



HAL
open science

Two-phase Branch & Cut for the Symmetric Weight Matrix Knapsack polytope

Alexandre Heintzmann, Pascale Bendotti, Cécile Rottner

► **To cite this version:**

Alexandre Heintzmann, Pascale Bendotti, Cécile Rottner. Two-phase Branch & Cut for the Symmetric Weight Matrix Knapsack polytope. 2023. hal-03992007

HAL Id: hal-03992007

<https://hal.science/hal-03992007>

Preprint submitted on 16 Feb 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Abstract

In this paper, we define a new variant of the knapsack problem, the Symmetric Weight Matrix Knapsack Problem (SMKP). The (SMKP) is the core structure of the Hydro Unit Commitment being a production scheduling problem relative to hydroelectric units. The (SMKP) is proven to be NP-hard. The main contribution is a polyedral study of the (SMKP) and a dedicated two-phase Branch & Cut scheme. The polyedral study focuses on inequalities with 0-1 coefficients. Necessary facet-defining conditions are described, through a new structure called pattern encoding the symmetries of the (SMKP). A special case of patterns is identified where these conditions are necessary and sufficient. A two-phase Branch & Cut scheme is defined to exploit the symmetries on the pattern inequalities. As a pre-processing step, the first phase aims to produce patterns verifying these conditions. The second phase separates the inequalities associated to the patterns produced in the first phase at the nodes of a Branch & Cut tree. Experimental results demonstrate the efficiency of the proposed scheme in particular with respect to default CPLEX.

Two-phase Branch & Cut for the Symmetric Weight Matrix Knapsack polytope

Alexandre Heintzmann^{1,2}, Pascale Bendotti¹, and Cécile Rottner¹

¹EDF Lab Paris-Saclay, 7 Bd. Gaspard Monge, 91120 Palaiseau, France
{alexandre.heintzmann, pascale.bendotti, cecile.rottner@edf.fr}

²LAAS-CNRS, Université de Toulouse, CNRS, INP, Toulouse, France
{alexandre.heintzmann@laas.fr}

February 16, 2023

1 Introduction

Consider N groups of M elements, where N and M are positive integers. Let item (i, j) be element j of group i . Item (i, j) has weight w_{ij} and value v_{ij} , both non-negative. Within each group, order constraints are such that any item (i, j) can be selected provided item $(i, j - 1)$ is selected. Let C be the maximum capacity. The Matrix Knapsack Problem (MKP) is to maximize the total value of the selected items, while the order constraints are verified, and the total weight of the selected items is less than or equal to C . The Symmetric Weight Matrix Knapsack Problem (SMKP) is a Matrix Knapsack Problem where item (i, j) has weight $w_{ij} = w_j$, i.e., the weight of item (i, j) does not depend on the group index i . It means that items (i, j) and (i', j) have the same weight, thus the knapsack is symmetrically weighted with respect to the groups. We define (N, M, w, v, C) an instance of the (SMKP).

Two well known formulations can be used for both the (MKP) and (SMKP), namely the so-called *incremental* formulation and the *multiple choice* formulation. The former involves order constraints, where the latter uses disjunctive constraints instead. Both formulations yield the same LP relaxation [4]. Note that in the incremental formulation, there is a particular precedence graph induced by order constraints as depicted in **Figure 1**(a). In the multiple choice formulation, it is a conflict graph induced by disjunctive constraints, namely a graph with edges between any pair of items in the same group. **Figure 1**(b) also shows the corresponding matrix representation that will be convenient in the sequel to illustrate on examples some properties.

The motivation for (SMKP) is that it is the core structure of the Hydro Unit Commitment (HUC) [8], which is a production scheduling problem relative to hydroelectric units.

