A dual ascent heuristic for obtaining a lower bound of the generalized set partitioning problem with convexity constraints

Stefania Pan, Roberto Wolfler Calvo, Mahuna Akplogan, Lucas Létocart, Nora Touati

A dual ascent heuristic for obtaining a lower bound of the generalized set partitioning problem with convexity constraints

Stefania Pan[a,b,*], Mahuna Akplogan[b], Lucas Létocart[a], Nora Touati[b], Roberto Wolfler Calvo[a]

[a] *LIPN (UMR CNRS 7030) - Institut Galilée - Université Paris-Nord, 99, avenue Jean-Baptiste Clément, 93430 Villetaneuse, France*
[b] *Horizontal Software, 9, Rue de l'Isly, 75008 Paris, France*

Abstract

In this paper we propose a dual ascent heuristic for solving the linear relaxation of the generalized set partitioning problem with convexity constraints, which often models the master problem of a column generation approach. The generalized set partitioning problem contains at the same time set covering, set packing and set partitioning constraints. The proposed dual ascent heuristic is based on a reformulation and it uses Lagrangian relaxation and subgradient method. It is inspired by the dual ascent procedure already proposed in literature, but it is able to deal with right hand side greater than one, together with under and over coverage. To prove its validity, it has been applied to the minimum sum coloring problem, the multi-activity tour scheduling problem, and some newly generated instances. The reported computational results show the effectiveness of the proposed method.

Keywords: dual ascent heuristic, Lagrangian relaxation, subgradient method, generalized set partitioning
2010 MSC: 90-02, 90C10

1. Introduction

Set partitioning, set covering and set packing problems are fundamental models in combinatorial optimization and they are concerned with finding an optimal family of subsets of elements from a set. They are formally presented as follows: suppose we are given a finite set $M = \{1, \dots, m\}$, a finite set $N =$

*Corresponding author: Tel. +33 1 4940 4071
Email addresses: `pan@lipn.univ-paris13.fr` (Stefania Pan), `makplogan@horizontalsoftware.com` (Mahuna Akplogan), `letocart@lipn.univ-paris13.fr` (Lucas Létocart), `ntouati@horizontalsoftware.com` (Nora Touati), `wolfler@lipn.univ-paris13.fr` (Roberto Wolfler Calvo)

$\{1, \ldots, n\}$, and a finite collection $\{M_j\}_{j \in N}$ of subset of M. Furthermore, let $A = (a_{ij})$ be a 0–1 $m \times n$ matrix, where each column $j \in N$ is the characteristic vector of M_j, i.e., $a_{ij} = 1$ if $i \in M_j$; let c be an n–dimensional integer vector, e the m-vector of all one, and $x \in \{0,1\}^n$ the vector of decision variables, where, for each $j \in N$, $x_j = 1$ if the subset M_j is selected in the solution. The set covering problem is to find a minimum cost cover of M, i.e., a subset $F \subseteq N$ such that $\bigcup_{j \in F} M_j = M$, and it can be formulated with an integer program as follows:

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax \geq e \\ & x \in \{0,1\}^n. \end{aligned}$$

The set packing problem is to find a maximum cost packing of M, i.e., a subset $F \subseteq N$ such that $M_j \cap M_k = \emptyset$ for all $j, k \in N$, $j \neq k$, and it can be formulated with an integer program as follows:

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax \leq e \\ & x \in \{0,1\}^n. \end{aligned}$$

Finally, the set partitioning problem is to find a minimum cost partition of M, i.e., a subset $F \subseteq N$ which is both a cover and packing, and it can be formulated with an integer program as follows:

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax = e \\ & x \in \{0,1\}^n. \end{aligned}$$

A comprehensive survey on theory and applications of these three models is presented for example in Balas & Padberg (1976) and Vemuganti (1998). The set partitioning, covering and packing models are strictly related. Indeed, a set partitioning problem can be restated as a set covering problem, and there is equivalence between set partitioning and set packing problems (see Balas & Padberg (1976)). The three problems can be combined into a *unified set partitioning problem*, which allows under and over coverage, yielding set partitioning and set covering constraints respectively. This model was first proposed by Darby-Dowman & Mitra (1985) and more recently by Rasmussen & Larsen (2011).

The problem addressed in this paper considers the generalized version of the unified set partitioning constraints (i.e., constraints (2)), where the right hand side is allowed to be a positive integer vector $b \in \mathbb{Z}_+^m$. In addition, it considers convexity constraints (i.e., constraints (3)). The problem is stated as follows:

$$
\begin{aligned}
\min \quad & c^\top x + \underline{c}^\top z + \overline{c}^\top t && (1)\\
\text{s.t.} \quad & Ax + z - t = b && (2)\\
& Ex = e && (3)\\
& z, t \geq 0 && (4)\\
& x \in \{0,1\}^n && (5)
\end{aligned}
$$

where $\underline{c}$ and $\overline{c}$ are the m-vectors of under and over coverage costs, z and t are the m-vectors of decision variables controlling whether or not constraints (2) are under or over covered respectively, and $E \in \{0,1\}^{p\times n}$ is the coefficient matrix of the convexity constraints. Matrix E has the peculiarity that each column has only one entry equal to 1. Model (1)-(5) is easily transformed in the unified set partitioning problem by removing constraints (3) and by setting vector b equal to 1. In addition, it is not difficult to see that constraints (2) catches set partitioning, covering and packing constraints. In order to obtain set partitioning constraints, we only need to set all components of vectors $\underline{c}$ and $\overline{c}$ equal to a sufficiently large positive number, thereby preventing both under and over coverage. In this case, if a feasible solution exists for set partitioning, the unified model will have the same optimal solution. Similarly, in order to solve set covering (resp. packing) constraints, we need to set all components of vector $\underline{c}$ (resp. $\overline{c}$) equal to a large positive number, and $\overline{c}$ (resp. $\underline{c}$) equal to 0. When set partitioning is used as model, an exact cover may not exist or a solution with under and over coverage could be more interesting than a set partitioning solution. For instance, in the multi-activity tour scheduling problem (cf. Section 6.2), satisfying constraints (2) without allowing under and over coverage may result impossible due to the fluctuation of the demand. The proposed method for solving the generalized set partitioning problem with convexity constraints (i.e., model (1)-(5)), usually encountered in column generation algorithms, is a generalization of the dual ascent procedure (DA). The DA is based on a parametric reformulation and it uses Lagrangian relaxation and subgradient method. The novelty of the proposed method consists in managing the generalized set partitioning constraints, where the right hand side can be different from the unit vector, and under and over coverage are allowed.

Two different applications are used to prove the validity of the proposed method. The first one is the minimum sum coloring problem, which is a variant of the vertex coloring problem. A review of recent algorithms to solve the minimum sum coloring problem can be found in Jin et al. (2017). Recently, Furini et al. (2018) have proposed a set covering based formulation for this problem. The second application is the multi-activity tour scheduling problem, which is a particular problem in personnel scheduling. We refer to Ernst et al. (2004) and Alfares (2004) for comprehensive surveys. Gérard et al. (2016) propose a formulation based on the generalized set partitioning in order to take into account under and over coverage of the demand. Since none of these two applications consider together constraints of set partitioning, covering, packing and generalized set partitioning, the approach has been used also to solve newly generated instances.

The paper is organised as follows: the next section contains a literature review. Section 3 shows how the generalized set partitioning problem with convexity constraints arises in Dantzig-Wolfe decomposition. The dual ascent heuristic and the mathematical tools used are presented in Section 4. Section 5 briefly recalls how the problem can be solved by Lagrangian relaxation. Different applications, together with some computational results are presented in Sections 6 and 7, while Section 8 presents the conclusions.

2. Literature review

Set partitioning (SPT), covering (SC) and packing (SP) problems have been used to model a great variety of problems in the literature, such as crew scheduling, cutting stock, facilities location, graphs coloring, personnel scheduling, vehicle routing, timetabling and many others. Below we report on some examples for each type of problem. The list is limited since it is out of the scope of this paper being exhaustive about the applications.

Set covering. Problems where every customer is served by some location, vehicle or person often requires the set covering structure. Balas & Carrera (1996) formulate airline crew scheduling and bus driver scheduling using a SC model. Ceria et al. (1998a) propose a large-scale SC model for railways crew scheduling. Muter et al. (2010) make use of SC to model vehicle routing problem with time windows, while Malaguti et al. (2011) address the vertex coloring problem.

Set partitioning. When every customer must be served exactly once, the problem takes the set partitioning structure. The vehicle routing problem and its variants widely use formulations based on the SPT model, originally proposed by Balinski & Quandt (1964). Among many different papers, we cite Baldacci et al. (2008), which address the capacitated vehicle routing problem. Desaulniers et al. (1997) use a SPT model to solve a crew scheduling problem for Air France. In Rezanova & Ryan (2010), a recovery problem for train driver duties is modelled as SPT. Brønmo et al. (2010) use SPT for a ship scheduling problem.

Set packing. The goal of satisfying as much demand as possible, without creating conflicts, generally requires the set packing format. Rönnqvist (1995) propose a SP model for a cutting stock problem. Mingozzi et al. (1998) used a SP formulation for a resource constrained project scheduling problem. Rossi & Smriglio (2001) considered a SP formulation for a ground holding problem. In Lusby et al. (2011) is given a survey of models and methods for railway track allocation, including formulations that rely on the SP model.

In the literature there are also papers addressing problems whose formulation combines partitioning, covering and packing constraints. Below we report a short and non-exhaustive list of examples. Boschetti & Maniezzo (2015) use an extended covering formulation to model a city logistics problem, where covering constraints impose all clients to be served at least once, while set packing-like constraints, where the right hand side can be different from one, limit the number of vehicles available in each work shift. Baldacci et al. (2016) use a set partitioning based model for the vehicle routing problem with transhipment facilities, where clients have to be served exactly once, while facilities may be used or not. A very similar model combining partitioning and packing constraints is used by Baldacci et al. (2017) for the capacitated ring-star problem. Cacchiani et al. (2014) present a set covering based formulation for the periodic vehicle routing, where packing constraints limit the daily fleet size, while covering constraints define the relation between combinations and routes and ensure that at least one combination is selected for each client. All the examples reported above deal with routing problems, while, as far as we know, the generalized set partitioning problem has been addressed only in personnel scheduling problem (see Gérard et al. (2016)).

Most of the models cited above have an exponential number of variables, since based on a Dantzig-Wolfe decomposition approach. Typically, only a small fraction of them is needed to prove optimality and this aspect makes column generation an interesting technique. Column generation is an iterative process that solves a restricted master problem and one or several subproblems (see Desrosiers & Lübbecke (2005)).

Primal or dual simplex methods are commonly used to solve the reduced master problem. Despite all the progress in linear programming, solving these linear programs can be a challenge. Various heuristics to obtain optimal and near-optimal dual solutions have been proposed. Fisher & Kedia (1990) solve a mixed set covering-partitioning model using dual heuristics that include greedy and 3-opt heuristics and, in some cases, the subgradient method. It is applied to the dual of the linear relaxation to provide lower bounds for a branch-and-bound algorithm. Ceria et al. (1998b) propose a primal-dual Lagrangian heuristic for the set covering problem. The proposed method solves simultaneously the Lagrangian relaxation of the primal and the dual problems. Then, primal and dual multipliers are used for fixing variables and reducing the problem. An extension of the subgradient method, called volume algorithm, has been proposed by Barahona & Anbil (2002) to produce a valid lower bound as well as an approximation of the primal solution. More recently, Boschetti et al. (2008) presented both a dual ascent procedure and an exact method for the set partitioning problem. The dual ascent procedure makes use of parametric and Lagrangian relaxations to produce feasible dual solutions of the linear relaxation of the set partitioning problem. The exact method described uses the dual solution found by the heuristic to define a reduced problem with a limited subset of variables that is solved by an integer programming solver. The reduced problem is augmented until optimality can be proven. A similar heuristic approach has been later proposed by Boschetti & Maniezzo (2015) to solve a real-world city logistic problem, for which the reduced master problem consists of an extended set covering problem. An exact solution framework that employs dual ascent procedures was proposed by Baldacci et al. (2008). The method is used for the capacitated vehicle routing problem, but it can be tailored to solve several variants of the vehicle routing problem, as shown in Baldacci et al. (2010). Indeed, Baldacci et al. (2011a) address the pickup and delivery problem with time windows, Baldacci et al. (2011b) consider the periodic routing problem, while Baldacci et al. (2016) recently solve the vehicle routing problem with transhipment facilities. In the following we explain how the dual ascent heuristic is the extension of the dual ascent procedure, and how it is used for solving the linear relaxation of the generalized set partitioning problem with convexity constraints. The proposed method adopts reformulation, Lagrangian relaxation and subgradient method used in the procedure proposed by Boschetti et al. (2008) for the set partitioning problem. The goal is to deal with generalized set partitioning constraints, where both under and over coverage are allowed, and the right hand side can be different from one.

