

# A Continuous Optimization Algorithm Using Equal Frequency Discretization Applied to a Fictitious Play

Chang-Yong Lee<sup>†</sup>

Department of Industrial and Systems Engineering, Kongju National University

## 동일 빈도 이산화를 가상 경기에 적용한 연속형 최적화 알고리즘

이 창 용<sup>†</sup>

공주대학교 산업시스템공학과

In this paper, we proposed a new method for the determination of strategies that are required in a continuous optimization algorithm based on the fictitious play theory. In order to apply the fictitious play theory to continuous optimization problems, it is necessary to express continuous values of a variable in terms of discrete strategies. In this paper, we proposed a method in which all strategies contain an equal number of selected real values that are sorted in their magnitudes. For comparative analysis of the characteristics and performance of the proposed method of representing strategies with respect to the conventional method, we applied the method to the two types of benchmarking functions: separable and inseparable functions. From the experimental results, we can infer that, in the case of the separable functions, the proposed method not only outperforms but is more stable. In the case of inseparable functions, on the contrary, the performance of the optimization depends on the benchmarking functions. In particular, there is a rather strong correlation between the performance and stability regardless of the benchmarking functions.

Keywords : Fictitious Play, Continuous Optimization Algorithm, Equal Frequency Discretization, Equal width Discretization

### 1. 서 론

지역 최적해(local optimum)를 많이 가지는 복잡한 시스템에서 전역 최적해(global optimum)를 찾는 최적화 문제(optimization problem)는 많은 경우 다항식 시간(polynomial time)에 전역 최적해를 찾기 어렵다고 판단되는 NP-hard 문제[9]에 속한다. 따라서 주어진 시간 내에 전역 최적해에 가까운 근사해를 찾는 방법론이 보다 현실적인 대안인데, 경험 및 학습을 통한 문제 해결법으로 알려진 메타 휴리스틱스(meta-heuristics)[21]를 그 대안의 예로 들 수

있다. 메타 휴리스틱스는 주어진 문제의 특성에 크게 구속되지 않는 유연성을 가지고 있기 때문에 다양한 문제에 적용 가능한 특징이 있으며, 유전자 알고리즘(genetic algorithm)[15], 진화 프로그래밍(evolutionary programming)[1], 진화 전략(evolutionary strategies)[5] 등의 방법론들이 있다.

최근 들어 가상 경기 이론(fictitious play theory)[2, 6, 7]을 메타 휴리스틱스 유형의 최적화 방법론에 적용하려는 시도가 새로운 연구 분야로 대두되고 있다. 그 이유는 가상 경기 이론은 내쉬 균형(Nash equilibrium)[17, 18] 자체를 중요시 하는 비협조 게임 이론(non-cooperative game theory)과 달리, '게임을 통한 학습'(learning in games) 관점에서

최적화 문제 해결을 위한 방법론으로 적용될 수 있기 때문이다. 가상 경기는 진화 게임 이론 중에서 학습 규칙을 위한 대표적인 모델로 1951년 Brown[2]에 의해 처음으로 제안되었고, 그 이후 Fudenberg and Levine에 의하여 더욱 발전되었다[6, 7].

가상 경기를 이용한 최적화 모델 및 방법론에 관한 연구는 아직 초기 단계라 할 수 있다. 가상 경기를 최적화 문제를 위한 방법론으로 적용할 수 있는 이론적 기초를 제공한 연구로 Monderer and Shapley의 연구[16]와 Lambert 등의 연구[10]를 들 수 있다. Monderer and Shapley는 모든 경기자가 동일한 효용치 함수(utility function)를 가지는 경우에도 가상 경기의 특성인 수렴 현상이 성립함을 증명하였다. 가상 경기에서 경기자는 일반적으로 서로 다른 효용치 함수를 가질 수 있는데, 모든 경기자가 동일한 효용치 함수를 가지는 경우는 최적화 문제에 해당한다. 따라서 Monderer and Shapley의 연구는 가상 경기 이론을 최적화 방법론으로 적용할 수 있는 이론적인 근거를 제공한다. 또한 Lambert 등은 가상 경기를 대규모 최적화 문제에 적용하기 위하여 경기자가 선택할 수 있는 모든 전략을 고려하지 않고 표본 추출을 통한 표본 전략을 가상 경기에 적용하여 수렴 현상을 연구하였다.

가상 경기에서 각 경기자에게 주어진 전략은 이산적(discrete)이라는 특성으로 인하여 가상 경기의 최적화 문제 응용은 조합적 최적화 문제(combinatorial optimization problem)에 먼저 적용되었다. 조합적 최적화 문제에 가상 경기 이론을 적용한 연구는 주로 컴퓨터 네트워크 분야로, 동적 트래픽 네트워크에서 실시간 최적 라우팅(routing)을 위해 가상 경기를 적용한 연구[8]와 무선 네트워크에서 사용자들의 파워 조절(power control)을 위하여 비협조적 게임 이론과 가상 경기를 적용한 연구[14]를 예로 들 수 있다. 또한 가상 경기와 기계 학습(machine learning) 간의 상호 관련성 및 유사성에 대한 연구[19]도 흥미할 만하다.

가상 경기 이론은 그 구조적 특성으로 인하여 조합적 최적화 문제에 적용이 비교적 용이한 반면, 연속형 최적화 문제(continuous optimization problem)의 경우에는 연속적인 값을 이산적인 전략으로 표현하는 방법을 고안해야 함으로 조합적 최적화 문제에 비하여 연구가 부족한 실정이다. 최근 가상 경기 이론을 연속형 최적화 문제에 적용할 수 있는 알고리즘에 대한 연구[12]가 진행되었다. 이 연구에서는 연속적인 값을 가지는 변수의 영역을 정해진 개수의 부분 영역으로 나누고, 부분 영역들을 전략으로 간주하는 방법을 선택하였다. 또한 각 부분 영역(즉, 전략)의 범위는 동일한 크기를 가지고 게임이 진행되는 동안 전략의 크기는 변하지 않는다.

