



Adaptive parameter control of evolutionary algorithms to improve quality-time trade-off

Sandip Aine, Rajeev Kumar^{*}, P.P. Chakrabarti

Department of Computer Science & Engineering, Indian Institute of Technology Kharagpur, WB 721302, India

ARTICLE INFO

Article history:

Received 14 June 2006

Received in revised form 29 July 2007

Accepted 27 July 2008

Available online 5 August 2008

Keywords:

Evolutionary algorithm

Parameter control

Meta-reasoning

Quality-time trade-off

Dynamic programming

Traveling salesman problem

Standard cell placement problem

ABSTRACT

Parameter control of evolutionary algorithms (EAs) poses special challenges as EA uses a population and requires many parameters to be controlled for an effective search. Quality improvement is dependent on several factors, such as, fitness estimation, population diversity and convergence rate. A widely practiced approach to identify a good set of parameters for a particular class of problem is through experimentation. Ideally, the parameter selection should depend on the resource availability, and thus, a rigid choice may not be suitable. In this work, we propose an automated framework for parameter selection, which can adapt according to the constraints specified. To condition the parameter choice through resource constraint/utilization, we consider two typical scenarios, one where maximum available run-time is pre-specified and the other in which a *utility* function modeling the quality-time trade-off is used instead of a rigid deadline. We present static and dynamic parameter selection strategies based on a probabilistic profiling method. Experiments performed with traveling salesman problem (TSP) and standard cell placement problem show that an informed adaptive parameter control mechanism can yield better results than a static selection.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

The rate of convergence of evolutionary algorithms (EAs) is strongly influenced by the choice of certain parameters, such as population size [1], and mutation [2–4] and crossover probabilities [5], collectively termed as control parameters of the algorithm. In the past, a considerable amount of effort has been put to devise strategies for choosing a good of parameter configuration to improve the performance of evolutionary algorithms [6]. However, it has been shown that it is not possible to find a control parameter setting which works across all problem domains [7]. This observation prompted researchers to concentrate on particular classes of problems. Generally, researchers experiment with a set of problems from a particular domain and tune the control parameters on the basis of such experimentation. Another approach is to select the parameter configurations dynamically using some adaptive equations [2,3,8] or a priori knowledge [9]. Some researchers have also proposed meta-genetic algorithms [10] or fuzzy logic based techniques [11,12] to select an appropriate parameter setting. Eiben et al. [6] presented a survey

of the different parameter selection strategies. He has distinguished two broad classes of parameter setting strategies, namely, *parameter tuning* [10] where the parameters are chosen before the run of an algorithm and the selection is not revised afterward, and *parameter control* where the parameters can be adapted dynamically at run-time. Adaptive [2,3,13] and self-adaptive [14] strategies fall in the second category. Eiben et al. [6] argued that EAs are implicitly dynamic in nature. Therefore, the use of a static parameter setting is in contrast to the general evolutionary spirit and an adaptive strategy is preferable.

Whenever a parameter selection or an adaptation strategy is proposed, the standard methodology is to measure the performance of an algorithm under several parameter choices, and to report the best possible configuration. Generally, the objective is to maximize the solution quality within the given limits of time/resources, or to minimize the time/resources when a given quality target is specified. In some cases, a trade-off between the obtained quality and the required computational effort is targeted. In real-life situations, the limits on the available resources (or the intended trade-off) may vary for different environments. Thus, a selection of parameters reported, is only useful when the experimental assumptions (reported by the researchers) and the real constraints (for the users) match. For example, if a parameter configuration choice C is reported to perform best for a given time limit T_1 (for a given class of problems), there is no guarantee that C

^{*} Corresponding author.

E-mail addresses: sandip@cse.iitkgp.ernet.in (S. Aine),
rkumar@cse.iitkgp.ernet.in (R. Kumar), ppchak@cse.iitkgp.ernet.in
(P.P. Chakrabarti).

will still be the best choice if the time given is T_2 . Similarly, an adaptation strategy which gives good performance under a particular constraint may not be suitable for others.

Anytime algorithm [15,16] and *Flexible computation* [17] based methods have been proposed to address the problem of quality-time trade-off. These algorithms can be interrupted at any stage of execution, yielding some (probably sub-optimal) solution, thus, provide opportunities to reason about the convergence. EAs, by nature, are *anytime*, since each point in the search space is essentially a complete solution. Using these anytime algorithms, a systematic approach based on meta-reasoning to handle the quality-time trade-offs has been proposed by researchers in Artificial Intelligence (AI) [15,17,18].