The knapsack problem and its variants have been widely studied in the literature [9]. The (SMKP) and the (MKP) have not been studied yet, but can be related to some variants of the knapsack problem in the literature. As the order constraints are a special case of precedence constraints, the (MKP) is a direct special case of the Precedence Knapsack Problem (PKP) [2]. Similarly as disjunctive constraints can be used alternatively for the (MKP), the (MKP) is also a special case of the Disjunctive Knapsack Problem (DKP) [14]. Both the (PKP) and the (DKP) are generalizations of the classical Binary knapsack problem (BKP). Indeed, the (BKP) is a (PKP) without any precedence, as well as a (DKP) without any disjunctive constraints. Interestingly, the (MKP) is also a generalization of the (BKP), whereas the (SMKP) is not. Indeed, an (MKP) with exactly one item for each group does not have any order constraints, which is exactly a (BKP). However, the (SMKP) is not a generalization of the

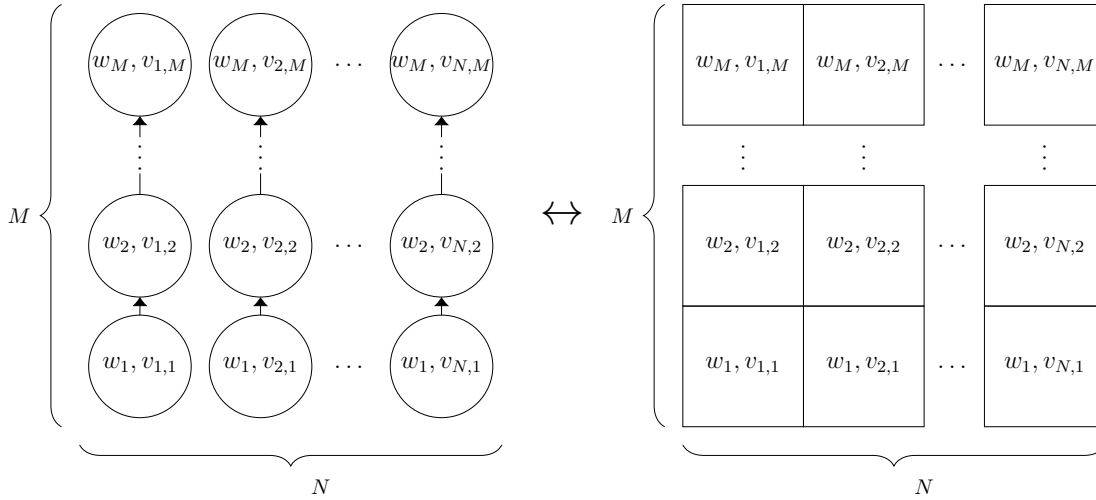


Figure 1: Graphic representation of the (SMKP)

(BKP). Indeed, because of the symmetric weight, an (SMKP) with one item per group would be a trivial (BKP), where each item has the exact same weight. Furthermore, the (SMKP) is not a Multiple knapsack problem [7] despite the presence of multiple groups. Indeed, a capacity constraint is applied on all groups simultaneously in the (SMKP), whereas it is applied for each group individually in the Multiple knapsack problem.

In this paper, the (SMKP) is proven to be NP-hard. A compact formulation is defined with its corresponding polytope. A literature review of facet-defining inequalities for problems related to the (SMKP) is exposed. The main contributions are the polyhedral study of the (SMKP) and a two-phase Branch & Cut (B&C) scheme. A new structure, called pattern, is defined, embedding the symmetries of the (SMKP). New inequalities, associated to the patterns and covering all the facets of the (SMKP) with 0-1 coefficients are introduced. Necessary facet-conditions are defined for these inequalities, which are also proven to be sufficient for a special case. An algorithm to generate patterns verifying these conditions is described and is used as pre-processing in the first phase of the B&C scheme. The separation algorithms, involved in the second phase of the B&C scheme, reduces to solve a maximum matching problem for the pool of the first phase generated patterns. To evaluate the efficiency of the corresponding inequalities, a numerical comparison shows that pattern inequalities largely outperform CPLEX generated cuts.

In **Section 2** the complexity of the (SMKP) is stated and some polyhedral results from the literature for variants of the KP are reported. In **Section 3** the patterns are defined as well as inequalities and facet-defining conditions are provided. In **Section 4** the two-phase B&C scheme is described. In **Section 5** are presented the experimental results on the efficiency of the proposed inequalities and compared to CPLEX. In **Section 6**, concluding remarks and perspectives for further research are drawn.

