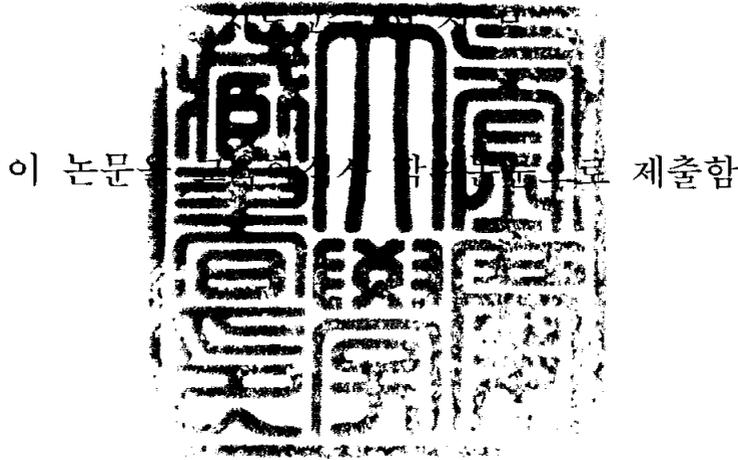


교육학석사 학위논문

EMFG의 시간트랜지션을 이용한
최단경로에 대한 연구



2004년 2월

부경대학교 교육대학원

전 산 교 육 전 공

김 남 혁

김남혁의 교육학석사 학위논문을 인준함

2003년 12월 13일

주 심 이학박사 박 흥 복 

위 원 공학박사 김 영 봉 

위 원 공학박사 여 정 모 

[차 례]

[표차례]	iii
[그림차례]	iv
[Abstract]	vi
1. 서론	1
2. 관련 연구	4
2. 1. 방향그래프	4
2. 2. EMFG	6
2. 2. 1 EMFG의 정의	6
2. 2. 2 EMFG의 동작	8
2. 2. 3 EMFG의 특징	9
3. 방향그래프를 EMFG로 변환	11
3. 1 EMFG의 마킹시간	11
3. 2 EMFG의 박스	12

3. 3	EMFG의 시간트랜지션 과 아크	12
3. 4	EMFG 시간트랜지션의 짐화시간	13
4.	최단경로 알고리즘	17
4. 1	EMFG 시간트랜지션을 이용한 최단경로 알고리즘	17
4. 2	계산복잡도 계산 및 비교·분석	22
5.	알고리즘 적용 예	26
5. 1	방향그래프를 EMFG로 변환 적용 예	26
5. 2	EMFG 시간트랜지션을 이용한 최단경로 적용 예	28
5. 3	마크벡터경로흐름에 따른 최단경로	37
6.	결론	39
	[참고문헌]	41

[표차례]

[표 1] 최단경로 문제의 계산복잡도	23
[표 2] 알고리즘 비교·분석	25
[표 3] 최단경로 시뮬레이션 진행과정1	32
[표 4] 최단경로 시뮬레이션 진행과정2	32
[표 5] 최단경로 시뮬레이션 진행과정3	33
[표 6] 최단경로 시뮬레이션 진행과정4	34
[표 7] 최단경로 시뮬레이션 진행과정5	35
[표 8] 최단경로 시뮬레이션 진행과정6	36
[표 9] EMFG의 시뮬레이션 전체 실행 결과	37

[그림차례]

[그림 1] 방향그래프의 예1	5
[그림 2] EMFG의 예1	6
[그림 3] 방향그래프 정점을 EMFG로 표현	12
[그림 4] 방향그래프 간선을 EMFG로 표현1	14
[그림 5] 방향그래프 간선을 EMFG로 표현2	16
[그림 6] 방향그래프의 예2	26
[그림 7] EMFG로 변환된 방향그래프	27
[그림 8] EMFG의 시뮬레이션 과정1	31
[그림 9] EMFG의 시뮬레이션 과정2	31
[그림 10] EMFG의 시뮬레이션 과정3	32
[그림 11] EMFG의 시뮬레이션 과정4	33
[그림 12] EMFG의 시뮬레이션 과정5	34
[그림 13] EMFG의 시뮬레이션 과정6	35
[그림 14] EMFG의 시뮬레이션 과정7	36

[그림 15] EMFG의 시뮬레이션 전체 실행 과정37

[그림 16] 마크백터 흐름에 따른 최단경로(예: A \Rightarrow G)37

A Study on Shorted Path by using EMFG's Time Transition

Major in Computer Science Education Kim Nam Hyuk
Directed by Professor Yeo Jeong Mo

Shortest path problem as, problem that application is possible in communication network or various field of application such as road network, Efficient algorithm have developed.

In this paper, Embodied process that solve shortest path from a particular vertex of directed graph to all different vertex using mathematical analysis method algorithm of EMFG.

To solve shortest path from a particular Vertex of directed graph to all other vertexes, First, EMFG that is presented method that changed directed graph to EMFG, and changes by presented algorithm solved problem using shortest path algorithm that apply mathematical analysis method.

Proposed algorithm is to grasp flowing of EMFG's mark vector and solves shortest path, Paper that design because does fetters in traditional computer model that existent methods present and does not analyze, and sees presented

solution method of new form that solve shortest path problem through process that EMFG's mark vector alters according to flowing of time. And use embodying program, is expected that confirmed through Simulation process that system acts rightly, and eases design of system, and helps a lot of on interpretation of system.

1. 서론

최단경로문제는 특정한 두 정점 사이의 경로 중에서 가장 짧은 길이의 경로를 찾는 문제로서 판매에서 물류의 수·배송망이나 네트워크 형태의 통신망, 고속 도로망 등의 다양한 응용분야에 적용되어 왔으며, 이를 표현하기 위한 방법으로는 방향 그래프가 사용되었다. 방향 그래프의 각 간선을 이용하여 노드들의 연결 상태를 표현하였으며, 연결된 거리나 통과하는데 걸리는 시간 등은 가중 값을 이용하여 표현하였다. 방향그래프의 특정한 두 정점 사이에는 여러 경로가 존재하고 그 중 최소 가중치 값을 가지는 경로를 최단경로라고 한다. 하나의 정점에서 다른 모든 정점까지의 최단경로 문제는, 방향그래프의 시작정점에서 나머지 모든 정점까지의 최단경로를 구하는 것으로서, 시작정점과 목표정점까지의 경로 중, 경로를 구성하는 간선들의 가중치 합이 최소가 되는 경로를 의미한다.

본 논문에서는 방향그래프의 특정한 한 정점에서 다른 모든 정점까지의 최단경로를 구하는 과정을 EMFG의 수학적 해석 방법 알고리즘[15]을 이용하여 구현하였다. 방향그래프의 한 정점에서 다른 모든 정점까지의 최단경로를 구하기 위해, 먼저 방향그래프를 확장된 마크흐류선도(EMFG: Extended Mark Flow Graph)로 변환하는 방법을 제시하였다. 제시된 알고리즘에 의해 변환된 EMFG를 EMFG의 시간트랜지션(Time

Transition)에 수학적 해석 방법[15]을 적용한 최단경로 알고리즘을 이용하여 문제를 해결하였다.

제안된 알고리즘은 EMFG의 마크벡터의 흐름을 파악하여 최단경로를 구하는 것으로서, 기존의 방법들이 고전적인 컴퓨터 모델에 기반 하여 설계·분석되어진 반면, 본 논문에서는 EMFG의 마크벡터가 시간의 흐름에 따라 변화하는 과정을 통해 최단경로 문제를 구하는 새로운 형태의 해결 방법을 제시하였다.

제안된 알고리즘을 이용하면 EMFG의 모든 동작 과정을 수학적 해석을 통한 박스(Box)의 마크벡터 변화과정을 통해 특정한 한 정점에서 다른 모든 정점까지의 최단경로를 보다 쉽고 정확하게 확인할 수 있으며, 특히 방향그래프 G 의 모든 정점의 개수가 n 이라 할 때, 간선의 수가 $n(n-1)/2$ 개인 완전방향그래프에서, 특정한 한 정점에서 다른 모든 정점으로 연결된 간선의 가중치가 동일하다면, 계산복잡도는 $O(n^2)$ 의 시간이 걸리게 된다. 이 과정은 구현되어진 프로그램을 이용한 시뮬레이션과정을 통해 시스템의 동작이 올바른 것을 확인하였다. 그리고, EMFG는 시스템을 설계하여 직접 실제회로로 변환할 수 있으므로 최단경로 알고리즘이 내포된 이산제어시스템의 모델링이나 설계가 용이하며, 나아가 설계된 시스템의 실험과 분석을 통하여 시스템의 성능향상에 도움을 줄 것이다.

