

# 다수 물류기지 재고 및 경로 문제의 유전알고리즘에 의한 해법\*

\*정 재 현\*\*

## An Effective Genetic Algorithm for Solving the Joint Inventory and Routing Problem with Multi-warehouses

\*Jaeheon Jung\*\*

### ■ Abstract ■

In this paper we propose an effective genetic algorithm for solving the integrated inventory and routing problem of supply chain composed of multi-warehouses and multi-retailers. Unlike extant studies dealing with integrated inventory and routing problem of supply chain, our model incorporates more realistic aspect such as positive inventory at the multi-warehouses under the assumption of inventory policy of power of two-replenishment-cycle. The objective is to determine replenishment intervals for the retailers and warehouses as well as the vehicles routes so that the total cost of delivery and inventory cost is minimized. A notable feature of our algorithm is that the procedure for evaluating the fitness of objective function has the computational complexity closing to linear function. Computational results show effectiveness of our algorithm.

Keywords : Transportation, Inventory, Genetic Algorithm, SCM

## 1. 서론

효율적인 구매, 생산, 저장, 운반 및 분배를 아우르는 공급망 관리는 기업의 성공적인 운영을 위해서 매우 중요하다. 공급망 관리의 많은 내용들은 재고 및 수송 관리와 연관되어 있다. 전통적으로 공급망 관리에서는 이 두 가지 관리 요소에 대한 의사결정을 독립적으로 다루어 왔다. 그러나 많은 경우에 이 두 요소에 대한 의사결정은 통합되어 이루어질 필요가 있다. 예컨대, 공급자 관리 재고(VMI : Vendor Managed Inventory)와 같은 경우, 슈퍼마켓이나 백화점에 납품하는 공급자들은 백화점이나 슈퍼마켓의 재고를 스스로의 책임 하에 보충한다. 이 경우 공급자는 재고와 재고 보충을 위한 수송에 대한 의사결정을 동시에 내려야 한다. 또한 다수의 차고지(이하 물류기지)와 전국적인 유통체인점들로 구성된 대형 유통점의 경우에도, 효율적인 운송경로를 따라 재고를 보충하여야하므로 재고와 수송의 통합된 의사결정이 필요하다. 이러한 재고-수송 통합 의사결정문제는 재고 및 경로문제(Inventory Routing Problem)로 알려져 있으며, 이는 크게 유한기간, 무한 기간, 확률적인 3가지 모형이 있다. 본 연구는 무한기간 재고 및 경로 문제를 다루고 있다.

무한기간 재고 경로 문제는 수송 및 재고 비용의 장기 평균 비용 최소화를 목적으로 하여 운송 차량 경로 및 재고주기를 동시에 결정하는 문제이다. 운송 차량경로문제(외판원순회문제 : Traveling Salesman Problem)만 하더라도 매우 복잡한 문제(Non-polynomial complex)로 널리 알려진 문제이므로 이 문제는 복잡도가 증첩된 문제이다. 문제의 복잡도를 낮추기 위해 다수의 연구들은 소매점과 물류기지가 2의 멱수배(POW : power of two policy)의 재고보충주기를 가진다는 가정을 전제하고 있다. 왜냐하면 차량경로를 차치하고라도, 다수 소매점에 대한 최적 재고 의사결정 문제는 Schwarz[16]에 의해 일정하지 않은(non-stationary) 최적의 재고보충주기를 가진다는 것이 증명되었기 때문이다. Roundy

[15]에 의해, 소매점들의 재고 보충시기가 특정한 기본 주기의 2의 멱수배로 이루어질 때(POW : power of two policy) 그 해는 최적해의 6% 이내에 있으며 일정한(stationary) 재고 보충주기를 가진다는 사실이 증명되었다. 본 연구도 이러한 흐름을 따르고 있다. 그러나 다음의 2가지 점에서 기존 연구들과 차별된다.

첫째, 본 연구는 재고를 보유한 다수의 물류기지를 전제로 하고 있다. 대부분의 연구들은 물류기지에서의 재고 보유 또는 다수의 물류기지를 가정하고 있지 않다. 현실적으로는 전국적 체인을 가지고 있는 유통점이나, 그들 유통점에 납품해야 하는 공급자(supplier)는 다수의 물류기지를 보유하고 있으며, 그들 개개의 유통점(이하에서 이들을 소매점이라 부름)들을 수송 비용과 재고비용을 동시에 고려하여 어느 물류기지에 배정해야 되는지의 결정을 기존의 재고 및 경로 문제의 의사결정에 추가해야 된다. 또한 물류기지들도 그 재고비용이 소매점보다 값싼 경우, 재고를 보유하게 될 것이다. 두 번째로, 기존의 해법들은 대부분 점근적으로(Asymptotically) 최적이거나(소매점 수가 많아질 때 최적으로 수렴), 단순한 절차에 의한 휴리스틱을 제공하고 있어 개선의 여지가 있다. 본 연구는 다수의 물류기지를 보유한 재고 및 경로문제에 유전알고리즘(Genetic Algorithm)을 적용하였다. 유전알고리즘은 최근 이산적 조합최적화(Discrete Combinatorial Optimization) 문제에 많이 적용되고 있다. 후술하겠지만, 재고 및 경로 문제에 유전알고리즘을 적용한 연구는 일부 있지만 알고리즘의 전제가 되는 가정에서 본 연구와는 차이가 있다(제 2.1절 참조).

이 논문의 구성은 다음과 같다. 제 2장에서 재고 및 경로 문제의 기존연구를 고찰하고 그 연장선상에서 본 논문이 다루고 있는 문제를 수리모델로 정식화한다. 제 3장에서 유전알고리즘을 설계하며, 제 4장에서는 그 해법의 효율성을 보여주기 위한 계산실험의 결과를 제시한다.

## 2. 문제의 정식화와 가정

### 2.1 기존 연구 고찰

재고 및 경로 문제는 수송과 재고의 복합 의사결정 문제에서 시작되었다[7, 6]. 그러나 이들은 수송 차량경로를 의사 결정 변수로 모델에서 고려하지 않았다. Gallego and Simchi-Levi[8]은 1대의 차량이 다수 소매점들을 동시에 재고 보충하는 것이 아닌, 점 대 점 재고보충(direct shipping)을 가정한 모델에서의 해법을 제시하였다. 그들은 모든 소매점들의 재고 보충량들이 차량 용량의 71% 이상인 경우에, 해당 모델의 장기적인 해의 평균치가 최적치에서 6% 이내에 있음을 보여주었다.

