Engineering Applications of Artificial Intelligence 65 (2017) 230–239

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Simulation-based multimodal optimization of decoy system design using an archived noise-tolerant genetic algorithm

Jeong Hee Hong [a], Kwang Ryel Ryu [b,*]

[a] Department of Computer Engineering, Dongseo University, 47, Jurye-ro, Sasang-gu, Busan, 47011, Republic of Korea
[b] Department of Electrical and Computer Engineering, Pusan National University, 2, Busandaehark-ro 63-gil, Geumjeong-gu, Busan, 46241, Republic of Korea

ARTICLE INFO

Keywords:
Simulation-based optimization
Multimodal optimization
Noisy optimization
Genetic algorithm

ABSTRACT

The difficulty of warship decoy system design problem is twofold. First, we need to find not just one but as many optimal solutions as possible. Second, it demands a heavy computation to evaluate a candidate solution through a long series of underwater warfare simulations. The previous approach tried to reduce the amount of search by heuristically selecting a set of plausible starting points for the search by a simulated annealing algorithm. However, it shows only limited success and cannot easily scale up to larger problems. This paper proposes an efficient and easy-to-scale-up multimodal optimization algorithm named A-NTGA that is based on a genetic algorithm. A-NTGA quickly evaluates candidate solutions by conducting only a small number of simulations, but instead copes with these inaccurate or noisy fitness values by using a noisy optimization technique. To further enhance the efficiency of search by promoting the population diversity, A-NTGA is provided with an archive to which some good-looking solutions are migrated in order to prevent the population from being too crowded with similar solutions. Usually at the end of the search, many optimal solutions are retrieved from the archive as well as the population. The experimental results show that our method can find multiple optimal solutions more efficiently compared to other methods and can be easily scaled up to larger problems.

1. Introduction

When solving various engineering problems in design, control, or planning, we often pursue not a single but multiple solutions. In some cases we attempt to find all globally optimal solutions, while in other cases we look for all the solutions with qualities above a certain threshold. In this paper, we deal with one such *multimodal optimization* problem in the military domain, which is the problem of designing a warship decoy system to protect the warship against enemy torpedo attacks in underwater warfare (Liang and Wang, 2006; Grootendorst et al., 2007). Torpedoes launched by an enemy submarine at close range are fatal threat because the warship is not fast enough to evade from high speed torpedoes that are also equipped with acoustic homing device. Decoy is a device that can distract the torpedo's acoustic homing function by transmitting radiated noise in order to be falsely regarded as a target for itself. Detecting a torpedo attack, a warship usually launches a few decoys into the water to have the torpedoes diverted to those false targets, during which time the warship can detour the danger area. The likelihood of survival of the warship depends on several factors including the deployment pattern (i.e., number and locations) of the decoys, their speeds, their noise source levels, and so on. For our decoy system design, we use a genetic algorithm to search for optimal values of five parameters that specify deployment pattern, launch type (drop or shoot), decoy speed, battery life, and noise source level. Each candidate design (i.e., parameter specification) is evaluated by running a series of underwater warfare simulations and then measuring the survival rate of the warship. The warship can either escape from or be hit by torpedoes in some scenarios or others in sessions of *stochastic* simulations. The survival rate refers to the ratio of the number of successful escapes to that of all the scenarios simulated. An accurate evaluation of a candidate design usually requires a number of simulations until the survival rate converges, demanding a nontrivial amount of CPU time. For our decoy system design problem, we want to find all the candidate solutions that meet the survival rate above a given threshold. Among them, only those that satisfy both technical and financial constraints after further analyses will be accepted as final designs. In this paper, our main goal is not to find these final designs requiring implementation-oriented technical constraints and cost analyses but to find just as many candidate designs as possible within a reasonable period of time.

---

[*] Corresponding author.
E-mail addresses: jhhong@dongseo.ac.kr (J.H. Hong), krryu@pusan.ac.kr (K.R. Ryu).

http://dx.doi.org/10.1016/j.engappai.2017.07.026
Received 25 February 2017; Received in revised form 22 June 2017; Accepted 26 July 2017
0952-1976/

A previous study by Hong et al. (2013) and Hong (2013) on this problem attempts to decrease the computational complexity by reducing the search space and recommending good starting points for a *simulated annealing* (SA) search. For this, the design parameter space is partitioned into subspaces, with several points in each space sampled and simulated to obtain a linear regression model that maps the sample points to a range of objective evaluations. Based on the investigation of these linear regression models, the subspaces are heuristically classified to either plausible or implausible ones, and then good initial search points in the plausible ones are again heuristically determined. In spite of this hard work to prepare for the SA search, the resulting reduction of the search space is not so satisfactory and their algorithm cannot be easily scaled up to problems with a large search space. Its weakness comes from the fact that the remaining amount of search after heuristic reduction can still be large if the size of the overall search space is very large. When an exponentially larger search space is given, the amount of search to be done in the reduced search space can also get larger exponentially.

