Applied Soft Computing 48 (2016) 254–264

Contents lists available at ScienceDirect

Applied Soft Computing

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

A genetic algorithm for the minimum generating set problem

Manuel Lozano[a], Manuel Laguna[b], Rafael Martí[c], Francisco J. Rodríguez[d,*], Carlos García-Martínez[e]

[a] Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain
[b] Leeds School of Business, University of Colorado, Boulder, USA
[c] Department of Statistics and Operations Research, University of Valencia, Valencia, Spain
[d] Department of Computer Science, University of Extremadura, Mérida, Spain
[e] Department of Computing and Numerical Analysis, University of Córdoba, Córdoba, Spain

ARTICLE INFO

Article history:
Received 28 September 2015
Received in revised form 6 July 2016
Accepted 11 July 2016
Available online 20 July 2016

Keywords:
Minimum generating set problem
Genetic algorithms
Multiple knapsack problem
Real-parameter crossover operator

ABSTRACT

Given a set of positive integers S, the minimum generating set problem consists in finding a set of positive integers T with a minimum cardinality such that every element of S can be expressed as the sum of a subset of elements in T. It constitutes a natural problem in combinatorial number theory and is related to some real-world problems, such as planning radiation therapies.

We present a new formulation to this problem (based on the terminology for the multiple knapsack problem) that is used to design an evolutionary approach whose performance is driven by three search strategies; a novel random greedy heuristic scheme that is employed to construct initial solutions, a specialized crossover operator inspired by real-parameter crossovers and a restart mechanism that is incorporated to avoid premature convergence. Computational results for problem instances involving up to 100,000 elements show that our innovative genetic algorithm is a very attractive alternative to the existing approaches.

1. Introduction

The minimum generating set (MGS) problem, a natural problem in combinatorial number theory [1], is defined as follows: given a set of positive integers, $S = \{s_1, \ldots, s_n\}$, the problem consists of finding a minimum cardinality set of distinct integers $T = \{t_1, \ldots, t_m\}$, called *generating set*, such that every element of S is equal to the sum of a subset of T. The MGS problem has been shown to be NP-hard [1], and is related, among other problems, to planning radiation therapies [2–4]: the elements of S represent radiation dosages required at various points, while an element of T represents a dose delivered simultaneously to multiple points. Then, the objective is to find the set of doses that properly combined (subsets of T), produces the initial requirements (S). Other variants, namely the cases in which the elements of T can be negative or fractional, were considered elsewhere [5,6].

The greedy algorithm presented by Collins et al. [1] is the unique proposed approach for the MGS problem so far. Its idea is to represent the largest set of integers s_i by means of the combination of other integers s_j, previously accepted solution components t_k, and a new candidate solution component d. The process is repeated until all the integers $s_i \in S$ have a representation based on solution components. Fagnot et al. [7] gave some elementary properties of the minimum 2-generating set, a natural restriction of the MGS problem where each element of S must be represented by the sum of at most two elements from T, and proved its hardness. However, surprisingly, not a single metaheuristic approach has been applied so far (to our knowledge) to tackle the problem from a practical point of view. This fact was our main motivation for the development of a genetic algorithm (GA) that aims at optimizing the MGS problem. GA is a well known metaheuristic that has proved to be very effective in solving hard optimization problems [8,9].

In GA, a population of candidate solutions, called chromosomes, evolves over successive generations using three genetic operators: selection, crossover, and mutation. First of all, based on some criteria, every chromosome is assigned a fitness value, and then the selection mechanism is invoked to choose relatively fit chromosomes to be part of the reproduction process. Then, new chromosomes are created through the crossover and mutation operators. The crossover generates new individuals by recombining the characteristics of existing ones, whereas the mutation operator is used to maintain population diversity with the goal of avoiding premature convergence.

* Corresponding author.

http://dx.doi.org/10.1016/j.asoc.2016.07.020
1568-4946/

The proposal presented in this paper to successfully address the MGS problem rests on four pillars:

- We redefine the MGS problem using the terminology employed in the well-known multiple knapsack (MK) problem, which has been extensively studied within this class of algorithms.
- We devise a randomized greedy procedure specifically designed for generating feasible solutions for the given MGS case in reasonable computing times. Highly constrained combinatorial optimization problems such as the MGS problem have proved to be a challenge for metaheuristic solvers [10]. This is a situation in which it is difficult to define an efficient neighborhood, thus no local search is available [11]. Therefore, the incorporation of specialized constructive greedy heuristics is often necessary in order to produce practical implementations [10].
- On the basis of the new formulation, we propose a GA approach to deal with the MGS problem that comprises an initial population generation method (based on the proposed randomized greedy algorithm) with the goal of acquiring a population of diversified, yet adequate quality solutions, and a restart mechanism, substituting the usual GA mutation, to regenerate population diversity when chromosomes become very similar.
- In addition, the proposed GA incorporates an innovative specialized recombination operator, inspired by real-parameter crossovers [12], which maintains the feasibility and legality of the offspring as solutions to the problem.

