Computers & Industrial Engineering 112 (2017) 706–720

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Integrating estimation of distribution algorithms versus Q-learning into Meta-RaPS for solving the 0-1 multidimensional knapsack problem

Arif Arin Ph.D. [a], Ghaith Rabadi Ph.D. Professor [b,*]

[a] Middle East Technical University, Ankara, Turkey
[b] Engineering Management & Systems Engineering, Old Dominion University, Engineering Systems Building, Room 2102, Norfolk, VA 23529, United States

ARTICLE INFO

Article history:
Available online 27 October 2016

Keywords:
Machine learning
Estimation of distribution algorithms
Q-learning
Meta-RaPS
0-1 multidimensional knapsack problem

ABSTRACT

Finding near-optimal solutions in an acceptable amount of time is a challenge when developing sophisticated approximate approaches. A powerful answer to this challenge might be reached by incorporating intelligence into metaheuristics. We propose integrating two methods into Meta-RaPS (Metaheuristic for Randomized Priority Search), which is currently classified as a memoryless metaheuristic. The first method is the Estimation of Distribution Algorithms (EDA), and the second is utilizing a machine learning algorithm known as Q-Learning. To evaluate their performance, the proposed algorithms are tested on the 0-1 Multidimensional Knapsack Problem (MKP). Meta-RaPS EDA appears to perform better than Meta-RaPS Q-Learning. However, both showed promising results compared to other approaches presented in the literature for the 0-1 MKP.

1. Introduction

With the growing complexity of today's large scale problems, using exact optimization methods has become more difficult and more time consuming. Due to computational efficiency concerns, the need to find near-optimal solutions in an acceptable amount of time requires using heuristic approaches; however, heuristics are vulnerable for falling into local optima far from the optimum. Metaheuristics confront this challenge by adding strategies and mechanisms to the construction and local search algorithms in existing heuristics to escape local optima. A more effective performance may be obtained by incorporating computational intelligence (CI) into these heuristics.

In the problem solving arena, the definition of intelligence emerges in metaheuristics via memory and learning. Learning, according to Fogel (1995), is an intelligent process in which the basic unit of mutability is the idea. "Good" adaptive ideas are maintained, much as good genes increase in a population, while poor ideas are forgotten. Similarly, memory and learning mechanisms in metaheuristics can learn and remember "good" features related to the search process to make it possible to create high quality solutions for optimization problems. Glover and Laguna (1997) introduced a classification framework for metaheuristics based on three design choices: the use of adaptive memory, the type of neighborhood exploration used, and the number of current solutions carried from one iteration to the next. The metaheuristic classification notation can be illustrated in the form a|b|c. If the metaheuristic has adaptive memory, the first field (a) will be *A*, and *M* if the method is memoryless. Depending on the neighborhood mechanism, the second field (b) will be *N* for somehow systematic neighborhood search, and *S* for using random sampling. The third field (c) can be 1 for a single-solution approach or *P* for a population-based approach with population size of P. Tabu Search (TS) was among the first metaheuristics to explicitly use memory, and Glover and Laguna (1997) presented a more sophisticated version of TS to include longer term memory with associated intensification and diversification strategies. The authors defined this approach as Adaptive Memory Programming (AMP) because it is based on exploiting the strategic memory components. Taillard, Gambardella, Gendreau, and Potvin (2001) sketched the following algorithm of AMP based on the common features of the methods that use these strategic memory components:

(1) Initialize memory.
(2) Until stopping criteria are met, do:
   (a) generate a temporary solution s using data stored in the memory;
   (b) improve s by implementing local search, s'; and
   (c) update the memory using data brought by s'.

* Corresponding author.
E-mail addresses: arifarin@yahoo.com (A. Arin), grabadi@odu.edu (G. Rabadi).
URL: http://www.GhaithRabadi.com (G. Rabadi).

http://dx.doi.org/10.1016/j.cie.2016.10.022

In the literature there are various learning approaches in metaheuristics that show different performance levels (Arin & Rabadi, 2012a). There are also successful hybrid applications in which metaheuristics are empowered by intelligent approaches to improve their effectiveness, such as in TS with linear programming (Flisberga, Lidéna, & Rönnqvist, 2009), Genetic Algorithms (GA) (Thamilselvan & Balasubramanie, 2009), Simulated Annealing (SA) (Yeh, Chu, Chang, & Lin, 2011), and Evolutionary Algorithms (EA) (Wu, Wang, & Lü, 2015); GA with adaptive local search scheme (YoungSu, Chiung, & Daeho, 2009); Evolutionary Programming (EP) with fuzzy systems (Tan & Lim, 2011) and Reinforcement Learning (Huaxiang & Jing, 2008); Ant Colony Optimization (ACO) with fuzzy systems (Yeong-Hwa, Chia-Wen, Chin-Wang, Hung-Wei, & Jin-Shiuh, 2012); and Particle Swarm Optimization (PSO) with Memetic Algorithm (Hu, YukunBao, & Xiong, 2014), Artificial Bee Colony (Li, Wang, & Zheng, 2015), ACO and 3-Opt algorithms (Mahi, Baykan, & Kodaz, 2015). Such frameworks store and utilize various information related to search history to reach high quality solutions. Blum, Puchinger, Raidl, and Roli (2011) presented a survey on hybrid metaheuristics in combinatorial optimization.

