# Comparing heuristic algorithms for the logistics network design model considering demand fill rate constraint*

Hyun-Woong Jin
Associate Professor, Dept. of Business Administration, Hannam University
(*jin@hannam.ac.kr*)

The design of logistics system satisfying demand of each retailer has been one of the important issues in Supply Chain Management. Therefore, various models including the fixed charge facility location model (FCFLM) have been proposed so as to present logistics network.

As more researchers are interested in FCFLM, some practical issues are incorporated into the traditional FCFLM. For example, Shen et al.(2003) considered new location model combining FCFLM with inventory management model. Another extension of FCFLM is to consider demand fill rate which is measured by the ratio of demand fulfilled within the given time or fulfilled by a DC located within given distance.(Melo et al., 2009)

This paper deals with the combined model of FCFLM and inventory management model considering demand fill rate constraint. The considered model is formulated as a non-linear integer programming and two heuristic algorithms based on Tabu Search and GRASP are proposed. To compare the performance of the proposed algorithms, 1620 data sets considering 5 design factors are randomly generated. The performances of the algorithms are measured in terms of the average objective function value and the average elapsed time. Test results show that Tabu Search based heuristic algorithm outperforms GRASP based heuristic algorithm in terms of the effectiveness and GRASP based heuristic algorithm outperforms in terms of Tabu Search based heuristic algorithm in terms of the efficiency.

The contributions of this paper are two-fold. First, it developed a new heuristic algorithm based on Tabu Search to solve the combined model of FCFLM and inventory management model considering demand fill rate constraint. Second, it compared the performance of the proposed algorithm with the existing GRASP algorithm.

Key words: Logistics System, Inventory management, GRASP, Tabu Search, Demand fill rate

## Ⅰ. Introduction

The design of logistics system satisfying demand of each retailer is one of the important issues in Supply Chain Management. Therefore, various models including the fixed charge facility location model (FCFLM) have been proposed so as to present logistics network. (Daskin, 1995)

For FCFLM, the amount of demand and location of each retailer are given as well as

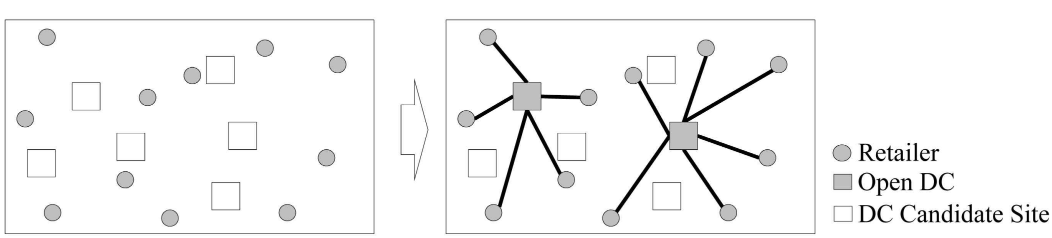
논문접수일: 2013. 2 게재확정일: 2013. 4
* This work was supported by 2013 Hannam University Research Fund.

the location of candidate sites for DC. Given the above parameters, FCFLM decides the number of DCs to be installed and their locations. Moreover, FCFLM allocates each retailer to one of the installed DC so as to minimize the total cost which is the sum of setup cost for opening DCs and delivery cost from DC to its assigned retailers. Figure 1 shows a simple example of FCFLM.

As more researchers are interested in FCFLM, some practical issues are incorporated into the traditional FCFLM. For example, Shen et al.(2003) considered new location model combining FCFLM with inventory management model. Usually, inventory management corresponds to the short term tactical level decision whereas facility location corresponds to the long term strategic level decision. By incorporating the tactical level decision with the strategic level decision, more consistent decisions between the different levels are expected.

Another extension of FCFLM is to consider demand fill rate which is measured by the ratio of demand fulfilled within the given time or fulfilled by a DC located within given distance.(Melo et al., 2009) Initially, demand fill rate has been introduced for the emergency service facility location problem. (Toregas et al., 1971; Khumawala, 1975) Toregas et al.(1971) introduced the emergency service facility location model in which a maximum distance requirement is explicitly imposed between demand nodes and their nearest facility site. The objective of the problem is to minimize the total number of service facilities which are required to meet the response time or distance standards for each of the users.

Even though demand fill rate has been introduced for the emergence service facilities, it could be applied to the private service facilities also.(Berman et al., 2003) Considering the demand fill rate in the private sector would be practical in situations where customers wait for their service requests to be satisfied until a certain time limit.(Jin, 2003) In such a situation, if the delivery time to a retailer from its supplying DC takes longer than allowable time limit, then the demands




〈Figure 1〉 An Example of FCFLM

are treated as lost. Since the importance of the customer satisfaction increased, more researchers become to be interested in demand fill rate.(Banerjee and Paul, 2005; Rim and Park, 2008; Tempelmeier, 2011) Some of them considered demand fill rate as an objective function(Sabri and Beamon, 2000; Axsater, 2006; Altiparmak et al., 2006) and others considered it as a constraint(Nozick, 2001; Rim and Park, 2008; Tempelmeier, 2011).

This paper deals with the combined model of FCFLM and inventory management model considering demand fill rate constraint. Recently, this model was considered by Jin(2010) and a GRASP based heuristic algorithm was proposed. However, the effectiveness of the algorithm was not verified since there were no algorithms to be compared. Therefore, in this paper, new heuristic algorithm based on Tabu Search is proposed and compared with the existing GRASP based heuristic algorithm.

The rest of this paper is organized as follows. Section 2 describes the proposed problem and derives its mathematical formulation. Section 3 describes the heuristic solution approach based on Tabu Search. Section 4 discusses the results of numerical experiments to verify the performance of the proposed solution approach. Section 5 makes concluding remarks.

