

이항 반응 실험의 확률적 전역최적화 기법연구

이동훈[†] · 황근철 · 이상일 · 윤원영

A Study on the Stochastic Optimization of Binary-response Experimentation

Donghoon Lee[†] · Kun-Chul Hwang · Sangil Lee · Won Young Yun

ABSTRACT

The purpose of this paper is to review global stochastic optimization algorithms(GSOA) in case binary response experimentation is used and to compare the performances of them. GSOAs utilise estimator of probability of success \hat{p} instead of population probability of success p , since p is unknown and only known by its estimator which has stochastic characteristics. Hill climbing algorithm, simple random search, random search with random restart, random optimization, simulated annealing and particle swarm algorithm as a population based algorithm are considered as global stochastic optimization algorithms. For the purpose of comparing the algorithms, two types of test functions(one is simple uni-modal the other is complex multi-modal) are proposed and Monte Carlo simulation study is done to measure the performances of the algorithms. All algorithms show similar performances for simple test function. Less greedy algorithms such as Random optimization with Random Restart and Simulated Annealing, Particle Swarm Optimization(PSO) based on population show much better performances for complex multi-modal function.

Key words : Stochastic Global Optimization, Binary-response experimentation, Hill Climbing Algorithm, Simulated Annealing, Particle Swarm Optimization

요약

본 논문의 목적은 이항출력 실험을 이용할 경우에 확률적 전역 최적화 방법론들을 검토하고 알고리즘들간의 성능을 비교하기 위한 것이다. 모 성공확률은 알수 없고 확률적 특성을 갖기 때문에 확률적 전역 최적화 방법론에서는 모 성공확률 대신 성공확률의 추정치를 이용한다. 언덕오르기 알고리즘, 단순랜덤탐색, 랜덤재출발 랜덤탐색, 랜덤 최적화, 담금질 기법 및 군집기반의 알고리즘인 입자 군집 최적화 알고리즘을 확률적 전역 최적화 알고리즘으로 사용하였다. 알고리즘의 비교를 위하여 두가지 테스트 함수(하나는 단봉이고 나머지는 다봉임)가 제안되었고 몬테카를로 시뮬레이션을 이용하여 알고리즘의 성능을 평가하였다. 단순 테스트 함수에 대하여는 모든 알고리즘이 유사한 성능을 보이고 있다. 복잡한 다봉의 테스트 함수에 대하여는 랜덤재출발 랜덤최적화, 담금질 기법과 군집 기반의 입자군집 알고리즘이 훨씬 더 좋은 성능을 보임을 알 수 있다.

주요어 : 확률적 전역 최적화, 이항 반응 실험, 언덕오르기 알고리즘, 담금질 기법, 입자 군집 최적화

1. 서론

유도무기체계를 이용한 표적에 대한 최적공격방법을

도출하거나 유도무기의 공격을 방어하기 위한 최적방어 방법을 도출하는 실험은 많은 경우 무기체계의 정확성이나 운용전술상의 불확실성 등의 확률적 요소를 포함하고 있어서 결정론적으로 모델링하면 문제를 외곽할 수 있다. 따라서 잡음환경을 바탕으로 실험이 분석되어야 하며 최적화의 문제도 실험을 통한 성능이나 효과도의 출력을 최대화시키는 확률적 전역 최적화문제로 정형화하여야 한다. 이때의 목적함수는 격추소요시간, 최근접거리와 같

Received: 2 December 2022, **Revised:** 27 February 2023,
Accepted: 27 February 2023

[†] **Corresponding Author:** Donghoon Lee
E-mail: dhlee@add.re.kr
Agency for Defence Development &
Pusan National University

은 연속변수값 또는 명중 횟수, 소요수량 같은 이산변수 더 나아가 명중/불명중, 격추/생존과 승/패와 같은 이항값으로 측정될 수 있다.

잡음이 유발한 목적함수의 반응표면(response surface)의 동적인 변이로 인하여 목적함수는 동일한 시행 솔루션(trial solution)을 평가할 때마다 다른 값을 낸다. 이런 잡음환경에서는 뛰어난 성능의 시행 솔루션이 적합도 추정치가 열세로 보임으로 인해 다음 세대에 살아남지 못할 수 있으며 반대로 근본적으로 열세인 솔루션이 기만적으로 높은 적합도 추정으로 인해 살아남을 수 있는 문제가 있다(Akimoto et al(2015), Bäck&Hammel(1994)). 일반적으로 이러한 불확실성을 보완하고 추정오차를 줄이기 위해서는 많은 수의 값비싼 실험(시간과 수행비용 측면을 고려)을 수행하여야 한다.

이러한 문제를 해결하기 위하여 잡음환경에 적합한 진화적 알고리즘을 비롯한 여러가지 전역 최적화 알고리즘이 연구되어 왔다(Bäck T.(1996)). 목적함수를 구성하는 측정(measurement)과 미래상황의 추정(estimation)이 잡음에 의해서 오염되어 있거나 근원적으로 잡음을 가진 경우의 잡음최적화(noisy optimization) 문제는 원천적으로 불연속이기 때문에 전통적인 미분기반의 최적화 기법은 적용하기 어려워 Alarie et al(2021)이 제안한 블랙박스 최적화(black-box optimization)라고 하는 미분을 적용하지 않는 전역최적화(derivative free optimization) 방법을 이용한다. 본 논문에서는 다수의 이항 반응 실험을 수행하여 추정되는 적합도를 이용하는 경우에 확률적 전역 최적화문제를 풀기 위한 기존의 방법들을 검토하고 몬테카를로 시뮬레이션을 통하여 실험 반복횟수 등을 비롯한 설정 파라미터의 영향을 분석하고 방법들 간의 비교를 수행하고자 한다.

본 논문의 구성으로는 2장에서는 다양한 형태의 확률적 최적화 방법에 대하여 기술하고 3장에서는 테스트 함수를 정의하고 몬테카를로 시뮬레이션을 통하여 각 방법들의 설정 파라미터의 영향을 분석하고 방법들간의 성능을 비교하였다. 4장에서는 결론 및 향후 연구 방향에 대하여 제시한다.

2. 확률적 전역 최적화 알고리즘

본 장에서는 이항 반응 실험을 이용한 성공확률의 특성을 검토하고 이항 출력 시뮬레이션의 확률적 전역 최적화에 적용할 일반적인 확률적 전역 최적화 방법을 종합적으로 검토하고 특성을 분석하고자 한다.