The rest of this paper is organized as follows. Section 2 presents the MK-based interpretation of the MGS problem, their similarities and differences. Section 3 introduces the new randomized greedy heuristic for the MGS problem, which constitutes one of the essential components of the proposed GA. Section 4 describes the evolutionary approach for the MGS problem. Section 5 provides an analysis of the GA performance and draws comparisons with the existing literature. Finally, Section 6 contains a summary of results and conclusions.

2. The MGS problem as searching objects for knapsacks

In the MK problem, we are given a set of objects O and a set of knapsacks K. Each object $o_j \in O$ has a profit, $p(o_j)$, and a weight, $w(o_j)$, and each knapsack $k_i \in K$ has a capacity, $C(k_i)$. The objective in the MK problem is to allocate each object to at most one knapsack in such a way that the total weight of the objects in each knapsack does not exceed its capacity and the total profit of all the objects included in the knapsacks is maximized. Its mathematical formulation, shown in Fig. 1(left) [13,14], is based on a set of binary variables $x_{o_j k_i}$, where $x_{o_j k_i} = 1$ indicates that object o_j is included in knapsack k_i, and $x_{o_j k_i} = 0$, otherwise. GAs and other metaheuristics applications to the MK problem and its variants [15–17,14] usually encode solutions as integer arrays whose lengths are equal to the number of objects, and the respective o_jth element indicates the knapsack k_i where it is included into, or an invalid value if it is not assigned to any knapsack.

In the MGS problem, the elements $s_i \in S$ may be recognized as knapsacks with capacities equal to their s_i values ($C(s_i) = s_i, \forall s_i \in S$). Then, the elements in a candidate generating set, $t_j \in T$, are objects that may be inserted in the knapsacks, with weights equal to their values ($w(o_j) = t_j, \quad \forall t_j \in T$). In this fashion, the objective of the MGS problem may be reformulated as constructing the smaller set of objects T, such that every knapsack is completely filled by including replicas of different objects from T. Noticing that no integer value j greater than the maximal element in S, S_{max}, may belong to a generating set T, we can reformulate the problem of constructing the set of objects T as the one of *selecting* those from the set $\{1, \ldots, S_{max}\}$ that will belong to T.

The MK mathematical formulation can be adapted for the MGS problem as shown in Fig. 1(right). There $x_{js_i} = 1$ indicates that integer j contributes to fill knapsack s_i ($x_{js_i} = 0$, otherwise); and consequently, x_j must be equal to 1, which expresses that integer j belongs to the generating set T.

Whereas this knapsack-based interpretation provides a mathematical adaptation and a pictorial analogy for the MGS problem, which is finding objects that properly combined fill all the knapsacks, their differences should be clearly remarked:

- The capacity constraints (2) become equality constraints, i.e., the sum of the weights of the objects in a knapsack must be equal to its capacity.
- There are not profits, so they disappear from the objective (1). Additionally, the objective is transformed into a minimization problem, to reduce the number of created objects.
- Objects may be placed in more than one knapsack, so the constraint (3) is not present.
- Knapsacks cannot carry two or more objects with the same weight, so integers j apply only once in the summation in the constraint (2).
- Objects must be created for solving the problem, whereas they are initially given in the MK problem.

  This becomes a hard restriction, since the solver has to consider combinations of every possible object. This can be addressed by searching in the space of combinations of elements in the set $\{1, \ldots, S_{max}\}$, either exploiting the mathematical formulation (Fig. 1, right) or applying a metaheuristic with integer arrays for the possible S_{max} objects. However, this becomes impractical for large S_{max} values. For example, we could not obtain any valid solution with CPLEX V12.1 and the model in Fig. 1(right) for a random instance with $|S| = 20$ and $S_{max} = 4096$ after one hour.

Regarding our proposed GA, since the direct adaptation of GAs for the MK problem to the MGS one is not viable, we will propose a randomized greedy heuristic that evaluates sets with a restricted number of samples from $\{1, \ldots, S_{max}\}$ (Section 3). Our GA will use it at different stages, namely initialization, restart, and crossover. To address extremely hard problems, a common strategy concerns to include heuristic subordinate procedures into the stages of metaheuristics [18–20].

Finally, note that given a solution for this reformulated knapsack problem, i.e. the set of objects ($O = \{o_j\}$), we may directly obtain a generating set T for S by building a set with the weight values of the objects ($T = \{t_j = w(o_j), o_j \in O\}$).

3. Randomized greedy heuristic

In this section, we propose a randomized greedy heuristic for the MGS problem, which is called RG-MGS. The design of RG-MGS (Fig. 2) is specified under the new formulation for the MGS problem presented in this paper. Therefore, one of its inputs is the set of knapsacks K associated with S, and the output is the set of created objects, O.

RG-MGS starts with all the knapsacks empty and constructs one object at a time, which is added to the current partial solution, O, until all the knapsacks are completed. Specifically, the algorithm manages the free spaces in the knapsacks, $F = \{f_1, \ldots, f_n\}$ (f_i stores the free space in knapsack k_i), and creates an object with a weight value belonging to the set $\{1, \ldots, F_{max}\}$ (the weight of the biggest possible object is equal to the greatest free space in any knapsack, F_{max}) with the aim of minimizing the global free space in the knapsacks after the insertion of the new object. To do this, RG-MGS uses