# Ⅱ. Problem Formulation

## 2.1 Problem Description

The considered problem in this paper is described as follows. Consider a distribution network consisting of multiple DC candidate sites and retailers as shown in Figure 1. If the location of each DCs are decided, each retailer is assigned to one of the installed DCs and products are delivered from DC to retailer so as to meet its daily demands. It is assumed that daily demands at each retailer are independent and follow a Poisson distribution. (Daskin et al., 2002; Ozsen et al., 2008; Park et al., 2010) Moreover, it is also assumed that aggregated daily demands at each DC are independent and follow a normal distribution because it is known that a Poisson demand process can be approximated by normal distribution for sufficiently large values. (Montgomery et al., 1998)

A DC holds on-hand inventory so as to satisfy the customer demands. On-hand inventory at DC is controlled under $(r, Q)$ policy where the fixed amount $Q$ is ordered if the level of on-hand inventory reaches the re-order point $r$. A DC also holds safety-stocks to meet the demands during its lead time. It is assumed that lead time is fixed but it may vary between DCs. The level of safety stock at each DC is determined according to the service level during the lead time.

The considered model makes both decisions on the facility location and the inventory management. Decisions on the facility location include the number of DCs to be installed and their location as well as the assignment of each retailer. Decisions on the inventory management include ordering quantity ($Q$), re-order point($r$) and level of safety stocks ($SS$) at each installed DC. The objective of the model is to minimize the sum of setup cost for DCs and their operating cost which consists of delivery cost and inventory cost. Moreover, inventory cost at each DC is also composed of on-hand inventory management cost and safety stock holding cost.

## 2.2 Mathematical Formulation

This section derives a mathematical formulation of the proposed problem in Section 2.1. For a mathematical formulation of the considered problem, the following notations are introduced.

- Indices
  - $I$ : index for retailers ($i$=1,2,⋯,$n$)
  - $j$ : index for DC candidates sites ($j$=1,2, ⋯,$m$)
- Decision Variables
  - $Y_j$ : 1 if a DC is installed on the candidate site $j$, and 0 otherwise
  - $X_{ij}$ : 1 if retailer $i$ is assigned to DC $j$, and 0 otherwise
- Parameters
  - $\mu_i$ : mean of daily demands at retailer $i$
  - $\chi$ : number of working days per year
  - $F_j$ : setup cost for opening a DC at candidate site $j$
  - $O_j$ : fixed cost per order at DC candidate site $j$
  - $H_j$ : inventory holding cost per unit at DC candidate site $j$
  - $C_{ij}$ : delivery cost per unit from DC candidate $j$ to retailer $i$
  - $L_j$ : lead time at DC candidate site $j$
  - $\alpha$ : required service level at each DC
  - $Z_\alpha$ : standard normal deviate such that $\Pr(Z \leq Z_\alpha) = \alpha$.
  - $\delta$ : required demand fill rate
  - $COVER_{ij}$ : 1 if demand at retailer $i$ can be fulfilled from DC $j$ within given time, and 0 otherwise

As mentioned above, the objective function of the considered model consists of setup cost for DCs and delivery cost from each DC to its retailers and inventory cost at each DC. The setup cost can be formulated as $(\Sigma_{j\in M} F_j Y_j)$ and the delivery cost can be formulated as $(\Sigma_{i\in N}\Sigma_{j\in M} C_{ij}\, \chi\, \mu_i\, X_{ij})$. The inventory cost is composed of on-hand inventory management cost and safety stock holding cost. Moreover, the on-hand inventory cost is also composed of ordering cost and holding cost. Denoting by $Q_j$ ordering quantity at DC $j$, the on-hand inventory cost at DC $j$ would be as follows.

$$O_j \frac{\Sigma_{i \in N} \chi \ \mu_i X_{ij}}{Q_j} + H_j \frac{Q_j}{2}$$

The optimal ordering quantity at DC $j$ can be derived as follows by differentiating the above formula.

$$Q_j^* = \sqrt{\frac{2 O_j \Sigma_{i \in N} \chi \ \mu_i X_{ij}}{H_j}}$$

The safety stock holding cost at DC $j$ depends on the aggregated demands from assigned retailers during the lead time and predetermined service level. Since demands at each retailer are assumed to follow a Poisson distribution and demands at each DC are approximated by a normal distribution, safety stock holding cost at DC $j$ can be formulated as $\left(H_j Z_\alpha \sqrt{L_j \Sigma_{i \in N} \mu_i X_{ij}}\right)$.

By considering the above discussion, the considered model can be formulated as follows.

$$min \sum_{j \in M} F_j Y_j + \sum_{i \in N} \sum_{j \in M} C_{ij} \chi \mu_i X_{ij} + \sum_{j \in M} \left( \sqrt{2 O_j H_j \sum_{i \in N} \chi \mu_i X_{ij}} + H_j Z_\alpha \sqrt{L_j \sum_{i \in N} \mu_i X_{ij}} \right) \quad \text{Eq.(1)}$$

$$s.t. \quad \frac{\sum_{i \in N} \sum_{j \in M} COVER_{ij} \ \mu_i \ X_{ij}}{\sum_{i \in N} \mu_i} \geq \delta \quad \text{Eq.(2)}$$

$$\sum_{j \in M} X_{ij} = 1 \quad \forall i \in N \quad \text{Eq.(3)}$$

$$X_{ij} \leq Y_j \quad \forall i \in N, \ \forall j \in M \quad \text{Eq.(4)}$$

$$X_{ij}, Y_j \in \{0,1\} \quad \forall i \in N, \ \forall j \in M \quad \text{Eq.(5)}$$

The objective function (1) is to minimize the sum of setup cost, delivery cost and inventory management cost. Eq.(2) represents demand fill rate constraint where the ratio of demands fulfilled within a given time should be larger than the predetermined level $\delta$. Eq.(3) requires that each retailer is assigned to one DC and Eq.(4) does not allow any delivery unless the corresponding DC is open. Eq.(5) means that the decision variables are binary variables.

## Ⅲ. Solution Approach

This section introduces heuristic algorithms to solve the considered model. The first subsection describes GRASP based heuristic algorithm which was proposed by Jin(2010). And the second subsection describes Tabu based heuristic algorithm which is developed in this paper.

### 3.1 GRASP based heuristic algorithm

A GRASP(Greedy Randomized Adaptive Search Procedure) is a meta-heuristic proposed by Feo and Resende(1995). It is an iterative process consisting of a solution construction phase and a local search phase.

