Computers & Operations Research 40 (2013) 1038–1046

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

Enhancing the performance of hybrid genetic algorithms by differential improvement

Zvi Drezner [a,*], Alfonsas Misevičius [b]

[a] College of Business and Economics, California State University-Fullerton, Fullerton, CA 92834, United States
[b] Department of Multimedia Engineering, Kaunas University of Technology, Lithuania

ARTICLE INFO

Available online 1 November 2012

Keywords:
Genetic Algorithms
Hybrid Genetic Algorithms
Memetic Algorithms
Tabu Search
Modified Robust Tabu
Quadratic assignment problem

ABSTRACT

A differential improvement modification to Hybrid Genetic Algorithms is proposed. The general idea is to perform more extensive improvement algorithms on higher quality solutions. Our proposed Differential Improvement (DI) approach is of rather general character. It can be implemented in many different ways. The paradigm remains invariant and can be easily applied to a wider class of optimization problems. Moreover, the DI framework can also be used within other Hybrid metaheuristics like Hybrid Scatter Search algorithms, Particle Swarm Optimization, or Bee Colony Optimization techniques.

Extensive experiments show that the new approach enables to improve significantly the performance of Hybrid Genetic Algorithms without adding extra computer time. Additional experiments investigated the trade-off between the number of generations and the number of iterations of the improvement algorithm. These experiments yielded six new best known solutions to benchmark quadratic assignment problems. Many other variants of the proposed algorithm are suggested for future research.

1. Introduction

1.1. Review of Genetic Algorithms

Genetic Algorithms (GAs) are a class of the artificial intelligence methods which are based on a Darwinian notion of natural selection, the principle of survival of the fittest, and Mendel's laws of transfer of traits [25]. From a broader point of view, they can be seen as a successful example of the virtualization of reality and transplanting ideas from the natural to the artificial. Such virtualization is also called "Biomimetics". See for example, Drezner and Drezner [8].

In the context of optimization, the basic attributes of GAs are the inclusion of a "population" of solutions and the use of special operators called "selection", "recombination" (crossover) and/or "mutation", and "replacement". The solutions of an optimization problem are metaphorically treated as individuals of a biological system and the cost of a solution (the value of the objective function) is associated with the fitness of an individual. The purpose is to successively produce better and better solutions by iteratively applying the above-mentioned operators. Although the optimality of the best obtained solution is not assured, sufficiently high quality solutions can usually be found within reasonable computation time. For a more thorough description of the principles of GAs, the interested reader is referred to Sivanandam and Deepa [47].

The most important advantages of GAs are versatility, parallelism, integration of rationality and randomness, ability to explore wider search space, ability to handle complex fitness landscapes and multiple local optima. However, the GAs also face severe barriers, like the possible loss of genetic variance (genetic drift), premature convergence, slow convergence, stalled evolution. This is even more pronounced in the crude canonical GA schemes.

To overcome these drawbacks, researchers have considered the fine-tuning of the parameter settings of GAs [46], the amendment of the genetic operators [15,19,57], and introduction of new features [9,36]. Also, more general methodological modifications have been proposed, for example, Compounded GAs [11], Parallel (Island-based) GAs [2], Messy GAs [23].

1.2. Review of Hybrid Genetic Algorithms

Another common idea is to incorporate additional local optimization techniques (improving algorithms) to refine the individuals, i.e. enhance their fitness. Note that due to some similarities of the role of local optimization within GAs and the role of the use of knowledge within the evolutionary process, the local

* Corresponding author. Tel.: +1 714 278 2712.
E-mail addresses: zdrezner@fullerton.edu (Z. Drezner), alfonsas.misevicius@ktu.lt (A. Misevičius).

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2012.10.014

optimization can also be thought of as a learning process. The main goal is to increase the performance of the traditional GAs by combining the explorative and exploitative capabilities of both the Genetic Algorithm and embedded algorithms, and balancing the global search (i.e. discovering new, more promising regions of the solution space) and local search (i.e. concentrating the search around good solutions in promising localized regions). It is also important in this case that the problem-specific information can be employed and integrated with the exploring abilities of the problem independent operators, which can accelerate convergence to solutions of high quality. This category of improved GAs is commonly termed Hybrid Genetic Algorithms (HGAs) [17], also named as Memetic Algorithms [40] or Genetic-Local Searches [20].

In the last decade, HGAs have attracted considerable attention of many algorithmists and the power of HGAs has been demonstrated in many domains of computer science, industrial engineering and operations research, including various types of hard optimization problems among them: continuous optimization [3], graph partitioning [49], location problems [7], quadratic assignment problem [10,33,34], scheduling problems [53,55], set covering problems [54], vehicle routing [24] and others [41]. However, despite the remarkable progress in this field, the design and investigation of advanced HGAs is still a very active area of research.

Hybrid GAs are a general scheme for cooperative optimization, where the incorporated algorithms work together with the genetic operators. There exists a great variety in the choice of how to design particular components of the Hybrid algorithm and many new opportunities for innovation arise [17,28,29,37,52]. It should, however, be stressed that implementing the HGA in a straightforward naive manner does not necessarily yield good solutions in reasonable computation time. This is especially evident in the cases of time-expensive heuristic optimizers (like Simulated Annealing, Tabu Search, Iterated Local Search). A number of important issues must be carefully addressed when an effective HGA is constructed [30].

Two popular ways of Hybridization rely on the concepts "Baldwin effect" [31] and "Lamarckism" [39].

In the first case, the local optimization can interact, allowing the local search to change the fitness of an individual. However, the genetic code itself remains unchanged. One of the disadvantages of the Baldwinian search strategies is that they are relatively slow [17].

The other way of combining GA and local optimization is known as Lamarckian evolution (or Lamarckian learning). This term is named after J.B. de Lamarck, who argued that individual's characteristics obtained during lifetime may become heritable traits. The central philosophy of this approach is that both the fitness and the genetic information of an individual are changed during the local optimization (learning) phase. The lifetime transformations and adaptations (the interactions between individuals and the environment) are likely of greater importance than the direct transmission of the parents' genetic code.

1.3. The contribution of the present paper

In this paper, we continue our endeavor to further increase the effectiveness of the Lamarckian-type Hybrid Genetic Algorithms. A novel conceptual modification referred to as a "Differential Improvement" is introduced. The intent is to reveal new potential positive synergetic effects of combining the global explorative and local exploitative processes, with strong focus on the enhanced, intensified improvement of particular population individuals.

This is a simple and effective concept that could easily be replicated and applied to other population based metaheuristics. Computational experiments for solving quadratic assignment problems demonstrate the effectiveness of the proposed Differential Improvement (DI) approach.

2. The Hybrid Genetic Algorithm framework

In our Hybrid Genetic Algorithm (HGA) framework, an improvement heuristic on every offspring is applied before considering its inclusion into the new population.

The following is a brief outline of our framework. The population size remains constant at P. The number of generations is prespecified at G. The improvement heuristic is a variant of Tabu Search [21,22] with pre-determined parameters. The crossover operator generates an offspring using the genes of two parents. The replacement strategy of our HGA is based on a steady state scheme [48]. The following is a general structure of the algorithm. It is also depicted in Fig. 1. A more detailed description is given in Section 5.2.

Algorithm 1 (*Hybrid Genetic Algorithm*).

1. A starting population of size P is randomly selected, and the improvement heuristic is applied on each starting population member. The current generation number is set to $g=1$.
2. In standard HGA this step is inactive. In the DI approach, four additional steps are inserted here as proposed below.
3. Two population members are randomly selected and merged by a crossover operator to produce an offspring.
4. The improvement heuristic is applied on the merged solution, possibly improving it.
5. If the value of the offspring's objective function is not better than the worst population member's objective function, the offspring is ignored. Go to Step 7.
6. Otherwise,
   (a) If the offspring is identical to an existing population member, it is ignored. Go to Step 7.
   (b) If the offspring is different from all population members, the offspring replaces the worst population member.
7. Set $g=g+1$. If $g \le G$ go to Step 2.
8. Otherwise ($g=G+1$), stop with the best population member as the final solution of the algorithm.

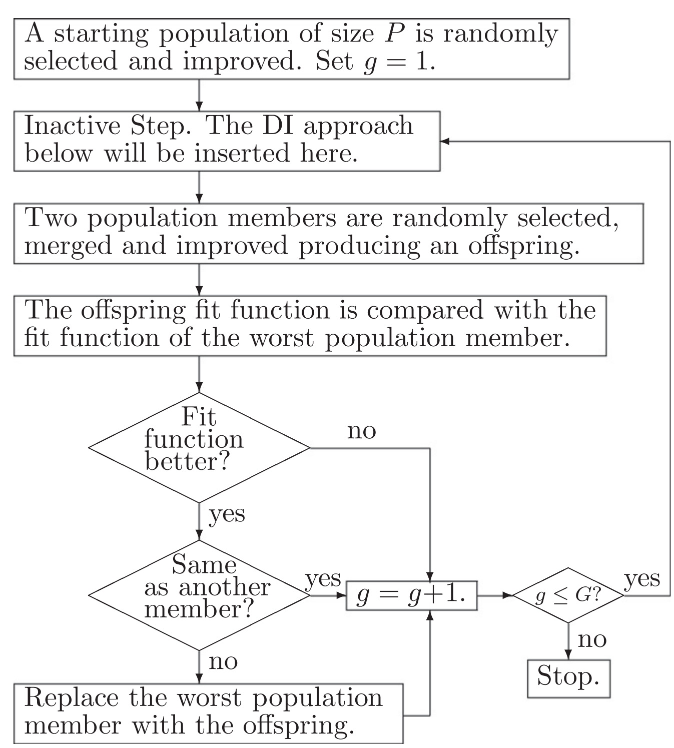

Fig. 1. A general flow chart of the Hybrid Genetic Algorithm.