가상 경기를 연속형 최적화 문제에 적용하기 위해서

해결해야 할 중요한 문제 중 하나가 연속형 변수의 값을 이산적인 전략으로 표현하는 것임을 고려할 때, 다양한 이산화 방법에 대한 연구가 필요하다. 연속적인 값을 이산화(discretization)하는 기본적인 방법으로 동일 너비 이산화(EWD, Equal Width Discretization)와 동일 빈도 이산화(EFD, Equal Frequency Discretization) 등으로 크게 구분할 수 있다[3, 4]. EWD는 모든 부분 영역이 동일한 너비(혹은 크기)를 가지도록 변수의 영역을 미리 정한 개수의 부분 영역으로 나누는 방법을 말하며, 이때 부분 영역의 범위는 일정하며 변하지 않는다. 이에 반하여 EFD는 선택된 실수 값들을 크기순으로 나열한 다음, 같은 개수의 값들이 각 부분 영역에 속하도록 부분 영역의 범위를 정하는 방법이다. 따라서 EFD에서 각 부분 영역의 크기는 선택된 실수 값들에 의존하며 일반적으로 동일하지 않다. 가상 경기를 적용한 연속형 최적화 알고리즘에서 전략 결정 방법의 중요성을 고려할 때, 다양한 이산화 방법론을 성능 측면에서 비교 분석하는 것은 중요한 연구 주제라 할 수 있다.

본 연구에서는 EFD를 가상 경기 이론에 적용한 연속형 최적화 알고리즘을 제안하고, 기존의 EWD를 사용한 최적화 알고리즘과 성능 및 특성 측면에서 비교 분석하고자 한다. 즉, 본 연구에서는 EFD를 기반으로 한 전략 결정 방법을 사용하여 최적화 알고리즘을 구현하고 벤치마킹 문제에 적용하여, 제안한 방법이 어느 정도 효율적으로 적용될 수 있는지 기존 방법과 비교를 통하여 파악하고자 한다. 위에서 언급한 두 가지 이산화 방법은 각기 고유한 장단점을 가지고 있기 때문에 어느 한 방법이 다른 방법에 비하여 모든 최적화 문제에 우수한 성능을 보일 것으로 예상되지는 않는다. 다만 본 논문에서는 연속형 최적화 문제에 제안한 이산화 방법의 적용 가능성을 파악하고, 두 가지 이산화 방법의 장단점을 분석하고자 한다.

본 논문은 다음과 같이 구성되어 있다. 제 2장에서는 가상 경기 이론에 대한 개요를 언급하고, EFD를 사용한 연속형 최적화 알고리즘을 제안하였다. 제 3장에서는 제안한 이산화 방법의 특성 및 성능을 분석하기 위하여 벤치마킹 함수에 대하여 실험을 수행하여 그 결과를 기존 방법과 대비하여 분석하였고, 마지막 장에 향후 연구를 포함한 본 논문의 요약과 결론을 맺었다.

## 2. 동일 빈도 이산화를 사용한 최적화

### 2.1 가상 경기 이론과 전략 표현 방법

전형적인 가상 경기에서 경기자  $i(i = 1, 2, \dots, n)$ 의 표

현형(genotype)은 고유한 전략 집합과 각 전략이 선택될 확률 집합으로 나타낼 수 있다. 경기자  $i$ 가  $n_i$ 개의 전략을 가지고 있을 때, 경기자  $i$ 의 전략 집합은  $s^i = \{s_1^i, s_2^i, \dots, s_{n_i}^i\}$ 로 표현할 수 있으며, 경기가 반복되는 동안 각 전략이 선택될 확률로 구성된 확률 집합은  $\Sigma^i = \{\sigma_1^i, \sigma_2^i, \dots, \sigma_{n_i}^i\}$ 로 나타낼 수 있다. 여기서  $\sigma_k^i$ 는 경기자  $i$ 의  $k$ 번째 전략  $s_k^i$ 가 선택될 확률로 모든  $k$ 에 대하여  $\sigma_k^i > 0$ 이며  $\Sigma_{k=1}^{n_i} \sigma_k^i = 1$ 을 만족한다[2, 6, 7].

가상 경기에서 각 경기자는 경기를 반복하는 과정에서 자신의 효용치 함수를 최대로 하는 전략을 선택한다. 경기자  $i$ 의 효용치 함수  $u^i(s)$ 는 실수 값으로 전략 프로파일(strategy profile)  $s = (s^i, s^{-i})$ 의 함수이다. 여기서  $-i$ 는 경기자  $i$ 를 제외한 모든 경기자를 집합적으로 표현한 것이다. 따라서 경기자  $i$ 의 효용치 함수  $u^i(s)$ 는 비단 자신의 전략  $s^i$ 뿐만 아니라 상대 경기자들의 전략  $s^{-i}$ 에도 의존한다. 상대 경기자들이  $\sigma^{-i}$ 로 주어지는 확률을 사용하여 전략  $s^{-i}$ 을 선택할 때, 경기자  $i$ 는 자신의 모든 전략  $s^i \in S^i$ 에 대하여  $u^i(s^i, \sigma^{-i}) \geq u^i(s^j, \sigma^{-i})$ 를 만족하도록 전략  $\hat{s}^i$ 을 선택한다. 이때  $\hat{s}^i$ 를 경기자  $i$ 의 최상 반응(best response)이라 하며

$$\hat{s}^i \in \operatorname{argmax}_{s^i \in S^i} \{u^i(s^i, \sigma^{-i})\} \quad (1)$$

로 주어진다. 따라서 각 경기자가 선택하는 전략은 경기를 반복하는 과정에서 갱신될 수 있다. 경기자  $i$ 에 대한 최상 반응을 구하기 위해서는 모든 경기자가 선택할 수 있는 전략 공간 전체에 대하여 효용치 함수를 계산해야 하는데, 전략 공간 전체를 탐색하는 것은 변수의 개수와 전략 개수가 많은 경우에는 비현실적임으로 확률  $\sigma^{-i}$ 를 사용하여 적절한 개수의 표본을 추출하여 구한 효용치 함수의 평균을 일반적으로 사용한다[10].

