

용량제한과 직접수송이 있는 Hub-and-Spoke 네트워크 설계: 진화알고리즘 기반의 접근법에 의해

이현수 · 신경석 · 김여근[†]

전남대학교 산업공학과

A Design of Capacitated Hub-and-Spoke Networks with Direct Shipment: Evolutionary Algorithm based Approach

Hyun Soo Lee · Kyoung Seok Shin · Yeo Keun Kim

Department of Industrial Engineering, Chonnam National University, Gwangju, 500-757

In this paper we address a design problem for hub-and-spoke networks and then consider a capacitated hub locations problem with direct shipment (CHLPwD). We determine the location of hubs, the allocation of nodes to hubs, and direct shipment paths in the network, with the objective of minimizing the total cost in the network. In this paper, CHLPwD is formulated as 0-1 integer programming. We develop an evolutionary algorithm here to solve the large sized CHLPwD. To do this, we present the representation and the genetic operators suitable for the problem and propose a heuristic method for the allocation of nodes to hubs. To enhance the search capability, problem-specific information is used in our evolutionary algorithm. The proposed algorithm is compared with the heuristic method in terms of solution quality and computation time. The experimental results show that our algorithm can provide better solutions than the heuristic.

Keywords: hub-and-spoke, hub location problem, evolutionary algorithm, direct shipment

1. 서론

Hub-and-spoke(H&S) 네트워크는 상호간에 직접 연결된 몇 개의 허브(hub, 중심기지)들과 이들 허브에 연결된 스포크(spoke, 가지 노드)들로 구성되며, 대부분의 화물이 기점(origin)에서 기점과 연결된 허브를 통해 종점(destination)과 연결된 허브를 경유하여 종점으로 가는 수송 경로를 갖는다. 이 구조는 화물 수송에 있어서 공동화와 대량화를 통한 규모의 경제를 잘 반영할 수 있는 물류수송 네트워크이다(Elhedhli and Hu, 2005). H&S 네트워크의 구조는 항공수송에서 시작하여, 점차 항만 물류, 우편시스템, 통신시스템 등에서 널리 사용되고 있다(Ebery *et al.*, 2000).

이러한 H&S 네트워크를 설계하기 위한 허브위치선정문제(Hub Location Problem; HLP)는 지점들과 이들 지점들의 각 쌍

에 흐르는 화물량 및 관련 비용이 주어진 상태에서, 네트워크상의 총 물류비가 최소가 되도록, 허브의 입지선정과 함께 남아있는 지점들을 어떤 허브에 할당할 것인가를 결정하는 것이다. 여기서 허브가 아닌 지점은 스포크가 된다.

이러한 HLP는 허브의 수, 각 허브의 용량제한, 화물이동경로, 그리고 한 스포크가 연결 가능한 허브 수 등에 따라 여러 형태의 문제로 분류될 수 있다. 여기서 화물이동경로에는 허브들을 통한 수송과 허브를 거치지 않고 기점에서 종점으로의 직접수송(direct shipment 또는 non-stop service라고 함)이 있다.

물류시스템에서 허브의 용량제한이 있는 HLP를 다룬 연구는 Hasan and Vaidyanathan(1998), Ernst and Krishnamoorthy(1999), Ebery *et al.*(2000), 그리고 Melkote and Daskin(2001) 등이 있다. 그러나 이들 연구는 모두 직접수송을 고려하지 않고 있으며 허브의 수 또한 고정된 경우를 다루고 있다. H&S 네트

이 논문은 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(R05-2004-000-10909-0)

[†] 연락저자 : 김여근 교수, 500-757 광주광역시 북구 용봉동 300 전남대학교 산업공학과, Fax : 062-530-1789, E-mail : kimyk@chonnam.ac.kr

2005년 6월 접수; 2005년 9월 수정본 접수; 2005년 10월 게재 확정.

워크 설계에 있어서 직접수송을 고려함으로써 보다 효율적인 물류네트워크를 설계할 수 있다(Lumsden, 1999). 이를 다룬 연구에는 Aykin(1995), Zpfel and Wasner(2002), 그리고 Elhedhli et al.(2005) 등이 있다. 하지만 이들 연구들에서는 허브의 용량은 무한하다고 가정하고 있으며, 또한 허브의 수가 미리 정해져 있다고 보고 있다. 이와 같은 문제를 p -HLP라 한다.

그러나 물류네트워크의 설계에 있어서 허브의 수를 하나의 결정변수로 두는 것이 보다 합리적이다(Helm and Venkataramanan, 1998). 이에 관한 연구는 Topcuoglu et al.(2005)과 Boland et al.(2003) 등이 있다. Boland et al.(2004)은 허브 용량 제한을 고려하고 있다. 그러나 이들 두 연구 모두 직접수송은 다루지 않고 있다.

본 연구에서는 H&S 네트워크 설계에 있어서 보다 현실적이고 그 적용성을 높이기 위하여 허브의 처리용량 제약이 있고, 기점과 종점 간 직접수송을 고려하면서 허브의 수를 결정변수로 두는 HLP를 다룬다. 본 연구에서 다루는 문제를 일반적으로 취급하는 HLP와 구별하기 위하여, CHLPwD(Capacitated Hub Location Problem with Direct shipment)로 표기한다.

본 연구에서는 CHLPwD의 수리모형을 제시하고, 이를 해결하기 위한 방법론으로 진화알고리즘(evolutionary algorithm: EA)을 사용한다. 진화알고리즘은 자연의 진화과정을 모방한 일종의 메타휴리스틱 탐색기법으로 복잡도가 높은 조합최적화 문제에 아주 효율적인 것으로 알려져 있다(Kim et al., 1997; Michalewicz, 1999).

본 연구에서 다루는 CHLPwD는 복잡도가 높은 일종의 조합 최적화 문제이므로 진화알고리즘을 하나의 적절한 방법론으로 기대할 수 있다. 진화알고리즘 적용을 위해 직접수송을 고려한 개체표현과 유전연산자를 개발한다. 본 연구에서는 CHLPwD에 진화알고리즘의 적용성을 보이고, 그 성능을 최적해 및 발견적 기법과 비교한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구에서 다루는 CHLPwD의 문제를 정의하고 수리모형을 제시한다. 3장에서는 CHLPwD를 위한 진화알고리즘의 구조와 절차 그리고 알고리즘의 적용을 위한 유전요소에 관하여 기술한다. 4장에서는 실험설계와 실험결과를 제시하고, 그 결과를 분석한다. 마지막으로, 5장은 결론으로 구성되어 있다.

2. 문제정의 및 수리모형

2.1 문제정의

본 연구에서 다루는 CHLPwD의 문제를 좀더 구체적으로 정의하면 다음과 같다.

첫째, CHLPwD에서 허브의 수와 위치는 총 물류비용을 최소화하기 위한 결정변수이다. 즉, 허브 후보지 중에서 적정 수의 허브를 결정하는 것이다. 허브의 수와 위치에 따라 H&S 네

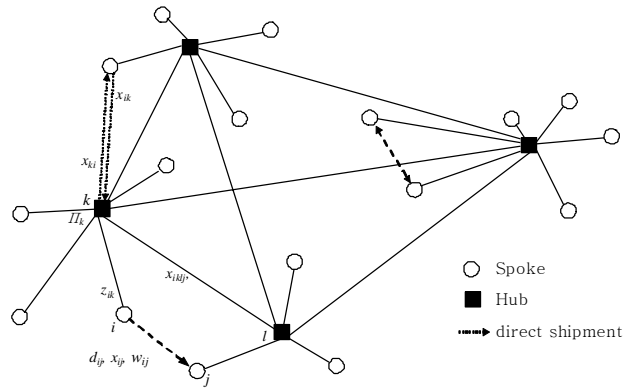


Figure 1. Hub-and-spoke network.

트워크 설계가 달라지고 네트워크 설계의 변화는 총 물류비용을 변화시키기 때문이다. 본 연구에서 허브 후보지는 모든 지점이 가능한 것으로 한다. 둘째, 허브용량이 제한되어 있다. 즉, 한 허브에서 처리할 수 있는 화물량에 제한을 둬으로써 허브에 할당될 수 있는 스포크의 수가 한정된다. 허브용량은 고객서비스의 질에 결정적일 수 있다. 한 예로, 우편물이 모이는 우편물 분류센터(hub)에서는 시간제약으로 인하여 한정된 양만의 분류가 허용되며, 허브에서 과도한 화물의 유입과 유출은 허브의 혼잡 또는 정체를 초래할 수 있다.

셋째, CHLPwD에서는 기점과 종점 간에 직접수송을 고려한다. 또한 본 연구에서는 모든 지점 간에 직접수송이 가능하다고 본다. 즉, <Figure 1>과 같이 기점 i 에서 종점 j 로 화물을 보내는 경로는 기점 i 에서 허브 k 허브 l 를 경유하여 종점 j 로 보내는 경로($i \rightarrow k \rightarrow l \rightarrow j$)와 기점 i 의 화물을 종점 j 로 직접 보내는 경로($i \rightarrow j$)가 있다. 마지막으로 각 스포크는 하나의 허브에만 할당된다고 본다.

본 연구에서는 이러한 특성과 제약을 갖는 H&S 수송네트워크에서 총 물류비용을 최소로 하는 네트워크를 설계한다. 고려하는 비용은 허브를 통한 수송비, 직접수송비, 허브설치비용이다(식 (1)). 결정변수는 허브의 수와 위치, 비 허브지점(스포크)의 허브 할당, 그리고 모든 지점들에 대한 수송경로(허브를 통한 수송 또는 직접수송)의 결정이다.