2절에서는 방향그래프와 EMFG의 이론에 대한 내용을 설명한다. 3절에서는 방향그래프의 기준가중치 S_m , 정점 V , 간선 E ,

간선 E의 가중치 w 가 EMFG의 마킹시간 t_w , 박스 B, 시간트랜지션(Time Transition)과 아크(Arc), 시간트랜지션(Time Transition)의 점화시간으로 변환하는 알고리즘을 제안한다. 4절에서는 점속행렬을 이용한 EMFG의 수학적 해석[15]을 적용한 EMFG의 시간트랜지션(Time Transition)을 이용한 최단경로 알고리즘을 제안하고, 계산복잡도 계산과 비교·분석과정을 나타내었다. 5절에서는 방향그래프를 EMFG로 변환, EMFG의 시간트랜지션(Time Transition)을 이용한 최단경로 알고리즘, 마크벡터 경로흐름에 따른 최단경로 알고리즘을 적용한 예를 보여주며, 이는 구현되어진 프로그램을 통해 실험·분석되었다.

2. 관련연구

2.1 방향그래프

방향그래프 G 는 (V, E) 의 두 쌍으로 이루어지는데, V 는 정점(Vertex)의 모임이고, E 는 V 의 서로 다른 두 정점으로 이루어진 간선(Edge)의 모임으로 정점의 개수가 한정된 집합이 있을 때 이들 정점과 정점들 사이의 관계가 선형 구조가 아닌 비선형 구조로서 정점들 사이에 다양한 관계를 표현할 수가 있으며, 컴퓨터 통신망이나 도로망 등 복잡한 응용 문제를 효과적으로 표현할 수 있다.

방향 그래프 G 는 공집합이 아닌 정점의 유한 집합과 공집합도 허용하는 간선의 유한 집합으로 구성되고, 각 간선에 대한 가중치($w_{ij} \geq 0$)의 집합을 w 라고 할 때, 그 표현법은 다음과 같다.

$$G=(V, E) \quad \text{식 1}$$

V : 방향그래프 G 의 정점의 집합

E : 방향그래프 G 의 간선의 집합

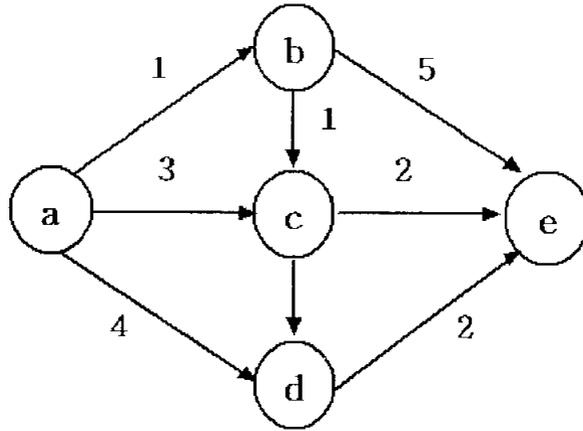


그림 1. 방향그래프의 예1

방향그래프 G 는 상태를 나타내는 노드들의 집합인 정점의 집합과 임의의 정점 v_1 에서 다른 정점 v_2 에 이르는 경로가 주어진 간선의 집합으로 구성되어 있으며, 그림 1은 방향그래프 $G = (V, E)$ 로서 정점의 집합 $V(G) = \{a, b, c, d, e\}$ 와 간선의 집합 $E(G) = \{ \langle a, b \rangle, \langle a, c \rangle, \langle a, d \rangle, \langle b, c \rangle, \langle b, e \rangle, \langle c, e \rangle, \langle c, d \rangle, \langle d, e \rangle \}$ 과 각 간선에 대한 가중치 $w = \{1, 3, 4, 1, 5, 2, 2\}$ ($w \geq 0$)로 구성된 방향그래프의 예를 보여준다.

2.2 EMFG

2.2.1 EMFG의 정의

EMFG는 박스(Box), 트랜지션(Transition), 아크(Arc)들로 구성되는 마크를 갖는 방향성 선도로 정의되며, 다음과 같이 5개의 구성 원소로 이루어져 있다[11].

$$G = \{B, T, I, O, M\}$$

B : 박스(Box)의 집합

T : 트랜지션(Transition)의 집합 식 2

I : 트랜지션(Transition)에 대한 입력함수

O : 트랜지션(Transition)에 대한 출력함수

M : 각 박스(Box)의 마크 상태

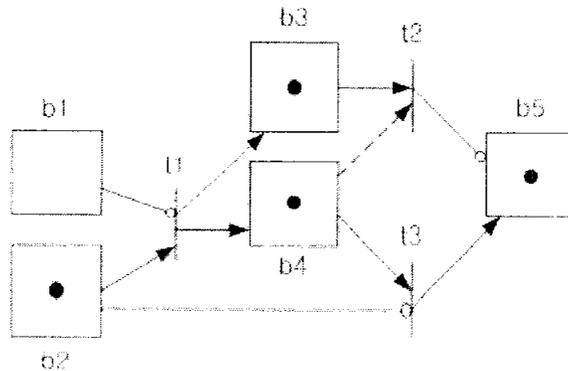


그림 2. EMFG의 예1

박스(Box)는 상태(개념적인 상태, 제어 상태, 신호 상태 등)를 나타내며, 실선의 사각형으로 표기한다. 상태의 만족 여부를

는 박스(Box) 내에 마크로 표현하는데 박스(Box)의 상태가 만족되면 마크를 두고, 만족되지 않으면 마크를 두지 않는다.

트랜지션(Transition)은 박스(Box)의 상태가 조합되어 박스(Box) 자신의 상태가 변화하거나 다른 박스(Box)의 상태를 변화시키는 곳, 즉 전이가 일어나는 곳으로 막대를 두어 표기한다. 그리고 트랜지션(Transition)에서 전이가 일어나는 과정을 트랜지션(Transition)이 점화(fire)한다고 한다. 트랜지션(Transition)은 일반트랜지션(Transition)과 시간트랜지션(Time Transition)으로 구분되어 동작하며, 일반트랜지션(Transition)은 점화조건이 만족되는 순간에 즉시 점화 완료한다. 그러나 시간트랜지션(Time Transition)은 점화조건이 만족되어도 트랜지션(Transition)에 설정된 점화시간이 경과하지 않으면 점화를 완료하지 못한다. 또한 시간트랜지션(Time Transition)은 점화조건이 만족되어 점화 중이라도 다시 점화조건이 만족되지 않는다면 점화를 중지하게 된다.

아크(Arc)는 박스(Box)와 트랜지션(Transition) 사이에서 트랜지션(Transition)의 점화조건을 결정하는 것으로 일반 아크와 조건 아크, 역 아크로 구분되며, 트랜지션(Transition)이 점화될 때의 박스(Box)의 마크 상태를 결정한다. 일반 아크는 일반적인 화살표로 표시하며, 조건 아크는 점선인 화살표로, 역 아크는 둥근 머리 화살표로 표현한다.

2.2.2 EMFG의 동작

1) 트랜지션의 점화조건

트랜지션(Transition)의 점화조건은 트랜지션(Transition)의 모든 입력 박스(Box)의 마크 상태가 모두 조건에 부합되어야 한다. 이 조건들은 연결된 아크(Arc)의 종류에 따라 달라진다. 즉 일반 아크와 조건 아크로 연결된 입력 박스(Box)에는 마크가 존재하여야 하며, 역 아크로 연결된 입력 박스(Box)에는 마크가 없어야 조건에 부합된다.

예를 들어 그림 2의 EMFG에서, 트랜지션 t_1 으로 연결된 입력 박스(Box)의 상태를 보면 역 아크로 연결된 박스 b_1 에는 마크가 없고, 일반 아크로 연결된 박스 b_2 에는 마크가 존재하므로 트랜지션 t_1 의 점화 조건에 부합된다. 하지만, 트랜지션 t_3 의 경우에는 역 아크로 연결된 박스 b_2 에 마크가 존재하므로 점화할 수 없다.

2) 트랜지션의 점화동작

트랜지션(Transition)이 점화하는 경우, 트랜지션(Transition)에 연결된 박스(Box)의 마크 상태는 연결된 아크(Arc)의 종류에 따라 다르게 변화한다.