가상 경기에 기초한 연속형 최적화 알고리즘에서 효용치 함수는 최적화 대상이 되는 목적 함수(objective function)에 해당하며, 각 경기자는 목적 함수의 독립 변수에 해당하고, 각 경기자가 선택하는 전략은 독립 변수가 취할 수 있는 값으로 구성된다. 가상 경기에서 각 경기자는 유한개의 전략 중에서 하나를 선택하기 때문에 가상 경기 이론을 연속형 최적화 알고리즘에 적용하기 위해서는 연속적인 변수 값을 이산적인 전략으로 표현해야 한다.

EWD에서 경기자가 전략을 선택한다는 것은 독립 변수(혹은 경기자)  $i$ 의 영역을  $n_i$ 개의 부분 영역으로 나누고 그 중 하나를 선택한 다음, 선택된 부분 영역에 속하는 값을 임의로 선택하는 것을 의미한다[12]. 즉, 변수  $i$  ( $i=1, 2, \dots, n$ )의 범위가  $R^i = [L^i, U^i]$ 로 주어지고  $n_i$ 개

의 전략을 가지고 있다고 하면,  $k$ 번째 부분 영역  $s_k^i$ 는

$$s_k^i = [L^i + r^i(k-1), L^i + r^i k], r^i = (U^i - L^i)/n_i \quad (k=1, 2, \dots, n) \quad (2)$$

로 표현되며, 경기자  $i$ 가  $k$ 번째 부분 영역(혹은 전략)  $s_k^i$ 를 선택한다는 것은 부분 영역  $[L^i + r^i(k-1), L^i + r^i k]$ 에 속하는 임의의 값을 선택하는 것을 의미한다. EWD의 특징은 모든 부분 영역의 크기가  $r^i$ 로 동일하고, 경기가 반복적으로 진행되는 동안 각 부분 영역의 크기는 변하지 않는다는 것이다.

이와 달리 EFD는 변수  $i$ 의 범위  $R^i = [L^i, U^i]$ 내에 가상 경기가 반복되는 동안 최상 반응에 의해 선택된 실수 값들의 빈도(frequency)가 동일하도록  $n_i$ 개의 부분 영역을 정하는 방법이다. 모든 부분 영역에 동일한 개수의 실수 값이 할당됨으로 각 부분 영역의 크기는 선택된 실수 값에 의존하며 가상 경기가 반복되는 동안 선택된 값의 개수에 따라 각 부분 영역의 크기가 변한다. EFD의 특징은 선택된 값들을 오름차순으로 정렬하였을 때 그 값들의 차이가 작은 경우는 변수의 해당 영역을 조밀하게 조사(exploitation)하는 것에 해당하며, 그 반대로 큰 경우에는 넓게 조사(exploration)하는 것에 해당한다. 따라서 EFD는 위의 두 가지 경우가 조화를 이루도록 하여 가상 경기를 통한 학습의 효율성을 높이는 결과를 가져다준다.

본 연구에서는 모든 경기자가 같은  $q$ 개의 전략을 가지고 있다고 가정하고 EFD를 다음과 같이 구현하였다.

- 초기화 : 경기자  $i$ 의 변수 범위  $R^i = [L^i, U^i]$  ( $i=1, 2, \dots, n$ )을 동일한 크기를 가지는  $q$ 개의 부분 영역(즉, 전략)으로 나눈다. 즉, 변수  $i$ 의  $k$ 번째 부분 영역인  $s_k^i$ 는 EWD의 경우와 유사하게

$$s_k^i = [L^i + r^i(k-1), L^i + r^i k], r^i = (U^i - L^i)/q \quad (k=1, 2, \dots, q) \quad (3)$$

고 주어진다.

- 전략 범위 결정 : 가상 경기를  $q$ 번 반복할 때마다 각 경기자의 전략 범위를 갱신한다. 가상 경기를  $jq$ 번 ( $j=1, 2, \dots$ ) 반복하여  $jq$ 개의 실수 값이 선택되면, 각 부분 영역에 동일한  $j$ 개의 실수 값이 포함되도록 각 부분 영역을 다음과 같이 정한다. 경기자  $i$ 에 의하여 선택된 실수 값들을 오름차순으로 정렬하여  $x_1^i, x_2^i, \dots, x_{jq}^i$  이라 하면,  $j$ 개의 실수 값으로 구성된  $k$  ( $k=1, 2, \dots, q$ )번째 전략 범위  $s_k^i(j)$ 는

$$s_k^i(j) = \begin{cases} \left[ L^i, \frac{x_{jk}^i + x_{jk+1}^i}{2} \right] & k=1 \\ \left[ \frac{x_{j(k-1)}^i + x_{j(k-1)+1}^i}{2}, \frac{x_{jk}^i + x_{jk+1}^i}{2} \right], & 2 \leq k \leq q-1 \\ \left[ \frac{x_{j(k-1)}^i + x_{j(k-1)+1}^i}{2}, U^i \right], & k=q \end{cases} \quad (4)$$

로 주어진다. 따라서 각 부분 영역의 크기는 선택된 실수 값에 의존한다.

