병렬구조를 가진 전산기에 의한 최단경로 계산

Shortest Path Calculation Using Parallel Computer System

서 창 진* 서울대학교 공과대학원 제어계측공학과 석사과정

이 장 규 서울대학교 제어계측 공학과 조교수

1. 서론

회로망은 점과 같이 위치만을 표시하는 정점(vertex)과 이 정점들을 이어주고 있는 가지(edge)로 이루어진다. 만일 두 정점 사이에 직접 연결되어 있는 가지가 존재치 않을 때는, 중간에 여러 정점을 거치면서 연결되어 있을 것이다. 이와같이 연속적으로 연결된 여러개의 가지를 우리는 경로(path)라 한다. 최단경로 계산을 할 경우에는 각각의 가지들에 실수의 가지값(weight)이 주어지게 된다. 이때 한 정점에서 다른 정점까지의 가능한 모든 경로중 각각의 가지값을 합한 경로값이 가장 적어지는 경로를 최단경로라 하며 최단경로 계산이란 모든 정점에서 모든 정점까지의 최단경로값을 알려주는 비용행렬(cost matrix)과 최단경로를 나타내는 경로행렬(successor matrix)을 구하는 문제로서, 운송시스템, 정보통신회로망, 초대형집적회로 디자인등에 이용되고 있다.

이를 구하는 방법으로서는 Dantzig법 [1], Floyd법 [2], Rosenthal법 [3]등 여러가지가 있는데 정점이 많아질수록 계산의 회수가 제곱이상으로 늘어나는 까닭에 회로망이 매우 큰 경우에는 이러한 방법들은 사용이 곤란하게 된다. 이 문제를 해결하기 위해서는 전체회로망을 몇개의 회로망으로 나누어서 각각의 소회로망의 비용행렬과 경로행렬을 구한 후 이 값들로부터 전체회로망의 비용행렬과 경로행렬을 구하여야 하며 이것을 분할방법이라 부른다. 분할방법에 의하여 최단경로를 계산하는 알고리즘이 최근 많이 개발되어 그 효율성이 인정되었다 [4]. 분할방법의 잇점은 회로망을 작게 나누어 필요한 계산을 함으로 적은 용량을 가진 콤퓨타로도 최단경로계산을 수행할 수 있고 계산 수행시간이 회로망 크기에 비례하므로 시간을 단축시킬 수 있다는 것이다. 이러한 잇점은 병렬구조를 가진 전산기를 사용하여 여러개의 소회로망을 동시에 처리하는 경우 훨씬 증대하게 된다. 이 논문에서는 기존의 분할방법을 발전시켜 병렬구조에 적합한 새로운 알고리즘 개발과 병렬구조 전산기를 구성하여 최단경로를 계산한 결과를 보고하고저 한다. 다음절에서 알고리즘을 소개하고 3절에서 분할방법에 의한 최단경로 계산을 MZ-80 8 bit 마이크로 전산기에 2대의 Z-80 CRC kit를 연결한 병렬구조 시스템에 의하여 수행한 예를 설명한다.

2. 알고리즘

알고리즘을 소개하기 전에 논리의 전개를 위해 우선 몇개의 술어를 정의한다.

(정의 1) 경계 정점 및 경계정점집합(boundary vertex set): 경계정점집합의 모든 원소와 이것과 직접 연결된 가지를 전체 회로망에서 제거할 경우 원하는 수 만큼의 소회로망으로 분리하게 되는 정점 및 그들의 집합

경계정점집합은 회로를 자르는 방법에 따라 집합이 다르게 이루어진다.

(정의 2) 절단집합(cutset): 경계 정점집합과 이것과 직접 연결된 가지를 합친 집합

(정의 3) 소회로망(subnetwork): 전체 회로망에서 절단집합을 제거함으로써 생긴 나누어진 몇개의 회로망중의 한 회로망

(정의 4) 소경계 정점집합 : 경계정점의 부분집합으로 다음의 (i)이나 (ii)의 성질중 하나를 만족하는 원소로 이루어진 집합중 제일 적은수의 원소로 소획로망 i를 나머지 획로망으로 부터 분리시키는 집합

(i) 소획로망 i의 임의의 한 정점으로부터 하나의 가지에 의해 연결된 경계정점

(ii) 경계정점으로 부터 하나의 가지에 의해 연결된 경계정점

(정의 5) 소절단집합i : 소경계집합과 이것과 직접 연결된 가지를 합친 집합

이와같이 정의하면 한 획로망에서 다른 획로망으로 가는 경로는 한개이상의 경계정점을 거쳐야 한다. 그림1은 4개의 소획로망으로 나누어진 획로망을 보여주고 있으며 그림에서 검은 점들은 경계정점을 나타낸다. 그림1에서 {1,2,3} 을 소획로망 1이라 하면 4,5,7이 정의4 (i)에 의하여 6이 (ii)의 성질을 만족하여 {4,5,6,7}이 소경계정점집합1을 만든다.

