

# 이항 반응 시뮬레이션의 성공확률 최적화를 위한 대체모델 및 리샘플링을 이용한 유전 알고리즘 응용

이동훈<sup>\*,1)</sup> · 황근철<sup>1)</sup> · 이상일<sup>1)</sup> · 윤원영<sup>2)</sup>

<sup>1)</sup> 국방과학연구소 해양기술연구원

<sup>2)</sup> 부산대학교 산업공학과

## An Application of Surrogate and Resampling for the Optimization of Success Probability from Binary-Response Type Simulation

Donghoon Lee<sup>\*,1)</sup> · Kunchul Hwang<sup>1)</sup> · Sangil Lee<sup>1)</sup> · Won-young Yun<sup>2)</sup>

<sup>1)</sup> *The 1st Research and Development Institute, Agency for Defense Development, Korea*

<sup>2)</sup> *Department of Industrial Engineering, Pusan National University, Korea*

(Received 15 August 2021 / Revised 3 January 2022 / Accepted 16 May 2022)

### Abstract

Since traditional derivative-based optimization for noisy simulation shows bad performance, evolutionary algorithms are considered as substitutes. Especially in case when outputs are binary, more simulation trials are needed to get near-optimal solution since the outputs are discrete and have high and heterogeneous variance. In this paper, we propose a genetic algorithm called SARAGA which adopts dynamic resampling and fitness approximation using surrogate. SARAGA reduces unnecessary numbers of expensive simulations to estimate success probabilities estimated from binary simulation outputs. SARAGA allocates number of samples to each solution dynamically and sometimes approximates the fitness without additional expensive experiments. Experimental results show that this novel approach is effective and proper hyper parameter choice of surrogate and resampling can improve the performance of algorithm.

Key Words : Noisy Optimization(잡음 최적화), Genetic Algorithm(유전 알고리즘), Bernoulli Simulation(베르누이 시뮬레이션), Dynamic Resampling(동적 리샘플링), Surrogate(대체모델)

### 기 호 설 명

$P$  : 개체군(population)

$E$  : 엘리트 솔루션 풀

$M$  : 메이팅 풀

\* Corresponding author, E-mail: dhlee@add.re.kr

Copyright © The Korea Institute of Military Science and Technology

- $S$  : 솔루션 저장소
- $n_x, m_x$  : 솔루션  $x$ 의 표본수와 성공횟수
- $n_0$  : (최초 평가를 위한) 최소 표본수
- $n_{max}$  : 솔루션  $x$ 의 최대허용표본수
- $N_{TOT}$  : 총 시행(시뮬레이션) 횟수
- $N_{MAX}$  : 총 시행(시뮬레이션) 횟수제한(종료조건)
- $x = [x_i, i = 0, 1, \dots, k]$  :  $k$ -차원의 시행 솔루션
- $p_x = m_x/n_x$  : 솔루션  $x$ 에서의 적합도 추정치
- $f_x$  : 솔루션  $x$ 에서의 실제성공확률
- $f_x^s$  : 솔루션  $x$ 에서의 적합도의 대체모델 추정치
- $n_{GEN}$  : 세대의 수
- $n_{pop}$  : 개체군의 크기
- $N(\mu, \sigma^2)$  : 평균  $\mu$ , 분산  $\sigma^2$ 인 정규분포
- $B(n, p)$  : 시행횟수  $n$ , 성공확률  $p$ 인 이항분포
- $t_v(x)$  : 자유도  $v$ 인  $t$ -분포의 누적분포함수

## 1. 서론

잡음 최적화(Noise Optimization) 문제는 항공, 기계, 의학, 수송과 제정 등 많은 분야에서 실재적으로 부딪히는 문제이다. 항공기나 유도탄의 설계, 교통신호등 체계 개발, 경제적 투자결정 등의 많은 문제들은 확률적인 특성들을 내포하고 있어 결정론적 문제에 적합한 전통적인 최적화 방법은 효율이 낮거나 불가능한 경우가 많다. 이를 해결하기 위해 진화적 알고리즘<sup>[1]</sup>을 이용한 최적화가 잡음환경에서 광범위하게 성공적으로 채택되어왔다<sup>[2,3]</sup>.

잡음환경에서 시스템의 출력/효율을 최대화시키는 문제는 실세계의 최적화 문제로서 확률적 최적화모형으로 정형화될 수 있다. 시스템은 일련의 측정치와 미래 상황의 추정치의 수학적인 관계식을 이용하여 전형적인 시시스템의 운용문제를 모델링할 수 있다. 이때의 목적함수는 격추시간, 최근접거리와 같은 연속변수 값 또는 명중/불명중, 격추/생존과 승/패와 같은 이항값으로 측정된다.

목적함수를 구성하는 측정(measurement)과 미래상황의 추정(estimation)이 잡음에 의해서 오염되어 있거나 근원적으로 잡음을 가진 경우의 잡음최적화(optimization with noise) 문제는 원천적으로 불연속이기 때문에 전통적인 미분기반의 최적화 기법은 적용하기 어렵다.

그리드 탐색(grid search)는 탐색공간을 완전 탐색하기 위한 방법으로 파라메타의 차원이 커지면 차원의 저주(curse of dimensionality)에 의하여 현실적으로 불가능하다. 탐색공간의 표본을 랜덤으로 취하여 탐색하는 랜덤 탐색(random search)<sup>[4]</sup>을 이용하여 성능은 낮지만 현실적인 대안으로 고려할 수 있다. 확률적 탐색을 변형하여 방향성을 가진 랜덤탐색 방법이 오랫동안 연구되어 왔는데 시뮬레이티드 어닐링(simulated annealing)<sup>[5]</sup>, 입자군 최적화(particla swarm optimization)<sup>[6]</sup>, 개미집단 최적화(ant colony optimization)<sup>[7]</sup>, 유전 알고리즘(genetic algorithm)<sup>[8]</sup> 등을 포함하여 많은 연구가 이루어지고 있다. 본 논문에서는 잡음최적화문제를 풀기 위하여 대표적인 메타 휴리스틱(meta-heuristics)의 하나로 진화 알고리즘의 일종인 유전 알고리즘<sup>[9]</sup>을 적용하고자 한다.

잡음이 유발한 적합도 함수의 반응표면(response surface)의 동적인 변이로 인하여 동일한 시행 솔루션(trial solution)을 평가할 때마다 적합도 함수는 다른 결과를 출력한다. 이런 잡음환경에서는 뛰어난 성능의 시행 솔루션이 적합도 추정치가 열세로 보임으로 인해 다음 세대에 살아남지 못할 수 있는 반면 근본적으로 열세인 솔루션이 기만적인 적합도 추정치로 인해 살아남을 수 있다<sup>[10,11]</sup>.

많은 경우 확률적 시뮬레이션의 출력은 승/패나 성공/실패으로 나타나는 확률사건으로 나타나며 0 또는 1 두가지 값만 가지는 베르누이 확률변수로 표현이 가능하다. 베르누이 확률변수의 평균으로 계산되는 적합도 추정치는 추정오차를 줄이기 위해서 다수의 고가의 시뮬레이션(expensive simulation)을 수행하여야 한다. 솔루션을 평가하기 위해서 많은 횟수의 시뮬레이션이 필요하여 시간 또는 수행 비용이 많이 든다는 의미에서 값비싼 시뮬레이션이라하고 근사화한 대체 모델은 상대적으로 거의 비용과 시간이 소요되지 않는다. 본 논문에서는 베르누이 분포를 따르는 이항 시뮬레이션 출력의 평균을 적합도로 이용하는 고가의 시뮬레이션의 시행을 줄이는 경제적인 유전 알고리즘을 제안하고자 한다. 이 문제를 해결하기 위하여 두가지 전략을 적용하는데 하나는 리샘플링(resampling)이고 다른 하나는 대체모델(surrogate)에 의한 근사화(approximation)이다.