2.1 이항 반응 확률적 실험을 이용한 성공확률의 특성

우리의 목적은 다수의 베르누이 시행(Bernoulli trial)인 이항출력 실험을 수행하여 알지 못하는 솔루션 x 에서의 모 성공확률(population success probability) f_x 를 추정하여 f_x 를 최대화하는 전역 최적 솔루션(global optimal solution)을 구하는 것이다. 목적함수인 모 성공확률 f_x 는 몬테카를로 실험에서 추정될 수 있으며 실험의 결과로부터 시행 솔루션 x 에서의 시행횟수 n_x 에 대한 성공횟수 m_x 와 관련된 정보는 식 (1)과 같이 주어지고 성공확률 추정치의 분포는 식 (2)와 같으며 두 솔루션간의 차이의 분포는 식 (3)과 같이 주어진다.

$$m_x \sim B(n_x, f_x) \tag{1}$$

$$p_x = \frac{m_x}{n_x} \approx N\left(f_x, \frac{f_x(1-f_x)}{n_x}\right) \tag{2}$$

$$p_x - p_y = \frac{m_x}{n_x} - \frac{m_y}{n_y} \tag{3}$$

$$\approx N\left(f_x - f_y, \frac{f_x(1-f_x)}{n_x} + \frac{f_y(1-f_y)}{n_y}\right)$$

두 확률의 차이의 분포는 식 (4)와 같은 t-분포에 의하여 근사시킬 수 있다.

$$p\{p_x > p_y\} = p\{p_x - p_y > 0\} \tag{4}$$

$$\approx t_v\left(\frac{\mu_d}{s_d}\right) = t_v\left(\frac{p_x - p_y}{\sqrt{s_x^2 + s_y^2}}\right)$$

여기서, $s_x^2 = \frac{p_x(1-p_x)}{n_x}$ 이며 자유도는 $v \equiv \frac{(s_x^2 + s_y^2)}{\frac{s_x^2}{n_x - 1} + \frac{s_y^2}{n_y - 1}}$

와 같다.

2.2 알고리즘 리뷰

2.2.1 등반 알고리즘(Hill Climbing Algorithm)

등반 알고리즘은 매우 오래된 단일 목적함수에 대한 탐색과 최적화를 위한 알고리즘이다(Russel & Novic(2002)). 등반 알고리즘은 현재까지의 최적해를 이용하여 하나의 후계자(successor)를 생성하고 평가하여 기존해를 대체하는 과정을 반복수행한다. 새로운 개체가 부모 개체보다 나은 경우 최적해는 자손에 의해서 대체된다. 이런 의미에서 등반 알고리즘은 개체군의 크기가 1인 경우의 진화 알고리즘의 일종으로 볼 수 있다. 등반 알고리즘

은 모수없는 탐색연산자를 이용하여 첫 번째 후보해를 만들고 그로부터 후계자 생성자를 이용하여 자손을 만든다.

Table 1은 등반 알고리즘의 일반적인 흐름을 보여준다. 등반 알고리즘의 주요 문제점은 국부 피크치로의 조기수렴인데 본 알고리즘은 알고 있는 후보해를 이용하여 문제 공간 내의 새로운 점을 후계자 생성자 $x \leftarrow successor(x^*)$ 를 이용하여 찾는다.

Table 1. Hill Climbing Algorithm

Algorithm $(x^*, p^*) \leftarrow hillClimber(f)$
<pre> generate random candidate x^* estimate fitness $p^* \leftarrow p(x^*)$ while terminationCriterion() do choose individual x by $x \leftarrow successor(x^*)$ estimate fitness of x by $p_x \leftarrow p(x)$ if $p_x < p^*$ then $[x, p^*] = [x, p_x]$ return (x^*, p^*) </pre>

기존의 점 x^* 를 중심으로 새로운 점을 찾는 후계자 생성자 $x \leftarrow successor(x^*)$ 를 정의함에 따라 다양한 알고리즘이 가능하다.

확률적 언덕오르기의 극단적인 방법으로 순수 랜덤 탐색(PRS: pure random search)(Brooks(1958))는 전역 균일 랜덤 탐색(Global Uniform Random Search) 또는 맹목 탐색(blind search)라고도 하며 현 솔루션과 독립적으로 다음 솔루션을 전역에서 균일한 확률로 구하는 방법으로 후계자 생성자는 식 (5)와 같다.

$$successor(x^*) = \begin{pmatrix} uniform_1(l_1, u_1) \\ uniform_2(l_2, u_2) \\ \dots \\ uniform_n(l_n, u_n) \end{pmatrix} \quad (5)$$

랜덤 재출발을 이용하는 등반 알고리즘은 수렴의 조기성숙을 억제하기 위한 또 다른 방법으로 Hill Climbing with Random Restart: HCRR)이 있다. 이 알고리즘의 의사코드는 Table 2와 같다. 탐색공간에서 출발점을 다양하게 발생시킴으로써 등반 알고리즘의 국부 최적치에 조기성숙하는 문제를 해결할 수 있다. 후계자 생성자 식 (5)를 적용할 경우 랜덤재출발 랜덤 최적화(random optimization with random restart: RORR)이라고 명명한다.

Table 2. Hill Climbing with Random Restart

Algorithm $(x_g^*, p_g^*) \leftarrow hillClimberRR(f)$
<pre> generate random global best individual x_g^* estimate fitness of global best ind. $p_g^* = p(x_g^*)$ while terminationCriterion() do $(x^*, p^*) \leftarrow hillClimber(f)$ if $p^* < p_g^*$ then $(x_g^*, p_g^*) = (x^*, p^*)$ return (x_g^*, p_g^*) </pre>

확률적 언덕오르기의 일종인 랜덤 최적화(Random Optimization: RO)는 (Rastrigin(1963))에 의해서 처음으로 제안되었다. 이후에 Gurin&Leonard(1965), Schumer M. A(1967), Schumer&Steiglitz(1968)et al에 의하여 많은 개선이 이루어졌다. 또 다른 RO 알고리즘이 Matyas (1965)에 의하여 유사한 시기에 제안하였고 단봉 문제에서 수렴문제에 대하여 정리했으며 목적함수가 다봉일 경우에도 RO가 최적해를 찾을 수 있음을 이론적으로 증명했다.

