

물류 센터 위치 선정 및 대리점 할당 모형에 대한 휴리스틱 해법

Meta-heuristic Method for the Single Source Capacitated Facility Location Problem

석상문*, 이상욱**

특허청 정보심사과*, 목원대학교 정보통신공학과**

Sang-Moon Soak(soakbong@hotmail.com)*, Sang-Wook Lee(slee@mokwon.ac.kr)**

요약

시설물 입지 선정 문제(FLP)는 전통적인 최적화 문제중에 하나이다. FLP에 공급제약과 하나의 고객은 하나의 시설물에서만 제품을 공급받을 수 있다는 제약을 추가하면 단일 시설물 공급제약을 가지는 시설물 위치 설정 문제(SSFLP)가 된다. SSFLP는 NP-hard 문제로 알려져 있으며 진화 알고리즘과 같은 휴리스틱 알고리즘을 사용하여 해결하는 것이 일반적이다. 본 논문에서는 SSFLP를 위한 효율적인 진화 알고리즘을 제안한다. 제안하는 알고리즘은 적응형 링크 조절 진화 알고리즘과 3가지 휴리스틱 해 개선 방법을 조합하여 고안되었다. 제안하는 알고리즘을 벤치마크 문제에 적용하여 다른 알고리즘과 성능을 비교분석해 본 결과, 제안하는 알고리즘은 중간 크기의 문제에서 대부분 최적해를 찾았으며 큰 문제에서도 안정된 결과를 보여주었다.

■ 중심어 : | 클러스터링 | 최적의 클러스터 수 | 클러스터링 타당성 평가기준 | 휴리스틱 해 개선 방법 |

Abstract

The facility location problem is one of the traditional optimization problems. In this paper, we deal with the single source capacitated facility location problem (SSCFLP) and it is known as an NP-hard problem. Thus, it seems to be natural to use a heuristic approach such as evolutionary algorithms for solving the SSCFLP. This paper introduces a new efficient evolutionary algorithm for the SSCFLP. The proposed algorithm is devised by incorporating a general adaptive link adjustment evolutionary algorithm and three heuristic local search methods. Finally we compare the proposed algorithm with the previous algorithms and show the proposed algorithm finds optimum solutions at almost all middle size test instances and very stable solutions at larger size test instances.

■ keyword : | Single Source Capacitated Location Problem | Evolutionary Algorithm | Heuristic Local Search Algorithm |

I. 서론

물류 네트워크에 있어서 창고, 물류센터 등과 같은 고정시설물의 입지를 결정하는 것은 전체 물류 시스템

의 형태 및 구조를 결정짓는 중요한 의사결정 문제이다. 이러한 결정은 먼저 대안을 설정하고, 시스템을 운영하는 데 소요되는 제반비용을 산정하는 순으로 이루어지며, 고정 시설물의 입지결정은 세부적으로 고정시

설물의 수, 위치, 그리고 규모를 결정하는 문제이다.

최근 기업은 물류부문에서 고객 서비스 증대와 비용 절감을 위해 효율적인 물류시스템을 구축하고 운영하는 것에 많은 노력을 기울이고 있는데, 물류센터와 같은 시설물의 입지 선정이 중요한 요소로 등장하고 있다. 즉, 물류센터는 생산자와 수요자를 연결하여 시간적 가치를 창출하며, 수요자의 요구량을 공장에서 직접 생산하여 배송하는 것은 비용 면에서 경제적이지 못할 뿐만 아니라 고객에 대한 서비스의 품질 또한 떨어뜨리기 때문에 수요자와 생산자 사이에 일종의 버퍼 역할을 하는 물류 센터의 설치 및 위치 선정은 물류 부분의 중요한 이슈 중에 하나이다.

일반적으로 물류센터를 중심으로 이루어지는 물류시스템은 크게 3 부분으로 나누어진다. 대리점으로부터 제품을 구매하는 고객과, 고객에게 제품을 직접적으로 공급하며, 고객의 수요를 충족시키기 위해 물류센터로부터 제품을 공급받는 대리점 및 대리점의 요구를 충족시키기 위해 제품을 공급하는 물류센터가 바로 그것이다.

이러한 물류시스템을 가진 회사가 초기에 다양한 고객의 요구를 충족시키기 위해 미리 대리점들이 입주하게 될 위치가 알려져 있을 경우 이들 대리점의 수요량을 충족시키는 최적의 물류센터의 위치를 선정하는 문제를 고려해 보자.

일반적으로 시설물 입지 선정 문제(the facility location problem : FLP)는 수송비용과 설비 설치비용(고정비용)의 합을 최소화시키는 방식으로 일련의 시설물들의 위치를 선택하는 문제로 Balinski[2]에 의해 처음으로 다루어졌다. 그 이후 많은 다양한 시설물 입지 선정 모형들이 개발되었는데, 각 시설물이 공급할 수 있는 능력(capacity)에 대한 제약을 가지는 경우인 공급 제약을 가지는 시설물 위치 설정 문제(the capacitated facility location problem: CFLP)도 그 중의 하나이다. 이전의 대부분의 연구는 CFLP 문제를 해결하기 위해 라그랑지언 휴리스틱(a Lagrangian heuristic)을 기반으로 하여 개발되었으며, Daskin[1]은 휴리스틱 접근법들과 최적해법을 포함하여 CFLP를 해결하기 위한 다양한 접근법들을 소개하였다.