3. Problem description

Let us consider the following *compact* mixed-integer linear program formulation representing the problem at hand that we want to solve:

$$\begin{aligned} (P) \quad \min \quad & c^\top \tilde{x} + \underline{c}^\top z + \overline{c}^\top t && (6)\\ \text{s.t.} \quad & \tilde{A}\tilde{x} + z - t = b && (7)\\ & D\tilde{x} = d && (8)\\ & \tilde{x} \in \{0,1\}^{\tilde{n}} && (9)\\ & z, t \geq 0 && (10) \end{aligned}$$

where $c \in \mathbb{R}^{\tilde{n}}$, $\underline{c} \in \mathbb{R}^m$ and $\overline{c} \in \mathbb{R}^m$ are the vectors of costs, $\tilde{A} \in \{0,1\}^{m\times\tilde{n}}$ and $D \in \mathbb{R}^{l\times\tilde{n}}$ are the matrices of coefficients. $\tilde{A}$ describes the *linking constraints* while D may have a block diagonal structure. Furthermore, $b \in \mathbb{Z}^m_+$ and $d \in \mathbb{R}^l$ are the right hand side vectors, $\tilde{x} \in \{0,1\}^{\tilde{n}}$, $z \in \mathbb{R}^m_+$ and $t \in \mathbb{R}^m_+$ are the decision variables. Costs $\underline{c}$ and $\overline{c}$ are assumed to satisfy $-\overline{c} \leq \underline{c}$. Let $X = \{\tilde{x} \in \{0,1\}^{\tilde{n}} : D\tilde{x} = d\}$ be the finite set defined by constraints (8) and (9). Model (6)-(10) can be reformulated by applying the Dantzig-Wolfe (DW) decomposition for mixed-integer programs, and the following extensive formulation is obtained:

$$\begin{aligned} (EP) \quad \min \quad & \sum_{j\in N} c_j x_j + \underline{c}^\top z + \overline{c}^\top t && (11)\\ \text{s.t.} \quad & \sum_{j\in N} a_j x_j + z - t = b && (12)\\ & \sum_{j\in N} x_j = 1 && (13)\\ & x \in \{0,1\}^n && (14)\\ & z, t \geq 0 && (15) \end{aligned}$$

where $c_j = c^\top \tilde{x}_j$, $a_j = \tilde{A}\tilde{x}_j$ and $\tilde{x}_j$ are the extreme points of $conv(X)$. Moreover, if constraints (8) have a block diagonal structure, X can be decomposed. Suppose $D^k \in \mathbb{R}^{g\times h}$ is the k-th block of D, for $k = 1, \ldots, p$, and $d^k \in \mathbb{R}^g$ is the corresponding block in d. We assume for simplicity that each block has the same dimensions, hence $l = pg$ and $\tilde{n} = ph$. X decomposes in p finite sets $X_1, \ldots, X_p$, i.e., $X = X_1 \times \cdots \times X_p$, where $\times$ denotes the Cartesian product and $X_k = \{\tilde{x}^k \in \{0,1\}^h : D^k\tilde{x}^k = d^k\}$. We denote by $K = \{1, \ldots, p\}$ the set of indices in the partition, $\tilde{N}_k = \{1, \ldots, \tilde{n}_k\}$ the set of indices j of extreme points $\tilde{x}^k_j$ of $conv(X_k)$, and $M = \{1, \ldots, m\}$ the set of indices of constraints (7). The DW reformulation of problem (P) takes the following form:

$$
\begin{aligned}
(EP) \quad \min \quad & \sum_{k\in K}\sum_{j\in\tilde{N}_k} c^k_j x^k_j + \sum_{i\in M} \underline{c}_i z_i + \sum_{i\in M} \overline{c}_i t_i && && (16)\\
\text{s.t.} \quad & \sum_{k\in K}\sum_{j\in\tilde{N}_k} a^k_j x^k_j + z - t = b && && (17)\\
& \sum_{j\in\tilde{N}_k} x^k_j = 1 && \forall k \in K && (18)\\
& x^k_j \in \{0,1\} && \forall k \in K, j \in \tilde{N}_k && (19)\\
& z_i, t_i \geq 0 && \forall i \in M. && (20)
\end{aligned}
$$

We assume that the decomposition results in 0–1 coefficient matrix, i.e., $a^k_j \in \{0,1\}^m$. The DW decomposition results in an extended formulation with an exponential number of variables. The linear relaxation of problem (EP), obtained relaxing the integrality constraints on binary variables x^k_j, provides a lower of the problem. Typically, only a small fraction of the variables is needed to prove optimality of the linear relaxation, and this aspect makes column generation an interesting technique. Column generation is a mathematical programming technique that enables to solve a wide class of large linear problems by iteratively adding the variables of the model. It replaces the linear relaxation of (EP) by a restricted version, where only a subset of variables is considered. The resulting problem is called restricted master problem (RMP), which is iteratively solved for obtaining each time dual variables. These latter are passed to the subproblems, which are solved looking for new variables with negative reduced cost to be added to the RMP. Optimality of the linear relaxation of (EP) is achieved as soon as no negative reduced cost variable is found. A deeper description of the column generation mechanism is out of the scope of this paper, but we address the interested reader to Desrosiers & Lübbecke (2005) to have an exhaustive insight to the subject. As a consequence, the RMP must be solved several times quickly in order to update the dual variable passed to the subproblems. Let us consider the linear relaxation of the RMP, made up of subsets of columns

$N_k \subseteq \tilde{N}_k$ for all k. The restricted master problem (RMP) appears as follows:

$$
\begin{aligned}
(RMP) \quad \min \quad & \sum_{k \in K} \sum_{j \in N_k} c_j^k x_j^k + \sum_{i \in M} \underline{c}_i z_i + \sum_{i \in M} \overline{c}_i t_i && && (21)\\
\text{s.t.} \quad & \sum_{k \in K} \sum_{j \in N_k^i} x_j^k + z_i - t_i = b_i && \forall i \in M && (22)\\
& \sum_{j \in N_k} x_j^k = 1 && \forall k \in K && (23)\\
& x_j^k \geq 0 && \forall k \in K, j \in N_k && (24)\\
& z_i, t_i \geq 0 && \forall i \in M && (25)
\end{aligned}
$$

where $N_k^i = \{j \in N_k : (a_j^k)_i = 1\}$ denotes the set of columns $j \in N_k$ that cover row $i \in M$. Problem (RMP) is the linear relaxation of the generalized set partitioning problem with convexity constraints. It consists in selecting columns that satisfy coverage (cf. constraints (22)), minimizing the total cost given by the columns and the under and over coverage (cf. objective function (21)). Constraints (23) state that each row $k \in K$ is covered by a convex combination of extreme points of $conv(X_k)$. Finally, constraints (24) and (25) define the domain of the variables. Let us denote $u = (u_1, \ldots, u_m)$ and $v = (v_1, \ldots, v_p)$ the vectors of the dual variables associated respectively with constraints (22) and (23). The p subproblems, solved to generate new variables, have the following formulation: for $k \in K$

$$
\begin{aligned}
(SP_k) \quad c_k^* = & \min c^\top \tilde{x}^k - u^\top \tilde{A} \tilde{x}^k - v_k \\
\text{s.t.} \quad & \tilde{x}^k \in X_k. && (26)
\end{aligned}
$$

The objective function evaluates the so called *reduced costs*. The role of the subproblems is to provide columns that price out profitably, i.e., that have negative reduced costs, or to prove that none of them exists and, therefore, optimality of the linear relaxation of (EP) is achieved.

4. A dual ascent procedure

In this section we describe a dual ascent heuristic to efficiently compute dual solutions of problem (RMP). The dual problem of (RMP) has the following formulation:

$$
\begin{aligned}
(D) \quad \max z_D = & \sum_{i \in M} b_i u_i + \sum_{k \in K} v_k && && (27)\\
\text{s.t.} \quad & \sum_{i \in R_j^k} u_i + v_k \leq c_j^k && \forall k \in K, j \in N_k && (28)\\
& -\overline{c}_i \leq u_i \leq \underline{c}_i && \forall i \in M && (29)
\end{aligned}
$$

where $R_j^k = \{i \in M : (a_j^k)_i = 1\}$ is the set of rows i covered by column $j \in N_k$. We remark that due to the assumption that $-\overline{c} \leq \underline{c}$, constraints (29) do not lead to an empty solution space. The dual ascent heuristic is based on a reformulation of (RMP). Then, coverage and convexity constraints are relaxed by means of penalty vectors, to derive the Lagrangian relaxation. We explain in details these techniques in the next sections.

4.1. Reformulation

The first element of the dual ascent heuristic consists of a reformulation of problem (RMP). The main idea is to associate to each binary variable x_j^k, a binary variable for each row $i \in M$ covered by the associated column. Therefore, each variable x_j^k is associated with a set of $|R_j^k| + 1$ binary variables $\hat{y}_j^k$, y_j^h for all $h \in R_j^k$, according the following expression:

$$x_j^k = \frac{1}{|R_j^k| + 1}\Big(\sum_{h \in R_j^k} y_j^h + \hat{y}_j^k\Big) \qquad \forall k \in K, j \in N_k. \tag{30}$$

By replacing variables x_j^k in (RMP) according the expression (30) , we obtain the reformulation of the reduced master problem (RMP), which is called $(RRMP)$ and appears as follows:

$$(RRMP)\ \min \sum_{k \in K} \sum_{j \in N_k} \frac{c_j^k}{|R_j^k| + 1}\Big(\sum_{h \in R_j^k} y_j^h + \hat{y}_j^k\Big) + \sum_{i \in M} \underline{c}_i z_i + \sum_{i \in M} \overline{c}_i t_i \tag{31}$$

$$\text{s.t.} \sum_{k \in K} \sum_{j \in N_k^i} \frac{1}{|R_j^k| + 1}\Big(\sum_{h \in R_j^k} y_j^h + \hat{y}_j^k\Big) + z_i - t_i = b_i \quad \forall i \in M \tag{32}$$

$$\sum_{j \in N_k} \frac{1}{|R_j^k| + 1}\Big(\sum_{h \in R_j^k} y_j^h + \hat{y}_j^k\Big) = 1 \qquad \forall k \in K \tag{33}$$

$$z_i, t_i \geq 0 \qquad \forall i \in M \tag{34}$$

$$y_j^h, \hat{y}_j^k \in \{0, 1\} \qquad \forall k \in K, j \in N_k, h \in R_j^k. \tag{35}$$

We will see in the next section that the use of this reformulation allows the addition of new constraints to the Lagrangian relaxation, leading to Lagrangian subproblems with possibly higher objective function values (cf. Corollary 1).

