

고정비용 비선형 수송문제를 위한 유전자 표현법들의 비교 연구

동서대학교 컴퓨터정보공학부
김병기, 장지훈, 김종율, 조정복

The Comparison of Genetic Representations for the Fixed Charge Non-linear Transportation Problems

Byung ki, Kim, Ji hoon Jang, Jong ryul Kim, Jung bok Jo

Dongseo University Computer Information Engineering Division

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

요 약

본 논문에서는 고정비용을 고려한 비선형 수송문제(Fixed Charge Non-linear Transportation Problem)에 대해 다룬다. 이는 한 종류의 상품을 다수의 공급처에서 다수의 수급처로 수송할 때, 총 수송비용과 고정 비용이 최소가 되도록 각 공급처와 수급처간의 수송량을 결정하는 문제이다. 현재 비선형수송문제에 대한 다양한 해법들이 제안되고 있으며 그 중에서도 메타 휴리스틱을 이용한 해법들이 가장 활발히 연구되고 있다. 본 논문에서는 메타휴리스틱 방법들 중에 가장 널리 이용되고 있는 유전 알고리즘을 이용한 해법을 제시하고자 한다. 유전 알고리즘을 적용함에 있어서 제일 첫 관문은 해의 유전자표현을 어떻게 나타낼 것인가이다. 본 논문에서는 수송문제의 해를 걸침나무로 표현할 수 있다는 점에 착안하여 다양한 트리 표현법을 수송문제에 적용해 보고 수치 실험을 통해 그 성능에 대한 비교 연구를 한다.

1. 서 론

수송문제는 1941년 F. Hitchcock[1]이 처음으로 정식화 한 것으로 알려져 있다. 고정비용을 고려한 비선형 수송문제는 한 종류의 상품을 다수의 공급처에서 다수의 수급처로 수송할 때, 총 수송비용 및 고정비용이 최소가 되도록 수송량을 결정하는 문제이다. 이것은 각 공급처에서 공급량과 각 수급처의 수요량 그리고 각 공장으로부터 각 수급처간의 수송비용과 고정 비용을 모두 고려한 것이다. 고정 비용 수송 문제의 목표는 각 수급처의 요구를 만족 시키면서 최적화하도록 각 공급처에서 공급 가능한 공급량을 적절히 할당 시키는 것이다. 여기서 각 공급처와 수급처를 노드(node)로 나타내고 각 수송경로를 에지(edge)로써 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, 고정 비용 수송 문제는 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 푸는데 다양한 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 처음 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 문제, 물류 창고와 고객 서비스 지역 할당 및 위치

설계 문제에 많이 사용되고 있다. 위의 문제들은 전통적인 방법들을 이용하면, 문제의 규모가 큰 경우, 네트워크 토폴로지 설계 문제를 풀이하기가 힘들어진다. 이 때문에 유전 알고리즘을 기초로 하는 메타 휴리스틱 알고리즘들이 주목을 받고 있다.

본 논문에서는 고정비용 비선형 수송문제를 위한 유전 표현법 중 Degree based 표현법, Edge based 표현법, Prüfer number 표현법 세 가지 트리 표현법을 수송문제에 적용해 보고 수치 실험을 통해 그 성능에 대한 비교 연구를 수행하였다.

Degree based 표현법, Edge based 표현법, Prüfer number 표현법은 최적 통신 걸침 나무 문제(Optimum Communication Spanning Tree Problem : OCSTP)를 해결하기 위해 제안된 표현법으로써 항상 트리 형성이 가능하다. 수송문제 적용 시, 제약조건으로 공급처간의 연결과 수급처간의 연결은 허용되지 않게 한다.

2. 고정비용 비선형 수송문제를 위한 유전자 표현법

고정비용 수송문제는 총 수송비용이 최소가 되도록 각 공급처와 수급처간의 수송량을 결정하는 것이다.

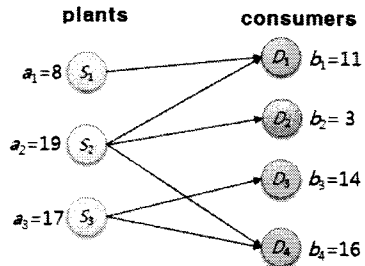


그림 1 수송그래프의 예

예를 들어 총 3개의 공급처에서 4개의 수급처로 수송을 할 때, a_m ($m=1,2,3$)은 각 공급처에서 공급할 수 있는 최대량이며, b_m ($m=1,2,3,4$)은 각 수급처에서 필요로 하는 요구량이다. 이 때 각 공급처에서 수급처로 상품을 수송할 때, 총 수송비용이 최소가 되도록 하여 그래프로 나타내면 그림 1의 수송그래프를 형성할 수 있다.