리샘플링 전략은 기본적으로 정적인 리샘플링(static resampling)과 동적인 리샘플링(dynamic resampling)의 두가지 범주로 구분될 수 있다. 확률 잡음에 의한 목적함수의 오염된 반응표면상의 시행 솔루션의 적합도

는 몬테칼로 적분(Monte Carlo integration)에 의해서 근사적으로 추정될 수 있다. 즉, 리샘플링을 통해 반복 평가횟수를 늘임으로써 몬테칼로 적분에 따르면 잡음에 기인한 개발 솔루션의 적합도의 추정의 불확실성은 감소될 수 있다. 진화형 잡음최적화를 위한 리샘플링 전략은 Miller 등<sup>[12]</sup>에 의하여 논의되었다. 불확실성 감소를 위해서 시행횟수를 모든 솔루션에 일괄적으로 부여하는 정적 리샘플링과 솔루션 별로 필요에 따라서 시행횟수를 조정하는 동적 리샘플링 방법이 있다.

정적인 리샘플링은 모든 개체군의 구성원에 동일한 표본수를 할당하는 것이다. 이 전략은 불필요한 솔루션에도 같은 수의 시행횟수를 부여하는 방법으로 총 자원의 크기와 계산의 정확도에 대하여 상호절충할 수 없다는 것이 큰 단점이다.

동적인 리샘플링은 표본의 요구에 따라서 개발 개체군 구성원에 상이한 표본수를 할당하는 것이다<sup>[13-17]</sup>. 후보 솔루션의 샘플링 요구사항은 적합도의 분산, 표본수의 갱신주기와 최적화 문제의 특성에 주로 의존한다. 전형적으로 잘 알려진 리샘플링 전략은 표준오차 동적 리샘플링(standard error dynamic resampling), 적합도 기반 동적 리샘플링(Fitness Based Dynamic Resampling(FBDR))<sup>[18,19]</sup>, 적합도기반 표준오차 동적 리샘플링(fitness based standard error dynamic rasampling) 외에 문제에 따라서 다양한 동적인 리샘플링 기법이 있다.

적합도 함수가 전산자원소모 차원에서 고가인 경우이거나, 명확한 수학적 적합도 함수가 없는 경우이거나 원 적합도 함수가 잡음이 섞여 있는 경우에 원 적합도 함수의 효과적인 근사가 필요하다. 다항식 (polynomial), 반응표면법(response surface methodologies), 신경망(neural networks), 방사형 기저함수망(radial-basis-function networks), 서포트 벡터 머신(support vector machines), 가우시안 프로세스(Gaussian processes)와 크리징 모델(Kriging models)을 이용한 대체 모델을 적용할 수 있다. 본 논문에서는 단순한 삼각 커널(Trianglar Kernel)을 적용한 비모수 커널 회귀(Nonparametric Kernel Regression)를 이용한 대체모델을 적용하였다.

본 논문의 구성은 2절에서는 적합도 함수로서의 성공확률의 특성을 분석하고 정적, 동적인 리샘플링 전략을 선택하는 방안을 설명하고, 3장은 삼각커널회귀를 이용한 성공확률의 대체모델을 설명하고, 4장에서는 컴퓨터 자원과 성능을 조절하기 위한 대체모델과 리샘

플링을 이용한 유전 알고리즘(Surrogate and Resampling Assisted Genetic Algorithm(SARAGA))을 제안하고, 5절은 실험 결과를 통해서 제안된 알고리즘의 효과를 살펴보고 하이퍼 파라메타에 따른 성능을 비교하였으며 마지막으로 6절에서는 결론과 향후 연구할 방향을 제시한다.

## 2. 성공 확률과 리샘플링 전략

### 2.1 이항반응 확률적 시뮬레이션을 이용한 성공확률의 평가

우리의 목적은 다수의 시뮬레이션을 수행하여 알지 못하는 실제 성공확률  $f_x$ 를 추정하여  $f_x$ 를 최대화하는 솔루션  $x$ 를 구하는 것이다. 각 시뮬레이션의 실행결과는 독립적인 베르누이 분포를 따른다. 실제 문제에서 알 수 없는 목적함수인 실제성공확률  $f_x$ 는 몬테칼로 실험에서 추정될 수 있으며 실험의 결과로부터 시행 솔루션  $x$ 에서의 시행횟수  $n_x$ 에 대한 성공횟수  $m_x$ 와 관련된 정보는 식 (1), (2)와 (3)과 같이 주어진다.

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

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

$$p_x - p_y = \frac{m_x}{n_x} - \frac{m_y}{n_y} \approx N(f_x - f_y, \frac{f_x(1-f_x)}{n_x} + \frac{f_y(1-f_y)}{n_y}) \tag{3}$$

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

$$p\{p_x > p_y\} = p\{p_x - p_y > 0\} \approx t_v \left( \frac{\mu_d}{s_d} \right) = t_v \left( \frac{p_x - p_y}{\sqrt{\frac{s_x^2}{n_x} + \frac{s_y^2}{n_y}}} \right) \tag{4}$$

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 리샘플링 전략

Pietro<sup>[11]</sup>는 확률적 적합도를 갖는 잡음 최적화 문제를 위한 동적인 리샘플링 기법을 연구했으며 그 내용을 정리하면 다음과 같다. 탐색영역의 모든 솔루션의 잡음 강도가 상이한 문제의 경우 다른 리샘플링 비율을 사용하여 추가적인 리샘플링으로 적은 이득을 주는 영역에서 샘플의 낭비를 피할 수 있도록 하고, 반면에 추가적인 샘플이 유익한 경우 추가적인 샘플을 가능하게 한다. 각 후보의 샘플수를 동적으로 할당하는 기법을 동적인 리샘플링이라 한다.

표준오차 동적 리샘플링(SEDR)은 샘플에 대한 평균의 표준오차가 일정 역치 이하가 될 때까지 후보의 적합도의 샘플을 반복하도록 하는 것이다. 평균의 표준오차는 식 (5)와 같이 정의된다.

$$s_e = \frac{s_x}{\sqrt{n_x}} \quad (5)$$

여기서  $s_x = \sqrt{\frac{\sum_{i=1}^{n_x} x_i^2 - n_x \cdot \bar{x}^2}{n_x - 1}}$  는 표준편차이고  $\bar{x} = \sum_{i=1}^{n_x} x_i / n_x$  이며  $n_x$  는 표본의 수이다.

적합도 기반의 동적 리샘플링(FBDR)은 리샘플링 비율을 잡음강도가 아닌 적합도에 기반하는 리샘플링 기법이다. 최초 표본수  $n_0$ 는 후보해의 적합도의 최초 추정치에 이용되고 리샘플링이 반복적으로 수행되어 표본의 수  $n_x$ 가 현재 적합도의 추정치의 함수가 요구하는 리샘플링 비율보다 크도록 한다. 요구되는 표본수는 수식 (6)과 같이 주어진다.

$$r(f) = c_0 + c_1 f \quad (6)$$

여기서,  $c_0, c_1$ 은 각각 임의의 상수이며  $f = \frac{g - \bar{x}}{g}$  는 표준화된 적합도의 척도이며  $g$ 는 적합도 함수의 범위이다.

적합도 기반의 표준오차 동적 리샘플링(FBSEDR)은 FBDR과 SEDR을 결합함으로써 잡음강도와 적합도를 동시에 고려한다. 소요되는 표본수는 식 (7)과 같이 주어진다.

$$r(f) = c_0 + c_1 f + \left( \frac{s_x}{se_{thr}} \right)^2 \quad (7)$$

여기서  $f = \frac{g - \bar{x}}{g}$ 는 표준화된 적합도의 척도이며  $se_{thr}$ 는 표준오차 문턱값(threshold)이며  $g$ 는 적합도 함수의 범위이다.

