

TSP 경로탐색을 위한 S-MINE 알고리즘

황 숙 희[†] · 원 일 용^{**} · 고 성 범^{***} · 이 창 훈^{****}

요 약

메타 휴리스틱 알고리즘을 이용해 TSP (Traveling Salesman Problem) 문제를 풀고자 하는 많은 시도가 이루어지고 있다. TSP 문제는 대표적인 NP-Hard 문제로 탐색 알고리즘이나 최적화 알고리즘을 실험하는데 많이 사용되고 있으며, 복잡한 사회의 많은 문제들의 표준 모델로 제시되고 있다. 본 논문에서는 2009년 제안된 MINE 알고리즘을 TSP 에 적용시켜 메타 휴리스틱 알고리즘으로서의 탐색성능을 알아보고자 하였다. 이에 S-MINE (Search - MINE) 알고리즘을 제안하였으며, TSP 에 적용하여 그 결과를 고찰하였다.

키워드 : 메타 휴리스틱 알고리즘, 탐색 알고리즘, 순회판매원문제

S-MINE Algorithm for the TSP

SOOK-HI HWANG[†] · IL-YONG WEON^{**} · SUNG-BUM KO^{***} · CHANG-HOON LEE^{****}

ABSTRACT

There are a lot of people trying to solve the Traveling Salesman Problem (TSP) by using the Meta Heuristic Algorithms. TSP is an NP-Hard problem, and is used in testing search algorithms and optimization algorithms. Also TSP is one of the models of social problems. Many methods are proposed like Hybrid methods and Custom-built methods in Meta Heuristic. In this paper, we propose the S-MINE Algorithm to use the MINE Algorithm introduced in 2009 on the TSP.

Keywords : Meta Heuristic Algorithm, Search Algorithm, Traveling Salesman Problem

1. 서 론

TSP(Traveling Salesman Problem)는 주어진 N개의 City 들을 단 한번씩만 거쳐 출발지로 되돌아 오는 경로들 중 가장 작은 비용이 소요되는 경로를 찾는 문제이며, 고전적인 최적화 문제로 널리 알려져 있다[1]. 네트워크 최적화 문제, 공장에서의 공정처리 순서결정문제, 전력전송 경로설정문제 등 광범위한 문제를 TSP에 환원시켜 최적화 문제로 적용시킬 수 있다[2]. 이와 같이 다양한 응용분야의 실험에 널리 활용되는 TSP는 City개수 증가에 따라 해를 구하는 데에 소요되는 계산량이 기하급수적으로 증가하여 NP-Hard문제로 분류된다[2].

지금까지 이러한 TSP에서 최단경로를 찾기 위하여 다양한 최적화 알고리즘이 제안되었다[3][4][5]. 메타 휴리스틱 형태의 최적화 알고리즘은 제한된 공간과 시간에서 해결되

지 않는 문제를 해결하고자 많이 사용되고 있다[2]. 메타 휴리스틱은 특정문제가 갖는 정보에 영향을 받지 않는 수준에서 다양한 문제에 적용 가능한 기법이다[7]. 메타 휴리스틱의 대표적인 알고리즘에는 GA(Genetic Algorithm)[3], SA (Simulated Annealing)[24], ACO(Ant colony optimization) [5], PSO(Particle Swarm Optimization)[4] 등이 있다. 메타 휴리스틱 알고리즘은 자연현상이나 행동을 모델링하여 조합 최적화문제로 이해될 수 있는 현실의 문제들을 해결하고자 하였다[6]. 이들 알고리즘들은 각각 다른 특성이 있지만, 개념과 이론이 단순하고 해공간의 탐색성능이 우수하다는 공통점을 가지고 있다[7]. 특히 공학, 자연과학, 경영학, 사회과학 등에서의 최적화 분야나 의사결정 분야에서 폭넓게 응용되고 있다[6].

그러나 메타 휴리스틱 알고리즘은 각각의 알고리즘 특성에 의하여 문제의 유형에 따라 그 성능의 차이를 보이고 있다[6]. GA는 최종단계 탐색능력이 부족하여 알고리즘을 수렴시키는 부분에 많은 조작이 필요하다[8][3]. ACO는 TSP 처럼 최단거리 문제로 환원되는 문제에서 특히 좋은 성능을 보인다[5]. 그러나 많은 최적해 탐색문제들은 그래프로 표현될 수는 있지만 최단거리 문제로 환원되지는 않는다. 따라

[†] 준 회원 : 건국대학교 컴퓨터공학과 박사과정 수료

^{**} 정 회원 : 서울호서전문대학교 사이버행정보안과 교수

^{***} 정 회원 : 공주대학교 컴퓨터공학부 교수

^{****} 종신회원 : 건국대학교 컴퓨터공학과 교수

논문접수: 2010년 11월 4일

수정일: 1차 2011년 1월 4일, 2차 2011년 1월 18일, 3차 2011년 2월 1일

심사완료: 2011년 2월 14일

서 응용 최적해 문제에서 좋은 성능을 내지 못하고 있다[8]. PSO는 다른 메타 휴리스틱 알고리즘들에 비해 수렴속도가 빠른 반면, 지역해에서 탈출하는 능력이 부족하다[4]. 이러한 단점들로 인하여 각각의 알고리즘마다 정의되는 문제들에 대한 낮은 범용성을 갖게 된다[8].

최근 메타 휴리스틱 알고리즘의 범용성을 높이기 위하여 두 개 이상의 알고리즘을 결합하는 하이브리드 방식이 제안되고 있다[9]. 알고리즘의 특성에 따라 적절처리 방식으로 두 알고리즘을 결합하는 약한 결합을 할 수 있는데, 약한 결합은 전처리 과정이나 데이터 처리 과정에서 한 부분을 결합시키는 결합방법이라 할 수 있다. 이러한 결합방식은 각각의 알고리즘의 특성을 잘 살리지 못하며, 주 알고리즘과 부 알고리즘으로 역할이 생기기 쉬워 한 알고리즘에 의존적으로 결합이 된다[9].

MINE 알고리즘은 다른 메타 휴리스틱 알고리즘들과는 달리 인간의 사회활동을 모델링 하였다. 여러 가지 인간의 사회활동 중에서 광산활동을 모델링 한 알고리즘이다[10]. MINE 알고리즘의 중요한 특징은 알고리즘 창발을 유도한다는 것이다. 다양한 기법들을 결합하거나 조정하는 별도의 장치 없이 자기 조직화 기능을 이용하여 목적을 달성하도록 한다[10]. MINE 알고리즘은 다른 메타 휴리스틱 알고리즘(GA, PSO, ACO, HC, RS)등 과 비교하기 위하여 3가지 표준문제 (Sphere function, De Jong I, De Jong II)에 알고리즘을 적용하여 실험하였으며[11][12][13], 실험결과 3가지 함수에서 좋은 성능을 나타냈다[10].

