

고정비용 수송문제를 위한 우선순위기반 유전자 표현법을 이용한 유전 알고리즘 개발

김동훈*, 김종율*, 조정복*
*동서대학교 컴퓨터정보공학부

The Development of GA with Priority-based Genetic Representation for Fixed Charge Transportation Problem

Dong-hun Kim* · Jong Ryul Kim* · Jung-bok Jo*

*Division of Computer Information Engineering, Dongseo University

E-mail : genetic.kim@gmail.com, xmaskjr@gdsu.dongseo.ac.kr, jobok@gdsu.dongseo.ac.kr

요 약

본 논문은 생산 물류 시스템 최적화의 실현에 가장 대표적인 생산수송계획문재인 수송문제(TP: Transportation Problem)에 고정비용을 고려한 고정비용 수송문제(fcTP: Fixed charge Transportation Problem)를 다룬다. 특히 NP-hard 문제로 널리 알려진 TP에서 수송량에 비례하는 가변비용과 함께 추가적으로 모든 경로에서 발생하는 고정비용을 함께 고려한 fcTP를 다룬다. 따라서 이러한 fcTP를 해결하기 위해 메타 휴리스틱 기법 중에 가장 널리 이용되고 있는 유전 알고리즘(GA: Genetic Algorithm)을 이용한 해법을 제시하고자 한다. 본 논문에서는 GA를 이용해 고정비용 수송문제의 해를 우선순위기반 유전자 표현법을 이용해 fcTP에 적용해 보고 수치 실험을 통해 그 성능에 대한 연구를 한다.

키워드

고정비용수송문제(Fixed Charge Transportation Problem), 유전 알고리즘(Genetic Algorithm), 유전자 표현법(Genetic Representation), 우선순위기반 인코딩(Priority-based encoding)

1. 서 론

수송문제(TP: Transportation Problem)는 1941년에 Hitchcock[1]에 의해서 정식화 되었다. 지난 수년간 많은 연구자들에 의해서 새로운 모델과 기법들이 제안되고 있다. 고정비용 수송문제(fcTP: Fixed Charge Transportation Problem)는 TP의 연장으로[2] fcTP는 전통적인 수송 문제에서 고려되는 수송비용뿐만 아니라 고정비용도 동시에 최소화 하는 것을 목적으로 한다. 수송문제는, 공급측으로부터 수요지까지 총 수송비용이 최소가 되는 경로를 결정하는 문제이다. 여기서 각 공급처와 수급처를 노드(node)로 나타내고 각 수송경로를 에지(edge)로 나타내면 수송문제를 수송 트리(tree)로 표현할 수 있다. 하지만 실세계의 문제에 응용하기 위해서는 네트워크 모델의 확장이나 더

복잡한 제약 조건의 추가 등이 요구된다. 이런 경우에 fcTP문제는 고전적인 방법으로 해결하기 어려운 NP-hard 문제로 널리 알려져 있다.

최근에 보편적이면서도 널리 실세계에 적용되는 최적화 문제들을 푸는데 많은 메타휴리스틱 방법들이 제안되고 있다. 이러한 메타휴리스틱 방법들 중에는 진화 연산법, 유전적 프로그래밍, 진화 전략 혹은 타부 검색(Tabu Search), 시뮬레이티드 어닐링(Simulated Annealing)등과 같은 방법들이 있다. 이들 가운데 유전 알고리즘은 매우 주목받은 최적화 방법들 중에 하나이다. 유전 알고리즘은 생물의 유전현상을 모방한 해 탐색 방법을 사용하고 있다. 유전 알고리즘은 비선형의 문제를 비교적 단시간에 용이하게 풀 수 있다는 점에서 주목받고 있고 여러 가지 분야에 적용되고 있다.

유전 알고리즘을 이용한 수송문제로의 적용도 그 하나이다.[3]

TP에 대한 최초의 GA 어플리케이션은, Michalewicz[4]에 의해서 제안되었으며, 그때 사용된 네트워크의 표현에는 행렬 베이스의 염색체가 사용되었다. 그들은 염색체를 구성하기 위해서 자체 개발한 매트릭스 기반 교차 및 돌연변이를 통한 매트릭스 표현을 사용했다.

Syarif와 Gen[5]은 이단계 배급 문제를 고려하는 하이브리드 유전 알고리즘을 제시했다. 그들은 Prüfer number 표현법을 사용하였고, 교배율과 돌연변이율을 위하여 퍼지 룰 테이블을 이용한 하이브리드 유전 알고리즘을 사용하였다. Gen과 Cheng[3]은 2단계 수송문제에 우선순위기반 표현법을 이용한 새로운 디코딩과 인코딩 방법을 제시하였다.

본 논문에서는 fcTP를 해결하기 위해 유전 알고리즘을 이용해서 수송문제의 해를 우선순위기반 표현법을 이용해서 fcTP에 적용해 보고 수치 실험을 통해 그 성능에 대한 연구를 한다.

II. 문제기술

이번 장은 운송비와 고정비를 동시에 고려한 fcTP의 수리모형을 제시한다. 수확모델은 그림 1과 같이 다수의 수급처에서 다수의 공급처로 한 종류의 상품을 최소 비용으로 수송계획을 수립하는 것으로 모든 공급처에서 생산되는 상품의 총수는 수급처에서 요구하는 상품의 총 요구량과 동일하다고 가정한다. 이 모델의 목적은 각 공급처의 공급 가능한 기준에서 최적화 하며 동시에 각 수급처의 요구를 만족시키는 것이다.

이 수송문제는 i 개의 공급처와 j 개의 수급처를 가진다고 가정하면 다음과 같은 수송계획 모델을 수립할 수 있다.

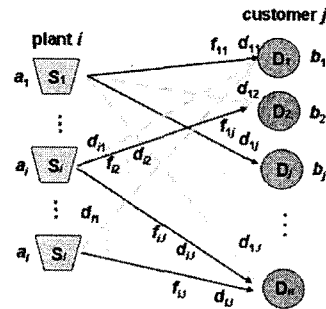


그림 1. 고정비용 수송문제의 모델 예

fcTP를 설명하기 위해 필요한 기호들을 정의하면 다음과 같다.

Indices

i : 공급처의 집합 ($i=1,2,\dots, I$)

j : 수급처의 집합 ($j=1,2,\dots, J$)

매개변수(Parameters)

a_i : 공급처 i 에서 공급 가능한 상품의 개수

b_j : 수급처 j 에서 요구하는 상품의 개수

c_{ij} : 공급처 i 에서 수급처 j 로의 수송비용

d_{ij} : 공급처 i 에서 수급처 j 로의 고정 비용

의사결정변수(Decision Variables)

x_{ij} : 공급처 i 에서 수급처 j 로 수송이 정해지면

그에 해당하는 수송량

목적함수(Objective Function)

$f_{ij}(x)$: 공급처 i 에서 수급처 j 까지의 총 운송비용

총 수송비용에서 공급처 i 에서 수급처

j 로의 수송에 대해 선형이 될 때

$f_{ij}(x) = c_{ij} x_{ij}$ 가 된다.

그림 1에서 나타난 고정비용 수송문제에 관한 수학 모델은 다음과 같이 정식화 할 수 있다.

$$\min f(x) = \sum_{i=1}^I \sum_{j=1}^J (f_{ij}(x) + d_{ij} \delta_{ij}) \quad (1)$$

$$\text{s.t.} \sum_{j=1}^J x_{ij} \leq a_i, \quad i = 1, 2, \dots, I$$

$$\sum_{i=1}^I x_{ij} \geq b_j, \quad j = 1, 2, \dots, J$$

$$x_{ij} \geq 0, \quad \forall i, j$$

$$\text{with } \delta_{ij}(x) = \begin{cases} 1, & \text{if } x_{ij} > 0 \\ 0, & \text{if } x_{ij} \leq 0 \end{cases}$$

본 목적 함수 (1)은 총 운송비용과 고정비용을 최소화하는 식이다. 제약식은 각각 공급 가능한 상품의 개수와 요구하는 상품의 개수에 대한 제약식이다. 제일 아래 제약식은 비용조건을 나타낸다.

III. fcTP를 위한 유전 알고리즘

3.1 Priority-based 표현법

수송 문제에서 염색체는 공급처와 수급처로 구성되고 운송트리의 길이는 공급처 m 과 수급처 n 의 총 수($m+n$)와 같다. 염색체와 대응하는 수송 트리는 공급처와 수급처 사이에서 추가 연속되는 링크에 의해 생성된다.