이와 같은 고정비용 수송 문제를 해결하기 위해 많은 표현법들이 제안되어 왔다. 이들 표현법들 중 Degree based 표현법, Edge based 표현법, Prüfer number 표현법에 대해 소개하였다.

2.1 Degree based(d -based) 표현법

d -based 표현법[2]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 개발한 표현법이다.

d -based 표현법은 위의 개념을 확장하여 CF-TCR (Cycle Free - Tree Construction Rule)이라는 방법을 이용한다. 노드의 개수가 n 일 때, $n \times 2$ 의 길이를 가진 유전자 표현을 고려해보면, 각각의 노드는 유전자 표현에 두 번씩 나타나야만 한다. 예를 들면 7개의 노드가 존재하고 이를 $n \times 2$ 길이의 노드 스트링을 형성하면 {3,1,1,4,2,5,2,7,6,3,6,4,5,7}과 같이 나타낼 수 있고, 이는 d -based 표현법으로 형성된 염색체의 예이다. 이렇게 형성된 스트링은 CF-TCR 방법을 이용하여 트리를 형성하게 된다. CF-TCR 방법은 먼저, $SelNode(f)$ 와 $EdgeSet(f)$ 를 설정한다. $SelNode(\cdot)$ 는 트리를 형성하며 선택된 노드의 집합이고, $EdgeSet(\cdot)$ 은 선택된 에지의 집합을 나타낸다. 노드 집합과 에지 집합을 설정하고 나면 스트링의 첫 번째 노드와 두 번째 노드를 비교를 하는데, 비교하는 두 개의 노드 중 하나라도 이전에 선택된 적이 없다면 두 개의 노드를 연결하고, 한 칸 옆으로 이동하여 다시 두 노드를 비교한다. 만약, 두 노드 모두 이전에 선택된 적이 있다면, 두 노드를 연결하지 않고 한 칸 옆으로 이동하여 다시 비교를 한다. 이렇게 연결하여 n 개의 노드들에 대해 에지의 개수가 $n-1$ 개만큼 연결이 되면 트리형성을 끝낸다. 하지만 위에서 제시한 d -based 표현법을 수송문제에 적용 할 경우 제약 조건이 생기게 된다.

수송문제에서 각 공급처와 수급처간의 관계에 의해서 공급처는 공급처끼리, 수급처는 수급처끼리 연결이 될 수 없다. 즉, 8개의 노드 중 1,2,3번 노드가 공급처이고, 4,5,6,7이 수급처라면 1,2,3 노드는 서로 연결할 수 없으며, 4,5,6,7 노드도 서로 연결을 할 수 없다. 이 제약 조건을 포함한 상태에서 위의 스트링 {3, 1,1,4,2,5,2,7,6,3,6,4,5,7}을 트리로 연결하면 그림 2와 같이 트리를 형성할 수 있다.

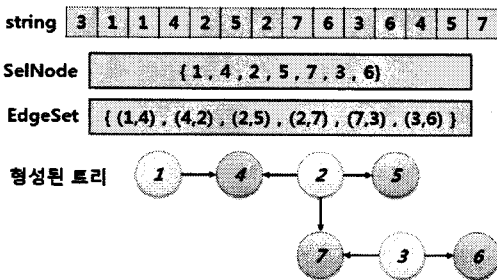


그림 2 수송문제 적용 시 d -based 트리 표현

d -based 표현법은 노드의 개수가 n 개일 때 $n \times 2$ 개의 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 -based 표현법이 사이클을 허용하지 않는 방법으로 트리를 생성하기 때문이다. 이를 해결하기 위해 e -based 표현법이 제안되었다.[3]

2.2 Edge based(e -based) 표현법

e -based 표현법[3]은 최적 통신 걸침 나무 문제를 해결하기 위해 제안된 방법이며, CB-TCR(Cycle Breaking - Tree Construction Rule)이라고도 한다. e -based 표현법은 $(n-1) \times 2$ 의 길이를 가지는 스트링을 이용하여 유전자를 표현한다. 단, 모든 노드는 최소한 한번씩은 나타나야 하고, 이 때 본 표현법으로 $n-1$ 개의 에지를 형성할 수 있게 된다. 이렇게 만들어진 스트링을 이용하여 트리를 구성하게 된다.

