Neurocomputing 146 (2014) 17–29

Contents lists available at ScienceDirect

Neurocomputing

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

On the performance of linkage-tree genetic algorithms for the multidimensional knapsack problem

Jean P. Martins [a,*], Carlos M. Fonseca [b], Alexandre C.B. Delbem [a]

[a] University of São Paulo, Institute of Mathematical and Computer Sciences, São Carlos 13566-590, SP, Brazil
[b] CISUC, Department of Informatics Engineering, University of Coimbra, 3030-290, Portugal

ARTICLE INFO

Article history:
Received 30 October 2013
Received in revised form
7 March 2014
Accepted 30 April 2014
Available online 14 July 2014

Keywords:
Linkage-tree GA
Multidimensional knapsack problem
Linkage-learning usefulness

ABSTRACT

Model-based *Genetic Algorithms* (GAs), as the *Linkage Tree Genetic Algorithm* (LTGA) and most *Estimation of Distribution Algorithms* (EDAs), assume a reductionist perspective when solving optimization problems. They use machine-learning techniques to discover problem's substructures that might be useful to generate new solutions. This idea was grounded on Simon's near-decomposability principle and Holland's *Building Block* (BB)-hypothesis, and have enabled the development of effective algorithms in some contexts. Although near-decomposability is commonly seen in nature, we cannot assume the same occurs for optimization problems. Therefore, the existence of problems where these algorithms are not effective is also focus of research. Recent studies have argued that *Multidimensional Knapsack Problems* (MKPs) are examples of such cases. This paper extends these studies with an extensive comparison of various LTGA variants for the MKP. Using a well-known GA as reference, we analyzed the difficulties faced by the LTGA and explained why its linkage-tree model is not of much help to solve the problem. The results have shown that the LTGA was not able to outperform the GA and performed very similarly to a LTGA using random linkage-models. Further analysis of the linkage-trees, grounded on the knapsack-core concept, enabled interesting conclusions about the reason that linkage-learning did not provide useful information to solve MKPs in the settings used for the experiments.

1. Introduction

Divide and conquer is a well-known principle of algorithm design, which basically states that a large problem should be divided in smaller subproblems in order to solve the whole problem more efficiently. Although such ideas fit very well in computer science, decomposition is a much older subject, dating back to Descartes in 1637. In natural sciences, such reductionist view is not an alternative, but a characteristic presented by many physical, chemical and biological structures. In the 1960s, Simon [1] argued that the ubiquitousness of hierarchical structures found in complex systems (which he characterized as nearly decomposable systems) has important role in the feasibility of evolution, being an explanation to the effectiveness of evolution in natural and artificial systems [2].

Influenced by such context and based on the evolutionary theory, Holland [3] proposed *Genetic Algorithms* (GAs) in the 1970s and developed the schema theorem and *Building Block* (BB) hypothesis to explain the performance of GAs. In light of history, it is possible to establish a connection between the BB hypothesis and Simon's near-decomposability principle [1,3], and although the BB hypothesis has been strongly criticized these days, its general ideas have been successfully used as insights for more effective algorithms [4,5].

The most significant advances in this field were found under the umbrella of *linkage learning* [6,7]. In Simon's perspective, linkage-learning would be a procedure to identify the subsystems from a whole large system. In Holland and GA's perspective, linkage-learning would be a procedure to identify the BBs of a solution for a problem. This whole idea assumes that there are substructures to be discovered and that by knowing them, GAs could reach nearly optimal performance in the exploration of the search space. As we could expect, linkage-learning procedures are implemented using machine learning techniques, therefore, the limitations of these techniques also limit the applicability of linkage-learning.

The first multivariate linkage-learning implementations came from *Estimation of Distribution Algorithms* (EDAs) as the *Extended Compact Genetic Algorithm* (eCGA) [6] and *Bayesian Optimization Algorithms* (BOAs) [8]. These algorithms build multivariate probabilistic models to describe statistical dependences among variables and use those dependencies as linkage information to sample new solutions and guide the search [9]. Early developments of EDAs considered the representativity of a model crucial for an efficient EDA, therefore, algorithms as BOA (using Bayesian network models) have been

* Corresponding author.
E-mail addresses: jean@icmc.usp.br (J.P. Martins), cmfonseca@dei.uc.pt (C.M. Fonseca), acbd@icmc.usp.br (A.C.B. Delbem).

http://dx.doi.org/10.1016/j.neucom.2014.04.069
0925-2312/

extensively studied and, indeed, have shown interesting results in many hard optimization problems [8]. Recently, the *Linkage Tree Genetic Algorithm* (LTGA) introduced a new way to use linkage information not based in model sampling [10]. Differently from EDAs, the LTGA does not use a probabilistic model, but only a linkage model which guides an ingenious hierarchical crossover. This new approach has provided interesting results in complex nearly decomposable benchmark problems and required smaller population sizes, what enabled its model-building step to be considerably less expensive than its counterparts [10,11].

Model-based GAs have shown relevant results for nearly decomposable problems as NK-landscapes and spin-glasses [10,8,12]. Some methods have even enforced the building of modular models in order to exploit the near-decomposability assumption [13]. However, there are also few studies concerning the usefulness of such models to solve a broader class of optimization problems, where the underlying structure of the problems is not well-known or when such structures are not easy to learn [14–17]. Most multivariate model-building algorithms assume the existence of substructures, i.e. near-decomposability, but what if this characteristic is not present, how these algorithms would perform? Recently, Martins and Delbem [18] have shown that LTGA seems not to benefit from its linkage-learning when solving *Multidimensional Knapsack Problem* (MKP), in a sense that by using a learning procedure or using a random linkage model very similar results were obtained. The existence of problems not suitable for model-based GAs is a well-known fact (as justified, in general, by the no free-lunch theorem [19]), but how to identify those problems in a general black-box scenario is still an open question. Due to these aspects, it is of great importance for the field to better understand the limitations of linkage-learning in extracting useful information about optimization problems beyond the context of benchmark functions.

