개미 알고리즘을 융합한 적응형 유전 알고리즘

(An Ant System Extrapolated Genetic Algorithm)

김 중 항 † 이 세 영 † 장 형 수 ††
(Joong Hang Kim) (Se-young Lee) (Hyeong Soo Chang)

요 약 본 논문에서는 개미 군 집단 알고리즘을 융합한 새로운 적응형 유전 알고리즘을 제안하고, 제안된 알고리즘이 확률적으로 최적 해에 수렴함을 증명한다. 실험을 통해서, 제안된 알고리즘은 최적 해로의 수렴이 어려운 여러 가지 대표적인 함수들에 대하여 elitist 전략을 사용한 유전 알고리즘보다 더 빠른 속도로 최적 해에 수렴하고 한 군집 내의 모든 해들이 최적 해로 수렴하며 파라미터 값에 따라 새로운 탐색이나 현 상태로의 귀착의 정도를 조절할 수 있는 유연성 있는 알고리즘인 것을 보인다.
키워드 : 유전 알고리즘, 개미 군 집단 최적화

Abstract This paper proposes a novel adaptive genetic algorithm (GA) extrapolated by an ant colony optimization. We first prove that the algorithm converges to the unique global optimal solution with probability arbitrarily close to one and then, by experimental studies, show that the algorithm converges faster to the optimal solution than GA with elitism and the population average fitness value also converges to the optimal fitness value. We further discuss controlling the tradeoff of exploration and exploitation by a parameter associated with the proposed algorithm.
Key words : Genetic Algorithm, Ant Colony Optimization

1. 서 론

본 논문에서는, 개미 집단을 적용한 새로운 적응 유전 알고리즘(Genetic Algorithm, GA)을 제안한다. 이미 잘 알려져 있는 기본적인 유전 알고리즘(Canonical Genetic Algorithms, CGA)[1]은 수행 과정에서 주어진 최적화 문제에 대한 2진 비트들로 구성된 문자열(string)로 하나의 해를 인코딩시켜 각각의 비트를 고정된 확률로 변이(mutation)시킨다. 이에 반하여, 본 알고리즘에서의 각 비트의 변이는 2진 문자열의 각 비트에 대한 변이 확률 분포와 "군지능(swarm intelligence)"[2]을 적용하여 이루어진다. 본 알고리즘은 선택법(selection)과 교배법(crossover)의 종류에 무관하다.

CGA와 CGA의 수많은 변형들[3]은 보통 개체(chromosome)들이 가진 비트들의 상관 관계나 상호 작용을 이용하지 않는다. 본 알고리즘은 한 군집(population)내의 개체들의 사회적 행동(social behaviour)들에 기초하여 각 비트의 위치에 대한 변이 확률 분포를 갱신(즉, 각각의 개체들이 얼마나 최적의 개체인가에 대한 집합적인 지식을 만들어냄)함으로써 개체군들의 조망(landscape)을 학습하고, 다음 세대의 군집을 만들어내는 데 사용하게 된다.

최근, 비트들 간의 연관성(dependency) 학습이나 생성된 문자열 혹은 개체들 사이에서 어떤 패턴을 찾아내려는 노력의 패러다임은 연관성에 대한 확률적 모델을 이끌어내는 알고리즘들(Probabilistic Model Building Genetic Algorithms, PMBGAs)에 서 볼 수 있다. 또한 분포 추정 알고리즘(Estimation of Distribution Algorithms, EDAs)[4,5]으로도 알려져 있으며, 해당 패러다임을 기초로 한 여러 접근방식들은 CGA가 잘 동작하지 못하는 많은 문제에 대해서도 성공적으로 동작한다는 것이 알려져 왔다. 주된 아이디어는 2진 문자열이 담고 있는 부분적인 비트의 패턴이 다음 세대에도 나타날 수 있도록 높은 확률을 주어 해당 비트 패턴이 바뀌지 않도록 막는 것이다. 이것은 2진 문자열의 각각의 비트의

· 이 논문은 한국학술진흥재단의 지원에 의하여 연구되었습니다. (KRF-2004-003-D00294)
† 학생회원 : 서강대학교 컴퓨터학과
peropero@sogang.ac.kr
sylee@smolab.sogang.ac.kr
†† 정 회 원 : 서강대학교 컴퓨터학과 교수
hschang@smolab.sogang.ac.kr
논문접수 : 2005년 3월 2일
심사완료 : 2005년 5월 26일

위치를 확률 변수(random variable)로 나타내고, 각 변수들 간의 상호 관계를 나타내는 확률적 모델을 설계함으로써 만들 수 있으며 이 확률적 모델은 다음 세대를 생성하는 데 쓰인다. 그러나 이러한 알 수 없는 연관 관계(즉, 변수들의 결합 확률 분포(joint probability distribution))를 학습하는 것은 어려운 과정이다. 주어진 최적화 문제에 대한 전역 최적 해(global optimal solution)로의 수렴을 불확실하게 하고 부분 최적(local optimum)으로 수렴하게 하는 몇 가지의 근사 값들은 필연적으로 등장한다. PMBGA의 대부분의 경우에 사용된 모델은 베이지언 네트워크(Bayesian network)나 그것의 변형들이며, 이러한 경우에 더 좋은 근사치를 위하여 더 좋은 베이지언 네트워크를 선택하는 과정은 그것 자체가 또 다른 어려운 문제가 된다. 게다가, 교배나 변이 모두 새로운 군집을 생성하는 과정에 포함되어 있지 않기 때문에(교배와 변이는 PMBGA에서 기본적으로 사용되는 계산(calculation)과 추출(sampling)로 대치된다), PMBGA는 GA의 기본적인 성질을 잃어버리게 된다. 이에 반하여, 본 알고리즘은 CGA의 기본 구조에 최근에 개발된 군집 기반의 진화적 접근 방식인 개미 집단 최적화(Ant Colony Optimization, ACO)[6]의 방법론을 사용한 적응형 변이 과정을 적용하였고 전역 최적해에 수렴하는 것을 확률적으로 보장한다. 이것에 대한 증명은 Gutjahr의 Graph-based Ant Colony Optimization(GACO)[7]의 증명을 확장시킴으로써 가능하다.

ANT-GA로 불리는 본 알고리즘은 변이 연산에 있어서 Gutjahr의 GACO[7]의 확장을 기본으로 한다. ACO의 기본적인 아이디어를 설명하기 위한 가장 좋은 방법은 생물학적 은유(biological metaphor)를 참고로 하여, 생물학적인 개미 집단의 탐색 습성을 알아보는 방법이다. 개미들은 주어진 환경에서 다소 임의의 길로 이곳저곳을 움직이는 것처럼 보이지만, 개미들은 다른 개미가 지나간 길을 따라가는 성향을 보인다. 이렇게 개미들이 다른 개미의 길을 인식할 수 있는 것은 개미들이 움직일 때 페로몬(pheromone)이라고 불리는 화학 물질을 땅에 묻히기 때문이다. 어떤 경로(path)의 페로몬을 감지하는 것은 개미가 해당 경로를 따라갈 확률을 높이게 되고, 해당 경로를 지나가게 되면 또다시 페로몬을 묻혀서 해당 경로의 페로몬을 더욱 강화시키게 된다. 이 메커니즘은 시작점과 끝점 사이의 경로 중 더 짧은 경로에 더 많은 가중치를 주는 효과를 주는 데 사용될 수 있으며, 휴리스틱 최적화 작용(heuristic optimization behavior)의 한 종류가 된다.