등반 알고리즘이 새로운 후보해가 기존 후보해와 매우 인접한 결정론적 위치에 생성되어지는 반면 랜덤 탐색은 기존 후보해를 중심으로 확률적으로만 결정된다. 랜덤 탐색은 진화 전략에서의 돌연변이와 유사하게 새로운 탐색 연산자를 적용한다. 식 (6)와 같은 후계자 생성자는 탐색 공간 $G = R^n$ 에서의 모든 점들은 특정 다른 점들로부터 확률적으로 한번에 도달할 수 있는 특성을 가진다.

$$successor(x^*) = x^* + \begin{pmatrix} random_1(\mu_1, \sigma_1^2) \\ random_2(\mu_2, \sigma_2^2) \\ \dots \\ random_n(\mu_n, \sigma_n^2) \end{pmatrix} \quad (6)$$

이 수식은 지금까지 발견된 최적해 후보의 유전형질에 $r \in R^n$ 이 더해진다. 각 구성요소 $r[i]$ 는 μ_i 주변에 분포한다. μ_i 는 일반적인 탐색방향을 정의한다. μ_i 와 σ_i^2 가 작을 경우 국부 최적치를 찾는 반면 클 경우 탐색공간을 대략적으로 탐색한다. $\mu_i = 0, \forall i$ 일 경우 현재의 유전형질을 중심으로 분포가 적용된다. 정규분포는 $-\infty$ 에서 ∞ 까지 발생이 가능하므로 분산 σ^2 가 0보다 클 경우 국부 최적에서 벗어날 가능성이 존재하게 해준다. 정규분포 대신에 다른 분포를 이용할 수 있으나 종형상의 분포가 좋은 특성을 갖는 것을 보였다(Worakul et al(2002)).

μ 와 σ 값의 적절한 설정으로 수렴속도를 많이 향상시

킬 수 있다. Li와 Rhinehart에 의해서 소개된 휴리스틱 랜덤 탐색과 Chandran&Rhinehart(2002)의 랜덤 최적화는 경사정보와 강화학습을 이용해서 갱신한다.

2.2.2 담금질 기법(Simulated Annealing: SA)

Metropolis et. al.(1953)는 담금질 과정 속에서 균형에 이르는 방법이 모의하는 몬테카를로 방법을 고안하였다. 이는 Kirkpatrick et al(1983)에 영감을 주어 1980년 초에 전역 최적화를 위한 담금질 기법을 개발하여 다양한 조합 최적화 문제에 적용하였다. 담금질 기법은 임의의 탐색 및 문제 공간에 적용될 수 있는 최적화 방법이다.

SA는 Table 3과 같은 흐름을 갖는다. 단순 등반 알고리즘과 같이 출발점으로 하나의 초기 개체와 후계자 생성자 $x \leftarrow successor(x^*)$ 를 RO에서 적용된 식 (5)와 동일한 것을 이용한다. x 가 연속변수일 경우 식 (7)과 같은 함수를 적용하여 $\Delta E < 0$ 일 경우 최적치를 갱신하는 것을 기본적으로 적용하나 그렇지 않을 경우에도 수식의 선택확률에 따라서 최적치를 갱신함으로써 전역 최적화 알고리즘으로 적용이 가능하다.

$$P(x, x^*, T) = \begin{cases} \exp\left\{-\frac{\Delta E}{T}\right\} & \text{if } \Delta E > 0 \\ 1 & \text{otherwise} \end{cases} \quad (7)$$

여기서, $\Delta E = E(x) - E(x^*)$

$T \leftarrow getTemperature(t)$ 함수는 초기에는 T 를 큰 수를 선택하여 덜 탐욕적으로 전역을 탐색하다가 시간 t 가 지나감에 따라서 작은 $T = T_{init} \times \alpha^t$, $\alpha < 1$ 를 선택($T = 0$ 일 경우 등반 알고리즘과 동일)하여 등반 알고리즘에 가까워지도록 설계되어 있다.

Table 3. Simulated Annealing Algorithm

Algorithm $(x_g^*, p^g) \leftarrow simulatedAnnealing(f)$
T_{init} : initial temperature q_a : acceptance parameter
generate random candidate x_g^* $T \leftarrow T_{init}$ while terminationCriterion() do $T \leftarrow getTemperature(t)$ $x \leftarrow successor(x_g^*)$ $\Delta E = E(x) - E(x_g^*)$ if $\Delta E < 0$: $x^* \leftarrow x$ elif $P(E(x), E(x_g^*), T) \geq uniform(0, 1)$: $x_g^* \leftarrow x$ return (x_g^*, p^g)

2.2.3 입자 군집 최적화(Particle Swarm Optimization: PSO)

PSO는 조류의 무리 또는 어뢰의 무리와 같은 생물학적 사회 시스템의 행태를 모의하는 군집지능(swarm intelligence)의 한 형태로서 1955년 Eberhart & Kennedy (1995)가 개발하였고 Venter & Sobieszcanski-Sobieski (2003), Cai et al(2004), Gao & Duan과 Gao & Ren (2007) et al에 의해서 논의되고 개선되고 정제되어왔다. 군집이 먹이를 찾을 때 각 개체는 퍼져서 독립적으로 먹이를 찾아 돌아다닌다. 각 개체는 운동에 일정 자유도를 가지거나 불확실성을 갖는다. 조만간 이들 중 바람직한 것을 찾게 되고 사회성으로 인해 정보가 다른 개체에 통보되어 먹이로 향하게 된다.

PSO 알고리즘의 흐름은 Table 4와 같다. 입자들의 갱신은 입자들의 방향 v_i 에 의하여 결정된다. 입자들의 방향은 관성(inertia) w , 인지력(cognitive force) c_1 , 사회적(social force) c_2 에 따라서 계산된다. 개체들의 최적점 x_i^* 는 등반 알고리즘과 같이 탐욕적인 방법으로 갱신된다. 전역최적점 g 는 모든입자들 중에서 선택된다.

Table 4. Particle Swarm Optimization Algorithm

Algorithm $(x_g^*, p^g) \leftarrow ParticleSwarmOptimizer(f)$
v_i^* : new particle velocity of i^{th} member v_i : current particle velocity of i^{th} member g : best swarm position w : constant inertia weight c_1, c_2 : cognitive and social parameters respectively r_1, r_2 : random numbers between 0 and 1
for i in range(n_{pop}): # pinitial fitness update $x_i^* \leftarrow generateRandomParticle()$ $p_i^* = p(x_i^*)$ while terminationCriterion() do for i in range(n_{pop}): $p_i = p(x_i)$ if $p_i < p_i^*$: $p_i^* = p(x_i)$, $x_i^* = x_i$ if $p_i < p^g$: $p^g = p(x_i)$, $x_g^* = x_i$ for i in range(n_{pop}): $v_i \leftarrow wv_i + c_1r_1(x_i - x_i^*) + c_2r_2(x_g^* - x_i)$ $x_i \leftarrow successor(x_i^*) = x_i^* + v_i$ return (x_g^*, p^g)

