

## 다중 라그랑지안 승수를 이용한 제한 진화 최적화

명 현

한국전자통신연구원 컴퓨터·소프트웨어연구소 휴먼컴퓨팅연구부  
대전광역시 유성구 가정동 161번지  
E-mail: myung@etri.re.kr

## Constrained Evolutionary Optimization Using Multiple Lagrange Multipliers

Hyun Myung

Human Computing Dept., CSTL, ETRI  
161 Kajong-dong Yusong-gu Taejon  
E-mail : myung@etri.re.kr

### 요약 (Abstract)

진화 연산을 이용하여 최적화 문제를 푸는데 있어서 가장 잘 알려져 있는 문제 중의 하나는 미완속 수렴이다. 일반적인 제한 최적화 문제를 푸는 기법으로서 제안된 하이브리드 진화프로그래밍(EP), 이상 EP(TPEP), Evolian 등과 같은 알고리즘도 첫 번째 상에서 이와 같은 문제점을 내포하고 있다. 본 논문에서는 이같은 문제점을 극복하기 위해서 Evolian 알고리즘에 공유 함수 기법을 적용하고 다음 상들을 위해서는 다중 라그랑지안 승수를 사용하고자 한다. 부개체군 영역에서 각각의 라그랑지안 승수들을 설정하고 병렬적으로 갱신해 나가면서 전역적인 최적해를 병렬적으로 찾아나간다. 컴퓨터 모의 실험을 통해서 제안된 공유 기법 및 다중 라그랑지안 승수 기법의 유용성을 보인다.

### I. 서론

본 논문은 다음과 같이 연속적인 변수들로 정의된 일반적인 제한 최적화 문제에 대해서 다루고자 한다: 다음의 제한 조건을 만족하고  $f(\vec{x})$ 를 최소화 하라.

$$\begin{aligned} g_1(\vec{x}) \leq 0, \dots, g_r(\vec{x}) \leq 0, \\ h_1(\vec{x}) = 0, \dots, h_m(\vec{x}) = 0 \end{aligned} \quad (1)$$

여기서  $f$  와  $g_k$ 는  $R^n$  함수이고,  $h_j$ 는  $m \leq n$ 인  $R^n$  함수이며  $\vec{x} = [x_1, \dots, x_n]^T \in R^n$ 이다.

일반적인 최적화 문제를 풀기 위해 진화 연산을 적용할 때의 문제점 중 가장 잘 알려진 특성은 유전적 표류(genetic drift)이다 [1]. 국부점들이 같은 최적값을 가지는 멀티 모달 함수에서 단순한 진화 연산을 적용하면 단지 하나의 국부점으로만 수렴하게 되고 그 국부점은 유전 연산자와 관련한 통계적 변위에 의해서 랜덤하게 선택되어 진다. 최적점에 충분히 다가가기 전에 유전 인자가 고정되어 버리는 이러한 미완속 수렴의 문제로 인해 진화 연산 알고리즘들은 비판을 받기

도 했다 [2]. 같은 맥락에서 하이브리드 진화 프로그래밍(EP) [3], 이상 EP(TPEP) [4], Evolian 알고리즘 [5] 등에서도 가장 큰 문제점은 첫번째 상에서의 미완속 수렴이다.

이와 같은 문제들을 극복하기 위해서 Goldberg와 Richardson은 유전 알고리즘내에서 공유 함수 기법을 도입하였다 [2]. 이 방법은 다른 스트림을 가진 안정적인 부개체군이 형성되도록 도와주어서 서로 다른 국부점들을 병렬적으로 찾아나갈 수 있도록 한다.