In this paper, we will examine the performance of two well known learning approaches when integrated into Meta-RaPS (Meta-heuristic for Randomized Priority Search): Estimation of Distribution Algorithms (EDA) as a stochastic learning approach, and Q-learning as a machine learning approach. Meta-RaPS has been generating very promising solutions when applied to optimization problems and is currently classified as a memoryless metaheuristic with no incorporated memory nor learning mechanisms. To reveal the performances of these proposed algorithms, the 0-1 Multidimensional Knapsack Problem (MKP), which is a special case of the general linear 0-1 integer programming problem with nonnegative coefficients, will be used as test bed.

The 0-1 MKP is the generalized form of the classical knapsack problem (KP). In KP there is a knapsack with an upper weight limit b, and a set of n items with different profits c_j and weights a_j per item j. The problem is to select the items from the set such that the total profit of the selected items is maximized without exceeding the upper weight limit of the knapsack b. If m knapsacks exist, the problem becomes the MKP in which each knapsack has a different upper weight limit b_i, and an item j has a different weight a_{ij} for each knapsack i. The objective is to find a set of items with maximal profit such that the capacity of each knapsack is not exceeded (Gallardo, Cotta, & Fernández, 2009). The MKP can be formulated as in the Eqs. (1)–(3):

$$\text{Maximize} \sum_{j=1}^{n} c_j x_j. \tag{1}$$

$$\text{Subject to} \sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \quad i = 1, \ldots, m;\ j = 1, \ldots, n. \tag{2}$$

$$x_j \in \{0, 1\}, \quad j = 1, \ldots, n \tag{3}$$

where x is a vector of binary variables such that $x_j = 1$ if item j is selected, and $x_j = 0$ otherwise. In the literature it is assumed that profits, weights and capacities are positive integers. However, they can be easily extended to the case of real values (Martello & Toth, 1990).

The MKP is often used as a platform to evaluate new metaheuristics, and there are two main reasons for this academic interest. First, the 0-1 MKP is a special version of the well-known constrained integer programming problems, and also a subproblem of many general integer programs. According to Fréville (2004), the renewed interest in the research community in computational integer programming has intensified the use of MKP benchmarks. The theoretical interest arises mainly from their simple structure which allows exploitation of a number of combinatorial properties and, more complex optimization problems to be solved through a series of knapsack-type subproblems (Martello & Toth, 1990). From the practical point of view, the MKP can model many industrial situations, such as project selection, cargo loading and capital budgeting, which is the first context of MKP developed by Lorie and Savage (1955) and Manne and Markowitz (1957). Second, due to its well-known NP-Hardness, many researchers choose the 0-1 MKP as a test bed for their new heuristics. Although the MKP is a straightforward generalization of the single case, the new model is quite different when several constraints are taken into account.

Algorithms proposed in the literature to solve MKPs can be grouped into three classes: exact, heuristic and metaheuristic algorithms. Exact techniques include Lagrangian methods and surrogate relaxation techniques, special enumeration techniques and reduction schemes, and branch-and-bound. The exact approaches evolved around the same time of the development of the 0-1 MKP, and spanned the Lagrangian and surrogate relaxation techniques, special enumeration techniques, reduction schemes, and the branch-and-bound method (Varnamkhasti, 2012). Glover (1965) replaced the original constraints by one surrogate constraint. Greenberg and Pierskalla (1970) applied the first principal handling of surrogate constraints in general mathematical programming. Fréville (2004) claimed that, since the Lagrangean relaxation method is not suitable for handling the homogeneous MKP structure, the surrogate relaxation method is more beneficial than the Lagrangean relaxation. Balas (1965) applied implicit enumeration techniques to resolve the 0-1 linear programs, and Shih (1979) proposed the first linear programming-based branch-and-bound technique taking advantage of the special structure of the MKP. Marsten and Morin (1977) combined dynamic programming and branch-and-bound approaches for solving the MKP. Gavish and Pirkul (1985) demonstrated a faster branch-and-bound approach examining problems with seven constraints and 80 variables. Constraint programming techniques integrated with integer programming is another developing research area for solving mixed-integer programming problems in the context of MKP (Oliva, Michelon, & Artigues, 2001). Wilbaut and Hanafi (2009) proposed several convergent algorithms to solve a series of small sub-problems of 0-1 MKP generated by relaxations.

Although these exact approaches were proposed to present good upper and lower bounds, these methods can only solve small and medium size instances optimally. Therefore, for solving MKPs instances of large size, several heuristics and metaheuristic techniques have been developed. Hillier (1969) presented the first multi-stage algorithms for the MKP; the first stage identifies a route leading from the LP solution to other adjacent solutions pertaining to the integer viable region and the second stage traverses this route to find a better possible integer solution. The last stage employs local search to enhance the current solution by modifying one variable or more at the same time. This first step of this search strategy was improved later by Glover (1989) as a concept called Path Relinking. Loulou and Michaelides (1979) developed a method for the MKP starting from the origin and assigning ones to the values of the parameters in accordance with declining ratios until additional variables violate feasibility. Balas and Martin (1980) proposed the so-called "Pivot and Complement" method, a LP-based procedure for the MKP to solve a small sub-set of existing items to reach a high possibility of finding a global optimum in the core, and next to enhance the 0–1 solution gained in pivoting. Pirkul (1987) showed a faster and comparable technique for solving the MKP containing a descent method for determining the surrogate constraints. Lee and Guignard (1988) developed a multi-stage technique for solving the MKP tuned with user-defined variables that manage the trade-off between time of computation and