



A parameter control method of evolutionary algorithms using exploration and exploitation measures with a practical application for fitting Sovova's mass transfer model



Shih-Hsi Liu^{a,*}, Marjan Mernik^b, Dejan Hrnčič^b, Matej Črepinšek^b

^a California State University, Fresno, Department of Computer Science, 2576 E San Ramon Drive, Fresno, CA 93740, USA

^b University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova 17, 2000 Maribor, Slovenia

ARTICLE INFO

Article history:

Received 30 October 2012

Received in revised form 5 March 2013

Accepted 8 May 2013

Available online 6 June 2013

Keywords:

Exploration

Exploitation

Parameter control

Sovova model

ABSTRACT

Exploration and exploitation are omnipresent terms in evolutionary computation community that have been broadly utilized to explain how evolutionary algorithms perform search. However, only recently exploration and exploitation measures were presented in a quantitative way enabling to measure amounts of exploration and exploitation. To move a step further, this paper introduces a parameter control approach that utilizes such measures as feedback to adaptively control evolution processes. The paper shows that with new exploration and exploitation measures, the evolution process generates relatively well results in terms of fitness and/or convergence rate when applying to a practical chemical engineering problem of fitting Sovova's model. We also conducted an objective statistical analysis using Bonferroni–Dunn test and sensitivity analysis on the experimental results. The statistical analysis results again proved that the parameter control strategy using exploration and exploitation measures is competitive to the other approaches presented in the paper. The sensitivity analysis results also showed that different initial values may affect output in different magnitude.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Exploration and exploitation [10,14] are two essential cornerstones of evolutionary algorithms (EAs) [4,17,22,38] that drive an evolution process toward optimization and/or convergence. In fact, these two processes are essential for search processes when using any metaheuristic approach [6,7]. Exploration is defined as visiting entirely new regions of a search space, while exploitation is defined as visiting those regions of a search space within the neighborhood of previously visited points [10]. However, to our best knowledge, not until exploration and exploitation measures using an ancestry tree approach were recently introduced by Črepinšek et al. [9], there had not been a *quantitative* way to measure exploration and exploitation and analyze how these essential cornerstones influence and balance the inner work of an evolution process. Our previous work in Ref. [9] primarily focused on introducing exploration and exploitation measures. In Ref. [33] we further applied

such measures to investigate and explain the inner work of VEGA [43] and SPEA2 [49].

Given the usefulness on investigating the inner work of EAs using exploration and exploitation measures, this paper attempts to validate a hypothesis: by using finer-grained exploration and exploitation measures as feedback, an evolution process adapted by parameter control approaches [15] may perform relatively competitive or generate even better results in terms of optimization and/or convergence on a selected practical chemical engineering problem. To validate the hypothesis, this paper extends the implementation of an existing domain-specific language [37], called PPCea (Programmable Parameter Control for Evolutionary Algorithms) introduced by Liu et al. [31,32], so that all exploration and exploitation measures from Črepinšek et al. [9] can be computed by the PPCea interpreter on-the-fly. With such, users may introduce PPCea programs to adaptively control evolution processes using the measures as feedback.

The paper is organized as follows. Section 2 reviews the ancestor-tree approach. Section 3 presents how PPCea can be used to (re)produce the parameter tuning and four parameter control algorithms, including the one controlled by exploration and exploitation measures. A practical Chemical Engineering problem is shown in Section 4. The experimental results of the four parameter

* Corresponding author. Tel.: +1 5592784789.

E-mail addresses: shliu@CSUFresno.edu (S.-H. Liu), marjan.mernik@uni-mb.si (M. Mernik), dejan.hrnccic@uni-mb.si (D. Hrnčič), matej.crepinsek@uni-mb.si (M. Črepinšek).

control strategies and two parameter tuning approaches are presented in Section 5, followed by the conclusion in Section 6.

2. The ancestry tree approach

Exploration and exploitation are fundamental concepts of any search algorithm. Despite this fact, exploration and exploitation are still not well understood by practitioners and researchers [14]. To remedy this situation Črepinšek et al. [10] introduced a fresh treatment of exploration and exploitation in EAs discussing:

- what parts of EAs contribute to exploration and exploitation: selection, variation operators, population size, and representation;
- how balance between exploration and exploitation is achieved: implicitly by parameter tuning and parameter control using uni-process or multi-processes driven approaches;
- when balance between exploration and exploitation should be controlled: online using deterministic, adaptive, or self-adaptive approaches; and
- how balance between exploration and exploitation can be controlled: by diversity maintaining, by diversity control, by diversity learning, or by other direct approaches.