2 The Symmetric Weight Matrix Knapsack problem

We first prove the NP-hardness of the (SMKP). A literature review of related problems is done, and first polyhedral properties are provided.

2.1 Complexity

To state the complexity of the (SMKP), we introduce the Unbounded Integer knapsack problem (UIKP), which is a knapsack problem where items can be selected an unlimited number of times. The (UIKP) is NP-hard [15] and a reduction from this problem is considered.

Theorem 1. *The (SMKP) is NP-hard*

Proof. Let (M, w', v', C) be an instance of the (UIKP) with M items, with w'_j and v'_j the weight and the value of item $j \leq M$. The aim is to maximize the value of the selected items, each can be selected multiple times, while satisfying the maximum knapsack capacity C .

Let (N, M, w, v, C) be an instance of the (SMKP). The aim is to maximize the value of the selected items, verifying the order constraints within a group, and satisfying the maximum knapsack capacity C .

We set the following relations, without loss of generality, between the weights and the value of each knapsack: $w'_j = \sum_{k=1}^j w_k$ and $v'_j = \sum_{k=1}^j v_{i,k}$

As such, selecting once item j for (M, w', v', C) is equivalent to selecting elements 1 to j in a single group for (N, M, w, v, C) . In the following, we make the reduction in the case $N > C / \min(w'_j)$.

For a solution of (M, w', v', C) there is a solution of (N, M, w, v, C) . Let p_j be the number of times item j is selected in a solution of (M, w', v', C) . There is a solution of (N, M, w, v, C) that is equivalent: for each element j , there are p_j unique groups of (M, w', v', C) where only elements 1 to j are selected.

Similarly, for a solution of (N, M, w, v, C) there is a solution of (M, w', v', C) . Let p_j be the number of groups where only elements 1 to j are selected in a solution of (N, M, w, v, C) . There is a solution of (M, w', v', C) that is equivalent: each item j is selected exactly p_j times.

Thus, the (UIKP) is a special case of the (SMKP) where $N > C / \min(w'_j)$, and because the (UIKP) is NP-hard, the (SMKP) is NP-hard. \square

The (SMKP), and the (MKP), are generalizations of the Unbounded Integer Knapsack problem (UIKP). Consequently, the (MKP) is a generalization of both the (BKP) and the (UIKP). The Bounded Integer Knapsack problem (BIKP)[3] is also a generalization of the (BKP) and the (UIKP). Indeed, the (BKP) is a special case with upper bounds 1 on the number of repetition for each item, and the (UIKP) is a special case with finite upper bound sufficiently large to not be restrictive. However, the (BIKP) and the (SMKP) are not related. In fact, the (BIKP) considers a maximum number of repetitions for each item, while the (SMKP) would be closer to an Integer Knapsack problem with a shared upper bound on the total number of items, repetition included. This shared upper bound is the number of groups N .

All connections between the knapsack problem variants are depicted in **Figure 2** with a graph in which each vertex represents a variant and each arc indicates that the variant at the tail is a generalization of the variant at the head.

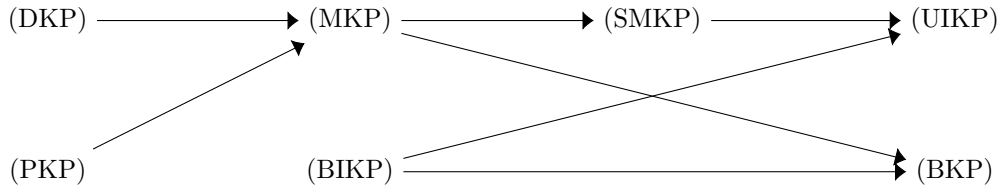


Figure 2: Generalization orders of some variants of the knapsack problem

2.2 The Symmetric Weight Matrix Knapsack polytope

Let x_{ij} be the binary variable such that $x_{ij} = 1$ if item (i, j) is selected in the solution. We denote \mathcal{V} the set of variables x_{ij} for the (SMKP). The total number of variables is $n = N \times M$. A compact *incremental formulation* of the (SMKP) is the following.