본 논문에서는 MINE 알고리즘을 확장하고, 알고리즘의 성능을 알아보기 위하여 문제의 복잡도를 높여 적용하였다. MINE 알고리즘은 세가지 표준 함수문제에 정형화되어 설계되어있기 때문에 MINE 알고리즘을 TSP 문제에 적용하려면 새로운 탐색 알고리즘이 필요하다. 따라서 새로운 평가 파라미터와 탐색 알고리즘을 포함한 S-MINE (Search-MINE) 알고리즘을 제안한다. S-MINE 알고리즘은 알고리즘 내부요소로 평가 파라미터와 조절 파라미터를 재정의 하였다. 또한 TSP에서 경로탐색을 위하여 새로운 탐색 알고리즘을 제안한다. 제안된 알고리즘을 TSP에 적용 실험하였으며, 그 결과를 고찰하였다.

본 논문에서 사용한 TSP 데이터 셋은 대표적인 TSP 예제로 알려진 TSPLIB에서 추출하였다[14]. TSPLIB 데이터 셋은 총 145개의 TSP 샘플 데이터 셋을 제공하고 있으며, 여러 가지 노드수와 데이터형식을 함께 제공하여, 다양한 문제를 실험할 수 있도록 하고 있다. TSPLIB에서는 각 TSP 샘플데이터 셋마다 지금까지 알려진 최적해 값이 함께 제공되고 있어 알고리즘의 실험 결과에 대한 평가 기준으로 사용하도록 하고 있다. 이 값은 지속적으로 갱신되고 있다.

본 논문은 2장 관련연구, 3장 S-MINE 알고리즘, 4장 실험 및 고찰, 5장 결론으로 구성하였다. 2장에서는 대표적인 메타 휴리스틱 알고리즘의 특징과 MINE 알고리즘을 살펴보고, 3장에서는 본 논문에서 제안하는 S-MINE 알고리즘의 특징과 구조, 탐색 알고리즘에 대하여 설명하였다. 4장에서

는 제안하는 S-MINE 알고리즘을 TSP에 적용 실험하여 결과를 제시하였다. 5장은 실험 결과를 고찰하여, 향후 발전 방향과 과제를 제시하였다.

2. 관련연구

2.1 메타 휴리스틱 알고리즘

GA(Genetic Algorithm), ACO(Ant colony optimization), PSO(Particle Swarm Optimization)은 대표적인 메타 휴리스틱 알고리즘이다. 각 알고리즘의 특징을 살펴보고 각각 TSP에 적용한 방법을 알아보려고 한다.

GA는 인공유전 시스템으로 자연세계의 진화과정에 기초한 계산모델로 John Holland에 의해서 1975년 개발된 전역적인 최적화 알고리즘이다[3]. GA는 풀고자 하는 문제에 대한 가능한 해들을 정해진 형태의 자료구조로 표현한 다음 이들을 점차적으로 변형함으로써 점점 더 좋은 해들을 만들어 낸다[15]. GA에서는 2진 형태로 개체를 표현하기 때문에 문제를 재정의하여 해를 표현하는 과정이 중요하다[16]. 따라서 GA에서는 '표현', '적응도', '유전연산자' 등의 요소들이 결정되어야 한다[17].

GA를 TSP에 적용하기 위해서는 적합도와 교차연산자나 전위연산자 등을 문제에 맞게 수정해야 한다. 특히 두 개의 방문순서를 가지고 단순교차연산을 했을 때 중복되는 도시나 방문을 하지 않은 도시가 생기게 될 수 있다. 따라서 TSP에 적합한 제약조건을 만족하는 연산자를 사용해야 한다[15]. 이와 같이 제약조건을 만족하는 알려진 GA 연산자로는 REVERSE, TRANSPORT, SWEEP, ORDER CROSSOVER, CYCLE CROSSOVER, PARTIALLY MATCHED CROSSOVER, EDGE RECOMBINATION 등의 연산자가 있다[3][16][15].

ACO는 그래프에서 최적의 경로를 찾기 위한 확률에 기반한 최적화 알고리즘이다[5]. ACO는 1992년 Marco Dorigo의 박사논문에서 처음으로 제안되었으며, 개미들이 먹이와 집 사이의 경로를 찾는 행동을 모델링 하며 만들어진 알고리즘이다[18]. 개미는 가능한 모든 해를 이동하면서 그 중 최적의 해를 찾게 된다[18]. 실제 개미는 서로 다른 개체에 개 영향을 줄 수 있는 페로몬을 개미 집단의 활동영역을 이동하면서 남긴다. 개미는 위치와 각 개체가 선택한 해의 최적 정도를 기록하고 반복과정에서 반복적으로 사용됨으로써 최적에 가까워 진다[5].

ACO는 TSP 문제에 적용시킬 때 좋은 예가 된다. '모든 도시는 연결되어 있다.', '모든 도시는 한번씩만 거쳐야 한다.', '모든 도시들을 방문한 뒤 제자리로 돌아온다.', 의 전제조건을 주고 여러 마리의 개미를 보낸다[19]. 탐색 가능한 예지들을 이용하여 경로를 생성하는 개미들이 최적해를 선택하기 위해서는 예지들의 페로몬 양에 대한 정보가 중요한 역할을 한다[5]. 따라서 예지의 페로몬 정보를 갱신하는 연구가 많이 이루어 지고 있다[18][19].

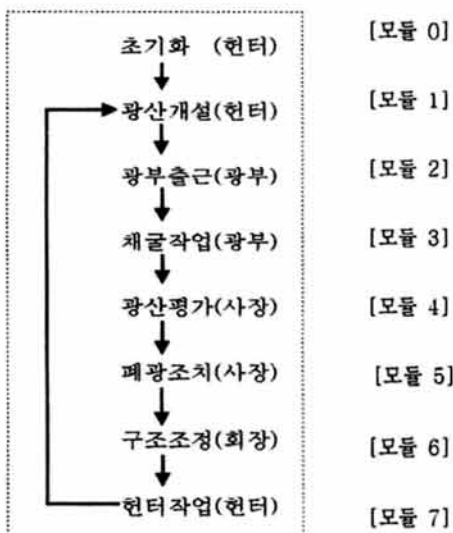
PSO는 1995년 J.Kennedy와 R.Eberhart에 의해 제안된

진화형 계산 기법의 일종으로, 새나 물고기 무리의 움직임에 관한 연구로부터 유도되었다[4]. 무리가 먹이를 찾아가는 과정에서 무리 전체가 정보를 공유한다는 가설과, 무리 내부의 파티클이 자신의 경험과 무리 전체에 공유되어 있는 정보를 기초로 하여 행동한다는 개념을 최적화 과정에 도입한 기법이다[20]. PSO는 알고리즘이 단순하고, 계산시간이 짧으며 대량의 메모리가 필요 없고, 연속형과 비연속형의 모든 문제에 적용 가능하다는 장점을 가지고 있다[21]. PSO는 GA등과 같이 복수의 탐색점을 가지며, 각 탐색점을 확률적으로 변경시켜가는 것에 의해 전역적인 최적해를 발견하는 방법이다[22]. 이러한 PSO를 TSP에 적용하기 위하여 GA 이론과 결합을 이루고 있다[4]. PSO의 변경량은 연속치로 표현된다. 따라서 연속형 변수를 포함하는 문제와 일반적인 이산형 변수를 포함하는 문제에 확장 적용된다. 이에 PSO에서는 개별 경로탐색 알고리즘을 삽입하는 방법으로 제안되고 있다[22].