This paper extends the studies on the MKP with a performance analysis of different versions of the LTGA using the well-known *Chu and Beasley Genetic Algorithm* (CBGA) as reference. We modified the original LTGA in order to make it similar to the CBGA. By implementing mutation and a diversity preservation mechanism, we could evaluate the importance of each of these CBGA's operators. Concerning the LTGA's elements, we also evaluated a *complete-linkage* clustering in place of the *Unweighted Pair Grouping Method with Arithmetic-mean* (UPGMA) originally used by the LTGA. Results for the MKP enabled an analysis of the linkage-tree learning implemented by these algorithms and, using concepts from information-theory, an explanation of why the model built was not of much help to solve MKP instances. Model-based GAs bridge machine-learning and population-based metaheuristics in a very natural way, therefore, a systematic analysis of linkage-learning in the MKP context can increase knowledge about the methods implemented, stimulating progress in the field [20].

This paper is organized as follows: Section 2 reviews some details about the MKP and describe the CBGA and LTGA used in the experiments. Section 3 clarifies the methods used to prove our hypothesis and defines the settings and the different implementations used in the experiments. Section 4 describes the performance comparisons among many LTGA versions and the CBGA in many MKP instances, focusing the difficulties of using linkage-learning to solve them. Section 5 summarizes the results and discusses how linkage-learning limitations could be circumvented by the use of context-specific reproduction operators.

2. Background

2.1. The multidimensional knapsack problem

The *multidimensional knapsack problem* (MKP) is an well-known strongly NP-Hard combinatorial optimization problem [21,22]. The objective is, from a set of items, to choose a subset which gives maximum total profit, restricted by knapsack-capacity constraints. Differently from the unidimensional case, in the MKP the knapsack capacity is bounded by $m > 1$ constraints, formulated as follows:

$$\max f(\mathbf{x}) = \sum_{j=1}^{n} p_j x_j,$$

$$\text{subject to } \sum_{j=1}^{n} w_{ij} x_j \le c_i, \quad i = 1, \ldots, m,$$

$$x_j \in \{0, 1\}, \quad j = 1, \ldots, n.$$

where, there are n items available, with profits $p_j > 0$ $(j = 1, \ldots, n)$, and m resources available in amounts $c_i > 0$ $(i = 1, \ldots, m)$. Each chosen item contributes with p_j for the total profit and consumes w_{ij} from each resource i. The number of resources available defines the knapsack dimensionality, in *feasible* solutions the total resource consumption do not surpasses any capacity c_i, $\forall i \in \{1, \ldots, m\}$.

The MKP is considerably harder than its unidimensional counterpart, not admitting an efficient polynomial-time approximation scheme even for $m=2$ [23]. Furthermore, MKP hardness increases with m and larger instances still cannot be effectively solved to optimality. Due to these characteristics a large number of papers have been published regarding the MKP, concerning both exact and heuristic methods.

One of the earliest references concerning MKPs was a dynamic programming approach proposed by Gilmore and Gomory [24], which was further extended by Weingartner and Ness [25] by embedding heuristics to the algorithm. Branch-and-bound was also used to improve dynamic programming approaches, as proposed by Marsten [26,27]. In the same context, Shih [28] proposed the first branch-and-bound implementation based on linear programming. However, due to high space requirements and the limited computational capacity, such approaches were limited to very small problems. In order to circumvent such limitations, Gavish and Pirkul [29] developed a branch-and-bound using rules to reduce the problem size, obtaining better results than Shih's method. A more recent approach based on an approximated dynamic programming was proposed by Bertsimas [30], while a hybridization with a branch-and-cut-procedure was proposed by Boyer [31].

In the 1990s, many important results were obtained through a more extensive use of meta-heuristics. Tabu-search methods, as developed by Glover and Kochenberger [32] and further improved by Hanafi and Freville [33] in 1998, were able to solve to optimality all the public instances available so far [21]. At the same year, Chu and Beasley [34] proposed the first competitive GA for the MKP and they also made available a well-designed instance set, containing large correlated instances ($n = 100, 250, 500$ and $m = 5, 10, 30$). Several of their results for these instances were considerably improved by Vasquez [35,36], through a tabu-search algorithm. After that, many studies proposed new tabu-search methods for the MKP [37–39]. In GA's field, CBGA has influenced a series of studies concerning new GA's operators and fitness-landscapes analysis [40–44], and still is used as reference for more recent works with univariate EDAs [45,46,17,18], particle swarm optimization [47,48], differential evolution [49]. From now on, we use the CBGA as reference framework, for a more complete review of MKP's peculiarities see [50,21,22,51], for state-of-the-art results we suggest [52–54].

2.1.1. Efficiencies and the core concept

Among the most successful algorithms for the MKP, both exact and heuristics, the concept of *core* is usually applied. The main idea of *core* is very simple, some items are more valuable than others due to the relation between their profit and size (*pseudo-utility ratio* or *efficiency*). Items with high/low efficiencies are commonly set to 1/0 in high-quality solutions, therefore, using this idea a dimensionality reduction is possible, because the search