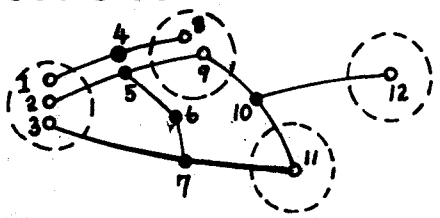


그림1. 4개로 분할된 획로망

위에 정의된 술어들을 사용하여 알고리즘을 써보면:

(단계 0) 획로를 자르고 경계정점을 정하고 이를 소경계정점으로 나눈후 행렬에 초기치를 준다.

(단계 1) ㄱ. 소획로망i와 소절단집합 i를 합쳐 구한 경로행렬과 비용행렬을 i=1,...,n의 경우 행한다.

ㄴ. 경계정점끼리의 수치들을 경계경로행렬에 자리에 맞게 집어넣는다. 단 기존의 숫자 a_{ij}가 집어 넣으려는 숫자보다 적을 경우는 그 값을 버리고 그다음 a_{ij+1} 혹은 $a_{i+1,1}$를 행하고 클 경우는 비용행렬을 집어넣는다. (ㄴ은반복적으로 i=1,...,n의 경우 행한다.)

ㄷ. 만들어진 경계경로행렬과 경계비용행렬로부터 기존의 최단경로 계산을 행하여 경계점끼리의 최단경로와 최소비용을 얻는다.

(단계 2) 소획로망i와 절단집합과 합한 획로망에서 비용행렬과 경로행렬을 구한다. 여기에서 소획로망내의 최단경로와 최소비용을 모두 구할 수 있다.

(단계 3) 서로 다른 소획로망에 속한 두 정점 사이의 비용행렬과 경로행렬을 구한다. 이것은 $\min_{1\le k\le m(\ell)} (a_{ik}+a_{kj})$을 구함으로써 얻게 된다. 이때 i은 i가 속해있는 소획로망의 번호이고 $m(\ell)$은 ℓ의 소경계정점의 수이다.

이 알고리즘이 기존의 분할방법[4]와 다른점이 단계1과 단계3에 나타나 있으므로 그 단계에 대하여만 설명하기로 한다. 단계 1-ㄴ은 경계정점만을 정점으로 인정하는 새로운 획로망을 생각하여 설명되어 질 수 있다. 즉 원래의 획로망을 그림2에서와 같이 경계정점들 사이에 일련의 소획로망의 정점들이 한 줄로 늘어서 있는 가지들에 의해 연결된 형태는 다르나 동일한 획로망으로 생각할 수 있다.

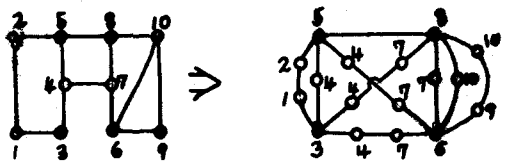


그림2. 경계정점사이의 최단경로 계산

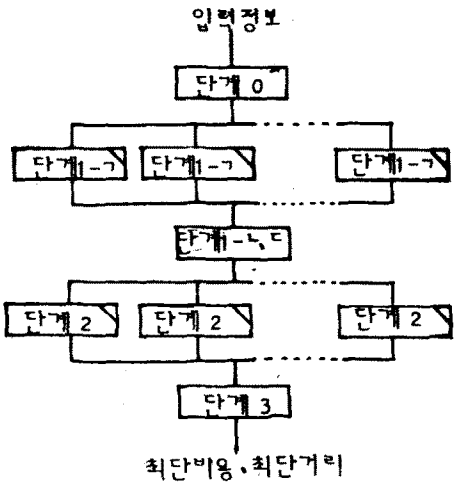
하나의 ij에 대하여 2개 이상의 a_{ij}가 나올 경우는 제일 적은 a_{ij}만이 필요하게 되는데 이것이 단계 2-ㄴ에 해당한다. 이 때 만들어진 획로망으로 최단계산을 하면 모든 경계정점 사이의 답이 나오게 된다.