Although numerous measures and means were summarized and classified in Ref. [10], *direct* measures to analyze exploration and exploitation barely exist. Hence, controlling exploration and exploitation is up to now mostly *indirect* in EAs (e.g., by varying different control parameters or by maintaining diversity without explicitly measuring exploration and exploitation). To the best of our knowledge, the only direct measures for exploration and exploitation is our own work: an ancestry-based approach presented in Ref. [9]. We have used such measures (e.g., *exploreRatio*, *exploreGap*) defined in Ref. [9] to directly measure exploration and exploitation ability of a few EAs. Particularly, in Ref. [33] the analysis and comparison between Vector Evaluated Genetic Algorithm (VEGA) [43] and Strength Pareto Evolutionary Algorithm 2 (SPEA2) [49] were performed using the recently introduced exploration and exploitation measures, where it was shown that the inner work of both multi-objective algorithms can be observed and explained in a refined way in terms of exploration and exploitation. However, in this case exploration and exploitation measures were computed *off-line* from data collected during the run of the algorithms. To be able to perform adaptive parameter control with the proposed exploration and exploitation measures they need to be computed on-line during the run of algorithms. Preliminary results of on-line computation of exploration and exploitation measures and adaptive parameter control with exploration and exploitation measures has been presented by Liu et al. [34], where the results were shown only on functions f_2 and f_{14} from the benchmark test suite in Yao et al. [48]. This paper is an extension of the paper [34], where adaptive parameter control with exploration and exploitation measures has been now applied to a practical chemical engineering problem. With newly proposed adaptive parameter control using exploration and exploitation measures, the performance of Differential Evolution (DE) [45] for fitting Sovova's model [44] has been improved. But, before showing the results (see Section 5) the ancestry tree based approach for directly measuring exploration and exploitation is briefly introduced for self-contained purpose. For more information, readers are directed to [9,10]. In this paper the following notation is used. An ancestry tree originated from i th individual from initial population is denoted as τ_i , where $i \in [0, \dots, pop_size - 1]$. The entire ancestry trees are denoted with τ , where $\tau = \{\tau_0, \dots, \tau_{pop_size-1}\}$. A j th exploitation tree originated from i th individual from initial population is denoted with τ_{ij} . An

exploration tree originated from i th individual from initial population is denoted as $exploreTree(\tau_i)$, where $i \in [0, \dots, pop_size - 1]$. When threshold value X , important for splitting an ancestry tree, is also mentioned an ancestry tree, exploitation tree, and exploration tree are denoted as $\tau_i^{(X)}$, $\tau_{ij}^{(X)}$, $exploreTree(\tau_i^{(X)})$, respectively.

The ancestry tree approach borrows the ideas from genealogy. Ancestry trees describe the history of all individuals that were created during an evolution process: an individual's parent, its representation (genome), as well as how (e.g., by mutation, by crossover) and when (generation) the individual was created are all stored during the construction of ancestry trees. For example, in Fig. 1(left), τ_2 represents the ancestry tree originated from the 2nd individual generated at the initial stage. During each generation, if a candidate child (generated by mutation ($T=m$), crossover ($T=c$), random creation ($T=rd$), repair ($T=r$), or cloning ($T=cln$), etc.) survives after selection, the ancestry tree will be expanded to further depth until no more offspring survives. Note that building such an ancestry tree is similar to the approach presented by Davis [11], where whenever a new member was added to the population, a pointer was established to its parents, as well as another pointer was established to the operator that had created the new member. Davis [11] used such a structure to adaptively compute operator probabilities. While in our approach an ancestry tree is used to compute an amount of exploration/exploitation. But, how can we see which node is a result of exploration or exploitation? In Ref. [9], a threshold variable X is introduced to delimit exploration from exploitation and can be seen as identification of an individual's neighborhood. Because X is problem dependent, various diversity measures (e.g., hamming distance and Euclidian distance) can be used for this purpose. When the selected diversity measure is larger than X , the splitting process will split an ancestry tree τ_i into several subtrees (see Fig. 1(middle)). Each particular subtree is called exploitation tree τ_{ij} , because all nodes, except the root node, are obtained by exploitation of neighborhood regions. Such a process can be observed in Fig. 1: There are six exploitation trees ($\tau_{2,0}, \dots, \tau_{2,5}$, in Fig. 1(middle)). For each exploitation tree, gray colored nodes represent exploitation nodes (obtained by exploiting within the individual's neighborhood) and black root node represents exploration node (obtained by exploring outside the neighborhood). As for exploration tree $exploreTree(\tau_i)$, it is constructed by linking all the black root nodes together (see Fig. 1(right)). With such, a number of exploration and exploitation measures can be defined based on the characteristics of exploration and exploitation trees.

Careful readers might notice that we did a slight simplification, as every individual in ancestry tree τ_i , except in initial generation, has exactly one parent called a dominant parent. A dominant parent of individual ind_i is an individual selected from τ_i that passes a genetic material to its offspring. In the case when more parents contribute genetic materials to an individual (e.g., as a result of crossover), a parent with most similarities to the individual is selected as dominant parent. Every individual that has not been selected as dominant parent is represented as a leaf in an ancestry tree. It is also clear from Fig. 1 that fitter individuals, those who are selected for next generations, contribute to growth of an ancestry tree. From now, let $size()$ represent the number of nodes of an ancestry/exploitation/exploration tree. The number of exploitation trees obtained from all ancestry trees using the threshold X is:

$$count(X) = \sum_{i=0}^{pop_size-1} splits(\tau_i^{(X)}) \quad (1)$$

where $splits(\tau_i^{(X)})$ represents the number of the splitting processes triggered by the threshold X in ancestry tree $\tau_i^{(X)}$, as can be seen in

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات