Proceedings of the 11th
KACC, October 1996

시간논리구조에서 이산사건시스템의 최적화

Optimization of Discrete Event System in a Temporal Logic Fmework

황 형 수*, 오 성 권*, 주 수 종**, °정 용 만*

* 원광대학교 제어계측공과(Tel : 50-6345; Fax :53-2196; E-mail :hshwang@engine.wonkwang.ac.kr)
** 원광대학교 컴퓨터공학과(Tel :50-6750;Fax :856-8009; E-mail :scjoo@cs.wonkwang.ac.kr)

Abstracts In this paper, we consider the optimal control problem based on *Discrete Event Dynamic Systems(DEDS)* in the *Temporal Logic framework(TLF)* which have studied for a convenient modeling technique. The *TLF* is enhanced with objective functions(event cost indices) and a measurement space is also defined. Our research goal is the design of the optimal controller for DEDSs. This procedure could be guided by the heuristic search methods. For the heuristic search, we suggested the Stochastic Ruler algorithm, instead of the A* algorithm with difficulties as following; the uniqueness of solutions, the computational complexity and how to select a heuristic function. This SR algorithm is used for solving the optimal problem. An example is shown to illustrate our results.

1. 서론

이산사건시스템은 최근 많은 연구가 이루어지고 있으며 많은 대규모 동적 시스템들은 이산사건구조로 이루어져 있다. 이산사건시스템은 미리 정의된 사건 집합으로부터 사건의 발생에 따라 상태가 변화하는 시스템으로 제조공정의 조립라인, 통신시스템, 전문가시스템 등이 있다. 이산사건시스템을 제어하고 모델링 하는 방법은 Automata and formal Language [1], Finitely Recursive Process, Mini-Max Algebra, Petri-Net 그리고 시간논리구조를 이용한 방법이 제안되었다. 가제어성, 가관측성, 감시제어 그리고 최적제어 등과 같은 많은 제어이론들이 이산사건시스템에 대하여도 연구되었다. 특히 이산사건 시스템의 최적제어에 관하여 Ramadge and Wonham[1]은 주어진 언어에 대한 제어 가능한 부언어의 최적성을 고려하였고, Sengupta and Lafortune[3]는 최적제어 문제에 관하여 그래프-이론 형식을 제안하였으며, Passino and Antsaklis[4]는 이산사건시스템의 모든 상태천이의 비용을 고려하였다. 즉,사건의 비용함수를 이용하여 최적화를 연구하였다. Lin and Ionescu[5]은 시간논리구조에서 A^*알고리즘을 이용하여 이산사건시스템의 최적화 방법을 제시하였으며 이 논문에서 해의 유일성, 계산의 복잡성 그리고 경험함수의 선택 등의 몇가지 문제점을 제시 하였다.

본 논문에서는 지금까지 연구된 방법중 가장 우수한 방법으로 평가되는 시간논리구조의 최적화 방법[5]에서 A*알고리즘대신 Stochastic Ruler 알고리즘을 이용하여 A*알고리즘을 이용하였을 때 발생하는 해의 유일성, 계산의 복잡성 그리고 경험함수의 선택 등의 문제점을 해결하여 최적제어기 설계를 위한 최적화 방법에 대하여 연구하였다.

2. 시간 논리 구조

시간 논리는 시간 순차열에 따른 추론 지향적인 논리이며, 논리를 이용하고 시제 표현식으로 표현될 수 있는 두 가지 장점을 갖는 시간논리는 주로 Software 증명에 이용되었으며 최근에는 이산사건시스템의 제어문제에 적용되었다[2]. 시간논리는 시간의 개념을 포함시킨 일반 논리의 확장이며 또한 시간논리는 종래의 부울 연결자 ¬(not), ∧(and), ∨(or), →(implies), ↔(if and only if))와 시간의 변화와 시간의 량을 표현하기 위한 시제 연산자 □(henceforth), ◇(eventually), ○(next), U(until), P(proceed)와 같은 연결자와 연산자를 이용하여 시간과 시제관계에 관한 추론이 가능한 구조를 갖고 있다

