

퍼지추론을 적용한 교통 신호 제어 시스템

김 미 경[†] · 이 윤 배^{††}

요 약

현행 교통 신호 제어기는 사전에 계획된 신호 시간 체계 또는 시간대별로 선택되는 방법을 취하고 있다. 이와 같은 신호 체계는 교통 상황 변화에 적절하게 대응하기 어려운 문제점을 갖고 있다. 특히, 혼잡 상황과 같은 문제들은 이진 논리로써 해결하기 어렵다. 따라서, 본 논문에서는 교통 혼잡 상황에 신속하게 대처할 수 있는 신호기 제어 시스템을 제안하였다. 본 논문에서 제안한 제어기는 확실성 및 퍼지환경에서 작동한다. 따라서, 도로의 혼잡 상황을 퍼지 논리를 사용하여 표현하고 퍼지 추론기에 의해 신호 시간을 결정하도록 하였다. 본 논문에서 제안한 신호기 제어 시스템의 타당성을 검증하고자 페트리네트를 이용하여 모델링 하였다.

The Traffic Signal Control System Applying Fuzzy Reasoning

Mi-Kyoung Kim[†] · Yun-Bae Lee^{††}

ABSTRACT

The current traffic signal control systems are operated depending on the pre-planned control scheme or the selected control scheme according to a period of time. The problem with these types of traffic control systems is that they can not cope with variant traffic flows appropriately. Such a problem can be difficult to solve by using binary logic. Therefore, in this paper, we propose a traffic signal control system which can deal with various traffic flows quickly and effectively. The proposed controller is operated under uncertainty and in a fuzzy environment. It shows the congestion of road traffic by using fuzzy logic, and it determines the length of green signal by means of a fuzzy inference engine. It modeled using petri-net to verify its validation.

1. 서 론

오늘날 도로는 도로에 비해 차량이 급속도로 증가함에 따라 심각한 교통난을 야기 하고 있다. 이와 같은 도로 교통 문제를 해결하기 위한 많은 연구가 진행되어 왔는데, 그 대표적인 분야가 신호 제어 시스템이다. 그러나, 대부분의 신호체계는 오늘날과 같이 다양한 교통류에 대처하는데는 많은 문제점을 가지고 있다.

그런데 현행 교통신호 체계는 다음과 같이 4가지 방식으로 분류할 수 있다[9].

- 첫째, 수동제어(Manual mode)방식
- 둘째, 시간대별 제어(Time of day)방식
- 셋째, 교통대응제어(Traffic Response mode)방식
- 넷째, 감응식 제어(Acurated control)방식

현행 교통 신호 제어 시스템은 사전에 계획된 신호 시간 체계에 의해 교통 상황에 따라 자동 선택되거나 시간대별로 선택되는 방법을 취하고 있다. 이와 같은 신호체계는 교통상황 변화에 적절하게 대응하기 어렵다는 문제점을 갖고 있어 교통 혼잡 상황에 빠르게 대

* 본 연구는 조선대학교 '97교비지원에 의해 수행되었음.

† 준 회 원 : 춘천기술대학 정보기술과

†† 정 회 원 : 조선대학교 전자계산학과 교수

논문접수 : 1998년 9월 14일, 심사완료 : 1999년 2월 5일

처할 수 있는 신호기 제어 시스템이 요구된다. 이와 같은 필요에 따라 본 논문에서는 교통량에 적절히 대처할 수 있는 교통 신호 제어 시스템을 제안하고자 한다. 그러나, 혼잡상황과 같은 문제들은 이진 논리로써 해결하기 어렵기 때문에 퍼지 논리를 사용하여 표현하고, 퍼지 추론기에 의해 신호시간을 결정하는 신호기 제어 시스템을 설계한다. 이 시스템은 직좌 동시 신호를 원칙으로 한다. 그러나 진행차선의 혼잡으로 녹색 신호시에도 진행이 불가능한 경우가 발생할 수 있다. 이 경우에는 다른 차선에 신호를 양보하도록 한다. 본 논문에서 제안한 교통 신호 제어 시스템은 페트리네트를 사용하여 모델링함으로써 타당성을 검증한다.

2. 퍼지추론

2.1 퍼지 집합의 표현

퍼지집합은 언어의 의미와 개념의 정의에서 보여지는 애매성을 정량적으로 나타내기 위한 집합 개념이다. 애매한 개념을 다루기 위해 의미 정도에 따라 0에서 1까지의 등급으로 나타낸다. 이와 같은 등급을 퍼지집합의 소속함수(S)라 하며, 예를들어, 다음과 같이 표현한다[6].

$$S(x; \alpha, \beta, \gamma) = \begin{cases} 0 & x \leq \alpha \\ 2\left(\frac{x-\alpha}{\gamma-\alpha}\right)^2 & \alpha \leq x \leq \beta \\ 1-2\left(\frac{x-\gamma}{\gamma-\alpha}\right)^2 & \beta \leq x \leq \gamma \\ 1 & x \geq \gamma \end{cases} \quad (2.1)$$

$$\begin{aligned} \Pi(x; \beta, \gamma) = & \\ & \left(\begin{array}{ll} S(x; \gamma - \beta, \gamma - \beta/2, \gamma) & x \leq \gamma \\ 1 - S(x; \gamma, \gamma + \beta/2, \gamma + \beta) & x > \gamma \end{array} \right) \end{aligned} \quad (2.2)$$

여기서, α, β, γ : 소속 함수를 조정할 수 있는 파라미터
 S 함수: $x \leq \alpha$ 에서는 $S = 0$
 $x \geq \gamma$ 에서는 $S = 1$
 또한, α 와 γ 사이에서는 x 의 이차함수가 되며, 파라미터 β 는 변화 대응된다.
 Π 함수: 함수 S를 이용하여 정의하고, 파라미터 β 는 변곡점간의 폭을 의미한다.
 또한, $x = \gamma \pm \beta$ 일 때 $\Pi = 0$ 이다.