4.2. Lagrangian relaxation

Problem $(RRMP)$ is relaxed dualizing constraints (32) and (33) by means of penalty vectors $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^p$ respectively. However, the relaxation considered by the dual ascent heuristic is not exactly the Lagrangian one in the classical sense. This is due to the fact that further constraints are added. These constraints, i.e., (37) and (38), are similar to the relaxed ones, but they state that the coverage of each row is satisfied independently from all the other rows. Note that constraints (37) and (38) are redundant if added to problem $(RRMP)$, but they strength the Lagrangian subproblem (cf. Corollary 1, inequality (58)). Let us define the Lagrangian costs $\underline{c}_i(\lambda) := \underline{c}_i - \lambda_i$ and $\overline{c}_i(\lambda) := \overline{c}_i + \lambda_i$. Furthermore, let us define the Lagrangian costs $c_j^k(\lambda, \mu)$ according the following:

$$c_j^k(\lambda, \mu) := \frac{c_j^k - \sum_{i \in R_j^k} \lambda_i - \mu_k}{|R_j| + 1}.$$

The Lagrangian subproblem $(LRP(\lambda, \mu))$ has the following formulation:

$$\begin{aligned} z_{LRP}(\lambda, \mu) = \min \sum_{i \in M} \Big(\sum_{k \in K} \sum_{j \in N_k^i} c_j^k(\lambda, \mu) y_j^i + \underline{c}_i(\lambda) z_i + \overline{c}_i(\lambda) t_i + b_i \lambda_i \Big) + & \\ + \sum_{k \in K} \Big(\sum_{j \in N_k} c_j^k(\lambda, \mu) \hat{y}_j^k + \mu_k \Big) & \quad (36) \\ \text{s.t.} \sum_{k \in K} \sum_{j \in N_k^i} y_j^i + z_i - t_i = b_i \qquad \forall i \in M & \quad (37) \\ \sum_{j \in N_k} \hat{y}_j^k = 1 \qquad \forall k \in K & \quad (38) \\ z_i, t_i \geq 0 \qquad \forall i \in M, & \quad (39) \\ y_j^i, \hat{y}_j^k \in \{0, 1\} \qquad \forall k \in K, j \in N_k, i \in R_j^k. & \quad (40) \end{aligned}$$

Note that the sum $\sum_{i \in M} \sum_{k \in K} \sum_{j \in N_k^i} y_j^i$ is obtained by rearranging the indices of $\sum_{k \in K} \sum_{j \in N_k} \sum_{h \in R_j^k} y_j^h$. The addition of constraints (37) and (38) does not prevent problem $(LRP(\lambda, \mu))$ from being decomposable into $m+p$ subproblems, one for each row $i \in M$, and one for each row $k \in K$. In the following, we show how an optimal solution of each subproblem i and k can be defined. These solutions are then used to determine an optimal solution of $(LRP(\lambda, \mu))$. We first consider an index $k \in K$ and the corresponding subproblem $(LRP^k(\lambda, \mu))$:

$$\begin{aligned} z_{LRP}^k(\lambda, \mu) = \min \sum_{j \in N_k} c_j^k(\lambda, \mu) \hat{y}_j^k + \mu_k & \quad (41) \\ \text{s.t.} \sum_{j \in N_k} \hat{y}_j^k = 1 & \quad (42) \\ \hat{y}_j^k \in \{0, 1\} \qquad \forall j \in N_k. & \quad (43) \end{aligned}$$

Let $j_k \in N$ be the column covering row k such that $j_k = \arg\min_{j \in N_k} c_j^k(\lambda, \mu)$. An optimal solution of problem $(LRP^k(\lambda, \mu))$ can be easily obtained setting $\hat{y}_{j_k}^k = 1$ and $\hat{y}_j^k = 0$ for all $j \in N_k \setminus \{j_k\}$. The optimal objective function value results

$$z_{LRP}^k(\lambda, \mu) = c_{j_k}^k(\lambda, \mu) + \mu_k. \quad (44)$$

We now consider the subproblems $(LRP^i(\lambda, \mu))$ concerning index $i \in M$, which has the following formulation:

$$z^i_{LRP}(\lambda,\mu) = \min \sum_{k\in K}\sum_{j\in N^i_k} c^k_j(\lambda,\mu) y^i_j + \underline{c}_i(\lambda) z_i + \overline{c}_i(\lambda) t_i + b_i\lambda_i \tag{45}$$

$$\text{s.t.} \sum_{k\in K}\sum_{j\in N^i_k} y^i_j + z_i - t_i = b_i \tag{46}$$

$$z_i, t_i \geq 0 \tag{47}$$

$$y^i_j \in \{0,1\} \qquad \forall k\in K, j\in N^i_k. \tag{48}$$

In order to find an optimal solution of problem $(LRP^i(\lambda,\mu))$, we consider the Lagrangian costs $c^k_j(\lambda,\mu)$ in ascending order $(c^{k_1}_{j_1}(\lambda,\mu) \leq c^{k_2}_{j_2}(\lambda,\mu) \leq \dots)$ and we compare them with $\underline{c}_i(\lambda)$ and $-\overline{c}_i(\lambda)$. The following three different cases can be identified:

C1 : $-\overline{c}_i(\lambda) \leq c^{k_{b_i}}_{j_{b_i}}(\lambda,\mu) \leq \underline{c}_i(\lambda)$, an optimal solution is given by setting $y^i_j = 1$ for all indices $j = j_1,\dots,j_{b_i}$, while all under and over coverage variables are set to 0:

$$z_i = 0, \quad t_i = 0, \quad y^i_j = \begin{cases} 1, & j = j_1,\dots,j_{b_i}, \\ 0, & \text{otherwise.} \end{cases}$$

The optimal objective function value results

$$z^i_{LRP}(\lambda,\mu) = \sum_{n=1}^{b_i} c^{k_n}_{j_n}(\lambda,\mu) + b_i\lambda_i. \tag{49}$$

C2 : $c^{k_\ell}_{j_\ell}(\lambda,\mu) \leq \underline{c}_i(\lambda) < c^{k_{b_i}}_{j_{b_i}}(\lambda,\mu)$, where ℓ is the maximal index satisfying the inequality, an optimal solution is given by setting $y^i_j = 1$ for all indices $j = j_1,\dots,j_\ell$. Furthermore we set under coverage variable $z_i = b_i - \ell$, while over coverage variable is set to 0:

$$z_i = b_i - \ell, \quad t_i = 0, \quad y^i_j = \begin{cases} 1, & j = j_1,\dots,j_\ell, \\ 0, & \text{otherwise.} \end{cases}$$

The optimal objective function value results

$$z^i_{LRP}(\lambda,\mu) = \sum_{n=1}^{\ell} c^{k_n}_{j_n}(\lambda,\mu) + \underline{c}_i(\lambda)(b_i-\ell) + b_i\lambda_i. \tag{50}$$

C3 : $c^{k_{b_i}}_{j_{b_i}}(\lambda,\mu) < c^{k_\ell}_{j_\ell}(\lambda,\mu) \leq -\overline{c}_i(\lambda)$, where ℓ is the maximal index satisfying the inequality, an optimal solution is given by setting $y^i_j = 1$ for all indices $j = j_1,\dots,j_\ell$. Furthermore we set over coverage variable $t_i = \ell - b_i$, while under coverage variable is set to 0:

$$z_i = 0, \quad t_i = \ell - b_i, \quad y_j^i = \begin{cases} 1, & j = j_1, \dots, j_\ell, \\ 0, & \text{otherwise.} \end{cases}$$

The optimal objective function value results

$$z^i_{LRP}(\lambda, \mu) = \sum_{n=1}^{\ell} c^{k_n}_{j_n}(\lambda, \mu) + \overline{c}_i(\lambda)(\ell - b_i) + b_i \lambda_i. \tag{51}$$

We remark that we do not need to assume that each row $i \in M$ contains at least b_i elements equal to 1, i.e., $b_i \le \sum_{k \in K} |N^i_k|$. Indeed, if this inequality does not hold for a row $\tilde{i}$, the solution of the corresponding subproblem $(LRP^{\tilde{i}}(\lambda, \mu))$ falls in **C2**. Optimal solutions of subproblems $(LRP^i(\lambda, \mu))$ and $(LRP^k(\lambda, \mu))$, for $i \in M$ and $k \in K$, allow us to define an optimal solution of problem $(LRP(\lambda, \mu))$, with value

$$z_{LRP}(\lambda, \mu) = \sum_{i \in M} z^i_{LRP}(\lambda, \mu) + \sum_{k \in K} z^k_{LRP}(\lambda, \mu). \tag{52}$$

This solution will be used to solve the Lagrangian dual problem with the subgradient method. The following theorem shows how a dual feasible solution (u, v) of cost $z_D \ge z_{LRP}(\lambda, \mu)$ can be obtained from the Lagrangian costs of problem $(LRP(\lambda, \mu))$. This theorem and the following corollary (cf. Corollary 1) is the extension of similar statements from Baldacci et al. (2008) and Boschetti et al. (2008).

Theorem 1. *Let us consider two vectors $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^p$. A feasible dual solution (u, v) of the dual problem (D) is given by the following expression:*

$$\begin{aligned} u_i &= \tilde{u}_i + \lambda_i, & \forall i \in M, \\ v_k &= \tilde{v}_k + \mu_k, & \forall k \in K. \end{aligned} \tag{53}$$

where

$$\tilde{u}_i = \max\Big\{ -\overline{c}_i(\lambda), \min\{ c^{k_{b_i}}_{j_{b_i}}(\lambda, \mu), \underline{c}_i(\lambda) \} \Big\},$$

and

$$\tilde{v}_k = c^k_{j_k}(\lambda, \mu) - \max_{j \in N_k} \Big\{ \sum_{i \in R^k_j} (\tilde{u}_i - c^k_j(\lambda, \mu))^+ \Big\}.$$

Furthermore, the cost z_D of this dual solution is greater than or equal to $z_{LRP}(\lambda, \mu)$.

Proof. We need to prove that (u, v), defined as in (53), is a feasible solution of problem (D). We can easily see that u satisfies the bound constraints (29). Indeed,

$$-\overline{c}_i = -\overline{c}_i(\lambda) + \lambda_i \le \tilde{u}_i + \lambda_i = u_i$$

and

$$u_i = \tilde{u}_i + \lambda_i \le \underline{c}_i(\lambda) + \lambda_i \le \underline{c}_i.$$

