Expert Systems with Applications 41 (2014) 4939–4949

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Review

Reactive Search strategies using Reinforcement Learning, local search algorithms and Variable Neighborhood Search

João Paulo Queiroz dos Santos [a,*], Jorge Dantas de Melo [b], Adrião Dória Duarte Neto [b], Daniel Aloise [b]

[a] *Federal Institute of Education, Science and Technology of Rio Grande do Norte, Campus Natal-Zona Norte, Natal, RN, Brazil*
[b] *Federal University of Rio Grande do Norte, Department of Computer Engineering and Automation, Natal, RN, Brazil*

ARTICLE INFO

Keywords:
Reactive Search
Reinforcement Learning
Local search
Variable Neighborhood Search
Combinatorial optimization

ABSTRACT

Optimization techniques known as metaheuristics have been applied successfully to solve different problems, in which their development is characterized by the appropriate selection of parameters (values) for its execution. Where the adjustment of a parameter is required, this parameter will be tested until viable results are obtained. Normally, such adjustments are made by the developer deploying the metaheuristic. The quality of the results of a test instance [The term instance is used to refer to the assignment of values to the input variables of a problem.] will not be transferred to the instances that were not tested yet and its feedback may require a slow process of "trial and error" where the algorithm has to be adjusted for a specific application. Within this context of metaheuristics the Reactive Search emerged defending the integration of machine learning within heuristic searches for solving complex optimization problems. Based in the integration that the Reactive Search proposes between machine learning and metaheuristics, emerged the idea of putting Reinforcement Learning, more specifically the Q-learning algorithm with a reactive behavior, to select which local search is the most appropriate in a given time of a search, to succeed another local search that can not improve the current solution in the VNS metaheuristic. In this work we propose a reactive implementation using Reinforcement Learning for the self-tuning of the implemented algorithm, applied to the Symmetric Travelling Salesman Problem.

1. Introduction

The diversity of complex problems for optimization encountered in the real world such as, telecommunications, logistics, transportation and financial planning are not a trivial task, as there are many situations where it is impossible to build a detailed model for the problem, due to its high complexity. On the other hand, a process for simplification of such a model can cause loss of relevant information that may compromise their quality. Besides the difficulties inherent in the construction of models for such problems, a feature that follows them during the resolution phase is the need for large computational processing, which in most cases leads such problems to be considered untreatable. In this context, numerous studies have been devoted to the development of techniques the facilitate modeling, and especially the resolution of these problems (Lima Junior, Melo, & Dória Neto, 2008).

One of these developed techniques was the local search by Croes (1958) and Bock (1958) for optimization problems where it generates a starting solution through a constructive heuristics, for each solution, and a neighborhood comprising of a set of solutions with very similar characteristics. Given a current solution, this neighborhood is traveled in search of another solution with a better value. If this solution is found it becomes the best solution found, and the algorithm continues the search. Otherwise the solution is an optimum site in relation to the neighborhood adopted. This type of search is determined by a decision based only on local knowledge of the search space. In addition, the local search methods can visit the same location within the search space more than once. In fact, many local search algorithms are prone to get stuck in some part of the search space that they can not escape without the use of special mechanisms, such as restart the process of finding the optimal or perform some kind of action towards diversification such as a perturbation of the solution. Consequently the local search does not produce an uniform result across the search space and has no uniform behavior for all classes and instances of optimization problems (Hoos & Stützle, 2005).

One approach that can be proposed to solve this problem of not having a good coverage in the search space, comes with the use of

* Corresponding author. Tel.: +55 84 40069500.
E-mail addresses: joao.queiroz@ifrn.edu.br (J.P. Queiroz dos Santos), jdmelo@dca.ufrn.br (J.D. de Melo), adriao@dca.ufrn.br (A.D. Duarte Neto), aloise@dca.ufrn.br (D. Aloise).

http://dx.doi.org/10.1016/j.eswa.2014.01.040
0957-4174/

various techniques of local searches for this search space in order to make it more robust regarding respect to the process of finding the optimal solution. This approach can be addressed by a metaheuristic, which is based on heuristic procedures, particularly applicable to combinatorial optimization problems which produces a stochastic search process in the search space (Talbi, 2009). The metaheuristic that uses multiple local searches is called VNS (Variable Neighborhood Search) whose main feature is exploring the search space through systematic exchange of neighborhood structures (local searches) randomly. VNS gradually explores neighborhoods more "distant" from the current solution rather than other local search strategies that follow a path.

One characteristic of VNS is that local searches are chosen randomly, thus not having a way to get a learning experience to select which local search can be the best alternative to get out of the stagnation of the previously selected local search, at any given time in which the search is located. Through this same learning after a certain period we can select another search because the current one does not provide the adequate return anymore and another local search may provide it.

