

# 고정비용 비선형 수송문제 해결을 위한 유전자 표현법들의 성능 비교

동서대학교 컴퓨터정보공학부

장지훈, 김병기, 김종울, 조정복

## The Comparison of Genetic Representation methods for Solving The Fixed Charge Non-linear Transportation Problems

Ji hoon Jang, Byung ki, Kim, Jong ryul Kim, Jungbok Jo

Dongseo University Computer Information Engineering Division

E-mail : lab344@lycos.co.kr, bbangbbang@gmail.com, {xmaskjr jobok}@gdsu.dongseo.ac.kr

### 요 약

수송문제는 산업공학 및 전자계산학 분야에서 중요한 문제 중의 하나로 인식된다. 수송문제가 시설을 수립하거나 고객들의 요구를 이행하기 위한 추가적인 고정 비용과 연관될 때, 이를 고정비용을 고려한 비선형 수송문제(Fixed Charge Non-linear Transportation Problem)라 한다. 고정비용을 고려한 비선형 수송문제는 한 종류의 상품을 다수의 공급처에서 다수의 수급처로 수송할 때, 수송비용과 고정비용이 최소가 되도록 수송량을 결정하는 문제이다. 본 논문에서는 이 비선형 수송문제에 가장 많이 쓰이는 메타 휴리스틱 방법들 중 유전 알고리즘을 이용한 해법을 제시한다. 유전 알고리즘을 적용함에 있어서 가장 중요한 것 중에 하나는 해의 유전자표현을 어떻게 나타낼 것인가인데, 본 논문에서는 수송문제의 해를 걸침나무로 표현할 수 있다는 점에 착안하여 유전자 표현법들을 수송문제에 적용해 보고 수치 실험을 통해 그 성능에 대한 비교를 한다.

### 키워드

비선형수송문제, 유전 알고리즘, 유전자표현, 걸침나무

### 1. 서 론

수송문제는 산업공학 및 전자계산학 분야에서 중요한 문제 중의 하나로 인식되며, 이는 1941년 F. Hitchcock[1]이 처음으로 정식화 한 것으로 알려져 있다. 고정비용을 고려한 비선형 수송문제는 한 종류의 상품을 다수의 공급처에서 다수의 수급처로 수송할 때, 수송비용과 고정 비용이 최소가 되도록 수송량을 결정하는 문제이다. 여기서 각 공급처와 수급처를 노드(node)로 나타내고 각 수송경로를 에지(edge)로 나타내면, 수송문제는 트리(tree)로 표현할 수 있다.

일반적으로 목적 함수는 여러 가능한 할당 형태에 따른 가변적인 비용과 고정 비용을 고려한 것으로 이러한 2가지 비용을 모두 동시에 최소화 하는 것이다. 또한, 고정비용 수송문제는 제약조건을 포함하고 있으며 조합 최적화 문제로 정식화 된다. 이때의 고정 비용 수송 문제는 쉽게 풀기 곤란한 NP-hard 문제로서, 이를 해결하기 위한 많은 시도들이 있어왔다.

최근에 보편적이면서도 널리 실세계에 적용되는 최적화 문제를 해결하는데 다양한 메타휴리스틱

방법들이 제안되고 있으며, 이러한 메타휴리스틱 방법 들 중에는 유전 알고리즘을 이용한 문제의 해법을 소개한다.[2],[3].

본 논문에서는 고정비용 비선형 수송문제를 위한 유전 표현법 중 Netkey(Network Random Key)표현법, LNB(Link and Node Biased)표현법, Degree based표현법, Edge based표현법, Prüfer number표현법 다섯 가지 트리 표현법을 수송 문제에 적용하고 수치 실험을 통해 그 성능에 대한 비교 연구를 수행하였다.

Netkey표현법, LNB표현법, Degree based표현법, Edge based표현법, Prüfer number표현법은 최적 통신 걸침 나무 문제(Optimum Communication Spanning Tree Problem : OCSTP)를 해결하기 위해 제안된 표현법으로써 수송문제 적용 시, 제약조건으로 공급처 간의 연결과 수급처간의 연결은 허용되지 않게 한다.

### 2. 고정비용 비선형 수송문제를 위한 유전자 표현법

고정비용 수송 문제는 총 수송비용이 최소가 되도록 공급처와 수급처간의 수송량을 결정하는 것이

다. 이 때 각 공급처에서 수급처로 상품을 수송할 때, 총 수송비용이 최소가 되도록 하여 수송그래프를 형성할 수 있다.

