그래프 감소를 위한 auction 알고리즘에 관한 연구

김현기*, 하기종**, 우경환***, 류기한****, 이천희*****
극동정보대학*, 영동전문대**, 우송공업대학***, 충주산업대****, 청주대학교*****
충북 청주시 상당구 내덕동 산 36.
yicheon@alpha94.chongju.ac.kr

A study on auction algorithms for reduced graph

Kim Hyungi*, Hah Gijong**, Woo Kyonghwan***, Ryu Gihan****, Yi Cheonhee*****
Keukdong College*, Youngdong College**, Woosong Technical College***,
Chungju Industrial Univ.**** Cheongju Univ.*****

Abstract

In this paper we consider strongly polynomial variations of the auction algorithm for the single origin/all destinations shortest path problem. These variations are based on the idea of graph reduction, that is, deleting unnecessary arcs of the graph by using certain bounds naturally obtained in the course of the algorithm. We study the structure of the reduced graph and we exploit this structure to obtain algorithm with $O(n \ \min\{m, n\log n\})$ and $O(n^2)$ running time.

1. 서론

본 논문에서는 Bertsekas[1, 2]에 의해 제안된 최단 경로 문제를 위해 auction 알고리즘에 초점을 맞추었다. 이 알고리즘은 Bertsekas가[3] 처음 도입하였고 할당 문제를 위해 원시적인 auction 알고리즘에 대한 문제는[4, 5] 다른 네트워크에서 auction 알고리즘에 밀접하게 관계된다. 또한 이 관계를 해석한 단일 원점과 단일 목적지인 경우에[2] 이 알고리즘은 매우 간단하므로 원점에서 출발하는 단일 패스를 유지하게 된다. 각 반복에서 패스는 새로운 노드를 추가하기 위해 확장하거나 그것의 터미널 노드를 빼기 위해 감소하며 목적지의 패스가 터미널 노드가 될 때 알고리즘은 종료하게 된다.

Auction 알고리즘의 강력한 다항식 버전은 감소 동작에 확장과 삼축 동작을 추가한 Pallottino와Scutella 에[6] 의해 얻어진다. 여기서 노드 1의 각 시간은 첫 번째 시간의 패스 터미널 노드가 되며 패스 한 개를 제외한 나머지에 들어오는 모든 호가 삭제된다. 패스가 i에 대해 최단일 때 이들 호는 삭제된다. 그러므로 auction 알고리즘은 $O(m^2)$의 수행 시간을 가진다. 여기서 m은 호의 수이다. 비감소 길이로 하기 위해 각 노드의 수출 호를 정리하는 개념을 사용하면 수행 시간은 $O(mn)$으로 더 감소하게 되며 여기서 n은 노드의 수를 나타낸다. 따라서 본 논문에서는 그래프 감소의 처리를 위해 확장 트리의 그래프 구조를 반들이 알고리즘의 이론과 실제 성능을 향상시키기 위한 방법을 제안하였다.

2. 최단 경로를 위한 auction 알고리즘

노드 i를 P의 터미널 노드로 할 때 식 (1)라면 스텝 1로 가라; 그렇지 않으면 스텝 2로 가라.

$$p_i < \min_{(i,j)\in A} \{ a_{ij} + p_j \} \tag{1}$$

스텝 1 : (감축) 세트

$$p_i := \min_{(i,j)\in A} \{ a_{ij} + p_j \} \tag{2}$$

i≠1이라면 P는 감축된다. 다음 반복으로 가라.
스텝 2 : (확장) 노드 j에의해 P는 확장된다.

$$j_i = \arg \min_{(i,j)\in A} \{ a_{ij} + p_j \} \tag{3}$$

j_i가 목적지 t라면 멈춘다: P는 최단 경로를 요구하게 되며 그렇지 않으면 다음 반복으로 가라.

1에서 t까지 적어도 한 개의 패스가 존재하는지를 알수있다. 여기서 t는 반복의 유한수에서 패스 P의 터미널 노드가 될 것이다. 패스 P는 1에서 t까지의 최단

경로가 된다. 호길이 a_{ij}는 상수라고 가정하면 반복의 수가 n^2L 에의해 제한됨을 알 수 있다. 여기서 $L = \max_{(i,j)\in A}|a_{ij}|$은 호 길이 범위를 나타낸다. 그러므로 불규칙하게 발생되는 문제에서 평균수행시간은 L에 의존하지 않는다. 알고리즘의 가장 나쁜 경우의 작용의 예는 그림 1에서 보여준다.