3. 성능비교

3.1 테스트 함수

실제 상황에서는 목적함수인 성공확률 형태는 알 수

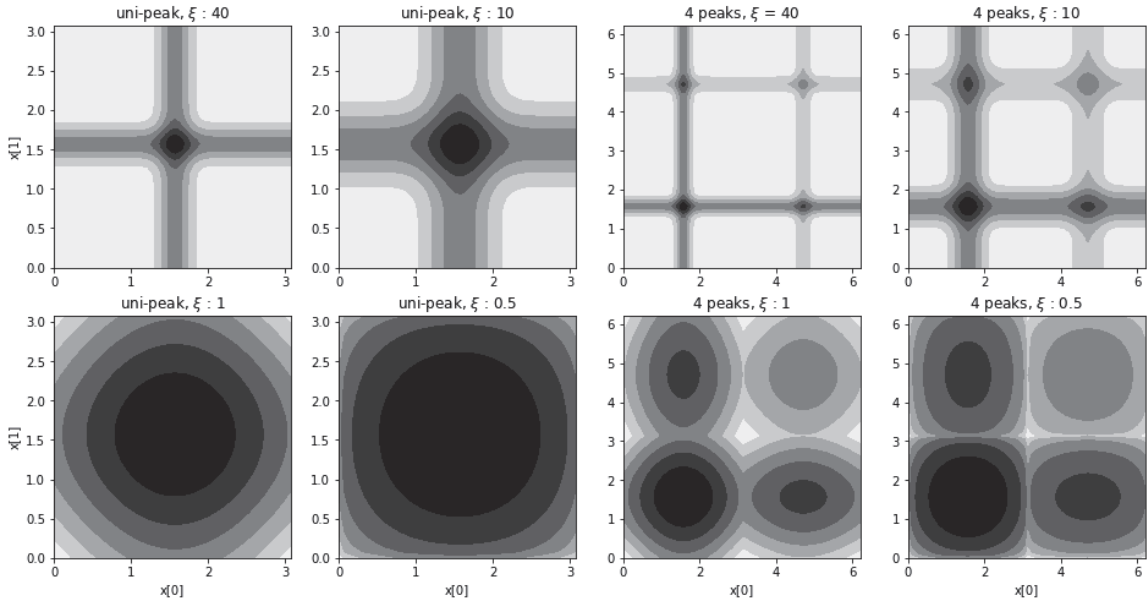


Fig. 1. Shape of test functions of 2 dimension

없는 전형적인 블랙박스 문제로 성공확률의 추정치의 분산은 식 (2)와 같이 반복횟수에 의존한다. 성공확률함수의 추정에 기초하여 최적 해를 찾는 것은 시간과 비용 측면에서 매우 비 효율적인 작업이다. 최적화 알고리즘의 성능을 평가하기 위하여 식 (8)과 식 (9)의 두가지 테스트 함수를 고려한다. 식 (8)은 $\left[\frac{\pi}{2}, i=1,2,\dots,\nu\right]$ 에서 1개의 국부 최소치를 갖는 단봉 테스트 함수이고 식 (9)는 k' 개의 국부 최소치를 가지며 x 가 $\left[\frac{\pi}{2}, i=1,2,\dots,\nu\right]$ 에서 최소 확률이 $1-\theta$ 인 테스트 함수를 선택하였다.

$$f_x^u = 1 - \theta \times \left[\sum_{i=1}^{\nu} L |\sin(x_i)| / \nu \right]^{\xi}, 0 < x_i < \pi \quad (8)$$

$$f_x^m = 1 - \theta \times \left[\sum_{i=1}^{\nu} L |\sin(x_i)| / \nu \right]^{\xi}, 0 < x_i < 2\pi \quad (9)$$

여기서 $L = \begin{cases} 1 & \text{if } x_i < \pi \\ 1/2 & \text{otherwise} \end{cases}$, ν 는 변수의 개수, ξ 는 성공확률 함수의 첨도(뾰족한 정도의 척도)를 결정하는 파라미터이다.

Fig 1은 $k=2$ 일 경우의 단봉 테스트 함수 f_x^u 와 다봉 테스트 함수 f_x^m 의 형상을 보여준다. ξ 가 클 경우 피크 부분의 영역의 크기가 달라짐을 알 수 있으며 다봉일 경우

변수의 개수에 따라서 봉의 개수가 기하급수적으로 늘어날 수 있다. 본 논문에서는 변수의 개수 ν 가 6이고 ξ 를 2인 경우에 대하여 단봉인 경우와 다봉인 경우의 테스트 함수를 정의하고 비교할 것이다.

3.2 성능척도(measure of performance)

추정된 최적해 x_g^* 와 추정 최적 성공확률 p^g 은 각각 식 (10)와 (11)과 같이 실험 결과로부터 계산된다.

$$x_g^* = x' \text{ s.t. } p_{x'} \geq p_x \quad \forall x \quad (10)$$

$$f_g = f(x_g^*) \quad (11)$$

식 (1)과 (2)에서 보듯이 이항분포의 추정최적성공확률 p_{opt} 은 모 성공확률 값보다 과대 추정되는 경향을 보인다. 따라서 알고리즘의 평가는 추정된 최적해의 추정 최적 성공확률이 아닌 모 성공확률 식 (11)에 추정된 최적해 x_g^* 를 넣어서 나오는 값으로 평가되어야 한다. 최적해의 공정한 평가를 위해서는 최적해를 도출하는 데에 필요한 총 실험 횟수 N_{TOT} 에 따른 평가가 필요하다.

3.3 비교방법

대상 알고리즘은 등반알고리즘 중에서 순수 랜덤 탐

색(PRS), 랜덤 최적화(RO), 랜덤재출발 랜덤 최적화(RORR)와 담금질기법(SSA), 입자 군집 최적화(PSO)를 선택하였다. 비교 대상 알고리즘별 설정 파라미터는 Table 5와 같다. 비교대상 테스트 함수는 단봉인 식 (11)의 f_x^u 과 다봉인 (12)의 f_x^m 와 같으며 추정 파라미터의 개수 ν 는 6, θ 는 1, ξ 는 2일 경우를 대상으로 비교하였다. 공정한 비교를 위해서 알고리즘 시행별 총 실험 횟수 (N_{TOT})를 1,000,000으로 제한하였고 모든 알고리즘에 공통적으로 솔루션 당 시행횟수 n_x 는 50, 100, 400, 600, 800, 1000, 2000, 4000, 10000의 조합에 대하여 적용하였으며 기타 파라미터는 Table 5에서 보는 바와 같이 알고리즘 별로 상이하게 적용된다. 성능 평가를 위하여 동일한 조건의 알고리즘을 100회 시행하여 최적값의 평균을 비교하였다.