다른 방법으로는 다목적함수 최적화 기법이 미완속 수렴을 극복하는데 도움을 줄 수 있다. 유전 알고리즘 분야에서는 다목적함수에 기반을 둔 알고리즘들이 개발되고 있다. 그 대표적인 예가 VEGA (Vector Evaluated GA) [6]인데, VEGA 시스템에서는 전체 개체를 동일한 크기의 부개체군으로 나누어지고 각각의 부개체군은 하나의 목적함수를 담당한다. 선택 과정 중에는 각각의 목적함수마다 독립적으로 개체들을 선택하게 되지만 교차 연산시에는 부개체군들 사이에서 연산이 행해진다. 어떠한 목적함수에서도 좋지 않은 개체들에 대해서는 수렴하는 경향을 줄여주는 휴리스틱도 개발되고 있다.

최근에는 Srinivas와 Deb [7]에 의해 NSGA (Non-dominated Sorting Genetic Algorithm)라는 새로운 기법이 제안되었다. 이 기법은 개체들을 몇 개의 계층으로 분류한다. 선택을 하기 전에 모든 개체들이 반우성(nondomination)에 기반하여 랭크가 매겨진다. 이 반우성 개체들은 하나의 카테고리로 분류되는데 개체수에 비례하는 의사 적합도(dummy fitness)를 가지도록 한다. 그 이유는 이 개체들이 동등한 재생산 확률을 가지도록 하기 위함이다. 개체들의 다양성을 유지하기 위해서 분류된 개체들은 의사 적합도를 가지고 공유하게 된다. 그리고 나서 이 개체들은 무시되고 다른 반우성 개체들이 고려되어진다. 모든 개체들이 분류될 때까지 이러한 과정을 반복한다.

최근에 Beasley, Bull, Martin은 순차적 물새 방법

(sequential niching)이라는 새로운 방법을 개발하였다 [8]. 이 기법은 멀티 모달 함수에 대해서 공유함수 기법이 가지고 있는 단점들 (예를 들면 공유 적합도를 계산의 복잡성과 최적점의 개수에 비례하게 커져야 하는 전체 개체수)을 극복하기 위해 제안되었다. 순차적 틈새 방법은 일단 최적점이 찾아지면 평가 함수 자체를 수정하여 그 부분에서 높은 목적 함수값을 갖도록 한다. 한번 찾아진 최적점은 다시 탐색할 필요가 없기 때문이다. 어떤 면에서는 이전의 시도에서 얻어진 지식을 이용하여 다음번의 시도를 한다고도 볼 수 있다.

그러나 Mahfoud [9]는 순차적 틈새 방법과 병렬적인 기법을 비교했을 때 병렬화, 수렴 속도, 개체의 다양성 면에서 병렬적인 기법이 순차적 기법보다 우수하다고 보고했다. 공유 함수 기법과 같은 병렬적 기법은 개체들 중에서 틈새를 동시에 형성하고 보존하며 유인점(attractor)을 빠져 나가서 원하는 해를 찾아가는 능력이 있다는 것이다.

결론적으로 Evolian의 첫번째 상에서 공유함수 기법이나 다목적 함수 기법이 적용된다면 성능이 개선될 것이다. Evolian의 첫번째 상에서는 라그랑지 승수가 0으로 세팅되므로 외부 페널티 함수 방법과 동일하다. 제한 조건이 있을 때는 Evolian의 다음 상들이 적용되어야 한다. 최적점들이 여러개 존재할 때는 라그랑지 승수가 여러개 필요하게 된다. 왜냐하면 하나의 최적점은 하나의 라그랑지 승수 조건을 만족하기 때문이다. 그렇기 때문에 연속되는 상들을 위해서 라그랑지 승수들은 국부 최적점이라고 판단되는 영역에서 초기화 되어야 한다.

본 논문에서는 Evolian에서 공유함수 사용의 유용성을 고찰하고 제한 최적화를 위한 다중 라그랑지 승수 방법을 제안한다.

## II. 공유함수

공유함수(sharing function)는 어떤 한 개체로부터  $d$ 라는 거리에 있는 이웃 개체로 인해 그 개체의 적합도가 감소하는 정도를 나타낸다. 공유함수  $sh$ 는 다음과 같은 특징을 가지는 거리의 함수이다.