e -based 표현법은 우선 d -based 표현법과 같은 방법으로 에지를 추가해 나가다가 비교하는 두 노드가 연결될 경우 사이클을 형성하게 되는데 이를 형성하는 에지들을 찾은 다음 가장 비용값이 큰 에지를 제거하는 방식으로 트리를 구성한다.

e -based 표현법의 절차는 먼저 선택된 노드의 집합 $NodeSet(f)$ 와 선택된 에지들의 집합 $EdgeSet(f)$ 를 설정한다. 첫 번째로 선택된 노드를 $NodeSet(\cdot)$ 에 추가하고 다음노드를 비교하게 되는데, 이때 $NodeSet(\cdot)$ 에 포함되어있지 않을 경우 $NodeSet(\cdot)$ 에 추가하고 형성된 에지를 $EdgeSet(\cdot)$ 에 추가 시킨다. 그리고 d -based 표현법과 마찬가지로 스트링을 이동하여 이 과정을 수행 하면서 에지를 형성하는데, 이미 $NodeSet(\cdot)$ 에 포함되어있는 노드가 발생되었을 경우 사이클을 형성하게 된다. 이때 사이클을 형성하는 모든 에지를 찾고 그 에지들 간의 거리를 비교하고 가장 긴 에지를 제거하여 사이클을 깨뜨린다. $n-1$ 개의 에지 형성 시 종료를 하게 되고, 아니면 이를 반복 수행한다.

만약 7개의 노드가 존재하고 이를 $(n-1) \times 2$ 길이의 노드 스트링을 형성하면 {3,1,4,2,5,1,2,7,3,6,7,4}와 같이 나타낼 수 있다. e -based 표현법도 d -based 표현법과 마찬가지로 수송문제에 적용을 할 경우 제약 조건이 생기게 된다.

수송문제에서 각 공급처와 수급처간의 관계에 의해서 공급처는 공급처끼리, 수급처는 수급처끼리 연결이 될 수 없다. 즉, 7개의 노드 중 1,2,3번 노드가 공급처이고, 4,5,6,7이 수급처라면 1,2,3 노드는 서로 연결할 수 없으며, 4,5,6,7 노드도 서로 연결을 할 수 없다. 이 제약 조건을 포함한 상태에서 위의 스트링 {3,1,4,2,5,1,2,7,3,6,7,4}을 트리로 연결하면 그림 3과 같이 트리를 형성할 수 있다.

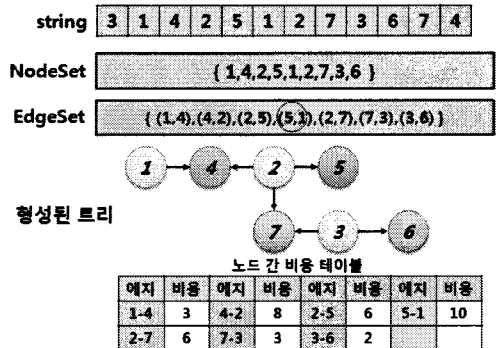


그림 3 수송문제 적용 시 e -based 트리 표현

e -based 표현법은 노드의 개수가 n 개일 때 $(n-1) \times 2$ 개의 스트링을 형성한다. 이는 d -based 표현법 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 계산시간이 오래 걸린다.

2.3 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.[4][5] 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 [6].

Prüfer number를 이용한 트리 형성 시 노드의 개수가 n 개일 때, $n-2$ 개 길이의 노드 스트링을 형성하게 되며, d -based 표현법과 e -based 표현법과 같이 1,2,3번 노드가 공급처 노드라 할 때 공급처 노드끼리는 서로 연결을 할 수 없고, 4,5,6,7이 수급처 노드라고 할 때, 수급처 노드끼리는 연결할 수 없다. 그리고 예지의 개수는 $n-1$ 개가 생성이 된다.

