ARTICLE IN PRESS

Journal of Computational and Applied Mathematics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

A simplified binary artificial fish swarm algorithm for 0–1 quadratic knapsack problems

Md. Abul Kalam Azad [a,*], Ana Maria A.C. Rocha [a,b], Edite M.G.P. Fernandes [a]

[a] *Algoritmi R&D Centre, Portugal*

[b] *Department of Production and Systems, School of Engineering, University of Minho, 4710-057 Braga, Portugal*

ARTICLE INFO

Article history:
Received 14 February 2013
Received in revised form 7 August 2013

Keywords:
0–1 knapsack problem
Heuristic
Artificial fish swarm
Swap move

ABSTRACT

This paper proposes a simplified binary version of the artificial fish swarm algorithm (S-bAFSA) for solving 0–1 quadratic knapsack problems. This is a combinatorial optimization problem, which arises in many fields of optimization. In S-bAFSA, trial points are created by using crossover and mutation. In order to make the points feasible, a random heuristic drop_item procedure is used. The heuristic add_item is also implemented to improve the quality of the solutions, and a cyclic reinitialization of the population is carried out to avoid convergence to non-optimal solutions. To enhance the accuracy of the solution, a swap move heuristic search is applied on a predefined number of points. The method is tested on a set of benchmark 0–1 knapsack problems.

1. Introduction

In this paper, we are particularly interested in the 0–1 quadratic knapsack problem (QKP) consisting in maximizing a quadratic objective function subject to a linear capacity constraint. This problem was introduced in [1] and may be expressed as follows:

$$\begin{aligned} &\text{maximize } f(\mathbf{x}) \equiv \sum_{i=1}^{n} p_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} p_{ij} x_i x_j \\ &\text{subject to } \sum_{i=1}^{n} w_i x_i \le c \\ &x_i \in \{0,1\}, \quad i = 1,2,\ldots,n, \end{aligned} \tag{1}$$

where $\mathbf{x}$ is the n-dimensional vector of the $0/1$ decision variables (items), p_i is a profit achieved if item i is selected and p_{ij} $(i = 1,2,\ldots,n-1, j = i+1,\ldots,n)$ is a profit achieved if both items i and j $(j > i)$ are selected. w_i is the weight coefficient of item i and c is the capacity of the knapsack. p_i, p_{ij} and w_i are positive integers and c is an integer such that $\max\{w_i : i = 1,2,\ldots,n\} \le c < \sum_{i=1}^{n} w_i$. The goal is to find a subset of n items that yields maximum profit f without exceeding knapsack capacity c. We may observe that if $p_{ij} = 0$ then the problem becomes a 0–1 linear knapsack problem (LKP).

The 0–1 QKP arises in a variety of real world applications, including finance, VLSI design, compiler construction, telecommunication, flexible manufacturing systems, location of airports, railway stations, freight handling terminals, and

* Corresponding author. Tel.: +351 253604740; fax: +351 253604741.
E-mail addresses: akazad@outlook.com (Md.A.K. Azad), arocha@dps.uminho.pt (A.M.A.C. Rocha), emgpf@dps.uminho.pt (E.M.G.P. Fernandes).

0377-0427/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cam.2013.09.052

hydrological studies. Classical graph and hypergraph partitioning problems can also be formulated as the 0–1 QKP. Several deterministic solution methods [2–5,1,6–11] as well as stochastic solution methods [12–15] have been proposed to solve (1). Billionnet and Soutif [2] used a linear reformulation technique for the 0–1 QKP and solved them efficiently using a standard mixed integer programming tool. In [3], an exact method based on the computation of an upper bound by the Lagrangian decomposition is proposed. Caprara et al. [5] investigated an exact branch and bound algorithm for the 0–1 QKP, where upper bounds are computed by considering a Lagrangian relaxation which is solvable through a number of (continuous) knapsack problems. Létocart et al. [9] presented reoptimization techniques for improving the efficiency of the preprocessing phase of the 0–1 quadratic knapsack resolution. In [10], an exact algorithm which makes usage of aggressive reduction techniques to decrease the size of the instance to a manageable size is introduced. An exact solution method based on a new linearization scheme is proposed in Rodrigues et al. [11].

The deterministic and exact methods are suitable for small dimensional problems. However, when the dimension increases, they cannot solve the problems within a reasonable time period. This is the main motivation to develop stochastic methods and heuristics for solving QKP. In the context of constrained problems, the widely used approach is based on penalty functions. In this approach, a penalty term is added to the objective function aiming to penalize constraint violation. The penalty function method can be applied to any type of constraints, but the performance of penalty-type method is not always satisfactory due to the choice of appropriate penalty parameter values. Hence, other alternative constraint handling techniques have emerged in the last decades.

