ARTICLE IN PRESS

Knowledge-Based Systems xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Multi-Objective Artificial Bee Colony algorithm applied to the bi-objective orienteering problem

Rodrigo Martín-Moreno, Miguel A. Vega-Rodríguez*

Department of Computer and Communications Technologies, University of Extremadura, Escuela Politécnica, Campus Universitario s/n, Cáceres 10003, Spain

ARTICLE INFO

Keywords:
Artificial Bee Colony
Multi-objective optimization
Bi-objective orienteering problem
Swarm intelligence
Evolutionary computation

ABSTRACT

We propose a new evolutionary computation approach for solving the multi-objective orienteering problem. This problem has applications in different fields like routing problems or logistic problems. In our case, the final motivation is the design of individual tourist routes. The tourists have different priorities about points of interests grouped into categories (for example, cultural or leisure), so, a multi-objective solution system is needed. In order to obtain the best Pareto solutions, the Artificial Bee Colony algorithm (based on swarm intelligence) has been adapted to the multi-objective context. The performance of this approach has been compared with two previous algorithms from the literature for the bi-objective orienteering problem (P-ACO and P-VNS), in benchmark instances and real-world instances. The results indicate that this new approach is good for solving the multi-objective orienteering problem.

1. Introduction

There is a huge variety of decision support systems that are used for different purposes in the companies such as marketing, finances, manufacturing, logistics or human resources, for example. However, it is very infrequent that companies provide systems to people which help them designing a tourist route that fits in their preferences, rather they offer different preplanned routes with low possibility of customization. When visitors are planning a stay in a tourist destination, they commonly want to visit some locations or Points Of Interests (POIs). These locations are from cathedrals, historical places or museums to restaurants, pubs or theaters. In a destination, there are a lot of POIs, chiefly if it has an important cultural heritage or a large variety for leisure. As it is impossible to visit all the places that destinations offer, the visitor must prioritize which POIs are worth to be visited using budget, time and interest, and decide what is their order in the route. Hence a decision support system that helps visitors in that process might be very interesting. Nevertheless the final decision will be made by the visitor, because this system will offer only the best solutions found that satisfy the user requirements, but will never consider the emotional part that biases in the process of planning a visit.

The fact of creating routes that connect POIs can be defined as an Orienteering Problem (OP) [1]. In this case, our focus is on the Multi-Objective Orienteering Problem (MOOP), where there are several categories for each point of interest (for example, cultural or leisure) and each of them has distinct benefits for every category. We have developed a Multi-Objective Artificial Bee Colony (MOABC) algorithm, based on the single-objective ABC algorithm proposed by Karaboga and Basturk [2], in order to solve MOOP in a competitive way. This is the first time that MOABC algorithm is applied to solve multi-objective orienteering problems and the conclusions obtained are very interesting. As we will see, MOABC results are very competitive when they are compared with the results obtained by other multi-objective algorithms (P-ACO and P-VNS) from the state-of-the-art in this field. The experiments have been made in both benchmark instances and real-world instances (a total of 216 instances grouped in 10 sets), and we have used three state-of-art performance multi-objective indicators to report and compare the results. Moreover, the outperformance of MOABC has been confirmed after a statistical analysis.

This paper is structured as follows. Section 2 introduces the reader to the work made in this field. Section 3 presents the formal problem definition and its mathematical formulation. Section 4 describes our multi-objective approach to solve this problem. Section 5 discusses about the results obtained and compares the quality metrics of our algorithm with respect to other previously published algorithms, including a statistical analysis. Section 6 concludes this paper and summarizes possible future works.

2. Related work

Multi-objective optimization is an important field, with a lot of activity in the past two decades. Within multi-objective optimization,

* Corresponding author.
E-mail addresses: rmartinky@alumnos.unex.es (R. Martín-Moreno), mavega@unex.es (M.A. Vega-Rodríguez).

https://doi.org/10.1016/j.knosys.2018.05.005
Received 14 November 2017; Received in revised form 2 May 2018; Accepted 5 May 2018

Please cite this article as: Martín-Moreno, R., Knowledge-Based Systems (2018), https://doi.org/10.1016/j.knosys.2018.05.005

the solving of MOOPs is an area of high applicability. Regrettably, there is not much research work in this area.

Mainly, there are three different approaches with multiple objective functions. The first of them implies building a function that combines the objectives using some kind of weights or thresholds, but the problem is that these have to be previously defined. The result obtained with this approach is a solution that fits well with the weighted relation among objectives given by the fitness function. A second approach is based on switching among the objectives (criteria) during the selection phase. Every time a selection is made, potentially a different objective function (criterion) will be used. The main problem of the objective (criterion) switching is that the solution population tends to converge to solutions which are superior in one objective, but poor at others. Finally, the third alternative is based on including the objectives isolated from each other and calculating the best (that is, Pareto-optimal or non-dominated) candidate solutions in order to provide the decision-maker the possibility of exploring all the best trade-off solutions. Here, we use this third alternative.

In the literature we can find extensions of well-known metaheuristics to the multi-objective context: genetic algorithms [3], evolutionary algorithms (see [4]), and some swarm algorithms [5], for example. A complete survey on multi-objective evolutionary algorithms can be found in [6]. In our case, the ABC algorithm has been adapted to the multi-objective context. We have based our work on the ABC algorithm because it has been widely and successfully studied and applied to solve real-world problems in multiple fields [7], including single-objective optimization (e.g. [8–10]) and multi-objective optimization (e.g. [11–13]).