위에서 기술한 근본 원리는 개미들의 이동을 시뮬레이션하고 목적 함수(objective function)를 추후 평가(post-hoc evaluation)하여 여러 개의 경로 중 더 좋은 경로를 강화함으로써, 임의의 조합 문제(combinatorial problem)의 목적 함수를 최적화하는 데 응용된다. Graph-based ant system[7]에서는 개미들의 이동을 나타내기 위한 기본 구조로 유향 그래프(directed graph)를 사용하며, 조합 최적화 문제의 가능한 해들은 이 그래프의 경로들로 인코딩된다. 최초에 개미들은 그래프상의 같은 node, 즉 출발점에 위치하며, 각 iteration마다 각각의 개미들은 현재 위치한 node에서 자신이 이동해 나갈 하위 node를 임의로 선택하게 된다. 각각의 하위 node로 이동할 확률은 그래프상의 edge, 즉 간선(arc)들에 부여된 "페로몬 값"으로부터 계산된다(추가로 "가시도 값(visibility value)"이라 불리는, 계속적인 개미들의 이동이 가능한지에 대한 적합도를 선평가(pre-evaluation)한 값이 사용될 수 있다). 모든 개미들이 이동을 끝내고 나면, 해당 이동 경로들은 문제에 대한 해로 디코딩되고, 목적 함수에 따라 추후 평가된다. 이 때 유망한(가능성 있는) 경로에 속한 간선들에 부여된 페로몬 값을 증가시켜서 강화시키게 된다. 그리고 이러한 전체적인 과정이 반복된다. 이 최적화 과정은, 주어진 문제에 대한 가장 좋은 경로를 찾아내는 과정(GA에서 가장 좋은 문자열을 찾아내는 과정)으로 ANT-GA의 변이 단계에 외삽(extrapolation)된다.

기본적인 시스템의 개요를 간단한 예를 들어 설명하면, 현재 어떤 node n_1에서 나와서 다음 node n_2로 들어가는 2개의 간선을 가지는 그래프가 있고, 개미가 node n_1에 있다고 할 때, GA에서 하나의 해를 나타내는 하나의 2진 문자열 중 한 개의 비트가 가지는 값을 개미가 다음 node인 n_2로 연결된 2개의 간선(편의상 e_1, e_2라 칭한다)중 하나를 선택하여 이동하는 일련의 동작에 대응시킨다. 즉, n_1에 도착한 개미가 n_2로 이동하기 위해 e_1을 따라 움직였을 경우는 해당 위치의 bit가 0을 가지는 것으로 보고, e_2를 따라 움직였을 경우는 1의 값을 가지는 것으로 본다(그림 1).

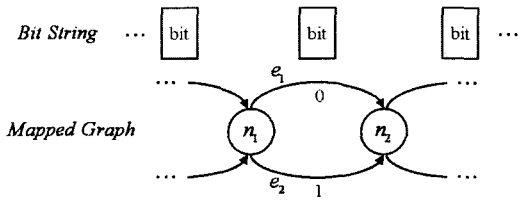


그림 1 2진 문자열과 그래프의 대응 관계

이제 그래프 상에서 출발점에서 도착점까지의 개미의 이동을 각각의 비트에 대응시키고 나면, 적합도 평가 함수에 따라 개미의 이동에 대한 적합도를 평가하고 평가

된 값에 따라 각 간선에 묻힐 페로몬 값을 계산한다. 이제 이러한 페로몬 값에 의하여 해당 비트의 확률 분포가 변화하게 된다. 이러한 확률 분포는 ANT-GA의 변이 단계에서 해당 비트를 어떤 특정한 값(0 또는 1)으로 변이시키는 데 사용된다.

즉, 어떤 간선에 대한 페로몬 값이 많으면 많을수록, 변이 과정에서 해당 간선에 대응되는 비트로 변이할 확률을 높여주게 되는 것이다. 이러한 페로몬 값의 생성 과정은 전적으로 그래프 상에서의 개미의 이동에 의한 것이며, 이러한 생성 과정이 GA에서의 변이 과정에 결합된다.

이제 GA의 기본적인 선택, 교배 작용과 새롭게 결합된 변이 과정을 가진 GA를 수행함으로써 알고리즘을 GA로 확장시킨다. 짧게 요약하면, 주어진 군집(GA의 2진 문자열의 집합)에서 특정한 선택과 교배를 하여 임시의 개체군을 얻고, 얻어진 개체군의 각각의 해를 개미의 그래프 상에서의 이동에 대응시킨 뒤, 개미들의 이동을 평가한 값을 가지고 그래프 상의 간선에 묻혀질 페로몬 값을 갱신한 뒤, 이것을 이용하여 각각의 비트가 가지는 특정한 값들에 대한 확률 분포를 갱신한다.

변이 과정에서는 새롭게 만들어진 한 문자열에 대한 각 비트의 확률 분포로 현재 군집의 각각의 해들을 변이시켜 새로운 다음 세대의 군집을 생성한다(보다 자세한 내용은 2장 참조).

변이 확률 P_m이나 교배 확률 P_c를 적합하게 조절하는 CGA의 다양한 시도들은 이미 존재하고 있다[8-10]. 하지만, 이러한 것들은 모두 변이 연산을 함에 있어서 적합하게 조절된 고정 변이 확률 P_m을 각 비트에 대하여 동일하게 적용하여 변이 연산을 수행하며, 풀려고 하는 어떤 문제에만 잘 들어맞도록 휴리스틱하게 P_m을 조절하기 때문에, 일반적으로 전역 최적 해에 수렴하는 것을 보장하지 못한다.

Rudolph[1]는 elitist 전략(elitist strategy)을 사용한 CGA가 수렴하는 것을 보였다. 본 논문과 관련된 가장 최근의 연구는 Baulja[11]의 각각 다른 변이 확률 갱신을 하는 군집 기반의 증가 학습(Population Based Incremental Learning, PBIL)이다. 그러나 이 연구에서 수렴성에 대한 분석은 존재하지 않는다. 본 논문에서는, 제안된 ANT-GA가 1에 임의로 근접한 확률로 전역 최적 해에 수렴한다는 것을 증명한다.

본 논문의 개괄적인 차례는 다음과 같다. 2장에서는 ANT-GA의 세부 알고리즘에 대한 설명과 ANT-GA가 전역 최적 해로 수렴함을 증명하고, 3장에서는 잘 알려져 있는 여러 가지 전역 최적화 문제에 대해서 실제적인 시뮬레이션을 통하여 ANT-GA의 성능을 평가하였다. 마지막으로, 4장에서의 결론으로 논문의 끝을 맺는다.

2. Graph-based ACO의 융합 : ANT-GA

2.1 알고리즘 설명

$X=\{0,1\}^n$가 길이 $n<\infty$이고 $f: X\rightarrow R^+$인 양의 적합도 함수를 가진 2진 문자열의 유한 해 공간일 때,

$$\min_{x\in X} f(x)$$

인 최적화 문제를 생각해 보자. 문제의 목표는 $\min_{x\in X} f(x)$를 만족하는 $x^*\in X$를 찾는 것이다. 본 논문에서는 해 $x\in X$의 i번째 비트의 위치를 x_i로 나타낼 것이다.

ANT-GA의 각 단계에 대한 개념적인 설명은 다음과 같다(ANT-GA에서의 우성(Elite) 해는 현재 및 과거의 군집들과 임시 군집들 중에서 가장 좋은 해를 뜻하며, 이것은 elitist 전략에서의 elitist와는 그 의미를 달리한다).

ANT-GA
- Initialization:
  군집의 크기 $m>0$을 결정하고, $\mu\in(0,1]$을 결정한다. $k=0$으로 초기화한다.
  초기 군집 $P(0)$를 임의로 초기화하고, $x_0^e=\arg\min_{\{x\in P(0)\}} f(x)$으로 정한다.

- Loop:
  - 적합도 평가: $x\in X$에서, $f(x)$를 평가하여 $x_k^*=\arg\min_{\{x\in P(k)\}} f(x)$를 구한다.
  - 선택/교배: 선택/교배로 임시 군집 $P_I(k)$를 생성한다.
  - 변이 확률 θ의 적응:
    * 우성 문자열 x_k^e를 구한다. $x_k^e=\arg\min_{\{x\in \cup_{j=1}^{k} P(j)\cup P_I(j)\}} f(x)$
    * x_k^e를 가지고 식(1)과 (2)를 사용하여 개미의 페로몬 함수인 ψ함수 값을 갱신한다.
    * 식(3)으로 θ값을 갱신한다.
  - 변이:
    θ값을 통하여 $P_I(k)$를 변이시켜, 다음 세대의 군집 $P(k+1)$를 생성한다.
  - $k\leftarrow k+1$

$x_l\in X$이고 $l=1,...,m$(m은 전체 군집에 속한 해의 개수이며, 이 값은 알고리즘이 진행함에 있어서 고정된 값)인 현재 군집 $P(k)=\{x^1,...,x^m\}$으로부터, 각각의 k ($k\geq 0$)번째 iteration에서 $P(k)$의 각각의 해 혹은 문자열을 평가하여

$$x_k^* = \underset{\{x \in P(k)\}}{\arg\min} f(x)$$

를 얻는다. 즉, 현재 군집에서의 가장 좋은 해를 얻는다. 그 다음 해들을 선택하고, 임의의 방법으로 $P_c \in [0,1]$의 확률을 가지고 교배를 시행한다. 이 때 교배 방법은 1점 교배(one-point crossover), 2점 교배(two-point crossover), 균등 교배(uniform crossover), 혹은 P_c를 tuning한 적응 교배(adaptive crossover)등을 사용한다. 교배를 통하여, $x_c^l \in X$이고 $l=1,...,m-1$인 임시 군집 $P_I(k) = \{x_k^*, x_c^1, ..., x_c^{m-1}\}$를 생성한다. 본 알고리즘에서는 선택과 교배의 방법을 자유롭게 선택하여 $P(k)$로부터 $m-1$개의 문자열을 생성하는 것을 가정한다. 이 때, 만약 $P_c = 0$이면 $P(k) = P_I(k)$이다.