한편 단계3에서는 모든 경계정점의 수만큼 비교할 필요없이 출발하는 정점의 소경계정점만을 비교하게 하였다. 이것은 중간의 여러 경계정점을 거쳐 목적 정점에 도달하는 경우에도 중간에 지나가는 경계정점들은 생각할 필요가 없이 처음 만나는 경계정점의 정보만으로 최단경로를 구할 수 있음을 뜻한다.

3. 병렬구조를 가진 전산기에 의한 분할방법의 풀이

2.의 알고리즘은 병렬적인 구조를 가지고 있다. 하지만 병렬적으로 수행되는 하나하나의 과정은

매우 복잡하며, 계산중간에 전입 혹은 전출되는 정보의 이동도 전혀 없기 때문에 멀티프로세서나 패럴렐프로세서를 사용하는 것은 부적당하며 독립적인 몇대의 전산기를 결합하여 사용하는 편이 효율적이다. 그리고 전산기의 연결방법은 한 대의 주전산기 (MZ-80)에 소회로망 수 만큼의 종전산기(연습용 키트인 CRC-800A)를 연결시키는 것이다. 동작방법은 주전산기에서 프로그램 및 데이타의 입출력을 담당하며 종전산기에는 수행할 프로그램을 입력하기 위한 부팅 프로그램만을 PROM에 입력시키며 이후 동작은 주전산기에서 전입된 프로그램에 의하여 수행된다. 그리고 주전산기와 종전산기 사이의 정보교환은 PIO를 이용한 벡터드 인터럽트(Vectored interrupt) 방식을 사용한다.



: 종전산기

: 주전산기

그림3. 병렬구조에 의한 최단경로 알고리즘 수행

동작 알고리즘을 살펴보면 각 소회로망에서의 최단거리계산을 하는 (단계 1-ㄱ)과 (단계 2)의 부분과 서로 다른 소회로망끼리의 계산인 (단계 3)의 부분이 병렬계산이 가능하므로 이 부분은 종전산기에서 처리하게 되며 입력 정보의 관리, 종전산기의 입출력 관할, 병렬 수행이 불가능한 부분의 처리 및 최종결과의 출력을 주전산기가 처리한다. 주전산기에서 종전산기로 보내는 방법은 소회로망 1,...,n을 종전산기 1,...,n에 하나씩 보내게 하며 받을때는 계산이 먼저 끝난 순서대로 정보를 받게 한다.

실제로는 2대의 종전산기를 달아서 IEEE 39 bus를 변형시킨 39개의 정점을 가진 회로망을 5개의 소경계정점에 의하여 2개의 소회로망으로 작르게 한 후 동작시켜 보았다. 이 경우에는 사용하는 언어가 다른 관계로(주전산기는 basic, 종전산기는 어쎔블러) 다른 방법들과 동작시간의 비교는 할 수가 없었다. 그러나 회로망이 커질수록 이러한 시도가 동작시간을 가장 짧게하는 시도라는 것은 자명한 일이다. 더군다나 전산기간의 데이타 전송도 그리 복잡하지 않아 설계또한 간단하게 처리될 수 있다.

4. 결론

이번 논문은 경계정점의 집합과 각 소회로망을 합친 회로망으로부터 최단경로계산을 소경계정점의 집합을 정의하여 소경계정점의 집합과 소회로망을 합한 회로망으로부터 계산을 함으로서 계산량을 줄이고, 동시에 좀 더 병렬적인 방법으로 바꾸었다. 그리고 이 바뀐 알고리즘을 이용하여 간단한 병렬구조를 가진 전산기 전산시스템을 구성하여 프로그램을 동작시켜 보았다. 그리하여 회로망이 매우 크고, 리얼타임으로 최단거리 계산을 행해야 될 경우에 성능이 좋은 고가의 전산기를 사용하는 대신, 값이 싼 몇대의 전산기를 이용하여 분할 방법으로 푸는 것이 가장 빠르고, 경제적인 방법임을 확신하게 되었다.

5. 참고문헌

1 G.B. Dantzig, "On the shortest route through a network," Management Sci., vol.6, pp. 187-190, 1960

2 R.W. Floyd, "Algorithm 97: Shortest path," Comm. Ass. Comput.,Mach, vol.5, pp.345, 1962

3 A. Rosenthal, "On finding shortest distances in a graph-An improvement to Dantzig's inductive algorithm," Discrete Math., vol.10, pp.159-162, 1974

4 Jang G. LEE "Calculation of the shortest paths by optimal decomposition", IEEE Trans on S.M.C. vol.12, No.3, pp.1410-415