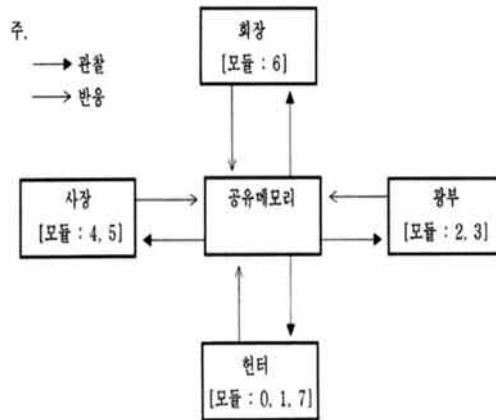
2.2 MINE 알고리즘

MINE 알고리즘은 광산 업무를 형식화한 메타 휴리스틱 알고리즘이다[10]. 시장경제에서 광산을 하나의 독립된 작은 시장으로 보고 구현하였다. MINE 알고리즘은 행위자 중심으로 돌아가도록 구조화 되었다. MINE 알고리즘은 회장, 사장, 광부, 헌터 등 4종류의 개념적 행위자를 두고 각각의 행위를 정의하였다[10].

MINE 알고리즘은 직렬 모드와 병렬 모드 두 가지 방법으로 모두 구축할 수 있다. 병렬모드의 경우 5개 이상의 프로세서가 지원되어야 하며, 이중 4개는 행위자들에게 배당하고, 나머지 하나는 공유 메모리로 사용하게 된다. 직렬 모드에서는 행위자보다는 프로세스 기준으로 배려했다. 병렬 모드에서는 일의 주기와 선후 관계를 경험적 설정이 가능하도록 하고 있다[10].



(그림 1) MINE 알고리즘의 직렬구조



(그림 2) MINE 알고리즘의 병렬구조

(그림 1)과 (그림 2)는 MINE 알고리즘의 직렬구조와 병렬구조를 보여준다. 병렬구조에서 사각형은 독립된 프로세서를 의미하여 공유메모리를 제외한 4개의 프로세서는 4가지 유형의 행위자에 대응하도록 하고 있다. MINE 알고리즘에서 행위자는 직렬구조의 모듈세트를 완전한 형태로 가지고 있도록 하였으며, 행위자는 8개의 모듈 중에서 일부 모듈만 선택적으로 수행하도록 하였다. MINE 알고리즘에서 행위자들은 주변상태를 관찰하고 반응하는 일을 반복적으로 수행한다[23].

MINE 알고리즘에서는 '성과'와 '인기도'를 이용하여 광산을 평가하였다. '성과'와 '인기도'는 PID 개념을 사용하여 정의하였다. PID의 의미는 다음과 같다.

- P (Proportional) : 비례요소이며 현 시점의 성과를 의미한다.
- I (Integral) : 적분요소이며 누적성과를 의미한다.
- D (Differential) : 미분요소이며 현 시점에서 본 성과의 개선 정도를 의미한다.

광산 m의 시점 t에서의 P값을 P(m,t)라 하고 D값과 I값을 구하면 다음과 같다.

$$D(m,t) = P(m,t) - P(m,t-1)$$

$$I(m,t) = \sum_{i=t-h}^t D(m,i) \quad (h \text{는 적분 항목의 개수})$$

광산 m의 시점 t에서의 광산레벨 ML(m,t)는 PID 개념을 이용해서 다음과 같이 정의된다.

$$ML(m,t) = \alpha(P(m,t) \times I(m,t)) + \beta(P(m,t) \times D(m,t))$$

이 식으로 보았을 때 PI와 PD는 상대적인 비중이 된다. 즉, 어떠한 일의 평가를 내릴 때 실력(P), 저력(I), 잠재력(D)을 종합적으로 고려한다[10].

자기조직화 모델에서는 환경의 변화와 거의 무관한 행위자들이 내부 계층에 있다. 이들은 환경의 변화를 의식하지 않고 어떤 정해진 행위를 반복한다. MINE 알고리즘에서 내부 계층은 알고리즘 구조에서 보았듯이 네 종류의 행위자들

〈표 1〉 MINE 알고리즘의 표준문제 적용 결과

모델	표준문제 I Sphere Function		표준문제 II De Jong I			표준문제 III De Jong II		
	평균(시간)	등수	평균(점수)	표준편차	등수	평균(점수)	표준편차	등수
MINE	1.3	3	99.97	0.04	1	16137	5382	1
PSO	0.82	1	87.16	2.86	5	4038	529	2
GA	0.828	2	98.61	0.58	3	356	99	5
ACO	15	6	96.57	0.48	4	78	19	6
HC	2.98	4	99.51	0.19	2	2214	969	3
RS	4.186	5	79.69	0.14	6	1366	553	4

이 독립적으로 일한다. 또한 제조적 계층에서 광산의 모든 현상들을 파악하여 광산의 상태를 파악하게 한다. MINE 알고리즘의 외부 계층에서는 시스템 차원의 적응능력을 보여 주고 있다. 내부계층의 정적이고 단순한 행위들이 제조적 계층에서 동적이고 복잡한 재 조직화 현상으로 표출되는 것은 자기조직화의 전형적인 특성이며, MINE 알고리즘은 이러한 자기조직화 구조를 가지고 있다[23].

2009년 제안된 MINE 알고리즘은 3가지 표준문제에 적용하여 그 결과를 기존의 5가지 메타 휴리스틱 알고리즘과 비교하여 평가를 하였다. 그 결과 MINE 알고리즘은 다른 메타 휴리스틱 알고리즘과 비교하여 개선된 범용성을 보였다. <표 1>은 MINE 알고리즘의 표준문제 적용 결과를 보이고 있다[10].

3. S-MINE 알고리즘

3.1 S-MINE 알고리즘의 특징

S-MINE 알고리즘은 MINE 알고리즘에 기반을 두고 있다. MINE 알고리즘은 인간의 사회활동 중에서 광산활동을 모델링 한 메타 휴리스틱 알고리즘이다. MINE 알고리즘은 3가지 표준문제에 정형화되어 개발되었기 때문에 TSP에 적용시키기 위해서는 TSP 경로탐색을 위한 새로운 탐색기법이 필요하다. 따라서 본 논문에서는 S-MINE (Search-MINE) 알고리즘을 제안한다.