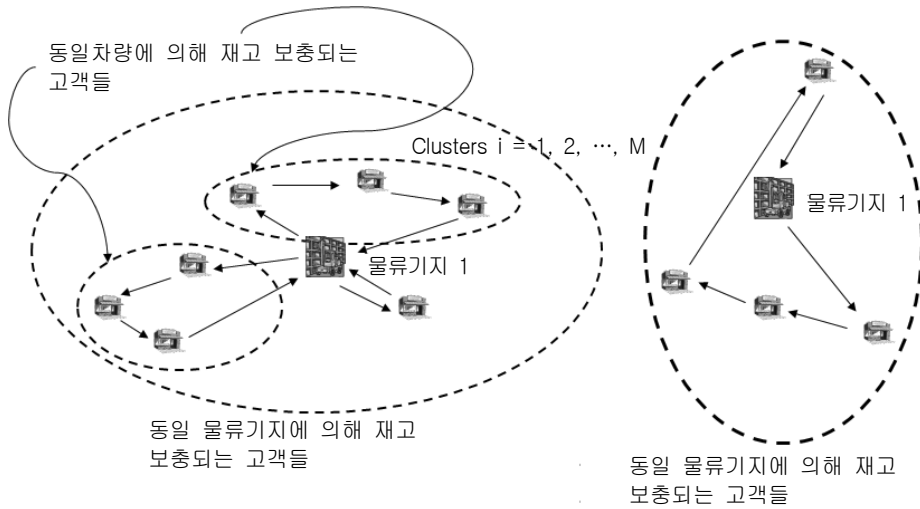
차량 경로를 고려하는 재고 및 경로 문제에서, 재고보충주기가 동일한 소매점들은 동일 차량에 의해 동시에 재고가 보충될 수 있다(<그림 1> 참조), 따라서 동시에 재고 보충될 때의 통합 주문과 재고 유지비용을 최소화 하는 소매점 그룹(동일한 차량 경로를 따라 재고 보충되는 그룹)을 만드는 그룹핑이 중요하게 된다. 연구 흐름은 이 그룹핑을 하는 방법에 따라 크게 3가지의 연구 흐름으로 구별된다.

Anily and Federgruen[3, 4]은 단순한 절차에 따른 소매점들에 대한 분할 정책(partitioning policy)에 의한 해법을 제시하였다. 또한 이들은 재고보충주기가 2의 멱수배이어야 한다는 가정을 전제로, 소매점 수가 많아질 때, 그 해법이 최적의 해에 가까운 해를 제시한다는 것(점근적 최적)을 증명하였다. 물류기지에 재고를 유지하는 것을 가정하지 않았던 이 연구흐름은 물류기지에서 재고를 유지하는 가정을 추가시켜 더욱 발전되었다[4]. 그러나 이 흐름은 점근적 최적해의 성질을 유지하기 위하여 다수의 가정들을 전제해야 하는 한계가 있다.

이와 유사한 연구흐름은 Chan et al.[12]에 의해서 시도되었다. 그러나 이들은 소매점 그룹과 물류기지 사이의 재고보충주기를 동기화시켜, 물류기지에서의 재고보유를 배제하고 있으며, 2의 멱수배 정책

과도 다르게, 재고보충주기가 기본주기의 정수배가 되도록 하는 정책을 취하고 있다. 앞의 경우와 동일하게 단순한 절차에 의한 점근적 최적에 의한 분할 해법을 제시하고 있다. 이 연구흐름은 다수의 물류기지와 단일한 공급자를 가정한 3단계 네트워크의 케이스로 확장되기도 하였으며[13], 동일한 가정 하에 유전알고리즘이 개발되기도 하였다[14].

전술한 두 흐름은 모두 동일차량에 의하여 재고 보충되는 소매점들이 모두 동일한 재고보충주기를 가지는 것을 전제로 하고 있다. 이 점에서 차이가 있는 연구흐름이 Herer and Roundy[9]에 의해 시작되었다. 이들은 동일 차량에 의해 재고 보충되는 소매점들이 각기 다른 2의 멱수배 재고보충주기를 가진다고 가정함으로써, 문제의 구조가 최적화가 보다 쉽게 되는 특성-제 2.2절에서 기술되는 중첩순열(nestedness sequence)의 특성-을 갖게 되었다. 이들은 물류기지에서의 재고 유지를 전제로 유사한 모델에 대한 휴리스틱 해법을 제시하고 있으나, 차량의 용량이 제한되지 않는 비현실적 가정에서 출발하여 적용에 한계가 있었다. 이후 Viswanathan and Mathur[17]에 의해 이 흐름은 더욱 발전하여 간단한 휴리스틱에 의해 소매점 그룹과 최적의 루트, 주문량을 정하는 휴리스틱이 제안되었다. 단 이들은 물류기지에서의 재고보유를 전제로 하지 않았다. 이후 Jung and Mathur[10]는 물류기지에서의 재고보유를 전제로, 먼저 휴리스틱에 의해 소매점 그룹을 정하고, 소매점 및 물류기지에서의 재고보충주기를 최적 해법으로 구하는 순차적인 해법을 제시하고, 이것이 Anily and Federgruen[4]의 점근적 알고리즘보다 우월함을 계산실험을 통해 보여주었다. 본 연구는 이 흐름을 따르면서, 다수의 물류기지에서 재고보유를 하는 경우로 확장시켜, 유전알고리즘을 적용하여 해를 도출하고자 하였다. 유전알고리즘의 경영 의사결정에 대한 적용 연구 사례는 다수 존재하지만, 본 연구와 유사한 다수 차량 경로문제(Vehicle Routing Problem)에 관한 적용 사례는 Prins[11], Baker and Ayeche[5], 양병학[1], 정영훈 외 2인[2]을 들 수 있다.



<그림 1> 다수 물류기지 재고 및 경로 문제의 구조

## 2.2 수리모형 및 기본가정

다수 물류기지를 가진 재고 및 경로 문제는 아래의 <그림 1>과 같이 표현될 수 있다. 이 문제에서는 단일 종류의 차량이(homeogenous vehicles) 특정 물류기지에서 출발하여 다수 소매점(고객)들의 재고를 보충한 이후, 다시 물류기지로 돌아온다. 물류기지 및 소매점 모두가 재고를 보충하여야 한다. 물류기지들은 공장 등의 보다 상위의 동일한 공급지로부터 재고를 보충한다. 우리는 물류기지, 소매점 모두의 주문 비용, 운송거리에 비례해서 증가하는 수송비용, 재고비용의 동시적인 최적화를 목적으로 한다. 이를 위해서 물류기지 중에서, 각 소매점을 위하여 어떤 특정 물류기지를 선택할 것인지를 결정한다(물류기지의 할당). 배정된 물류기지 중에서 동일 차량으로 재고 보충되는 소매점들을 골라 한 그룹으로 묶고(이하에서 이 그룹을 클러스터로 부름) 그 그룹에 속한 모든 소매점들에 대하여 재고 보충주기를 정하며, 동일시기에 같이 재고 보충되는 소매점들에 대해 차량운송을 위한 차량경로를 정한다-즉 외판원순회문제의 해를 구하여야 한다. 제 5장의 계산실험에서는 외판원순회문제의 해법으로 최소 삽입휴리스틱(cheapest insertion heu-

ristic)을 이용하였다-문제의 구조에서 명확하듯이 이러한 의사결정은 순차적으로 이루어지는 것이 아니라 동시에 이루어져야 한다. 그리고 물류기지 및 소매점들의 재고보충주기가 모두 2의 멱수배를 취한다고 가정한다. 후술하겠지만 이 정책은 클러스터 내에서의 소매점들간의 순서를 가정하고(중첩순열 : nestedness sequence), 소매점들의 재고보충주기는 이 순서 값이 큰 순서대로 값을 가져야 한다고 전제한다. 순열은 물류기지 배정, 클러스터와 함께 의사결정변수의 한 부분을 이룬다. 이 전제 아래 모형화를 위해 다음과 같은 의사결정변수들과 파라미터들을 도입한다

- 의사결정변수

$MS_m$  : 물류기지 m에 배정되는 소매점들의 집합

$R_i^m$  : m 물류기지에 배정된 소매점들 중 동일한 차량에 의해 재고 보충될 i번째 소매점들의 집합 (클러스터)

$Seq_i^m$  :  $R_i^m$ 에 속하는 모든 소매점들의 순서를 표시한 순열

$T_{ij}^m$  :  $Seq_i^m$ 의 j번째 소매점의-이하에서  $(i, j)^m$ 로 표현-재고보충주기

$T_m$  : 물류기지 m의 재고보충주기

• 파라미터

$f_{ij}^m$  :  $Seq_i^m$ 의 j번째 소매점까지의 차량 순회비용(vehicle routing cost). 곧 j번째 소매점까지의 외판원순회문제의 해.

