판매원 문제를 이용한 2-상품 네트워크 흐름 문제의 효율적인 계산방법

황 인 극*, 박 동 진*, 윤 광 식**

* 공주대학교 산업공학과 ** 전남대학교 농공학과

The Efficient Computation Method of Two-commodity Network Flow Problem Using TSP

Hwang In keuk*, Park Dong jin*, Yoon Kwang sik**

*Dept. of Industrial Engineering, Kongju National University

**Dept. of Agricultural Engineering, Chonnan National University

ABSTRACT

Our interest in this paper is in the efficient computation of a good low bound for the traveling salesman problem and is in the application of a network problem in agriculture.

We base our approach on a relatively new formulation of the TSP as a two-commodity network flow problem. By assigning Lagrangian multipliers to certain constraints and relaxing them, the problem separates into two single-commodity network flow problems and an assignment problem, for which efficient algorithms are available.

1. 서론

농작물을 저장하는 창고의 크기가 점차 대형화되고, 창고의 내부 온도가 고온 혹은 고냉이 필요한 경우가 많아짐으로 인하여 사람이 직접 저장물에 접근하기 어려워지는 경향이 있다.

이런 경우에는 일반적으로 AGV(Automated Guided Vehicle) 혹은 S/R (Storage/ Retrieval) 기계가 사람을 대신하여 물건을 저장하고 반출한다. 이러한 기계들은 움직이는 거리에 따라 많은 비용이 초래되기 때문에 경영자는 될 수 있는대로 기계가 움직이는 거리를 최소화하는 노력을 경주하게 된다. 이러한 유형의 문제는 판매원 문제를 적용함으로 해결할 수 있다.

이 연구는 대규모 양조장이나 혹은 대형 농업창고에서 볼 수 있는 2-상품(two- commodity) 네트워크 흐름에 관한 문제이다. 이 문제는 네트워 상에서 정의되고, 시작점에서 가고자하는 모든 장소를 단지 한번만 방문하고 다시 원점으로 돌아와야 하는 판매원 문제(Traveling salesman problem)와 함께 고려되어 진다. 일반적인 판매원 문제와 다른 점은 2-상품을 함께 다룬다는 점이다. 즉, 판매원 문제는 물건의 개수와는 상관없이 각 도시를 단지 한번씩 방문하는 최단의 거리에 관심을 두는 반면 2-상품 네트워 문제는 각 도시를 방문하는 최단거리 뿐만 아니라 수량에도 관심을 갖는다는 점이 다르다.

이 문제를 풀기 위한 수학적 모델은 Finke et al.(1984)

제시되었다. 그러나 TSP가 문제의 크기 또는 복잡도의 정도에 따라 이를 풀기 위한 해법의 가능한 최대 계산수가 지수함수에 비례하여 증가하는 NP-complete 문제이기 때문에 도시의 숫자가 증가하면 정확한 해를 찾기 힘들고 또한 TSP 문제의 정확한 해를 제시할 수 있는 일반적인 algorithm 역시 발견되지 않았다. 이 논문에서는 여행자문제에 대하여 보다 적은 computaion time과, 좋은 해를 제시하는 라그랑지안(Lagrangian Relaxation) 방법을 전개할 것이다.

2. TSP와 2-상품 네트워크 문제

2. 1 판매원 문제

판매원 문제는 다음과 같이 설명되어 질 수 있다. 판매원은 n개 도시를 각각 한번씩 방문하여야 하며, 돌아다닌 총 거리를 최소화하면서 다시 시작한 곳으로 돌아오는 순서를 정하는 것이다. 주어진 마디(nodes)들과 각 마디사이의 거리(혹은 각 마디사이를 움직이는데 드는 비용)가 주어지면, 경로(path)는 각 마디를 단지 한번만 통과하여 초기 마디에 되돌아 와야 하는데 이것은 Hamiltonian circuit이 될 것이다. 그러므로 최소의 여행 거리를 갖는 Hamiltonian circuit을 찾는다는 것은 판매원 문제로써 정의되어 질 수 있다.