일반 아크로 연결된 입력 박스(Box)의 마크는 소멸되고, 조건 아크로 연결된 입력 박스(Box)의 마크 상태에는 변화가 없다. 트랜지션(Transition)에서 일반 아크나 조건 아크로 연결된

출력 박스(Box)에는 마크가 없으면 생성시키고, 마크가 있으면 그대로 유지시킨다. 또 트랜지션(Transition)에서 역 아크로 연결된 출력 박스(Box)에는 마크가 있으면 소멸시키고, 마크가 없으면 그대로 유지시킨다. 또한 EMFG는 충돌박스(Collision Box : 두 개 이상의 출력 아크를 가진 박스)의 마크가 다수의 출력 트랜지션(Transition)의 점화조건을 동시에 만족시키는 경우, 모든 트랜지션(Transition)이 동시에 점화된다[11].

그림 2의 EMFG에서 트랜지션 t_1 이 점화한 결과로 박스 b_2 의 마크는 소멸되고, 박스 b_3 와 b_4 에는 마크가 생성되어야 하는데, 마크가 이미 존재하므로 그대로 유지시킨다.

2.2.3 EMFG의 특징

EMFG는 Petri Net에서 파생된 그래프 이론으로 Petri Net의 도달가능성, 생동성, 안전성 등의 모든 성질을 가지고 있다. 특히, EMFG는 Petri Net에서 시스템의 안전성을 확보하기 위해 파생된 이론이므로 시스템의 안전성을 쉽게 나타낼 수 있다. 안전성은 하드웨어 장비의 모델링에서 매우 중요한 특성으로, 박스(Box)가 안전하면 그 박스(Box)에 있는 마크의 수는 0이거나 1이 되며, 모든 박스(Box)가 안전하면 EMFG의 안전성, 즉 시스템의 안전성을 확보할 수 있다.

EMFG의 용도를 살펴보면 설계자의 생각을 그대로 표현하기에 적합한 강력한 설계 도구로서 시스템의 개념적 설계나 시스템 동작의 상세 설계 가능하고, EMFG를 구성하고 있는

요소를 각각 일대 일로 변환하면 직접적인 회로를 얻을 수 있어 시스템의 구현이 아주 쉽다.

또한 기존의 EMFG를 단순화시켜 표현하므로 시스템의 설계가 용이하고, 수학적으로 표현가능하며, 시스템의 분석이나 해석이 쉽게 이루어지고 컴퓨터 프로그램에 의해서도 그 분석 및 해석이 가능하다는 장점을 가지고 있다.

3. 방향그래프를 EMFG로 변환

정점과 간선의 집합으로 구성된 방향그래프 G 를 박스(Box)와 시간트랜지션(Time Transition)과 아크(Arc)의 집합으로 구성된 EMFG로 변환하며, 각 간선에 대한 가중치는 EMFG 트랜지션(Transition)의 점화시간으로 변환한다.

3.1 EMFG의 마킹시간(t_w)

방향그래프 G 의 각 간선에 대한 가중치(w)를 이용하여 EMFG의 마킹시간(t_w)을 구한다.

정의 1. 방향그래프에서 각 간선에 대한 가중치를 평가하는 기준 단위 가중치를 방향그래프 기준가중치 s_w 라고 정의하며, 기준가중치 s_w 는 EMFG의 마킹시간 t_w 로 표현된다.

정리 1. 방향그래프의 기준가중치 s_w 는 다음과 같다.

$$s_w = G.C.M(w_1, w_2, \dots, w_i, \dots, w_n) \quad \text{식 3}$$

식 3에서 w_i 는 임의의 간선에 대한 가중치를 나타내며, n 은

간선의 수이다.

3.2 EMFG의 박스

방향그래프 G 의 정점은 EMFG에서 개념적인 상태, 제어 상태, 신호 상태 등을 나타내는 박스(Box)로 표현한다.

정의 2. 방향그래프 G 에서 정점은 EMFG의 박스(Box)로 표현된다.

방향그래프 G 의 한 정점을 a 라 할 때, 정점 a 는 EMFG의 박스(Box) a 로 표현되며, 그림 3에서 표현된 예를 보여주고 있다.

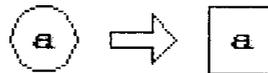


그림 3. 방향그래프 정점을 EMFG로 표현

3.3 EMFG의 시간트랜지션과 아크

방향그래프 G 의 간선은 EMFG에서 박스(Box)의 상태가 변화하거나 변화시키는 곳, 전이가 일어나도록 하는 트랜지션(Transition)에 집화시간이 포함된 시간트랜지션(Time Transition)과 트랜지션(Transition)의 집화조건과 박스(Box)의

마크상태를 결정하는 아크(Arc)로 표현된다.

정의 3. 방향그래프 G 에서 간선은 EMFG의 시간트랜지션(Time Transition)과 아크(Arc)로 표현된다.

3.4 EMFG 시간트랜지션의 점화시간

방향그래프 G 의 정점에서 출력되는 간선의 가중치는 EMFG에서 특정한 시간 후 박스(Box)의 상태가 변화하거나 변화시키는, 전이가 일어나도록 하는 시간트랜지션(Time Transition)의 점화시간으로 표현된다.

정의 4. 방향그래프 G 의 정점에서 출력되는 간선의 가중치는 EMFG의 시간트랜지션(Time Transition)의 점화시간으로 표현된다.

방향그래프 G 의 정점에서 출력되는 간선의 수와 각 간선에 대한 가중치에 따라 EMFG의 표현법이 달라지며, 그 표현법은 다음과 같다.

(1) 방향그래프의 정점에서 출력되는 간선의 수가 1 또는 간선에 대한 가중치 w 가 기준가중치 s_w 인 경우

방향그래프 G 의 간선은 EMFG의 시간트랜지션(Time Transition)과 아크(Arc)로 표현되며, 간선에 대한 가중치 w

는 EMFG 시간트랜지션(Time Transition) 점화시간으로 표현된다.

방향그래프 G 의 한 정점 a 에서 출력되는 간선에 대한 가중치를 w 라고 할 때, w 는 EMFG의 시간트랜지션(Time Transition) 점화시간 $t_{(w)}$ 와 아크(Arc)로 표현되며, 그림 4에서 표현된 예를 보여주고 있다.

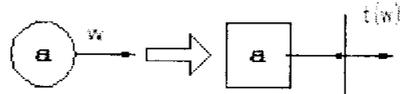


그림 4. 방향그래프 간선을 EMFG로 표현1

(2) 방향그래프 G 의 정점에서 출력되는 간선의 수가 1보다 큰 경우

방향그래프 G 의 간선은 EMFG의 시간트랜지션(Time Transition)과 아크(Arc)로 표현되며, 간선에 대한 가중치 w 는 EMFG의 박스(Box)와 아크(Arc), 시간트랜지션(Time Transition) 점화시간으로 표현된다. 이때 간선에 대한 가중치 w 에 의해 표현된 시간트랜지션(Time Transition)의 점화시간은 $w - s_w$ (기준가중치)의 시간을 가지게 된다. 표현된 EMFG는 점화조건이 고려되고, EMFG의 시간트랜지션(Time Transition)의 점화시간 합이 방향그래프 G 의 간선 E 의 가중치 w 값과 일치해야 한다. 생성된 EMFG의 박스(Box)는 s_w

를 점화시간으로 하는 시간트랜지션(Time Transition)의 출력측에 위치하게 된다.

간선에 대한 가중치 w 를 EMFG의 박스(Box)와 아크(Arc), 시간트랜지션(Time Transition) 점화시간으로 표현하는 방법은 다음과 같다.

