대한산업공학회/한국경영과학회 2002 춘계공동학술대회
한국과학기술원(KAIST) 2002년 5월 3일~4일

강화학습기법을 이용한 TSP의 해법
(A learning based algorithm for Traveling Salesman Problem)

임준묵
대전시 유성구 덕명동 한밭대학교 산업경영공학과

吉本一穂
早稲田大学，経営システム工学科

林載国
早稲田大学，アシアー太平洋研究センタ

강진규
대전시 유성구 덕명동 한밭대학교 산업경영공학과

Abstract

본 연구에서는 각 수요지간의 시간이 확률적으로 주어지는 경우의 TSP(Traveling Salesman Problem)를 다루고자 한다. 현실적으로, 도심의 교통 체증 등으로 인해서 각 지점간의 걸리는 시간은 시간대별로 요일별로 심한 변화를 일으키기 마련이다. 그러나, 현재까지의 연구 결과는 수요지간의 경과시간이 확정적으로 주어지는 경우가 대부분으로, 도심 물류 등에서 나타나는 현실적인 문제를 해결하는데는 많은 한계가 있다. 본 연구에서는 문제의 해법으로 강화학습기법의 하나인 Q학습(Q-Learning)과 Neural Network를 활용한 효율적인 알고리즘을 제시한다.

Key words: TSP, Q-Learning, Neural Network.

1. 서론

TSP는 물류문제 등에서 매우 폭넓게 활용될 수 있는 기본적이면서도 중요한 문제 중의 하나이다. 지금까지 연구되어온 일반적인 TSP에서는 각 수요지간의 시간이 확정적으로 주어지는 경우가 대부분이었다. 그러나, 도심 등에서 일어나는 물류문제에서는 이러한 가정은 현실적이지 못하다. 예를 들어, 택시로 서울의 종로1가에서 동대문을 간다고 가정해보자 즉, 종로1가→종로3가→종로5가→동대문의 경로를 택했다고 하자. 원활하게 소통이 될 경우는 20분 정도면 충분할 것이다. 그러나, 때로는 똑 같은 길로 간다고 해도 1시간도 넘게 걸리는 경우도 자주 발생한다. 이것은 각 지점간의 거리는 같더라도 걸리는 시간은 커다란 편차가 있음을 의미한다. 또한, 종로1가가 정체를 이루고 있으면 종로3가, 종로5가도 정체가 될 가능성이 높다. 이것은 각 지점간의 걸리는 시간이 편차(분산)를 가질 뿐 만 아니라 서로 의존적(독립이 아님)이라는 사실을 나타내고 있다.

본 연구에서는 이렇게 각 지점간의 시간이 확률적으로 주어지며, 서로 독립적이지도 않을 수 있는 환경의 TSP를 다루고자한다.

다음절에서는 문제의 설정을, 그리고 제3절에서는 Q학습을 활용한 TSP알고리즘을 제시한다. 제4절에서는 수치실험결과를 보여주고, 마지막으로 결론을 제시한다.

2. 문제의 설정

2.1 전제조건

본 연구에서는 일반적인 TSP상황에 추가로 다음을 가정한다.

(1) 각 지점간의 시간은 평균과 분산에 의해서 주어진다.

(2) 각 지점간의 걸리는 시간의 분포는 통계적으로 서로 독립적이지 않다.

(3) 각 지점을 방문하는 시각에 대한 제약 즉, Time Window는 고려하지 않는다.

2.2 문제의 목표

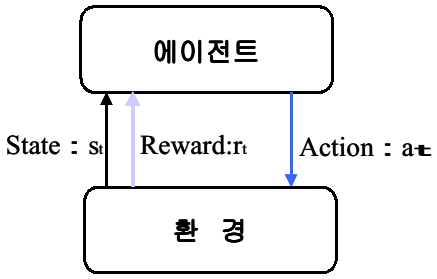
위에서 주어진 환경 하에서의 TSP의 목표는 모든 지점을 방문하고 돌아오는데 걸리는 시간의 기대치는 물론 전체경로의 분산의 최소화를 목표로 한다.

