Journal of the Society of Korea Industrial and Systems Engineering
Vol, 30, No. 3, pp.44－53, September 2007.

중앙창고의 수와 위치 결정을 위한 거리 기반 Simulated Annealing 앨고리듬

이동주† · 김진호

공주대학교 산업시스템공학과

A Distance-Based Simulated Annealing Algorithm for the determination of the Number and the Location of Centralized Warehouses

Dongju Lee† · Jinho Kim

Department of Industrial & Systems Engineering

Forming central warehouses for a number of stores can save costs in the continuous review inventory model due to economy of scale and information sharing. In this paper, transportation costs are included in this inventory model. Hence, the tradeoff between inventory-related costs and transportation costs is required. The main concern of this paper is to determine the number and location of central warehouses. Transportation costs are dependent on the distance from several central warehouses to each store. Hence, we develop an efficient simulated annealing algorithm using distance-based local search heuristic and merging heuristic to determine the location and the number of central warehouses. The objective of this paper is to minimize total costs such as holding, setup, penalty, and transportation costs. The performance of the proposed approach is tested by using some computational experiments.

Keywords : Simulated Annealing, Inventory Model, Supply Chain Management

1. 서 론

기업 내의 최적화 뿐 아니라 기업 간의 협력과 조정을 통한 전체 최적화를 추구하는 공급사슬경영(Supply Chain Management)이 오늘날 각광을 받고 있다(권재현 외(2004), 이승근 외(2004)). 이동주 외(2006)에서 제시한 본 논문의 문제도 스토어들(store)간의 협력을 통한 비용 최소화에 초점을 맞추고 있다.

즉, 각 스토어들이 (Q, r)재고모델에 따라 운영되는 개별창고를 보유하는 대신 주변에 큰 창고(중앙창고라 부르기로 하자)를 두고 이 중앙창고가 스토어들의 수요를 담당하는 경우 몇 개의 중앙창고를 어디에 두어야 할지 결정하고 이때의 비용은 얼마나 절감되는지 알아보는 문제이다.

(Q, r)재고 모델은 재고수준을 항상 알 수 있고 제품 수요가 불확실할 때, 재고가 재주문점인 r이하로 떨어지면, Q만큼 주문하는 것인데, 이때 비용을 최소화하도록 Q와 r을 결정하여야 한다.

중앙창고도 역시 (Q, r)재고모델을 따른다고 한다면 각 스토어가 개별 창고를 가질 때보다 중앙창고는 규모

† 교신저자 djlee@kongju.ac.kr

의 경제로 인한 안전재고의 감소로 비용절감이 발생하는 반면 중앙창고에서 스토어로 상품을 운송하므로 운송비가 유발된다. 이때 중앙창고가 (Q, r)재고모델에 의해 운영된다고 할 때 중앙창고의 운영비용과 운송비를 최소화하도록 중앙창고의 위치와 수를 결정하는 문제이다.

이 문제는 널리 알려진 설비위치결정문제(Facility Location Problem)과 동일한 제약식을 가진다. 설비 위치결정문제에서는 설비 혹은 창고를 특정위치에 두게 될 때 발생하는 고정비와 설비를 수요지와 연결하는 운송비가 목적식으로 쓰인다. 하지만 이 문제는 운송비는 동일하지만 고정비대신 (Q, r)재고모델의 창고운영비가 적용된다는 점에서 차별된다. 다음 절에서 살펴보겠지만 이러한 창고운영비는 고정비보다 훨씬 복잡하다.

이동주 외(2006)는 본 문제를 효율적으로 푸는 휴리스틱을 제안하고 10개의 스토어가 있는 경우에 대해 풀었다. 이 휴리스틱은 작은 크기의 문제에 적당한 휴리스틱으로 본 논문에서는 이들 보다 큰 크기의 문제에 적합한 시뮬레이티드 어닐링(Simulated Annealing)기법으로 100개의 스토어에 대해 풀었다. 또한, 거리를 이용한 휴리스틱을 개발하여 Simulated Annealing(SA)과 혼용하여 해를 구하고 일반적인 SA를 적용하여 구한 해와 계산시간과 해의 우수성 면에서 비교하여 보았다.

이전의 설비위치결정문제와 달리 최근에는 비선형의 목적식을 가지는 설비위치결정문제가 연구되고 있다. Holmberg(1999)는 운송비가 convex 함수인 경우의 설비위치결정문제를 Benders decomposition과 Dual ascent기법을 이용하여 최적해를 탐색하였다.

Hackness(2003)는 생산비가 convex 함수인 경우의 설비위치결정문제를 분지한계법(Branch and Bound)을 이용하여 해를 제시하였다. Wu et al.(2006)는 창고에 용량제약이 있고 여러 개의 창고를 한 지역에 세울 수 있을 때 창고 수에 따라 비선형적으로 증가하는 고정비가 있는 경우에 효과적인 라그랑쥐(Lagrangian Heuristic)휴리스틱을 제안하였다. 본 논문의 문제는 창고운영비가 선형이 아니라는 점에서는 위의 연구들과 유사하지만 반복적인 방법을 통해 주문량인 Q와 재주문점인 r을 구한 후 Q와 r을 이용하여 창고운영비를 계산한다는 점에서 이들 연구와 차별된다.

SA는 조합최적화 문제에 접근하는 여러 메타휴리스틱 기법 중 하나로 Kirkpatrick et al.(1983)에 의해 제안되어 폭 넓게 사용되고 있다. 우훈식(1997)은 순열 flow-shop에서 makespan을 최소화하는 문제를 SA를 이용해 풀어보았다. 또한, 김정자 외(1999)는 TSP(Travelling salesman problem)를 엔트로피 개념을 이용해 개선한 SA를 적용하여 좋은 해를 구했다. 이현남 외(1999)는 Set Covering 문제를 유전자 앨고리듬(Genetic Algorithm)의 교배기법과 국부 탐색기법을 이용한 SA를 적용하였다. SA는 이전에 방문한 해를 다시 방문하는 낭비를 할 수 있기에 Mishra(2005)는 제약이론(TOC)의 다중제약을 지닌 제품 혼합 생산문제에서 타부탐색(tabu search)의 타부목록(tabu list)과 열망함수(aspiration function)를 SA에 도입하여 우수한 해를 탐색하였다.