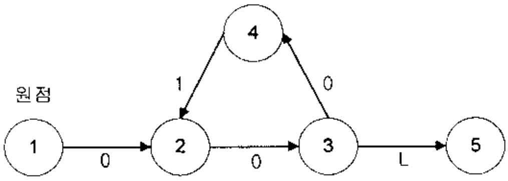
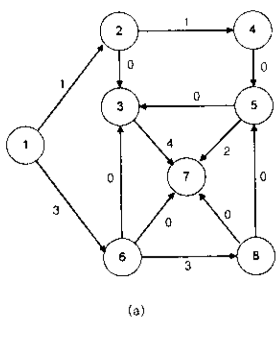


그림 1 : Auction 알고리즘의 모조 다항식의 작용

단일 목적 알고리즘은 모든 목적지에서 최단 경로 트리를 찾는데 사용된다. 사실상 남아있는 페어(P,p)가 변하지 않는 동안에 주어진 목적지에서 최단 경로를 찾는 새로운 목적지로 전환할 수 있다. 등가적으로 최단 경로 트리를 찾는 단계에서 모든 목적지가 P의 터미널 노드가 될 때까지 반복을 지속한다.

3. 그래프 감소를 가진 auction 알고리즘

그래프 감소를 가진 auction 알고리즘의 대표적인 반복은 기본적인 감축과 확장 단계가 그래프 감소 단계에서 진행되는 것을 제외하고 auction 알고리즘에서는 같이 본질적으로 같게 된다. 그래프 감소 단계는 호 또는 노드를 지울 수 있는 두 가지 방법이 있다.
(a) 현재 터미널 노드가 밖으로 나가는 호를 가지고 있지 않을 때 노드는 지워지고 반복은 끝나게 된다.
(b) 현재 터미널 노드 i가 약간의 밖으로 나가는 호를 가지고 있을때 i가 트리 노드일 때 첫 번째 반복이다. 이 경우에 i의 모든 안으로 들어오는 호는 P에 속한 트리 호를 제외하고 삭제한다. 그리고 약간의 경계선 상의 호는 만들어지게 된다. 그리고 삭제에 의해서 약간의 추가로 호가 발생되는 원인이 된다.
i를 P의 터미널 노드가 되게 하기 위해 i가 밖으로 나가는 호가 없고 i=1이라면 멈추고 그렇지 않으면 스텝 1로 가라.



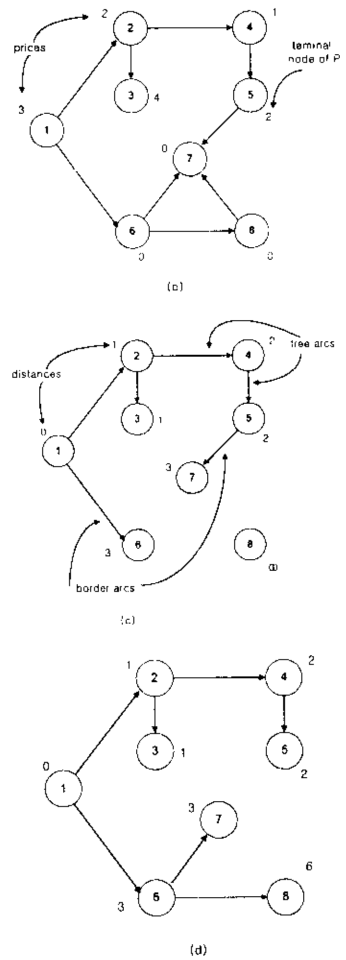


그림 2: 최단 경로 트리와 확장 트리

스텝 1 : (그래프 감소) i에 밖으로 나가는 호가 없다면 스텝 1(a)로 가고 그렇지 않으면 스텝 2나 i가 어떤 초기 반복에서 P의 터미널 노드가 되거나 아닌 경우에 스텝 1(b)로 가라.

스텝 1(a) : (터미널 노드의 삭제) 감축 P, i와 i에 들어오는 P의 호를 삭제하고 다음 반복으로 가라.

