



An adaptive parameter binary-real coded genetic algorithm for constraint optimization problems: Performance analysis and estimation of optimal control parameters

Omar Arif Abdul-Rahman^{a,*}, Masaharu Munetomo^b, Kiyoshi Akama^b

^a Graduate School of Information Science and Technology, Hokkaido University, Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido 060-0814, Japan

^b Information Initiative Center, Hokkaido University, North 11, West 5, Sapporo, Hokkaido 060-0811, Japan

ARTICLE INFO

Article history:

Received 29 November 2011

Received in revised form 5 January 2013

Accepted 13 January 2013

Available online 31 January 2013

Keywords:

Binary coded GA (BGA)

Real coded GA (RGA)

Hybrid scheme

Design of experiment

ABSTRACT

Real parameter constrained problems are an important class of optimization problems that are encountered frequently in a variety of real world problems. On one hand, Genetic Algorithms (GAs) are an efficient search metaheuristic and a prominent member within the family of Evolutionary Algorithms (EAs), which have been applied successfully to global optimization problems. However, genetic operators in their standard forms are blind to the presence of constraints. Thus, the extension of GAs to constrained optimization problems by incorporating suitable handling techniques is an active direction within GAs research. Recently, we have proposed a Binary Real coded Genetic Algorithm (BRGA). BRGA is a new hybrid approach that combines cooperative Binary coded GA (BGA) with Real coded GA (RGA). It employs an adaptive parameter-based hybrid scheme that distributes the computational power and regulates the interactions between the cooperative versions, which operate in a sequential time-interleaving manner. In this study, we aim to extend BRGA to constrained problems by introducing a modified dynamic penalty function into the architecture of BRGA. We use the CEC'2010 benchmark suite of 18 functions to analyze the quality, time and scalability performance of BRGA. To investigate the effectiveness of the proposed modification, we compare the performance of BRGA under both the original and the modified penalty functions. Moreover, to demonstrate the performance of BRGA, we compare it with the performance of some other EAs from the literature. We also implement a robust parameter tuning procedure that relies on techniques from statistical testing, experimental design and Response Surface Methodology (RSM) to estimate the optimal values for the control parameters to secure a good performance by BRGA against specific problems at hand.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

Many optimization problems in science and engineering involve constraints. In general, a real-parameter constraint optimization function can be formulated as follows:

* Corresponding author. Address: Sapporo International House #1102, 6-5-35 Toyohira 6-jo, Toyohira-Ku, Sapporo-shi, Hokkaido 062-0906, Japan. Tel.: +81 (0)80 3294 8078.

E-mail addresses: omar@ist.hokudai.ac.jp (O.A. Abdul-Rahman), munetomo@iic.hokudai.ac.jp (M. Munetomo), akama@iic.hokudai.ac.jp (K. Akama).

$$\text{Minimize } F(X), \quad \text{where } X = (x_1, x_2, \dots, x_n) \text{ and } X \in S, \quad (1)$$

$$\text{Subject to: } g_i(X) \leq 0, \quad i = 1, \dots, p, \quad (2)$$

$$H_j(X) = 0, \quad j = p + 1, \dots, m.$$

Inequality constraints are usually transformed into inequalities in the following form:

$$\text{Minimize } F(X), \quad \text{where } X = (x_1, x_2, \dots, x_n) \text{ and } X \in S, \quad (3)$$

where X is a vector of a real number candidate solution and S is a continuous domain design space. A solution X is regarded as feasible when the constraints in Eq. (2) are satisfied. In this article, the value of the resolution is set to 0.0001, as recommended by Mallipeddi and Suganthan [46]. Constrained optimization problems involve a variety of great challenges arising from the various limits on the decision variables, the constraints involved, the interference among constraints and the inter-relationship between the constraints and the objective functions [67].