3. Q학습을 활용한 TSP알고리즘

3.1 Q학습의 개요

본 연구에서는 확률적 환경의 최적경로를 찾기 위해서 강화학습(Reinforcement Learning) 기법의 하나인 Q학습을 이용하였다. 강화학습기법이란 [그림 1]과 같이, 어떤 환경(Environment)이 주어져있고 이 환경을 이용하는 에이전트(Agent)가 있다고 하면, 먼저 에이전트가 환경으로부터 어떤 상태(State (s_t))를 인식하게된다. 이 때 에이전트는 특정 상태에 대한 여러 가지의 행동(Action(a_t))을

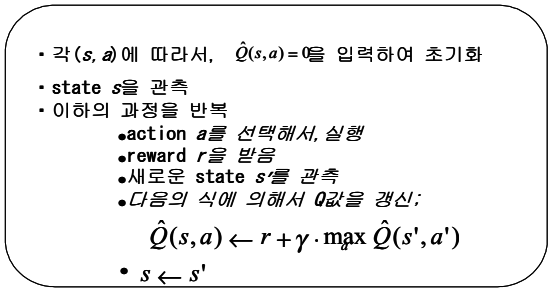
취할 수가 있는데 그 행동 중 하나를 취하면, 그 행동에 대해 환경으로부터 상금(Reward (r_t)) 또는 벌점(Penalty)을 받게 된다. 이러한 일련의 과정의 반복을 통해 에이전트는 각 상태에 대한 행동들 가운데 좋은 상금(또는 벌점)을 받는 쪽으로 점점 행동을 취할 가능성(확률)을 높여 가게된다. 이러한 방식으로 무한하게 반복을 하게 되면, 결국 하나의 상태에 대해 가장 좋은 행동 하나가 선택되어질 확률이 1에 가깝게 되어 상태와 행동이 일대일 대응 관계로 되게 된다. 이러한 학습과정이 끝나면, 하나의 상태에 대해 취할 수 있는 여러 가지 행동 가운데 가장 성능이 좋았던 행동 하나만을 추출하여 최종적으로 하나의 상태에 대해 하나의 행동만을 취할 수 있도록 하는 기법이다. 이것은 마치 인간의 행동과도 흡사하다고 할 수 있다. 인간의 행동도 착한 일을 하면 상을 주고 잘못을 했을 경우에는 벌을 주어 점점 더 좋은 행동을 취할 수 있도록 하는 것과 흡사한 학습 방법이라 할 수 있다.[5]



[그림 1] 강화학습기법

3.2 Q학습 알고리즘

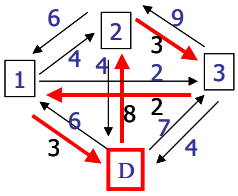
일반적인 Q학습의 알고리즘은 [그림 2]와 같다. 앞 절에서 언급한 바와 같이 초기의 Q값으로 시작하여 (State의 관측 → Action의 실행 → Reward의 수리 → Q값의 갱신)의 절차를 반복적으로 수행한다.



[그림 2] Q학습 알고리즘

3.3 일반 Q학습을 활용한 TSP 알고리즘

TSP를 Q학습을 활용하여 풀기 위해서는 Q학습의 요소들을 정의함에 의해서 가능하다. 출발지가 D이고 수요지가 1,2,3으로 주어지는 간단한 TSP문제를 예로들어 설명하기로 한다.



[그림 3] TSP 예제

[그림 3]에서 보는 바와 같이 각 수요지간의 이동시간은 확정적인 수치에 의해서 주어진 것으로 가정하자. 매우 간단한 문제이므로 최적해는 D→2→3→1→D이고 총 소요시간은 16이 됨을 쉽게 알 수 있다. 그러면 TSP를 Q학습의 각 요소에 대응시키기로 하자.

(0) Goal : D를 출발하여 가장 짧은 시간에 모든 노드를 한번씩 방문하고 D로 돌아옴.
(1) State : 현재까지 방문한 노드의 집합 + 현재위치
(2) Action : 다음 방문지(노드)
(3) Reward : 노드간의 거리(시간)
(4) Q-value : 현재 노드에서부터 남은 노드 전부를 방문하고 목적지(D)까지 갈 때의 총 소요시간의 할인된 기대치.
(5) Learning Rule:
$\hat{Q}(s,a) = r(s,a) + \gamma \cdot \min_{a'} \hat{Q}(s',a')$, 여기서 $r(s,a)$는 상태 s에서 Action a를 취했을 때 받게되는 Reward를 의미하며, γ는 할인율(discount rate), $\hat{Q}(s,a)$는 Q값의 추정치를 말한다.