그리고 CFLP에서 각 고객이 단지 하나의 시설물에

서만 제품을 공급받을 수 있다는 제약을 부가하게 되면 단일 시설물 공급제약을 가지는 시설물 위치 설정 문제(the single source capacitated facility location problem: SSCFLP)로 변형이 되는데, 본 논문에서 다물류 센터 설비 입지 문제는 SSCFLP로 모델링이 가능하다.

CFLP에 비해 SSCFLP는 그 동안 상대적으로 많은 연구가 이루어지지 않았으며, SSCFLP도 NP-hard 조합 최적화 문제로 분류되고 있기 때문에 최적해를 구하는 해법을 이용하는 것에 한계가 있다. 따라서 많은 연구들이 휴리스틱 기법을 개발하는 방향으로 이루어졌다.

Klincewicz와 Luss[8]은 일종의 라그랑지언 휴리스틱을 사용하여 이 문제를 해결하였으며, Beasley[3] 또한 물류센터의 용량 제약과 고객 할당 제약을 완화시킨 라그랑지언 휴리스틱을 제안하였다. Holmberg 등[6]도 일련의 매칭 문제를 해결하는 반복된 매칭 알고리즘에 기반한 휴리스틱을 라그랑지언 휴리스틱과 결합하는 방법을 제안하였다.

최근에는 복잡한 조합 최적화 문제를 해결하기 위해서 메타 휴리스틱 접근법들이 많이 사용되고 있는데, Cortinhal과 Captivo[4] 등은 SSCFLP의 하한값을 얻기 위해서 라그랑지언 완화법을 제안하였고, 상한값을 얻기 위해서는 지역탐색법과 타부 서치를 이용하였다. 또한, Cortinhal과 Captivo[5]는 네 가지 유전 알고리즘을 이용하여 SSCFLP를 해결하려고 시도하였으며, Julstrom[7] 또한 임의로 선택된 물류 센터 중 각 고객이 가장 적은 수송비용으로 서비스 받을 수 있는 물류센터에 고객을 할당하는 방식으로 해를 디코딩하는 진화 알고리즘을 제안하였으며, Cortinhal과 Captivo[5] 등이 제안한 네 번째 유전 알고리즘과 비교 실험을 통해 제안하는 방법의 우수성을 보여주었다.

본 논문에서도 최근 SSCFLP 문제를 해결하기 위해 주로 이용되고 있는 메타 휴리스틱 접근법을 이용하는 새로운 방법을 제안하고 기존의 메타 휴리스틱 기법과의 비교를 통해 제안하는 방법의 성능을 보이는 것을 목적으로 한다. 제안하는 방법은 기존의 Soak[10]이 제안한 진화알고리즘을 기반으로 하고 있으며, 해를 개선시키기 위해 새로운 초기해 생성 방법과 3가지 지역 탐

색 휴리스틱 기법을 제안한다.

본 논문은 제2장에서 물류 센터 입지 선정 및 대리점 할당 모형에 대해 설명하고, 제3장에서는 제안하는 알고리즘을 구체적으로 기술한다. 그리고 제4장에서 기존의 기법들과의 비교 실험을 통해 제안하는 방법의 성능을 보이며, 마지막으로 결론을 맺는다.

II. 물류 센터 입지 선정 및 대리점 할당 모형

우선 물류 센터 입지 선정 및 대리점 할당 모형에서는 n 대리점과 m 물류 센터 후보지가 이미 알려져 있다고 가정하고, 각각의 대리점 i 는 요구량 d_i 를 가지며 각 대리점 i 는 하나의 물류 센터 j 로부터 물량을 공급받을 수 있다. 그리고 대리점 i 가 각 물류 센터 후보지 j 로부터 물량을 공급 받을 경우 수송비용 c_{ij} 가 발생하게 되며, 각 물류 센터 후보지 j 의 공급 가능량 s_j 및 물류 센터 후보지 j 가 선정될 경우 발생하게 되는 고정비용 f_j 도 미리 알려져 있다고 가정한다. 결국 목적함수는 최소의 수송비용 c_{ij} 와 고정비용 f_j 을 가지고 모든 대리점의 요구량을 충족시킬 수 있는 물류 센터 후보지를 선정하는 것이다. 따라서 이러한 조건을 정형화하면, 일반적으로 잘 알려져 있는 단일 시설물 공급제약을 가지는 시설물 위치 설정 문제(the single source capacitated facility location problem: SSCFLP)로 모형이 가능하다.

$$\text{Min} \sum_{i=1}^n \sum_{j=1}^m c_{ij}y_{ij} + \sum_{j=1}^m f_jx_j \quad (1)$$

$$\text{s.t.} \quad \sum_{j=1}^m y_{ij} = 1, \quad i = 1, \dots, n \quad (2)$$

$$\sum_{i=1}^n d_iy_{ij} \leq s_jx_j, \quad j = 1, \dots, m \quad (3)$$

$$x_j \in \{0, 1\}, \quad j = 1, \dots, m \quad (4)$$

$$y_{ij} \in \{0, 1\}, \quad i = 1, \dots, n, \quad j = 1, \dots, m \quad (5)$$

여기서, x_j 는 만약 어떤 물류 센터 j 가 후보지로 선

정이 되면 1을 그렇지 않으면 0의 값을 가지는 변수이며, y_{ij} 는 만약 어떤 대리점 i 가 물류 센터 j 에 할당이 되면 1을 그렇지 않으면 0의 값을 가지는 변수이다.

