

Comparison of Exact Algorithms for Rectilinear Distance Single-Source Capacitated Multi-Facility Weber Problems

Chansiri Singhtaun and Peerayuth Charnsethikul
Operations Research and Management Science Units, Department of Industrial Engineering,
Kasetsart University, Bangkok 10903, Thailand

Abstract: Problem statement: The objective of this study is to develop efficient exact algorithms for a single source capacitated multi-facility location problem with rectilinear distance. This problem is concerned with locating m capacitated facilities in the two dimensional plane to satisfy the demand of n customers with minimum total transportation cost which is proportional to the rectilinear distance between the facilities and their customers. **Approach:** Two exact algorithms are proposed and compared. The first algorithm, decomposition algorithm, uses explicit branching on the allocation variables and then solve for location variable corresponding to each branch as the original Mixed Integer Programming (MIP) formulation with nonlinear objective function of the problem. For the other algorithm, the new formulation of the problem is first created by making use of a well-known condition for the optimal facility locations. The problem is considered as a p -median problem and the original formulation is transformed to a binary integer programming problem. The classical exact algorithm based on this formulation which is branch-and-bound algorithm (implicit branching) is then used. **Results:** Computational results show that decomposition algorithm can provide the optimum solution for larger size of the studied problem with much less processing time than the implicit branching on the discrete reformulated problem. **Conclusion:** The decomposition algorithm has a higher efficiency to deal with the studied NP-hard problems but is required to have efficient MIP software to support.

Key words: Location-allocation problem, decomposition algorithms, mixed integer programming

INTRODUCTION

The Location-Allocation Problem (LAP) is to select the locations of a set of facilities and simultaneously allocate to these facilities demand for service of a set of customers in order to minimize the transportation cost. Facilities may be considered as plants, warehouses, supply centers and hospitals while dealers, working stations and retailers may be considered as customers. LAP has been shown to be NP-hard by (Francis *et al.*, 1974).

When facility locations are selected from a set of pre-designed locations, the corresponding LAP is called a discrete LAP or a Capacitated Multi-facility Location Problem (CMLP). In this problem, distance between each customer and facility is the real or estimated distances between two corresponding nodes, which is a known constant value. If the demand of a customer can be serviced by many facilities separately, the mathematical model formulation of the problems is a mixed integer programming problem. Most of the

algorithms for such problems deal with finding the efficient technique to tighten bound or to relax the constraints in branch-and-bound algorithm such as the algorithm proposed by (Akino and Khumawala, 1977; Nauss, 1978; Sa, 1969). When a customer should receive the whole service quantities from a facility, the problem is known as p -median problem. The mathematical model is a binary integer problem with two set of 0-1 variables to allocate the node for the facilities and customers. Since the efficient exact algorithm for such problem is branch-and-bound algorithm, most of the studies emphasize on improving the quality of initial solution such as the algorithm proposed in (Koskosidis and Powell, 1992; Mulvey and Beck, 1984).

When the facilities can be located anywhere in a continuous plane, the corresponding LAP is known as the Capacitated Multi-facility Weber Problem (CMWP). The transportation distances between customers and facilities are considered in different distance functions. These functions can be a rectilinear

Corresponding Author: Chansiri Singhtaun, Operations Research and Management Science Units,
Department of Industrial Engineering, Kasetsart University, Bangkok 10903, Thailand

distance, squared-Euclidean distance, Euclidean distance or the general distance function called l_p . If the demand of customer can be satisfied from different facilities, the CMWP is a Multi-source problem (MCMWP). On contrary, if every customer should be served by a single facility, the problem is formulated as a Single-source CMWP (SCMWP), which requires the use of additional binary variables. To solve CMWP, the original formulation of the problem is usually reformulated using the special condition for optimal solution of considered distance function. The reformulated equation will then be solved by the specific methods for such kind of formulation such as sub-gradient method, Extreme point raking method, cutting plane method etc. that are shown in (Sherali and Tuncbilek, 1992; Singhtaun and Charnsethikul, 2007; Singhtaun and Charnsethikul, 2008).