$c_{ij}^m$  :  $(i, j)^m$  소매점의 주문비용

$Q$  : 사용되는 차량의 크기

$v_j$  : 1회 차량 운행시 발생하는 고정 차량 비용

$d_{ij}^m$  :  $(i, j)^m$  소매점의 연간 발생 수요량

$h_{ij}^{R^m}$  : 소매점  $(i, j)^m$ 에서의 1단위 제품 보유의 단위 시간당 재고보유 비용

$h_{ij}^m$  : 물류기지 m에서의 소매점  $(i, j)^m$ 에서 요구되는 1단위 제품보유에 발생하는 단위 시간당 재고보유비용

$M$  : 물류기지의 총수

$K_m$  : 물류기지 m에 배정된 소매점들이 이루는 클러스터들의 총수

$N_i^m$  : m 물류기지에 배정된 i번째 클러스터에 속하는 소매점 개수

$C_m$  : 물류기지 m에서의 거래비용(주문비용)

• 수리모형

$$\sum_m M N_{MS} \left\{ M n_{Seq_i^m \subset MS_m} \sum_{Seq_i^m, i=1}^{K_m} \left\{ M N_{T_{ij}^m, T_m} \sum_{j \in Seq_i^m}^{N_i^m} \left[ \frac{b_{ij}^m}{T_{ij}^m} + a_{ij}^{R^m} T_{ij}^m + a_{ij}^m (T_{ij}^m \vee T_m) \right] + C_m / T_m \right\} \right\}$$

s.t.

$$T_{i1}^m \leq T_{i2}^m \leq \dots \leq T_{iN_i}^m \quad (1)$$

$(m = 1, \dots, M), (i = 1, 2, \dots, K_m)$

$$\sum_{j=1}^{N_i} d_{ij}^m T_{ij}^m \leq Q \quad (m = 1, \dots, M), (i = 1, 2, \dots, K_m) \quad (2)$$

$$T_{ij}^m = 2^{t_{ij}^m} B \quad (m = 1, \dots, M) \quad (3)$$

(모든  $i, j$ 에 대하여  $t_{ij}^m \geq 0$ , 정수)

$$T_m = 2^n B \quad (n \geq 0, \text{정수}) \quad (4)$$

• 수리모형 설명 : 결합주문비용의 계산

맨 안쪽의 목적항의 1번째 항은 결합주문비용을

의미한다. 이는  $(i, j)^m$ 까지의 차량 순회비용에 주문비용을 합한 값을 대변하는  $b_{ij}^m$ 에 재고보충 횟수를 곱한 값이다- $b_{ij}^m$ 는 이하에서 설명, m 첨자는 생략-. 상세한 내용은 다음과 같다. B의 2의 멱수배로 모든 소매점과 물류기지의 보충주기가 정하여져 있기 때문에 보충주기가 큰 순서의 소매점이 주문되면, 보충주기가 작은 소매점은 언제든지 같이 재고 보충된다. 이와 같은 특성은 중첩순열이라 불린다. 예컨대, 일년을 52주라고 하고, (i, 1), (i, 2)들의 재고 보충 주기를 모두 2주씩이라고 하자( $T_{i1}, T_{i2} = 2/52$ 년, 여기서 B는  $1/51$ 년, 곧 1주). 그리고 (i, 3), (i, 4) 소매점의 보충주기는 각각 4주와 16주로 즉  $T_{i3} = 4/52, T_{i4} = 16/52$ 로 한다고 하면, (i, 3) 소매점이 재고 보충될 때는 (i, 1), (i, 2) 소매점이 같이 재고 보충된다. 마찬가지로 (i, 4) 소매점들이 재고 보충될 때는 (i, 1), (i, 2), (i, 3) 소매점과 같이 재고가 보충된다. 재고가 보충될 때마다 각 소매점들은 독립적인 주문비용을 가지지 않는다. 주문비용은 각 소매점들의 독립적인 주문비용에 같이 재고 보충되는 소매점들을 순회하는 차량경로비용(routing cost)을 합해서 결합주문비용으로 계산되기 때문이다. 수치 사례에서 같이 재고 보충되는 소매점들의  $\{(i, 1), (i, 2)\}$ 의 연간 총 재고보충 횟수는,  $T_{i2}^{-1}$  곧 26번이 아니다. 왜냐하면, 실제로는 소매점  $\{(i, 1), (i, 2), (i, 3)\}$ 이 같이 재고 보충되는 횟수 13번( $= T_{i3}^{-1}$ )을 제외해줘야 하기 때문이다. 따라서 특정 클러스터 i의 소매점들  $\{(i, 1), (i, 2), (i, 3), \dots, (i, j)\}$ 만의 재고보충 곧 주문횟수는  $T_{ij}^{-1} - T_{ij+1}^{-1}$ 로 계산된다. 또한 (i, j) 소매점까지의 결합주문비용  $K_i(1, 2, \dots, j) = f_{ij} + \sum_{k=1}^j c_{ik}$ 이다. 따라서 (i, j) 소매점까지의 연간 총 결합주문비용은 다음과 같이 계산된다.

$$\sum_{j=1}^{N_i} K_i(1, 2, \dots, j) (T_{ij}^{-1} - T_{ij+1}^{-1})$$

다시 정리하면 다음과 같다(아래에서  $b_{ij}$ 는 수식

에서 해당되는 부분 그대로 정의된다).

$$\begin{aligned} & \sum_{j=1}^{N_i} [K_j(1, 2, \dots, j) - K_j(1, 2, \dots, j-1)] T_{ij}^{-1} \\ &= \sum_{j=1}^{N_i} [f_{i,j} - f_{i,j-1} + c_{ij}] T_{ij}^{-1} = \sum_{j=1}^{N_i} b_{ij} T_{ij}^{-1} \end{aligned}$$

- 수리모형 설명 : 재고보유비용의 계산 및 제약식 맨 안쪽의 2, 3번째 목적항에 대한 설명이다. 물류기지 m과 소매점 모두 재고를 보유하고 있으므로, 물류기지 m에 속한 소매점 (i, j)<sup>m</sup>를 위한 재고보유비용은 다음과 같이 계산된다(Herer and Roundy[7] 참조). 아래에서  $T_{ij}^m \vee T_m = \max(T_{ij}^m, T_m)$ 이며,  $a_{ij}^{Rm}, a_{ij}^{Am}$ 는 수식에서 해당되는 부분 그대로 정의된다.

$$\begin{aligned} & (a_{ij}^{Rm} T_{ij}^m / 2)(h_{ij}^{Rm} - h_{ij}^m) + (h_{ij}^m a_{ij}^{Am} / 2)(T_{ij}^m \vee T_m) \\ &= a_{ij}^{Rm} T_{ij} + a_{ij}^{Am} (T_{ij} \vee T_m) \end{aligned}$$

그리고 제약식 (1)은 순열  $Seq_i^m$  순으로 재고보충주기가 커야하며, 제약식 (2)는 각 차량의 용량 제약을 의미한다. 제약식 (3)과 제약식 (4)는 소매점과 물류기지의 재고보충주기가 B의 2의 멱수배이어야 함을 의미한다. 참고로 Jung and Mathur[10]는 단일물류기지를 전체로 하고(M = 1), 클러스터 및 순열  $Seq_1^1 \dots Seq_{K_1}^1$ 은 다른 휴리스틱에 의하여 구하였다. 그리고 주어진 클러스터 및 순열을 전체로 목적항 가장 안쪽의 재고보충주기만을 최적해법으로 구하여, 위의 수리모형이 의미하는 동시적인 방법이 아닌 순차적인 방법을 취하였다.

