

⊗ 연구논문

외판원문제에 대한 효율적인
새로운 경험적 방법 개발
**A New Heuristic Algorithm for
Traveling Salesman Problems**

백 시 현
Paik, Si Hyun*
김 내 현
Kim, Nae Hyun**

ABSTRACT

The TSP(Traveling Salesman Problem) is one of the most widely studied problems in combinatorial optimization. The most common interpretation of TSP is finding a shortest Hamiltonian tour of all cities.

The objective of this paper proposes a new heuristic algorithm MCH(Multi-Convex hulls Heuristic). MCH is a algorithm for finding good approximate solutions to practical TSP. The MCH algorithm is using the characteristics of the optimal tour. The performance results of MCH algorithm are superior to others algorithms (NNH, CCA) in CPU time.

1. 서론

공학적인 문제에서 최적화의 문제가 포함되지 않은 경우는 거의 없다. 정보검색, 제조업, 전파, 신소재, 우주산업 등 많은 영역에서 최적화에 대한 연구가 진행되고 있는지 혹은 이미 완성된 연구를 기반으로 하여 또 다른 새로운 연구가 이루어지고 있다. 예를 들어, 외판원문제(Traveling Salesman Problems)와 차량경로문제(Vehicle routing Problems)의 최단경로를 찾는 연구는 많은 연구자들이 심혈을 기울여 연구하는 하는 단골 문제이다. 이러한 문제는 공학보다도 과학으로 보는 경향이 있을 정도로 원론적이며, 많은 실제 상황에 적용되고 있으며, 새로운 경험적(Heuristic)방법이 개발될 때마다 언제나 평가의 대상이 되곤 한다.

이와 같은 경로문제는 많은 설비, 시스템들의 효율에 응용되고 있어, 생산성 향상을 위한 좋은 관리 방법의 요소가 되고 있다. 제조산업의 PCB의 천공과 부품 삽입문제, AGV(Automatic Guide Vehicle) 경로문제, 자동창고의 상·하적(loading/unloading)순서 등과 유통산업에서의 수배송의 경로설정 등이 좋은 예이다. 특히 교통개발연구원에서 발표한 국가물류비 현황에 따르면, 96년 우리나라 물류비 중 수송비가 차지하는 비용이 42조 4천억에 달해 총 물류비의 66.5%를 차지하고 있다. 따라서 기업과 국가의 경쟁력을 높이기 위해서, 효율적인 경로를 설정하는 것

* 아주대학교 산업공학과 대학원

** 아주대학교 산업공학과

이 매우 중요한 일이라고 생각되며, 본 연구와도 깊은 관계가 있다. 본 연구에서는 차량의 수 배송이나, PCB의 천공순서와 같은 문제를 더 빨리 효율적으로 해결할 수 있도록 새로운 경험적 방법을 제시하여 물류관리 및 생산성 향상에 기여하고자 한다.

우선 2장에서는 적용 대상 문제인 외판원문제의 관련연구에 대한 전반적인 소개를 하였고, 3장에서는 좋은 경로를 구성하기 위한 새로운 경험적 방법(Multi-Convex hulls Heuristic ; MCH)을 제시하고 MCH의 원리에 대해서 설명하였다. 4장에서는 새롭게 제시한 MCH의 성능을 다른 기존의 경험적 방법들과 비교, 평가하였으며, 5장에서는 결론과 앞으로의 연구방향에 대해 서술하였다.

2. 연구동향

경로문제는 많은 최적화 연구자들에게 인기 있는 유명한 문제로써, 한 차량이 여러 도시를 방문하는 외판원문제(Traveling Salesman problem)와 여러 차량들로 수배송을 하는 차량경로문제(Vehicle Routing Problem)가 유명하다. 'TSP(Traveling Salesman Problem)'라는 용어가 처음 소개된 것은, 1954년 'the Journal of the Operations Research Society of America'의 잡지에 실린 Dantig, Fulkerson and Johnson의 논문이라고 밝히고 있다[8]. 그 후에 1958년 Croes가 k-exchange를 이용해 TSP를 푸는 등, 많은 연구자들이 좋은 경험적 방법들을 개발하였다[9]. 기타 많은 연구들이 있지만, 좀 더 자세한 내용을 위해 Lawler[8]와 Reinelt[12]의 책을 추천하고 싶다. 또한 백시현, 김내현[1]에도 외판원문제에 대한 전반적인 소개를 하였다.

