the relaxed problem (10)–(13).

Let us consider an arbitrary pair i, k of locations, for which $y_i^* = 1$ and $y_k^* = 1$ hold. Let us have an arbitrary pair of customers j, s , for which $z_{ij}^* = 1$ and $z_{ks}^* = 1$ hold.

From the optimality of the solution, it follows that $c_{ij} + u_i b_j \leq c_{kj} + u_k b_j$ and $c_{ks} + u_k b_s \leq c_{is} + u_i b_s$.

After substitution for c_{ij} , we obtain

$$(e_1 d_{ri} + e_0 d_{ij} + g_i) b_j + u_i b_j \leq (e_1 d_{rk} + e_0 d_{kj} + g_k) b_j + u_k b_j$$

and

$$(e_1 d_{rk} + e_0 d_{ks} + g_k) b_s + u_k b_s \leq (e_1 d_{ri} + e_0 d_{is} + g_i) b_s + u_i b_s.$$

These two inequalities can be rewritten as $e_1d_{ri}+e_0d_{ij}+g_i+u_i \leq e_1d_{rk}+e_0d_{kj}+g_k+u_k$ and $e_1d_{rk}+e_0d_{ks}+g_k+u_k \leq e_1d_{ri}+e_0d_{is}+g_i+u_i$.

The summation of the inequalities gives $e_1d_{ri}+e_0d_{ij}+e_1d_{rk}+e_0d_{ks}+g_k+u_k+g_i+u_i \leq e_1d_{rk}+e_0d_{kj}+e_1d_{ri}+e_0d_{is}+g_i+u_i+g_k+u_k$, that implies $d_{ij}+d_{ks} \leq d_{kj}+d_{is}$.

We have proved that an optimal solution of the relaxed problem (10) – (13) satisfies the condition of compactness but it need not satisfy relaxed capacity constraints. This defect can be partly removed using various techniques based on multiplier adjustment. Nevertheless, these techniques have a limited ability to improve solution infeasibility and fail in obtaining a feasible solution. A classical approach to decreasing this infeasibility is based on the use of some exchange heuristic moving surplus customers from their original subset to another one. This classical approach is little used in our case due to the demanded compactness property of the sub-sets. Therefore, we have studied the property of compactness in a more detailed way. Based on studies, we have suggested a special exchange heuristic considerably improving the infeasibility of the solution and preserving the compactness of the sub-sets.

4. Reformulating the Definition of Partition Compactness

Let us remind the original definition and consider that partitioning is given by a system of sub-sets $S_p, i \in I_1$ where each sub-set S_i is represented by a seed node $n(i)$.

Let d_{kj} denote the distance between nodes k and j of the underlying network. We accept that each sub-set of the system $S_p, i \in I_1$ is compact if the following condition holds for each pair $\langle S_p, S_k \rangle$ of sub-sets and for each pair $\langle j, s \rangle$ of nodes, where $j \in S_i$ and $s \in S_k$: $d_{jn(i)}+d_{sn(k)} \leq d_{jn(k)}+d_{sn(i)}$.

Let us introduce the following denotations:

$v_{ik}(x) = d_{xn(k)} - d_{xn(i)}$ and $v_{ik} = \min \{v_{ik}(x): x \in S_i\}$ for $i, k \in \{1, \dots, m\}, m = |I_1|$.

Theorem 4

The definition of original partitioning compactness can be now expressed as follows: partition S_1, \dots, S_m of node set N with seeds $n(1), \dots, n(m)$ is called to be compact if for each pair $\langle i, k \rangle$ from set $\{1, \dots, m\}, i \neq k$ the following inequality holds: $-v_{ik} \leq v_{ki}$.