2.2 수리계획 모형

먼저, CHLPwD의 수리모형화를 위해 사용하는 변수와 기호를 정의하고, 이 문제의 0-1 정수계획 모형을 제시한다.

<데이터>

V : 지점의 집합

w_{ij} : 지점 i 와 j 간의 화물량

d_{ij} : 지점 i 와 j 간의 거리

Π_k : 허브 k 의 용량

F_k : 지점 k 에서 허브설치비용

M : 매우 큰 수

<파라미터>

λ : 직접수송과 관련된 고정비용(fixed cost)

α : 기점에서 허브까지의 수송에 관계된 비용계수

β : 허브 간의 수송에 관계된 비용계수

γ : 허브에서 종점까지의 수송에 관계된 비용계수

δ : 직접수송에 관계된 비용계수

<결정변수>

$$z_{ik} = \begin{cases} 1, & \text{지점 } i \text{가 허브 } k \text{에 할당되면,} \\ 0, & \text{그렇지 않으면} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{지점 } i \text{에서 지점 } j \text{로 직접수송되면,} \\ 0, & \text{그렇지 않으면} \end{cases}$$

$$x_{iklj} = \begin{cases} 1, & \text{지점 } i \text{에서 허브 } k \text{와 } l \text{을 경유하여 지점 } j \text{로 수송되면,} \\ 0, & \text{그렇지 않으면} \end{cases}$$

<0-1 정수계획 모형>

$$\begin{aligned} \text{Min} \quad & \sum_i \sum_{j \neq i} (\lambda + w_{ij} \delta d_{ij}) x_{ij} \\ & + \sum_i \sum_k \sum_l \sum_{j \neq i} w_{ij} (\alpha d_{ik} + \beta d_{kl} + \gamma d_{lj}) x_{iklj} + \sum_k F_k z_{kk} \end{aligned} \quad (1)$$

Subject to

$$z_{ik} \leq z_{kk} \quad \forall i, k \in V \quad (2)$$

$$\sum_k z_{ik} = 1 \quad \forall i \in V \quad (3)$$

$$\sum_{i \neq k} (w_{ik} x_{ik} + w_{ki} x_{ki}) + \sum_i \sum_{j \neq i} \left\{ w_{ij} x_{iklj} + \sum_{l \neq k} (w_{ij} x_{iklj} + w_{ji} x_{jlki}) \right\} \leq \Pi_k z_{kk} + M(1 - z_{kk}) \quad \forall k \in V \quad (4)$$

$$z_{ik} + z_{jl} - 2x_{iklj} \geq 0 \quad \forall i, j \neq i, k, l \in V \quad (5)$$

$$x_{ij} + \sum_k \sum_l x_{iklj} = 1 \quad \forall i, j \neq i \in V \quad (6)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \neq i \in V \quad (7)$$

$$x_{iklj} \in \{0, 1\} \quad \forall i, j \neq i, k, l \in V \quad (8)$$

$$z_{ik} \in \{0, 1\} \quad \forall i, k \in V \quad (9)$$

위에서 식 (1)은 목적함수로 물류 관련비용의 최소화를 나타낸다. 첫째 항은 직접수송 비용을 나타낸다. 여기서 λ 는 직접수송에 요구되는 고정비용이다. 둘째 항은 허브를 경유한 수송에 소요되는 비용을, 마지막 항은 허브 설치비용을 나타낸다. 식 (2)는 지점 i 는 지점 k 가 허브로 선정되는 경우에만 이

지점 k 에 할당(연결)될 수 있다는 제약을 의미하고, 식 (3)은 각 지점이 하나의 허브에만 할당되어야 함을 나타낸다. 식 (4)는 허브 용량과 관련된 제약식이다. 여기서 첫째 항은 허브가 기점 또는 종점이면서 직접수송되는 화물량을 나타낸다. 둘째 항은 허브를 경유한 수송과 관련해서 발생하는 화물량을 나타낸다. 이는 다시 두 가지로 분류할 수 있는데 앞부분은 기점과 종점이 모두 같은 허브에 할당된 경우이고, 뒷부분은 기점과 종점이 각각 서로 다른 허브에 할당되어 있는 경우를 나타낸다. 마지막 항, $M(1-z_{kk})$ 은 지점 k 가 허브가 아닐 때, 지점 k 가 기점이거나 종점인 지점 간의 직접수송이 허용될 수 있도록 한다. 식 (5)는 허브를 통한 수송경로는 반드시 기점과 종점이 각각 연결된 허브를 경유해야 함을 나타낸다. 또한, 각각의 모든 지점간의 수송은 직접수송 또는 허브를 통한 수송경로 중에서 이루어져야 하는데, 이는 식 (6)과 같이 표현된다. 끝으로 식 (7), (8) 그리고 (9)는 정수조건을 나타낸다.

3. CHLPwD를 위한 이웃 진화알고리즘

3.1 이웃 진화알고리즘의 구조와 절차

진화알고리즘은 여러 개의 개체로 이루어진 모집단을 운영 하는 일종의 반복적인 인공지능기법이며, 다윈의 자연선택과 멘델의 유전법칙을 모방한 일종의 확률적 탐색기법이다. 진화 알고리즘은 제약의 첨가나 목적함수의 변경이 용이하고, 해 공간의 탐색능력이 우수하여 공학과 자연과학뿐만 아니라 경제학, 사회과학에서 최적화 또는 의사결정 문제 등에 널리 적용되고 있다(Goldberg, 1989; Kim and Hyun, 1994).

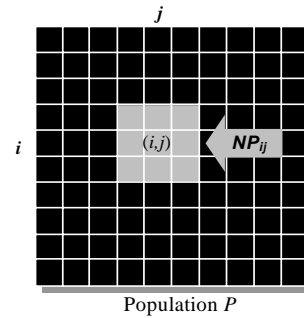


Figure 2. Structure of population.

전통적 진화알고리즘에서 전체 모집단이 진화하는 구조는 모집단 내에 높은 적응도(fitness)를 갖는 개체들로 인해 다양한 해공간을 탐색하지 못하고 조기 수렴할 가능성이 높다. 이러한 단점을 보완하기 위해 본 연구에서는 이웃 진화구조를 사용한다. 이를 위해, 모집단 P 는 토러스(torus) 형태의 2차원 정방형 격자구조를 갖는다고 본다. 이웃 크기와 형태는 다양하게 정의될 수 있으나, 본 연구에서는 (3x3) 격자구조 형태를 사용한다. 이웃 NP_{ij} 는 모집단의 위치 (i, j) 에 있는 개체와 그 주

변의 8개 개체를 나타낸다. <Figure 2>는 모집단의 구조와 이웃 NP_{ij} 를 나타내고 있다. 이러한 이웃진화구조는 모집단의 다양성 및 적소(niche) 형성에 기여하는 것으로 알려져 있다 (Pagie and Hogeweg, 1998; Kim *et al.*, 2001).

제안된 알고리즘의 절차는 다음과 같다.

단계 1: 초기화

2차원 격자구조를 갖는 모집단 P 를 구성하고, 초기해를 생성하여 격자의 각 위치에 둔다.

단계 2: 초기 적응도 평가

모든 개체들의 적응도를 평가한다.

단계 3: 이웃 설정

모집단 P 에서 임의의 위치 (i, j) 를 선택하고, 이웃 NP_{ij} 를 정의한다.

단계 4: 이웃의 진화

- 4.1: NP_{ij} 에서 적응도를 기준으로 두 개체 P_1, P_2 를 확률적으로 선택한다.
- 4.2: P_1, P_2 를 유전연산(교차와 돌연변이)하여 두 자손 o_1, o_2 를 생산하고, 적응도를 평가한다.
- 4.3: o_1, o_2 를 NP_{ij} 중 가장 낮은 적응도를 갖는 두 개체와 교체한다.
- 4.4: 이웃진화의 종료조건을 만족하면 단계 5로 가고, 그렇지 않으면 단계 4.1로 간다.

단계 5: 종료 판단

알고리즘의 종료조건을 만족하면 끝내고, 그렇지 않으면 단계 3으로 간다.

3.2 유전요소

3.2.1 개체 표현

진화알고리즘에서 개체는 자연스럽게 명확하게 해를 나타내야 하고, 가능한 해가 중복 표현되지 않아야 한다. 중복 표현은 해공간의 효율적 탐색을 방해할 수 있다. 또한, 표현방법은 유전연산자와 밀접하게 관련되어 있다. 따라서 개체는 개체가 갖는 중요한 정보가 유전연산자에 의해 추출되고 자손에게 잘 전달될 수 있도록 표현되어야 한다.

Helm and Venkataramanan(1998)과 Helm(1998)는 진화알고리즘을 이용한 허브의 수와 입지선정문제를 다루었다. Topcuoglu *et al.*(2005)은 앞의 두 연구(Helm and Venkataramanan, 1998; Helm, 1998)가 허브의 입지선정만을 하나의 염색체로 표현한 데 반하여, 입지선정을 위한 염색체와 스포크 할당을 위한 염색체로, 즉 두 개의 염색체로 개체를 표현하고 이를 유전연산하였다. 이 표현의 장점은 유전연산에서 허브의 입지선정과 스포크의 허브 할당정보를 효과적으로 자손에게 전달할 수 있다는 데 있다. 그러나 이 방법은 허브의 선정과 스포크의 허브 할당을 별도로 표현하고 있다. 이는 단일화 된 표현보다 더 많은 메모리 공간과 계산시간을 필요로 한다. 또한, 이들 세

연구 모두 허브 용량과 직접수송을 고려하지 않고 있다. 본 연구에서는 Topcuoglu *et al.*(2005)이 제안한 기법을 CHLPwD에 맞게 개선하여 사용한다.