외판원문제에 대한 많은 연구가 이루어져 왔고, 이에 따른 많은 종류의 외판원문제들이 생겼다. 외판원문제를 특성별로 구분하여 이해하기 쉽게 다음의 <그림1>과 같이 정리하였다. 외판원 문제는 환경의 제약들에 따라, Bottleneck 외판원문제, Time Windows를 갖는 외판원문제, Stochastic 외판원문제 등으로 나누어질 수 있다. 또한 거리의 특성별로 구분해서 알아보면,

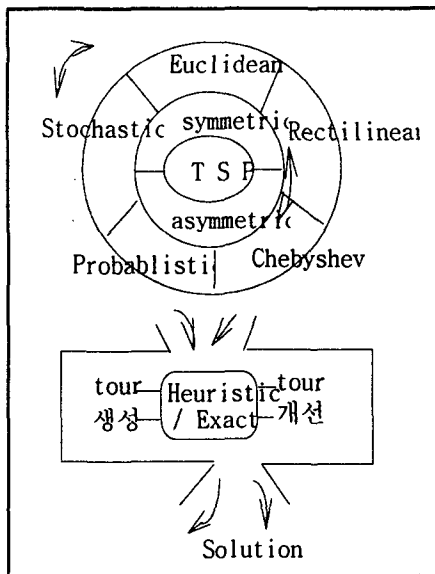


그림 1 외판원 문제의 특성별 구분

Barachet[3], Norback and Love[10], Papadimitriou [11], Su, Wang, Egbelu and Cannon [13]은 유클리디안 (euclidean) 거리를 가진 외판원문제를 다루었고, Allison and Noga [2]는 렉티리니어(rectilinear) 거리를 가진 외판원문제를, Goetschalckx[6]과 Bozer, Schorn and Shapr[4]은 체비세프(chebyshev) 거리를 가진 외판원문제에 대해서 연구하였다. 또한 백시현, 김내현[1]의 논문에서는 일반적으로 경험적 방법들은 주로 유클리디안 거리를 가지고 개발되었는데, 이러한 경험적 방법들이 다른 종류의 거리(렉티리니어 거리와 체비세프 거리)에서도 좋은 성능을 갖는지를 알기 위해 실험과 검정을 해 본 결과, 거리에 상관없이 경험적 방법들이 어느 종류의 거리(유클리디안, 렉티리니어, 체비세프 거리)에서나 동일한 성능을 가짐을 알 수 있었다.

본 절에서는 본 논문과 비교할 기존의 몇 가지의 외판원문제 알고리즘들에 대해서 소개하였다. Golden

and Stewart가 개발한 CCAO[9]는 가장 좋은 2-phase method 방법들 중 하나로써, convex hull을 이용해 초기 경로를 구하고, Cheapest insertion 할 수 있는 점들을 찾고, 초기경로와 삽입할 도시 사이에서 최대 각(Largest Angle)을 이루는 도시를 찾아 부분경로에 삽입하고, 생성된 하나의 완전경로를 Or-OPT post optimization 절차를 사용하여 개선시키는 방법이다. 이 방법은 기존의 다른 경험적 방법보다 성능이 우수하다고 밝혔다[9]. 또한 성능이 비슷한 GENIUS는 Gendreau, Hertz and Laporte[5]가 개발한 알고리즘으로 일반적인 삽입 방법과는 달리, 반드시 이웃한 두 점 사이에 삽입할 필요가 없는 정점을 GENERALIZATION Insertion Procedure를 이용하여 삽입한 후, Unstringing and Stringing 절차를 이용해 다시 제거하여 다시 삽입한 post optimization algorithm을 수행하였는데, 이 방법에서는 삽입대상이 모든 점들에서 각각의 p개의 이웃한 점들(neighbors)을 구해 계산 횟수를 줄였다. 또한 CCAO와 GENIUS의 성능을 비교해 본 결과 성능에 유의한 차이가 없음을 시뮬레이션을 통해 Gendreau et.al.[5]가 밝혔다.