$$\begin{aligned} \max \quad & \sum_{i=1}^N \sum_{j=1}^M x_{ij} v_{ij} \\ \text{s.c.} \quad & \sum_{i=1}^N \sum_{j=1}^M x_{ij} w_j \leq C \end{aligned} \tag{1}$$

$$\begin{aligned} x_{ij} &\leq x_{i,j-1} & \forall x_{ij} \in \mathcal{V}, j \geq 2 \\ x_{ij} &\in \{0, 1\} & x_{ij} \in \mathcal{V} \end{aligned} \tag{2}$$

In this formulation, the objective function is to maximize the total value of the selected items. Inequality (1) is the capacity constraint, inequalities (2) correspond to the order constraints within each group. We define the polytope $P_{(SMKP)}$ the convex hull of the feasible solutions of the (SMKP):

$$P_{(SMKP)} = \text{conv} \left\{ x \in \{0, 1\}^n : x \text{ satisfies (1) - (2)} \right\}$$

Let z_{ij} be the binary variable such that $z_{ij} = 1$ if and only if items $(i, 1)$ to (i, j) are selected in the solution. We denote \mathcal{W} the set of variables z_{ij} for the (SMKP). A compact *multiple choice formulation* of the (SMKP) is the following.

$$\begin{aligned} \max \quad & \sum_{i=1}^N \sum_{j=1}^M z_{ij} \sum_{k=1}^j v_{ik} \\ \text{s.c.} \quad & \sum_{i=1}^N \sum_{j=1}^M z_{ij} \sum_{k=1}^j w_k \leq C \end{aligned} \tag{3}$$

$$\sum_{j=1}^M z_{ij} \leq 1 \quad \forall i \leq N \tag{4}$$

$$z_{ij} \in \{0, 1\} \quad \forall z_{ij} \in \mathcal{W}$$

The major difference is inequality (4), corresponding to the disjunctive constraints within each group. Otherwise these formulations are equivalent with respect to the variable change.

Because of the symmetric weights, if a solution is valid, then any symmetric solution with respect to the group indices is also valid. Moreover, the symmetries also appears in the facet-defining inequalities of the (SMKP).

Property 1. *If an inequality is facet-defining for the (SMKP), any of its symmetries is also facet-defining for the (SMKP).*

Proof. If an inequality is facet-defining for the (SMKP), there are $n + 1$ affinely independent valid solutions verifying the inequality to equality. As the weights are symmetric with respect to the groups, if a solution is valid, then any permutation of groups yields another valid solution. Hence, one can prove any symmetry of a facet-defining inequality to also be facet-defining, as it suffices to deduce the $n + 1$ valid solutions following the same permutation of groups. These new $n + 1$ points are necessarily affinely independent as they all undergo the exact same permutation of groups. \square

Example 1. Let $(4, 3, [3, 4, 2], v, 9)$ be an instance of the (SMKP). Besides trivial facets and inequalities (2), the convex hull of the incremental formulation for $(4, 3, [3, 4, 2], v, 9)$ contains 5 inequalities:

$$x_{12} + x_{22} + x_{32} + x_{41} \leq 1 \quad (i1)$$

$$x_{12} + x_{22} + x_{31} + x_{42} \leq 1 \quad (i2)$$

$$x_{12} + x_{21} + x_{32} + x_{42} \leq 1 \quad (i3)$$

$$x_{11} + x_{22} + x_{32} + x_{42} \leq 1 \quad (i4)$$

$$x_{11} + 2x_{12} + x_{21} + 2x_{22} + x_{31} + 2x_{32} + x_{41} + 2x_{42} \leq 3 \quad (i5)$$