본 연구에서 표현은 허브의 위치선정, 스포크의 허브 할당, 그리고 직접수송을 하나의 개체에 나타낸다. 개체에서 앞부분에는 허브-스포크 할당을 표현하고, 뒷부분에는 직접수송을 표현한다. 허브-스포크 표현에서 인자의 위치는 지점을 나타내고, 인자값에 의해 허브와 스포크를 구분한다. 0의 값을 갖는 위치는 허브가 설치되는 지점을, 그 외의 값을 갖는 위치는 스포크가 되고, 인자값은 해당 스포크가 연결된 허브의 위치를 나타낸다. <Figure 3>의 (1)과 (2)는 허브가 지점 2과 6에 설치되고, 스포크 1과 7은 허브 6에, 스포크 3, 4, 5, 8, 9는 허브 2에 연결됨을 나타낸다. 이러한 표현은 Topcuoglu *et al.*(2005)의 표현보다 컴퓨터 메모리와 계산시간을 줄일 수 있다는 장점이 있다.

(1)	Hub-spoke part																		
	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr> <tr><td>P_1</td><td>0</td><td></td><td></td><td></td><td>0</td><td></td><td></td><td></td></tr> </table>	1	2	3	4	5	6	7	8	9	P_1	0				0			
1	2	3	4	5	6	7	8	9											
P_1	0				0														
(2)	Hub-spoke part																		
	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr> <tr><td>P_1</td><td>6</td><td>0</td><td>2</td><td>2</td><td>2</td><td>0</td><td>6</td><td>2</td></tr> </table>	1	2	3	4	5	6	7	8	9	P_1	6	0	2	2	2	0	6	2
1	2	3	4	5	6	7	8	9											
P_1	6	0	2	2	2	0	6	2											
(3)	Hub-spoke part	Direct shipment part																	
	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr> <tr><td>P_1</td><td>6</td><td>0</td><td>2</td><td>2</td><td>2</td><td>0</td><td>6</td><td>2</td></tr> </table>	1	2	3	4	5	6	7	8	9	P_1	6	0	2	2	2	0	6	2
1	2	3	4	5	6	7	8	9											
P_1	6	0	2	2	2	0	6	2											

Figure 3. Representation of CHLPwD.

그리고 모든 지점들 중에서 직접수송되는 두 지점, (i, j) 를 사전식으로 정렬하여 뒷부분에 추가한다. <Figure 3>의 (3)은 이를 보여 주고 있다.

3.2.2 초기 모집단에서 개체의 허브 수와 위치결정

초기 모집단 생성에서 허브의 수와 위치는 각 지점의 총 화물량(유입량+유출량)과 허브의 용량에 관한 정보를 이용하여 결정한다. 이는 보다 효율적으로 해공간을 탐색하기 위해서이다.

Topcuoglu *et al.*(2005)은 초기 모집단에서 허브의 수와 위치를 결정하는 데 있어 각 지점의 총 화물량에 대한 정보를 이용하였다. 본 연구에서는 여기에 허브 용량과 설치비용정보를 추가한 다음 세 방법에 의해 초기 모집단을 구성한다.

먼저, 모집단의 30%의 개체들은 각 개체가 갖는 허브 수를 $[1, \dots, n/4]$ 범위에서 임의로 결정한다. 그리고, 각 개체의 허브 위치는 화물량 목록에서 상위 2/3 내에 있는 지점들 중에서 화물량의 크기에 비례한 확률을 각 지점에 부과하여 지정된 허브의 수만큼 확률적으로 선택한다. 여기서 n 은 지점의 수이고 화물량 목록은 모든 지점 각각의 총 화물량에 대해 내림차순으로 정렬해 놓은 목록이다. 다음은, 모집단의 30%의 개체들

에 대해 각 개체가 갖는 허브 수는 $[1, \dots, n/4]$ 범위에서 임의로 결정하고, 이에 대한 허브의 위치는 허브 비용목록에서 상위 $2/3$ 내에 있는 지점들 중에서 각 지점에 대해 허브의 용량을 설치비용으로 나눈 값이 큰 지점일수록 허브로 선정될 확률을 높게 하여 허브 수만큼 선택한다. 여기서 허브 비용목록은 허브 용량을 설치비용으로 나눈 값에 대하여 내림차순으로 정렬해 놓은 목록이다. 이 두 방법은 강한 임의성을 탈피하여 해의 탐색 능력을 강화하기 위함이다. 마지막으로, 모집단의 40%에 해당하는 개체에 대해 각 개체가 갖는 허브 수는 $[n/4, \dots, n/2]$ 범위에서 임의로 결정하고, 이에 대한 허브의 위치는 전체 지점들의 집합에서 임의로 결정한다. 이 방법은 다양한 해를 생성하여 해의 조기 수렴을 방지하기 위해서이다.

3.2.3 적응도 평가와 선택

최적화 문제에서 적응도는 흔히 목적함수에 의해 평가된다. 그러나 본 연구에서 다루는 CHLPwD는 다양한 제약들을 포함하고 있다. 다루는 문제의 복잡성으로 인하여 유전표현과 유전연산에서 모든 제약을 만족하는 해의 생산이 어려운 경우가 흔히 발생한다. 본 연구에서는 제약들 중에서 개체표현과 유전연산에 의해 항상 만족시킬 수 있는 제약을 직접제약이라고 하고 그렇지 않은 제약을 간접제약이라 부르도록 한다. 본 연구에서 직접제약은 허브와 스포크의 할당과 관련된 제약이고, 간접제약은 허브의 용량제약이다. 간접 제약을 만족하는 해는 목적함수가 곧 적응도가 된다. 하지만 허브의 용량을 초과하는 화물량이 허브에 부과되면 평가함수에 벌금을 부과하는 방법을 사용한다. 즉, 개체의 평가 결과 나타난 수송비용과 허브 설치비용의 합을 Q_{sum} 이라 두자. 그러면, Q_{sum} 은 식 (1)과 같고, 여기에 허브 용량제약의 위반여부에 따라 벌금을 부과하여 적응도를 평가한다. 본 연구에서 사용되는 개체의 적응도 평가함수, $eval$ 는 다음과 같다.

$$eval = Q_{sum} + c \cdot \sum_{k, k \in H} PW_k^\theta \quad (10)$$

여기서, H 는 허브의 집합이고 $PW_k = \max(HW_k - \Pi_k, 0)$ 이다. 또한, HW_k 는 허브 k 에서 처리해야 할 화물량을 나타내고 상수 c 와 θ 는 파라미터이다. 이러한 적응도를 기준으로 정의된 이웃 NP_{ij} 에서 유전연산에 참여할 두 부모(개체)를 확률적으로 선택한다. 선택된 부모는 유전연산을 통하여 두 자손을 생산하고 적응도를 평가한 다음 NP_{ij} 중 가장 낮은 적응도를 갖는 두 개체와 교체시켜 준다. 이러한 이웃진화 과정을 이웃진화 종료조건이 만족할 때까지 반복한다. 이렇게 몇 개의 개체를 반복적으로 재생산하여 이들 개체를 적응도가 낮은 개체와 대체하는 안정상태 진화전략을 사용한다. 이러한 안정상태 진화 전략은 해의 탐색 효율을 높이는 것으로 알려져 있다.

3.2.4 유전연산자

유전연산자에는 크게 교차(crossover)와 돌연변이(mutation)

가 있다.

1) 교차

교차는 두 부모의 좋은 특성을 자손에게 효과적으로 상속할 수 있어야 한다. CHLPwD에서 교차는 부모가 갖는 허브의 위치와 스포크의 할당 정보 그리고 직접수송경로에 대한 정보를 자손에게 적절히 상속시켜 주어야 한다. 또한, 직접수송은 허브의 용량제한과 밀접한 관계를 갖는다. 따라서 이들도 함께 고려해 주어야 한다.

직접수송은 허브의 용량부하를 덜어준다. 그리고, 직접수송은 스포크의 허브할당에 영향을 주고, 스포크의 할당 위치에 따라 허브를 경유한 수송비용이 달라지게 된다. 즉, 이는 모든 기-중점 간의 수송경로 결정에 영향을 준다. 따라서 CHLPwD를 효율적으로 해결하기 위해서는 허브의 용량제한과 직접수송이 서로 연관되어 동작할 수 있는 교차연산자를 필요로 한다. 이를 위해 본 연구에서 제안한 교차연산의 절차는 다음과 같다.

단계 1: 허브-스포크 표현부분에서 범위 $[1, n]$ 에서 임의의 교차점 r 을 선택한다.

단계 2: 허브-스포크 표현부분에 대해 일점교차를 한다.

단계 3: 교차 후 스포크 중에서 허브가 아닌 곳에 할당된 인자들을 제거하고 이들 인자를 미할당 인자(*)로 표기한다.

단계 4: 직접수송 표현부분에서 지점들의 쌍 (i, j) 에 대해 $i = r-1$ 인 지점을 포함하는 부분까지를 교차점으로 선택한다.

단계 5: 직접수송 표현부분에 대해 일점교차를 한다.

단계 6: 허브-스포크 표현부분의 미할당 인자(*)들에 대해 재할당 기법(3.2.5절 참조)을 이용하여 허브에 할당한다.