스텝 1(b) : (터미널 노드의 첫 번째 스캔) i에 들어오는 P의 호를(만약 i≠1이라면) 제외한 i의 모든 수입 호를 삭제한다. 각 수출 호 $(i,j)\in A$,에 대해서 식 (4)이라면 (i,j)를 삭제하고 그렇지 않으면 k는 i와 달리 트리 노드가 되는 호 $(k,j)\in A$,를 삭제한다. 그리고 식 (5)과 같이 세트한다.

$$u_i + a_{ij} \geq u_j \tag{4}$$

$$u_j := u_i + a_{ij} \tag{5}$$

i가 좌측의 수출 호가 없다면 감축 P는 i와 그것의 수입 호를 삭제하고 다음 반복으로 가라. 그렇지 않으면

스텝 2로 가라.
스텝 2 : (감축 또는 확장의 결정) 만약 식 (6)이라면 스텝 3으로 가라. 그렇지 않으면 스텝 4로 가라.

$$p_i < \min_{(i,j)\in A_r}\{a_{ij}+p_j\} \tag{6}$$

스텝 3 : (감축) 세트

$$p_i := \min_{(i,j)\in A_r}\{a_{ij}+p_j\}$$

그리고 i≠1이라면 P를 감축, 다음 반복으로 가라.
스텝 4 : (확장) j_i 노드에 의해 P를 확장

$$j_i := \arg\min_{(i,j)\in A_r}\{a_{ij}+p_j\}$$

트리 노드의 수가 n과 같다면 멈춰라: 트리 호의 세트는 최단 경로 트리로 정의한다. 그렇지 않으면 다음 반복으로 가라.

그림 2는 트리 T의 최단 경로와 트리 E가 확장된 알고리즘의 설명이다. 원점 그래프는 그림 2(a)에서 보여준다. 먼저 가격 p1은 1에서 결정된다. 노드 1은 트리 노드가 되고 노드 2와 6은 경계 노드가 된다. 노드 2가 확장되어 노드 3과 4은 경계 노드가 된다. 노드 3에서 확장되어 호(5,3)과 (6,3)은 삭제된다. 노드 7은 경계 노드가 된다. 나중에 원점의 범위 내에 있는 P의 계속적인 감축과 노드 5의 범위 내에 계속적인 확장은 호(8,5)와 (3,7)에서 그림 2(b)에서 보여진 것과 같이 삭제된다. 경계 호(3,7)의 삭제될 때까지 관여하는 한 그것은 노드 5를 통해 최단 경로가 되었기 때문에 u7의 향상에 기인한다. 트리 E의 확장은 그림 2(c)에서 보여준다. 더나아가 원점으로부터 계속되는 P의 감축에서 다음의 확장은 노드 6과 노드 7에서 이루어진다. 노드 8은 호 패스 트리인 동안 경계 노드일지라도 경계 노드와 같이 확장되어 트리 E에 들어간다. 아래 반복의 원점에서 모든 방법으로 P의 계속적인 감축이 수행된다. 노드 5의 계속적인 확장과 노드 1의 감축은 다른 순서에 의해 수행된다. 이러한 동작 동안에 호(2,4)와 (4,5)를 가진 노드 4와 5는 삭제된다. 노드 3에서 확장의 순서는 원점에서 감축의 순서에 의해 수행되고 노드 2, 3 그리고 호(1,2)와 (2,3)은 삭제 원인이 된다. 결국 P는 마지막 노드 8에서 확장되고 알고리즘의 끝은 최단 경로 트리 T에 들어간다.

알고리즘의 수행시간을 평가하기 위해서 그것들은 호(i,j)당 u_i+a_{ij}와 $a_{ij}+p_j$ 형태의 총합 계산의 하나를 포함하므로 먼저 첫 번째 스캔 반복이 총시간 $O(m)$과 $O(m)$ 호 삭제의 총계를 취한다. 다음에 확장 주기가 주기의 시작에서 존재하는 확장 트리 E의 호의 부분 집합의 시험을 포함하기 때문에 $O(n)$ 시간이 된다. 그리고 E가 제안 1(f)에 의한 트리이므로 많아야 (n-1)이다. 유사하게 감축 주기는 주기의 시작에 존재하는 확장 트리의 호에 대한 부분 집합의 시험을 포함하기 때문에 $O(n)$ 시간이 되므로 처음 스캔 반복 n보다도 적다. 그것들의 수는 $O(\min\{m, n\log n\})$이다. 따라서 m≥n을 가정하면 $O(n\min\{m, n\log n\})$ 수행 시간이 유도된다.