### 3. 유전알고리즘의 설계

유전알고리즘의 유전자에는 각 물류기지에 배정된 소매점 집합, 클러스터, 순열이 반영되어 있어야 한다. 이들 유전자는 후술하는 평가알고리즘에 의해 적합도가 평가되어 배제, 선택된다. 평가알고리

즘은 위에서 언급된 수리모형의 맨 안쪽의 해를 구하는 것에 해당되며, 최종해법은 Jung and Mathur [10]의 해법을 이용할 수 있다. 그러나 이 해법은 해를 도출하기 위한 전산소요시간이  $O(n \log(n))$ 으로 선형이 아니어서, 유전알고리즘 전체의 전산소요시간이 문제가 될 수 있다. 우리는 선형의 전산시간을 가진 다른 평가해법을 제 3.4절에서 고안하였다.

#### 3.1 유전자 표현

이 문제를 위한 유전알고리즘에서 유전자 표현은 <표 1>과 같은 세 개의 염색체(chromosomes)로 이루어진다. 3개의 염색체로서 물류기지로의 할당, 클러스터링의 형성, 클러스터 내에서의 순열이 결정된다. <표 1>에서 보이듯이, 염색체 1로부터 물류기지 2로 소매점 1, 5, 7이 배정되었음을 알 수 있다. 여기서 소매점은 유전인자 순서로 표시되며, 유전인자 순서는 유전자는 아니다. 클러스터 구성은 염색체 2가 주는 단일 유전인자인 숫자(이하  $M_d$ )를 각 소매점들의 연간 수요량 합이 넘지 않는 범위 내에서 최다의 소매점들을 클러스터로 묶는 것으로 이루어진다. 한도를 초과하면 새로 클러스터를 시작한다. 묶는 순서는 염색체 3의 순서를 따른다. 예컨대, 물류기지 2로 할당된 소매점들은 3번째 염색체에서의 순서가 (1, 7, 5)순이므로, 각 소매점들이 모두 1의 연간수요량을 가졌고  $M_d = 2$  라면, 1과 7가 하나의 클러스터로, 5은 다른 하나의 클러스터로 묶여진다. 그리고 염색체 3의 순서에 따라 1번째 클러스터 내에서는 1, 7의 순서가 클러스터내의 순열이다.

<표 1> 유전자 표현 : 염색체 1(물류기지 할당), 염색체 2(클러스터 크기), 염색체 3(소매점들의 순열표시)

유전인자 순서	1	2	3	4	5	6	7
염색체 1	2	3	1	3	2	1	2
염색체 2	2						
염색체 3	2	1	4	3	7	5	6

### 3.2 초기해 생성/개선

최초의 모집단(population)는 지역해로의 조기수렴을 방지하기 위해 임의의 생성과 발견적 기법을 혼용하여, 다음 <그림 2>과 같이 생성하였다. 주어진 확률 ( $p_d, p$ )로 염색체 1, 3을 발견적 기법을 적용하여 형성하였다. 염색체 1, 3의 나머지는 완전히 임의(random)로 형성한다. 2번째 염색체는 전부 임의로 형성하였다. 단 그 값의 범위는 사용되는 차량크기 Q에 따르도록 하였다. 전산실험에서 모집단은  $\delta$ 개의 유전자로 형성하였다. 또한 생성

- 1 단계 : 모든 소매점들을  $p_d$ 의 확률로 가장 가까운 물류기지로 배정하고,  $1-p_d$ 의 확률로 임의로(random) 배정한다.
  - 2 단계 :  $p$ 의 확률로 각 물류기지 m에 배정된 소매점들을 대상으로 각 물류기지에서부터의 래디안 각을 중심으로 시계방향으로 배열하여 배열을 구성한다. 전체 배열은 각 물류기지에 속한 소매점들의 배열 순서를 유지하면서, 염색체 3에서의 전체 순열에서의 순서는 완전히 임의로 될 수 있게 물류기지별 배열을 섞어 붙여서 구성한다.  $1-p$ 로 확률로 전체 배열은 완전히 임의로 구성된다.
  - 3 단계 : 2번째 염색체의 단일 유전인자의 값은 Q의 (0.25~4)배의 일양분포를 따르도록 형성하였다.

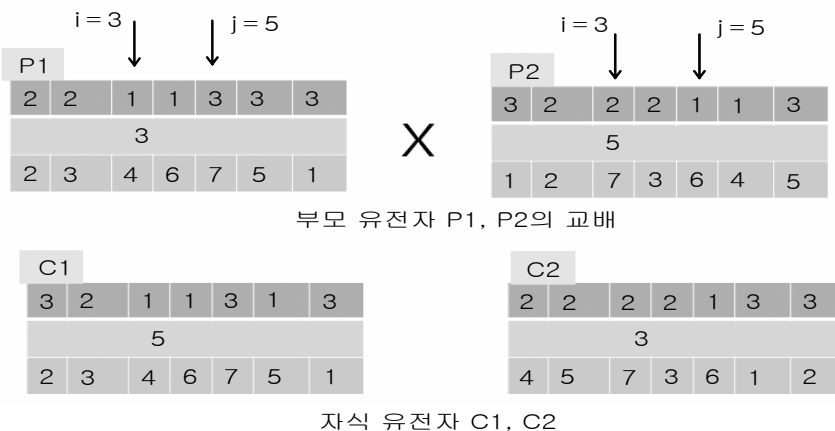
<그림 2> 초기해 생성을 위한 알고리즘

되는 유전자들은 해가 기존 모집단의 모든 유전자와  $\sigma$ 값 이상의 차이가 있어야 채택되도록 하여 모집단 초기해의 다양성을 확보, 해의 조기 수렴을 방지하였다.

### 3.3 모집단 교체를 위한 연산

#### 3.3.1 선택 및 교배(Selection and Crossover)

모집단의 교체를 위한 1번째 유전연산자의 선택은 엘리트 보존 전략(elite strategy)을 택하였다. 즉 모집단 평가값 상위 50% 내에서 임의로 두 개의 부모 유전자를 선택하였다. 두 번째 유전 연산자인 교배는 여러 연산자 중에서 order crossover(OX) 또는 linear order crossover에 가까운 방식을 택하였다. 염색체별로 교배의 방법은 다르다. <그림 3>에서 보이듯이, 염색체 1은 임의의 두 점 사이에서 부모유전자 P1, P2의 해당염색체가 자식 유전자 C1, C2의 해당 염색체에 교차하여 교배가 이루어지는 단순 order crossover이다. 예컨대 임의의 두 점, <그림 3>의 사례에서  $i = 3, j = 5$ 을 정하여 자식 유전자 C1은 교차되는 두 점 사이는 P1의 유전인자를 그대로 가져오고, 나머지 부분은 다른 부모 유전자 P2의 유전인자를 그대로 이어받는다. C2도 동일한 방식으로 교차되어 유전인자를 상속한다.