단계 7: 할당된 허브-스포크 표현부분을 기준으로 직접수송 표현부분에서 직접수송이 허브를 경유한 수송보다 더 큰 비용을 갖는 지점들의 쌍을 제거한다.

단계 8: 단계 3에서 미할당 인자(*)들이 있는 지점으로 들어 오고 나가는 모든 수송경로 중에서 허브를 경유한 수송에 비해 직접수송 비용이 더 저렴한 쌍을 직접수송 표현부분에 추가한다.

허브-스포크 표현부분과 직접수송 표현부분은 모두 일점교차를 한다. 먼저 단계 1과 2의 과정을 통해 허브-스포크 표현부분에 있는 부모의 정보를 자손에게 상속한다(만약, 허브가 하나도 없는 자손이 생성되면 새롭게 하나의 개체를 만들어 자손으로 대체한다). 단계 3에서 허브가 아닌 지점에 할당된 인자를 포함하는 개체들은 비가능해이므로, 해당 인자값을 제거하고 차후 가능해로 만들기 위해 이들을 미할당 인자(*)로 표기한다. 단계 4에서는 직접수송 표현부분에 대한 교차점을 선택한다. 이때, 직접수송 표현부분에서 교차점의 위치는 허브-스포크 표현부분에서의 교차점에 따라 결정된다. 즉, 허브-스포크 표현부분에서 사용한 교차점의 위치가 r 이었다면 직접

수송 표현부분에서의 교차점 위치는 직접수송 표현부분의 (i, j) 에서 i 는 $r-1$ 을 포함하는 부분까지를 교차점으로 사용한다. 이는 허브-스포크 표현부분에서 자손에게 상속한 허브 위치와 스포크 할당정보를 직접수송 표현부분에서도 이와 관련된 직접수송 정보를 자손에게 상속시키기 위함이다. 이를 토대로 단계 5에서는 직접수송 표현부분에 대하여 교차를 수행한다. <Figure 4>에서 (1)과 (2)는 위에서 설명한 교차연산의 과정 중 단계 1에서 5까지를 나타낸 것이다.

단계 6에서는 미할당 인자들에 대해 재할당 기법을 이용하여 허브에 할당한다. 앞에서 언급했듯이 재할당 기법은 미할당 인자(*)에 대해, 기회비용의 개념을 이용하여 현재 직접수송 표현부분의 직접수송과 허브용량을 고려하면서 비용이 가장 저렴한 허브에 할당하는 것이다. 즉, 직접수송은 허브의 용량제약을 완화시킨다. 이는 미할당 인자가 여러 허브들 중에서 보다 좋은 위치의 허브에 할당될 수 있는 기회를 제공한다. 예로 <Figure 4>의 (2)에서 자손 o_1 의 미할당 지점 1을 허브 2에 할당할 때, <Figure 4>의 (3)의 직접수송 표현부분 (1,3), (1,5), (3,1), (4,1), (5,2)는 허브 2의 용량부하를 덜어주어 미할당 지점 1이 허브 2에 할당되는 것을 돕는다. 여기서 (1,3), (1,5), (3,1), (4,1)은 미할당 지점 1의 화물량이고 (5,2)는 허브 2

에서 처리해야 하는 화물량이다. 단계 7에서는 새롭게 만들어진 허브-스포크 표현부분을 기준으로 직접수송 표현부분에서 허브를 경유하는 수송보다 직접수송 비용이 더 큰 두 지점의 쌍을 제거한다(<Figure 4>의 (4)). 이때, 직접수송 비용은 허브와 스포크의 위치에 상관없이 항상 동일하므로 데이터로 저장하여 필요한 경우에 호출하여 사용할 수 있다. 따라서 이들 쌍들과 관련된 허브를 경유한 수송비용만을 구하여 비교하면 된다.

마지막으로 단계 8에서는 허브-스포크 표현부분에서 미할당 지점(단계 3에서 미할당 인자*)들에 대해 그 지점들을 통하여 들어오고 나가는 모든 수송경로들 중 허브를 경유한 수송에 비해 직접수송 비용이 더 작은 지점들의 쌍을 직접수송 표현부분에 추가한다(<Figure 4>의 (5)).

2) 돌연변이

진화알고리즘에서 돌연변이는 해공간의 다양성을 유지하고 부분최적에 조기 수렴하는 것을 방지하는 역할을 한다. 따라서 CHLPwD에서 돌연변이 연산자는 해의 구조를 다양하게 변화시킬 수 있어야 한다. 본 연구에서 사용한 돌연변이 연산자는 다음과 같다.

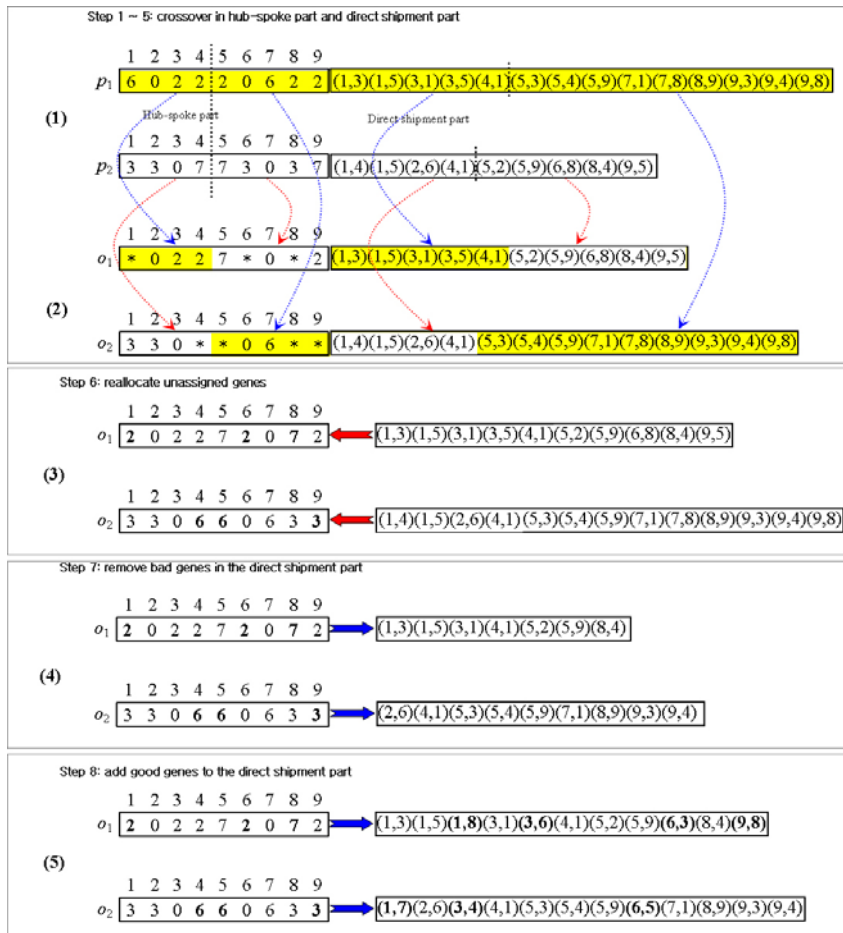


Figure 4. Crossover.

- (1) 스포크교환(SwapNode): 두 개의 허브에서 각 허브에 할당된 스포크를 임의로 하나씩 선택하여 허브를 교환하여 할당한다.
- (2) 스포크 재할당(ReallocateNode): 임의로 하나의 스포크를 선택하여 다른 허브에 할당한다.
- (3) 허브교체(RelocateHub): 임의의 스포크 s 를 새로이 허브로 만들고, 이 스포크 s 가 연결된 허브와 이 허브에 연결된 모든 스포크를 새로운 허브에 할당(연결)시킨다.
- (4) 허브생성(NewHub): 임의의 스포크 s 를 새로이 허브로 만들고 이 스포크 s 가 연결된 기존 허브에 할당되어 있는 모든 스포크들을 기존 허브와 새로운 허브에 재할당한다(재할당은 3.2.5절의 기법을 사용함).
- (5) 허브제거(DeleteHub): 허브들 중에서 임의로 하나의 허브를 선택하여 제거하고 그 허브에 할당되어 있던 스포크들은 다른 허브들에 할당한다(재할당은 3.2.5절의 기법을 사용함).

개체 돌연변이율에 따라 개체의 돌연변이 여부를 결정하고 위의 5가지 돌연변이 연산자 중 임의로 하나를 선택하여 사용한다. 5가지 돌연변이 연산자들 모두 허브-스포크 표현부분에만 적용된다. 이는 다양한 해공간을 탐색하는 데 있어 직접수송 표현부분은 큰 영향을 미치지 않기 때문이다. 여기서 돌연변이 연산자 ‘스포크교환’과 ‘스포크재할당’은 허브의 위치와 개수에는 변화를 주지 않는 반면, ‘허브교체’, ‘허브생성’, ‘허브제거’ 연산자는 허브의 위치나 수를 변화시켜서 새로운 해공간의 탐색을 도모한다.

3.2.5 재할당 기법

전통적인 진화알고리즘은 강한 제약을 갖는 문제에 적용할 경우, 진화알고리즘의 내재된 임의성에 의해 낮은 탐색성능을 보일 수 있다(Falkenauer, 1994). 이러한 문제에서 알고리즘의 탐색성능 향상을 위해 문제에 적합한 발견적 기법이 흔히 사용된다(Eiben et al., 1995). 본 연구에서는 제안하는 알고리즘에 CHLPwD가 갖는 문제의 특성을 이용한 발견적 기법을 결합하여 사용한다. 재할당에 사용되는 기호 정의와 그 절차는 다음과 같다.