Solution construction phase produces a feasible solution by adding one element of the solution at each iteration. To select a candidate element, a greedy function which measures the attractiveness of each element is adopted. The attractiveness of an element is updated at each iteration by reflecting previous selection. GRASP keeps elements showing high attraction in a list called RCL(Restricted Candidate List) and randomly selects an element among RCL. The difference between GRASP and the traditional greedy heuristic is that GRASP randomly selects promising element in RCL whereas traditional greedy heuristic selects the best element only. Therefore, it may be claimed that GRASP incorporates the diversification strategy by considering RCL and the intensification strategy by considering the greedy function.

Local search phase of GRASP is an iterative process to find a local optimum solution by searching the neighbourhood of current solution found in the previous iteration. Local search phase ends when it does not improve solution any more. The solution found in local search phase is compared with current best solution and updates current best solution, if needed. Solution construction phase and local search phase are iteratively repeated until termination condition is satisfied.

Solution construction phase of the GRASP based heuristic algorithm for the considered model is composed of 3 subphases. The first subphase iteratively selects DCs satisfying demand fill rate constraint. The second subphase selects additional DCs which can improve the objective function value. The last subphase closes unnecessary DCs.

In local search phase of the considered GRASP algorithm, the set of open DCs are fixed and the assignments of retailer are changed. To design a local search phase, it is needed to define a "*move*" from current solution to a neighbourhood solution. In this algorithm, two types of move are considered. One is a shift move in which single retailer changes its assignment. The other is a swap move in which a pair of retailer assigned different DCs exchanges their assignments.

The detail of GRASP based heuristic algorithm for the considered model can be found in Jin(2010).

## 3.2 Tabu Search based heuristic algorithm

### 3.2.1 Tabu Search heuristic

Tabu Search is a local search type metaheuristic proposed by Glover(1989, 1990) to solve various combinatorial optimization problems. It uses a short-term memory called Tabu list which records historical information of pervious solutions to guide the local search process.

Tabu list helps the search process escape from being stuck at a local optimum solution since it prevents the solution process from repeating the track of previous solution search.

Since Tabu Search is a kind of local search heuristic, it needs to define a neighborhood solution which can be reached with one move from the current solution. In this paper, two types of moves which are used in the GRASP based heuristic algorithm are adopted also; a shift move and a swap move.

A shift move means that one retailer changes its assigned DC. For example, let assume a retailer $i$ moves its assigned DC from $j_1$ to $j_2$. If DC candidate $j_2$ is already open and there are other retailers assigned to DC candidate $j_1$, then the set of open DC is not changed with the shift move. Therefore, the objective function value of the neighborhood solution can be calculated by updating the delivery cost of retailer $i$ and the inventory cost of DC candidate $j_1$ and $j_2$. If DC candidate $j_2$ is not open yet, then DC should be open at candidate site $j_2$ so that the set of open DCs is changed. In this case, the setup cost for DC candidate $j_2$ needs to be added as well as the delivery cost and inventory management cost are updated. If retailer $i$ is the only retailer assigned to DC candidate $j_1$, then DC $j_1$ needs to be closed so that the set of open DCs is changed. In this case, the setup cost for DC candidate $j_1$ needs to be deleted as well as the delivery cost and inventory management cost are updated.

A swap move means that a pair of retailers assigned to different DCs exchange their assignment. For example, let assume retailer $i_1$ which is assigned to DC candidate $j_1$ and retailer $i_2$ which is assigned to DC candidate $j_2$ exchange their assignments so that retailer $i_1$ is assigned to DC candidate $j_2$ and retailer $i_2$ is assigned to DC candidate $j_1$. Since the assigned DCs are already open, with the swap move, the set of open DCs is not changed so that it needs to update the delivery cost and the inventory cost only to calculate the objective function value of the neighborhood solution.

Traditional Tabu Search algorithm searches all the possible neighbourhood solutions so as to move the current solution to the best neighborhood solution. Therefore, in this problem, the number of possible neighbourhood solution is $\{n(m-1)+n(n-1)/2\}$. Since large number of neighborhood solution takes long time to search all the possible neighbourhood solutions at every iteration, in this algorithm, the iteration ends when a neighborhood solution providing lower objective function value than the current solution is found and the current solution moves to the neighborhood solution.

Tabu Search algorithm records the characteristics of the previous solution in the Tabu List so as to escape from local optimum solution. In this algorithm, the changed retailer and its previously allocated DC are recorded in the Tabu list. Therefore, at the specific iteration, if retailer $i$'s allocation is changed from DC $j_1$ to $j_2$, then retailer $i$ is not allowed to be assigned to $j_1$ again at the next iteration. However, if the assignment generates the best solution which provides lower objective function value than the current best solution, then the assignment is allowed. This is called as an aspiration condition.

An element of tabu list graduates its tabu status after some iterations. The number of iterations where an element continues its tabu status is called Tabu tenure. In this algorithm, dynamic tabu tenure is adopted. That is, if the objective function value of the best neighborhood solution is better than the current solution, then tabu tenure is decreased by one so that the algorithm searches promising area more thoroughly. If the objective function value of the best neighborhood solution is worse than the current solution, then tabu tenure is increased by one so that the algorithm searches wider area. In this algorithm, the minimum number of Tabu tenure is set to 2 and the maximum of Tabu tenure is set to $n$ which can restrict all retailers from returning to the previously assigned DC. At the first iteration, Tabu tenure is set to $n/2$.

The iteration ends when the predetermined termination condition is satisfied. In this algorithm, the algorithm ends when the number of iteration reaches the maximum value which is set to $(nm)$ or the best objective function value is not improved within $(2n)$ iterations.