이런 종류의 최적화문제를 푼다는 것은 매우 어려운 일이다. 왜냐하면 이 문제는 NP - complete 문제로 잘 알려져 있고, 정확한 해를 보장하는 어떠한 알고리듬도 존재하지 않고 오직 문제의 크기에 따라 지수적으로(exponentially)으로 증가하는 계산상의 노력을 필요로 하기 때문이다.

TSP 문제의 공식은 다음과 같이 정의되어 질 수 있다. 판매원이 방문하여야 하는 n 개의 도시가 있다고 가정하자. 그리고 Cij(ij=1, 2, ..., n)는 각 도시간에 여행비용이라고 하자. 만약 i에서 j까지 갈 수 없다면 그 때의 Cij=∞이라 한다. 문제의 공식은 정수(0 혹은 1) 흐름 변수 xij (i, j=1, 2, ..., n)을 포함하는데, 만약 도시 j까지 직접 갈 수 있다면, 그 때 변수 xij, i≠j, j=1, 2, ..., n,는 1의 값을 갖고, 만약 그렇지 못하다면 xij=0의 값을 갖는다. 그 때 최적의 경로는

$$\text{Min} \sum_i \sum_j c_{ij}\, x_{ij}$$

제약조건

$$\sum_j x_{ij} = 1\,(i = 1, 2, \ldots, n) \tag{1}$$

$$\sum_i x_{ij} = 1\,(i = 1, 2, \ldots, n) \tag{2}$$

$$x_{ij} \subseteq 0, 1$$

$$i \neq j$$

를 해결함으로 얻을 수 있다.

2. 2 2-상품 (Two-commodity) 네트워크 흐름

Finke, Claus, and Gunn (1984)는 TSP의 새로운 모델을 제시하였는데, 그것은 2 상품, P와 Q를 다루는 네트워크 문제이다. 그들이 제시한 네트워크 문제는 다음과 같다. 판매원은 장소 1에서 시작하여 Hamiltonian circuit을 수행한다. 판매원은 상품 P를 시작점에서 n-1만큼 사용할 수 있다. 방문중 첫 번째 장소에서 상품 P를 한 단위 내려놓고 그리고 두 번째 상품 Q를 한 단위 가져간다. 그러므로 그는 다시 다음 장소에도 P와 Q를 합하여 총 n-1단위를 가지며, 그 때 n-1단위는 P가 n-2단위 그리고 Q는 1단위로 구성되어 진다. 각 장소에서 상품 P와 상품 Q의 교환은 여행의 끝, 즉 다시 시작점으로 돌아올 때는 Q의 수는 n-1단위가 될 것이고 P의 수는 0이 될 것이다.

이 문제의 실제 활용은 우리 주변에 쉽게 발견할 수 있다. 농산물 창고에서 혹은 양조장, 우유 배달 등 여러 분야에 응용영역을 확장할 수 있다. 우리는 얼마 전 까지 병에 우유를 담아 배달하는 것을 볼 수 있었다. 처음 우유 배달할 때, 우유가 담긴 n-1개의 병을 가지고 출발해서, 첫 장소에서 빈 병을 수거하고 새 우유를 내려놓는데, 이 작업은 모든 우유를 배달할 때까지 계속된다. 최종적으로 빈 병은 시작점에 되돌아올 때 n-1이, 그리고 처음 출발할 때 가지고 갔던 새 우유의 수는 0이 되어야 한다.

총 개수가 n이 아니라 n-1이 되는 이유는 농작물 등 물건을 저장할 경우, 먼저 원하는 저장물을 수거하고 새로운 저장물을 저장한다는 것을 고려한 것이다. 이 문제의 최적해를 얻기 위한 공식은 다음과 같다.