본 논문에서는 결과가 0또는 1로 나타나는 베르누이 형태의 시뮬레이션에 의하여 추정되는 성공확률을 적합도로 활용하는 유전 알고리즘을 적용하기 위하여 FBSEDR을 변형한 알고리즘을 적용한다.

## 3. 성공확률 근사화를 위한 비모수적 커널회귀모델 (nonparametric kernel regression model)

통계적으로 의미있고 신뢰할만한 결과를 내기 위하여 높은 해상도와 신뢰도를 갖는 반복횟수가 많고 긴 수행시간이 소요되는 시뮬레이션을 다양한 조합에 대하여 반복적으로 수행하여야 한다. 따라서 수시간 또는 수일이 소요되는 총 시뮬레이션 횟수를 줄이기 위한 실험방법이 매우 중요하다. 유전 알고리즘에서는 총 시뮬레이션 횟수가 개체군의 크기, 세대의 수와 확률 추정을 위한 반복횟수에 의존하여 많은 횟수의 시뮬레이션을 수행하여야 한다. 본 논문에서는 각 솔루션의 성공확률 추정을 위한 반복횟수를 동적으로 결정하는 것과 아울러 추정된 성공확률 결과를 분석하여 얻어진 목적함수  $f$ 에 대한 대체모델  $f^s$ 를 이용하여 불필요한 시뮬레이션의 횟수를 줄이면서 탐색의 질을 향상시키고자 한다. 대체모델은 실제 적합도를 근사적으로 추정함으로써 시뮬레이션 횟수를 줄이기 위한 효과적인 방법의 하나로 알려져 있다. 대체모델 기반의 최적화는 실제 적합도의 평가가 시간과 비용이라는 측면에서 비 효율적이고 상대적으로 대체모델 학습이 효율적인 경우에 유용하다. 유전 알고리즘 수행 중에 현재 세대(generation)까지 발생된 모든 시행 솔루션의 집합  $D = \bigcup_{i=n}^g D_i (D_i : i \text{ 번째 세대의 솔루션의 집합})$ 에 속한 모든 솔루션  $x$ 와 그에 상응하는 값 비싼 시뮬레이션에 의한 적합도의 추정치  $\hat{f}_x$ 로 이루어진 대체모델 저장소  $S = \{(x, \hat{f}_x), x \in D\}$ 에 대체모델 학습을 위한 데이터가 누적된다. 대체 모델 저장소

S는 대체모델 학습을 위한 데이터를 제공하며 고성능 병렬 컴퓨터에 의해서 수행되는 반복횟수가 많고 긴 시간의 수행이 요구되는 값비싼 시뮬레이션을 줄이는 역할을 한다.

대체모델로서 방사형기저함수<sup>[20]</sup>, 크리징 모델<sup>[21]</sup>, 서포트 벡터 머신<sup>[22]</sup>, 다항회귀<sup>[23]</sup>, 스플라인<sup>[24]</sup> 등 다양한 모델이 적용되어 왔다. 유전 알고리즘을 적용할 경우 세대가 진행됨에 따라서 대체모델 저장소에 시행솔루션과 적합도 추정치가 누적됨으로써 대체모델 계산의 신뢰성을 높일 수 있으나 과대한 표본의 수와 차원의 저주 등에 따라서 대체 모델의 계산시간이 크게 늘어나기도 한다. 본 논문에서는 개념적으로 단순하고 방대한 데이터 셋에 적용하기가 용이한 삼각커널을 적용한 커널회귀(Trigonal Kernel Regression : TKR)를 대체모델로 적용하였다. 대체모델을 위하여 각 세대의 모든 적합도 평가 데이터는 대체모델 저장소 S에 누적하여 저장되며 대체모델은 저장소 S의 데이터가 이용하여 계산된다. TKR에서  $x$ 로부터 반경  $r$ 이내의 점들을 찾아 그 점들에 의한 추정성공확률의 가중평균이 성공확률의 대체추정치가 된다. 최종적인 성공확률의 대체 추정치는 식 (8)와 같다.

$$f_x^S = \frac{\sum_{y \in ngh(x)} n_y \times p_y \times w_y}{\sum_{y \in ngh(x)} n_y \times w_y} \quad (8)$$

여기서,  $ngh(x) = \{y \in D : |x, y| < r\}$ ,  $D$  : 시행 솔루션 집합,  $w_s = \frac{|x - s|}{r}$ ,  $|a|$  :  $a$ 의 놈(norm)이며  $f_x^S$ 는 개체  $x$ 의 적합도(성공확률)의 대체모델 추정치를 의미한다.

$\sum_{y \in ngh(x)} n_y \times w_y$ 이 작아지면(즉  $x$  근처의 솔루션에 대한 많은 실험이 이루어지면),  $f_x^S$ 는 낮은 정확도를 가지며  $\sum_{y \in ngh(x)} n_y \times w_y$ 가 커지면(즉  $x$  근처에 솔루션에 대하여 많은 실험이 이루어지면),  $f_x^S$ 는 높은 정확도를 갖는다. 또한 유전 알고리즘 수행의 초기 단계에는  $x$  근처에 확률적으로 희박한 데이터가 있고 저장소 S가 누적됨에 따라서  $x$  근처에 데이터의 밀도가 높아져서 대체모델의 신뢰수준이 높아진다.

제안된 알고리즘은 기존의 대체모델 저장소의 데이터를 이용하여 TKR 대체모델을 계산한다. 이 때 추가

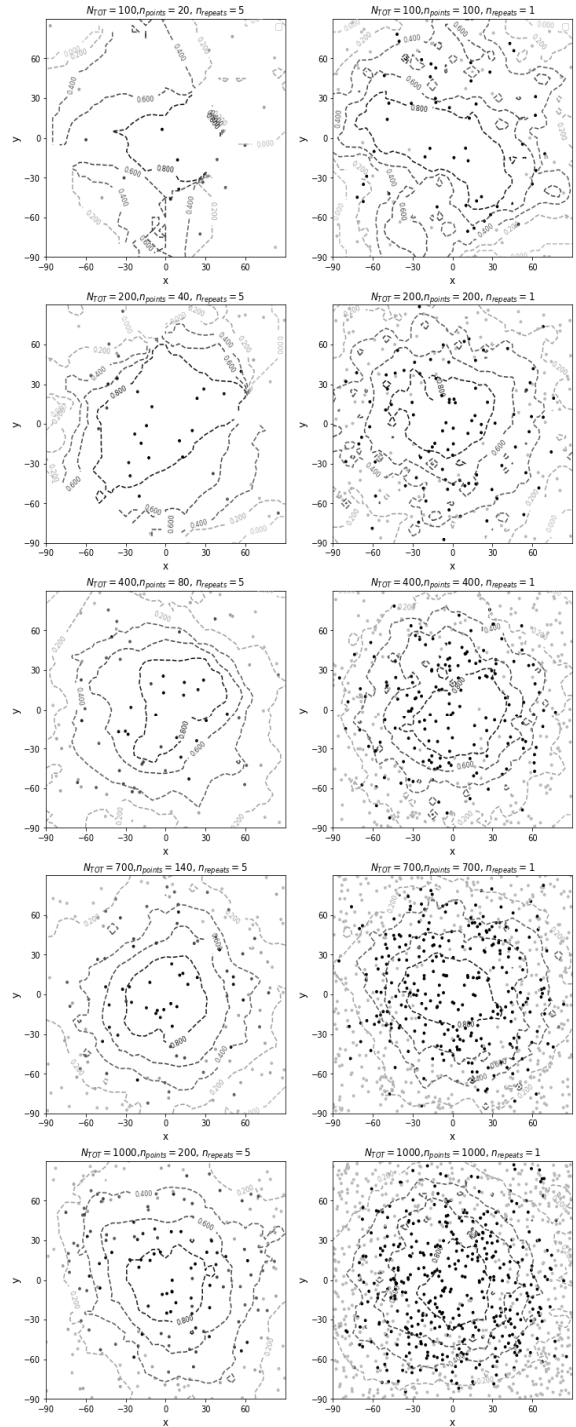


Fig. 1. Comparison of TKR fitness estimations according to number of solutions and number of trials

적인 실험 없이 적합도(성공확률)를 추정함으로써 시뮬레이션의 시행횟수를 줄임으로써 소요시간을 단축할 수 있도록 한다. 또한 세대가 진행됨에 따라서 대체모델 저장소의 데이터가 누적됨으로써 더 정확한 대체모델이 최신화가 수행될 수 있다.