이제, $\theta_i(k)$가 길이가 n인 어떤 2진 문자열의 i번째에 위치한 비트가 '1'의 값을 가질 확률('0'의 값을 가질 확률은 $1-\theta_i(k)$)이라 하고, $i=1,...,n$에 대하여 변이 확률 $\theta_i(k)$를 갱신한다(CGA의 경우, $\theta_i(k) = P_m \in (0,1)$이다. 즉, 문자열의 각각의 비트에 대한 변이 확률은 변하지 않는다). 변이 확률 $\theta_i(k)$가 갱신되는 동안에, 우성 해를 함께 갱신한다. 즉, iteration $k \geq 1$일 때, $P(0) \cup P_I(0) \cup \cdots \cup P(k-1) \cup P_I(k-1)$중에서 우성 해 x_k^e,

$$x_k^e = \underset{\{x \in \cup_{j=0}^{k-1} P(j) \cup P_I(j)\}}{\arg\min} f(x)$$

를 구한다.

이제, 갱신된 변이 확률을 가지고 $P_I(k)$의 각각의 해를 변이시켜 $P(k+1)$을 얻는다. 예를 들면, $n=3$인 k번째의 iteration에서, 2진 문자열 001은 확률 $\theta_1(k)(1-\theta_2(k))(1-\theta_3(k))$로 인하여 100으로 변이된다.

변이 확률들은 $P_I(k)$의 해들의 적합도 값과 우성 해, 즉 x_k^e(강조된 "유망한" 해)의 적합도 값을 기초로 하여 갱신된다. $P_I(k)$를 정렬된 집합$\{x^1,...,x^m\}$이라고 하고, 각각의 해에 대한 index인 $l=1,...,m$와 각각의 비트의 위치 $i=1,...,n$, 그리고 $s=0,1$에 대하여 개미의 페로몬 함수 ψ를 다음과 같이 정의한다.

$$\psi_k^l(i,s) = \begin{cases} \varphi(f(x^l)) \text{ if } f(x^l) \leq f(x_k^e) \text{ and } x_i^l = s \\ 0 \text{ otherwise.} \end{cases} \quad (1)$$

φ는 유망한 해를 강조하기 위한 감소함수(non-increasing function)이다. 예를 들어 $f(x^l) \leq f(x_k^e)$일 때, φ를 $\varphi(f(x^l)) = 1/f(x^l)$로 선택할 수 있다. 이는 일반적인 최적화 문제에서, 더 작은 목적함수 값을 가지는 해에 대하여 더 많은 페로몬을 부여하는 의미를 가진다.

만일 $\sum_{i',s'}\sum_{l=1}^{m} \psi_k^l(i',s') = 0$일 경우 각각의 i와 s에 대하여, $\psi_k(i,s) = \psi_{k-1}(i,s)$라고 하고, 그렇지 않을 경우는 $i=1,...,n$과 $s=0,1$에 대하여

$$\psi_k(i,s) = (1-\mu)\psi_{k-1}(i,s) + \mu \frac{\sum_{l=1}^{m} \psi_k^l(i,s)}{\sum_{i',s'}\sum_{l=1}^{m} \psi_k^l(i',s')} \quad (2)$$

을 사용하여 $\psi_k(i,s)$를 구한다. 계수 $\mu \in (0,1]$은 이전 값에 대한 강도를 조절하기 위한 알고리즘의 매개 변수로 사용된다.

ψ를 갱신하는 식 (1)과 (2)에 의하여, "유망한" 해의 비트 패턴, 즉 "유망한" 비트들의 위치와 그에 해당하는 0과 1의 값에 대한 ψ 함수 값은 증가되고, 비트의 값과 위치는 이후의 변이 과정에 의하여 다음 세대에서 더 많이 나타나게 된다.

$\theta_i(k)$의 갱신은 $i=1,...,n$에 대하여, 다음과 같다.

$$\theta_i(k) = \frac{\psi_k(i,1)}{\psi_k(i,0) + \psi_k(i,1)} \quad (3)$$

최초의 군집 $P(0)$는 임의의 2진 문자열들로 초기화하고, x_0^e는 $x_0^e = \arg\min_{x \in P(0)} f(x)$으로, ψ_{-1}^l은 $l=1,...,m$과 $i=1,...,n$, 그리고 $s=0,1$에 대하여 $\psi_{-1}^l(i,s) = 0.5$로 초기화 한다.

참고 1: 알고리즘의 성능 향상을 위하여, 확률의 연산에 $\eta_k(i,s)$라는 가시도 값(visibility value)을 추가할 수 있다 : 어떤 비트의 위치에 대하여 가망 있는 것으로 보이는, 우선(priori)으로 고려될 0 또는 1의 값을 선호할 수 있도록 인위적으로 수정한다. 만약 이러한 옵션이 선택된다면, 마지막의 식 (3)은 다음과 같이 바꾸어질 수 있다.

$$\theta_i(k) = \frac{\psi_k(i,1) \cdot \eta_k(i,1)}{\psi_k(i,0) \cdot \eta_k(i,0) + \psi_k(i,1) \cdot \eta_k(i,1)}$$

여기서 $\eta_k(i,s) > 0$은 비트의 위치 i에 대한 s의 선평가된 값이다.

변이 확률의 적응 과정을 요약하면, $P_I(k)$에 있는 각각의 해 $x^l (l=1,...,m)$은 페로몬 함수인 $\psi_k^l(i,s)$를 통하여 자신의 해에 대한 질(quality)을 우성 해의 질에 대비하여 측정하며, 이렇게 측정된 x^l의 질(quality), 즉 $f(x^l)$로 결정되는 질을 가진 x^l값을 가지고 해당되는 비트의 위치를 강조함으로써, 자신의 해인 x^l가 얼마나 "유망"한가에 대한 의견(opinion)을 피력한다. 그리고 이러한 $\psi_k^l(i,s)$값들은 해 공간 X에 대한 확률 분포에 의해, 가능한 최적 해(possible optimal solution)에 대한 전체 해들의 집단적인 소신(belief)인 $\theta_i(k+1)$을 만들기 위해 결합된다.

2.2 수렴(Convergence)

Theorem 2.1 다음의 조건들이 만족된다고 가정하자 :

(1) x^*는 유일하다.

(2) 모든 $k \geq 0$에 대하여, $x^* = (x_1^*,...,x_n^*)$이고, $i=1,...,n$, $s=0,1$일 때, 만일 $x_i^* = s$ 이면 $\eta_k(i,s) \in (0,1]$이다.

(3) 함수 $\varphi = \varphi_k$는 모든 $k \geq 0$에 대하여 만족하며, $f(x^l) \leq f(x_k^e)$일 때 $\varphi_k(f(x^l)) > 0$이고 나머지 경우에 대하여는 0이다.

이 때, 어떠한 $e>0$와 고정된 m에 대하여 μ가 충분히 0에 가까운 작은 값을 가지고 있다고 하면, 모든 $k \geq k_0$(k_0는 e에 의존하는 정수 값)에 대하여 $\Pr\{x^* \in P(k)\} \geq 1-e$이 성립한다.

증명 : $V=\{v_1,...,v_n,v_t\}$에 대하여 유향 그래프인 $G(V,E)$를 정의하자. $i=1,...,n$에 대한 각각의 정점(vertex) $v_i \in V$는 $i=1,...,n$에 대하여 확률 ρ_{is}로 $s \in \{0,1\}$의 값을 가지는 이산 확률 변수 X_i에 대응한다. 각각의 $i=1,...,n$에 대한 $v_i \in V$는 $e_{i0}=(v_i,v_{i+1}) \in E$와 $e_{i1}=(v_i,v_{i+1}) \in E$의 두 개의 간선과 연결되며, 최종(terminal) 정점은 $v_t(=v_{n+1})$로 정의된다. v_1에서 출발하여 각 node를 차례대로 거쳐 최종 node v_t에 도착할 때까지를 하나의 경로로 보고, 우리는 각 간선을 무작위로 선택하여 지나가는 하나의 무작위 경로로 그래프 위를 움직일 수 있다. 각각의 정점 v_i에서 어떤 간선 e_{is}를 통하여 움직일 확률은 ρ_{is}에 의하여 결정된다. 이러한 무작위 경로에 의하여, 무작위의 2진 문자열이 생성된다.

다음의 알고리즘 A를 생각해 보자 : 알고리즘 A의 각 iteration $k \geq 0$에 대하여, 그래프 G상의 m개의 무작위 경로로 부터, $x^l \in X$, $i=1,...,m$인 현재 군집 $P_A(k)=\{x^1,...,x^m\}$을 $\rho_i(k)$를 고려하여 생성한다. 이 때, $\rho_{i1} := p_i(k)$이고, $p_i(0)=0.5$이다.

이제 어떤 함수 $\pi_k^l(i,s)$로부터 $\rho_i(k)$를 갱신한다. $l=1,...m$과 $i=1,...,n$, $s=0,1$에 대하여,

$$\pi_k^l(i,s) = \begin{cases} \varphi_k(f(x^l)) & \text{if } f(x^l) \leq f(x_k^b) \text{ and } x_i^l = s \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

φ_k는 각각의 k에 대하여 $f(x^l) \leq f(x_k^b)$일 때 $\varphi_k(f(x^l)) > 0$, 나머지 경우에 대하여는 0이고

$$x_k^b = \underset{\{x \in \cup_{i=0}^{k-1} P_A(i)\}}{\operatorname{argmin}} f(x)$$

이며, 이 때 $x_0^b = \operatorname{argmin}_{x \in P_A(0)} f(x)$이다.

만일 $\sum_{i',s'} \sum_{l=1}^{m} \pi_k^l(i',s') = 0$일 경우에 각각의 i와 s에 대하여, $\pi_k(i,s) = \pi_{k-1}(i,s)$라고 하고, 그렇지 않을 경우에는 상수 $\mu \in (0,1]$에 대하여,

$$\pi_k(i,s) = (1-\mu)\pi_{k-1}(i,s) + \mu \frac{\sum_{l=1}^{m} \pi_k^l(i,s)}{\sum_{i',s'} \sum_{l=1}^{m} \pi_k^l(i',s')}$$

이라고 한다.

마지막으로, $\rho_i(k)$를 $i=1,...,n$에 대하여 다음과 같이 갱신한다.

$$\rho_i(k) = \frac{\pi_k(i,1)}{\pi_k(i,0) + \pi_k(i,1)}$$

모든 $l=1,...,m$과 $i=1,...,n$, $s=0,1$에 대하여 $\pi_{-1}^l(i,s) = 0.5$로 초기화 한다. 이러한 알고리즘 A는 다음과 같은 수렴 성질을 가지고 있다는 것이 Gutjahr의 Lemma 4.1부터 4.4[7]까지에 의해서 증명되었다. 해당되는 Lemma들을 간단하게 요약하면, Lemma 4.1은 GACO의 어떤 무작위 경로가 iteration k 이전의 iteration에서는 어떤 최적 해 x^*를 생성하지 않은 상태에서, k번째의 iteration에 도달하여 적어도 하나의 x^*를 생성할 확률의 하한(lower bound)은

$$P_{x^*} = 1-(1-(1-\mu)^{nk-n}p)^m$$

이며, 이 때 $\gamma = \min_{k,i,s,x_i^l=s} \eta_k(i,s) \in (0,1]$이고 $p = \gamma^n \prod_{i,s,x_i^l=s} \pi_n(i,s)$라는 것을 말한다. Lemma 4.2부터 4.4까지는 ACO의 그래프 G 상의 무작위 경로가 iteration k에서 최초로 x^*를 생성하는 확률의 조건이 되는 iteration의 횟수 j($j>k$)에 관한 조건부 확률(conditional probability)에 대한 것이다.

Lemma 2.1 Theorem 2.1에서의 조건 (1)과 (2)가 성립한다고 가정하면, 알고리즘 A에 대하여 다음의 조건 (3′)이 성립한다:

(3′) 함수 $\varphi = \varphi_k$는 모든 $k \geq 0$에 대하여 만족하며, $f(x^l) \leq f(x_k^b)$일 때 $\varphi_k(f(x^l)) > 0$이고 $f(x^l) > f(x_k^b)$일 때 0이다.

이 때, 어떠한 $e>0$와 고정된 m에 대하여 μ가 충분히 0에 가까운 작은 값을 가지고 있다고 하면, 모든 $k \geq k_0$(k_0는 e에 의존하는 정수 값)에 대하여 $\Pr\{x^* \in P_A(k)\} \geq 1-e$이 성립한다.

또한, 다음의 Lemma 2.2가 참이라는 것 역시 [7]의 증명을 변형하여 쉽게 증명할 수 있다.

Lemma 2.2 Lemma 2.1에서의 조건 (1)과 (2)가 성립한다고 가정하면, 다음과 같은 φ함수에 대한 수정된 조건 (3″)이 성립한다:

(3″) 함수 $\varphi = \varphi_k$는 모든 $k \geq 0$에 대하여 만족하며,

$\varphi_k(f(x^l))>0$이고 $f(x^l)>f(x_k^b)$일 때 0이다.

이 때, 어떠한 $\epsilon>0$와 고정된 m에 대하여 μ가 충분히 0에 가까운 작은 값을 가지고 있다고 하면, 모든 $k\geq k_0$(k_0는 e에 의존하는 정수 값)에 대하여 $\Pr\{x^*\in P_{A'}(k)\}\geq 1-e$이 성립한다.

Lemma 2.2의 수정된 조건을 가진 알고리즘을 A'라 하자. ANT-GA에 대한 Theorem 2.1의 증명은 직접적으로 알고리즘 A'에 대한 Lemma 2.2를 따른다. 처음에 해 공간 X의 각 문자열이 그래프 G의 하나의 경로로 나타내어지는(혹은 그 반대) G를 정의하였고, 가정에 의하여 최적 문자열 x^*가 유일하기 때문에, Theorem 2.1에 명시된 조건 (1)은 Lemma 2.2에 의하여 만족된다. 조건 (2) 또한 가정에 의하여 만족된다.

ANT-GA에 있어서, $x_0^b=x_0^e$일 때,

$$x_k^b:=\underset{\{x\in\cup_{j=0}^{k-1}P(j)\}}{\arg\min} f(x)$$

로 정의한다. 이제 x_k^b의 정의에 의하여, 모든 k에 대하여 $x_k^e\leq x_k^b$임을 알 수 있다. 그러므로 ANT-GA에서 φ의 선택으로부터, 위에서 정의된 x_k^b에 대하여 $f(x^l)>f(x_k^b)\geq f(x_k^e)$일 때 $\varphi(f(x^l))=0$이다. 게다가, $f(x^l)\leq f(x_k^e)$에 대하여$\varphi(f(x^l))>0$이고 모든 k에 대하여 $f(x^*)\leq f(x_k^e)$이기 때문에, $\varphi(f(x^*))>0$이다. 따라서 조건 (3")는 만족된다. 이는 ANT-GA에 의해서 생성된 x_k^b의 순열이 알고리즘 A'에 의해서도 생성될 수 있으며, Lemma 2.2의 수렴 성질에 의하여 x^*로 수렴한다는 사실을 내포한다. □

참고 1: 다음의 문장 또한 성립될 수 있다: 각각의 $e>0$와 고정된 μ에 대하여, m이 충분히 큰 값을 가지고 있다고 할 때, 모든 $k\geq k_0$(k_0는 e에 의존하는 정수 값)에 대하여 $\Pr\{x^*\in P(k)\}\geq 1-e$이 성립한다.