We now consider constraints (28): for each $k \in K$ and $j \in N_k$,

$$\begin{aligned}
\sum_{i \in R_j^k} u_i + v_k &= \sum_{i \in R_j^k} (\tilde{u}_i + \lambda_i) + \tilde{v}_k + \mu_k \\
&= \sum_{i \in R_j^k} \tilde{u}_i + c_{j_k}^k(\lambda, \mu) - \max_{j' \in N_k} \Big\{ \sum_{i \in R_{j'}^k} (\tilde{u}_i - c_{j'}^k(\lambda, \mu))^+ \Big\} + \sum_{i \in R_j^k} \lambda_i + \mu_k \\
&\leq \sum_{i \in R_j^k} \tilde{u}_i + c_{j_k}^k(\lambda, \mu) - \sum_{i \in R_j^k} (\tilde{u}_i - c_j^k(\lambda, \mu))^+ + \sum_{i \in R_j^k} \lambda_i + \mu_k \\
&\leq \sum_{i \in R_j^k} \tilde{u}_i + c_{j_k}^k(\lambda, \mu) - \sum_{i \in R_j^k} (\tilde{u}_i - c_j^k(\lambda, \mu)) + \sum_{i \in R_j^k} \lambda_i + \mu_k \\
&= \sum_{i \in R_j^k} \tilde{u}_i + c_{j_k}^k(\lambda, \mu) - \sum_{i \in R_j^k} \tilde{u}_i + \sum_{i \in R_j^k} c_j^k(\lambda, \mu) + \sum_{i \in R_j^k} \lambda_i + \mu_k \\
&\leq c_j^k(\lambda, \mu) + |R_j^k| c_j^k(\lambda, \mu) + \sum_{i \in R_j^k} \lambda_i + \mu_k \\
&= (|R_j^k| + 1) c_j^k(\lambda, \mu) + \sum_{i \in R_j^k} \lambda_i + \mu_k = c_j^k(\lambda, \mu)
\end{aligned}$$

where the last inequality holds due to the fact that $j_k = \arg\min_{j \in N_k} c_j^k(\lambda, \mu)$, hence $c_{j_k}^k(\lambda, \mu) \leq c_j^k(\lambda, \mu)$. We now show that the cost z_D of the dual solution defined with vectors λ and μ is greater than or equal to the Lagrangian cost $z_{LRP}(\lambda, \mu)$:

$$\begin{aligned}
z_D &= \sum_{i \in M} b_i u_i + \sum_{k \in K} v_k \\
&= \sum_{i \in M} b_i(\tilde{u}_i + \lambda_i) + \sum_{k \in K} \Big(c_{j_k}^k(\lambda, \mu) - \max_{j \in N_k} \Big\{ \sum_{i \in R_j^k} (\tilde{u}_i - c_j^k(\lambda, \mu))^+ \Big\} + \mu_k \Big) \\
&\geq \sum_{i \in M} b_i(\tilde{u}_i + \lambda_i) + \sum_{k \in K} (c_{j_k}^k(\lambda, \mu) + \mu_k) - \sum_{k \in K} \sum_{j \in N_k} \sum_{i \in R_j^k} (\tilde{u}_i - c_j^k(\lambda, \mu))^+ \\
&= \sum_{i \in M} \Big(b_i \tilde{u}_i - \sum_{k \in K} \sum_{j \in N_k^i} (\tilde{u}_i - c_j^k(\lambda, \mu))^+ + b_i \lambda_i \Big) + \sum_{k \in K} (c_{j_k}^k(\lambda, \mu) + \mu_k).
\end{aligned}$$

Let us first introduce z_D^i and z_D^k defined as $z_D^i := (b_i \tilde{u}_i - \sum_{k \in K} \sum_{j \in N_k^i} (\tilde{u}_i - c_j^k(\lambda, \mu))^+ + b_i \lambda_i)$ and $z_D^k := (c_{j_k}^k(\lambda, \mu) + \mu_k)$. We show that $z_D^i = z_{LRP}^i(\lambda, \mu)$ and $z_D^k = z_{LRP}^k(\lambda, \mu)$, for all indices $i \in M$ and $k \in K$.
It is easy to see from (44) that $z_D^k = z_{LRP}^k(\lambda, \mu)$ for all $k \in K$. We now show that $z_D^i = z_{LRP}^i(\lambda, \mu)$ for all $i \in M$. We have three different cases, one for each solution defined in **C1**, **C2** or **C3**.

If the solution is the one defined in case **C1**, we have that $\tilde{u}_i = c_{j_{b_i}}^{k_{b_i}}(\lambda, \mu)$, and the following equalities hold:

$$\begin{aligned}
z_D^i &= b_i c_{j_{b_i}}^{k_{b_i}}(\lambda,\mu) - \sum_{k\in K}\sum_{j\in N_k^i}(c_{j_{b_i}}^{k_{b_i}}(\lambda,\mu) - c_j^k(\lambda,\mu))^+ + b_i\lambda_i \\
&= b_i c_{j_{b_i}}^{k_{b_i}}(\lambda,\mu) - \sum_{n=1}^{b_i}(c_{j_{b_i}}^{k_{b_i}}(\lambda,\mu) - c_{j_n}^{k_n}(\lambda,\mu)) + b_i\lambda_i \\
&= \sum_{n=1}^{b_i} c_{j_n}^{k_n}(\lambda,\mu) + b_i\lambda_i = z_{LRP}^i(\lambda,\mu)
\end{aligned}$$

where the second equality comes from the fact that $c_{j_{b_i}}^{k_{b_i}}(\lambda,\mu) - c_{j_n}^{k_n}(\lambda,\mu) \geq 0$ only for $n = 1,\ldots,b_i$, while the last equality comes from (49). If the solution is the one defined in case **C2**, we have that $\tilde{u}_i = \underline{c}_i(\lambda)$, and the following equalities hold:

$$\begin{aligned}
z_D^i &= b_i\underline{c}_i(\lambda) - \sum_{k\in K}\sum_{j\in N_k^i}(\underline{c}_i(\lambda) - c_j^k(\lambda,\mu))^+ + b_i\lambda_i \\
&= b_i\underline{c}_i(\lambda) - \sum_{n=1}^{\ell}(\underline{c}_i(\lambda) - c_{j_n}^{k_n}(\lambda,\mu)) + b_i\lambda_i \\
&= \sum_{n=1}^{\ell} c_{j_n}^{k_n}(\lambda,\mu) + \underline{c}_i(\lambda)(b_i - \ell) + b_i\lambda_i = z_{LRP}^i(\lambda,\mu)
\end{aligned}$$

where the second equality comes from the fact that $c_{j_{b_i}}^{k_{b_i}}(\lambda,\mu) - c_{j_n}^{k_n}(\lambda,\mu) \geq 0$ only for $n = 1,\ldots,\ell$, while the last equality comes from (50). If the solution is the one defined in **C3**, we have that $\tilde{u}_i = -\overline{c}_i(\lambda)$, and the following equalities hold:

$$\begin{aligned}
z_D^i &= b_i\overline{c}_i(\lambda) - \sum_{k\in K}\sum_{j\in N_k^i}(\overline{c}_i(\lambda) - c_j^k(\lambda,\mu))^+ + b_i\lambda_i \\
&= -b_i\overline{c}_i(\lambda) - \sum_{n=1}^{\ell}(-\overline{c}_i(\lambda) - c_{j_n}^{k_n}(\lambda,\mu)) + b_i\lambda_i \\
&= \sum_{n=1}^{\ell} c_{j_n}^{k_n}(\lambda,\mu) + \overline{c}_i(\lambda)(\ell - b_i) + b_i\lambda_i = z_{LRP}^i(\lambda,\mu)
\end{aligned}$$

where the second equality comes from the fact that $c_{j_{b_i}}^{k_{b_i}}(\lambda,\mu) - c_{j_n}^{k_n}(\lambda,\mu) \geq 0$ only for $n = 1,\ldots,\ell$, while the last equality comes from (51). We have shown that $z_D^i = z_{LRP}^i(\lambda,\mu)$ and $z_D^k = z_{LRP}^k(\lambda,\mu)$, for all indices $i \in M$ and $k \in K$. From equation (52) we can conclude that $z_D \geq z_{LRP}(\lambda,\mu)$. □

The following Corollary states that maximizing the function $z_{LRP}(\lambda,\mu)$ with respect to λ and μ, we achieve the optimal dual value z_D^*.

Corollary 1. *The following equality holds:*

$$\max_{\lambda,\mu} z_{LRP}(\lambda,\mu) = z_D^*. \tag{54}$$

Proof. Let us consider the Lagrangian relaxation of problem (RMP), dualizing constraints (22) and (23) by means of penalty vectors $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^p$ respectively. The resulting Lagrangian subproblem is denoted with ($LR(\lambda,\mu)$):

$$z_{LR}(\lambda,\mu) = \min \sum_{k\in K}\sum_{j\in N_k} \hat{c}_j^k(\lambda,\mu)x_j^k + \sum_{i\in M}\Big(\underline{c}_i(\lambda)z_i + \overline{c}_i(\lambda)t_i + b_i\lambda_i\Big) + \sum_{k\in K}\mu_k \tag{55}$$

$$\text{s.t.}\quad z_i, t_i \geq 0 \qquad \forall i \in M \tag{56}$$

$$0 \leq x_j \leq 1 \qquad \forall j \in N \tag{57}$$

where $\hat{c}_j^k(\lambda,\mu) := c_j^k - \sum_{i\in R_j^k}\lambda_i - \mu_k$, $\underline{c}_i(\lambda) := \underline{c}_i - \lambda_i$ and $\overline{c}_i(\lambda) := \overline{c}_i + \lambda_i$ are the Lagrangian costs. The goal is to show that the following inequality

$$z_{LR}(\lambda,\mu) \leq z_{LRP}(\lambda,\mu) \tag{58}$$

holds for each vectors λ and μ. It is easy to see that $z_{LR}(\lambda,\mu) = -\infty$ if $\underline{c}_i(\lambda)$ or $\overline{c}_i(\lambda)$ is negative for some index i. Since $z_{LRP}(\lambda,\mu)$ is always finite, inequality (58) holds. If both $\underline{c}_i(\lambda)$ and $\overline{c}_i(\lambda)$ are non-negative, let us define the index set $N_k' = \{j \in N_k : \hat{c}_j^k(\lambda,\mu) < 0\}$ for each $k \in K$, then we have

$$z_{LR}(\lambda,\mu) = \sum_{j\in N_k'} \hat{c}_j^k(\lambda,\mu) + \sum_{i\in M} b_i\lambda_i + \sum_{k\in K}\mu_k.$$

Using the solution y of the problem ($LRP(\lambda,\mu)$), we define the variable x_j^k according to expression (30). Finally, let us define $J_k = \{j \in N_k : x_j^k > 0\}$ and $\tilde{N}_k = \{j \in J_k : \hat{c}_j^k(\lambda,\mu) < 0\}$. Then we have:

$$\begin{aligned} z_{LRP}(\lambda,\mu) &= \sum_{k\in K}\sum_{j\in J_k} \hat{c}_j^k(\lambda,\mu)x_j^k + \sum_{i\in M}\Big(\underline{c}_i(\lambda)z_i + \overline{c}_i(\lambda)t_i + b_i\lambda_i\Big) + \sum_{k\in K}\mu_k \\ &\geq \sum_{k\in K}\sum_{j\in \tilde{N}_k} \hat{c}_j^k(\lambda,\mu)x_j^k + \sum_{i\in M} b_i\lambda_i + \sum_{k\in K}\mu_k \\ &\geq \sum_{k\in K}\sum_{j\in N_k'} \hat{c}_j^k(\lambda,\mu) + \sum_{i\in M} b_i\lambda_i + \sum_{k\in K}\mu_k = z_{LR}(\lambda,\mu). \end{aligned}$$

The first inequality comes from the fact that $\hat{c}_j^k(\lambda,\mu)x_j^k \geq 0$ for each $j \in J_k \setminus \tilde{N}_k$, $\underline{c}_i(\lambda) \geq 0$, and $\overline{c}_i(\lambda) \geq 0$. The second inequality comes from the fact that $x_j^k \leq 1$ and $\tilde{N}_k \subseteq N_k'$. Since the Lagrangian relaxation $LR(\lambda,\mu)$ has the integrality property, we have that the Lagrangian dual is equal to z_P. Therefore

$$\max_{\lambda,\mu} z_{LR}(\lambda,\mu) = \max_{\lambda,\mu} z_{LRP}(\lambda,\mu) = z_D^*.$$

□

From Corollary 1 it follows that we need to solve the Lagrangian dual problem $\max_{\lambda,\mu} z_{LRP}(\lambda,\mu)$ in order to find an optimal dual solution.

4.3. A column generation method based on dual ascent

In this section we describe a column generation method to solve the linear relaxation of problem (EP). This method differs from classical column generation since it solves the reduced master problem (RMP) by means of a dual ascent heuristic, instead of using the simplex algorithm (similar to Baldacci et al. (2016) and Baldacci et al. (2017)). It can be eventually combined with simplex LP solver to find the optimal dual variables of problem (RMP) and continue with the classical column generation. The proposed method is described as follows. The parameters have been set after preliminary tests.