2.1 측정 공간에서 시간논리구조식

공정의 시간논리구조는 6-변수, 즉 $M=(S,E,F^*,f,s_0,l)$로 정의된다. 여기서 S는 유한한 상태들의 집합이며, 그리고 E는 유한한 사건들의 집합을 나타낸다. 각 사건들은 한 상태를 다른 상태로 천이 시킨다. s_0 는 초기상태, f는 발화(firing) 함수이며 상태들의 집합 S로부터 E^*(E^*는 사건들의 집합 E의 부분집합으로 각 원소들은 실행 가능한 사건들을 나타낸다.)에 대응시킨다. 즉 $\forall s\in S$ 그리고 $\forall e\in f(s)$일 때 $s'=f(s,e)$로 정의되며 $f(s)$는 상태 s에서 발생되는 사건들의 집합을 나타낸다. l은 사건들의 집합 E로부터 F^*(F^*는 논리식의 집합)로의 labeling 함수를 나타낸다.

발생함수 f에 의하여 $e(s)=f(s,e)=s'$는 상태 s로부터 다음상태 s'의 천이를 나타낸다. 따라서 발화함수를 다음상태함

수(next state function)라 부른다. 시간논리구조에서 한 상태는 정확하게 한 사건의 실행에 대응한다. 이러한 사실을 반영하기 위하여 한 사건의 발생의 결과를 나타내기 위한 식 $l(e_i, s_i)$는 다음과 같다.

$$\Box[s_i(x) \wedge e_{i+1}(x) \rightarrow s_{i+1}(\bigcirc x)]$$

여기서 $s_i(x)$는 원소 x의 현재상태를 나타내며, $e_{i+1}(x)$는 x에 대응하는 사건의 발생을 나타내고, $s_{i+1}(\bigcirc x)$는 다음상태를 표현한다. ,

2.2 측정 공간에서 도달성

이장에서는 시간논리구조에서 도달성을 소개한다. 그리고 상태들의 도달성과 시간논리구조식 사이의 관계를 규명하고 끝으로 도달성 그래프 작성과정을 소개한다. 임의의 한 상태 s 에서 한 사건의 발생의 결과는 다음상태 s'으로 천이를 의미한다. 즉 다음 상태 s'은 전 상태 s로부터 즉각적인 도달을 의미한다. 시간논리구조 $M=(S, E, F^*, f, s_0, l)$에서 다음상태 s'은 식 $f(e, s_0)=s'$를 만족하는 임의의 사건 $e \in Es_0$가 존재한다면 초기상태 s_0로부터 바로 도달된다. 여기서 Es_0는 초기상태 s_0로부터 발생할 수 있는 사건의 집합을 나타낸다. 이 개념은 주어진 시간논리구조 M에 대하여 도달 가능한 상태들의 집합을 정의하도록 확장할 수 있다. 만일 상태 s'이 초기 상태 s_0로부터 즉각적으로 도달되고 상태 s''은 상태 s'으로부터 즉각 도달되어 진다면 그때 상태 s''은 초기 상태 s_0로부터 도달 가능하다고 할 수 있다. 초기상태 s_0로부터 도달될 수 있는 모든 상태들로 이루어진 임의의 시간논리구조 M의 도달성 집합 $R(M, s_0)$를 정의한다. 즉 $s_0 \in R(M, s_0)$이며, 만일 $s \in R(M, s_0)$이고 $e \in E_s$에 대하여 $s'=f(e, s)$이 성립하면 그때 역시 $s' \in R(M, s_0)$이 성립하며 결국 $R(M, s_0)$)은 이들에 의해서 정의된 가장 적은 상태들의 집합이다. 한 상태와 일련의 사건들의 열을 새로운 상태로 대응시키는 발화함수(다음상태함수)는 편리하게 확장될 수 있다. 따라서 시간논리구조 $M=(S, E, F^*, f, s_0, l)$가 주어졌을 때 만일 상태 $s'=f(e, s)$가 $e \in E_s$에 대하여 상태 s로부터 도달 가능하며 상태 s_0에서 상태 s로 변화시킬 수 있는 일련의 사건 발생이 존재하고 동적 방정식 $l(e, s) \in F$가 참이면 상태 s 는 집합 $R(M, s_0)$에 존재한다. 그리고 사건의 비용을 측정하기위하여 시간논리구조와 관련이 잇는 측정공간을 정의하기 위한 측정함수를 소개한다. 사건비용함수 $\theta : E \rightarrow R(\theta(\bar{e}_{ij})=\theta(s_i, s_j)$이다.