Proof:

Inequality $-v_{ik} \leq v_{ki}$ can be expressed as:

$$-\min \{d_{xn(k)} - d_{xn(i)}: x \in S_i\} \leq \min \{d_{yn(i)} - d_{yn(k)}: y \in S_k\}$$

or

$$\max \{d_{xn(i)} - d_{xn(k)}: x \in S_i\} \leq \min \{d_{yn(i)} - d_{yn(k)}: y \in S_k\}.$$

This is equivalent to $d_{xn(i)} - d_{xn(k)} \leq d_{yn(i)} - d_{yn(k)}$ for each $x \in S_i$ and $y \in S_k$.

Theorem 5

Let us assume that a partition S_1, \dots, S_m of node set N with seeds $n(1), \dots, n(m)$ is compact.

Let node x is a node from cluster S_i .

Then the necessary and sufficient condition of preserving compactness when moving x from S_i to S_k is:

- a) $v_{ik}(x) = v_{ik}$
- b) $-\min \{v_{kj}(x), v_{kj}\} \leq v_{jk}$, or equivalently $v_{jk}(x) \leq v_{jk}$ for $j = 1, \dots, m, j \neq i, k$.

Proof:

Let us prove the necessity of a) contradiction. Let us assume that $v_{ik}(x) > v_{ik}$.

Having moved x from S_i to S_k , we obtain

$$\underline{S}_k = S_k \cup \{x\}, \underline{S}_i = S_i - \{x\}, \underline{v}_{ik} = v_{ik} \text{ and } \underline{v}_{ki} = \min \{v_{ki}(x), v_{ki}\} = \min \{-v_{ik}(x), v_{ki}\} = -v_{ik}(x).$$

Now, it is obvious that $\underline{v}_{ki} < -\underline{v}_{ik}$, what leads to contradiction with compactness condition $-\underline{v}_{ik} \leq \underline{v}_{ki}$ for \underline{S}_i and \underline{S}_k .

Let us prove the necessity (b) now.

Let us assume that the condition (b) is not satisfied with some particular j , i.e. $-\min \{v_{kj}(x), v_{kj}\} > v_{jk}$.

By moving the node x from S_i to S_k , we obtain $\underline{S}_j = S_j$ and $\underline{S}_k = S_k \cup \{x\}$ with $\underline{v}_{jk} = v_{jk}$ and $\underline{v}_{kj} = \min \{v_{kj}(x), v_{kj}\}$.

It can be seen that $\underline{v}_{kj} \geq v_{jk}$, which is in contradiction with compactness condition.

If conditions a) and b) are satisfied for given x, i, k and for all $j \neq i, k$, then compactness conditions hold for all pairs of changed clusters and it follows that new partitioning obtained by the move of the node x from S_i to S_k is compact.

5. Exchange Algorithm

The exchange algorithm preserving compactness needs the following input structures: $S_p, i=1, \dots, m$ is a list of customers assigned to particular clusters, $n(i), i=1, \dots, m$ is a list of the cluster seeds.

In accordance to theorems 4 and 5, the following auxiliary structures are created, initialised and updated during the run of the algorithm: $v = \{v_{ik}\}$ is introduced accordingly to the upper definition and, in addition, the matrix $x = \{x_{ik}\}$ is defined so that $x_{ik} \in S_i$ and equality $v_{ik}(x_{ik}) = v_{ik}$ holds.

Furthermore, matrix $a = \{a_{ik}\}$ is introduced to identify pairs $\langle i, k \rangle$ using $a_{ik} = 1$, for which the assumption (b) of Theorem 5 is satisfied. Otherwise, the value of a_{ik} equals to zero.

The key structure of the algorithm is an ordered list L of pairs $\langle S_p, S_q \rangle$ of clusters where a total demand of S_p is greater than its capacity, the associated overload is greater than the possible overload of S_q and the move of customer x_{pq} is feasible from the point of compactness ($a_{pq} = 1$).