## 2.2 EFD를 적용한 연속형 최적화 알고리즘

연속형 최적화 문제는 다음과 같이 정의할 수 있다.  $\text{find min}\{u(s) : s = (s^1, s^2, \dots, s^n) \in S^1 \times \dots \times S^n\}$  여기서  $n$ 은 변수 개수(혹은 경기자 수)를 나타내며,  $u(s)$ 는 모든 경기자들에 대해 동일한 효용치 함수로 목적 함수에 해당하고,  $S^i$ 는 경기자  $i$ 의 전략 집합이다. 따라서 최적화 문제는 각 경기자  $i$ 가 유한개의 원소로 구성된 전략 집합  $S^i$ 과 동일한 효용치 함수  $u(s)$ 를 가지는 가상 경기로 간주할 수 있다. 전략 표현을 위해 EFD를 적용한 연속형 최적화 알고리즘은 아래와 같이 표현할 수 있다.

**Step 1** : 전략 범위 갱신 횟수를  $j=0$ , 반복을  $t=1$ 이라 두고, 각 경기자  $i(i=1, 2, \dots, n)$ 의 전략들이 선택될 확률을 동일하게

$$\sigma_{t=1}^i(s_1^i) = \sigma_{t=1}^i(s_2^i) = \dots = \sigma_{t=1}^i(s_{n_i}^i) = 1/q \quad (5)$$

로 선정한다. 전략 범위는 주어진 변수 영역을 전략 개수로 나눈 동일한 크기를 가지도록 식 (3)과 같이 정한다.

**Step 2** : 주어진 확률을 사용하여 각 경기자는 전략  $s^i$ 를 선택한다. 선택된 전략을 사용하여 경기자  $i(i=1, 2, \dots, n)$ 의 최상 반응  $s^i(t+1)$ 을 아래와 같은 방법으로 구한다.

$$s^i(t+1) \in \text{argmin}_{s^i \in S^i} \left\{ \overline{u}_m^i(s^i, \sigma_t^{-i}(s)) \right\} \quad (6)$$

여기서  $\overline{u}_m^i(s^i, \sigma_t^{-i}(s))$ 는 반복  $t$ 에서  $m$ 개의 표본을 통해 구한 효용치 함수들의 평균값이다. 즉, 주어진 확률  $\sigma_t^{-i}(s)$ 를 사용하여  $m$ 개의 서로 독립인 표본 전략  $Y_j^{-i}(t)$ , ( $j=1, 2, \dots, m$ )을 선택하고,  $m$ 개의 표본을 사용하여 효용치 함수들의 표

본 평균  $\overline{u}_m^i(s^i, \sigma_t^{-i}(s)) = \frac{1}{m} \sum_{i=1}^m u^i(s^i, Y_j^{-i}(t))$ 를 구한다. 이러한 과정을 모든  $s^i \in S^i$ 에 대해 실행하여  $\overline{u}_m^i(s^i, \sigma_t^{-i}(s))$ 가 최소가 되는  $s^i$ 를 반복 ( $t+1$ )에서 최상 반응  $s^i(t+1)$ 로 정한다. 모든 경기자의 최상 반응  $s^i(t+1)$ , ( $i=1, 2, \dots, n$ )를 사용하여 목적 함수  $u(s)$ 를 계산한다.

**Step 3** : 경기자  $i(i=1, 2, \dots, n)$ 가 선택한 전략  $s^i(t+1)$ 을 사용하여 전략 선택 확률을 아래 식 (7)을 사용하여 갱신한다.

$$\sigma_{t+1}^i(s_j^i) = \sigma_t^i(s_j^i) + \frac{1}{t+1} \left[ I_{t+1}^i(s^i) - \sigma_t^i(s_j^i) \right] \quad (7)$$

여기서  $I_{t+1}^i(s^i) = \begin{cases} 1, & \text{if } s^i(t+1) = s_j^i \text{이다.} \\ 0, & \text{otherwise} \end{cases}$

**Step 4** :  $t \leftarrow t+1$ 로 두고  $t = jq$ 일 때까지 step 2와 step 3을  $q$ 번 반복한다.

**Step 5** : 각 경기자에 대하여  $q$ 개의 실수 값이 선택되면, 각 전략에 동일한 개수의 값이 포함되도록 각 경기자의 전략 범위를 식 (4)와 같이 정한다. 그리고  $j \leftarrow j+1$ 로 둔다.

**Step 6** : step 2에서 step 5를 정해진 횟수만큼 반복한다.

## 3. 실험 결과 및 토론

제 2장에서 제안한 EFD를 사용한 연속형 최적화 알고리즘의 성능 및 특징을 EWD와 비교 분석하기 위하여 벤치마킹 함수[11, 20]를 사용하여 실험을 수행하였다. 본 실험에서 사용한 벤치마킹 함수와 변수 개수 및 변수 범위는 <Table 1>에 나타내었다.

모든 벤치마킹 함수는 지역 최적해가 많은 함수(multimodal function)로 정하였는데, 그 이유는 지역 최적해가 없는 함수(unimodal function)는 기존에 잘 알려진 최대 경사법(steepest descent method) 혹은 이와 유사한 방법을 사용하여 최적해를 찾을 수 있기 때문이다. 또한 벤치마킹 함수를 두 가지 유형으로 나누어 실험 결과를 분석하였다. 함수  $f_1$ 과  $f_2$ 는 분리 가능한 함수(separable function)로 변수들 간에 상호 작용이 없는 함수이며, 함수  $f_3$ - $f_5$ 는 분리 가능하지 않는 함수(inseparable function)로 변수 간에 상호 작용이 있는 함수이다.