식 (2)는 각 대리점이 하나의 물류 센터에서만 물량을 공급받아야한다는 것을 나타내는 제약식이며, 식 (3)은 선택된 각 물류 센터에 할당되는 대리점의 총 요구량이 각 물류 센터의 공급가능량을 초과할 수 없음을 나타내는 제약식이다.

III. 제안하는 알고리즘

1. 적응형 링크 조절 진화 알고리즘

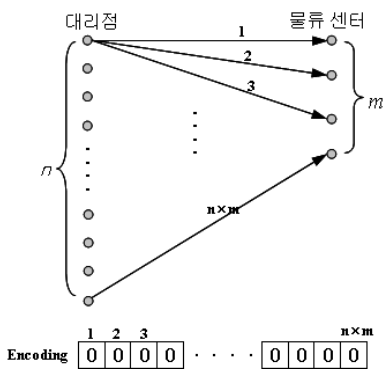
본 논문에서는 최근에 Soak[10]이 제안한 적응형 링크 조절 진화 알고리즘(adaptive link adjustment evolutionary algorithm : ALA-EA)을 기본으로 하는 새로운 진화 알고리즘을 개발한다. 따라서 우선 적응형 링크 조절 진화 알고리즘의 기본 구조에 대해 간단히 설명한다. 보다 자세한 내용은 참고문헌을 참조하기 바란다.

ALA-EA는 기존의 다른 진화 알고리즘들과 많은 차이가 있다. 각 유전인자는 다루는 문제의 각 링크(또는 에지)를 나타내고 초기해는 모두 '0'으로 설정이 된다 ([그림 1]의 (a) 참조).

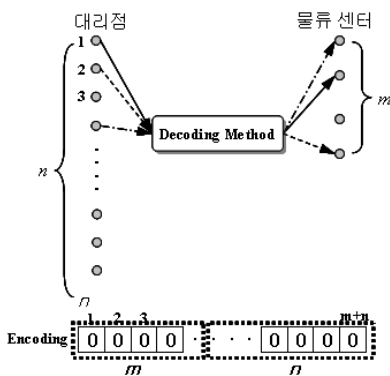
그 다음 각 유전인자에 있는 값들을 오름차순으로 정렬하고 정렬된 순서대로 해당 링크를 선택할지를 결정하게 된다. 즉, 해당 링크가 다루는 문제의 제약 조건을 만족한다면 선택을 하게 되고, 그렇지 않다면 정렬된 다음 링크를 고려하는 식으로 모든 링크를 정렬된 순서에 따라 고려하고 나면 하나의 해가 생성된다. 이렇게 생성된 해를 목적함수에 따라 평가를 하게 되고, 평가 결과에 따라 'α'와 'β'과정을 적용하게 된다 ([그림 3] Step 3의 3.3 참조).

만약 평가 결과, 해 i 가 현재까지의 찾은 가장 우수한 해보다 더 좋은 결과 값을 나타낸다면 'α'과정을 그렇지 않다면 'β'과정 적용한다. 'α'과정은 모든 유전인자들 중에서 해에 포함된 유전인자들에만 '+ α'값을 더하고, 'β'과정은 모든 유전인자들 중에서 'θ'만큼의 유

전인자들에게 ‘-β’값을 더한다. 결국 디코딩 과정에서 유전인자들이 오름차순으로 정렬된 순서대로 고려가 되기 때문에 ‘α’과정은 일종의 패널티(penalty)의 역할을 하게 되고, ‘β’과정은 일종의 보상(reward)의 역할을 하게 된다. 여기서, 우수한 해에게 패널티를 부여하는 것은 2가지의 이유가 있다. 첫째는 이미 찾은 해를 강화하는 것은 현재 해에 전혀 변화를 주지 못하기 때문이며, 둘째는 현재 해를 강화할 경우 강화된 해로 인해 빠른 시간 내에 해 집단이 이 해에 의해 지배되어 조기 수렴을 하게 되기 때문이다. 그리고, 상대적으로 열등한 해들에게 보상을 부여하는 것은 유전인자의 값의 근소한 차이를 가지는(즉, 오름차순으로 정렬된 순서에 따라 해당 링크를 고려하기 때문에) 링크들끼리의 경쟁을 촉진시켜 이웃해들로의 탐색을 촉진시킬 수 있기 때문이다.