Fig. 1은 솔루션의 개수  $n(p)$ 와 솔루션당 반복 횟수  $n_x$ 의 10가지 조합에 대하여 시뮬레이션 결과를 비교할 수 있다. 각 경우에 대하여 솔루션의 시뮬레이션에 의한 적합도 추정치의 산포도와 TKR 대체 모델 추정치의 등고선 그래프를 겹쳐서 그린 것이다. 총시뮬레이션 횟수  $N_{TOT} = n(p) \times n_x$ 의 증가는 학습된 대체모델의 정확도를 향상시킴을 알 수 있다.  $N_{TOT}$ 가 작을 경우에는 반복횟수보다 솔루션의 개수를 높이는 것이 좋은 것으로 보이며  $N_{TOT}$ 가 큰 경우에는 그 경향이 줄어들고 있다.

#### 4. 리샘플링과 대체모델을 이용한 유전 알고리즘

베르누이 시행과 유사한 출력을 내는 시뮬레이션을 이용한 성공확률의 최적화를 위해서는 대체모델과 리샘플링을 이용한 유전 알고리즘(Surrogate and Resampling

Assisted Genetic Algorithm: 이하 SARAGA)은 고가의 시뮬레이션을 이용한 총 평가 횟수(total number of expensive evaluations)의 절약 기제를 제공한다.

Table 1은 제안된 알고리즘의 유사코드이다. 유전 알고리즘 수행 중 최초로 무작위로 생성된 솔루션 개체군  $P$ 에 대하여 각 개체별로  $n_0$ 회의 값비싼 시뮬레이션을 수행하여 성공횟수  $m_0$ 를 구하고  $n_x, m_x$ 에 각각 입력한 후  $S = S \cup \{x, n_x, m_x, x \in P\}$ 를 저장한다. 누적된  $S$ 의 모든 개체는 Table 2에서와 같이 추가적인 평가가 이루어져서  $S$ 에 누적된다. 또한  $S$ 를 이용하여 대체모델을 구하고 대체모델을 기반으로 엘리트를 선정하고 토너먼트 선택 단계에서도 추가적인 시뮬레이션 없이 대체모델을 이용한다. 총 시뮬레이션 수행단계에서 알고리즘의 성능을 비교하기 위하여 총시뮬레이션 횟수  $N_{TOT}$ 가 기준 횟수  $N_{MAX}$ 를 초과하면 추가적인 진화를 멈추도록 설정하였다. 진화가 종료되면 최대 생존확률을 갖는 솔루션을 바로 최적 솔루션으로 구하지 않고 Table 3의 최종 평가과정을 통하여 확률적 최적 솔루션 추정치의 불확실성을 제거하는 과정을 거친다.

본 알고리즘에서는  $n_0$ 를 작게 설정하여 적은 시뮬레이션 횟수만으로 조기에 낮은 적합도를 갖는 후보를

Table 1. Pseudo code of proposed algorithm(SARAGA)

<ol style="list-style-type: none"> <li>1. <math>M, S, P = \phi</math></li> <li>2. initialize population <math>P</math> randomly</li> <li>3. <b>for</b> <math>x</math> <b>in</b> <math>P</math> : <math>m_x, n_x = 0</math></li> <li>4. <b>repeat</b></li> <li>5.     <b>for</b> <math>x</math> <b>in</b> <math>P</math> :</li> <li>6.         get <math>m_0</math> successes in <math>n_0</math> simulations, <math>n_x = n_x + n_0, m_x = m_x + m_0</math></li> <li>7.         <math>S = S \cup \{x, m_x, n_x, x \in P\}</math></li> <li>8.         <i>additional - evaluate</i>(<math>S; n_{max}</math>)(see table 2)</li> <li>9.         calculate Surrogate(<math>S</math>)</li> <li>10.         select <math>E</math> of <math>e</math> members which have highest surrogate fitness from <math>P</math></li> <li>11.         select <math>M</math> of size <math>len(P) - e</math> selected by tournament according to surrogate fitness</li> <li>12.         <math>P \leftarrow E \cup M</math></li> <li>13.         <math>N_{TOT} = \sum_{x \in S; [0]} n_x</math></li> <li>13. <b>until</b> <math>N_{TOT} \geq N_{MAX}</math></li> <li>14. <b>return</b> <math>\{x^*, p_x\} = final\_evaluate(n_{maxf})</math> # calculating final optimal solution(see table 3)</li> </ol>
--

도태되도록 하였다. 이로써 더 이상의 자원이 투입되지 않고 높은 적합도를 갖는 후보는 살아남아서 이후에 알고리즘에 의해 적절히 자원이 투입되도록 하여 알고리즘의 효율을 높인다. 경우에 따라서  $n_0$ 를 1로 설정함으로써 탐색능력을 최대화하도록 할 수도 있다.

Table 2는 대체모델 저장소의 모든 솔루션을 대상으로 동적 리샘플링을 통하여 추가적인 평가가 이루어지는 절차를 보여준다. 각 솔루션에 대하여  $n_0$ 회의 추가적인 시뮬레이션을 반복적으로 수행하여 평가를 갱신함으로써 최신화 할 수 있도록 한다. 총 시뮬레이션 횟수  $n_x$ 가 주어진 최대 시뮬레이션 횟수  $n_{max}$ 보다 크거나 갱신된 평가가 최대 평가 솔루션보다 확률적으로 충분히 작을 경우 평가의 갱신은 멈추게 된다. 이러한 동적인 리샘플링 과정을 통하여 저평가되는 솔루션에 시뮬레이션 자원을 할당하는 것을 막고 가능성이 높은 솔루션에 자원을 많이 할당하는 전략을 택한다.

Table 2. Additional fitness evaluation of solutions

Algorithm : <i>additional-evaluate</i> ( $S;n_{max}$ )
<p><b>for</b> <math>x</math> <b>in</b> <math>S</math>:</p> <p><b>while</b> <math>n_x &lt; n_{max}</math>:</p> $p_x = \frac{m_x}{n_x}, s_x^2 = \frac{p_x(1-p_x)}{n_x}$ $x_{max} = \mathop{\text{argmax}}_{x \in S}(p_x)$ $p_{x_{max}} = \frac{m_{x_{max}}}{n_{x_{max}}}, s_{x_{max}}^2 = \frac{p_{x_{max}}(1-p_{x_{max}})}{n_{x_{max}}}$ <p><b>if</b> <math>p_x + s_x &lt; p_{x_{max}} - s_{x_{max}}</math> : <b>break</b></p> <p>additional <math>n_0</math> simulations for <math>x</math> &amp; get <math>m_0</math></p> $n_x = n_x + n_0, m_x = m_x + m_0$

선택 과정을 통해서 다음 세대를 결정하는데 엘리트 선택과 재생산과정으로 분리하여 이루어진다. 먼저 부분적으로  $e$ 개의 최대 생존확률을 갖는 엘리트 선택을 수행하고 나머지는 토너먼트 선택을 수행하는데 엘리트의 선택은 대체모델을 이용하여 상위 대체모델 적합도의  $e$ 개의 솔루션이 선택된  $E$ 가 다음 세대에 그대로 전달된다. 토너먼트 선택은 두 솔루션씩 대체모델 적합도를 비교하여 높은 적합도의 솔루션이 선택되어지는데 엘리트 선택과 마찬가지로 추가적인 값

비싼 시뮬레이션은 없다. 토너먼트 선택에 의해서 선택된 솔루션은 교차 및 돌연변이 연산자를 통하여 다음 세대에 전달될 재생산된 새로운 솔루션으로 구성된 자손  $M$ 을 발생시킨다. 다음 세대에 적용될 새로운 개체군  $P$ 는  $E$ 와  $M$ 의 합집합에 의해서 구해진다. 즉  $P = E \cup M$ . 총 시뮬레이션 횟수  $N_{TOT} = \sum_{x \in S} n_x$ 이며 주어진  $N_{MAX}$ 보다 크게 되면 추가적인 진화과정을 멈추고 저장소  $S$ 에 있는 개체를 대상으로 최대의 적합도를 내는 솔루션을 찾는 절차를 수행한다.