<Table 1> Test Functions Used for the Experiment.  $n$  is the Number of Variables and  $R$  Represents the Range of a Variable

Test Functions	$n$	$R$
$f_1 = -\sum_{i=1}^n x_i \sin( x_i )$	10	[-20, 20]
$f_2 = -\sum_{i=1}^n \{x_i^2 - 10 \cos(2\pi x_i) + 10\}$	10	[-20, 20]
$f_2 = -20 \exp\left\{-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right\} - \prod_{i=1}^n \exp\left\{\frac{1}{n} \cos(2\pi x_i)\right\} + 20 + e$	10	[-20, 20]
$f_4 = \left(\sum_{i=1}^n x_i^2\right)^{0.25} \left\{\sin^2\left(50 \left(\sum_{i=1}^n x_i^2\right)^{0.1}\right) + 1\right\}$	10	[-20, 20]
$f_5 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	10	[-20, 20]

의 갱신 횟수가 증가함에 따라 최적해의 변화가 줄어들 수 있는데, 이것은 반복이 진행되면서 제한한 알고리즘이 전역 최적해로 점차적으로 수렴하기 때문으로 분석할 수 있다. 위에서 언급한 특징들은 정도의 차이는 있지만 다른 함수에 대해서도 유사하게 나타났다.

본 연구에서 적용한 EFD의 특징 중 하나는 전략 개수 만큼 가상 경기를 반복할 때마다 전략 범위를 갱신하는 것인데, 전략 범위 갱신에 따른 각 전략의 크기(즉, 각 전략 범위의 상한과 하한의 차이)의 변화를 함수  $f_2$ 에 대하여 <Figure 2>에 나타내었다. <Figure 2>를 통해 볼 수 있듯이 각 전략 범위는 동일한 크기로 고정되지 않고 전략이 갱신될 때마다 변화함을 알 수 있다. 이러한 특징은 모든 전략의 크기가 동일하고 전략 범위가 변하지 않는 EWD의 경우와 대조를 이룬다. 특히 전략 범위 갱신 횟수가 증가함에 따라 전략 크기가 줄어드는 전략(예를 들어 <Figure 2>에서 6번째에서 16번째 전략)들은 최상반응을 통한 전략 선택에서 이들 전략 범위에 속하는 값들이 반복이 진행됨에 따라 많이 선택됨을 의미한다.



<Figure 1> The Minimum Values Versus Iteration for Test Function  $f_1$ . The Arrows Indicate Iterations Right After Updating Strategy. (a)  $q=10$ , (b)  $q=20$

전략 범위의 갱신과 최적해 변화 사이의 관계를 살펴 보기 위하여 반복에 따른 최적해 변화를 함수  $f_1$ 에 대하여 구하여 <Figure 1>에 나타내었다. <Figure 1>을 통해 반복에 따른 최적해의 변화에 대한 세 가지 일반적인 특징을 알 수 있다. 첫째, 반복이 진행됨에 따라 최적해는 수렴하는 경향을 띤다. 이러한 경향은 대부분의 최적화 휴리스틱스에서 볼 수 있는 현상으로 반복이 진행됨에 따라 보다 향상된 최적해를 찾아감을 의미한다. 둘째, 전략 개수만큼의 반복이 진행될 때마다 전략 범위가 갱신되는데, 전략 범위가 유지되는 동안에는 반복에 따른 최적해의 변화가 크지 않으나 전략 범위가 갱신된 직후의 반복에서는 최적해의 변화가 상대적으로 현저함을 알 수 있다. 이것은 전략 범위의 갱신이 최적화에 효율적으로 적용됨을 나타내는 결과라 할 수 있다. 셋째, 전략 범위

<Figure 2> The Variation of Strategy Interval for Test Function  $f_2$  as the Strategy Intervals are Updated ( $\square$  : 2 Times,  $\circ$  : 3 Times,  $\triangle$  : 4 Times). The Abscissa Represents Strategy index with  $q=20$ , and the Ordinate Stands for the Magnitude of Strategy Interval. The Dotted line Indicates the Magnitude of Strategy Interval for EWD

두 가지 전략 방법(EWD, EFD)에 따른 최적화 성능을 비교 분석하기 위하여 각 함수들에 대한 최적해 결과를 전략 개수 및 표본 개수를 중심으로 도출하였으며, 대표적인 전략 및 표본 개수에 대한 결과를 <Table 2>에 요약하였다. 모든 결과는 10번 독립 시행한 통계치(평균과 표준편차)이며, 각 독립 시행은 100번 반복으로 구성되어 있다. <Table 2>를 통해 두 가지 전략 방법의 결과에 대한 대체적인 분석을 할 수 있다. 분리 가능한 함수( $f_1$

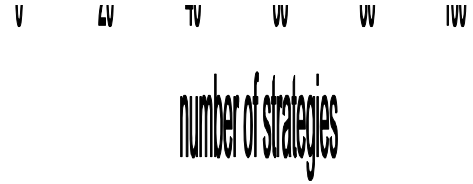
과  $f_2$ 의 경우에는 모든 매개 변수에 대하여 EFD이 EWD보다 우수한 결과를 나타내었으며, 분리 가능하지 않은 함수( $f_3 - f_5$ )에서는  $f_4$ 를 제외하면 EFD이 상대적으로 좋은 결과를 나타냈다. 이러한 결과를 통해 볼 때, EFD이 DWD보다 전반적으로 향상된 최적해를 찾을 수 있는 방법이라 할 수 있다. 또한  $f_4$ 를 제외한 모든 함수와 두 방법 모두에서, 표본 개수를 증가시키는 것보다 전략 개수를 증가시키는 것이 보다 향상된 결과를 도출할 수 있음을 알 수 있다. 이것은 최적해 성능은 표본 개수보다 전략 개수에 더 민감함을 의미한다. 이와 대조적으로 함수  $f_4$ 의 경우에는 표본 개수가 전략 개수보다 최적화 관점에서는 더 중요함을 알 수 있는데, 이것은  $f_4$ 는 최상 반응을 위한 전략 선택이 표본 개수에 민감함을 의미한다.

<Table 2> The Experimental Results for All Test Functions. The Values are the Average and Standard Deviation (in Parentheses) over 10 Independent Runs

	number of strategies = 50 number of samples = 10		number of samples = 10 number of strategies = 50	
	EFD	EWD	EFD	EWD
$f_1$	-194.15892 (0.03732)	-190.86683 (0.31254)	-194.25479 (0.000258)	-174.58012 (1.33347)
$f_2$	13.02272 (2.4321)	53.91057 (8.07947)	38.83202 (4.68554)	81.5518 (12.40174)
$f_3$	1.89458 (0.27821)	2.66789 (0.14502)	5.98968 (0.67931)	6.41847 (0.68814)
$f_4$	5.14456 (0.54404)	5.04352 (0.32859)	3.72991 (0.38775)	2.46209 (0.18613)
$f_5$	0.09052 (0.02433)	0.13915 (0.03274)	0.68825 (0.14006)	0.85393 (0.05514)

위에서 언급한 일반적인 결과에 추가적으로 더욱 세밀한 분석을 위해 제안한 알고리즘의 중요한 매개변수인 전략 개수와 표본 개수의 변화에 따른 두 방법(EFD와 EWD)의 결과를 비교 분석하였다. <Figure 3>은 분리 가능한 함수인  $f_1$ 과  $f_2$ 에 대하여 표본 개수 10개, 반복 횟수 100회 조건하에 구한 최적해의 통계치(10번 독립 시행에 대한 평균과 표준편차)를 나타낸 것이다. <Figure 3>에서 볼 수 있듯이 모든 전략 개수에 대하여 EFD이 EWD에 비하여 최적해 성능 면에서 우수하고 보다 안정적(즉, 표준편차가 적음)임을 알 수 있다. 일반적으로 전략 개수를 증가시키면 보다 많은 후보해를 조사할 수 있기 때문에 최적해 면에서도 나은 결과를 얻을 수 있는데,  $f_1$ 의 EFD를 제외한 모든 경우에서 같은 성향을 보였다. 그러나 <Figure 3>(a)에서 볼 수 있듯이  $f_1$ 의 EFD 경우에는 최적해가 전략 개수에 거의 무관한데, 이것은 EFD

를 사용한 전략 범위의 변화가 전략 개수 효과를 포함하는 것으로 추측된다.



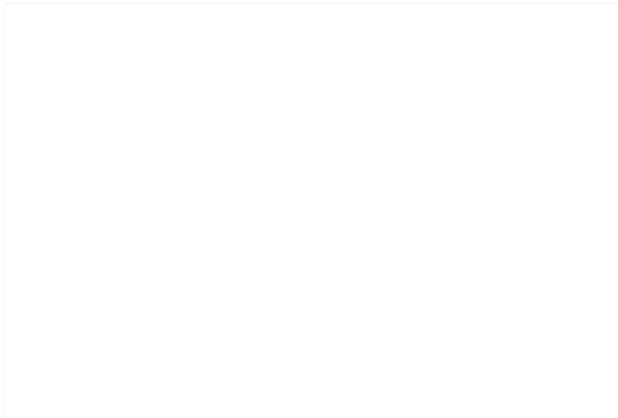
<Figure 3> The Average and Standard Deviation of Minimum Values Versus the Number of Strategies : EFD(■) and EWD(□). (a) The Result of Test Function  $f_1$ , (b) The Result of Test Function  $f_2$

<Figure 4> The Average and Standard Deviation of Minimum Values Versus the Number of Samples for Test Function  $f_2$  : EFD(■) and EWD(□)

표본 개수의 변화에 따른 최적해의 변동 추이를 비교하기 위하여 EFD와 EWD를 적용하여 실험을 수행하였으며, <Figure 4>에 함수  $f_2$ 에 대한 결과를 나타내었다. <Figure 4>를 통해 볼 때 표본 개수에 무관하게 EFD가 EWD에 비하여 성능 면에서도 우수하고 보다 안정적임을 알 수 있다. 특이할 만한 사항은 각 방법 내에서 최적해의 성능은 표본 개수에 많은 영향을 받지 않고, 두 방법의 성능 차이는 표본 개수에 비교적 무관하다는 것이다. 이러한 결과는 최상 반응을 위한 표본 개수는 최적화 결과에 큰 영향을 미치지 않음을 의미하며, 따라서 가능

한 모든 변수 영역을 조사하지 않고 표본을 통한 최상 반응을 구하는 방법에 대한 정당성을 부여한다고 볼 수 있다. 또한 함수  $f_1$ 의 경우에도  $f_2$ 의 경우와 유사한 결과를 얻었다.

<Figure 5>는 분리 가능하지 않는 함수  $f_3$ 과  $f_4$ 에 대하여 전략 개수에 따른 최적해의 성능을 두 가지 방법에 대하여 비교한 것이다.  $f_3$ 의 경우에는 전략 개수가 증가함에 따라 두 방법 모두 최적해의 성능도 향상되나, 모든 전략 개수에 대하여 EFD가 EWD에 비하여 우수함을 알 수 있다.  $f_5$ 에 대해서도  $f_3$ 과 유사한 결과를 얻었다. 이와 대조적으로 함수  $f_4$ 의 경우에는 전략 개수와 최적해의 성능이 대체적으로 무관하며, 전략 개수가 100개인 경우를 제외한 모든 전략 개수에 대하여 EWD가 우수함을 알 수 있다. 또한 10번 독립 시행에 대한 표준 편차를 통해 볼 때 두 방법 모두에서  $f_4$ 의 안정성이  $f_3$ 에 비하여 떨어지는데, 이것은  $f_4$ 가  $f_3$ 에 비하여 최적화하기 어려운 함수이기 때문이라고 추측할 수 있다.