M'Hallah(2007)와 Yoo et al.(2007)는 일정관리 문제를 GA(Genetich Algorithm)과 SA기법을 혼용한 hybrid search를 제안하였다. Kaisa et. al(2006)은 국부 탐색기법인 a local proximal bundle method와 SA기법을 혼용하여 연속최적화문제(continuous optimization)를 효율적으로 풀었다(2006). Kaisa는 국부탐색기법을 쓸 때 정도(accuracy)를 SA 수행 초기에서 말기로 갈수록 높일 때 가장 좋은 해를 제공한다는 것을 실험을 통해 밝혔다. 본 논문에서는 다양한 휴리스틱과 SA를 혼용하고 휴리스틱의 혼용빈도와 휴리스틱의 적용순서가 결과에 어떤 영향을 미치는지를 실험을 통해 알아보았다.

이어지는 제 2장에서는 본 논문에서 사용되는 기호를 설명하고 문제를 정의하였다. 제 3장에서는 본 논문의 문제를 메타휴리스틱을 이용하여 풀어야 하는 이유를 설명하고 국부최적해를 찾는 휴리스틱들을 제안하였다. 제 4장에서는 SA를 설명하고, 제 5장에서는 실험을 통해 제안한 SA의 성능을 알아보았다. 마지막으로 제 6장에서는 결론을 제시하였다.

2. 기호와 문제정의

몇 개의 스토어들이 모여서 그 스토어들을 위한 중앙창고(central warehouse)를 이용할 수 있다고 할 때, 이러한 스토어들의 모임을 **연합(coalition)**이라고 부르기로 하자. 또한, 하나의 중앙창고와 그와 연결된 스토어들을 "연결그룹"이라고 하자. 즉, 연합은 스토어들의 모임이며 연결그룹은 연합과 연합을 담당하는 하나의 중앙창고로 구성된다. 엄밀하게는 이 둘을 구분할 수 있지만, 내용을 전개하는데 있어 이 둘을 구분하는 것은 무의미하므로 이후로는 이 둘을 구분치 않기로 한다.

본 논문에서 사용된 가정은 다음과 같다.

- 각 스토어의 창고와 각 연합들의 중앙창고는 동일한 주문비용(setup cost) A, 재고유지비용(holding cost) h, 유실판매 벌과비용(penalty cost) p를 가진다.
- 중앙창고의 위치는 미리 정해져 있지 않고, 기존의 스토어의 위치 중 연간평균총비용을 최소로 하는 곳으로 정한다.
- 각 스토어들과 연합들은 연속조사정책(continuous review policy) (Q, r)을 따른다.

또한, 사용된 기호를 아래와 같이 정리하였다.

N : 스토어 수

U : 전체 스토어들의 집합, $\{1, 2, 3, \cdots, N\}$

s : 임의의 연합 혹은 연결그룹

d_{ij} : i에 위치한 중앙창고로부터 스토어 j로의 운송거리, 단 $d_{ii} = 0$.

t : 하물당 km당 운송비

D_i : 스토어 i의 년간 기대수요.

D_s : 연합 s가 만드는 중앙창고의 년간 기대수요, $D_s = \sum_{i \in s} D_i$

μ_i, σ_i : 선행기간(lead time) 동안의 스토어 i의 수요의 평균과 표준편차

μ_s, σ_s : 선행기간(lead time) 동안의 연합 s의 수요의 평균과 표준편차

$$\mu_s = \sum_{i \in s} \mu_i \quad \sigma_s = \sqrt{\sum_{i \in s} \sigma_i^2}$$

X_i : 스토어 $i \in U$의 선행기간(lead time) 동안의 수요. 정규분포를 따르고 X_i의 확률밀도함수(pdf, probability density function)는 g_i, 누적분포함수(cdf, cumulative distribution function)는 G_i이며 각각의 X_i는 서로 독립이다.

X_s : 연합 s의 선행기간(lead time) 동안의 수요.

Q_s, r_s : 연합 s의 주문량과 재주문점

$R_s(r_s)$: 연합 s의 주기말의 기대부족수요

$$R_s(r_s) = E[(X_s - r_s)^+] = \int_{r_s}^{\infty} (X_s - R_s(r_s))\, g_s(x_s)\, dx_s \quad \cdots\cdots (1)$$

단, $(X_s\text{-}r_s)^+ = \max[0, X_s\text{-}r_s]$.

스토어들의 연합인 s가 정해진다면 (Q, r)재고모델에서는 운송비를 제외한 창고의 운영비용은 다음과 같다.

$$M(Q_s, r_s) = A\frac{D_s}{Q_s} + h\left(\frac{Q_s}{2} + r_s - \mu_s\right) + p\frac{D_s R_s(r_s)}{Q_s} \quad \cdots\cdots (2)$$

창고운영비인 식 (2)는 연간총주문비(setup cost), 연간총재고유지비(holding cost), 선행기간동안의 수요가 재고량을 초과할 경우 발생하는 연간총유실판매벌과비(penalty cost)의 합이다.

이 비용 $M(Q_s, r_s)$를 최소로 하는 Q_s^*, r_s^*를 구하는 방법은 재고 관련 책들에 설명이 되어 있기에 자세한 설명은 생략한다(Johnson and Montgomery(1974) 참조). $M(Q_s, r_s)$를 Q_s 와 r_s에 대해 각각 편미분하여 0으로 놓고 풀면 식 (3)과 식 (4)를 구할 수 있다.

$$Q_s^* = \sqrt{\frac{2D_s(A + pR_s(r_s^*))}{h}} \quad \cdots\cdots (3)$$

$$1 - G_s(r_s^*) = \frac{hQ_s^*}{pD_s} \quad \cdots\cdots (4)$$

이때 식 (3)과 식 (4)를 통해서 Q_s^*, r_s^*가 한번에 구해지는 것이 아니라 Q_s와 r_s가 수렴할 때까지 반복적으로 풀어야 Q_s^*, r_s^*를 구할 수 있다는 것에 주의해야 한다. 반복적인 계산이 이뤄져야 하기에 설비위치결정문제처럼 수식모형을 세워 혼합정수계획법으로 풀 수가 없다.