- $0 \leq sh(d) \leq 1$ , for all distances  $d$
- $sh(0) = 1$
- $\lim_{d \rightarrow \infty} sh(d) = 0$ .

위의 조건을 만족하는 공유함수는 많이 있을 수 있다. [2]에서는 공유의 정도를 제어하는 매개변수  $\sigma_{share}$ 를 써서 다음과 같이 두 개체간의 거리에 대한 함수로서 공유함수를 나타낸다.

$$sh(d) = \begin{cases} 1 - (\frac{d}{\sigma_{share}})^\alpha, & \text{if } d < \sigma_{share} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

여기서  $\alpha$ 는 공유함수의 컨벡시티를 결정하는 상수이다. 개체의 적합도를 틈새 계수(niche count) 만큼 나누어줌으로써 공유가 일어난다.  $i$ 번째 개체인  $\vec{x}^i$ 는 공유를 거쳐 새로운 적합도인  $eval'(\vec{x}^i)$ 를 가지게 된다.

$$eval'(\vec{x}^i) = eval(\vec{x}^i)/m(\vec{x}^i)$$

여기서  $m(\vec{x}^i)$ 은  $\vec{x}^i$ 라는 개체의 틈새 계수를 나타낸다.

$$m(\vec{x}^i) = \sum_{j=1}^{2N_p} sh(d(\vec{x}^i, \vec{x}^j)) \quad (3)$$

여기서  $N_p$ 는 한 세대의 개체수이고 자신을 포함한 모든 개체에 대해 합을 한다. 결론적으로 만일  $\vec{x}^i$  개체가 자신의 틈새에 있으면 자신의 적합도는 감소하지 않고 ( $m(\vec{x}^i) = 1$ ), 그렇지 않으면 이웃한 개체의 개수와 근접 정도에 따라서 그만큼 적합도가 감소하게 된다. 이렇게 함으로써 공유함수 기법은 전체 개체 내에서 특별한 증(우성종)이 급작스럽게 늘어나는 것을 제한할 수 있다. 더불어 공유함수 기법은 개체의 다양성을 유지할 수 있도록 하고 미완속 수렴을 방지할 수 있다 [2].

진화프로그래밍 기법은 최소화 문제를 다루기 때문에 본 논문에서는 유전 알고리즘과는 달리 목적함수를 다음과 같이 구현하였다:

$$\begin{aligned} \Phi'(\vec{x}^i) &= \Phi(\vec{x}^i) + \eta(t)(m(\vec{x}^i) - 1), \\ \eta(t) &= r_s(\bar{\Phi} - \Phi(\vec{x}^i))/2N_o, \\ \bar{\Phi} &= \frac{1}{2N_p} \sum_{j=1}^{2N_p} \Phi(\vec{x}^j) \end{aligned} \quad (4)$$

여기서  $m(\vec{x}^i)$ 은 (3)에 의해 계산된  $\vec{x}^i$  개체의 틈새 계수이다. 새로이 도입된 적응 매개변수  $\eta(t)$ 는  $t$ 번째 세대에서의 개체들의 통계치에 의존하는데, 전체 개체수에 의해 정규화된 틈새 계수에 비례하여 목적함수를 증가시킨다. 스케일링 인수인  $r_s < 1.0$ 는 양의 상수이고  $\bar{\Phi}$ 는 현재 개체들의 평균 목적함수이며  $\vec{x}^i$ 은 개체내에서 가장 좋은 성능의 개체를 의미한다.  $\vec{x}^i$  개체가 혼자 자신의 틈새에 있을 때 틈새 계수는 1이 되고, (4) 식의 마지막 항은 사라지고 공유된 목적 함수는 원래의 목적함수와 동일하게 된다.