근래에는 대부분의 경험적 방법들의 단점인 극소(local)해에 빠지는 것을 해결하기 위해, 자연현상(적자생존, 자연선택설)속에서 일어나는 유전자의 생성, 교배, 변형을 모델로 한 유전자 알고리즘(Genetic algorithms)과 온도(매개변수)를 통해 얻은 해에 확률 값을 부여해 전역(global)해를 구해 나가는 모의 단조(Simulated Annealing) 알고리즘에 대한 연구가 많이 이루어지고 있다. 또한 유전자 알고리즘의 탐색능력(Exploration)과 기존의 경험적 방법들의 극대 혹은 극소 값을 찾는 능력(Exploitation)을 잘 조합한 혼합형(Hybrid) 알고리즘들이 많이 개발되고 있다.

3. 새로운 경로 구성방법

경로생성의 중요성질

- [성질 1] 모든 최적경로는 꼬이지 않는다
- [성질 2] 최적경로는 부분적 경로를 구성하는 Convex hull의 순서를 그대로 유지한다

기존의 많은 경험적 방법들의 접근 방식을 분석하여 보면, 주로 좋은 초기해를 구성하는 방법에 관한 연구, 좋은 해를 구성하도록 삽입하는 방법에 관한 연구, 경로를 개선시키는 방법에 관한 연구에 초점을 맞추고 있다. 본 연구에서 제시하는 방법은 좋은 경로를 구성할 수 있도록 새로운 삽입방법에 관한 연구이다. 최근에 convex hull의 개념을 이용한 연구가 Takenaka and

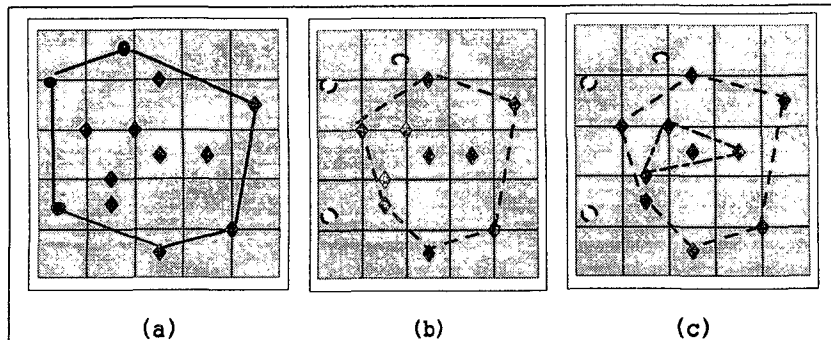


그림 2. Convex hull과 삽입점과의 관계

Funabiki [14]에 이루어졌는데, convex hull을 이용하여 방문할 수 있는 경로의 수를 제한하여 유전자 알고리즘의 성능을 개선시켰다. 본 논문에서 새롭게 제시한 알고리즘인 MCH도 Convex hull들의 성질을 이용한 방법으로 기존의 방법들보다 비슷한 성능을 유지하면서 속도는 월등히 우수하다. MCH의 특징은, 그림2의 (a)와 같이 부분경로에 아직 추가되지 않은 점들의 삽입에 있다. 기존의 경험적 방법들은 내부의 점들을 삽입시 모든 점들의 절약되는 거리(Saving distance)를 구하여 가장 많이 절약되는 부분에 삽입한다. 그리하여 많은 계산시간이 소요된다. MCH는 내부점을 삽입시, 좀 더 타당한 방법으로 빠른 시간내에 삽입하여 좋은 경로를 구성하는 방법이다. MCH의 성능을 이론적으로 설명하기 위해 표1의 기준에 밝혀진 다음의 두 가지 성질들을 이용해 그 타당성을 보였다.