Besides trivial facets and inequalities (4), the convex hull of the multiple choice formulation for $(4, 3, [3, 4, 2], v, 9)$ also contains 5 inequalities:

$$x_{12} + x_{13} + x_{22} + x_{23} + x_{32} + x_{33} + x_{41} + x_{42} + x_{43} \leq 1 \quad (m1)$$

$$x_{12} + x_{13} + x_{22} + x_{23} + x_{31} + x_{32} + x_{33} + x_{42} + x_{43} \leq 1 \quad (m2)$$

$$x_{12} + x_{13} + x_{21} + x_{22} + x_{23} + x_{32} + x_{33} + x_{42} + x_{43} \leq 1 \quad (m3)$$

$$x_{11} + x_{12} + x_{13} + x_{22} + x_{23} + x_{32} + x_{33} + x_{42} + x_{43} \leq 1 \quad (m4)$$

$$x_{11} + 3x_{12} + 3x_{13} + x_{21} + 3x_{22} + 3x_{23} + x_{31} + 3x_{32} + 3x_{33} \\ + x_{41} + 3x_{42} + 3x_{43} \leq 3 \quad (m5)$$

One can notice that the 4 inequalities with 0-1 coefficients are symmetries of one another with respect to the group indices, for both formulations of the (SMKP). Also, each facet-defining inequality of one formulation has an equivalent for the other formulation, which is consistent with the LP relaxation of both formulations being the same [4].

Based on the convex hull of many small instances, we experimentally distinguished three types of inequalities: the ones from the initial formulation, *binary inequalities* with 0-1 coefficients, and *integer inequalities*, with non-negative integer coefficients. In the article, the polyhedral study focuses on the binary inequalities, through a new structure to handle their symmetries.

2.3 Related knapsack polytopes

The (SMKP) is a knapsack variant. Indeed, it is a generalization of the (UIKP) [13], as shown in **Theorem 1**. Thus, the facet-defining inequalities of the (UIKP) may not be facet-defining for the (SMKP).

The (SMKP) is neither a generalization, nor a special case of both the (BIKP) and the (BKP). Hence, the facet-defining inequalities of these problems may not be related to those of the (SMKP). However, we introduce some concepts of the (BKP) as they have been extended to variants that are related to the (SMKP).

Let V be a set of items, the (BKP) can be formulated as follows:

$$\begin{aligned} \max \quad & \sum_{j \in V} v_j x_j \\ & \sum_{j \in V} w_j x_j \leq C \\ & x_j \in \{0, 1\} \quad \forall j \in V \end{aligned}$$

A *cover* $U \subseteq V$ is a set of items such that $\sum_{j \in U} w_j > C$. A *minimal cover* $U \subseteq V$ is a cover such that no subset of U is also a cover, $\sum_{j \in V \setminus \{i\}} w_j \leq C, \forall i \in U$. The *minimal cover inequalities* containing

facet-defining inequalities of the (BKP), are as follows, with U a minimal cover:

$$\sum_{j \in U} x_j \leq |U| - 1$$

The minimal cover and related inequalities have been extended to the (PKP), which is a generalization of the (SMKP). The (PKP), defined as follows. Let (V, \preceq) be a partial order set of items. Item j covers item i if there is no k such that $j \prec k \prec i$. A compact formulation of the (PKP) is:

$$\begin{aligned} \max \quad & \sum_{j \in V} v_j x_j \\ & \sum_{j \in V} w_j x_j \leq C \\ & x_j \leq x_i && \text{if } j \text{ covers } i \\ & x_j \in \{0, 1\} && \forall j \in V \end{aligned}$$

A *lower-ideal* is a set $U \subset V$ such that if $i \in U$ and $i \prec j$, then $j \in U$. The minimal cover inequalities have been extended to the case of the (PKP) with U a minimal cover as well as a lower-ideal:

$$\sum_{j \in U} x_j \leq |U| - 1$$

Such inequalities cannot contain the binary facet-defining inequalities of the (SMKP), as most of these inequalities are for variable sets that are not lower-ideals. When referring to the convex hull of incremental formulation of **Example 1**, inequality (i1) is facet-defining for the (SMKP), for the set of variables $\{x_{12}, x_{22}, x_{32}, x_{41}\}$. However, this set is not a lower ideal. In fact, it contains the second variable of groups 1 to 3, without containing the first variable of these groups despite the order constraints. The minimal cover inequalities have been enhanced in the literature with various liftings. In [2] a lifting procedure is described, starting from a facet of a lower-dimensional polyhedron. The lifting procedure provides valid inequalities that are not necessarily facet-defining for the (PKP).