<그림 3> 교배 연산자의 작용 예

클러스터링의 크기를 결정하는 염색체 2는 단일유전자인 숫자( $M_j$ )로서 1번째 자식 C1의 해당 염색체는 2번째 부모 P2로부터, 2번째 자식 C2의 해당 염색체는 1번째 부모 P1로부터 숫자를 그대로 이어 받는다.

염색체 3은 다소 복잡한 교배구조를 가지고 있다. 염색체 3에서 교차되면서 단절되는 지점은 임의로 선택된다. 결과로서 이루어지는 두 개의 자식 유전자 C1, C2의 3번째 염색체는 다음과 같이 이루어진다. 교차되는 두 지점 사이(전과 같이 3에서 5사이)의 자식 유전자 C1의 해당 염색체는 부모 유전자 P1의 해당 염색체 유전인자가 그대로 복사된다([467]). 나머지 위치의 자식 유전자 C1의 해당 염색체는 부모 유전자 P2의 해당 염색체  $i-1$ 번째(즉 2번째) 유전인자로부터 우측으로 읽어서 복사하여-즉 2, 7, 3순으로-1번째 유전인자부터 채워나간다. 단 이 때, P2 유전자의 유전인자가 C1의 교차되는 두 지점 사이의 유전인자 즉 [467]과 중복되면 건너뛴다. 자식 유전자 C2의 해당 염색체는 부모 유전자 P1, P2의 역할을 바꾸어 같은 방식으로 복사된다. 최종적으로 2개의 자식 중 하나만 임의로 선택되어 모집단에서 열위 50% 중 임의의 하나를 대체한다.

### 3.3.2 돌연변이(Mutation)

조기해 수렴을 방지하기 위한 돌연변이는 <그림 4>와 같이 실행된다. 단 본 연구의 유전알고리즘의 적용에서 돌연변이 알고리즘에 의해 3번째 염색체만이 변형되어 유전되며, 돌연변이의 적용은 주어진 확률(pp)로 적용된다. 돌연변이 알고리즘에서 문제되는 것은 운송 차량경로비용을 계산하기 위해 반복적으로 순회외판원문제의 해를 얻어야 한다는 점이다.  $k$ 가 하나의 클러스터 내에서의 소매점들의 총수라면, 최소삽입비용 휴리스틱을 이용하는 경우, 전산소요시간은  $O(k^2)$ 이다. 그런데,  $k$ 는 차량용량에 의해 제한되므로 상수에 근사하게 취급될 수 있다. 따라서 돌연변이 알고리즘의 전산 처리 속도는  $O(n)$ 에 근사하여 무리가 없다.

1단계 : 유전자에 의해 결정되어지는 가장 낮은 순서의 클러스터( $c$ )를 선택하여 가장 낮은 순서의 소매점( $r$ )을 골라 다른 클러스터( $c'$ )의 모든 소매점 앞에 옮겨보아 두 클러스터에서 다음과 같이 표현되는 연관비용의 가장 큰 감소를 보이는 클러스터  $c'$ 을 선택하여 그 지점으로 소매점  $r$ 을 옮긴다.

$$t_r^{c'} = MIN \left[ \sqrt{K(R_c \cup r) / \sum_{j \in R_c} (d_j/2)h_j^R + (d_r/2)h_r^R}, \right. \\ \left. Q / \sum_{j \in R_c} d_j + d_r \right] \\ OBJ_r^{c'} = t_r^{c'} \left( \sum_{j \in R_c} (d_j/2)h_j^R + (d_r/2)h_r^R \right) \\ + K(R_c \cup r) / t_r^{c'}$$

만약 감소가 없으면 다른 클러스터를 고려하여 가장 큰 연관비용의 감소를 보이는 지점으로 옮긴다. 단 다른 클러스터( $c'$ )는 주어진 상수  $T_R$ 에서  $c' - c < T_R$ 와 같은 관계를 유지해야 한다. 이런 관계를 만족하는 모든 다른 클러스터가  $c'$ 로 고려되었음에도 연관비용 감소가 없으면 하나의 소매점  $r$ 로 이루어지는 새 클러스터를 만들어 연관비용 감소를 체크한다. 연관비용 감소가 있으면 해당되는 클러스터로 옮긴다.

2단계 : 클러스터  $c$ 에서 아직 고려되지 않은 다른 소매점  $r$ 을 선택하여 1단계를 반복한다. 다른 고려되지 않은 소매점이 클러스터  $c$ 에 없으면 3단계로 간다.

3단계 : 다른 클러스터  $c$ 를 골라 1, 2단계를 반복한다.

4단계 : 1~3단계의 과정을 모든 클러스터의 소매점 움직임이 고려될 때까지 반복한다.

5단계 : 다음 절차를 각 클러스터에 대해 적용한다.

5-1단계 : 클러스터안의 소매점 중에서 가장 낮은 비율  $K(\{j\}) / [(d_j/2)h_j^R]$ 을 가진  $j^*$ 를 선택하여 클러스터 배열에서 첫 번째 소매점으로 선택한다. 그리고 그 배열을  $P_i = j^*$ 로 한다.

5-2단계 : 클러스터 내에서 아직 선택되지 않은 소매점을 골라 최소의 비율  $[K(P_i \cup \{k\}) - K(P_i)] / [(d_k/2)h_k^R]$ 을 가진  $k$ 를 골라 현재 배열의 맨 끝에 위치시키고 그 배열을  $P_i = P_i \cup \{k\}$ 로 한다. 클러스터 안의 모든 소매점들이 배열되면 중단 한다. 그렇지 않으면 단계 5-2의 과정을 반복한다.

<그림 4> 돌연변이 알고리즘



### 3.4 적합도 평가

유전알고리즘은 반복적인 교배와 평가가 필수적이므로 개별 유전자의 적합도를 평가하는 알고리즘의 전산소요시간이 짧아야 한다. 본 연구의 문제는 복잡도가 증첩되어 있는 문제로서 기존의 전산소요시간이 긴 알고리즘을 적용하지 않고, 선형의 전산소요시간을 가진 다음의 알고리즘을 제시하였다. 이 알고리즘은 유사한 형태가 Viswanathan and Mathur[17]에 의해 제시되었으나, 변형되어 이용되었다. 또한 1번째 염색체의 물류기지가 동일한 소매점들을 대상으로 각각 알고리즘이 적용된다. 따라서 사용되는 물류기지의 개수 M만큼 평가알고리즘은 반복되어 시행된다.

아래 알고리즘을 간략히 설명하면, 3번째 염색체의 순열 순서에 따라 선택된 소매점들은 유전자에 정해진 클러스터 내에서 가장 바깥쪽에 위치하되 같은 재고보충주기를 가진 소매점들의 집합에 포함된다(만약 이 소매점이 새로 시작되는 클러스터의 1번째 소매점이라면 다음 소매점으로 넘어감). 다음으로 이 집합에 속하는 모든 소매점들에 공통적으로 단순 경제적 주문공식(EOQ)를 적용하여 재고보충주기를 얻는다. 이 재고보충주기가 바로 안쪽의 동일 재고보충주기를 가진 집합의 재고보충주기보다 크지를 검사하여 크면, 중단하고 다른 소매점을 골라서 위의 과정을 반복한다. 만약 작으면 안쪽의 집합과 현재의 집합을 합하고 새로운 집합의 공통적인 재고보충주기를 계산한다. 이후 또 다른 안쪽의 동일재고보충주기를 가진 소매점 집합과 재고보충주기를 검사하여 위의 과정을 반복한다. 평가알고리즘은 각 물류기지 m에 대해 반복적으로 시행되어 현재 유전자의 평가값은 이들의 합으로 계산된다.