참고 2: x^*의 유일성에 대한 조건은 적합도 함수에 대한 약간의 변형, 즉 어떠한 준최적(sub-optimal)해가 실제 최적 해로 나타나지 않도록 하는 변형에 의하여 항상 만족될 수 있다: 예를 들어, $\chi(x)$가 어떤 고정 순서로 정렬된 해 $x\in X$의 번호(index)라고 하자. 충분히 작은 값 $e>0$에 대하여, 각 해 $x\in X$의 적합도 값에 $e\cdot\chi(x)$를 추가한다. 그러면 최적 해의 유일성(uniqueness)이 얻어진다.

2.3 확장(Extensions)

2.3.1 제한 조건 문제들(Constrained problems)

$c>0$는 상수이고, $g: X\rightarrow R^+$인 양의 적합도 함수를 가진, 다음과 같은 X의 부분집합 Δ를 생각해 보자.

$$\Delta=\{x|g(x)\leq c, x\in X\}$$

여기서의 목적은, Δ가 공집합이 아니라고 가정했을 때

$$\min_{x\in\Delta} f(x)$$

을 만족하는 $x^*\in\Delta$를 구하는 것이다.

ANT-GA에 간단한 변형을 가하여, 제한 조건 문제에 대한 수렴하는 알고리즘을 만들 수 있다. 식 (1)의 ψ-함수의 갱신 법칙으로부터, 비제한(unconstrained) 조건 문제에 대한 알고리즘의 수렴을 보장하기 위하여 φ는 다음을 만족해야 한다:

$$\begin{aligned}&\varphi(f(x^l))>0 \quad \text{if } f(x^l)\leq f(x_k^e)\\ &\varphi(f(x^l))=0 \quad \text{otherwise.}\end{aligned}$$

φ가 다음을 만족할 필요가 있도록, 간단하게 φ의 요구를 바꾼다.

$$\begin{aligned}&\varphi(f(x^l))>0 \quad \text{if } f(x^l)\leq f(x_k^e) \text{ and } g(x^l)\leq c\\ &\varphi(f(x^l))=0 \quad \text{otherwise.}\end{aligned}$$

변형에 의하여, ANT-GA는 제한을 만족하는 "유망한" 해에만 초점을 맞춘다.

이러한 간단한 변형으로 다음의 결과를 보일 수 있다 (증명은 본질적으로 Theorem 2.1의 증명과 비슷하므로, 생략한다):

Theorem 2.2 Theorem 2.1의 성립 조건을 가정한다. 변형된 φ-함수를 가진 ANT-GA에 대하여, 다음은 참이다:

1. 어떤 $e>0$와 고정된 μ에 대하여, 군집의 크기 m이 충분히 큰 값을 가지고 있다고 할 때, 모든 $k\geq\tilde{k}_0$ ($\tilde{k}_0$는 e에 의존하는 정수 값)에 대하여 $P_k\geq 1-e$이 성립한다.

2. 어떤 $e>0$와 고정된 m에 대하여, μ가 충분히 0에 가까운 작은 값을 가지고 있다고 하면, 모든 $k\geq\tilde{k}_0$($\tilde{k}_0$는 e에 의존하는 정수 값)에 대하여 $P_k\geq 1-e$이 성립한다.

2.3.2 PMBGA의 ANT-GA로의 융합 기법

가시도 값인 η을 사용하기 위한 접근 방식은 PMBGA나 EDA의 경우와 같이 표본(sample)(여기서는 2진 문자열의 표본)으로부터의 확률 변수 사이의 관계를 학습하고 해당 관계의 정보를 가시도 값을 추출하는 데에 사용하기 위한 것이다. 유망한 해에 대하여 더 많이 집중하게 하는(유망한 해, 즉 유망한 경로 위에 있는 각각의 간선은 공동으로 가중치를 얻는다) ψ-함수 값의 갱신 과정은, 해 공간 X의 알려지지 않은 구조를 찾아낼 수 있다는 점에 주목하자.

$i=1,\ldots,n$에 대하여, Y_i는 어떤 비트의 위치 i에 대하여 $\{0,1\}$의 값을 가지며, 이러한 Y_i에 대하여 정렬된

확률 변수 $Y=\{Y_i\}$의 결합 분포인 $p(Y)$를 평가 또는 학습하기 위하여, 다음과 같은 식을 얻는다.

$$P(Y) = p(Y_n|Y_{n-1},...,Y_1)p(Y_{n-1}|Y_{n-2},...,Y_1)\cdots p(Y_1)$$

여기서 $p(Y_n|Y_{n-1},...,Y_1)$는, 확률 변수 $Y_{i-1},...,Y_1$와 연관된 비트의 위치들을 가지고 생성된 확률 변수 Y_i, 즉 "부분 해(partial solution)"와 연관되어 어떤 비트 위치의 특정 값(0 또는 1)에 대한 선택/사상(mapping)의 확률로 해석된다. 각 iteration k에 대하여, $p_k(Y)$를 이용하여 $p(Y)$를 평가하고, Y_i에 관련된 비트의 위치 i에 대하여, 가시도 값인 $\eta_k(i,s)$를 다음과 같이 구하여 사용한다.

$$\eta_k(i,s) = p_k(Y_i = s|Y_{i-1},..., Y_1)$$

이러한 접근 방식을 위한 [12]와 [12]의 참고 논문들은 조합 최적화(combinatorial optimization)를 위한 베이지언 네트워크에 기반한다. 수렴성을 보존하기 위하여, $\eta_k(i,s)$는 양의 값을 가져야 한다. 따라서, 만약 가시도 값이 수용될 경우, 모든 k에 대하여 p_k에 의하여 얻어진 가시도 값이 0이라면, $\eta_k(i,s)$의 값을 매우 작은 양의 숫자로 대치할 필요가 있다.

3. 성능 평가

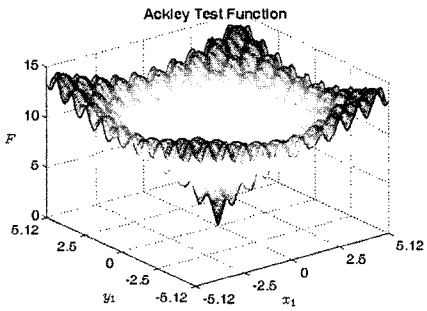
ANT-GA의 성능을 평가하기 위하여, elitist전략을 사용한 유전 알고리즘(Genetic Algorithm with Elitism, EGA)이라 불리우는 알고리즘[13]을 평가 대상으로 삼았다. 두 가지의 알고리즘은 동일한 기계와 환경에서 실험하였다.

ANT-GA와 EGA를 구현하기 위한, 문제에 대한 해들의 인코딩 및 디코딩은 실수의 디지털 샘플링 기법(quantization of real value)[14]을 사용하여 실수를 문자열의 길이 $N=30$의 2진 문자열로 인코딩하였다.

초기 군집인 $P(0)$에서의 임의로 설정된 각 해들은 동일하게 적용하였으며, ANT-GA와 EGA에서의 매개 변수의 설정은 De Jong과 Spears가 제시한 함수 최적화에 대한 일반적인 GA의 최적 매개 변수 설정[15]을 사용하였다. ANT-GA, EGA 모두 개체군의 수는 50으로 고정하고, 토너먼트 선택법(tournament selection)과 2점 교배법을 사용하였다. 교배의 경우, 교배된 해가 다음 세대의 전체 개체군에 포함될 비율(crossover fraction)을 0.6으로 설정하였다. EGA의 경우에는 변이 확률을 0.001로 고정하였으며, 변이의 종류는 비트-플립(bit-flip)방식을 사용하였다. 이때의 변이 확률은 각 문자열별의 변이 확률이 아니라 문자열의 각 비트별 변이 확률이므로, 각 문자열별의 변이 확률로 볼 때 변이 확률이 매우 낮은 것이 아님을 염두에 두어야 한다. 한편 ANT-GA의 경우는 μ값을 0.1로 고정한 뒤 실험을 수행하였다. ANT-GA에 추가되는 또 다른 매개 변수인 μ값에 따른 알고리즘의 성능 변화에 대해서는 뒤에서 자세히 설명하였으며, 실험에서는 ANT-GA와 EGA를 30번 수행한 후 각 세대별 최적 해와 평균 해의 30회의 평균값을 구하여 수치화 하였다.

실험에 사용된 각 함수는 다음과 같으며, 각 함수들은 전역 최적 해로의 수렴이 어려운 함수들 중에서 선택되었다. 각 함수는 10차원의 해, 즉 한 개체가 포함하는 문자열의 수가 10개$(x_1,\cdots,x_{10})$인 함수로 실험을 수행하였다.

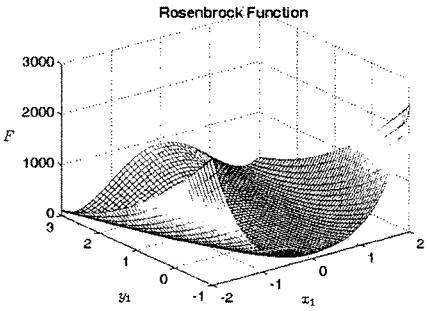
F1) Ackley의 함수(Ackley's test function)[16]



$$F=-20\exp\left(-0.2\sqrt{\frac{1}{2}\sum_{l=1}^{10}x_l^2}\right)-\exp\left(\frac{1}{2}\sum_{l=1}^{10}\cos 2\pi x_l\right)+20+e$$

$$x_l\in[-32.768,32.768],\ l=1,\cdots,10,\ F_{\min}=0$$

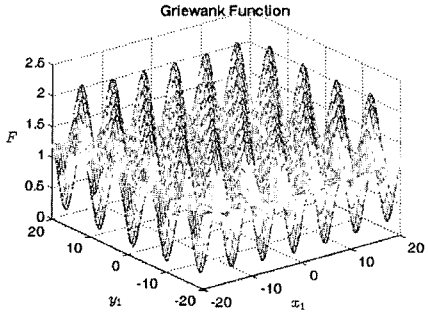
F2) Rosenbrock의 함수(Rosenbrock's function)[16].



$$F=\sum_{l=1}^{9}\left[100(x_{l+1}-x_l^2)^2+(x_l-1)^2\right]$$

$$x_l\in[-5,5],\ l=1,\cdots,10,\ F_{\min}=0$$

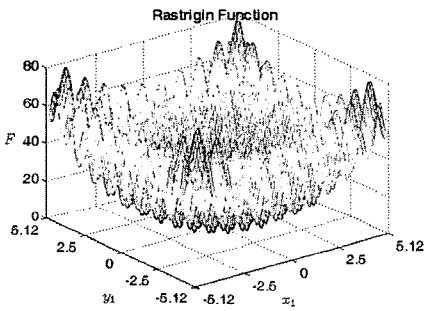
F3) Griewank의 함수(Griewank's function)[16]



$$F=-\prod_{l=1}^{10}\cos\left(\frac{x_l}{\sqrt{l}}\right)+\sum_{l=1}^{10}\frac{x_l^2}{4000}+1$$

$$x_l\in[-512,512],\ l=1,\cdots,10,\ F_{\min}=0$$

F4) Rastrigin의 함수(Rastrigin's function)[16]



$$F=\sum_{l=1}^{10}\left[x_l^2-10\cos(2\pi x_l)+10\right]$$

$$x_l\in[-5.12,5.12],\ l=1,\cdots,10,\ F_{\min}=0$$

그림 2~5에서는 ANT-GA와 EGA의 300세대까지의 해당 세대의 군집에서의 최적 해를 표시하였다. F1~F4까지의 4개의 함수 모두에서, 초기에는 ANT-GA의 탐색 시간이 오래 걸리는 경향을 보이지만, 일정 세대 이후가 되면 EGA보다 더 좋은 해를 탐색해 내는 것을 볼 수 있다.

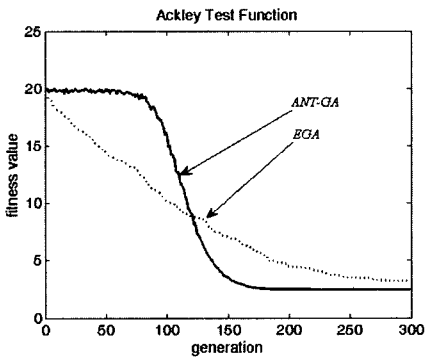


그림 2 F1에 대한 실험 결과(각 세대에서 발견된 최적 해)

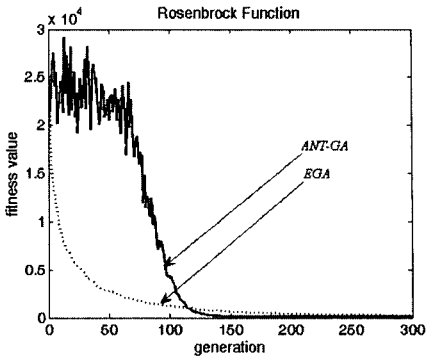


그림 3 F2에 대한 실험 결과(각 세대에서 발견된 최적 해)

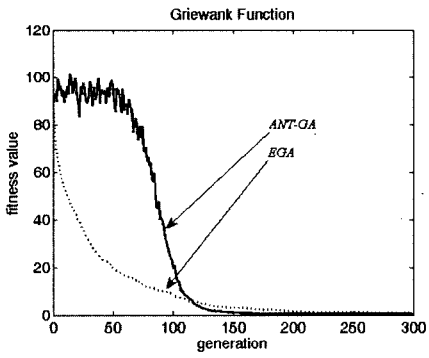


그림 4 F3에 대한 실험 결과(각 세대에서 발견된 최적 해)

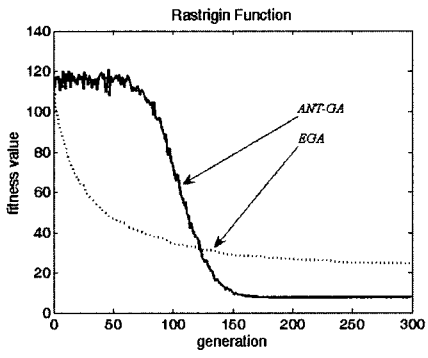


그림 5 F4에 대한 실험 결과(각 세대에서 발견된 최적 해)

그림 6~9에서는 F3과 F4의 함수를 가지고 ANT-GA를 실험하되, 식 (2)에서의 μ값을 0, 0.1, 0.3, 0.5, 0.7, 0.9의 차례로 변화시키면서 300세대까지의 해당 세대의 최적 해와 해당 군집의 해의 평균값을 구하였다. μ값이 작을 경우(0.1)에는 초기에 최적 해를 찾는 속도

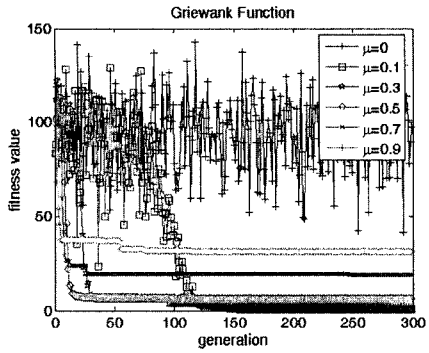


그림 6 F3에 대한 실험 결과(각 세대에서 발견된 최적 해)

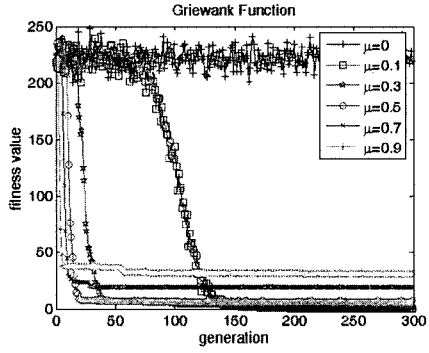


그림 7 F3에 대한 실험 결과(군집의 전체 해의 평균값)

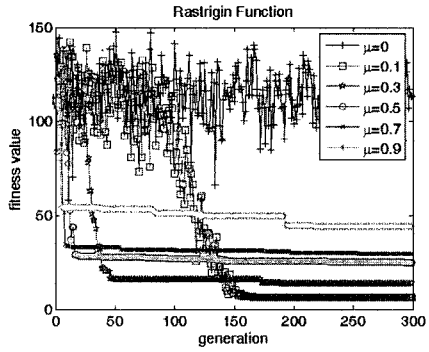


그림 8 F4에 대한 실험 결과(각 세대에서 발견된 최적 해)

가 느리지만 후기에 전역 최적 해로의 수렴이 잘 일어나는 반면, μ값이 클 경우(0.9)에는 초기에 빠른 속도로 해당 군집의 해들이 현재까지 발견된 해 중 최적 해로 수렴하지만, 해당 비트의 위치에 대한 θ의 값이 초기에 너무 커지거나 작아지기 때문에 후기에 지역 최소(local minima)에서 벗어나지 못하는 경향을 보인다. 이러한 점으로 볼 때 μ값은 현재의 최적 해로 얼마나 빨리 수렴(exploitation)할 것인지, 혹은 현재의 최적 해보다 더 나은 해를 탐색할 것인지(exploration)를 결정하는 중요

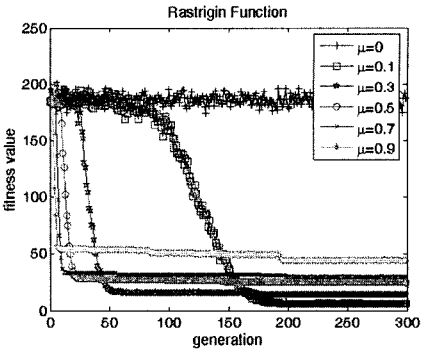


그림 9 F4에 대한 실험 결과(군집의 전체 해의 평균값)

한 매개 변수가 된다는 것을 알 수 있다.