The minimal cover can also be adapted to the (PKP) [12] [11] without the use of lower-ideals. The corresponding inequalities are more likely to be facet-defining inequalities of the (SMKP), which cannot be obtained with lower-ideals as previously mentioned. Let A be the arcs of the precedence graph. The arc set A is modified such that if $(i, j) \in A$ and $(j, k) \in A$ then $(i, k) \in A$. An *induced cover* is a set $U \subseteq V$, such that there is no arc between a pair of items in U and $\sum_{j \in U} w_j + \sum_{(k,j) \in A, j \in V} w_k \geq C$. A *minimal induced cover* is an induced cover U such that $\sum_{j \in U \setminus \{i\}} w_j + \sum_{(k,j) \in A, j \in V} w_k \geq C, \forall i \in U$. The *minimal induced cover inequalities* are as follows, with U a minimal induced cover:

$$\sum_{j \in U} x_j \leq |U| - 1$$

The minimal induced cover inequalities do not contain the facet-defining inequalities of the (SMKP). Indeed, most facet-defining inequalities are for induced covers that are not minimum. When referring to the convex hull of incremental formulation of **Example 1**, inequality (i1) is facet-defining for the set of variables $\{x_{12}, x_{22}, x_{32}, x_{41}\}$. However, this set is not a minimal induced cover, as its subset $\{x_{12}, x_{41}\}$ is a minimal induced cover.

These minimal induced covers have been enhanced with a lifting [12]. The procedure described uses items that have at least two successors in the precedence graph. For the (SMKP) instance $(4, 3, [3, 4, 2], v, 9)$, this lifting only considers variables for the last item of each group. As mentioned previously, variable set $\{x_{12}, x_{22}, x_{32}, x_{41}\}$ is not a minimal induced cover and it also excludes all variables for the last item of each group. Consequently, the inequality corresponding to the variable

set $\{x_{12}, x_{22}, x_{32}, x_{41}\}$ cannot be obtained through this lifting, and the lifting procedure described cannot produce the facet-defining inequalities of the (SMKP). A sequential lifting procedure has also been developed [11]. Consider U a minimal induced cover, with U_p containing all variables $i \notin U$ such that $(i, j) \in A$ and $j \in U$, and with $U_r = V \setminus (U \cup U_p)$. The sequential lifting procedure computes the coefficients $\alpha_j, \forall j \in U_p$ and $\beta_j, \forall j \in U_r$, leading to a lifted inequality:

$$\sum_{j \in U} x_j \sum_{j \in U_p} \alpha_j (1 - x_j) + \sum_{j \in U_r} \beta_j x_j \leq |U| - 1$$

Referring to **Example 1** with the (SMKP) instance $(4, 3, [3, 4, 2], v, 9)$, the procedure described can produce inequalities (i1) to (i4), but not (i5), meaning it could produce the binary inequality of the (SMKP). However, this is not always the case as shown with **Example 2**.

Example 2. Let $(4, 3, [6, 3, 2], v, 18)$ be an instance of the (SMKP). The following inequality is facet-defining for the incremental formulation:

$$x_{11} + x_{13} + x_{22} + x_{32} + x_{42} \leq 3 \tag{i6}$$

The only minimal induced cover with the variables of (i6) are $U_1 = \{x_{11}, x_{22}, x_{32}\}$, $U_2 = \{x_{13}, x_{22}\}$ or any of their symmetry with respect to the groups 2 to 4. The sequential lifting procedure could produce inequality (i6) from neither U_1 nor U_2 with the lifting process. No variable in inequality (i6) is in U_p associated to U_1 , and $|U_1| - 1 = 2$, the bound 3 cannot be obtained with the sequential lifting procedure. Only one variable in (i6) is in U_p associated to U_2 : x_{11} . However, as $|U_2| - 1 = 1$ and x_{11} has coefficient 1, the bound 3 cannot be obtained with the sequential lifting procedure. Hence, no variable set can lead to (i6) with the sequential procedure.