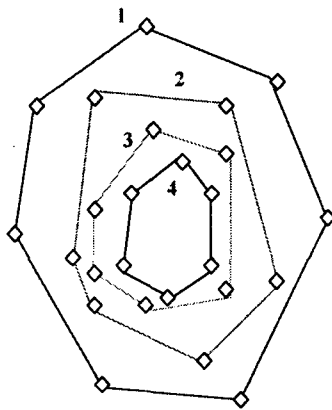


그림 3. 삽입을 위해 Convex hull들로 구성

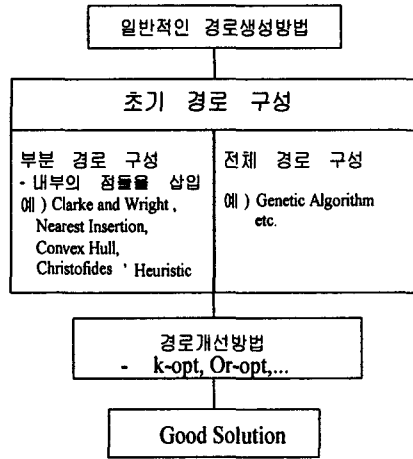


그림 4 전통적인 경로생성 경험적방법

3.1 경로생성 원리

일반적으로 앞의 그림 2와 같이 최단 경로문제를 풀기 위한 경험적 방법들은, 부분경로 생성하고, 하나의 완전경로를 구성하고, 구성된 경로를 개선시켜 나간다. MCH알고리즘도 이러한 전통적인 방법을 따르지만, 내부 점들을 삽입시 다음과 같은 원리를 이용해 빠른 시간내에 좋은 경로를 구한다.

앞의 그림 2는 새롭게 제시하는 알고리즘의 특징을 알아보기 위해 임의로 만든 최단 경로문제이다. (a)는 전체 문제에서의 convex hull을 이룬 부분경로를 나타내고 있다. 즉 최적경로는 convex hull을 이루는 점들의 순서를 유지하면서 내부의 점들이 각각의 점들 사이에 삽입되어 이루어진다. 결국 최적경로는 convex hull을 이루고 있는 점들의 순서는 변하지 않는다. 만일 convex hull을 이루는 점들 중에서 (b)와 같이 3개의 점들이 없다고 가정하자. 그러면, 이 문제의 convex hull은 (b)의 점선과 같게 된다. 즉 (b)에서의 최적경로는 점선을 이루고 있는 점들의 순서를 유지하게 된다. 이때 (b)에서 무시되었던 3개의 점들을 다시 삽입하고 싶을 때, 좋은 경로를 만들기 위해 어느 곳에 삽입하는 것이 좋은가? (c)를 살펴보자. 외부의 세 개의 점들은, 바로 이웃한 점선으로 된 convex hull사이에 삽입될 가능성이 높을 것이다. 왜냐하면 안의 convex hull과 연결된다면, 꼬이는 부분이 생기기 때문이다. 그러므로 최적경로를 이루고 있는

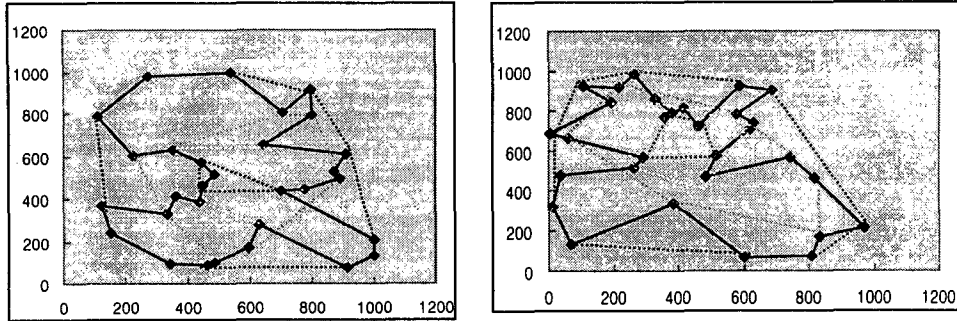


그림 5 최적경로와 Convex hulls과의 관계

발견된 최적경로의 경향

- [추가 경향] 최적경로를 이루고 있는 인접한 점들은 적어도 한쪽은
- 같은 Convex hull 을 이루는 점과 연결되거나 ,
 - 바로 이웃한 Convex hull 에 연결되어 있다 .

