Les Cahiers du GERAD ISSN: 0711–2440

Binary Clustering Problems: Symmetric, Asymmetric and Decomposition Formulations

R. Jans
J. Desrosiers

G–2010–44

August 2010

Binary Clustering Problems: Symmetric, Asymmetric and Decomposition Formulations

Raf Jans
Jacques Desrosiers

GERAD & HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada, H3T 2A7

raf.jans@hec.ca
jacques.desrosiers@hec.ca

August 2010

Les Cahiers du GERAD
G–2010–44

Abstract

In this paper, we generalize the Asymmetric Representatives Formulation, which was first introduced by Campêlo et al. (2008) for the Node Coloring Problem. The main idea from this formulation can be used to model a variety of Binary Clustering Problems. The new asymmetric decision variables indicate if an object belongs to a specific cluster, but the cluster is identified by the lowest indexed object. The advantage of this formulation is that it eliminates the symmetry between the clusters which exists in the classical formulation. We prove that the LP relaxation bound of the Asymmetric Representatives Formulation is at least as good as the one from the classical Symmetric Formulation. We further show that applying Dantzig-Wolfe decomposition to the classical Symmetric Formulation or to the Asymmetric Representatives Formulation leads to the same symmetry-breaking model, and hence the same LP bound that can be much stronger. Finally, we show that in specific cases, the LP relaxation bound of the Asymmetric Representatives Formulation depends on the order of the input data.

Key Words: Integer Programming; Binary Clustering Problems; Dantzig-Wolfe Decomposition; Symmetry-Breaking Formulations.

Résumé

Dans cet article, nous généralisons aux problèmes de regroupement binaire la formulation des représentants asymétriques ARF qui a d'abord été introduite par Campêlo et al. (2008) pour le problème du coloriage des nœuds sur un graphe. Les nouvelles variables asymétriques binaires indiquent si un objet appartient ou non à un groupe donné, et ce groupe est identifié par l'objet de plus petit indice. L'avantage de cette formulation est qu'elle élimine la symétrie entre les groupes que l'on rencontre dans la formulation classique SF. Nous démontrons que la borne obtenue par la relaxation linéaire d'ARF est au moins aussi bonne que celle de la relaxation linéaire de SF. Nous montrons également que l'application de la décomposition Dantzig-Wolfe à SF ou ARF donne en fait le même modèle non-symétrique, d'où la même borne qui, en pratique, est bien meilleure. Finalement, nous montrons par un exemple que la borne linéaire d'ARF dépend de l'ordonnancement initial des groupes.

Mots clés : Programmation en nombres entiers; décomposition Dantzig-Wolfe; problèmes de regroupement binaire; formulations asymétriques.

1 Introduction

Many Integer Programming (IP) formulations suffer from symmetry in their solution space. Due to the possible permutation of the variable values, alternative solutions exist at the same cost and need to be checked during the exploration of a branch-and-bound tree. Consequently, the search is unnecessarily duplicated and computation times can increase drastically.

A well known example is the Node Coloring Problem (also known as the Graph or Vertex Coloring Problem). Given is a set of $K = \{1, \ldots, m\}$ colors and a graph $G = (N, E)$, where $N = \{1, \ldots, n\}$ is the set of nodes and E the set of arcs. The aim is to assign a color to each node, using the minimum number of colors, so that both nodes of each arc have a different color. In the classical *Symmetric Formulation* (SF), see e.g. Méndez-Díaz and Zabala (2006) or Kaibel and Pfetsch (2008), we have the following two types of binary variables: $x_i^k = 1$ if node $i \in N$ has color $k \in K$, 0 otherwise, and $x_0^k = 1$ if color $k \in K$ is used, 0 otherwise:

$$\text{Min} \sum_{k \in K} x_0^k \tag{1.1}$$

$$\text{s.t.} \sum_{k \in K} x_i^k = 1 \qquad \forall i \in N \tag{1.2}$$

$$x_i^k + x_j^k \leq x_0^k \qquad \forall (i,j) \in E : i < j, \forall k \in K \tag{1.3}$$

$$x_i^k \in \{0,\, 1\} \qquad \forall i \in N \cup \{0\}, \forall k \in K \tag{1.4}$$

In the objective function (1.1), we minimize the number of colors used. Each node has exactly one assigned color (1.2). The endpoints of an arc cannot have the same color (1.3). The symmetry in this formulation stems from the fact that the colors can be arbitrarily permuted. Symmetry, however, is a property of a formulation and not of the problem itself. Other formulations for this problem do not exhibit this symmetry between colors. Mehrotra and Trick (1996) propose the Independent Set Formulation, which can be obtained by applying a Dantzig-Wolfe reformulation (Dantzig-Wolfe 1960) to (1.1)–(1.4). In their formulation, a binary variable is associated with each maximal independent set of nodes. The objective function minimizes the number of sets used subject to the constraints that each node is contained in at least one set or cluster. This formulation avoids the symmetry induced by numbering the clusters in the symmetric formulation.

Proposed by Campêlo et al. (2008) and also briefly discussed in Margot (2010), the *Asymmetric Representatives Formulation* (ARF) is another formulation that does not exhibit symmetry. The general idea behind this ARF is to identify a cluster by the node with the lowest index in that cluster using the following binary variables: $v_i^h = 1$ if node $i \in N$ is in cluster $h \in \{1, \ldots, i\}$ and node h is the lowest indexed node in that cluster. The variable $v_h^h,\ h \in K$, indicates whether the cluster with node identifier h is used or not.

An ARF for the Node Coloring is as follows:

$$\text{Min} \sum_{h \in N} v_h^h \tag{2.1}$$

$$\text{s.t.} \sum_{h \in \{1,\ldots,i\}} v_i^h = 1 \qquad \forall i \in N \tag{2.2}$$

$$v_i^h \leq v_h^h \qquad \forall i \in N, \forall h \in \{1, \ldots, i-1\} \tag{2.3}$$

$$v_i^h + v_j^h \leq v_h^h \qquad \forall (i,j) \in E : i < j, \forall h \in \{1, \ldots, i\} \tag{2.4}$$

$$v_i^h \in \{0,\, 1\} \qquad \forall i \in N, \forall h \in \{1, \ldots, i\} \tag{2.5}$$

The number of clusters used is minimized in the objective function (2.1). Constraint set (2.2) imposes that each node must be part of exactly one cluster, and the cluster identifier has to be lower than or equal to the node number. Constraints (2.3) impose that node i can only be part of the cluster with identifier h if that cluster is used. The end nodes of each edge must be in different clusters (2.4). Campêlo et al. (2008) show that applying a Dantzig-Wolfe decomposition on the ARF results in a reformulation that is equivalent to the Independent Set Formulation proposed by Mehrotra and Trick (1996).

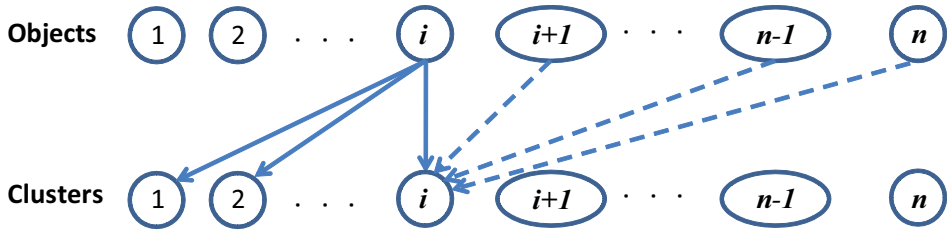

Figure 1: Feasible assignments in the Asymmetric Representatives Formulation

In this note, we show that the results from Campêlo et al. (2008) for the Node Coloring can be generalized in two ways. First, although the ARF is a little known, it is very flexible and can be used to model various Binary Clustering Problems. Second, for Binary Clustering Problems, we show in general that applying a Dantzig-Wolfe decomposition on the SF and the ARF lead to the same symmetry-breaking model, hence the same Linear Programming (LP) relaxation lower bound. As a third contribution, we show that the LP relaxation of the ARF is always at least as good as the LP relaxation of the SF. A fourth contribution of this note is to show that for some Binary Clustering Problems, the LP relaxation of the ARF depends on the order of the input data. There are however problems for which this is not the case, like the Bin Packing Problem.

The paper is organized as follows. In Section 2, we examine various Binary Clustering Problems for which we give the ARF. For one specific example (a Scene Selection), we provide computational results showing that the LP value of the ARF is better than the LP value of the SF. This is formally and generally proven in Section 3 where we also prove that the Dantzig-Wolfe reformulations of both the SF and the ARF always result in the same model, usually a much stronger one. Our conclusions follow.