S-MINE 알고리즘은 네 가지 개념적 특징을 가지고 있다.

- 첫째, S-MINE 알고리즘은 탐색중심을 이동하면서 탐색한다.
- 둘째, S-MINE 알고리즘은 탐색영역의 범위를 변화시키면서 탐색한다.
- 셋째, S-MINE 알고리즘은 탐색영역의 탐색량을 조절하며 탐색한다.
- 넷째, S-MINE 알고리즘은 독립적, 경쟁적으로 탐색한다.

S-MINE 알고리즘의 개념적 특징은 광산활동에서 채광작업의 특징을 알고리즘으로 표현하기 위하여 개념적으로 정의한 것이다. 광산에서 채광작업은 한곳에서만 하지 않으며

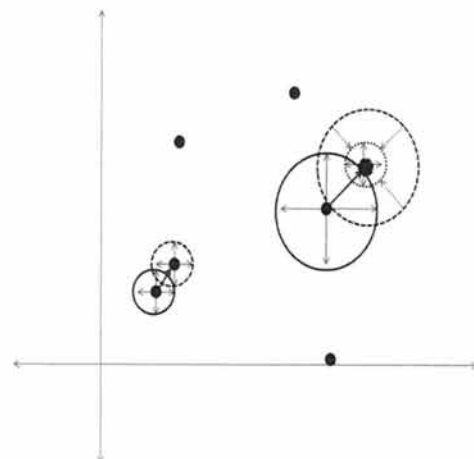
성과에 따라 채광위치를 변경한다. 따라서 S-MINE 알고리즘에서는 탐색중심을 이동할 수 있도록 설계하였다.

탐색영역의 크기는 이동범위와 관계가 있다. 즉, 성과가 좋은 광산의 이동범위는 작고, 성과가 좋지 않은 광산은 더 멀리 이동하여 새로운 채광위치를 찾는다. 따라서 S-MINE 알고리즘에서는 탐색영역의 범위에 따라 이동을 결정하도록 설계하였다.

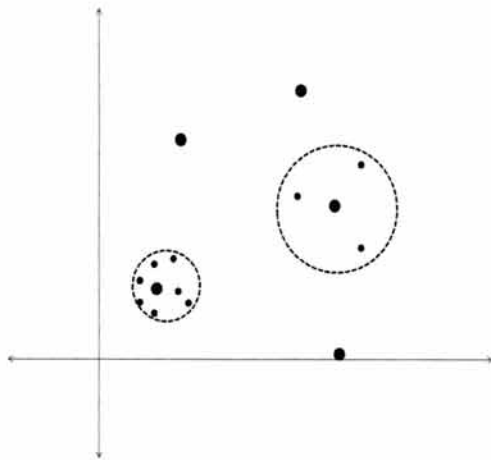
광산활동에서 광부들은 매일 원하는 광산을 선택할 수 있다. 즉, 성과가 좋은 광산에 더 많은 광부들이 가기를 원하며, 광부의 수가 많은 광산은 좋은 광산임을 알 수 있다. S-MINE 알고리즘에서는 광부의 수를 탐색영역의 탐색량으로 정의하여 광부의 수를 조절할 수 있도록 설계하였다.

채광작업에서 광부들은 서로 작업에 영향을 미치지 않는다. 즉, 독립적으로 채광작업을 하며, 경쟁관계에서 그 작업량을 평가 받게 된다. 따라서 S-MINE 알고리즘에서는 각각의 프로세스동작과 탐색과정에서 독립성을 유지시켰으며, 평가를 통하여 프로세스들이 경쟁상태를 유지하도록 설계하였다.

(그림 3)과 (그림 4)는 S-MINE의 개념적 특징을 그림으로 나타낸 것이다. 즉, 2차원 해공간에서 탐색중심의 이동과 탐색영역의 범위변화, 그리고 탐색영역의 탐색량을 그림으로 보여주고 있다.



(그림 3) 2차원 해공간에서 탐색영역의 이동과 탐색영역의 범위



(그림 4) 2차원 해공간에서 탐색영역의 탐색량

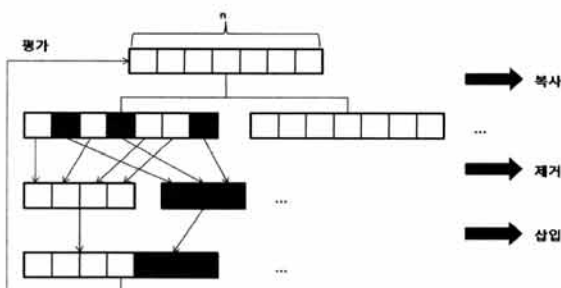
S-MINE 알고리즘에서는 위의 네 가지 개념적 특징을 구현하기 위하여 세션, 광산, 광부, 채광작업등 네 가지 요소를 정의하였다. 이 네 가지 요소를 TSP에 적용하여 재 정의하면 다음과 같다.

세션은 알고리즘수행을 나타내며, 탐색지점을 표현한 광산은 하나의 완성된 TSP경로이다. 광부는 새롭게 생성되는 TSP경로이며 알고리즘수행 프로세스로 정의한다. 채광작업은 실제 탐색과정을 나타내며 채광작업을 통하여 TSP 경로를 찾는다. 탐색과정은 3.2절에서 다시 설명하였다.

S-MINE 알고리즘에서는 변경률과 집중도 두 가지 평가값을 정의한다. 이 평가값은 탐색영역크기의 변화와 탐색영역의 탐색량을 조절하기 위한 평가요소이다. 즉, 변경률이 높으면 탐색영역이 확대되어 광산의 위치변화의 폭이 커지게 되며, 변경률이 낮으면 탐색영역이 좁아져 광산의 위치변화의 폭이 줄어들게 된다. 또한 집중도는 탐색량을 조절하는 평가요소이며, 집중도가 높으면 배당광부의 수가 늘어나 탐색량이 늘어나고, 집중도가 낮으면 배당광부의 수가 줄어 탐색량이 줄어든다. 변경률과 집중도는 3.3절에서 다시 설명하였다.

3.2 S-MINE 알고리즘 탐색과정

S-MINE 알고리즘에는 각 광산마다 광부들이 채굴하는 행위를 탐색과정으로 새롭게 모델링 하였다. S-MINE 알고리즘의 탐색과정은 다음과 같다. S-MINE 알고리즘의 탐색과정은 복사-제거-삽입-평가 의 과정을 반복한다. (그림 5) 은 S-MINE 알고리즘의 탐색과정을 그림으로 보이고 있다.



(그림 5) S-MINE 알고리즘의 탐색과정

복사 - 광부는 포함된 광산의 경로를 복사하여 탐색을 준비한다. 이때 복사되는 경로는 광산에서 항상 최적값을 가진 경로이므로 매번 복사과정을 거치게 된다.