대부분의 점들은, 그림 3과 같이 자신이 포함되어 있는 convex hull이나 바로 이웃한 convex hull에 연결되는 경향이 있다. 여기서는 '성질'이라고 말보다 '경향'이라고 언급한 이유는 아직 수리적으로 증명된 것이 아니고, 단지 실험을 통해 얻은 결론이기 때문에 여운을 남겼다. 하지만, 수십개의 실제 데이터와 공개데이터, 일정하게(Uniform) 난수로 만든 데이터들을 가지고 시뮬레이션 하여 본 결과, 모든 문제의 모든 점들이 바로 이웃한 Convex hull에 연결되는 경향을 만족하였다. 그림 5는 위에서 제시한 최적 경로의 경향을 알아보기 위해 최적해와 Convex hull 들 사이의 관계를 나타낸 그림이다.

3.2 MCH 알고리즘

표 1 MCH 알고리즘

- [초기단계] Convex hull을 이용해 부분경로를 구성한다
- [삽입단계]
1. 내부 점들을 삽입시키기 위해 내부 점들을 구성하는 convex hull들을 찾는다 (그림 3 참조)
 2. 내부점을 삽입 시 가장 인접한 Convex hull들상에 놓여 있는 점들만을 고려해 절약값이 가장 많이 되는 위치와 점을 찾아 삽입한다(그림2참조)
 3. 완전한 경로가 될 때까지 반복
- [개선단계] 기존의 개선시키는 알고리즘(k-Opt)을 이용해 개선 시킨다

본 연구에서는 삽입 단계에서, 모든 점들을 비교 분석하는 것이 아니라, 이웃한 Convex hull 들을 구성하는 점들만을 비교하여 결정하였기 때문에 시간적인 측면에서 매우 우수하였다. 즉 MCH알고리즘은 앞의 그림 3과 같이 모든 점들이 Convex hulls을 구성하도록 한 후, 삽입할 경우에는 이웃한 Convex hull상에 있는 점들만을 비교하여 가장 적은 값의 saving을 갖는 점(가장 적은 거리로 삽입되는 점)을 삽입하는 방법이다. 즉 MCH알고리즘은, CCAO의 삽입단계에서

위에서 새로 찾은 최적경로의 경향을 이용하여 만든 알고리즘이다.

4. 실험 결과

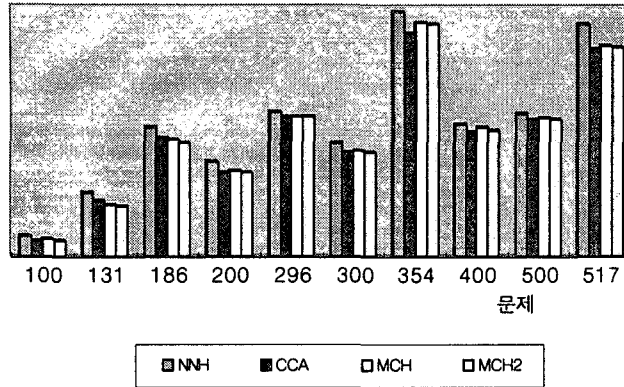
본 논문에서 새롭게 제시한 MCH알고리즘을 이용하여 여러 문제를 풀어보고, 이를 다른 경험적 알고리즘과 비교하였다. 본 연구에서 비교, 평가할 알고리즘은 가장 간단한 NNH(Nearest Neighbor Heuristic)와 좋은 경로를 구성하는 CCA(Convex hull Cheapest insertion, largest Angle) 알고리즘과 본 논문에서 제시한 MCH를 비교하여 보았다. 이때 알고리즘의 성능을 알아보기 위해 NNH, CCAO, MCH에서 개선단계는 생략하였다. 다음의 알고리즘들은 Pentium 166에서 C++로 코딩하였다. 위의 표2에서 선정된 문제들은 S전자회사의 실제 PCB의 천공데이터 5개와 랜덤데이터 4개, 그리고 Krolak[7]의 공개 데이터 1개를 가지고 4가지의 경험적 방법들을 이용하여 시뮬레이션을 하였다. MCH는 삼입단계에서 이웃한 1개의 convex hull상에 있는 점들만을 고려한 것이고, MCH2는 2개의 이웃한 convex hulls상에 있는 점들을 고려한 것이다.

CCAO는 CCA로 문제를 푼 다음 Or-opt를 이용하여 개선을 하여야하지만, 본 실험에서는 개선절차를 생략하였다. 왜냐하면 개선절차로 인해 성능이 좋아질 수도 있으므로 MCH알고리즘의 의한 성능과 교락(confound -ing) 되어지므로 MCH의 성능을 정확히 알 수 없게 된다. 그래서 CCAO와 MCH 알고리즘의 개선 절차는 생략하였다.