$$\text{Min } Z = \sum_i \sum_j C_{ij}\,(x^p_{ij} + x^q_{ij})/(n-1)$$

제약조건

$$\sum_j x^p_{ij} - \sum_j x^q_{ij} = P_i \qquad i \in (1, 2, \cdots, n) \tag{3}$$

$$\sum_j x^p_{ij} - \sum_j x^q_{ij} = Q_i \qquad i \in (1,2,\cdots,n) \tag{4}$$

$$\sum_j (x^p_{ij} + x^q_{ij}) = n-1 \qquad i \in (1,2,\cdots,n) \tag{5}$$

$$x^p_{ij} + x^q_{ij} = (n-1)\, y_{ij} \qquad i,j \in (1,2,\cdots,n) \tag{6}$$

$$y_{ij} \in 0,1$$

$$x^p_{ij} \ge 0 \qquad x^q_{ij} \ge 0 \tag{7}$$

X^p는 상품 P에 대한 가능한 흐름을 나타내며, Xq는 상품 Q에 대한 가능한 흐름을 나타낸다.

P_i는 장소 i에서 상품 P의 흐름을 나타내며,

$$P_i = \begin{cases} n-1, & \text{if } i=1 \\ -1, & otherwise \end{cases}$$

Q_i는 장소 i에서 상품 Q의 흐름을 나타낸다.

$$Q_i = \begin{cases} -(n-1), & \text{if } i=1 \\ 1, & otherwise \end{cases}$$

Pi와 Qi의 양의 값은 공급을 나타내고, 반면 음수는 장소 i 에서의 수요를 나타낸다.

3. 2-상품 문제를 위한 라그랑지안 방법

위의 문제를 풀기 위하여 Kusiak and Finke(1987)은 LP 완화법(relaxation)을 이용하여 분기 가지법(Branch and Bound)으로 문제 해결을 제안하였다. 그러나 그들의 방법은 이 문제가 네트워크 구조를 갖는다는 것을 무시한 것이었다. 이 문제의 보다 좋은 해를 얻기 위한 방법으로 제약식 (5)와 (6)을 완화시키는(relaxed) 라그랑지안 방법이 보다 좋은 해를 제시해 줄 수 있을 것으로 보인다. 왜냐하면 가장 복잡한 제약식 (5)와 (6)을 목적함수로 이동시켜 문제를 간소함으로 보다 안정적인 해를 얻을 수 있기 때문이다.

제약식 (5)와 (6)을 완화시키기 위해서 라그랑지안 승수들(multipliers) μ와 λ가 이용된다. 이것은 2-상품 네트워크 문제를 두 개의 독립된 하나의 상품문제로 분리시키고 또한 이진변수(binaryvariable) yij를 비제약식(unconstraint) 최적화 문제로 해결할 수 있다.

수식으로 표현하면 다음과 같다.

$$PR_{\lambda,\mu} = \text{Min} \sum_i \sum_j C_{ij}(x^p_{ij} + x^q_{ij})/(n-1)$$
$$+ \sum_i \lambda_i \Big(\sum_j (x^p_{ij} + x^q_{ij}) - [n-1]\Big)$$
$$+ \sum_i \sum_j \mu_{ij}(x^p_{ij} + x^q_{ij} - [n-1]\, y_{ij})$$

제약조건

$$\sum_j x^p_{ij} - \sum_j x^q_{ij} = P_i \qquad i \in (1,2,\cdots,n)$$

$$\sum_j x^p_{ij} - \sum_j x^q_{ij} = Q_i \qquad i \in (1,2,\cdots,n)$$

$$\sum_j y_{ij} = 1 \qquad i \in (1,2,\cdots,n)$$

$$\sum_i y_{ij} = 1 \qquad j \in (1,2,\cdots,n)$$

$$y_{ij} \in 0,1$$

$$x^p_{ij} \ge 0$$

$$x^q_{ij} \ge 0$$

위의 식은 3가지 문제로 다시 나눌 수 있다.