Table 3. Final evaluation of optimal solutions

Algorithm : <i>final-evaluate</i> ( $n_{maxf}$ )
$L_k = \{(x_{(i)}, n_{x_{(i)}}, m_{x_{(i)}}), i = 1, \dots, k\}$ <p><b>while</b> <math>\min(n_x) &lt; n_{maxf}</math></p> <p><b>for</b> <math>(x, n_x, m_x)</math> <b>in</b> <math>L_k</math>:</p> <p><b>if</b> <math>n_x \geq n_{maxf}</math>: <b>continue</b></p> <p>additional <math>n_0</math> simulations for <math>x</math></p> <p>update <math>n_x, m_x</math></p> $p_x = \frac{m_x}{n_x}, s_x = \sqrt{\frac{p_x(1-p_x)}{n_x}}$ $x_{(1)} = \mathop{\text{argmin}}(p_x), x_{(k)} = \mathop{\text{argmax}}(p_x)$ <p><b>if</b> <math>p_{x_{(k)}} - p_{x_{(1)}} &gt; \sqrt{s_{x_{(k)}}^2 + s_{x_{(1)}}^2}</math>:</p> <p><b>del</b> <math>(x_{(1)}, p_{x_{(1)}})</math> <b>from</b> <math>L_k</math></p> <p><math>k \leftarrow k - 1</math></p> <p><b>return</b> <math>L_k</math></p>

Table 3은 지금까지 도출된 솔루션을 대상으로 추가적인 시뮬레이션을 수행하여 최종적인 최적해를 도출하는 과정을 보여준다. 먼저 식 (9)와 같이 상위 적합도를 갖는  $k$ 개의 개체들로 구성된 리스트  $L_k$ 를 구하고  $L_k$ 내의 각 개체를  $n_0$ 회씩 추가적으로 평가를 수행함으로써 이루어진다. 추가적인 평가는 최대  $n_{maxf}$ 회까지만 수행하고  $L_k$ 중 최소 성공확률인 개체  $x_{(1)}$ 가 최대 성공확률 개체  $x_{(k)}$ 와의 성공확률 차이가  $\sqrt{s_{x_{(1)}}^2 + s_{x_{(k)}}^2}$ 보다 크면  $L_k$ 에서 삭제되어 추가적인 평가를 하지 않도록 하였다. 또한  $L_k$  리스트의 개체의 반복횟수가 모두  $n_{maxf}$ 보다 크거나 같으면 추가적인

평가는 종료된다. 재평가 대상 개체수  $k$ 를 너무 크게 할 경우 추가적인 평가가 너무 많아지는 단점이 있고 너무 작게 하는 경우 우월한 개체가 누락될 가능성이 있으므로 적절한  $k$ 를 선정하는 것이 중요하다.

$$L_k = \{(x_{(i)}, n_{x_{(i)}}, m_{x_{(i)}}), i = 1, \dots, k\} \quad (9)$$

여기서  $x_{(i)}$ 는  $i$  번째 상위 성공확률의 추정치  $p_x = \frac{m_x}{n_x}$ 을 갖는 솔루션을 의미한다.

## 5. 제안된 알고리즘의 성능비교

### 5.1 테스트 함수

실제 상황에서는 성공확률 형태에 대한 적합도함수의 형태는 일반적으로 알 수 없으며 적합도의 추정치의 분산은 식 (2)와 같이 반복횟수에 의존한다. 적합도함수의 추정에 기초하여 최적 해를 찾는 것은 시간과 비용 측면에서 매우 비효율적인 작업이다. 최적화 알고리즘의 성능을 평가하기 위하여 식 (10)과 같이 단봉이며 최대 확률이  $\theta$ 인 단순한 테스트 함수를 제안하고 테스트 함수로 이용하였다.

$$f_x = \theta \times \left[ \sum_{i=1}^{\nu} \sin \left( \frac{x_i - b_i^j}{b_u^j - b_l^j} \pi \right) / \nu \right]^{\xi}, b_l^j < x_i < b_u^j \quad (10)$$

여기서  $\nu$ 는 변수의 개수,  $\xi$ 는 적합도 함수의 형상을 결정하는 파라메타,  $b_u^j$ ,  $b_l^j$ 는  $i$ 번째 변수  $x_i$ 의 상한과 하한이다.

### 5.2 성능척도(measure of performance)

실제 최적해는 실제 적합도  $f$ 를 가장 높여주는  $x$ 를 찾는 문제로 최적해는 식 (11)과 (12)와 같이 나타나며 결정론적으로 오차 없이 구해질 수 있다. 그러나 실제 확률적 시뮬레이션을 이용하여 최적해를 구할 때는 값비싼 시뮬레이션을 이용하여 제한된 횟수의 시뮬레이션을 수행하여야 하므로 오차를 갖는 추정 적합도  $p$ 를 가장 높이는  $x$ 를 찾는 문제가 되어 식 (13), (14)과 같이 추정된다. 이때의 추정 최적해  $\hat{x}_{opt}$ 는 실제 최적해  $x_{opt}$ 와 거리가 멀고  $p$ 를 통해 솔루션에 대한 평가를 하게되면 과대평가하게 된다. 실제적

합도 함수는 알 수 없는 것이 일반적이나 본 논문에서는 제안된 알고리즘의 적합도 함수를 이용하여 모의 실험을 수행하기 때문에 알고리즘의 평가는  $p$ 를 최대화 시킨 솔루션  $\hat{x}_{opt}$ 을 실제 적합도 함수  $f$ 에 대입한 평가함수인 식 (15)를 이용하여 평가한다.

$$\text{실제 최적해} : x_{opt} = x' \text{ s.t. } f_{x'} \geq f_x \forall x \in R^{\nu} \quad (11)$$

$$\text{실제 최적적합도} : f_{opt} = f_{x_{opt}} \quad (12)$$

$$\text{추정 최적해} : \hat{x}_{opt} = x' \text{ s.t. } p_{x'} \geq p_x \forall x \in S \quad (13)$$

$$\text{추정최적적합도} : p_{opt} = p_{\hat{x}_{opt}} \quad (14)$$

$$\text{추정최적해 평가척도} : Perf = f_{\hat{x}_{opt}} \quad (15)$$

### 5.3 실험 계획

실험계획은 SARAGA가 최적화 성능의 향상을 가져오는지를 확인하고 중요한 하이퍼 파라메타를 찾으며 대체모델의 역할에 대하여 분석하기 위하여 수립되었다. SARAGA 알고리즘의 실제 시뮬레이션에서의 성능을 비교하기 위하여 Intel® Core™ i7-8700 CPU (3.20 GHz)와 16.0 GB of RAM이장착된 Windows 10 OS 환경의 PC에서 실험을 수행하였다.