<변수 정의>  
 $seq_m$  : 염색체 3에서 염색체 1을 참조, 물류기지  $m(\in M)$ 에 배정된 소매점들만 추출한 부분 순열  
 $seq$  : 염색체의 3의 순열

$R_c$  : 유전자에 의해 결정된 c번째 클러스터의 소매점 집합

$K(P)$  : 특정 집합 P에 포함된 모든 소매점들의 주문비용 및 순회 수송비용의 총합

$H_r$  : 소매점 r에서의 1단위 제품보유의 단위 시간당 재고보유비용

$P_c(l_c)$  : 클러스터 c의  $l_c$ 번째 증첩순열집합. 즉 하나의 클러스터내에서 동일한 재고보충주기를 가지는 소매점들의  $l_c$ 번째 집합. 예컨대, 같은 차량에 의해 재고 보충되는 2번째 클러스터가  $R_2 = \{3, 4, 5, 6\}$ 이고 이들이 2, 2, 4, 8의 재고보충주기를 가진다면,

$$P_2(3) = \{6\}$$

<평가 알고리즘>

단계 0 :  $n = 1, c = 0, R_c = \emptyset, l_c = 0$

단계 1 :  $r = seq_m$ 의 n번째 소매점 r이 유전자에 의해 결정된 새로 시작되는 클러스터링의 1번째 소매점이면  $c = c+1$ 로 하고 다음을 계산한다.

$$s_r = K(r)$$

$$t_r = B[\sqrt{s_r/2B^2H_r}]_2$$

소매점 r이 첫 번째 소매점이 아니면 다음을 계산한다.

$$s_r = K(R_c \cup r) - K(R_c)$$

$$t_r = B[\sqrt{s_r/2B^2H_r}]_2$$

단계 2 :  $k_c(l_c) = s_r, l_c = l_c + 1$ 로 하고 단계 3으로 간다.

단계 3 : 단계 3을  $M_c(l_c) > M_c(l_c - 1)$ 일 때까지 다음을 되풀이 한다.

$$k_c(l_c) = k_c(l_c - 1) + k_c(l_c),$$

$$P_c(l_c) = P_c(l_c - 1) \cup P_c(l_c), l_c = l_c - 1$$

$$M_c(l_c) = B[\sqrt{k_c(l_c)/2B^2 \sum_{j \in P_c(l_c)} H_j}]_2$$

단계 4 : 모든 소매점  $j \in P_c(l_c)$ 에 대하여,

$$t_j = M_c(l_c) \text{로 놓는다}$$

단계 5 :  $q_c = \sum_{j \in R_c} d_j t_j$  를 계산하고 만약  $q_c > Q$

이면, 다음 단계 6으로 진행한다. 그렇지 않으면 단계 7로 간다.

단계 6 : 만약  $l_c = 1$ 이면,

$$M_c(l_c) = B \left[ Q / \left( B \sum_{j \in R_c} d_j \right) \right]_2$$

그렇지 않으면  $M_c(L_c) = M_c(L_c)/2$ 로 하고, 이 경우 만약

$M_c(L_c) = M_c(L_c - 1)$  이면  
 $k_c(l_c) = k_c(l_c - 1) + k_c(l_c)$ ,  
 $P_c(l_c) = P_c(l_c - 1) \cup P_c(l_c)$ ,  $l_c = l_c - 1$ ,  
 로 한다. 그리고 모든 소매점  $j \in P_c(l_c)$ 에  
 대하여,  $t_j = M_c(l_c)$ 하고 단계 7으로 간다.  
 단계 7: 유전자에 의해 결정된 배열  $seq_m$ 의 모든  
 소매점이 고려되었으면 현재 유전자의  
 평가값을 다음과 같이 계산하고,  
 알고리즘을 끝낸다. 그렇지 않으면,  
 $n = n+1$ 로 하고, 단계 1로 간다.  

$$cost(m) = MIN_{T_m} [ \sum_{c \leq C} \sum_{l_c \leq L_c} [(M_c(l_c) \sum_{j \in P_c(l_c)} a_{cj}^R)$$
  

$$+ k_c(l_c) / M_c(l_c)]$$
  

$$+ \sum_{c \leq C} \sum_{l_c \leq L_c} a_{cj}^m (T_m \vee M_c(l_c)) + C_m / T_m ]$$

〈그림 5〉 적합도 평가를 위한 해의 생성 알고리즘

적합도 평가 알고리즘 단계 7의 최소화문제에서  
 물류기지  $m$ 의 재고보충주기  $T_m$ 은 최소의 기본 주문  
 간격(B)의 반복된 2배수를 지속적으로 시도하여 가  
 장 작은 비용  $cost(m)$ 을 주는 값을 선택하면 된다. 이  
 알고리즘은 모든 소매점 각각에 대하여  $K(R_c \cup \{r\})$   
 를 구하기 위해 외판원순회문제를 풀어야 하므로, 전  
 산소요시간의 대부분이 여기에 소모된다. 그러나 전  
 술한 바와 같이 외판원순회문제의 크기는 차량의  
 크기에 의존하므로 전체 전산소요시간은 선형에 근  
 사하게 제한된다.

### 3.5 전체 유전알고리즘

전체의 유전알고리즘은 다음과 같은 가코드(pse-  
 udo code)로 기술된다.

```

k = 0; iter = 0;
repeat
   $\pi_{k+1}$  = initialize ( )
  // 제 3.2절의 초기해 생성 알고리즘에
  // 의해 k+1번째 유전자  $\pi_{k+1}$  생성;
  if (spaced( $\pi_1 \dots \pi_{k+1}$ )) then k = k+1;
  // 평가 알고리즘을 적용한 모든
  // 유전자들의 해 차이가
  shift  $\pi_{k+1}$ 
  // 일정수준  $\sigma$  이상 되면 초기해로
  // 채택, 모집단 유전자들을
  
```

```

end if // 해가 작은 순으로 정렬
until k ==  $\delta$  // 모집단의 크기는  $\delta$ 
repeat
  select P1, P2 in { $\pi_1 \dots \pi_{\lfloor \delta/2 \rfloor}$ }
  // 해가 상위 50% 안에 있는 유전자
  // 중에 임의로 두 유전자 선택
  crossover // 제 3.3.1절 기술
  iter = iter+1;
  select k at random in [ $\delta/2$ ,  $\delta$ ]
  if random < pp then
    C = M;
    // M은 crossover 자식유전자 둘 중
    // 하나를 임의로 골라 돌연변이시킨
    // 결과(제 3.3.2절 기술)
  end if // C는 crossover의 결과로 나오는
  // 2개의 자식 유전자중 임의의 하나
  if (spaced( $\pi \setminus \{\pi_k\} \cup \{C\}$ )) then
    // 자식유전자와 모집단의 다른 유전
    // 자들과의 해 차이가  $\sigma$  이상 되면
     $\beta = \beta + 1$ ; //  $\pi_k$ 를 C로 대체하고
     $\pi_k = C$ ;
    shift  $\pi_k$  // 모집단 재정렬
  end if
  if (F(C) < F( $\pi_1$ )) then
    // F(C)는 자식 유전자 C의 해
     $\alpha = 0$ ; // 해의 개선이 있으면  $\alpha = 0$ 
  else
     $\alpha = \alpha + 1$ 
  end if
until  $\beta \leq \beta_{max}$ , or  $\alpha \leq \alpha_{max}$  or iter  $\leq$  itermax
  