Table 5. control parameters for each algorithm

algorithm	control parameters	# of cases
PRS	n_x	1
RO	n_x, σ	2
RORR	$n_x, \sigma, n_{restart}$	3
SA	$n_x, K_B\sigma, T_{init}$	3
PSO	n_x, w, c_1, c_2	4

3.4 몬테카를로 시뮬레이션 결과 및 고찰

3.4.1 단순랜덤탐색(pure random search: PRS)

PRS은 통제 대상 파라미터는 솔루션 당 시행횟수 n_x 의 조합에 대하여 몬테 칼로 실험을 수행하였다. n_x 에 따른 최적값의 추세는 Fig 3과 같다. 대체로 n_x 이 커지면 성능이 좋아지며 테스트 함수가 f_x^u 일 경우 f_x^m 경우보다 0.12 정도 낮은 실패확률을 가짐을 볼 수 있다.

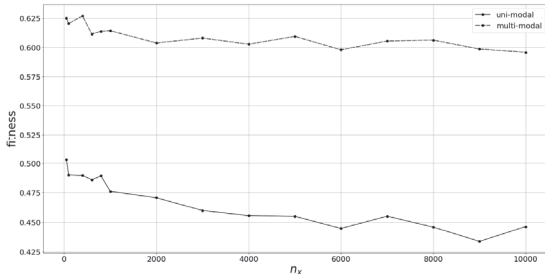


Fig. 2. Results of PRS simulation for test functions

3.4.2 랜덤최적화(random optimization: RO)

RO는 시행횟수 n_x 와 후계자 생성자의 표준편차 σ 에 의하여 탐험(exploration)과 이용(exploitation)을 통제하는 것이 가능하며 몬테카를로 시뮬레이션을 위한 두 가지 파라미터의 조합은 Table 6과 같다.

Table 6. simulation cases for RO

parms	cases	# of cases
n_x	50,100,400,1000,2000,4000,10000	7
σ	1,2,3,4,5,10,20,30,...,100,200,400	17
total cases		119

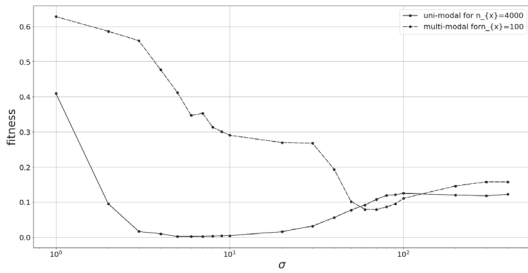
Table 7은 RO 알고리즘의 몬테카를로 시뮬레이션 결과로 도출된 최적 파라미터 조합과 성공확률 값을 보여주고 있다.

단봉일 경우 $\sigma=5$ 에서 다봉일 경우 $\sigma=70$ 에서 각각 최적 성공확률을 보이고 있다. 테스트 함수가 단봉일 경우 다봉일 경우에 탐험에 비중을 많이 두기 때문에 높은 분산의 후계자 생성이 선호되는 것으로 파악된다.

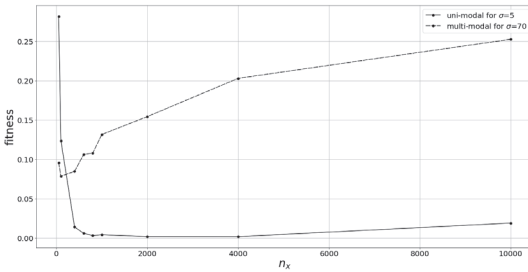
Table 7. optimal parameter combination for RO

parms	test function	
	f_x^u	f_x^m
n_x	4000	100
σ	5	70
$1-p^g$	0.0017	0.0786

Fig 4는 RO의 몬테카를로 시뮬레이션 수행결과 중 최적값을 중심으로 변수별 민감도를 보여주는 결과이다. (a)는 σ 에 따른 영향을 보여주는데 f_x^u 의 경우 σ 가 50에서 100 정도에서 대체로 최적에 가까운 결과를 보이고 있다. f_x^m 의 경우 σ 가 60 및 70에서 최적의 결과를 보이고 있음을 확인할 수 있다. 대체로 테스트 함수가 f_x^u 와 f_x^m 에 따라서 상이한 결과를 보이고 있음을 알 수 있어 테스트 함수의 형태를 고려한 알고리즘의 선정이 필요함을 알 수 있다.



(a) fitness vs σ



(b) fitness vs n_x

Fig. 3. Results of RO simulation for test functions

3.4.3 랜덤 재출발 랜덤최적화(random optimization with random restart:RORR)

RORR은 시행횟수 n_x 와 이웃 생성자의 표준편차 σ 에 의하여 탐험(exploration)과 이용(exploitation)을 통제하는 것이 가능하며 몬테카를로 시뮬레이션을 위한 두가지 파라미터의 조합은 Table 8과 같다.

Table 8. simulation cases for RORR

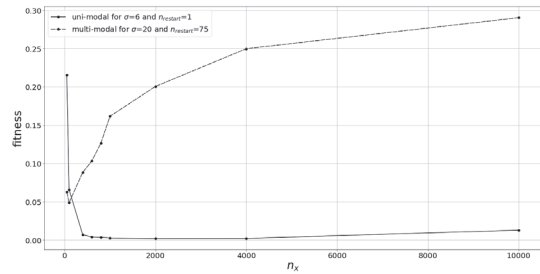
parms	cases	# of cases
n_x	50,100,400,1000,2000,4000,10000	7
σ	1,2,4,6,8,10,15,20,30,50,70,100,200,400	14
$n_{restart}$	1,25,50,75,100,200,300,400,500	9
total cases		882

Table 9는 RORR 알고리즘의 몬테카를로 시뮬레이션 결과로 도출된 최적 파라미터 조합과 성공확률 값을 보여주고 있다. 단봉 테스트 함수 f_x^u 에 대하여는 f_x^m 에 대한 랜덤재출발 랜덤최적화는 $n_{restart}=1$ 이고 $\sigma=6$ 이고 $n_x=2000$ 이 최적으로 선택되어 RO의 결과와 거의 일치함을 확인할 수 있다. 다봉의 경우 $n_{restart}=75$ 가 선정되고 $\sigma=20$, $n_x=2000$ 으로 선정되어 전역적인 탐색에 비중을

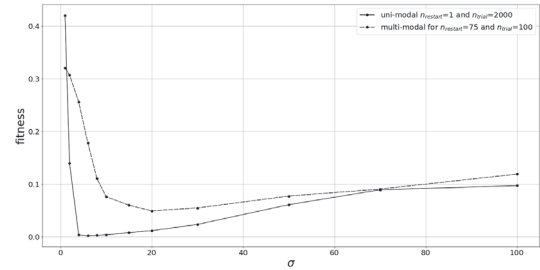
두는 전략이 필요함을 알 수 있다.