$\bar{e}_{ij} \in Es_i$이며, $s_i=f(\bar{e}_{ij}, s_i)$이다. 여기서 R은 도달성 그래프를 말하며, $\bar{e}_{ij}$는 상태 s_i와 s_j사이 호(Arcs)를 나타낸다. 사건비용함수는 사건 $\bar{e}_{ij} \in Es_i$가 성립해야 정의된다. 분명히 $\theta(\varepsilon)=0$이며, 여기서 ε은 영 사건을 의미한다. 만일 $e_{1k}=e_1 e_2 \cdots e_k$면 그때 $\theta(\bar{e}_{1k})=\sum_{i=1}^{k} \theta(e_i)$이다. 따라서 우리는 시간 논리구조 M에 비용함수 θ를 추가함으로써 개선된 시간논리구조 $M'=(S, E, F^*, f, s_0, l, \theta)$을 얻는다.

사건의 발생은 수행되어진 행동을 나타내며, $\theta_{ij}=\theta(\bar{e}_{ij})$은 사건 e_{ij}의 발생에 의하여 상태 s_i로부터 상태 s_j로 진행된 비용 측정을 나타낸다. 개선된 시간논리구조의 실행은 일련의 상태들의 열, 일련의 사건들의열 그리고 일련의 사건들의 비용의 순열등의 세 순열을 얻게 된다. 이것은 도달성 그래프에 의하여 분명하게 표현된다. 다음 장에서 Stochastic Ruler 알고리즘을 이용하여 이산사건시스템의 최적제어 문제를 다루기 위하여 앞에서 다룬 세 순열을 정의한다.

- $S=\{s_0\ s_1 \cdots\}$은 상태들의 집합이다.
- $E=\{\bar{e}_{ij} \mid s_j=f(\bar{e}_{ij}, s_i)$ for all $s_i, s_j \in S\}$는 상태 s_i로부터 s_j를 향하는 사건 e_{ij}에 대한 유향호 (Directed arcs)의 집합이다.
- $R=\{\{\theta_{ij} \mid \theta_{ij}=\theta(\bar{e}_{ij}), \bar{e}_{ij} \in E\} \vee \{\theta_j \mid \theta_j=\theta(s_j), s_j \in S\}\}$.

여기서 θ_{ij}는 상태 s_i로부터 상태 s_j사이의 각 호(Arcs)에 관련된 사건 비용함수의 집합이며, $\forall \theta_{ij} \in R$에 대하여 $\theta_{ij} \geq 0$이다. 여기서 θ_{ij}는 균일하게 분포된 임의 변수로서 Stochastic Ruler를 나타낸다. θ_j, $\theta_j \geq 0$는, 현재상태 s_i로부터 존재할 수 있는 다음 상태들 $s_j(j=1, 2, \cdots, i=j)$에 관련된 사건비용함수의 집합이며 이 함수는 SR알고리즘을 이용한 모의실험에 의해 얻어진다.

3. 이산사건시스템의 최적제어 문제

제어이론에서 최적화는 제어기 합성 및 최적제어에서 중요한 부분이었다. 지금까지 최적화 과정은 경험적 탐색 방법에 의한 A^*알고리즘을 이용한 연구가 이루어졌었다, 그러나 A^*알고리즘을 사용했을 때 발생하는 해의 유일성, 계산의 복잡성 그리고 경험함수의 선택 등의 문제점등이[5] 발생했으며, 이문제점 등을 해결하기 위하여 본논문에서는 A^*알고리즘 대신에 Stochastic Ruler 알고리즘을 사용한다. .

3.1 최적 목적 함수

플랜트의 시간논리구조 M에서 구할 수 있는 최적 궤도 ($\sigma \in S^*$)를 결정하는 것은 어려운 최적화 문제이다. 즉 $g(s) \leq g(s')$, $\forall s \in S^* \subset S$를 만족하는 최적 상태 집합 s^*를 구하는 것이다. 여기서 $S=\{s_i, \cdots, s_k\}$는 σ에서 실행 가능한 상태들의 집합이며 $g(s)$는 상태 s에 대응하는 목적함수 이다. 목적함수 $g(s)$가 분리 가능할 때, $g(s)$가 $g(s)=\sum_{i=1}^{n} g_i(s^{(i)})$로서 쓰여질 수 있다. 여기서 $s=(s^1, \cdots, s^n)$이다. 그러나 본 논문에서의 이산사건 시스템의 최적제어 문제는 분리될 수 있는 목적함수가 아니며 또한 탐색 알고리즘은 측정공간이라 불리는 상태공간에서 어떻게 다음상태를 탐색할 것인가에 의하여 특성지어 진다. 탐색 방법의 대부분은 brand-and-bound 알고리즘과 simulated annealing 알고리즘이다.[6] 이 두 알고리즘은 목적함

수 $g(s)$의 해석적 표현이 요구된다. 우리가 지적한 것처럼 고려되어진 목적함수는 해석적인 형태로도 표현될수 없다. 단지 목적함수 $g(s)$는 경험적인 데이터나 혹은 이산사건 모의실험[7]에 근거를 두고 평가되어질 수 있다. 특히 우리는 상태 s_0로부터 시작해서 상태 s_k에서 끝나는 궤도의 경우에 흥미를 갖고 있다. $M(s_0, s_k)$는 초기상태 s_0에서 시작해서 s_k에서 끝나는 시간논리구조 M의 모든 유한 궤적($\sigma = s_0, s_1, \cdots, s_k$)의 집합을 의미한다. 우리의 목적은 목적함수를 최소화하거나 최적궤도 σ^*를 구할 수 있는 일련의 상태들의 열을 구하는 것이다. 즉 $g(s) = \min\{g(s) | s \in S\}$, 여기서 실행 가능한 집합 $S = \{s_1, s_2, \cdots, s_k\} = \{s_i | i \in I\}$와 지수집합 $I = \{1, 2, \cdots, k\}$는 유한하며 그리고 목적함수 g는 $g: S \rightarrow IR$이다. 여기서 문제의 중요한 특징은 실행가능한 해의 집합이 이산특성을 갖고 있는 것이다. 그러나 s의 크기가 임의의 크기로 만들어 질 수 있고 그리고 이론적 해석이 편리하다고 할 지라도 s의 유한성은 실제 응용에서의 제한 때문에 고려 될 수 없다.

최적제어기 합성 문제는 궤도 $\sigma^* \in M(s_0, s_k)$를 따라 시스템 M을 구동하는 일련의 상태들의 열을 얻을 수 있는 제어기를 찾는 것이다. 그러한 상태들의 열을 최적제어경로라 부르며 그 궤도 σ^*를 최적 궤도라 부른다. 그러한 상태들의 열 σ^*이 결정되면 상태열은 궤도 σ^*을 따라 플랜트 M을 구동시킬 수 있다.

3.3 Stochastic Ruler알고리즘을 이용한 제어기합성의 최적화

시간논리구조에서 제어기의 설계는 일반적으로 목적함수 값을 최소화할 수 있는 상태 열에 대응하는 경로를 찾기 위한 상태 탐색공간을 포함한다. 여기서 본 논문은 Stochastic Ruler 알고리즘을 이용한 경험적 탐색 이론을 이용한다.[7] 따라서 우리는 위의 문제에 대한 부분적인 해 보다는 전체적인 해를 찾는다. 일반적으로 이산사건 시스템에서 최적화 문제는 아래와 같이 전역 최적 집합 S*(global optimum set)을 찾도록 요구한다.