Prüfer number는 노드의 개수가 n 개일 때 $n-2$ 개 길이의 노드 스트링을 형성하며 이를 이용하여 트리를 형성하게 된다. Prüfer number의 스트링 $P(T)$ 가 {4,2,2,7,3}일 때, $P(T)$ 가 아닌 나머지 노드를 $P'(T)$ 로 지정하면 $P'(T)$ 는 {1,5,6}이 된다. $P(T)$ 의 가장 왼편의 노드를 j , $P'(T)$ 의 가장 왼편의 노드를 i 라 하고 i 와 j 가 서로 공급처Set 또는 수급처Set가 아니면 트리 T 에서 (i,j) 를 연결한다. 만약 i 와 j 가 같은 공급처Set 또는 수급처Set에 포함이 된다면 $P(T)$ 로부터 다음 수 k 를 선택하고 j 와 k 를 바꾸 후 트리 T 에서 (i,k) 를 연결하고 $P(T)$ 의 j (또는 k)를 제거한다. 만약 $P(T)$ 의 남은 노드 중 j (또는 k)와 같은 노드가 없을 경우 $P'(T)$ 의 i 를 삭제하고 이 위치에 j 를 넣어 j' 으로 설정한다. $P(T)$ 의 새로운 j 와 $P'(T)$ 의 j' 을 트리 T 에 연결하면 (j',j) 가 되고 $P'(T)$ 의 j' 를 삭제한다. 이때 j' 이 $P'(T)$ 에 남은 노드와 같은 Set에 포함된다면 j' 을 $P'(T)$ 의 가장 오른편 위치에 넣고, $P(T)$ 와 $P'(T)$ 의 남은 노드로 계속해서 트리를 형성해 나간다. $P(T)$ 의 모든 노드를 $P'(T)$ 와 연결하여 더 이상의 노드가 $P(T)$ 에 남아 있지 않고 $P'(T)$ 에 두 개의 노드가 남으면 이 두 개의 노드를 서로 연결한다.

이와 같은 방법으로 만들어진 Prüfer number $P(T) = [4\ 2\ 2\ 7\ 3]$ 는 그림 4와 같으며 트리로 나타 낼 수 있다.

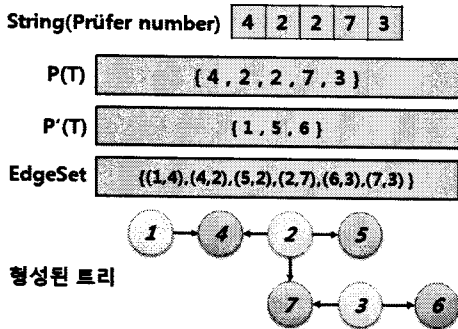


그림 4 수송 신장 트리의 예

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.

본 논문의 실험을 위해 사용된 교배연산자는 균일교배(Uniform Crossover)로 각 유전자가 독립적으로 교환될 수 있도록 하는 것으로서 Random으로 0 또는 1의 마크를 띄워 유전자를 교환 하는 연산자이다.

표 1 각 알고리즘에 사용된 파라메타 및 연산자들

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

돌연변이 연산자로 역순(inversion)연산자와 전환(displacement) 연산자를 함께 사용 하였다. 역순 연산자는 부모유전자에서 두 개의 절단점을 임의로 선택하고 절단점 사이의 인자들을 역순으로 하여 자손을 생산하는 방법이다. 전환연산자는 부모 유전자에서 두 개의 절단점을 임의로 선택하고 절단점 사이의 인자들을 임의의 위치로 전위하여 자손을 생산하는 방법이다. 그리고 선택연산자로 ($\mu+\lambda$)선택법을 이용하였다. ($\mu+\lambda$) 선택법은 μ 개의 개체로 된 부모의 집단 중에서 λ 개의 개체의 자손을 생성한다. 이때 λ 개의 ($\mu+\lambda$)개체의 집단 중에서 μ 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 하여 실험을 수행했으며, 모든 실험 데이터 상에서 각 알고리즘을 10번씩 수행하였고, 실험 결과는 수행 후 출력된 결과값에 대한 가장 낮은 최소값(best minvalue)과 이에 대한 표준편차를 중심으로 나타내었다. 종료 조건은 generation 을 500으로 설정하였으며, 그 결과 데이터는 표 2와 같다.