The list L is ordered in accordance to the criterion that prefers a maximal decrease in the overloads of

clusters and a minimal increase in the original objective function. The proposed algorithm follows these steps:

1. Initialize \mathbf{v} , \mathbf{x} , \mathbf{a} and L for the current partitioning.
2. If list L is empty, then terminate, otherwise go to step 3.
3. While L is nonempty, perform steps 3.1, 3.2 and 3.3.
 - 3.1. Let $\langle S_p, S_q \rangle$ be the first element of L . Move x_{pq} from S_p to S_q .
 - 3.2. Update matrices \mathbf{v} , \mathbf{x} and \mathbf{a} .
 - 3.3. Update L by discharging elements ceased to satisfy the conditions of L .
4. Go to step 1.

6. Numerical Experiments

The series of 192 problem instances were designed and solved to verify the proposed methods and find out the behaviour of the implemented algorithms. The solved instances were derived from the Slovak road network with 2907 customers representing towns, villages and hamlets. The numbers of the inhabitants (taken in hundreds) of these places were used as volumes of demand mentioned as b_j in models (1)–(5) and (10)–(11). The set of possible facility locations was represented by a set of 71 district centres.

This set of possible facility locations was used in all above-mentioned instances as the set I . The series of 192 instances was divided into four sets accordingly to the generated capacities a_i of the located facilities. Each of these sets contained 48 instances and the same capacity was used for all possible locations in each instance from one set. These sets of instances are denoted as SI1, SI2, SI3 and SI4 in the remainder of this paper and the associated capacities (also taken in hundreds of inhabitants) were 3408, 5380, 5390 and 8710 respectively.

The instances of one set differ in the fixed charges f_i . There is an assumption that facilities at all possible locations from the set I are the same in every instance, however, in each set of instances, four different values of the fixed charge were used to vary the discussed instances.

Further differences among the instances of one set were achieved modifying costs c_{ij} . Cost c_{ij} in the considered problems is computed accordingly to the expression $c_{ij} = (e_1 d_{ri} + e_0 d_{ij} + g_i) b_j$ where the meanings of coefficients e_1 , e_0 , g_i , d_{ri} and d_{ij} are explained in Theorem 2. In all instances, we use coefficients $e_0=1$, $g_i=0$ and coefficient e_1 taking the values of 2, 4 and 6 in each set of the instances.

The next modification of coefficients c_{ij} consisted of choosing a position of the so-called primary source r . Four different primary sources were used in each set of the instances to modify the coefficients of the associated models. A set of cities including Bratislava, Kosice, Banská Bystrica and Zilina was used as the set of possible primary sources. Combining these three parameters, we obtained 48 different instances in each of sets SI1, ..., SI4.

The results reported in the remainder of this paper are the average values of resulting values obtained by solving all instances of a particular set of instances.

The resulting version of the designed methods consists of three phases that can be denoted as *BBDual*, *Subgrad* and *Part*. The first phase *BBDual* solves the relaxed problem where capacity constraints are neglected. Then, the solution satisfies the constraints of compactness and its objective function value represents a lower bound of the optimal solution of the original problem, but nevertheless, may break considerably capacity constraints.

The second phase *Subgrad* starts from the solution obtained by *BBDual* and tries to improve capacity constraint satisfaction by a sequence of multiplier adjustments. The resulting solution also satisfies the constraints of compactness and gives a better lower bound of the objective function of the original problem. Nevertheless, the solution given by *Subgrad* may also break capacity constraints.

The third phase *Part* starts from the solution obtained by *BBDual* or *Subgrad* and uses exchange heuristic to keep compactness constraint and to improve infeasibility in capacity constraints. The objective function value of the resulting solution need not represent a lower bound of the objective function value of the original problem.

As this compound method is focused on minimizing the facility overload of goods, however, in general, we get an infeasible solution considering capacity constraints where the overloads of resulting solutions were studied. Especially, a relative overload of facility located at i was introduced as the value of the following expression:

$$\frac{\sum_{j \in I} b_j z_{ij} - a_i y_i}{a_i} \quad \text{for all } i \in I \tag{14}$$

for the values of z_{ij} and y_i of the resulting solution. Considering each of the solved instances, it is obvious that several located facilities may be overloaded. To evaluate such solution, two parameters were used the first of which was an average value of all relative overloads of the solution. This parameter was denoted as AO. The second parameter was a maximal value of all relative overloads of the solution and was denoted as MO.

As the results obtained for one set of instances must be reported in a comprehensive form, we report an average value of average relative overloads (denoted as AAO) and the maximum of maximal relative overloads (denoted as MMO) in the results of numerical experiments. The average computation time in seconds of one set of instances is denoted as AvgT and also a standard deviation StdT is reported. The average of objective function values is referred as AF (in thousands) in the following tables.

To be able to study an impact of a particular phase on the observed parameters of resulting solutions, we completed four algorithms by combining three above mentioned phases. This way we performed numerical experiments with the following compositions of phases *BBDual*, *BBDual* and *Part*, *BBDual* and *Subgrad*, *BBDual* and *Subgrad* and *Part*. These algorithms are denoted as B, BP, BS and BSP respectively in Tables 1–6.

Table 1. The average results of the algorithm *BBDual* computed for individual sets SI1, ..., SI4

Set	AF	AvgT	StdT	AAO	MMO
SI1	21352	0.65	0.58	1.56	5.14
SI2	21352	0.65	0.58	0.76	2.89
SI3	21352	0.65	0.58	0.76	2.88
SI4	21352	0.65	0.58	0.36	1.40

Table 2. The average results of the compound algorithm *BBDual* and *Part* computed for sets SI1, ..., SI4

Set	AF	AvgT	StdT	AAO	MMO
SI1	22905	0.03	0.01	1.52	3.01
SI2	22761	0.03	0.01	0.61	1.54
SI3	22791	0.03	0.01	0.61	1.54
SI4	22008	0.02	0.01	0.18	0.57

Table 3. The average results of the compound algorithm *BBDual* and *Subgrad* computed for individual sets SI1, ..., SI4

Set	AF	AvgT	StdT	AAO	MMO
SI1	26200	168.04	65.24	0.13	0.88
SI2	23462	177.69	179.1	0.13	0.44
SI3	23472	144.58	117.05	0.13	0.43
SI4	22205	46.38	74.18	0.04	0.35

Table 4. The average results of the compound algorithm *BBDual*, *Subgrad* and *Part* computed for individual sets SI1, ..., SI4

Set	AF	AvgT	StdT	AAO	MMO
SI1	26566	168.07	65.26	0.03	0.15
SI2	23699	177.7	179.11	0.02	0.07
SI3	23732	144.6	117.06	0.02	0.12
SI4	22255	46.38	74.19	0.00	0.03

Table 5. The average relative overloads (AAO) computed by the particular algorithms for sets SI1, ..., SI4

Set	B	BP	BS	BSP
SI1	1.56	1.52	0.13	0.03
SI2	0.76	0.61	0.13	0.02
SI3	0.76	0.61	0.13	0.02
SI4	0.36	0.18	0.04	0.00

Table 6. Maximal relative overloads (MMO) computed by the particular algorithms for sets SI1, ..., SI4

Set	B	BP	BS	BSP
SI1	5.14	3.01	0.88	0.15
SI2	2.89	1.54	0.44	0.07
SI3	2.88	1.54	0.43	0.12
SI4	1.40	0.57	0.35	0.03

To evaluate the objective function values of the resulting solutions obtained by the suggested compound algorithm, we introduce the following denotations. Let $AF(B)$ and $AF(BSP)$ denote the objective function values of solutions obtained by the algorithms *BBDual* and *BBDual*, *Subgrad*, *Part* respectively. We use the symbol SLB to denote a lower bound obtained from the optimal solution of the Lagrangean relaxation of the original problem where capacity constraints are relaxed and the objective function is charged by the sum of the products of the Lagrangean multiplier and the difference between a total demand satisfied by a facility and the capacity of the facility. Let Gap denote the difference between the objective function value of the resulting solution and the highest associated lower bound. The symbol $Gap\ [%]$ denotes the value of Gap given in a percentage of the highest associated lower bound (see Table 7).