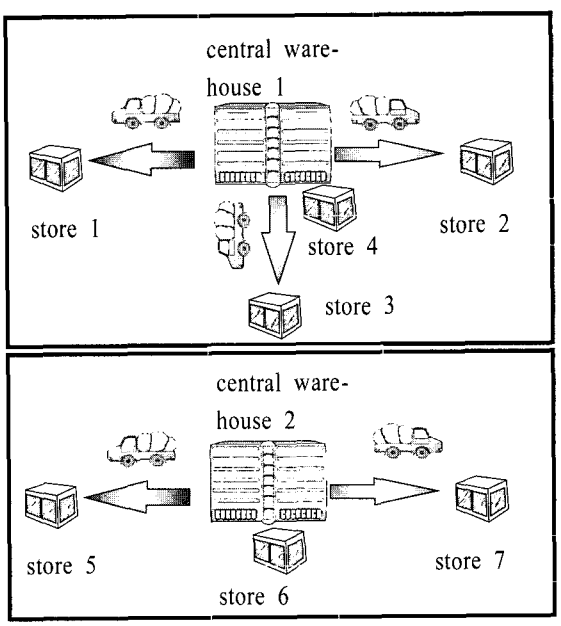
운송비를 포함한 연결그룹 s의 연간 평균 총비용은 식 (5)와 같다.

$$C(Q_s, r_s) = A\frac{D_s}{Q_s} + h\left(\frac{Q_s}{2} + r_s - \mu_s\right) + p\frac{D_s R_s(r_s)}{Q_s} + \min_i \sum_{j \in s} t d_{ij} D_j \quad \cdots\cdots (5)$$

어떤 스토어들이 연합 s를 구성할 것인지 정해진다면 Q_s^*와 r_s^*를 계산할 수 있기에 창고운영비를 최소화하는 $C(Q_s^*, r_s^*)$를 구할 수 있다. 연합 s에서 둘 수 있는 여러 중앙창고입지 중 운송비를 최소화하는 곳에 중앙창고를 두면 된다.

그러므로 각 스토어들이 하나의 연합에는 속해야 하므로 전체 비용(TC)은 이러한 연합들의 비용의 합이므로 식 (6)과 같다.

$$TC = \sum_{\bigcup s_i = U} C(Q_{s_i}^*, r_{s_i}^*) \quad \cdots\cdots (6)$$



<그림 1> 연결그룹 s_1(스토어 1, 2, 3, 4)과 s_2(스토어 5, 6, 7)에서 $Q_{s_1}^*$, $r_{s_1}^*$, $Q_{s_2}^*$, $r_{s_2}^*$의 결정

<그림 1>에 스토어 1, 2, 3, 4, 5, 6, 7이 두 개의 연결그룹들을 구성할 때 Q_s^*, r_s^*를 결정하는 경우를 보여주고 있다. 2개의 중앙창고가 존재하므로 2개의 각 중앙창고에 적절한 Q_s^*, r_s^*를 구하여야 한다.

3. 초기해와 제안된 휴리스틱

3.1 초기해 설정

본 연구에서는 각 스토어들이 개별 중앙창고를 가지는 것으로 설정하였다. 즉, 100개의 스토어가 있는 경우 각 스토어가 개별 창고를 가져 100개의 중앙창고가 있도록 초기해를 설정하였다. 이후의 실험에서 사용되는 모든 해법들은 본 절에서 소개한 대로 초기해를 구하였다.

3.2 제안된 휴리스틱

이동주(2006)외는 100개의 스토어와 5개의 중앙창고를 가지는 연결그룹들인 경우의 수는 약 10^{68}개가 있으며, 스토어의 수가 많아질수록 가능한 해의 대안 수는 기하급수적으로 증가함을 설명하였다. 이에 최적해를 구하는 방법보다는 짧은 시간 안에 근사해를 구하는 메타 휴리스틱 방법이 더욱 의미가 있다.

본 논문에서는 국부최적해 탐색기법을 제시하고 이를 메타휴리스틱 중 SA를 혼용하여 적용하였고 100개의 스토어들이 있는 예제들을 생성하여 제시된 SA의 성능을 검증하였다.

본 논문에서는 2개의 거리를 기반으로 한 국부 탐색 휴리스틱과 두 연결그룹끼리의 합병을 고려한 국부 탐색 휴리스틱을 소개한다.

3.2.1 거리 기반 Heuristic 1

거리(Euclidean distance)를 계산해서 특정 창고(W_k)에 대하여 창고 W_k에 배치되지 않은 스토어들 중 가까운 거리에 있는 것부터 먼 거리에 있는 것들을 정렬하여 가까운 것부터 W_k에 배치하여 전체비용(TC)이 감소하면 W_k가 그 스토어를 커버하고 아니면 기존의 창고가 커버하게 한다. 스토어를 재할당할 때 TC가 감소하지 않는 순간 그 창고에 대한 탐색은 그만 둔다. 다른 모든 창고도 같은 방법으로 고려한다. 즉, 거리를 근거로 한 휴리스틱이며 앨고리듬은 다음과 같다.

다음 절차를 더 이상 고려할 창고가 없을 때까지 반복한다.

1. (정렬) 중앙창고 W_k가 커버하지 않는 스토어들을 거리(Euclidean distance)를 계산해서 가까운 것부터 먼 것 순으로 정렬한다.
2. (재할당) 스토어를 창고 W_k로 할당시켜 TC의 감소여부를 파악한다.
   2.1 W_k가 커버하지 않는 스토어들 중 가까운 거리에 있는 스토어(i)와 현재 i를 담당하고 있는 창고(W_m)를 찾는다.
   2.2 스토어 i를 W_k로 배치하였을 때 TC가 감소하면 i를 W_k로 재배치한다. 아니면 기존에 담당하던 창고(W_m)에 그대로 두고 3으로 간다.
3. (종료) 고려되지 않은 중앙창고를 찾고 1로 간다. 없으면 종료한다.

3.2.2 거리 기반 Heuristic 2

스토어 i를 커버하지 않는 오픈된 중앙창고 중 가장 가까운 중앙창고를 찾고 스토어 i를 할당하였을 때 비용절감이 발생하면 스토어 i를 할당한다. 제안된 휴리스틱은 다음과 같다.

다음 절차를 더 이상 고려할 스토어가 없을 때까지 반복한다.

스토어 i를 커버하지 않는 창고 중 가장 가까운 창고(W_k)를 찾는다. 스토어 i를 W_k에 재할당하였을 때 비용절감이 발생한다면 W_k에 할당하고 아니면 기존에 스토어 i를 담당하던 창고에 할당한다.

3.2.3 가까운 거리의 두 연결그룹의 합병 Heuristic 3

가까운 거리의 두 연결그룹을 합하여 새로운 연결그룹을 형성하고 합쳐진 연결그룹의 중앙창고는 운송비를 최소화하는 곳으로 새로이 정한다. 즉, 두 중앙창고를 없애고 두 중앙창고가 담당하던 스토어들을 새로운 하나의 중앙창고가 담당하게 되고 이 중앙창고의 위치는 운송비를 최소로 하는 곳으로 정한다. 이러한 절차를 합병이라고 부르기로 하자. 제시된 휴리스틱은 다음과 같다.

다음 절차를 더 이상 고려할 연결그룹이 없을 때까지 반복한다.

연결그룹 k의 중앙창고인 W_k와 가장 가까운 오픈된 중앙창고를 가진 연결그룹을 탐색하고 합병했을 때 TC가 감소한다면 합병하고 아니면 그대로 둔다.

4. Simulated Annealing 앨고리듬

SA(Simulated Annealing)방법은 후보해를 바꾸어 가면

서 목적함수를 개선해가는 방법으로서, 국부 최적해에서 벗어나기 위해 국부 탐색방법에 수락과 종료기준 및 무작위 검색을 부과한다. 이렇듯 어떤 해에서 다른 해로 이동해가는 규칙을 이동(move)이라 부른다.

국부 탐색방법은 항상 현재 해보다 나은 해로 찾아가기에 국부 최적해(local optimal solution)에 빨리 수렴해 가지만, 일단 국부 최적해에 수렴이 된 이후에는 근처에 있는 다른 국부 최적해들로 옮겨 갈 수가 없다. 이러한 단점을 극복하기 위해서 SA는 "uphill move"라는 개념을 도입하여 목적함수를 증진시키지 못하는 해에 대하여도 교체 확률 함수 하에 받아들여 옮겨가게 하였다. 이러한 개념을 메트로폴리스 기준이라 부른다.

4.1 패러미터 설정

SA 앨고리듬과 같은 경험적 탐색기법은 관련된 패러미터에 의해 그 성능이 영향을 많이 받는다. 이러한 패러미터에는 초기온도(initial temperature, T_0), 최종온도(final temperature, T_f), 온도제어 패러미터의 감소방법(cooling schedule)등이 있다.

SA의 초기온도는 초기에 이루어지는 대부분의 이동이 수락되도록 충분히 높게 설정되어야 한다. 그러나 지나치게 높은 초기온도는 SA의 탐색기간이 불필요하게 길어지는 문제가 있으므로 예비실험을 통해 적절한 수준의 초기온도를 설정하기로 한다. 즉, 예비실험을 통해 설정된 수락률(수락된 해이동/총 해이동)이상이 되는 최소의 온도를 초기온도로 정한다. 본 논문에서는 수락률이 90%이상이 되도록 초기온도를 설정하였다.

온도제어 패러미터를 탐색과정에 따라 감소시키는 방법을 쿨링스케쥴이라고 한다. uphill move가 발생하면 쿨링스케쥴에서의 현재 온도 T에 1보다 작은 상수 a를 곱한다. 일반적으로 a값은 0.9이상으로 설정되는데 사전 실험을 통해 최상의 해를 가지는 $a = 0.95$를 쓰기로 한다.

현재의 온도(T)가 최종온도(T_f)보다 낮은 경우나 반복횟수(*iter*) 미리 정해둔 최대 반복횟수(*max_iter*)보다 큰 경우에는 종료한다.

4.2 사용된 SA 앨고리듬

본 논문에서 사용한 이동과 내부루프와 외부루프를 먼저 알아보고 SA앨고리듬을 소개하기로 한다. 먼저 "이동"은 다음과 같다. 임의의 스토어 j에 대해 현재 오픈된 중앙창고수를 *no_ware*라고 할 때 1에서 *no_ware*+1사이의 랜덤(random)한 수(m)을 발생시킨다. m에 해당하는 중앙창고에 스토어 j를 배치한다. 만약 $m = no_ware+1$이면 새로운 중앙창고를 오픈하고 스토어 j를 배치한다.

내부루프와 외부루프는 다음과 같다. 내부루프는 이동을 미리 정해진 L번의 반복 수동안 행하여 해를 개선하는 과정이고 외부루프는 내부루프를 종료할 때마다 휴리스틱을 일회 적용하는 단계이다. 즉, 내부루프를 L번 반복한 후 외부루프에서 휴리스틱을 한 번 적용하는데 이 과정을 *max_iter*만큼 반복한다.

*best*를 SA가 탐색한 최소의 TC를 가지는 해라고 하고 TC_{best}를 *best*의 TC라고 할 때 본 연구에 적용된 SA의 앨고리듬은 다음과 같다.