In this study, two exact algorithms under two different approaches are proposed to solve two different formulations of Rectilinear distance SCMWP (RSCMWP). The next two sections show methodology of the two algorithms. The result section compares the computational time of the two algorithms on the test problems with various sizes. The conclusion is then summarized in the last section.

MATERIALS AND METHODS

There are two algorithms proposed in this study which are a decomposition algorithm and enumeration algorithm. Their methodology can be shown as follows:

A decomposition algorithm:

Mathematical model: The CMLCP is described as follows. There are $m > 1$ new facilities with a certain capacity to be located on the continuous plane. They have to serve n customers in their responsibilities whose locations and inseparable products or demands are known and deterministic. The objective is to find the good locations of these new facilities and allocations of customers to them so as to minimize total distances measured in squared-Euclidean metric with respect to facility capacity. It can be mathematically formulated as follows:

$$\begin{aligned} \text{Minimize } & \sum_{i=1}^m \sum_{j=1}^n z_{ij} w_j |(x_i - a_j) + (y_i - b_j)| \\ \text{subject to } & \sum_{i=1}^m z_{ij} = 1 \\ & \sum_{j=1}^n z_{ij} w_j \leq s_i \end{aligned} \quad (1)$$

Where:

$$\begin{aligned} s_i &= \text{Capacity of facility } i; i = 1, \dots, m \\ w_j &= \text{Demand of customer } j; j = 1, \dots, n \\ z_{ij} &= \begin{cases} 1, & \text{if customer } j \text{ is assigned to facility } i \\ 0, & \text{otherwise} \end{cases} \\ (a_j, b_j) &= \text{Known co-ordinate of customer } j \text{ on plane} \\ (x_i, y_i) &= \text{Unkown co-ordinate of facility } i \text{ on plane} \end{aligned}$$

The objective function above gives the total transportation distance, while the first constraint set ensures that all customer demands are satisfied and the second constraint set ensures that each facility capacity is controlled respectively.

Algorithm development: Even though the LAP is NP-hard in nature, it composes of two sub-problems, which can be solved in polynomial time. They are location problem (finding the location variables (x_i, y_i)) and allocation problem (finding the allocation variables z_{ij}). Observe from Eq. 1 that if the allocation variables z_{ij} are known, the optimum location is readily obtained by solving m single facility location problems with rectilinear distance independently. On contrary, if the location variables (x_i, y_i) are known, the best allocation can be found by solving binary integer programming problem with $m \times n$ variables. The decomposition algorithm uses this fact to solve Eq. 1 by decomposing the RSCMWP into two sub-problems. For location sub-problem, all feasible facility locations are listed as candidate nodes using the special property for LAP with rectilinear distance. All nodes in the list are then selected with m nodes to be m facility locations. At each set of m facility locations the allocation variables corresponding to the known m location variables are solved. The optimum solutions are the set of location and allocation variables that give minimum objective function value. It can be elaborately explained as follows.

Step 1: Find all feasible location solutions:

Francis *et al.* (1974) shows that the rectilinear distance multi-facility location problem always has a minimum-cost solution where the x coordinate of each facility is equal to the x coordinate of some customers and the y coordinate of each facility is equal to the y coordinate of some customers. The feasible locations are listed from all possible pairs of coordinate (a_j, b_j) of n customers. Therefore, there are n^2 possible coordinates or nodes to be the candidate facility locations and a set of m nodes out of these n^2 nodes is an optimum set of locations for m facilities. All possible set of selecting m nodes from n^2 nodes are considered.

Step 2: Solve the allocation solutions: At each set of m nodes selecting from n^2 nodes, the allocation sub-problem corresponding to the set of locations is solved. Plugging in the set of m coordinates to location variables (x_i, y_i) , an Eq. 1 becomes as follows:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^m \sum_{j=1}^n w_j d_{ij} z_{ij} \\ \text{subject to} \quad & \sum_{i=1}^m z_{ij} = 1 \quad ; j = 1, \dots, n \\ & \sum_{j=1}^n z_{ij} w_j \leq s_i \quad ; i = 1, \dots, m \\ & z_{ij} \in \{0, 1\} \end{aligned} \quad (2)$$

