Swarm and Evolutionary Computation 38 (2018) 267–274

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Computing budget allocation in multi-objective evolutionary algorithms for stochastic problems

Mengmei Liu*, Aaron M. Cramer

Department of Electrical and Computer Engineering, University of Kentucky, Lexington 40506, KY, USA

A R T I C L E I N F O

Keywords:
Computational complexity
Evolutionary computation
Gaussian noise
Genetic algorithms
Pareto analysis

A B S T R A C T

Multi-objective stochastic problems are important problems in practice and are often solved through multi-objective evolutionary algorithms. Researchers have developed different noise handling techniques to improve the efficiency and accuracy of such algorithms, primarily by integrating these methods into the evaluation or environmental selection steps of the algorithms. In this work, a combination of studies that compare integration of different computing budget allocation methods into either the evaluation or the environmental selection steps are conducted. These comparisons are performed on stochastic problems derived from benchmark multi-objective optimization problems and consider varying levels of noise. The algorithms are compared in terms of both proximity to and coverage of the true Pareto-optimal front and sufficient studies are performed to allow statistically significant conclusions to be drawn. It is shown that integrating computing budget allocation methods into the environmental selection step is better than integration within the evaluation step.

1. Introduction

Real-world optimization problems are often multi-objective and stochastic problems [1,2]. Multi-objective evolutionary algorithms (MOEAs) are widely applied to solve those problems [3]. MOEAs combine operators such as mating selection [4], which selects child genes, crossover, and mutation to construct new generations of individuals. Among MOEAs, popular elitist approaches archive non-dominated solutions from the previous generation and combine them with non-dominated solutions from the current generation to produce the subsequent generation, a process which is referred to as environmental selection [4].

Researchers have developed different noise handling techniques to solve stochastic problems, aiming to improve the accuracy and efficiency of the algorithms. For example, a probabilistic method to improve sampling using loopy belief propagation for probabilistic model building genetic programming is described in [5]. Population statistics based re-sampling technique is introduced in [6] with the particle swarm optimization algorithm to solve stochastic optimization problems. For the elitist MOEAs, because of the stochastic nature of the objective functions, MOEAs must perform repeated computationally expensive samples in order to assess the fitness of each individual. The noise handling techniques seek to obtain more accurate results with fewer total evaluations. For example, the optimal computing budget allocation (OCBA) method proposed in [7] is integrated into the evaluation procedure to reduce the computing cost in [8]. In [9], fitness inheritance from parent genes is proposed to reduce the computational intensity required for evaluation. A probabilistic method based on statistical analysis of dominance is used to estimate Pareto-optimal front in [10]. In [11], confidence-based dynamic re-sampling is proposed to improve the confidence of Pareto ranking. A reliability-based optimization method which utilizes mathematical approximations of a solution's reliability to integrate into evolutionary algorithms is described in [12]. Optimal design considering the worst-case scenario for safety can be approached by applying anti-optimization factors into stochastic optimization problems [13]. A noise-aware dominance operator is integrated into the mating selection in [14]. However, due to the nature of the elitist MOEAs, environmental selection plays a more critical role than mating selection because it controls the evolving set of non-dominated solutions [15].

Herein, a fundamental question regarding the application of computing budget allocation (CBA) methods to MOEAs is considered. The CBA methods refer to methods of allocating a fixed number of total samples to a pool of individuals in the solution set of a stochastic problem. The effect of integrating CBA methods in either the evaluation or environmental selection on the accuracy of the MOEA is examined. In previous work with re-sampling applied as a noise handling technique, it is proposed either in evaluation [8] or in environmental selection [11]. However, there is no clear comparison between these two techniques; the comparisons focused only on whether the propo-

* Corresponding author.
E-mail address: mengmei.liu@uky.edu (M. Liu).

http://dx.doi.org/10.1016/j.swevo.2017.08.004
Received 16 August 2016; Received in revised form 18 July 2017; Accepted 18 August 2017
Available online 24 August 2017
2210-6502/

sals improved the results or not. There exists no comprehensive study that examines, for a fixed total computing budget, where the re-sampling procedure should be integrated to best improve the algorithms. A previous work in [16] proposed integration of CBA techniques into selection procedure instead of evaluation procedure of genetic algorithm for single-objective stochastic problems and showed that the selection integration method greatly improved the accuracy of the algorithm. Inspired by this work, a combination of studies that compare the alternative approaches to integrating CBA methods into genetic algorithms (GA), namely evaluation-integrated GA (EIGA) and selection-integrated GA (SIGA), are described. It is shown that the SIGA outperforms the EIGA statistically for the reason that the SIGA allocates fitness evaluations toward specific individuals when the algorithm needs more information for evolving. Various CBA techniques are compared, including the most basic equal allocation (EQU) method [17], OCBA method [7,18], and the proportional-to-variance (PTV) method [19]. These algorithms and CBA techniques are applied to stochastic multi-objective problems constructed from benchmark multi-objective optimization problems [20–22]. Numerical experiments are performed, and statistical testing is used to validate the significance of the comparisons. The results suggest two significant findings: 1) applying SIGA other than EIGA, the generational distance (GD) metrics had improved for at least one decimal level to up to two decimal levels with statistical significance for all the test cases, which implies that the SIGA method produces more accurate front when solving multi-objective stochastic problems; 2) even though OCBA method is a better allocation method for correct selection from a static pool [7], it performs worse than the EQU and PTV allocation methods when integrated into an elitist MOEA where the pool changes dynamically from generation to generation.