이렇게 공유된 목적함수는 일반적인 진화프로그래밍 기법 [3, 4, 5]의 5번째 스텝에서 확률적인 토너먼트 선택을 행할 때 다음과 같이 쓰여진다:

모든 개체에 대해서 정해진 회수만큼 짝을 지은 비교를 행한다. 각각의 해에 대해서 모든 개체 중에서 동일한 확률로  $N_c$ 번 만큼 랜덤하게 비교할 개체를 선택한다. 각 비교시에 공유된 목적함수를 비교하게 되는데 만일 자신의 값이 상대방의 값보다 작으면 자신의 승수(win count)를 하나 증가시킨다.

모든 틈새 계수를 계산하는 데에는 많은 계산량이 필요로 하게 되므로 공유 목적함수는 첫번째 상에서만 적용되도록 한다. 한 개체의 틈새 계수를 계산하려면 유클리디안 거리 계산과 공유함수 계산을 매 세대마다  $2N_p$ 번 행해야 하기 때문이다. 참고로 비교 회수  $N_c$ 는  $\min(2N_p - 1, 10)$ 로 세팅을 하여 전체 개체수인  $2N_p$ 보다 작으면서 최대 10번의 비교만을 행하도록 한다.

공유함수의 유용성을 고찰하기 위해서 [2]에 나와있는 두개의 함수에 대해서 모의 실험을 해보자.

문제 #1:

$$f_1(x) = -\sin^6(5.1\pi x + 0.5) \text{를 최적화 하라.}$$

문제 #2:

$$f_2(x) = f_1(x) \cdot e^{-4 \log 2 \frac{(x-0.0887)^2}{0.8^2}} \text{를 최적화 하라.}$$

개체수  $N_p = 30$ 으로 하고 최대 세대는 30으로 하였을 때 Evolian의 첫번째 상단을 이용한 결과가 그림 1에 보여진다. Evolian의 수행 과정은 지면 관계상 생략하며 [5]을 참고하기 바란다.

그림 1에서 볼 수 있듯이 공유함수 기법을 도입하지 않은 Evolian의 첫번째 상은 외부 페널티 함수 방법과 동일하게 다수개의 최적점을 찾아내지 못한다. 반면 공유함수를 도입하였을 때는 탐색 공간상에서 극부 최적점에 개체들을 분산시킬 수 있다. 또한 각각의 극부 점에서의 개체의 개수는 그 극부점의 목적함수에 반비례함이 [2]에서 보고되고 있다.

### III. 다중 라그랑지 승수

제한 조건이 있을 때에는 Evolian의 첫번째 상이 끝난 후에 다음 상들이 적용되어야 한다. 극부 최적점의 피크들이 여러개 있을 때에는 하나의 라그랑지 승수 세트만이 아니라 그 피크 수 만큼 라그랑지 승수 세트가 필요로 하게 된다. 그러므로 후속 상들에서는 라그랑지 승수가 예상되는 극부 최적 영역 마다 초기화 되도록 한다 (그림 2 참조). 이것을 위해서 Evolian은 현재 개의 개체 영역에서 다수개의 피크를 판별하는 루틴을 가지고 있다.

[피크 판별 알고리즘]

$$N_{peak} := 0;$$

$$Peak(1) := \bar{x}^1;$$

do  $i = 1, \dots, \mu + \lambda$

if  $\frac{|\Phi(\bar{x}^i) - \Phi(\bar{x}^1)|}{\max(10^{-3}, |\Phi(\bar{x}^1)|)} \leq \sigma_{tot}$  and  $\|\bar{x}^i - Peak(j)\| \geq \sigma_{share}$