$$S^* = \{s \in S \mid g(s) \leq g(s') \ \forall \ s' \in S\}$$

여기에서 g(s)는 상태s에 대한 시스템의 목적 함수이다. 이 함수는 분석적인 수식이 불가능하므로 시뮬레이션 평가를 위해 g(s)에 대한 추정값으로써 H(s)를 사용한다. 또한 θ(a,b)는 a<b 조건하에서 a 와 b간의 일양분포에서 얻어진 무작위 값을 나타내며, 이를 Stochastic Ruler라고 부른다. g(s)의 상한 과 하한 경계값을 a(s)와 b(s)로 각각 정의한다.

max= { P(s,a,b) | s∈S }가 최대값이 되도록 하기 위해 본 연구에서는 다음과 같은 방법을 제시한다.

$\Sigma^* = \{ s \in S \mid P(s,a,b) \geq P(s',a,b) \ \forall \ s' \in S\}$, where $P(s,a,b) = P[H(s) \leq \theta(a,b)]$.

상태들의 집합 Σ^*가 S^*에 수렴한다는 사실을 SR 알고리즘을 통해 보여주는 것은 어렵지 않다. 위 과정을 설명하기 위해 SR 알고리즘에 대해 세부적으로 살펴보면, SR 알고리즘은 s∈S에 대해 미리 정의된 초기 상태 s0에서 출발한다. S의 이웃(neighbor : 다음 상태가 될 수 있는 상태들의 집합)들을 N(s), N(s)⊂S라고 정의한다면, 그 다음에 임의로 N(s)에 있는 상태들 중에서 하나를 선택하여 s'라 할 때, s'에 대한 시뮬레이션을 통해 얻은 목적함수의 표본값들 $x_1, x_2, \ldots, x_n$과 Stochastic Ruler [a,b]의해 일양 분포로 부터 무작위로 추출한 표본값 $y_1, y_2, \ldots, y_n$을 서로 비교한다. 만일 $x_i \leq y_i$ (for $\forall i = 1, \ldots, n$)이면, 상태 s에서 다음 상태 s'로 탐색을 이동시킨다. 그렇치 않으면, 현재 상태에서 표본값들 $x_1, x_2, \ldots, x_n$과 표본값 $y_1, y_2, \ldots, y_n$을 서로 비교하는 수행을 반복한다. 즉, 이 방법은 목적함수의 표본값과 Stochastic Ruler 표본값들을 서로 비교·탐색하므로써 목적함수를 최소화 시키게 된다.

본 SR 알고리즘의 정형화하기 위해 R(s,s'), s,s'∈ S를 현재 상태 s에서 다음 상태 s'로 탐색이 이동될 확률값이라 할 때, 여기서 R(s,s')과 R(s',s)은 대칭특성을 요구하며, 무한 증가 변수로서 M_k는 k -> ∞일 때, M_k -> ∞임을 만족한다.

SR 알고리즘은 비교될 수 있는 두 상태 s, s'을 고려한다. 그리고 상태 s와 s'의 목적함수의 실질적인 의미는 각각 $P(s)$와 $P(s')$으로 나타내어진다. 그때 만일 $|P(s) - P(s')|$가 크다면 그들 사이의 차에 대하여 그들 사이에 상태가 존재한다고 추측하기 힘들 뿐만 아니라, 상태들의 사이에 좋은 성능을 기대하기도 힘들다. 대조적으로 $|P(s) - P(s')|$가 적다면 정확한 평가를 하는 것은 불가능하나, 두 상태들 사이에서 오류를 찾아내는데 관련된 문제점은 거의 없다. 다시 말하면 SR 알고리즘은 성능(performance)이 큰 차를 나타내는 두 상태들 사이의 비교에서는 견실하다.