고정비용 수송 문제를 해결하기 위해 많은 표현법들이 제안되고 있으며, 이런 표현법들 중 본 논문에서는 Netkey표현법, LNB표현법, *d*-based표현법, *e*-based표현법, Prüfer number표현법에 대해 소개하며, 제안된 모든 표현법은 수송문제에 적용을 할 경우 제약조건이 생기게 된다. 수송문제에서 각 공급처와 수급처간의 관계에 의해서 공급처는 공급처끼리, 수급처는 수급처끼리 연결이 될 수 없다.

**2.1 Network Random Key (NetKey)**

NetKey를 이용하는 방법은 조합 최적화문제를 해결하기 위해 Bean[4]에 의해 처음 제안된 방법이다. 걸침나무(spanning tree) 문제에서 사용되는 NetKey 표현법은 기존의 MST를 해결하기 위한 Kruskal 알고리즘과 유사하며, 랜덤으로 생성된 벡터(vector)값을 이용하여 벡터값의 오름차순 정렬을 통해 에지를 연결하여 트리를 형성한다.[4]

Netkey표현법은 Kruskal의 알고리즘과 똑같은 결정을 지니고 있는데, 만약 네트워크가 완벽하게 연결되어 있고 많은 노드를 포함하는 경우, 많은 계산 시간과 메모리 량을 필요로 하게 된다는 단점이 있다.[6]

보다 자세한 사항은 참고문헌 [5]를 참조 바란다.

**2.2 Link and Node Biased(LNB)**

LNB표현법은 Palmer와 Kershenbaum[7]이 OCST 문제를 해결하기 위해 제안한 방법이다. 이 방법은 가중치가 부여된 벡터(weight vector)를 사용해서 트리 네트워크의 구조를 표현하고 진화 알고리즘이 네트워크에 있는 노드와 에지들 중에서 우수한 것들을 가려 낼 수 있도록 만든다.

LNB표현법은 각 평가(evaluation) 이후에 비용 테이블이 수정되고, 수정된 비용 테이블을 이용해서 Prim알고리즘을 수행하여 트리를 만들어내기 때문에 많은 계산시간을 필요로 한다 [6].

이 표현법의 TP문제에 적용한 것에 대한 자세한 정보는 [5]를 참조하기 바란다.

**2.3 Degree based(*d*-based) 표현법**

*d*-based 표현법[8]은 최적 통신 걸침 나무 문제를 해결하기 위해 제안된 방법이다. 순회 판매원 문제(Traveling Salesman Problem : TSP)에서 각각의 노드가 유전자 표현에 단지 한 번씩만 나타나는데 이를 시작 노드와 종료 노드를 연결하지 않고 모든 노드를 연결하면 degree가 2인 하나의 트리가 형성된다는 점에 착안하여 이 개념을 확장한 CF-TCR (Cycle Free - Tree Construction Rule)이라는 방법을 이용한다. 노드의 개수가 *n*일 때, *n*×2의 길이를 가진 유전자 표현을 고려해보면, 각각의 노드는 유전자 표현에 두 번씩 나타나야만 한다. 예를 들면 7개의 노드가 존재하고 이를 *n*×2 길이의 노드 스트링을 형성하면 {3,1,1,4,2,5,2,7,6,3,6,4,5,7}과 같이 나타낼 수 있고, 이렇게 형성된 스트링은 CF-TCR방법을 이용하여 그림 3

과 같은 트리를 형성하게 된다.