표2와 그림5를 살펴보면, 이미 알려진 바와 같이 NNH는 매우 간단한 알고리즘이기 때문에 시간이 가장 짧았으나, 거리는 10%이상 차이가 있어 좋은 경로를 구하지 못하였다. 하지만 CCA와 새롭게 개발한 MCH, MCH2를 비교·분석하여 보면, CCA가 MCH보다는 약간 우수하

표2. 성능 비교

문제 방법		Pr100 [®]	Pr131 [#]	Pr186 [*]	Pr200 [^]	Pr296 [#]	Pr300 [^]	Pr354 [#]	Pr400 [^]	Pr500 [^]	Pr517 [#]
		NNH	거리	27772	89312	180616	132477	202556	158087	341879	185251
NNH	시간	51.6	73.8	89.9	86.6	117.9	143.9	162.8	290.3	342.9	324.0
CCA	거리	23011	78091	164930	116630	195278	145452	311779	174689	190763	290496
CCA	시간	1217.2	4562.3	8171.9	9986.3	30511.2	34258.6	54299.1	79450.6	153880.7	166434.4
MCH	거리	23611 (-2.6%)	72002 (7.8%)	162545 (1.4%)	118810 (-1.8%)	194855 (0.2%)	147614 (-1.5%)	327403 (-4.8%)	179458 (-2.7%)	192882 (-1.1%)	294593 (-1.4%)
	시간	238.0 (5.1배)	1392.7 (3.3배)	925.6 (8.8배)	1131.2 (8.8배)	4486.1 (6.8배)	2734.1 (12.5배)	4135.4 (13.1배)	5259.2 (14.1배)	8772.8 (16.5배)	9822.7 (16.9배)
MCH2	거리	22690 (1.4%)	69283 (1.3%)	158238 (4.1%)	117594 (-0.1%)	194658 (0.3%)	144802 (0.4%)	325048 (-4.3%)	175879 (-0.7%)	192003 (-0.7%)	292576 (-0.7%)
	시간	362.0 (3.4배)	1705.8 (2.7 배)	1439.4 (5.7 배)	1875.3 (5.3 배)	5528.4 (5.5 배)	4166.1 (8.2 배)	6190.1 (8.8 배)	8484.7 (9.4 배)	13514.2 (11.4 배)	14342.9 (11.6 배)



<그림 5> 경험적 방법들과 MCH의 성능 분석

였으나, MCH2와는 유의한 차이가 없었다. 하지만 속도면에서는 CCA보다 MCH가 약 10배, MCH2가 약 7배 이상이 빨랐고, 문제의 크기가 커질수록 더 큰 속도의 차이가 생겼는데, MCH2인 경우와의 속도는 대략 $(1.51 + 0.02 * \text{도시의 개수})$ 의 상관관계를 갖고 있었고, 더 정확한 상관모형은 도시의 개수가 증가할때마다 속도는 지수적으로 향상된다. 즉 100개의 도시인 경우에는 약 3.5배가 빠르고, 500개의 도시인 경우는 약 11.5배가 빠르다.

그러므로 MCH2는 CCA와의 성능과 비슷하면서 수행 속도는 문제크기와 선형의 관계를 가지며 증가함을 알 수 있었다. 즉 새로 제시한 MCH2는 CCA의 좋은 성능을 유지하면서 더 빨리 얻을 수 있었다.

5. 결론

본 연구에서는 차량의 수배송이나, 전자산업의 꽃인 PCB의 천공순서와 같은 문제를 해결하기 위한 새로운 경험적 방법을 개발하였다.

그 동안의 많은 좋은 경험적 방법들이 개발되어 왔고, NP-hard에 속하는 문제이므로 좋은 경로의 구하기 위한 시간과 경로의 질(Quality)을 향상시키기 위한 연구에 초점을 맞추어 왔다. 새로 개발한 MCH 알고리즘에 사용된 최적경로의 새로운 경향에 관해서는 수리적으로 증명할 수 없지만, 기존의 밝혀진 두 가지의 좋은 성질을 이용하여 MCH 알고리즘의 우수성을 보일 수 있었고, 많은 문제들을 시뮬레이션하여 새로 제시한 알고리즘의 우수성을 확인할 수 있었다.