The (SMKP) can also be a special case of the (DKP) formulated as follows. Consider a disjunctive graph with vertex set V and edge set E . A compact multiple choice formulation of the (DKP) is:

$$\begin{aligned} \max \quad & \sum_{j \in V} v_j x_j \\ & \sum_{j \in V} w_j x_j \leq C \\ & x_j + x_i \leq 1 && \text{if } (j, i) \in E \\ & x_j \in \{0, 1\} && \forall j \in V \end{aligned}$$

As for the precedence graph of the (SMKP), the corresponding disjunctive graph is very special. Indeed, the only edges are between any pair of items in the same group. Consequently, every group of items is a clique, and no item is adjacent to an item in another group. Five families of efficient inequalities have been reported for the (DKP) [14]: the clique inequalities; the cover inequalities; odd-cycle and hypergraph inequalities, the clique-cover inequalities; the clique-cover-partition inequalities. The cover inequality for a set $U \subseteq V$ is close to the one of the (PKP): instead of U being a lower-ideal, U is such that there is no edge between two items of U in the disjunctive graph. Such inequalities cannot contain the binary facet-defining inequalities of the (SMKP), as most of these inequalities are for sets of variables with pairs of variables being neighbors in the disjunctive graph. When referring to the convex hulls of the multiple choice formulation for (SMKP) instance $(4, 3, [3, 4, 2], v, 9)$ (see **Example 1**), the set of variables $\{x_{12}, x_{13}, x_{22}, x_{23}, x_{42}, x_{43}, x_{41}, x_{42}, x_{43}\}$ contains multiple items in a same group, hence being neighbors in the disjunctive graph. Yet, this set yields a facet-defining inequality of the (SMKP). Because of this specific disjunctive graphs, the cliques inequalities cannot be adapted to the (SMKP), and similarly for the odd-cycle and hypergraph inequalities. As the clique-cover and clique-cover-partition inequalities rely on the cliques, these inequalities cannot be adapted to the (SMKP) either.

2.4 First polyhedral properties

Definition 1 (Full-dimensional condition (fd)). An (SMKP) verifies (fd) if any item (i, j) , $j < M$ can be selected without item $(i, j + 1)$ in at least one feasible solution, and any item (i, M) can be selected in at least one feasible solution.

Let (N, M, w, v, C) be an instance of the (SMKP) where an item (i, j) cannot be selected in any feasible solution, and $(i, j - 1)$ can be selected in a feasible solution. Clearly, (N, M, w, v, C) does not verify (fd). Because of the symmetric weights, item j cannot be selected in any group, therefore $x_{ij'} = 0$, $\forall i \leq N, \forall j' \geq j$ in any integer solution. It is possible to create another instance of the (SMKP) $(N, j - 1, w, v, C)$ with the exact same integer solutions. Because $(i, j - 1)$ can be selected in a feasible solution, and because of the symmetric weight, any item of $(N, j - 1, w, v, C)$ can be selected. Hence, $(N, j - 1, w, v, C)$ verifies (fd).

As any instance of the (SMKP) can be transformed into an instance of the (SMKP) verifying (fd), in the following we will only consider (SMKP) instances verifying (fd) without loss of generality.

Definition 2 (Solution $X_{\mathcal{X}}$). For a variable set \mathcal{X} , solution $X_{\mathcal{X}}$ is the solution with for any $i \leq N$, $x_{ij} = 1$ if $x_{ij'} \in \mathcal{X}$ and $j \leq j'$ and $x_{ij} = 0$ otherwise.

We define the special case X_{ij} if $\mathcal{X} = \{x_{ij}\}$, and X_{\emptyset} if $\mathcal{X} = \emptyset$.

To ensure that the order constraints are taken into account while considering a solution $X_{\mathcal{X}}$, we introduce the set weights associated to a set of variables \mathcal{X} .