제거 - 준비된 경로에서 광산의 변경률에 따라 변경할 노드의 수만큼 노드를 선택하고 삭제한다. 이때 할당되는 변경률은 한 세션에 한번씩 변경되며, 한 세션에서는 변경률이 유지되므로 변경할 노드의 수가 유지된다.

삽입 - 제거된 노드는 제거가 이루어진 경로의 마지막 노드 뒤에 삽입시킨다.

평가 - 노드가 삽입되어 새롭게 생성된 경로의 전체 웨이트를 구한다. 새로운 경로의 웨이트와 현재 광산의 1차 평가 후 최적 탐색경로의 웨이트를 비교하여 두 경로를 평가한다. 이때 새로운 경로의 웨이트가 기존의 최적탐색 경로의 웨이트보다 개선되었다면 업데이트하고 그렇지 않다면 현재경로를 유지한다.

탐색 알고리즘에서 복사된 경로들은 동시에 작업을 수행하며, 경쟁적으로 최적경로를 업데이트 한다. 따라서 복사 단계에서는 매번 광산의 최근 경로로 탐색경로를 복사를 해야만 한다.

3.3 변경률과 집중도

변경률과 집중도는 S-MINE 알고리즘에서 평가값과 조절값이다. 집중도와 변경률은 광산의 레벨값으로 계산된 비율값이다. 이때 사용되는 레벨값은 MINE 알고리즘의 PID 값을 이용하여 구하였다. 집중도와 변경률은 한 세션 안에서 광산의 변화를 분석하고 평가하는 값이다. 이 값은 다음 세션에 적용될 광산의 영역과 광부의 작업량이 할당되므로 변경률과 집중도는 광산의 탐색의 기회와 영역 확장 및 축소를 표현해 주는 수학적 요소이다.

광산 레벨값이 높을 때 집중도는 높게 나타나 광부가 많이 배당되어 현재 경로를 중심으로 탐색의 기회를 많이 주게 되고, 변경률은 낮게 나타나 변동의 폭을 좁혀 좋은 시점의 경로를 유지하면서 탐색하도록 하였다. 따라서 집중도와 변경률은 서로 반비례관계로 정의된다.

집중도를 C, 변경률을 S 라고 했을 때 각각은 식(2)와 식(3)과 같이 구할 수 있다. 식(1)은 광산의 레벨값을 비율값으로 재정의 한 식이다.

m개의 광산이 생성되었고, 각 광산의 경로의 수를 n, i 번째 광산의 레벨값을 ML_i , i번째 광산 레벨값의 비율을 $NewML_i$ 라고 했을 때 집중도 C와 변경률 S는 다음과 같이 구할 수 있다.

$$NewML_i = \frac{\sum_{j=1}^n ML_j}{ML_i} \tag{1}$$

$$C = rm \times \frac{NewML_i}{\sum_{j=1}^n NewML_j} \quad (2)$$

$$S = \left(n \times \frac{m}{c} \right) + 1 \quad (3)$$

식(2)에서 집중도 C 는 전체 레벨값에 대한 현재 레벨값의 비율로 결정하며, 전체 광산 개수의 r 배까지 허용하도록 하였다. 이때 사용되는 변수 r 은 광산수의 배수로 하여 광산의 탐색 기회를 높일 수 있도록 조절 하였다. 식(3)에서 변경률 S 는 집중도 C 의 역 비율을 적용 하였으며, 주어진 경로의 노드수를 넘지 않도록 하고, 한 세션에서 적어도 1개 이상의 경로를 제거 할 수 있도록 하였다. 경로 탐색 비율을 높이면서 변화율은 낮게 하도록 하기 위하여 C 와 S 는 반비례 관계를 갖도록 하였다.

식(2),(3)은 각 광산의 레벨값에 따라 할당되는 광산의 영역과 집중도를 표현해준다. 즉, 현재 광산의 레벨값이 높을 경우 집중도가 높아져 탐색의 기회를 많이 갖게 된다. 현재 광산의 경로를 집중도 만큼 복사되어 같은 경로를 여러 번 탐색할 수 있게 한다. 또 영역의 표현은 현재 광산의 경로에서 변경되는 범위를 결정해 준다. 즉, 변경할 노드의 수를 조절하여 경로의 변화에 영향을 줄 수 있도록 하였다. 이는 레벨값이 높은 광산의 경로는 최적경로 확률이 높다는 가정에 기인한다. S-MINE 알고리즘에서 사용한 MINE 알고리즘의 PID개념을 이용한 PID함수를 살펴보면 다음과 같다. 시점 i 에서 경로값을 라고 했을 때,

$$P = W_i \quad (4)$$

$$I = \sum_{i=1}^n (W_{i-1} - W_i) \quad (5)$$

$$D = \frac{I}{G}, \quad (G = \text{update 횟수}) \quad (6)$$

식(6)에서처럼 S-MINE 알고리즘에서는 개선 정도를 나타내는 D 를 I 값의 평균값으로 한다. D 는 개선 정도의 기울기와 개선속도를 나타낸다. 기울기에 따라 관리자 프로세스는 광산의 지속여부를 결정하며, 새로운 광산이 활성화 되도록 한다. 또한 일정 시간 동안 기울기 변화가 없을 경우 최적해로 결정하여 세션을 종료할 수 있도록 하였다.

식(5),(6)은 탐색알고리즘이 끝나고 각 광산의 상태를 평가할 때 사용한다. I 값은 탐색알고리즘이 수행되는 과정에서 생기는 변화량의 합을 나타내고 D 는 평균 변화량을 나타낸다. I 와 D 값은 현재 광산을 중심으로 탐색 알고리즘 수행에서 생성된 경로들의 모든 값을 가지고 개선된 정도를 계산하도록 하였다. I 와 D 는 상호 보완적 관계를 유지한다. 전체 변화량이 크더라도 평균 변화량이 작으면 탐색을 지속할 확률이 낮아질 수 있고, 전체 변화량은 적지만 평균 변화량이 크면 탐색을 지속할 확률이 높아질 수 있다.

4. 실험 및 결과

실험에 사용한 TSP 데이터 셋은 대표적인 TSP 예제로 알려진 TSPLIB에서 추출하여 사용하였다.

실험 TSP 데이터 셋은 'gr17', 'fri26', 'bays29', 'swiss42', 'hk48', 'brazil58' 등 총 6개를 선정하여 사용하였다. 비교 최적값의 기준은 TSPLIB에서 제공하는 TSPLIB Best known solution 으로 하였다. 본 실험에서는 S-MINE 알고리즘의 1차 탐색 성능을 평가하기 위하여 데이터 셋의 노드수를 100개 미만에서 다양하게 선정하였다. 노드를 17개 à 26개 à 29개 à 42개 à 48개 à 84개로 변화를 주어 선정하였고, 이는 각각 처음 노드수 대비 각각 52%, 70%, 147%, 182%, 241% 증가시킨 것이다. 실험을 통하여 S-MINE 알고리즘 탐색성능을 알아보았다.