단계 1 : (초기화) 초기온도(T_0)와 최종온도(T_f)를 정한다. 초기해 s와 초기해의 비용 $TC(s)$라 하고 현재 해를 x라 하자. $iter = 0$, $best = s$, $x = s$, $TC_{best} = TC(s)$로 정한다.

단계 2 : (내부루프) 다음을 L번 반복한다.

(1) (이동) 현재의 해(x)에서 임의의 스토어 j를 선택한다. 현재 오픈된 창고의 수를 *no_ware*라고 할 때 j가 속하지 않은 중앙창고 중 $W_m(1 \le m \le no_ware+1)$를 임의로 선택하였을 때 $m = no_ware+1$이면 새로운 중앙창고 W_{no_ware+1} 오픈하고 스토어 j를 배치하고 아니면 W_m에 j를 추가한다. 이러한 이동에 의해 얻어진 새로운 해를 y라 하자.

(2) $\triangle TC = TC(y) - TC(x)$를 계산한다.

- $\triangle TC \le 0$이고 $TC(y) < TC_{best}$이면 $best = y$, $TC_{best} = C(y)$으로 두고, $x = y$, $TC(x) = TC(y)$로 둔다.
- $\triangle TC \le 0$이고 $TC(y) \ge TC_{best}$이면 $x = y$, $TC(x) = TC(y)$로 둔다.
- $\triangle TC > 0$이면, 확률 $exp(-\triangle TC/T)$으로 $x = y$, $TC(x) = TC(y)$, $T = a \times T$로 갱신하고(**uphill move**), 아니면 $x = x$로 둔다.

단계 3 : (외부루프) 3장에서 소개한 휴리스틱을 적용한다.

단계 4 : (종료조건) 만약 $iter \le max_iter$이고 $T_f < T$이면 $iter = iter + 1$로 갱신하고 단계 2로 가고 아니면 종료한다.

5. 실험 결과

본 절에서는 먼저 작은 크기의 문제에 적용할 수 있는 이동주 외(2006)의 h2-1, h3-1 휴리스틱과 SA를 3.2절의 휴리스틱과 혼용한 경우를 작은 크기의 문제를 이용하여 비교해 보았다. 그 후 보다 큰 크기의 문제를 이용하여 혼용한 SA간의 수행도를 평가하였다. 마지막으로 3.2절에서 제시한 휴리스틱만 적용한 경우를 혼용한 SA들과 수행도를 비교하여 보았다.

5.1 SA의 수행도 평가

본 논문에서 사용한 SA와 혼용한 SA의 수행도를 평가하기 위해 이동주외(2006)에서 제안한 휴리스틱 중 가장 좋은 결과를 보여주는 h2-1과 h3-1라고 부르는 두 개의 휴리스틱과 SA, SA1-SA7의 결과를 비교하였다. 여기서 SA1-SA7는 SA와 3.2절에서 제시한 휴리스틱을 혼용하는 것인데 다음과 같다.

- SA : 일반적인 SA 알고리듬
- SA1 : SA와 거리 기반 휴리스틱 1을 혼용
- SA2 : SA와 합병 휴리스틱 혼용
- SA3 : SA와 거리기반 1, 합병 휴리스틱 순으로 혼용
- SA4 : SA와 합병, 거리기반 휴리스틱 1 순으로 혼용
- SA5 : SA와 거리기반 휴리스틱 2를 혼용
- SA6 : SA와 거리기반 2, 합병 휴리스틱 순으로 혼용
- SA7 : SA와 합병, 거리기반 휴리스틱 2 순으로 혼용

SA1, SA2, SA5는 내부루프가 끝날 때마다 거리기반 1, 합병, 거리기반 2 휴리스틱을 각각 적용하였다. SA3와 SA6은 내부루프가 끝날때마다 거리기반1 혹은 거리기반2를 적용하고 후에 합병 휴리스틱을 적용하였다. 그리고, SA4와 SA7은 내부루프가 끝날때마다 합병 휴리스틱을 적용하고 후에 거리기반 2 휴리스틱을 적용하였다.

h2-1과 h3-1은 작은 규모의 문제에 적합한 휴리스틱으로 휴리스틱의 오차율을 "(휴리스틱의 해의 TC - 최적해의 TC)/최적해의 TC"라고 할 때 1.49%와 0.86%의 오차율을 나타내었다(이동주외(2006)). 이들 휴리스틱들은 일종의 탐욕적 앨고리듬(Greedy Algorithm)으로 모든 연합을 고려하여 효율이 높은 연합을 하나씩 선택하는 휴리스틱이다. 즉, 스토어가 10개 있는 경우의 연합의 수는 2^{10}개 이며 100개의 경우는 2^{100}개이다. 스토어가 100개인 경우에는 고려해야할 연합의 수가 너무 많아 적용이 불가능하므로 SA와 같은 메타휴리스틱을 적용하여야 한다.

그러므로 스토어가 10개인 경우를 비교한 결과 SA, SA1-SA7이 모든 경우에 있어 h2-1과 h3-1보다 나은 결과를 보여주었다. 특히, SA, SA1-SA3, SA5-SA7는 모든 경우에 최적해를 찾았으며, SA4는 데이터 3을 제외하고는 모든 경우에 최적해를 찾았다. <표 1>에 SA 4와 h2-1과 h3-1의 결과가 주어져 있다.

SA, SA1-SA7의 실험조건과 데이터 생성조건은 스토어 수가 10개인 것만 제외하고는 앞으로 설명할 5.2절의 내용과 동일하며 모든 경우에 있어 3초 이내에 결과를 구할 수 있었다.