2.2 퍼지 근사 추론

전통적 논리에서 추론의 주된 도구는 긍정식(modus ponens)과 같은 동어반복(tautology)이다. 즉 $(A \Rightarrow (A \Rightarrow B)) \Rightarrow B$ 또는 A와 B는 서술문 또는 명백하게 정의된 명제이고 규칙에서 정의된 B는 결론의 B와 동일하다. 다음은 퍼지 근사 추론을 나타낸 것이다.

$$\begin{aligned} &<규칙> \text{ IF } a=X \text{ THEN } b=Y \text{ (CF}=\phi r) \\ &\underline{\text{<사실> } a=X' \text{ (CF}=\phi f)} \\ &\text{<결론> } b=Y' \text{ (CF}=\phi c) \end{aligned}$$

여기서, X, Y : 퍼지변수
 $\phi r, \phi f, \phi c$: 각각 규칙, 사실, 결론에 대한 확신도

우리는 일상 생활에서 어떤 객체 a, b와 이들간의 관계, 그리고 a에 관한 정보가 주어지면 정보객체 b에 대한 연역추론을 할 수 있다. 이와 같은 추론 방법을 합성추론규칙(compositional rule of inference)이라 한다[1].

퍼지생성 규칙은 전제부 또는 결론부에서 and, or와 같은 논리연산자를 이용하여 둘 이상의 명제를 합성수 있다. 합성 명제가 전제부 또는 결론부에 있는 명제를 합성 퍼지생성규칙이라 한다. 그리고 논리연산자가 없는 규칙을 단순 퍼지생성규칙이라한다.

2.3 퍼지 추론과 퍼지 제어

플랜트 제어에 필요한 플랜트의 상태에 관한 정보를 x_1, x_2 라 하고, 플랜트로의 입력, x_2 을 y 라 하면, if(전제부)-then(결론부)형의 퍼지 제어 규칙은 전제부와 결론부로 나누어지지만 x_1, x_2, y 등이 취하는 언어 라벨이 붙은 퍼지 집합으로 표현하기 때문에 언어적 제어 규칙이라 한다.

일반적인 퍼지 제어 규칙 R은 다음과 같이 표현할 수 있다.

$$R^i : \text{ if } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ then } y \text{ is } B_i \text{ (} i = 1, 2, \dots, n \text{)}$$

여기서, i : 제어 규칙의 번호, A_i, B_i : 퍼지 변수

퍼지 제어기는 복수 개의 퍼지 제어 규칙의 집합과 퍼지 추론부로 구성된다. 제어 규칙의 x_1, x_2 는 퍼지 제어기의 입력, y 는 출력이라고 한다. 퍼지 추론부는

실제 입력이 주어졌을 때 적용 가능한 제어 규칙을 이용하여 퍼지 추론을 함으로써 출력을 계산한다. 플랜트 제어를 위한 퍼지 추론법 중 Mamdani의 Max-Min 추론법이 많이 사용되고 있다[3]. 이 추론법은 퍼지관계의 합성규칙을 따르며 제어를 위한 추론법 중에 가장 많이 이용되고 있다. 여기서 이용되는 퍼지변수는 멤버쉽 함수 (2.3)과 같은 범종형과 (2.4)와 같은 삼각형이 있다.

$$A(x) = e^{-\frac{1}{a}(x-b)^2} \quad (2.3)$$

$$A(x) = \frac{1}{a}(-|x-b+a| \vee 0, (a > 0)) \quad (2.4)$$

퍼지제어규칙이 n개 있고 or로 결합되어 있다고 할 때, 다음과 같이 나타낸다.

$$\begin{aligned} R^1: & \text{if } x_1 \text{ is } A_{11} \text{ and } x_2 \text{ is } A_{12} \text{ then } y \text{ is } B_1 \\ & \text{or} \\ R^2: & \text{if } x_1 \text{ is } A_{21} \text{ and } x_2 \text{ is } A_{22} \text{ then } y \text{ is } B_2 \\ & \text{or} \\ & \vdots \\ R^i: & \text{if } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ then } y \text{ is } B_i \\ & \vdots \\ R^n: & \text{if } x_1 \text{ is } A_{n1} \text{ and } x_2 \text{ is } A_{n2} \text{ then } y \text{ is } B_n \end{aligned}$$

이때 i번째의 제어규칙은 x_1, x_2, y 의 전체영역(universal set)을 각각 X_1, X_2, Y 로 하면 $X_1 \times X_2 \times Y$ 상의 퍼지관계 R_i 에 의해

$$(x_1, x_2, y) \text{ is } R_i \quad (2.5)$$

또,

$$R_i = (A_{i1} \times A_{i2}) \times B_i \quad (2.6)$$

로 표현할 수 있다.

그리고, n개의 제어규칙의 전체 R 은 다음과 같이 표현할 수 있다.

$$R = R_1 \cup R_2 \cup \dots \cup R_n = \bigcup_{i=1}^n R_i \quad (2.7)$$

전제부 변수 x_1, x_2 등은 확정 수치이기 때문에 그 값을 x_1^0, x_2^0 라 하면, 다음과 같이 나타낼 수 있다.

$$R(x_1^0, x_2^0, y) = R_1(x_1^0, x_2^0, y) \vee R_2(x_1^0, x_2^0, y) \vee \dots \vee R_n(x_1^0, x_2^0, y) \quad (2.8)$$

$$R(x_1^0, x_2^0, y) = A_{i1}(x_1^0) \wedge A_{i2}(x_2^0) \wedge B_i(y) \quad (2.9)$$

$$\begin{aligned} B^0(y) &= [\omega_1 \wedge B_1(y)] \vee [\omega_2 \wedge B_2(y)] \vee \dots \vee [\omega_n \wedge B_n(y)] \\ &= \bigvee_{i=1}^n [\omega_i \wedge B_i(y)] \end{aligned} \quad (2.10)$$

단, $\omega_i = A_{i1}(x_1^0) \wedge A_{i2}(x_2^0)$: ω_i 는 i번째 규칙의 (전체의 입력에 대한) 적합도

i번째 규칙의 전제부: $x_1 = A_{i1}, x_2 = A_{i2}$ 이것은 규칙을 적용하기 위한 조건, 즉 규칙이 적용되는 변수 x_1, x_2 의 범위를 의미한다. 이것에 대한 현재 상태는 다음과 같다.