3.2.2 Initial Solution

In general, a greedy type method generates a local optimum solution in an efficient way. That is why, in some papers, a greedy method is successfully adopted as a solution procedure to find an initial solution.(Ahuja et al., 2000; Feo and Resende, 1995)

In this algorithm, initial solution is obtained by applying greedy method to the construction phase of GRASP based heuristic algorithm. That is, in the first step, a DC candidate site providing the lowest value of the following greedy function $G^1(j)$ among the set of undecided candidate sites is selected to open DC until the demand fill rate constraint is satisfied.

$$G^1(j)= \frac{F_j + \sum_{i \in N_j^1} C_{ij}\,\chi\,\mu_i + \sqrt{2O_jH_j \sum_{i \in N_j^1} \chi\,\mu_i} + H_jZ_\alpha \sqrt{L_j \sum_{i \in N_j^1} \mu_i}}{\left|N_j^1\right|} \quad \text{Eq.(6)}$$

In Eq.(6), $N_j^1$ is the set of unassigned retailers whose demand can be satisfied from the candidate site $j$ within the given time limit.

In the second step, a DC candidate site providing the lowest value of the following greedy function $G^2(j)$ among the set of undecided candidate sites is selected to open DC until the improvement of the objective function value is not possible.

$$\begin{aligned} G^2(j) = F_j + \sum_{i \in N_j^2} C_{ij}\, \chi\, \mu_i + \sqrt{2 O_j H_j \sum_{i \in N_j^2} \chi\, \mu_i} \\ + H_j Z_\alpha \sqrt{L_j \sum_{i \in N_j^2} \mu_i} \qquad \text{Eq.(7)} \\ - \Big\{ \sum_{i \in S_j^2} C_{iJ(i)}\, \chi\, \mu_i + \sum_{i \in S_j^2} \sqrt{2 O_{J(i)} H_{J(i)} \chi\, \mu_i} \\ + H_{J(i)} Z_\alpha \sqrt{L_{J(i)}\, \mu_i} \Big\} \end{aligned}$$

In Eq.(7), $N_j^2$ is the set of retailers whose demand can be satisfied from the candidate site $j$ within the given time limit. And $S_j^2$ is the subset of $N_j^2$ in which the retailer $i$ is already assigned to different DC $J(i)$.

In the third step, a DC candidate site providing the lowest value of the following greedy function $G^3(j)$ among the set of open DCs is selected to close the DC until the improvement of the objective function value is not possible.

$$\begin{aligned} G^3(j) = -F_j + \sum_{i \in N_j^3} C_{iK(i)}\, \chi\, \mu_i \\ + \sum_{i \in N_j^3} \left( \sqrt{2 O_{K(i)} H_{K(i)} \chi\, \mu_i} + H_{K(i)} Z_\alpha \sqrt{L_{K(i)} \mu_i} \right) \\ \text{Eq.(8)} \end{aligned}$$

In Eq.(8), $N_j^3$ is the set of retailers which is currently assigned to $j$ but its demand can be satisfied from other DCs within the given time limit. And $K(i)$ is the installed DC satisfying the demand of retailer $i$ within a given time with the lowest cost except DC $j$.

The detailed description of the procedure for initial solution can be found in Jin(2010).

#### 3.2.3 Intensification Strategy and Diversification Strategy

As more researchers are interested in Tabu Search, various methodologies to enhance the performance of the traditional Tabu Search have been proposed. One of them is the intensification strategy which focuses on the area including elite solutions and other one is the diversification strategy which guides the search process to the unvisited solution area.

In this algorithm, dynamic Tabu tenure can be recognized as an implementation of the intensification strategy since it searches the area having elite solutions more thoroughly by reducing Tabu tenure. Moreover, it also can be recognized as an implementation of the diversification strategy since it guides the search process to the unvisited area by increasing Tabu tenure.

Besides the dynamic Tabu tenure, in this algorithm, a long-term memory and a strategic oscillation are adapted so as to enhance the performance of the algorithm.

A long-term memory is a kind of the diversification strategy and it records how often the candidate sites are selected during the previous iterations. That is, there are $m$

elements in the long-term memory and at each iteration the set of open DCs are recorded in the long-term memory so as to the corresponding element of long-term memory increases by one. If there are no improvement of the current best objective function value during the last $n/2$ iterations, the current solution and tabu list are initialized and new set of open DCs are constructed from long-term memory. In this case, the element having the smallest value is open and retailers whose demands are covered by the candidate sites within the given time limit are assigned to the newly open DC until the demand fill rate constraint is satisfied. If the demand fill rate constraint is satisfied, then the remaining retailers are assigned to the open DC which provides the smallest delivery cost. Since this procedure constructs the set of open DCs among the least often selected DCs, newly constructed solution seems to be located in the unvisited solution area. And the following iterations will search the unvisited solution area.

A strategic oscillation is a kind of the diversification strategy which allows a move to the infeasible solution so as to search diverse solution area. In this algorithm, if the best feasible neighborhood solution does not improve the current solution, then the neighborhood solution providing the best objective function value is selected as a next solution regardless its feasibility. If the selected solution is infeasible, at the next iteration, then the infeasibility of the neighborhood solution is considered as a form of penalty function at the objective function value so as to guide its search to the feasible solution area. The penalty function is set to the product of the infeasibility and the penalty constant. And the penalty constant is set to the objective function value of the initial solution.

### 3.2.4 Tabu search solution procedure

The above discussed issues on Tabu search algorithm are now put together as the following solution procedure:

**STEP 1. Initialization.**

1.1. Find the initial solution.
   - Derive the initial solution as described in the subsection 3.2.2.

1.2. Initialize the parameters.
   - Initialize *Current_Solution*, *Current_Best_Solution*, *Tabu_List*, *Tabu_Tenure* and *Longterm_ Memory*.

**STEP 2. Updating *Current_Solution*.**

2.1. If *Current_Solution* is infeasible,
   - Select the first found solution providing less infeasibility than *Current_Solution* among the neighborhood. If all the neighborhood solution provide more infeasibility than *Current_Solution*, select the solution provid-

ing the smallest infeasibility among the neighborhood. The selected solution should not be included in *Tabu_List* unless it satisfies the aspiration condition.

2.2. If *Current_Solution* is feasible,
- Select the first found feasible solution providing less objective function value than *Current_Solution* among the neighborhood. If all the neighborhood solution provide bigger objective function value than *Current_Solution*, select the solution providing the smallest objective function value regardless of feasibility. The selected solution should not be included in *Tabu_List* unless it satisfies the aspiration condition.