(a) ALA-EA의 기본적인 해 표현법



(b) 제안하는 방법의 해 표현법

그림 1. ALA-EA의 해 표현법

2. 초기해 생성 방법

본 논문에서 사용하는 해 표현법은 [그림 1]의 (b)에서처럼 기본적인 ALA-EA의 해 표현법과 다른 방법을 사용한다. 해의 전반부 m 개의 유전인자는 각 물류 센터의 번호를 의미하며 후반부 n 개의 유전인자는 각 대리점의 번호를 의미한다. 따라서 각 해의 길이는 $m + n$ 이 된다. 또한, 전반부와 후반부는 각각 따로 적합도 함수 값을 계산하기 위해 오름차순으로 정렬된다.

그리고 본 논문에서는 초기해를 좀 더 우수한 해들로 구성하기 위한 방법을 제안한다. 우선 각 물류 센터들에 대해 식 (6)에 따라 모든 대리점들에 대해 비용의 총합을 계산하고, 총 비용이 적게 드는 물류 센터 순으로 정렬하여 정렬된 물류 센터들에 1부터 m 의 유전인자 값을 부여한다 ([그림 3] Step 1의 1.1 참조).

$$PC_j = f_j + \sum_{i=1}^n c_{ij}, \quad \forall j = 1, \dots, m \quad (6)$$

다음은 각 대리점에 대해 식 (7)에 따라 모든 수송 비용의 총합이 가장 적게 드는 대리점들을 정렬하여 정렬된 대리점들에 1부터 n 의 유전인자 값을 부여한다 ([그림 3] Step 1의 1.4 참조).

$$CC_i = \sum_{j=1}^m c_{ij}, \quad \forall i = 1, \dots, n \quad (7)$$

결국 이는 총 비용이 적게 드는 물류 센터와 대리점을 우선적으로 고려하기 위해서이다.

그리고 이렇게 생성된 첫 번째 해의 후반부 즉 대리점들에 대해서만 상호 교환 번이 연산을 적용하여 해 집단을 생성한다 ([그림 3] Step 1의 1.7 참조).

3. 휴리스틱 해 개선 방법

본 논문에서는 목적함수 값 계산 방법을 통해 형성된 해를 개선시키기 위해 3가지 휴리스틱 해 개선 방법을 제안한다.

첫 번째 방법(휴리스틱 1)은 어떤 물류 센터 a ($x_a = 1$)에 할당된 대리점들을 모두 선택되지 않은 물류 센터 b ($x_b = 0$)로 옮겼을 때 공급 가능량을 초과하지 않으면서 비용 이득이 발생하는 경우가 있는지를 확인하고 비용 이득이 발생하면 물류 센터 a 에 할당된 모

든 대리점들을 물류 센터 b 에 할당하고, $x_a = 0$, $x_b = 1$ 로 변경한다 ([그림 3] Step 2의 2.1 참조).

두 번째 방법(휴리스틱 2)은 어떤 물류 센터 a ($x_a = 1$)에 할당된 대리점들을 다른 물류 센터 b ($x_b = 1$)로 이동할 경우 공급 가능량을 초과하지 않으면서 비용 이득이 발생하는 경우가 있는지를 확인하고 해당 대리점을 비용 이득이 가장 크게 발생하는 물류 센터 b 로 이동시키고, 물류 센터 a 의 남아 있는 공급 가능량과 물류 센터 b 의 남아 있는 공급 가능량을 조정한다. 이러한 과정을 모든 대리점들에 대해서 수행한다 ([그림 3] Step 2의 2.2 참조).

마지막으로 세 번째 방법(휴리스틱 3)은 어떤 물류 센터 a ($x_a = 1$)에 할당된 대리점 p 를 다른 물류 센터 b ($x_b = 1$)에 할당된 대리점 q 와 교환할 경우 각 물류 센터 a 및 b 의 공급 가능량을 초과하지 않으면서 비용 이득이 발생하는 경우가 있는지를 확인하고 비용 이득이 가장 크게 발생하는 해당 대리점들을 상호 교환하고, 물류 센터 a 의 남아 있는 공급 가능량과 물류 센터 b 의 남아 있는 공급 가능량을 조정한다. 이러한 과정을 모든 대리점들에 대해서 수행한다 ([그림 3] Step 2의 2.3 참조).

4. 목적함수 값 계산 방법

생성된 해들로부터 목적함수 값을 계산하기 위해서 Julstrom이 제안한 탐욕적 방법(greedy decoding)[7]을 변형하여 사용한다.

우선 랜덤 넘버를 $R = Rand[0, m]$ 의 범위에서 발생시키고, 정렬된 R 개의 물류 센터 중 대리점의 요구량을 만족시키면서 가장 비용이 적게 드는 물류 센터에 대리점을 정렬된 순서대로 할당한다. 만약 R 개의 물류 센터가 어떤 대리점의 요구 사항을 만족시킬 수 없다면 R 을 하나 늘려 해당 물류 센터에 대리점을 할당한다. 다음 대리점은 $R+1$ 개의 물류 센터 중에서 가장 비용이 적게 드는 물류 센터에 할당된다.

이러한 방법을 모든 대리점이 물류 센터에 할당될 때까지 반복 수행한다 ([그림 3] Step 3의 3.1 및 3.2 참조).