```

〈그림 6〉 전체 유전알고리즘의 기술

## 4. 전산 실험

본 논문에서 제시된 알고리즘-이하에서 GEN으  
 로 표기-을 기존에 잘 알려진 동일 문제의 해법을  
 주는 알고리즘들과 비교하여 해의 품질과 계산소  
 요시간이 더 나은지를 비교하면 GEN의 성능을 평  
 가할 수 있을 것이다. 그러나 2의 먹수배 정책하에  
 서 다수 물류기지에서의 재고보유를 전제로 한 재  
 고 및 경로 문제의 해법을 제시하는 연구는 찾을  
 수 없다. 그러므로 우리는 단일 물류기지에서의 재  
 고보유를 전제로 하면서, 2의 먹수배 정책 하에서  
 의 최적해를 찾는 Jung and Mathur[10]의 알고리

증(아래에서 JM로 표기)을 다수 물류기지를 가진 경우로 확대하여, 비교를 시도하였다. 물론 아주 작은 수의 소매점들을 대상으로 최적해를 구하여 비교할 수도 있겠지만, 15개의 소매점을 대상으로 하는 단일 물류기지에서의 재고 및 경로 문제도 최적해 도출에 수 시간의 전산소요시간이 필요하여 (전개서 참조) 현실적으로 불가능하다.

비교를 위한 확대된 JM 알고리즘은 이하와 같이 설명된다. 먼저 JM 알고리즘은 단일한 물류기지를 전제로 하기 때문에 JM 알고리즘을 적용하기 전에 우선, 각 소매점들의 물류기지로의 배정이 가장 가까운 물류기지로 이루어졌다. 다음으로 JM 알고리즘은 특정 물류기지에 배정된 소매점들을 대상으로, 물류기지에서부터의 래디안(radian) 각의 순으로 소매점들을 정렬하여,  $M_i$ 의 수치-다수의 값을 시도하여 가장 좋은 해를 주는 것을 선택-를 포함한 소매점들의 수요합이 넘지 않는 한도 내에서 가장 크게 클러스터들을 형성한다. 다음 제 3.3절의 돌연변이 알고리즘을  $T_R$ 변수 크기를 무한대로 하여 적용한다. 그렇게 형성된 각 물류기지별 클러스터와 중첩순열을 전제로 물류기지 및 해당 물류기지에 배정된 소매점들의 재고보충주기를 구하는 문제(제 2.2절의 수리모형 참조)를 최적 알고리즘으로 푼다.

해의 비교를 위해 우리는 문제의 생성을 임의로 하여 비교하였다. 소매점 및 각 물류기지들의 위치는 원점으로부터 300의 반지름 내에 임의로 위치

하게 하였다. 비교를 위한 파라메터들을 아래 <표 2>와 같이 설정하여 총 8개 Case의 문제들을 생성하였다. 각 소매점들의 재고비용( $h_j^r, h_j^m$ ), 차량의 용량( $Q$ ) 및 기본 주기(B), 소매점들의 숫자(n), 물류기지에서 차량 출발시마다 발생하는 고정비용( $v_f$ )은 유형별 비교를 위해, 각 케이스에 따라 다르게 하였다. 각 물류기지의 주문비용( $C_m$ )은 500에서 1500사이의 일양분포(uniform distribution)와 50에서 150사이의 값을 가진 두 가지 범주를 비교하였다. 소매점별로 일정 비율로 발생하는 수요량 또한 두 가지 범주로, 모든 소매점들이 동일하게 단위 시간 동안 1의 수요량을 가진 케이스와 각 소매점별로 1에서 10까지 일양분포의 수요량을 가진 케이스를 다루었다.

유전알고리즘은 삼성 매직스테이션 2.99Hz를 사용하고 C 언어로 코딩되어 실행되었다. 주요 변수들의 세팅은  $\alpha = 2000, \beta = 4000, \text{iter max} = 6000$ 으로 놓고,  $pp = 0.2, p = 0.3, p_d = 0.5$ 로 놓고 해를 얻었다.

전체적으로 우리는 해의 비교를 위해 각 케이스별로 5개의 문제들을 생성하였으며, 그 평균 결과-유전알고리즘과 JM 알고리즘 해의 평균 비율 및 유전알고리즘의 평균 전산소요시간(초). 괄호안은 JM 알고리즘의 전산소요시간-를 <표 3>으로 요약하였다.

결론적으로 유전알고리즘(GEN)은 기준에 알려진 알고리즘의 적용에 비해, 실험에 포함된 모든

<표 2> 문제 Case별 변수 세팅

	case 1	case 2	case 3	case 4	case 5	case 6	case 7	case 8
n	150	150	150	150	150	150	150	300
$C_m$	U(500~1500)	U(50~150)	U(500~1500)	U(500~1500)	U(500~1500)	U(500~1500)	U(500~1500)	U(500~1500)
d	1	1	1	1	1	1	U(1~10)	U(1~10)
$h_j^r, h_j^0$	150, 50	150, 50	60, 50	150, 50	150, 50	150, 50	150, 50	150, 50
Q	2	2	2	2	2	0.8	8	8
$v_f$	1	1	1	250	1	1	1	1
B	0.125	0.125	0.125	0.125	0.5	0.125	0.2	0.2

〈표 3〉 물류기지 수(M) 및 각 문제 Case별 알고리즘 수행 결과 비교

	M = 1		M = 3		M = 5	
	GEN/JM	전산소요 시간(초)	GEN/JM	전산소요 시간(초)	GEN/JM	전산소요 시간(초)
Case 1	0.86	100(4)	0.85	289(5)	0.82	465(5)
Case 2	0.85	152(5)	0.86	441(5)	0.84	536(6)
Case 3	0.79	68(4)	0.79	347(4)	0.75	484(6)
Case 4	0.81	168(4)	0.81	456(4)	0.77	660(4)
Case 5	0.94	129(3)	0.94	486(5)	0.96	399(5)
Case 6	0.85	100(5)	0.87	411(5)	0.86	482(5)
Case 7	0.97	100(11)	1.00	401(18)	0.98	555(22)
Case 8	0.94	576(30)	0.94	543(37)	0.99	710(37)

문제 케이스에서 비교되는 알고리즘(JM)보다 좋은 결과를 가져온다. 케이스에 따라서는 최대 25%의 큰 폭의 개선효과를 가져온다. 내용을 본다면, 우선 물류기지의 수를 1, 3, 5개의 경우로 한 각각의 경우를 보았을 때, 대체로 물류기지 수가 많아지는 경우 유전알고리즘의 개선효과가 커지는 것을 볼 수 있다. 물류기지 수가 많아지면 물류기지에서 소매점까지의 거리가 작아지므로, 재고비용에 비해 수송비용이 비중이 작아진다(즉 재고모형에서 주문비용이 준다). 그러므로 더 자주 재고보충을 하는 것이 최적에 가까워져 재고보충시간이 짧아진다. 이에 따라 수송용량의 제약이 있는 차량이 좀 더 많은 소매점들을 재고 보충할 수 있게 된다. 즉 동일차량이 재고보충을 담당하는 소매점들의 집합인 클러스터 크기가 커지게 된다. 따라서 클러스터 내의 순열의 다양성이 커지므로 유전알고리즘의 개선효과가 더 커지는 것으로 해석될 수 있다.

