

# 확률적 타부 탐색 전략을 이용한 새로운 함수 최적화 방법에 관한 연구

論 文
50D-11-4

## A Study on a New Function Optimization Method Using Probabilistic Tabu Search Strategy

金亨洙\* · 琪 鉉\*\* · 朴俊灝\*\*\*  
(Hyung-Su Kim · Gi-Hyun Hwang · June Ho Park)

**Abstract** - In this paper, we propose a probabilistic tabu search strategy for function optimization. It is composed of two procedures, one is Basic search procedure that plays a role in local search, and the other is Restarting procedure that enables to diversify search region. In Basic search procedure, we use Belief space and Near region to create neighbors. Belief space is made of high-rank neighbors to effectively restrict searching space, so it can improve searching time and local or global searching capability. When a solution is converged in a local area, Restarting procedure works to search other regions. In this time, we use Probabilistic Tabu Strategy(PTS) to adjust parameters such as a reducing rate, initial searching region etc., which makes enhance the performance of searching ability in various problems. In order to show the usefulness of the proposed method, the PTS is applied to the minimization problems such as De Jong functions, Ackley function, and Griewank functions etc., the results are compared with those of GA or EP.

**Key Words** : Probabilistic Tabu Search, Belief Space, Function Optimization

### 1. 서 론

현실적으로 해결해야 하는 많은 최적화 문제들은 비선형적이며 많은 국부최소값들이 존재하기 때문에 일반적인 방법으로는 최적해를 찾기가 매우 어렵다. 지금까지 이러한 최적화 문제를 해결하기 위해서 여러 방법들이 제시되었는데, 최급경사법, Newton-Rapson법 등과 같은 수치적 해석법, 확률적 전역 탐색방법인 시뮬레이티드 어닐링(simulated annealing), 그리고 최근에는 자연의 진화과정을 모의한 확률적인 최적화 알고리즘인 진화연산 등이 적용되어 비교적 좋은 결과를 얻고 있다[1-5]. 그러나 이러한 방법들은 전역 탐색 및 빠른 계산시간을 동시에 만족시키는데 있어서는 많은 시행착오를 필요로 하며, 문제에 따라 그 성능의 편차가 심하여 일반적인 최적화 방법으로는 제한적인 성능을 나타낸다. 또한 탐색과정에서 얻어진 정보를 효율적으로 다루지 못함으로써 탐색 효율이 낮은 문제를 갖고 있다. 이에 반해 최근에 제시된 타부 탐색법은 경험적 탐색 방법으로서 일반적인 수치적 방법에 비해 국부 최소값에 빠질 위험이 적을 뿐만 아니라 유전 알고리즘이나 시뮬레이티드 어닐링과 같은 전역 최적해 탐색 방법에 비해 대상 문제에 관한 지식 및 탐색 중에 얻어진 정보를 효과적으로 이용할 수 있는 최적화 방법이다. 기존의

최적화 알고리즘은 대상 문제의 복잡도가 높고 탐색공간이 클 경우 최적해에 수렴하기 위해 많은 계산시간이 소요된다. 단점을 가지는데 반해, 타부 탐색법은 후보해의 생성에 직접 관여함으로써 복잡한 문제에 대해 효율적인 탐색으로 빠른 수렴속도를 구현할 수 있으며, 특히 다른 최적화 알고리즘과 결합하기 쉬운 특성 때문에 다양한 분야에서 활발히 연구되고 있다[6-11]. 그러나 지금까지 타부 탐색법의 적용 분야는 주로 이산적인 조합 최적화문제에 국한되고 있는데, 연속적 값을 갖는 문제에 적용하기 위해서는 이웃해 생성, 타부 정의 및 타부 리스트 등의 구현방법 등이 선행되어야 하지만 이러한 작업이 용이하지 않아 지금까지 적용된 경우가 매우 드문 실정이다.

본 논문에서는 연속적인 값을 갖는 함수 문제에 대해서 각 문제의 특성을 고려하면서 빠른 시간 내에 전역 최적해를 탐색할 수 있는 새로운 확률적 타부 탐색전략을 제안하였다. 제안한 방법은 크게 두 가지 과정으로 문제를 해결하는데, 이웃해로부터 빠르게 해에 수렴하도록 하는 타부 탐색과정과 국부수렴시 다시 탐색영역을 확장하는 재설정 과정으로 나눌 수 있다. 타부 탐색과정은 타부 탐색법의 이웃해 생성 및 탐색과정을 모의한 것으로 전체 탐색영역을 임의의 두 구간으로 구분한 후, 각 구간으로부터 일정비율로 이웃해를 생성한 후, 생성된 이웃해로부터 언덕등반(hill climbing) 방법에 의해 가장 우수한 해의 방향으로 탐색을 진행한다. 이산적 문제와는 다르게 연속적 문제에서는 탐색구간이 가변적이고 구간내에서 랜덤한 방법으로 이웃해를 생성함으로써 적절한 타부상태를 정의하기가 매우 어려우며 그 효과도 크지 않아 기본탐색과정에서는 타부 상태를 고려하지 않았다. 재설정 과

\* 正 會 員 : 釜山大 電氣工學科 博士課程  
\*\* 正 會 員 : 釜山大 大學院 電氣工學科 博士課程  
\*\*\* 正 會 員 : 釜山大 컴퓨터 및 情報通信研究所 基金教授  
接受日字 : 2001年 8月 29日  
最終完了 : 2001年 10月 16日

정은 타부 탐색과정에 의해 해가 어느 정도 수렴한 경우에 새로운 영역을 탐색할 수 있도록 탐색영역을 확장하는 과정으로, 제안한 확률 타부전략을 이용하여 함수의 수렴특성에 따라 재설정 이후의 탐색전략을 변경함으로써 전역적 탐색과 더불어 빠른 시간에 최적해로의 수렴이 가능토록 하였다.

제안한 방법을 서로 다른 수렴 특성 및 많은 국부 최적해를 갖는 다양한 평가 함수에 적용하여 그 유용성을 확인하였으며, 다른 방법들과 비교를 통해 성능을 검토하였다. 검토 결과, 제안한 방법이 다른 방법들에 비해 전역 최적해의 탐색 능력과 수렴 속도 면에서 크게 개선되었음을 알 수 있었다. 또한 제안한 방법은 문제에 따른 복잡한 파라미터를 선택할 필요가 없으므로 보다 일반화된 탐색 방법으로 사용이 가능하다.

## 2. 타부 탐색전략(Tabu Search Strategy)

함수 최적화 문제는 식 (1)에서 나타난 것처럼 주어진 범위에서 최적의 값을 구하는 문제로 정의할 수 있다. 여기서  $X$ 는 탐색공간  $R^n$ 에서 정의되며,  $R^n$ 은  $n$ 차원의 직사각형을 나타낸다.

$$\min f(x),$$

$$X = \{ (x_1, \dots, x_n) \in R^n \mid a_i \leq x_i \leq b_i, i=1, \dots, n \} \quad (1)$$

기존의 수치적 방법으로 이러한 문제를 해결할 경우 변수 개수나 제약조건이 증가함에 따라 최적값에 수렴하기 위해 많은 계산시간이 소요되거나, 또는 국부최적값에 수렴할 가능성이 커진다. 이러한 문제점을 개선하기 위해 많은 방법들이 제시되었으나, 여전히 제한적인 성능을 나타내고 있다. 최근에 제시된 타부 탐색법은 문제의 특성에 따른 경험적 지식을 쉽게 활용할 수 있으며, 탐색 중에 얻어진 정보를 효과적으로 이용할 수 있기 때문에 다양한 분야에 널리 적용되고 있다. 그러나 연속적인 값을 갖는 문제에 있어서는 적용이 어려워 아직까지 적용된 사례가 매우 드물며, 몇 가지 적용 사례에 있어서도 일반화가 충분히 이루어지지 않아 제한적인 성능을 나타내고 있다.

연속적 함수 최적화 문제에 적용된 사례를 보면, 참고문헌 [7]에서는 함수의 최적화 문제를 해결하기 위해 다단계 타부 탐색법을 제시하였다. 즉, 이웃해 및 타부 범위를 여러 단계로 설정하였으며, 각 단계는 독립적인 타부리스트를 가진다. 각 단계를 일정횟수만큼 반복한 후, 순차적으로 다음단계를 수행하였으며, 모든 단계를 수행한 후에는 다시 처음단계부터 반복하여 탐색을 수행하였다. 종료조건을 만족할 때까지 이러한 과정을 반복함으로써 보다 넓은 영역의 탐색과 더불어 수렴성을 도모하도록 하였다. 참고문헌[8]의 경우에는 반복횟수에 따라 탐색을 위한 이동거리를 점차 감소시킴으로써 일정값에 수렴하도록 하는 탐색방법을 사용하였다. 이때 타부리스트를 사용하지 않음으로서 탐색과정을 보다 단순화시켰으며, 몇 가지 사례에 대해 탐색속도와 수렴값이 개선됨을 보였다. 참고문헌[7-8]의 방법들은 연속적인 실수형 최적화문제에 있어서도 타부탐색법이 효과적으로 적용될 수 있음을 보여주었으나 적용한 문제의 복잡도가 비교적 낮고, 연속적인 탐색영역에 있어 타부 탐색법의 여러 가지 설정변수의 영향

이 충분히 검토되지 않아 다른 문제에 적용하기가 매우 어렵다. 따라서 보다 일반적이고 복잡한 문제를 효과적으로 해결하기 위해서는 연속적 탐색 영역의 특성을 잘 반영한 타부 탐색 전략이 요구된다. 본 논문에서는 탐색 속도 및 전역 탐색능력을 개선한 2단계의 타부 탐색 전략을 제안한다.

### 2.1 제안한 알고리즘 구성

본 논문에서 제안한 확률적 타부 탐색전략은 그림 1과 같이 전체적으로 국부탐색을 담당하는 타부 탐색 과정과 국부 수렴한 경우 탐색영역을 다시 확장하는 재설정(Restarting) 과정으로 구성되며, 재설정시 국부탐색 비율과 전역탐색 비율을 적용적으로 조절하기 위해 확률 타부 전략(Probabilistic Tabu Strategy : PTS)을 사용하였다. 먼저 국부탐색을 하는 타부 탐색 과정을 살펴보면, 전체 구간으로부터 신뢰구간[5]을 설정한 후 신뢰구간과 초기해 부근의 근접구간으로부터 이웃해를 생성하며 탐색을 진행한다. 여기서 신뢰구간(Belief Space)은 효율적인 탐색을 가능하도록 하기 위해서 도입된 것으로 수렴정도에 따라 이웃해 생성 구간을 제한하는 역할을 한다. 생성된 이웃해의 적합도를 평가하여 상위 일정비율의 값으로 후보해( $C_{sol}$ )를 구성하며, 후보해 중에서 가장 좋은 값을 다음 탐색의 현재해로 설정한 후 탐색을 반복하게 된다. 이때 근접구간과 신뢰구간의 크기는 점차 감소하게 되는데, 신뢰구간의 크기가 일정 크기이하로 줄어들거나 일정 반복동안 해의 변동이 없을 경우에 국부수렴한 것으로 간주하여 타부 탐색과정을 종료한다. 타부 탐색과정에 의해 국부 수렴한 경우 재설정 과정에 의해 다시 탐색영역을 확장하게 되며, 이때 확률 타부 전략에 의해 새로운 탐색 전략을 선택한다. 선택된 전략에 의해 탐색 파라미터값을 조절한 후 타부 탐색과정을 반복하게 되며, 이때 각 전략의 선택확률은 해의 개선정도에 따라 적용적으로 조절되는데, 어느 한 전략의 선택확률이 지나치게 크지 않도록 제한함으로써 한 전략에 의한 국부 수렴의 가능성을 방지하였다. 이러한 방법으로 구해진 해가 허용값 이내로 수렴하거나 정해진 반복횟수를 초과할 때까지 전체 탐색과정을 반복한다.

### 2.2 타부 탐색 과정

타부 탐색 과정은 기존의 타부 탐색법을 연속 함수의 최적화 문제에 적합하도록 변경한 것으로 이웃해 구성 방법과 타부 사용 여부에 큰 차이점이 있다. 먼저 이웃해를 구성하는 방법을 보면, 전체 탐색구간을 크게 두가지 영역, 즉 근접구간과 신뢰구간으로 구분한 후, 각각의 영역으로부터 일정비율의 이웃해를 생성하게 된다. 근접구간은 현재해를 중심으로 작은 범위의 구간을 나타내며, 신뢰구간은 이전 탐색정보에 의해 적용적으로 결정되는 탐색 영역을 나타낸다. 그림 2는 두 개의 독립변수( $x, y$ )를 가진 경우에 대해 나타낸 것으로, 그림에서 보듯이 현재해(CP)를 중심으로 한 근접구간(A)과 신뢰구간(B), 그리고 각 변수값이 서로 다른 구간에 위치하게 되는 혼합구간(AB, BA)으로 구분되는데, 이때 혼합구간은 현재해로부터 근접구간의 미세한 변화와 신뢰구간의 비교적 큰 변화를 동시에 가짐으로서 특정 변수값에 대한 탐색을 강화하는 역할을 한다.

Function TabuSearch(  $s^*$ ,  $N_{size}$ ,  $\alpha$  )  
 Set  $k = 0$   
 Initialize  $BCR^n$   
 Set  $s^0 = s^*$   
 Do  
     Identify  $N_{sol}(s^k) \in N_{size}$   
     Identify  $N_B(s^k) \in B$   
     Identify  $C_{sol}(s^k) \subset N_B \cup N_{sol}$   
      $f(s^k) = \text{Min}_{s \in C_{sol}} (f(s))$   
     If  $f(s^k) < f(s^*)$  then, set  $s^* = s^k$   
     Update  $B = \text{Range}(C_{sol})$   
     Update  $N_{size} = \alpha^k N_{size}$   
      $k = k + 1$   
 Repeat  $f(s^k) \neq f(s^{k-1})$  and  $B > \text{Threshold}$   
 return best solution  $s^*$   
 End

(a) 기본 탐색과정

Procedure ProbabilisticTabuStrategy  
 Set  $t = 0$   
 Initialize  $N_{size} \subset R^n$   
 Select initial  $s_{st}^0 \in R^s$ ,  $\alpha \in R^a$   
 Select initial  $s_{sol}^0 \in R^n$   
 Do  
      $s_{sol}^{t+1} = \text{BasicSearch}(s_{sol}^t, N_{size}, \alpha)$   
     If  $f(s_{sol}^{t+1})$  is better than  $f(s_{sol}^t)$  then,  
          $P(s_{st}^t)$  is increasing, others are decreasing  
          $P(a)$  is increasing, others are decreasing  
     Identify  $N_{st}(s_{st}^t) \subset R^s$ ,  $N_a(s_a^t) \subset R^a$   
     Select strategy  $s_{st}^{t+1} \in N_{st}$  in proportion to  $P(s_{st})$   
     Select reducing rate  $a \in N_a$  in proportion to  $P(a)$   
     Update  $N_{size}$  by  $s_{st}^{t+1}$   
      $t = t + 1$   
 Repeat  $f(s_{sol}^t) > \epsilon$  and  $k < \text{IterationCriterion}$   
 End