시뮬레이션 결과, 이웃한 convex hull을 우선 고려해 삽입하는 MCH과 MCH2 (Multi-Convex hull Heuristic) 알고리즘은 기존의 우수한 경험적 방법중 하나인 CCA와 비슷한 성능을 가지고 있었고, 속도면에서는 $(1.51 + 0.02 * \text{도시의 개수})$ 배나 빠른 상관관계를 갖고 있었고, 더 정확한 상관모형은 도시의 개수가 증가할때마다 속도는 지수적으로 향상된다. 즉 100개의 도시인 경우에는 약 3.5배정도 빠르고, 500개의 도시인 경우는 약 11.5배나 빠른 결과를 얻을 수 있었다. 즉 좋은 경로를 얻는 동시에 시간적인 면에서는 도시의 개수가 커질수록 MCH 알고리즘의 성능은 더욱 좋아진다.

앞으로의 과제로는, 본 알고리즘에 적용된 새로운 경향의 타당성을 보이기 위해, 수리적으로 최적해와 MCH와의 관계를 증명할 필요가 있으며, 계속해서 최적해가 가지고 있는 많은 좋은

성질(규칙)들을 발견하여 좀 더 좋은 경험적 알고리즘을 개발이 필요하겠고, 요사이의 유전자 알고리즘으로 최적경로문제를 풀기 위한 많은 연구가 이루어지고 있는데, 이러한 중요한 성질들을 유전자 알고리즘에 잘 반영시킨다면 더 좋은 성능을 가질 수 있을 것으로 기대된다.

參考文獻

- [1] 백시현, 김내헌, "회로기판 생산에서의 대형 외판원문제를 위한 경험적 해법 응용", 공업경영학회지. 20(41), pp.177-188, 1997
- [2] Allison and Noga, "The Rectilinear Traveling Salesman Problem", Information Processing Letters, Vol.18, No.4, pp.195-199, 1984
- [3] Barachet, "Graphic Solution of the Traveling Salesman Problem", Oper. Res., Vol.5, pp.841-845, 1957
- [4] Bozer, Schorn and Shapr, "Geometric Approaches to Solve the Chebyshev Traveling Salesman Problem", IIE Trans., Vol.22, No.3, pp.238-254, 1990
- [5] Gendreau, Hertz and Laporte, "New Insertion and Post-optimization Procedures for The Traveling Salesman Problem", Oper. Res., Vol.40, No.6, pp.1086-1094, 1992
- [6] Goetschalckx, "Storage and Retrieval Policies for Efficient Order Picking", Unpublished Ph.D. dissertation, Georgia Institute of Technology, Atlanta, GA, 1983
- [7] Krolak, Felts "A Man-Machine Approach Toward Solving the TSP", Communications, Vol.14, ACM No.5, May, pp. 327-334, 1971
- [8] Lawler, Lenstra, Rinnooy Kan, and Shmoys, "The Traveling Salesman Problems", Wiley-International publication, pp.1-6, 1985
- [9] Lawler, Lenstra, Rinnooy Kan, and Shmoys, "The Traveling Salesman Problems", Wiley-International publication, pp.207-249, 1985
- [10] Norback and Love, "Geometric approaches to solving the traveling salesman problem", Manag.Sci., 23, pp.1208-1223, 1977
- [11] Papadimitriou, "The Euclidean Traveling Salesman Problem is NP-Complete", Theoretical Computer Science, Vol.4, No.3, pp.237-244, 1977
- [12] Reinelt, "Lecture Notes in Computer Science 840; The Traveling Salesman - Computational solutions for TSP Applications", Springer-Verlag, 1994
- [13] Su, Wang, Egbulu and Cannon, "A dynamic point specification approach to sequencing robot moves for PCB assembly", Int. J. Comput. Integ. Manuf., Vol.8, No.6, pp. 448-456, 1995
- [14] Takenaka and Funabiki, "An Improved Genetic Algorithm Using the Convex Hull for Traveling Salesman Problem", Proc. IEEE Inter. Conf. on System, Man and Cybernetics, Vol.3, pp.2279-2284, 1998