Expert Systems with Applications 41 (2014) 3712–3725

Contents lists available at ScienceDirect

Expert Systems with Applications

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

An improved firefly algorithm for solving dynamic multidimensional knapsack problems

Adil Baykasoğlu *, Fehmi Burcin Ozsoydan

Dokuz Eylül University, Faculty of Engineering, Department of Industrial Engineering, İzmir, Turkey

ARTICLE INFO

Keywords:
Firefly algorithm
Genetic algorithm
Differential evolution
Dynamic optimization
Multidimensional knapsack problem

ABSTRACT

There is a wide range of publications reported in the literature, considering optimization problems where the entire problem related data remains stationary throughout optimization. However, most of the real-life problems have indeed a dynamic nature arising from the uncertainty of future events. Optimization in dynamic environments is a relatively new and hot research area and has attracted notable attention of the researchers in the past decade. Firefly Algorithm (FA), Genetic Algorithm (GA) and Differential Evolution (DE) have been widely used for static optimization problems, but the applications of those algorithms in dynamic environments are relatively lacking. In the present study, an effective FA introducing diversity with partial random restarts and with an adaptive move procedure is developed and proposed for solving dynamic multidimensional knapsack problems. To the best of our knowledge this paper constitutes the first study on the performance of FA on a dynamic combinatorial problem. In order to evaluate the performance of the proposed algorithm the same problem is also modeled and solved by GA, DE and original FA. Based on the computational results and convergence capabilities we concluded that improved FA is a very powerful algorithm for solving the multidimensional knapsack problems for both static and dynamic environments.

1. Introduction

The area of evolutionary computation widely focuses on static optimization where the entire problem related data or variables remain stationary throughout optimization procedure. However, most of the real-life problems are dynamic in nature which means that a solution approach is expected to be adaptive to environmental changes. Therefore, tracking moving optima or high-quality solutions becomes the main purpose rather than optimizing a precisely predefined system. Evolutionary algorithms which have been widely studied for static optimization problems attracted an increasing attention over the past years on dynamic optimization problems (DOPs) (Branke, 2002; Jin & Branke, 2005; Morrison, 2004; Weicker, 2003). There are reported research theses (Liekens, 2005; Wilke, 1999) and edited volumes (Yang, Ong, & Jin, 2007) devoted to this research area in its infancy.

DOPs can be divided into two major fields as combinatorial and continuous. Because this study focuses on the combinatorial part, continuous DOPs are outside of the scope of this paper. However, a recently published study (Brest et al., 2013) provides a comprehensive research for continuous part. On combinatorial DOP part Rohlfshagen and Yao (2011) presented an extended work of Rohlfshagen and Yao (2008) on the dynamic subset sum problem. The authors used a binary encoding technique and a penalizing procedure to avoid infeasibility. Various stationary extensions of knapsack problems (Kellerer, Pferschy, & Pisinger, 2004) have been commonly studied by the researches. Because the dynamic version of multidimensional knapsack problem (dynMKP) is discussed here, related work is provided in the following.

Branke, Orbayı, and Uyar (2006) studied the effects of solution representation techniques for dynMKP. They used three types of representations which are, *binary encoding*, *permutation encoding*, and *real valued encoding with weight-coding*. The authors generated dynamic environments using a stationary environment as a base, and updated the changing parameters using normal distribution. A penalizing procedure is also presented in their studies in order to handle with infeasibilities.

Karaman and Uyar (2004) introduced a novel method which uses the *environment-quality measuring* technique which was applied to 0/1 knapsack problem for detection of the changes in the environment. Afterwards, Karaman, Uyar, and Eryiğit (2005) classified the dynamic evolutionary algorithms into four categories and they proposed an evolutionary algorithm considering the previous related work for 0/1 knapsack problem.

Kleywegt & Papastavrou, 1998, 2001 reported a stochastic/dynamic knapsack problem where items dynamically arrive with respect to a Poisson distribution. A distinctive feature was that the profit and unit resource consumption values of the items become visible as they arrive. In their problem the aim was to

* Corresponding author. Tel.: +90 232 3017600.
E-mail address: adil.baykasoglu@deu.edu.tr (A. Baykasoğlu).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.11.040

maximize the profit accepting the most appropriate items, rejecting rest of the items, where rejection causes a penalty. The same problem was reported by Hartman and Perry (2006). The authors utilized linear programming and duality for a quick approximation particularly on large scaled problems.

Karve et al. (2006) introduced and evaluated a middleware technology, capable of allocating resources to web applications through dynamic application instance placement which is a related problem of dynMKP. Kimbrel, Steinder, Sviridenko, and Tantawi (2005) reported a similar study.

Farina, Deb, and Amato (2004) proposed several test cases for dynamic multi-objective problems and the authors addressed the dynMKP as an appropriate problem for applications and implementations. For a similar purpose, Li and Yang (2008) proposed a generalized dynamic environment generator which can be instantiated into binary, real-valued and combinatorial problem domains. As reported in their studies, the proposed dynamic environment generator could be implemented to knapsack problems as well.

Adaptation of parameters was widely studied in the field of dynamic evolutionary algorithms. Thierens (2002) proposed adaptive mutation rate schemes for GAs, in order to prevent non-trivial decisions beforehand. Dean, Goemans, and Vondrák (2008) presented a stochastic version of 0/1 knapsack problem where the value of the items are deterministic but the unit resource consumptions are assumed as randomly distributed.

As it can be seen from the previous studies related to the dynamic version of knapsack problems, compared to well known static extensions, there are relatively far less reported publications. This is one of the main motivations behind the present study as dynamic knapsack problems have many practical implications. The second motivation is to investigate the performance of a relatively new and promising optimizer FA and its improved version (as proposed in this paper) on this dynamic optimization problem in comparison to very well known and widely used evolutionary algorithms GA and DE.

The rest of the paper is organized as follows: static and dynamic versions of MKP is presented in Section 2, solution approaches DE, GA and FA are discussed in detail in Section 3. Sections 4–5 represent experimental results and conclusion, respectively.

2. Dynamic multidimensional knapsack problem

As reported in the studies of Branke et al. (2006), MKP is in the NP-complete class and according to Kellerer et al. (2004) knapsack problems have been widely used as combinatorial benchmark problems of evolutionary algorithms. In addition, because MKP has numerous dynamic real-life applications such as cargo loading, selecting project funds, budget management, cutting stock, etc., dynamic version of MKP was used as benchmark environment in this study. Well known stationary version of MKP was formalized as follows in the scientific literature:

$$\text{maximize} \quad \sum_{j=1}^{n} p_j x_j \tag{1}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} w_{ij} x_j \leqslant c_i \quad \forall i \in I \tag{2}$$

$$x_j \in \{0,1\} \forall j \in J \tag{3}$$

where $J = \{1, 2, 3, \ldots, n\}$ set of items, $I = \{1, 2, 3, \ldots, m\}$ set of dimensions of knapsack, n is the number of items, m is the number of dimensions of the knapsack, x_j is the binary variable denoting whether the jth item is selected or not, p_j is profit of the jth item, c_j is the capacity of the ith dimension of the knapsack and w_{ij} is the unit resource consumption of the jth item for the ith dimension of the knapsack and all parameters are assumed to be positive.

Literature includes well known benchmark generators for DOPs (Morrison, 2004; Rohlfshagen, Lehre, & Yao, 2009; Ursem, Krink, Jensen, & Michalewicz, 2002; Yang, 2003). Those generators briefly translate a well-known static problem into a dynamic version using specialized procedures. Dynamism is adopted here as the changes of the parameters after a predefined simulation time units similar to Branke et al. (2006). Simulation time unit represents number of iterations allocated for each environment. In other words it represents a frequency of dynamic change. The less number of simulation time units yields to more frequent changes and vice versa. In this study, a series of 1000 iterations is adopted as the frequency of changes.

An initial environment is required as a basis in order to dynamically generate other environments. Therefore, as Branke et al. (2006), the first instance of *mknapcb4.txt*[1] was used as the basis environment. This instance includes 100 items and 10 dimensions with a tightness ratio of 0.25. After a change occurs, the parameters are updated as stated below.

$$p_j = p_j * (1 + N(0, \sigma_p)) \tag{4}$$

$$w_{ij} = w_{ij} * (1 + N(0, \sigma_w)) \tag{5}$$

$$c_i = c_i * (1 + N(0, \sigma_c)) \tag{6}$$

The standard deviation of normal distributions of each parameter are assumed to be equal and $\sigma_p = \sigma_w = \sigma_c = 0.05$ which yields to an average 11 out of the 100 items to be assigned or removed (Branke et al., 2006).

Distinctly from the other authors, neither lower nor upper bounds for those dynamic parameters are employed here. It's thought that if a change (whether with respect to a probability distribution or other sources of dynamism) occurs in dynamic environment, bounding it might not exactly reflect real-life applications. Solution approaches for DE, GA and FA are discussed in the following section.

3. Solution approaches

As stated in the aforementioned sections DOPs considerably differ from the traditional stationary problems (Baykasoğlu & Durmuşoğlu, 2011, 2013). In the typical black box optimizations, a population based algorithm starts from randomly chosen points in the search space or mapped representations because of the lack of information. The algorithm is expected to converge through generations. Like many other factors, complexity of problem has an apparent impact on the convergence capability of the algorithm. However, assuming that the changing optimum is around the previous one, intuition suggests that using the information of the previously visited points might make a complex problem to be solved. In other words, once a high-quality solution is reached for a time period of a DOP then the tracking good solutions might be easier unless a severe change occurs. This can also reduce the computational complexity of the problem (Rohlfshagen & Yao, 2011). A basic approach to solve a DOP might be restarting the algorithm with its initially set parameters once a change occurs. In other words DOP can be handled as a sequence of stationary problems. Conversely, the algorithm can be allowed to run in a continuous manner and not take any action to changes. Considering those opposite approaches, it can be seen that, convergence which is a desirable feature for stationary environments might not be enough

[1] http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/mknapcb4.txt.