Examples of stochastic population-based methods to solve the 0–1 QKP are as follows: Glover and Kochenberger [12] reformulated the 0–1 QKP to unconstrained binary quadratic problem and solved using Tabu search. In [13], a hybridization of the genetic algorithm with greedy heuristic based on the *absolute-profit to weight* ratio is proposed. Here, the capacity constraint is handled by never generating chromosomes whose solutions violate it. Narayan and Patvardhan [14] introduced a novel quantum evolutionary algorithm for the 0–1 QKP and Xie and Liu [15] presented an agent-based mini-swarm algorithm using the *absolute-profit to weight* ratio to repair and improve the solutions.

Unlike the stochastic methods, the outcome of a deterministic method does not depend on pseudo-random variables. In general, its performance depends heavily on the structure of the problem since the design relies on the mathematical attributes of the optimization problem. In comparison with the deterministic methods, the implementation of stochastic algorithms is often easier. A survey of different methods for solving the 0–1 QKP is found in [16].

The artificial fish swarm algorithm (AFSA) is an example of a stochastic method that has recently appeared to solve continuous and engineering design optimization problems [17–20]. When applied to an optimization problem, a 'fish' represents an individual point in a population. The algorithm simulates the behaviors of a fish swarm inside water. At each iteration, trial points are generated from the current ones using either a chasing behavior, a swarming behavior, a searching behavior or a random behavior. Each trial point competes with the corresponding current and the one with best fitness is passed to the next iteration as a current point. There are in the scientific literature different versions and hybridizations of AFSA [21–23].

This paper presents a simplified binary version of AFSA for solving the 0–1 QKP. A previous binary version of AFSA, denoted by bAFSA, is presented in [24], where a set of small 0–1 multidimensional knapsack problems were successfully solved. Nevertheless, the computational effort required by bAFSA when solving large dimensional problems is not satisfactory. To create the trial points from the current ones in a population, bAFSA chooses each point/fish behavior according to the number of points inside its 'visual scope', i.e., inside a closed neighborhood centered at the point. To identify those points, the Hamming distance between pairs of points is used. When the chasing behavior is chosen, the trial point is created after performing a uniform crossover between the individual point and the best point inside the 'visual scope'. On the other hand, when the swarming behavior is chosen, a uniform crossover between the individual point and the central point of the 'visual scope' is performed to create the trial point. When the searching behavior is chosen, the trial point is created by performing a uniform crossover between the individual point and a randomly chosen point from the 'visual scope'. Finally, in the random behavior, the trial point is created by randomly setting a binary string of 0/1 bits of length n. Past experience has shown that the time related with the computation of the 'visual scope' of all points, at each iteration, is $\mathcal{O}(Nn^2)$, where N is the number of points in the population.

The purpose of the herein presented study is to simplify the procedures that are used to choose which behavior is to be performed to each current point in order to create the corresponding trial point. The main goal is to reduce the computational requirements, in terms of the number of iterations and execution time, to reach the optimal solution. This is a new simplified binary version of AFSA, henceforth denoted by S-bAFSA. Briefly, for all points of the population, except the best, random, searching and chasing behavior are randomly chosen using two target probability values $0 \leq \tau_1 \leq \tau_2 \leq 1$, and thereafter a uniform crossover is operated to create the trial points. A simple 4-flip mutation is performed in the best point of the population to generate the corresponding trial point. To make the points feasible, the new S-bAFSA uses a random heuristic drop_item procedure followed by an add_item operation aiming to increase the profit throughout the adding of more items in the knapsack. Furthermore, to improve the accuracy of the solutions obtained by the algorithm, a swap move heuristic search [25] and a cyclic reinitialization of the population are implemented. A benchmark set of 0–1 knapsack problems is used to test the performance of the S-bAFSA.

The paper is organized as follows. The proposed simplified binary version of the artificial fish swarm algorithm is described in Section 2. Section 3 describes the experimental results and finally we draw the conclusions of this study in Section 4.

Please cite this article in press as: Md.A.K. Azad, et al., A simplified binary artificial fish swarm algorithm for 0–1 quadratic knapsack problems, Journal of Computational and Applied Mathematics (2013), http://dx.doi.org/10.1016/j.cam.2013.09.052