각 스텝에서 하나의 링크는 가장 큰 우선순위와 연결되는 최소비용을 찾아서 수급처(공급처)에 연결하여 공급처(수급처)를 선택하여 추가한다.[7]

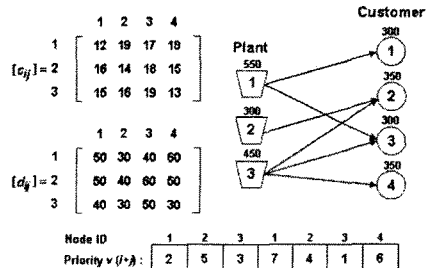


그림 2. 우선순위기반 유전자 표현법의 운송트리 염색체 예[3]

그림 2는 Plant와 Customer간의 운송비용과 고정비용의 비용행렬과 탐색체를 예로 들고 있다. 여기서는 3개의 Plant와 4개의 Customer를 고려하였기 때문에 탐색체의 길이는 7이 되고 그림 3 우선순위기반 인코딩 프로시저에 의해 탐색체는 생성된다.

```

procedure 1.1 : Encoding between plant and customer
input:  $l$ : number of plants
         $J$ : number of customers
         $a_i$ : capacity of plant  $i, \forall i \in I$ 
         $b_j$ : capacity of customers  $j, \forall j \in J$ 
         $c_{ij}$ : transportation cost from plant  $i$  to customer  $j$ 
         $x_j$ : fixed charge cost from plant  $i$  to customer  $j$ 
output:  $v_i(i^*)$ : chromosome,  $\forall i \in I, \forall j \in J$ 
step 1:  $priority\ p \leftarrow (|I|+|J|), v_i(i^*) \leftarrow 0, \forall i^* \in |I|+|J|$ ;
step 2:  $(i^*, j^*) \leftarrow \text{argmin}\{c_{ij} \mid x_j = 0 \ \& \ (a_i = x_{ij}, b_j = x_j)\}$ ;
step 3:  $a_p = a_p - x_{p,j^*}, b_{j^*} = b_{j^*} - x_{p,j^*}$ ;
step 4: if  $a_p = 0$  then  $v_i(i^*) \leftarrow p, p \leftarrow p-1$ ;
step 5: if  $(a_p = 0, \forall p \in I)$  &  $(b_{j^*} = 0, \forall j^* \in J)$  then goto step 6;
        else return step 1;
step 6: for  $l = 1$  to  $p$  do
         $v_i(i^*) \leftarrow l, l = \text{random}(1, (|I|+|J|) \ \& \ v_i(i^*) = 0)$ ;
step 7: output encoding  $v_i(i^*), \forall i \in |I|+|J|$ 
    
```

그림 3. 우선순위기반 인코딩 프로시저[7]

그림 3에서 얻은 탐색체의 최고 우선순위의 노드를 선택하여 그 노드와 연결되는 최소비용을 찾는다. 그리고, 각 노드의 용량을 비교하여 적은 쪽의 용량만큼 운반하게 된다. 만약 최고 우선순위 노드의 용량을 만족하면 두 번째 우선순위로 넘어가게 되고 그렇지 않을 경우 용량이 만족될 때까지 그 다음 최소비용의 노드로 넘어가게 된다. 공급처의 용량을 다 만족할 때까지 이 과정을 되풀이한다. 그림 4 우선순위기반 디코딩 프로시저에 의해 정리하였다.

```

procedure 1.2 : Decoding between plant and customer
input:  $l$ : number of plants
         $J$ : number of customers
         $a_i$ : capacity of plant  $i, \forall i \in I$ 
         $b_j$ : capacity of customers  $j, \forall j \in J$ 
         $c_{ij}$ : transportation cost from plant  $i$  to customer  $j$ 
         $d_j$ : fixed charge cost from plant  $i$  to customer  $j$ 
         $v_i(i^*)$ : chromosome,  $\forall i \in I, \forall j \in J$ 
output:  $x_{ij}$ : the amount of shipment from plant  $i$  to customer  $j$ 
step 0:  $x_{ij} = 0, \forall i \in I, \forall j \in J$ 
step 1:  $i \leftarrow \text{argmax}\{v_i(i^*) \mid i^* \in |I|+|J|\}$ ; select a node
step 2: if  $i \in I$ , then  $i \leftarrow i$ ; select a plant
         $j^* \leftarrow \text{argmin}\{c_{ij} \mid a_i + d_j \mid v_i(i^*) = 0, j \in J\}$ ; select a  $j$  with lowest cost
        else  $j \leftarrow j$ ; select a customer
         $i^* \leftarrow \text{argmin}\{c_{ij} \mid b_j - a_i \mid v_i(i^*) = 0, i \in I\}$ ; select a  $i$  with lowest cost
step 4:  $x_{i^*j^*} \leftarrow \min\{a_{i^*}, b_{j^*}\}$ ; assign available amount of units
        update the availabilities on  $i$   $(a_{i^*})$  and  $j$   $(b_{j^*})$ 
         $a_{i^*} = a_{i^*} - x_{i^*j^*}$ , and  $b_{j^*} = b_{j^*} - x_{i^*j^*}$ 
step 5: if  $a_{i^*} = 0$ , then  $v_i(i^*) \leftarrow 0$ 
        if  $b_{j^*} = 0$ , then  $v_i(i^*) \leftarrow 0$ 
step 6: if  $v_i(i^*) = 0, \forall i \in I, \forall j \in J$ , output  $x_{ij}$ 
        else return step 1
    
```