즉, μ값이 클 경우에는, 이전 세대에서 갱신된 페로몬 값들은 조금만 고려되고 현 세대에서의 우성 해의 2진 문자열에 대한 페로몬 값이 크게 고려되어 군집 안의 모든 해들이 현 세대의 우성 해에 귀착되는 형태가 되며(그림 7, 9), 이전 세대의 우성 해에 대한 페로몬 값들은 무시한 채 현재 세대의 우성 해에 대한 값들에 큰 비중을 두어 고려하게 되므로, 결과적으로는 이전에 갱신된 0 또는 1에 대한 페로몬의 양과는 상관없이 현재의 우성 해로 수렴하는 형태가 된다. 이 값이 만약 전역 최적 해라면 큰 문제가 되지 않지만, 실험에서와 같이 부분 최적 해가 많은 복잡한 함수라면 문제는 달라진다. 현재의 우성 해가 어떤 부분 최적 해라고 할 경우에, 이 값에 한 군집 안의 모든 개체가 수렴한다면 지역 최소의 문제가 발생하게 된다.

그러므로 μ값이 너무 커지지 않도록 조절할 필요성이 있으며, 또한 μ값이 매우 작을 경우에는 이전 세대에서의 페로몬 값들의 영향이 더 많이 고려되며 이는 실제로 각 비트 위치의 0 또는 1의 값에 대한 페로몬 초기값인 0.5에서 시작하여 현재 세대까지 고려된 페로몬의 값들을 무시하지 않는다는 것을 뜻하기 때문에, 그림 6와 같이 어떤 최적 해에 대한 수렴 속도는 느려지지만 이전 세대에서 좋은 결과를 보였던 해들에 대한 페로몬 값도 함께 고려되므로 결과적으로 어떤 새로운 값을 탐색하는 데 있어서는 μ값이 클 경우보다 더 좋은 결과를 내게 된다. 실험 결과에서도 나타나듯이, 이러한 작은 μ값을 사용하면 일정한 세대가 지난 후에 EGA보다 더 좋은 최적 해를 찾아낼 수 있게 된다. 극단적인 경우에는($\mu=0$) 초기 값인 0.5를 계속적으로 사용하게 되므로, 그림 6, 7에서의 그래프와 같이 모든 해들이 각각의 비트 위치의 0 또는 1의 값에 대하여 0.5의 비율로 탐색만을 진행하는 꼴이 된다.

결과적으로 매개 변수인 μ의 값이 너무 커지거나 작

아지지 않도록 하여 수렴이나 탐색에 대한 정도를 조절할 필요성이 있다. 실험에서는 μ값이 0.1일 때 가장 좋은 결과를 보였다. 즉 0과 1사이의 값 중에서 작은 값일 때 좋은 결과를 보였으며, 이는 [7]의 Theorem 4.1.2에서 페로몬 증발 상수(evaporation factor)가 0에 가까운 작은 값을 가질 때 GACO가 수렴한다는 사실과 일맥상통한다.

"수렴"의 또다른 정의로써, 한 군집 안에 있는 모든 해가 얼마나 빨리 발견된 최적 해로 수렴하는가에 대한 문제는 유전자 알고리즘의 응용에 대한 연구[17,18]나 진화 연산 알고리즘(예: Linear Genetic Programming (LGP)[19])을 사용한 여러 응용 분야에서 중요한 문제가 되고 있다[20]. 이에 따른 실험으로, ANT-GA와 EGA의 해당 군집에 대한 평균 해의 값을 나타내었다 (그림 10~13).

ANT-GA의 군집에 속해 있는 모든 해들은 빠른 속도로 최적 해에 수렴해가는 반면, EGA의 군집에 속해 있는 모든 해들은 최적 해로 수렴하는 경향이 초기에 약간 나타나다가 후기에는 수렴하지 못하는 경향을 보인다.

이러한 결과의 이유는 CGA나 EGA에서는 변이 확률이 고정되어 있기 때문에 전체 군집의 최적 해로의 수렴이 어려우나, ANT-GA는 우성 해를 가지고 개미의 페로몬 갱신의 방법을 통해 변이의 확률을 조정하여 최적 해로의 수렴을 촉진시킬 뿐 아니라, 동시에 전체 군집의 최적 해에 대한 수렴 또한 촉진시키기 때문이다.

한 군집 안에 있는 모든 해가 발견된 최적 해로 수렴하지 못하는 EGA의 성질은 EGA의 다음 세대의 군집에 포함시킬 elitist의 비율을 지정하는 것에 관련되어 있는 점에 착안하여, 그림 13, 15에서는 F4의 함수를 가지고 elitist의 비율을 변화시켜 가며 이러한 EGA의 성질을 실험하였다.

실험 결과와 같이 elitist의 비율을 높일 경우는 지역 최소의 문제가 두드러지게 나타나며, elitist의 비율을 낮출 경우는 지역 최소의 문제는 해결되나 전체 군집의 최적 해로의 수렴은 초기에 약간 나타나다가 후기에는 오히려 증가하는 경향을 보이는 것을 알 수 있다. 이는 elitist의 비율이 높아지면 그만큼 다음 세대의 군집에서 elitist들이 많이 나타나게 되고 선택이나 교배로 인하여 새로이 탐색된 해가 차지하는 비율이 적어지기 때문에, 이러한 elitist가 지역 최소일 경우 지역 최소에서 잘 벗어나지 못하기 때문이다.

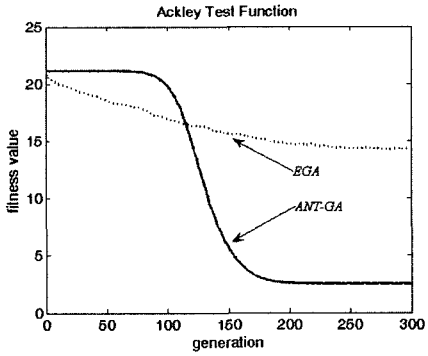


그림 10 F1에 대한 실험 결과(군집의 전체 해의 평균값)

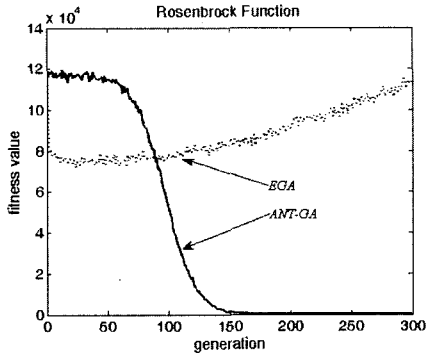


그림 11 F2에 대한 실험 결과(군집의 전체 해의 평균값)

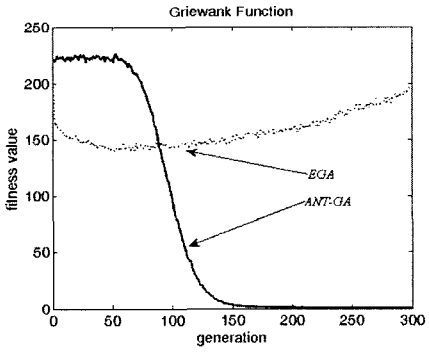


그림 12 F3에 대한 실험 결과(군집의 전체 해의 평균값)