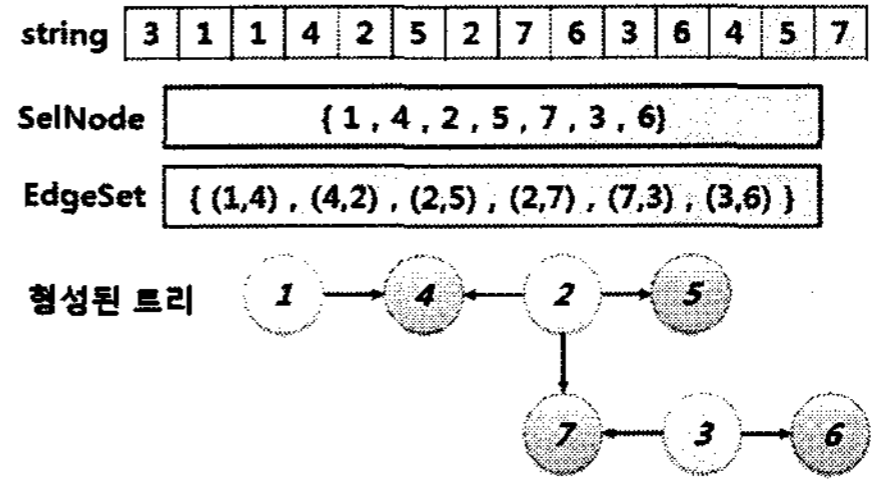


그림 1 수송문제 적용 시 *d*-based 트리 표현

*d*-based표현법은 노드의 개수가 *n*개일 때 *n*×2개의 스트링을 형성하기 때문에 많은 메모리를 필요로 하고, 항상 왼쪽에서 나타나는 에지들을 우선적으로 고려하기 때문에 뒤쪽에 유용한 정보를 지닌 유전자(gene)가 있을 경우 트리 생성 시 배제되게 된다는 것인데 이는 *d*-based표현법이 사이클을 허용하지 않는 방법으로 트리를 생성하기 때문이다.

**2.4 Edge based(*e*-based) 표현법**

*e*-based 표현법[8]은 최적 통신 걸침 나무 문제를 해결하기 위해 제안된 방법이며, CB-TCR(Cycle Breaking - Tree Construction Rule)이라고도 한다. *e*-based표현법은 (*n*-1)×2 의 길이를 가지는 스트링을 이용하여 유전자를 표현한다. 단, 모든 노드는 최소한 한번씩은 나타나야 하고, 이 때 본 표현법으로 *n*-1개의 에지를 형성할 수 있게 된다. 이렇게 만들어진 스트링을 이용하여 트리를 구성하게 된다.

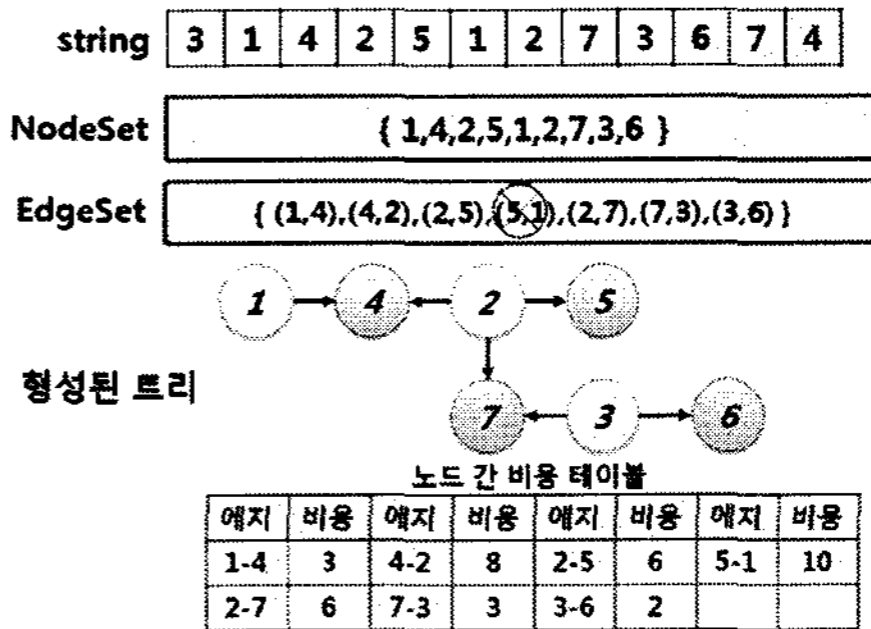


그림 2 수송문제 적용 시 *e*-based 트리 표현

*e*-based표현법은 우선 *d*-based표현법과 같은 방법으로 에지를 추가해 나가며 노드에 연결된 에지가 사이클이 발생할 경우 가장 비용값이 큰 에지를 제거하는 방식으로 그림 4와 같은 트리를 구성한다.

*e*-based표현법은 노드의 개수가 *n*개일 때 (*n*-1)×2개의 스트링을 형성한다. 이는 *d*-based표현법 보다는 적은 메모리를 필요로 하고 있으며, 사이클 형성 시 비용값이 가장 큰 에지를 제거하기 때문에 최적해에 더욱 가까운 값을 찾는데 유리하다. 하지만 사이클을 형성하는 에지들을 찾는 과정으로 인해 수행시간이 오래 걸린다.