The orienteering problem was defined by Tsiligirides [1], and some authors have considered it as a kind of TSP (Traveling Salesman Problem) with profits (see [14]) or selective TSP. In OP, every vertex has associated some benefit, and the goal is to visit a group of vertices that maximize the sum of benefits, while fulfilling the corresponding tour length/cost constraint. Other related problems are the vehicle routing problem with profits [15] and the team orienteering problem (see [16,17]), where the problem is extended to multiple tours. Two complete surveys on orienteering problem can be found in [18,19].

Regarding the multi-objective orienteering problem, the problem addressed in this paper, very few proposals can be found in the literature. Within bi-objective orienteering problem we can highlight [20] that used a P-ACO (Pareto Ant Colony Optimization) algorithm and a P-VNS (Pareto Variable Neighborhood Search) algorithm combined with path relinking and [21] that used an evolutionary algorithm also combined with path relinking.Martí et al. [22] used GRASP combined with path relinking.

3. Problem definition

The multi-objective orienteering problem can be specified based on a directed graph $G = (V, A)$. This directed graph has a set of vertices, $V = \{v_0, v_1, v_2, ..., v_{n+1}\}$, and a set of arcs, $A = \{(v_i, v_j)\colon v_i, v_j \in V \wedge v_i \neq v_j \wedge v_i \neq v_{n+1} \wedge v_j \neq v_0\}$. Every vertex $v_i \in V \setminus \{v_0, v_{n+1}\}$ has associated K benefits b_{ik} $(k = 1,...,K)$. The starting and ending vertices, v_0 and v_{n+1}, do not have benefits associated. Furthermore, each arc $(v_i, v_j) \in A$ has a cost c_{ij} that can be interpreted as distance, money or time spent for going from v_i to v_j.

In all the instances used in this work, v_0 and v_{n+1} represent the same point. For this reason, we call a solution as "tour" instead of "path". The goal of the multi-objective orienteering problem is to find the best tours (which maximize the benefits in all the objectives) from v_0 to v_{n+1}, while fulfilling the tour length/cost constraint $C_{\max}$.

Therefore, we can mathematically define the problem as:

$$maximize \quad F(x) = (f_1(x), \; ...,f_K(x)), \tag{1}$$

$$f_k = \sum_{v_i \in V \setminus \{v_0, v_n+1\}} (b_{ik} \cdot y_i) \quad (k = 1,...,K), \tag{2}$$

where y_i, a binary variable, has a value equal to 1 when v_i is visited, and else a value of 0. Furthermore, we have to take into account that:

$$\sum_{v_i \in V \setminus \{v_j\}} a_{ij} = y_j \quad (v_j \in V \setminus \{v_0\}), \tag{3}$$

$$\sum_{v_j \in V \setminus \{v_i\}} a_{ij} = y_i \quad (v_i \in V \setminus \{v_n + 1\}), \tag{4}$$

$$\sum_{\{v_i, v_j\} \in S} a_{ij} \leq |S| - 1 \quad (S \subseteq V \wedge S \neq \varnothing), \tag{5}$$

$$y_0 = y_{n+1} = 1, \tag{6}$$

$$\sum_{(v_i, v_j) \in A} c_{ij} \cdot a_{ij} \leq C_{max}, \tag{7}$$

$$a_{ij} \in \{0, 1\} \quad \Big((v_i, v_j) \in A\Big), \tag{8}$$

$$y_i \in \{0, 1\} \quad (v_i \in V). \tag{9}$$

The binary variable a_{ij} has a value equal to 1 when $(v_i, v_j) \in A$ is used, and else a value of 0. Eq. (1) indicates that for solving MOOP we have to maximize the different objective functions. Eq. (2) defines each objective function as the addition of the corresponding benefits. Eq. (3),4 imply that for each vertex visited only one arc is ingoing and only one arc is outgoing. Eq. (5) avoids subtours. Eq. (6) implies that the starting and ending points are used in all the tours. Eq. (7) guarantees that the tour cost is not greater than the established limit $C_{\max}$. In our case, we will solve the bi-objective orienteering problem, that is, $K = 2$.

4. Solution procedure

Golden et al. [23] demonstrated that OP is NP-hard; no polynomial time algorithm has been designed, or is expected to be designed, to solve this problem to optimality, and especially when multiple objectives exist.

Therefore, we need to apply a metaheuristic solution technique to the multi-objective orienteering problem. In this section, we describe the MOABC algorithm that we have designed and developed. The single-objective Artificial Bee Colony (ABC) algorithm was originally proposed by Karaboga and Basturk [2], and we adapt it to the multi-objective context and the particular solving of MOOP.

We represent every solution to MOOP as a list of points (those included in the corresponding tour), which is the most natural way for representing a solution for this problem.

4.1. Multi-objective optimization

Due to the multi-objective nature of the problem to solve, it is very difficult to choose exactly an optimal solution where all the objectives are maximized. Nevertheless, if we restrict to non-dominated solutions the choice will be bounded to a reasonable amount of candidate solutions. The next definitions help to clear this aspect.

A solution x dominates a solution x' if x is not worse than x' in any of the objective functions, and is better at least in one of the objective functions. Formally: for $F(x) = (f_1(x), \; ...,f_K(x))$ to be maximized, x dominates x' if $f_k(x) \geq f_k(x')$ for all $k = 1, \; ...,K$, and $f_k(x) > f_k(x')$ for at least one k. If this happens, we write $x \succ x'$.

If no solution dominates the solution x^*, we say that x^* is non-dominated or Pareto-efficient. In this case, we say that $z^* = F(x^*) = (f_1(x^*),...,f_K(x^*))$ is a non-dominated vector. The set of all non-dominated vectors is called non-dominated frontier or Pareto front. The relation $\succ$ can be extended from the solution space to the objective space. In that case, given two vectors $z = (z_1, \; ...,z_K)$ and $z' = (z'_1,...,z'_K)$,