① 방향그래프 G의 간선에 대한 가중치 w 를 식 4에 의해

s_w, w_x 로 분리

$$w = s_w + w_x \quad \text{식 4}$$

② 식 4를 참고하여

(a) EMFG 시간트랜지션(Time Transition) 점화시간 s_w

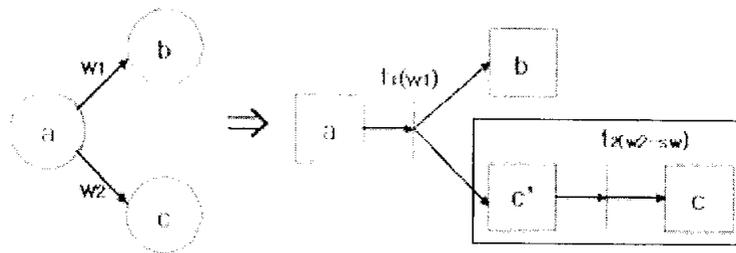
출력측에 새로운 EMFG 박스(Box)와 아크(Arc)를 생성

(b) w_x 는 생성된 EMFG 박스(Box)의 출력측에 아크

(Arc)와 EMFG 시간트랜지션(Time Transition) 점화시간으로 변환

방향그래프 G의 한 정점 a에서 출력되는 간선의 수가 두 개이고, 간선에서 출력되는 가중치값이 w_1 과 w_2 라고 하고, 기준가중치 $s_w = \text{G.C.M}(w_1, w_2) = w_1$ 이라고 할 때, 방향그래프 G의 간선의 가중치 w_1 은 (1)의 방향그래프의 정점에

서 출력되는 간선의 수가 1 또는 간선에 대한 가중치 w 가 기준가중치 s_w 인 경우이기 때문에 EMFG의 표현법은 아크(Arc)와 시간트랜지션(Time Transition) 점화시간 $t_1(w_1)$ 로 표현되며, 방향그래프 G 의 간선의 가중치 w_2 는 (2)의 방향그래프의 정점에서 출력되는 간선의 수가 1 보다 큰 경우이기 때문에 EMFG의 표현법은 EMFG 시간트랜지션(Time Transition) 점화시간 s_w 의 출력측에 새로운 박스 c' 와 아크(Arc)를 생성하고, 생성된 박스 c' 의 출력측에 아크(Arc)와 식 4에 의해 구해진 $w_2 - w_1$ 는 시간트랜지션(Time Transition) 점화시간으로 표현되며, 그림 5에서 표현된 예를 보여주고 있다.



$$s_w = \text{G.C.M}(w_1, w_2) = w_1$$

그림 5. 방향그래프 간선을 EMFG로 표현2

방향그래프 G 에서 변환된 EMFG의 동작을 수학적으로 해석하기 위하여, 먼저 EMFG에서 트랜지션(Transition)들에 대한 박스(Box)들의 접속 상태를 수학적으로 표현하기로 한다.

4. 최단경로 알고리즘

4.1 EMFG 시간트랜지션을 이용한 최단경로 알고리즘

단계 1 : 초기화 단계

EMFG의 동작에 필요한 초기값을 구한다.

① EMFG의 마킹시간 t_m [15] 구함

식 5를 사용하여 EMFG의 마킹시간을 구한다.

$$t_m = \text{G.C.M.}(s_1, s_2, \dots, s_j, \dots, s_n) \quad \text{식 5}$$

여기서 n 은 트랜지션(Transition)의 수이며, s_j 는 트랜지션(Transition) t_j 의 점화시간이다.

② 초기 완료비 벡터 V_0 [15] 구함

식 6를 사용하여 EMFG의 초기 완료비 벡터를 구한다.

$$V_0 = \left[\frac{s_1}{t_m}, \frac{s_2}{t_m}, \dots, \frac{s_n}{t_m} \right] \quad \text{식 6}$$

여기서 t_m 은 마킹시간, n 은 트랜지션(Transition)의 수, s_j

는 트랜지션(Transition) t_j 의 점화시간이다.

③ 입력 행렬(Input Matrix) T_i 구함

식 7를 사용하여 EMFG의 입력 행렬을 구한다.

$$T_i = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad \text{식 7}$$

여기서 n 은 트랜지션(Transition)의 수, m 은 박스(Box)의 수, a_{ij} 는 트랜지션(Transition) t_j 에 연결된 입력박스 b_i 의 연결 상태를 나타내며, a_{ij} 가 일반 아크로 연결되었으면 $a_{ij} = 1$ 의 값이 되고, a_{ij} 가 아크(Arc)연결이 없는 경우에는 $a_{ij} = 0$ 의 값을 가지게 된다.

④ 출력 행렬(Output Matrix) T_o 구함

식 8를 사용하여 EMFG의 출력 행렬을 구한다.

$$T_o = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad \text{식 8}$$

a_{ij} 가 일반 아크로 연결되었으면 $a_{ij} = n$ 의 값이 되고, a_{ij} 가 아크

(Arc) 연결이 없는 경우에는 $a_{ij} = 0$ 의 값을 가지게 된다.

⑤ 접속 행렬 T_b [15] 구함

식 9을 사용하여 EMFG의 접속 행렬을 구한다.

$$T_b = T_o - T_i. \quad \text{식 9}$$

⑥ 점화조건 행렬 C 구함

식 10을 사용하여 EMFG의 점화조건 행렬을 구한다.

$$C = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad \text{식 10}$$

a_{ij} 가 일반 아크로 연결되었으면 $a_{ij} = 1$ 의 값이 되고, a_{ij} 가 아크(Arc) 연결이 없는 경우에는 $a_{ij} = \times$ 의 값을 가지게 된다.

⑦ 초기 마크 벡터 M_0 [15] 구함

박스(Box)들의 마크 상태에 따라 EMFG의 초기 마크 벡터를 구한다.

단계 2 : 실행단계

① 점화가능 벡터 F_k 구함

식 11을 사용하여 점화가능 벡터 F_k 을 구한다.

$$F_k = M_{k-1} \blacksquare C \quad \text{식 11}$$

여기서 M_{k-1} 은 마크벡터, C 는 점화조건 행렬을 의미하며, 마크벡터 M_{k-1} 와 점화조건 행렬 C 의 각 대응값이 (0, 1)의 요소를 가지면 $F_k = 0$, 그렇지 않으면 $F_k = 1$ 의 값을 가진다.

② 예상완료비 벡터 V_k 구함

k 가 1인 경우에는 초기 완료비 벡터 V_0 로 하고, k 가 2 이상인 경우는 경과 완료비 X_{k-1} 에서 0 이하의 모든 원소를 초기 완료비로 대체하여 구한다.

③ 현완료비 벡터 W_k [15] 구함

식 12을 사용하여 현완료비 벡터를 구한다.

$$W_k = V_k \star F_k \quad \text{식 12}$$

W_k 는 예상완료비 벡터 V_k 의 각 원소와 점화가능 벡터 F_k 의 각 원소의 곱의 값을 가진다.

④ 경과완료비 벡터 X_k [15] 구함

식 13를 사용하여 경과완료비 벡터를 구한다.

$$X_k = W_k - 1 \quad \text{식 13}$$

⑤ 점화완료 벡터 Y_k 구함

Y_k 는 $X_k = 0$ 이면 $Y_k = 1$ 의 값으로 대치하고, $X_k \neq 0$ 이면 $Y_k = 0$ 의 값으로 대치하여 구한다.

⑥ 다음 마크 벡터 M_k [15] 구함

식 14를 사용하여 다음 마크 벡터를 구한다.

$$U = M_{k-1} + Y \cdot T_b^T \quad \text{식 14}$$

$U \leq 0$ 이면 $M_k = 0$ 의 값으로 대치하고 $U > 0$ 이면 $M_k = 1$ 의 값으로 대치하여 구한다.

단계 3 : 최단경로 확인 단계

현재 마크벡터 M_k 와 이전 마크벡터 M_{k-1} 을 비교하여 다음 과정에 따라 최단경로를 구한다.

$N = M_k$ 의 마크의 수 라고 할 때,

① M_k 에 새로운 마크가 생성되면, 마크가 생성된 박스

(Box)의 최단경로가 결정되고, 마크가 생성된 박스(Box)를 true로 설정하고 모든 박스(Box)가 true가 되면 프로그램을 종료한다.

② $N = 1$ 이면, 단계 3으로 가고, $N > 1$ 이면, M_k 에 생성된 마크를 각각 구분한 뒤, 각각의 마크에 대해 단계 3으로 간다.

방향그래프 G 의 모든 정점의 개수가 n 이라 할 때, 모든 정점 n 개간에 간선이 $\frac{n(n-1)}{2}$ 개로 모두 연결된 완전방향그래프에서, 특정한 한 정점에서 다른 모든 정점으로 연결된 간선의 가중치가 동일하다면, 점화조건행렬 C 의 모든 요소 값은 '1'을 가지게 되며, 점속행렬 T_b 의 모든 요소의 값은 'EMFG'의 트랜지션(Transition)의 수 - 1'을 가지게 된다. 따라서 점화조건행렬 C 와 점속행렬 T_b 에 의해 계산되는 마크벡터의 흐름을 간략화 할 수 있다.

