Engineering Applications of Artificial Intelligence 50 (2016) 226–235

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Randomized greedy multi-start algorithm for the minimum common integer partition problem

Manuel Lozano [a], Francisco J. Rodriguez [a,*], Daniel Peralta [a], Carlos García-Martínez [b]

[a] Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain
[b] Department of Computing and Numerical Analysis, University of Córdoba, Córdoba, Spain

ARTICLE INFO

Article history:
Received 21 January 2015
Received in revised form
30 September 2015
Accepted 2 January 2016

Keywords:
Minimum common integer partition problem
Randomized greedy algorithm
Multi-start methods

ABSTRACT

In this paper, we propose a randomized greedy multi-start algorithm for the minimum common integer partition problem. Given k multisets $S_1, \ldots, S_k$ of positive integers ($S_i = \{s_{i1}, \ldots, s_{ij}, \ldots, s_{im_i}\}$), the goal is to find the *common integer partition T with minimal cardinality*, i.e., a unique and reduced multiset T that, for each S_i, it can be partitioned into m_i multisets T_j so that the elements in T_j sum to s_{ij}. This mathematical problem is reported to appear in computational molecular biology, when assigning orthologs on a genome scale or assembling DNA fingerprints in particular. Our proposed metaheuristic approach constitutes the construction of multiple solutions by a new greedy method that embeds a diversification agent to allow diverse and promising solutions to be reached. Furthermore, we formulate an integer programming model for this problem and show that the CPLEX solver can only solve small instances of the problem. However, computational results for problem instances involving up to 1000 multisets (each one with up to 1000 elements) show that our innovative metaheuristic produces very good feasible solutions in reasonable computing times, arising as a very attractive alternative to the existing approaches.

1. Introduction

Consider two multisets $S = \{s_1, \ldots, s_m\}$ and T of positive integers. T is called an *integer partition of S*, if there exists a partition of T into multisets T_i such that for each i the sum of integers in T_i equals s_i. Given several sets $\mathcal{S} = \{S_1, \ldots, S_k\}$, if T is an integer partition of every S_i, then T is a *common integer partition of* $\mathcal{S}$. For example, given $\mathcal{S} = \{S_1 = \{3,3,4\}, S_2 = \{2,2,6\}\}$, $T = \{1,1,2,2,4\}$ is a common integer partition of $\mathcal{S}$, as shown in Fig. 1. Notice that each integer in T is associated to a unique element of S_1 and S_2 and that each element in S_1 and S_2 can be obtained by adding their associated elements in T.

The *minimum common integer partition* (MCIP) problem (Chen et al., 2006) consists of finding a common integer partition T of $\mathcal{S}$ with minimum cardinality. Considering the above example, we can check that $T' = \{2,2,3,3\}$ is another common integer partition of $\mathcal{S}$. Since the cardinality of T' is less than that of T, the former is preferred. The problem has practical applications in computational molecular biology, in particular, ortholog assignment (Chen et al., 2005a,b; Fu et al., 2006) and DNA hybridization fingerprint assembly (Valinsky et al., 2004) (see Chen et al., 2008; Woodruff, 2006 for more details on these and some other applications).

The problem is known to be NP-hard (Chen et al., 2006) (it generalizes the well-known subset sum problem Corten et al., 2009) and APX-hard (Chen et al., 2006, 2008), so exact methods rapidly become impractical even for small cases, and there is not polynomial-time $(1+\epsilon)$-approximation algorithms for arbitrarily small $\epsilon > 0$ either, unless $P{=}NP$. This means that approximation algorithms, which have the advantage of guaranteeing minimal performance, either run in polynomial time but with high/large exponents, or may not assure sufficiently good performance. Nevertheless, the progress in the literature has been mainly focused in approximation methods. (Chen et al., 2006, 2008) presented a $\frac{5}{4}$-approximation algorithm[1] for the MCIP of two sets ($|\mathcal{S}| = 2$) by using a heuristic for the maximum set packing problem, and a $3k(k-1)/(3k-2)$-approximation algorithm for MCIP problems with $k \geq 3$ multisets ($|\mathcal{S}| = k$). Woodruff (2006) proved an asymptotic worst-case performance ratio of $0.6139k$ for this problem by looking at what he called redundancy of $\mathcal{S}$, which is a quantity capturing the frequency of integers across the different

