DOI: 10.3745/KIPSTB.2009.16B.5.411

Mine 알고리즘 : 인간의 행동을 모방한 메타휴리스틱

고 성 범†

요 약

대부분의 메타휴리스틱들은 동물의 행동을 모방한 것이다. 본 논문에서는 Mine 알고리즘을 제안한다. Mine 알고리즘(Mine Algorithm)은 인간의 행동을 모방한 메타휴리스틱이다. 탐색의 관점에서 인간의 노하우와 휴리스틱이 가장 잘 녹아 있는 업종은 광산업(mining industry)이다. Mine 알고리즘에서는 광산 업무에 초점을 맞추어서 인간의 행동패턴을 형식화한다. Mine 알고리즘은 다양한 탐색기법을 유연하게 구사하며, 그 때문에 광범위한 문제에서 고른 성능을 보인다. 즉, 범용성이 양호하다. 우리는 기존 메타휴리스틱들과의 비교 실험을 통하여 Mine 알고리즘의 개선된 범용성을 보인다.

키워드 : 탐색, 메타휴리스틱, 범용성, 자기조직화, Mine 알고리즘

Mine Algorithm : A Metaheuristic Imitating The Action of The Human Being

Ko Sung-Bum†

ABSTRACT

Most of the metaheuristics are made by imitating the action of the animals. In this paper, we proposed Mine Algorithm. The Mine Algorithm is a metaheuristic that imitates the action of the human being. Speaking of search, the field in which the know-how and the heuristics of the human being are melted best is the mining industry. In the Mine Algorithm we formalize the action pattern of the human being by focusing the mine business. The Mine Algorithm uses various searching techniques fluently and shows equally good performance for broad problems. That is, it has good generality. We show the improved generality of the Mine Algorithm by the comparing experiments with the conventional metaheuristics.

Keywords : Search, Metaheuristic, Generality, Self-Organization, Mine Algorithm

1. 서 론

범용성의 측면에서 휴리스틱 개념을 한 단계 개선한 것이 메타휴리스틱이다. 즉, 메타휴리스틱은 추상성이 매우 높은 해공간 탐색 모델이다. 메타휴리스틱을 사용하면 보다 광범위한 문제들을 쉽게 다룰 수 있다. 잘 알려진 메타휴리스틱으로는 유전알고리즘(Genetic Algorithm : GA), 입자군집최적화(Particle Swarm Optimization : PSO), 개미군락최적화(Ant Colony Optimization : ACO), 시뮬레이티드 어닐링(Simulated Annealing : SA), 터부탐색(Tabu Search : TS), 언덕 오르기(Hill Climbing : HC) 그리고 무작위 탐색(Random Search : RS) 등을 들 수 있다.

메타휴리스틱의 대부분은 동물 특히 군집성 동물의 행동을 모방한 것이다[1-3]. 예를 들어, GA 는 동물들이 교미를 통해서 번식하는 행동을 모방한 것이다. PSO는 새나 물고기에서 볼 수 있는 이동 패턴을 모방한 것이다. ACO는 개미들이 페로몬을 이용해서 최단거리를 찾아내는 행동을 모방한 것이다. HC는 구배 개념을 이용한 것인데, 이것은 산을 오르는 들짐승의 행동을 연상시킨다. TS는 뇌세포의 기억 패턴을 모방한 것이다. SA만이 예외이며, 이 모델은 담금질을 모방한 것으로 알려져 있다.

행동 패턴의 관점에서 사람에게는 동물(군집성 동물)과 구별되는 점들이 있다. 첫째, 동물은 이타적인데 반해, 사람은 이기적이다. 둘째, 동물과는 달리, 사람은 개성을 갖는다. 셋째, 동물은 눈에 보이는 성과만을 중시하지만, 사람은 눈에 보이지 않는 성과도 함께 고려한다. 넷째, 사람은 더 큰 만족을 위해서 만족스러운 현실을 과감히 버린다. 동물은 절대로 그런 짓을 하지 않는다. 다섯째, 특성화 측면에서 사

† 정 회 원 : 공주대학교 컴퓨터공학부 교수
논문접수 : 2009년 5월 13일
수 정 일 : 1차 2009년 6월 29일
심사완료 : 2009년 6월 29일

람은 동물보다 유연하며 얼마든지 자신의 역할을 바꿀 수 있다. 여섯째, 사람은 때로 현실의 기본 틀을 혁명적으로 바꾸는 시도를 한다. 일곱째, 사람은 동물보다 훨씬 다양하고 또 고차원적인 휴리스틱을 구사한다.

본 논문에서는 Mine 알고리즘(Mine Algorithm : MA)을 제안한다. Mine 알고리즘은 인간의 행동패턴을 모방한 메타휴리스틱이다. 탐색의 관점에서 인간의 노하우와 휴리스틱이 가장 잘 녹아 있는 산업 분야는 광산업일 것이다. 광산업은 뭔가를 찾아내는 작업 즉, 탐색 행위 그 자체이다. Mine 알고리즘에서는 광산 업무에 초점을 맞추어서 인간의 행동패턴을 추상화하고 형식화(formalize)한다.

Mine 알고리즘의 중요한 특징은 다양한 탐색 기법들을 구사한다는 점이다. 그러나 이들 다양한 기법들을 결합하거나 조정하는 별도의 장치를 갖고 있지는 않다. 대신에 자기조직화 기능을 이용해서 동일한 목적을 달성한다. Mine 알고리즘은 광범위한 문제에서 비교적 고른 성능을 보인다. 즉, 범용성이 한 단계 개선되어 있다. 우리는 기존 메타휴리스틱들과의 비교 실험을 통하여 Mine 알고리즘의 개선된 범용성을 보인다.

논문의 구성은 다음과 같다. 2장에서는 관련 연구를 살펴본다. 3장에서는 Mine 알고리즘을 소개한다. 4장에서는 Mine 알고리즘이 갖는 특징을 분석한다. 5장에서는 실험을 통해 Mine 알고리즘의 범용성을 검증한다. 끝으로 6장에서는 결론과 함께 다음 과제를 알아본다.

2. 관련 연구

2장에서는 대표적인 메타휴리스틱들의 구조와 기능 그리고 장단점 등을 종합 정리한다. 첫째, 유전 알고리즘(Genetic Algorithm : GA)은 동물의 번식 패턴(혹은 유전 현상)을 흉내 낸 메타휴리스틱이다[4]. 시스템이 수행되면 일정한 해 집단을 유지하며 부모와 다르지만 일정 부분 닮은 자식해들을 만든다. 대체로 숲을 보는 능력은 뛰어나지만 나무를 보는 능력이 취약하다. 즉, 섬세한 최종단계 탐색 능력이 부실하다.

둘째, 입자군집 최적화(Particle Swarm Optimization : PSO)는 새나 물고기 등의 독특한 이동 양식을 모방한 메타휴리스틱이다[5]. PSO의 각 파티클은 다차원 탐색공간을 날아다니며, 자신과 이웃들의 경험 정보를 이용하여 최적의 위치로 이동해 간다. 전반적으로 이론이 간결하고 구현이 용이한 편이다. GA 등 다른 메타휴리스틱에 비해 수렴 속도가 빠른 반면, 지역해에서 탈출하는 능력은 부족한 감이 있다.

셋째, 개미군락 최적화(Ant Colony Optimization : ACO)는 개미들이 페로몬을 이용하여 최단거리를 찾아내는 것을 모방한 메타휴리스틱이다[6]. 개미들은 페로몬을 선호하며 스스로도 이동 중에 페로몬을 뿌린다. 그래프로 표현된 최단거리 문제에서 어떤 경로가 최단거리일 경우 그 경로의 페로몬이 가장 많아진다. ACO는 여행자 문제(travelling salesman problem)처럼 최단거리 문제로 환원되는 문제에서 특히 좋은 성능을 보인다. 그런데 많은 최적해 탐색문제들은, 그래프로 표현될 수는 있지만, 최단거리 문제로 환원되지는 않는다. 그런 문제에서 ACO는 그다지 좋은 성능을 내지 못한다.

넷째, 시뮬레이티드 어닐링(Simulated Annealing : SA)은 금속의 담금질(annealing) 과정을 모방한 메타휴리스틱이다[7]. 이 모델에서는 지역해에서 머무르는 것을 막기 위해 개선되지 않은 이웃해 즉, 열등해로의 이동을 확률적으로 허용한다. 담금질 개념의 핵심은 개선되지 않은 이웃해로의 이동확률을 단계적으로 줄여 나가는 데 있다. 이것이 바로 담금질의 온도 개념이다. SA의 큰 흐름은 탐색의 초점이 다양성에서 집중성으로 옮겨가는 것이다.

다섯째, 터부 탐색(Tabu Search : TS)은 뇌세포의 기억과정 즉, 메모리 구조에서 힌트를 얻은 메타휴리스틱이다[8]. 뇌의 메모리 구조는 단기, 장기, 중기 메모리로 나누어지는데, 각각은 탐색에서의 제약/열망, 다양성, 집중성 개념에 대응한다. TS에서는 터부제약과 열망 수준을 참조하여 현재해로부터 새로운 해를 선택한다. 터부 개념은 순환을 탈출하는 데 효과가 있으며 예를 들어, 언덕처럼 높낮이가 별로 없는 문제에서 좋은 성능을 발휘한다.

여섯째, 언덕 오르기(Hill Climbing : HC)는 구배(gradient)를 찾아서 이동하는 메타휴리스틱으로 정의될 수 있다[9]. 구배(gradient)란 적합도가 가장 가파르게 변화하는 방향을 의미하는 수학 용어이다. 더 나은 곳으로 이동 할 수 없으면 종료하는데, 이 때문에 지역해에 빠질 수 있다. 이를 막기 위해서는 적합도에 비례해서 이동확률을 주거나 시작점을 여러 개 주는 방법(다중 시작 HC) 등이 고려될 수 있다. 다중 시작 HC를 사용하면 지역해에 빠질 가능성은 적다. 단, 구배가 정의되기 어려운 곳에서는 방향을 잡기 어려운 단점이 있다.

일곱째, 무작위 탐색(Random Search : RS)은 시행착오에만 의존하는 매우 단순한 메타휴리스틱이다[10]. 즉, 휴리스틱이나 지능을 전혀 사용하지 않는다. 그래서 하찮게 보일지 모르지만 아무런 특징이 없는 카오스적 해공간에서는 이것이 가장 좋은 해법이 된다. 이런 공간에서는 RS를 포함해서 어떤 탐색모델도 최적해를 찾을 확률이 동일하다. 따라서 시스템이 가벼운 RS가 유리해 진다. 일반적으로 RS는 다른 탐색 모델을 평가하는 기준이 된다.