TSP 문제는 노드수가 증가함에 따라 계산량이 기하급수적으로 증가하여 탐색시간에도 큰 영향을 미치게 된다. 따라서 알고리즘이 TSP 문제공간에서 얼마나 효율적으로 탐색하고 있는지 알아보기 위하여 노드수 증가에 따른 탐색시간 증가율을 알아보았다. 이때 외부 입력조건인 광산 수, 광부 수, 세션 수를 같은 조건으로 하여 노드수만 변경시켜 실험하였다.

4.1 TSP 데이터 셋 적용 실험

각 데이터 셋마다 광산 수, 광부 수, 세션 수를 조절하여, 각각 100회씩 반복실험 하였다. 광산 수는 알고리즘의 시작 조건에서 영역의 범위조건이며, 광부 수는 실제 알고리즘 수행 프로세스의 수로 광산 수의 8배, 10배로 조절할 수 있도록 하였다. 세션 수는 전체 알고리즘 수행 세대를 의미하며 외부조건으로 조절하여 실험할 수 있도록 하였다. 광산 수, 광부 수, 세션 수는 알고리즘을 수행할 때 외부조건으로 입력할 수 있도록 하였다. 본 실험은 S-MINE 알고리즘에서 찾은 경로값과 TSPLIB Best known solution값을 비교하여 알고리즘의 탐색성능을 평가하였다. 이때 S-MINE 알고리즘에서 탐색한 경로값이 TSPLIB Best known solution값에 얼마나 근접했는가를 알아보기 위해 TSPLIB Best known solution값과의 정확도를 계산, 분석하였다. 또한 실험결과에 따른 평균 정확도를 계산하여 알고리즘의 안정성과 최적도를 알아보았다. <표 2>는 각각의 데이터 셋에 대한 실험 결과를 보이고 있다.

이러한 실험과 분석은 S-MINE 알고리즘의 메타 휴리스틱 알고리즘적 특징을 알아볼 수 있고, 실제 실험 결과에 의하여 메타 휴리스틱 알고리즘으로써의 완성도를 알아보았다. 메타 휴리스틱 알고리즘의 해는 알고리즘 수행 시 탐색한 최적해를 말한다. 따라서 본 논문에서는 매 수행 때 마다 탐색하는 최적해의 변화량을 측정하여 오차범위를 분석하였다.

<표 2> S-MINE 알고리즘의 탐색 결과를 보면 gr17, fri26, bays29 데이터 셋에서는 탐색오차 10% 이내의 탐색률을 보였다. hk48 데이터 셋의 경우 탐색오차 12.43%를 보

〈표 2〉 S-MINE 알고리즘의 TSP 경로 탐색 결과

TSPLIB dataset	TSPLIB Best Known	S-MINE Best	S-MINE Average	Best Accuracy(%)	Average Accuracy(%)	Accuracy Distance
gr17	2085	2085	2172.5	100	95.81	4.19
fri26	937	956	1009.3	97.98	93.29	4.69
bays29	2020	2131	2183.2	94.51	91.92	2.59
swiss42	1273	1397	1421.7	90.26	88.32	1.94
hk48	11461	12886	12994	87.57	86.63	0.94
brazil58	25395	27086	27578.2	93.35	91.41	1.94

이고 있지만 평균오차범위가 +0.9%로 나타나 반복실험에서 안정적인 탐색 결과를 보였다. brazil58 데이터 셋의 경우 문제공간이 가장 크지만 탐색오차가 6.65%로 낮게 나타났다. 17개 노드로 구성된 gr17 데이터 셋 경우 TSPLIB Best known solution값과 동일한 경로값을 찾았고, fri26 데이터 셋의 경우 TSPLIB Best known solution값과 오차율 2%를 보이는 경로값을 찾았다. gr17과 fri26 데이터 셋에 적용한 탐색 결과 아주 높은 탐색율을 보이고 있으나 평균오차율과 4%정도 차이를 보이고 있어 약간 불안정한 탐색율을 나타내고 있다. bays29 데이터 셋과 swiss42 데이터 셋의 경우 TSPLIB Best known solution값과 오차율 5%, 9%를 보이고 있어 좋은 탐색율을 보이고 있고 평균오차율과도 각각 2%, 1%정도의 차이를 보여 안정적인 탐색율을 보이고 있다. 이러한 탐색의 안정성 차이는 세션수의 조절에 따른 변화로 보인다. 노드수 증가에 따라 세션수를 증가시켰을 때 높은 탐색율을 보였으며 평균오차율과의 차이도 낮아졌다.

노드수를 17개 대비 182%, 241% 증가시킨 hk48, brazil58 데이터 셋에서는 접근율이 3~4% 정도 낮아지고 있지만 이때에도 평균오차율과의 차이는 낮게 나타났다. hk48 데이터 셋의 경우 TSPLIB Best known solution값과의 오차율이 12%정도로 나타나 낮은 탐색율을 보이고 있지만, 평균오차율과의 차이가 0.9%로 반복실험에서 가장 안정적인 탐색율을 나타냈다.

정확도 분석을 위하여 신뢰도 95% 신뢰구간을 계산하였다. 표본개수 n은 반복실험수로 하였다. 표본의 표준편차를

σ , 평균을 X , 계수를 k 라고 했을 때 모평균 m 의 신뢰구간은 $X-k \leq m \leq X+k$ 이 되고, 신뢰구간의 폭은 $2k \frac{\sigma}{\sqrt{n}}$ 이다. 실험 데이터 셋 중에 gr17에 적용하여 신뢰구간을 분석하였다.

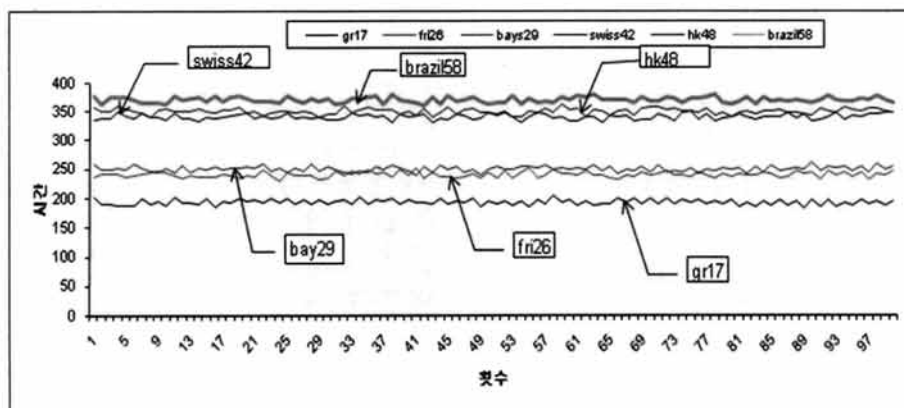
gr17 데이터 셋의 평균은 2172.54, 95% 신뢰구간이므로 계수 k 는 1.96, 따라서 gr17 데이터 셋의 신뢰구간은 $2172.54 - 1.96 \leq m \leq 2172.54 + 1.96 \equiv 2170.58 \leq m \leq 2174.5$ 가 된다. gr17의 표준편차 σ 는 15.2, 표본개수는 100, 이때의 신뢰구간의 폭은 다음과 같이 구할 수 있다.