Step 1. *Initialization.* Initialize the reduced master problem (RMP) with a set of columns N containing a feasible solution. Furthermore, initialize the penalty vectors $(\lambda,\mu) := (0,0)$, the iteration counter $i := 1$, $i^{max} := 10000$, $i_{DA}^{max} := 10$, $\beta := 2$, and $\beta^{min} := 0.0001$.

Step 2. *Dual ascent heuristic to find a dual feasible solution* (u,v) *of* (RMP). Set $\bar{z}_{LRP}(\lambda,\mu) := -\infty$, $i_{DA} := 1$. Perform the following steps:

Step 2a. *Solve* $(LRP(\lambda,\mu))$. Using the current multipliers λ and μ, solve all subproblems (41)-(43) and (45)-(48) and get an optimal solution of $(LRP(\lambda,\mu))$. If $z_{LRP}(\lambda,\mu) > \bar{z}_{LRP}(\lambda,\mu)$, then update $\bar{z}_{LRP}(\lambda,\mu) := z_{LRP}(\lambda,\mu)$, update the dual solution (u,v) using expression (53), and set $\beta := \min\{2, 1.2\cdot\beta\}$.

Step 2b. Update the multipliers (λ,μ) using the subgradients vectors $(g_i)_{i\in M}$ and $(g_k)_{k\in K}$ defined as $g_i = b_i - \sum_{k\in K}\sum_{j\in N_k^i} x_j^k - z_i + t_i$ and $g_k = 1 - \sum_{j\in N_k} x_j^k$, where x_j^k is defined using expression (30) and the solution y of $(LRP(\lambda,\mu))$. Modify the multipliers $\lambda_i := \lambda_i + \alpha g_i$ and $\mu_k := \mu_k + \alpha g_k$, where $\alpha := \beta(0.1\cdot z_{LRP}(\lambda,\mu))/(\sum_{i\in M} g_i^2 + \sum_{k\in K} g_k^2)$.

Step 2c. Set $i_{DA} := i_{DA} + 1$. If after 2 consecutive iterations, $\bar{z}_{LRP}(\lambda,\mu)$ has not improved, halve β (i.e., $\beta = 0.5\cdot\beta$). If $i_{DA} = i_{DA}^{max}$ or $\beta < \beta^{min}$, stop. Otherwise, return to Step 2a.

Step 3. *Generate new columns.* Generate, for each subproblem (26) with $k \in K$, a column with minimum reduced cost c_k^*. Define as N^* the set of columns from all subproblems such that the minimum reduced costs $c_k^* < -0.1$.

Step 4. *Stopping criteria.* If $N^* = \emptyset$ or $i = i^{max}$, stop. Otherwise, update the set of columns $N = N \cup N^*$, set $i := i + 1$ and return to Step 2.

We remark that $\bar{z}_{LRP}(\lambda, \mu)$ is a lower bound valid for problem (RMP), but it is not a valid for the complete master problem. A valid lower bound is given by the Lagrangian dual bound (see Desrosiers & Lübbecke (2005)):

$$\sum_{i \in M} b_i u_i + \sum_{k \in K} (c_k^* + v_k) \leq z^* \tag{59}$$

where we recall that c_k^* is the minimum reduced costs obtained by solving the subproblem (SP_k) during one iteration of column generation, and z^* is the optimal objective function value of the master problem.

5. Classical Lagrangian relaxation

The problem (P) can be also solved by applying the Lagrangian relaxation to the compact formulation, by dualizing exactly the coverage constraints (7), which are the linking constraints in the Dantzig-Wolfe decomposition. The resulting subproblems are identical, and the columns generated by the Lagrangian subproblems can be added to the reduced master problem (Huisman et al. (2005)). The Lagrangian relaxation has the following formulation:

$$\begin{array}{llll} (CLR(\lambda)) & z_{CLR}(\lambda) = & \min & c^\top \tilde{x} + \underline{c}^\top z + \bar{c}^\top t + \lambda^\top (b - \tilde{A}\tilde{x} - z + t) \\ & & \text{s.t.} & \tilde{x} \in X \\ & & & z, t \geq 0 \end{array}$$

where $X = \{\tilde{x} \in \{0,1\}^{\tilde{n}} : D\tilde{x} = d\}$ is the finite set defined by constraints (8) and (9) of problem (P). We remark that problem $(CLR(\lambda))$ is unbounded if the Lagrangian cost vector $\underline{c} - \lambda$ or $\bar{c} + \lambda$ has at least one negative component. Therefore, we assume that both vectors are non-negative. As consequence, in the optimal solution, vectors z and t are equal to 0. The resulting Lagrangian problem decomposes into p subproblems, one for each $k \in K$, due to the assumption that X is decomposable:

$$\begin{array}{llll} (CLR^k(\lambda)) & z^k_{CLR}(\lambda) = & \min & c^\top \tilde{x}_k - \lambda^\top \tilde{A}\tilde{x}_k \\ & & \text{s.t.} & \tilde{x}_k \in X_k. \end{array} \tag{60}$$

We can see that the subproblem (26) of the Dantzig-Wolfe decomposition and the subproblem (60) of the Lagrangian relaxation are identical, except for the constant term v_k in the objective function. An optimal solution of the Lagrangian dual problem $\bar{z}_{CLR} = \max_\lambda z_{CLR}(\lambda)$ gives the maximum lower bound.

6. Applications

In this paragraph we show two problems whose Dantzig-Wolfe decomposition leads to a particular case of (RMP). The first is the minimum sum coloring problem that consists in minimizing the sum of the cardinality of subsets of vertices receiving the same color, weighted with the index of the color, while ensuring that adjacent vertices receive different colors. The second is the multi-activity tour scheduling problem, which consists in constructing feasible planning to be assigned to company's employees. The goal is to satisfy workload requirements and minimize under and over coverage. These two applications do not incorporate a combination of set partitioning, set covering, set packing and generalized set partitioning constraints. For this reason, we generated further instances involving all four types of constraints. We will refer to these instances as the *generated instances* and they will be presented in Section 7.1.

6.1. Minimum sum coloring

In the minimum sum coloring problem, we are given an undirected graph $G = (M, E)$ with $|M| = m$ vertices and $|E|$ edges. A coloring C of G is a partition of M into p stable sets $C = \{M^1, \dots, M^p\}$, where all the vertices in M^k are colored with the same color k. The sum coloring of C is given by the sum $\sum_{k=1,\dots,p}(k \cdot |M^k|)$. The minimum sum coloring problem consists of finding a coloring C that minimizes its sum coloring. Furini et al. (2018) introduce directly an extended formulation for this problem, without using DW decomposition. The model uses binary variables x_j^k associated with each stable set $j \in N_k$ and each color $k \in K$. When variable x_j^k takes value 1, it means that all vertices in the stable set j are colored with color k. The problem appears as follows:

$$\min \quad \sum_{k \in K} \sum_{j \in N_k} c_j^k x_j^k \tag{61}$$

$$\text{s.t.} \quad \sum_{k \in K} \sum_{j \in N_k^i} x_j^k \geq 1 \qquad \forall i \in M \tag{62}$$

$$\sum_{j \in N_k} x_j^k \leq 1 \qquad \forall k \in K \tag{63}$$

$$x_j^k \in \{0, 1\} \qquad \forall k \in K, j \in N_k \tag{64}$$

where N_k contains the stable sets colored with color k, $N_k^i \subseteq N_k$ contains the stable sets covering vertex $i \in M$, and $c_j^k = k \cdot |M_j^k|$ is the cost of stable set j colored with color k. Constraints (62) impose each vertex i to be contained in at least one stable set, while constraints (63) impose each color k to be assigned to at most one stable set. The objective function (61) aims at finding a solution with the lowest sum coloring. We remark that constraints (63) can be rewritten using equalities, if we consider the empty stable set for each color k. The linear relaxation of problem (61)-(64) is obtained by replacing the integrality constraints (64) with the following relaxed constraints:

$$x_j^k \geq 0 \qquad \forall k \in K, j \in N_k. \tag{65}$$

The resulting problem is a particular case of (RMP). Indeed, it is sufficient to set all components of $\overline{c}$ equal to zero, and all components of $\underline{c}$ equal to a sufficient large positive number. The reduced master problem (61)-(65) combines stable sets in order to cover all vertices, while the subproblems defined as follows, one for each color $k \in K$, generate new stable sets:

$$\begin{aligned}
\min \quad & \sum_{i \in M} k\, \tilde{x}_i^k - \sum_{i \in M} u_i\, \tilde{x}_i^k - v_k && && (66)\\
\text{s.t.} \quad & \tilde{x}_i^k + \tilde{x}_{i'}^k \leq 1 && \forall (i,i') \in E && (67)\\
& \tilde{x}_i^k \in \{0,1\} && \forall i \in M && (68)
\end{aligned}$$

where $u \in \mathbb{R}^m$ denotes the dual variables associated with coverage constraints (62), while $v \in \mathbb{R}^p$ denotes the dual variables associated with constraints (63). By changing the sign of all coefficients and the sense of the objective function, each subproblem becomes a maximum weight stable set problem on graph G, where the weight of each vertex $i \in M$ is defined as $u_i - k$.

6.2. Multi-activity tour scheduling

Personnel scheduling problems consist of assigning employees to activities over a given time horizon, taking into account organizational, legal and social constraints. One of the first classification methods for personnel scheduling problems was proposed by Baker (1976). According to Baker, three main groups can be distinguished: shift scheduling, days-off scheduling and tour scheduling. In shift scheduling one has to schedule the employees' working periods during their working days. Days-off scheduling concerns the determination of rest days. The third case is a combination of the shift scheduling and the days-off scheduling problem. When more than one work activity has to be scheduled, the problem becomes a multi-activity tour scheduling. In this problem, we need not only to define the working days and the working periods, but also to specify the allocation of work activities. The extended formulation appears as follows:

$$\begin{aligned}
\min \quad & \sum_{i \in M} \underline{c}_i z_i + \sum_{i \in M} \overline{c}_i t_i && && (69)\\
\text{s.t.} \quad & \sum_{k \in K} \sum_{j \in N_k^i} x_j^k + z_i - t_i = b_i && \forall i \in M && (70)\\
& \sum_{j \in N_k} x_j^k = 1 && \forall k \in K && (71)\\
& x_j^k \in \{0,1\} && \forall k \in K, j \in N_k && (72)\\
& z_i, t_i \geq 0 && \forall i \in M && (73)
\end{aligned}$$

where N_k contains feasible schedules of employee $k \in K$ and $N_k^i \subseteq N_k$ is the set of schedules in which employee k works during period $i \in M$. Problem (69)-(73) assigns a feasible schedule to each employee (cf. constraints (71)), in order to cover the demand (cf. constraints (70)) while minimizing the total costs given by under and over coverage (cf. objective function (69)). The linear relaxation of problem (69)-(73) is obtained by replacing the integrality constraints (72) with the following relaxed constraints:

$$x_j^k \geq 0 \qquad \forall k \in K, j \in N_k \tag{74}$$

The resulting problem is a particular case of (RMP), where the cost vector c has all components equal to 0. The subproblems, one for each employee $k \in K$, generate new schedules:

$$\min \quad -\sum_{i \in M} u_i \, \tilde{x}_i^k - v_k \tag{75}$$

$$\text{s.t.} \quad \tilde{x}^k \in X_k \tag{76}$$

where X_k is the set of feasible schedules for employee $k \in K$, and variables $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^p$ denote the dual variables associated with constraints (70) and (71), respectively. The subproblems take into account skills and legal constraints defined by contract's regulations, such as consecutive working hours, breaks, daily working hours and amplitude of the working day.