4.2 계산복잡도 계산 및 비교·분석

최단경로 문제는 여러가지의 경로 문제들 중 기본형태이다. N.Deo, C.Pang(1984) 등은 최단경로 문제의 연구 현황을 잘 분류 정리하였다. 현재까지 연구된 바 있는 최단경로 문제의 해법의 계산복잡도를 정리하면 다음과 표 1과 같다.[18]

방향성	가중치	계산복잡도	비고
유방향	비음인 정수	$O(n^2)$	Dijkstra
	정수	$O(n^3)$	Bellman-Ford
무방향	정수	$O(n^3)$	Non-Bipartit Matching

표 1. 최단경로 문제의 계산 복잡도

EMFG의 시간트랜지션(Time Transition)을 이용한 최단경로 알고리즘의 계산복잡도는 트랜지션(Transition) 수를 n , 박스(Box) 수를 m , 최단거리를 s 라고 할 때

마킹시간 t_m 계산에 $O(n)$, 초기완료비벡터 V_0 계산에 $O(n)$, 접속행렬 T_b 계산에 $O(n \times m)$, 점화가능벡터 F_k 계산에 $O(n \times m \times s)$, 예상완료비벡터 V_k 계산에 $O(m \times s)$, 현완료비벡터 W_k 계산에 $O(m \times s)$, 경과완료비벡터 X_k 계산에 $O(m \times s)$, 점화완료비벡터 Y_k 계산에 $O(m \times s)$, 마크벡터 M_k 계산에 $O(n \times m \times s)$ 이므로

전체계산복잡도는 $n = m = s$ 라고 할 때, $O(n) + O(n) + O(n^2) + O(n^3) + O(n^2) + O(n^2) + O(n^2) + O(n^2) + O(n^3) = O(n^3)$ 이다.

복수 최단경로 문제의 새로운 해법연구[17]에서 이중 최단경로나무(DSA)에 의해 복수 최단경로 해법을 구함에 있어, 전체 계산복잡도는 $O(n^3)$ 이지만 특정 부분, 즉 $k = 3$ 인 경우에는

계산복잡도는 $O_{(n^3)}$ 으로 제시되고 있다.

본 논문에서도 방향그래프 G의 모든 정점이 완전하게 연결되어진 완전방향그래프에서, 특정한 한 정점에서 다른 모든 정점으로 연결된 간선의 가중치가 동일하다면, EMFG의 시간트랜지션(Time Transition)을 이용한 최단경로 알고리즘의 계산 복잡도는 트랜지션(Transition) 수를 n , 박스(Box) 수를 m , 최단거리를 s 라고 할 때,

마킹시간 t_m 계산에 $O_{(n)}$, 초기완료비벡터 V_0 계산에 $O_{(n)}$, 점속행렬 T_b 계산에 $O_{(n \times m)}$, 점화가능벡터 F_k 계산에 $O_{(m \times s)}$, 예상완료비벡터 V_k 계산에 $O_{(m \times s)}$, 현완료비벡터 W_k 계산에 $O_{(m \times s)}$, 경과완료비벡터 X_k 계산에 $O_{(m \times s)}$, 점화완료비벡터 Y_k 계산에 $O_{(m \times s)}$, 마크벡터 M_k 계산에 $O_{(m \times s)}$ 이므로

전체계산복잡도는 $n = m = s$ 라고 할 때, $O_{(n)} + O_{(n)} + O_{(n^2)} = O_{(n^2)}$ 이 된다.

EMFG의 수학적 해석 알고리즘의 특성상 계산복잡도가 $O_{(n^3)}$ 로서 기존의 최단경로 알고리즘에 비해 높지만, 특정 조건에서는 계산복잡도가 $O_{(n^2)}$ 을 가지게 되고, 방향그래프를 EMFG로 변환하는 알고리즘을 이용함으로써, 기존의 최단경로 알고리즘에서 제시된 고전적인 컴퓨터 모델에 의한 실계 분석

에서 벗어나 시간의 경과에 따른 경로의 흐름을 파악하는 EMFG의 마크 상태변화를 통해 최단경로를 구하는 새로운 해법이 제시되어 마크벡터의 변화과정을 통한 개념적, 직관적 해석이 가능한 최단경로 알고리즘의 설계가 가능해졌다. 뿐만 아니라 EMFG의 특징상 하드웨어적 시스템의 구현을 통한 설계가 가능하며, 이를 통한 분석을 통하여 시스템의 성능향상을 기대할 수 있다.

최단경로 알고리즘에서 기존에 제시된 문제해법과 본 논문에서 제시된 알고리즘의 특징을 비교·분석하면 다음과 같다.

	기존방식	제안된 방식
해석	컴퓨터 모델	마크벡터의 변화과정
구현	소프트웨어	소프트웨어와 하드웨어 모두 가능
계산복잡도	$O(n^2)$	$O(n^3)$, 특정 조건에서 $O(n^2)$

표 2. 알고리즘 비교·분석

5. 알고리즘 적용 예

5.1 방향그래프를 EMFG로 변환 적용 예

방향그래프 G 의 정점 V 와 간선 E 의 집합은 EMFG의 박스 B 와 트랜지션(Transition) T 의 집합으로 변환되며, 방향그래프 G 의 각 간선에 대한 가중치 w 는 EMFG 트랜지션(Transition) T 의 점화시간으로 변환한다.

그림 6은 방향그래프 G 의 예로서

$V(G) = \{a, b, c, d, e, f, g\}$ 과

$E(G) = \{\langle a,b \rangle, \langle a,c \rangle, \langle a,d \rangle, \langle b,c \rangle, \langle b,e \rangle, \langle c,d \rangle, \langle c,e \rangle, \langle d,f \rangle, \langle e,c \rangle, \langle e,g \rangle, \langle f,g \rangle\}$,

$w = \{1, 3, 1, 1, 5, 1, 2, 3, 2, 4\}$ 인 방향그래프이다.

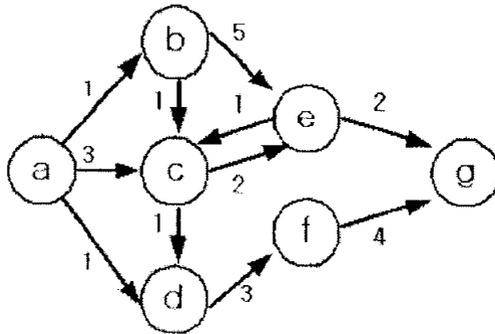


그림 6. 방향그래프의 예2

그림 6의 방향그래프 G의 간선의 기준가중치 $s_w = G.C.M(1, 3, 1, 1, 1, 5, 1, 2, 3, 2, 4) - 1$ 이며, 방향그래프를 EMFG로 변환하는 방법을 적용하여 EMFG로 변환한다면 그림 5와 같이 표현되며, 다음과 같다.

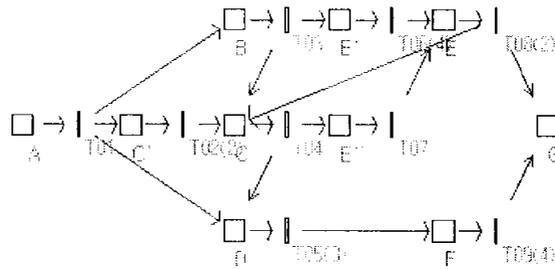


그림 7. EMFG로 변환된 방향그래프

그림 7에서 EMFG의 박스 B와 트랜지션(Transition) T, 그리고 각 트랜지션(Transition)의 입력함수와 출력함수를 구하면 다음과 같다.

$$B = \{a, b, c', c, d, e', e'', e, f, g\} \text{과}$$

$$T = \{t01, t02(2), t03, t04, t05(3), t06(4), t07, t08(2), t09(4)\}$$

$$I(t01) = \{a\} \qquad O(t01) = \{c', b, d\}$$

$$I(t02(2)) = \{c'\} \qquad O(t02(2)) = \{c\}$$

$$I(t03) = \{b\} \qquad O(t03) = \{c, e'\}$$

$$I(t04) = \{c\} \qquad O(t04) = \{d, e''\}$$

$$I(t05(3)) = \{d\}$$

$$O(t05(3)) = \{f\}$$

$$I(t06(4)) = \{e'\}$$

$$O(t06(4)) = \{e\}$$

$$I(t07) = \{e''\}$$

$$O(t07) = \{e\}$$

$$I(t08(2)) = \{c\}$$

$$O(t08(2)) = \{c, g\}$$

$$I(t09(4)) = \{f\}$$

$$O(t09(4)) = \{g\}$$

5.2 EMFG 시간트랜지션을 이용한 최단경로 적용 예

EMFG 시간트랜지션(Time Transition)을 이용한 최단경로를 해석하기 위해 점속행렬을 이용한 EMFG의 수학적 해석[15]이 적용된 알고리즘을 프로그램 구현을 통한 시뮬레이션과정을 그림 7에 적용하여 표현하면 다음과 같다.