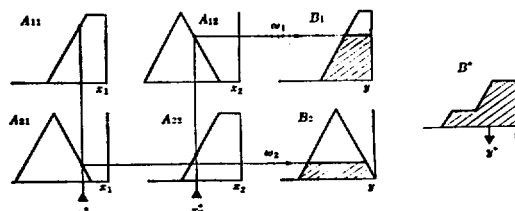
$$x_1 = x_1^0, x_2 = x_2^0 \quad (2.11)$$

퍼지추론은 B^0 를 구하면 종료되지만 제어에서는 퍼지집합 B^0 를 플랜트의 조작량으로 출력해도 플랜트를 움직일 수 없다. 만약 퍼지제어기와 플랜트 사이에 '조작량을 크게 하시오' 라는 것과 같은 애매한 지시를 내려도 인간은 그 의미를 이해하고 어떤 수치로 변환하여 조작량을 결정할 수 있다. 이것은 추론결과의 해석이라고 할 수 있는데 Max-Min 추론법에서는 B^0 를 다음과 같이 해석하여 하나의 수치 y^0 로 변환한다.

$$y^0 = \frac{\int B^0(y) y \, dy}{\int B^0(y) \, dy} \quad (2.12)$$

식 (2.12)는 퍼지집합 B의 멤버쉽 함수 B(y)의 중심 좌표 y^0 를 부여한 것이다. 본 논문에서는 Max-Min 추론법을 이용하여 퍼지집합 B의 해석, B의 비퍼지화 방법으로 무게중심법을 사용하였다.

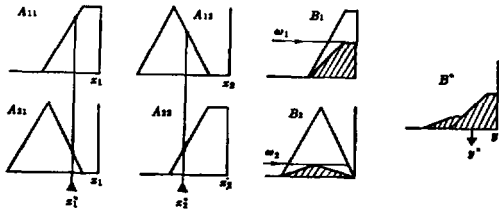
이상과 같은 추론 프로세서를 도해하면 (그림 1)과 같다.



ω_i : 각 규칙의 적합도, B_i^* : 각 규칙의 추론결과
 B^0 : 추론 결과, y^0 : 무게중심으로 해서 규칙전체의 추론 결과

(그림 1) 무게중심 비퍼지화 방법의 Max-Min 추론(a)
 (Fig. 1) The Max-Min inference of defuzzification method for the center of weight(a)

퍼지제어는 추론의 전체가 퍼지가 아니고 확정된 수치로써 플랜트 오퍼레이터가 감각으로 인식한 "이상한 음이난다", "진동이 크다" 등의 감각량을 사용하는 경우라든가, 계측 오차 등을 위해 측정치를 퍼지를 이용하여 해석한 쪽의 제어기 성능이 좋을 경우에는 퍼지로서 전체가 주어진다. 이 경우에는 규칙의 전체부를 간단히 하기 위해 $x=A$ 라고 하면 전체는 $x=A^0$ 이고 이 때의 적합도 ω 는 $A(x)\wedge A^0(x)$ 의 최대 등급인 $\omega = \max [A(x)\wedge A^0(x)]$ 로 하면 된다. 이 추론법은 적합도 ω_i 를 구하는 식과 B^0 를 구하는 식으로 min 연산을 이용하는 데, 이들의 min 연산은 퍼지의 직곱을 정의하는 연산이 기본이다. 그런데, 여기서 적합도의 값이나 중심의 값이 유연하게 변화하지 않을 경우도 있고 min 대신에 보통의 곱셈을 이용하는 경우도 있는데, 이때 적합도는 $\omega_i = A_{i1}(x_1^0)A_{i2}(x_2^0)$ 로 나타낼 수 있으며, 또 추론 결과는 $B^0(y) = \bigvee_{i=1}^n \omega_i B_i(y)$ 가 된다. 추론 프로세스는 (그림 2)와 같다. (그림 2)에서 $\omega_i, B_i(y)$ 로 나타내는 퍼지집합은 B_i 에 대하여 최대 등급이 1인 곳이 ω_i 가 되도록 압축된다.



(그림 2) 무게중심 비퍼지화 방법의 Max-Min 추론(b)
(Fig. 2) The Max-Min inference of defuzzification method for the center of weight(b)

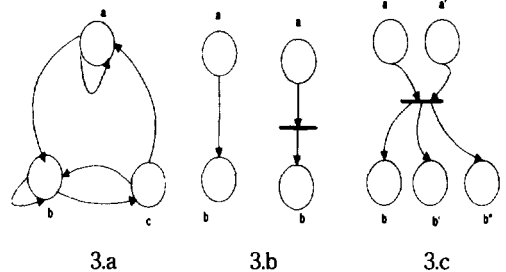
3. 페트리네트(Petri-net)

3.1 페트리네트의 정의

페트리네트는 상호작용(interaction)과 병렬성(parallelism), 동시성(concurrency) 등의 구성 요소를 갖춘 시스템으로 추상적이며 정형화된 모델이다. 또한 동시에 발생하는 사건을 포함하는 시스템의 모델링, 검증, 분석에 효과적으로 사용되고 있다. 페트리네트의 장점을 요약하면 다음과 같다.

1. 표현 구조가 그래픽 형태로 되어 있으므로 이해가 용이하고 사용이 간결함
2. 동시적, 비동기적인 시스템을 표현할 수 있는 강력한 모델링 및 분석 도구
3. 자동화가 가능

트리네트는 순차적 오토마타의 확장으로 볼 수 있다. (그림 3)은 3-상태 오토마타를 기술한 것이다. 여기에서 한 호를 선택하여 (그림 3.b)와 같이 상태 a로부터 상태 b로 전이한다고 하자. 이와 같은 전이는 막대를 침가하여 그림 3.b의 오른쪽과 같이 상태 a로부터 상태 b로 전이하는 사건을 표현할 수 있다. 이와 같은 표현은 각 전이가 'place'라고 하는 오직 하나의 입력 상태를 갖고, 역시 'place'라 불리는 하나의 출력을 가짐으로써 각각의 호가 '상태 기계'라 불리는 페트리네트의 가장 낮은 단계를 표현할 수 있다. place와 상태의 차이는 상태 기계에서는 명백하지 않지만, 상태 기계의 확장인 페트리네트를 고려했을 때 명확하게 설명될 수 있다. (그림 3.c)에 나타낸 바와 같이 모든 전이에서, 다중 입력과 다중 출력 place가 있을 수 있다.



(그림 3) 순차적 오토마타로부터 페트리네트로의 전환
(Fig. 3) The conversion from sequential automata to petri-net

페트리네트를 정의하면 다음과 같다.