Where:

$$d_{ij} = \left[|x_i - a_j| + |y_i - b_j| \right]$$

x_i = x co-ordinate of a candidate location

y_i = y co-ordinate of a candidate location

This problem is then solved by branch-and-bound algorithm combining with logic based method (Hooker, 2000). The logic based method is using the logic of integer according to the problem conditions to construct the constraints so as to tighten bounds of the variables. Since the problem contains binary integer variables, tightening bounds of the variables is equivalent to determining the value of these variables and then can reduce the problem size to be considered. For this problem, the logic based constraints are constructed from the fact that every facility cannot serve its customers over their capacity. This condition can be transformed to the logic condition as follows. If $w_j > s_i$, then $z_{ij} = 0$. Obviously, the more logic based constraints are created, the more number of variables are cut. Moreover, if $m \times n^{-1}$ logic based constraints are constructed, all $m \times n$ allocation variables z_{ij} will be fixed.

Step 3: Update and select the optimum solutions: To update the solution at each loop of solving Eq. 2, the solution with less objective function is kept while the worse is discarded. After passing through all possible set of m nodes, the latest updated solution is the optimum solution.

Algorithm complexity: The complexity of decomposition algorithm is shown as follows. The algorithm requires complexity $O(n^{2m})$ for selecting m locations from n^2 the candidate locations. At each set of m selected locations needs 2^{mn} nodes to be searched in the worst case. Hence, the complexity of the decomposition algorithm is $O(n^{2m} 2^{mn})$.

An enumeration algorithm: The idea of this algorithm is to transform Eq. 1 to the standard formulation, which is an integer programming and then solve it with the specific method of such problem. Under this approach, the location and allocation variables are solved simultaneously. It can be elaborated as follows.

Discrete mathematical formulation of RSCMWP and the algorithm: The RSCMWP is reformulated as a discrete LAP using the optimal solution property for location variables described in step 1 of a decomposition method. Since the optimal locations are a set of m nodes selecting from n^2 feasible nodes created from a set of customer coordinates, the distance between the candidate nodes and customers are known. The RCMWP can be reformulated as follows:

The RCMWP can be reformulated as follows:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^m \sum_{j=1}^n z_{kj} w_j d_{kj} \\ \text{subject to} \quad & \sum_{k=1}^v y_{ki} = 1 \\ & \sum_{k=1}^v z_{kj} = 1 \\ & \sum_{j=1}^n z_{kj} w_j \leq \sum_{i=1}^m s_i y_{ki} \end{aligned} \quad (3)$$

Where:

s_i = Capacity of facility i; $i = 1, \dots, m$

w_j = Demand of customer j; $j = 1, \dots, n$

d_{kj} = Rectilinear distance from node k to customer

$$j = \left[|a_k - a_j| + |b_k - b_j| \right]; \quad k = 1, \dots, v$$

$$y_{kj} = \begin{cases} 1, & \text{if facility } i \text{ is located at node } k \\ 0, & \text{otherwise} \end{cases}$$

$$z_{kj} = \begin{cases} 1, & \text{if customer } j \text{ is assigned to facility at node } k \\ 0, & \text{otherwise} \end{cases}$$

(a_j, b_j) = Known co-ordinate of customer j on plane

(x_i, y_i) = Known co-ordinate of candidate location

The objective function above gives the total transportation distance from selected candidate nodes (facility locations) and customers, while the first and second constraint set ensures that a facility and a customer select only one candidate nodes to be a location and a service node respectively. The last constraint shows that each facility capacity is controlled.

The classical exact algorithm for Eq. 3 which is an integer programming problem is a branch-and-bound algorithm.

Algorithm complexity: Since number of variables in Eq. 3 equals $n^2(m+n)$ for z_{kj} and y_{ki} , then the complexity of enumeration algorithm in the worst case is $O(2^{n^3})$.

To evaluate the efficiency of both algorithms proposed above, they are tested on benchmark instances. The numerical experiments are constructed with various problem sizes of which the number of facility (m) varies from 2-6 and $m \times n$ is less than 60. For each problem size, 5-10 sets of data are generated and then solved by two algorithms: decomposition algorithm and enumeration algorithm. Both algorithms are coded with MATLAB and use command “bintprog” to do branch-and-bound algorithm. These experiments process on a personal computer with 2.4 GHz, 2.00 GB of RAM, Pentium IV Core 2 Duo.