In this context, the Reactive Search Optimization (RSO) or self-adaptive search appeared to support the integration of machine learning[1] within search techniques. The word "reactive" suggests a ready response to events during a search through an incremental internal feedback for self-tuning and dynamic adaptation. The Reactive Search differs from other techniques in literature for its context adaptive search, looking for new search areas for finding solutions that are better than the ones previously found, which is possibly a local optimum. This local-optima are responsible for the stagnation of the search process for the global optimum solution (Battiti & Tecchiolli, 1994).

Reinforcement Learning, more specifically the Q-learning algorithm, will lead this self-adaptive or Reactive Search, as a strategy of which local search should be chosen by the VNS metaheuristic applied to a combinatorial optimization problem. Adding also the dilemma of balancing the process intensification or diversification, another aspect to consider is the large number of possible solutions that exist in combinatorial optimization problems.

Based on the acknowledged techniques mentioned above and on the encountered difficulties on the real world, this work proposes the development of a reactive method using Reinforcement Learning and local searches with Variable Neighborhood Spaces, in order to self-adapt to the context of the search and this way to not getting stuck to a local minimum.

2. Theoretical foundation

2.1. Introduction

Every metaheuristic has as challenge during the search for the optimal solution, the balance between the processes of exploration and exploitation. Establishing this balance consists of deciding properly on the need or not to try new situations (explore new areas of the solution space) at the expense of those already experienced, i.e., the challenge is to solve the dilemma of intensifying the search in regions at the time considered promising, or explore in the expectation of finding the best regions in the future.

The development of metaheuristics is characterized by the appropriate selection of parameters (values) for its execution. Where the adjustment of a parameter is required, this parameter will be tested until viable results are obtained. Normally, such adjustments are made by the developer deploying the metaheuristic. The quality of the results of a test instance will not be transferred to the instances that were not tested yet and its feedback may require a slow process of "trial and error" where the algorithm has to be adjusted for a specific application.

Within this context of metaheuristics the Reactive Search emerged defending the integration of machine learning within heuristic searches for solving complex optimization problems. The word reactive suggests a rapid response to events during the search through an iterative feedback for an autotuning and a dynamic adaptation. In the Reactive Search the search history and the knowledge accumulated while there is movement in the search space configuration is used for a self-adaptation in an autonomous way: the algorithm keeps the internal flexibility to handle different situations during the search, and with its automated adaptation the algorithm uses past experience to reflect the current phase (Battiti, Brunato, & Mascia, 2008).

Based on the integration that the Reactive Search proposes between machine learning and metaheuristics, emerged the idea of putting Reinforcement Learning, more specifically the Q-learning algorithm with a reactive behavior, to select which local search is the most appropriate in a given time of a search, to succeed another local search that can not improve the current solution in the VNS metaheuristic. By understanding that such techniques can jointly cooperate with each other for a good computational performance it is proposed in this paper the implementation of the Reactive Search with the VNS metaheuristic to form reactive or self-adaptive in order to select the most suitable local search for a particular instant of a search. In the literature there are many interesting publications regarding Reactive Search, specifically regarding local searches that are the core of this work. Very pertinent publications are those of Hifi, Michrafy, and Sbihi (2006) proposing the Reactive Local Search to solve the Multiple-Choice Multi-Dimensional Knapsack Problem. The search begins with a greedy initial solution which is improved by a rapid and iterative method exchanging units of the solution, and after that methods to escape the local minimum and to introduce diversification in the search space are introduced respectively. To avoid repetition of the search results, a "memory" of the results already found is applied to the search. In the work of Bogl, Zapfel, and Affenzeller (2011) he analyzes the performance of the local search (first-improvement and best-improvement), the reactive tabu search and VNS in terms of solution quality on the Traveling Salesman Problem with Time Windows. The local searches used by the Reactive Tabu Search were the same used by the VNS. Then he makes a comparison of results between the impact of local search isolated with the Reactive Tabu Search and VNS, being able to accurately identify the impact on each one. For the Genomic Median Problem it was developed a Reactive Stochastic Local Search by Lenne, Solnon, Stützle, Tannier, and Birattari (2008, Chap. Reactive S) which is based on the iterated local search and tabu search. A reactive method that automatically adapts the length of the tabu list for the tabu search and also the intensity of the disturbance to the iterated local search was developed. Bräysy (2003) presents a new metaheuristic with deterministic procedure, but not in the broad sense (metaheuristics have stochastic character), but choosing a neighborhood of fixed size and in this neighborhood an exhaustive search is made, i.e., all possible solutions in this neighborhood are verified, based on changes in the VNS metaheuristic with a reactive behavior to solve the Vehicle-Routing Problem with Time-Windows. The proposed method is based on four phases:

- First phase: several initial solutions are created using construction heuristics with different combinations of values in the parameters.

[1] It is a sub-field of artificial intelligence dedicated to the development of algorithms and techniques that allow the computer to learn, that is to allow the computer to improve its performance at some task.