앞의 경우와 유사한 이유로,  $Q$ 가 발생하는 수요 크기에 비해 작아지는 경우(문제 케이스 6은 케이스 1에 비해  $Q$ 만 감소) 유전알고리즘의 개선효과가 작아지는 것을 볼 수 있다. 또 케이스 7, 8의 경우에도 수요크기에 비해 차량크기가 케이스 1~5에 비해 작다. 이 경우에도 유전알고리즘에 의한 개선효과가 케이스 1~5에 비해 크게 떨어지는 것을 알 수 있다. 하나의 클러스터에 포함되는 소매점수가 차량 크기에 의해 제약받는 경우에 작아지

고, 따라서 하나의 클러스터 내에서 순열의 다양성에 의한 해의 개선이 유전알고리즘 내에서 이루어지기 어려운 까닭으로 해석된다. 또한 케이스 4의 경우는 케이스 1의 경우에 비해 차량 1회 순회 시 발생하는 고정비용만이 큰 경우이다. 이때는 고정비용이 크기 때문에 하나의 클러스터 내에 포함되는 소매점 수가 많게 되어서, 유전알고리즘의 해가 더 좋아지는 경우로 해석할 수 있다.

한편 케이스 3의 경우, 제품 1단위당 재고비용이 줄어, 재고비용보다 셋업비용을 이루는 차량순회비용(수송비용)의 비중이 커지는 경우이다. 이때는 보다 더 긴 재고보충주기를 가지는 것이 최적에 가까워진다. 케이스 3이 케이스 1과 동일한 기본주기(B)를 가지므로, 긴 보충주기를 동일한 기본주기의 2의 멱수배로서 반영하게 된다. 따라서 해가 가지는 정밀도가 좀 더 높아질 수 있으므로, 유전알고리즘이 더욱 더 좋은 개선해를 제시한다고 볼 수 있다. 이와 반대의 경우는 케이스 5의 경우로서, 케이스 1에 비해 기본주기가 커지는 경우, 유전알고리즘의 개선효과가 감소한다.

한편 물류기지에서의 주문비용이 커지는 경우(케이스 2)는 유전알고리즘에 미치는 영향이 미미한 것으로 나온다. 또한 소매점 수가 늘어나는 경우 개선효과는 더욱 더 좋아지는 것을 알 수 있다. 케이스 8의 경우, 케이스 7의 소매점수가 2배로 된 경우이며, 케이스 7의 개선효과가 훨씬 커지는 것

을 볼 수 있다. 이는 역시 하나의 클러스터 내에 포함되는 순열의 다양성이 커져서 유전알고리즘의 개선 폭이 더욱 커지는 것으로 볼 수 있다.

<표 3>에서 보듯이 유전알고리즘은 JM 알고리즘과 비해서 훨씬 많은 전산시간을 소모하지만, 재고 및 경로 문제의 전략적 중요성을 감안해본다면 충분히 감당할 수 있는 수준이다. 표에서 보이듯이 소매점 수가 300개에 달하는 큰 문제도 대체로 10분 내외의 전산소요시간을 가진다.

## 5. 결 론

본 논문은 2의 멱수배 재고보충주기를 가정하고, 다수의 물류기지 및 소매점 모두에서 재고를 보유하고, 물류기지들에서 출발한 차량이 소매점들을 방문하여 재고를 보충할 때, 재고 및 수송비용의 합을 최소화하는 복합적인 의사결정문제를 다루고 있다. 이 문제 곧 다수 물류기지 재고 및 경로 문제에 우리는 유전알고리즘을 적용하여 기존의 알고리즘과 비교한 결과, 대부분의 경우에 큰 폭의 개선효과를 가져오는 것을 알 수 있었다. 이러한 재고 및 수송의 복합문제와 같은 전략적인 문제에 있어서는 해의 개선이 실제 현실에서 큰 폭의 비용 감소로 이어지므로, 유전알고리즘 적용의 가치는 충분히 있다.

본 논문이 다룬 모델의 확장도 향후 연구과제로 가치가 있을 것이다. 예컨대, 각 소매점들이 재고를 보충할 수 있는 시간대가 정해져 있는 모델 또는 소매점들에서 발생하는 수요가 확률변수인 모델 등은 보다 더 실용적인 가치를 지니지만, 이들 모델에 유전알고리즘을 적용하는 것은 어려운 연구과제로 남겨져 있다.

## 참 고 문 헌

- [1] 양병학, “연동계획과 확장된 기억 세포를 이용한 재고 및 경로문제의 복제선택해법”, 『경영과학』, 제26권, 제1호(2009), pp.171-182.
- [2] 정영훈, 김각규, 이상현, “역물류 환경을 고려한 복수차고지 다용량 차량경로문제”, 『경영과학』, 제29권, 제1호(2012), pp.89-99.
- [3] Anily, S. and A. Federgruen, "A Class of Euclidean Routing Problems with General Route Cost Functions," *Mathematics of Operations Research*, Vol.15(1990), pp.268-285.
- [4] Anily, S. and A. Federgruen, "Two Echelon Distribution System with Vehicle Routing Cost and Central Warehouse," *Operations Research*, Vol.41(1993), pp.37-47.
- [5] Baker, B.M. and M.A. Ayechev, "A Genetic Algorithm for the Vehicle Routing Problem," *Computers and Operations research*, Vol.30(2003), pp.787-800.
- [6] Burns, L.D., R.W. Hall, and D.E. Blumenfeld, "Distribution Strategy that Minimize Transportation and Inventory Costs," *Operations Research*, Vol.33(1985), pp.469-490.
- [7] Daganzo, C.F. and G.F. Newell, "Configuration of Physical Distribution Networks," *Networks*, Vol.16(1986), pp.113-132.
- [8] Gallego, G. and D. Simchi-Levi, "On the Effectiveness of Direct Shipping Strategy for the One-Warehouse Multi-Retailer-Systems," *Management Science*, Vol.36(1990), pp.240-243.
- [9] Herer, Y.R. and G. Roundy, "Heuristic for a One-Warehouse Multi-Retailer Distribution Problem with Performance Bound," *Operations Research*, Vol.45(1997), pp.102-115.
- [10] Jung, J. and K. Mathur, "An Efficient Heuristic Algorithm for a two-echelon Joint Inventory and Routing Problem," *Transportation Science*, Vol.41(2007), pp.55-73.
- [11] Prins, C., "A Simple and Effective Evolutionary Algorithm for the Vehicle Routing Problem," *Computers and Operations Research*, Vol.31(2004), pp.1985-2002.

- [12] Chan, L.M.A., A. Federgruen, and D. Simchi-Levi, "Probabilistic analyses and practical algorithms for inventory-routing models," *Operations Research*, Vol.46, No.1(1998), pp.96-106.
- [13] Chan, L.M.A. and D. Simchi-Levi, "Probabilistic Analyses and Algorithms for Three Level Distribution Systems," *Management Science*, Vol.41(1998), pp.1562-1576.
- [14] Zhao, Q.-H., S.-Y. Wang, and K.K. Lai, "A Partition Approach to the Inventory/Routing Problem," *European Journal of Operational research*, Vol.177(2007), pp.786-802.
- [15] Roundy, R., "98% Effective Integer-Ratio Lot-Sizing for One-Warehouse Multi-Retailer Systems," *Management Science*, Vol.31(1985), pp.1416-1429.
- [16] Schwarz, L.B., "Simple Continuous Review Deterministic One Warehouse N Retailer Inventory Problem," *Management Science*, Vol.19(1973), pp.555-566.
- [17] Viswanathan, S. and K. Mathur, "Integrating Routing and Inventory Decisions in One-Warehouse Multi-retailer Multi-product Distribution Systems," *Management Science*, Vol.43(1997), pp.294-312.