<표 1> 스토어가 10개인 경우의 결과

데이터	최적해	SA4	h2-1	h3-1
1	17403	17403	17403	17975
2	20280	20280	20280	20868
3	26221	26491	27217	26951
4	14059	14059	14059	14059
5	17886	17886	18779	17886
6	13176	13176	13254	13176
7	13114	13114	13114	13114
8	9467	9467	9467	9467
9	17860	17860	17860	18081
10	18212	18212	18212	18369

5.2 여러 가지 혼용 SA의 실험결과

100개의 스토어가 있는 각 10개의 예제를 생성하였고, 모든 예제에서 스토어는 동일한 하나의 품목만 취급하고, 선행기간(lead time)은 모든 스토어들에 대해 3주이며 제품단위당 운송비는 0.01$/km이다. 각 스토어의 창고와 중앙창고는 같은 주문당 주문비용(setup cost), 제품단위당 재고유지비용(holding cost), 유실판매 벌과비용(penalty cost)을 가지며 예제에 따라 각각 U[$50, $150], U[$1, $5], U[$5, $9]이다. 또한, 선행기간동안의 수요의 평균은 U[100, 1000]이며, 표준편차는 U[10, 55]이다. 각 스토어간의 거리는 2차원 평면으로 가정하여 x축, y축 각각에 대해 U[0km, 50km]이다. 제안된 앨고리듬은 'C 언어'로 구현되어 Intel Core2 Dual(1.83GHz) CPU가 탑재된 pc로 실행되었다.

본 논문에서는 아래와 같은 점에 대해서도 실험을 통해 알아보았다.

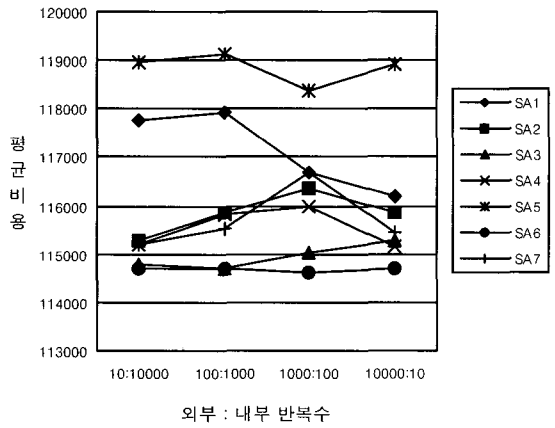
- SA와 제안된 휴리스틱의 혼용은 결과에 얼마나 영향을 미치나?
- 휴리스틱을 쓰는 빈도에 따라 계산된 해는 얼마나 영향을 받는가?
- 제안된 3개의 휴리스틱을 적용하는 순서에 따라 계산된 해는 얼마나 영향을 받는가?

SA에 영향을 미치는 많은 패러미터들이 있는데, 이 실험에 쓰인 요소들은 다음과 같다.

- *max_iter* : 외부 반복횟수.
- L : 내부루프의 반복수,
- T_0 : 초기온도, 5000(수락율이 92%로 90% 이상이 되도록 정함).
- T_f : 최종온도 1,

• α : 0.95(사전실험을 통해 최고의 해를 가지는 값으로 정함).

본 연구에서는 해가 충분히 수렴하도록 총 탐색횟수를 100,000이 되도록 하였다. 즉, 내부루프의 반복수와 외부 반복수를 곱하면 100,000으로 최소한 이 횟수만큼의 해를 탐색하게 된다. SA는 4.2절에 소개된 SA 앨고리듬에서 단계 3의 휴리스틱을 적용치 않고 L=100,000, max_iter=1로 하였다.



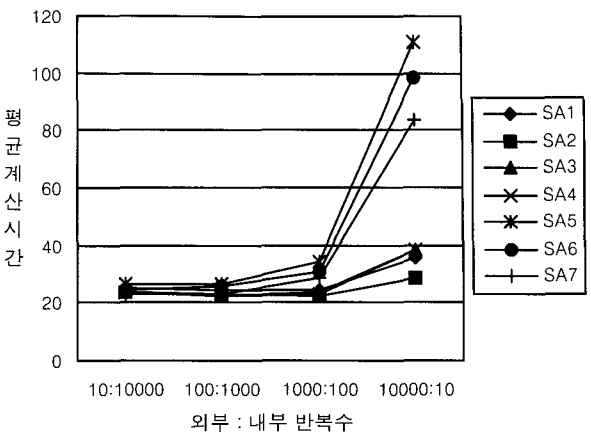
<그림 2> 외부 대 내부반복수의 변화에 따른 비용의 변화

<그림 2>는 외부 대 내부 반복수의 변화에 따른 10개의 예제의 평균비용을 나타내고 있다.

SA3과 SA6이 가장 좋은 해를 제공하였고, 그 다음으로는 SA2, SA4, SA7이 좋은 해를 제공하였다. SA3과 SA4, SA6과 SA7은 거리기반 휴리스틱과 합병휴리스틱을 적용하는 순서가 다른 경우인데 결과가 차이가 있는 것으로 보인다. 또한, SA3는 외부 대 내부 반복수가 100 : 1000인 경우에 가장 좋은 해를 제공한 반면 SA4는 10 : 10000과 10000 : 10인 경우에 좋은 해를 제공하였다. 하지만, 10000 : 10의 경우에는 <그림 3>에서 보는 대로 다른 경우와 달리 오랜 계산시간을 요하므로 평균비용의 차이만으로 좋은 해를 제공했다고 하기에는 무리가 있다.

SA1은 10000 : 10인 경우에 SA2는 10 : 10000인 경우에, SA5는 1000 : 100인 경우에 가장 좋은 해를 제공하였다. 대부분의 경우에 있어 SA2가 SA1이나 SA5보다 나은 해를 제공하는 것으로 보인다.

<그림 3>에 외부 대 내부 반복수의 변화에 따른 계산시간이 나타나 있는데 외부 반복수가 커서 국부탐색해를 적용하는 횟수가 많아지는 경우에 계산시간이 증가하는 것으로 나타났다.