단계 1 : 초기화 단계

① EMFG의 마킹시간 t_m [15] 구함

$$t_m = \text{G.C.M.}(1, 2, 1, 1, 3, 4, 1, 2, 4) = 1$$

② 초기 완료비 백터 V_0 [15] 구함

$$\begin{aligned} V_0 &= [1/1, 2/1, 1/1, 1/1, 3/1, 4/1, 1/1, 2/1, 4/1] \\ &= [1, 2, 1, 1, 3, 4, 1, 2, 4] \end{aligned}$$

③ 입력 행렬(Input Matrix) T_i 구함

$$T_i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

④ 출력 행렬(Output Matrix) T_o 구함

$$T_o = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 9 & 0 & 0 & 0 & 0 & 9 & 0 \\ 9 & 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 9 \end{bmatrix}$$

④ 접속 행렬 T_b [15] 구함

$$T_b = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 9 & -1 & 0 & 0 & 0 & 9 & 0 \\ 9 & 0 & 0 & 9 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 & 9 & -1 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 9 \end{bmatrix}$$

⑤ 점화조건 행렬 C 구함

$$C = \begin{bmatrix} 1 & x & x & x & x & x & x & x & x \\ x & 1 & x & x & x & x & x & x & x \\ x & x & 1 & x & x & x & x & x & x \\ x & x & x & 1 & x & x & x & x & x \\ x & x & x & x & 1 & x & x & x & x \\ x & x & x & x & x & 1 & x & x & x \\ x & x & x & x & x & x & 1 & x & x \\ x & x & x & x & x & x & x & 1 & x \\ x & x & x & x & x & x & x & x & 1 \\ x & x & x & x & x & x & x & x & x \end{bmatrix}$$

⑥ 초기 마크 벡터 M_0 [15] 구함

$$M_0 = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

단계 2 : 실행단계

① 점화가능 벡터 F_k 구함

$$F_1 = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

② 예상완료비 벡터 V_k 구함

$$V_1 = [1, 2, 1, 1, 3, 4, 1, 2, 4]$$

③ 현완료비 벡터 W_k [15] 구함

$$W_1 = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

④ 경과완료비 벡터 X_k [15] 구함

$$X_1 = [0, -1, -1, -1, -1, -1, -1, -1, -1, -1]$$

⑤ 점화완료 벡터 Y_k 구함

$$Y_1 = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

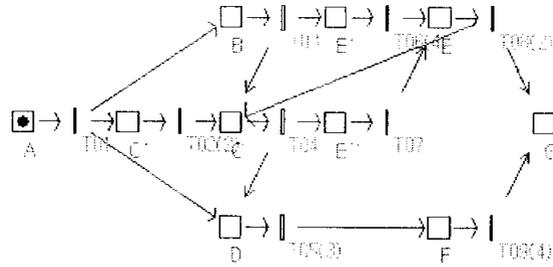
⑥ 다음 마크 벡터 M_k [15] 구함

$$M_1 = [0, 1, 1, 0, 1, 0, 0, 0, 0, 0]$$

단계 3 : 최단경로 확인 단계

① 초기 마크 벡터 M_0 가 설정된 단계

여기선 EMFG의 박스 A를 출발점으로 한다.

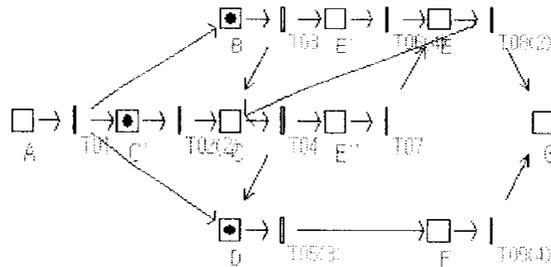


$$M_0 = [1, 0, 0, 0, 0, 0, 0, 0, 0]$$

그림 8. EMFG의 시뮬레이션과정1

② 최단경로 알고리즘을 1회 진행한 결과

EMFG 박스 C', B, D에 마크가 생성되었으며, 시작점 A에서 B, D의 최단경로가 결정됨.



$$M_1 = [0, 1, 1, 0, 1, 0, 0, 0, 0]$$

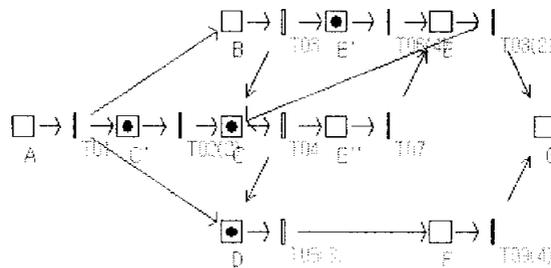
그림 9. EMFG의 시뮬레이션과정2

단계	경로		거리
	시작점	도착점	
1	A	C', B, D	1
단계 1 최단경로 확인 : A → B, A → D			

표 3. 최단경로 시뮬레이션 진행과정1

③ 최단경로 알고리즘을 2회 진행한 결과

EMFG 박스 C, E'에 마크가 생성되었으며, 시작점 A에서 C의 최단경로가 결정됨.



$$M_1 = [0, 1, 0, 1, 1, 1, 0, 0, 0, 0]$$

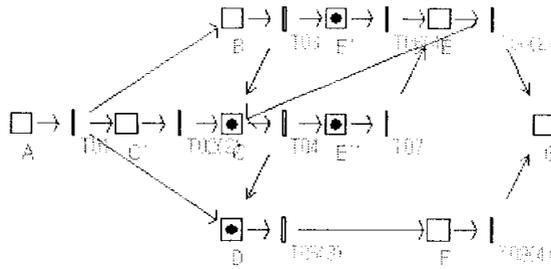
그림 10. EMFG의 시뮬레이션과정3

단계	경로		거리
	시작점	도착점	
1	A	C', B, D	1
2	A	C, E'	2
단계 2 최단경로 확인 : A → C			

표 4. 최단경로 시뮬레이션 진행과정2

④ 최단경로 알고리즘을 3회 진행한 결과

EMFG 박스 E''에 마크가 생성되었음.



$$M_1 = [0, 0, 0, 1, 1, 1, 1, 0, 0, 0]$$

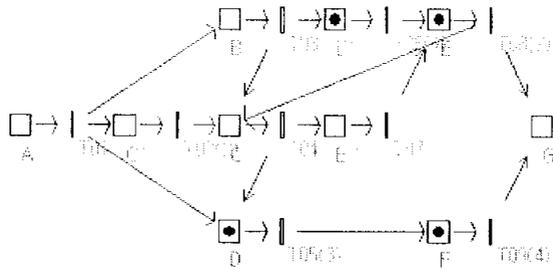
그림 11. EMFG의 시뮬레이션과정4

단계	경로		거리
	시작점	도착점	
1	A	C', B, D	1
2	A	C, E'	2
3	A	E''	3

표 5. 최단경로 시뮬레이션 진행과정3

⑤ 최단경로 알고리즘을 4회 진행한 결과

EMFG 박스 E, F에 마크가 생성되었으며, 시작점 A에서 E, F의 최단경로가 결정됨.



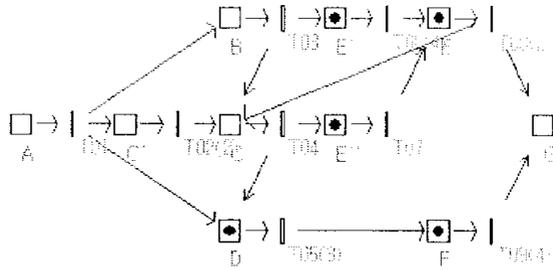
$$M_1 = [0, 0, 0, 0, 1, 1, 0, 1, 1, 0]$$

그림 12. EMFG의 시뮬레이션과정5

단계	경로		거리
	시작점	도착점	
1	A	C', B, D	1
2	A	C, E'	2
3	A	E''	3
4	A	E, F	4
단계 4 최단경로 확인 : A → E, A → F			