상기 7개의 메타휴리스틱 중 SA와 TS의 경우는 사용하는 세부 기법들의 일반성이 매우 강해서 다른 메타휴리스틱에서도 쉽게 적용할 수 있다. 더구나 다른 기법과도 비교적 잘 어울린다. 실제로 이 기법들은 대부분의 메타휴리스틱에서 사용되고 있으며, 일종의 공유 기술처럼 되었다. 따라서 이 두 모델은 더 이상 독립된 메타휴리스틱으로 보기 어렵다.

메타휴리스틱이 갖는 약점을 보완하기 위해서 다른 메타휴리스틱과 부분적인 결합이 시도되기도 한다. 이를 하이브리드 메타휴리스틱이라고 부른다. 일테면 GA의 경우, 태어난 자식을 국소탐색(HC를 이용)을 통해 개선한 다음에 해 집단에 넣는 방식이 있다[11]. 교차를 이용해서 자식을 만들

때 path-relinking같은 강력한 기법을 활용하는 방식도 있다[12]. 하나의 탐색모델을 이용해서 만든 불완전한 해를 디코더라 부르는 다른 모델이 이어 받아서 다듬는 방식도 있다[13]. GE (Grammatical Evolution)는 해공간 부분과 탐색 알고리즘 부분이 분리되어서 다양한 탐색 모델의 사용이 가능한 이점이 있다[14]. GA와 역전파 알고리즘도 결합이 쉬운 사례로 알려져 있다[15]. 메타휴리스틱의 성능 개선을 위한 또 다른 접근으로 병렬 메타휴리스틱 분야도 있다. 이것은 복수개의 프로세서를 이용해서 여러 모델을 동시에 가동시키는 방법인데, 여기서는 효율적인 정보 공유가 중요하다[16].

3. Mine 알고리즘의 소개

Mine 알고리즘은 광산 업무를 형식화(formalize)한 메타휴리스틱이다. 즉, 광산업무의 기본 틀을 프로그램 언어로 구현한 것이다. 여기서 광산의 목적은 '최고의 보석(최적해에 대응)을 찾는 것'으로 가정한다. 즉, 보석의 양은 문제 삼지 않는다.

3.1 시장 개념

시장경제 하에서, 광산업도 하나의 독립된 작은 시장으로 간주된다. 시장은 시장을 구성하는 행위자(actor) 집합으로 정의된다. 즉, 시장을 구현한다는 것은 시장을 구성하는 행위자(혹은 행위자의 행동 패턴)들을 구현하는 것과 같은 의미이다.

3.2 광산시장의 행위자

인간이 만든 전형적인 조직은 최고책임자, 중간책임자, 정해진 일을 하는 그룹, 새로운 일을 개척하는 그룹 등 4개의 역할로 구성된다. 현실세계의 조직 대부분이 이 패턴을 따르고 있으며 심지어는 동네의 작은 교회도 그러하다. 광산조직도 기본적으로 이 패턴을 따르고 있다. 다만 구체적으로 하는 일이 다를 뿐이다. 본 논문에서는 이들의 이름을 회장, 사장, 광부, 헌터로 부르는 데, 물론 이름이 중요한 것은 아니다. 이들이 하는 일은 다음과 같다.

3.2.1 회장

회장은 광산그룹 전체를 책임지는 최고 경영자이다. 광산그룹이 장기간 침체에 빠질 경우 회장은 그 능력을 의심받게 된다. 보수도 떨어진다. 이 경우 회장은 국면 전환을 위해 대규모 구조조정을 고려한다. 구조조정은 다음 세 가지 원칙에 따른다. 첫째, 일차적으로 경영진(리더 그룹)을 문책한다. 즉, 구조조정의 대상은 서열이 높은 광산들이다. 둘째, 구조조정은 일차, 이차, 삼차 등 단계적으로 이루어진다. 각각의 시간차는 점점 짧아지며 구조조정의 폭은 점점 커진다. 즉, 뒤에 오는 구조조정일수록 더 과격해진다. 셋째, 구조조정으로 문을 닫은 광산은, 폐광으로 문을 닫은 광산과는 달리, 폐광 리스트에 등록되지 않는다.

3.2.2 사장

사장은 광산의 운영으로 돈을 번다. 광산은 사장이 가진 보석을 중심으로 일정 크기의 원으로 정의된다. 사장은 광부를 고용하여 보석을 채굴하며, 좋은 보석이 발견되는 곳으로 계속해서 광산을 이동시킨다. 두 가지 경우에 광산의 운영은 어려워진다. 첫째는 광산의 서열이 너무 낮은 경우이고, 둘째는 서열은 높지만 개선이 안 되고 있는 경우이다. 사장은 둘 중 하나에 직면하면 폐광을 고려한다. 전자에 대한 폐광을 '악성폐광'이라 하고 후자에 대한 폐광을 '양성폐광'이라 한다. 하나의 광산이 폐광되면 새로운 광산이 다른 지역에서 태어난다. 폐광된 광산은 새로운 광산의 탄생 지역과 광산의 이동 대상에서 제외된다.

3.2.3 광부

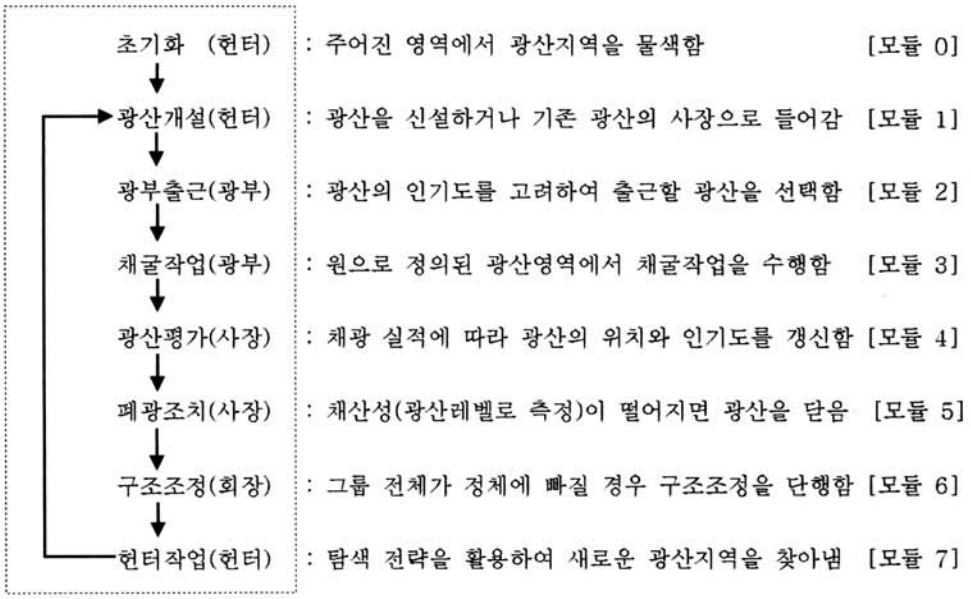
광산에서 채굴업무를 담당하는 광부는 일당을 받는다. 단, 보석의 질에 따라 보수를 받는 소위 성과급이다. 당연히 광부들은 인기 있는(실적과 장래성이 있는) 광산을 더 선호한다. 결과적으로 인기도가 더 높은 광산이 더 많은 광부를 끌어들이게 된다. 그런데 광부마다 광산의 레벨에 대한 최소한의 요구치가 있다. 광부들은 이 기준에 미달되는 광산에 대해서는 아예 출근을 거부한다. 만일 광산 그룹 전체가 정체에 빠질 경우 대부분의 광부들이 출근하지 않을 것이다. 즉, 이 경우 결과적으로는 대규모 파업과 비슷한 상황이 전개될 수 있다.

3.2.4 헌터

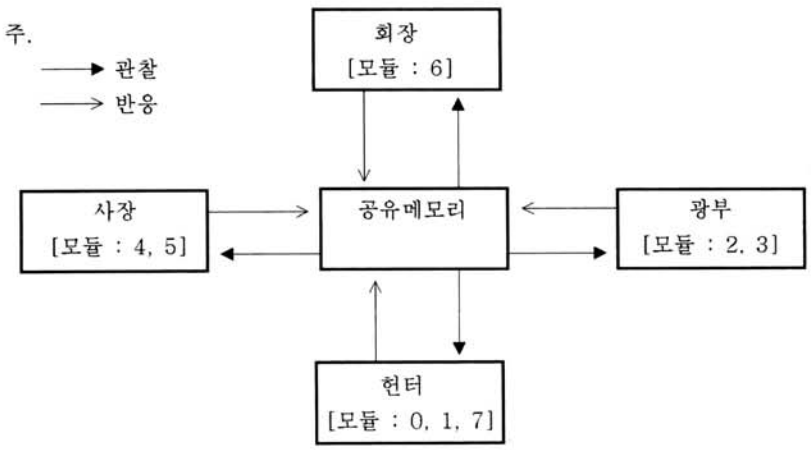
광산을 개척하는 역할을 하는 헌터는 좋은 보석들이 있을 만한 지역을 부지런히 찾아다닌다. 좋은 보석을 캐면 스스로 광산을 개설할 수도 있고, 자신의 성과를 이용해서 기존 광산의 사장으로 치고 들어갈 수도 있다. 헌터들은 과거 혹은 현재의 우량광산들이 갖는 특징들을 참고하여 탐색지역을 정한다. 통상 네 가지 전략이 사용된다. 첫째, 우량해의 특징들을 교차시켜서 탐색지역을 정한다(교차 탐색). 둘째, 우량해를 적절히 모방하여 탐색지역을 정한다(모방 탐색). 셋째, 우량해에 돌연변이를 첨가하여 탐색지역을 정한다(돌연변이 탐색). 넷째, 아무도 가본 적이 없는 공터를 찾아서 탐색지역을 정한다(공터 탐색).

3.3 알고리즘

Mine 알고리즘은 기본적으로 행위자 중심으로 돌아가는 구조이다. 알고리즘 구축은 직렬 모드와 병렬 모드 모두 가능하다. 후자의 경우는 5개 이상의 프로세서가 지원되어야 한다. 이 중 4개는 행위자들에게 배당하고, 나머지 하나는 공유 메모리로 사용하면 된다. 직렬 모드에서는 행위자보다는 프로세스 기준으로 배열하는 게 편리하다. 병렬 모드에서는 일의 주기와 선후 관계가 문제될 수 있는 데, 어느 정도 경험적 설정이 가능하다. 예를 들어, 광산의 평가나 폐광은 한 템포 늦는 게 좋고 특히 구조조정 같은 것은 너무 자주 하면 역효과가 날 가능성이 있다. (그림 3)(1)과 (그림



(그림 3-1) Mine 알고리즘의 직렬구조



(그림 3-2) Mine 알고리즘의 병렬구조

3)(2)는 각각 직렬 모드와 병렬 모드를 보여준다.