[정의 1] 페트리네트는 네 개 톱플로 이루어진다.

$$PN = \{P, T, \alpha, \beta\}$$

P : p1, p2, p3, ..., pn ; place들의 집합이며 원으로 표현된다.

T : t1, t2, t3, ..., tn ; (T ≠ 0, and P ∩ T = 0)인 트랜지션(transition)들의 집합으로 막대로 표현된다.

α : P × T → N, 트랜지션(transition)의 입력 함수(N = 정수의 집합).

β : T × P → N, 트랜지션(transition)의 출력 함수

[정의 2] 주어진 페트리네트에서 $t \in T$ 일 때, 다음과 같은 식이 성립한다.

$$Pre(t) = \{p \in P / \alpha(p, t) \neq 0\} \quad (Pre(t) : \text{트랜지션 } t \text{의 입력 플레이스의 집합})$$

$$Post(t) = \{p \in P / \beta(t, p) \neq 0\} \quad (Post(t) : \text{트랜지션 } t \text{의 출력 플레이스의 집합})$$

place에는 토큰(token)이라는 마킹(marking)을 위한 한 개의 정수를 갖는다.

[정의 3] 페트리네트의 마킹 M은 place P로부터 음이 아닌 정수의 집합인 N의 집합으로 사상(mapping)된다. $|P|=m$ 에서, 표시된 M은 열벡터 $M \in N^m$ 으로 표현된다.

(단, i번째 요소는 $M(p_i)$ 로 표현된다.)

(그림 4)에 대한 페트리네트 구조는 다음과 같다.

$$PN = \{P, T, \alpha, \beta\}$$

$$P = \{p_1, p_2, p_3, p_4, p_5\}$$

$$T = \{t_1, t_2, t_3\}$$

$$\alpha(p_1, t_1) = 1 \quad \beta(t_1, p_2) = 1$$

$$\alpha(p_5, t_2) = 1 \quad \beta(t_1, p_3) = 1$$

$$\alpha(p_2, t_3) = 1 \quad \beta(t_2, p_1) = 1$$

$$\alpha(p_4, t_3) = 1 \quad \beta(t_2, p_4) = 1$$

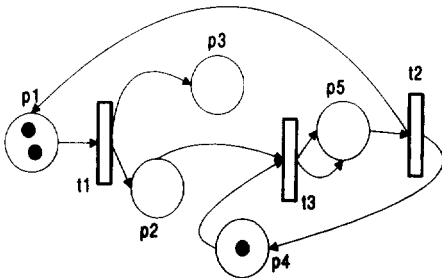
$$\beta(t_3, p_5) = 2$$

$$Pre(t_1) = \{p_1\} \quad Post(t_1) = \{p_2, p_3\}$$

$$Pre(t_2) = \{p_5\} \quad Post(t_2) = \{p_1, p_4\}$$

$$Pre(t_3) = \{p_2, p_4\} \quad Post(t_3) = \{p_5, p_5\}$$

$$M = (2, 0, 0, 1, 0)$$



(그림 4) 페트리네트 그래프
(Fig. 4) The diagram of petri-net

[정의 4] 페트리네트의 트랜지션 t는 마킹 M에 대하여 $\forall p \in Pre(t), M(p) \geq \alpha(p, t)$ 이 성립할 때 실행가능하다. 여기서, 트랜지션을 실행가능할 때 점화 가능하다고 한다.

고 한다.

[정의 5] 트랜지션 t_i 의 점화에 따른 마킹 M의 M'으로의 전환은 다음과 같이 정의된다.

$$\forall p \in P, M'(p) = M(p) + \beta(t_i, p) - \alpha(p, t_i)$$

(그림 4)에서 전이 t1의 firing은 새로운 표시 M'에 다음과 같이 주어진다 :

$$M'(p_1) = M(p_1) + \beta(t_1, p_1) - \alpha(p_1, t_1) = 2 + 0 - 1 = 1$$

$$M'(p_2) = M(p_2) + \beta(t_1, p_2) - \alpha(p_2, t_1) = 0 + 1 - 0 = 1$$

$$M'(p_3) = M(p_3) + \beta(t_1, p_3) - \alpha(p_3, t_1) = 0 + 1 - 0 = 1$$

[정의 6] 페트리네트에서 다음 조건이 성립되면, 'self-loop free(또는 loop free)'라고 한다.

$$\forall p \in P, Pre(t) \cap Post(t) = \emptyset \quad \text{이고} \quad \forall t \in T, Pre(t) \cap Post(t) = \emptyset$$

여기서, $C = \beta - \alpha = \beta(t, p) - \alpha(p, t)$ 라고 할 때, $M' = M + C$ 가 성립한다.

[정의 7] σ 를 전이 $t_1 t_2 \dots t_k$ 의 finite sequence(유한 연속)이라 하면, σ 는 M_i 로부터 출발한 정당한 점화 순서이다. 이것은 다음과 같이 표현된다.

$$M_1 \xrightarrow{t_1} M_2 \dots \xrightarrow{t_k} M_{k+1}$$

마킹 M_1 은 σ 의 점화에 의해 M_{k+1} 에 도달된다.

초기마킹 M_0 부터 진화하며 페트리네트는 n까지의 유계성(boundness)을 갖는다.

$$M \in M_0, \forall p \in P, M(p) \leq n$$

페트리네트에는 초기 마킹 M_0 가 무엇이든지 간에, 어떤 전이 t에 대해 다음과 같은 전이를 하는 정당한 점화 순서가 존재할 때 이를 deadlock-free라고 한다.

$$\forall M \in M_0 (\forall t \in T, \exists \sigma \{M \xrightarrow{\sigma} M' \wedge M' \xrightarrow{t} M''\})$$

[정의 8] Place-coloured 페트리네트는 다음과 같이 구성된다

1. 정점이 플레이스와 트랜지션으로 이루어진 양분된 그래프
2. 호는 색깔을 갖는 토큰의 기능표현을 한다.
3. 초기 마킹 M_0 는 모든 플레이스 p와 초기 색깔 값의 bag을 연합한다.