개체별 최대 시뮬레이션 횟수( $n_{max}$ ), TKR의 반경( $r$ ) 그리고 총 시뮬레이션 횟수( $N_{TOT}$ )는 대체모델과 리샘플링을 이용한 유전 알고리즘의 시뮬레이션에 의하여 추정되는 성공확률의 추정을 위해서 중요한 설정 파라메타이다. 제안된 알고리즘의 파라메타의 설정 값에 따른 효과를 분석하기 위한 실험 계획은 Table 4와 같다. 여기서 세대의 수( $n_{GEN}$ )를 총 시뮬레이션 횟수( $N_{TOT}$ ), 개체군의 크기( $n_{pop}$ )와 솔루션의 표본수( $n_x \leq n_{max}$ )의 변화에 연동되도록 설정하였으며 엘리트 비율( $r_{elit}$ )은 10 %로 설정하였다. 엘리트 비율을 10 %로 선정한 것은 최적해가 확률적으로 도태될 가능성을 줄이기 위함이다.

정적인 리샘플링을 적용한 naive SARAGA를 비교대상에 포함시켰다. naive SARAGA는  $n_{pop} = \frac{N_{TOT}}{n_x}$ 개의 랜덤 해를 발생시키고 시뮬레이션 횟수( $n_x = \frac{N_{TOT}}{n_{pop}}$ )는 고정되고 세대의 갯수는 1인 경우이다.



5.4 실험 결과 및 분석

Table 4는 보편 총 180개 하이퍼 파라메타 조합의 경우에 대하여 20개의 SARAGA 시행이 이루어졌다. 이 실험을 통하여 여러 가지 의미있는 추론결과가 도출이 되었다.

Table 4. Experimental plan to compare SARAGA

parameter	combination	# of cases
$n_{pop}$	100	1
$\theta$	0.95	1
$\hat{p}_{mut}$	$1/\nu$	1
$\xi$	1.5	1
$\nu$	12	1
$\hat{p}_{cross}$	1/2	1
$n_0$	1	1
$r_{elit}$	10 %	1
$r$	naive, 0, 5, 10, 15, 20	6
$n_{max}$	2, 5, 10, 25, 50, 100	6
$N_{TOT}$	2000, 3000, 6000, 10000, 15000	5
num of trials per each combination		20
total number of trials		3600

Fig. 2는 동일한 시뮬레이션 시행횟수  $n_x$ 를 각 솔루션에 할당된 naive SARAGA의 수행결과이다.  $n_{max}$ 가 증가하면 참조 최적알고리즘인 naive SARAGA은 더 좋은 성능을 보인다. 총 시뮬레이션 횟수( $N_{TOT}$ )가 증가하면 naive SARAGA는 성능이 미세하게 좋아진다.  $n_{max} = 100$ 이고  $N_{TOT} = 15000$ 일 때 최대 성능치인 0.72를 보이고 있다.

Fig. 3은 SARAGA( $r = 0$ )를 수행한 결과로 대체모델은 적용하지 않았지만 각 솔루션에 동적으로 다른 횟수의 시뮬레이션 횟수를 할당하는 동적인 리샘플링을 적용한 결과를 보여준다.  $n_{max} = 2$ 인 경우  $N_{TOT}$ 와 상관없이 단순 SARAGA와 SARAGA( $r = 0$ )의 추정 적합도는 1에 이른다. 이것이 의미하는 것은  $n_{max} = 2$ 는 과대 평가된 적합도의 추정치가 너무 일찍 나타나 다른 세대의 어떠한 개체도 이를 추월하기 어렵다는 것이다.

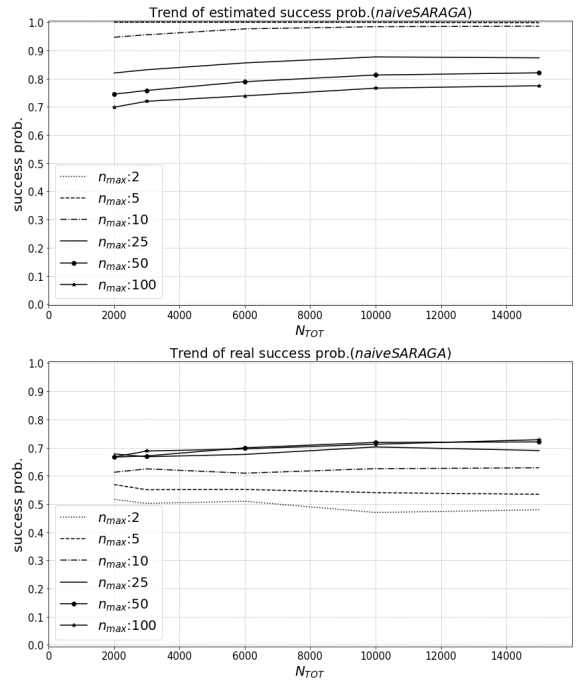


Fig. 2. Performance comparison of naive SARAGA

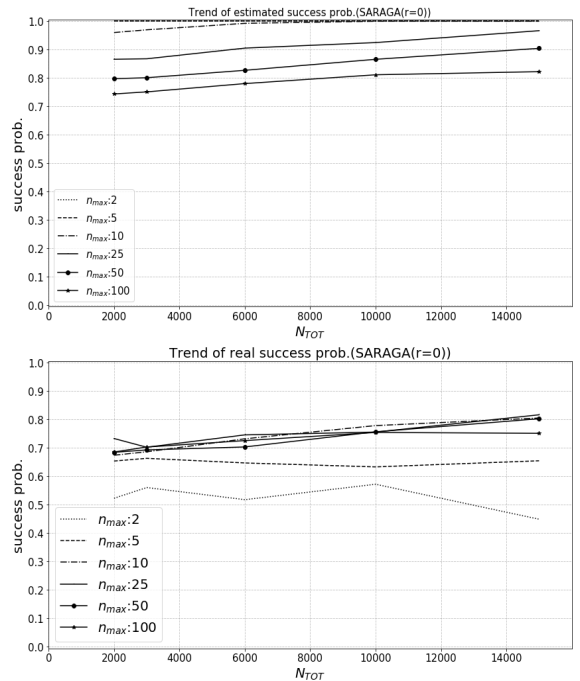


Fig. 3. Performance comparison of SARAGA without surrogate model

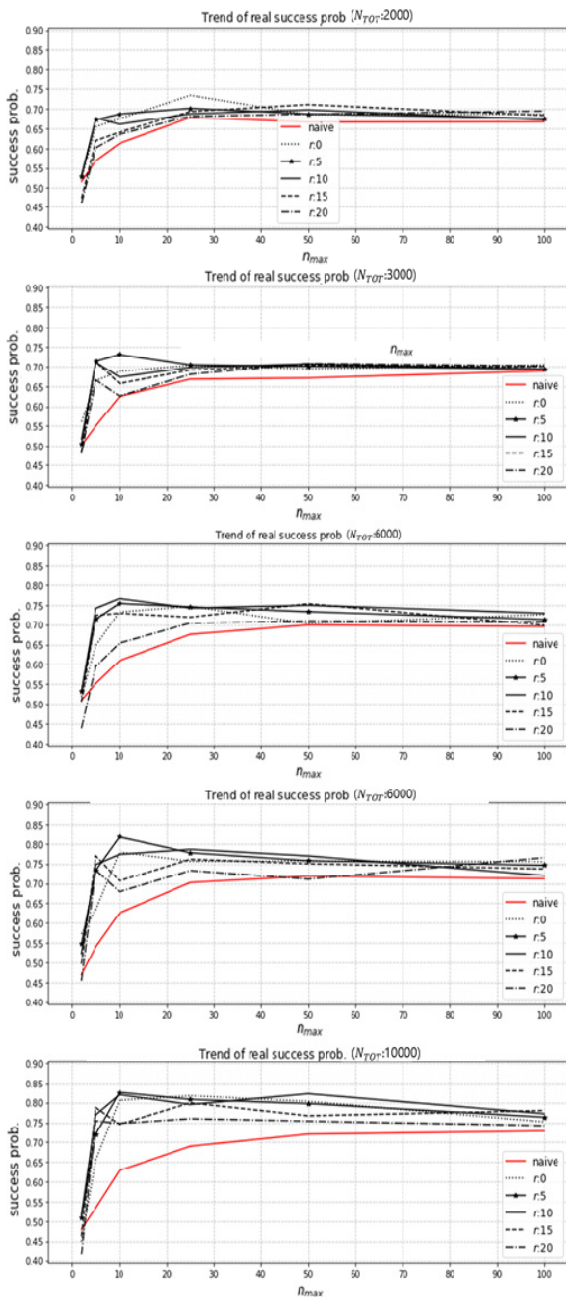


Fig. 4. Performance comparison of SARAGA according to  $n_{max}$ ,  $r$  and  $N_{TOT}$

SARAGA( $r = 0$ )은 명백하게 naive SARAGA보다 성능이 월등함을 알 수 있다. 이유는 SARAGA가 값비싼 시뮬레이션을 절약하도록  $n_x$ 를 동적으로 조절할 수

있는 반면에 naive SARAGA는  $n_x = m_{max}$ 로 고정함에 기인하는 것으로 보인다.  $n_{max} = 5$ 일 때 SARAGA( $r = 0$ )는 단순 SARAGA보다 10% 정도 우세함을 알 수 있다. 이로써 동적인 리샘플링이 적정한 리샘플링보다 좋은 선택임을 실증적으로 확인할 수 있었다.

Fig. 4는 총 시뮬레이션 횟수( $N_{TOT}$ )가 주어졌을 때의 SARAGA 알고리즘을 ( $n_{max}, r$ )의 조합에 따라서 성능을 비교한 것이다.  $N_{TOT}$ 이 증가함에 따른 성능의 증가는 명백하고 자연스러운 것이다.  $N_{TOT} = 2000$ 인 경우를 제외하면  $n_{max}$ 에 따른 실제 성공확률 일양 증가하는 패턴을 보이지 않는데  $N_{TOT}$ 이 증가함에 따라서 ( $r = 5, 10, n_{max} = 10$ ) 조합에서 눈에 띄이게 좋은 성능을 보이고 있다. 이는 너무 높은  $n_{max}$ 는 자원을 낭비할 수 있고 너무 낮은  $n_{max}$ 는 SARAGA가 개체의 평가를 상향평가하게 만들므로써 더 나은 개체의 진입을 방해하기 때문으로 추정된다.

또한  $r$ 의 경우에도 너무 높거나 낮을 경우 성능이 떨어지고 있음을 확인할 수 있다. 이는 너무 낮은  $r$ 는 SARAGA가 값비싼 시뮬레이션에만 의존함으로써 효율이 떨어지고 반대의 경우 실제 적합도의 형태를 왜곡함으로써 성능을 저하시키는 것으로 판단된다. 본 결과는 SARAGA의 적용시  $r$ 과  $n_{max}$ 의 선택이 중요함을 보여준다.

## 6. 음향어뢰방어 시뮬레이션 적용

본 절에서는 본 논문에서 제안된 방법론을 실제 최적화 문제에 응용한 사례연구를 제시하고자 한다. 급속하게 발전하고 있는 수중의 음향어뢰는 함정에 매우 위협적인 무기체계로 함정의 생존성을 높이기 위하여 음향어뢰 방어체계의 보유는 필수적이다. 음향어뢰 방어체계의 효과에 대한 평가는 실제 해양환경에서의 실제 발사시험의 어려움으로 인해 대부분 시뮬레이션을 수행하여 이루어진다. 어뢰방어 시뮬레이션의 결과는 생존과 침몰로 나타나며 생존확률은 어뢰 방어체계의 효과의 척도이다. 음향어뢰 방어체계의 운용개념은 Fig. 5와 같이 어뢰의 공격에 대하여 부유식 및 자항식 기만기로 이루어진 음향어뢰 방어체계를 전술적으로 운용함으로써 함정의 생존성을 높일 수 있다.

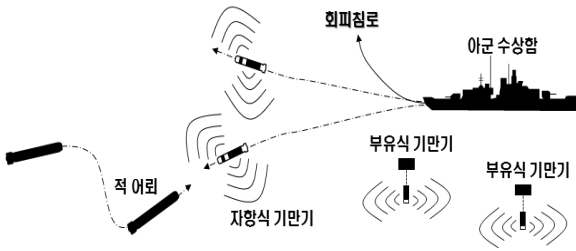


Fig. 5. Concept of operations of acoustic torpedo defense system

Table 5. Tactical decision variables

tactical variable	range	error*
E	-180-180 degree	1 degree
V	15-30 kts	1 %
FD1, FD2	000-000 m	3 degree
FR1, FR2	0-360 degree	5 %
FT1, FT2	0-00 sec	1 sec
MD1, MD2	0-360 degree	3 degree
MT1, MT2	0-00 sec	1 sec

- E : evasive course,
- V : evasive speed,
- FD<sub>i</sub> : launching direction of *i*th floating decoy,
- FR<sub>i</sub> : launching range of *i*th floating decoy,
- MT<sub>i</sub> : activation time of *i*th floating decoy,
- MD<sub>i</sub> : launching direction of *i*th mobile decoy,
- MT<sub>i</sub> : activation time of *i*th mobile decoy,
- \* standard deviation of normal distribution

어뢰방어체계의 전술적 운용은 어뢰경보 정보에 따른 복수의 부유식 기만기와 자항식 기만기의 투척방법의 선정 및 함정회피 방향 및 속도의 선정이다. Table 5는 전술적 의사결정 변수들을 설명하고 있으며 함정의 회피 침로/속도와 세 발의 부유식 기만기 발사 거리/방향 두 발의 자항식 기만기 기동방향/속도 등 12개의 전술 변수로 구성되며 각 변수는 정규분포로 실제 시행에서 반영된다고 가정하여 시뮬레이션에 반영하였다.

최적전술을 도출하기 위한 유전 알고리즘의 하이퍼 파라메타 설정은 Table 5와 같다. 최대 시뮬레이션 횟수( $n_{max}$ )가 10이고, TKR의 반경( $r$ )이 10인 경우와

TKR 반경( $r$ )이 20인 경우를 비교하도록 조합을 설정하였다. 개체군의 크기( $n_{pop}$ )는 100으로 세대의 수( $n_{GEN}$ )는 고정하지 않고 최대실험횟수( $n_{TOT}$ )가 15000이 넘으면 중단하도록 하였다. 돌연변이 확률은 개체당 평균 1회 발생하도록 1/12로 설정하였고 엘리트의 비율은 10 %로 설정하였다. 엘리트 비율 10 %를 고려하는 것은 확률적으로 우수한 개체가 남아있을 확률을 높이기 위함이다. 본 실험은 5절의 성능평가와 동일한 PC환경에서 실험이 이루어졌으며 시뮬레이션 1회당 약 5초의 시간이 소요되었으며, 1회의 SARAGA 알고리즘을 수행하여 최적화수행에 소요된 시간은 약 28.7시간이 소요되었다.

Fig. 6은 SARAGA 알고리즘을 적용하여 최적 전술을 도출하는 적합도(fitness)의 성장과정을 보여준다.  $r = 10$ 이거나  $r = 20$ 와 관계없이 시뮬레이션 횟수가 1500에서 적합도(fitness)가 최고에 이르렀다가 이후 오르내림을 보여주며 변동폭이 점차 작아지는 경향을 보여주고 있다.  $r = 20$ 인 경우 변동의 정도가 눈에 띄게 작아져서 추정된 생존확률보다 일정수준 높은 값으로 수렴하며 미세하게 상승하는 경향을 보여주고 있다.  $r = 10$ 인 경우 추정된 생존확률보다 일정수준 높은 범위에서 오르내림을 반복하고 있음을 확인할 수 있다.

