A Novel Co-evolutionary Approach for Constrained Genetic Algorithms

Emmanuel Kieffer,
Mateusz Guzek
University of Luxembourg
firstname.name@uni.lu

Grégoire Danoy,
Pascal Bouvry
University of Luxembourg
firstname.name@uni.lu

Anass Nagih
Université de Lorraine
firstname.name@univ-lorraine.fr

ABSTRACT

In this paper, a novel type of co-evolutionary algorithm based on constraints decomposition (CHCGA) is proposed. Its principle consists in dividing an initial constrained problem into a sufficient number of sub-problems with weak constrained domains where feasible solutions can be easily determined. One sub-population for each sub-problems are then evolved independently and merged when they become compatible with each other, i.e. they contain enough mutually feasible solutions. Experimental results on the Cloud Brokering optimization problem have demonstrated a strong solution quality gain compared to a standard genetic algorithm.

Keywords

Constraints Handling, Co-evolutionary Algorithms, Problem Decomposition

1. INTRODUCTION

Evolutionary algorithms are nature-inspired algorithms which have proved their efficiency on a large number of free optimization problems. In this paper, the problem of constraints handling by genetic algorithms (GAs) is tackled since it still represents an important challenge. Most of population-based algorithms, such as GAs, start from a random initial population where solutions can be far from the feasible decision set. Therefore, during the first evaluations, they waste time to reach it. In this case, penalty functions can be used to speed up this phase. However a large penalty factor may force solutions to get closer to the border of the decision set while a too small penalty factor does not ensure to find feasible solutions at all. Furthermore, we can sometimes observe that feasible solutions with poor fitness performance are often rejected while unfeasible ones close to the boundaries are kept. A promising approach would be to start directly with an initial population having a reasonable proportion of feasible solutions while penalizing new unfeasible ones. By doing so, some knowledge on the decision set is introduced very early in the population and transmitted to the next generations. The question of starting with some feasible solutions in the population is not trivial at all when problems are strongly constrained. That is the reason why we propose to devote the first generations of a genetic algorithm to evolve the initial population in order to gain feasible solutions without focusing on the fitness function. In the proposed algorithm, feasibility is a characteristic that has to be achieved first. Thus, the initial population of a GA should contain a number of feasible solutions even if their fitness is weak.

GECCO '16 July 20-24, 2016, Denver, CO, USA

ACM ISBN 978-1-4503-4323-7/16/07.

DOI: http://dx.doi.org/10.1145/2908961.2908969

2. PROPOSED METHODOLOGY

In this paper, we propose the Constraint Handling Co-evolutionary Genetic Algorithm (CHCGA), which is inspired by the decomposition of large scale problems [2]. Indeed, it could be more efficient to decompose the initial problem into several less constrained problems. A similar strategy has been proposed by Schoenauer in [3] where constraints are added one by one, and a death penalty then rejects solutions which do not satisfy the previous constraints. The CHCGA is different since constraints are not added one by one but instead sets of satisfied constraints are merged. In addition, Schoenauer's algorithm uses a single population while the CHCGA works with several sub-populations which brings intrinsic parallelism properties. The CHCGA aims at generating a population filled with a certain ratio of feasible solutions. This population is then used as the initial population for a GA. As a result we can classify it as an initialisation procedure based on a co-evolutionary paradigm. To generate the initial population, the proposed algorithm decomposes the original constraints' set into a number of subsets having less constrained domains, such that it is trivial to obtain feasible solutions. A sub-population is then assigned to each subset and initialised with solutions satisfying it. Groups of 2 sub-populations are then formed. One group may contain 3 sub-population if the number of sub-population is odd. These groups of sub-populations are chosen at random in the current version of the CHCGA. During the evolutionary process, sub-populations of the same group can merge if their solutions satisfy the union of their subsets of constraints, i.e. the constraint set of the group. The goal of this procedure is to evolve these sub-populations and lead them to merge until reaching a single (sub-)population with feasible solutions for the global constrained domain. The final and unique sub-population is further used to initialise another evolutionary algorithm.

3. CLOUD BROKERING PROBLEM

Cloud brokering has gained recently interest among researcher and professionals as a real-life, relevant problem for both consumers and providers of cloud computing services [1]. In this paper, the cloud brokering optimization problem is defined as a general covering problem where customers want to select a set of bundled services among all providers knowing that the total cost should be minimized. The optimisation problem is formally defined as follows:

$$\min \quad F = \sum_{i}^{|P|} \sum_{j}^{|B^i|} d_{ij} X_{ij} \tag{1}$$

$$\text{s.t.} \quad \sum_{i}^{|P|} \sum_{j}^{|B^i|} q_{ij}^k X_{ij} \geq b^k \quad \forall k \in \{1, ..., M\} \tag{2}$$

where P is the set of providers, B^i is the bundle set of provider i, d_{ij} is the price associated to bundle j sold by provider i, q_{ij}^k is the amount of service k included in bundle j sold by provider i, b^k is the required amount of service k and X_{ij} is the number of bundle j to buy from provider i. Eq.(1) defines the objective, that is the minimal cost function for the whole set of purchased bundles. Constraints are expressed by Eq.(2), which ensures that the selected bundles provide at least the amount of service(s) requested by the user.