[정의 9] Predicate-Transition 페트리네트는 다음으로 구성된다 :

1. Place-coloured net
2. 각 트랜지션에 연합된 Predicate
: 여기서 입력 토큰과 점화될 트랜지션 사이의 관계를 형성한다.

[정의 10] Predicate-Action 페트리네트는 다음과 같이 구성된다.

1. 안전한 페트리네트 : 호의 가중치가 오직 하나일 때 안전하다.
2. 트랜지션에 부착된 라벨의 집합
: 각 트랜지션 t 는 *when* $p_i(x)$ *do* $x' \leftarrow F_i(x)$ 의 부착된 형식 표현을 갖는다.
(Pt : predicate, Ft:action, 둘 다 프로그램 변수의 벡터 x 에 존재)

[정의 11] Timed 페트리네트

한정된 시간을 필요로 하는 시스템은 그 실행 시간을 표현할 수 있어야 한다. 그 방법으로 두 가지가 이용된다. 첫째, Timed 페트리네트는 각 전이에 값을 부여함으로써 정의된다. 이것은 트랜지션의 점화 시간을 나타낸다. 둘째, Time(Out) 페트리네트는 각 전이에 부착되는 한 쌍의 (t_{min}, t_{max}) 에 의해 정의된다. 이것은 트랜지션이 점화되어야 하는 시간 간격을 의미한다.

4. 교통 신호 제어 시스템

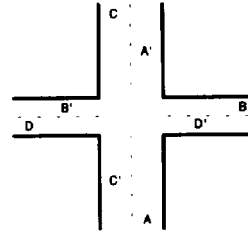
4.1 교통 신호 제어 시스템의 설계

시스템 설계를 위한 전체 조건은 다음과 같다.

1. 각 도로의 검지선에 의해 도로 정체 정도를 파악할 수 있다.
2. 녹색 신호는 4거리에서 직좌 동시신호를 순환한다
3. 녹색 신호시에 진행 불가능한 경우는 진행불가 방향에 따라 다른 도로에 신호를 양보한다.
4. 황색 시간은 교차로의 폭, 차량의 길이 등의 요소에 의해 결정된다.

상술한 전체조건과 같은 도로 상황에서 교통 신호 제어 시스템의 동작은 다음과 같다.

step 1. 도로 상황 입력



(그림 5) 도로의 상황도
(Fig. 5) The diagram of road situations

이 단계에서는 신호 시간 배정을 위한 도로의 정체 정도를 파악하기 위해 각 도로의 상황이 입력된다. 도로의 정체도는 차량 정체 길이가 길수록, 같은 정체 길이라 하더라도 적색 신호 경과 시간이 짧을수록 정체 정도는 심각한 것으로 간주한다. 다음, 어떤 도로의 정체도가 심각하여 긴 녹색 신호시간을 할당하여도 진행 방향의 도로 정체가 심하다면 진행할 수 없을 뿐만 아니라 교차로 중간에 정지하게 되므로 다음 녹색 신호를 받는 도로의 진행에 방해가 될 수 있다. 이것은 진행 방향의 도로 상황 또는 녹색 신호 시간을 결정하는데 중요한 요소가 될 수 있다. 따라서 다음과 같은 요소를 퍼지 신호 제어 시스템의 입력 파라미터로 결정한다.

- 각 도로의 정체길이 $road[1:4]$: 각 차도에 대기중인 차량의 길이를 나타낸다.
- 각 도로의 지체시간 $wait[1:4]$: 각 차도의 적색 신호 경과 시간을 나타낸다.
- 상대도로의 정체길이 $part[1:4]$: 녹색 신호를 받을 예정인 차량의 진행 차선의 정체도를 나타낸다.

상기의 입력 요소는 신호의 '녹 -> 황' 전이와 동시에 이루어진다.

step 2. 입력값의 퍼지화

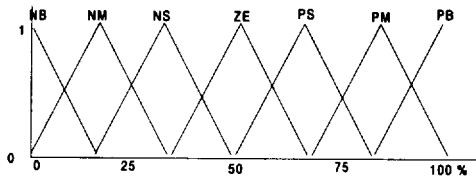
이 단계에서는 입력된 값으로부터 도로의 정체 정도를 추론하고 보다 유연하고 적절하게 녹색 신호시간을 결정하기 위해 입력된 값을 퍼지 값으로 변환한다.

(그림 6, 7)과 같은 소속함수 그래프에 의해 이루어진다. 퍼지값은 도로 상황에 따라 조정될 수 있다.

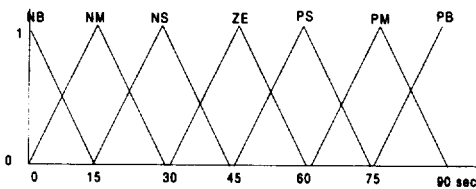
여기서, (그림 6)과 (그림 7)의 퍼지집합 라벨은 다음과 정의한다.

- NB : Negative Big NM : Negative Medium
- NS : Negative Small ZE : Zero

PS : Positive Small PM : Positive Medium
PB : Positive Big



(그림 6) 차량 정체도에 대한 퍼지 집합의 소속함수
(Fig. 6) The membership function of fuzzy sets for the congestion of vehicles



(그림 7) 도로의 지체시간에 대한 퍼지 집합의 소속함수
(Fig. 7) The membership function of fuzzy sets for the delay time of road

step 3. 각 도로의 복잡도에 대한 퍼지 추론이 단계에서는 퍼지 추론에 의해 도로의 정체도를 계산한다. 도로의 정체도는 상술한 바와 같이 각 도로(차량)의 정체길이와 각 도로의 지체시간(적색 신호 경과시간)에 의해 결정된다. 따라서 도로의 정체도를 추론하는 규칙은 다음과 같이 나타낼 수 있다.

- R_1 : if R is NB and W is NB then C is ZE
- R_2 : if R is NB and W is NM then C is NS
- ...
- R_n : if R is PB and W is PB then C is ZE

여기서, R: 정체길이, W 적색 신호 경과 시간, C: 도로의 정체도

이와 같은 규칙을 도표화하면 <표 1>과 같다.