(1) 상품 P에 대한 최소 비용 네트워크 문제

$$\text{Min} \sum_i \sum_j \Big[\Big(\frac{1}{n-1}\Big) c_{ij} + \lambda_i + \mu_{ij}\Big] x^p_{ij}$$

제약조건

$$\sum_j x^p_{ij} - \sum_j x^p_{ji} = P_i \qquad i \in (1,2,\cdots,n)$$

$$x^p_{ij} \ge 0 \qquad \text{for all i, j}$$

(2) 상품 Q 에 대한 최소 비용 네트워크 문제

$$\text{Min} \sum_i \sum_j \Big[\Big(\frac{1}{n-1}\Big) c_{ij} + \lambda_i + \mu_{ij}\Big] x^q_{ij}$$

제약조건

$$\sum_j x^q_{ij} - \sum_j x^q_{ji} = Q_i \qquad i \in (1,2,\cdots,n)$$

$$x^q_{ij} \ge 0 \qquad \text{모든 i, j에대해}$$

(3) y에 대한 할당 문제:

$$\text{Min} \; -\sum_i \sum_j [n-1]\, \mu_{ij}\, y_{ij}$$

제약조건:

$$\sum_j y_{ij} = 1 \qquad i \in (1,2,\cdots,n)$$

$$\sum_i y_{ij} = 1 \qquad j \in (1,2,\cdots,n)$$

$$y_{ij} \in 0,1$$

두 개의 1-상품 네트워크 문제는 Out-of-Kilter 방법으로 효과적으로 풀어 질 수 있고 또한 할당문제는 헝가리안 방법 등으로 풀어 질 수 있다.

4. 예

어느 축산업자는 사육용으로 음식물을 수거하기 위해서 4장소를 방문한다고 가정하자. 처음 출발점에서는 4개의 빈 통과 함께 출발하여, 첫 장소에 도착하면, 채워진 음식통을 수거하면서, 가지고 간 새로운 통을 내려 놓는다. 그 다음 장소에서도 역시 처음 장소에 행하였던 것과 같은 일을 반복하고, 마지막 장소를 방문하고 난 뒤에는 빈 4개의 통 대신에 음식이 채원진 4 개의 통으로 대체되어 돌아오게 된다. 이 문제의 조건은 총 수량(=4)은 언제나 동일하여야 하며, 방문한 장소는 다시 방문하지 않으면서 최단거리를 이용하여 수행되어야 한다는 것이다.

표 4.1에서는 4X4의 비대칭적 거리(혹은 비용) 행렬과 대칭행렬을 나타낸다. 상품 P에 대한 네트워크는 그림 1에서 보여진다. Out-of-kilter방법은 모든 노드(node)안으로 들어오는 흐름은 0이 될 것을 요구한다. 그림 1에서 나타난 화살표 위의 (i, j)는 (Lij, Uij)를 나타내는데, Lij는 하한 값을 Uij는 상한 값을 나타낸다.

위에 설명한 방법을 사용하여, 우리는 예제에 대한 완벽한 해를 얻을 수 있다. 이 과정은 그림 3, 4, 5에서 보여지며, 그림 5에서는 각 화살표 위의 괄호 안의 두 숫자는 두 상품, x_{ij}^{p}와 x_{ij}^{q} 의 흐름을 나타낸다.

표 4.1. 예를 위한 비대칭형 비용 행렬

		까지			
		1	2	3	4
부터	1	0	24	41	60
	2	23	0	13	30
	3	36	1	0	12
	4	58	19	14	0

최적의 해는 109의 비용을 가진 (1, 2, 3, 4, 1)의 경로를 가진다. 우리의 해는 모든 제약조건 (3), (4), (5), (6), (7)을 만족한다. 즉, 선택된 시작점 장소1은 상품 P를 n-1=3 단위 이용할 수 있고, 각 장소에 한 단위씩 할당한다. 상품 P와는 반대로, 각 장소에서 한 단위의 상품 Q를 얻고, 최종적으로 원점에 돌아오면, 상품 Q는 n-1=3 단위가 되어 돌아온다.

대칭인 경우의 최적해는 110의 비용을 갖는 (1, 2, 3, 4, 1)의 경로를 갖는 동일한 결과를 얻게 되었다.

5. 계산 결과

우리는 이 라그랑지안 방법이 일반적으로 사용되는 분기가지 방법을 이용하여 LP 완화법을 계산함으로써 우리가 전개한 방법이 2-상품 네트워크 흐름 문제에 적합한지 아닌지를 비교하여 보았다.LP 완화법은 R.E. Marsten[1987]에 의해 프로그램화된 분기가지법을 사용하였다. 또한 최적값을 얻기 위해 최근 개발된 Branch-&-Bound방법에 기초를 두어 만들어진 ZOOM(Zero-One Optimization Method)을 사용하였고, 우리의 방법과 ZOOM으로부터 얻을 수 있는 해의 CPU와 그 값들을 비교하여 보았다.