4. EXPERIMENTATIONS

As aforementioned, the CHCGA can be considered as an initialisation technique. For that reason, experiments were performed using the same constrained genetic algorithm. We have considered two cases: 1) the initial population is generated uniformly at random and 2) the initial population is the remaining population after applying the co-evolutionary approach (CHCGA). In both cases, cloud brokering instances were solved for a varying number of providers (20,50), services (50,100) and bundles per provider (25,50,75). Solutions were represented using an integer encoding in which each gene represents a variable X_{ij}. Figures 1 and 2 shows large differences between the average fitness results of the algorithms. One can also observe that the average fitness obtained with the co-evolutionary approach is better and the difference increases with the instances size. These results can be explained by the fact that knowledge about the decision set has been obtained during the co-evolutionary phase. In our case, the decision variables for the Cloud Brokering Problem are not upper-bounded which means that a random initialisation can lead to feasible but very low-quality solutions. An advantage brought by the CHCGA is that solutions close to the decision set frontier are generated. Indeed, each population tackles a unique constraint at the beginning of the algorithm. Finding feasible solutions for each sub-population is trivial in this case, since it suffices to take points belonging to the corresponding constraint. Then, the role of the co-evolutionary method is to evolve all of these sub-populations in order to generate solutions which satisfy more and more constraints until a single (sub-)population satisfies the full constraints set with an acceptable rate. We notice another advantage of this new approach concerning the co-evolutionary phase. The latter could be used to determine if the constraint set is hard to satisfy or not. Figures 1 and 2 also show the convergence for both algorithms as well as the co-evolutionary phase. It can be seen that more iterations are required by the co-evolutionary phase when the number of constraints increases (Figure 2).

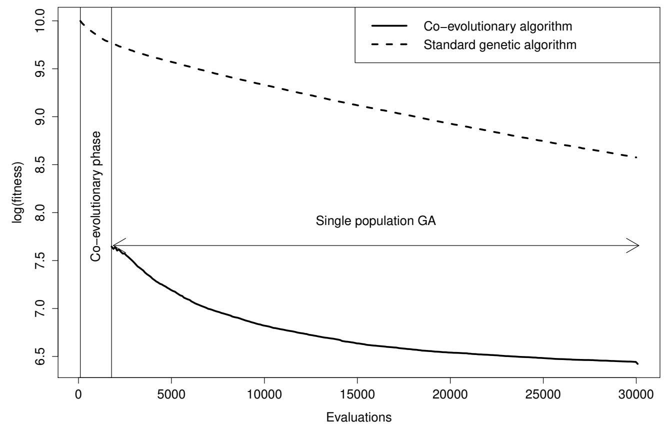

Figure 1: Convergence for CBO20_50_25 instance

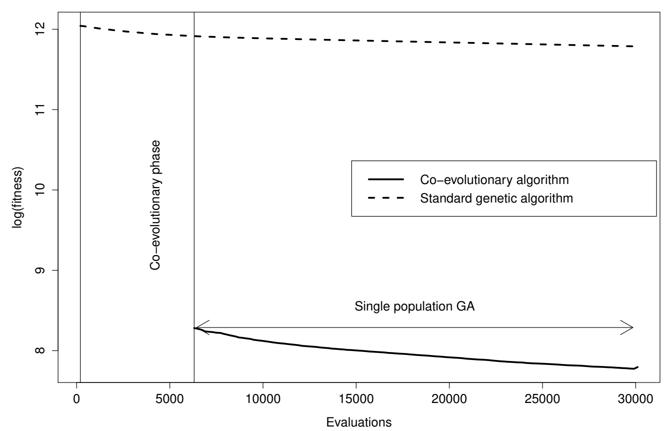

Figure 2: Convergence for CBO50_100_75 instance

5. PERSPECTIVES

Future work will investigate a new approach consisting in splitting the population when feasibility rate drops during the execution of genetic algorithms. In this way, merging and splitting procedures could improve the ability of genetic algorithms to keep a requested number of feasible solutions during all generations and not only at the initialization step. Group generation could be also enhanced by replacing the random procedure with an interaction-based rule.

6. REFERENCES

[1] M. Guzek, A. Gniewek, P. Bouvry, J. Musial, and J. Blazewicz. Cloud brokering: Current practices and upcoming challenges. *IEEE Cloud Computing*, 2(2):40–47, Mar 2015.

[2] L. Lasdon. Optimization theory for large systems, 1970.

[3] M. Schoenauer and S. Xanthakis. Constrained ga optimization. In *Proc. of the 5th Int'l Conf. on Genetic Algorithms*, pages 573–580. Morgan Kaufmann, 1993.