for all  $j = 1, \dots, N_{peak}$ ,

$$N_{peak} := N_{peak} + 1;$$

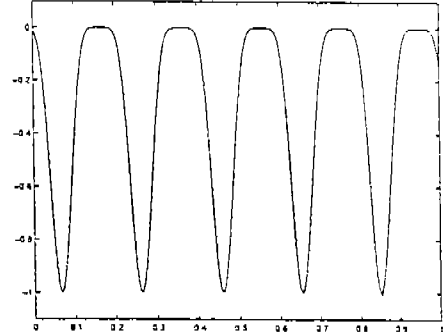
$$Peak(N_{peak}) := \bar{x}^i;$$

fi

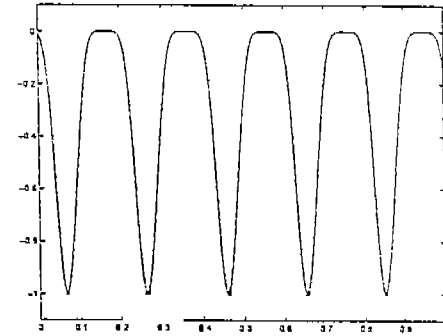
od

여기서  $N_{peak}$ 는 개체 공간에서의 피크의 개수를 나타내고  $Peak(j)$ 는  $j$ 번째 피크를 나타내는 변수이며  $\sigma_{tot}$ 은 목적함수 공간에서의 유사도를 측정하는 매개 변수이다. 우리는 전역 최소점에만 관심이 있기 때문에 전역 최소점이라고 판단되는 피크들만 피크 판별 알고리즘에 의해 판별된다. 판별하는 과정을 용이하게 하기 위해 개체들은 목적함수가 증가하는 순서대로 소트된다. 높은 순위의 개체들 중에서 가장 좋은 개체와 근사한 목적함수 값을 가지고 있고 이전에 판별된 개체와 거리가  $\sigma_{share}$  이상 떨어져 있는 개체가 피크로 판별된다.

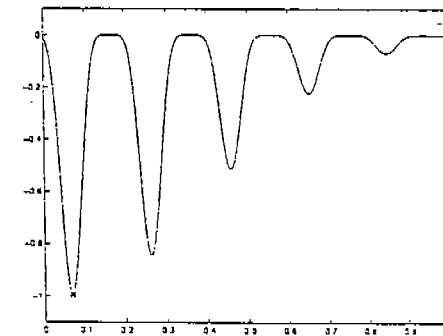
피크 판별 알고리즘은  $f_1$  함수에 대해서 피크 점들을 잘 판별하였다. 가장 좋은 피크값과 비슷한 값을 가지는 다수개의 피크들이 판별된 것이다. 함수  $f_1$ 에 대



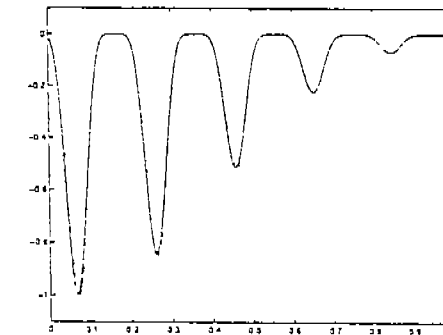
(a)  $f_1(\bar{x})$ : 공유없는 개체.



(b)  $f_1(\bar{x})$ : 공유한 개체.



(c)  $f_2(\bar{x})$ : 공유없는 개체.



(d)  $f_2(\bar{x})$ : 공유한 개체.

그림 1: 공유한 것과 아닌 것의 Evolian의 첫번째 상.

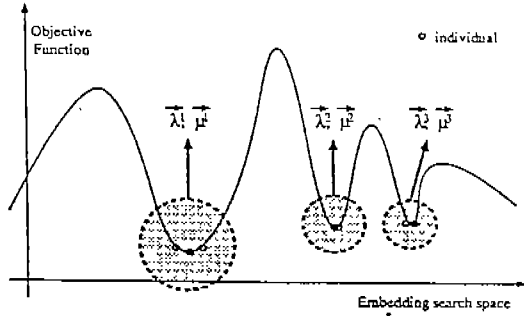


그림 2: Evolian에서 라그랑지 승수는 각 극부 최적 영역에서 병렬적으로 갱신된다.