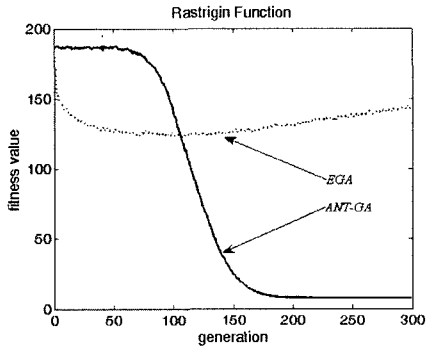


그림 13 F4에 대한 실험 결과(군집의 전체 해의 평균값)

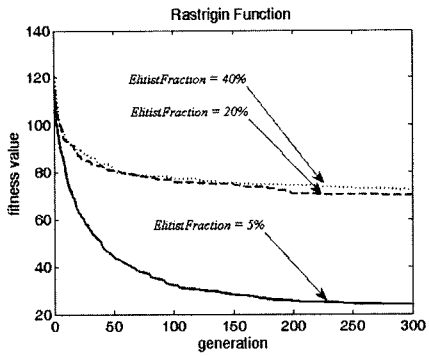


그림 14 EGA에 대한 실험 결과(각 세대에서 발견된 최적 해)

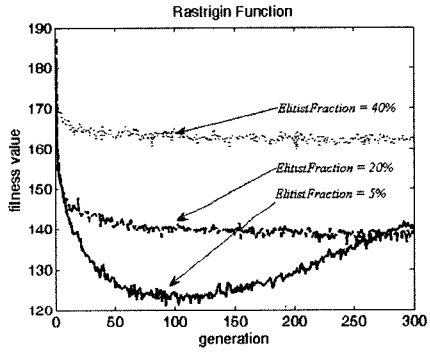


그림 15 EGA에 대한 실험 결과(군집의 전체 해의 평균값)

또한, 비율이 5%일 경우에는 후기에 elitist를 제외한 나머지 해들은 계속해서 탐색이 일어나기 때문에, 전체의 해들이 최적 해에 수렴하여 머물기보다는 다른 해를 계속해서 찾아감으로 인하여 그래프가 후반기에 증가하는 경향을 보이는 것으로 해석된다.

4. 결 론

우리는 군지능의 아이디어를 CGA의 변이 과정에 적용한 새로운 적응형 유전 알고리즘을 개발하고, 본 알고리즘이 최적 해에 수렴한다는 것을 확률적으로 증명하였다. 여러 흥미로운 문제에 대하여, 개미 집단 최적화가 CGA보다 좋은 결과를 보이는 것은 여러 논문에서 다루어져 왔다[6,21]. CGA의 구조에 개미 군지능 최적화의 방법론을 융합함으로써, 그 결과인 본 논문의 hybrid 알고리즘은 CGA의 약점을 보완할 수 있게 되었다.

본 논문에서의 여러 가지 전체 최적화 문제에 대하여, 새롭게 제안된 우리의 알고리즘이 EGA에 비해 최적 해로 빠르게 수렴 하는 것을 실험을 통하여 확인하였으며, EGA와는 달리 한 군집 안의 모든 해들이 최적 해로 수렴하는 것을 확인하였다.

또한 알고리즘의 파라미터인 μ값을 변경함으로써 최적 해로의 수렴 속도와 최적 해의 탐색 가능성을 조절할 수 있는 유연성 있는 알고리즘임을 확인하였다.

참 고 문 헌

[1] G. Rudolph, "Convergence Analysis of Canonical Genetic Algorithms," *IEEE Trans. on Neural Networks*, vol. 5, no. 1, 96-101, 1994.

[2] R. Eberhart and J. Kennedy, *Swarm Intelligence*, Morgan Kaufmann, 2001.

[3] C. R. Reeves, "Genetic Algorithms for the Operations Reserchers," *INFORMS J. on Computing*, vol.9, no. 3, 231-250, 1997.

[4] H. Mühlenbein, G. Paab, "From Recombination of Genes to the Estimation of Distributions I. Binary parameters," *Parallel Problem Solving from Nature*, H. Voigt, W. Ebeling, I. Rechenberg, and H. Schwefel (eds.), LNCS, vol. 1141, 178-187, 1996.

[5] S. K. Shakya, "Probabilistic Model Building Genetic Algorithm(PMBGA): A survey," Tech. Rep., School of Computing, The Robert Gordon University, UK, 2003.

[6] M. Dorigo, V. Maniezzo, A. Colorni, "The Ant System: Optimization by a Colony of Cooperating Agents," *IEEE Trans. Systems Man Cybernet.*, vol. 25, 29-41, 1996.

[7] W. J. Gutjahr, "A Graph-based Ant System and Its Convergence," *Future Generation Computer Systems*, vol. 16, no. 8, 2000.

[8] M. Srinivas and L. M. Patnaik, "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithm," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 24, no. 4, 656-667, 1994.

[9] J. Yang, J. Horng, and C. Kao, "A genetic algorithm with adaptive mutations and family competition for training neural networks," *Int. J. of Neural Systems*, vol. 10, no. 5, 333-352, 2000.

[10] D. Thierens, "Adaptive mutation rate control schemes in genetic algorithms," in *Proc. of the 2002 IEEE World Congress on Computational Intelligence*, 980-985, 2002.

[11] S. Baluja, "Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning," Tech. Rep. No. CMUCS94163, Department of Computer Science, Carnegie Mellon University, 1994.

[12] P. Larranaga, R. Etxeberria, J.A. Lozano and J.M Pena, "Combinational optimization by learning and simulation of Bayesian networks," in *Proc. of the Conference in Uncertainty in Artificial Intelligence*, 343-352, 2000.

[13] D. Bhandari, C. A. Murthy, and S. K. Pal, "Genetic Algorithms with Elitist Model and Its Convergence," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 10, 731-747, 1996.
[14] K. Najim, A. S. Poznyak, and E. Ikonen, "Optimization based on a team of automata with binary outputs," *Automatica*, vol. 40, 1349-1359, 2004.
[15] K. De Jong and W. Spears, "An Analysis of the Interacting Roles of Population Size and Crossover in Genetic Algorithms," in *Proc. First Workshop Parallel Problem Solving from Nature*, Springer-Verlag, 38-47, 1990.
[16] X. Yao, Y. Liu, and G. Lin, "Evolutionary Programming Made Faster," *IEEE Trans. on Evloutionary Computation*, vol 5, no. 2, 82-102, 1999.
[17] C. Fernandes, R. Tavares, C. Munteanu, and A. Rosa, "Using Assortative Mating in Genetic Algorithms for Vector Quantization Problems," in *Proc. of the 2001 ACM symposium on Applied computing*, 361-365, 2001.
[18] F. Villegas, T. Cwik, Y. Rahmat-Samii, and M. Manteghi, "A Parallel Electromagnetic Genetic-Algorithm Optimization (EGO) Application for Patch Antenna Design," *IEEE Trans. on Antennas and Propagation*, vol. 52, no. 9, 2424-2435, 2004.
[19] M. Brameier and W. Banzhaf, "A Comparison of Linear Genetic Programming and Neural Networks in Medical Data Mining," *IEEE Trans. on Evolutionary Computation*, vol. 5, 17-26, 2001.
[20] J. Fernandez and A. Caballero, "A Comparison of Management Strategies for Conservation with regard to Population Fitness," *Conservation Genetics 2*, 121-131, Kluwer Academic Publishers, 2001.
[21] M. Dorigo, G. Di Caro, and T. Stützle, Special Issue on "Ant Algorithms," *Future Generation Computer Systems*, vol. 16, no. 8, 2000.

김 중 항
2004년 서강대 컴퓨터학과 졸업. 2004년~현재 서강대학교 컴퓨터학과 석사과정 재학. 관심분야는 최적화 알고리즘, 센서 네트워크, P2P 네트워크 등

이 세 영
2004년 서강대 컴퓨터학과 졸업. 2004년~현재 서강대학교 컴퓨터학과 석사과정 재학. 관심분야는 최적화 알고리즘, 무선 네트워크 등

장 형 수
1994년 미국 Purdue University, 전기 및 컴퓨터공학과 졸업, 동 대학 석사, 박사 (1996, 2001). 2001년 9월~2002년 6월 Institute for Systems Research, University of Maryland, College Park, Postdoc, 2002년 6월~2003년 2월 고려대학교 정보통신 기술 공동연구소 연구교수, 2003년 3월~현재 서강대학교 공과대학 컴퓨터학과 조교수