<기호 정의>

- H : 설치된 허브들의 집합
- US : 미할당된 스포크들의 집합
- AS : 할당된 스포크들의 집합
- HS_k : 허브 k 에 할당된 스포크의 집합
- HW_k : 이미 허브 k 에 할당된 화물량
- DW_i : 지점 i 의 직접수송 화물량

$$(DW_i = \sum_{j \in AS, j \neq i} (w_{ij}x_{ij} + w_{ji}x_{ji}))$$

- W_i : 지점 i 의 허브를 경유하여 수송되는 화물량

$$(W_i = \sum_{j \in AS, j \neq i} (w_{ij} + w_{ji}) - DW_i)$$

AC_{ik} : 미할당 지점 i 를 허브 $k, k \in H$ 에 할당할 때, 지점 i 에서 다른 모든 지점 j 로 유입·유출되는 화물량에 따른 수송 비용

$$(AC_{ik} = \sum_{j \in AS, j \neq i} (w_{ij} + w_{ji})(\alpha d_{ik} + \beta d_{ki} + \gamma d_{ij})x_{ikj} + \sum_{j \in AS, j \neq i} (\lambda + w_{ij}\delta d_{ji})x_{ji}).$$

\hat{k}_i : 미할당 지점 i 를 허브에 할당하고자 할 때, 가장 적은 수송비용이 발생하는 허브. 즉, $\hat{k}_i = \arg \min_{k \in H} AC_{ik}$

<재할당 절차>

단계 1: $HS_k, HW_k, k \in H$ 를 구한다. 그리고 $TS \leftarrow US$ 로 둔다.

단계 2: TS 에서 임의로 스포크 i^* 를 선택한다.

2.1 허브 $k, k \in H$ 에 대해 AC_{i^*k} 를 계산한다. 이 때, TS 에서 i^* 를 뺀 나머지 스포크들은 가장 가까운 허브에 할당되었다고 가정한다.

2.2 $k1 = \arg \min_{k \in H} AC_{i^*k}$ 과 $k2 = \arg \min_{k \in H, k \neq k1} AC_{i^*k}$ 를 구하고

$\hat{k}_{i^*} = k1$ 으로 둔다. $O_{i^*} = AC_{i^*k2} - AC_{i^*k1}$ 를 계산한다.

단계 3: $TS \leftarrow TS - \{i^*\}$ 로 둔다. $TS \neq \emptyset$ 이면 단계 2로 간다.

단계 4: $UA = \emptyset$ 로 둔다.

4.1 $i' = \arg \max_{i \in US} O_i$ 인 i' 를 구한다.

4.2 $HW_{\hat{k}_{i'}} + W_{i'} \leq \pi_{\hat{k}_{i'}}$ 이면, $HS_{\hat{k}_{i'}} \leftarrow HS_{\hat{k}_{i'}} \cup \{i'\}$, $HW_{\hat{k}_{i'}} \leftarrow HW_{\hat{k}_{i'}} + W_{i'}$ 로 두고, 그렇지 않으면 $UA \leftarrow UA \cup \{i'\}$ 로 둔다.

단계 5: $US \leftarrow US - \{i'\}$ 로 둔다. $US \neq \emptyset$ 이면 단계 4.1로 간다.

단계 6: $US \neq \emptyset$ 이면 종료한다.

6.1 UA 에서 $W_i, i \in UA$ 가 가장 큰 스포크 i 를 선택하고 스포크 i 의 화물량 W_i 를 수용할 수 있는 가장 가까운 허브에 할당한다.

6.2 화물량 W_i 를 수용할 수 있는 허브가 존재하지 않으면, 가장 가까운 허브에 할당한다. 그리고 $UA \leftarrow UA - \{i\}$ 로 두고, 단계 6.1로 간다.

앞서 언급했듯이 미할당 스포크의 할당 기법은 기회비용의 개념을 사용한 것이다. 미할당 스포크에 대해, 이 스포크를 각 허브에 할당하여 가장 적은 비용(식 (1)에 표현된 비용)과 그 다음 적은 비용의 차를 구한다(단계 2와 단계 3). 비용을 구할 때 다른 미할당 스포크는 가장 가까운 허브에 할당된 것으로 본다. 이 비용 차이가 가장 큰 스포크부터 차례로 용량제한을 만족하면 허브에 할당하고 그렇지 않으면 UA 에 놓는다(단계 4와 5).

UA 에 있는 스포크들(허브 용량제약을 만족하지 못한 스포크들)은 화물량이 가장 큰 지점부터 차례로 이 화물량을 수용할 수 있는 가장 가까운 허브에 할당한다(단계 6.1). 만약 화물량을 수용할 수 있는 허브가 존재하지 않으면 가장 가까운 허브에 할당한다(단계 6.2). 위와 같은 과정을 모든 미할당 인자 값이 모두 할당될 때까지 반복한다.

4. 실험과 결과

4.1 실험설계

본 연구에서 다루는 문제를 실험하기 위해 사용된 데이터는 OR-Library(Beasley, 1990)의 AP data이다. AP data는 Australia Post의 실제 허브 위치선정문제에 관한 것으로, 200개의 지점들로 구성되어 있다. 실험에 사용된 문제의 크기(지점 수) $n=10, 20, 25, 40, 50, 100, 200$ 은 원래의 데이터 집합으로부터 얻을 수 있으며, 이들 각각의 문제는 각 지점에 대한 허브 설치비용과 허브용량을 포함하고 있다. 또한, 허브 설치비용과 허브용량은 각각 두 경우로 나누어진다. 허브 설치비용은 낮음(low(L))과 높음(high(H))으로, 허브용량 제약은 느슨함(loose(L))과 강함(tight(T))으로 분류된다. 허브 설치비용과 허브용량에 따른 문제의 유형은 LL, LT, HL, HT의 네 가지로 구분한다. 그리고 허브 경우 수송에서 수집(collection), 수송(transfer), 분배(distribution) 비용에 관계된 계수들은 각각 $a=3, \beta=0.75, \gamma=2$ 로 주어진다.

본 연구에서 제안한 EA의 알고리즘 탐색성능은 두 측면에서 비교한다. 절대적 평가를 위해 앞의 2.2절에서 제시한 수리모형에 의해 구한 최적해와 비교하며 또한, 상대적 평가를 위해 발견적 기법과 비교한다.

발견적 기법은 Klincewicz(1991)이 제안한 기법을 CHLPwD

에 맞게 변형한 것이다. 즉, p -HLP에 적용된 기법을 허브의 용량제한과 직접수송을 함께 고려할 수 있도록 수정하였다. 이 기법은 <부록>을 참조할 수 있다.

모든 알고리즘은 JAVA언어로 구현되었으며, 3.2GHz Pentium CPU를 장착한 IBM-PC에서 수행되었다. 유전 파라미터들은 예비실험을 통하여 결정하였다. 첫째, 모집단의 크기는 $100(10 \times 10$ 의 격자구조)으로 하였다. 둘째, 평가함수식 (10)에서 $c=1000, \theta=0.1$ 로 두었다. 셋째, 돌연변이는 인자단위가 아닌 개체단위로 이루어지고 그율은 0.1로 두었다. 사용되는 돌연변이 연산자의 선택은 3.2.4절에서 언급한 다섯 연산자 중에서 임의로 선택하였다. 넷째, 종료조건은 재생산 개체수로 하였고, 각 문제마다 해공간의 차이를 반영하기 위하여 다르게 주었다. 마지막으로, 직접수송에 요구되는 고정비용 g 는 문제의 크기에 따라 1,000에서 3,000 사이의 값을 사용하였다.

4.2 진화알고리즘의 성능평가

제안한 EA의 성능평가를 위해서 먼저 0-1 정수계획모형을 이용하여 구한 최적해와 비교하였다. 여기서 수리모형의 해는 CPLEX 7.5를 사용하여 구하였다.

<Table 1>은 EA와 최적해의 비교를 위한 실험결과이다. 비교적 작은 크기의 문제이지만 각 지점의 허브용량과 설치비용에 따른 제약의 강약을 표현할 수 있도록 실험을 설계하였다.