$2k \frac{\sigma}{\sqrt{n}} = 2 \cdot 1.96 \cdot \frac{15.2}{\sqrt{100}} = 5.95$ 이고, 이 값은 gr17 데이터 셋의 경우 최대 허용오차 범위가 신뢰도 95% 수준에서 $\pm 5.95\%$ 라고 할 수 있다.

4.2 S-MINE 알고리즘 탐색시간 분석

TSP 문제는 노드수가 증가함에 따라 계산량이 기하급수적으로 증가하여 탐색시간에도 영향을 미치게 된다. 따라서 알고리즘이 TSP 문제공간에서 얼마나 효율적으로 탐색하고 있는지 알아보기 위하여 노드수 증가에 따른 탐색시간 증가율을 알아보았다.

실험조건은 외부 입력조건인 광산 수, 광부 수, 세션 수를 각각 30, 300, 40으로 동일하게 하였으며, 같은 조건에서 노드수만 변경시켜 실험하였다. 100회 반복 실험을 하였으며, 이때 탐색시간의 변화를 알아보았다. (그림 6)은 반복실험을 통한 각 데이터 셋의 탐색시간변화의 결과를 그래프로 나타내고 있다.



(그림 6) S-MINE 알고리즘 탐색시간변화 그래프

〈표 3〉 TSP 노드수 증가에 따른 S-MINE 알고리즘 탐색시간 증가율

TSPLIB dataset	gr17	fri26	bays29	swiss42	hk48	brazil58
Node No.	17	26	29	42	48	58
Node increase rate	0	52.9%	70.5%	147%	182.3%	241.1%
Space	3.55687E+14	4.03291E+26	8.84176E+30	1.40501E+51	1.24139E+61	2.35056E+78
Space increase rate	0	1.13384E+14	2.48582E+18	3.95011E+38	3.49012E+48	6.6085E+65
S-MINE Average Time (Sec)	191.910	239.105	248.704	340.510	349.446	369.461
Deviation (Sec)	4.640	4.884	4.909	5.010	4.627	5.238

(그림 6) S-MINE 알고리즘 탐색시간변화 그래프에서와 같이 실험횟수에 따른 탐색시간이 각 데이터 셋마다 일정하게 변화하고 있는 것을 알 수 있다. gr17 데이터 셋은 190초대에서, fri26은 230초대에서, bay29는 240초대에서, swiss42는 340초대에서, hk48은 350초대에서, brazil58은 370초대에서 각각 표준오차 5초 이내로 변화하고 있다. <표 3>은 TSP 노드수 증가에 따른 S-MINE 알고리즘의 탐색시간 증가율을 보여준다.

TSP 노드수를 가장 작은 노드수 대비 각각 25.9%, 70.5%, 147%, 182.3%, 241.1%씩 증가시켰고, 노드수 증가에 따라 계산량이 n! 만큼 증가하여 문제의 복잡도가 기하급수적으로 증가한다. 이때 S-MINE 알고리즘의 탐색시간을 측정 한 결과 탐색시간도 노드수 증가에 따라 증가하고 있으나 노드수 증가율 대비 S-MINE 탐색시간 증가율이 50% 감소되고 있음을 알 수 있다.

5. 결 론

2009년 제안된 MINE 알고리즘은 세가지 표준문제에 적용하여 기존의 메타휴리스틱 알고리즘보다 개선된 범용성을 보였다. 본 논문에서는 S-MINE 알고리즘으로 확장된 MINE 알고리즘이 TSP 문제에 적용가능하며, 최단경로를 찾기 위한 메타 휴리스틱 형태의 최적화 알고리즘임을 고찰하였다. S-MINE 알고리즘은 인간 사회활동의 조직, 독립, 경쟁, 작업형태 등의 특징을 개념적으로 재정의하고 각각의 관계를 모델링 하였다. MINE 알고리즘에 기반한 S-MINE 알고리즘은 평가 파라미터를 변경률과 집중도로 정의 하였으며, TSP 에서 경로탐색을 위하여 탐색 알고리즘을 새롭게 제안하였다.

또한 제안된 S-MINE 알고리즘의 성능을 평가하고, 특징을 알아보기 위하여 TSP 에 적용 실험 하였다. TSP 문제에 적용한 결과 6개의 실험 데이터 셋에서 TSPLIB에서 제공하는 알려진 최적 경로값에 90% 이상 접근하는 탐색률을 보였다. 노드수를 17개 대비 147%, 182%, 241% 증가시킨 swiss42, hk48, brazil58 데이터 셋 에서는 접근율이 3-4% 정도 낮아지고 있지만 역시 90% 이상의 탐색률을 보이고 있어 100개 미만의 노드에서는 전체적으로 안정적인 모습을 보인다.

실험과정에서 S-MINE 알고리즘의 외부입력 조건인 광산

수, 광부 수, 세션 수 의 값이 노드수에 따라 변화되고 있는 것을 알 수 있었다. 노드수 17, 26개의 데이터 셋 에서는 탐색 경로값 조건으로 광산은 30개, 광부는 광산의 8배, 세션은 40으로 나타났다. 광산의 수를 30 이상으로 증가시키거나 30 미만으로 하여 실험하였을 때 알고리즘 성능 5~10% 정도 떨어졌다. 광부는 광산의 수에 따라 8배 이상 요구되었다. 알고리즘 수행 세션은 40회 이상으로 넘어가면 탐색 그래프가 진동하며 탐색 경로값에 수렴하였다. 노드수를 29, 42, 48, 58으로 증가시키면 두 배 이상의 세션이 요구된다. 즉, 광산은 30개, 광부는 광산의 10배, 세션은 100으로 조절했을 때 더 좋은 결과값이 나왔다. S-MINE 알고리즘은 광산수 보다는 세션수에 더 의존적으로 보인다. 노드수 40 이상에서도 1차 수렴 후 진동하면서 탐색하는 자기 조직화 특징을 보여주고 있다. 수렴, 진동, 창발을 반복하면서 알려진 최적 경로값에 다가가고 있는 특징을 보인다.

S-MINE 알고리즘의 탐색효율성을 알아보기 위하여 노드수 증가에 따른 탐색시간 증가율을 비교하였다. 이때 탐색시간이 표준오차 5초 이내에서 변하고 있음을 알았으며, 노드수 증가에 따라 탐색시간도 증가하고 있으나 노드수 증가율 대비 S-MINE 탐색시간 증가율이 50% 정도 감소되고 있음을 알 수 있다.