여기서,  $k$ : 반복횟수,  $s_{sol}^k$ :  $k$ 번째 반복시의 현재해  
 $R^n$ : 전체 탐색 영역,  $N_{sol}$ : 근접구간내의 이웃해 집합  
 $N_B$ : 신뢰구간내 이웃해 집합,  $C_{sol}$ : 후보해 집합  
 $N_{size}$ : 근접구간 크기,  $B$ : 신뢰구간 크기  
 $t$ : 재설정 횟수,  $s_{st}^t$ :  $t$ 번째 재설정시의 현재해  
 $P(\cdot)$ : 선택확률,  $s_{st}^t$ :  $t$ 번째의 재설정시의 현재 전략  
 $R^s$ : 모든 가능전략,  $N_{st}$ : 이웃 전략 집합

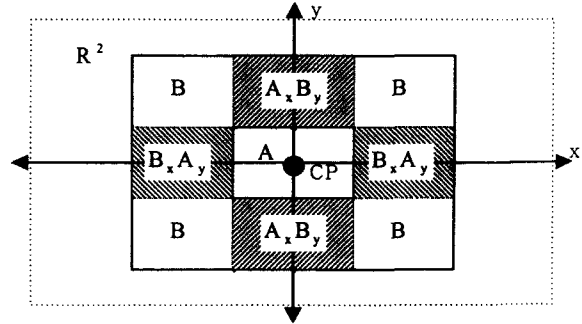
(b) 확률 타부 전략

그림 1 제안한 타부 탐색 전략의 알고리즘

Fig. 1 Algorithm of the proposed Tabu Search Strategy

$N$ 개의 독립변수를 갖는 문제의 경우 이웃해를 구성하는 방법은 그림 3과 같다. 그림 3에서 보는 바와같이 모든 변수 영역을 근접구간(A)과 신뢰구간(B) 및 혼합구간(AB)으로 나누어 각각의 구간으로부터 정해진 비율만큼의 이웃해를 생성한다. 이렇게 생성된 이웃해를 평가하여 일정비율만큼의 상

위값들로 후보해를 구성한 후, 후보해의 분포범위에 의해 신뢰구간을 결정하게 된다. 신뢰구간은 전체 탐색영역으로부터 탐색영역을 제한하여 효율적인 탐색을 가능하게 하는 것으로, 대체로 이웃해로부터 후보해의 설정비율에 비례하여 크기가 감소한다.



여기서,  $R^2$ : 전체구간,

그림 2 이웃해 생성을 위한 구간 구분

Fig. 2 Region partitioning for generation of neighborhoods

변수

$x_1$	$x_2$	$x_3$	$x_4$	$\dots$	$x_N$
AB	AB	AB	AB	$\dots$	A
AB	AB	AB	AB	$\dots$	AB
B	B	B	B	$\dots$	B

그림 3 N개의 변수에 대한 이웃해 구성 방법

Fig. 3 Composing method of neighborhoods in N-variable case

타부 탐색 과정이 일반적인 타부 탐색법과 또 다른 특징은 타부 리스트가 포함되어 있지 않다는 것인데, 이것은 몇 가지 이유 때문에 타부를 정의하는 것이 비효율적이기 때문이다. 우선 이웃해 생성을 보면, 각 구간내에서 랜덤하게 이루어지므로 같은 지점이 반복되어 타부조건을 만족할 가능성이 매우 적다. 따라서 대부분의 경우 탐색지점에 대해 임의크기의 구간을 타부로 설정하게 되지만 너무 크게 설정한 경우에는 수렴이 되지 않으며, 너무 작은 경우에는 타부조건을 만족하기가 힘들다. 따라서 적절한 크기의 타부범위를 정의해야 하지만 일정한 타부범위를 사용할 경우 함수 최적화 문제와 같이 수렴하는 문제에 있어서는 탐색이 진행됨에 따라 탐색구간이 감소하므로 점점 타부 영향을 많이 받게 되어 수렴속도가 떨어지게 된다. 타부 범위를 가변하는 방법도 있으나 가변 범위를 정의하기가 힘들며, 문제의 수렴 특성에 따라 효율이 달라지므로 효과적인 방법이 되지 못한다. 이러한 이유로 기본 탐색 과정에서 타부조건 및 타부리스트를 제외하였다.

### 2.1.1 근접구간(Near Range)

현재해를 중심으로 작은 크기의 구간을 설정한다. 구간의 크기는 매 반복횟수마다  $\alpha$ 의 비율로 감소시켜 점차적으로 수렴하도록 하였다. 식 (2)는 근접구간내에서의 이웃해 집합을 나타낸다.

$$N_{so}(x^k) = \{x \mid |x - x^k| \leq Aa^k, x \in R^n\} \quad (2)$$

여기서,  $Aa^k$  : k 반복횟수일 경우의 근접구간 크기  
 $x^k$  : k 반복횟수에서의 현재해  
 $a$  : 근접구간 크기의 감소율,  $0 < a < 1$   
 $A$  : 근접구간의 초기치

2. 2. 2 신뢰구간(Belief Space)

문제를 풀면서 생기는 정보를 효율적으로 이용한다면 탐색능력을 더욱 향상시킬 수 있을 것이다. 문화 알고리즘(culture algorithm)[5]에서 사용된 신뢰구간의 개념은 주어진 문제의 제약조건이나 얻어진 정보를 이용하여 탐색 영역을 제한하는 것으로 탐색구간이 큰 경우 탐색구간을 효과적으로 축소함으로써 탐색성능을 높일 수 있도록 한 것이다. 본 논문에서는 신뢰구간을 결정하기 위해 우선 탐색과정 중에 생성된 이웃해의 상위집단에 의해 후보해를 구성하였으며, 구성된 후보해의 전체 분포 범위를 이용하여 다음 탐색시의 신뢰구간 범위로 설정하였다. 이것은 이웃해 중에서도 비교적 우수한 해의 근처로부터 전역 최적값이 발견될 가능성이 크다는 가정을 기초로 하였다. 신뢰구간에 의해 전체 탐색영역으로부터 탐색구간을 제한함으로써 탐색성능 및 수렴 속도를 크게 개선할 수 있다. 식 (3)은 신뢰구간내에서 생성된 이웃해 집합을 나타낸 것이다.

$$N_B(x^k) = \{x \mid l \leq x \leq u, x \in R^n\} \quad (3)$$

$l = \min_{x \in B} x, u = \max_{x \in B} x$   
 여기서, B : 신뢰구간  
 l : 신뢰구간의 하한값  
 u : 신뢰구간의 상한값

2. 2. 3 혼합구간(Mixed Region)

N개의 독립변수를 가진 문제를 최적화하려고 할 경우, 모든 독립변수에 대해 제각기 동일구간의 조합으로만 이웃해를 생성한다면 특정 변수의 변동성분을 고려하기가 어렵게 된다. 따라서 본 논문에서는 그림 3에서 보듯이 확률적으로 선택한 임의값수의 변수에 대해서는 비교적 변동이 큰 신뢰구간 내에서, 나머지는 근접구간 내에서 발생시키는 방법을 이용하여 근접구간과 신뢰구간의 혼합된 구간을 구현하였으며, 이 구간내에서 적절한 비율만큼의 이웃해를 생성함으로써 보다 효율적인 이웃해를 구성하도록 하였다.