Table 1. Comparison of CPLEX and the proposed EA: $n=10$

Problem	Solution	CPLEX	Proposed EA
10LL	Opt Sol	206456.6	206456.6
	Opt hubs	4, 7	4, 7
	Opt allocate	(4, 4, 4, 4, 7, 4, 7, 7, 7, 7)	(4, 4, 4, 4, 7, 4, 7, 7, 7, 7)
	Opt DS pairs	(1, 2) (1, 3) (3, 1) (3, 5) (8, 6) (8, 10) (10, 8) (1, 2) (1, 3) (3, 1) (3, 5) (8, 6) (8, 10) (10, 8)	(1, 2) (1, 3) (3, 1) (3, 5) (8, 6) (8, 10) (10, 8)
	Time	13.52 sec	0.84 sec
10LT	Opt Sol	223190.1	223190.1
	Opt hubs	5, 6	5, 6
	Opt allocate	(5, 6, 5, 6, 5, 6, 5, 6, 5, 6)	(5, 6, 5, 6, 5, 6, 5, 6, 5, 6)
	Opt DS pairs	(7, 10), (8, 7), (8, 10), (10, 7)	(7, 10), (8, 7), (8, 10), (10, 7)
	Time	150.05 sec	1.11 sec
10HL	Opt sol	223462.0	223462.0
	Opt hubs	4, 5	4, 5
	Opt allocate	(5, 4, 5, 4, 5, 4, 5, 5, 5, 5)	(5, 4, 5, 4, 5, 4, 5, 5, 5, 5)
	Opt DS pairs	(7, 10), (8, 7), (8, 10), (10, 7)	(7, 10), (8, 7), (8, 10), (10, 7)
	Time	106.14 sec	1.48 sec
10HT	Opt sol	223462.0	223462.0
	Opt hubs	4, 5	4, 5
	Opt allocate	(5, 4, 5, 4, 5, 4, 5, 5, 5, 5)	(5, 4, 5, 4, 5, 4, 5, 5, 5, 5)
	Opt DS pairs	(7, 10), (8, 7), (8, 10), (10, 7)	(7, 10), (8, 7), (8, 10), (10, 7)
	Time	98.36 sec	3.77 sec

Problem의 nAB에서 n은 지점의 수를, A와 B는 각각 허브 설치 비용과 허브용량에 대한 문제의 분류를 나타낸다. <Table 1>의 결과값들은 보다 쉽게 이해할 수 있도록 표현의 결과를 해석하여 허브의 위치(Opt hubs)와 스포크의 허브 할당(Opt allocate)으로 나누어 나타내었다. 이때, 허브인 지점은 자신의 허브에 할당된다. 그리고 표에서 Opt DS pairs는 직접수송되는 두 지점들을 나타낸다.

<Table 1>에서 CPLEX 및 EA의 결과값은 모든 계약을 만족한 해들이고, EA의 결과값은 10회 반복 실험하여 얻은 가장 좋은 해를 나타낸 것이다. 실험결과, EA는 모든 문제에서 최적해와 동일한 해를 찾았으며 계산시간은 현저히 적게 소요됨을 확인할 수 있다. 이로부터 제안한 EA가 다루는 문제에 적합하게 설계되었음을 확인할 수 있다. $n > 12$ 인 경우에는 컴퓨터 계산시간으로 인해 최적해를 구할 수 없었다.

Table 2. Performance comparison of heuristic and the proposed EA: n=20, 25, 40

Problem	Solution	Heuristic	Proposed EA
20LL	Cost	230908.3	220447.7
	Hubs	10, 14	7, 14
20HL	Cost	265788.6	251549.1
	Hubs	10, 19	10
20LT	Cost	274850.0	259481.6
	Hubs	6, 9, 14, 15	9, 10, 14
20HT	Cost	340480.4	311468.1
	Hubs	1, 5, 10, 12	6, 8, 10
25LL	Cost	245641.5	233090.8
	Hubs	8, 14, 19	8, 23
25HL	Cost	297779.8	292132.5
	Hubs	9, 23	14
25LT	Cost	306768.7	284672.0
	Hubs	4, 6, 9, 13	6, 9, 12, 14
25HT	Cost	379831.9	363418.1
	Hubs	9, 10, 12	6, 9, 14, 25
40LL	Cost	265362.6	233630.1
	Hubs	12, 15, 36	14, 29
40HL	Cost	319770.0	297437.4
	Hubs	14, 29	14, 36
40LT	Cost	368056.9	316490.3
	Hubs	6, 9, 10, 25, 33, 40	10, 12, 14, 40
40HT	Cost	549567.3	422587.9
	Hubs	10, 12, 13, 25, 40	6, 14, 19, 25, 40

규모가 큰 문제에 대해 제안한 EA의 성능은 본 연구에서 수정하여 새롭게 만든 발견적 기법(부록 참조)과 비교하여 평가하였다. <Table 2>과 <Table 3>는 이 결과를 보여 주고 있다. 실험결과, 모든 문제에서 제안한 EA는 heuristic 기법보다 상대적으로 좋은 결과를 보였다. 특히, 문제의 규모가 크고 허브 용량 제약이 강할수록 제안한 EA의 성능이 우수하였다. 또한, 허브 설치비용이 높을 때 허브 수가 적어지는 결과를 보였다. 제안한 EA의 계산시간은 n=20, 25, 40일 때, 각각 약 8, 15, 80초가

소요되고, n=100, 200일 때, 각각 약 500, 1800초가 소요되는데 허브 용량제약이 강할 때 다소 더 많은 시간이 소요되었다. 이는 강한 허브 용량제약은 스포크의 허브 할당을 보다 어렵게 하기 때문으로 보인다. <Table 1>, <Table 2> 그리고 <Table 3>의 결과를 토대로 CHLPwD를 해결하는 데 있어 해공간의 크기나 문제의 복잡도와 상관없이 제안한 EA는 합리적인 시간 내에 좋은 해의 탐색능력을 가지고 있다 할 수 있다.

Table 3. Performance comparison of heuristic and the proposed EA: n=100, 200

Problem	Solution	Heuristic	Proposed EA
100LL	Cost	257038.1	228419.9
	Hubs	29, 95	28, 92
100HL	Cost	340264.2	286095.1
	Hubs	23, 28, 76	28, 92
100LT	Cost	398218.7	315377.8
	Hubs	12, 28, 30, 32, 33, 41, 96	21, 30, 51, 59, 64, 76, 91
100HT	Cost	726535.2	509023.0
	Hubs	8, 19, 41, 43, 82	26, 30, 40, 76
200LL	Cost	267182.7	233793.5
	Hubs	9, 104, 125, 141	53, 145
200HL	Cost	316339.1	280778.4
	Hubs	53, 91, 141, 191	91, 145
200LT	Cost	438373.2	285708.8
	Hubs	9, 56, 109, 132, 141, 143, 170	26, 59, 109, 130, 140, 191
200HT	Cost	915013.7	447823.1
	Hubs	49, 69, 88, 102, 106, 169, 171, 182	9, 91, 109, 126, 180, 191, 198

<Figure 5>는 실험문제 40HT와 200HT의 진화과정을 보인 것이다. 이 그림에서 알 수 있듯이 진화 초기에 상당히 좋은 해를 찾음을 알 수 있다. 이는 3.2.2절에서 언급한 초기 모집단에서 개체의 허브 수와 위치결정 방법에 기인한 것으로 판단된다.

4.3 발견적 정보이용의 효과분석

제안한 EA에서 다루는 문제가 갖는 특정 정보는 초기 모집단 구성과 교차의 재할당 기법에서 이용되고 있다. 이러한 정보의 이용이 EA의 탐색성능에 미치는 영향을 분석하고자 한다.

먼저 초기 모집단 구성에서 모집단을 구성하는 각 개체에 대해 3.2.2절에서 언급한 방법으로 허브의 수와 위치를 결정하는 경우(UsedInfo)와 임의로 허브의 수와 위치를 결정하는 경우(Random)를 비교한 것이다. <Table 4>는 이에 대한 결과를 보여준다. 여기서 ‘Best’, ‘Mean’, ‘Std’는 각각 실험문제마다 10회 반복실험을 통해 얻은 값에 대한 가장 좋은 값, 평균값, 그리고 표준편차를 의미한다. 마지막 열은 각 문제마다 비교되는 두 방법에 대한 개선율을 보인 것이다. 개선율은 {(Random

의 Mean - UsedInfo의 Mean) / Random의 Mean} × 100(%)로 계산하였다. 실험결과, 모든 실험문제에서 초기해를 생성할 때 정보를 이용하는 것이 좋은 결과를 보였다. 문제의 크기와 허브의 용량제약이 클수록 정보를 이용한 초기 모집단 구성이

정보를 이용하지 않는 경우보다 해를 효과적으로 탐색함을 알 수 있으며, 표준편차도 낮은 수준을 보임으로써 안정적으로 해를 탐색함을 알 수 있다.

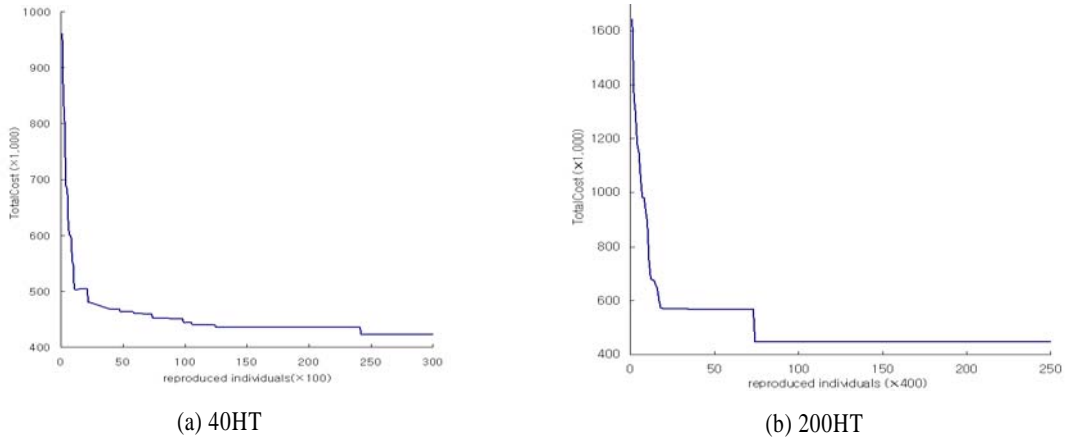


Figure 5. Evolution speed of the proposed EA.

Table 4. Comparison of methods for generating initial population

Problem	Random			UsedInfo			Improved rate(%)
	Best	Mean	Std	Best	Mean	Std	
20LL	221572.0	225051.9	1739.96	220447.7	222507.9	1030.12	1.13
20HT	313028.5	323703.6	5337.56	311468.1	319167.0	3849.45	1.40
25LL	233090.8	234304.7	606.97	233090.8	233493.6	201.41	0.34
25HT	381780.0	405008.3	11614.19	363418.1	371925.5	4253.64	8.17
40LL	236545.7	240864.6	2159.49	233630.1	235621.8	995.86	2.17
40HT	558095.8	607608.6	24756.39	422587.9	439128.4	8270.25	27.72
100LL	262762.8	273943.3	5590.26	228419.9	229961.7	770.91	16.05
100HT	608265.4	624315.8	8025.16	509022.9	526359.4	8668.23	15.69
200LL	243275.3	254647.4	5686.08	233793.5	238463.7	2335.11	6.35
200HT	711543.2	769449.9	28953.31	447823.1	469344.5	3760.71	39.00

Table 5. Comparison of rearrangement methods

Problem	H&V Method			Proposed Method			Improved rate(%)
	Best	Mean	Std	Best	Mean	Std	
20LL	221572.0	224084.9	1256.43	220447.7	222507.9	1030.12	0.70
20HT	320140.4	326464.8	3162.20	311468.1	319167.0	3849.45	2.24
25LL	233090.8	234149.5	529.38	233090.8	233493.6	201.41	0.28
25HT	370343.2	376210.1	2933.46	363418.1	371925.5	4253.64	1.14
40LL	233630.1	240980.7	3675.31	233630.1	235621.8	995.86	2.22
40HT	531796.3	549247.9	8725.84	422587.9	439128.4	8270.25	20.45
100LL	228914.3	233823.8	2454.78	228419.9	229961.7	770.91	1.65
100HT	515966.9	542572.7	13302.93	509022.9	526359.4	8668.23	2.99
200LL	243275.3	246796.6	1760.67	233793.5	238463.7	2335.11	3.38
200HT	609610.8	645820.6	18104.82	447823.1	469344.5	3760.71	29.49

다음은 재할당 기법에 대한 분석이다. 재할당 기법에 대한 비교는 3.2.5절에서 제안한 재할당 기법과 Helm and Venkataramanan(1998)이 제안한 방법을 CHLPwD에 맞게 수정한 기법(H&V 기법)과 비교한다. 수정 H&V 기법은 미할당 스포크 중에서 화물량이 가장 큰 것부터 차례로 허브 용량제한을 만족하는 가장 가까운 허브에 할당한다. 만약, 이러한 허브가 없으면 가장 가까운 허브에 할당하는 방법이다.

<Table 5>는 재할당 기법에 대한 실험결과이다. 실험결과, 본 연구에서 제안한 기법은 H&V 기법에 비해 ‘Best’, ‘Mean’ 모두에서 더 좋은 탐색 성능을 보였다. 이는 미할당 지점을 허브에 할당할 때, H&V 기법은 화물량이 가장 큰 지점부터 차례로 가장 가까운 허브에 할당하는 반면, 제안한 기법은 3.2.5절에서 언급한 기회비용의 개념을 이용하여 이 비용이 큰 것부

터 차례로 할당하는 것의 차이로 판단된다. 반면, 반복실험에 대한 표준편차는 그리 큰 차이를 보이지 않았다.

4.4 돌연변이 연산자의 성능분석

본 절에서는 돌연변이 연산자가 알고리즘의 성능에 미치는 영향에 대하여 비교, 분석한다. <Table 6>은 본 연구에서 사용한 각각의 돌연변이 연산자들이 해의 탐색에 어떤 영향을 미치는지에 대해 비교한 것이다.

비교는 3.2.4절의 2)에 제시된 돌연변이 연산자를 각기 단독으로 사용할 때와 5가지 연산자를 혼합하여 사용할 경우(‘Mixed’ 로 표기)에 대한 것이다.

<Table 6>에서 보는 바와 같이 문제의 크기와 허브 용량제한

Table 6. Effect of mutation operators on performance

Problem	SwapNode		ReallocateNode		RelocateHub		NewHub		DeleteHub		Mixed	
	Best	Mean	Best	Mean	Best	Mean	Best	Mean	Best	Mean	Best	Mean
20LL	220447.7	228567.8	220587.9	229541.8	220447.7	224360.4	220447.7	230055.3	220447.7	231584.2	220447.7	222507.9
20HT	333041.9	350842.2	330072.6	348920.6	333541.8	352884.8	339143.8	359922.5	321604.3	349022.3	311468.1	319167.0
40LL	238509.5	250263.0	233630.1	248515.6	234919.9	251159.2	240517.1	254783.4	242424.8	255513.2	233630.1	235621.8
40HT	445687.8	458395.4	454134.8	461109.6	439586.0	441255.9	459124.9	488255.1	457975.5	485988.5	438478.6	439128.4
100LL	265668.9	276778.8	228942.7	234155.6	228855.0	232233.3	249915.8	251238.9	237009.7	244484.5	228419.9	229961.7
100HT	571844.4	599666.7	579350.1	610008.7	555237.1	575779.4	604147.3	633574.7	623440.1	647724.4	509022.9	526359.4
200LL	341565.5	379994.2	234002.0	251583.4	246529.6	249282.4	265097.2	271445.5	270644.7	276988.9	233793.5	238463.7
200HT	557358.7	610441.5	510958.6	553224.3	461513.9	498117.6	542071.5	590003.8	563171.9	621110.7	447823.1	469344.5

Table 7. Effect of direct shipment

Problem	Not considered directshipment		Considered direct shipment		Improved rate(%)
	Best	Mean	Best	Mean	
20LL	222278.2	239925.1	220447.7	222107.9	7.43
20HL	259985.7	270008.3	251549.1	255213.6	5.48
20LT	267852.6	294142.5	259481.6	265330.7	9.80
20HT	331139.7	347241.3	311468.1	319167.0	8.08
40LL	242603.8	253586.3	233630.1	235621.8	7.08
40HL	312546.3	324975.1	297437.4	301116.2	7.34
40LT	348219.3	370083.4	316490.3	328237.1	11.31
40HT	496658.6	511001.2	422587.9	439128.4	14.07
100LL	234297.1	235792.7	228419.9	229961.7	2.47
100HL	313244.5	310082.4	286095.1	290027.3	6.47
100LT	457660.9	471167.8	315377.8	333116.0	29.30
100HT	745708.3	760211.3	509022.9	526359.4	30.76
200LL	242858.9	247703.6	233793.5	238463.7	3.73
200HL	302112.1	308331.0	280778.4	289727.9	6.03
200LT	377989.6	389721.0	285708.8	314177.5	19.38
200HT	547428.4	562077.7	447823.1	469344.5	16.50

이 커질수록 돌연변이 연산자가 알고리즘의 성능에 미치는 영향도 커짐을 알 수 있다. 또한, 하나의 돌연변이 연산자만을 사용하는 것 보다 혼합하여 사용하는 것이 'Best', 'Mean'에서 모두 보다 좋은 해를 보였다. 반면, 각각의 돌연변이 연산자에 대한 성능은 그리 큰 차이를 보이지 않았다. 단, 문제의 크기가 크고 허브용량 제한이 큰 경우에 'RelocateHub' 연산자가 다른 연산자들에 비해 다소 더 좋은 성능을 보였다. 이는 이 연산자가 네트워크 전체에는 많은 변화를 주지 않으면서 허브와 스포크의 위치를 적절히 바꿔 주기 때문인 것으로 판단된다.

4.5 직접수송의 영향 분석

직접수송이 네트워크 설계에 미치는 영향에 대하여 살펴보자. <Table 7>로 부터 직접수송은 허브 설치비용에는 많은 영향을 미치지 않으나 허브용량 제한에는 많은 영향을 미치는 것으로 보인다. 즉, 허브 설치비용에 관계없이 허브 용량제한이 느슨한 경우(LL, HL)에는 직접수송이 네트워크 설계에 큰 영향을 미치지 않지만, 허브 용량제한이 강한 경우(LT, HT))에는 직접수송을 고려함으로써 보다 저렴한 비용으로 네트워크를 설계할 수 있음을 알 수 있다.

이는 단순히 수송경로들(허브를 경유한 수송경로, 직접수송 경로) 중 직접수송을 통해서 얻는 비용절감의 효과에 기인한 것만이 아니다. 앞에서 언급했듯이 직접수송은 허브의 용량제한을 완화시켜 스포크들이 좋은 위치의 허브에 할당할 기회를 증가시킨다. 이러한 스포크의 허브 할당은 시스템 전체 네트워크 구조에 변화를 주게 된다.

5. 결 론

본 연구에서는 허브의 용량제한과 직접수송이 있는 hub-and-spoke 수송네트워크에서 물류관련비용의 최소화를 목적으로 하는 허브위치 선정문제를 다루었다. 이를 위한 0-1 정수계획 모형을 제시하고, 대형 문제의 해결을 위한 방법론으로 진화 알고리즘을 채용하였다. 진화알고리즘의 구현을 위해 다루는 문제에 적합한 개체표현과 유전연산자를 개발하였다. 그리고 초기 모집단의 개체생성과 유전 연산에서CHLPwD가 갖는 문제의 정보를 이용한 발견적 기법을 제시하였다. 제안한 알고리즘의 검증에 위해, 크기와 복잡도가 다른 여러 유형의 문제에 대해 알고리즘의 성능, 정보의 이용이EA의 성능에 미치는 영향, 현재 개발되어 있는 유전연산자(돌연변이)의 특성, 그리고 네트워크 설계에 직접수송이 미치는 영향에 대하여 실험하고 그 결과를 분석하였다.