The meta-reasoning frameworks use past data based on some sort of profiling to take informed decisions on time allocation to problem instances [18–20]. Quality-time trade-off is generally modeled by a utility function [15,21] that combines both quality and time. The meta-controllers try to take informed decisions on the time allocation so as to maximize the expected utility. The frameworks are either static [16,21], where the total time allocation is decided before the run, or adaptive [17,18,22], where the progress of the algorithm is monitored at intervals and a decision of whether to stop or allocate some more time is taken based on the amount of progress. Hansen and Zilberstein [18] proposed dynamic programming schemes for adaptive time deliberation of anytime algorithms, optimizing the utility of computation. They have shown that under uncertainty, an adaptive allocation is expected to perform better than a static allocation. Finkelstein and Markovitch [23] also studied the problem of monitoring the anytime algorithms and proposed an optimal monitoring schedule based on the quality distribution. Some studies on design-to-time scheduling [24,25] has also been directed in this direction of trade-off control. In general, these methods do not control any parameter other than the execution time. For EAs, we identify the need to develop methods that can control other parameters of the underlying algorithm being adjusted (depending on its progress) in addition to execution time, to optimize the trade-off. Aine et al. [26,27] showed how the quality-time trade-off is improved using parameter control along with time adjustment for Simulated Annealing (SA) algorithm.

Fukunaga [28] suggested an anytime portfolio technique (adapted from the work presented in [29]) to select appropriate control parameters and showed that a portfolio of several independent EA runs with different parameters can outperform a set of restarts using a single best control choice. However, the methodology suggested is intrinsically a static one as no revision of choices is performed during a particular run. Due to the stochastic nature of EAs, the convergence may vary from one run to another. As it is intuitively obvious that different values of parameters are suitable for different stages [6] of the algorithm, a dynamic strategy may be preferred over such static selections. More importantly, irrespective of whether static or dynamic selection methodology is used, the parameter choices will certainly be dependent on the amount of time in hand (or the trade-off equation).

In this paper, we present methodologies for parameter selection/adaptation conditioned with run-time constraints. We attempt to formulate a meta-level reasoning framework for parameter selection that can possibly adapt according to the given time constraints. We use a profile based methodology to select the appropriate parameter settings (crossover and mutation probabilities) when a given constraint is specified. We discuss both static (parameter tuning) and adaptive (parameter control) frameworks and compare their performances.

In our formulation, we attempt to model a steady-state genetic algorithm [30] as an interruptible *anytime* algorithm and devise a strategy for parameter selection based on the pre-computed profiles. For the static parameter selection, the profile information is used to choose the best possible parameter configuration from a set of choices depending on the constraint specification. For adaptive strategies, a particular run of EA is interrupted at pre-defined intervals and the parameter vector is adjusted monitoring the current state of the algorithm.

We also consider the scenarios where there is no hard deadline (i.e., run-time constraints) specified, instead a trade-off between the quality of solutions and the computational effort spent is targeted. A popular approach to model the quality-time trade-off is to define a utility function [15,17] which combines both quality and time. For utility based scenarios, the meta-controllers should try to take informed decisions on the time allocation as well as the parameter configurations so as to maximize the expected utility. For this purpose, we enhance the parameter controlling model to include time, and present an integrated meta-level framework which can decide both time allocation and parameter choice in a way that the specified utility of computation is optimized.

In our approach, the parameter selections/adjustments are done depending on the profile of the EA. These profiles are created with an initial set of parameter vectors (crossover and mutation probability choices). Essentially, the parameter space for an evolutionary algorithm is an unbounded one, i.e., we may have any number of parameter choices in our initial set. Though, there are some studies pointing to probable parameter options which can be used as guidance for choosing an appropriate initial set, there are no formal methodologies to select a more probable choice set over others. On the other hand, both space and time complexities of the meta-reasoning algorithm are dependent on the cardinality of the parameter choice set. To reduce this overhead, we develop a concept of dominance among control parameter vectors and show how it can be effectively used to reduce the storage (as well as time complexity) required, without sacrificing quality of the decision.

We evaluate the efficacy of the methodology using a standard steady-state genetic algorithm to two optimization problems, namely, the traveling salesman problem (TSP) and the standard cell placement problem. For TSP, we use integer encoding for the chromosomes and permutation based crossover strategies [31]. For standard cell placement, binary chromosome encodings are used along with cyclic crossover and pairwise interchange mutation strategies (as described in [32]). The experimental results show that a profile based dynamic adaptive strategy can significantly outperform the profile based/meta-genetic static selection strategies, for both the time constrained and the utility based scenarios.

The rest of the paper is organized as follows. In Section 2, we present the basic profiling framework and the modifications required for profiling EAs. Sections 3 and 4 describe the static and dynamic parameter control frameworks, for time constrained and utility based scenarios, respectively. In Section 5 we describe the dominance relation among control parameter configurations and show how it can be used to reduce the profile data. We present the experimental results in Section 6 and conclude in Section 7.

2. Profiling evolutionary algorithms

The basic problem for meta-level controller, we address in this work, is to decide on a control parameter setting (also the time allocation for utility based framework) for EAs, which optimizes the chosen objective function, under the specified constraints. For

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