2. 3 확률 타부 전략(PTS)에 의한 재설정(Restarting) 방법

타부 탐색과정에 의한 탐색 방법은 탐색영역이 수렴하여 국부값에 도달한 경우 더 이상 새로운 값을 찾기가 매우 힘들게 된다. 이러한 경우 탐색영역을 다시 재설정(Restarting)하는 방법을 사용하여 탐색영역을 확장할 수 있다. 본 논문에서는 확률 타부 전략에 의해 재설정시 초기값과 감소율 등을 결정하도록 하였다. 그림 4는 이러한 방법에 의한 탐색과정을 도식적으로 나타낸 것이다. 초기해로부터 탐색 영역 A내에서 근접구간과 신뢰구간에 의해 기본 탐색 과정을 수행하다가 국부수렴한 것으로 판단될 경우, 확률 타부 전략에 의한 재설정 과정이 수행된다. 재설정 이후 탐색지점은 탐색영역 B로 이동하여 다시 타

부 탐색과정이 수행된다. 이러한 과정을 전역최적해를 찾을 때까지 또는 일정횟수동안 반복하게 되는데, 이것은 타부 탐색 과정의 빠른 수렴특성을 이용하면서 전역탐색이 가능하도록 한다.

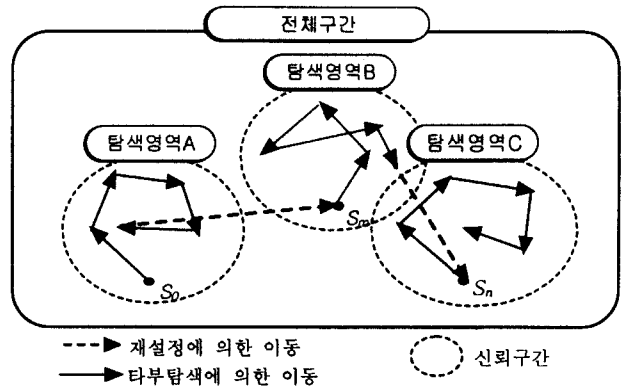


그림 4 재설정에 의한 탐색 영역의 다양화  
 Fig. 4 Diversification of search region by Restarting

2. 3. 1 재설정(Restarting)

타부 탐색과정에 의한 탐색은 탐색이 진행됨에 따라 근접구간의 크기 및 신뢰구간의 크기를 점차 감소시킨다. 이때 신뢰구간의 크기가 매우 작아질 경우에는 더 이상 새로운 영역의 탐색이 어려워지며 국부 수렴하게 되는데, 이러한 경우 신뢰구간과 근접구간을 동시에 확장시키는 재설정 방법을 사용함으로써 새로운 탐색 영역을 탐색할 수 있도록 하였다. 재설정 과정에 의해 신뢰구간의 크기는 전체구간으로 확장되며, 근접구간의 크기는 식 (4)와 같이 초기화된다.

$$A^t = A_0\beta^t \quad (4)$$

여기서, t : 재설정 횟수  
 $A_0$  : 최초의 근접구간 초기치  
 $A^t$  : t번째 재설정시의 근접구간 초기치  
 $\beta$  : 근접구간 초기치의 감소율,  $0 < \beta \leq 1$

2. 3. 2 이웃해 구성 및 타부 상태 정의

일정한 비율을 갖는 재설정 과정은 문제 유형에 따라 수렴특성을 적절하게 고려하지 못해 탐색성능이 떨어질 수 있다. 본 논문에서는 재설정 이후의 파라미터를 조절하기 위해 확률 타부 전략(PTS)을 사용하였다. 타부 전략을 구현하기 위해서는 우선 문제의 특성을 잘 반영한 이웃해 구성이 필수적이다. 본 논문에서는 재설정 이후의 탐색전략을 선택하기 위해 식 (5)와 같이 두 종류의 이웃해를 구성한다.  $N_{st}$ 에 의한 집합은 재설정이후에 근접구간 초기값을 변경하기 위한 전략을 나타내며,  $N_{st}$ 는 근접구간의 감소율 집합을 나타낸다. 모든 이웃해들은 선택확률( $P_x$ )을 가지고 있으며, 각 전략 및 감소율은 선택확률에 비례하여 확률적으로 선택된다. 그리고 각각의 선택확률은 식 (6)과 같은 방법으로 현재해의 개선여부에 따라 수정된다. 즉 선택된 전략에 의해 해가 개선된 경우와 개선되지 않은 경우에 대해 표 1과 같은 비율로서 각각의 선택확률을 조절하여 해의 개선정도에 따라 적절한 선택확률을 가지게 하였다. 그림 5에서 보듯이 A 전략이 선택되어 해가 개선된

경우, 표 1의 비율에 의해 A 전략의 선택확률은 20%만큼 증가하고 다른 전략의 선택확률은 10%만큼씩 감소한다. 해를 개선하지 못한 경우에는 이와반대로 선택되었던 전략의 확률을 감소시키며 다른 전략의 선택확률은 다소 증가시키게 되는데, 이때 변화율은 개선된 경우보다 낮다. 이것은 해를 개선한 경우에 보다 큰 가중치를 부여함으로써 보다 빠르게 수렴할 수 있도록 한 것이다.

$$N_s(x) = \{x | x \in R^s\}, R^s = \{A, B, C\} \tag{5}$$

$$N_a(x) = \{x | x \in R^a\}, R^a = \{\alpha^{high}, \alpha^{mid}, \alpha^{low}\}$$

여기서, A :  $N_{upperlimit} = N_{upperlimit} \times \frac{1}{\beta}$ ,  $A' = N_{upperlimit}$   
 B :  $A' = A'^{-1}$   
 C :  $N_{lowerlimit} = N_{lowerlimit} \times \beta$ ,  $A' = N_{lowerlimit}$   
 $\alpha, \beta$  : 근접구간 및 근접구간 초기치 감소율,  $0 < \alpha, \beta < 1$   
 $A^t$  : t 번째 재설정시의 근접구간 초기치

$$P'_x = P_x^{t-1}(1 + \Delta P'_x) \tag{6}$$

$$P_A + P_B + P_C = 1, P_{\alpha^{high}} + P_{\alpha^{mid}} + P_{\alpha^{low}} = 1$$

여기서,  $P'_x$  : t번째 재설정시에 x의 선택확률  
 $\Delta P'_x$  : t 번째 재설정시의 선택확률 변화율

표 1 해의 개선여부에 따른 각 전략의 선택확률 변경  
 Table 1 Changing probabilities of selection

변화율 전략	해가 개선된 경우			해가 개선되지 않은 경우		
	$\Delta P_A$	$\Delta P_B$	$\Delta P_C$	$\Delta P_A$	$\Delta P_B$	$\Delta P_C$
A	+20	-10	-10	-10	+5	+5
B	-10	+20	-10	+5	-10	+5
C	-10	-10	+20	+5	+5	-10

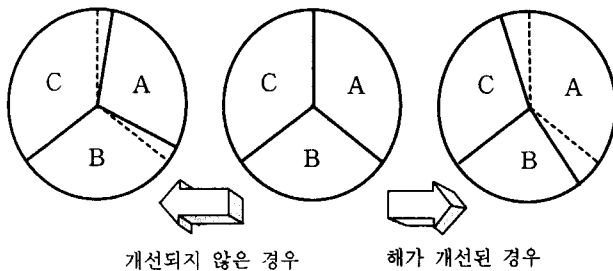


그림 5 A 전략이 선택된 경우 각 전략의 선택확률 변경  
 Fig 5 Changing probabilities of selection when A strategy is selected

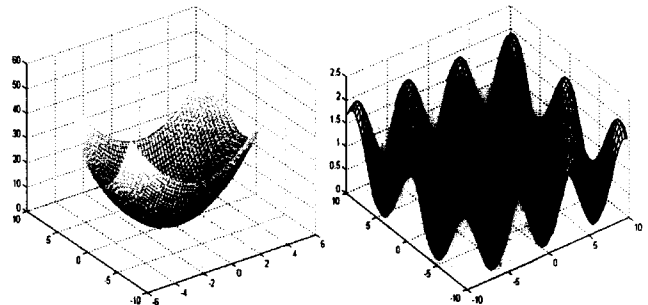
### 2. 3. 3 타부 리스트

일반적으로 타부 탐색법은 타부 리스트를 이용하여 이전의 탐색한 지점으로의 이동을 금지(Tabu)하게 된다. 본 논문에서 두 번째로 구현된 확률 타부 전략은 모든 이웃해가 선택 확률을 갖도록 구성함으로써 타부 리스트를 대체하였다. 즉, 해의 개선 여부에 따라 적응적으로 선택확률을 변경하였고, 모든 이웃해는 선택확률에 의해 선택 및 금지가 가능하도록 함으로써 타부 리스트와 같은 역할을 하도록 하였다.

### 3. 평가 함수

본 논문에서는 제안한 최적화 기법의 성능을 검토하기 위

해 다양한 평가 함수들을 이용하였다. 사용된 평가 함수[4-5, 10]는 크게 단봉(unimodal) 함수와 다봉(multimodal) 함수로 구분할 수 있는데, 단봉 함수의 경우는 최적값이 하나이므로 최적해를 구하기가 용이한 반면, 다봉 함수의 경우에는 많은 국부 최적값을 가지고 있어 일반적인 수치적 해석 방법에 의해서 전역해를 찾기가 매우 어렵다. 그림 6의 (a), (b)는 각각 대표적 단봉 함수인 De Jong 함수와 다봉 함수인 Griewangk 함수를 3차원 영역에서 나타내었다.



(a) De Jong 함수(Sphere 함수) (b) Griewangk 함수

그림 6 대표적 함수의 2-변수에 대한 목적함수값  
 Fig. 6 2-dimensional landscape of the test functions

본 논문에서 제안한 방법에 대한 성능평가 함수와 각 변수의 범위를 식 (7)~(20)에 나타내었으며, 탐색 성능의 비교에 주안점을 두기 위해 다른 제약조건은 없는 것으로 가정하였다.  $f_1 \sim f_9$ 은 비교적 낮은 차원 또는 적은 수의 국부 최소해를 갖는 함수들이며,  $f_{10} \sim f_{14}$ 는 변수 수에 따른 국부 최소값의 수가 지수적으로 증가하는 매우 어려운 문제들로 알려져 있다. 모든 경우 전역 최적해 지점에서 목적함수의 값은 0이다.

#### ① De Jong 함수

$$f_1(x) = \sum_{i=1}^{30} x_i^2, -5.12 \leq x_i \leq 5.12 \tag{7}$$

$$f_2(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2, -2.048 \leq x_1 \leq 2.048 \tag{8}$$

$$f_3(x_1, \dots, x_{30}) = \sum_{i=1}^{30} i x_i^4, -512 \leq x_i \leq 512 \tag{9}$$

#### ② Davis 함수

$$f_4(x_1, x_2) = 0.5 - \frac{(\sin(\sqrt{x_1^2 + x_2^2}))^2 - 0.5}{1.0 + 0.001(x_1^2 + x_2^2)^2}, -100 \leq x_i \leq 100 \tag{10}$$

#### ③ Bohachevsky 함수

$$f_5(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7, -6 \leq x_i \leq 6 \tag{11}$$

#### ④ Matyas 함수

$$f_6(x_1, x_2) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2, -100 \leq x_i \leq 100 \tag{12}$$

#### ⑤ Box 함수

$$f_7(x_1, x_2, x_3) = \sum_{i=1}^{10} [e^{-0.1ix_1} - e^{-0.1ix_2} - x_3(e^{-0.1i} - e^{-i})]^2, -100 \leq x_i \leq 100 \tag{13}$$

⑥ Beale 함수

$$f_8(x_1, x_2) = [1.5 - x_1(1 - x_2)]^2 + [2.25 - x_1(1 - x_2^2)]^2 + [2.625 - x_1(1 - x_2^3)]^2, \quad -100 \leq x_i \leq 100 \quad (14)$$

⑦ Floudas 함수

$$f_9(x_1, \dots, x_4) = 6 - 5 \sin(x_1) \sin(x_2) \sin(x_3) \sin(x_4) \sin(x_5) - \sin(5x_1) \sin(5x_2) \sin(5x_3) \sin(5x_4) \sin(5x_5), \quad -10 \leq x_i \leq 10 \quad (15)$$

⑧ Colville 함수

$$f_{10}(x_1, \dots, x_4) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1), \quad -10 \leq x_i \leq 10 \quad (16)$$

⑨ Griewank 함수

$$f_{11}(x_1, \dots, x_{30}) = 1 + \frac{1}{4000} \sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} \left( \cos\left(\frac{x_i}{\sqrt{i}}\right) \right), \quad -512 \leq x_i \leq 512 \quad (17)$$

⑩ Rastrigin 함수

$$f_{12}(x_1, \dots, x_{20}) = 200 + \sum_{i=1}^{20} (x_i^2 - 10 \cos(2\pi x_i)), \quad -5.12 \leq x_i \leq 5.12 \quad (18)$$

⑪ Ackley 함수

$$f_{13}(x_1, \dots, x_{30}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^{30} x_i^2}\right) - \exp\left(\frac{1}{30} \sum_{i=1}^{30} \cos(2\pi x_i)\right) + 20 + e, \quad -30 \leq x_i \leq 30 \quad (19)$$

⑫ Schwefel 함수

$$f_{14}(x_1, \dots, x_{10}) = 418.9829 + \sum_{i=1}^{10} [x_i \sin(\sqrt{|x_i|})], \quad -500 \leq x_i \leq 500 \quad (20)$$

4. 사례연구

본 논문에서 제안한 타부 탐색 전략의 성능을 평가하기 위해 앞장에서 보여진 평가함수[4-5, 10]에 대해 시뮬레이션을 수행하였다. 시뮬레이션은 단계적으로 수행하였는데, 첫 번째로 기본 탐색과정에 의한 결과를 보였으며, 두 번째로는 타부 탐색과정과 일정 감소율에 의한 재설정 방법의 결과를 보였고, 마지막으로 재설정 방법을 수정하기 위해 확률적 타부 전략을 이용한 결과를 보였다. 이러한 과정에 따라 가장 적합한 최적화 방법을 선택할 수 있도록 하였으며, 제안한 방법에 의해 구해진 결과를 기존의 다른 방법들과도 비교, 검토함으로써 제안한 방법이 기존의 방법에 비해 수렴 속도 및 전역 탐색 능력을 크게 향상시킬 수 있음을 보였다.

4.1 타부 탐색 과정에 의한 시뮬레이션 결과

타부 탐색과정은 초기해로부터 근접구간과 신뢰구간 내에서 이웃해를 생성하며 탐색하는 과정으로, 구간의 크기를 점차 감소시키며 국부 또는 전역최적값으로 수렴할 때까지 탐색을 진행한다. 이때 근접구간의 감소율( $\alpha$ )과 이웃해로부터 후보해 생성 비율에 의해 탐색 성능이 달라지므로 이러한 비율에 따라 시뮬레이션을 수행하여 성능을 검토하였다. 이때 사용된 시뮬레이션 파라미터는 표 2와 같다. 이웃해 구성하기 위한 구간별 비율은 표 2에서와 같이 경험적으로 선정하였고, 근접구간 감소율은 앞의 식 (2)를 이용하여 설정하였다. 예를 들어 50회 반복에 의해 수렴조건  $10^{-5}$ 값 이내로 수렴하고자 한다면 식 (21)을 만족하도록 근접구간 감소율  $\alpha$ 를

구할 수 있다.

$$\alpha^k < \epsilon \Leftrightarrow \alpha^{50} < 10^{-5} \Rightarrow \alpha \approx 0.79 \quad (21)$$

표 2 기본 탐색 과정시의 시뮬레이션 파라미터

Table 2 Simulation parameters for basic search procedure

계수	이웃해 개수	최대 반복 횟수	실행 횟수	수렴 조건	근접구간 초기값	이웃해구성			후보해 비율	근접구간 감소율 ( $\alpha$ )
						근접	혼합	신뢰		
설정 값	100	1000	100	$10^{-5}$	전체의 1%	20%	20%	60%	50~90%	0.32~0.98

그림 7에서는 서로 다른 수렴특성을 갖는 함수에 대해, 감소율( $\alpha$ )에 따른 최적해 탐색 성공률과 평균 반복횟수를 보여 주고 있다. 그림에서 보듯이 각 함수별로 가장 나은 탐색 성능을 나타내는 감소율은 서로 달라, 적절한 감소율을 선택하는 것에 어려움이 있음을 알 수 있다. 그림 8은 후보해 비율을 달리한 경우에 대한 수렴 특성을 보였다. 이때 후보해 비율이 낮은 경우 보다 빠르게 수렴하는 것을 알 수 있다. 전체 함수를 수렴특성에 따라 유형별(A, B, C, D, E)로 나누어 각 유형별 수행결과를 표 3에 나타내었다. A 유형의 경우에는 국부값이 하나이므로 어떤 감소율을 적용하더라도 항상 수렴하였으며, B, C, D 유형의 경우는 소수의 국부값을 가지는 것으로 적절한 비율의 감소율에서 가장 나은 성능을 나타내었다. 그리고 E 유형의 경우는 많은 독립변수와 국부값이 존재하여 기본 탐색 과정만으로는 국부값에 쉽게 빠져 전역 최적값을 찾기가 매우 어려웠다.

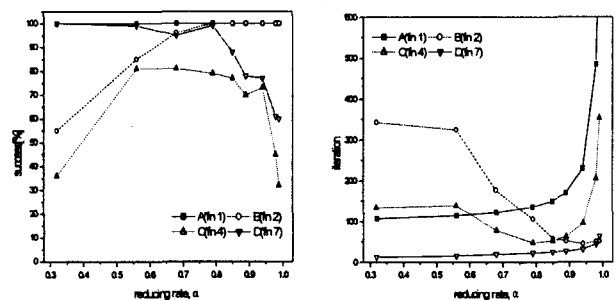


그림 7 근접구간 감소율( $\alpha$ )에 따른 탐색 성능(성공률, 평균탐색횟수)  
Fig. 7 Search performance according to reducing rates of near range

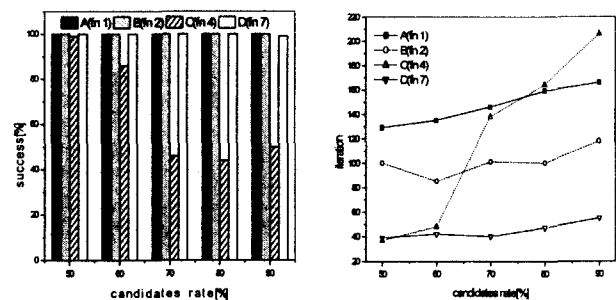


그림 8 각 함수의 후보해 비율에 따른 탐색 성능  
Fig. 8 Search performance according to candidate rates from neighbors

표 3 함수 유형별 수렴 특성

Table 3 Convergence features according to the types of functions

함수 유형	함수	국부값 개수	최적해탐색 성공률[%]	근접구간 감소율( $\alpha$ )	
				수렴경향	최적값
A	$f_1, f_5, f_6, f_8$	1	100	무관	0.32
B	$f_2, f_{10}$	소수	100	크다	0.99
C	$f_4, f_9$	소수	100	중간	0.68
D	$f_7$	소수	100	작다	0.32
E	$f_3, f_{11}, f_{12}, f_{13}, f_{14}$	다수	0	-	-

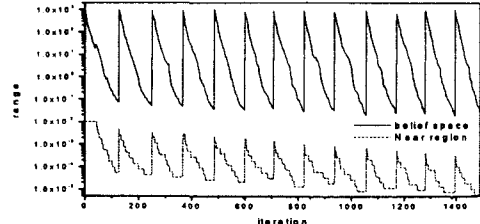
4. 2 재설정 방법에 의한 타부 탐색과정의 확장

타부 탐색 과정만으로 탐색할 경우 국부값이 많은 문제(E 유형)에 있어서는 쉽게 국부값에 수렴하여 더 이상 새로운 영역의 탐색이 매우 힘들어진다. 이러한 경우 탐색영역을 다시 재설정 방법에 의해 새로운 영역을 탐색할 수 있다. 본 절에서는 타부 탐색과정 중 국부수렴한 것으로 판단된 경우 일정비율에 의해 신뢰구간과 근접구간을 새로운 값으로 재설정하는 방법을 사용하여 시뮬레이션을 수행하였다. 표 4는 이때 사용된 시뮬레이션 파라미터를 나타낸 것으로, 여기서 감소율( $\beta$ )은 재설정시 근접구간 초기치의 감소율을 나타낸다.  $\beta$ 값 역시 앞의 식 (21)과 동일한 방법으로 재설정횟수에 의해 구해진다. 그림 9는 Griewank 함수에서 재설정 방법에 의한 수렴 특성을 나타내고 있다. 그림 9(a)는 신뢰구간과 근접구간의 크기가 점차로 감소하다가 재설정 과정에 의해 다시 확장되는 것을 보여주고 있으며, 그림 9(b)에서는 설정구간내에서의 탐색에 의한 현재해와 최적해의 수렴추이를 보여준다. 그림에서 보는 바와 같이 재설정 방법에 의해 점차로 전역최적해로 수렴하는 것을 알 수 있다.

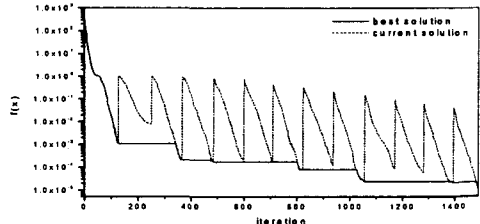
그림 10은 함수 유형에 따른 재설정 방법의 결과를 보여준다. 그림 10(a)에서 보는 바와 같이 A, B, C, D 유형의 함수에 대해서는 기본 탐색과정과 크게 차이가 나지 않는 것을 알 수 있으나, E 유형의 함수에 대해서는 성공률이 매우 높아졌다. 즉, 국부값이 많은 E 유형의 경우에도 탐색 성능이 크게 개선되는 것을 알 수 있다. 그림에서 R(0.56~1.0)으로 나타낸 것은 재설정시의 근접구간초기치 감소율( $\beta$ )을 나타낸 것으로, 앞의 식 (21)을 이용하여 구한 것이다. 그림 10으로부터 감소율( $\beta$ )이 클수록 전역최적해에 수렴할 가능성은 높으나 탐색횟수가 다소 증가하는 것을 알 수 있다. 그림 11은 후보해 비율에 따른 탐색 성능의 변화를 검토한 것이다. 그림으로부터 후보해 비율이 너무 적거나 크지않다면 감소율( $\beta$ )에 비해서 적은 영향을 미치는 것을 알 수 있다.