(그림 3-2)의 병렬 구조에서 사각형은 독립된 프로세서를 의미한다. 여기서 공유메모리를 제외한 4개의 프로세서는 4가지 유형의 행위자에 대응한다. 각각의 행위자는 직렬구조(그림 3-1)의 모듈세트를 완전한 형태로 갖고 있으면서 이 중 일부 모듈만을 선택적으로 수행한다. 만일 어떤 사유로 일부 행위자들이 파괴된다면 남은 행위자들이 파괴된 부분을 추가 활성화하는 방법으로 대처할 수 있다. 이것은 강건성(robustness) 측면에서 유용하다.

Mine 알고리즘에서 행위자들이 하는 일은 의외로 간단한데, 이들은 한두 개의 주변 상태를 관찰하고 이에 반응한다. 행위자들은 이 단순한 작업을 끊임없이 반복하며, 이것이 이들이 하는 일의 전부이다. <표 2-1>은 행위자의 관찰/반응 관계를 보여준다.

〈표 2-1〉 행위자의 관찰/반응 관계

행위자	관찰 대상	반응
회장	정체기간	구조조정
사장	광산실적 채굴된 보석	폐광조치 보석의 교체(광산의 이동)
광부	광산의 인기도 광산의 중심	출근할 광산 선택 중심 부근에서 채굴
헌터	광산들의 실적 우량 광산 리스트	부진한 광산의 사장 교체 탐사 지역 선정 및 채굴

4. Mine 알고리즘의 특징

4장에서는 3장에서 소개한 Mine 알고리즘을 정량화, 구체화하고 그 특징을 분석한다.

4.1 수치 분석

4.1.1성과와 인기도

광산을 평가하는 두 가지 기준은 '성과'와 '인기도'이다. 광산의 성과는 광산이 소유한 보석의 점수로 표현되고, 광산의 인기도는 광산레벨로 표현된다. 광산레벨은 약간 추상적인 용어로 PID 개념을 이용해서 정의된다[17]. 여기서 PID의 의미는 다음과 같다.

- P(Proportional) : 비례 요소이며 현 시점의 성과(찾은 보석의 점수)를 의미한다.
- I(Integral) : 적분 요소이며 그 동안의 누적 성과를 의미한다.
- D(Differential) : 미분 요소이며 현 시점에서 본 성과의 개선 정도를 의미한다.

이산적 시간을 가정하고, 광산 m의 시점 t에서의 P값을 P(m,t)라 하자. 광산 m의 시점 t에서의 D값 D(m,t)는

$$D(m,t) = P(m,t) - P(m, t-1) \tag{1}$$

이다. 그러면 광산 m의 시점 t에서의 I값 I(m,t)는

$$I(m,t) = \sum_{i=t-h}^{t} D(m,i) \tag{2}$$

여기서, h : 적분 항목의 개수

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

$$ML(m,t) = \alpha\,(P(m,t)\times I(m,t)) + \beta\,(P(m,t)\times D(m,t)) \tag{3}$$

여기서 α 와 β는 각각 PI와 PD에 대한 상대적인 비중이다. 식 (3)을 보면, P값이 아무리 높아도 I와 D값이 부실하면 광산레벨이 낮아진다는 걸 알 수 있다. 즉, 광산은 아무리 좋은 실적을 올렸어도 그 실적이 계속해서 개선되지 않으면 인기를 유지할 수 없다. 사실 (3)식은 평가에 대한 사람들의 상식을 반영하고 있다. 즉, 우리가 누군가를 평가할 때는 실력(P)과 저력(I)과 잠재력(D)을 종합적으로 고려한다.

4.1.2 광부의 수

광산에 출근하는 광부의 수는 의무배당과 옵션배당으로 나눈다. 모든 광산은 최소한 1명의 광부가 출근하는 데 이를 의무배당이라 한다. 한편 옵션배당은 광산의 인기도에 끌려 출근하는 광부의 수이다. 인기도는 광산레벨로 표현되는데, 시점 t에서 광산 m의 광산레벨을 ML(m,t)이라 하자. 그러면 시점 t에서 광산 m에 1명의 광부가 옵션 배당될 확률 $P_r(m,t)$는

$$P_r(m,t) = \begin{cases} \dfrac{ML(m,t)}{\sum_{i=1}^{M} ML(i,t)} & \text{If } ML(m,t) \geq R \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

여기서,
R : 광산레벨 임계치(광부가 요구하는 광산레벨 최저치)
M : 광산의 총 수

와 같이 계산된다. 따라서 시점 t에서 광산 m에 배정될 광부의 수에 대한 기대치 N(m,t)는

$$N(m,t) = N_a P_r(m,t) + 1 \tag{5}$$

여기서, N_a : 광부의 총수

이다. 따라서 시점 t에서 광산에 출근하는 광부의 총수 N_b는

$$N_b = \sum_{i=1}^{M} N(i,t) \tag{6}$$

이다. 그러면 시점 t에 출근하지 않는 광부의 총수 N_c 는 $N_c = N_a - N_b$ 가 된다. 그런데 광부에게 배당된 시간과 헌터에게 배당된 시간의 합 T는 일정하다. 처음 광부와 헌터에게 배당된 시간을 각각 T_1, T_2라 하면, $T = T_1+T_2$이다. 광부들이 실제로 사용하는 시간 T_1'는 출근하지 않는 광부의 수만큼 줄어든다. 즉, $T_1' = T_1\times\frac{N_b}{N_a}$ 이다. 광부가 사용하는 시간이 줄어들면 헌터가 사용하는 시간이 늘어난다. 즉, 헌터가 사용하는 시간 T_2'는

$$\begin{aligned} T_2' &= T - T_1' \\ &= T - (T_1\times\frac{N_b}{N_a}) \end{aligned} \tag{7}$$

와 같다. 광부가 활동하는 시간의 양과 헌터가 활동하는 시간의 양은 서로 배타적 관계인데, 양자간의 비 δ를 다음과 같이 계산할 수 있다.

$$\begin{aligned} \delta &= \frac{T_2'}{T_1'} \\ &= \frac{T - \frac{T_1 N_b}{N_a}}{\frac{T_1 N_b}{N_a}} \\ &= \frac{TN_a}{T_1\sum_{i=1}^{M} N(i,t)} - 1 \end{aligned} \tag{8}$$

4.2 정리 증명

Mine 알고리즘에서의 채굴 작업(해공간 탐색 작업)은 광산을 중심으로 행해진다. 광산과 관련하여 몇 가지 중요한 사실들을 확실하게 정리해둘 필요가 있다.

[정리 1] 광산은 이동하거나 멈추는 데, 일정 시간 멈춘 광산은 소멸한다.

[증명] 광산은 이동하거나 멈추거나 둘 중 하나이다. 광산이 멈추었다는 것은 ML(광산레벨)의 D값(미분요소)이 0이라는 뜻이다. 이 상태에서 일정 시간이 지나면 식 (2)에 의해서 I값(적분요소) 역시 0이 된다. 따라서 식 (3)에 의해 ML은 PI와 PD의 합이므로 ML 값도 0이 된다. (그림 3-1)의 모듈 5에서 광산은 ML값이 일정시간 동안 임계치를 넘지 못하는 경우 폐광처리 된다. 그런데 임계치는 0이상이므로 일정시간 멈춘 광산은 반드시 소멸한다.

[정리 2] 이동하는 광산은 일정 시간 이내에 멈춘다.

[증명] 식 (1)에 의해서 D값이 0이 아니기 위해서는 ML의 P값(비례요소)이 계속 증가해야 한다. 여기서 P값은 특정 후보해의 적합도를 의미한다. 그런데 적합도는 정의에 의해서 유한한 값을 갖는다. 따라서 P값이 계속해서 증가할 수는 없다. 따라서 이동하는 광산은 유한한 시간 이내에 멈추어야 한다. 여기서 유한한 시간이란 최적해에 이르기까지 걸리는 시간을 의미한다. 실제로 대부분의 광산(후보해)들은 최적해에 이르기 전에 멈춘다.

[정리 3] 광산은 하나의 상태에서 일정 시간 이상 정체하지 않는다.

[증명] 광산은 이동하거나 멈추거나 둘 중 한 상태에 있다. [정리 2]에 의해 모든 광산은 멈출 수밖에 없고 멈춘 광산은 소멸될 수밖에 없다. 그리고 소멸되기 이전에는 [정리 1]에 의해서 일정 시간 이상 정체 상태에 있을 수 없다.

[정리 4] 흩어진 광산들은 모이려는 경향이 있다.

[증명] 두 가지 이유로 광산들은 모여드는 경향을 갖는다. 첫째, [정리 1]과 [정리 2]에 의해 모든 광산은 언젠가는 폐광될 운명에 있다. 즉, 시간이 흐르면서 폐광리스트는 계속 커져 간다. 폐광지역은 정의에 의해 광산의 탄생 장소나 이동 경로에서 제외된다. 이것은 광산들이 새로 태어나거나 이동하기 위한 후보 지역이 계속 좁아진다는 걸 의미한다. 결과적으로 광산들은 모여들 수밖에 없다. 둘째, 유의미한 해공간 탐색 문제를 가정할 때, 우량해는 특정 지역에 몰려 있다고 보아야 한다. 모든 광산들은 당연히 우량해를 지향한다. 따라서 광산들은 모여들 수밖에 없다.

[정리 5] 모인 광산들은 흩어지는 경향이 있다.

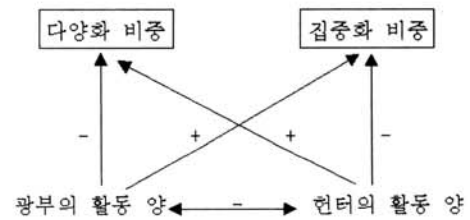
[증명] 광산들이 모여 있다는 것은 좁은 영역에 많은 광산이 밀집해 있다는 뜻이다. 즉, 좁은 영역에서 많은 광산들이 탐색 작업을 수행하는 것이므로, 우량해들은 조만간 모두 발견되고 더 이상의 진전이 불가능한 상태에 이른다. 이 경우 두 가지 사건이 일어난다. 첫째는 광부들의 대량 출근 거부 사태이고(모듈 2), 둘째는 회장에 의해서 단행되는 구조조정(모듈 6)이다. 두 사건 모두 폐광사태를 야기하는 데 후자가 좀더 큰 규모이다. 광산들이 사라지면 그 만큼 다른 지역에서 보충된다. 이로 인해 모인 광산들은 흩어질 수밖에 없다.