Table 9. optimal parameter combination for RORR

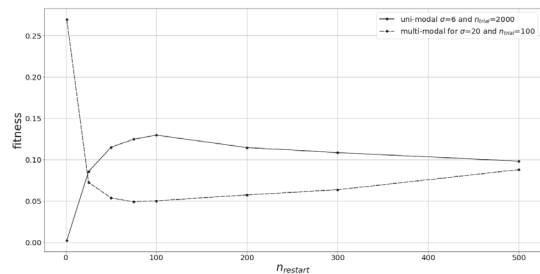
parms	test function	
	f_x^u	f_x^m
n_x	2000	100
σ	6	20
$n_{restart}$	1	75
$1-p^j$	0.0020	0.0489



(a) fitness vs n_x



(b) fitness vs σ



(c) fitness vs $n_{restart}$

Fig. 4. Results of RORR simulation for test functions

Fig 5는 RORR의 몬테카를로 시뮬레이션 수행결과 중 최적값을 중심으로 변수별 민감도를 보여주는 결과이다. (a)는 σ 에 따른 영향을 보여주는데 f_x^u 의 경우 σ 가 50에

서 100 정도에서 대체로 최적에 가까운 결과를 보이고 있다. f_x^m 의 경우 σ 가 60과 70에서 최적의 결과를 보이고 있음을 확인할 수 있다. (b)의 경우 최적 σ 에 이르기까지 급격히 감소하다가 완만하게 증가하는 패턴을 보인다. (c)를 보면 f_x^u 에 대하여는 $n_{restart}=1$ 에서 최적이고 이후 급격히 증가하다가 100 부근부터 완만하게 감소하고 있음을 알 수 있고 f_x^m 에 대하여는 90 부근에 이르기까지 급격히 감소하다가 이후 완만하게 증가함을 알 수 있다. 대체로 테스트 함수가 f_x^u 와 f_x^m 에 따라서 상이한 결과를 보이고 있음을 알 수 있어 테스트 함수의 형태를 고려한 알고리즘의 선정이 필요함을 알 수 있다.

3.4.4 담금질 기법(simulated annealing: SA)

담금질 기법은 n_x 와 σ 에 T_{init} 를 추가함으로써 탐험(exploration)과 이용(exploitation)을 통제하는 것이 가능하며 몬테카를로 시뮬레이션을 위한 세가지 파라미터의 조합은 Table 10과 같다.

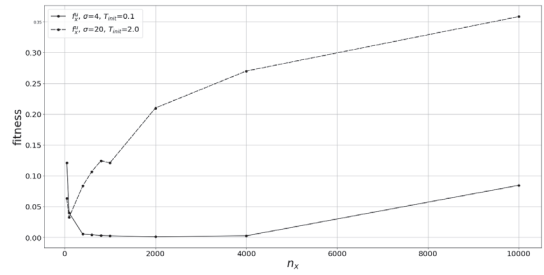
Table 10. simulation cases for SA

parm	cases	# of cases
n_x	50,100,400,1000,2000,4000,10000	7
σ	1,2,3,4,5,10,20,30,...,100,200,400	7
T_{init}	0.001, 0.1, 1,2,3,5,10	7
total cases		343

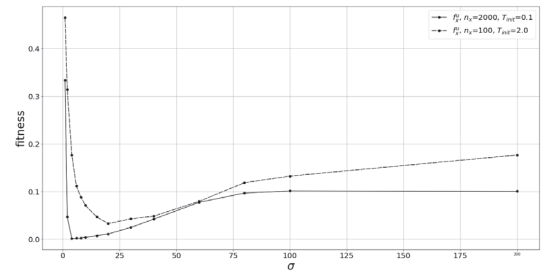
Table 11은 SA 알고리즘의 몬테카를로 시뮬레이션 결과로 도출된 최적 파라미터 조합과 성공확률 값을 보여주고 있다. 단봉일 경우 T_{init} 이 0.1이고 σ 가 4로 선택되었고 다봉일 경우 T_{init} 이 2.0이고 σ 가 20로 선택되었는데 단봉일 경우는 탐색능력보다 탐욕적인 방법을 선호하고 있으며 다봉일 경우는 탐색능력이 중요하기 때문으로 보인다.

Table 11. optimal parameter combination for SA

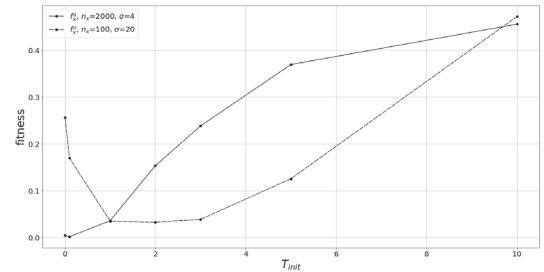
parms	test function	
	f_x^u	f_x^m
n_x	2000	100
σ	4	20
T_{init}	0.1	2.0
$1-p^g$	0.0013	0.0326



(a) fitness vs n_x



(b) fitness vs σ



(c) fitness vs T_{init}

Fig. 5. Results of SA simulation for test functions

Fig 6는 SA의 몬테카를로 시뮬레이션 수행결과 중 최적값을 중심으로 변수별 민감도를 보여주는 결과이다. (a)는 n_x 에 따른 영향을 보여주는데 f_x^u 의 경우 σ 가 50에서 100 정도에서 대체로 최적에 가까운 결과를 보이고 있다. f_x^m 의 경우 σ 가 60과 70에서 최적의 결과를 보이고 있음을 확인할 수 있다. (b)의 경우 최적 σ 에 이르기까지 급격히 감소하다가 완만하게 증가하는 패턴을 보인다. (c)를 보면 f_x^u 에 대하여는 $n_{restart}=1$ 에서 최적이고 이후 급격히 증가하다가 100 부근부터 완만하게 감소하고 있음을 알 수 있고 f_x^m 에 대하여는 90 부근에 이르기까지 급격히 감소하다가 이후 완만하게 증가함을 알 수 있다. 대체로 테스트 함수가 f_x^u 와 f_x^m 에 따라서 상이한

결과를 보이고 있음을 알 수 있어 테스트 함수의 형태를 고려한 알고리즘의 선정이 필요함을 알 수 있다.

3.4.5 입자 군집 최적화

입자 군집 최적화는 n_x 에 ω 와 c_1 과 c_2 를 추가함으로써 탐험(exploration)과 이용(exploitation)을 통제하는 것이 가능하며 몬테카를로 시뮬레이션을 위한 네가지 파라미터의 조합은 Table 12와 같다.