노드의 개수가 8개일 때 Prüfer number 표현법, d-based 표현법, e-based 표현법 중 Prüfer number 표현법이 가장 낮은 best minvalue를 산출하였으며, 10회 모두 같은 minvalue를 산출하여 표준편차의 변동이 없었다. e-based 표현법은 Prüfer number와 같은 best minvalue를 산출하였으나 표준편차에서 가장 나쁜 변동폭을 보였으며, d-based 표현법이 minvalue에서 가장 나쁜 값을 산출하였다. 하지만 평균 수행 시간에서는 e-based 표현법이 가장 빠른 수행 시간을 보였다.

노드의 개수가 10개일 때 Prüfer number 표현법이 가장 낮은 best minvalue를 산출하였으며, 10회의 수행 모두 같은 minvalue를 산출하여 표준편차의 변동이 없었다. e-based 표현법은 Prüfer number 표현법에 비해 minvalue를 큰 차이를 보이지 않았지만 노드의 개수가 8개일 때와 같이 표준편차에서 가장 나쁜 변동폭을 보였으며, d-based 표현법이 minvalue에서 가장 나쁜 값을 산출하였다. 평균 수행 시간에서도 e-based 표현법이 가장 빠른 수행 시간을 보였다.

노드의 개수가 13개일 때 Prüfer number 표현법의 best minvalue와 평균 minvalue가 동일하지만 10회의 수행동안 큰 지 않은 변동으로 인해 매우 낮은 표준편차를 기록하였다. 다음으로는 e-based 표현법의 best minvalue가 최소값을 산출하였지만, d-based 표현법의 best minvalue와 매우 근사한 최소 값을 산출하였다. 표준편차는 이전의 실험에서와 같이 e-based 표현법이 가장 나쁜 표준편차를 보였다.

노드의 개수가 15개일 때 Prüfer number 표현법의 best minvalue가 다른 두 표현법 보다 더 나은 최소값을 산출하였고, 표준편차도 큰 차이를 보이지 않았다. e-based 표현법이

Prüfer number 표현법 다음으로 최소의 minvalue를 산출하였지만 *d*-based 표현법에 비해 큰 차이를 보이지 않았다. 표준편차 역시 이전의 실험에서와 같이 *e*-based 표현법이 가장 나쁜 값을 기록했다.

표 2 실험결과 비교

		Prüfer	<i>d</i> -based	<i>e</i> -based
N=8 (3×5)	Avg. CPU time	0.5	0.5	0.2
	Avg. minvalue	17110	17542	17379
	best minvalue	17110	17370	17110
	표준편차	0	190	356
N=10 (3×7)	Avg. CPU time	0.8	0.8	0.6
	Avg. minvalue	17330	18333	18345
	best minvalue	17330	17920	17610
	표준편차	0	302	462
N=13 (5×8)	Avg. CPU time	0.7	1.3	1
	Avg. minvalue	20780	24559	24555
	best minvalue	20780	23420	23290
	표준편차	19.5	706	788
N=15 (5×10)	Avg. CPU time	1.5	6.3	5
	Avg. minvalue	19963	24665	24981
	best minvalue	19790	23050	22840
	표준편차	87.2	1094	1163

전체 실험 결과 Prüfer number 표현법이 다른 두 표현법에 비해 가장 좋은 minvalue를 산출 하였고 표준편차의 변동폭도 낮아 가장 높은 신뢰성을 보였다. 그리고 노드의 개수가 늘어남에 따라 minvalue가 증가하는 하였지만 큰 폭으로 증가하지는 않았다. *d*-based 표현법은 다른 두 표현법에 비해 minvalue가 가장 나쁘게 산출되었다. 표준편차는 노드의 개수가 적을 때는 *e*-based 표현법과 비교해서 변동폭이 *e*-based 표현법 보다 더 나은 변동폭을 보였지만 노드의 개수가 증가함에 따라 *e*-based 표현법의 표준편차가 크지 않은 변동폭을 보였다. *e*-based 표현법은 노드의 개수가 적을 때는 Prüfer number 표현법과 동일한 minvalue를 산출하여 좋은 결과를 보였지만 노드의 개수가 커짐에 따라 Prüfer number 표현법의 minvalue와의 큰 차이를 보였으며 *d*-based 표현법의 minvalue와 근사한 값의 minvalue를 나타내었다.