<그림 3> 외부 대 내부 반복수의 변화에 따른 계산시간(초)

특히, 외부 대 내부 반복 수의 비율이 10000 : 10인 경우 SA5, SA6, SA7의 평균계산시간이 급격히 상승하였는데 이는 공통으로 쓰인 거리기반 휴리스틱 2의 계산시간이 오래 걸리기 때문이다. <표 2>와 <표 3>에는 <그림 2>에 나타난 것 중 최선의 외부 대 내부반복수에 대해 각 예제(data 1-data 10)를 10회씩 반복실험 한 결과이다.

'평균', '표준편차', '시간(초)'은 각각의 방법이 데이터별로 보인 해들의 평균값($), 표준편차, 평균계산시간(CPU time, 초)를 나타낸다.

각 경우별 사용된 내부 대 외부 반복수는 SA1은 10000 : 10, SA2는 10 : 10000, SA5는 1000 : 100로 하였다. <표 3>에서는 SA3는 100 : 1000, SA4는 10000 : 10, SA6는 10 : 10000, SA7은 10 : 10000으로 하였다. 대부분의 경우 가장 좋은 해를 보이는 외부 대 내부 반복수의 경우를 선택하였지만, 해가 큰 차이가 없고 계산시간이 짧은 경우가 있다면 선택하였다. 예를 들면, SA6의 경우 1000 : 100이 가장 좋은 결과를 보이지만 10 : 10000의 결과와 비교해 해의 질은 큰 차이가 없으며 계산시간에 오래 걸리므로 10 : 10000으로 하였다.

<표 2>와 <표 3>의 결과를 보면 평균이 SA6가 가장 좋은 해를 탐색했으며 표준편차도 작아서 안정적으로 좋은 해를 탐색하는 것으로 보인다. SA3과 SA6이 SA4와 SA7에비해 좋은 결과를 보이므로 거리기반 휴리스틱을 먼저 적용 후에 합병알고리듬을 적용하는 것이 좋은 결과를 보이는 것으로 나타났다.

흥미로운 사실은 거리기반 휴리스틱 2를 적용한 SA5가 다른 휴리스틱을 적용한 SA1이나 SA2보다 못한 결과를 보여주나 거리기반 2와 합병휴리스틱을 함께 적용한 SA6이 SA3, SA4, SA7보다 좋은 결과를 보여주는 것으로 나타났다.

<표 2> SA, SA1, SA2, SA5의 알고리듬의 성능 비교

데이터	SA			SA1			SA2			SA5		
	평균	표준편차	시간(초)	평균	표준편차	시간(초)	평균	표준편차	시간(초)	평균	표준편차	시간(초)
1	80003	310	29	79608	97	41	79732	364	28	79874	225	41
2	155863	2020	23	152323	1274	30	149579	522	20	157479	1712	29
3	118757	878	24	117427	1192	31	116094	721	21	119660	881	31
4	160814	3482	22	156514	1604	29	155019	1075	20	164373	1734	29
5	145452	1430	24	142833	777	31	142260	1210	21	146657	1449	30
6	85790	677	28	85993	716	38	84961	348	25	85972	709	39
7	104444	282	25	103824	550	35	103251	433	23	104274	663	33
8	103194	678	25	102663	933	33	100510	277	22	104411	781	36
9	93988	603	28	93389	520	39	92309	202	25	94491	250	40
10	132490	1457	23	128159	706	28	127364	206	21	133381	858	28
평균	118116	1186	25	116273	837	34	115108	536	23	119057	926	34

<표 3> SA3, SA4, SA6, SA7의 알고리듬의 성능 비교

데이터	SA3			SA4			SA6			SA7		
	평균	표준편차	시간(초)	평균	표준편차	시간(초)	평균	표준편차	시간(초)	평균	표준편차	시간(초)
1	79587	122	27	80217	473	47	79477	9	28	79674	305	28
2	150120	706	20	149807	871	34	149345	370	21	150292	975	21
3	115559	384	20	115631	338	34	115440	372	21	115411	309	21
4	154594	786	20	154924	1020	33	154006	477	20	154507	1506	20
5	141154	537	21	142013	1075	34	140786	155	21	142182	1496	21
6	84817	430	25	85335	433	42	84761	73	27	84849	323	26
7	103033	62	22	103627	613	37	103017	157	24	103522	679	23
8	100385	88	22	100802	387	35	100251	142	23	100496	174	22
9	92201	271	24	92582	339	41	92069	223	25	92370	307	25
10	127580	401	20	127688	327	34	127493	413	21	127435	319	21
평균	114903	379	22	115263	588	37	114664	239	23	115074	639	23

이상 실험을 통해 드러난 결과를 요약해 보면 다음과 같다.

- SA6이 가장 좋은 해를 제공하였고, 외부 대 내부 반복수에 민감하게 반응하지 않는 것으로 나타났다.
- SA2가 SA1과 SA5보다 일반적으로 나은 결과를 보여주었다. 즉, 하나의 국부 휴리스틱만 적용하는 경우 합병 휴리스틱이 거리기반 휴리스틱을 적용하는 것보다 나은 해를 제공하였다.
- SA3와 SA6가 SA4와 SA7보다 나은 해를 제공하는 것으로 보아 본 논문의 문제의 경우 휴리스틱의 적용순서가 결과에 영향을 미치는 것으로 보인다. 즉, 거리기반 휴리스틱을 먼저 적용하고 난 후에 합병 휴리스틱을 적용하는 것이 반대로 적용하는 것보다 나은 해를 제공한다. 그 이유는 합병 휴리스틱을 적용한 SA2가 거리기반 휴리스틱을 적용한 SA1이나 SA5보다 작은 평균과 표준편차를 보여주듯이 SA3과 SA6은 합병 휴리스틱을 먼저 적용하여 TC의 평균과 표준편차를 SA2나 SA5보다 낮게 하고 이후 거리기반을 적용하여 해를 향상시킨 것이기 때문인 듯하다.
- 두 개의 휴리스틱을 적용한 SA3, SA4, SA6, SA7이 한 개의 휴리스틱을 적용한 SA1, SA2, SA5는 보다 나은 해를 제공하였다.
- SA5의 경우 가장 나쁜 결과를 보여주지만, 합병과 함께 적용한 SA6의 경우 가장 안정적이고 좋은 해를 제공하고 있다.
- 모든 경우에 있어 외부 대 내부 반복수의 변화에 따라 결과에 차이가 있는 것으로 보이며 서로 다른 외부 대 내부반복수의 영역에서 나은 결과를 보이는 것으로 보인다.