표 4 재설정 방법에서 사용한 시뮬레이션 파라미터  
Table 4 Simulation parameters for restarting method

계수	최대 반복횟수	수렴 조건	후보해 비율	최소 재설정횟수	근접구간 감소율	
					$\alpha$	$\beta$
설정값	5000	$10^{-5}$	60%	50	0.56~0.79	0.56~1.0

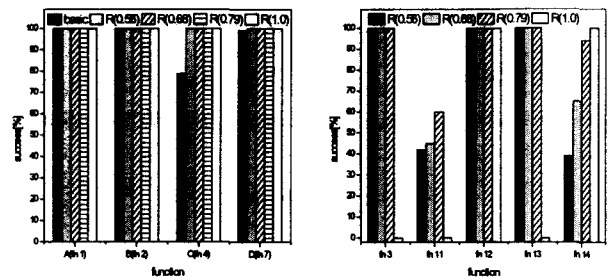


(a) 근접구간 및 신뢰구간의 크기 변화(Griewank 함수)

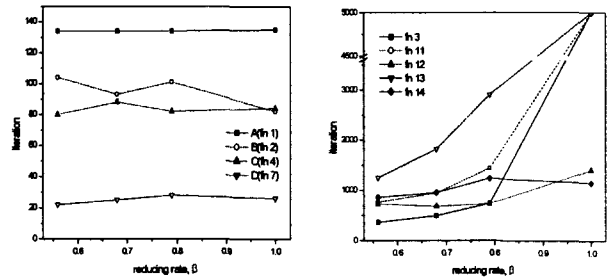


(b) 현재해와 최적해의 수렴 곡선(Griewank 함수)

그림 9 재설정시의 탐색 영역 변동 및 목적함수값 수렴 곡선  
Fig. 9 Variations of search range and objective value by Restarting



(a) 함수유형에 따른 성공률



(b) 함수 유형에 따른 평균 탐색 횟수( $\alpha=0.79$ )

그림 10 근접구간의 초기값 변화율에 따른 함수별 탐색 성능  
Fig. 10 Search performance of functions by reduction rates  $\beta$

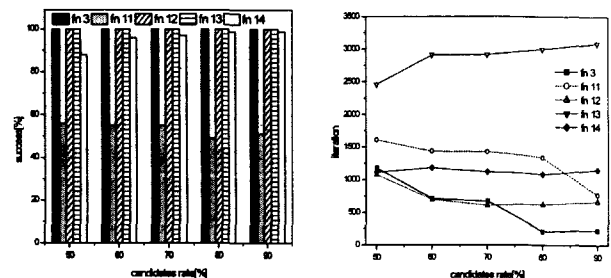


그림 11 E 유형 함수의 후보해 비율에 따른 탐색 성능

변동 ( $\alpha=0.79, \beta=0.79$ )

Fig. 11 Variation of search performance of E-type functions according to candidates rate

4. 3 확률 타부 전략을 이용한 재설정 방법

앞 절에서는 재설정시 일정한 비율로서 구간을 변경한 결과에 대해 알아보았다. 그러나 이러한 경우 문제에 따른 수렴특성을 제대로 반영하기가 어려워 다양한 함수의 최적해를 효과적으로 찾기는 어렵다. 따라서 본 논문에서는 확률 타부 전략을 이용하여 재설정시에 구간의 크기 및 감소율을 수렴특성에 따라 적응적으로 변경하는 방법을 제안하였다. 표 5는 이때 사용된 시뮬레이션 파라미터를 나타내었다. 그림 12는 일정한 감소율을 갖는 재설정 방법과 제안한 확률 타부 전략에 의한 탐색과정을 비교한 것으로, 제안한 방법은 재설정 이후의 전략에 따라 근접구간 감소율( $\alpha$ )과 초기값( $\beta$ )이 적응적으로 바뀌며 빠르게 최적해를 찾아감을 알 수 있다. 표 6은 이제까지의 방법을 평균 반복횟수와 성공률로서 비교하여 나타낸 것이다. 표 6으로부터 확률 타부 전략(PTS)에 의한 방법이 전체적으로 가장 우수한 결과를 나타냄을 알 수 있다.

표 5 확률 타부 전략에서의 시뮬레이션 계수  
Table 5 Simulation parameters for probabilistic tabu strategy

계수	최대 반복횟수	수렴조건	후보해 비율	근접구간 감소율	
				$\alpha$	$\beta$
설정값	5000	$10^{-5}$	60%	0.56~0.79	$10^{-3} \sim 0.02$

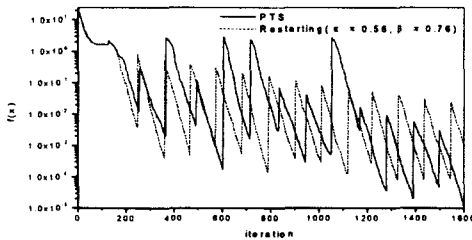


그림 12 함수  $f_{13}$ (Ackley)에 대한 목적함수값 비교

Fig. 12 Comparison of objective function values for function  $f_{13}$

표 6 각 방법에 따른 수행결과 비교

Table 6 Performance comparison of each method

함수	변수개수	평균반복횟수					성공률[%]				
		$A_{0.79}$	$B_{0.56}^{0.56}$	$B_{0.56}^{0.79}$	$B_{0.79}^{1.0}$	C	$A_{0.79}$	$B_{0.56}^{0.56}$	$B_{0.56}^{0.79}$	$B_{0.79}^{1.0}$	C
f1, Sphere	30	135	113	113	208	135	100	100	100	100	100
f2, Rosenbrock	2	105	168	153	79	100	100	100	100	100	100
f3, Quadratic	30	-	304	409	-	326	0	100	100	0	100
f4, Davis	2	47	134	134	83	95	79	100	100	100	100
f5, Bohachevsky	2	23	15	15	24	18	100	100	100	100	100
f6, Matyas	2	28	19	19	27	22	100	100	100	100	100
f7, Box	3	22	16	15	31	20	99	100	100	100	100
f8, Beale	2	40	30	27	41	31	100	100	100	100	100
f9, Floudas	5	59	64	73	85	78	92	100	100	100	100
f10, Colville	4	-	1335	1035	1577	1054	0	100	100	96	100
f11, Griewank	30	-	740	1325	-	1140	0	45	59	0	100
f12, Rastrigin	20	-	739	723	886	758	0	100	100	100	100
f13, Ackley	30	-	1098	2351	-	1615	0	100	100	0	100
f14, Schwefel	10	-	1098	1327	1447	1391	0	39	98	100	100