5. 유전 연산자들

본 논문에서는 Soak[10]이 기본적인 ALA-EA가 우수한 성능을 보이는 것으로 제안한 실세계 토너먼트 선택전략[9], uniform 교차 연산자(uniform crossover) 및 상호 교환 변이 연산자(exchange mutation)를 사용한다.

우선 실세계 토너먼트 선택전략은 현실 세계의 토너먼트 스포츠 게임처럼 해 집단에 있는 모든 해들을 이웃하는 해들끼리 쌍을 이루어 경쟁을 시키고, 경쟁에서 승리한 해들을 다음 세대의 해 집단에 포함시키며, 승리한 해들은 또 다시 이웃하는 다른 승리한 해들끼리 쌍을 이루어 경쟁을 시키는 과정을 반복 수행하여 해 집단을 형성하는 방법이다 ([그림 3] Step 4의 4.1 참조).

uniform 교차 연산자는 각 유전인자에 랜덤 넘버를 발생시켜 0이 나오면 부모1의 유전인자 값을 1이 나오면 부모2의 유전인자 값을 가져와서 하나의 자손을 형성하는 방법이다 ([그림 3] Step 4의 4.2 참조).

상호 교환 변이 연산자는 임의로 두 개의 유전인자를 선택하여 서로 유전인자의 값을 교환하는 연산자이다. 본 논문에서는 상호 교환 변이 연산자를 대리점과 물류 센터 각각에 대해 독립적으로 수행하였다. 즉, [그림 1]의 (b)에서 해의 1에서 m 번째에 있는 유전 인자끼리 한번, $m+1$ 에서 $m+n$ 번째에 있는 유전 인자끼리 한번 각각 수행된다 ([그림 3] Step 4의 4.3 참조).

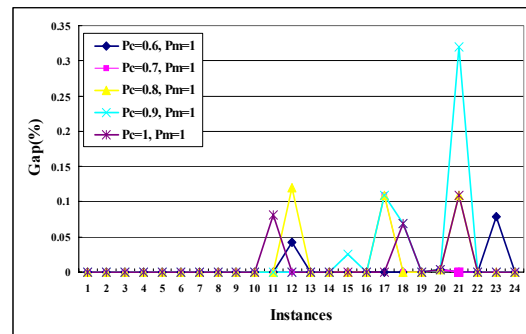


그림 2. 교차 연산자 및 변이 연산자 적용 비율 실험

그리고 본 논문에서는 제안하는 알고리즘에서 사용되는 uniform 교차 연산자와 상호 교환 변이 연산자의 최적 적용 파라미터 값을 찾기 위해 실험을 수행하였

다. [그림 2]는 그 결과를 보여준다. 상호 교환 변이 연산자의 경우 100% 적용하는 것이 가장 우수한 결과를 보였으므로, 여기서는 상호 교환 변이 연산자의 적용 비율을 100%로 두고 uniform 교차 연산자의 적용 비율이 변화할 경우의 결과만을 보인다. 실험결과 uniform 교차 연산자의 적용 비율을 70%, 상호 교환 변이 연산자의 적용 비율을 100%로 하는 경우가 가장 우수한 결과를 보여 준다는 것을 확인할 수 있었다. 따라서 비교 실험에서는 두 연산자의 적용비율로 이들 값들을 사용하였다. 또한, ALA-EA를 위한 변수는 기존의 연구에서 가장 좋은 조합으로 제시하고 있는 α 와 β 값은 1로 θ 값은 70%로 설정하여 비교 실험을 수행하였다.

그리고 [그림 3]은 본 논문에서 제안하는 알고리즘의 유사 코드를 나타낸다.

Step 1. 초기해 생성

- 1.1 $PC_j = f_j + \sum_{i=1}^n c_{ij}, \quad \forall j = 1, \dots, m$
- 1.2 $Asort(PC_j)$; *//* 각 물류 센터를 PC_j 값에 따라 오름차순정렬 **/**
- 1.3 $S_1 = \{g_1, \dots, g_m\}$;
// 정렬된 순서대로 해당 물류센터를 나타내는 유전인자 g_i 에 1에서 m 값을 부여 **/**
- 1.4 $CC_i = \sum_{j=1}^m c_{ij}, \quad \forall i = 1, \dots, n$
- 1.5 $Asort(CC_i)$; *//* 각 대리점을 CC_i 값에 따라 오름차순정렬 **/**
- 1.6 $S_1 = \{g_{m+1}, \dots, g_{m+n}\}$;
// 정렬된 순서대로 해당 대리점을 나타내는 유전인자 g_i 에 1에서 n 값을 부여 **/**
- 1.7 **for** $k = 2$ **to** pop_size
 $S_k = swap(S_1, rand(m+1, m+n), rand(m+1, m+n))$;

Step 2. 휴리스틱 해 개선법

- 2.1 휴리스틱 1 적용
- 2.2 휴리스틱 2 적용

2.3 휴리스틱 3 적용

Step 3. 목적함수값 계산

3.1 모든 해 S_i 를 물류센터 부분 $\{g_1, \dots, g_m\}$ 과 대리점 부분 $\{g_{m+1}, \dots, g_{m+n}\}$ 으로 나눠서 각각 오름차순으로 정렬함.