The remainder of this paper is organized as follows. First, descriptions of the considered stochastic multi-objective problems and different CBA methods are given in Section 2. In Section 3, the structure of the EIGA and SIGA approaches for integrating CBA methods into GAs are described. Test functions, performance metrics, and experimental results are presented in Section 4. Conclusion for this study is in Section 5.

2. Multi-objective stochastic problems and computing budget allocation methods

The multi-objective stochastic problems and the CBA methods considered herein are described below.

2.1. Stochastic problem statement

The multi-objective stochastic problems considered herein can be defined as

$$\min_X J_1(X),\ J_2(X), \ldots, J_H(X), \tag{1}$$

where X is the (possibly multi-dimensional) decision variable, J_1, J_2, ..., and J_H are the H objectives to be minimized, $J_l(X) = E[L_l(X, \xi)]$, $l \in \{1, 2, \ldots, H\}$, $L_l(\cdot,\cdot)$ is the sample performance of lth objective, and ξ is a random variable describing the problem noise. For the test problems considered herein, Noise can be introduced in variable values or environment levels and can follow different kinds of distributions. The stochastic nature of problems varies from case to case, depending on application details and difficult to generalize. For the problems considered herein, the noise for each objective is modeled with additive independent and identical Gaussian distributions, with the noise sampled at each function evaluation. It is also assumed that such a problem is unconstrained in the sense that any constraints that bind the solution are appropriately penalized in $L(\cdot,\cdot)$.

For a deterministic multi-objective problem where $J_l(X) = L_l(X)$, the Pareto optimal front [23] is the complete set of non-dominated solutions. A solution for the multi-objective problem is defined as a non-dominated solution if it is not dominated by any other solutions. A solution a dominates solution b if $J_l(a) \le J_l(b)\ \forall\ l \in \{1, 2, \ldots, H\}$ and $\exists\ l \in \{1, 2, \ldots, H\}$ such that $J_l(a) < J_l(b)$. For the non-dominated solutions, each objective is minimized to the extent that it is not possible to further minimize one objective without making one or more other objectives bigger (worse).

For a stochastic multi-objective problem, the fitness function of each objective can only be estimated by a limited number of random samples. Mean value from the random samples is used to evaluate the fitness in this study. For a certain number of realistic problems, it is assumed that the evaluation of the samples takes far more computation time and effort than the algorithm itself. It is desired to be able to allocate the samples, or the total computing budget, appropriately to obtain the best approximation of the fitness function when evaluates, thus to obtain the best approximation of the Pareto optimal front when the search terminates. The quality of a Pareto optimal front approximation is measured by both its proximity to the true front and the degree to which it covers the true front.

2.2. Computing budget allocation methods

The CBA methods determine, given a total number of samples, how samples should be allocated to each individual in the solution set. The motivation of applying CBA methods to a stochastic problem is to improve the accuracy of solving a stochastic problem and avoid wasting samples on unwanted individuals. Various CBA methods have been studied, and the three such CBA methods applied in this study are discussed below. For each of these methods, it is assumed that N samples are being allocated among k individuals.

2.2.1. Equal allocation method

The simplest allocation technique to conduct sampling is the EQU technique, and it often serves as a benchmark for comparison [24]. The available computing budget is equally distributed to all individuals:

$$\tilde{n}_i = \frac{N}{k}, \tag{2}$$

where $\tilde{n}_i$ is the number of additional samples to be allocated to individual i.

2.2.2. Optimal computing budget allocation method

The OCBA method [7] for multi-objective optimization is based on maximizing the asymptotic probability that the selected subset is the non-dominated set. One such implementation is described below.

For a set of unique individuals S, S_P is defined as the non-dominated set, and S_D is defined as the dominated set. In deterministic problems, S_P and S_D can be determined by non-dominated sorting [25]. The OCBA allocation rule aims to maximize the probability of correctly selecting the Pareto optimal set in stochastic problems.

For an individual i, which has previously been sampled, $\overline{L}_{il}$ is the sample mean, and σ_{il}^2 denotes the sample variance corresponding to the lth objective. For two individuals, i and j, the difference of sample means for objective l is expressed as

$$\delta_{ijl} = \overline{L}_{jl} - \overline{L}_{il}. \tag{3}$$

The individual that dominates i with the highest probability is approximated as

$$j_i \approx \arg\max_{j \in S, j \ne i} \prod_{l=1}^{H} P(L_{jl} \le L_{il}) \approx \arg\min_{j \in S, j \ne i} \frac{\delta_{ijl_j^i}\left|\delta_{ijl_j^i}\right|}{\sigma_{il_j^i}^2 + \sigma_{jl_j^i}^2}, \tag{4}$$

where l_j^i denotes the objective for which j is better than i with the lowest probability and can be calculated as