Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Lagrangian relaxation versus genetic algorithm based matheuristic for a large partitioning problem

Oliver G. Czibula *, Hanyu Gu, Yakov Zinder

School of Mathematical and Physical Sciences, University of Technology Sydney, 15 Broadway, Ultimo, NSW 2007, Australia

ARTICLE INFO

Article history:
Received 11 May 2016
Received in revised form 15 December 2016
Accepted 1 January 2017
Available online xxxx

Keywords:
Partitioning problem
Integer programming
Lagrangian relaxation
Genetic algorithm
Heuristic

ABSTRACT

This paper is concerned with a partitioning problem. One of the applications, and the motivation for this research, is the problem of class formation for training and retraining sessions at large electricity distributors. Two different approaches are developed. One is based on the Quadratic Multiple Knapsack formulation and Lagrangian relaxation. The other is a matheuristic developed as an amalgamation of Genetic Algorithms and Integer Programming. The approaches are tested by means of computational experiments. Both heuristics outperformed the direct application of quadratic programming, with the Lagrangian relaxation based approach performing the best on average, and the Genetic Algorithm based approach performing the best on the larger test cases.

1. Introduction

This paper is concerned with the combinatorial optimisation problem where the union of disjoint sets $F_1, \ldots, F_K$ (referred to as families) must be partitioned into sets $S_1, \ldots, S_N$ in such a way that each S_i is either empty, or has cardinality between a_i and b_i, i.e. $a_i \leq |S_i| \leq b_i$, and each nonempty set S_i must contain elements from at least p_i and at most q_i families. All a_i, b_i, p_i, and q_i are given.

There is a cost $c_{i,j}$ associated with the inclusion of each element j of $F_1 \cup \ldots \cup F_n$ into each set S_i. For any two distinct families F_k and F_l, and any set S_i, the cost $b_{k,l}$ is incurred for the simultaneous presence of elements from both families F_k and F_l in S_i. This cost does not depend on the number of elements from F_k and F_l in S_i. The objective is to minimise the total cost associated with the partition $S_1, \ldots, S_N$.

The above mentioned combinatorial optimisation problem was motivated in particular by the problem of class formation for training and retraining at large electricity distributors. Such organisations typically have thousands of workers of different types, and some also provide training to their contractors and to third parties. Due to the multitude of hazards that exist when working with high voltages, at heights, or in confined spaces, local laws often require all such workers to undergo regular safety, technical, and professional training.

Many of these workers, referred to as students for the purpose of this study, have very different learning outcomes from these courses, different learning styles, different levels of education or English proficiency, and different levels of technical proficiency for certain tasks. Although assigning only one type of student into any class enables the training delivery to be better tailored to the needs of the specific group, in some cases it is beneficial to combine several compatible types of

* Corresponding author.
E-mail address: oliver.g.czibula@student.uts.edu.au (O.G. Czibula).

http://dx.doi.org/10.1016/j.tcs.2017.01.003
0304-3975/

students in the same class to facilitate the exchange of knowledge and experience between student types. Furthermore, due to the cost of delivering training and the scarcity of training resources, it is often not possible to run segregated classes, and some undesirable blending of student types may be necessary. Each class can either be empty, in which case it will be cancelled with no penalty, otherwise it must contain between a minimum and a maximum number of students, and between a minimum and maximum number of student types.

Each potential class is associated with a specific date at which it will be conducted. On the other hand, each student, who is of exactly one type, should complete their training by a certain date, resulting in a penalty (cost) incurred for assigning a student to a particular class. Another type of penalty reflects the incompatibility of the student types assigned to the same class. The objective is to minimise the total penalty. When a single course is considered, the above leads to the combinatorial optimisation problem studied in this paper. We denote the problem studied in this paper to be "Partition into Inhomogeneous Classes" (PIC).

The considered combinatorial optimisation problem is distinct from, but shares certain similarity with, the Graph Partitioning Problem (GPP) [1]. This problem arises in various practical situations, for example: partitioning subroutines that are compiled into computer code into clusters, such that the communication between clusters is minimised [2]; assigning data or processes evenly to processors, such that inter-process communication is minimised in parallel computing environments [3]; in aircraft control, where the flow of aircraft within blocks is maximised and flow of aircraft between blocks is minimised [4]; partitioning the electronic subcircuits in very large-scale integration (VLSI) systems, such that the electrical connections between partitions is minimised [5]; partitioning a power network into so-called islands to prevent the propagation of cascading failures [6,7]; for route planning along road networks [8,9]; and in many other domains. Although the combinatorial optimisation problem considered in our paper and the GPP are different, each provides an insight in the design of solution techniques for the other.