Table 12. simulation cases for PSO

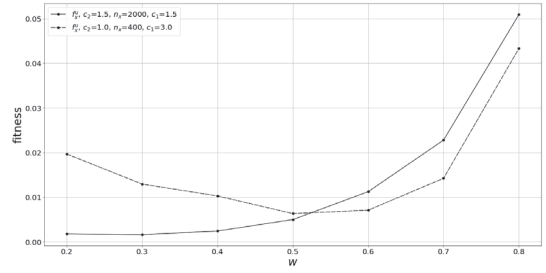
parms	cases	cases
n_x	50,100,400,1000,2000, 4000,10000	7
ω	0.2, 0.5, 0.8	3
c_1	0.5, 1, 1.5, 2.0, 2.5, 3.0	6
c_2	0.5, 1, 1.5, 2.0, 2.5, 3.0	6
total cases		756

Table 13은 PSO 알고리즘의 몬테카를로 시뮬레이션 결과로 도출된 최적 파라미터 조합과 성공확률 값을 보여주고 있다. 단봉일 경우 n_x 이 2000이고 w 가 0.3로 선택되었고 다봉일 경우 c_1, c_2 가 각각 1.5, 1.0으로 선택되었고 다봉일 경우 n_x 이 400이고 w 가 0.5로 선택되었고 다봉일 경우 c_1, c_2 가 각각 3.0, 1.0으로 선택되었다. 단봉일 경우 n_x 가 2000으로 큰 수치가 선택되었는데 변별력이 크게 요구되기 때문으로 보인다.

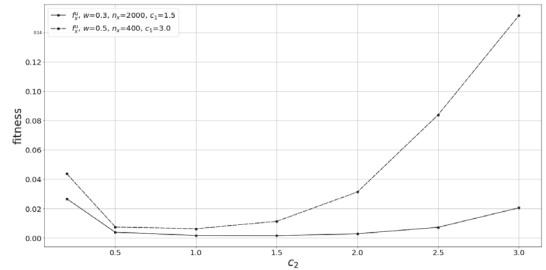
Table 13. optimal parameter combination for PSO

parms	test function	
	f_x^u	f_x^m
n_x	2000	400
ω	0.3	0.5
c_1	1.5	3.0
c_2	1.0	1.0
$1-p^g$	0.0016	0.0049

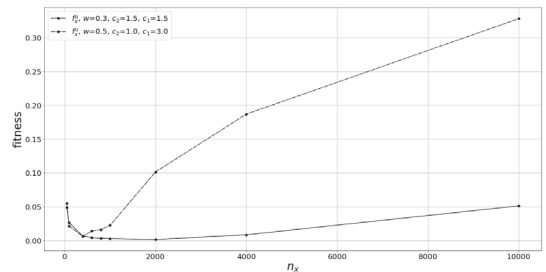
Fig 7는 PSO의 몬테카를로 시뮬레이션 수행결과 중 최적값을 중심으로 변수별 민감도를 보여주는 결과이다. ω 는 단봉일 경우 낮은 값일 때 높은 성능을 보이고 다봉일 경우는 0.5에서 높은 성능을 있다. 다봉의 목적함수에 적합한 ω 를 선택할 경우 단봉에 대한 성능은 0.05정도 손실을 보게 된다.



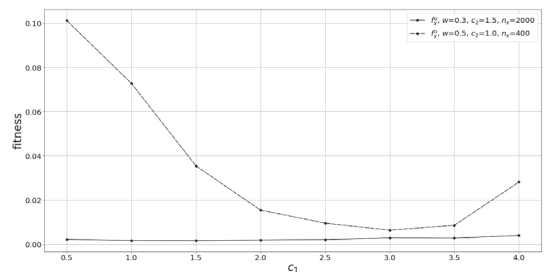
(a) fitness vs ω



(b) fitness vs c_2



(c) fitness vs n_x



(d) fitness vs c_1

Fig. 6. Results of PSO simulation for test functions

3.4.6 소결론

Table 14는 알고리즘간의 성능을 테스트 함수에 따라서 비교한 것이다. 테스트 함수가 단봉일 경우 PRS 대비 다른 모든 알고리즘이 대비 월등히 향상된 성능을 보

고 있음을 알 수 있다. 문제의 복잡도가 높아진 다봉의 가랑 향상됨을 확인할 수 있었고 RORR을 적용할 때 RO 대비 30%정도 향상을 보이고 있으며 전통적으로 전역탐색 알고리즘으로 이용되는 SA는 RO 대비 2배이상의 향상을 보이고 있다. PSO는 RO 대비 10배 정도의 높은 성능향상을 보이고 있으며 군집에 의한 집단 지능이 반영된 효과로 판단된다.

본 분석은 테스트 함수의 특정한 형태에 대하여 분석된 결과로 일반화시킬 수는 없으나 대체로 복잡한 목적함수의 구조가 복잡한 경우의 하나인 국부 최대치의 개수가 증가할 경우에도 상당부분 탐색능력을 높일 수 있음을 알 수 있었다.

Table 14. comparison among algorithms

algorithm	test function	
	$1 - f_x^n$	$1 - f_x^m$
PRS	0.59	0.43
RO	0.0017	0.0786
RORR	0.0020	0.0489
SA	0.0013	0.0326
PSO	0.0016	0.0063

4. 결론

본 논문에서는 이항 반응 실험을 이용해서 몬테카를로 시뮬레이션에 의해서 추정 성공확률을 이용하는 상황에서 등반 알고리즘 기반의 알고리즘, 담금질 기법, 입자 군집 최적화를 비롯한 다양한 알고리즘의 특징을 분석하고 몬테 칼로 시뮬레이션을 통하여 각 알고리즘의 구성 파라미터가 탐색성능에 미치는 영향을 분석하고 알고리즘 간의 성능을 비교하였다.

이항 반응 실험을 이용한 알고리즘의 성능을 비교하기 위하여 단봉과 다봉과 피크의 크기 및 첩도(뾰족한 정도의 척도)를 변화시킬 수 있는 테스트 함수를 제안하였다. 제한된 테스트 함수에 대하여 수행한 결과이지만 단봉의 테스트 함수에 대하여는 대부분의 알고리즘이 높은 수준의 성능 향상을 보이고 있음을 확인할 수 있었지만 다봉의 테스트 함수에 대하여는 알고리즘별로 약간의 성능차이를 보이고 있음을 확인하였고 군집기반의 알고리즘인 PSO는 특별히 높은 성능을 보임을 알 수 있었다.

