Applied Soft Computing 29 (2015) 288–297

Contents lists available at ScienceDirect

Applied Soft Computing

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

An effective hybrid harmony search-based algorithm for solving multidimensional knapsack problems

Biao Zhang [a], Quan-Ke Pan [a,b,*], Xin-Li Zhang [c], Pei-Yong Duan [a]

[a] College of Computer Science, Liaocheng University, Liaocheng 252059, PR China
[b] State Key Lab of Digital Manufacturing Equipment & Technology in Huazhong University of Science & Technology, Wuhan 430074, PR China
[c] College of Mathematic Science, Liaocheng University, Liaocheng 252059, PR China

ARTICLE INFO

Article history:
Received 25 August 2014
Received in revised form 11 January 2015
Accepted 11 January 2015
Available online 19 January 2015

Keywords:
Multidimensional knapsack problem
Harmony search
Fruit fly optimization
Evolutionary algorithms
Binary algorithm
Meta-heuristic

ABSTRACT

This study presents an effective hybrid algorithm based on harmony search (HHS) for solving multidimensional knapsack problems (MKPs). In the proposed HHS algorithm, a novel harmony improvisation mechanism is developed with the modified memory consideration rule and the global-best pitch adjustment scheme to enhance the global exploration. A parallel updating strategy is employed to enrich the harmony memory diversity. To well balance the exploration and the exploitation, the fruit fly optimization (FFO) scheme is integrated as a local search strategy. For solving MKPs, binary strings are used to represent solutions and two repair operators are applied to guarantee the feasibility of the solutions. The HHS is calibrated based on the Taguchi method of design-of-experiment. Extensive numerical investigations based on well-known benchmark instances are conducted. The comparative evaluations indicate the HHS is much more effective than the existing HS and FFO variants in solving MKPs.

1. Introduction

The multidimensional knapsack problem (MKP) is a well-known combinatorial optimization, which is also an NP-hard problem [1]. The objective is to maximize the total profit of the selected given items with all resource constraints satisfied. A variety of practical applications can be formulated as MKPs, for example, capital budgeting [1], cargo loading [2], resource allocating [3], cutting stock [4], etc. Thus, it is significant in the development of effective and efficient algorithms for solving MKPs.

Exact algorithms were always applied to deal with MKPs in early studies, such as the branch and bound algorithm [2] and dynamic programming (DP) [5]. Because of the NP-hardness property of MKPs, these exact methods perform poorly when the scales come to be large, although they can produce optimal solutions in solving small-scale problems. In recent decades, some meta-heuristic algorithms have been employed to solve MKPs, including tabu search (TS) [6–8], genetic algorithm (GA) [9,10], simulated annealing (SA) [11], ant colony optimization (ACO) [12–15], particle swarm optimization (PSO) [16–18], and estimation of distribution algorithm (EDA) [19], etc.

Harmony search (HS) firstly presented by Geem et al. [20] is one of new meta-heuristics. It imitates the musical improvisation process to search for a perfect state of harmony. Since firstly proposed, HS has drawn much attention and it has been successfully applied to various mathematical and engineering optimization problems [21–25]. For the 0–1 knapsack problems: Zou et al. [26] proposed a novel global harmony search algorithm (NGHS) in which position updating and genetic mutation operators are applied to enhance its ability. Wang et al. [27] considered various adaptive mechanisms in the basic HS algorithm and developed a scalable adaptive strategy to improve its robustness. Abdesslem Layeb presented another modified algorithm named QIHSA [28] where quantum representation was integrated to achieve a better balance between the exploration and exploitation.

Inspired by the intelligent behavior of fruit flies in finding food, a new evolutionary optimization approach, namely fruit fly optimization algorithm (FFO), has recently proposed by Pan [29]. The FFO algorithm, because of its easy implementation and quick convergence, has been applied to diverse fields such as financial distress [30], power load forecasting [31] web auction logistics service [32] and PID controller tuning [36,37]. As for MKPs, to the best of our knowledge, there is only one research by Wang et al. [33], who proposed a binary fruit fly optimization algorithm (bFOA) in which three main search processes (smell-based process, search-based process and vision-based process) are designed to perform evolutionary search.

* Corresponding author.
E-mail address: panquanke@hust.edu.cn (Q.-K. Pan).

http://dx.doi.org/10.1016/j.asoc.2015.01.022
1568-4946/

Overall, despite identifying the high-performance regions in the search space well, the HS algorithm has difficulty in performing local search [22] whereas the FFO algorithm can be expert in local search but easily become trapped in local optima [31]. In view of these, in this paper, the FFO scheme is integrated into the HS as a local search strategy. During every evolution, HS operators are performed on the population group for global search and then each harmony in HM moves independently based on the FFO scheme to find an improving neighbor nearby. In this way, the resulting algorithm, named a hybrid harmony search-based algorithm (HHS), achieves good balance between the exploration capability of the HS algorithm and the exploitation ability of the FFO scheme. Additionally, combined with the parallel updating strategy, a novel improvisation rule with the modified memory consideration rule and the global-best pitch adjustment rule is developed to enhance the exploration ability of the HS operator. Furthermore, for solving MKPs, binary strings are used to represent the solutions, and two repair operators are applied to guarantee their feasibility. Finally, extensive numerical testing results and comparisons are provided to demonstrate the effectiveness of the proposed HHS.