$$S_i = \{G_1, \dots, G_m, G_{m+1}, \dots, G_{m+n}\}$$

// 정렬된 해 **/**

3.2 $R = rand(1, m)$;

for $i = 1$ **to** m

$A = G_{m+1}$ 에 있는 대리점의 수요량을

G_1 에서 G_R 에 있는 물류센터에서 수용할 수 있는 지를 확인;

if 수용할 수 있다면, G_{m+1} 을 수용 가능한 물류센터에 할당하고 각 물류센터의 공급 가능한 물량을 조정함.

else

$$R = R + 1;$$

end for

3.3 각 해의 목적함수 값에 따라 α 와 β 과정 적용

if $S_i < Best$

α 과정 적용;

else

if $rand() < \theta$

β 과정 적용;

Step 4. 유전 연산자

- 4.1 실세계 토너먼트 선택전략
- 4.2 uniform 교차 연산자
- 4.3 상호교환변이 연산자

Step 5. 종료조건

if 종료조건 == yes

종료;

else

go to step 2;

그림 3. 제안하는 알고리즘의 유사 코드

표 1. Beasley의 cap61~134 문제에서의 실험 결과 비교

Inst.	m	n	C&C +#[7]	Julstrom's EA[7]			The proposed algorithm				
				Best +%	Mean+%	Std(%)	Hits +%	Best +%	Mean +%	Std(%)	Time (s)
cap61	16	50	0.00	0.000	0.000	0.000	100	0.000	0.000	0.000	0
cap62	16	50	0.00	0.000	0.000	0.000	100	0.000	0.000	0.000	0
cap63	16	50	0.01	0.004	0.004	0.000	100	0.000	0.000	0.000	0
cap64	16	50	0.05	0.722	0.722	0.000	100	0.000	0.000	0.000	1.1
cap71	16	50	0.00	0.000	0.000	0.000	100	0.000	0.000	0.000	0.6
cap72	16	50	0.00	0.000	0.000	0.000	100	0.000	0.000	0.000	0.2
cap73	16	50	0.00	0.000	0.000	0.000	100	0.000	0.000	0.000	0.1
cap74	16	50	0.00	0.000	0.000	0.000	100	0.000	0.000	0.000	0.1
cap91	25	50	0.00	0.000	0.000	0.000	100	0.000	0.000	0.000	3.6
cap92	25	50	0.00	0.278	0.286	0.023	100	0.000	0.000	0.000	0.3
cap93	25	50	0.22	0.462	0.462	0.000	100	0.000	0.000	0.000	0.6
cap94	25	50	0.21	0.482	0.485	0.013	100	0.000	0.000	0.000	0.6
cap101	25	50	0.00	0.000	0.000	0.000	100	0.000	0.000	0.000	0.3
cap102	25	50	0.00	0.000	0.000	0.000	100	0.000	0.000	0.000	0.3
cap103	25	50	0.00	0.000	0.000	0.000	100	0.000	0.000	0.000	0.5
cap104	25	50	0.00	0.000	0.000	0.000	100	0.000	0.000	0.000	0.0
cap121	50	50	0.00	0.000	0.108	0.093	100	0.000	0.000	0.000	1.5
cap122	50	50	0.00	0.279	0.286	0.021	100	0.000	0.000	0.000	2.6
cap123	50	50	0.19	0.331	0.331	0.000	100	0.000	0.000	0.000	0.8
cap124	50	50	0.00	0.482	0.485	0.001	100	0.000	0.000	0.000	1.2
cap131	50	50	0.00	0.000	0.043	0.054	80	0.000	0.043	0.101	43.8
cap132	50	50	0.00	0.000	0.000	0.000	100	0.000	0.000	0.000	1.5
cap133	50	50	0.00	0.000	0.033	0.053	100	0.000	0.000	0.000	1.8
cap134	50	50	0.00	0.000	0.000	0.000	100	0.000	0.000	0.000	1.2

IV. 실험

본 논문에서는 보다 정확한 비교 실험을 위해 기존의 알고리즘들이 벤치마크 문제로 사용한 Beasley의 OR-Library¹에 있는 데이터를 이용하였으며, 총 24개의 중형의 문제와 12개의 대형의 문제에서 실험을 수행하였다. 중형의 문제는 50개의 대리점과 16, 25, 50개의 물류 센터 후보지가 존재하는 문제이며, 대형의 문제는 1000개의 대리점과 100개의 물류 센터 후보지가 존재하는 문제이다. 그리고 이들 문제에서 우수한 성능을 보인 Cortinhal과 Captive[5] 및 Julstrom [7]의 실험결과를 비교를 위한 기준으로 사용하였으며 제안하는 알고리즘은 C++를 이용해서 프로그램되었고 Pentium IV 3.4G CPU와 1.5Gbyte Ram 컴퓨터 상에서 실험을 하였다.

제안하는 알고리즘은 중형 및 대형의 문제들에서 각

각 10회 및 5회씩 반복실험을 수행하였고, 종료 조건은 각각 5000세대, 100세대 동안 해의 개선이 없을 경우로 설정하였다.