**STEP 3. Updating Parameters.**

3.1. Update *Tabu_List*.
- Include the changed retailer and its previously allocated DC in STEP 2 as a new element of *Tabu_List*.
- Delete the oldest element from *Tabu_List*.

3.2. Update *Tabu_Tenure*.
- If *Current_Solution* provides better objective function value than the previous one and *Tabu_Tenure* is larger than 2, then decrease *Tabu_Tenure* by 1.
- If *Current_Solution* does not provide better objective function value than the previous one and *Tabu_Tenure* is smaller than $n$, then increase *Tabu_Tenure* by 1.

3.3. Update *Current_Best_Solution*, if needed.
- If *Current_Solution* provides better objective function value than *Current_Best_Solution*, then update *Current_Best_Solution* with *Current_Solution*.

3.4. Update *Longterm_Memory*.
- For the open DCs in *Current_Solution*, increase the corresponding elements of *Longterm_ Memory* by 1.
- If *Current_Best_Solution* has not been updated during the last $n/2$ iterations, new *Current_Solution* is derived from *Longterm_Memory* as described in the subsection 3.2.3.

**STEP 4. Termination Condition.**

4.1. Check the termination condition described in the subsection 3.2.1.
- If it is satisfied, then stop.
- Otherwise, go to STEP 2.

3.2.5 Validity of the algorithm

Tabu Search is a kind of meta-heuristics which are general combinatorial optimization techniques rather than dedicated to the specific problem. These general techniques are flexible so as to handle various types of combinatorial optimization problems. Because of the flexibility, it is important to adjust the main ingredients of the general technique to the considered problem. And it is needed to validate whether the adjustment process is appropriate or not.

Even though there does not exist a specific methodology to validate meta-heuristic based algorithms, Hertz and Widmer(2003) proposed the following basic principles which are helpful to successful design of meta-heuristic based algorithm.

*Principle 1.* It should be easy to generate neighborhood solutions.

*Principle 2.* All the considered solution should contain a path linking to an optimal solution.

*Principle 3.* Neighborhood solutions should be in some sense close to the current solution.

*Principle 4.* The topology induced by the cost function of neighborhood solutions should not be too flat.

Since a shift move and a swap move are the only ways to generate neighborhood solution, the proposed Tabu Search based heuristic algorithm may conform to Principle 1 and Principle 3. And the proposed algorithm may conform to Principle 2 because all the feasible solutions can be reached from any feasible solution by combining the above mentioned moves. Principle 4 requires that the objective function values of the neighborhood solutions need to be different from each other so that they can be prioritized. In the proposed algorithm, the objective function value of a neighborhood solution cannot be the same to that of any other neighborhood solution unless the delivery costs and the inventory costs are same. Therefore, the proposed Tabu Search based heuristic algorithm seems to conform to Principle 4.

As a result, it may be claimed that the proposed Tabu Search based heuristic algorithm is acceptable since it conforms to the Principles of meta-heuristics proposed by Hertz and Widmer(2003).

## Ⅳ. Experimental Results

To evaluate and compare the performance of the GRASP based heuristic algorithm described in Section 3.1 and the Tabu Search based heuristic algorithm described in Section 3.2, experimental tests under various environments were performed. Both algorithms were coded in C++ language on a Pentium IV CPU 2.0 GHz 1GB RAM desktop computer.

### 4.1 Data Set

For the experimental tests, randomly generated data sets were constructed by considering the following design factors; the size of network, the level of demand variation between retailers, the level of cost variation between DC candidates, the level of coverage ratio and the level of demand fill rate.

The size of network is defined by the number of retailers and the number of DC candidates. In this paper, four different network sizes are considered to have (30,20), (50,30), (80,40), (100,50), where $(n,m)$ means that the considered network has $n$ retailers and $m$ DC candidates. Given the network size, the coordinate of each retailer and DC candidate is randomly generated within a predetermined rectangular area. The delivery cost from a DC to a retailer is randomly generated from 0 to its Euclidean distance. Therefore, if DC candidate $j_1$ is located closer to retailer $i$ than DC candidate $j_2$, the delivery cost from $j_1$ to $i$ has high chance to be smaller than the delivery cost from $j_2$, even though there are some probability that delivery cost from $j_1$ to $i$ is higher than the delivery cost from $j_2$ to $i$.

The level of demand variation between retailers is defined as $\max_{i\in N}(\mu_i)/\min_{i\in N}(\mu_i)$. The level is set to small if the ratio is between 1.0 and 2.0, medium if the ratio is between 2.0 and 5.0 and large if the ratio is between 5.0 and 10.0. Table 1 shows actual variation between the maximum daily demand and the minimum daily demand of sample data sets.

The level of cost variation between DC candidates is defined as the ratio of the maximum and minimum value of each cost (ordering cost and inventory holding cost) between DC candidates. The level is set to small if the ratio is between 1.0 and 2.0, medium if the ratio is between 2.0 and 5.0 and large if the ratio is between 5.0 and 10.0. Table 2 shows actual variation between the maximum cost and the minimum cost of sample data sets.

The level of coverage ratio is defined as the ratio of the number of DC-retailer pairs satisfying the demand fill rate. That is, $\Sigma_{i\in N}\Sigma_{j\in M}COVER_{ij}/(nm)$. The level is set to small if the ratio is between 0.1 and 0.3, medium if the ratio is between 0.3 and 0.5, large if the ratio is between 0.5 and 0.7. Table 3 shows actual ratio of the coverage of sample data sets.