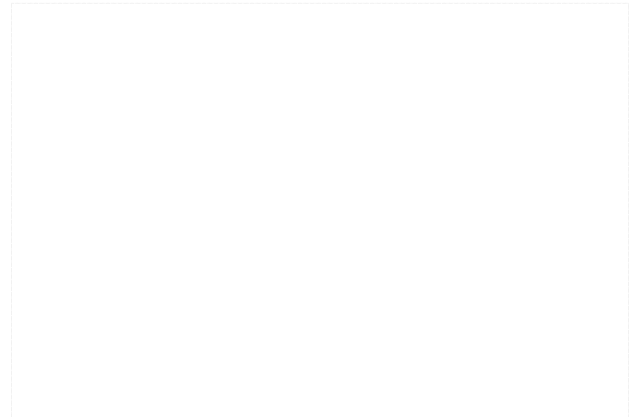


<Figure 5> The Average and Standard Deviation of Minimum Values Versus the Number of Strategies : EFD(■) and EWD(□). (a) The Result of Test Function  $f_3$ , (b) The Result of Test Function  $f_4$

<Figure 6>은  $f_4$ 와  $f_5$ 에 대하여 표본 개수에 따른 EWD와 EFD를 사용하여 구한 최적해 성능을 비교한 것이다. <Figure 6>을 통해 볼 때  $f_4$ 의 경우에는 모든 표본 개수에 대하여 EWD가 EFD보다 나은 결과를 나타냄을 알 수 있다. 또한 EWD는 표본 개수가 30보다 크면 표본 개수에 대체적으로 무관한데, 이것은 실험적으로 볼 때  $f_4$ 의 경우 최상 반응을 위한 최적의 표본 개수가 대략적으로 30개 정도임을 의미한다. 이와 대조적으로 EFD는 표본 개수가 증가할수록 더욱 향상된 결과를 나타내었다. 이러한 현상은  $f_5$ 에 대한 결과와 대조를 이룬다. 즉,  $f_5$ 의 경우에는 EFD가 EWD보다 성능 면에서 우수하며,

두 방법 모두 표본 개수에 비교적 영향을 받지 않음을 알 수 있다.  $f_3$ 에 대해서도  $f_5$ 와 유사한 결과를 얻었다.

위에서 분석한 표본 개수와 전략 개수에 따른 두가지 전략 결정 방법(EFD와 EWD)의 결과를 요약하면 다음과 같다. 첫째, 분리 가능한 함수의 경우에는 EFD가 EWD에 비하여 모든 조건(표본 개수 및 전략 개수)에 대하여 향상된 최적해와 안정적인 결과를 얻을 수 있는 방법이다. 둘째, 분리 가능하지 않는 함수의 경우에는 함수에 따라 다른 결과를 나타내었는데, 일반적으로 EFD는  $f_3$ 와  $f_5$ 에서 보다 우수한 성능을 보였으며 EWD는  $f_5$ 에 대하여 우수한 성능을 나타내었다. 셋째, 벤치마킹 함수에 무관하게 대체적으로 최적해와 안정성에는 높은 상관관계가 존재하였는데 우수한 최적해를 가지는 전략 결정 방법은 안정성도 높은 것을 알 수 있었다.



<Figure 6> The Average and Standard Deviation of Minimum Values Versus the Number of Samples for Test Function  $f_4$  (a) and  $f_5$  (b) with EFD(■) and EWD(□)

#### 4. 요약 및 결론

본 연구에서는 가상 경기 이론을 적용한 연속형 최적화 알고리즘에 필요한 전략 결정을 위한 새로운 방법을 제안하고 벤치마킹 함수에 적용하여 그 성능을 분석하였다. 가상 경기 이론을 연속형 최적화 문제에 적용하기 위해서는 연속적인 독립 변수 값을 이산적인 전략으로 표현해야 한다. 기존에 사용된 방법은 변수의 영역을 동일한 크기를 가지는 부분 영역으로 나누는 방법인 동일 너비 이산화(EWD) 방법을 사용했다. 본 연구에서는 선택된 실수 값들을 크기순으로 나열한 다음 동일한 개수의 값들이 각 부분 영역에 속하도록 부분 영역의 범위를 정하여 선택된 값의 빈도가 동일하도록 모든 부분 영역을 정하는 동일 빈도 이산화(EFD) 방법을 사용하였다.

제안한 방법의 성능을 기존 방법과 대비하여 분석하기 위하여 두 가지 유형(분리 가능한 함수와 분리가 가능하지 않는 함수)의 벤치마킹 함수에 적용하여 실험을 수행하였다. 분리 가능한 함수의 경우에는 표본 개수 및 전략 개수 등의 매개변수에 대하여 EFD가 EWD에 비하여 보다 나은 최적해 및 안정적인 결과를 얻을 수 있음을 보였다. 분리 가능하지 않는 함수의 경우에는 함수에 따라 다른 결과를 나타내었는데, 성능 측면에서는 EFD는  $f_3$ 와  $f_5$ 에서 보다 우수한 성능을 보였으며, 이와 대조적으로  $f_4$ 에 대하여는 열악한 성능을 보임을 알 수 있었다. 또한 벤치마킹 함수에 무관하게 대체적으로 최적해와 안정성 사이에는 높은 상관관계가 존재하였는데, 우수한 최적해를 가지는 이산화 방법은 안정성도 높은 것을 알 수 있었다.