* Corresponding author.
E-mail addresses: lozano@decsai.ugr.es (M. Lozano), fjrodriguez@decsai.ugr.es (F.J. Rodriguez), dperalta@decsai.ugr.es (D. Peralta), cgarcia@uco.es (C. García-Martínez).

http://dx.doi.org/10.1016/j.engappai.2016.01.037
0952-1976/

[1] An α-approximation algorithm guarantees to provide a solution with a quality approximation ratio, with regards to an optimal solution, equal to or less than α (1 is for optimal algorithms).

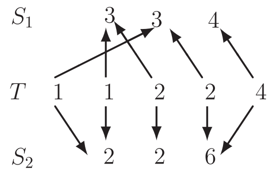

Fig. 1. An example where T is a common integer partition of S_1 and S_2.

$S_i \in \mathcal{S}$. Recently, Tong and Lin have proved an absolute 0.6k ratio, but the best asymptotic worst-case ratio was previously provided by Zhao et al. (2006), 0.5625k, by viewing this problem as a flow decomposition problem in some flow network.

Metaheuristics (Blum and Roli, 2003; Boussaïd et al., 2013; Chiong and Weise, 2011; Lozano and García-Martínez, 2010) are algorithms that usually provide excellent results for optimization problems in practice, though they do not normally guarantee a bound on the approximation ratio. Highly constrained combinatorial optimization problems such as the MCIP problem have proved to be a challenge for metaheuristic solvers (Randall and Lewis, 2010). This is a situation in which it is difficult to define an efficient neighborhood, thus no local search is available (Maniezzo et al., 2002). Therefore, the incorporation of specialized constructive greedy heuristics is often necessary in order to produce practical implementations (Randall and Lewis, 2010). Strong constraints are also very usual in engineering optimization problems (Cagnina et al., 2008) such as the *welded beam design optimization problem* (Ragsdell and Phillips, 1976) and the *speed reducer design optimization problem* (Golinski, 1973).

In this work, we tackle the MCIP problem with a *randomized greedy multi-start* (Lozano et al., 2011; Martí et al., 2013; Resende and Ribeiro, 2003) algorithm. This metaheuristic repeatedly applies a controlled randomization of a greedy method as the strategy to effectively explore the huge search space of computationally hard optimization problems. In particular, our approach adds some randomization to a greedy procedure specifically designed for generating feasible solutions for the given MCIP case. We also present an *integer quadratic programming* (IQP) formulation for the MCIP problem

The rest of this paper is organized as follows. Section 2 introduces an integer programming formulation for the MCIP problem. Section 3 presents our multi-start metaheuristic for this problem, the novel and effective greedy subordinate heuristic and the randomized framework it is included in. Section 4 provides an analysis of the performance of our algorithm and comparisons with regards to the existing approaches from the literature and the results of CPLEX. Finally, Section 5 contains a summary of results and conclusions.

2. Integer quadratic programming model

In this section, we propose an IQP formulation of the MCIP problem, which is based in the following notation. Let $\mathcal{S}_{mm}$ be the minimum among the maxima of each multiset $S_i \in \mathcal{S}$ ($\mathcal{S}_{mm} = \min_{S_i \in \mathcal{S}}\{\max(S_i)\}$), $\tilde{\mathcal{S}}$ be the summation of the elements of any multiset in $\mathcal{S}$ ($\tilde{\mathcal{S}} = \sum_{s_{ij} \in S_i} s_{ij}$, for any $S_i \in \mathcal{S}$), [2] and without loss of generality, let m be the cardinality of the multisets ($m = |S_i|$, for any $S_i \in \mathcal{S}$). In case that the multisets had different cardinalities, we could complete the smaller ones with zeros, solve this case, and remove the zeros at the end. The model, shown in Fig. 2, is based on the following three sets of indexes and three sets of variables:

[2] Chen et al. (2006) proved that the necessary and sufficient condition for a set of multisets $\mathcal{S}$ to have a common integer partition is that the multisets have the same summation over their elements.

$$\min \sum_{l=1}^{\tilde{\mathcal{S}}} x_l \quad (1)$$

subject to:

$$s_{ij} = \sum_{l=1}^{\tilde{\mathcal{S}}} c_l \cdot x_{ijl}, \ \forall i \in \{1, \cdots, k\}, j \in \{1, \cdots, m\} \quad (2)$$

$$\sum_{j=1}^{m} x_{ijl} \le 1, \ \forall i \in \{1, \cdots, k\}, l \in \{1, \cdots, \tilde{\mathcal{S}}\} \quad (3)$$

$$k \cdot x_l = \sum_{i=1}^{k} \sum_{j=1}^{m} x_{ijl}, \ \forall l \in \{1, \cdots, \tilde{\mathcal{S}}\} \quad (4)$$

$$c_l \in \{1, \cdots, \mathcal{S}_{mm}\}, x_{ijl} \in \{0,1\}, x_l \in \{0,1\} \quad (5)$$
$$\forall l \in \{1, \cdots, \tilde{\mathcal{S}}\}, \ i \in \{1, \cdots, k\}, \ j \in \{1, \cdots, m\}$$

Fig. 2. Integer quadratic programming model of the MCIP problem.

- Index $i \in \{1, \ldots, k\}$ is used to get the ith multiset $S_i \in \mathcal{S}$.
- Index $j \in \{1, \ldots, m\}$ is used to access to the jth element of a multiset, $s_{ij} \in S_i$.
- Index $l \in \{1, \ldots, \tilde{\mathcal{S}}\}$, given a set of integer candidate elements of the solution $C = \{c_1, \ldots, c_{\tilde{\mathcal{S}}}\}$ of the MCIP problem (the real solution T is a subset of C), this index is used to locate the lth candidate element.
- Integer variables $c_l \in \{0, \ldots, \mathcal{S}_{mm}\}$ represent integer candidate elements of the solution of the MCIP case. The number of c_l variables is $\tilde{\mathcal{S}}$, according to the range of the index l.
- Binary variables $x_{ijl} \in \{0,1\}$, where $x_{ijl} = 1$ indicates that variable c_l contributes to the integer partition of the value $s_{ij} \in S_i$, and $x_{ijl} = 0$ otherwise. Notice that the number of x_{ijl} variables is $k \cdot m \cdot \tilde{\mathcal{S}}$.
- Binary variables $x_l \in \{0,1\}$, where $x_l = 1$ indicates that the candidate element c_l belongs to the solution T of the MCIP problem, and $x_l = 0$ otherwise. Notice that these variables can be deduced from the values of x_{ijl} variables. They are introduced in order to simplify the IQP model.

In the model shown in Fig. 2:

- Eq. (1) specifies the objective, which is to obtain a solution T with a minimal number of elements.
- Constraints in Eq. (2) establish the integer partition constraints, i.e., the summation of the candidate elements c_l that contributes to the integer partition of s_{ij}, according to the values of the x_{ijl} variables, must be exactly equal to s_{ij}.
- Constraints in Eq. (3) ensure that each candidate component of the solution participates at most in the integer partition of one element s_{ij} per multiset S_i.
- Constraints in Eq. (4) make sure that the selected candidate components of the solution, those with $x_l = 1$, participate exactly in m integer partitions, or none otherwise. Together with constraints (3), they guarantee that the selected candidate components participate in the integer partition of one and only one element per multiset.
- Constraints in Eq. (5) are bound constraints. Notice that no integer value greater than $\mathcal{S}_{mm}$ can belong to a solution of the MCIP problem, because it cannot participate in any integer partition of multisets without elements greater than $\mathcal{S}_{mm}$.
- Finally, notice that $\tilde{\mathcal{S}}$ candidate components are sufficient because the all-ones (exactly $\tilde{\mathcal{S}}$ ones) is a trivial solution, the worst feasible one.