Definition 3 (Set weights). For a given set of variable \mathcal{X} the set weights are

$$s_{ij}(\mathcal{X}) = \begin{cases} 0 & \text{if } x_{ij} \notin \mathcal{X} \\ \sum_{k=j'+1}^j w_k & \text{if } x_{ij} \in \mathcal{X} \text{ with } j' = \max\{j' | x_{ij'} \in \mathcal{X}, j' < j\} \\ \sum_{k=1}^j w_k & \text{if } x_{ij} \in \mathcal{X} \text{ and } x_{ij'} \notin \mathcal{X}, \forall j' < j \end{cases}$$

The coefficient $s_{ij}(\mathcal{X})$ embed the order constraints. Indeed, if $x_{ij} = 1$, $x_{ij} \in \mathcal{X}$, then all $x_{ij'} = 1$, $j' \leq j$, even for $x_{ij'} \notin \mathcal{X}$. Thus, if $x_{ij} = 1$ then the weights of all variable $x_{ij'} \notin \mathcal{X}$ should be accounted for, which is the purpose of coefficients $s_{ij}(\mathcal{X})$.

For the sake of simplicity, we define \mathcal{Y} as a k -intersection of \mathcal{X} if \mathcal{Y} contains k elements of \mathcal{X} .

Definition 4 (k -intersection). Let \mathcal{X}, \mathcal{Y} be sets of variables. Variable set \mathcal{Y} is a k -intersection of \mathcal{X} if $|\mathcal{Y} \cap \mathcal{X}| = k$, $\forall x_{ij} \in \mathcal{Y}$ if $x_{ij'} \in \mathcal{X}$ with $j' \leq j$, then $x_{ij'} \in \mathcal{Y}$ and $\sum_{x_{ij} \in \mathcal{Y}} s_{ij}(\mathcal{Y}) \leq C$.

If \mathcal{Y} is k -intersection of \mathcal{X} , the reverse can also be true. Consequently, a k -intersection is not necessarily a subset.

Example 3. Let $(3, 3, [1, 3, 2], v, 12)$ be an instance of the (SMKP). Let two variable sets $\mathcal{X} = \{x_{11}, x_{13}, x_{22}, x_{31}\}$ and $\mathcal{Y} = \{x_{11}, x_{13}, x_{21}, x_{22}\}$. In this case, \mathcal{Y} is a 3-intersection of \mathcal{X} . Indeed, $|\mathcal{X} \cap \mathcal{Y}| = 3$, $s_{11}(\mathcal{X}) + s_{13}(\mathcal{X}) + s_{22}(\mathcal{X}) + s_{31}(\mathcal{X}) = 1 + 5 + 4 + 1 = 11 \leq C = 12$, and $x_{12} \notin \mathcal{X}$. However, \mathcal{X} is not a 3-intersection of \mathcal{Y} . Indeed, $|\mathcal{X} \cap \mathcal{Y}| = 3$ and $s_{11}(\mathcal{Y}) + s_{13}(\mathcal{Y}) + s_{21}(\mathcal{Y}) + s_{22}(\mathcal{Y}) = 1 + 5 + 1 + 3 = 10 \leq C = 12$, but there is $x_{21} \in \mathcal{Y}$ and $x_{21} \notin \mathcal{X}$ even if $x_{22} \in \mathcal{X}$.

Theorem 2. $P_{(SMKP)}$ is full dimensional.

Proof. As there are N items and M groups, solutions X_{ij} and X_{\emptyset} yield a total of $N \times M + 1 = n + 1$ different solutions, which are feasible, otherwise (fd) is not verified. For any i , solution X_{iM} is the only solution with $x_{iM} = 1$, thus being affinely independent to other solutions. For any i , solution X_{ij} , $j < M$ is the only solution with $x_{ij} = 1$ and $x_{ij+1} = 0$, this being affinely independent to other solutions. Clearly, X_{\emptyset} is affinely independent to other solutions, which means that there are $n + 1$ affinely independent solutions. \square

With the dimensionality of $P_{(SMKP)}$ it becomes possible to characterize when order inequalities and trivial inequalities with bounds at 0 are facet-defining.

Property 2. Inequalities $x_{iM} \geq 0$ and $x_{ij} \leq x_{ij-1}$ are facet-defining for $P_{(SMKP)}$.

Proof. As there are N items and M groups, solutions X_{ij} and X_\emptyset yield a total of $N \times M + 1 = n + 1$ different solutions, which are feasible otherwise (fd) is not verified.

For a given i , besides solution X_{iM} , every of the other n solutions verify $x_{iM} \geq 0$ to equality, and are proven affinely independent. