

비선형 수송문제를 위한 유전자 표현법들의 비교 연구

동서대학교 컴퓨터정보공학부

장지훈, 김병기, 김종율, 조정복

The Comparison of Genetic Representations for Non-linear Transportation Problems

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

Dongseo University Computer Information Engineering Division

E-mail : implant11@nate.com, bbangbbang@gmail.com, {xmaskjr jobok}@gdsu.dongseo.ac.kr

요 약

본 논문에서 다루는 고정비용을 고려한 비선형 수송문제 (Non-linear Transportation Problem)는 한 종류의 상품을 다수의 공급처에서 다수의 수요처로 수송 할 때, 총 수송비용 및 고정비용이 최소가 되도록 각 공급처와 수요처간의 수송량을 결정하는 문제로 현재 비선형 수송문제에 대한 다양한 해법들이 제안되고 있으며 그중에도 메타 휴리스틱이 가장 활발히 연구되고 있다. 본 논문에서는 이 메타휴리스틱 방법들 중에 가장 주목 받고 있는 유전 알고리즘을 이용하여 비선형 수송문제에 대한 해법을 제시하고자 한다. 유전 알고리즘을 적용함에 있어서 가장 중요한 것 중에 하나가 어떻게 해를 유전자 표현으로 나타낼 것인가인데 본 논문에서는 수송문제의 해를 걸침나무로 표현할 수 있다는 점에 착안하여 다양한 트리 표현법을 수송 문제에 적용해 보고 수치 실험을 통해 그 성능에 대한 비교 연구를 한다.

키워드

비선형수송문제, 유전 알고리즘, 유전자표현, 걸침나무

1. 서 론

고정비용을 고려한 비선형 수송문제 (Non-linear Transportation Problem)는 한 종류의 상품을 다수의 공급처에서 다수의 수요처로 수송 할 때, 총 수송비용 및 고정비용이 최소가 되도록 각 공급처와 수요처간의 수송량을 결정하는 문제이다. 여기서 각 공급처와 수요처를 노드로 나타내고 각 수송경로를 가지로써 나타내면, 수송문제는 네트워크로 표현할 수 있다. 이와 같은 수송문제는 1941년 F. Hitchcock[9]이 처음으로 정식화 한 것으로 알려져 있다.

최근에 보편적이면서도 널리 실세계에 적용되는 최적화 문제를 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 소개되어져 왔다. 이들 가운데 유전 알고리즘은 매우 주목받는 최적화 방법들 중 하나이다 [2, 3].

Holland에 의해 소개된 유전 알고리즘은 최근

많은 연구자들로부터 아주 큰 관심을 불러 일으켰고 주로 공학 분야 및 OR 분야에서 매우 어려운 조합 최적화 문제를 풀기 위하여 사용되어 왔다. 최근에는 유전 알고리즘을 이용하여서 통신네트워크 설계 문제와 물류 창고와 고객 서비스 지역 할당 및 위치 설정 문제에 많이 사용되고 있다. 일반적으로 최적의 고정 비용 및 운송 비용을 고려한 수송 계획안을 찾아내는 것은 매우 복잡한 조합최적화 문제이며 NP-hard 최적화 문제로 분류된다. 전통적인 방법들을 이용하면, 문제의 규모가 지수적으로 확대되어지는 경우, 네트워크 토폴로지 설계 문제를 풀이하기가 힘들어진다. 이 때문에 유전 알고리즘을 기초로 하는 휴리스틱 알고리즘이 주목을 받고 있다.

고정 비용 수송 문제(Fixed Charge Transportation Problem)는 동질의 상품에 대하여 여러 개의 공장에서 소비자들에게 최소 비용의 수송 계획의 결정을 찾는 것이다. 이것은 각 공장에서 공급량과 각 소비자의 수요량 그리고 각 공장으로부터 각 소비자에게 수송비용과 고정 비용을 모두 고려한 것이다. 고정 비용 수송 문제의 목표는 각 소비자에게 요구를 만족 시키면서 기준에 최

적화하도록 각 공장에서 이용 가능한 공급을 할당 시키는 것이다. 일반적으로 목적 함수는 여러 가능한 할당 형태에 따른 가변적인 비용과 고정 비용을 고려한 것으로 이러한 2가지 비용을 모두 동시에 최소화 하는 것이다. 또한, 고정 비용 수송 문제는 제약조건을 포함하고 있으며 조합 최적화 문제로 정식화 된다. 어떤 경우에 있어서 가변적인 비용은 비용 함수가 비선형의 2차 변수와 관련이 있을 수 있다. 이때의 고정 비용 수송 문제는 쉽게 풀기 곤란한 NP-hard 문제로 인식될 수 있다.