Table 7. The average lower bounds ($AF(B)$, SLB), objective values ($AF(BSP)$) and gaps computed for individual sets SI1, ..., SI4

Set	AF(B)	SLB	AF(BSP)	Gap	Gap[%]
SI1	21352	25992	26200	208	0.8
SI2	21352	23207	23462	255	1.1
SI3	21352	23222	23472	250	1.08
SI4	21352	21969	22205	237	1.08

7. Conclusions

This paper outlines a framework for the solution of the capacitated facility location problem with demand for a cluster shape of resulting clusters.

Demand for a cluster shape denoted as compactness was exactly defined, its properties were explored and some associated theorems were formulated and proved. The main result of the theoretical study reveals that any optimal solution of the uncapacitated facility location problem induces compact clusters under the assumption that the costs of customer's demand satisfaction are computed from the network distances and prime costs.

It has been proved that this result holds even for solutions to such uncapacitated facility location problem that is obtained by the Lagrangean relaxation of the capacitated facility location problem.

Based on these findings, a compound algorithm was suggested, in which an initial solution is obtained as an optimal solution of the uncapacitated facility location problem and the infeasibility of this solution concerning capacity constraints is subsequently improved by the sub-gradient algorithm.

The resulting solution, that usually remains slightly infeasible, is further improved by special exchange heuristic designed so that it does not break the compactness of the processed solutions.

The suggested compound algorithm was thoroughly tested using the pool of 192 instances of the capacitated facility location problem and the characteristics of the resulting solutions were studied. Great attention was

paid to the average relative overload and maximal relative overload and to the contribution of the particular phases of the algorithm in order to improve these characteristics.

Based on the results reported in comprehensive Tables 5 and 6, we can state that the suggested algorithm is able to considerably decrease the relative overload. In the worst case, which is connected with the set SII of problem instances, the maximal relative overload was 15% of the original capacity of the most overloaded facility.

Nevertheless, the average relative overload was less than 3% and the procedures used in the algorithm guarantee that each resulting solution keeps the property of compactness. Tables 5 and 6 show a synergy effect of sub-gradient and exchange procedures. When only an exchange procedure is used to improve the rough infeasible solution obtained by the procedure *BBDual*, the results are much worse than the findings obtained by the sub-gradient method and the overload of an input solution is improved at most by 20% or 50% concerning the average and maximum overload respectively (see column B and BP in tables 5 and 6). However, when the exchanged heuristic is combined with the sub-gradient method, it is improved by 80% of the overload obtained by the sub-gradient method.

The objective function of an optimal solution of any instances of the original problem is bounded from downside by a rough lower bound represented by the objective function value of the optimal solution of the relaxed problem (see column AF in Table 1).

If we compare these values with the values of the objective function reported in Table 4, it could be seen that the resulting solution of instances from SI2, SI3 and SI4 can be worse at most by 10% than the unknown optimal solution. For set SII, it is 20% of a lower bound.

Nevertheless, the sub-gradient method can provide us with a more precise lower bound if values of Lagrangean multipliers and the associated overloads are taken into account. These lower bounds are reported in column SLB in Table 7 and show that optimal solutions are much nearer to the obtained solutions than the suggested tough lower bounds. Taking into consideration slight infeasibility, we can declare that the difference between the optimal and obtained solutions is equal approximately to one percent.

The carried out experiments show that the suggested compound algorithm can effectively reduce the maximal overload and constitutes an effective tool for obtaining a near optimal solution of the original problem. This approach to the capacitated facility location problem seems to be worth enough for further development in the future research. Especially, other approaches to Lagrangean multiplier adjustment seem to constitute a promising way to better solutions. Besides designing a private distribution system, the suggested algorithm can be used to develop a public service system with a weakly limited capacity of servicing facilities (Marianov and Serra 2002; Jánošíková 2007).

Acknowledgment

This work was supported by the grant VEGA 1/0591/09 'Optimal Designing of Public Service Systems'.