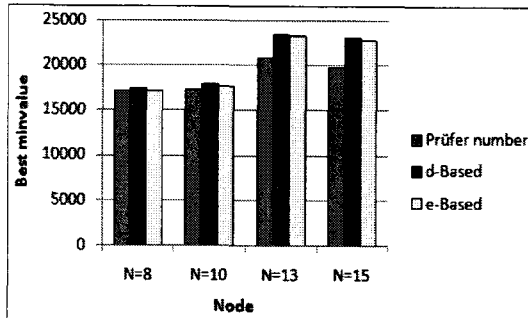


그림 5 노드개수에 따른 최소의 결과값 비교

또한 이론상의 *d*-based 표현법과 *e*-based 표현법이 실 실험에서는 다른 결과를 보이는 것을 확인 할 수 있었다. 가장 큰 차이는 수행 시간의 차이 이다. 이론에서는 *e*-based 표현법이 사이클을 형성하는 예제들을 찾는 과정으로 인해 많은 계산 시간이 걸린다고 하였으나 실험을 통한 수행 시간의 결과는 *e*-based 표현법이 *d*-based 표현법에 비해 빠른 수행 시간을 보였다. 또한 *d*-based 표현법의 단점을 보완하기 위해 제안된 *e*-based 표현법이 노드의 개수가 증가함에 따라 *d*-based 표현법

과의 minvalue가 근접해 진다는 것을 알 수 있었다.

4. 결론

본 논문에서는 고정비용을 고려한 비선형 수송문제에 대해 *d*-based 표현법과 *e*-based 표현법, Prüfer number 표현법을 이용하여 각각에 대한 성능 비교를 하였다.

Prüfer number 표현법은 노드의 개수가 적은 경우, 표준편차가 0으로 균일한 minvalue를 산출해 냈으며, 노드의 개수가 점차 늘어날수록 약간의 변동은 있지만 안정된 최소비용을 산출해냈다.

d-based 표현법은 노드의 개수가 늘어날수록 수행 시간은 큰 폭으로 늘어났고, 다른 세 가지 표현법 중에서 가장 나쁜 minvalue를 산출하였다. 그리고 노드의 개수가 늘어갈수록 표준편차가 큰 폭으로 증가함을 알 수 있었다.

e-based 표현법은 노드개수가 적을 때에는 Prüfer number 표현법과 비슷한 결과 값을 산출하였으며, 노드의 개수가 증가할수록 Prüfer number 표현법에 비해 나쁜 값을 산출하였지만, *d*-based 표현법에 비해 좋은 값을 산출하였다. 표준편차는 다른 두 표현법에 비해 높은 수치를 보였다.

그리고 *d*-based 표현법과 *e*-based 표현법이 이론과 실험 결과가 서로 차이를 보이는 것은 항상 트리 형성이 가능한 두 표현법이 수송문제 해결을 위해 각각의 표현법에 적용한 제약 조건으로 인하여 트리 형성에 많은 제약을 받게 되고 그로 인해 두 표현법의 실험 결과에 영향을 미친것이라고 판단된다. 또한 *e*-based 표현법에서의 수행 시간은 노드의 개수가 적을 경우에는 큰 영향을 미치지 않으나 노드의 개수가 증가함에 따라 수행시간이 점차 큰 폭으로 증가한다는 것을 실험을 통해 예측 가능하다.

본 실험을 통해 세 가지의 표현법을 비교해본 결과 Prüfer number 표현법이 다른 표현법에 비해 가장 안정되고 최적의 비용값을 산출 하였다.이로 미루어 본 결과 비선형 수송 문제에 있어서 Prüfer number 표현법이 가장 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] 석상문, 안병하, "차수 제약 걸침문제를 해결하기 위한 트리 표현법", 한국정보과학회 추계 학술 대회, 2003.
- [3] 석상문, 장석철, 변성철, 안병하, "최적 통신 걸침 나무 문제를 해결하기 위한 진화 알고리즘", 한국정보과학회 논문지, 소프트웨어 및 응용 제 32권 제 4호, 2005.4.
- [4] Gen, M. and Cheng, R., "Genetic Algorithms and Engineering Design", John Wiley & Sons, New York, 1997.
- [5] Gen, M. and Cheng, R., Genetic Algorithms and Engineering Optimization, John Wiley & Sons, New York, 2000.
- [6] 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).
- [7] 조정복, 고석범, M. Gen, "신장트리 기반 유전자 알고리즘에 의한 비선형 fCTP 해법" 한국정보과학회논문지:SA 제 32권 제 8호, 2005.