The level of demand fill rate corresponds to the parameter $\delta$ in Eq.(2). The level is set to

〈Table 1〉 Demand variation of sample data sets.

| | small | | | medium | | | large | | |
|---|---|---|---|---|---|---|---|---|---|
| | min | max | ratio | min | max | ratio | min | max | ratio |
| daily demand | 2.09 | 3.84 | 1.84 | 1.13 | 4.83 | 4.26 | 0.67 | 5.28 | 7.89 |

〈Table 2〉 Cost variation of sample data sets.

| | small | | | medium | | | large | | |
|---|---|---|---|---|---|---|---|---|---|
| | min | max | ratio | min | max | ratio | min | max | ratio |
| ordering cost | 109.2 | 200.8 | 1.84 | 57.5 | 229.3 | 3.98 | 40.8 | 226.3 | 5.56 |
| holding cost | 36.6 | 61.2 | 1.68 | 20.5 | 76.1 | 3.72 | 8.5 | 76.1 | 9.01 |

〈Table 3〉 Coverage ratio of sample data sets.

| small | | | medium | | | large | | |
|---|---|---|---|---|---|---|---|---|
| total pairs | covered pairs | ratio | total pairs | covered pairs | ratio | total pairs | covered pairs | ratio |
| 600 | 111 | 0.19 | 600 | 273 | 0.46 | 600 | 377 | 0.63 |

small if the ratio is 0.5, medium if the ratio is 0.7, large if the ratio is 0.9.

To secure the reliability of the experimental test, 5 data sets are randomly generated at each combination of the design factors. Therefore, total 1620 data sets are randomly generated since there are 4 levels of network sizes, 3 levels of the demand variation, 3 levels of the cost variation,3 levels of the coverage ratio and 3 levels of the demand fill rate.

### 4.2 Experimental Test

To compare the performance of the GRASP based heuristic algorithm(GRASP) and the Tabu Search based heuristic algorithm(TS), the average objective function value and the average elapsed time at each combination of the design factors are measured. The average objective function value is used to evaluate the effectiveness of those algorithms and the average elapsed time is used to evaluate the efficiency of those algorithms. Table 4 shows the summary of the experimental test.

The average objective function value of GRASP with the whole data set is 1,748,432.1 and the average elapsed time of GRASP is 1.421 seconds(1,421 milliseconds) whereas the average objective function value of TS with the whole data set is 1,630,367.5 and the average elapsed time of TS is 5.099 seconds(5,099 milliseconds). The overall test results show that TS outperforms GRASP in terms of the effectiveness whereas GRASP outperforms TS in terms of the efficiency.

The average objective function value obtained by TS is less than GRASP by 7.2% and the average elapsed time of GRASP is 27.9% of TS. It conforms to the test result of Delmaire et al.(1999) which considered the

〈Table 4〉 Test results with the randomly generated data sets.

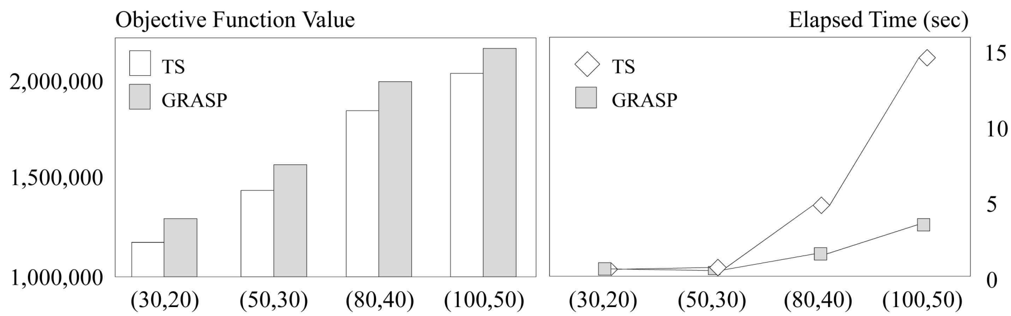
| Design Factor | Level | Objective Value | | Elapsed Time(msec) | |
|---|---|---|---|---|---|
| | | GRASP | TS | GRASP | TS |
| Network size | (30,20) | 1,300,847.1 | 1,185,035.3 | 90.3 | 74.8 |
| | (50,30) | 1,566,643.6 | 1,446,356.0 | 410.0 | 696.6 |
| | (80,40) | 1,985,477.5 | 1,858,376.2 | 1,620.0 | 4,884.1 |
| | (100,50) | 2,140,760.1 | 2,031,702.3 | 3,564.8 | 14,741.5 |
| Level of demand variation | S | 1,786,344.3 | 1,644,160.2 | 1,418.7 | 5,127.9 |
| | M | 1,738,451.3 | 1,626,184.3 | 1,414.1 | 5,023.8 |
| | L | 1,720,500.6 | 1,620,757.9 | 1,431.0 | 5,146.1 |
| Level of cost variation | S | 1,742,280.3 | 1,633,736.2 | 1,404.9 | 5,031.6 |
| | M | 1,784,442.2 | 1,655,326.1 | 1,417.9 | 5,150.2 |
| | L | 1,718,573.7 | 1,602,040.1 | 1,441.0 | 5,116.0 |
| Level of coverage ratio | S | 1,924,425.0 | 1,701,376.9 | 731.4 | 4,682.6 |
| | M | 1,700,843.0 | 1,629,868.5 | 1,454.7 | 5,156.1 |
| | L | 1,620,028.2 | 1,559,857.0 | 2,077.7 | 5,459.2 |
| Level of demand fill rate | S | 1,694,131.2 | 1,550,883.8 | 1,627.6 | 5,639.3 |
| | M | 1,718,412.0 | 1,557,216.7 | 1,471.5 | 5,282.4 |
| | L | 1,832,753.0 | 1,783,001.9 | 1,164.7 | 4,376.1 |

single source capacitated plant location problem with several meta-heuristic algorithms including Evolutive Algorithms, GRASP, Simulated Annealing and Tabu Search. Demlaire et al. (1999) reported that Tabu Search showed the best performance in terms of the effectiveness whereas GRASP showed the best performance in terms of the efficiency.