실험 결과, 먼저 알고리즘의 적용성 측면에서 크기가 작은 문제에 대해서는 최적해를 찾는 CPLEX 소프트웨어를 이용하여 얻은 해와 동일한 해를 탐색함을 보였고, 큰 문제에 대해서는 발견적 기법을 통하여 얻은 해와 비교하여 상대적으로 진

화알고리즘으로 구한 해가 우수함을 보였다. 다음으로 EA에 적절한 정보를 이용하는 것이 CHLPwD와 같이 제약이 강한 경우에 효율적임을 알 수 있었다. 돌연변이 연산자들에 대한 실험을 통하여 돌연변이 연산자들 각각에 대한 성능에는 큰 차이가 없으나 하나의 돌연변이만을 사용하는 것보다 여러 돌연변이를 혼합하여 사용하는 것이 더 효과적인 탐색성능을 보임을 확인했다. 마지막으로 용량제한이 있는 네트워크 설계에 직접수송을 고려함으로써 보다 효과적으로 네트워크를 구성할 수 있음을 보였다.

본 연구에서 제안한 진화알고리즘은 여러 유형의H&S구조를 갖는 네트워크의 설계나 재설계에 적용될 수 있다. 예로 육상수송과 항공·항만수송, 우편시스템 등이다. 추후 연구로는 하나의 스포크가 여러 허브에 연결될 수 있는 모형과 스포크에 여러 허부 지점이 연결된 확장된 모형 등을 고려할 수 있다.

부 록 1. CHLPwD를 위한 발견적 기법

<기호정의>

L : $\Pi_i / F_i, i = 1, \dots, n$ 를 내림차순으로 나열한 목록

$L_{(i)}$: 목록 L 에서 i 번째의 지점

W : 모든 지점간의 총 화물량($W = \sum_i \sum_{j \neq i} w_{ij}$)

h : $h \leftarrow \text{Max}\{t | \sum_{i=1}^t \Pi_{L(i)} \leq W\}$

SS_{ikl} : 스포크 i 를 현재 연결된 허브 k 대신에 다른 허브 l 에 연결할 때 절약되는 목적함수 값

SH_{ik} : 허브 k 와 i 에 연결된 스포크 i 를 허브 i 와 스포크 k 로 교체하고, 기존 허브 k 에 연결된 다른 모든 스포크를 새로 만든 허브 i 에 연결할 때 절약되는 목적함수 값

<절 차>

단계 1. 초기화

L 과 h 를 구한다. $z^* \leftarrow \infty$ 로 둔다.

단계 2. 허브가 아닌 지점들을 허브에 할당

화물량이 가장 큰 지점부터 차례로 이 화물량을 수용할 수 있는 가장 가까운 허브에 할당한다. 만약 화물량을 수용할 수 있는 허브가 존재하지 않으면 용량제한을 가장 적게 어기는 허브에 할당한다.

단계 3. 목적함수 값 계산

3.1 허브를 통한 수송에 비해 직접수송 비용이 더 저렴한 것을 직접수송한다.

3.2 목적함수 값 z 를 계산한다. 만약, $z < z^*$ 이면 $z^* \leftarrow z$ 로 갱신하고 단계 4로 간다.

단계 4. 스포크 재할당

4.1 각 스포크 $i \in (V-H)$ 에 대해 $SS_{ikl}, \forall l \in H$ 을 구하고, 그 중 절약값이 가장 큰 $SS_{i^*kl^*}$ 를 선택한다.

4.2 만약 $SS_{i^*kl^*} > 0$ 이면, 스포크 i^* 를 허브 l^* 에 재할당하고 단

계 4.1로 간다. 그렇지 않으면 단계 5로 간다.

단계 5. 허브 교체

5.1 허브가 아닌 각 스포크 $i \in (V-H)$ 에 대해 SH_{ik} 를 계산하고, 그 중 절약값이 가장 큰 $SH_{i^*k^*}$ 를 선택한다.

5.2 만약 $SH_{i^*k^*} > 0$ 이면, 지점 i^* 와 허브 k^* 를 서로 교체한다. 그리고 단계 3으로 간다. 그렇지 않으면 단계 6으로 간다.

단계 6. 종료조건을 만족하면 끝내고, 그렇지 않으면 $h \leftarrow h+1$ 로 놓고 단계 2로 간다.

단계 1에서 초기 허브는 $L_{(i)}, i = 1, \dots, h$ 까지의 지점들로 구성한다. 단계 6의 종료조건은 단계 4와 5에서 해의 개선이 이루어 지지 않은 반복횟수를 사용한다.

참고문헌

- Aykin, T. (1994), Lagrangian relaxation based approaches to capacitated hub and spoke network design problem, *European Journal of Operational Research*, **79**(3), 501-523.
- Aykin, T. (1995), The hub location and routing problem, *European Journal of Operational Research*, **83**(1), 200-219.
- Beasley J. E. (1990), OR-library: distributing test problem by electronic mail, *Journal of the Operational Research*, **41**, 1069-1072.
- Boland, N., Krishnamoorthy, M., Ernst, A.T. and Ebery, J. (2004), Preprocessing and cutting for multiple allocation hub location problems, *European Journal of Operational Research*, **155**(3), 638-653.
- Campbell, J.F. (1994), Integer programming formulations of discrete hub location problems, *European Journal of Operational Research*, **72**(2), 387-405.
- Chung, H.Y. and Kim, Y.K. (2001), Development of evolutionary algorithms for determining the k most vital arcs in shortest path problem, *Journal of the Korean Operations Research and Management Science Society*, **16**(3), 47-58.
- Ebery, J., Krishnamoorthy, M., Ernst, A. and Boland, N. (2000), The capacitated multiple allocation hub location problem: formulations and algorithms, *European Journal of Operational Research*, **120**(3), 614-631.
- Eiben, A. E., Raue, P. E., and Ruttkay, Zs. (1995), How to apply genetic algorithms to constrained problems. *Practical Handbook of Genetic Algorithms 1 : Applications* (edited by L. Chambers), 307-365. CRC Press, Boca Raton.
- Elhedhli, S. and Hu, F.X. (2005), Hub-and-spoke network design with congestion, *Computers & Operations Research*, **32**(6), 1615-1632.
- Ernst, A.T. and Krishnamoorthy, M. (1999), Solution algorithms for the capacitated single allocation hub location problem, *Annals of Operations Research*, **86**(0), 141-159.
- Falkenauer, E., (1994), A new representation and operators for genetic algorithms applied to grouping problems. *Evolutionary Computation*, **2**, 123-144.
- Goldberg, D.E.(1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Reading, Massachusetts.
- Hasan, P. and Vaidyanathan, J. (1998), A multi commodity, multi plant, capacitated facility location problem: formulation and efficient heuristic solution, *Computers & Operations Research*, **25**(10), 869-878.
- Helm, S.A. and Venkataramanan, M.A. (1998), Solution approaches to hub location problems, *Annals of Operations Research*, **78**(0), 31-50.
- Helm, S.A. (1998), A hybrid heuristic for the uncapacitated hub location problem, *European Journal of Operational Research*, **106**(3), 489-499.
- Kim, Y. K., Hyun, C. J. (1994), Genetic Algorithms for mixed model assembly line sequencing, *Journal of the Korean Institute of Industrial Engineering*, **20**(3), 409-428.
- Kim, Y. K., Yun, B. S. and Lee, S. B. (1997), Metaheuristics, Yeongji Moonhwas, Seoul, Korea.
- Kim, Y. K., Lee, H. Y. and Kim, J. Y. (2001), An artificial adaptation model by means of endoparasitic evolution process, *Journal of the Korean Institute of Industrial Engineering*, **27**(3), 239-249.
- Klincewicz, J. G.(1991), Heuristics for the p hub location problem, *European Journal of Operational Research*, **53**, 25-37.
- Lumsden, K., Dallari, F. and Ruggeri, R. (1999), Improving the efficiency of the hub and spoke system for the SKF European distribution network, *International Journal of Physical Distribution & Logistics Management*, **29**(1/2), 50-64.
- Melkote, S. and Daskin, M.S. (2001), Capacitated facility location/network design problems, *European Journal of Operational Research*, **129**(3), 481-495.
- Michalewicz, Z. (1999), *Genetic Algorithms + Data Structures = Evolution Programs*, the forth edition, Springer Verlag.
- O'Kelly, M.E. (1987), A quadratic integer program for the location of interacting hub facilities, *European Journal of Operational Research*, **32**(3), 393-404.
- Pagie, L., and Hogeweg, P. (1998), Evolutionary consequences of coevolving targets, *Evolutionary Computation* **5**(4), 401-418.
- Skorin Kapov, D. and Skorin Kapov, J. (1994), On tabu search for the location of interacting hub facilities, *European Journal of Operational Research*, **73**(3), 502-509.
- Skorin Kapov, D., Skorin Kapov, J., and O'Kelly, M. (1996), Tight linear programming relaxations of uncapacitated p hub median problems, *European Journal of Operational Research* **94**(3), 582-593.
- Topcuoglu, H., Corut, F., Ermis, M. and Yilmaz, G. (2005), Solving the uncapacitated hub location problem using genetic algorithms, *Computer and Operations Research*, **32**(4), 967-984.
- Zpfel, G. and Wasner, M. (2002), Planning and optimization of hub-and-spoke transportation network of cooperative third-party logistics providers, *International Journal of Production Economics*, **78**(2), 207-220.