본 연구는 가상 경기 이론을 연속형 최적화 알고리즘에 효율적으로 적용하기 위한 전략 선택 방법론에 관한 것으로 아래와 같은 향후 연구가 필요하다. 첫째, 연속형 변수를 이산적인 전략으로 변환하는 혼합형 전략 범위에 관한 연구이다. 본 연구를 통해 파악한 EFD와 EWD의 장단점을 기반으로 EFD와 EWD의 장점을 살린 보다 효율적인 전략 범위 선택에 관한 연구가 필요하다. 둘째, 본 논문에서 제안한 방법인 EFD에 대한 심도 깊은 성능 분석이다. 일반적인 연속형 최적화 알고리즘의 성능을 분석하기 위한 다양한 벤치마킹 문제들이 알려져 있으므로 이러한 문제들을 대상으로 제안한 방법을 적용하고, 실험 결과를 이론적으로 평가 분석하는 연구는 향후 효율적인 최적화 알고리즘을 창출하기 위해 필요한 분야라 할 수 있다[13]. 마지막으로 제안한 최적화 알고리즘을 실제 응용 분야에 적용하는 연구이다. 본 연구를 통해 알고리즘 측면에서 가상 경기 이론과 EFD를 적용한 최적화 방법론의 가능성을 보였기 때문에, 공학적 최적화 문제 해결에 본 연구에서 제안한 알고리즘을 적용함으로써 제안한 알고리즘의 효용성뿐만 아니라 알고리즘 자체를 더욱 개발시킬 것으로 판단된다. 따라서 앞으로 이러한 분야에서 연구가 더욱 진행되어야 할 것으로 생각된다.

## Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012-0007627).

## References

- [1] Back, T. and Schwefel, H.-P., An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1993, Vol. 1, p 1-23.
- [2] Brown, G., Iterative Solutions of Games by Fictitious Play, in *Activity Analysis of Production and Allocation*, T. Koopmans (Ed.), New York : Wiley, 1951.
- [3] Das, K. and Vyas, P., A suitability study of discretization methods for associative classifiers. *International Journal of Computer Applications*, 2010, Vol. 5, p 46-51.
- [4] Dougherty, J., Kohavi, R., and Sahami, M., Supervised and unsupervised discretization of continuous features, in *Machine Learning : Proceedings of the 12th International conference*, A. Prieditis and S. Russell eds, Morgan Kaufmann Publisher, San Francisco, CA, 1995.
- [5] Fogel, D., *Evolutionary Computation : Toward a New Philosophy of Machine Intelligence*. IEEE Press, NY, 1995.
- [6] Fudenberg, D. and Kreps, D., Learning mixed equilibria, *Games Econ. Behav.*, 1993, Vol. 5, p 320-367.
- [7] Fudenberg, D. and Levine, D., "The Theory of Learning in Games," Cambridge : MIT Press, 1998.
- [8] Garcia, A., Reaume, D., and Smith, R., Fictitious play for finding system optimal routings in dynamic traffic networks. *Transportation Research Part B : Methodological*, 2000, Vol. 34, p 147-156.
- [9] Garey, M. and Johnson, D., *Computers and Intractability : A Guide to the Theory of NP-Completeness*, W.H. Freeman, 1979.
- [10] Lambert, T., Epelman, M., and Smith, R., A fictitious play approach to large-scale optimization. *Operations Research*, 2005, Vol. 53, p 477-489.
- [11] Lee, C. and Yao, X., Evolutionary programming using mutations based on the Levy probability distribution. *IEEE Trans. Evol. Comput.*, 2004, Vol. 8, p 1-13.
- [12] Lee, C., A study on a continuous optimization algorithm based on the fictitious play theory. *Journal of KIISE : Software and Applications*, 2012, Vol. 39, No. 10, p 787-795.
- [13] Lee, D. and Lee, C.-Y., Simulated Annealing Algorithm Using Cauchy-Gaussian Probability Distributions. *Journal of the society of Korea industrial and systems engineering*, 2010, Vol. 33, p 130-136.
- [14] Long, C., Zhang, Q., Li, B., Yang, H., and Guan, X., Non-Cooperative Power Control for Wireless Ad Hoc Networks with Repeated Games. *IEEE Journal of selected areas in communications*, 2007, Vol. 25, p 1101-1112.

[1] Back, T. and Schwefel, H.-P., An overview of evolutionary



- [15] Mitchell, M., An Introduction to Genetic Algorithms. Cambridge, MA : MIT Press, 1996.
- [16] Monderer, D. and Shapley, L., Fictitious play property for games with identical interests. *Journal of Economic Theory*, 1996, Vol. 68, p 258-265.
- [17] Nash, J., Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 1950, Vol. 36, p 48-49.
- [18] Nash, J., Non-Cooperative Games. *The Annals of Mathematics*, 1951, Vol. 54, p 286-295.
- [19] Rezek, I., Leslie, D., Reece, S., Roberts, S., Rogers, A., Dash, R., and Jennings, N., On similarities between Inference in Game Theory and Machine Learning, *Journal of Artificial Intelligence Research*, 2008, Vol. 33, p 259-283.
- [20] Schaffer, J., Caruana, R., Eshelman, L., and Das, R., *A study of control parameters affecting online performance of genetic algorithms for function optimization*, in Proc. of the 3rd Int'l Conf. on GAs, edited by J. Schaffer. Morgan Kaufman, San Francisco, 1989, p 51-60.
- [21] Talbi, E.-G., *Metaheuristics : from design to implementation* (Wiley Series on Parallel and Distributed Computing) Wiley, Hoboken, 2009.