<표 1> 도로의 정체도 추론 규칙
(Table 1) The inference rules for congestion of road

지체시간(W) 정체길이(R)	NB	NM	NS	ZE	PS	PM	PB
NB	ZE	NS	NM	NB	NB	NB	NB
NM	PS	ZE	NS	NM	NB	NB	NB
NS	PM	PS	ZE	NS	NM	NB	NB
ZE	PB	PM	PS	ZE	NS	NM	NB
PS	PB	PB	PM	PS	ZE	NS	NM
PM	PB	PB	PB	PM	PS	ZE	NS
PB	PB	PB	PB	PB	PM	PS	ZE

상기 규칙에 의한 도로의 정체도 계산은 다음과 같이 할 수 있다.

$$\text{정체도} = \bigvee_{i=1}^n \text{정체길이}(\text{road}(i)) \wedge \text{대기시간}(\text{wait}(i)) \quad (4.1)$$

정체도는 (그림 1)과 (그림 2)에서 제시한 바와 같이 정체길이와 대기시간의 퍼지곱(Min 연산) 연산을 하여 최대등급인 $\max[\text{정체길이} \wedge \text{대기시간}]$ 로 계산하는 Max-Min 추론의 방법을 이용하는 것이 가장 적합 하다.

step 4. 진행방향 도로의 정체도에 따른 녹색신호 시간 결정

step 3의 결과에 따라 녹색시간을 결정하는 일은 경우에 따라 비효율적인 결과를 가져올 수 있다. 따라서, 최종적인 녹색신호 시간은 정체도뿐만 아니라 진행방향(상대도로)의 정체도를 고려하여 결정하여야 한다. 따라서, 녹색 신호시간은 직진과 좌회전 신호 각각을 계산하여야 하며 그 결과는 다른 값으로 나타날 수 있다. 녹색신호시간을 결정하는 퍼지 추론 규칙은 다음과 같다.

- R_1 : if C is NB and FC is NB then G is NB
 - R_2 : if C is ZE and FC is PM then G is NM
 - ...
 - R_n : if C is PB and FC is PM then G is NM
- C: 정체도, FC: 상대도로의 정체도, G: 녹색신호시간

이와 같은 규칙을 도표화하면 <표 2>와 같다.

<표 2> 녹색 신호 시간 결정 추론 규칙
(Table 2) The inference rules for decision of green signal time

정체도(C)	NB	NM	NS	ZE	PS	PM	PB
상대도로 정체(FC)	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NM	NS	ZE	PS	PM	PB
NM	NB	NM	NS	ZE	PS	PM	PM
NS	NB	NM	NS	ZE	PS	PS	PS
ZE	NB	NM	NS	ZE	ZE	ZE	ZE
PS	NB	NM	NS	NS	NS	NS	NS
PM	NB	NM	NM	NM	NM	NM	NM
PB	NB	NB	NB	NB	NB	NB	NB

상기 규칙에 의한 녹색 신호 시간 계산은 다음과 같이 할 수 있다.

녹색시간길이 =

$$\bigvee_{i=1}^n \text{정체도}(\text{congest}(i)) \wedge \text{상대도로정체도}(\text{part}(j)) \quad (4.2)$$

step 5. 결정된 녹색신호시간에 대한 비퍼지화

계산된 결과를 신호기에 적용하기 위해서, 무게중심법에 따라 비퍼지화한다. 비퍼지화식은 다음과 같다.

$$\frac{\sum_{i=1}^n \text{area}(i) \times \text{distance}(i)}{\sum_{i=1}^n \text{area}(i)} \quad (4.3)$$

n : 녹색신호 최대시간, $\text{area}(i)$: i번째 구간의 면적
 $\text{distance}(i)$: i번째 구간까지의 거리

step 6. 녹색신호시간 적용

이 단계는 step 5의 결과를 신호기에 적용하는 단계이다. 이때 서로 다른 직좌 시간은 조정되어야 한다. 직좌 신호시간의 차이가 많은 경우에 다른 도로에서 신호를 활용하는 것은 효율적이지만 한편으로 그것이 아주 짧은 시간인 경우에는 오히려 혼란의 원인이 될 수 있기 때문이다. 따라서, 직좌신호는 다음과 같이 적용한다. 현재 녹색 신호를 받는 도로가 A이고 황색 신호 시간이 3이라면 최소 녹색 시간 5를 보장하기 위해서는 8초 이상 차이가 있을 때에만 도로 C의 좌회전 또는 직진을 허용해야 한다. 그렇지 않을 경우에는 직좌 중 더 큰 값을 직좌 동시 적용한다.

이와 같은 시스템을 실제 도로에 적용한 결과를 사례 연구를 통해 보기로 하겠다.

먼저, 현재의 도로 상황은 <표 3>과 같다고 가정한다.

<표 3> 도로 상황 예
 <Table 3> The example of road situations

도로	입력 파라미터	차량 정체도(%)	적신호대기(sec)	진행차도의 정체도(%)
도로 A		10	2	도로 A' 50
도로 B		50	10	도로 B' 20
도로 C		62	40	도로 C' 50
도로 D		88	60	도로 D' 10

이 값을 퍼지화 하여 각 도로의 정체도를 추론한 결과는 <표 4>와 같다.

<표 4> 도로의 상황에서 도로의 정체도 추론 결과
 <Table 4> The inference results of congestion of road in the road situations

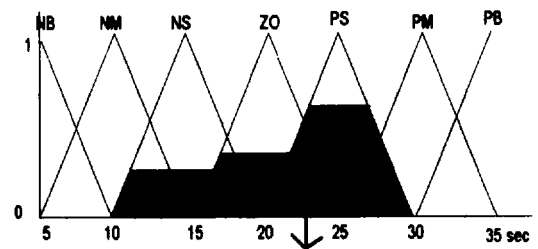
	NB	NM	NS	ZE	PS	PM	PB
도로 A	0	0	0.1	0.4	0.6	0	0
도로 B	0	0	0	0	0	0.7	0.3
도로 C	0	0	0	0.3	0.4	0.6	0
도로 D	0	0	0	0	0.6	0.4	0

<표 5>는 각 도로의 녹색신호 할당 시간을 추론한 결과이다.