2.5 Prüfer number 표현법

Prüfer number는 독일 수학자 Ernst Paul Heinz Prüfer (1918)에 의해 처음 고안되었으며, 네트워크 문제들을 푸는데 Prüfer number 표현 사용은 Gen과 Cheng에 의하여 소개되었다.[2][3] 그들은 네트워크 그래프에서 가능한 모든 트리들이 일대일로 표현할 수 있는 Prüfer number를 이용하였다. 그들이 주장한 Prüfer number의 이용은 신장 트리 기반 인코딩 방법에 있어서 매우 적은 메모리를 사용하여 표현할 수 있기 때문에 적절하며, 특히 어떤 확장된 수송 문제와 같은 연구 분야, 생산, 분배 문제, 최소 신장 트리 문제 등에 대하여 더욱 더 적절하다고 주장하였다 [8].

3. 실험 및 결과

본 논문에서는 세 가지 표현법의 성능을 나타내기 위해 각각의 수치 데이터를 수집하고 이를 비교 하였다. 실험을 위해 Intel Core2 CPU 6300 1.8Ghz 컴퓨터와 Visual C++를 이용하여 유전 알고리즘을 작성하였다.

각 알고리즘에 사용된 파라메타 및 연산자들은 표 1과 같다. 실험을 위해 사용된 교배연산자는 균일교배(Uniform Crossover)로 각 유전자가 독립적으로 교환될 수 있도록 Random으로 0 또는 1의 마크를 띄워 유전자를 교환하는 연산자이다.

돌연변이 연산자로 부모유전자에서 두 개의 절단점을 임의로 선택하고 절단점 사이의 인자들을 역순으로 하여 자손을 생산하는 방법인 역순(inversion)연산자와 부모 유전자에서 두 개의 절단점을 임의로 선택하고 절단점 사이의 인자들을 임의의 위치로 전위하여 자손을 생산하는 방법인 전환(displacement)연산자를 함께 사용 하였다. 그리고 선택연산자로  $(\mu+\lambda)$ 선택법으로  $\mu$ 개의 개체로 된 부모의 집단 중에서  $\lambda$ 개의 개체의 자손을 생성한다. 이때 함께  $(\mu+\lambda)$ 개체의 집단 중에서  $\mu$ 개의 개체를 선택하여 다음세대의 부모로 하는 방법이다. 그리고 최소 비용값을 minvalue로 나타내었다.

각각의 표현법에 대한 비교를 위해 노드의 개수를 8개(3×5 : 3개의 공급처와 5개의 수급처)와 10개(3×7), 13개(5×8), 15(5×10)로 실험을 수행했으며, 모든 실험 데이터 상에서 각 알고리즘을 10번씩 수행하였다. 실험 결과는 수행 후 출력된 결과값에 대한 가장 낮은 최소값(best minvalue)과 이에 대한 표준편차를 중심으로 나타내었다. 수치실험을 위한 유전 알고리즘의 유전 파라메타의 표 2로 설정했고, 그 결과 데이터는 표 2와 같다.

표 1 각 알고리즘에 사용된 파라메타 및 연산자들

Population	100
교배확률	0.2
돌연변이확률	0.4
교배 연산자	Uniform
돌연변이 연산자	Inversion, displacement
선택 연산자	$(\mu+\lambda)$ -ES
종료조건	Generation 500

노드의 개수가 8개일 때 5개의 표현법 중 Netkey표현법이 가장 낮은 best minvalue값을 출력하였고, 표준 편차는 LNB표현법 다음으로 높은 값이 나오는 것을 확인할 수 있었다. Prüfer number표현법은 best minvalue가 Netkey표현법 다음으로 낮은 값을 출력하였고, 표준편차는 0으로 10번의 수행결과가 매우 안정됨을 알 수 있었다. d-based표현법은 Prüfer number의 best minvalue와 비슷한 값을 출력하고 표준편차는 Prüfer number표현법 다음으로 안정적인 값을 보였다. e-based표현법의 best minvalue는 Prüfer number표현법과 동일한 값을 출력 하였으나, 표준편차가 다른 표현법에 비해 다소 나쁜 값을 나타내었다. LNB표현법은 best minvalue와 표준편차가 가장 나쁜 값이 나왔으며, 특히 표준편차는 다른 표현법에 비해서도 월등히 높은 값을 기록하며 결과값에 대한 신뢰도가 매우 낮았다.

