물류센터 입지 선정 문제를 위한 Network Flow Algorithm 연구

박율기*, 이홍철**
*고려대학교 정보경영공학과
**정보네트워크연구실
e-mail:xereness@korea.ac.kr

A Study of Network Flow Algorithm for location problem of distribution center

Youl-Kee Park*, Hong-Chul Lee**
*Graduate School of Information Management & Security, Korea University
**Information network Lab.

요 약

본 연구는 생산 공장, 거점창고, 지역창고, 대리점으로 이어지는 일련의 SCM상에서의 물류센터의 최적의 입지를 선정 문제에 Network Flow Algorithm을 적용하고자 한다. 경로최소화, 전체비용최소화, 손실함수 등의 목적에 따라 문제가 다양하게 있을 수 있다. 최적화 문제에서 대표적으로 사용되는 Maximut flow의 알고리즘을 자세히 알아본다.

1. 서론

최근의 국가 경쟁력 향상의 주요한 분야인 물류유통 시스템의 최적화(유통량, 시간, 비용 등을 고려한 유통 시스템)는 매우 중요한 분야이며 많은 연구가 추진되고 있다. 기업 활동에 있어서 유통 문제는 무시할 수 없는 중요한 부분이며, 최근 물류유통 비용의 상승에 의한 업계의 부담이 가중되고 있는 기업 현실에서 운송 효율화를 실현하고 대응책을 모색하는 것은 업계와 정부의 관심이 집중되고 있는 분야이다.

제품의 운송, 보관, 재고통제 등의 물류분야는 관리의 혁신을 통하여 대폭적인 비용절감을 기대할 수 있으나 현재 관심부족으로 다른 분야에 비하여 연구가 안 된 분야이다. 이러한 물류 유통의 중요성은 다음과 같이 설명할 수 있다. 유통 효율의 향상으로 기업 물류비를 절감하여, 이를 통해 물가의 상승을 억제할 수 있다. 또한 영리를 추구하는 기업 입장에서의 물류를 최소의 비용으로 유통시킴으로써 매출 신장을 꾀하는 역할을 가능하게 한다. 이처럼 기업의 산업 활동에서 중요한 부분을 차지하고 있는 물류 시스템을 보다 효율적으로 운용하기 위해 Newtwork Flow 알고리즘을 적용하고, 이로부터 한정된 유통 경로 상에서 보다 많은 물류의 배송을 가능하게 하는 물류 최적화 시스템을 구축한다.

본 연구에서는 유통 네트워크에 Maximum Flow Algorithm을 적용하여 물류 유통을 극대화하고, 물류 시스템의 효율성을 향상시켜 물류비용 절감을 위해 알고리즘에 대해 연구를 한다.

본 논문의 주요 목차는 다음과 같다. 2장에서는 관련연구를 살펴보고, 3장에서는 알고리즘 설명을 하였으며, 4장 추후과제로 구성되었다.

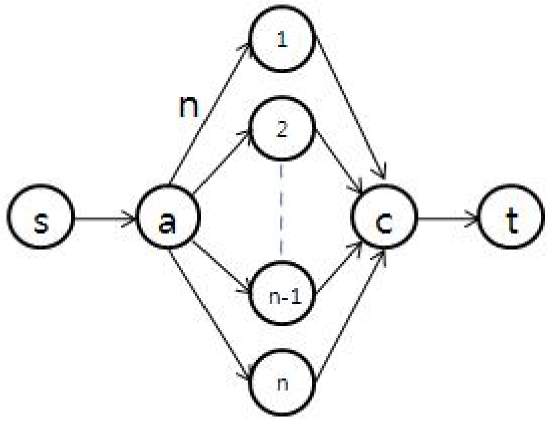
2. 관련연구

최대 흐름(MAX-Flow)문제는 단일-소스, 단일-싱크 흐름을 네트워크를 통해 최대화하여 나타낸다. 최대 흐름을 찾는 것은 흐름을 나타냄으로써 정의된다. 최대 흐름 문제는 순환문제와 같은 복잡한 네트워크 흐름 문제의 특별한 경우이다. 네트워크에서 최대 s-t (source to sink) 흐름은 최대 흐름 최소 잘라내기 이론에서 최소 s-t 문제와 같다. 주어진 유통 네트워크 $G=(V, E)$, 각각의 간선$(u, v)\in E$ 의 음이 아닌 용량 $c(u, v)>0$을 갖는 방향 그래프이다.

최대 흐름 문제를 푸는 방법에는 여러 가지가 있다. 첫 번째는 기본적인 방법으로 Menger's Theorem, Whitney's Theorem, Maximum flow / Minimum Cut, Ford Fulkerson, Edmonds-Karp, Maximum Flow by Scaling이 있다. 두 번째로, Blocking Flow를 이용한 알고리즘이 있다. 세 번째로, Preflow를 이용한 알고리즘, 네 번째로, Distance Labeling Function을 이용한 Shortest Augmenting Path Algorithm 등 다양한 방법이 존재한다.

3. Algorithm

Maximum Flow의 Network의 기본 구성은 아래와 같다.



[그림 1] 기본적인 네트워크 구조

Flow Networks G=(V,E)

- 각 edge (u,v)가 capacity c(u,v)≥0를 갖는 directed graph.
- source (s), sink (t)
- 모든 vertex v ∈ V 에 대하여,
  path s → v → t 존재 (connected)

$$|E| \geq |V| - 1$$

Flow f(u,v) : edge (u,v) 를 통과하는 양

(i) f(u,v) ≤ c(u,v), ∀u,v ∈ V

(capacity constraint)

(ii) f(u,v) = -f(v,u), ∀u,v ∈ V

(skew symmetry)

(iii) $\sum_{v \in V} f(u,v) = 0$,∀u ∈ V- {s,t}

(flow conservation)

value of flow: source 에서 방출되는 flow 의 합

$$|f| = \sum_{v \in V} f(s,v)$$

3.1 Ford Fulkerson

잔여 네트워크에 Augmenting Path가 존재하지 않을 때까지 계속 유량을 흘려보내는 알고리즘이다.