표 4. 2. 예를 위한 대칭형 비용 행렬

		까지			
		1	2	3	4
부터	1	0	24	41	60
	2	24	0	13	30
	3	41	13	0	12
	4	60	12	24	0

표 5. 1. 변수와 제약식의 수

크기	제약식의 수	제약변수의 수	정수변수의 수
4	38	42	18
6	92	110	50

표 5. 2. 해의 비교

N	Matrix	Lagrangian 방법에 의한 해	LP 완화법에 의한 해	ZOOM에 의한 해
4	대칭	133	125.3	141
	비대칭	151	148.7	156
6	대칭	201	172.7	202
	비대칭	278	262	293

표 5.1은 각 마디(node) 수에 따른 제약 식과 제약 변수 그리고 정수 변수의 크기를 나타내고 있다. 크기가 조금 증가함에도 제약식과 변수의 크기가 기하 급수적으로 늘어나는 것을 확인할 수 있다.

표 5. 3. 평균 CPU 시간의 비교.

	N	대칭형(Symmetric)			비대칭형(Asymmetric)		
		Lagrangian Relaxation	LP 완화법	ZOOM	Lagrangian Relaxation	LP 완화법	ZOOM
평균 CPU 시간 (Sample size-50)	4	3.246	0.83	6.098	3.282	0.945	5.392
	6	17.93	3.752	68.27	16.568	3.465	70.284

표 5.2는 Kusiak and Finke(1987)에 의해 발표된 LP 완화법과 이 연구논문에서 제시된 방법 그리고 ZOOM 프로그램을 통해 얻어진 결과들을 나타내고 있다. ZOOM에 의해 얻어진 결과와 비교해서 여기서 방법을 사용하여 얻어진 제시된 값이 훨씬 최적값에 가까운 값을 나타낸다는 것을 확인할 수 있다. 대칭일 때가 비대칭일 경우보다 더 좋은 결과를 도출하고 있다는 사실도 또한 확인할 수 있다. 우리의 실험에서 n의 크기를 8로 늘렸을 때, 제약식이 170, 변수의 수가 210, 정수 변수가 98로 늘어남으로 인하여 ZOOM에 의한 결과를 도출할 수 없어, 두 방법에 대한 차이를 확인할 수 없었다.

표 5.3은 각 방법에 의해 소비된 CPU 시간을 나타내고 있다. LP 완화법이 가장 짧은 시간으로 값을 주는 반면, ZOOM을 통해 계산된 시간은 LP 완화법에 소비된 시간보다 훨씬 많음을 알 수 있다. 마다의 수가 커짐에 따라 ZOOM의 사용이 불가능함으로, 보다 좋은 값을 얻기 위해 여기서 제시한 라그랑지안 방법의 사용이 필요하다.

6. 결론

우리는 TSP의 2-상품 네트워크 흐름문제를 풀기 위한 기본적인 과정을 설명하였다. 여기서 제시된 2-상품 네트워크 흐름 문제는 농업의 저장창고, 혹은 양조장이나 여러 농작물 수송문제에 적용될 수 있다. 제시된 방법의 적적할 응용은 비용의 절감과 노동력의 감축, 기계 수명의 연장 등 여러 형태로 나타날 수 있다. 현재 정확한 알고리듬이 제시되지 않은 상태에서 마디(node)수가 큰 문제에서는 라그랑지안 방법의 사용은 보다 좋은 해를 제시해 줄 것이다. 신경망(neural network)의 사용 또한 이러한 문제를 해결하는데 또하나의 대안이 될 수 있을 것이다.