7. Computational results

We present some computational results to show the performance of the dual ascent heuristic, by solving the linear relaxations of the problems presented above using column generation. The LP solver used for the reduced master problem is CPLEX 12.7. Then we combine it with the dual ascent heuristic previously presented, which is used during the first iterations of column generation. To be more precise, the dual ascent is used first and it stops as soon as the minimum reduced cost of all subproblems is greater than or equal to -0.1. The goal is to exploit the rapid decrease of the lower bound gap of the dual ascent, in order to speed up the convergence of the column generation. Finally, analogously to the dual ascent, we combine CPLEX with the classical Lagrangian relaxation. Experiments on applications 6.1 and 6.2 have been computed on a Intel Xeon E5-2650 v3 (2,3GHz), 64 GB of RAM (only one core is used), while an Intel Core i7-3770 CPU at 3,40GHz has been used for the experiments on the generated instances presented above.

7.1. Instances

The instances that has been used for testing the proposed approach are presented in the following paragraph. In particular, for the two applications, it was easy and possible to identify them. Moreover, we describe and report the algorithm used to generate the new complete instances.

Minimum sum coloring instances. We perform computational experiments on 43 benchmark instances, which are frequently used to evaluate the performance of minimum sum coloring algorithms (Jin et al. (2017)). These instances come from the COLOR 2002-2004 competitions[1].

Multi-activity tour scheduling instances. The multi-activity tour scheduling instances have been generated from real ones, given by the company Horizontal Software. The time horizon is fixed to one day and slots have time units of 1 hour or 30 minutes, resulting in instances with 24 and 48 slots. The instances consider 7 different activities, and 33 or 66 employees. They differ also in workload requirements, which are inspired by realistic demand coming from quick service restaurants. We consider 6 workload types named in alphabetic order, from A to F, with increasing demand. Instances are labeled with the format `S_E_W`, where `S`, `E` and `W` represent the number of slots, the number of employees and the workload type respectively.

Generated instances. We aim to further prove the validity of our approach by doing supplementary tests on more complete and diverse instances, involving set partitioning, covering, packing and generalized set partitioning constraints. They have been generated combining one instance from the covering data set ({*rail507*, *rail516*, *rail582*}), with one instance from the partitioning data set ({*sppaa01*, *sppaa02*, *sppaa03*, *sppaa05*, *sppaa06*, *sppus03*, *sppus04*}). These data sets are available at the Beasley's OR-Library[2]. More in details, we proceeded as follows:

- we consider one instance I_1 $(m_1 \times n_1)$ from the set covering data set and one I_2 $(m_2 \times n_2)$ from the set partitioning data set;
- for each instance I_1 and I_2, we add a number of convexity constraints corresponding to 22% of the total number of rows. Therefore $k_1 = 0.22 \cdot m_1$, $k_2 = 0.22 \cdot m_2$ and the total number of convexity rows is $|K| = k_1 + k_2$;
- for each instance I_1 and I_2, each column has been duplicated respectively k_1 and k_2 times, and assigned to the different convexity rows;
- half randomly chosen rows M_1^1 from the set covering instance I_1 are defined as generalized set partitioning constraints, while the other half M_1^2 is kept as covering constraints. Analogously, half randomly chosen rows M_2^1 from the set partitioning instance I_2 are defined as packing constraints while the other half M_2^2 is kept as partitioning constraints;
- all columns' costs c_j are equal either to 1 or to 2. The under and over assignment costs for the generalized set partitioning constraints are equal to 10. This means that $\underline{c}_i = \bar{c}_i = 10$ for all $i \in M_1^1$. All other under and over assignment costs are defined equal to 0 or to a large positive number M (given by the sum of all columns' costs) depending whether they correspond to a partitioning, covering or packing constraints.

[1]`http://mat.gsia.cmu.edu/COLOR02/`
[2]`http://people.brunel.ac.uk/~mastjjb/jeb/info.html`

7.2. Algorithmic details

We provide here some details on the methods used to solve the subproblems, together with the techniques employed to speed up the convergence of the column generation.

Minimum sum coloring problem. Subproblems (66)-(68) are modeled as maximum weight stable set problems and are solved using the open-source implementation[3] of the branch-and-bound algorithm described in Held et al. (2012). At each iteration of column generation, each subproblem is solved and one column, for each of them, with minimum reduced cost is added to the reduced master problem. In order to speed up the convergence of the column generation and to avoid solving subproblems that cannot generate negative reduced cost columns, we apply a technique proposed by Furini et al. (2018). As soon as no negative reduced cost stable set is found for a color k such that the corresponding constraint is not active (< 1), then no subproblem $h > k$ is solved, due to the fact that no color $h > k$ can generate stable sets with negative reduced costs. This implies that an optimal primal solution of problem (RMP) needs to be available. Therefore, this technique cannot be applied when the reduced master problem is solved with the dual ascent, since no optimal primal solution is produced. We apply a different procedure: as soon as no negative reduced cost stable set is found for color k with corresponding dual variable satisfying $v_k = 0$, we do not solve any subproblem for color $h > k$ with $v_h = 0$.

Multi-activity tour scheduling problem. Subproblems (75)-(76) are modeled as a resource-constrained shortest path problems in a directed acyclic graphs. For each slot, break and activities nodes are considered. The subproblems are solved using a labeling algorithm which starts with the trivial path containing only the source node, and extends paths one-by-one into all feasible directions. Intensification and diversification techniques have been used to speed up the convergence of the column generation. The first adds several negative reduced cost columns instead of adding only the one with the best reduced cost, while the second adds diverse and complementary columns at each iteration. These techniques are applied whenever the subproblems are solved, independently from which method is used to solve the reduced master problem.

Generated instances. Concerning the generated instances, we initialize the reduced master problem selecting only a subset of the whole set of columns. In order to assure that a feasible initial solutions exists, artificial columns with high costs have been added. The subproblems consists simply in pricing out the remaining columns and evaluate their reduced cost. At each iteration of column generation, one column with minimum reduced cost is selected for each subproblem. Then, the columns with negative reduced cost are added to the reduced master problem.

[3]`https://github.com/heldstephan/exactcolors`

7.3. Discussion of the results

In the following section we show the computational results performed on the different instances of generalized set partitioning problem with convexity constraint presented in Section 7.1. A lower bound is obtained by using the classical Lagrangian relaxation, marked with CLR and presented in Section 5. The same bound is evaluated by solving with column generation the linear relaxation of the extended formulation (EP), marked with SMOOTH and DA. In particular, with SMOOTH we mean that the optimal dual variables of (RMP) are obtained with the LP solver CPLEX, and new columns are generated using smoothed dual variables. Indeed, it is well known that the use of simplex method in column generation causes several drawbacks, such as the dual oscillations and the tailing-off effect (Vanderbeck (2005)). Several stabilization techniques have been proposed to deal with these difficulties. Among them, we can find smoothing techniques, in which dual solutions used for pricing are corrected and combined with previous duals. Every time the reduced master problem is solved using CPLEX, we apply the smoothing technique with a self adjusting parameter, presented in the recent work of Pessoa et al. (2017). With DA we mean that the dual variables of (RMP), used to generate new columns, are obtained with the dual ascent heuristic proposed. In order to prove optimality, as soon as the dual ascent heuristic does not find columns with reduced costs lower than -0.1, the optimal dual variables are evaluated with CPLEX, and the smoothing technique previously mentioned is applied.

Tables 1, 3 and 4 compare the three methods DA, SMOOTH and CLR on the minimum sum coloring, the multi-activity tour scheduling and the generated instances respectively. Concerning the results on the minimum sum coloring in Table 1, we report the name of the instance (*name*), the number of vertices (n) the number of edges (m), the graph density ($d = 2m/n(n-1)$), and the lower bound (LB) found by all the three methods. In addition, we report the number of iterations (*iter*) of column generation, the number of final columns in the reduced master problem (*cols*) and the total computational time (*time*) in seconds, for DA, SMOOTH and CLR. Instances unsolved within one hour time limit are marked with "tl" and "-". For each instance, bold values indicate the fastest method. In addition, the last two rows of the table shows the average values and the shifted geometric mean (SGM), which has the advantage to neither be compromised by very large nor by very small outliers. We recall that the shifted geometric mean of $t_1, \ldots, t_n$ is defined by $(\prod_{i=1}^{n}(t_i + s))^{1/n} - s$. We use a shift factors of $s = 10$ for the iterations and the time, and $s = 100$ for the number of columns. The results on the minimum sum coloring reported in Table 1, show that CLR fails in solving 15 instances, while SMOOTH fails in two of them. Furthermore, bold time values indicate that 30 instances are solved using DA in lower computational time compared to SMOOTH and CLR. Finally, the average and shifted geometric mean values on the last two rows, underline that DA needs fewer iterations and computational time to converge. It also generates almost one third of the columns generated by SMOOTH.

Table 2 reports the instances of sum coloring for which we compute a lower bound that improves the best lower bound known in the literature. The table

Instances					DA			SMOOTH			CLR		
name	n	m	d	LB	iter	cols	time(s)	iter	cols	time(s)	iter	cols	time(s)
2-Insertions_3	37	72	0.11	62	87	273	**0.1**	102	828	0.2	308	936	0.3
3-Insertions_3	56	110	0.07	92	151	487	**0.8**	205	2022	1.1	323	996	1.3
anna	138	493	0.05	276	929	3953	70.7	989	10584	103.4	623	5040	**35.5**
david	87	406	0.11	237	169	1200	**2.9**	194	2844	3.6	407	3655	7.3
DSJC125.1	125	736	0.09	314	201	1100	3211.6	320	5601	3390.7	346	1773	**2976.4**
DSJC125.5	125	3891	0.50	978	140	2391	55.0	166	7143	**42.6**	471	7258	151.7
DSJC125.9	125	6961	0.90	2500	143	5627	**8.5**	136	9242	9.0	631	22260	35.6
DSJC250.5	250	15668	0.50	3105	225	6690	**2827.9**	362	29301	2974.0	-	-	tl
DSJC250.9	250	27897	0.90	8235	301	18114	**217.5**	304	38305	265.2	-	-	tl
DSJR500.1c	500	121275	0.97	16234	273	31349	**466.5**	399	82380	877.7	-	-	tl
games120	120	638	0.09	443	104	1087	1434.7	142	4529	**972.3**	-	-	tl
huck	74	301	0.11	243	78	663	**0.7**	111	1836	0.9	504	4281	3.8
jean	80	254	0.08	217	120	868	**0.9**	192	2158	2.0	428	3411	4.1
miles1000	128	3216	0.40	1666	81	2808	**7.8**	103	5574	10.0	562	18002	81.0
miles1500	128	5198	0.64	3354	88	4764	**4.8**	60	5916	5.9	797	38954	63.9
miles250	128	387	0.05	325	258	1950	352.4	336	5700	**242.3**	353	2566	554.7
miles500	128	1170	0.14	705	103	1745	41.3	143	4635	**34.9**	455	7590	213.4
miles750	128	2113	0.26	1173	82	2381	**6.9**	107	4922	9.1	502	12165	73.8
mug100_1	100	166	0.03	202	167	859	**1735.4**	244	4036	2353.2	-	-	tl
mug100_25	100	166	0.03	202	163	761	2971.9	254	4015	**1716.5**	-	-	tl
mug88_1	88	146	0.04	178	130	628	**308.9**	206	3179	389.6	335	1055	1494.2
mug88_25	88	146	0.04	178	156	964	**151.1**	197	3257	199.7	324	1055	632.1
mulsol.i.1	197	3925	0.20	1957	281	7433	**46.9**	2140	23950	669.9	-	-	tl
mulsol.i.2	188	3885	0.22	1191	2708	9417	910.0	1832	24879	**908.1**	-	-	tl
mulsol.i.3	184	3916	0.23	1187	946	5806	**185.4**	1667	23425	725.5	-	-	tl
mulsol.i.4	185	3946	0.23	1189	342	4436	**45.3**	1505	23186	625.1	-	-	tl
mulsol.i.5	186	3973	0.23	1160	1839	9367	**514.1**	1622	23460	709.6	-	-	tl
myciel3	11	20	0.36	21	23	61	**0.0**	19	74	**0.0**	257	774	**0.0**
myciel4	23	71	0.28	44	36	127	**0.0**	38	278	**0.0**	308	1131	0.1
myciel5	47	236	0.22	88	88	382	**0.2**	99	1197	0.4	280	1103	0.6
myciel6	95	755	0.17	176	173	901	**1.7**	265	3932	3.8	345	1442	2.7
myciel7	191	2360	0.13	349	580	3332	47.2	858	13482	115.9	422	2139	**23.2**
queen10_10	100	1470	0.30	550	146	1807	130.2	178	5175	**105.5**	540	5284	515.5
queen11_11	121	1980	0.27	726	157	2155	721.1	199	7366	**641.3**	566	5877	3213.4
queen5_5	25	160	0.53	75	50	283	**0.1**	41	364	**0.1**	297	1594	**0.1**
queen6_6	36	290	0.46	138	47	376	**0.1**	46	643	0.2	321	2347	0.6
queen7_7	49	476	0.40	196	62	564	**0.5**	84	1430	0.8	464	3132	2.5
queen8_12	96	1368	0.30	624	90	1333	52.2	107	4140	**45.3**	421	4406	427.4
queen8_8	64	728	0.36	291	75	797	2.2	84	2090	**1.7**	388	3321	12.7
queen9_9	81	1056	0.33	405	115	1271	**13.5**	146	3443	15.7	-	-	tl
zeroin.i.1	211	4100	0.19	1822	272	11617	**47.2**	3535	30763	2091.3	-	-	tl
zeroin.i.2	211	3541	0.16	1004	387	7295	**46.7**	-	-	tl	-	-	tl
zeroin.i.3	206	3540	0.17	998	267	6590	**36.4**	-	-	tl	-	-	tl
Average					**298**	**3861**	**388**	**481**	**10519**	**639**	**428**	**5841**	**1501**
SGM					**170**	**1856**	**61**	**230**	**4851**	**103**	**412**	**3295**	**249**