[표 1] 및 [표 2]에 나타난 각각의 값들은 최적해와 찾아낸 해 사이의 차이(Gap)를 나타내는데, 이는 다음의 수식을 이용해서 계산되었다.

$$Gap = (C - C_{opt}) / C_{opt} \times 100 (\%)$$

또한, 대형의 문제는 현재까지 최적해가 알려져 있지 않기 때문에 알려진 하한값(Lower bound)을 사용하여 차이를 계산하였다.

[표 1]은 중형의 문제들에 제안하는 알고리즘의 성능을 보여준다. 총 24개의 문제들 중 한 문제(cap131)를 제외한 모든 문제의 반복 실험에서 최적해를 찾아내었으며, 계산 시간 또한 최대 평균 4초를 넘지 않았다. 8번의 최적해를 찾은 cap131 문제에서도 최적해를 찾지 못

¹ <http://people.brunel.ac.uk/~mastijb/jeb/info.html>

표 2. Beasley의 capa1~capc4 문제에서의 실험 결과 비교

Inst.	Lower bound	C&C +#[7]	Julstrom's EA[7]			The proposed algorithm			
			Best +%	Mean+%	Std(%)	Best +%	Mean +%	Std(%)	Time (s)
capa1	19,240,822.45	0.28	0.063	0.521	0.514	0.074	0.291	0.150	697.6
capa2	18,438,046.54	0.12	0.064	0.549	0.246	0.056	0.093	0.079	814
capa3	17,765,201.95	0.29	0.000	1.104	0.933	0.000	0.000	0.000	248.2
capa4	17,160,439.01	0.00	0.002	0.423	1.147	0.003	0.003	0.000	871
capb1	13,656,379.58	0.01	0.193	1.231	0.957	1.824	2.160	0.449	508.8
capb2	13,361,927.45	2.87	0.268	1.268	0.728	0.118	0.459	0.303	784.6
capb3	13,198,556.43	2.04	0.366	1.203	0.499	0.287	0.354	0.075	831.8
capb4	13,082,516.50	0.47	0.565	1.059	0.410	0.596	0.987	0.374	540.4
capc1	11,646,596.97	0.70	0.063	0.832	0.607	0.364	0.913	0.428	721.6
capc2	11,570,340.29	0.65	0.001	0.205	0.243	0.036	0.141	0.182	915.8
capc3	11,518,743.74	1.15	0.042	0.392	0.230	0.349	0.357	0.127	738.6
capc4	11,505,767.39	0.09	0.001	0.196	0.203	0.033	0.176	0.207	718.4
		0.725	0.135	0.748	0.559	0.311	0.495	0.197	

한 2번의 계산 시간으로 인해 상대적으로 많은 계산 시간을 필요로 하였지만, 최적해를 찾은 경우의 평균 계산 시간은 3.87초에 불과하였다.

중형의 문제들에서는 기존의 C&C 방법뿐만 아니라 Julstrom의 진화 알고리즘에 비해서 월등히 좋은 성능을 보여 준다는 것을 확인할 수 있다.

[표 2]는 대형의 문제들에서 제안하는 알고리즘의 실험 결과이다. 진하게 표시된 결과들은 세 가지 방법론 중에서 가장 좋은 결과를 찾은 것을 나타내고 있다. 가장 좋은 결과값(Best)들을 비교해 보면, 상대적으로 Julstrom의 진화 알고리즘이 가장 좋은 결과를 보여주며, 그 다음은 본 논문에서 제안하는 진화 알고리즘이 근소한 차이로 우수한 결과를 보여주었다. 하지만 평균 결과를 비교해 보면, 총 12개의 문제 중에서 2개의 문제를 제외하고 모든 문제들에서 더 좋은 결과를 찾아내었다. 전체 문제의 평균 또한 Julstrom의 진화 알고리즘이 0.748%, 제안하는 진화 알고리즘이 0.495%로 더 좋은 결과를 보여주었다. 특히 주목할 점은 표준편차의 결과값인데 이를 보면 평균 0.197%로 제안하는 방법이 아주 안정적으로 우수한 결과값을 찾았다는 것을 확인할 수 있다. 계산시간의 경우는 다른 알고리즘들의 경우 계산시간에 대한 결과가 알려져 있지 않아 직접적인 비교는 어렵지만, 제안하는 알고리즘의 경우 최대 915.8초에서 최소248.2초로 중형의 문제들에 비해 상당히 많은 계산시간을 필요로 하였다. 하지만, 물류 센터의 입지를

선정하고, 대리점을 할당하는 문제가 시간적으로 아주 시급하게 이루어져야하는 문제가 아닌 만큼 계산시간은 충분히 허용할 수 있는 범위라고 판단된다.