노드의 개수가 10개일 때 Prüfer number표현법이 가장 낮은 best minvalue와 안정적인 표준편차를 보

표 2 실험 결과 비교

		Prüfer	Netkey	LNB	d-basde	e-based
		N=8	Avg. CPU time	0.5	0.6	0.2
N=8	Avg. minvalue	17110	17333	20399	17542	17379
	best minvalue	17110	16610	17420	17370	17110
	표준편차	0	406	1965	190	356
	N=10	Avg. CPU time	0.8	0.8	0.2	0.8
N=10	Avg. minvalue	17330	18736	19022	18333	18345
	best minvalue	17330	17530	18140	17920	17610
	표준편차	0	1214	1065	302	462
	N=13	Avg. CPU time	0.7	0.7	0.1	1.3
Avg. minvalue		20797	26161	33115	24559	24555
best minvalue		20780	24080	29470	23420	23290
표준편차		19.5	1162	2009	706	788
N=15	Avg. CPU time	1.5	1.2	0.1	6.3	5
	Avg. minvalue	19963	27240	33115	24665	24981
	best minvalue	19790	25460	29470	24050	22840
	표준편차	87.3	1266	2597	1094	1163

였다. Netkey표현법은 두번째로 낮은 best minvalue를 출력하였지만 표준편차에서 가장 나쁜 값을 보였다. LNB표현법은 노드의 개수가 8개일 때에 비해서 표준편차가 많이 안정되었지만, 다른 표현법과 비교해서는 여전히 나쁜 표준편차를 보였으며, best minvalue도 가장 나쁜 결과를 출력하였다. d-based 표현법과 e-based표현법은 노드의 개수가 8개일 때의 실험과 비교하여 큰 차이를 보이지 않았으며, 표준편차는 이전의 실험보다 두 배 가량 높아졌다.

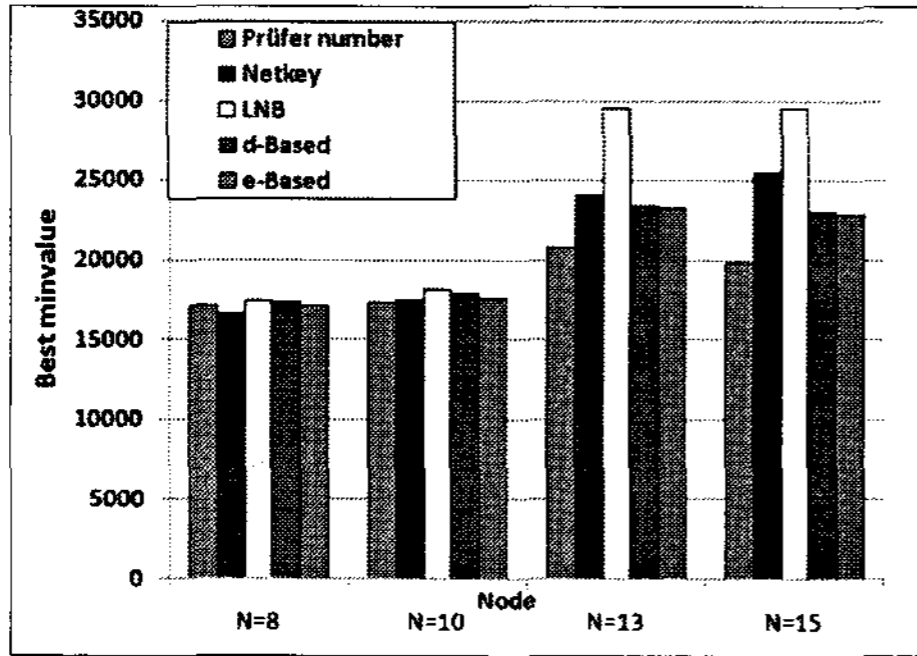


그림 3 노드 개수에 따른 최소값 비교

노드의 개수가 13개일 때 Prüfer number표현법의 best minvalue가 다른 표현법에 비해 가장 낮은 값을 출력하였으며, 표준편차도 매우 낮은 수준의 변동폭을 보였다. e-based표현법과 d-based표현법의 best minvalue와 표준편차는 비슷한 출력값을 나타내었다. Netkey표현법도 앞의 두 표현법과 비슷한 best minvalue를 나타내었으나, 표준편차는 높은 값을 나타내었다. LNB표현법은 네가지 다른 표현법들에 비해 매우 나쁜 best minvalue와 표준편차가 2000이 넘어가며 값의 신뢰도를 매우 낮게 만들었다.