RESULTS

Since the solutions of both algorithms are optimal solutions, the comparison of the quality of the solutions is ignored. Therefore, only average processing time of all instances or algorithm efficiency is considered. They are shown in Table 1.

The underlined values in Table 1 mean that there are some cases in these problem sizes that are premature terminated by time limitation. The limitation of processing time for decomposition algorithm is set at 36,000 sec or 10 h while for enumeration algorithm is set at 7,200 sec. At the premature time, decomposition algorithm can provide good or optimal solution while the other cannot give any solution because of too small premature time to solve the problem. The letter “N/A” means the algorithm cannot provide the solution because of inadequate computer memory.

Table 1: Processing time of decomposition and enumeration algorithms

Problem No.	Problem size		Decomposition algorithm		Enumeration algorithm	
	m	n	Complexity	Processing time (sec)	Complexity	Processing time (sec)
1	2	5	6.40E+05	2.5922	4.25E+37	18.1781
2	2	10	1.05E+10	125.1969	1.07E+301	7.20E+03
3	2	15	5.44E+13	1.32E+03	>1.07E+301	N/A
4	2	30	9.34E+23	3.42E+04	>1.07E+301	N/A
5	3	5	5.12E+08	19.2547	4.25E+37	35.8
6	3	10	1.07E+15	2.43E+04	1.07E+301	7.20E+03
7	3	15	4.01E+20	3.60E+04	>1.07E+301	N/A
8	3	20	7.38E+25	3.60E+04	>1.07E+301	N/A
9	4	8	7.21E+16	1.13E+04	1.34E+154	N/A
10	4	10	1.10E+20	3.60E+04	1.07E+301	N/A
11	4	15	2.95E+27	3.60E+04	>1.07E+301	N/A
12	5	8	1.18E+21	3.60E+04	1.34E+154	N/A
13	5	10	1.13E+25	3.60E+04	1.07E+301	N/A
14	6	8	1.93E+25	3.60E+04	1.34E+154	N/A
15	6	10	1.15E+30	3.60E+04	1.07E+301	N/A

DISCUSSION

From Table 1, the decomposition algorithm has much less complexity than enumeration algorithm for all instances. Therefore, it can provide the optimal solution or good solution (in case of no premature termination occur) for much more number of test instances than enumeration algorithm. The enumeration algorithm struck at only the small-scale problem with $(m,n) = (2,5)$ and $(3,5)$ whose complexity are not over $4.25 \text{ E}+37$ while the decomposition algorithm can give optimal solution to the problem size from $(m,n) = (2,5)$ - $(5,8)$. Its efficiency appears obviously when the problem sizes grow and the number of customer (n) is much more than the number of facility (m).

Increasing in both m and n requires more computational time. However, the number of facility has a higher impact on computational time than the number of customers. Increasing of m makes the computational time increase numerously even if number of customers is small. For example, both algorithms take less time to solve problem number 3 with $(m,n) = (2,15)$ than problem number 6 with $(m,n) = (3,10)$. However, there are some cases given the contradicted results. To explain this special condition easier, a graph that illustrates the relationship between the number of variables and computational time corresponding to each m is illustrated as shown in Fig. 1.

From Fig. 1, the beginning of graph m^{-1} lies below the beginning of graph m . After passing a certain point the graph of m is steeper rapidly and lies above graph of m^{-1} . The reason behind this condition is that when n is closed to m , the average number of customers in each facility ($n m^{-1}$) will be small and the probability to occur logical based constraints is high. Therefore, the number of variables to be considered is reduced. Moreover, the smaller $n m^{-1}$ is, the higher number of variables should be reduced.