Table 1: 함수  $f_3$ 를 최적화하기 위한 Evolian의 매개변수 설정

매개변수	값	의미
$N_p$	30	개체 수
$N_c$	10	비교하는 회수
$\rho$	$10^{-3}$	EP의 오차 허용치
$N_g$	7	EP의 세대 정지조건
$s_0$	1.0	초기 페널티 계수
$s_{max}$	$10^5$	최대 페널티 계수
$\gamma$	3.0	페널티 계수 증가율
$\epsilon_{\phi,1}$	$10^{-3}$	변수 정지조건
$\epsilon_{\phi,2}$	$10^{-5}$	목적함수 정지조건
$\sigma_{share}$	0.1	공유 매개변수
$r_s$	0.1	공유 스케일 인자
$\sigma_{tol}$	0.05	피크 유사도 매개변수

해서는 다수개의 피크가 판별되었지만 함수  $f_2$ 는 하나의 전역 최적점을 가지고 있기 때문에 하나의 피크만이 판별되었다. 이러한 극부적인 피크들이 판별되고 나서 라그랑지 승수들이 각 극부 피크에 할당되고 다음과 같은 갱신 법칙에 의해서 갱신된다:

$$\begin{aligned} \lambda_k[t+1] &= \lambda_k[t] + \epsilon s_k g_k^+(\bar{x}[t]) \\ \mu_j[t+1] &= \mu_j[t] + \epsilon s_j h_j(\bar{x}[t]) \end{aligned} \quad (5)$$

여기서  $\epsilon$ 은 작은 양수이다. 각각의 극부 영역들은 병렬적으로 Evolian의 후속 상들을 수행하여 정지 조건을 만족하면 멈추게 된다.

이제 다음과 같은 비선형 제한 최적화 문제를 고려해 보자 [10]:

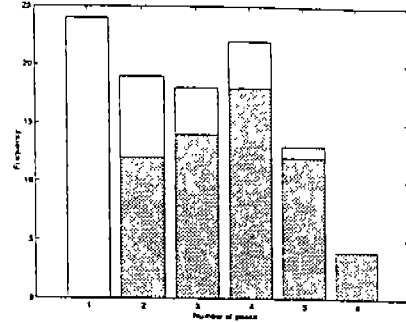
문제 #3:

제한조건  $h(x) = x_2 - x_1^2 = 0$ 을 만족하며 :

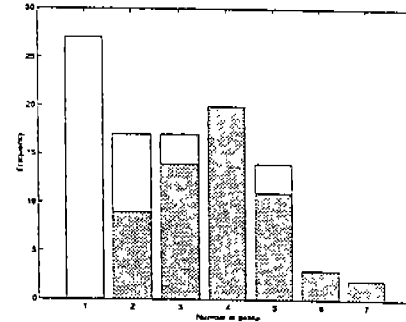
$f_3(x) = x_1^2 + (x_2 - 1)^2$ 를 최소화 하라.

이 문제는 두 개의 전역 최적점  $(x_1, x_2) = (\pm 1/\sqrt{2}, 1/2)$ 을 가진다. 표 1과 같이 매개변수를 설정하고 Evolian을 100번 수행하였을 때의 결과를 그림 3에 바 그래프로 나타내었다.

그림 3으로부터 공유를 하든지 안하든지 안정된 부개체군이 얻어지는 것은 약 60% 정도임을 알 수 있다. 첫번째 상에서 공유함수를 사용한 것은 사용하지 않은 경우와 비교했을 때 그다지 커다란 차이를 보이지 않음을 알 수 있다. Evolian에서 후속 상들을 적용했을 때 공유함수의 사용여부에 관계없이 다수개의 부개체군 영역을 형성하도록 하였다.