5.3 휴리스틱을 적용한 경우

3.2절에서 제시한 휴리스틱을 반복횟수를 100,000번으로 하여 구한 결과가 <표 4>와 같다. 평균을 보면 거리기반 1과 거리기반 2는 많은 시간을 소요하였지만 합병 앨고리듬을 적용한 경우보다 나쁜 결과를 보여주었다.

<표 4> 휴리스틱을 적용한 경우의 결과

데이터	거리 기반 1	시간 (초)	거리 기반 2	시간 (초)	합병	시간 (초)
1	84140	238	80726	1498	80784	83
2	153441	141	161547	741	152244	65
3	122692	199	121415	779	118097	64
4	158064	151	168809	761	157832	67
5	160556	205	149420	725	143777	64
6	87544	245	87508	1565	85782	81
7	108059	243	104913	849	104744	66
8	105966	235	116571	1150	102476	69
9	93583	245	95729	1311	93660	76
10	128638	159	136653	689	135531	61
평균	120268	206	121329	1007	117493	70

거리기반 1, 거리기반 2, 합병 앨고리듬을 적용한 경우와 <표 3>의 SA와 이들 앨고리듬을 혼용한 SA1, SA5, SA2의 결과를 각각 비교해보면 SA1, SA5, SA2가 항상 나은 결과를 보여주며 계산시간도 짧은 것으로 보인다. 이들 휴리스틱만 적용한 경우에는 국부최적해로 수렴한 이후로는 이웃의 다른 국부최적해로 옮겨갈 수가 없고 SA와 혼용한 경우에는 SA가 "uphill move"를 이용하여 이웃 국부최적해로 옮겨갈 수 있게 하기 때문인 듯 하다.

6. 결 론

본 논문에서는 일반적인 단품종, 단일기간의 재고모델에 운송비가 추가된 경우 비용을 최소화하는 중앙창고의 수와 위치를 결정하는 문제를 고려하였다. 특히 스토어의 수가 많은 경우에 이용할 수 있는 거리를 기반으로 한 휴리스틱과 합병 휴리스틱을 이용한 SA를 제안하고 예제를 통하여 성능을 알아보았다. 일반적으로 SA가 좋은 근사해를 제공해 주는 것으로 알려져 있는데 일반적인 SA의 해와 비교해 본 결과 제안된 휴리스틱들을 적용한 SA가 좋은 근사해를 제공해 주었다. 이를 통해 본 논문에서 제안한 기법은 유망한 하나의 기법임을 알 수 있었다.

미래의 연구과제로는 큰 규모의 문제에 대한 최적해를 찾는 기법과 좀 더 효율적인 앨고리듬의 개발이 필요하다.

참고문헌

[1] 권재현, 박상민, 남호기; "SCM 구축을 위한 협업적 수요예측 모형 개발 - 통신장비 제조산업의 협업 수요예측 실제 사례 모형 연구", 산업공학, 17(1) : 84-92, 2004.

[2] 김정자, 최규탁, 정재욱; 2 단계 엔트로피를 이용한 Simulated Annealing의 개선에 관한 연구, 설비관리학회, 4(1) : 159-165, 1999.

[3] 이동주, 황인극, 박동진; "휴리스틱을 이용한 중앙창고의 수와 위치결정", 산업공학(IE Interfaces), 19(1) : 78-85, 2006.

[6] 우훈식; "Simulated Annealing을 이용한 순열 Flow-shop 에서의 Makespan 최소화", 춘계산업공학회 논문집 : 42-45, 1997.

[5] 이승근, 이홍철; "선형계획법을 이용한 협업공급망계획 수립모델", 산업공학, 17(4) : 472-481, 2004.

[4] 이현남, 한치근; "Set Covering 문제의 해법을 위한 개선된 Simulated Annealing 알고리즘", 산업공학, 12(1) : 94-101, 1999.

[7] Hackness, J. and ReVelle C.; "Facility location with increasing production costs," *European Journal of Operational Research*, 145 : 1-13, 2003.

[8] Holmberg K.; "Exact solution methods for uncapacitated location problems with convex transportation costs," *European Journal of Operational Research* : 114, 127-140, 1999.

[9] Johnson, A. and Montgomery, D. C.; "Operations Research in Production Planning, Scheduling, and Inventory Control," Wiley, New York, 1974.

[10] Kaisa M., Makela, M. M., Maaranen, H.; "Efficient hybrid methods for global continuous optimization based on simulated annealing," *Computers & Operations Research*, 33 : 1102-1116, 2006.

[11] Kirkpatrick, S. Gelatt, C. D., and Vecchi, M. P.; "Optimization by simulated annealing," *Science*, 220 : 671-680, 1983.

[12] M'Hallah, R.; "Minimizing total earliness and tardiness on a single machine using a hybrid heuristic," *Computers & Operations Research*, 34 : 3126-3142, 2007.

[13] Mishra, N., Prakash, Tiwari, M. K., Shankar, R., and

Chan, T. S.; "Hybrid tabu_simulated annealing based approach to solve multi-constraint product mix decision problem," Expert System with Application : 446-454, 2005.

[14] Yoo, M. and Gen, M.; "Scheduling algorithm for real-time tasks using multiobjective hybrid genetic algorithm in heterogeneous multiprocessors system," *Computers & Operations Research*, 34 : 3084-3098, 2007.

[15] Wu, L. Y., Zhang, X. S., and Zhang, J. L.; "Capacitated facility location problem with general setup cost," *Computers & Operations Research*, 33 : 1226-1241, 2006.