표 6. 최단경로 시뮬레이션 진행과정4

(6) 최단경로 알고리즘을 5회 진행한 결과



$$M_1 = [0, 0, 0, 0, 1, 1, 0, 1, 1, 0]$$

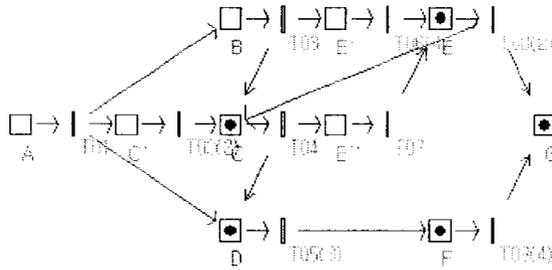
그림 13. EMFG의 시뮬레이션과정6

단계	경로		거리
	시작점	도착점	
1	A	C', B, D	1
2	A	C, E'	2
3	A	E''	3
4	A	E, F	4
5	A	.	5

표 7. 최단경로 시뮬레이션 진행과정5

(7) 최단경로 알고리즘을 6회 진행한 결과

EMFG 박스 G에 마크가 생성되었으며, 시작점 A에서 G의 최단경로가 결정됨.



$$M_1 = [0, 0, 0, 0, 1, 1, 0, 1, 1, 0]$$

그림 14. EMFG의 시뮬레이션과정7

단계	경로		거리
	시작점	도착점	
1	A	C', B, D	1
2	A	C, E'	2
3	A	E''	3
4	A	E, F	4
5	A	.	5
6	A	G	6

단계 6 최단경로 확인 : A → G
 시작점에서 모든 다른 정점까지의 최단경로확인

표 8. 최단경로 시뮬레이션 진행과정6

⑧ 전체 실행결과

Step	점화가능 벡터(F)	예상 완료비(V)	현완료비(W)	경과완료비(O)	점화완료벡터(Y)	마크벡터(M)
1	1 0 0 0 0 0 0 0 0	1 2 1 1 3 4 1 2 4	1 0 0 0 0 0 0 0 0	0 -1 -1 -1 -1 -1 -1 -1	1 0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0 0
3	0 1 1 0 1 0 0 0 0	1 2 1 1 3 4 1 2 4	0 2 1 0 0 0 0 0 0	-1 0 1 0 1 3 -1 -1	0 0 1 0 0 0 0 0 0	0 1 0 1 1 0 0 0 0
4	0 0 0 1 1 1 0 0 0	1 2 1 1 3 4 1 2 4	0 0 0 1 1 3 1 0 0	-1 -1 -1 0 0 2 0 -1	0 0 0 1 1 0 1 0 0	0 0 0 0 1 1 1 1 0
5	0 0 0 0 1 1 1 1 1	1 2 1 1 3 4 1 2 4	0 0 0 0 0 2 1 2 4	-1 -1 -1 2 1 0 1 3	0 0 0 0 0 1 0 1 0	0 0 0 0 1 1 0 1 1
6	0 0 0 0 1 1 0 1 1	1 2 1 1 3 4 1 2 4	0 0 0 0 0 2 1 0 1 3	-1 -1 -1 1 0 -1 0 2	0 0 0 0 0 1 0 1 0	0 0 0 0 1 1 0 0 1 1

그림 15. EMFG의 시뮬레이션 전체 실행 과정

단계	경로	경로 거리
1	A → B, A → D	1
2	A → B → C	2
4	A → B → C → E, A → D → F	4
6	A → B → C → E → G	6

표 9. EMFG의 시뮬레이션 전체 실행 결과

5.3 마크벡터 경로흐름에 따른 최단경로

Step	점화가능 벡터(F)	예상 완료비(V)	현완료비(W)	경과완료비(O)	점화완료벡터(Y)	마크벡터(M)
1	1 0 0 0 0 0 0 0 0	1 2 1 1 3 4 1 2 4	1 0 0 0 0 0 0 0 0	0 -1 -1 -1 -1 -1 -1 -1	1 0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0 0
→ 마크벡터(M) (0,1,1,0,0,0,0,0,0) (0,1,0,0,0,0,0,0,0) (0,0,1,0,0,0,0,0,0) (0,0,0,0,1,0,0,0,0)으로 분리						
→ 선정된 마크벡터(A)에서 (0,1,0,0,0,0,0,0,0)를 선택						
1-1	1 0 0 0 0 0 0 0 0	1 2 1 1 3 4 1 2 4	1 0 0 0 0 0 0 0 0	0 -1 -1 -1 -1 -1 -1 -1	1 0 0 0 0 0 0 0 0	0 1 0 0 0 0 0 0 0
2	0 0 1 0 0 0 0 0 0	1 2 1 1 3 4 1 2 4	0 0 1 0 0 0 0 0 0	-1 -1 0 -1 -1 -1 -1 -1	0 0 1 0 0 0 0 0 0	0 0 0 1 0 1 0 0 0 0
→ 마크벡터(X) (0,0,0,1,0,1,0,0,0,0)를 (0,0,0,1,0,0,0,0,0) (0,0,0,0,0,1,0,0,0,0)으로 분리						
→ 선정된 마크벡터(A)에서 (0,0,0,1,0,0,0,0,0)를 선택						
2-1	0 0 0 1 0 0 0 0 0	1 2 1 1 3 4 1 2 4	0 0 0 1 0 0 0 0 0	-1 -1 -1 0 -1 -1 -1 -1	0 0 0 1 0 0 0 0 0	0 0 0 1 0 0 0 0 0 0
3	0 0 0 1 0 0 0 0 0	1 2 1 1 3 4 1 2 4	0 0 0 1 0 0 0 0 0	-1 -1 -1 0 -1 -1 -1 -1	0 0 0 1 0 0 0 0 0	0 0 0 0 1 0 1 0 0 0
→ 마크벡터(O) (0,0,0,0,1,0,1,0,0,0)를 (0,0,0,0,1,0,0,0,0) (0,0,0,0,0,0,1,0,0,0)으로 분리						
→ 선정된 마크벡터(A)에서 (0,0,0,0,0,1,0,0,0)를 선택						
3-1	0 0 0 0 0 0 1 0 0	1 2 1 1 3 4 1 2 4	0 0 0 0 0 0 1 0 0	-1 -1 -1 -1 -1 0 0 -1 -1	0 0 0 0 0 0 1 0 0	0 0 0 0 0 0 1 0 0 0 0
4	0 0 0 0 0 0 1 0 0	1 2 1 1 3 4 1 2 4	0 0 0 0 0 0 1 0 0	-1 -1 -1 -1 -1 0 0 -1 -1	0 0 0 0 0 0 1 0 0	0 0 0 0 0 0 0 1 0 0 0
5	0 0 0 0 0 0 1 1 0	1 2 1 1 3 4 1 2 4	0 0 0 0 0 0 0 2 0	-1 -1 -1 -1 -1 0 0 -1 -1	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1 0 0 0
6	0 0 0 0 0 0 0 1 0	1 2 1 1 3 4 1 2 4	0 0 0 0 0 0 0 1 0	-1 -1 -1 -1 -1 0 0 -1 -1	0 0 0 0 0 0 0 1 0	0 0 0 0 1 0 0 0 0 0 0
→ 마크벡터(W) (0,0,0,1,0,0,0,0,0)에서 연속 3개 마크벡터 생성						
→ 선정된 마크벡터(A) (0,0,0,0,0,0,0,0,0) (0,1,0,0,0,0,0,0,0) (0,0,0,1,0,0,0,0,0) (0,0,0,0,0,0,1,0,0,0) (0,0,0,0,0,0,0,1,0,0) (0,0,0,0,0,0,0,0,1,0,0) (0,0,0,0,0,0,0,0,0,1,0,0) (0,0,0,0,0,0,0,0,0,0,1,0,0)						

그림 16. 마크벡터흐름에 따른 최단경로(예:A⇒G)

그림 16에서 시간의 흐름에 따라 각 박스(Box)에 마크가 변화한 과정을 나타내었으며, 마크의 흐름 파악을 통해 최단경로를 쉽게 파악할 수 있다.

6. 결 론

본 논문은 통신망이나 도로망과 같은 그래프형태의 구조에서 그래프의 각 요소들이 활동을 수행하는데 필요한 비용 또는 거리가 주어졌을 때 그래프의 특정한 한 정점에서 다른 모든 정점까지의 최단경로를 구하는 문제를 EMFG를 이용한 알고리즘을 제시하였다. 방향 그래프의 최단경로를 해석함에 있어 방향그래프를 EMFG로 변환하는 과정과 EMFG의 수학적 해석 알고리즘[15]이 적용된 제안된 알고리즘을 통해 시간의 변화에 따른 마크벡터의 변화를 정확하게 판단함으로써 특정한 한 정점사이의 다른 모든 정점까지의 경로흐름을 파악함으로써 최단경로를 보다 쉽고 정확하게 확인할 수 있다. 제안된 알고리즘은 기존의 컴퓨터 네트워크 구축이나 이동통신 회사의 기기국간 연결선 회사의 각 지점간의 전용망 구축, 주요도시를 모두 연결하는 도로건설, 국가 기간통신망 구축, 물류센터와 창고간의 연결경로설정 등의 다양한 분야에서 적용된 그래프이론을 EMFG로 변환을 가능하게 함으로서 문제해결에 대한 새로운 패러다임 설정을 가능하게 했으며, 최단경로 문제 해결에 대한 알고리즘을 기존의 해법에서 정의하는 고전적인 컴퓨터 모델에 의한 설계 분석에서 벗어나 시간의 경과에 따른 경로의 흐름을 파악함으로써 마크벡터의 변화과정을 통한 개념적, 직관적 해석이 가능한 알고리즘의 설계가 가능해졌다.