이제 위에서 TSP에 정의된 Q학습의 각 요소를 예를 들어 보기로 하자. 방문자가 D를 출발하여 2의 노드에 와 있다고 하자. 즉, D→2→?. 현재의 위치는 2이고 현재까지 방문한 노드집합은 {D,2}이 된다. 따라서, 현 상태는 {D,2}와 {2}로 정의된다. 다음의 Action으로는 노드1, 3 및 D가 예상되나 D로 가는 것은 TSP정의에 위배되어 현재의 상태에서 선택가능한 Action은 {1,3}이 된다. 현 상태에서 다음의 Action으로 노드1을 선택한다면, Reward는 6이 되고, 노드3을 선택하면 3의 Reward를 받게 된다. 현재의 상태에서 Q값은 현 위치에서 계속해서 Action을 취하고 Reward를 받아 나가서 최종 목적지에 도달할 때까지 받을 수 있는 Reward의 할인된 기대치의 총합을 의미한다. 다시 말해서, 현 위치를 출발하여, 현재까지 방문한 노드를 제외한 나머지 노드를 모두 방문하고 다시 D로 돌아

오는데 걸릴 것으로 예상되는 시간의 할인된 기대치를 말한다.

현재의 상태에서 다음의 Action을 선택하고 Reward를 받게되면 다음으로 Q값을 갱신하게 되는데, Q값은 Q테이블에 저장되어 있는 Q값과 현재의 Action에 대한 Reward로부터 최종 목적지까지 도달할 때까지 얻을 수 있는 Reward의 할인된 기대치가 최적화(TSP에서는 Reward가 시간치이므로 최소화가 된다.)되도록 Q값을 갱신한다.

위의 예제의 경우, Q테이블에서 Q값이 어떻게 갱신되는지 과정을 살펴보기로 하자.
학습을 위한 에피소드(Episode)로는 D→1→2→3→D를 사용하는 것으로 하고 현재의 에피소드를 가지고 4회 반복학습을 한다고 가정하자. 그리고 할인율은 γ=0.9로 한다.
초기의 모든 상태에 대한 Q값은 100으로 하였다. 본 예제에서는 TSP가 최소화문제이므로 0으로 하지 않고 충분히 큰 수인 100으로 하였다. 1회의 학습과정을 거치면 상태의 변화에 따른 Q값이 차례로 갱신된다. 예를 들어, 1회의 96은 6 + 0.9×min{100,100}에 의해서 얻어진다. 그리고 같은 에피소드의 4회째 반복학습에서 9.94는 4 + 0.9 × min{6.6}에 의해서 얻어진다.

[표 1] 학습에 의한 Q테이블 값의 변화

State			a	r	Q-value					
표기	기방문 노드집합	현재 위치			초기	1회	2회	3회	4회	최종
S(t)	{D}	D	1	6	100	96	90.6	84.9	14.95	14.95
			2	8	100					14.51
			3	7	100					14.96
S(t+1)	{D,1}	1	2	4	**100**	94	87.7	9.94	9.94	9.94
			3	2	**100**					13.34
	{D,2}	2	1	6	100					11.04
			3	3	100					7.23
	{D,3}	3	1	2	100					8.84
			2	9	100					16.83
S(t+2)	{D,1,2}	1	3	2	100					5.6
		2	3	3	100	93	6.6	6.6	6.6	6.6
	{D,1,3}	1	2	4	100					7.6
		3	2	9	100					12.6
	{D,2,3}	2	1	6	100					8.7
		3	1	2	100					4.7
S(t+3)	{D,1,2,3}	1	D	3	100					3
		2	D	4	100					4
		3	D	4	100	4	4	4	4	4
S(t+4)	{D,1,2,3,D}	D								

계속해서, 다양한 에피소드를 발생시켜 반복학습을 수행하면 최종적으로 Q값은 [표 1]의 '최종'열에 주어진 값으로 수렴을 하게된다.
이렇게 해서 Q학습이 완료되면 최종 Q테이블의 Q값으로부터 최적경로를 찾는다. D를 출발하여, 상태 S(t)에서 최소의 Q값은 14.51이므로 2를 선택한다. 그러면 새로운 상태는 D→2가되므로 S(t+1)의 {D,2}+{2}에서 최소값 7.23에 해당하는 수요지 3을 선택한다. 현재까지의 경로는 D→2→3이되고 상태는 {D,2,3}+{3}이되고, S(t+2)에서 최소값인 4.7에 해당하는 수요지 1을 선택하여 D→2→3→1이 된다. 마지막으로 D로 돌아옴에 의해서 최적 경로(D→2→3→1→D)가 완성된다.