(a) 공유없이 100번 Evolian을 수행했을 때 피크의 개수와 빈도.



(b) 공유하여 100번 Evolian을 수행했을 때 피크의 개수와 빈도.

그림 3: 공유를 하거나 안했을 때 Evolian의 결과. 흰색 바는 하나의 최적점으로 수렴한 것이고 회색 바는 두개의 최적점으로 수렴한 것을 나타낸다.

바 그래프를 통해 보았을 때 첫번째 상에서 많은 피크가 결정될 수록 최적점으로 수렴할 가능성이 커진다고 말할 수 있다. 즉 다수개의 최적점들을 찾기 위해서는 다수개의 부개체군 영역을 찾는 것이 중요하다.

결론적으로 다수개의 부개체군 영역에서 다수개의 라그랑지 승수를 이용하면 병렬적으로 다수개의 최적점들을 효과적으로 찾을 수 있다고 말할 수 있다.

#### IV. 정리

본 논문에서는 Evolian 알고리즘의 첫번째 상 이후에 피크 판별 알고리즘에 의해 극부 최적 영역을 결정한다. 이렇게 해서 찾아진 부개체군 영역에 각각 라그랑지 승수 세트를 초기화 하면 알고리즘의 전역적인 최적성을 향상시킬 수 있다. 여기에 더하여 다중 라그랑지 승수 기법은 병렬적으로 수행될 수 있기 때문에 병렬 프로세서로 계산시에 계산 시간을 부개체군의 수만큼 줄일 수 있다. 제안된 다중 라그랑지 승수 기법은 최적점들이 다수개일 때 병렬적으로 이 최적점들을 찾을 수 있음이 보여졌다.

#### References

- [1] C. Hocaoglu and A. C. Sanderson, "Multi-modal function optimization using minimal representation size clustering and its appli-

- cation to planning multipaths," *Evolutionary Computation*, vol. 5, no. 1, pp. 81-104, 1997.
- [2] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proc. of the Second International Conference on Genetic Algorithms* (J. J. Grefenstette, ed.), (Hillsdale, NJ), pp. 41-49, Lawrence Erlbaum, 1987.
- [3] H. Myung and J.-H. Kim, "Hybrid evolutionary programming for heavily constrained problems," *BioSystems*, vol. 38, pp. 29-43, 1996.
- [4] J.-H. Kim and H. Myung, "Evolutionary programming techniques for constrained optimization problems," *IEEE Trans. on Evolutionary Computation*, vol. 1, pp. 129-140, July 1997.
- [5] H. Myung and J.-H. Kim, "Evolian: Evolutionary optimization based on Lagrangian with constraint scaling," in *Proc. of the Sixth Annual Conference on Evolutionary Programming / Lecture Notes on Computer Science (LNCS) 1213* (P. J. Angeline, R. G. Reynolds, J. R. McDonnell, and R. Eberhart, eds.), (Indianapolis, USA), pp. 177-188, Springer-Verlag, April 1997.
- [6] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. of the First International Conference on Genetic Algorithms* (J. J. Grefenstette, ed.), (Hillsdale, NJ), pp. 93-100, Lawrence Erlbaum, 1985.
- [7] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221-248, 1994.
- [8] J. E. Beasley, D. R. Bull, and R. R. Martin, "A sequential niche technique for multimodal function optimization," *Evolutionary Computation*, vol. 1, no. 2, pp. 101-125, 1993.
- [9] S. W. Mahfound, "A comparison of parallel and sequential niching methods," in *Proc. of the Sixth International Conference on Genetic Algorithms* (L. J. Eshelman, ed.), (Los Altos, CA), pp. 136-143, Morgan Kaufmann, 1995.
- [10] C. Y. Maa and M. A. Shanblatt, "A two-phase optimization neural network," *IEEE Trans. Neural Networks*, vol. 3, no. 6, pp. 1003-1009, 1992.