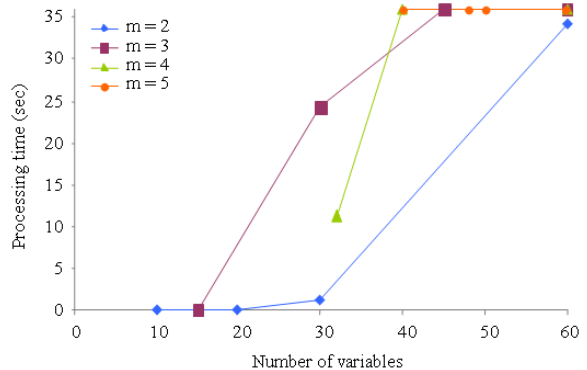


Fig. 1: The relationship between number of variables and computational time separated by m

CONCLUSION

In this study, the RSCMWP is studied, analyzed and solved. Two exact algorithms to solve the different formulations of the problem is developed and compared. The first algorithm, decomposition algorithm, formulates the RSCMWP as an original mathematical model which is a mixed integer programming formulation with nonlinear objective function. Using the special property of the optimal solution to rectilinear distance single facility location, all possible candidate locations are determined. At each location, the optimal allocation is provided using branch and bound algorithm. The other algorithm, enumeration algorithm, reformulates the problem as a discrete LAP which requires much more additional variables and solves this reformulated equation using branch-and-bound algorithm. The location and allocation variables are found simultaneously. The result shows that the decomposition algorithm has higher efficiency to solve RSCMWP than the enumeration algorithm because of less complexity than the other. Using the logic based technique; allocation sub-problem can be solved faster because of reduction on the number of variables to be considered. Owing to the logic based constraints the two-phase algorithm can solve the problem with m is up to 4 without premature terminated.

As a future direction, the method to expedite the computational time of solving the two independent sub-problems for decomposition algorithm or of solving the independent branches in branch-and-bound algorithm such as grid computing technique may be considered.

ACKNOWLEDGEMENT

This study was supported by Kasetsart University Research and Development of Thailand under New Researcher Grant No. 36.51.

REFERENCES

Akino, U. and B.M. Khumawala, 1977. An efficient branch and bound algorithm for the capacitated warehouse location problem. *Manage. Sci.*, 23: 585-594. DOI: 10.1287/mnsc.23.6.585

Francis, R.L., L.F. McGinnis Jr. and J.A. White, 1974. *Facility Layout and Location: An Analytical Approach*. 2nd Edn., Prentice Hall, Inc., ISBN: 0132992310, pp: 210-245.

Hooker, J.N., 2000. *Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction*. 2nd Edn., John Wiley and Sons, ISBN: 0471385212.

Koskosidis, I. and W.B. Powell, 1992. Clustering algorithms for consolidation of customer orders into vehicle shipments. *Transport. Res.*, 26: 365-379. DOI: 10.1016/0191-2615(92)90032-R

Mulvey, J. and M. Beck, 1984. Solving capacitated clustering problems. *Eur. J. Operat. Res.*, 18: 339-348. DOI: 10.1016/0377-2217(84)90155-3

Nauss, M.R., 1978. An improved algorithm for the capacitated facility location problem. *J. Operat. Res. Soc.*, 29: 1195-1201. DOI: 10.1057/jors.1978.263

Sa, G., 1969. Branch and bound and approximate solutions to the capacitated plant-location problem. *Operat. Res.*, 17: 1005-1016. DOI: 10.1287/opre.17.6.1005

Sherali, H.D. and C.H. Tuncbilek, 1992. A squared-Euclidean distance location-allocation problem. *Naval Res. Logist.*, 39: 447-469. DOI: 10.1002/1520-6750(199206)39:4<447::AID-NAV3220390403>3.0.CO;2-O

Singhtaun, C. and P. Charnsathikul, 2007. An efficient algorithms for capacitated multi-facility location problems. *J. Comput. Sci.*, 3: 583-591. <http://www.scipub.org/fulltext/jcs/jcs38583-591.pdf>

Singhtaun, C. and P. Charnsathikul, 2008. Efficient heuristics for single-source multi-facility Weber problems. *Proceedings of the 38th Computer and Industrial Engineering Conference*, Oct. 31-Nov. 2, Publishing House of Electronics Industry, Beijing, China, pp: 35-41. http://pirun.ku.ac.th/~fengcsr/Research/38cie_chan_siri.pdf