Genetic Algorithms (GAs) are efficient search metaheuristics. Two main concepts of evolution, natural selection and genetic dynamics, inspired the development of GAs. Basic principles of GAs were laid down by Holland [38] and his colleagues in 1975 and were elaborated in detail by Goldberg [29]. Currently, GAs play an increasingly important role in a variety of fields and applications such as bioinformatics, computational science, engineering, economics, chemistry, manufacturing and other fields. However, constrained optimization problems pose a challenge to the performance of GAs, as GAs in their original versions (i.e., the ones that use traditional crossover and mutation operators) are blind to constraints. Therefore, it is very likely that the candidate solutions generated by GAs during the search process would violate certain constraints.

Researchers have recently developed several constrained handling techniques to maintain the applicability of GAs to constrained optimization problems and prevent the genetic search from leaving the feasible region. Penalty functions, repair algorithms, separating the objective and constraints, and using special representations and operators are good examples of such constrained handling techniques. Coello Coello [16] reviews the comprehensive literature regarding constrained handling techniques in EAs. Penalty functions, however, represent one of the earliest and most commonly used approaches within the EA community. Courant first proposed penalty function in the 1940s [17], and Carroll [13] and Fiacco and McCormick [28] later expanded it. The treatment of constraints in penalty methods happens by converting the constrained optimization problem to an unconstrained one by augmenting the constraints to the objective function as a penalty term. When the solution is infeasible, the objective value is penalized using a pre-determined penalty, i.e., the degree of constraint violation. This method allows retaining some genetic information from the infeasible solutions so that they might contribute to improving later solutions.

We have recently proposed a hybrid Binary-Real coded Genetic Algorithm (BRGA) [1,2] for real parameter optimization. Fig. 1 shows a schematic diagram that clarifies the BRGA operation. BRGA employs a hybrid scheme that organizes the interactions and divides the computational power between two cooperative GA versions, i.e., the Binary-coded GA part (BGA part) and the Real-coded GA part (RGA part). The evolutionary search is primarily guided by the BGA part, which is used to identify promising regions within the search space. According to the schema theory [31], using a low cardinality alphabet is responsible for BGA's success in finding good solutions. In our proposed hybrid scheme, the BGA part plays a vital role in scanning the search space effectively and partitioning it into smaller potentially promising parts that the RGA part can later exploit. Whenever the BGA part completes its shares of the genetic search, the computational power switches from the BGA part to the RGA part in a process that can be best described as population handover.

During population handover, the BGA part injects some of its best population members into the RGA part of the population solutions. Another sample of injected members is randomly generated from the region bounded by those best members. Population handover is thus an opportunity for the RGA part to focus its search by receiving valuable feedback from the BGA part about the promising regions within the search space. The RGA part, which can directly process problem variables, can also effectively compensate for the poor ability of BGA to find accurate solutions from the limitations of mapping from the real problem space into the binary representation space and vice versa.

Starting from a population dominated by promising members, the RGA part extensively exploits these regions and adapts population members to identify higher quality solutions. Whenever the RGA part completes its share of a genetic search, the computational power switches back to the BGA part in a reverse population handover process. The RGA part injects a portion of its best population members into the BGA part solution population. The handover process is thus an opportunity for the BGA part to correct its behavior in scanning and partitioning the search space by receiving valuable feedback from the RGA part about the optimality of the regions considered promising in the previous cycle. The evolutionary search continues similarly until it satisfies the termination criteria.

To regulate the share of each part from the total computational power, the handover process frequency, and the amount and type of information exchange, BRGA employs a hybrid scheme that relies on selection criteria and a group of static, dynamic and adaptive parameters. We have successfully applied BRGA to global optimization problems [3]. In this study, we aim to extend BRGA to real parameter-constrained optimization problems by incorporating a modified dynamic penalty mechanism within the algorithm architecture.

One main critique against the penalty method is the difficulty of deriving good values for the penalty factors. If the penalty value is too high and the optimum lies at the boundary of the feasible region, the algorithm is quickly pushed inside the feasible region and cannot move back towards the boundary with the infeasible region. High penalty factors can discourage the algorithm from exploring a larger part of the infeasible region from the beginning of the search.

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