위의 알고리즘에서 각 수요지간의 시간이 확률적인 분포로 주어지는 경우는 Q값의 갱신을 위한 Learning Rule의 식을 다음의 식으로 바꾸어 계산하기만 하면 된다.

$$\hat{Q}(s,a) = (1-\alpha)\hat{Q}(s,a) + \alpha\{ r(s,a) + \gamma \cdot \min_{a'} \hat{Q}(s',a') \}$$

여기서, α는 Q학습의 학습율을 말한다.

3.4 Neural Network(NN)을 사용한 Q학습 알고리즘

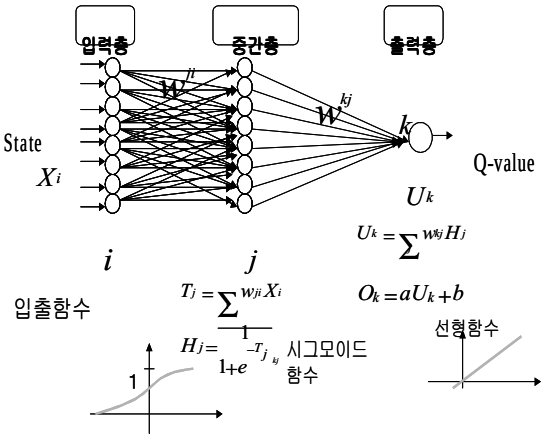
앞 절에서 제시된 일반Q학습에 의한 TSP 알고리즘은 다음과 같은 문제점을 가진다. State의 표현 시 앞의 정의를 사용하게 되면 수요지의 수가 증가하면서 State의 수는 기하급수적으로 증가하여 수요지의 수가 10이상의 경우는 Q테이블의 저장에 애로가 발생한다. 따라서, 본 연구에서는 State의 표현과 Q값들을 Q테이블에 직접 저장시키는 방식 대신에 NN에 학습시켜 저장시키는 방법을 사용한 새로운 알고리즘을 제안한다.

(1) State와 Q값의 학습을 위한 NN

본 연구에서 사용한 NN은 [그림 4]에서 보여지는 형태의 역전파NN을 사용한다. 입력층, 중간층 및 출력층의 3개의 층으로 구성하였다. 중간층에서의 입출력 함수는 일반적으로 널리 사용하는 시그모이드함수,

$$f(x) = 1/(1+e^{-x})$$

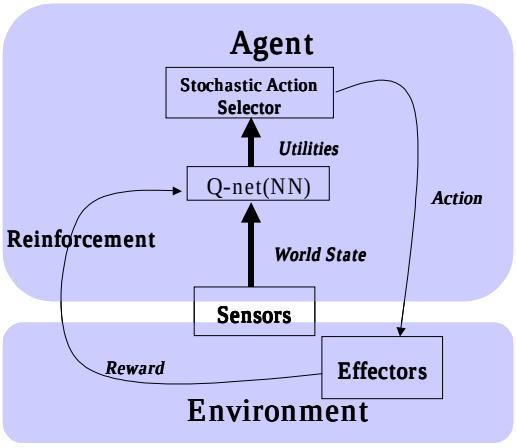
를 사용하였으며, 출력층에서는 Q값이 0이상의 실수값을 가지게 되므로 $f(x) = ax + b$형태의 선형함수를 사용하였다.



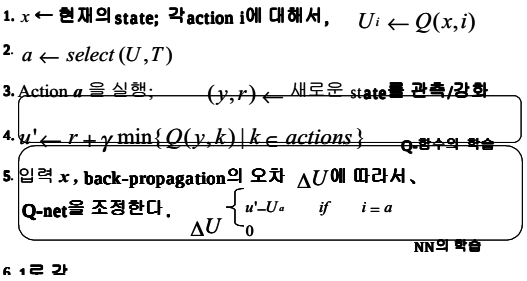
[그림 4] Q값의 학습을 위한 NN

NN과 Q학습을 사용한 TSP 알고리즘의 흐름을 [그림 5]에 도시하였으며, 알고리즘은 [그

림 6]에 주어진다. 알고리즘은 크게 두 부분의 학습모듈을 가진다. 하나는 전체 흐름에 있어서의 Q학습이고, 다른 하나는 Q값을 저장하기 위한 NN의 학습이다. 따라서, 본 알고리즘은 다양한 에피소드를 발생시켜가면서 Q학습과 NN학습을 동시에 행하는 알고리즘이다. 여기서, NN의 학습을 위해서는 역전파알고리즘(Back propagation algorithm)을 사용하였다.