4. 4 다른 방법과의 비교

제안한 확률적 타부 탐색 전략에 의해 실행한 결과의 성능을 검토하기 위해 참고문헌[4-5]의 진화 프로그래밍(EP) 및 실수형 유전 알고리즘(RGA)과 비교하였다. 각 방법에 대해 평균 세대수와 탐색시간 및 주어진 반복횟수이내에 최적해를 찾을 확률(성공률)을 비교하여 표 7에 나타내었다. 여기서 진화 프로그래밍에 의한 결과는 Griewank 함수( $f_{11}$ )와 Ackley 함수( $f_{13}$ )에서 비교적 빠른 수렴을 나타내었다. 그러나 이 방법[4]은 이웃해를 변경하는 방법에 따라 제안된 4가지 방법 중에 가장 뛰어난 결과를 나타낸 것으로 문제에 따라 적합한 방법을 선택하기가 어려운 단점이 있다. 실수형 유전 알고리즘[12]의 경우는 유전알고리즘의 특성상 최적해 근처로는 빨리 수렴하나 최적해를 찾는데 많은 시간을 요구하며, 대상 문제에 따라 성능 차이가 심하여 연속적 탐색영역을 갖는 문제의 최적화 방법으로는 그다지 적합하지 않은 것을 알 수 있다. 이에 반해 제안한 확률적 타부 탐색 전략은 다양한 문제에 있어 전역 최적해의 탐색 성능이 뛰어나고, 빠른 수렴시간을 가진다. 특히 제안한 방법은 파라미터의 선정 문제에 있어 확률 타부 전략에 의해 자동적으로 선택 가능하므로 많은 문제를 효과적으로 해결할 수 있다. 모든 결과는 Pentium III(800MHz)의 PC 환경에서 100번 수행한 평균값을 나타내었다.

5. 결 론

본 논문에서는 새로운 확률적 타부 탐색전략을 이용하여 함수 최적화 문제를 해결하는 방법을 제시하였다. 전체 탐색 구간을 신뢰구간과 근접구간으로 분리한 후, 이 구간내에서 이웃해를 생성함으로써 효과적인 이웃해를 구성하도록 하였고, 현재해가 국부수렴할 경우 이제까지 탐색하지 않은 새로운 영역을 탐색하기 위한 재설정 방법, 그리고 재설정시에 탐색 영역 및 감소율 등을 조절하기 위해 확률 타부 전략을 사용함으로써 탐색 성능을 높일 수 있었다. 제안한 방법을 다양한 평가함수에 단계적으로 적용하여 그 특성을 분석함으로써 다른 문제에 적용이 용이하도록 하였으며, 다른 방법(EP, RGA)들과도 비교하여 수렴 속도 및 탐색 능력에 있어서 매우 우수함을 입증하였다.



표 7 수렴조건이  $10^{-5}$ 일 때 진화 프로그램, 유전알고리즘과 타부 탐색법의 성능비교  
 Table 7 Performance comparison of EP[4], RGA and PTS when convergence criterion is  $10^{-5}$

함수	변수개수	평균세대수			성공률(%)			평균시간[sec]		
		EP	RGA	PTS	EP	RGA	PTS	EP	RGA	PTS
f1, Sphere	30	500	5807	135	100	100	100	8.58	13.34	0.14
f2, Rosenbrock	2	500	6259	100	100	100	100	1.59	10.55	0.04
f3, Quadratic	30	800	731	326	100	100	100	4.76	3.15	0.74
f4, Davis	2	-	263	95	-	100	100	-	0.53	0.05
f5, Bohachevsky	2	200	38	18	100	100	100	0.31	0.10	0.01
f6, Matyas	2	500	59	22	100	100	100	0.30	0.14	0.01
f7, Box	3	500	36	20	100	100	100	1.40	0.16	0.05
f8, Beale	2	500	1090	31	100	100	100	0.45	1.73	0.01
f9, Floudas	5	400	1563	78	100	100	100	0.73	3.08	0.05
f10, Colville	4	2000	18864	1054	100	4	100	20.31	32.02	0.48
f11, Griewank	30	1000	15461	1140	70	61	100	28.70	66.24	1.81
f12, Rastrigin	20	2000	1138	758	10	100	100	50.39	3.23	0.88
f13, Ackley	30	800	22954	1615	100	100	100	17.24	69.95	2.97
f14, Schwefel	10	-	7344	1391	-	100	100	-	18.42	1.31
가장우수한 경우수		2	0	12	10	12	14	-	0	14

참 고 문 헌

[1] J. M. Renders and S. P. Flasse, "Hybrid Methods Using Genetic Algorithms for Global Optimization", IEEE Trans. on Systems, Man and Cybernetics-Part B, Vol. 26, No. 2, pp. 243-258, April, 1996

[2] Zi Gang, Peng Chuwu, Zou Mingzhu, "A modified genetic algorithm based on the best schema and its application for function optimization", Proc. of the 3rd World Congress on Intelligent Control and Automation, Vol. 1, pp. 615-618, 2000

[3] C.H.M. van Kemenade, J.N. Kok and A.E. Eiben, "Raising GA Performance by Simultaneous Tuning of Selective Pressure and Recombination Disruptiveness", IEEE ICEC, Vol. 1, pp. 341-351, 1995

[4] Xin Yao, Yong Liu, and G. Lin, "Evolutionary Programming Made Faster", IEEE Trans. on Evolutionary Computation, Vol. 3, No. 2, pp. 82-102, July 1999

[5] R. G. Reynolds and ChanJin Chung, "Knowledge - based Self-adaptation in Evolutionary Programming Using Cultural Algorithms", IEEE ICEC, pp. 71-76, 1997

[6] Fred Glover, Manuel Laguna, "Tabu Search", Kluwer Academic Publishers, 1997

[7] M.D. Asic and V.V. Kovacevic-Vujcic, "TABU Search Methodology in Global Optimization", Computers and Mathematics with Applications 37, pp. 125-133, 1999

[8] Yang Shiyu, Ni Guangzheng, Li Yan, Tian Baoxia and Li Ronglin, "An Universal Tabu Search Algorithm for Global Optimization of Multimodal Functions with Continuous Variables in Electromagnetics", IEEE Trans. on Magnetics, Vol. 34, No. 5, pp. 2901-2904, Sep. 1998

[9] A. Schaerf, M. Cadoli and M. Lenzerini, LOCAL++: A C++ framework for local search algorithms, Software. Pract. Exper. 30, pp. 233-257, 2000

[10] 김여근, 윤복식, 이상복, "메타 휴리스틱", 영지문화사,

2000

[11] 천희주, 김형수, 황기현, 문경준, 박준호, "Tabu 탐색법을 이용한 화력 발전기의 기동정지계획", 대한전기학회 논문지, Vol. 49A, No. 2, pp. 70-77, Feb. 2000

[12] 황기현, 문경준, 박준호, 정정원, "진화전략과 유전 알고리즘을 이용한 적응진화연산", 대한전기학회 논문지, 47권 8호, pp. 1262-1268, Aug. 1998

저 자 소 개



김형수 (金亨洙)

1972년 1월 26일 생. 1994년 부산대 공대 전기공학과 졸업. 1997년 동 대학원 전기공학과 졸업(석사). 1997년~현재 동 대학원 전기공학과 박사과정. 현재 연구분야는 지능 시스템 및 전력계통 최적응용 분야

Tel : 051-510-3188

E-mail : kimhsu@hyowon.pusan.ac.kr

황기현 (黃琪鉉)

전기학회 논문지 제 50A권 제 5호 참조

Tel : 051-510-3527

E-mail : hwanggh@hyowon.pusan.ac.kr

박준호 (朴俊瀾)

전기학회 논문지 제 50A권 제 5호 참조

Tel : 051-510-3188

E-mail : jhpark@hyowon.pusan.ac.kr