4. 예제

제시된 방법의 예를 보이기 위하여 8-퍼즐이라 불리는 인공지능의 전형적인 실례를 들어 고찰해 본다. 8-퍼즐은 9 셀(cell)의 판이 있고, 셀에는 8개의 타일과 한 개의 빈 셀이 있다.(그림3) 타일은 1에서 8까지 번호가 붙여져 있고, 셀(cell)에는 1에서 9까지의 번호가 붙어 있다. 어떤 인접한 셀이 비어있다면, 타일은 한 셀에서 비어있는 다른 셀로 이동할 수 있다. 예를 들어, 그림 1에서 보인 바와 같이 타일 6은 셀 9를 비우고 셀 6으로 이동할 수 있다. 게임은 그림 1-a)에서 보인 것처럼 임의의 초1기 상태에서 시작한다. 타일 이동의 적절한 순서는 최소한의 이동으로 그림 1-d)에 보인 최종상태에 도달하도록 제어기에 의해 이루어 져야 한다.

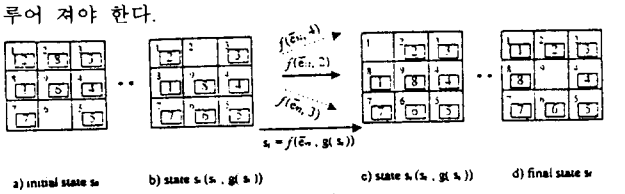

그림 1. 8-퍼즐과 f함수의 맵핑

Fig. 1 The 8-puzzle and the mapping of function f

함수 f의 맵핑(mapping) 예는 그림 1-b), c)에서 다음과 같이 보여진다. s_i는 그림 1-c)에서 구성의 상태라 하고, e_{ij}는 타일 2가 2에서 1로 이동하는 사건이라 하자. 그러면 $s_j = f(e_j, s_i)$는 구성의 상태이다. 여기서, 어떤 타일이 셀 y에서 z로 이동할 때 y와 z가 셀 이동 수라면 $e_j = \overline{e}_{yz}$이다. 우리는 $s_j = \{s_j \in S \mid P(s_i, a, b) \geq P(s_j, a, b) \ \forall s_i \in S\}$의 최적 해에 대해 SR 알고리즘을 설명한다. 여기서, $P(s_i, a, b) = P[H(s_i) \leq \theta$

(a, b)]이다. x를 타일에 붙여진 번호를 표시하는 지역변수라 하고, y와 z를 셀에 붙여진 번호를 표시하는 지역변수라 하자. 그러면 다음과 같은 속성을 수반한다.

- IN(x,y) - 타일 x는 셀 y에 있다.
- EM(z) - 셀 z는 비어있다.
- EQ(x,y) - x에 붙여진 숫자는 y와 같다.
- ADD1(n) - 수 n은 1이 증가된다.
- MIN1(n) - 수 n은 1이 감소된다.

어떤 다른 속성은 이 예제에서 다루어지지 않는다. 그리고 가능한 사건은 다음과 같다.

- move(x,y,z):타일 x가 셀 y에서 y에 인접한 셀 z로 이동.

플랜트에서 발화규칙은 다음에 나오는 TLF에 의해 주어진다.

$\Box$[IN(x,y)$\wedge$EM(z)$\wedge$move(x,y,z) $\rightarrow$ $\bigcirc$(IN(x,z)$\wedge$EM(y))].

위의 서술을 기초로 제어기 설계를 다음과 같이 주어진다.

$\Box$[move(x,y,z)$\rightarrow\bigcirc$[EQ(x,z)$\wedge$MIN1($g(s_i)$)) $\vee$

($\neg$EQ(x,z)$\wedge$ADD1($g(s_j)$))].

여기서, $g(s)$는 그들의 위치나 각 상태의 목적함수 밖에 있는 타일의 수를 나타내는 지역변수이고 s_i와 s_j에 대하여 $g(s_i)=3$과 $g(s_j)=2$이다. $g(s_i)$에 대한 상태 s_i와 $g(s_j)$에 대한 상태 s_j 그리고 폐루프 시스템에서 맵핑(mapping) f는 위의 그림 1-b), c)에서 보였다.