노드의 개수가 15개일 때 Prüfer number표현법은 이전의 실험들처럼 best minvalue와 표준편차 모두 가장 좋은 값을 출력 하였으며, 다음으로 e-based표현법이 좋은 best minvalue값을 출력하였다. d-based표현법은 e-based표현법 다음으로 낮은 best minvalue를 출력하였으며, 두 표현법 모두 비슷한 표준편차를 보였다. Netkey표현법과 LNB표현법은 다른 표현법에 비해 이전의 실험과 같이 best minvalue와 표준편차 모두 나쁜 값이 출력되었으며, LNB표현법의 표준편차는 다른 표현법에 비교해 매우 크게 차이를 보이는 것을 확인할 수 있었다.

4. 결론

본 논문에서는 고정비용을 고려한 비선형 수송 문제에 대해서 Netkey표현법, LNB표현법, d-based표현법, e-based표현법, Prüfer number 표현법을 이용하여 각각에 대한 성능 비교를 하였다.

Prüfer number표현법은 모든 실험에서 다른 네 개의 표현법보다 좋은 값을 출력하였으며, 노드의 개수가 적을 경우에는 10번의 수행결과 표준편차의 변동이 없었으며, 노드의 개수가 늘어나 표준편차는 큰 변동폭을 보이지 않았다.

Netkey표현법은 노드의 개수가 적을 때 Prüfer number표현법과 비슷한 minvalue를 출력하였지만, 노드의 개수가 증가할수록 점차 나쁜 값을 출력해 냈다. 표준편차는 노드의 개수가 증가하자 표준편차도 크게 증가하였다.

LNB표현법은 노드의 개수가 적을 때에는 다른 표현법과 비슷한 minvalue를 출력했으나 노드의 개수가 13개 이후부터 minvalue가 급격히 나빠졌다. 표준편차도 다른 네 개의 표현법에 비해서 매우 큰 편차를 보였다. d-based표현법과 e-based표현법은 모든 실험에서 비슷한 minvalue와 표준편차를 보였다.

본 실험을 통해 다섯가지의 표현법을 비교해본 결과 Prüfer number표현법이 다른 표현법에 비해 표준편차가 가장 안정되고 최적의 minvalue를 출력하였다. 이를 미루어 본 결과 비선형 수송문제에 적용한 Prüfer number표현법, Netkey표현법, LNB표현법, d-based표현법, e-based표현법 중 Prüfer number표현법이 가장 안정되고 적합하다고 본 논문에서는 제시한다.

참고문헌

- [1] Hitchcock, F. L., "The Distribution of a Product from Several Sources to Numerous Localities", Journal of Mathematical Physic, Vol. 20, pp. 224-230, 1941.
- [2] Gen, M. and R. Cheng, Genetic Algorithms and Engineering Design, John Wiley & Sons, New York, 1997.
- [3] Gen, M. and R. Cheng, Genetic Algorithms and Engineering Optimization, John Wiley & Sons, New York, 2000.
- [4] Bean, J. C, "Genetic algorithms and random keys for sequencing and optimization" ORSA Journal on Computing, Vol.6, No2, pp.155-160, 1994.
- [5] 장지훈, 김병기, 김종율, 조정복, "비선형 수송 문제를 위한 유전자 표현법들의 비교 연구", 한국해양정보통신학회 07 추계종합학술대회, pp.248-251, 2007
- [6] 석상문, 장석철, 변성철, 안병하, "최적통신 걸침 나무 문제를 해결하기 위한 진화 알고리즘" 정보과학회논문지 제32권 제4호, pp.268-276, 2005.
- [7] C, C Palmer. "An approach to a problem in network design using genetic algorithms", Ph.D. Dissertation, Polytechnic University, 1994.
- [8] 석상문, 장석철, 변성철, 안병하, "최적 통신 걸침 나무 문제를 해결하기 위한 진화 알고리즘", 한국정보과학회 논문지, 소프트웨어 및 응용 제 32권 제 4호, 2005.4.
- [9] Jo, Jung-bok. Li, Yinzen. and Gen, M. "Nonlinear fixed charge transportation problem by spanning tree-based genetic algorithm", Computer & Industrial Engineering An International Journal, 2007 (In Press).