본 논문에서는 비선형 수송문제를 위한 유전 표현법 중 Prüfer number, NetKey, LNB 세 가지 트리 표현법을 수송 문제에 적용해 보고 수치 실험을 통해 그 성능에 대한 비교 연구를 수행하였다.

II. 비선형 수송문제를 위한 유전자 표현법

2.1 Prüfer number

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 효율적이며, 특히 어떤 확장된 수송 문제와 같은 연구 분야, 생산, 분배 문제, 최소 신장 트리 문제 등에 대하여 더욱 더 적절하다고 주장하였다 [1].

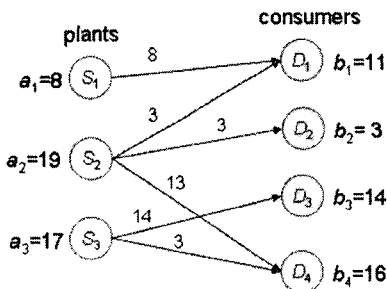
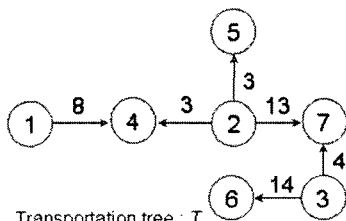


그림 1. 수송그래프의 예



Transportation tree : T

그림 2. 걸침나무 트리의 예

예를 들어 그림 1의 수송 그래프는 그림 2의 Prüfer number를 이용한 걸침나무와 그에 따른 Prüfer number $P(T) = [4\ 2\ 2\ 7\ 3]$ 으로 나타낼 수 있다.

2.2 Network Random Key (NetKey)

NetKey를 이용하는 방법은 조합 최적화문제를 해결하기 위해 Bean[5]에 의해 처음 제안된 방법이며 걸침나무(spanning tree) 문제에서 사용되는 NetKey 표현법은 기존의 MST를 해결하기 위한 Kruskal 알고리즘과 유사하다. Kruskal 알고리즘은 최소 비용 걸침나무 T를 찾기 위해 한번에 하나의 링크(link)를 선택하여 T에 추가해 나가는 방법이다. 이 알고리즘은 n개의 노드를 가진 그래프 G의 링크 집합 E(G)로부터 n-1개의 링크를 선정하는 것이며 링크의 선정 방법은 비용이 가장 적은 링크부터 선정하되 이미 먼저 선정된 링크들과 결합될 때 사이클이 만들어지는 것을 제외시킨다. 비용이 같은 링크들은 임의의 순서로 하나씩 선정할 수 있다. Kruskal 알고리즘의 핵심은 두 가지로, 첫째는 최소비용 순으로 가중치에 따라 오름 차순으로 정렬하고 링크를 차례로 선택하는 것이다. 두 번째는 만들고 있는 걸침나무 T에서 새로 선정된 링크가 사이클을 만들 가능성이 있는지를 판단하는 것이다.

Kruskal의 알고리즘은 네트워크 상의 링크들을 정렬하기 위해 링크의 비용값을 이용한다. 하지만 NetKey에서는 Random으로 부여된 벡터값을 이용한다. 이런 NetKey 표현법 역시 Kruskal의 알고리즘과 똑같은 결점을 지니고 있는데, 만약 네트워크가 완벽하게 연결되어 있고 많은 노드를 포함하는 경우, 많은 계산 시간과 메모리량을 필요로 하게 된다 [6].

2.3 Link and Node Biased

Link and Node Biased(LNB) 표현법은 palmer와 Kershenbaum[4]이 OCST문제를 해결하기 위해 제안한 방법이다. 이 방법은 가중치가 부여된 벡터(weight vector)를 사용해서 트리 네트워크의 구조를 표현하고 진화 알고리즘이 네트워크에 있는 노드와 링크들 중에서 우수한 것들을 가려낼 수 있도록 만든다. 이 표현법에서는 각 chromosome은 각 노드의 bias값을 이용해서 나타낸다. 각 노드 및 링크의 bias는 0과 255사이의 정수 값으로 표현되고, LNB의 코드와 일치하는 걸침나무(spanning tree)는 수정된 비용값(C')을 가지고 Prim 알고리즘을 수행하여 만들어 낸다. Prim 알고리즘은 처음에 주어진 하나의 노드 u를 트리의 노드 집합 V(T)에 포함시키는 것으로 시작한다. 노드 u는 주어진 그래프의 어떤 노드라도 관계가 없으며 V(T)에 있는 노드들과 인접한 노드들 중에서 최소 비용 링크 (u, v)를 선택하여 T에 포함시키고 새로 선정된 노드를 V(T)에 포함시킨다. 이 링크의 선택은 트리 T가 n-1개의 링크를 포함할 때까지 반복한다. 항상 새로 선택되는 링크 (u