What we propose in this paper is a more cost-effective approach to searching for multiple optimal solutions to the decoy system design problem. The approach is based on a *noisy optimization* technique that works on top of a *genetic algorithm* (GA) (Park and Ryu, 2011; Choe et al., 2013). The implementation of our approach can be considered as an extension of one such algorithm called NTGA developed by Choe et al. (2013). The most notable difference between the idea of NTGA and the method by Hong et al. (2013) and Hong (2013) is that NTGA does not thoroughly evaluate the candidate solutions as Hong's method does. Through only a few simulations, a candidate solution can still be evaluated, albeit roughly. The evaluation values obtained in this way may be inaccurate or *noisy*, but the reduction in computation time is enormous. If a rough evaluation can be made by m simulations while a thorough evaluation requires n simulations, the ratio of computation time m/n can be very small when $m \ll n$. NTGA deals with the problem of noisy evaluation by allotting more simulations to more critical-looking candidate solutions to enhance the accuracy of their evaluations. Note that NTGA does not attempt to reduce the search space, but instead smartly allocates computational resources to different candidate solutions within the search space.

The original version of NTGA does not search for multiple optimal solutions. Therefore, we extend NTGA by providing it with an *archive* that functions as a repository of plausible solutions found during the search. Since NTGA is equipped with a scheme that controls the diversity of the population to avoid premature convergence, good solutions are often discarded from the population if there exist other closely resembling solutions in the population. By selectively preserving some of those discarded solutions in the archive, the proposed algorithm can eventually harvest multiple qualified solutions not only from the population but also from the archive. The archiving scheme contributes to maintaining the diversity of population by having some good solutions migrated from crowded niches to the archive. The experimental results show that our method can find multiple optimal solutions more efficiently and more effectively compared to other methods. In a relatively small search space investigated by Hong et al. (2013), our method has found all the optimal solutions found by their method but about twice as fast. Given a much larger search space to which Hong's method cannot feasibly be applied, our method has found a considerable number of optimal solutions in a reasonable amount of time.

The remainder of this paper is organized as follows. Section 2 reviews some relevant literature on multimodal and noisy optimization techniques. Section 3 describes how we simulate the decoy operation of a warship in underwater warfare to measure its survival rate. Section 4 presents the details of the proposed method. Section 5 reports the experimental results, comparing the performance of the proposed algorithm with those of other methods. Finally, Section 6 gives some concluding remarks.

2. Related works

Multimodal optimization problems have often been dealt with by using evolutionary algorithms (EAs) because the population in EAs provides the possibility of locating multiple solutions in parallel. However, ordinary EAs are not suitable for pursuing multiple optimal solutions because they tend to be biased towards converging to a single best solution rather than maintaining the diversity of population. What extend the ordinary EAs to find multiple solutions are the *niching methods* that promote the formation and maintenance of multiple subpopulations during evolution. One of the well-known niching methods is *fitness sharing* (Goldberg and Richardson, 1987), according to which the *fitness* (i.e., the goodness of objective evaluation) value of an individual is decreased by having it shared with other individuals within the *niche*. An individual is considered to reside in the niche of another individual if the distance of the former from the latter is less than a predetermined *niche radius*. Since an individual in a crowded niche is made to look less promising than it really is due to fitness sharing, the search is diversified to other attractive region in the search space. One problem with fitness sharing is that its performance is sensitive to the niche radius value that should be determined empirically through trial and error. An appropriate value of niche radius, however, is hard to find especially under a multimodal optimization setting. Suppose all the optimal solutions are at least distance d apart in the search space. Then the niche radius of $d/2$ may easily identify all those prominent solutions. However, if some optimal solutions are very close to each other and the others are far apart, then d becomes too small and as a consequence fitness sharing does not work well because there may not be multiple individuals to share the fitness within such a small niche of radius $d/2$.

Crowding is another well-known niching method, in which each offspring (a newly born individual) competes for survival with old individuals close to it. In the crowding factor model proposed by De Jong (1975), *CF* individuals are randomly selected from the population and then the one most similar (or close) to the offspring gets replaced. Other variations of the crowding method include deterministic crowding (Mahfoud, 1995), probabilistic crowding (Mengshoel and Goldberg, 1999), and restricted tournament selection (RTS) algorithm (Harik, 1995). The differences are mainly in the selection of competitors for replacement. The idea of the crowding methods is not to allow too many similar individuals to reside together in the population, thus promoting the population diversity. Some previous studies (Sareni and Krähenbühl, 1998; Choe et al., 2013) report that the RTS algorithm shows better overall performance than the other niching methods. The NTGA algorithm based on which our algorithm has been developed is built on top of the RTS algorithm.

There have been various studies on using EAs to deal with noisy optimization problems through efficient allocation of samples for candidate evaluations. The effect of noise can be reduced by taking multiple fitness samples and averaging them. However, the computational cost of repeated *resampling* increases with the number of samples taken. The adaptive resampling methods (Aizawa and Wah, 1994; Cantú-Paz, 2004; Choe et al., 2013) dynamically adapt the number of samples during evolution for better use of limited computational resources. In particular, the duration scheduling method (Aizawa and Wah, 1994) adjusts the sample size at every generation based on the estimated noise level of the current population, assuming a *non-overlapping* generation GA. Aizawa and Wah (1994) also proposed a sample-allocation method to allocate more samples to better individuals within the population.

As mentioned above, NTGA (Choe et al., 2013) is built on top of RTS that is a *steady-state* GA based on an *overlapping-generation* model. Whenever an offspring is born, it has to compete for survival with an old individual close to it. It is at this *survival selection* step when additional samples are allotted to the old competitor to make the evaluations for survival selection more reliable, although usually a small number of samples are allotted to the newly born for its initial evaluation. The evaluation for the old individual is then renewed by averaging