[정리 6] 광산들은 모인 상태와 흩어진 상태 사이를 끊임없이 오고 간다.

[증명] [정리 4]와 [정리 5]에 의해 증명된다.

4.3 다양화(diversification)와 집중화(intensification)의 자율 조정

4.1에서 분석한 바와 같이, 광부들의 활동량과 헌터들의 활동량은 서로 배타적인 관계에 있다. 즉, 광산들이 침체에 빠지면 헌터의 활동이 커져서 다양화(탐험) 쪽으로 중심이 옮겨지고, 그 반대의 경우는 광부들의 활동이 커져서 집중화(탐사) 쪽으로 중심이 옮겨진다. 즉, Mine 알고리즘에서는 (그림 4-1)에서 보는 것처럼 둘 사이의 중심 이동이 자연스럽다.

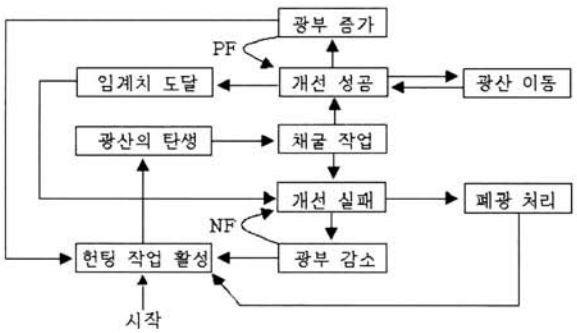


주) + : 긍정적 영향, - : 부정적 영향

(그림 4-1) 다양화와 집중화 간의 자율조정

4.4 광산의 생명주기

광산은 태어나고 성장하고 늙고 소멸한다. (그림 4-2)에 묘사된 이런 일련의 과정을 광산의 생명주기라고 부른다. 생명주기 안에는 곳곳에 되먹임 현상들이 숨어 있다. 예를 들어, 어떤 이유로 광산의 실적이 오르면, 다음 세션에서 더 많은 광부들을 끌어들이고 이로 인해 광산의 실적은 더 오를 수 있다. 이것은 전형적인 양의 되먹임이다. 반대로 어떤 이유로 광산의 실적이 떨어지면, 다음 세션에서 광부의 수가 줄어들어서 실적은 더 나빠질 수 있다. 이것은 전형적인 음의 되먹임이다.



주) PF : Positive feedback NF : Negative feedback

(그림 4-2) 광산의 생명주기

그러나 되먹임 과정이 언제나 이렇게 간단하지는 않다. 일테면 양의 되먹임 때문에 광부의 수가 너무 많아질 경우 광산자원이 빠르게 고갈될 수 있다. 그러면 광부의 수가 아무리 많아도 실적을 올릴 수 없게 된다. 이것은 오히려 음의 되먹임이다. 즉, 이런 특수한 경우에는 양의 되먹임과 음의 되먹임이 교묘하게 얽히게 된다.

4.5 광산타운의 생명주기

정리 6에 의해 광산 그룹은 이합집산을 반복한다. 보통 광산군도 형태에서 시작해서 광산마을 형태를 거쳐 광산타운 형태로 나아간다. 여기서 광산군도란 광산들이 섬처럼 흩어져 있는 상태이고, 광산마을은 광산들이 마을처럼 가까이 접근해 있는 상태이고, 광산타운은 좁은 영역에 대부분의 광산이 밀집되어 있는 상태이다. 광산타운은 시간이 흐르면서 침체기에 접어들게 되고 결국 구조조정에 의해 소멸된다. 이후 광산타운은 완전히 새로 태어나거나 광산마을 형태를 거쳐 광산군도로 역 진행을 하게 된다. 그 다음 다시 같은 과정이 반복된다. (그림 4-3)에 묘사된 이런 일련의 과정을 광산타운의 생명주기라고 부른다. 광산타운의 생명주기 곳곳에도 다양한 형태의 되먹임 현상이 숨어 있다.

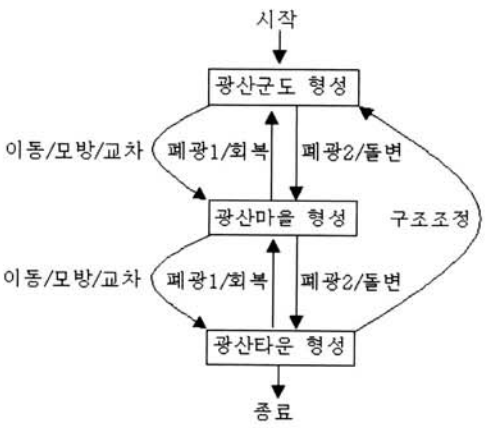
4.6 자기조직화(self-organization)

탐색 시스템을 포함한 모든 시스템에서 환경에 대한 적응 능력은 대단히 중요하다. 적응을 구현하는 방법론에는 학습 모델과 자기조직화 모델이 있다. 보통 적응 시스템은 3계층으로 구성할 수 있다. (그림 4-4)는 두 모델의 3계층 구조를 보여준다.

학습 모델에서는 각각 다른 환경에 대응하는 n개의 독립된 처리모듈이 내부 계층에 있다. 환경이 변화하면 학습계층에서 환경을 분석하여 필요한 지식을 습득한다. 그리고 이 지식에 근거해서 n개의 처리모듈 중 가장 적합한 놈을 선택하여 수행시킨다.

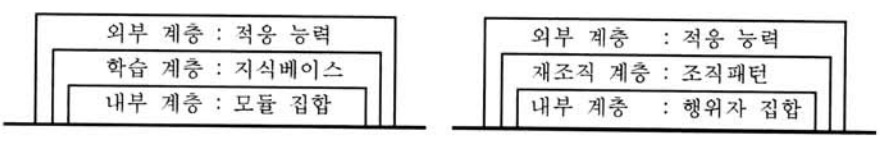
자기조직화 모델에서는 환경의 변화와 거의 무관한 행위자들이 내부 계층에 있다. 이들은 환경의 변화를 의식하지 않고 어떤 정해진 행위를 반복한다. 이와는 대조적으로 재조직 계층에서는 환경의 변화에 민감하게 반응하며 자신의 조직패턴을 유연하게 변화시킨다.

이처럼 두 방법론을 가르는 개념은 지식이다. 학습 모델에서는 지식을 구하고 그 지식을 이용해서 환경에 적응한다. 자기조직화 모델에서는 시스템의 특정 부분을 재조직하여 환경에 적응한다. 학습의 경우, 학습 역할을 전담하는 독립된 장치가 존재하며 학습의 결과로 지식이 얻어진다. 자기조직화의 경우, 자기조직화 역할을 전담하는 장치가 따로



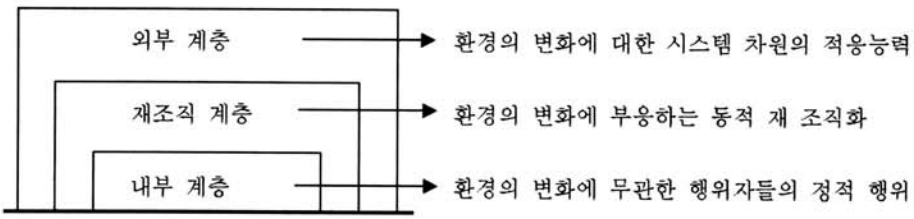
주)
이동 : 광산의 이동
모방 : 헌터의 모방 탐색
교차 : 헌터의 교차탐색
폐광1 : 우량광산의 폐광
폐광2 : 열등광산의 폐광
회복 : 폐광의 회복
돌변 : 헌터의 돌연변이 탐색
구조조정 : 광산그룹의 대량 폐쇄

(그림 4-3) 광산타운의 생명주기



(a) 학습 모델 (b) 자기조직화 모델

(그림 4-4) 두 적응 모델의 3계층 구조



(그림 4-5) Mine 알고리즘의 자기조직화 구조

없으며 당연히 부산물도 없다. 양자를 비교하면, 자기조직화 쪽이 좀더 가볍고 빠르지만 학습모델 쪽이 좀 더 정교하다. (그림 4-5)는 Mine 알고리즘에서의 자기조직화 구조이다.

첫째, 내부 계층에서는 네 종류의 행위자들이 독립적으로 일한다. 이들은 주변을 관찰하고, 그 결과에 반응하는 단순 행위를 거의 주기적으로 반복한다. 이들의 동작은 정적이고 환경의 변화와는 거의 무관하다. 둘째, 재조직 계층에서는 헌터의 탄생과 소멸, 광산의 탄생과 소멸, 광산의 이동, 광산의 팽창과 수축, 폐광, 폐광의 회복, 광부와 헌터의 비율 변화 그리고 구조조정 등의 다양한 사건들이 비주기적으로 일어난다. 이 모든 사건들은 환경 변화에 대응하는 재 조직화 과정으로 볼 수 있다. 셋째, 우리가 필요로 하는 시스템 차원의 적응능력은 외부계층에서 최종적으로 발현된다.

내부 계층의 정적이고 단순한 행위들이 재조직 계층에서 동적이고 복잡한 재 조직화 현상으로 표출되는 것은 자기조직화의 전형적인 특징이다. 물론 이런 역할을 담당하는 별도의 장치 같은 건 없다. 이 역할은 계층 내 혹은 계층 간에 내재되어 있는 다양한 형태의 요동, 되먹임, 상호작용 패턴에 의해 주도된다. 즉, 자기조직화 시스템의 성공 여부는 행위자들의 정적 행위 패턴보다는 요동, 되먹임, 상호작용 같은 동역학 패턴에 더 의존한다.

5. 실험 및 고찰

5장에서는 Mine 알고리즘의 범용성을 검증하기 위해서 소정의 실험을 수행한다.

5.1 실험 대상 문제

범용성이란 다양한 환경에서 고루 좋은 성능을 보이는 성질이다. 문제는 다양한 환경이 무엇인지 정확하게 정의하기가 어렵다는 점이다. 우리는 이 목적을 위해서 주의 깊게 문제 집합을 선정하였다. 문제 집합의 선정에는 특별히 세 가지 조건이 고려되었다. 첫째, 문제 집합이 광범위한 환경을 대변할 수 있어야 한다. 둘째, 문제집합이 여러 탐색 모델 중 어떤 쪽으로도 편향되지 않아야 한다. 셋째, 내부적 특징이 충분히 알려진 투명한 문제여야 한다.