Several particular cases of the GPP, including Integer Programming (IP) models for each case, are discussed in [10]. The authors take advantage of the graph structure when clustering. They also discuss several valid inequalities and facet-defining inequalities for the GPP. A column generation approach is presented in [2], which the authors tested on graphs with between 30 and 61 nodes, and between 47 and 187 edges. For the 12 test cases the authors considered, their proposed approach provided integer solutions for all but two, and for those, solutions obtained by a branch-and-bound scheme were very close to the fractional solutions provided by column generation. Our paper also presents IP models for the considered partitioning problem, but is concerned with a Lagrangian relaxation and a matheuristic, and is focused on solving large instances with hundreds of classes and thousands of students.

The problem considered in our paper is also related to the Quadratic Multiple Knapsack Problem (QMKP), which is a combination of the multiple knapsack and quadratic knapsack problems. The QMKP received little attention in the literature until recently, and most solution approaches are based on meta-heuristics [11–14]. In particular, greedy and genetic algorithms (GA) for the QMKP are discussed in [14]. The author presents two greedy heuristics that build solutions by choosing objects according to their value densities, and two GA heuristics. One GA is a standard implementation, while the other is extended with greedy techniques that probabilistically favour objects of high value density. The algorithms are tested on 20 problem instances, and the extended GA is reported to perform best on all but one test case. In our paper we present a matheuristic that is an amalgamation of Genetic Algorithms and IP.

A Lagrangian relaxation based approach to solving the QMKP exactly is presented in [15], whereby a tighter bound is computed using the subgradient method. The proposed approach is able to optimally solve instances with up to 400 binary variables. In our paper we also consider a Lagrangian relaxation based approach, but focus on problems with between 17 697 and 267 393 binary variables (and between 42 656 and 367 232 for the linearised models). The size of our problem, together with the complexity results presented in our paper, leads to the aim of obtaining a good approximation solution, which is compared with the alternative approach that is an amalgamation between Genetic Algorithms and IP.

The considered problem was previously studied in [16]. In contrast to [16] our current paper is focused on significantly larger test cases, which has greater significance for practical applications. The change of focus necessitated the revision of the proposed approaches, resulting in superior performance of the designed algorithms on the larger test cases.

The remainder of this paper is organised as follows. Section 2 presents a linearly constrained quadratic programming formulation, as well as its linearisation, for the studied problem. Section 3 analyses the complexity of the considered partitioning problem. Section 4 discusses a Lagrangian relaxation of the quadratic program. Section 5 presents the first approach, which is a heuristic based on the Lagrangian relaxation. Section 6 presents the second approach, which is an amalgamation of Genetic Algorithms and Integer Programming. Section 7 discusses the results of the computational experimentation. Our concluding remarks are given in Section 8.

2. Quadratic programming formulation

Let $\mathcal{N} = \{1, \cdots, N\}$, $\mathcal{M} = \{1, \cdots, M\}$, and $\mathcal{K} = \{1, \cdots, K\}$ be the set of available classes, the set of students to be assigned, and the set of student types respectively. Denote the penalty of assigning student $j \in \mathcal{M}$ to class $i \in \mathcal{N}$ by $c_{i,j}$, and the penalty of pairing student types $k \in \mathcal{K}$ and $l \in \mathcal{K}$ together in the same class by $b_{k,l}$. Each student has exactly one type, and the set of students who are of type k is represented by T_k, $k \in \mathcal{K}$. Each student must be assigned to exactly one class, but not all classes must be run. Each class $i \in \mathcal{N}$ that is run must contain at least a_i and at most b_i students, and at least p_i and at most q_i student types. Students or student types cannot be assigned to classes that are not run. The binary

Please cite this article in press as: O.G. Czibula et al., Lagrangian relaxation versus genetic algorithm based matheuristic for a large partitioning problem, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.01.003