2 The ARF for Binary Clustering Problems

Assume a set $N = \{1, \ldots, n\}$ of objects and a set $K = \{1, \ldots, m\}$ of possible cluster identifiers. In the SF, we define $(n+1)m$ binary variables, that is $x_i^k = 1$ if object $i \in N$ is in cluster $k \in K$, 0 otherwise, and $x_0^k = 1$ if cluster $k \in K$ is used. Symmetric solutions can be obtained by arbitrarily permuting the cluster identifiers. In the ARF, let $v_i^h = 1$ if object $i \in N$ is in the same cluster as object $h \in N$ and object h is the lowest numbered object in that cluster, 0 otherwise. In the ARF, $n(n+1)/2$ binary variables need to be defined, that is, $v_i^h,\ i \in N,\ h \in \{1, \ldots, i\}$, or equivalently, $v_i^h,\ h \in N,\ i \in \{h, \ldots, n\}$ (see Figure 1). Note that $v_h^h = 1$ indicates that cluster h is selected. With these new variables, the symmetry between clusters disappears.

The core sets of parameters and constraints in a Binary Clustering Problem are the number of objects and the number of possible cluster identifiers, the assignment of the objects to clusters exactly once, and the assignment of objects only in used clusters. In the SF, this becomes:

$$\sum_{k \in K} x_i^k = 1 \qquad \forall i \in N = \{1, \ldots, n\} \tag{3.1}$$

$$x_i^k \leq x_0^k \qquad \forall k \in K, \forall i \in N \tag{3.2}$$

$$x_i^k \in \{0, 1\} \qquad \forall k \in K = \{1, \ldots, m\}, \forall i \in N \cup \{0\} \tag{3.3}$$

In the ARF, these parameters and constraints are written as follows:

$$\sum_{h \in \{1, \ldots, i\}} v_i^h = 1 \qquad \forall i \in N \tag{3.4}$$

$$\sum_{h \in N} v_h^h \leq m \tag{3.5}$$

$$v_i^h \leq v_h^h \qquad \forall h \in N, \forall i \in \{h, \ldots, n\} \tag{3.6}$$

$$v_i^h \in \{0,\ 1\} \qquad \forall h \in N, \forall i \in \{h, \ldots, n\} \tag{3.7}$$

Note that the sets of constraints (3.2) and (3.6) are necessary, otherwise an object could be assigned to a cluster that is not used. However, in several applications, these constraints may be redundant as they are implicitly imposed by the structural constraints defining the cluster composition and can therefore be removed. To further show the versatility of the ARF, we next describe how various Binary Clustering Problems can be formulated using the asymmetric variables.

Note. In constraint sets (3.1)–(3.2), we can reduce the number of variables without losing any optimal solution. We can start by imposing that the first object is assigned to the first cluster as clusters can always be renumbered to satisfy this condition. For the second object, we can impose that if it does not belong to cluster one, it must belong to cluster two. The third object should belong to one of the first three clusters. We can continue this reasoning for the first $m-1$ objects. The remainder of the objects, from m to n, can be assigned to any of the m clusters. We further refer to this technique as *Variable Reduction* (VR). Note that this SF+VR, i.e. the formulation resulting from applying the VR technique to the SF formulation, is not equivalent to the ARF. Computational experiments on a Scene Selection Problem in Section 2.3 indicate that these two types of formulations can indeed have different LP values.

2.1 Bin Packing / Binary Cutting Stock

In the Bin Packing or Binary Cutting Stock, one has to pack a set $N = \{1, \ldots, n\}$ of objects, with a specific non-negative weight $w_i,\ i \in N$ into a set $K = \{1, \ldots, m\}$ of bins. Each bin has the same capacity W, and the objective is to minimize the number of bins used. The classical formulation (Kantorovitch 1960) using the binary variables $x_i^k = 1$ if object i is in bin k, and $x_0^k = 1$ if bin k is used, suffers from symmetry as one can arbitrarily permute the bins. In the ARF, let $v_i^h = 1$ if object $i \in N$ is in the same bin as object $h \in N$ and object h is the lowest numbered object in that bin, 0 otherwise. A bin is hence identified by its lowest numbered object. The ARF is as follows:

$$\text{Min} \sum_{h \in N} v_h^h \tag{4.1}$$

$$\text{s.t.} \sum_{h \in \{1,\ldots,i\}} v_i^h = 1 \qquad \forall i \in N \tag{4.2}$$

$$v_i^h \leq v_h^h \qquad \forall h \in N, \forall i \in \{h, \ldots, n\} \tag{4.3}$$

$$\sum_{i \in \{h,\ldots,n\}} w_i v_i^h \leq W v_h^h \qquad \forall h \in N \tag{4.4}$$

$$v_i^h \in \{0,\ 1\} \qquad \forall h \in N, \forall i \in \{h, \ldots, n\} \tag{4.5}$$

The objective (4.1) minimizes the number of bins used. In (4.2) each object must be assigned to exactly one bin and an object can only be assigned to a bin if the bin is used (4.3). The capacity of the bin is limited to W in (4.4). It is well known that the SF provides an LP lower bound value of $\sum_{i \in N} w_i / W$, the sum of the object lengths divided by the bin capacity. Elhedhli (2005) ranks LP bounds for various formulations, but he does not consider the above ARF. Indeed the ARF provides the same LP relaxation bound in most cases, except in the trivial case when the sum of the weights is less than the capacity of one bin. In the latter case, the ARF provides a better LP bound than the SF.

Proposition 1 *The ARF of the Bin Packing provides an LP value of $\sum_{i \in N} w_i / W$ if this value is greater than or equal to one, otherwise the LP value is one.*

Proof. Let $s = \left\lfloor \sum_{i \in N} w_i / W \right\rfloor \geq 1$ and $t = \sum_{i \in N} w_i / W - s \geq 0$. Assign one by one objects 1 to s to the first s bins ($v_h^h = 1,\ h \in \{1, \ldots, s\}$) and a fraction t of object $s+1$ to bin $s+1$, resulting in a reduced capacity of Wt for bin $s+1$ since $v_{s+1}^{s+1} = t$. Close all other bins, i.e., $v_h^h = 0$, $\forall h \in \{s+2, \ldots, n\}$. Assign (arbitrarily and using fractional amounts if necessary) the remaining fraction $1-t$ of item $s+1$ to the open bins 1 to

s and all the other items $s+2$ to n to the open bins 1 to $s+1$, so that the remaining capacity per bin is not exceeded. By construction, the total available capacity is $(s+t)W$, which is equal to the total lengths of all the objects, and the objective value is equal to $s+t$. Finally observe that if $s=0$, then a single bin is necessary and the LP value of the ARF is equal to one because $v_1^1 = 1$ according to (4.2). □

A variant of this problem is the Maximum Cardinality Bin Packing (Labbé, Laporte and Martello 2003, Peeters and Degraeve 2006) in which we must pack as many of the n objects as possible into m bins. The ARF is as follows, the meaning of the relations following the lines of the Bin Packing:

$$\text{Max} \sum_{h\in N} \sum_{i\in\{h,\ldots,n\}} v_i^h \tag{5.1}$$

$$\text{s.t.} \sum_{h\in\{1,\ldots,i\}} v_i^h \leq 1 \qquad \forall i \in N \tag{5.2}$$

$$\sum_{h\in N} v_h^h = m \tag{5.3}$$

$$v_i^h \leq v_h^h \qquad \forall h \in N, \forall i \in \{h,\ldots,n\} \tag{5.4}$$

$$\sum_{i\in\{h,\ldots,n\}} w_i v_i^h \leq W v_h^h \qquad \forall h \in N \tag{5.5}$$

$$v_i^h \in \{0,\ 1\} \qquad \forall h \in N, \forall i \in \{h,\ldots,n\} \tag{5.6}$$

A second variant is the Dual Bin Packing (Labbé, Laporte and Martello 1995, Peeters and Degraeve 2006) where the objective is to assign the n items to as many identical bins as possible, provided that the total weight assigned to each open bin is at least equal to a minimum value W. The ARF is as follows:

$$\text{Max} \sum_{k\in N} v_k^k \tag{6.1}$$

$$\text{s.t.} \sum_{h\in\{1,\ldots,i\}} v_i^h = 1 \qquad \forall i \in N \tag{6.2}$$

$$v_i^h \leq v_h^h \qquad \forall h \in N, \forall i \in \{h,\ldots,n\} \tag{6.4}$$

$$\sum_{i\in\{h,\ldots,n\}} w_i v_i^h \geq W v_h^h \qquad \forall h \in N \tag{6.5}$$

$$v_i^h \in \{0,\ 1\} \qquad \forall h \in N, \forall i \in \{h,\ldots,n\} \tag{6.6}$$

2.2 Node Capacitated Graph Partitioning

Assume a graph $G = (N, E)$ where $N = \{1,\ldots,n\}$ is the set of nodes and E is the set of edges. There is a non-negative weight w_i associated with each node and a cost c_{ij} for each edge $(i,j) \in E$. In the Node Capacitated Graph Partitioning, we assign the n nodes in G to at most m disjoint clusters so that the total cost of the edges between clusters is minimized or the total cost of the edges within a cluster is maximized. The total node weight within a cluster is restricted to the capacity W. A standard SF is described by Ferreira et al. (1996), for the case with non-negative edge costs. However, we can also provide an alternative ARF to this problem. In addition to the previously defined v_i^h variables, let $w_{ij}^h = 1$ if edge (i,j) lies within cluster h, i.e., the cluster in which h is the smallest numbered node, 0 otherwise. The ARF is then as follows:

$$\text{Max} \sum_{h\in N} \sum_{(i,j)\in E | h\leq i<j} c_{ij} w_{ij}^h \tag{7.1}$$

$$\text{s.t.} \sum_{h\in\{1,\ldots,i\}} v_i^h = 1 \qquad \forall i \in N \tag{7.2}$$

$$\sum_{h\in N} v_h^h \leq m \tag{7.3}$$

$$v_i^h \leq v_h^h \qquad \forall h \in N, \forall i \in \{h, \ldots, n\} \tag{7.4}$$

$$\sum_{i \in \{h,\ldots,n\}} w_i v_i^h \leq W v_h^h \qquad \forall h \in N \tag{7.5}$$

$$w_{ij}^h \leq v_i^h \qquad \forall h \in N, \forall (i,j) \in E : h \leq i < j \tag{7.6}$$

$$w_{ij}^h \leq v_j^h \qquad \forall h \in N, \forall (i,j) \in E : h \leq i < j \tag{7.7}$$

$$w_{ij}^h \in \{0,\ 1\} \qquad \forall h \in N, \forall (i,j) \in E : h \leq i < j \tag{7.8}$$

$$v_i^h \in \{0,\ 1\} \qquad \forall h \in N, \forall i \in \{h, \ldots, n\} \tag{7.9}$$

The objective function (7.1) maximizes the sum of the total cost of the edges within a cluster. In (7.2) each node is assigned exactly once to a cluster and in (7.3) at most m clusters are allowed. In constraint set (7.4), a node can only be assigned to a cluster that is chosen. The capacity of each cluster is limited to W in (7.5). Finally, (7.6)–(7.7) impose that an edge can only be within cluster h if both endpoints are assigned to that cluster h.

2.3 A Scene Selection Problem

The following Scene Selection Problem is taken from Van Hentenreyck (2002). In this problem, a movie producer has to decide which scenes should be shot on which days. In total, $n = 19$ scenes (set N of objects) have to be shot during at most $m = 5$ days (set K of clusters). It is not possible to shoot more than $W = 5$ scenes in a day. In total, 11 actors (set A of actors) are needed to make the movie. Each scene requires the presence of a specific set of actors: parameter $a_{ij} = 1$ if actor $j \in A$ is needed for scene $i \in N$, 0 otherwise. The actors are paid for each day they are required to be present, regardless of the number of scenes in which they appear on that day. The daily pay $p_j,\ j \in A$ varies per actor, but not per day. The objective is to minimize the total cost. Let the decision variables be defined as follows: $x_i^k = 1$ if scene $i \in N$ is shot on day $k \in K$, 0 otherwise; $y_j^k = 1$ if actor $j \in A$ is present on day $k \in K$, 0 otherwise. This optimization problem can then be expressed as follows:

$$\text{Min} \sum_{k \in K} \sum_{j \in A} p_j y_j^k \tag{8.1}$$

$$\text{s.t.} \sum_{k \in K} x_i^k = 1 \qquad \forall i \in N \tag{8.2}$$

$$\sum_{i \in N} x_i^k \leq W \qquad \forall k \in K \tag{8.3}$$

$$a_{ij} x_i^k \leq y_j^k \qquad \forall k \in K, \forall i \in N, \forall j \in A \tag{8.4}$$

$$y_j^k \in \{0,\ 1\} \qquad \forall k \in K, \forall j \in A \tag{8.5}$$

$$x_i^k \in \{0,\ 1\} \qquad \forall k \in K, \forall i \in N \tag{8.6}$$

The objective function (8.1) minimizes the total salary cost. Each scene is shot on exactly one day (8.2). The capacity constraints (8.3) require that at most W scenes are shot on each day. If a scene is shot on a specific day, then the required actors must be present (8.4). Both the x_i^k and y_j^k variables are binary, although it is technically sufficient to impose the binary restrictions only on the x_i^k variables.

Proposition 2 *The LP value of the Scene Selection Problem (8.1)–(8.6) is* $\sum_{j \in A} p_j$.

Proof. Assume n scenes, m days and a maximum of W scenes per day. Consider the following solution: $x_i^k = 1/m,\ \forall i \in N, \forall k \in K$ and $y_j^k = 1/m,\ \forall j \in A, \forall k \in K$. We first proof that this is a feasible solution. As the set K contains m days, it holds that $\sum_{k \in K} x_i^k = 1,\ \forall i \in N$. To have a feasible problem, we need at least $\lceil n/W \rceil$ days, hence $m \geq n/W$. Therefore $\sum_{i \in N} x_i^k = n(1/m) \leq n(W/n) = W$, and constraint (8.3) is satisfied. Constraints in set (8.4) are obviously satisfied. For this solution the objective value is $\sum_{j \in A} p_j$.

This is also the minimum value for the objective. For each actor j, there is at least one scene i for which $a_{ij} = 1$, so that summing both sides of (8.4) over the possible clusters gives: $\sum_{k\in K} a_{ij}x_i^k \leq \sum_{k\in K} y_j^k$. The left hand side is equal to 1 due to constraint (8.2) and $a_{ij} = 1$, hence $\sum_{j\in A} p_j(\sum_{k\in K} y_j^k) \geq \sum_{j\in A} p_j$. □

Formulation (8.1)–(8.6) contains a lot of symmetry. For any feasible solution, optimal or non-optimal, we can obtain an equivalent solution by permuting the numbering of the days, i.e., assigning a different day to the given clusters of scenes. We can decrease the symmetry using the VR technique. Take one scene at random, say scene one, and impose that it is shot on day one (days can always be renumbered to satisfy this condition). For scene two, we can impose that if it is not shot on the same day as scene one, it will be shot on day two, thereby imposing $x_2^3 = x_2^4 = x_2^5 = 0$, and constraint (8.2) for scene 2 becomes: $x_2^1 + x_2^2 = 1$. We can continue this reasoning for scenes 3 and 4. The remainder of the scenes can be shot on any day; hence for scene 5 and the following ones, we cannot reduce the variables.

We can also provide an ARF for this problem. Let $v_i^h = 1$ if scene i is shot on the same day as scene h and scene h is the lowest index scene shot that day, 0 otherwise. The variable y_j^h now indicates if actor j is present in the cluster identified by scene h, i.e., the cluster in which h is the smallest indexed scene.

$$\text{Min} \sum_{h\in N}\sum_{j\in A} p_j y_j^h \tag{9.1}$$

$$\text{s.t.} \sum_{h\in\{1,\ldots,i\}} v_i^h = 1 \qquad \forall i \in N \tag{9.2}$$

$$\sum_{h\in N} v_h^h \leq m \tag{9.3}$$

$$v_i^h \leq v_h^h \qquad \forall h \in N, \forall i \in \{h,\ldots,n\} \tag{9.4}$$

$$\sum_{i\in\{h,\ldots,n\}} v_i^h \leq W v_h^h \qquad \forall h \in N \tag{9.5}$$

$$a_{ij}v_i^h \leq y_j^h \qquad \forall h \in N, \forall i \in N, \forall j \in A \tag{9.6}$$

$$y_j^h \in \{0,\ 1\} \qquad \forall h \in N, \forall j \in A \tag{9.7}$$

$$v_i^h \in \{0,\ 1\} \qquad \forall h \in N, \forall i \in N \tag{9.8}$$

The objective function (9.1) minimizes the total salary cost. Constraint set (9.2) imposes that each scene is allocated to exactly one cluster and (9.3) indicates that we cannot use more than m clusters. In (9.4) we can only assign a scene to a cluster if that cluster is used. Further, if a cluster is used, it cannot have more than W scenes in it (9.5). Note that constraints (9.4) are implied by constraints (9.5). Finally constraint set (9.6) imposes that if scene i is in the cluster identified by scene h, then the actors required for scene i must be present in that cluster.

We now present some limited computational experiments conducted using the data provided in Van Hentenreyck (2002). We modeled the various formulations using ILOG OPL Development Studio 6.1 and the problems were solved using CPLEX 11.2 on a 2.4 GHz computer with 3GB RAM. We tested the following formulations identified in Table 1: the Symmetric Formulation (SF); the SF with variable reduction (SF+VR), and the Asymmetric Representatives Formulation (ARF) without the redundant constraint (9.4). Furthermore, we tested SF and SF+VR using three different orders for the input data. For the first arrangement called *Random*, we kept the numbering of the scenes as provided in Van Hentenreyck (2002). Next, we calculated for each scene the total cost for the actors needed in that particular scene. In the second arrangement (*LowHigh*), we ordered the scenes starting with the lowest total cost to the highest and in the third arrangement we inversed this (*HighLow*). Finally, as a preview of the Section 3, we also present the LP value provided by the Dantzig-Wolfe reformulation of both SF and ARF identified by DWSF and DWARF, respectively.

CPLEX 11.2 contains a feature which is *Default Symmetry Breaking*. This specific setting can be found under: Settings: Mathematical Programming – Preprocessing – Symmetry. The manual does, however, not

Table 1: Scene Selection Problem: CPU times in seconds, LP value and IP Gap

	Default SB LP (sec)	**No SB LP (sec)**	**LP value**	**IP Gap**
SF	4.00	14.00	137 739.00	58.8%
SF + VR (*Random*)	5.79	5.79	177 075.57	47.0%
SF + VR (*LowHigh*)	8.53	8.50	157 388.55	52.9%
SF + VR (*HighLow*)	2.75	2.73	219 013.83	34.5%
ARF (*Random*)	6.95	6.96	228 764.46	31.5%
ARF (*LowHigh*)	35.70	35.40	187 083.07	44.0%
ARF (*HighLow*)	0.78	0.79	317 201.97	5.1%
DWSF	1.34	1.34	330 405.41	1.1%
DWARF	1.13	1.12	330 405.41	1.1%

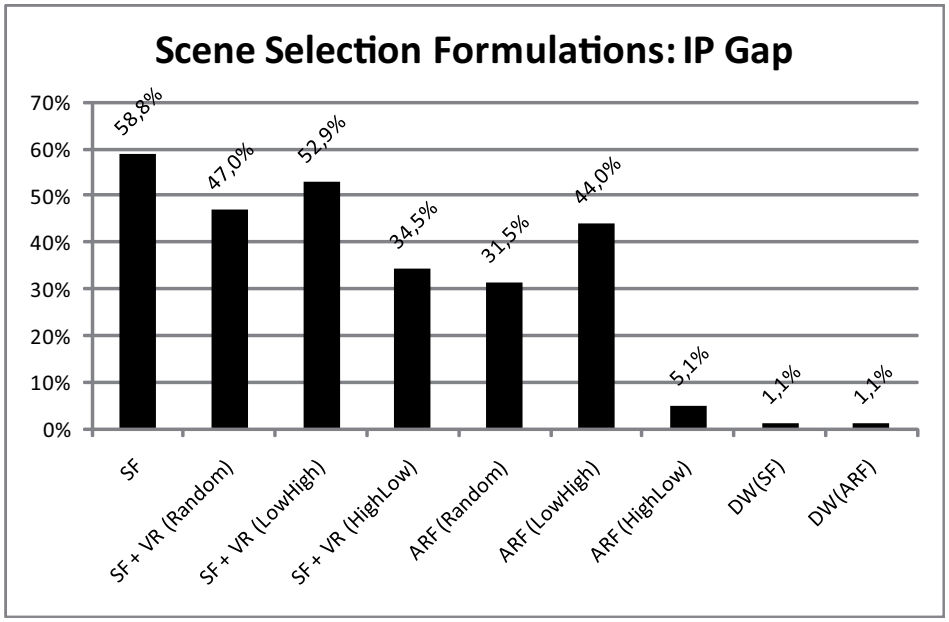

Figure 2: Scene Selection Formulations: IP Gap

give any further information on the way this is done. In Table 1, we provide the CPU time in seconds to calculate the LP value for both the default setting for symmetry breaking (Default SB LP (sec)) and the setting where this feature is turned off (No SB LP (sec)). Note that this symmetry breaking feature only has a significant effect for the SF. In the last two columns of Table 1, we provide the LP value and the IP Gap in percentage compared to the optimal integer value 334 144. The IP Gap is illustrated in Figure 1.

The computational results indicate two important issues. First, this is an example of a formulation where the various LP values of ARF (from 187 083.07 to 317 201.97) improve on the LP value provided by SF (137 739.00). The IP gap improved from 58.8% to 5.1% for ARF (*HighLow*). Second, we observe that the LP values of SF+VR and ARF depend on the order of the input data. This can partially be explained as follows. In the ARF, we fix the first variable: $v_1^1 = 1$. This also happens in the LP relaxation, and it forces that the actors needed for that first scene must be present via constraint (9.6). Hence in the LP relaxation, we are fully charged for the cost of the first scene. For the second scene, there are only two possibilities: either it is in the first cluster or in the second, so we induce less fractional values for the decision variables. A similar behavior occurs in SF+VR. From Table 1, we see indeed that the order *HighLow* performs substantially better in terms of LP value compared to the order *LowHigh*, for both SF+VR and ARF. For SF and the decomposition formulations, the order of the input data has no impact on the LP value, so we only report the results for the *Random* order.

2.4 The Vehicle Routing Problem with Time Windows

The vehicle routing problem with time windows (VRPTW) is defined on a directed graph with node set $N \cup \{0,\ n+1\}$ and arc set A. It can be described as finding least cost routes for a set $K = \{1, \ldots, m\}$ of identical vehicles to be routed and scheduled originating from a single depot (split into two for modeling reasons with start at node 0 and end at node $n+1$) such that the set $N = \{1, \ldots, n\}$ of customers is visited exactly once and the demand d_i, $i \in N$ is met. Furthermore, each node $i \in N$ must be visited within a time window $[e_i, l_i]$ and the load of a vehicle must not exceed its capacity D. The cost and travel time (including service time at one end node) for traversing the arc $(i, j) \in A$ are given as c_{ij} and t_{ij}, respectively.

The mathematical programming formulation (adapted from Desaulniers et al. 1998) involves two types of structural variables: binary flow variables X_{ij}^k equal to one if arc $(i, j) \in A$ is used by vehicle $k \in K$, 0 otherwise; and time variables T_i^k specifying the arrival time at node $i \in N \cup \{0, n+1\}$ of vehicle k. The SF also uses the binary assignment variables $x_i^k = \sum_{j|(j,i)\in A} X_{ji}^k$, for $k \in K$ and $i \in N \cup \{0\}$. For the ease of the presentation, for $k \in K$, let $z^k = \sum_{(i,j)\in A} c_{ij} X_{ij}^k$, $\boldsymbol{X}^k = (X_{ij}^k)_{(i,j)\in A}, \boldsymbol{T}^k = (T_i^k)_{i\in N\cup\{0,\ n+1\}}$, and $\boldsymbol{x}^k = (x_i^k)_{i\in N}$.

The SF for finding the minimum cost set of routes can be given as:

$$\text{Min} \sum_{k\in K} z^k \tag{10.1}$$

$$\text{s.t.} \sum_{k\in K} x_i^k = 1 \qquad \forall i \in N \tag{10.2}$$

$$(\boldsymbol{X}^k, \boldsymbol{T}^k, \boldsymbol{x}^k, z^k) \in S(x_0^k) \qquad \forall k \in K \tag{10.4}$$

$$x_0^k \text{ binary} \qquad \forall k \in K \tag{10.4}$$

where the generic solution space $S(x_0)$, which is identical for all $k \in K$, is defined as:

$$S(x_0) = \left\{\boldsymbol{X} = (X_{ij})_{(i,j)\in A}, \boldsymbol{T} = (T_i)_{i\in N\cup\{0,n+1\}}, \boldsymbol{a} = (a_i)_{i\in N}, c\right\} \text{ such that:}$$

$$\sum_{j|(i,j)\in A} X_{ij} - \sum_{j|(j,i)\in A} X_{ji} = \begin{cases} x_0 & i = 0 \\ 0 & \forall i \in N \\ -x_0 & i = n+1 \end{cases} \tag{11.1}$$

$$a_i = \sum_{j|(j,i)\in A} X_{ji} \qquad \forall i \in N \tag{11.2}$$

$$c = \sum_{(i,j)\in A} c_{ij} X_{ij} \tag{11.3}$$

$$\sum_{i\in N} d_i a_i \le D x_0 \tag{11.4}$$

$$X_{ij}(T_i + t_{ij} - T_j) \le 0 \qquad \forall (i,j) \in A \tag{11.5}$$

$$T_0 = 0 \tag{11.6}$$

$$e_i a_i \le T_i \le l_i a_i \qquad \forall i \in N \tag{11.7}$$

$$X_{ij} \in \{0,\ 1\} \qquad \forall (i,j) \in A \tag{11.8}$$

If $x_0 = 1$, $S(x_0)$ provides $\boldsymbol{X}$ and $\boldsymbol{T}$, the structural flow and time variables describing an elementary path (if it exits) from 0 to $n+1$ satisfying time window and capacity constraints, vector $\boldsymbol{a} = (a_i)_{i\in N}$ identifies the customers visited while c gives the cost of the path; otherwise all flow, time and cost variables vanish. Note that in (10.1)–(10.4), constraint sets $x_i^k \in \{0, 1\}$ and $x_i^k \le x_0^k$, for $k \in K,\ i \in N$ need not be present as they are imposed by the solution space $S(x_0), x_0$ binary.

The ARF rather uses the binary assignment variables v_i^h, for $h \in N$ and $i \in \{h, \ldots, n\}$. Specifically, $v_i^h = 1$ if client i is visited by the same vehicle as client h, and client h is the lowest indexed client in the cluster.

The solution space is now specific for every $h \in N$ and it is given by:

$$AS^1\left(v_0^1\right) = \left\{(\boldsymbol{X}, \boldsymbol{T}, \boldsymbol{a}, c) \in S(v_0^1) | a_1 = v_0^1\right\}$$

$AS^h\left(v_0^h\right) = \left\{(\boldsymbol{X}, \boldsymbol{T}, \boldsymbol{a}, c) \in S(v_0^h) | a_h = v_0^h a_i = 0,\ \forall i \in \{1, \ldots, h-1\}\right\}$, for $h \in N \backslash \{1\}$.

The ARF for the VRPTW can be expressed as:

$$\text{Min} \sum_{h \in N} w^h \tag{12.1}$$

$$\text{s.t.} \sum_{h \in \{1,\ldots,i\}} v_i^h = 1 \qquad \forall i \in N \tag{12.2}$$

$$\sum_{h \in N} v_0^h \leq m \tag{12.3}$$

$$(\boldsymbol{X}^h, \boldsymbol{T}^h, \boldsymbol{v}^h, w^h) \in AS^h(v_0^h) \qquad \forall h \in N \tag{12.4}$$

$$v_0^h \text{binary} \qquad \forall h \in N \tag{12.5}$$

If $v_0^h = 1$, $AS^h(v_0^h)$ provides an elementary path from 0 to $n+1$ satisfying time window and capacity constraints visiting customer h but none of the lower numbered customers; otherwise all flow, time and cost variables vanish. Constraint sets $v_i^h \in \{0, 1\}$ and $v_i^h \leq v_0^h$, for $h \in N,\ i \in \{h, \ldots, n\}$ need not be present in (12.1)–(12.5) as they are imposed in the definition of the solution space.

2.5 Discussion

As the previous examples have shown, the ARF (as well as SF) is very versatile and can be used to model side constraints that are found in Binary Clustering Problems. The side constraints include: bound on the maximum number of clusters (Node Coloring, Node Capacitated Graph Partitioning); upper and lower bounds on the number of objects in a cluster (Scene Selection); capacity constraints (Bin Packing, Node Capacitated Graph Partitioning, VRPTW); two objects not allowed in the same cluster (Node Coloring).

Another possible side constraint not present in the above examples is the requirement that two objects i and j, with $i < j$, are forced to be in the same cluster. In the ARF, this can be modeled as $v_i^h = v_j^h, \forall h \in \{1, \ldots, i\}$.

A typical objective function is to minimize the number of clusters used (Node Coloring, Bin Packing), but more complex objective functions requiring structural variables can be modeled as well (Node Capacitated Graph Partitioning, Scene Selection, VRPTW). The structural constraint set $S\left(x_0\right),\ x_0$ binary, in the VRPTW is formulated using non-linear constraints and the cost function can also be non-linear, e.g., a function of the traveling time, of the arrival time at the nodes, or of the vehicle load (see Desaulniers et al. 1998).

3 Dantzig-Wolfe Reformulations of SF and ARF

In this section, we first give generic models for both the SF and the ARF. Next, we prove that the LP relaxation of the ARF is at least as good as the LP relaxation of the SF. Moreover, we also prove that the Dantzig-Wolfe reformulation of both the SF and the ARF results in the same model, hence the same linear relaxation bound.

We impose the following general restriction on the binary clustering problems: the structural constraints are identical for each cluster, otherwise one could not obtain symmetrical solutions by the permutation of the cluster numbers. Additionally we also impose that the cost of a cluster is independent of the other clusters. In most of the practical situations this is true but not for all. For example, this assumption is not satisfied if the cost of a Cutting Stock cluster depends on the repositioning of the knives from the previously used cluster.

3.1 The SF vs. ARF

The SF uses the following notation:

- $\boldsymbol{x}^k = (x_i^k)_{i\in\{1,\ldots,n\}}$ where $x_i^k = 1$ if object i is in cluster $k \in \{1,\ldots,m\}$, 0 otherwise;
- $x_0^k = 1$ indicating if cluster $k \in \{1,\ldots,m\}$ is used, 0 otherwise;
- $z^k \in \mathbb{R}$ providing the cost of cluster $k \in \{1,\ldots,m\}$;
- $\boldsymbol{y}_{SF}^k \in C\left(x_o^k\right)$, $k \in \{1,\ldots,m\}$ expressing the cluster rules.

The generic solution space is the same for all clusters, and therefore we omit the cluster index k. It is given as:

$$S\left(x_0\right) = \{\boldsymbol{y} \in C\left(x_0\right), \boldsymbol{a} = \boldsymbol{a}\left(\boldsymbol{y}\right) \in \{0,\ 1\}^n, c = c\left(\boldsymbol{y}\right) \in \mathbb{R} \mid\ a_i(\boldsymbol{y}) \leq x_0,\ \forall i \in \{1,\ldots,n\}\}.$$

$C\left(x_0\right)$ expresses all the cluster rules in terms of the structural variable vector $\boldsymbol{y}$; $\boldsymbol{a}(\boldsymbol{y})$ is an n-dimensional function that sets to 1 variables in $\boldsymbol{a} = \left(a_1(\boldsymbol{y}),\ldots,a_n(\boldsymbol{y})\right)$ corresponding to the objects assigned to the cluster; $c(\boldsymbol{y})$ computes the cost c of the cluster (we assume that if the cluster is not used then $c\left(\boldsymbol{0}\right) = 0$). Since $a_i\left(\boldsymbol{y}\right) \leq x_0,\ \forall i \in \{1,\ldots,n\}$, no objects can be assigned to a cluster that is not used. There are at most $2^n - 1$ different non-empty assignment sets represented by $\boldsymbol{a} \in \{0,\ 1\}^{\ n}$. An SF for finding the minimum cost of a binary clustering problem is given by:

$$\text{Min} \sum_{k\in\{1,\ldots,m\}} z^k \tag{13.1}$$

$$\text{s.t.} \sum_{k\in\{1,\ldots,m\}} x_i^k = 1 \qquad \forall i \in \{1,\ldots,n\} \tag{13.2}$$

$$(\boldsymbol{y}_{SF}^k, \boldsymbol{x}^k, z^k) \in S(x_0^k) \qquad \forall k \in \{1,\ldots,m\} \tag{13.3}$$

$$x_0^k \text{ binary} \qquad \forall k \in \{1,\ldots,m\} \tag{13.4}$$

The ARF uses a similar notation:

- $\boldsymbol{v}^h = (v_i^h)_{i\in\{1,\ldots,n\}}$ where $v_i^h =\ 1$ if object $i \in \{h,\ldots,n\}$ is in cluster $h \in \{1,\ldots,n\}$, 0 otherwise (the cluster identifier is the lowest numbered object in it);
- $v_0^h =\ 1,\ h \in \{1,\ldots,n\}$ indicating if cluster h is used, 0 otherwise;
- $w^h \in \mathbb{R}$ providing the cost of cluster $h \in \{1,\ldots,n\}$;
- $\boldsymbol{y}_{ARF}^h \in C\left(v_o^h\right)$, $h \in \{1,\ldots,n\}$, expressing the cluster rules.