References

- Afandizadeh, Sh.; Moayedfar, R. 2008. The feasibility study on creation of freight village in Hormozgan province, *Transport* 23(2): 167–171.
doi:10.3846/1648-4142.2008.23.167-171
- Bagdonienė, D. 2008. Optimization of loading facilities at the terminal, *Transport* 23(2): 95–97.
doi:10.3846/1648-4142.2008.23.95-97
- Baublys, A. 2009. Principles for modelling technological processes in transport terminal, *Transport* 24(1): 5–13.
doi:10.3846/1648-4142.2009.24.5-13
- Baublys, A. 2008. Model for distribution of warehouses in the commercial network in optimising transportation of goods, *Transport* 23(1): 5–9.
doi:10.3846/1648-4142.2008.23.5-9
- Baublys, A. 2007. Probability models for assessing transport terminal operation, *Transport* 22(1): 3–8.
- Beasley, J. E. 1993. Lagrangian relaxation, in *Modern Heuristic Techniques for Combinatorial Problems*, 243–303.
- Brauers, W. K. M.; Zavadskas, E. K. 2008. Multiobjective optimization in local theory with a simulation for a department store, *Transformations in Business & Economics* 7(3): 163–183.
- Chakraborty, P.; Wivedi, T. 2002. Optimal route network design for transit systems using genetic algorithms, *Engineering Optimization* 34(1): 83–100. doi:10.1080/03052150210909
- Erlenkotter, D. 1978. A Dual-based procedure for uncapacitated facility location, *Operations Research* 26(6): 992–1009.
doi:10.1287/opre.26.6.992
- Janáček, J. 2008. Customers' Pool partitioning to compact sub-areas, in *Advances in Transport Systems Telematics*, 117–124.
- Janáček, J.; Buzna, L. 2007. *Facility Location in Distribution Systems*: monograph. Žilina, EDIS-University of Žilina. 142 p.
- Janáček, J.; Buzna, L. 2008. An acceleration of Erlenkotter-Körkel's algorithms for uncapacitated facility location problem, *Annals of Operations Research* 164(1): 97–109.
doi:10.1007/s10479-008-0343-0
- Janáček, J.; Gábrišová, L. 2006. Lagrangean relaxation based approximate approach to the capacitated location problem, *Komunikácie/Communications* 8(3): 19–24.
- Jánošíková, L. 2007. Emergency medical service planning, *Komunikácie/Communications* 9(2): 64–68.
- Jovanović, V. D.; Tica, S.; Milovanović, B.; Živanović, P. 2009. Researching and analyzing the features of oil and demand for transporting oil derivatives in the area of Belgrade, *Transport* 24(3): 249–256.
doi:10.3846/1648-4142.2009.24.249-256
- Kabashkin, I. 2007. Logistics centres development in Latvia, *Transport* 22(4): 241–246.
- Körkel, M. 1989. On the exact solution of large-scale simple plant location problem, *European Journal of Operational Research* 39(2): 157–173.
doi:10.1016/0377-2217(89)90189-6
- Lingaitienė, O. 2008. A mathematical model of selecting transport facilities for multimodal freight transportation, *Transport* 23(1): 10–15.
doi:10.3846/1648-4142.2008.23.10-15

- Liu, W.; Xu, H.; Zhao, X. 2009. Agile service oriented shipping companies in the container terminal, *Transport* 24(2): 143–153. doi:10.3846/1648-4142.2009.24.143-153
- Marianov, V.; Serra, D. 2002. Location problems in the public sector, in *Facility Location: Applications and Theory*. Berlin, Springer, 119–150.
- Matis, P. 2008. Decision support system for solving the street routing problem, *Transport* 23(3): 230–235. doi:10.3846/1648-4142.2008.23.230-235
- Pocklad, O. 2007. Development of the interurban service by use of alternative commercial road trains, *Transport* 22(2): 90–98.
- Ziari, H.; Keymanesh, M. R.; Khabiri, M. M. 2007. Locating stations of public transportation vehicles for improving transit accessibility, *Transport* 22(2): 99–104.