본 논문에서는 실험을 통하여 S-MINE 알고리즘이 100개 이하 TSP 데이터 셋에서 안정적으로 최적해 탐색을 하고 있음을 알았다. 이러한 결과는 MINE 알고리즘이 메타 휴리스틱 알고리즘으로써 TSP에서도 탐색 가능함을 보이고 있다. 또한 탐색 과정에서는 S-MINE 알고리즘의 빠른 수렴 속도의 결과는 알고리즘의 특징 중 하나인 독립 경쟁을 이용한 탐색결과를 잘 보여주고 있다. 광산의 이동을 통한 지역 해 탈출은 탐색률을 높여 진동을 통한 창발을 잘 유도하고 있다.

이러한 1차 100개 미만 노드 TSP 데이터 셋 실험을 토대로 앞으로 100개 이상 노드 TSP 데이터 셋에 확장 적용해 보아야 할 것이다. TSP에서 노드수의 증가는 해공간의 증가로 계산량의 기하급수적 증가를 의미한다. 따라서 MINE 알고리즘을 복잡한 사회 문제에 실제 적용하기 위해서는 대용량 해공간에서의 최적해 탐색 성능이 요구된다. 이에 S-MINE 탐색 알고리즘의 확장을 통한 대용량 TSP 적용은 본 논문의 향후 과제로서 폭넓은 응용분야를 모색하기 위한 중요한 과정이 될 것이다.

참 고 문 헌

- [1] S. Lin and B.W. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem," Bell Telephone Laboratories, Incorporated, Murray Hill, N.J. 1971.
- [2] M.R. Garey and D.S. Johnson, "A Guide to the Theory of NP-Completeness", Computers and Intractability, Freeman, 1979.
- [3] B. Freisleben and P. Merz, "Genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems," Proceedings of IEEE International Conference of Evolutionary Computation, IEEE-EC96, IEEE Press, pp.616-621, 1996.
- [4] Zhong Yiwen, Yang Xiangang, Ning zhengyuan. "A Discrete Particle Swarm Optimization Algorithm for Solving Traveling salesman Problem," System Engineering and Theory Practice, pp.88-94, 2006.
- [5] L.M. Gambardella and M. Dorigo, "Ant Colony System: A cooperative Learning Approach to the Traveling Salesman Problem," IEEE Transactions on Evolutionary Computation, Vol.1, No.1, 1997.
- [6] Christian Blum and Andrea Roli, "Metaheuristics in combinatorial optimization: overview and Conceptual comparison," ACM Computing Surveys, Vol.35, pp.268-308, 2003.
- [7] L. Bianchi, M. Dorigo, L.M. Gambarella, and W.J. Gutjahr, "A survey on metaheuristics for Stochastic combinatorial optimization," Natural Computing, 2008.
- [8] Sarayut Nonsiri an Siriporn Supratid, "Modifying Ant Colony Optimization," IEEE conference on soft computing in industrial applications, 2008.
- [9] X. Hu, R.C. Eberhart, and Y. Shi, "Swarm Intelligence for permutation Optimization: a case study of n-queens problem," in proceedings of the IEEE Swarm Intelligence Symposium, Indianapolis, USA, April, pp.243-246, 2003.
- [10] 고성범, "Mine 알고리즘: 인간의 행동을 모방한 메타휴리스틱", 정보처리논문지, 제16-B권 제5호, pp.411-426, 2009.
- [11] Solis, F.J. and Wets R.J. "Minimization by random search techniques," Mathematics of operations research, Vol.6, No.1, pp.19-30, 1981.
- [12] Moscato, P. "Memetic algorithms: A short introduction," In come, D, et al, eds: New Ideas in Optimization. McGraw Hill, pp.219-234, 1999.
- [13] Puchinger, J., Raidl, G.R., "Modules and Algorithms for Three-stage two-dimensional bin Packing," European Journal of Operational Research, Feature Issue on Cutting and Packing, 2006.
- [14] TSPLIB - A Traveling Salesman Problem Library.<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>
- [15] D.E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," Addison-Wesley, 1989.
- [16] Z. Michalewicz, "Genetic Algorithms + Data structure = Evolution Program," Springer-Verlag, 1992.
- [17] Dorigo M., Maniezzo V., Colorni A., "The Ant System: Optimization by a Colony of Cooperating Agents," IEEE trans. Syst. Man Cybern B 26, pp.29-41, 1996.
- [18] Dorigo M., Gambardella L.M., " Ant Colony System: A Cooperative Learning Approach," IEEE Trans. Evol. Comp.1, pp.53-66, 1997.
- [19] Raed Abu Zitar, Huessein Hiyassat, "Optimizing the parameters of Ant colony Algorithm Using the Genetics," Enformatika Transaction on Engineering, Computing and Technology VI, December, pp.228-231, 2004.
- [20] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," IEEE conference of Neural Network, Vol.IV, pp.1942-1948, 1995.
- [21] J. Kennedy, "The Particle Swarm: Social adaptation of knowledge," IEEE conference of Evolutionary Computing, pp.303-308, 1997.
- [22] X.H. Shi, Y.C. Liang, H.P. Lee, C. Lu, and L.M. Wang, "An improved GA and novel PSO-GA-based hybrid algorithm," Information Processing Letters, 1993.
- [23] 고성범, "인터넷기반 의사결정 공간의 설계" GS 인터뷰전, 2010.
- [24] J. De Vicente, J. Lanchares, R. Hermida, "Placement by Thermodynamic Simulated Annealing," Physics Letters A, Vol.317, Issue 5-6, pp.415-423, 2003.

황 숙 희



e-mail : mamipapa714@hanmail.net
 1994년 상명대학교 전자계산학과(학사)
 1997년 상명대학교 교육대학원 컴퓨터
 교육전공(교육학 석사)
 2004년 건국대학교 컴퓨터공학과 박사
 과정 수료
 관심분야: 인공지능, 알고리즘

원 일 용



e-mail : clccc@shoseo.ac.kr
 1998년 경원대학교 전자계산학과(학사)
 2000년 건국대학교 컴퓨터공학과(석사)
 2006년 건국대학교 컴퓨터공학과(박사)
 2008년~현 제 서울호서전문대학교 사이버
 해킹보안과 교수
 관심분야: 뇌과학, 복잡성의 과학, 보안



고성범

e-mail : kosb@kongju.ac.kr
1980년 송전대학교 전기공학과(학사)
1983년 서울대학교 전기공학과(석사)
2003년 건국대학교 컴퓨터공학과(박사)
2005년~현 재 공주대학교 컴퓨터공학부
교수
관심분야: 인공지능



이창훈

e-mail : chlee@konkuk.ac.kr
1977년 연세대학교 수학과(학사)
1980년 한국과학기술원 전산학과(석사)
1993년 한국과학기술원 전산학과(박사)
1996년~2002년 건국대학교 정보통신원
원장
2001년~2002년 건국대학교 정보통신대학 학장
1980년~현 재 건국대학교 컴퓨터공학과 교수
관심분야: 지능시스템, 운영체제, 보안