우리는 이 세 가지 조건을 종합적으로 감안하여 단봉형 문제, 다봉형 문제, 사막형 문제 등 세 가지 표준문제를 선택하였다. 여기서 단봉형은 봉우리가 하나인 형태이고, 다봉형은 봉우리가 여러 개인 형태이며, 사막형은 사막처럼 평탄한 공간에 가끔씩 봉우리가 있는 형태이다. <표 5-1>은 본 실험에서 사용한 세 가지 표준문제에 대한 명세이다.

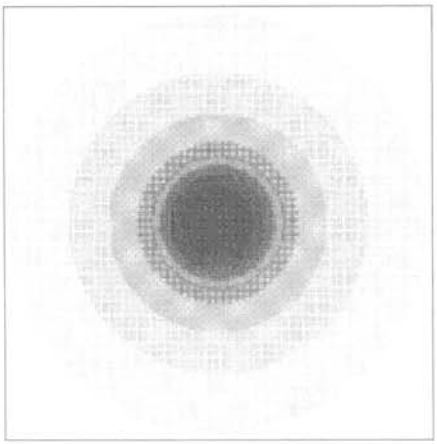
<표 5-1>에서 De Jong Ⅱ는 원래의 De Jong 함수(De Jong Ⅰ)를 약간 변형한 것이다. 즉, 원래 최대값을 찾는 문제인 것을 최소값을 찾는 문제로 바꾸었다. 실제 실험에서는 함수 F를 1/F로 바꾸어서 최대값을 구한다. 한편, 최소값을 찾는 문제인 표준문제 1의 경우도 F를 F1 = Max - F로 바꾸어서 F1에 대한 최대값을 구한다. 이렇게 하면 세 가지 표준문제 모두 최대값(점수로 표현)을 구하는 문제로 통일된다. 그리고 표준문제 1의 경우, 문제의 난이도를 다소 높일 필요가 있어서, 원래의 함수에서 해공간 영역을 100배 확대하였다.

〈표 5-1〉 세 가지 표준문제의 명세

항목	표준문제 1	표준문제 2	표준문제 3
문제이름	Sphere function	De Jong Ⅰ	De Jong Ⅱ
문제(함수)	$F=\sum_{i=1}^{5} x_i^2$	$F=\sum_{i=1}^{6} x_i \sin(\sqrt{\lvert x_i \rvert})$	$F=\sum_{i=1}^{6} x_i \sin(\sqrt{\lvert x_i \rvert})$
목표	최소값	최대값	최소값
제약조건(x_i)	[-10000, 10000]	[-500, 500]	[-500, 500]
차원	5	6	6
최적해	각 항목 : [0]	각 항목 : [420.9]	각 항목 : [0]
최적해 점수	0	2513.9	0
해공간의 특징	단봉형	다봉형	사막형

5.1.1 표준문제 1의 해공간

(그림 5-1)은 표준문제 1의 해공간에 대한 적합도 조망(fitness landscape)이다. 그림은 해공간 데이터 중 무작위로 일부 데이터만을 추출하여 그린 것이다. 즉, 개략적인 이미지이다. 5차원 좌표 중 2차원만 표현하였다. 보다 진한 이미지일수록 보다 높은 점수를 의미한다. 그림을 보면 봉우리가 한 개인데, 중심부로 갈수록 점진적으로 봉우리가 높아지는 것(즉, 점수가 높아지는 것)을 알 수 있다. 최적해는

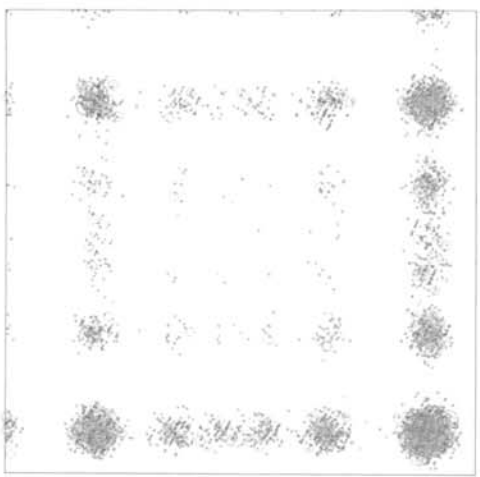


(그림 5-1) 표준문제 1의 해공간

당연히 중심점에 있다.

5.1.2 표준문제 2의 해공간

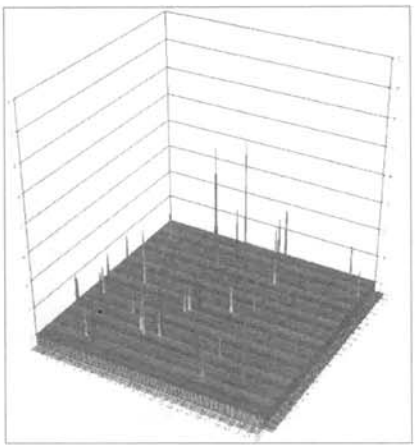
(그림 5-2)는 표준문제 2에 대한 해공간 조망이다. 역시 해공간 데이터 중 일부 데이터만을 추출하여 그린 개략적인 이미지이다. 여기서는 6차원 좌표 중 2차원만 표현하였다. 원으로 점수를 표현하였는데, 보다 큰 원일수록 보다 높은 점수를 의미한다. 그림에서 다수의 작은 봉우리들(즉, 지역해)을 볼 수 있다. 그 중 좌상, 좌하, 우상, 우하 등 4개의 봉우리가 큰 편이고, 최적해는 우하 쪽 봉우리에 있을 것으로 짐작된다.



(그림 5-2) 표준문제 2의 해공간

5.1.3 표준문제 3의 해공간

(그림 5-3)은 표준문제 3에 대한 해공간 조망이다. 이 그림 역시 해공간 데이터 중 일부 데이터만을 무작위로 추출하여 그린 개략적인 이미지이다. 6차원 좌표 중 2차원만 표현하였다. 이 경우는 3차원그래프 형식으로 표현하였는데, 이렇게 하면 해공간의 특징이 보다 분명하게 드러난다. 이



(그림 5-3) 표준문제 3의 해공간

공간은 사막같이 황량한 들판에 좁고 높은 봉우리들이 가끔씩 솟아 있는 데, 이런 형태의 해공간 탐색은 일반적으로 쉽지 않다.

5.2 비교 대상 모델

Mine 알고리즘에 대한 비교 대상으로는 GA(Genetic Algorithm), PSO (Particle Swarm Optimization), ACO(Ant Colony Optimization), HC(Hill Climbing), RS(Random Search), TS(Tabu Search), SA(Simulated Annealing) 등을 들 수 있는데, 이들 중 마지막 둘은 제외시켰다. 그 이유는 이들의 주요 기법들이 일종의 공유기법처럼 사용되고 있어서 더 이상 독립된 탐색 모델로 보기 어렵기 때문이다. 여기서 HC는 '다중시작 HC'를 의미한다.

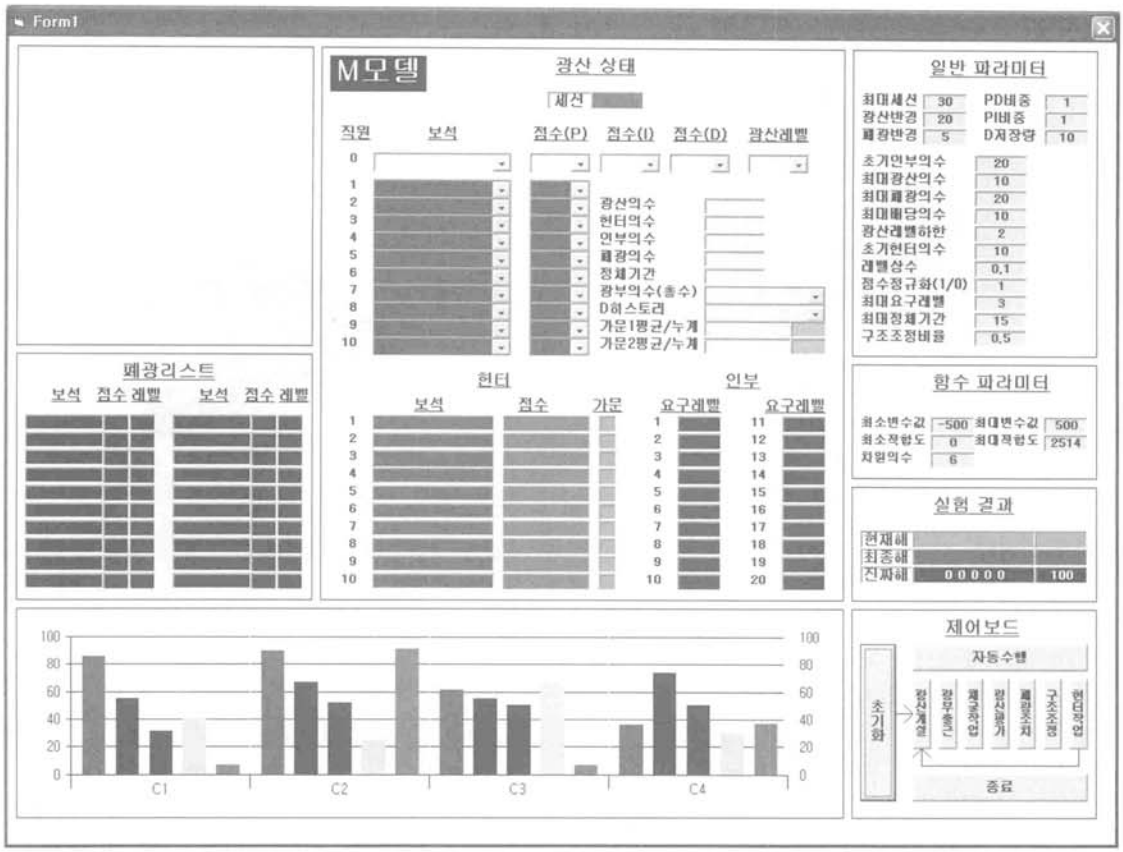
이들 다섯 개 메타휴리스틱들은 이미 소개 된지 오래 되었으며, 그동안 특별한 목표와 용도를 갖는 다양한 버전들이 수없이 발표되었다. 이들에 관한 종합적인 정보는 논문보다는 [18-21]등의 서적에서 상세하게 설명하고 있다.

본 논문의 실험에서는 이들 다섯 개 메타휴리스틱 모두에 대해서 교과서적인 표준 알고리즘을 사용하였다. 표준 알고리즘을 포함한 관련 정보 역시 상기의 서적들에서 찾아볼 수 있다. 참고로, 비록 공식적인 참고문헌으로 제시할 수는 없지만, Mine 알고리즘과의 비교 대상인 상기 다섯 개 모델에 대해서, 표준 알고리즘을 가장 알기 쉽고 명쾌한 형태로 소개하고 있는 곳은 웹의 위키피디아(Wikipedia) 영문 사이트이다.