The rest of this paper is organized as follows: In Section 2, the basic HS and FFO algorithms are introduced, while the mathematical information of the MKP is also presented. Then, the details of the proposed HHS and two repair operators are presented in Section 3. In Section 4, the experimental design and comparisons are given. Finally, Section 5 presents the conclusion and identifies future work.

2. HS, FFO and MKP

2.1. The basic HS algorithm

In the basic HS algorithm, each solution, represented by a n-dimensional real-valued vector, is analogous to a harmony and stored in the harmony memory (HM). The basic HS algorithm is composed of five steps described in detail below:

Step 1: Initialize the problem and algorithm parameters. Generally, the bound-constrained optimization problems can be defined as: Minimize $f(X)$ subject to $x(i) \in [\mathrm{LB}_i, \mathrm{UB}_i]$, $i = 1, 2, \ldots, n$. Here, $f(X)$ is the objective function; X is the set of decision variables $x(i)$; LB_i and UB_i are the lower and upper bounds for $x(i)$, respectively, and n is the number of decision variables. The four critical parameters are initialized in this step, including harmony memory size (HMS), the harmony memory considering rate (HMCR), the pitch adjusting rate (PAR) and the bandwidth (bw).

Step 2: Initialize the harmony memory (HM). The HM consists of HMS harmony vectors, and each decision variable is randomly generated as: $hm_{i,j} = \mathrm{LB}_i + r \times (\mathrm{UB}_i - \mathrm{LB}_i)$, where $i = 1, 2, \ldots, \mathrm{HMS}$, $j = 1, 2, \ldots, n$, and r is a random real-valued number between 0 and 1. Thus, the HM can be formulated in a matrix:

$$\mathrm{HM} = \begin{bmatrix} hm_{1,1} & hm_{1,2} & \cdots & hm_{1,n} \\ hm_{2,1} & hm_{2,2} & \cdots & hm_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ hm_{\mathrm{HMS},1} & hm_{\mathrm{HMS},2} & \cdots & hm_{\mathrm{HMS},n} \end{bmatrix} \tag{1}$$

Step 3: Generate a new harmony. The new harmony vector $\mathrm{HM}_{\mathrm{new}} = (hm_{\mathrm{new},1}, hm_{\mathrm{new},2}, \ldots, hm_{\mathrm{new},n})$ is generated by applying three basic rules including memory consideration, pitch adjustment and random selection. Decision variables in the generating vector will be sequentially produced following these three rules.

Firstly, in the memory consideration rule, the value of each decision variable is randomly chosen from the existing values stored in the HM with a probability HMCR or generated randomly from the possible range with a probability 1 − HMCR. The memory consideration rule is given as follows:

$$hm_{\mathrm{new},j} = \begin{cases} hm_{a,j} & \text{with HMCR probability} \\ \mathrm{LB}_j + r \times (\mathrm{UB}_j - \mathrm{LB}_j) & \text{with } 1-\text{HMCR probability} \end{cases} \tag{2}$$

where a is a random integer uniformly generated between 1 and n.

Then each decision variable obtained from the HM should conduct a pitch adjustment rule with a probability PAR. The pitch adjustment rule is showed as follows:

$$hm_{\mathrm{new},j} = hm_{\mathrm{new},j} \pm r \times bw \tag{3}$$

where r is a random real-valued number between 0 and 1.

Step 4: Update the harmony memory. The newly generated harmony is compared with the existing harmonies in HM and then replace the worst harmony if it has a better fitness value in terms of the objective function.

Step 5: Check the termination criterion. The evolutionary search repeats from Step 3 to Step 4 until it meets the termination criterion and then output the best harmony vector in HM.

2.2. The basic FFO algorithm

The FFO mimics the food finding behavior of fruit fly swarms, and consists of two main foraging processes: smell-based foraging process and vision-based foraging process. In the smell-based foraging phase, a group of fruit flies smell the food sources and they are randomly located. Then, fruit flies use sensitive vision to find the best food source location and fly towards it in the vision foraging phase. The basic FFO was presented for financial distress [30]. It is adapted to solve high-dimensional optimization problems by Pan [34] as follows:

Step 1: Initialize the fruit fly group location and set the control parameters. The fruit fly group location, $gl = (l_1, l_2, \ldots, l_n)$, is the set of decision variables, while n is the number of decision variables. The location is randomly initialized in the search space with control parameters, including maximum iteration and population size.

Step 2: Apply the smell-based process. A population size (PS) of food sources, $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$, are found which are generated around the current group location based on a random food finding distance (rd). This foraging phase procedure can be summarized as follows:

```
for (i = 1 to PS)
  for (j = 1 to n)
    x_{i,j} = l_j ± rd
  endfor
endfor
```

Step 3: Apply the vision-based process. Find the best food source location with best fitness value and then all fruit flies fly towards it. Thus, the current fruit fly swarm location is updated.

Step 4: Check the stopping criterion. If the stopping criterion is reached, terminate the algorithm. Otherwise, repeat the implementation of Steps 2–4.

2.3. Multidimensional knapsack problem

Mathematically, the multidimensional knapsack problem can be formulated as follows [4]:

$$\text{Maximize} \sum_{j=1}^{n} c_j d_j \tag{4}$$