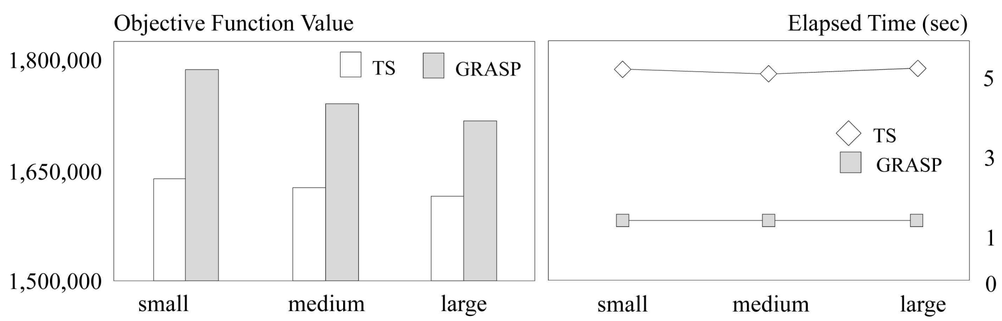
Detailed test results in terms of the design factors can be shown from Figure 2 to Figure 6. Figure 2 shows test results in terms of the network size.

As expected, the objective function value and the elapsed time of both algorithms increase as the network size increases. In terms of the objective function value, TS showed better results than GRASP at all the network sizes even though any trends on the gap between TS and GRASP regarding the network size cannot be found. In terms of the elapsed time, GRASP showed better results than TS at all the network sizes and the gap increased as the network size increases. Therefore, it may be claimed that GRASP is suitable at the large size network whereas TS is suitable at the small and medium size network.

Figure 3 shows that demand variation across retailers does not affect the objective function




〈Figure 2〉 Test results in terms of the network size

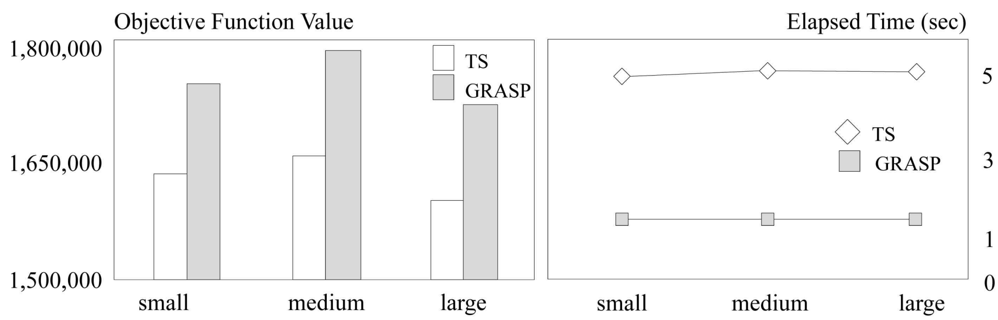



〈Figure 3〉 Test results in terms of the demand variation

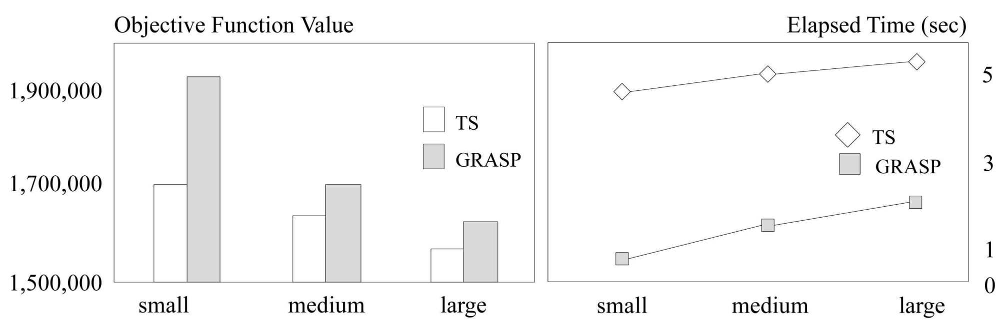
value nor the elapsed time of TS. However, GRASP is more effective at the large demand variation rather than the small demand variation. No trend is found in the elapsed time of GRASP corresponding to the demand variation. Therefore, it may be claimed that TS is robust in terms of the demand variation.

Figure 4 shows the effect of cost variation on the objective function value and the elapsed time of TS and GRASP. Test results show that the objective function value at the medium level of cost variation is larger than that of other levels. However, test results show that the cost variation does not affect the efficiency of both algorithms.

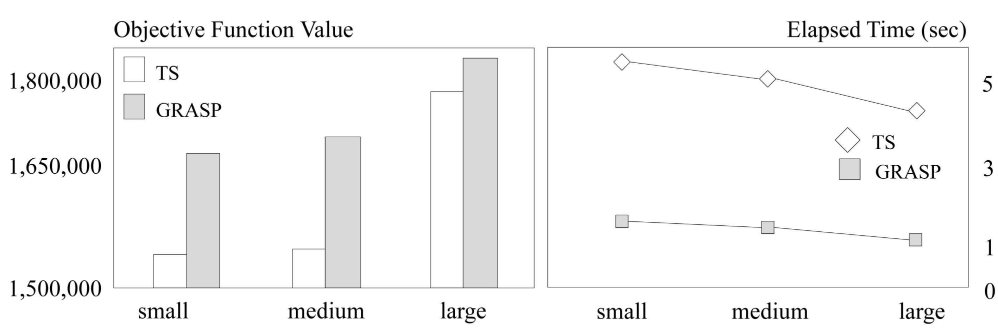
According to Figure 5, the objective function values of both algorithms tend to decrease as the coverage ratio increases. It may be inferred that the feasible solution area become wider as the coverage ratio increases. Wide feasible solution area results in the decrease of the objective function value of both algorithms as well as the increase of the elapsed time of both algorithms.




〈Figure 4〉 Test results in terms of the cost variation




〈Figure 5〉 Test results in terms of the level of the coverage ratio




〈Figure 6〉 Test results in terms of the level of the demand fill rate

High demand fill rate reduces the feasible solution area. Therefore, as we can see from Figure 6, the objective function values of both algorithms increase and the elapsed time of both algorithms decrease as the demand fill rate increases.

## Ⅴ. Conclusions

This paper deals with the logistics network design model considering demand fill rate constraint. The considered model makes facility location related decisions such as the number of installed DCs and their locations as well as inventory management decisions such as ordering quantity, re-order point and the level of safety stocks. To make facility location related decisions, the demand fill rate which is the ratio of retailer's demand satisfying within the given time limit is considered.