The solution spaces for the structural variables are now specialized for every cluster. The ARF variables $\left(\boldsymbol{y}_{ARF}^h, \boldsymbol{v}^h, w^h\right) \in AS^h\left(v_o^h\right)$ are defined in terms of $S\left(v_o^h\right)$ but are different for every $h \in \{1,\ldots,n\}$. They are given as:

$$AS^1\left(v_o^1\right) = \left\{(\boldsymbol{y}, \boldsymbol{a}, c) \in S\left(v_o^1\right)\ \mid a_1 = v_o^1\right\};$$

$AS^h\left(v_o^h\right) = \left\{(\boldsymbol{y}, \boldsymbol{a}, c) \in S\left(v_o^h\right)\ \mid a_h = v_o^h;; a_i = 0,\ \forall i \in \{1,\ldots,h-1\}\right\}$, for $h \in \{2,\ \ldots, n\}$.

For $h \in \{1,\ldots,n\}$, if $AS^h(1) \neq \emptyset$, assignment and cost variables are defined as in $S(1)$, except that cluster h contains object h ($a_h = 1$) but no objects (if any) with a smaller index number ($a_i = 0,\ \forall i \in \{1,\ldots,h-1\}$). An alternative way to say it is that object $i \in \{1,\ldots,n\}$ can only be assigned to clusters up to index number i, i.e., clusters numbered $h \in \{1,\ldots,i\}$. Given the solution space $AS^h\left(v_o^h\right)$, vector $\boldsymbol{v}^h \in \{0,\ 1\}^n$, i.e., the assignment variables v_i^h are defined for $i, h \in \{1,\ldots,n\}$. Additionally $v_h^h = v_0^h, h \in \{1,\ldots,n\}$, and v_h^h can be used to identify the lowest index object in cluster h and to count the number of clusters used. An ARF for finding the minimum cost of a binary clustering problem is expressed as:

$$\text{Min} \sum_{h\in\{1,\ldots,n\}} w^h \tag{14.1}$$

$$\text{s.t.} \sum_{h\in\{1,\ldots,i\}} v_i^h = 1 \qquad \forall i \in \{1,\ldots,n\} \tag{14.2}$$

$$\sum_{h\in\{1,\ldots,n\}} v_0^h \leq m \tag{14.3}$$

$$(\boldsymbol{y}_{ARF}^h, \boldsymbol{v}^h, w^h) \in AS^h(v_0^h) \qquad \forall h \in \{1,\ldots,n\} \tag{14.4}$$

$$v_0^h \text{ binary} \qquad \forall h \in \{1,\ldots,n\} \tag{14.5}$$

In the above ARF, note that the first constraint in the assignment set (14.2) imposes that $v_1^1 = 1$. No such result can be derived from the SF. In Proposition 1, we showed that for the Bin Packing/Binary Cutting Stock, the LP relaxation value is the same for the SF and the ARF except when the sum of the item lengths is less than the bin capacity for which the ARF LP bound is larger. In the Scene Selection Problem, the LP values of various ARF were better than that of the SF. In the next proposition, we clarify the relationship between the LP values of the SF and the ARF.

Proposition 3 *For the Binary Clustering Problem, the LP value of ARF is at least as good as the LP value of SF.*

Proof. Consider the LP relaxations of SF and ARF. $\boldsymbol{y} \in C\left(x_0\right),\ \boldsymbol{a} = \boldsymbol{a}\left(\boldsymbol{y}\right),\ c = c\left(\boldsymbol{y}\right)$ need to be replaced by linear systems of equations: $\boldsymbol{By} = \boldsymbol{b}x_0$, $\boldsymbol{a} = \boldsymbol{Ay}$, and $c = \boldsymbol{c}^T\boldsymbol{y}$, respectively. $\boldsymbol{By} = \boldsymbol{b}x_0$ provides the structural constraints. The assignment matrix $\boldsymbol{A}$ has n rows (given as $\boldsymbol{A}_i,\ i \in \{1,\ldots,n\}$) and the appropriate number of columns according to the dimension of $\boldsymbol{y}$. Finally, $\boldsymbol{c}$ is the cost vector of the structural variables $\boldsymbol{y}$. Therefore both SF and ARF can be written in terms of $\boldsymbol{y}$, the SF with $\boldsymbol{y}_{SF}^k,\ k \in \{1,\ldots,m\}$ and the ARF with $\boldsymbol{y}_{ARF}^h,\ h \in \{1,\ldots,n\}$.

On the one hand, the LP relaxation of the SF reads as:

$$\text{Min} \sum_{k\in\{1,\ldots,m\}} c^T y_{SF}^k \tag{15.1}$$

$$\text{s.t.} \sum_{k\in\{1,\ldots,m\}} A_i y_{SF}^k = 1 \qquad \forall i \in \{1,\ldots,n\} \tag{15.2}$$

$$\boldsymbol{B}\boldsymbol{y}_{SF}^k = \boldsymbol{b}x_0^k \qquad \forall k \in \{1,\ldots,m\} \tag{15.3}$$

$$0 \leq \boldsymbol{A}_i\boldsymbol{y}_{SF}^k \leq 1 \qquad \forall k \in \{1,\ldots,m\}, \forall i \in \{1,\ldots,n\} \tag{15.4}$$

$$0 \leq x_0^k \leq 1 \qquad \forall k \in \{1,\ldots,m\} \tag{15.5}$$

Constraints (15.2) are the assignment constraints. Constraints (15.3) define the structure of the clusters. Constraints (15.4) are the relaxed conditions on the binary assignment variables and (15.5) are the relaxed conditions on the binary use of each cluster.

On the other hand, using the same kind of linearization, the LP relaxation of the ARF reads as:

$$\text{Min} \sum_{h\in\{1,\ldots,n\}} \boldsymbol{c}^T\boldsymbol{y}_{ARF}^h \tag{16.1}$$

$$\text{s.t.} \sum_{h\in\{1,\ldots,i\}} \boldsymbol{A}_i\boldsymbol{y}_{ARF}^h = 1 \qquad \forall i \in \{1,\ldots,n\} \tag{16.2}$$

$$\sum_{h\in\{1,\ldots,n\}} v_0^h \leq m \tag{16.3}$$

$$\boldsymbol{B}\boldsymbol{y}_{ARF}^h = \boldsymbol{b}v_0^h \qquad \forall h \in \{1,\ldots,n\} \tag{16.4}$$

$$\begin{array}{l} \boldsymbol{A}_1\boldsymbol{y}_{ARF}^1 = v_0^1 \\ \boldsymbol{A}_h\boldsymbol{y}_{ARF}^h = v_0^h,\ \boldsymbol{A}_i\boldsymbol{y}_{ARF}^h = 0 \end{array} \qquad \forall h \in \{2,\ldots,n\},\ i \in \{1,\ldots,h-1\} \tag{16.5}$$

$$0 \leq \boldsymbol{A}_i \boldsymbol{y}^h_{ARF} \leq 1 \qquad \forall h \in \{1,\ldots,n\}, \forall i \in \{1,\ldots,n\} \tag{16.6}$$

$$0 \leq v^h_0 \leq 1 \qquad \forall h \in \{1,\ldots,n\} \tag{16.7}$$

To prove that the LP value of the ARF is at least as good as the LP value of the SF, we have to show that any LP solution $(v^h_0, \boldsymbol{y}^h_{ARF}, h \in \{1,\ldots,n\}$ to ARF is also feasible for the LP relaxation of SF. Since $\sum_{h\in\{1,\ldots,n\}} v^h_0 \leq m$, it is easy to verify that

$$x^k_0 = \frac{\sum_{h\in\{1,\ldots,n\}} v^h_0}{m}, \boldsymbol{y}^k_{SF} = \frac{\sum_{h\in\{1,\ldots,n\}} \boldsymbol{y}^h_{ARF}}{m}, \qquad \forall k \in \{1,\ldots,m\}$$

satisfy the three constraint sets of the LP relaxation of the SF. Indeed, all clusters in SF are identical. Obviously $0 \leq x^k_0 \leq 1,\ \forall k \in \{1,\ldots,m\}$. The structural constraints of SF are satisfied by simply summing up the structural constraints of ARF and dividing by m:

$$\boldsymbol{B}\left(\frac{\sum_{h\in\{1,\ldots,n\}} \boldsymbol{y}^h_{ARF}}{m}\right) = \boldsymbol{b}\left(\frac{\sum_{h\in\{1,\ldots,n\}} v^h_0}{m}\right) \quad \Longrightarrow \quad \boldsymbol{B}\boldsymbol{y}^k_{SF} = \boldsymbol{b}x^k_0, \quad \forall k \in \{1,\ldots,m\}.$$