중요 특성은 그림 2의 설명과 같이 노드의 두 개의 가능한 형태중의 하나로 유도되어 계속되는 확장의 순서에 이끌리는 호(i,j)에 의해 유도되는 확장 P와 반복이다.
(a) 트리 노드가 되는 k 경우에 경계 노드 k와 첫 번째 스캔 반복이 된다.
(b) 감축의 경우에 트리 노드 k는 노드 k가 삭제가 되는지 p_k가 증가된다.

k에 유도되는 확장 주기가 (a)의 경우에 성공적이라고 하고 (b)의 경우에 실패되었다고 한다. 계속되는 확장 주기의 수는 n보다 적다. 실패된 확장 주기의 수는 $O(\min\{n, n\log n\})$이 된 부록에서 보여준다. 그러므로 확장 주기의 총계는 $O(\min\{n, n\log n\})$이다. 마지막으로 감축 주기가 첫 번째 스캔 반복이거나 실패된 확장 주기에 의해 진행되므로 감축 주기의 수는 $O(\min\{n, n\log n\})$보다 적다.

4. 개선된 auction 알고리즘

터미널 노드 i에서 다음의 감축과 확장은 $(p_i = a_{ij} + p_j)$이므로 호(i,j)는 한 개 이상이 된다. 여기서 임의로 한 개를 선택하여 i에서 확장이나 감축이 발생할 때 다음 반복까지를 i의 예비 호라고 부른다. (i,j)가 i의 예비 호이고 확장은 i에서 취해진다면 (i,j)가 i의 예비 호가 되기 위해 지속되는 동안에 노드 j는 P의 터미널 노드가 된다. 현재 패스 P의 보충된 모든 호는 예비 호이다. 정의에 따르면 향상된 노드는 예비 호가 되므로 노드의 예비 호가 되는 동안 호를 삭제 할 수 있는 것이 가능하고 확장 주기가 실패되는 원인이 된다.

다중 패스로 불리고 M으로 표시되는 그래프를 도입하여 알고리즘에서 중요하게 사용하게 된다. M의 노드는 현재 감소된 그래프인 트리 노드와 경계 노드이다. 즉 여전히 삭제되지 않는 트리와 경계 노드가 있다. 호 M은 감소된 호에 속해 있고 $(p_i = a_{ij} + p_j)$를 만족하는 예비 호(i,j)이다.

이 목적을 위해 중요한 두 정의를 한다.
(a) 트리 노드 i와 종단 노드(i)가 확장된 패스 P가 경계 노드일때 패스 $P_M(i)$에서 경계 노드 last(i)까지 확장 순서는 P에 의해 도달되고 첫번째 반복 스캔은 각 노드에서 발생된다.
(b) M이 첫 번째 스캔 반복이 아닌 어떤 반복의 시작에서 완전하다면 M은 반복의 끝에서 완전하게 된다. 이 가정을 증명하기 위해 주어진 반복의 터미널 노드 i는 트리 노드이고 반복의 끝에서 보여준다. 모든 노드 j∉P의 패스 $P_M(j)$의 마지막 노드가 경계 노드이다. 실제로 i에 대해 제외된 모든 노드 j∉P의 패스 $P_M(j)$은 반복에 의해 영향을 미치지 않기 때문에 이러한 특성을 가지게 되며 터미널 노드 i가 P에서 나갈 때 감축의 경우를 관찰하기 위해 남는다. i는 반복 동안에 예

비 호(i,j)를 얻는다. 호(i,j)로 이루어진 반복의 끝에서 패스 $P_M(i)$는 $P_M(j)$에 의해 이루어진다. 여기서 M은 반복의 시작에서 완전한 경계 노드가 끝이 되야 한다. M의 완전함이 첫 번째 스캔 반복보다 다른 반복에 의해 보존되어 완전하다.