1. 잔여 네트워크의 Augmenting Path를 찾는다. 없을 경우에는 알고리즘을 종료한다.
2. AugmentingPath가 있는 경우, 지나온 Vertex들을 <s , v1 , … , vp , t>라 하자. L을 Path 내에서 거치는 잔여 네트워크의 간선들의 최소값이라 할 때, 그 Path 순서대로 L만큼의 유량을 source에서 target으로 흘려보낸다.
3. 다시 1로 돌아가서 알고리즘을 반복한다.

이 방법은 일단 Augmenting Path를 찾는데 약 O(E)가 걸리고, |f*|만큼의 최대유량이 존재하면 알고리즘 안에서 찾게 되는 Augmenting Path의 개수도 |f*|이 될 가능성이 있기에 시간복잡도는 위와 같이 된다.

3.2 Edmonds-Karp

Ford-Fulkerson을 개선한 것으로 residual network에서 확대 경로 p를 계산하기 위하여, 너비 우선 검색(breadth-first search)을 이용한다. 즉 확장 경로가 잔여 네트워크에서 s에서 t로의 가장 짧은 경로라면 가장 짧은 시간에 최대흐름을 나타낸다. 이 알고리즘의 실행시간은 $O(VE^2)$이 된다.

3.3 Goldberg's Generic Algorithm

Preflow를 이용한 알고리즘으로 f(u,v)에 대해서 플로우 보존이 성립하지 않고, 완화된 플로우 보존을 유지한다. 이 때 f를 선형 플로우라 한다.

즉, 일단 f(u,v)=-f(v,u)이며, $u \in V - \{s\}$에 대하여 $f(V,u) \geq 0$을 만족한다. 일단 몇개의 함수를 정의한다. 함수들은 다음과 같다.

- f(u,v) : Flow from u to v
- Cf(u,v) : c(u,v) - f(u,v) : 가능 용량
- excess;e(u) :f(V,u). $u \in V - \{s\}$: e(u)>0인 u를 오버플로우 되었다 혹은 active 상태에 있다고 한다.
- h(u) : distance labeling function (혹은 height function). 임의의 정점에 대해서 0 이상이다.

height function $h(u)$는, Cf(u,v)로 구성된 E_f의 간선 (u,v)에 대해서 $h(u) \leq h(v)+1$을 만족하며,

$h(s)=n$, $h(t)=0$을 만족하는 function이다. 따라서 만약에 $h(u)>h(v)+1$를 만족한다면 (u,v)는 E_f의 간선이 아니게 된다.
-허용 가능한 간선 : $(u,v) \in E_f$이고 $C_f(u,v)>0$인 경우 (u,v)는 허용 가능한 간선(admissible edge)라 한다.

4. 실험

Ford Fulkerson을 개선한 Edmonds-Karp(EK)와 Goldberg's Generic Algorithm(KAR) 2가지 방법의 성능을 비교했다.

[표 1] Sparse network : $E=\Theta(V)$

V	EK $O(VE^2)$	KAR $O(V^3)$
500	0.0017	0.248
1000	0.0047	4.30
2000	0.0094	38.2
4000	0.0193	-

[표 2] Medium network : $E=\Theta(V^{3/2})$

V	EK $O(VE^2)$	KAR $O(V^3)$
500	0.0099	0.939
1000	0.0565	9.34
2000	0.280	84.1
4000	0.780	-

[표 2] Medium network : $E=\Theta(V^2)$

V	EK $O(VE^2)$	KAR $O(V^3)$
500	0.282	3.64
1000	2.42	29.5
2000	16.5	190.0
4000	193.9	-

5. 결론 및 추후 연구과제

간단한 실험을 위해 그림 1에서 묘사한 네트워크를 실험에 사용하였다. 실제 유통 네트워크의 구조는 훨씬 복잡하기에 좀 더 복잡한 구성의 네트워크에 대해 실험이 필요하다. 인접 네트워크가 많고 복잡한 매트릭스 구조의 실제 유통구조에서 정교한 실험을 위해선 preflow-push algorithms이 필요하다. 그리고 위의 실험은 Maximum flow 문제에서 일반적으로 사용되는 Forward 방식으로만 진행되었다. backward 방식과 blanced 방식의 실험 비교가 필요하다.

참고문헌

[1] 물류 유통을 최적화하기 위한 네트워크-유통 알고리즘, 이충세, 김명하, 정보보안 논문지 제8권 제3호(2008.9)

[2] Introduction to algorithms Second edition , MIT Press , Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein

[3] Combinatorial Optimization , Theory and Algorithms Third edition, Springer , Bernhard Korte, Jens Vygen

[4] Determining the maximal flow in a network by the method of pre-flows, Karzanow, A.V, Sov. Math. Dokl. 15, 434 - 437 (1974)

[5] Efficient graph algorithms for sequential and parallel computers, Goldberg, Department of Electrical Engineering and Computer Science, MIT (1987)

[6] Network Flows : Theory, Algorithms, and Applications , Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin.

[7] Quick Max-flow Algorithm, Przemyslaw Gordinowicz, Journal of Mathematical Modelling and Algorithms : JMMA, v.8 no.1 ,2009, pp.19-34