參考文獻

1. M. S. Daskin (1995), Network and Discrete Location, Wiley, New Jersey
2. G. Finke, A. Claus and E. Gunn (9184), A Two-Commodity Network Flow Approach
to The Traveling Salesman Problem, Congress Numerantium 41, PP167-178
3. M. L. Fisher (1981), The Lagrangian Relaxation Method for Solving Integer
Programming Problems, Management Science, 27, PP1-18
4. A. Kusiac and G. Finke (1987), Modeling and solving the flexible forgoing module
scheduling problem, Engineering optimizaion, 12, PP1-12.
5. R. Marsten (1994), ZOOM User's Manual, Dept. of Management Information Systems,
University of Arizona, Tucson, AZ
6. K. G. Murty (1992), Network Programming, Prentice Hall, New Jersey
7. R. L. Rardin (1998), Optimization in Operations Research, Prentice Hall, New Jersey

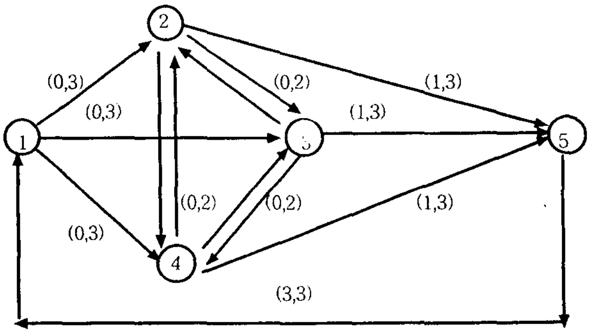

그림1. P 상품에 대한 네트워크 모델

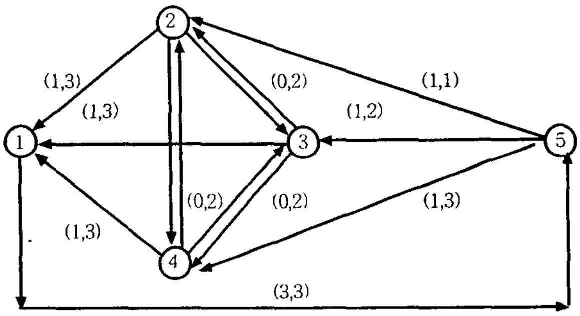

그림2. Q 상품에 대한 네트워크 모델

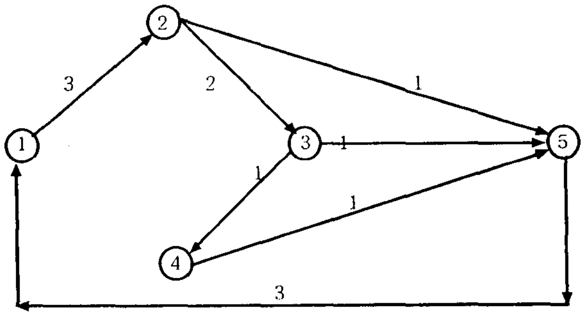
그림3. P 상품에 대한 최적 네트워크 흐름

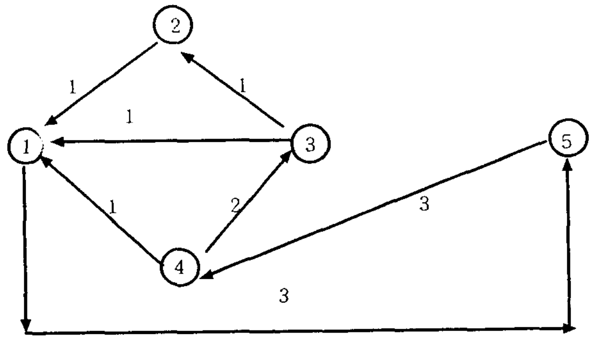
그림4. Q 상품에 대한 최적 네트워크 흐름

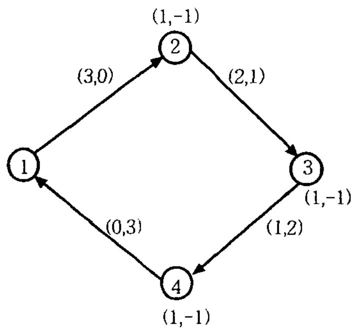

그림5. 가능한 2-상품 (x^p, x^q)의 흐름