, v)는 반드시 노드 u 나 v 어느 하나만 $V(T)$ 에 포함되어 있는 비트리 링크로서 최소 비용을 가진 링크여야 한다. 이 알고리즘에서는 최소 비용 링크를 선정할 때 T 에 있는 모든 노드에 대한 최소 비용을 가진 인접 노드들을 고려해야 한다. 즉, 최소 걸침나무의 노드 집합 $V(T)$ 에 어떤 새로운 노드를 첨가할 때 마다 모든 $(u, v) \in V(T)$ 에 대한 비용을 계산하며, 이 때 비용값을 수정하는 식은 다음과 같다.

$$C'_{(u,v)} = C_{(u,v)} + P_1 b_{(u,v)} C_{\max} + P_2 (b_u + b_v) C_{\max} \quad (1)$$

여기서 C_{\max} 는 그래프에서의 최대링크 비용(maximum link cost)이며, $b_{u,v}$ 는 노드 u, v 로 나타내어지는 링크의 bias값을, b_u 는 노드 u 의 bias값을 나타낸다. 노드와 링크의 bias값들은 우선 0과 1사이의 값으로 평균화(normalize) 시킨 다음에 (1)식을 적용한다. 그리고 P_1 과 P_2 는 제어 파라미터(control parameter)로 Palmer와 Kershenbaum은 실험에서 $P_1=0, P_2=1$ 의 값을 사용하였다. 즉, 링크에는 bias값으로 표현하지 않고, 각각의 노드만을 bias값을 이용하여 표현하였다.

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

III. 실험 및 결과

본 논문에서는 세 가지 표현법의 성능을 나타내

기 위해 위에서 제시한 세 가지 표현법을 이용하여 각각의 수치 데이터를 수집하고 이를 비교하였다. 이 비교를 위하여 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 Uniform 교배로 각 유전자가 독립적으로 교환될 수 있도록 하는 것으로서 Random으로 0 또는 1의 마크를 씌워 유전자를 교환하는 연산자이며, 돌연변이 연산자의 역순(inversion)연산자는 부모유전자에서 두 개의 절단점을 임의로 선택하고 절단점 사이의 인자들을 역순으로 하여 자손을 생산하는 방법이다. 전환(displacement)연산자는 부모 유전자에서 두 개의 절단점을 임의로 선택하고 절단점 사이의 인자들을 임의의 위치로 전위하여 자손을 생산하는 방법이다. 본 논문에서는 돌연변이 연산자를 위해 역순 연산자와 전환 연산자를 함께 사용하였고, 최소 비용값을 minvalue로 나타내었다.

각각의 표현법에 대한 비교를 위해 노드의 개수를 8개(3x5 : 3개의 plants와 5개의 consumers)와 10개(3x7), 15개(5x10), 30(10x20)로 하여 실험을 수행했으며, 모든 실험 데이터 상에서 각 알고리즘을 10번씩 수행하였고, 실험 결과는 이를 평균한 값이다. 종료 조건은 generation을 500으로 설정하였으며, 그 결과 데이터는 표 2와 같다.

노드의 개수가 8개일 때 Prüfer number는 10번의 수행결과가 동일하게 나왔으며, NetKey와 LNB도 빠른 수행 시간과 Prüfer number와 큰 차이를

표 1 각 알고리즘에 사용된 파라메타 및 연산자들

	Prüfer number	NetKey	LNB
Population 크기	100	100	100
교배확률	0.2	0.2	0.2
돌연변이 확률	0.4	0.4	0.4
교배 연산자	Uniform	Uniform	Uniform
돌연변이 연산자	Inversion, displacement	Inversion, displacement	Inversion, displacement
종료조건	Generation = 500	Generation = 500	Generation = 500

표 2 실험결과 비교

		Prüfer number	NetKey	LNB
N=8 (3x5)	Avg. CPU time	5	7	2
	Avg. minvalue	17110	17424	21285
	best minvalue	17110	16800	18780
N=10 (3x7)	Avg. CPU time	7	8	2
	Avg. minvalue	17330	18649	23398
	best minvalue	17330	17500	22070
N=15 (5x10)	Avg. CPU time	15	7	6
	Avg. minvalue	19905	27388	24076
	best minvalue	19740	25240	21550
N=30 (10x20)	Avg. CPU time	720	43	3
	Avg. minvalue	34212	56538	64181
	best minvalue	32750	49970	58250

보이지 않았다. 하지만 LNB는 다른 두 개의 표현법과 비교했을 때 CPU time은 빠르지만 minvalue의 값이 가장 나쁘게 나왔다.