Table 1: Results for minimum sum coloring problem with time limit 3600 seconds.

Instances	LB_ℓ	UB_ℓ	LB_{cg}	gap_ℓ (%)	gap_{cg}(%)
DSJC125.1	247	326	314	31.98	3.82
DSJC125.5	549	1012	978	84.34	3.48
DSJC250.5	1287	3210	3105	149.42	3.38
DSJR500.1c	15398	16286	16234	5.77	0.32
myciel7	286	381	349	33.22	9.17

Table 2: Lower bounds improved compared to the best bounds known in the literature.

Instances	DA			SMOOTH			CLR		
	iter	cols	time(s)	iter	cols	time(s)	iter	cols	time(s)
24_33_A	33	1084	**2.9**	40	1678	3.3	198	6826	14.2
24_33_B	36	980	**3.2**	37	1607	3.2	156	5039	11.9
24_33_C	35	1016	**3.2**	39	1709	3.3	167	5636	12.4
24_66_A	29	1772	**2.8**	43	3554	3.9	208	14312	15.0
24_66_B	43	1036	**4.3**	51	4138	4.8	193	13444	13.8
24_66_C	48	1174	4.9	41	3550	**4.1**	202	14044	14.9
24_66_D	25	2174	**2.5**	46	3824	4.3	238	16364	18.9
24_66_E	41	2032	4.1	41	3550	**4.0**	170	11272	14.0
24_66_F	31	2098	3.2	31	2890	**3.0**	80	4672	7.2
48_33_A	56	1557	**19.6**	86	3306	31.2	232	8124	75.8
48_33_B	71	1615	**25.3**	76	3034	27.4	230	5971	80.5
48_33_C	58	1628	**22.0**	78	3080	30.0	183	5753	61.9
48_66_A	48	3378	**19.0**	86	6612	32.1	198	14004	64.1
48_66_B	49	2900	**19.2**	83	6530	32.2	212	14120	70.2
48_66_C	53	3256	**20.7**	81	6358	30.9	232	16324	74.6
48_66_D	51	3460	**20.3**	89	6826	34.5	233	15868	76.6
48_66_E	75	3312	**33.1**	85	6678	35.0	208	12222	76.1
48_66_F	55	3256	**22.2**	71	5698	27.3	198	12100	70.9
Average	**46**	**2096**	**13**	**61**	**4145**	**18**	**196**	**10894**	**43**
SGM	**45**	**1911**	**11**	**58**	**3755**	**14**	**192**	**9985**	**34**

Table 3: Results for multi-activity tour scheduling problem.

shows the name of the instance (*Instance*), the best lower (LB_ℓ) and upper (UB_ℓ) bounds known in the literature, and the lower bound found solving the linear relaxation of problem (61)-(64). The last two columns show the gap (gap_ℓ *(%)*) between LB_ℓ and UB_ℓ, and the gap (gap_{cg} *(%)*) between $\mathrm{LB}_{\mathrm{cg}}$ and UB_ℓ. To the best of our knowledge, the best lower and upper bounds for the minimum sum coloring instances are reported in Jin et al. (2017). Recently, Furini et al. (2018) were able to solve to optimality many of these instances.

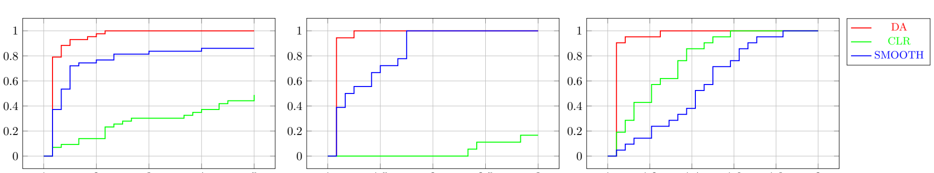
The results on the multi-activity tour scheduling problem and on the generated instance in Table 3 and 4, confirm that DA improves the computational time, since for most of the instances the solving time is lower compared to SMOOTH and CLR. Figure 1 presents the performance profiles introduced by Dolan & Moré (2002). In particular, the performance profiles of the minimum sum coloring, the multi-activity tour scheduling and the generated instances are shown respectively in Figures 1a, 1b and 1c. As we can see, DA results the method with the best performance in all the applications.

These experiments show us that the dual ascent heuristic presented in Section 4, embedded in a column generation framework improves the computational time in many cases. This is supported by the fact lower and upper bound gaps decrease faster when the dual ascent heuristic is employed in the first iterations, as it can be seen in Figures 2, 3 and 4. Figure 2 presents three graphics on the minimum sum coloring problem. In particular, Figure 2a shows the behavior of the average gap between the lower bound (59) and the optimal value z^* of the master problem in the bigger graphic, while the behavoir of one instance (*queen10_10*) is reported in the smaller graphic. Figure 2b shows the behavior of the gap between the upper bound, given by the optimal value of the reduced

Instances	DA			SMOOTH			CLR		
	iter	cols	time(s)	iter	cols	time(s)	iter	cols	time(s)
rail507_sppaa01	1099	268506	1190.1	1276	295902	1652.5	1126	296636	**1177.8**
rail507_sppaa02	780	144495	650.4	733	149499	683.4	654	145064	**522.6**
rail507_sppaa03	962	246043	**959.5**	1321	307713	1528.2	1110	313128	1072.5
rail507_sppaa05	1060	254513	1022.2	1192	275412	1507.4	1152	277445	**1080.0**
rail507_sppaa06	934	195915	**756.3**	1017	210849	1171.9	995	227171	850.4
rail507_sppus03	609	67991	431.3	565	63886	541.4	589	66385	**400.1**
rail507_sppus04	649	75780	529.5	598	74178	**510.9**	653	79657	524.9
rail516_sppaa01	1037	249011	**855.2**	1357	311959	1241.8	1294	307295	1112.9
rail516_sppaa02	512	105526	**332.3**	647	140314	460.8	856	156552	493.1
rail516_sppaa03	966	236608	**770.7**	1306	298856	1392.5	1252	295032	954.8
rail516_sppaa05	966	231800	**766.2**	1283	300056	1294.2	1176	283701	913.3
rail516_sppaa06	786	173239	**547.1**	1033	218254	896.5	973	233490	711.4
rail516_sppus03	536	60639	**365.0**	565	65108	399.5	581	66523	391.4
rail516_sppus04	515	61854	**324.5**	509	66783	340.6	597	74838	357.4
rail582_sppaa01	854	245416	**958.0**	1103	306578	1436.3	1347	323769	1372.6
rail582_sppaa02	681	154775	**618.5**	828	199207	734.6	910	203656	822.8
rail582_sppaa03	949	253826	**1004.0**	1299	341777	1616.7	1286	351219	1368.5
rail582_sppaa05	879	238048	**921.9**	1143	295174	1358.1	1222	323971	1266.4
rail582_sppaa06	779	192349	**760.4**	1018	253960	1055.4	1313	273858	1197.3
rail582_sppus03	574	73645	**532.2**	675	87840	685.2	701	90531	642.9
rail582_sppus04	642	85719	**591.8**	761	107934	706.9	764	107239	691.9
Average	**798**	**172176**	**709**	**963**	**208154**	**1010**	**978**	**214150**	**854**
SGM	**776**	**151808**	**665**	**916**	**179140**	**907**	**940**	**185262**	**787**

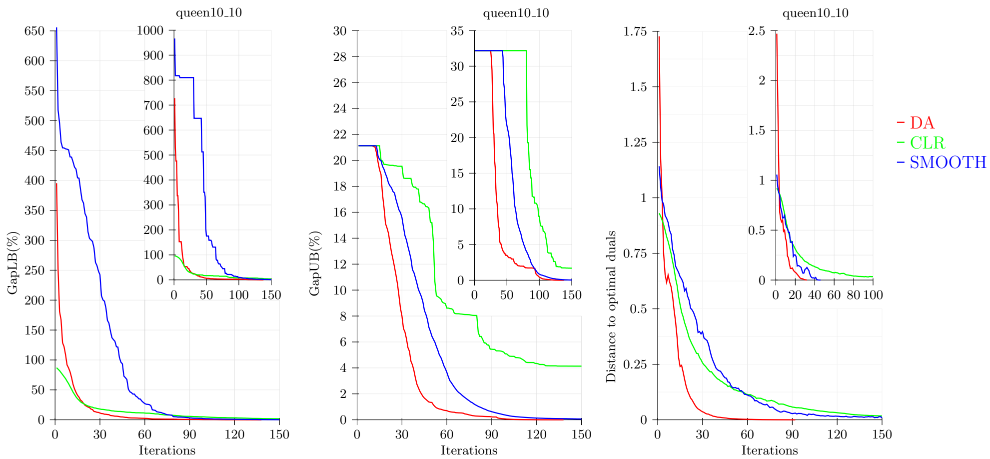
Table 4: Results for the generated instances.

master problem, and the optimal value z^*. Finally, Figure 2c shows the distance between the dual variables at an intermediate iteration t, (u^t, v^t), and the final dual solution (u^*, v^*). Analogously, Figures 3 and 4 presents the three graphics for the multi-activity tour scheduling problem and the generated instances respectively. We can see that the dual ascent heuristic allows to decrease significantly the lower and upper bounds gaps during the first iterations of column generation, yielding to a faster convergence compare with CLR and SMOOTH. Algorithm CLR is competitive with DA in decreasing the lower bound gap, while the upper bound gap improves slowly. This behavior can be explained by the quality of the generated columns. Indeed, the dual variables estimated by DA allow the generation of good quality columns and a faster decrease of the upper bound gap. Furthermore, the dual variables generated are more stable compared to SMOOTH.



(a) Minimum sum coloring (b) Tour scheduling (c) Generated instances

Figure 1: Performance profiles.



(a) Lower bound gap (b) Upper bound gap (c) Distance to optimal duals

Figure 2: Graphics for the minimum sum coloring problem.