5.3 Mine 알고리즘의 실험 환경

5.3.1 플랫폼

Mine 알고리즘의 실험을 위한 플랫폼은 (그림 5-4)와 같다. 그림에서 좌상부에서는 해공간에 대한 상태공간을 표현하며(상태공간의 이미지는 그림 5-5에서 보여줌) 좌 하부는 점수에 대한 성능 커브를 표현한다. 좌중부에서는 폐광에 관한 정보를, 우상부에서는 파라미터에 관한 정보를 그리고 중심부에서는 광산에 대한 전반적인 상태정보를 표현한다.



(그림 5-4) Mine 알고리즘의 실험을 위한 플랫폼

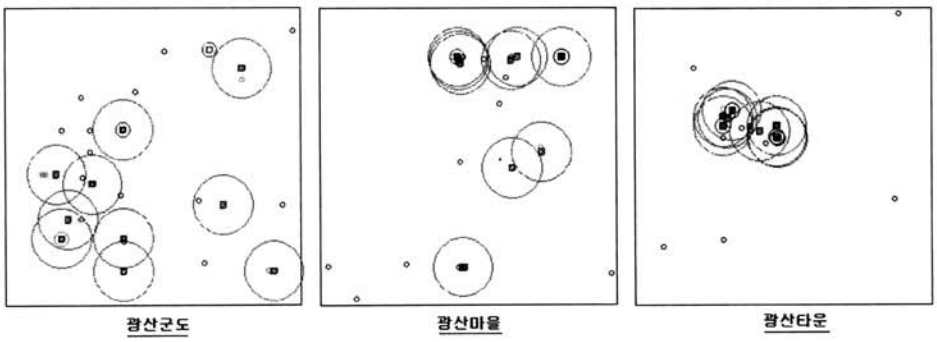
5.3.2 파라미터 설정

새로운 탐색 모델을 설계할 때 시간 투자가 가장 많은 부분 중 하나가 파라미터 설정(혹은 파라미터 튜닝)이다. 일반적으로 파라미터가 성능에 미치는 영향은 문제 영역마다 다르다. 일을 정말로 어렵게 하는 것은 파라미터 상호 간에 서로 영향을 주고받는다는 점이다. 이 때문에 최적의 파라미터 집합을 찾기 위한 실험의 가지 수가 기하급수적으로 증가한다. <표 5-2>는 Mine 알고리즘 실험에서 실제로 사용한 파라미터 값이다. 이것은 체계적인 실험을 통해서 찾은 값은 아니고 몇 번의 시행착오를 거쳐서 대략 구한 값이다.

이 문제는 너무 복잡하여 본 논문에서는 깊이 다루지 않는다. 다행히 Mine 알고리즘이 최적의 파라미터를 사용하지 않음으로서 Mine 알고리즘과 경쟁해야 하는 다른 모델이 당하는 불이익은 없다. 즉, Mine 알고리즘의 상대적인 경쟁력이 실험결과에 근거해서 주장될 때, 그 주장의 정당성은 이로 인해 훼손되지는 않는다. 참고로 <표 5-2>의 표현방법 항목에서 명시적이라 함은 파라미터를 밖에서 조정 가능하도록 독립적으로 표현한 것이고 묵시적이라 함은 파라미터 개념을 소스 코드 자체에 삽입한 것을 말한다.

5.4 광산타운 실험

(그림 5-5)는 전형적인 광산타운 형성과정을 보여준다. 그림에서 광산은 사장의 보석이 상징하는 해공간 좌표에서 일정한 반지름을 갖는 원으로 정의된다. 즉, 큰 원은 광산이



(그림 5-5) 상태공간에서 보여주는 전형적인 광산타운 형성 과정

〈표 5-2〉 Mine 알고리즘의 파라미터 집합

파라미터	파라미터의 범주	표현 방법	설정된 값	비고
회장의 수	n ≥ 1(n:정수)	묵시적	1	회장의 수는 그룹의 수를 말함
헌터의 탐색전략	우량해를 이용	묵시적	4개	우량해를 참조하여 설계함
헌터의 수명	n ≥ 1	묵시적	1	수명은 세션 단위로 설정
직병렬 모드	직렬 or 병렬	묵시적	직렬	프로세서의 수에 의해 결정됨
회장의 학습능력	True or False	묵시적	False	구조조정의 시점을 학습
사장의 학습능력	True or False	묵시적	False	폐광기준을 학습
광부의 학습능력	True or False	묵시적	False	광산레벨 요구 기준을 학습
헌터의 학습능력	True or False	묵시적	False	탐색 전략을 학습
광부의 탐색전략	free	묵시적	RS+HC	광산 내에서 어떤 전략도 가능
광부의 수	n ≥ 20	명시적	20	지향성은 광부의 수에 비례
최대폐광의 수	n ≥ 20	명시적	20	소요시간은 폐광의 수에 비례
광산의 최대배당	n ≥ 10	명시적	10	지향성에 긍정적 영향을 줌
광산의 의무배당	n ≥ 1	명시적	1	다양성에 긍정적 영향을 줌
헌터의 수	n ≥ 10	명시적	10	다양성은 헌터의 수에 비례
실험세션	n ≥ 50	명시적	100	세션=iteration
광산반경	f(성능, 광부의수)	명시적	20	성능 및 광부의 수에 반비례
폐광반경	n ≥ 1	명시적	5	지향성에 긍정적 영향을 줌
PD 비중	r ≥ 1(r:실수)	명시적	1	광산이 갖는 잠재력의 비중
PI 비중	r ≥ 1	명시적	1	광산이 갖는 저력의 비중
D의 저장량	n ≥ 10	명시적	10	I를 재는 기준
최대요구레벨	r ≥ 1	명시적	3	광부의 출근 거부의 기준
광산레벨 하한	r ≥ 1	명시적	2	폐광의 기준
최대정체기간	n ≥ 10	명시적	15	구조조정의 기준
구조조정비율	r ≥ 0	명시적	0.5	구조조정의 폭을 조정

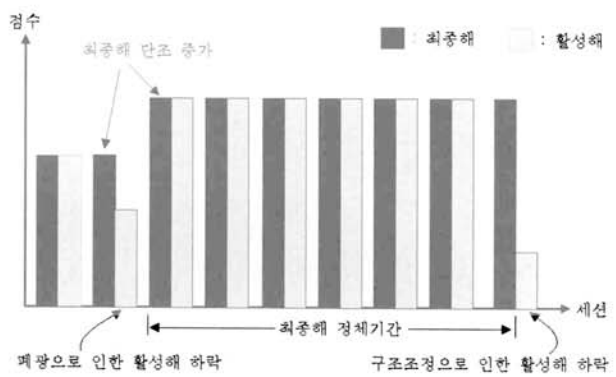
고, 아주 작은 원은 헌터의 작업장소이다. 광산그룹은 일반적으로 군도 형태에서 시작해서 마을 형태를 거쳐 타운 형태로 종결되어 간다. 이런 일련의 성장 과정에서 광산들 간의 중첩부분이 증가하는 것을 볼 수 있다. 극단적으로 모든 광산이 하나로 통합될 수도 있다.

5.5 성능 실험

성능 실험에서 우리가 주목해야 할 부분은 성능 커브에서 보여주는 점수의 변화 패턴과 그 과정에서 일어나는 폐광과 구조조정 부분이다.

5.5.1 성능 커브 상에서의 폐광과 구조조정

본 실험에서는 '활성해'와 '최종해'라는 용어를 사용한다. 활성해는 현재 활성화된 광산 중 최고 점수(즉, 최고의 보석)를 의미하고, 최종해는 지금까지 찾아낸 점수 중 최고 점수를 의미한다. (그림 5-6)의 성능 커브에서 활성해와 최종해를 볼 수 있다.



(그림 5-6) 성능 커브

폐광이 일어나는 형식에는 두 가지 유형이 있는 데, 하나는 광산이 열등하기 때문에 폐광되는 경우이고, 다른 하나는 상위 서열의 우수한 광산이지만 더 이상 개선 가능성이 없어서 폐광되는 경우이다. 둘 중 후자에 한해서만 성능 커브에서 관찰이 가능하다.

구조조정 역시 활성해의 하락을 야기하는 데, 다만 이 경우는 정체기간이 선행된다. 즉, 정체기간이 나타난 다음의 활성해 하락은 구조조정으로 인한 것이고, 그렇지 않은 활성해 하락은 폐광으로 인한 것이다. 한편, 최종해는 정체될 수는 있어도 하락하지는 않는다. 즉, 최종해는 항상 단조 증가한다.

5.5.2 성능 커브 분석

여기서는 세 가지 표준문제에 대하여 성능 커브를 분석한다.

5.5.2.1 표준문제 1의 경우

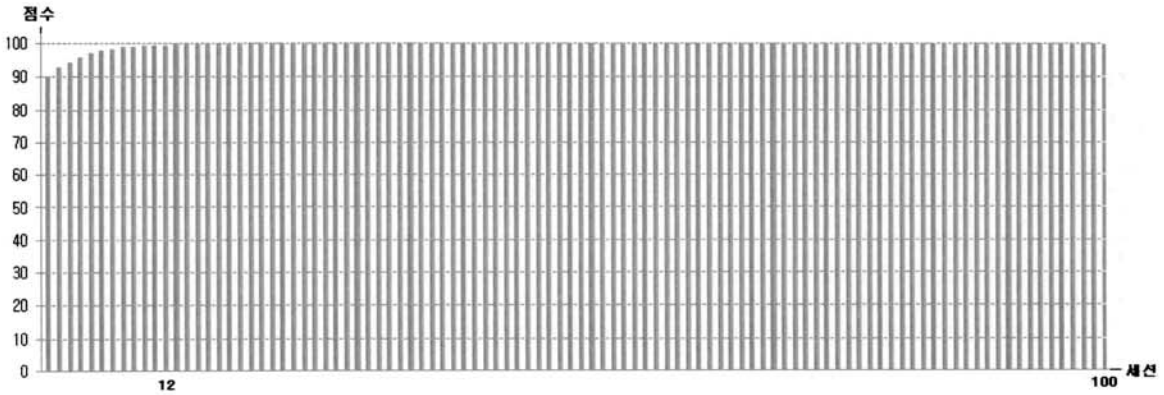
(그림 5-7)은 표준문제 1에 대한 성능 커브의 한 전형적인 예이다. 그림에서 커브는 처음 88점 정도에서 시작해서 세션 12지점에서 100점 만점에 도달한다. 표준문제 1은 연속이고 봉우리가 하나밖에 없는 단봉형 문제이다. 이것은 해공간 상의 어느 지점이든 구배(gradient)가 존재한다는 뜻이다. 따라서 광산은, 처음 어떤 위치에서 시작하든, 하나밖에 없는 봉우리(최적해)를 향해서 빠르게 이동해갈 것이다. 즉, 이 경우는 폐광이나 구조조정은 일어날 수 없다.

5.5.2.2 표준문제 2의 경우

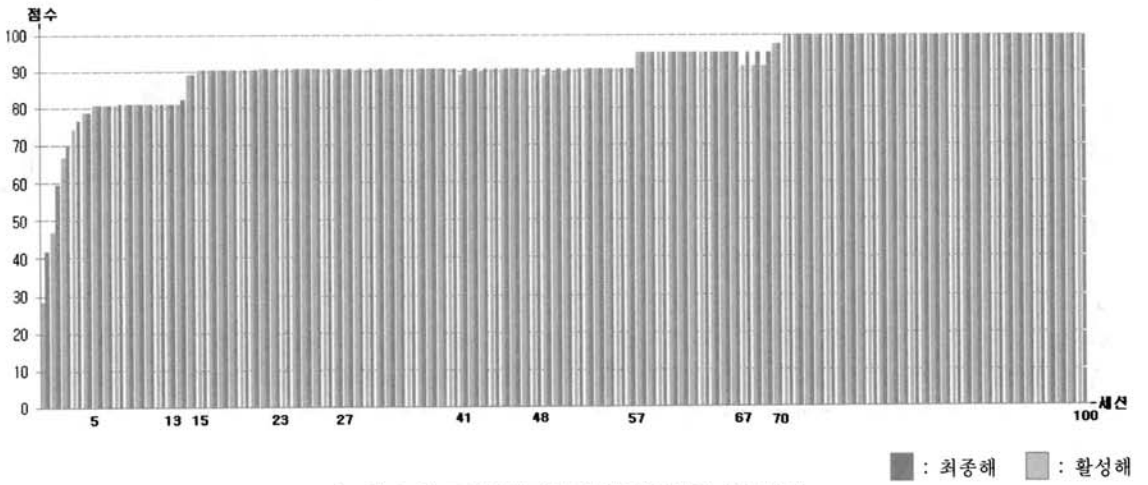
(그림 5-8)은 표준문제 2에 대한 성능 커브의 한 전형적인 예이다. 그림에서 커브의 특징을 요약하면 다음과 같다.

- 상승 구간 : [13~15], [15~23], 57, 70
- 급상승 구간 : [0~5]
- 폐광이 일어나는 지점 : 23, 27, 41, 48, 67
- 짧은 정체구간 : [5~13], [48~57]

표준문제 2는 봉우리가 여러 개 있는 문제이다. 즉, 이 공간에는 하나의 최적해와 여러 개의 지역해가 존재한다. 각각의 봉우리들은 그 안에 몇 개의 작은 봉우리들을 포함하므로 설사 봉우리를 제대로 선택했다고 해서 바로 최정상에 도달하는 것은 아니다. 예외적인 경우가 아니라면 작은 봉우리들을 선택한 광산은 폐광되거나 큰 봉우리 쪽으로 이동할 것이다. 즉, 대부분의 경우 광산타운은 큰 봉우리가 있는 지역에 형성될 것이다. 결국 이 문제의 경우는 폐광은 일어날 수 있지만 광산타운 자체가 붕괴되는 구조조정은 일어나기 어렵다.



(그림 5-7) 표준문제 1에 대한 전형적인 성능커브



(그림 5-8) 표준문제 2에 대한 전형적인 성능커브

5.5.2.3 표준문제 3의 경우

(그림 5-9)는 표준문제 3에 대한 성능 커브의 한 전형적인 예이다. 그림에서 커브의 특징을 요약하면 다음과 같다.

- 상승 구간 : [0~8],17, [33~44], 80
- 급상승 구간 : 13, 68, 92
- 폐광이 일어나는 지점 : 9, 62
- 구조조정
  - 1차 구조조정 = 33(정체구간 : [17~33])
  - 2차 구조조정 = 52(정체구간 : [44~52])
  - 3차 구조조정 = 74(정체구간 : [68~74])
  - 4차 구조조정 = 86(정체구간 : [80~86])
  - 5차 구조조정 = 97(정체구간 : [92~97])

표준문제 3은 넓은 사막에 가끔씩 봉우리가 있는 상당히 난해한 문제이다. 더구나 대부분의 봉우리는 좁고 높은 형태이다. 많은 지역은 높이(점수)가 0이거나 0에 가깝고, 해공간은 별다른 특징을 보이지 않는다. 이런 유형의 문제에서는 전형적인 탐색 기법들이 잘 듣지 않는다. Mine 알고리즘에서는 다양한 탐색기법을 동원하여 이 문제에 대처한다. (그림 5-9)를 보면 폐광이 두 번 일어나고 구조조정이 다섯 차례 일어나는 것을 볼 수 있다. 폐광이 두 차례만 관찰된 것은 폐광이 두 번 일어났다는 뜻은 아니다. 대부분의 경우 열등한 광산들이 폐광되는데, 이 경우 성능커브 곡선에서는 관찰되지 않는다.

5.6 결과 분석

여기서는 세 개의 표준문제에 대한 여섯 개 메타휴리스틱의 실험 결과를 시간과 점수의 관점에서 종합적으로 분석하고 정리한다.

5.6.1 개별 분석

5.6.1.1 표준문제 1

<표 5-3>은 표준문제 1에 대한 실험결과이다. 실험방법을 요약하면 다음과 같다. 첫째, 100회의 실험으로 목표점수(표의 괄호 안에 표현된 수치)에 도달하는 시간(초)을 비교하였다. 둘째, 점수는 100점 만점으로 정규화 하였다. 셋째, 등수는 99.9점과 99.7점을 대상으로 계산한 것이다. 넷째, '*'는 15초를 초과하는 시간을 나타낸다. 실험결과에 의하면 PSO 가 1위, Mine 알고리즘(MA)이 3위이다.

〈표 5-3〉 표준문제 1에 대한 모델 간 성능 비교표

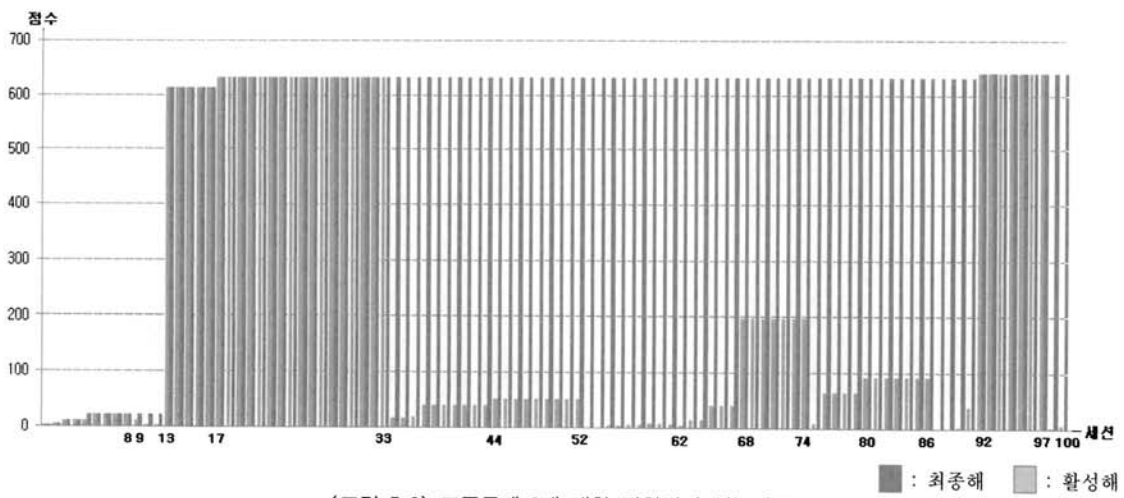
모델	시간 (99.9)	시간 (99.7)	시간 (99.5)	시간 (99.3)	시간 (99.0)	등수
PSO	1.00	0.85	0.80	0.75	0.70	1
GA	1.60	1.11	0.80	0.53	0.10	2
Mine	1.80	1.60	1.30	1.00	0.80	3
HC	3.40	3.15	3.00	2.80	2.60	4
RS	*	4.00	1.20	0.58	0.15	5
ACO	*	*	*	*	15.00	6

5.6.1.2 표준문제 2

<표 5-4>는 표준문제 2에 대한 실험결과이다. 실험방법을 요약하면 다음과 같다. 첫째, 100회의 실험으로 24초 동안에 취득한 점수를 비교하였다. 둘째, 점수는 100점 만점으로 정규화 하였다. 셋째, 등수는 평균점수 만을 고려한 것이다. 실험결과를 보면 Mine 알고리즘(MA)이 1위이고, HC가 2위이다.

〈표 5-4〉 표준문제 2에 대한 모델 간 성능 비교표

모델	평균점수	표준편차	시간(초)	등수
Mine	99.97	0.04	24	1
HC	99.51	0.19	24	2
GA	98.61	0.58	24	3
ACO	96.57	0.48	24	4
PSO	87.16	2.86	24	5
RS	79.69	0.14	24	6



(그림 5-9) 표준문제 3에 대한 전형적인 성능커브

5.6.1.3 표준문제 3

<표 5-5>는 표준문제 3에 대한 실험결과이다. 실험방법을 요약하면 다음과 같다. 첫째, 100회의 실험으로 24초 동안에 취득한 평균점수와 표준편차를 비교하였다. 둘째, 다른 문제들과는 달리, 점수는 100점 만점으로 정규화 하지 않았다. 셋째, 등수는 평균점수 만을 고려한 것이다. 실험결과를 보면 Mine 알고리즘(MA)이 1위이고, PSO 가 2위이다.

〈표 5-5〉 표준문제 3에 대한 모델 간 성능 비교표

모델	평균점수	표준편차	시간(초)	등수
Mine	16137	5382	24	1
PSO	4038	529	24	2
HC	2214	969	24	3
RS	1366	553	24	4
GA	356	99	24	5
ACO	78	19	24	6

5.6.2 종합 분석

(그림 5-10)은 여섯 개 탐색 모델에 대해서 등수만을 뽑아서 비교해본 그래프이다. Mine 알고리즘의 경우 전반적으로 양호하며 특히 범용성이 우수한 편이다. 범용성의 관점에서만 보자면 HC가 GA나 PSO보다 나은 점이 있다. 참고로, ACO가 전반적으로 부진한 이유는 ACO가 이런 유형의 문제에 잘 맞지 않기 때문이다. ACO는 순회판매원 문제처럼 최단거리 문제로 환원되는 문제에서는 대단히 양호한 성능을 보인다[22].