그림 4. 우선순위기반 디코딩 프로시저

3.2 Prüfer number 표현법

Prüfer number는 독일 수학자 Ernst Paul Heinz Prüfer[6]에 의해 처음 고안되었으며, 네트워크 문제들을 푸는데 Prüfer number 표현 사용은 Gen과 Cheng에 의하여 소개되었다.[3] 그들은 네트워크 그래프에서 가능한 모든 트리들이 일대일로 표현할 수 있는 Prüfer number를 이용하였다. 그들은 네트워크 그래프에서 가능한 모든 트리들이 일대일로 표현할 수 있는 Prüfer number

를 이용하였다.

3.3 LNB(Linked Node-based)

LNB 표현법은 Palmer와 Kershenbaum[8]이 OCST 문제를 해결하기 위해 제안한 방법이다. 이 방법은 가중치가 부여된 벡터(weight vector)를 사용하여 트리 네트워크의 구조를 표현하고 진화 알고리즘이 네트워크에 있는 노드와 에지들 사이에 좋은 것들을 찾아낼 수 있게 만든다. 이 표현법에서는 각 탐색체는 각 노드의 bias 값을 이용해서 나타낸다. LNB 표현법은 각 평가(evaluation) 이후에 비용 테이블이 수정되고, 수정된 비용이 테이블을 이용해서 Prim 알고리즘을 수행하여 트리를 만들기 때문에 많은 계산 시간을 필요로 한다.[9]

3.4 유전 연산자

3.4.1 교차 연산자

교차(Crossover)는 아주 중요한 유전 알고리즘의 연산자이다. 이는 서로 다른 부모 유전자를 교차에 의해 새로운 자손을 생성하는 과정에서 이용되는 연산자이다. 이는 유전 알고리즘에 있어서 새로운 해 영역을 탐색할 때, 해 영역의 지역탐색이라 한다. Prüfer number 표현법에서는 균일 교차(Uniform Crossover)를 이용한다. 균일교차란 각 유전자가 독립적으로 교환될 수 있도록 하는 것으로서 Random으로 0또는 1의 마스크를 씌워 유전자를 교환하는 방식이다.[3]

우선순위기반 표현법에서는 교차 처리에 WMX(Weight Mapping Crossover)를 사용한다. 이것은 1점 교차의 확장으로서 보고 잡혀 부모로부터 자손에게 교차 부분의 우선순위를 계승할 뿐만 아니라, 교차 처리의 후에 복수처리를 실행할 필요는 없다. 즉, 얻을 수 있는 자손 탐색체 안에 같은 우선순위를 가지는 값이 존재할 것은 없다.[7]

3.4.2 돌연변이 연산자

Prüfer number 표현법에서는 돌연변이 연산자로 역순(Inversion) 연산자와 전환(Displacement) 연산자를 함께 사용 하였고 우선순위기반 표현법에서는 삽입(Insertion) 연산자를 사용하였다. 역순 연산자는 탐색체 내에서 두 개의 절단점을 임의로 선택하고 절단점 사이의 인자들을 역순으로 하여 자손을 생산하는 방법이다. 전환 연산자는 탐색체 내에서 두 개의 절단점을 임의로 선택하고 절단점 사이의 인자들을 임의의 위치로 전위하여 자손을 생산하는 방법이다.[3]

삽입 연산자는 랜덤으로 한 유전자를 선택하여 다시 랜덤으로 위치를 선택하여 삽입하는 방법이다.

IV. 수치실험 및 결과

본 논문에서는 제안한 우선순위기반 표현법의 유효성과 효율성을 증명하기 위해 각각의 수치 데이터를 수집하고 다른 2가지 방법을 이용하여 비교하였다. 실험을 위해 Intel Pentium(R) 4CPU

2.00Ghz를 이용하여 유전 알고리즘을 작성하여 수치실험 하였다. 각 알고리즘에 사용된 파라메타 및 연산자들은 표 1과 같다.