The assignment constraints (15.2) of SF are satisfied because, $\forall i \in \{1,\ldots,n\}$:

$$\begin{aligned}\sum_{k\in\{1,\ldots,m\}} \boldsymbol{A}_i \boldsymbol{y}^k_{SF} &= \sum_{k\in\{1,\ldots,m\}} \boldsymbol{A}_i \left(\frac{\sum_{h\in\{1,\ldots,n\}} \boldsymbol{y}^h_{ARF}}{m}\right) \\ &= \frac{1}{m} \sum_{k\in\{1,\ldots,m\}} \left(\sum_{h\in\{1,\ldots,i\}} \boldsymbol{A}_i \boldsymbol{y}^h_{ARF} + \sum_{h\in\{i+1,\ldots,n\}} \boldsymbol{A}_i \boldsymbol{y}^h_{ARF}\right) \\ &= \frac{1}{m} \sum_{k\in\{1,\ldots,m\}} (1+0) = 1,\end{aligned}$$

where, in the last equation, the first term is equal to one because of (16.2) while the second is zero because of (16.5). To complete the proof, we need only to note that both objective functions are in fact the same:

$$\sum_{h\in\{1,\ldots,n\}} \boldsymbol{c}^T \boldsymbol{y}^h_{ARF} = \sum_{k\in\{1,\ldots,m\}} \left(\frac{1}{m} \sum_{h\in\{1,\ldots,n\}} \boldsymbol{c}^T \boldsymbol{y}^h_{ARF}\right) = \sum_{k\in\{1,\ldots,m\}} \left(\boldsymbol{c}^T \sum_{h\in\{1,\ldots,n\}} \frac{\boldsymbol{y}^h_{ARF}}{m}\right) = \sum_{k\in\{1,\ldots,m\}} \boldsymbol{c}^T \boldsymbol{y}^k_{SF}.$$

□

Note. In some applications, authors do not use the total set of the structural constraints but a relaxation of it. In the VRPTW for example, many authors replace the elementary paths by paths from the origin depot to the destination depot, allowing for multiple visits at the same customer, and enforcing feasibility of the solution in a branch-and-bound search tree. In that case, the above result is still valid as long as both SF and the ARF use the same set of structural constraints.

3.2 Dantzig-Wolfe Reformulations

Both the SF and the ARF have a block angular structure over the cluster indices, SF with m blocks and ARF with n blocks. These formulations are well suited for the application of the Dantzig-Wolfe decomposition principle.

We start with the reformulation of the SF, keeping the objective function (13.1) and the set of the assignment constraints (13.2) in the master problem and the m cluster blocks (13.3)–(13.4) in the subproblems. Let $\boldsymbol{0} \neq \boldsymbol{y_p},\ p \in P$ be the finite set of non-null cluster solutions to $C\,(1)$ and let $\boldsymbol{y}_0 = \boldsymbol{0}$ be the null vector solution to $C\,(0)$. Define $P_0 = P \cup \{0\}$ the set of indices. Let $\boldsymbol{a}_p = \boldsymbol{a}\left(\boldsymbol{y}_p\right),\ p \in P$ be the assignment vector and

$c_p = c(\boldsymbol{y}_p)$ its cost. Any SF solution vector $\left(\boldsymbol{x}^k,\ z^k\right),\ k \in \{1,\ldots,m\}$, can be expressed as a binary convex combination of vectors $(\boldsymbol{a}_p,\ c_p), p \in P_0$, such that:

$$\boldsymbol{x}^k = \sum_{p\in P_0} \boldsymbol{a}_p\theta_p^k,\ \sum_{p\in P_0} \theta_p^k = 1, \theta_p^k \in \{0,\ 1\},\ p \in P_0,\ \text{and } z^k = \sum_{p\in P_0} c_p\theta_p^k.$$

This is the discrete version of the Dantzig-Wolfe decomposition in terms of the vector points $(\boldsymbol{a}_p,\ c_p), p \in P_0$, see Lübbecke and Desrosiers (2005). Because binary and possibly continuous variables are present in the subproblem but only binary variables appear in the master problem, we can impose the binary conditions directly on the new θ_p^k variables (Jans 2010). The substitution in the SF provides the following SFDW reformulation:

$$\begin{aligned}
\text{Min} \quad & \sum_{k\in\{1,\ldots,m\}} \sum_{p\in P_0} c_p\theta_p^k & & \\
\text{s.t.} \quad & \sum_{k\in\{1,\ldots,m\}} \sum_{p\in P_0} a_{ip}\theta_p^k = 1 & & \forall i \in \{1,\ldots,n\} \\
& \sum_{p\in P_0} \theta_p^k = 1 & & \forall k \in \{1,\ldots,m\} \\
& \theta_p^k \text{ binary} & & \forall k \in \{1,\ldots,m\}, p \in P_0
\end{aligned}$$

Because $c_0 = 0$ and $a_{i0} = 0,\ \forall i \in \{1,\ldots,n\}$, the null index variables can be removed and the convexity constraints written as inequalities. Define variables $\theta_p = \sum_{k\in\{1,\ldots,m\}} \theta_p^k$. Summing the convexity constraints over $k \in \{1,\ldots,m\}$, we have $\sum_{k\in\{1,\ldots,m\}} \sum_{p\in P} \theta_p^k \leq m \Leftrightarrow \sum_{p\in P} \theta_p \leq m$. The SFDW reformulation becomes:

$$\begin{aligned}
\text{Min} \quad & \sum_{p\in P} c_p\theta_p & & & (17.1)\\
\text{s.t.} \quad & \sum_{p\in P} a_{ip}\theta_p = 1 & & \forall i \in \{1,\ldots,n\} & (17.2)\\
& \sum_{p\in P} \theta_p \leq m & & & (17.3)\\
& \theta_p \text{ binary} & & p \in P & (17.4)
\end{aligned}$$

Consider now the Dantzig-Wolfe reformulation of the ARF. Observe first that the sets $AS^h(1),\ h \in \{1,\ldots,n\}$ form a partition of $S(1)$. Note also that the null solution $(\mathbf{0},\mathbf{0},0)$ is the only solution to $S(0)$ and $AS^h(0),\ h \in \{1,\ldots,n\}$. For $h \in \{1,\ldots,n\}$, let $\left(\boldsymbol{y}_p,\boldsymbol{a}_p,c_p\right) \in S(1), p \in P^h$ be the set of non-empty solution points to $AS^h(1)$. Note that $P^h,\ \ h \in \{1,\ldots,n\}$ is a partition of the index set P, and therefore we have that $P = \cup_h P^h$. As a consequence, the index sets are defined as follows:

$$P^1 = \left\{1,\ 2,\ \ldots, \left|P^1\right|\right\},\ P^2 = \left\{\left|P^1\right|+1, \left|P^1\right|+2, \ldots, \left|P^1\right| + \left|P^2\right|\right\}, \ldots$$

Define $P_0^h = P^h \cup \{0\}$ to be the augmented index set. As before, any ARF solution vector $\left(\boldsymbol{v}^h,\ w^h\right),\ h \in \{1,\ldots,n\}$, can be expressed as a binary convex combination of the vectors $(\boldsymbol{a}_p,\ c_p), p \in P_0^h$ such that:

$$\boldsymbol{v}^h = \sum_{p\in P_0^h} \boldsymbol{a}_p\theta_p,\ \sum_{p\in P_0^h} \theta_p = 1.\ \theta_p \in \{0,\ 1\},\ p \in P_0^h,\ \text{and } w^h = \sum_{p\in P_0^h} c_p\theta_p.$$