이러한 실험 결과들을 분석해보면, 제안하는 방법에서 사용되는 휴리스틱 기법들이 상대적으로 탐색 공간이 덜 복잡한 중형의 문제에서는 아주 우수한 성능을 보인다는 것을 확인할 수 있었지만, 탐색 공간 내에 지역해가 아주 많이 존재하는 대형의 문제들에서는 빠른 시간에 아주 우수한 성능을 가진 해로 수렴하였음에도 불구하고, 기존에 알려진 하한값으로 수렴하지 못하는 단점을 보였다. 이는 제안하는 알고리즘의 기본틀로 사용되고 있는 ALA-EA가 가진 전역 탐색의 능력에 비해 제안하는 휴리스틱 방법들의 지역 탐색 능력이 더 우월하여 조기 수렴하여 전역해로의 이동에 제약을 주기 때문인 것으로 판단된다. 앞으로 이들 연산자들간의 균형을 고려할 수 있는 방법이 마련된다면 현재 보다 훨씬 더 우수한 결과를 얻을 수 있을 것으로 판단된다.

V. 결론

본 논문에서는 물류 센터의 입지 및 대리점을 할당하는 문제를 효율적으로 해결할 수 있는 새로운 진화 알고리즘을 제안하였다. 특히 제안하는 방법은 기존의 방법들과의 비교 실험을 통해 중형의 문제들에서는 아주

우수한 결과를 대형의 문제들에서는 상대적으로 아주 안정적으로 우수한 평균 결과값을 찾는 다는 것을 확인할 수 있었다.

향후 연구로는 실험결과에서 확인할 수 있었던 것처럼 대형의 문제를 좀 더 효율적이고 효과적으로 해결할 수 있는 ALA-EA와 휴리스틱 기법들 간의 탐색능력에 있어서 균형을 맞출 수 있는 방법에 대한 연구가 필요하다 판단되며, 특히나 본 논문에서 다루어진 물류 센터 입지 선정 문제의 경우 다양한 형태로 변형이 가능한데 특히, 현실 문제에 주로 발생하는 이미 선정된 물류센터의 입지를 평가한다든지 아니면 몇몇 물류 센터는 이미 입지해 있고, 새로운 물류센터를 선정하는 문제들에 대한 연구가 이루어질 필요가 있다.

참 고 문 헌

[1] M. Daskin, Network and discrete location models, algorithms and applications, Wiley, New York, 1995.

[2] M. L Balinski, "Integer Programming: Methods, uses, computations," Management Science, Vol.12, pp.253-313, 1965.

[3] J. E. Beasley, "Lagrangian heuristic for location problem," European Journal of Operational Research, Vol.65, pp.383-399, 1993.

[4] M. J. Cortinhal and M. E. Captivo, "Upper and lower bounds for the single source capacitated location problem," European Journal of Operational Research, Vol.151, pp.333-351, 2003.

[5] M. J. Cortinhal and M. E. Captivo, "Genetic algorithms for the single source capacitated location problem," In M.G.C. Resende and J.P. de Sousa, editors, Metaheuristics: Computer Decision-Making, pp.187-216. Kluwer, Boston, 2004.

[6] K. Holmberg, M. Ronnqvist, and D. Yuan, "An exact algorithm for the capacitated facility

location problems with single sourcing," European Journal of Operational Research, Vol.113, pp.544-559, 1999.

[7] B. A. Julstrom, "An evolutionary algorithm for some cases of the single-source constrained plant location problem," Proceedings of the 2008 Genetic and Evolutionary Computation Conference, pp.607-608, 2008.

[8] J. G. Klincewicz and H. Luss, "A Lagrangian relaxation heuristic for capacitated facility location with single source constraints," Journal of the Operational Research Society, Vol.37, pp.495-500, 1986.

[9] S. M. Soak and B. H. Ahn, "A new tree representation for evolutionary algorithms," Journal of the Korean Institute of Industrial Engineers, Vol.31, No.1, pp.10-19, 2005.

[10] S. M. Soak, "Adaptive Link Adjustment' Applied to The Fixed Charge Transportation Problem," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E90-A, No.12, pp.2863-2876, 2007.

저 자 소 개

석 상 문(Sang-Moon Soak)

정회원



- 1988년 8월 : 한국해양대학교 물류시스템공학과(공학사)
- 2000년 8월 : 한국해양대학교 물류시스템공학과(공학석사)
- 2005년 8월 : 광주과학기술원 기전공학과(공학박사)

- 2005년 8월 ~ 2006년 5월 : LG 전자 디지털 가전 연구소 선임연구원
 - 2006년 5월 ~ 현재 : 특허청 심사관
- <관심분야> : 휴리스틱 알고리즘, 최적화, 진화 컴퓨팅

이 상 욱(Sang-Wook Lee)

정회원



- 2000년 2월 : 한국과학기술원 기계공학과(공학사)
 - 2002년 2월 : 광주과학기술원 기전공학과(공학석사)
 - 2007년 8월 : 광주과학기술원 정보기전공학부(공학박사)
 - 2007년 9월 ~ 2008년 9월 : 조지아공과대학교 전산학과 박사후연구원
 - 2008년 11월 ~ 2009년 2월 : 삼성전자 통신연구소 책임연구원
 - 2009년 3월 ~ 현재 : 목원대학교 정보통신공학과 전임강사
- <관심분야> : 휴리스틱 알고리즘, 최적화, 데이터 마이닝, 유비쿼터스 컴퓨팅