5.7 정당성의 논거

최적해 탐색 능력과 관련하여 수학적 논증이 가능한 메타휴리스틱은 없다. 만일 있다면 용어 자체가 잘못 사용된 것이다. 메타휴리스틱은 소프트컴퓨팅에 속하기 때문이다. 그러나 모든 메타휴리스틱은 비록 엄밀성은 부족하지만 어느 정도 설득력을 갖는 직관적 논거를 갖추고 있다. 즉, 모든 메타휴리스틱에는 나름의 자랑거리 즉, 브랜드 개념이 있다.

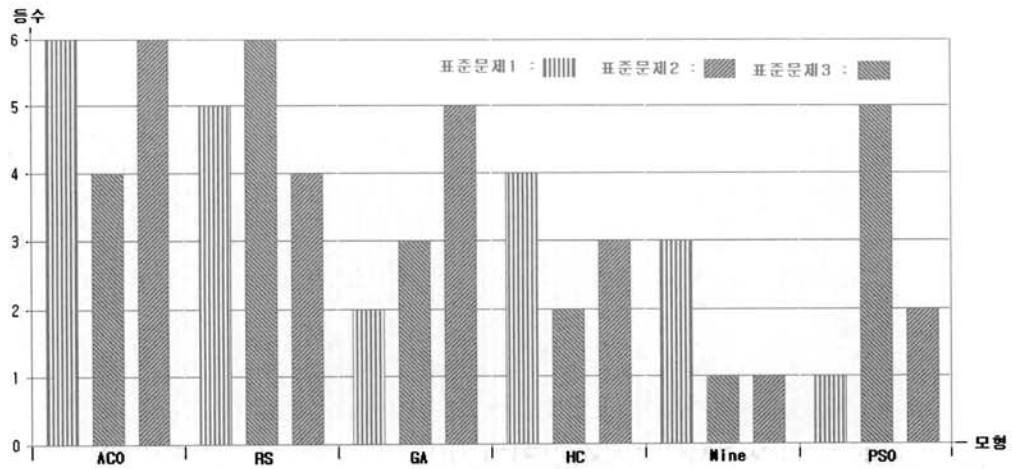
예를 들어, GA는 교차 개념이 그것이다(좋은 것 끼리 섞으면 좋은 것이 나온다는 주장은 확실히 일리가 있다). PSO는 모방 개념이 그것이다. ACO는 페로몬 개념이 그것이다. HC는 구배 개념이 그것이다. SA는 담금질 개념이 그것이다. 그리고 TS는 터부제약 개념이 그것이다. 심지어는 RS의 시행착오 전략도 나름의 설득력을 갖는다.

Mine 알고리즘이 내세우는 브랜드는 '시장' 개념이다. Mine 알고리즘이 갖는 힘은 다양한 탐색 기법을 자기조직화 기전으로 엮어내는 능력에서 나온다. 즉, 자기조직화가 진정한 의미의 자랑거리이다. 그런데 이 자기조직화 기전은 설계된 것이 아니라 이식된 것이다.

잘 알려진 것처럼, 시장경제의 시장(market)은 놀라운 적응 능력과 안정성을 보인다. 그러나 정작 시장을 구성하는 행위 주체들은 눈앞의 이익에만 몰두한다. 공익 개념 같은 건 이들의 안중에 없다. 시장 체계에서 시스템 레벨의 적응 능력과 안정성은 소위 보이지 않는 손이라고 불리는 자기조직화의 산물이다. 사회성을 가진 군집동물 중 시장 개념은 오직 인간 세계에만 있는 것이다. Mine 알고리즘은 광산업이라는 작은 시장을 베낀 것으로, 이 과정에서 시장 고유의 자기조직화 기전(mechanism)을 묵시적으로 이식하고 있다.

6. 결 론

광산 업무에는, 탐색 문제와 관련하여, 긴 세월 축적된 인간의 노하우와 휴리스틱이 녹아 있다. Mine 알고리즘은 특히 광산 업무에 초점을 맞추어서 인간의 행동 패턴을 모방한 해공간 탐색 모델 즉, 메타휴리스틱이다. Mine 알고리즘은 다양한 탐색 기법을 갖추고 있으며, 환경에 따라 이들 중 일부를 선택적으로 활용한다. 이런 적응 능력으로 인해



(그림 5-10) 6개 탐색 모델의 3개 표준문제에 대한 등수 비교

Mine 알고리즘은 개선된 범용성을 갖는다.

Mine 알고리즘의 범용성을 체크하기 위해서 단봉형 문제, 다봉형 문제, 사막형 문제 등 세 가지 표준문제를 선택하였다. 우리는 이 세 가지 표준문제가 다양한 환경을 어느 정도 대변할 수 있다고 가정하였다. 그리고 Mine 알고리즘과의 비교 대상으로는 GA, PSO, ACO, HC, RS 등 다섯 개 메타휴리스틱을 선택하였다.

실험결과 세 가지 표준문제 모두에서 Mine 알고리즘이 비교적 양호한 성적을 보이는 것이 확인되었다. 이것은 Mine 알고리즘의 범용성을 지지하는 유의미한 증거로 보인다. 그러나 이를 근거로 Mine 알고리즘이 다른 메타휴리스틱보다 일반적으로 우수하다고 말할 수는 없다. 메타휴리스틱마다 각자에게 잘 맞는 문제 유형이 따로 있기 때문이다.

사실 Mine 알고리즘이 범용성을 갖는다고는 하지만 만능은 아니다. 즉, 잘 맞는 문제 유형이 따로 있다. 금이든 석탄이든 모든 광물은 일정한 패턴을 갖고 매장되어 있다. 만일 광물이 사금처럼 온 산에 골고루 퍼져 있다면 광산 개념 자체가 성립하지 않는다. 즉, 광물이 매장된 패턴과 비슷한 유형의 해공간 문제가 Mine 알고리즘에 잘 맞는다고 할 수 있다.

한 마디로 메타휴리스틱이라고 하지만, 구체적으로 들어가면 특정한 용도를 갖는 소위 맞춤형 버전들이 얼마든지 있다. 또 여러 개의 메타휴리스틱을 섞은 하이브리드 메타휴리스틱 그룹과 여러 개의 프로세서를 사용하는 병렬 메타휴리스틱 그룹도 있다. Mine 알고리즘은 이제 겨우 교과서적인 표준형 버전들과의 비교 실험을 끝냈을 뿐이다. Mine 알고리즘의 특징과 장단점을 제대로 알아보기 위해서는 이들 다양한 버전들과의 비교 실험이 반드시 필요하다. 물론 검증용으로 사용하는 문제 집합의 폭도 더 확대할 필요가 있다. 대체로 이런 사안들이 다음 번 연구 과제라고 할 수 있다.

참 고 문 헌

[1] F. Glover, "Future paths for integer programming and links to artificial intelligence," Computers and Operations Research, Vol.13, No.5, pp.533-549, 1986.

[2] Christian Blum and Andrea Roli, "Metaheuristics in combinatorial optimization: overview and conceptual comparison," ACM Computing Surveys, Vol.35, No.3, pp.268-308, 2003.

[3] L. Bianchi, M. Dorigo, L.M. Gambardella, and W.J. Gutjahr, "A survey on metaheuristics for stochastic combinatorial optimization," Natural Computing, 2008.

[4] David E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," Addison-Wesley, 1989.

[5] M. Clerc, "Particle Swarm Optimization," ISTE, 2006.

[6] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," IEEE Computational Computer Science, Vol.344, pp.28-39, 2006.

[7] J. De Vicente, J. Lanchares, R. Hermida, "Placement by Thermodynamic Simulated Annealing," Physics Letters A, Vol.317, Issue 5-6, pp.415-423, 2003.

[8] Cvijovic, D.; Klinowski, J. "Taboo search - an approach to the multiple minima problem," Science 267, 664-666, 1995.

[9] Rusell, Stuart J. Norvig, Peter, "Artificial Intelligence : A Modern Approach (2nd ed)," Upper Saddle River, Nj: Prentice Hall, pp.11-114, 2003.

[10] Solis, F.J and Wets, R.J, "Minimization by random search techniques," Mathematics of operations research, Vol.6, No.1, pp.19-30, 1981.

[11] Moscato, P. "Memetic algorithms : A short introduction," In Corne, D., et al, eds: New Ideas in Optimization. McGraw Hill, pp. 219-234, 1999.

[12] Glover, F. Laguna, M. Marti, R, "Fundamentals of Scatter Search and Path Relinking," Control and Cybernetics 39(3) 653-684, 2000.

[13] Puchinger, J., Raidl, G.R, "Modles and Algorithms for Three-stage two-dimensional bin Packing," European Journal of Operational Research, Feature Issue on Cutting and Packing, 2006.

[14] Conor Ryan, J.J. Collins, Jj collins, Michael Oneill, "Grammatical Evolution: Evolving programs for an Arbitrary Language," Proceedings of the First European Workshop on Genetic Programming, 1998.

[15] 박찬란외, "유전자와 역전파 알고리즘을 이용한 효율적인 윤곽선 추출", 정보처리학회논문지, 제5권 제11호, pp.3010-3023, 1998.

[16] Alba, E., ed, "Parallel Metaheuristics, a New Class of Algorithms," John Wiley, New Jersey, 2005.

[17] Tan, Kok Kiong; Wang Qing-Gu, Hang Chang Chieh, "Advances in PID Control", London, UK: Springer-Verlag, 1999.

[18] 김여근, 윤복식, 이상복 공저, "메타휴리스틱", 영지문화사, 1997.

[19] 오일석, "패턴 인식", 교보문고, 2008.

[20] Gonzalez, Teofilo F., "Handbook of Approximation Algorithms And Metaheuristics," Taylor & Francis, 2007.

[21] Dreo, J., Petrowski, A., Siarry, P., Taillard, E., "Metaheuristics For Hard Optimization, Springer Verlag, 2005.

[22] L. Bianchi, L.M. Gambardella et M.Dorigo, "An ant colony optimization approach to the probabilistic traveling salesman problem," PPSN-VII, Seventh International Conference on Parallel Problem Solving from Nature, Lecture Notes in Computer Science, Springer Verlag, Berlin, Allemagne, 2002.

고 성 범

e-mail : kosb@kongju.ac.kr

1980년 숭전대학교 전기공학과(학사)

1983년 서울대학교 전기공학과(석사)

2003년 건국대학교 컴퓨터공학과(박사)

2005년~현 재 공주대학교 컴퓨터공학부 교수

관심분야 : 인공지능