This is again the discrete version of the Dantzig-Wolfe decomposition, this time in terms of the vector points $(\boldsymbol{a}_p,\ c_p), p \in P_0^h,\ \ h \in \{1,\ldots,n\}$. Making the substitution in (14.1)–(14.3) of the ARF, we obtain the following ARFDW reformulation:

$$\text{Min} \quad \sum_{h\in\{1,\ldots,n\}} \sum_{p\in P_0^h} c_p\theta_p$$

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{h\in\{1,\ldots,i\}} \sum_{p\in P_0^h} a_{ip}\theta_p = 1 && \forall i \in \{1,\ldots,n\} \\
& \sum_{h\in\{1,\ldots,n\}} \sum_{p\in P_0^h} a_{hp}\theta_p \leq m \\
& \sum_{p\in P_0^h} \theta_p = 1 && \forall h \in \{1,\ldots,n\} \\
& \theta_p \text{ binary} && \forall h \in \{1,\ldots,n\}, p \in P_0^h
\end{aligned}
$$

Because $c_0 = 0$ and $a_{i0} = 0,\ i \in \{1,\ldots,n\}$, we can discard the null index variables in the objective function and in the first two sets of constraints. We can also do it in the convexity constraints and replace them by inequalities. However these inequality constraints are redundant because of the assignment constraints and can simply be removed. Indeed, for all $i \in \{1,\ldots,n\}$, we have

$$
\begin{aligned}
\sum_{h\in\{1,\ldots,i\}} \sum_{p\in P^h} a_{ip}\theta_p &= \sum_{p\in P^i} a_{ip}\theta_p + \sum_{h\in\{1,\ldots,i-1\}} \sum_{p\in P^h} a_{ip}\theta_p \\
&= \sum_{p\in P^i} \theta_p + \sum_{h\in\{1,\ldots,i-1\}} \sum_{p\in P^h} a_{ip}\theta_p = 1 \Rightarrow \sum_{p\in P^i} \theta_p \leq 1.
\end{aligned}
$$

In the above relations, $a_{ip} = 1,\ \forall p \in P^i, i \in N$, that is, object $i \in N$ is by definition in the cluster identified by its index number (see Figure 1). Using the same argument, we can simplify the constraint on the number of clusters and the ARFDW reformulation becomes:

$$
\begin{aligned}
\text{Min} \quad & \sum_{h\in\{1,\ldots,n\}} \sum_{p\in P^h} c_p\theta_p && && (18.1) \\
\text{s.t.} \quad & \sum_{h\in\{1,\ldots,i\}} \sum_{p\in P^h} a_{ip}\theta_p = 1 && \forall i \in \{1,\ldots,n\} && (18.2) \\
& \sum_{h\in\{1,\ldots,n\}} \sum_{p\in P^h} \theta_p \leq m && && (18.3) \\
& \theta_p \text{ binary} && \forall h \in \{1,\ldots,n\}, p \in P^h && (18.4)
\end{aligned}
$$

Proposition 4 *For Binary Clustering Problems, the Dantzig-Wolfe reformulations SFDW and ARFDW of both the SF and the ARF result in the same model.*

Proof. Since $P^h,\ \ h \in \{1,\ldots,n\}$ is a partition of the index set P, we have that $P = \cup_h P^h$ and therefore, (17.1), (17.3) and (17.4) can directly be written as (18.1), (18.3) and (18.4), respectively. To complete the proof, we have to show the equivalence of the assignment constraints. Indeed, in (17.2), $\forall i \in N$,

$$
1 = \sum_{p\in P} a_{ip}\theta_p = \sum_{h\in\{1,\ldots,i\}} \sum_{p\in P^h} a_{ip}\theta_p + \sum_{h\in\{i+1,\ldots,n\}} \sum_{p\in P^h} a_{ip}\theta_p
$$

where $a_{ip} = 0,\ \forall p \in P^h, h > i$ (see Figure 1). The second part of the right-hand side can be removed and hence, both formulations result in the same model. □

4 Conclusion

In this paper, we show that the main idea of the Asymmetric Representatives Formulation, as proposed by Campêlo et al. (2008) for the Node Coloring Problem, can be applied to obtain symmetry-breaking formulations for a large class of problems, namely the Binary Clustering Problems. Many well-known problems such as bin packing, graph coloring, graph partitioning, vehicle routing with time windows, and min-cut clustering

fall into this general class of Binary Clustering Problems. The LP relaxation of the ARF is always as least as good as the LP relaxation of the Symmetric Formulation.

The main contribution of this paper is that both the Symmetric and Asymmetric formulations for Binary Clustering problems result in the same symmetry-breaking model when we apply Dantzig-Wolfe decomposition. This opens up avenues for better branch-and-price algorithms. Even though for binary integer programming problems, the binary conditions can be imposed on the new variables in a decomposed reformulation, in practice the branching is done on the original variables (see e.g. Barnhart et al 1998). When branching on the original variables in the ARF, we avoid the symmetry difficulties during the branch-and-price algorithm, whereas the symmetry is still present when branching on the original variables in the SF. Specifically, it will be worthwhile to investigate whether ARF performs much better than the SF in a practical implementation of a branch-and-price algorithm based on the decomposition.

Another research avenue is related to the observation that the input order can affect the LP value for the ARF. Even though ARFDW is a stronger formulation than ARF, it would be interesting to investigate if and how we can exploit and optimize the input order.

Finally we could investigate if these results can be generalized to a larger class of problems, namely the Integer Clustering Problems.

References

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H. 1998. Branch-and-Price: Column generation for solving huge integer programs. *Operations Research*, 46 (3), 316–329.

Campêlo, M., Campos, V.A., Corrêa, R. 2008. On the asymmetric representatives formulation for the vertex coloring polytope. *Discrete Applied Mathematics*, 156, 1097–1111.

Desaulniers, G., Desrosiers, J., Ioachim, I., Solomon, MM., Soumis, F., Villeneuve, D. 1998. A Unified Framework for Deterministic Time Constrained Vehicle Routing and Crew Scheduling Problems. In: T. Crainic and G. Laporte (eds.), *Fleet Management and Logistics*, Kluwer, Norwell, MA, 57–93.

Dantzig, G.B., Wolfe, P. 1960. Decomposition principle for linear programs. *Operations Research*, 8, 101–111.

Elhedhli, S. 2005. Ranking lower bounds for the bin-packing problem. *European Journal of Operational Research*, 160, 34–46.

Ferreira, C.E., Martin, A., de Souza, C.C., Weismantel, R., Wolsey, L.A. 1996. Formulations and valid inequalities for the node capacitated graph partitioning problem. *Mathematical Programming*, 74, 247–266.

Jans, R. 2010, Classification of Dantzig-Wolfe reformulations for binary mixed integer programming problems. *European Journal of Operational Research*, 204, 251–254.

Labbé, M., Laporte, G., Martello, S. 1995. An exact algorithm for the dual bin packing problem. *Operations Research Letters* 17 9-18.

Labbé, M., Laporte, G., Martello, S. 2003. Upper bounds and algorithms for the maximum cardinality bin packing problem. *European Journal of Operational Research*, 149, 490–498.

Lübbecke, M.E., Desrosiers, J. 2005. Selected Topics in Column Generation. *Operations Research*, 53 (6), 1007–1023.

Kaibel, V., Pfetsch, M.E. 2008, Packing and Partitioning Orbitopes. *Mathematical Programming Series A*, 114, 1–36.

Kantorovitch, L.V. 1960. Mathematical methods of organizing and planning production. *Management Science*, 6, 366–422. Translation from the Russian original, dated 1939.

Margot, F. 2010. Symmetry in Integer Linear Programming. In: Jünger, M. et al., *50 Years of Integer Programming 1958-2008: From the early years to the state-of-the-art*, Springer, 647–686.

Mehrotra, A., Trick, M.A. 1996. A column generation approach for graph coloring. *INFORMS Journal on Computing*, 8 (4), 344–354.

Méndez-Díaz, I., Zabala, P. 2006. A Branch-and-Cut algorithm for graph coloring. *Discrete Applied Mathematics*, 154, 826–847.

Peeters, M., Degraeve, Z. 2006. Branch-and-Price algorithms for the dual bin packing and maximum cardinality bin packing problem. *European Journal of Operational Research*, 170, 416–439.

Van Hentenryck, P. 2002. Constraint and Integer Programming in OPL. *INFORMS Journal on Computing*, 14 (4), 345–372.