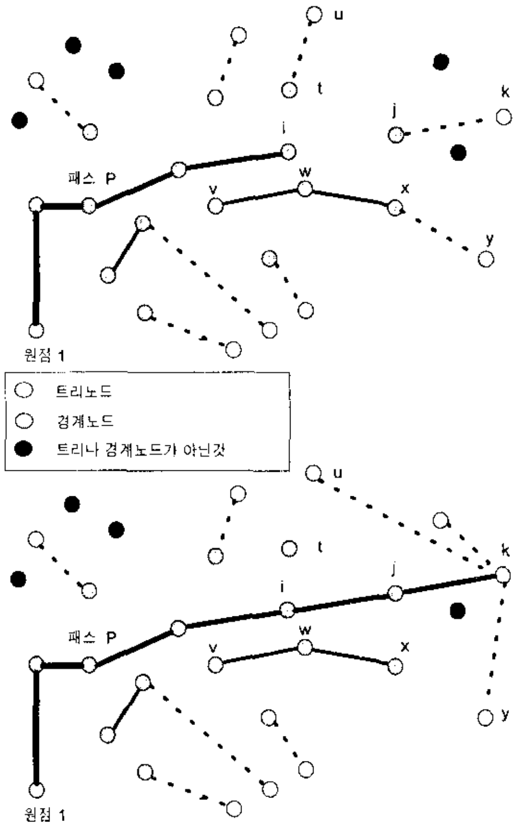


그림 3 : 노드 (i,j)에 대한 다중 패스

M이 모든 시간에서 완전하다고 보증된다면 실패한 확장 주기가 없을 것이라고 위의 관찰 (a)에서 보여준다. 각 첫 번째 스캔 반복은 감축 주기에 의해 따르게 되고 성공한 확장 주기에 의해 차례로 놓여지게 된다. 앞 절의 auction 알고리즘에서 다중 패스는 완전할 필요가 없고 관찰(b)에서 어떤 예비 호가 삭제될 때 첫 번째 스캔 반복에 기인한 것을 보여준다. 완전함을 유지하든지 그렇지 안으면 트리 노드 i에 대한 확장이 발생되는 것을 보장하든지 다중 패스를 개조해야하는 개념을 가지게 된다. 종단 노드(i)는 경계 노드이므로 첫 번째 스캔 반복 당 $O(n)$시간에서 행해진다면 재구성된 다중 패스를 위해 요구되는 총시간은 $O(n^2)$가 된다. 또한 알고리즘의 수행 시간은 앞 절의 해석에 의해 $O(n^2)$가 된다.

5. 결론

본 논문에서는 더 효과적으로 호를 삭제하기 위해 최단 거리의 노드에 상측 한계를 사용해 그래프 감소 개념을 적용한다. 이렇게 얻어진 감소 그래프의 구조를 연구하였다. 향상된 시간 복잡도를 지닌 알고리즘을 얻기 위해 이 구조를 개발했다. 특히 $O(n\min\{m, n\log n\})$의 수행 시간을 가진 알고리즘을 개발했다. 그리고 다른 하나는 좀 더 복잡하지만 수행시간은 $O(n^2)$이 되었다.

참 고 문 헌

1. Bertsekas, D. P., "A The Auction Algorithm for Shortest Paths," SIAM H. on Optimization, Vol. 1, 1991, pp. 425-447

2. Bertsekas, D. P., "Linear Network Optimization: Algorithm and Codes," M.I.T. Press, Cambridge, MA, 1991,

3. Bertsekas, D. P., "A Distributed Algorithm for the Assignment Problem," Lab. for Information and Decision Systems Working Paper, M.I.T., Cambrige, MA, March 1979.

4. Bertsekas, D. P., "A New Algorithm for the Assignment Problem" Math. Programming, Vol. 21, 1981, pp. 152-171.

5. Bertsekas, D. P., "The Auction Algorithm for Assignment and Other Network Flow Problems:A Tutorial," Interface, Vol. 20, 1990, pp. 133-149.

6. Pallottino, S., and Scutella', M. G., "Strongly Polynomial Algorithms for Shortest Paths," Dipartimento di Information Report TR-19/91, University of Pisa, Italy, 1991; also in Ricerca Operative, Vol. 60, 1991, pp. 33-53.