표 1. 각 알고리즘에 사용된 파라메타 및 연산자

Operator	Parameter
초기모집단	100
교배확률	0.3
돌연변이확률	0.5
교배연산자	Uniform, WMX
돌연변이연산자	Inversion, displacement Insertion
종료조건	500 generation

각각의 유전자 표현법에 대한 비교를 위해 노드의 수를 8개(3×5: 3개의 공급처와 5개의 수급처)와 13개(5×8), 15개(5×10), 25개(10×15)로 하고 각각의 노드 수에 따라 초기모집단수(Population)를 100으로 하여 실험을 수행하였다. 모든 실험 데이터상에서 각 알고리즘을 10번씩 수행하였으며, 실험결과를 수행 후 출력된 결과값에 대해 최대값(Max), 최소값(Min), 평균값(Average), 그리고 이에 대한 표준편차(STDV)를 중심으로 나타내었고, 그 결과 데이터는 표 2와 같다.

표 2. 유전자 표현법에 대한 실험결과 비교

Node		Priority-based	Prüfer number	LNB
8(3×5)	Max	2060	2060	2410
	Min	2060	2060	2060
	Average	2060	2060	2295
	STDV	0	0	108.74
13(5×8)	Max	3120	3120	3490
	Min	3120	3120	3120
	Average	3120	3120	3287
	STDV	0	0	137.48
15(5×10)	Max	3350	3360	3855
	Min	3350	3350	3390
	Average	3350	3353.5	3590
	STDV	0	5	141.74
25(10×15)	Max	5285	5385	6285
	Min	5245	5280	5690
	Average	5263.5	5307	5982
	STDV	12	30	182.6

V. 결 론

본 논문에서는 고정비용을 고려한 비선형 수송문제에 대해서 우선순위기반 표현법과 Prüfer number 표현법, LNB 표현법을 이용하여 각각에 대한 성능을 비교하였다.

우선순위기반 표현법은 노드의 개수가 적은 경우, 표준편차가 0으로 균일한 최소값을 산출해 냈으며, 노드의 개수가 점차 늘어날수록 약간의 변동은 있지만 가장 안정된 최소비용을 산출해냈다.

Prüfer number 표현법은 노드의 개수가 적은

경우, 우선순위기반 표현법과 균일하게 최소값을 산출해 내었지만, 노드의 개수가 증가할수록 우선순위기반 표현법에 비해 더 높은 표준편차와 최소값을 산출해냈다.

LNB 표현법은 노드의 개수가 늘어날수록 다른 두 가지 표현법보다 가장 나쁜 최소값을 산출하였다. 그리고 노드의 개수가 늘어날수록 표준편차가 증가함을 알 수 있었다. 표준편차는 다른 두 표현법에 비해 높은 수치를 보였다.

본 실험을 통해 세 가지의 표현법을 비교해 본 결과 우선순위기반 표현법이 다른 표현법에 비해 가장 안정되고 최적의 비용값을 산출함이 증명되었다. 따라서 비선형 수송문제에 있어서 우선순위기반 표현법이 가장 적합하다는 것을 확인 할 수 있었다.

참고문헌

- [1] F. Hitchcock (1941). The distribution of a product from several sources to numerous locations, *Journal of Mathematical Physics*, vol. 20, pp. 224-230.
- [2] W. M. Hirsch and G. B. Dantzig (1968). The fixed charge problem. *Naval Research Logistics Quarterly*, 15, pp. 413 - 425.
- [3] M. Gen and R. W. Cheng (2000). *Genetic Algorithm and Engineering Optimization*, Wiley, New York.
- [4] Z. Michalewicz, G. A. Vignaux and M. Hobbs (1991). A non-standard genetic algorithm for the nonlinear transportation problem, *ORSA Journal on Computing*, vol. 3, no. 4, pp. 307-316.
- [5] A. Syarif and M. Gen (2003). Double Spanning Tree-based Genetic algorithm For Two Stage Transportation Problem, *International Journal of Knowledge-Based Intelligent Engineering System*, vol. 7, no. 4, pp.388-389.
- [6] H. Prüfer (1981). Neuer beweis eines Satzes über Permutationen, *Archives of mathematical Physics*, vol. 27, pp. 742-744.
- [7] M. Gen, F. Altiparmak and L. Lin (2006). A genetic algorithm for two-stage transportation problem using priority-based encoding, *OR Spectrum*, vol.28, no.3, pp.337-354.
- [8] Palmer. C. C. and A. Kershenbaum, "Representing trees in genetic algorithm", in *Proc. IEEE Int. Conf. Evol. Comput.*, pp.379-384, 1994
- [9] 석상문, 장석철, 변성철, 안병하, "최적통신 결침 나무 문제를 해결하기 위한 진화 알고리즘", *정보과학회논문지 제32권 제4호*, pp.268-276, 2005