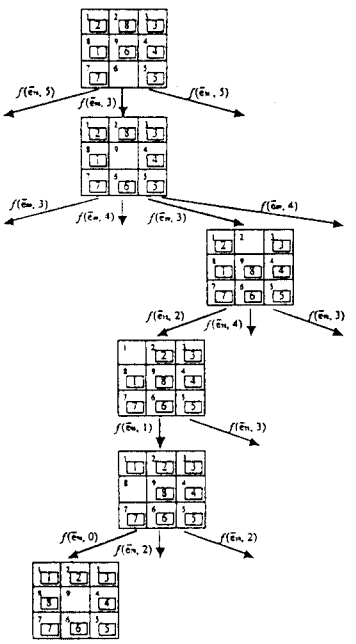

그림 2. 8-퍼즐의 탐색 구조

fig. 2 A search tree of the 8-puzzle

우리는 최종상태에 도달하기 위한 이동의 최적 경로를 찾을 수 있다. 이러한 최적 제어를 위한 모의실험에서 우리는 [1,10]에 의해 균일하게 분포된 θ와 $M_k=k/20$, 상태 s_0의 목적함수인 $g(s_0)$는 4임을 얻는다. 또한, 현재 상태는 3개의 인접한 상태를 최대로 이동할 수 있다. 위와 같은 과정에 의하여 그림 3에서 나타난 구해진 사건의 최적순서는 move(6,9,6), move(8,2,9), move(2,1,2), move(1,8,1), move(8,9,8)이다. 그림 3에서 최종단계의 두상태 move(1,8,1)과 move(8,9,8) 사이에서 진동하는 것을 볼 수 있는데 이것은 더좋은 목적함수 $g(S_f)$로 최종산태 s_f를 찾기위한 절차이다. 이것은 최적제어 문제인 DEDS의 최소 사건 비용 제어문제의 실예이다.

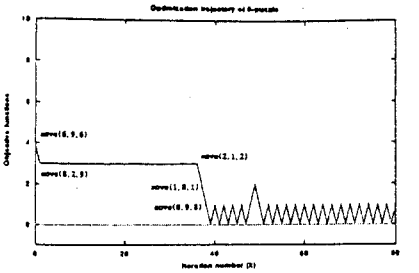

그림 3. 8-퍼즐의 최적화궤도

Gig. 5 the Optimization trajectory of 8-Puzzle

5. 결론

우리는 DES에 대한 최적제어 문제에 시간논리구조를 소개하고, 우리는 측정공간에서 사건비용함수를 정의하여 개선된 시간논리 구조를 제안하여 이산사건시스템에 대한 최적제어 문제를 수식화 했다. 이를 이용하여 A* 알고리즘을 이용한 방법에서 문제가되는 해의 유일성, 경험함수의 선택, 그리고 계산의 복잡성 등의 문제를 Stochastic Ruler 알고리즘을 이용하여 해결하였으며, 이 방법의 우수성을 8-퍼즐의 예를 들어 보였다.

참고문헌

[1] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event process" SIAM J. Control. Optimiz., Vol.25, no.1,pp206-230, 1987.

[2] D. Ionescu, J.-Y. Lin, and H. S. Hwang, "Specification of Intelligent controllers for discrete event system in a temporal logic framework" IFAC workshop on Algorithms and achitecture for real time control, Seoul, Korea Aug. 1992. pp. 237-242.

[3] R. Sengupta and S. Lafortune, "Optimal control of a class of discrete event systems", IFAC *Symposium on Distributed Intelligence Systems*, Arling, Virginia, August 13-15, 1991, pp.25-30.

[4] K. M. Passino and P. J. Antsaklis, "On the optimal control of discrete event systems" Proc. 28th *IEEE Conf. Decision and Control*, Tampa, Florida, Dec. 1989, pp.2713-2718.

[5] J. -Y. Lin and D. Ionescu, "Optimization of controller design for discrete event systems in a temporal logic framework"'92 American Control conference,pp.2819-2823. 1992

[6] N. J. Nilsson, Principles of Artificial Intelligence. Palo Alto, California, 1980.

[7] D. Yan and H. Mukai, "Stochastic Discrete Optimization," SIAM *Journal on Control Optimization*, 1992.