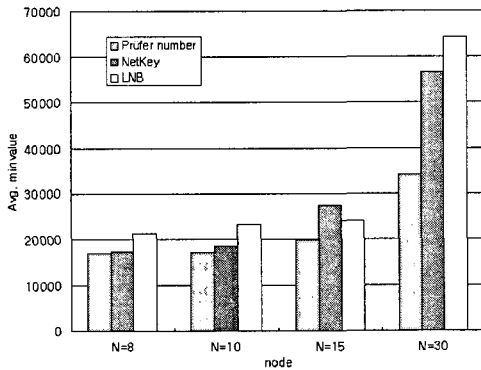


그림 3. 노드 개수에 따른 최선의 결과값들의 비교 (평균)

노드의 개수가 10개일 때 Prüfer number는 8개일 때와 같이 10번의 수행 결과가 동일하게 나온 것을 확인할 수 있었다. NetKey는 Prüfer number와 거의 비슷한 결과를 나타냈지만, LNB는 8개일 때와 같이 다른 두 개의 표현법보다 minvalue값이 나쁘게 나왔다.

노드의 개수가 15개 일 때 Prüfer number의 10번의 수행 시간이 10개일 때 보다 두 배 정도의 시간이 걸렸지만, 표현법들 중에 가장 좋은 값이 나왔다. NetKey와 LNB는 수행 시간이 Prüfer number보다 여전히 빨랐으며, minvalue 값은 큰 차이를 보이지 않았다.

노드 개수가 30개 일 때는 Prüfer number와 NetKey는 10번의 수행을 하는 시간이 큰 폭으로 증가 하였고, LNB의 수행 시간은 이전과 큰 차이를 보이지 않았다. 하지만 minvalue에서 세 가지 표현법의 결과 값은 큰 차이를 보였으며, Prüfer number가 가장 낮은 value 값을, LNB가 가장 높은 value 값을 산출해 냈다.

IV. 결론

본 논문에서는 고정비용을 고려한 비선형 수송 문제에 대해서 Prüfer number, NetKey, LNB 표현법을 이용하여 각각에 대한 성능 비교를 하였다. 노드의 개수가 적을 경우 Prüfer number는 균일한 최소비용을 산출해 냈으며, 노드의 개수가 점차 늘어날수록 약간의 변동은 있지만 안정된 최소비용을 산출해냈다. 하지만 노드의 개수가 늘어날수록 수행 시간이 큰 폭으로 증가함을 확인할 수 있었다. NetKey는 노드의 개수가 적을 경우 Prüfer number와 비슷한 수행 시간과 최소비용을 찾아냈다. 노드의 개수가 늘어날수록 수행 시간은 큰 폭으로 늘어났고, 노드의 개수가 적을 때도 Prüfer number의 최소비용보다 나쁜 값을 찾아내긴 했지만

큰 차이는 보이지 않았다. 하지만 노드의 개수가 늘어갈수록 그 차이는 점차 커짐을 확인할 수 있었다. LNB는 노드의 개수가 적을 때 많은 수행 시간은 크게 변하지 않았다. 하지만 최소비용은 다른 두 가지 표현법 중에서 가장 나쁜 값을 계속해서 산출해냈다.

본 실험을 통해 세 가지의 표현법을 비교해본 결과 Prüfer number 표현법이 다른 표현법에 비해 가장 안정되고 최적의 비용값을 산출 하였다.

이로 미루어 본 결과 TP 문제에 있어서 Prüfer number 표현법이 가장 적합하다고 본 논문에서는 제시한다.

참고문헌

- [1] Gen, M. Jung-bok Jo. "Nonlinear fixed charge transportation problem by spanning tree-based genetic algorithm", Computer & industrial engineering An International Journal, 2007(submitted).
- [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] C, C Palmer. "An approach to a problem in network design using genetic algorithms", Ph.D. Dissertation, Polytechnic University, 1994.
- [5] Bean, J. C. "Genetic algorithms and random keys for sequencing and optimization" ORSA Journal on Computing, Vol.6, No2, pp.155-160, 1994.
- [6] 석상문, 장석철, 변성철, 안병하, "최적통신 걸침 나무 문제를 해결하기 위한 진화 알고리즘" 정보과학회논문지 제32권 제4호, pp.268-276, 2005.
- [7] Rothlauf, F. Goldberg, D.E and Heinzl, A. "Network Random Keys-A Tree Network Representation Scheme for Genetic and Evolutionary Algorithms" Evolutionary Computation, Vol.10 (1), pp. 75-97, 2002.
- [8] Gaube, T. and Rothlauf, F. "The Link and Node Biased Encoding Revisited: Bias and Adjustment of parameter" EvoWorkshop 2001, LNCS Vol.2037, pp.1-10, 2001.
- [9] Hitchcock, F. L., "The Distribution of a Product from Several Sources to Numerous Localities", Journal of Mathematical Physic, Vol. 20, pp. 224-230, 1941.
- [10] 조정복, 고석범, M. Gen, "신장트리 기반 유전자 알고리즘에 의한 비선형 fcTP 해법" 한국정보과학회논문지:SA 제32권 제 8호, 2005.