[그림 5] NN과 Q학습을 사용한 TSP알고리즘의 흐름도



[그림 6] NN과 Q학습을 사용한 TSP알고리즘

4. 수치실험

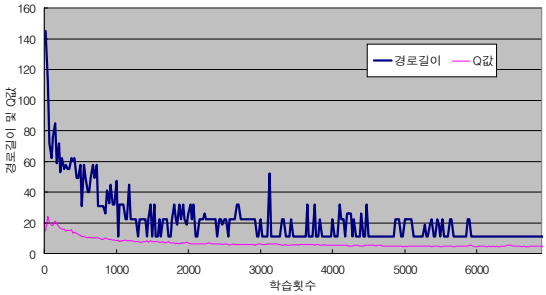
수치실험은 두 가지의 경우에 대해서 실시하였다. 실험1은 앞 절에서 제시한 NN과 Q학습을 사용한 TSP알고리즘이 과연 최적해에 수렴하는가에 관한 검증실험이고, 실험2는 각 수요지간의 시간자료가 확률적으로 주어질 경우와 확정적 자료만을 사용했을 경우와의 비교 평가에 관한 실험이다.

4.1 실험 1 - 알고리즘의 검증

제시된 알고리즘이 최적해에 수렴하는가를 알아보기 위해서 [표 2]에 주어진 것과 같이 임의로 발생시킨 확정적 시간치 자료를 가지고 알고리즘을 수행하였다. 학습 알고리즘에 사용된 모수로는 Q 학습율은 0.05, 할인율은 0.7, 그리고 NN의 학습율은 0.2를 사용하였다. [표 2]의 시간치 자료에 대한 최적해는 0→7→4→5→1→3→6→2→8→10→9→0 이고 경로에 대한 총시간은 11이다. [그림 6]은 알고리즘의 수행결과 학습횟수의 증가에 따른 경로시간과 Q값의 변화를 보여주고 있다. 학습횟수가 증가함에 따라 최적해에 수렴하고 있음을 볼 수 있다. 본 연구에서는 임의로 발생된 다른 여러 가지의 문제 셋트(set)에 대해서도 다양한 실험을 하여본 결과 대부분의 경우에서 적절히 수렴함을 알 수 있었다.

[표 2] 각 수요지간의 시간자료(확정적 경우)

	0	1	2	3	4	5	6	7	8	9	10
0	0	13	7	8	17	2	17	1	18	16	2
1	10	0	9	1	21	19	3	20	9	12	17
2	3	13	0	14	14	7	7	18	1	18	13
3	21	20	6	0	2	6	1	12	4	21	15
4	13	4	4	17	0	1	7	9	8	20	21
5	10	1	7	5	5	0	14	16	8	14	6
6	5	13	1	3	11	20	0	7	7	20	14
7	14	10	3	13	1	15	20	0	11	20	10
8	15	12	12	11	9	10	7	3	0	3	1
9	1	7	9	11	5	11	7	14	12	0	2
10	6	9	4	17	11	17	13	18	2	1	0



[그림 6] 확정적 시간치를 가지는 TSP의 Q학습결과

4.2 실험 2 - 확률적인 경우

실험 2에서는 수요지간의 시간자료가 [표 3]에 보는 바와 같이 평균과 분산에 의해서 주어지는 경우에 대해서 실험을 행하였다.

[표 3] 각 수요지간의 확률적인 시간자료(괄호의 자료는 (평균, 표준편차)를 나타낸다)

	0	1	2	3	4	5	6	7	8	9	10
0	(0,0)	(21,2)	(48,13)	(34,9)	(35,9)	(58,16)	(20,5)	(58,16)	(60,5)	(55,15)	(20,17)
1	(40,11)	(0,0)	(22,2)	(38,10)	(68,19)	(20,18)	(22,5)	(67,19)	(38,10)	(46,12)	(58,16)
2	(22,5)	(49,13)	(0,0)	(20,2)	(51,14)	(52,14)	(33,8)	(33,9)	(61,17)	(61,17)	(49,13)
3	(69,19)	(65,18)	(31,8)	(0,0)	(20,2)	(32,8)	(46,13)	(25,6)	(20,19)	(53,15)	(69,5)
4	(48,13)	(25,6)	(25,6)	(59,16)	(0,0)	(21,2)	(34,9)	(20,10)	(35,9)	(67,19)	(68,19)
5	(40,11)	(33,9)	(20,7)	(28,7)	(52,14)	(0,0)	(23,2)	(55,15)	(36,9)	(51,14)	(30,8)
6	(29,7)	(20,13)	(24,6)	(42,11)	(65,18)	(33,8)	(0,0)	(22,2)	(34,9)	(65,18)	(51,14)
7	(51,14)	(41,11)	(24,6)	(48,13)	(54,15)	(65,18)	(20,17)	(0,0)	(21,2)	(65,18)	(41,11)
8	(53,15)	(45,12)	(45,12)	(43,11)	(37,10)	(40,11)	(33,9)	(22,5)	(0,0)	(20,2)	(23,5)
9	(20,10)	(38,10)	(44,12)	(27,7)	(43,12)	(32,8)	(51,14)	(47,13)	(27,7)	(0,0)	(20,2)
10	(21,2)	(39,10)	(25,6)	(59,16)	(20,11)	(57,16)	(49,13)	(61,17)	(20,5)	(30,8)	(0,0)