대체로 테스트 함수가 단봉일 경우와 다봉일 경우에 따라서 최적의 파라미터의 선정은 서로 모순되는 경우가

발생하여 최적화 알고리즘의 적용시 목적함수의 형태에 대한 충분한 검토가 중요함을 알 수 있고 목적함수의 형태를 모르더라도 근집기반의 알고리즘인 PSO의 높은 성능을 활용하여 대규모 실험의 수행속도를 단축하고 성능이 향상된 결과를 도출할 수 있다.

100만 회 이상의 총실험 횟수의 변경에 따른 영향을 분석할 필요가 있고 솔루션 당 실험 횟수를 동적으로 변화시키면서 총 실험 횟수를 줄이거나 탐색성능을 향상시키기 위한 연구도 유전 알고리즘 등 다양한 알고리즘에 대한 검토도 필요하며 필요할 것으로 보인다.

References

Akimoto Y., S. A. Morales, and O. Teytaud(2015), "Analysis of runtime of optimization algorithms for noisy functions over discrete codomains", Theoretical Computer Science, vol. 605, pp. 42-50.

Alarie S., et al(2021), "Digabel Two decades of blackbox optimization applications", EURO Journal on Computational Optimization 9.

Bäck T.(1996), "Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming", Genetic Algorithms. Oxford University Press, Oxford, UK.

Bäck T., and U. Hammel(1994), "Evolution strategies applied to perturbed objective functions," in Proceedings of IEEE Congress on Evolutionary Computation, pp. 40-45.

Brooks S. H.(1958), "A Discussion of Random Methods for Seeking Maxima," Operations Research, 6, 244-251.

Cai T., F. Pan, and J. Chen(2004), "Adaptive particle swarm optimization algorithm." In Proceedings of Fifth World Congress on Intelligent Control and Automation, 2004. WCICA 2004, volume 3, pages "2245-2247

Chandran S. and R. R. Rhinehart.(2002) "Heuristic random optimizer-version II." In Proceedings of the American Control Conference, volume 4, pages 2589-2594. IEEE, Piscataway NJ, US

Eberhart R. C. and J. Kennedy(1995), "A new optimizer using particle swarm theory." In Proceedings of the Sixth International Symposium on Micro Machine

- and Human Science MHS'95, pages 39-43. IEEE Press
- Gao Y. and Y. Duan(2007), "An adaptive particle swarm optimization algorithm with new random inertia weight." In *Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques*, volume 2, pages 342-350.
- Gao Y. and Z. Ren(2007), "Adaptive particle swarm optimization algorithm with genetic mutation operation." In *Proceedings of the Third International Conference on Natural Computation (ICNC 2007)*, volume 2, pages 211-215.
- Glover F.(1986), "Future Paths for Integer Programming and Links to Artificial Intelligence". *Computers and Operations Research*, 13(5), 533-549.
- Gurin and A. R. Leonard(1965), "Convergence of the random search method in the presence of noise", *Automation and Remote Control*, 26, 1505-1511.
- Kirkpatrick S., C. D. Gelatt, Jr., and M. P. Vecchi (1983), "Optimization by simulated annealing." *Science*, 220(4598), 671-680.
- Li J. and R. R. Rhinehart(1998). "Heuristic random optimization." *Computers and Chemical Engineering*, 22(3), 427-444.
- Matyas J.(1965) "Random optimization." *Automation and Remote Control*, 26(2), 244-251.
- Metropolis N., et. al.(1953), "Equation of state calculations by fast computing machines." *The Journal of Chemical Physics*, 21(6), 1087-1092.
- Nelder J. A. and R. Mead,(1965), "A simplex method for function minimization", *Comput. J.*, 7, pp. 308-313.
- Rastrigin L. A.(1963), "The convergence of the random search method in the extremal control of many-parameter system". *Automation and Remote Control*, 24, 1337-1342.
- Russel S. and P. Novic(2002), *Artificial Intelligence: A Modern Approach*, Prentice Hall, second edition, December.
- Schumer M. A. and K. Steiglitz(1968), "Adaptive step size random search." *IEEE Transactions on Automatic Control*, AC-13(3), 270-276.
- Schumer M. A(1967). *Optimization by adaptive random search*. PhD thesis, Princeton University, NJ, Supervisor Kenneth Steiglitz.
- Spall J. C.(2000), *Introduction to Stochastic Search and Optimization. Estimation, Simulation, and Control* (2003), Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, first edition.
- Venter G. and J. Sobieszczanski-Sobieski(2003), "Particle swarm optimization." *AIAA Journal*, 41(8), 1583-1589.
- Worakul N., W. Wongpoowarak, and P. Boonme(2002) "Optimization in development of acetaminophen syrup formulation." *Drug Development and Industrial Pharmacy*, 28(3), 345-351.
- Yuret D. and Michael de la Maza(1993), "Dynamic hill climbing: Overcoming the limitations of optimization techniques", In *Proceedings of the Second Turkish Symposium on Artificial Intelligence and Neural Networks*, pages 208-212, June 24-25, Boğaziçi University, Istanbul, Turkey.



이 동 훈 (ORCID : <https://orcid.org/0009-0000-7591-4956> / ldh2214@paran.com)

1986 고려대학교 통계학과 경제학사
1988 고려대학교 통계학과 이학석사
1990~ 현재 국방과학연구소 해양기술연구원 수석연구원

관심분야 : 시뮬레이션, 국방 M&S, 최적화모델링



황 근 철 (ORCID : <https://orcid.org/0000-0001-8594-629X> / hkchul@chol.com)

2001 경북대학교 전자전기공학부 학사
2003 서울대학교 전기컴퓨터공학부 석사
2023 한국과학기술원 로봇공학학제 박사과정
2003~ 현재 국방과학연구소 해양기술연구원 책임연구원

관심분야 : 무기체계 모델링&시뮬레이션, 체계운용효과도 분석
모델기반 시뮬레이션, 로보틱스



이 상 일 (ORCID : <https://orcid.org/0000-0003-1881-9041> / peterlee2231@gmail.com)

2001 해군사관학교 기계조선공학과 공학사
2006 서울대학교 기계항공공학부 기계공학석사
2013 Texas A&M Univ. 재료공학과 박사수료
2013~ 2019 해군사관학교 기계조선공학과/무기체계공학과 조교수
2019~ 현재 국방과학연구소 해양기술연구원 현역연구원

관심분야: 국방 M&S, 최적화모델링, 소성재료전산해석, 해군무기체계



윤 원 영 (ORCID : <https://orcid.org/0000-0002-1096-8651> / wonyun@pusan.ac.kr)

1982 서울대학교 산업공학과 학사
1984 KAIST 산업공학과 석사
1988 KAIST 산업공학과 박사
1989~ 현재 부산대학교 교수

관심분야 : RAM 및 예방정비 최적화, 검사정책, 보증정책, 시뮬레이션