<표 5> 도로 상황 및 정체도에서 녹색신호시간 추론 결과
 <Table 5> The inference results of green signal time in the situations and congestion of road

직진	NB	NM	NS	ZE	PS	PM	PB
도로 A	0	0	0.1	0.6	0	0	0
도로 B	0	0	0	0	0.3	0.7	0
도로 C	0	0	0	0.6	0	0	0
도로 D	0	0	0	0	0.6	0.4	0
좌회전	NB	NM	NS	ZE	PS	PM	PB
도로 A	0	0	0.1	0.4	0.6	0	0
도로 B	0	0	0	0.7	0	0	0
도로 C	0	0	0.3	0.4	0.6	0	0
도로 D	0	0	0	0.6	0	0	0

다음은 녹색 신호 시간 추론 결과를 비퍼지화 해야 한다. 식(4.3)의 무게중심법에 의해 <표 5>의 결과가 산출되어 도로 C의 좌회전 신호 시간은 (그림 8)과 같이 구할 수 있다. 이 경우 각 도로의 직진 및 좌회전 신호 시간은 <표 6>과 같다.



(그림 8) 도로 C의 비퍼지화
 (Fig. 8) The defuzzification of road C

〈표 6〉 비퍼지화된 신호 시간
 〈Table 6〉 The signal time for defuzzication

	직진신호시간	좌신호시간
도로 A	19	22
도로 B	29	20
도로 C	20	23
도로 D	28	20

따라서, 도로 D에 녹색 신호를 보내야 한다면 신호 시간은 다음과 같이 결정된다.

- t = x 일 때 도로 D의 직좌 녹색 등화
- t = x+20 일 때 도로 D의 좌회전 황색 등화
- t = x+20+3 일 때 도로 B의 직진 녹색 등화
- 도로 D의 좌회전 적색 등화
- t = x+28 일 때 도로 B의 직진 황색 등화
- 도로 D의 직진 황색 등화

4.2 교통 신호 제어 시스템을 위한 페트리네트 모델링

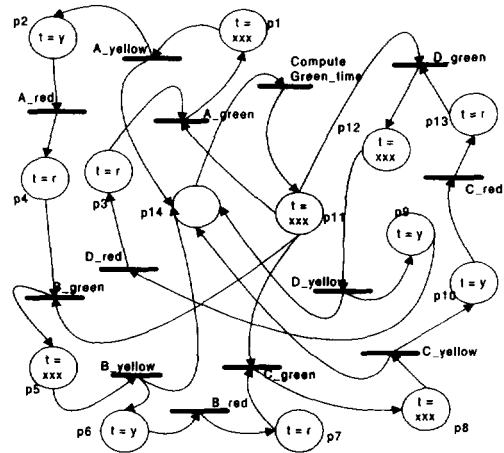
본 절에서는 4.1절에서 제안한 교통 신호 제어 시스템을 페트리네트를 이용하여 모델링하고 성능을 분석한다. 제안된 시스템은 기존 고정된 신호주기 제어 시스템의 단점을 최소화하여 도로 상황에 유연하게 대처하도록 하기 위한 것이다.

이 시스템은 페트리네트 기반의 시뮬레이션 언어인 SimNet을 이용하여 실험한다. 모델링 결과는 (그림 9), (그림 10)에 도시한 바와 같다. 신호기 제어 시스템의 페트리네트 모델은 트래픽 흐름(traffic flow)과 트래픽 제어 논리(traffic control logic)로 구성된다.

(그림 9)는 트래픽 흐름에 대한 모델이다. 여기에서 표현한 플레이스 및 트랜지션의 의미는 다음과 같다.

- | | |
|---------------------|---------------------|
| 1) place label | 2) transition label |
| P1: 도로 A의 녹색 신호 상태 | t1: 도로 A의 황색 신호 점등 |
| P2: 도로 A의 황색 신호 상태 | t2: 도로 A의 적색 신호 점등 |
| P3: 도로 D의 적색 신호 상태 | t3: 도로 A의 녹색 신호 점등 |
| P4: 도로 A의 적색 신호 상태 | t4: 도로 B의 녹색 신호 점등 |
| P5: 도로 B의 녹색 신호 상태 | t5: 도로 B의 황색 신호 점등 |
| P6: 도로 B의 황색 신호 상태 | t6: 도로 B의 적색 신호 점등 |
| P7: 도로 B의 적색 신호 상태 | t7: 도로 C의 녹색 신호 점등 |
| P8: 도로 C의 녹색 신호 상태 | t8: 도로 C의 황색 신호 점등 |
| P9: 도로 D의 황색 신호 상태 | t9: 도로 C의 적색 신호 점등 |
| P10: 도로 C의 황색 신호 상태 | t10: 도로 D의 황색 신호 점등 |

- P11: 각 도로의 녹색 신호 시간 t11: 도로 D의 녹색 신호 점등 결정 상태
- P12: 도로 D의 녹색 신호 상태 t12: 도로 D의 적색 신호 점등
- P13: 도로 C의 적색 신호 상태 t13: 각 도로의 녹색 신호 할당 시간 계산
- P14: 각 도로의 녹색 신호 시간 종료 상태



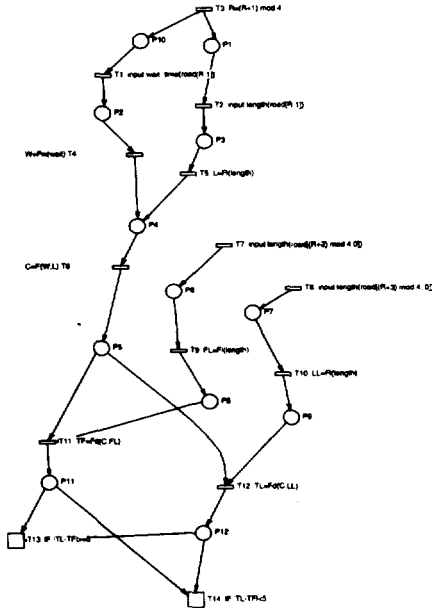
(그림 9) 교통 신호 제어 시스템의 페트리네트 모델링
 (Fig. 9) The Petri-net modeling of traffic signal control system

녹색 신호 시간이 결정되어 그 시간동안 상태를 유지한 후 황색 신호를 유지했다가 다른 도로들이 녹색 신호를 유지하는 동안 적색 신호를 준다. P1은 도로 A의 녹색 신호 상태이고 할당받은 신호 시간이 끝나게 되면 황색등을 점등(t1)하여 P2 상태를 유지했다가 적색등을 점등(t2)한다. P4 상태 후 B도로의 녹색 신호 시간을 P11로부터 할당받아 B의 녹색 신호를 점등(t4)한다. 그리고 녹색 신호 시간동안 상태를 유지하게 된다.