어뢰방어 솔루션의 실제 생존확률은 알 수 없기 때문에 최종적으로 구해진 두 가지 솔루션에 대한 정확한 평가를 위해서 10,000회의 시뮬레이션을 통해서 생존확률을 추정하였다. 두 가지 솔루션의 생존확률의 추정치는 각각 0.8625와 0.8875(표준오차 0.01)로 유사한 수준으로 나왔으며 naive SARAGA( $n_{pop} = 300, n_0 = 50$ , 생존확률 추정치 0.7075(표준오차 0.01))의 솔루션과 비교할 때 15 % 정도 우세함을 알 수 있다.

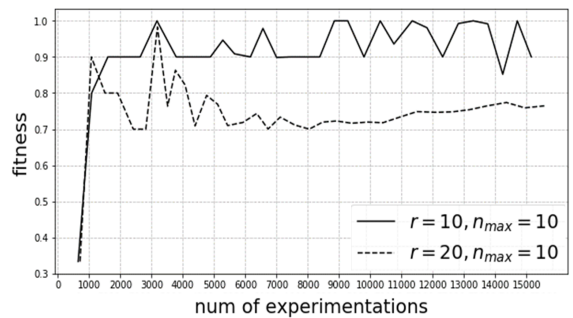


Fig. 6. SARAGA application for tactic optimization

## 7. 결론

본 논문에서는 이항 시뮬레이션을 이용해서 추정되는 성공확률을 적합도로 이용하는 상황에서 동적인 리샘플링과 대체모델을 채용하여 비교적 제한된 시뮬레이션 횟수 하에서 목표성능을 달성할 수 있는 SARAGA라 칭하는 유전 알고리즘을 제안하였다.

가장 단순한 형태의 단봉 최적화 문제에 적용한 경우 실험결과를 통해 보면 정적 리샘플링을 이용하는 naive SARAGA는 동적인 리샘플링을 이용하는 SARAGA 대비 낮은 성능을 보임을 확인하였다. 또한 대체모델의 적절한 조합을 통해서 탐색성능을 향상시킬 수 있음을 확인하였고 특히 리샘플링과 적절한 조합함으로써 더욱 성능을 향상시킬 수 있음을 확인하였다. 또한 음향어피방어 시뮬레이션에 적용하여 최적 전술을 도출하는 목적으로 적용하여 실제 문제에 적용하였으며 비교적 높은 생존률을 보이는 전술을 도출할 수 있음을 확인하였다.

본 논문에서는 시뮬레이션 소요시간을 고려하여 총 시뮬레이션 횟수가 15000회 정도로 제한된 경우에 대하여 탐색적인 최적화(exploratory optimization)에 대하여 우수한 성능을 보임을 확인하였다. 향후 실제 최적 솔루션에 더 접근하려면 더 많은 시뮬레이션을 수행할 수 있는 분산컴퓨팅 환경 기반의 최적화 환경의 구축이 필요하며 많은 시뮬레이션 횟수에 의해 수반되는 증가된 데이터에 따른 대체모델 추정시 발생할 차원의 저주를 완화하기 위한 연구도 필요하다. 또한 다봉과 같은 좀더 복잡한 반응표면을 갖는 문제와 다목적으로 정의된 문제를 해결하기 위한 알고리즘의 연구도 필요할 것으로 보인다.

## References

- [1] T. Bäck, "Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms," Oxford University Press, Oxford, UK, 1996.
- [2] T. Bartz-Beielstein, "New Experimentalism Applied to Evolutionary Computation," Ph. D. Thesis, University of Dortmund, 2005.
- [3] Y. Jin and J. Branke, "Evolutionary Optimization in Uncertain Environments-a Survey," IEEE Transactions on Evolutionary Computation, 9(3):303-317, 2005.
- [4] R. L. Anderson, "Recent Advances in Finding Best Operating Conditions," Journal of the American Statistical Association, 1953.
- [5] E. Aarts, and J. Korst, 1989, "Simulated Annealing and Boltzmann Machines - a Stochastic Approach to Combinatorial Optimization and Neural Computing," Wiley, New York, p. 272, 1989.
- [6] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", Proc. IEEE Int. Conf. Neural Networks, pp. 1942-1948, 1995.
- [7] L. M. Gambardella and M. Dorigo, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, pp. 53-66, 1997.
- [8] D. Beasley, D. R. Bull and Martin R. R., "An Overview of Genetic Algorithms : Fundamentals," Morgan Kaufmann, 1993.
- [9] T. Back, "Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms", Oxford University Press, 1996.
- [10] Y. Akimoto, S. A. Morales, and Teytaud O., "Analysis of Runtime of Optimization Algorithms for Noisy Functions Over Discrete Codomains," Theoretical Computer Science, Vol. 605, pp. 42-50, 2015.
- [11] T. Bäck, and U. Hammel, "Evolution Strategies Applied to Perturbed Objective Functions," in Proceedings of IEEE Congress on Evolutionary Computation, pp. 40-45, 1994.
- [12] B. L. Miller, "Noise, Sampling, and Efficient Genetic Algorithms," Ph. D. Thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, TR 97001, 1997.
- [13] A. N. Aizawa, and B. W. Wah, "Scheduling of Genetic Algorithms in a Noisy Environment," Evolutionary Computation, Vol. 2, No. 2, pp. 97-122, 1994.
- [14] J. E. Fieldsend, "Elite Accumulative Sampling Strategies for Noisy Multi-Objective Optimisation," Proceedings of Evolutionary MultiCriterion

- Optimization, Springer International Publishing, pp. 172-186, 2015.
- [15] T. Park, and K. R. Ryu, "Accumulative Sampling for Noisy Evolutionary Multi-Objective Optimization," Proceedings of the ACM 13th Annual Conference on Genetic and Evolutionary Computation, pp. 793-800, 2011.
- [16] A. D. Pietro, "Optimising Evolutionary Strategies for Problems with Varying Noise Strength," Ph. D. Thesis, University of Western Australia, 2007.
- [17] Z. Zhang, and T. Xin, "Immune Algorithm with Adaptive Sampling in Noisy Environments and its Application to Stochastic Optimization Problems," IEEE Computational Intelligence Magazine, Vol. 2, No. 4, pp. 29-40, 2007.
- [18] A. D. Pietro, "Optimising Evolutionary Strategies for Problems with Varying Noise Strength," Ph. D. Thesis, University of Western Australia, 2007.
- [19] F. Siegmund, "Sequential Sampling in Noisy Multi-Objective Evolutionary Optimization," Master's Thesis, University of Skövde, School of Humanities and Informatics, 2009.
- [20] R. L. Hardy, "Multiquadric Equations of Topography and Other Irregular Surfaces," Journal of Geophysical Research, Vol. 76, No. 8, pp. 1905-1915, 1971.
- [21] J. Sacks et. al., "Design and Analysis of Computer Experiments," Statistical Science, Vol. 4, No. 4, pp. 409-435, 1989.
- [22] R. H. Myers et. al., "Response Surface Methodology: Process and Product Optimization Using Designed Experiments," Vol. 705, John Wiley & Sons, 2009.
- [23] L. L. Schumaker, "Spline Functions: Basic Theory," Cambridge Mathematical Library, Cambridge University Press, Cambridge, Uk, 3rd edition, 2007.
- [24] V. N. Vapnik, "Statistical Learning Theory, Adaptive and Learning Systems for Signal Processing, Communications, and Control," John Wiley & Sons, New York, USA, 1998.