The considered model is formulated as a non-linear integer programming and two heuristic algorithms based on Tabu Search and GRASP are proposed. To compare the performance of the proposed algorithms, 1620 data sets considering 5 design factors are randomly generated. The performances of the algorithms are measured in terms of the average objective function value and the average elapsed time.

Test results show that Tabu Search based heuristic algorithm outperforms GRASP based heuristic algorithm in terms of the effectiveness and GRASP based heuristic algorithm outperforms in terms of Tabu Search based heuristic algorithm in terms of the efficiency. Therefore, it may be claimed that Tabu Search based heuristic algorithm is suitable for the small and medium size network design whereas GRASP based heuristic algorithm is suitable for the large size network design.

The contributions of this paper are two-fold. First, it developed a new heuristic algorithm based on Tabu Search to solve the combined model of FCFLM and inventory management model considering demand fill rate constraint. Second, it compared the performance of the proposed algorithm with the existing GRASP algorithm.

## References


Ahuja, R., J. Orlin and A. Tiwari(2000), "A greedy genetic algorithm for the quadratic assignment problem," *Computers & Operations Research*, 27, 917-934.

Altiparmak, F., M. Gen, L. Lin and T. Paksoy (2006), "A genetic algorithm approach for multi-objective optimization of supply chain networks," *Computers & Industrial Engineering*, 51, 196-215.

Axsater, S.(2006), "A simple procedure for determining order quantities under a fill rate constraint and normally distributed lead-time demand," *European Journal of Operational Research*, 174, 480-491.

Banerjee, A. and A. Paul(2005), "Average fill rate and horizon length," *Operations Research Letters*, 33, 525-530.

Berman, O., Z. Drezner and G. Wesolowsky(2003), "Locating service facilities whose reliability is distance dependent," *Computers & Opera-*

*tions Research*, 30, 1683-1695.

Daskin, M.(1995), *Network and discrete location: models, algorithms and applications,* New York, USA: John Wiley& Sons, Inc.

Daskin, M., C. Coullard and Z. Shen(2002), "An inventory-location model: Formulation, solution algorithm and computational results," *Annals of Operations Research*, 110, 83-106.

Delmaire, H., J. Diaz and E. Fernandez(1999), "Reactive GRASP and Tabu Search based heuristics for the single source capacitated plant location problem," *INFOR*, 37, 194-225.

Feo, T. and M. Resende(1995), "Greedy randomized adaptive search procedure," *Journal of Global Optimization*, 6, 109-133.

Glover, F.(1989), "Tabu Search - Part I," *ORSA Journal on Computing*, 1, 190-206.

Glover, F.(1990), "Tabu Search - Part II," *ORSA Journal on Computing*, 2, 4-32.

Hertz, A. and M. Widmer(2003), "Guidelines for the use of meta-heuristics in combinatorial optimization," *European Journal of Operational Research*, 151, 247-252.

Jin, H.(2003), "Optimal facility location considering the maximum delivery time constraints and the budget constraint," *Journal of Korean Production and Operations Management Society*, 14, 153-172.

Jin, H.(2010), "A study on the facility location and inventory management model using GRASP algorithm," *Journal of Korean Production and Operations Management Society*, 21, 441-456.

Khumawala, B.(1975), "Algorithm for the p-median problem with maximum distance constraints: Extension and Reply," *Geographical Analysis*, 7, 91-95.

Melo, M., S. Nickel and F. Saldanha-da-Gama (2009), "Facility location and supply chain management - A review," *European Journal of Operational Research*, 196, 401-412.

Montgomery, D., G. Runger and N. Hubele(1998), *Engineering statistics*, New York, USA: John Wiley& Sons, Inc.

Nozick, L.(2001), "The fixed charge facility location problem with coverage restrictions," *Transportation Research Part E: Logistics and Transportation Review*, 37, 281-296.

Ozsen, L., C. Coullard and M. Daskin(2008), "Capacitated warehouse location model with risk pooling," *Naval Research Logistics*, 55, 295-312.

Park, S., T. Lee and C. Sung(2010), "A three-level supply chain network design model with risk-pooling and lead times," *Transportation Research Part E*, 46, 563-581.

Rim, S. and I. Park(2008), "Order picking plan to maximize the order fill rate," *Computers & Industrial Engineering*, 55, 557-566.

Sabria, E. and B. Beamon(2000), "A multi-objective approach to simultaneous strategic and operational planning in supply chain design," *Omega*, 28, 581-598.

Shen, Z., C. Coullard and M. Daskin(2003), "A joint location-inventory model," *Transportation Science*, 37, 40-55.

Tempelmeier, H.(2011). "A column generation heuristic for dynamic capacitated lotsizing with random demand under a fill rate constraint," *Omega*, 39, 627-633.

Toregas, C., R. Swain, C. ReVelle and L. Berman (1971), "The location of emergency service facilities," *Operations Research*, 19, 1363-1373.

# 수요충족률 제약조건을 고려한 물류네트워크 설계 알고리듬*

진현웅**

요　약

본 논문은 물류네트워크의 설계를 위한 휴리스틱 알고리듬을 개발하고 유사 알고리듬과의 비교를 통한 성능평가를 주요 내용으로 한다. 본 논문에서 고려하는 모델은 물류센터의 개수 및 위치를 포함하는 전략단계의 의사결정과 개별 물류센터에서의 주문량 및 재주문점을 포함하는 운영단계의 의사결정을 모두 포함한다. 해당 모델은 비선형 정수계획 모형으로 표현되었으며, 해당 모형의 근사해를 얻기 위해 타부탐색 기법을 활용한 휴리스틱 알고리듬을 개발하였다. 개발된 모형은 기존의 알고리듬인 GRASP 기반 휴리스틱 알고리듬과의 성능비교를 위해, 다양한 환경에서 모의실행을 수행하였다.

주제어: 물류 네트워크, 재고관리, GRASP, Tabu Search, 수요충족률

* 본 연구는 2013년도 한남대학교 교비학술연구비 지원을 받았음.
** 한남대학교 경영학과 부교수