(그림 10)은 퍼지추론에 의한 신호기 제어논리를 모델링한 것으로 녹색 신호시간 결정 과정을 모델링한 것이다. 여기에서 나오고 있는 각 트랜지션의 의미는 다음과 같다.

- T1: 각 도로의 대기 시간
- T2: 각 도로의 정체 길이
- T7: 좌회전 방향의 상대 도로 정체 길이
- T8: 직진 방향의 상대 도로 정체 길이
- T4: T1의 퍼지화
- T5: T2의 퍼지화
- T6: T4, T5에 의한 도로의 정체도 계산
- T9: T7의 퍼지화
- T10: T8의 퍼지화

- T11 : 녹색 시간 추론(각 도로의 정체도와 좌회전 진행 방향의 정체도에 의한)
- T12 : 녹색 시간 추론(각 도로의 정체도와 직진 방향의 정체도에 의한)
- T13, T14 : 녹색 시간 적용 결정



(그림 10) 퍼지 추론 시스템의 모델링
(Fig. 10) The modeling of fuzzy inference system

5. 결 론

본 논문에서는 퍼지 추론을 적용한 신호기 제어 시스템을 구성하고 페트리네트를 이용하여 모델링하였다. 페트리네트는 제어의 흐름을 보이기 위해 토큰(token)을 사용하는 수학적 모델링 도구로서 hardware 뿐만아니라 Parallel, Asynchronous System들의 특징을 갖는 정보 처리 시스템을 표현하고 연구하는데 유용한 도구이다. 따라서, 본 논문에서 제안한 시스템의 타당성을 검증하는데 이용하였다. 시스템에 대한 제약 조건은 다음과 같다.

1. 각 도로의 검지선에 의해 도로 정체 정도 파악.
2. 녹색 신호는 네 거리에서 직좌 동시 신호 순환.
3. 황색 시간은 교차로의 폭, 차량의 길이 등의 요소에 의해 결정.
4. 녹색 신호시 진행 불가한 경우, 진행 불가 방향에 따라 다른 도로에 신호 양보.

그리고, 도로 상황 입력, 입력값의 퍼지화, 각도로의 복잡도에 대한 퍼지 추론, 진행 방향도로의 정체도에 따른 녹색 신호 시간의 결정, 결정된 녹색 신호에 대한 비퍼지화, 녹색 신호 시간 적용의 6단계로 제어되는 신호기 시스템을 구성하였다. 페트리네트 전용 시뮬레이션 언어인 SimNet으로 실험한 결과 제어 시스템은 dead-lock free, liveness, boundness를 유지하였고 트래픽 처리량을 늘리고 차량의 신호 대기 시간을 단축시킬 수 있었다.

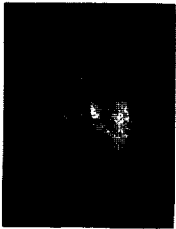
특히, 본 논문에서 제안한 신호기 제어 시스템은 신호 시간 결정 요인인 진행 방향 정체도까지 포함하였다. 그 결과, 진행 방향의 정체도가 심한 경우, 4장 4.1 절 step. 4의 규칙에 따라 녹색 신호를 결정함으로써 진행 차선의 정체로 인해 녹색 신호 동안에도 계속 정지되는 상황을 예방할 수 있었다. 그러나, 향후 혼잡한 교차로에서 보다 정교한 대기시간 계산과 정체도 파악을 위한 지능형 퍼지 교통 제어 시스템 연구가 필요하다.

참 고 문 헌

- [1] Zadeh, L.A., "Outline of New Approach to the Analysis of Complex Systems and Decision Process," IEEE Transactions on System, Man and Cybernetics, Vol.SMC-3, No.1, Jan. 1996.
- [2] R.M.Keller, "Formal Verification of parallel programs," Con. ACM19-7, Vol.19, No.7, 1996.
- [3] E.H.Mamdani, "Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis," IEEE Transactions on Computers, Vol. C-26, No.12, pp.1182-1191, December 1996.
- [4] J.L.Peterson, "Petri Nets", ACM Computing surveys, Vol.9, No.3, pp.223-252, Sep. 1995.
- [5] H.J.Genrich, 'System Modelling with high level Petrinets,' Theoretical compute science, 13, North-Holland, 1991.
- [6] Giarrantano, J., and Riley, G., 'Expert system : Principles and programming,' MT : PWS-KENT Publishing, 1989.
- [7] K.Jenson, "Colored Petrinets and the invariant method," Research report, DAIAMI PB-104-AARHUS University, 1990.
- [8] 최주행, "퍼지 추론법의 비교에 관한 연구", 한양

대학교, 1991.

- [9] 도철웅, '교통공학 원론', 청문각, 1993.
- [10] 이윤배 외 3인, "퍼지논리를 이용한 교통신호 제어", 한국정보과학회지, 제21권, 제2호, pp.385-388, 1994.
- [11] 이동익, "패트리넷 이론의 기초", 정보처리학회지 제2권, pp.70-84, 1995.
- [12] 이영신, "퍼지제어를 적용한 교차로에서의 최적교통신호시스템", 정보처리 논문지 제4권, 제1호, 1997.



김 미 경

e-mail : yblee@mina.chosun.ac.kr
 1991년 한국산업인력관리공단 전남직업전문학교
 1996년 춘천기능대학 정보기술학과 조교수
 1999년~현재 고창기능대학 정보통신설비학과 조교수

관심분야 : 인공지능, 지식베이스 시스템, 퍼지추론, 전문가 시스템

이 윤 배



e-mail : yblee@mina.chosun.ac.kr
 1980년 광운대학교 전자계산학과 졸업(이학사)
 1983년 광운대학교 대학원 전자계산학과 졸업(이학석사)
 1993년 숭실대학교 대학원 전자계산학과 졸업(공학박사)

1988년 9월~1999년 현재 조선대학교 컴퓨터공학부 부교수

1997년 9월~1999년 2월 조선대학교 정보과학대학장

1996년 7월~1999년 현재 광주광역시 및 전라남도 정보화추진위원

1999년 4월~현재 광주광역시 시정정책자문위원

관심분야 : 인공지능, 전문가 시스템, 멀티미디어, 화상인식, 데이터베이스