또한 제안된 알고리즘은 EMFG의 수학적 해석 알고리즘의 특성상 계산복잡도가 $O(n^3)$ 로서 기존의 최단경로 알고리즘에 비해 높지만, 방향그래프 G의 모든 정점의 개수가 n이라 할 때, 간선의 수가 $n(n-1)/2$ 개인 완전방향그래프에서, 특정한 정점에서 다른 모든 정점으로 연결된 간선의 가중치가 동일할 때는 계산복잡도가 $O(n^2)$ 을 가지다. 이 과정은 구현되어진 프로그램을 이용한 시뮬레이션과정을 통해 시스템의 동작이 올바른 것을 확인 분석하였다. 그리고, EMFG는 시스템을 설계하여 직접 실제회로로 변환할 수 있으므로 최단경로 알고리즘이 내포된 이산제어시스템의 모델링이나 설계를 용이하게 하며, 나아가 설계된 시스템의 실험과 분석을 통하여 시스템의 성능향상에 도움을 줄 것으로 기대된다.

향후 연구 과제로는 다양한 그래프에서 시스템의 성능저하 없이 EMFG를 이용한 최단경로 알고리즘 이론을 적용하여 최적화된 해를 찾는 것이다.

[참고문헌]

- [1] James L. Peterson, "Petri Net Theory And The Modeling Of Systems", Englewood Cliffs, NJ: Prentice-Hall, 1981
- [2] Tadao Murata, "Petri Nets : Properties, Analysis and Applications", Proceedings of the IEEE, Vol. 77, No. 4, p541-580, 1989
- [3] Herve P. Hillion and Jean-Marie Proth, "Performance Evaluation of Job-Shop Systems Using Timed Event-Graphs", IEEE Trans on Automatic Control, Vol. 34, No. 1, January 1989
- [4] C. S. Hwang and J. M. Lee, "Analysis of Matrix Equation Based on Petri Net for Discrete System Control", Proceedings of the 29th SICE Annual Conference International Session, p639-696, July 1990.
- [5] Said Laftit, Jean-Marie Proth, "Optimization of Invariant Criteria for Event Graphs", IEEE Trans on Automatic Control, Vol. 37, No 5, May 1992
- [6] Rene David and Hassane Alla, "Petri Nets for Modeling of Dynamic Systems a Server", Automatica, Vol. 30, No. 2, p175-202, 1994
- [7] 여정보, "마크호프텐선도의 확장", 부산대학교 대학원 석사학위 논문, 1982. 2.

- [8] 여정보, 황창선, “확장된 마크흐름선도와 시퀀셜 제어 시스템에의 응용”, 부산대학교 공과대학 연구보고 Vol. 25, p. 209 - 219, 1983. 6 .
- [9] 이재만, “확장된 세이프 Petri Net를 이용한 이산시스템의 해석과 설계에 관한 연구”, 부산대학교 대학원 박사학위 논문, 1995. 2.
- [10] 여정보, “EMFG 회로의 간략화에 관한 연구”, 부산개방대학 연구보고 Vol. 29, p741 - 760 1987. 12 .
- [11] 여정보, “이산 시스템의 설계와 해석을 위한 확장된 마크흐름선도의 재정의와 회로변환”, 멀티미디어학회 논문지 Vol. 1 No. 2, p224 - 238, 1998. 12.
- [12] 여정보, 하재목, “확장된 마크흐름선도의 재구성과 회로변환”, 한국멀티미디어학회, 1998년도 춘계학술발표논문집, p. 423 - 431, 1998. 6.
- [13] 여정보, “이산제어시스템 설계를 위한 확장된 마크흐름선도의 동작해석”, 정보처리논문지 Vol. 5 No. 7, p. 1896-1907, 1998. 7.
- [14] 여정보, “확장된 마크흐름선도의 성질 및 간소화 알고리즘”, 부경대학교 논문집 Vol. 5. No. 2, p. 17-28, 1998. 12.
- [15] 김희정, 허후숙, 정안나, 여정보, “접속행렬을 이용한 EMFG의 수학적 해석”, 멀티미디어공학회, 2001년 추계학술발표논문집,
- [16] 백형구, 허후숙, 정명희, 여정보, “조건 아크를 이용한 릴레이 회로의 EMFG 변환”, 멀티미디어공학회, 2001년 추계학

술발표논문집, p.821-826, 2001. 11.

[17] 장병만, “복수 최단경로 문제의 새로운 해법 연구”, 경영과학회지, p.229-237, 1998. 11

[18] 성기석, 박순달, “수송네트워크에서 최대물동량 경로문제의 최적해법”, 한국경영과학회지, 16권 1호, 1991. 6,

[19] 김남혁, 정명희, 이정모, “EMFG를 이용한 최단경로 알고리즘”, 한국멀티미디어학회, 2002년 춘계학술발표논문집,

[20] 백형구, “릴레이 회로의 개선된 EMFG 변환”, 부경대학교 대학원 석사학위 논문, 2002. 1.

감사의 글

2년이 넘는 시간동안 적지 않은 고민과 갈등, 그 속에서 느껴지는 새로운 경험들을 겪으며 지금 이 자리에 서게 되었습니다. 지금껏 유형무형으로 도움을 준 고마운 분들께 지면으로나마 감사의 글을 드리고자 합니다.

석사과정동안 노력하는 자세의 모습을 직접 보여주시며 많은 지도와 관심을 가져주신 지도교수 여정보 교수님께 깊이 감사드립니다. 또한, 바쁘신 와중에도 귀중한 시간을 내주시어 논문에 대한 충고를 해 주신 박홍복 교수님, 김영봉 교수님께도 감사드립니다. 그밖에도 많은 학문의 가르침을 주신 박승섭 교수님, 김창수 교수님, 윤성대 교수님, 박지환 교수님께도 감사의 마음 전합니다.

아울러 연구실 생활과 논문을 진행하는 동안 어려움도 만나 하지 않고 항상 많은 조언과 도움을 아끼지 않으신 연구실 동료들에게 고마움을 전합니다. 선배이자 동료이자 후배로서 석사과정동안 함께 생활한 홍섭, 형구, 태훈, 정수, 박희광 선생님, 허선자 선생님, 정명희 선생님, 안정숙, 이미순, 홍지연, 정은옥, 이재용, 그리고 이미 졸업한 허후숙, 정안나, 김희정 선생님, 박동진 선생님, 김종민 선생님께 다시 한번 고마움을 전합니다.

제 삶에 많은 영향을 주신 박종호 선생님, 최진상 선생님과 항상 격려의 말씀을 아끼지 않으신 박재일 선생님, 신동희 선생님, 김진태 선생님, 김영자 선생님, 학사일정 기간에 대학원 생활에 많은 배려와 고마움을 주신 박혜정 선생님께 감사드립니다. 그리고 물심양면으로 저에게 도움과 은혜를 주신 교장선생님, 교감선생님, 여러 선생님께도 감사의 마음을 전합니다.

무언의 격려와 함께 항상 저의 버팀목이 되어주며 오늘이 있게 해주신 아버님, 어머님, 장인, 장모님, 바쁜 핑계로 제대로 챙겨주지도 못하지만 마음속에는 항상 함께하는 동생 무성과 처남, 처제, 세심한 관심과 사랑을 보내주는 누나에게 고마움을 드립니다.

특히 삶의 활력을 주는 오늘도 예쁘고 귀엽게 자라는 딸 모경과 힘들고 지친 속에서도 내색 않고 항상 제 옆에서 도움을 주는 사랑하는 아내에게 고마움을 전합니다.

이제, 오늘을 발판으로 한 걸음 더 나아가기 위해 언제나 최선을 다하는 내일이 되도록 열심히 노력하겠습니다.

2004년 2월

김 남 혁