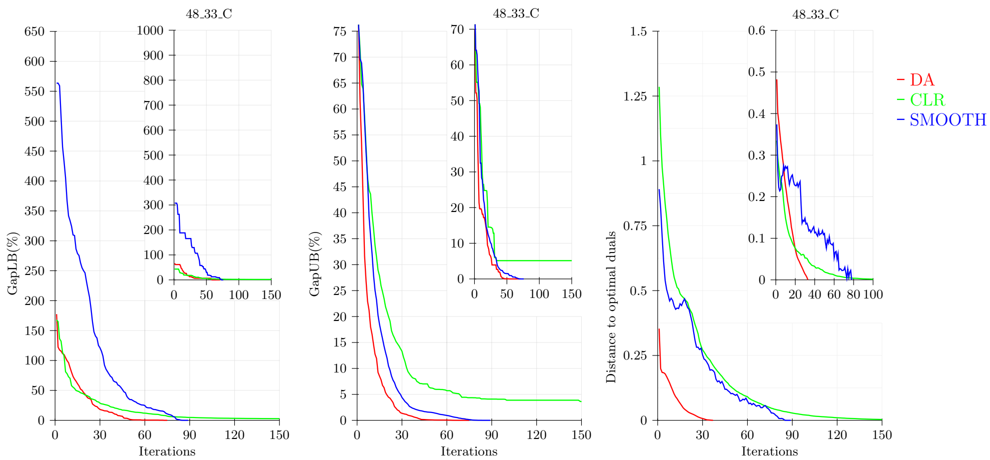
8. Conclusions

This paper describes a dual ascent heuristic, based on a reformulation and Lagrangian relaxation, to obtain efficient dual feasible solutions of the linear relaxation of the generalized set partitioning problem with convexity constraints. The proposed method is able to deal with set partitioning, covering, packing constraints and more general versions with right hand side different from one, together with under and over coverage variables. The computational experiments have been carried out on three different problem sets. The results indicate that the dual ascent is efficient and it can be integrated with methods based on simplex LP solver to speed up the convergence.

Future works will investigate the further extension of the dual ascent to the more general case, where the coverage constraints' matrix in (2) is assumed to have integer coefficients, and it is not limited the be binary. This would allow to implement a branch-and-cut-and-price framework to obtain an optimal integer solution of the generalized set partitioning problem with convexity constraints. The presented dual ascent heuristic could be employed at each node of the search tree to speed up the convergence of the column generation. However, only experiments can certify its effectiveness in that context. Indeed, non-root nodes already have a good hot start, given by the parent dual solution.

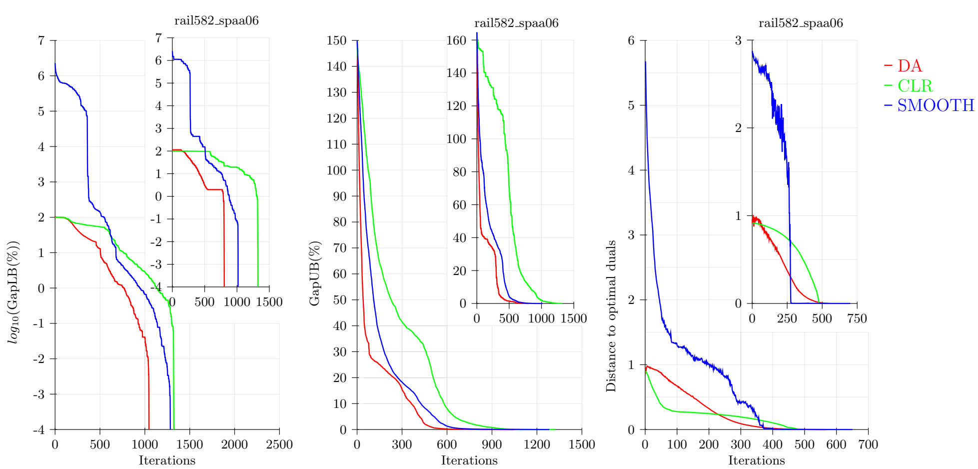
Acknowledgement

The authors would like to thank the anonymous reviewers and associate editor for their helpful suggestions. This study was funded by BPIfrance in the frame of the French cooperative project ADAMme, Projet Investissement Avenir (FSN: AAP CISN2). The authors also thank Fabio Furini, Enrico Malaguti,



(a) Lower bound gap (b) Upper bound gap (c) Distance to optimal duals

Figure 3: Graphics for multi-activity tour scheduling problem.



(a) Lower bound gap (b) Upper bound gap (c) Distance to optimal duals

Figure 4: Graphics for the generated instances.

Sébastien Martin and Ian-Christopher Ternier for providing part of the code to solve the minimum sum coloring problem.

References

Alfares, H. K. (2004). Survey, Categorization, and Comparison of Recent Tour Scheduling Literature. *Annals of Operations Research*, *127*, 145–175.

Baker, K. R. (1976). Workforce Allocation in Cyclical Scheduling Problems: A Survey. *Operational Research Quarterly (1970-1977)*, *27*, 155–167.

Balas, E., & Carrera, M. C. (1996). A Dynamic Subgradient-Based Branch-and-Bound Procedure for Set Covering. *Operations Research*, *44*, 875–890.

Balas, E., & Padberg, M. (1976). Set Partitioning: A survey. *SIAM Review*, *18*, 710–760.

Baldacci, R., Bartolini, E., & Mingozzi, A. (2011a). An Exact Algorithm for the Pickup and Delivery Problem with Time Windows. *Operations Research*, *59*, 414–426.

Baldacci, R., Bartolini, E., Mingozzi, A., & Roberti, R. (2010). An exact solution framework for a broad class of vehicle routing problems. *Computational Management Science*, *7*, 229–268.

Baldacci, R., Bartolini, E., Mingozzi, A., & Valletta, A. (2011b). An Exact Algorithm for the Period Routing Problem. *Operations Research*, *59*, 228–241.

Baldacci, R., Christofides, N., & Mingozzi, A. (2008). An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, *115*, 351–385.

Baldacci, R., Hill, A., Hoshino, E. A., & Lim, A. (2017). Pricing strategies for capacitated ring-star problems based on dynamic programming algorithms. *European Journal of Operational Research*, *262*, 879–893.

Baldacci, R., Ngueveu, S. U., & Wolfler Calvo, R. (2016). The Vehicle Routing Problem with Transhipment Facilities. *Transportation Science*, *51*, 592–606.

Balinski, M. L., & Quandt, R. E. (1964). On an Integer Program for a Delivery Problem. *Operations Research*, *12*, 300–304.

Barahona, F., & Anbil, R. (2002). On some difficult linear programs coming from set partitioning. *Discrete Applied Mathematics*, *118*, 3–11.

Boschetti, M., & Maniezzo, V. (2015). A set covering based matheuristic for a real-world city logistics problem. *International Transactions in Operational Research*, *22*, 169–195.

Boschetti, M. A., Mingozzi, A., & Ricciardelli, S. (2008). A dual ascent procedure for the set partitioning problem. *Discrete Optimization*, *5*, 735–747.

Brønmo, G., Nygreen, B., & Lysgaard, J. (2010). Column generation approaches to ship scheduling with flexible cargo sizes. *European Journal of Operational Research*, *200*, 139–150.

Cacchiani, V., Hemmelmayr, V. C., & Tricoire, F. (2014). A set-covering based heuristic algorithm for the periodic vehicle routing problem. *Discrete Applied Mathematics*, *163*, 53–64.

Ceria, S., Nobili, P., & Sassano, A. (1998a). A lagrangian-based heuristic for large-scale set covering problems. *Mathematical Programming*, *81*, 215–228.

Ceria, S., Nobili, P., & Sassano, A. (1998b). A Lagrangian-based heuristic for large-scale set covering problems. *Mathematical Programming*, *81*, 215–228.

Darby-Dowman, K., & Mitra, G. (1985). An extension of set partitioning with application to scheduling problems. *European Journal of Operational Research*, *21*, 200–205.

Desaulniers, G., Desrosiers, J., Dumas, Y., Marc, S., Rioux, B., Solomon, M. M., & Soumis, F. (1997). Crew pairing at Air France. *European Journal of Operational Research*, *97*, 245–259.

Desrosiers, J., & Lübbecke, M. E. (2005). A Primer in Column Generation. In G. Desaulniers, J. Desrosiers, & M. M. Solomon (Eds.), *Column Generation* (pp. 1–32). Springer US.

Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical programming*, *91*, 201–213.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., & Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, *153*, 3–27.

Fisher, M. L., & Kedia, P. (1990). Optimal Solution of Set Covering/Partitioning Problems Using Dual Heuristics. *Management Science*, *36*, 674–688.

Furini, F., Malaguti, E., Martin, S., & Ternier, I.-C. (2018). ILP Models and Column Generation for the Minimum Sum Coloring Problem. *Electronic Notes in Discrete Mathematics*, *64*, 215–224.

Gérard, M., Clautiaux, F., & Sadykov, R. (2016). Column generation based approaches for a tour scheduling problem with a multi-skill heterogeneous workforce. *European Journal of Operational Research*, *252*, 1019–1030.

Held, S., Cook, W., & Sewell, E. C. (2012). Maximum-weight stable sets and safe lower bounds for graph coloring. *Mathematical Programming Computation*, *4*, 363–381.

Huisman, D., Jans, R., Peeters, M., & Wagelmans, A. P. M. (2005). Combining Column Generation and Lagrangian Relaxation. In G. Desaulniers, J. Desrosiers, & M. M. Solomon (Eds.), *Column Generation* (pp. 247–270). Springer US.

Jin, Y., Hamiez, J.-P., & Hao, J.-K. (2017). Algorithms for the minimum sum coloring problem: a review. *Artificial Intelligence Review*, *47*, 367–394.

Lusby, R. M., Larsen, J., Ehrgott, M., & Ryan, D. (2011). Railway track allocation: models and methods. *OR Spectrum*, *33*, 843–883.

Malaguti, E., Monaci, M., & Toth, P. (2011). An exact approach for the Vertex Coloring Problem. *Discrete Optimization*, *8*, 174–190.

Mingozzi, A., Maniezzo, V., Ricciardelli, S., & Bianco, L. (1998). An Exact Algorithm for the Resource-Constrained Project Scheduling Problem Based on a New Mathematical Formulation. *Management Science*, *44*, 714–729.

Muter, İ., Birbil, Ş. İ., & Şahin, G. (2010). Combination of Metaheuristic and Exact Algorithms for Solving Set Covering-Type Optimization Problems. *INFORMS Journal on Computing*, *22*, 603–619.

Pessoa, A., Sadykov, R., Uchoa, E., & Vanderbeck, F. (2017). Automation and combination of linear-programming based stabilization techniques in column generation.

Rasmussen, M. S., & Larsen, J. (2011). *Optimisation-Based Solution Methods for Set Partitioning Models*. Ph.D. thesis Technical University of Denmark, Department of Informatics and Mathematical Modeling.

Rezanova, N. J., & Ryan, D. M. (2010). The train driver recovery problem—A set partitioning based model and solution method. *Computers & Operations Research*, *37*, 845–856.

Rönnqvist, M. (1995). A method for the cutting stock problem with different qualities. *European Journal of Operational Research*, *83*, 57–68.

Rossi, F., & Smriglio, S. (2001). A set packing model for the ground holding problem in congested networks. *European Journal of Operational Research*, *131*, 400–416.

Vanderbeck, F. (2005). Implementing Mixed Integer Column Generation. In G. Desaulniers, J. Desrosiers, & M. M. Solomon (Eds.), *Column Generation* (pp. 331–358). Springer US.

Vemuganti, R. R. (1998). Applications of Set Covering, Set Packing and Set Partitioning Models: A Survey. In D.-Z. Du, & P. M. Pardalos (Eds.), *Handbook of Combinatorial Optimization* (pp. 573–746). Springer US.