실험은 두 가지 방법으로 행하였는데, 하나는 [표 3]의 자료 중 평균치만의 자료만을 사용하여 알고리즘을 수행하였고, 다른 하나는 평균과 표준편차 모두를 고려하여 알고리즘을 수행하였다.

두 실험결과 얻어진 최적해 및 시뮬레이션 분석 결과는 [표 4]와 같다. [표 4]에서 최적경로는 평균만을 고려한 경우와 평균 및 표준편차를 모두 고려한 경우의 각각의 최적해를 의미한다. [표 4]의 3-6행은, 얻어진 각각의 최적경로에 대해서 10회의 시뮬레이션을 수행하여 경로시간의 최소, 최대, 평균 및 분산을 구한 결과를 나타낸다.

[표 4] 최적해 및 시뮬레이션 분석 $\left|\frac{(1)-(2)}{(1)}\right|\times 100$

	평균만을 고려한 경우(1)	평균 및 표준편차를 모두 고려한 경우(2)	편차(%) = $\left\|\frac{(1)-(2)}{(1)}\right\|\times 100$
최적경로	0→10→4→7→6→1→5→2→3→8→9→0	0→1→2→3→4→5→6→7→8→9→10→0	
평균	221.2	229.0	3.53
최소	166.9	221.2	32.53
최대	260.7	234.4	10.08
분산	1051.8	22.05	97.90

[표 4]의 결과를 살펴보면, 평균만을 고려한 경우에 비해서 평균과 표준편차를 모두 고려한 경우가 평균은 다소 높지만 최대-최소 편차가 매우 작고 분산도 작아서 매우 안정적인 경로임을 알 수 있다. 다시 말해, 본 연구에서 제시한 알고리즘을 사용하면 시간치가 확률적으로 주어지는 경우의 TSP에서 평균 뿐 만 아니라 분산도 최소화되는 매우 안정적인 경로를 얻을 수 있음을 알 수 있다. 따라서, 이렇게 얻어진 경로는 실제 물류문제의 응용에 있어서 차량의 경로 스케쥴 작성 시 매우 신뢰성이 높은 자료로 사용될 수 있을 것으로 기대된다.

5. 결론

본 연구에서는, 시간자료가 확률적으로 주어지는 경우의 Q학습과 NN을 사용한 TSP의 알고리즘을 새롭게 제시하였다. 실험을 통해서 제시된 알고리즘이 확정적 및 확률적인 시간자료를 가지는 TSP에서 모두 최적해에 수렴함을 알 수 있었다. 특히, 본 연구의 알고리즘은 시간치가 확률적으로 주어지는 경우에서 평균 및 분산을 동시에 최소화하여, 분산이 매우 작은 안정적인 경로를 발견할 수 있음을 알 수 있었다.

참고문헌

1. Gendreau, M., Laporte, G. and Seguin, R., "Stochasitc vehicle routing," European Journal of Operational Research, Vol.88, pp.3-12, 1996.
2. Kao, E. P. C., "A Preference Order Dynamic Program for a Stochastic Traveling Salesman Problem," Operations Research, Vol.26, No.6, pp.1033-1045, 1978.
3. Kaelbling, L. P., Littman, M. L. and Moore, A. W., "Reinforcement Learning: A Survey," Journal of Artificial Intelligence Research, Vol.4, 1996.
4. 萩原将文, ニューロ・ファジィ・遺伝的アルゴリズム. 産業図書, 1994.
5. 이경모, 트랜스퍼 크레인의 반입 및 반출 작업 순서 결정법. 부산대학교 대학원, 1999.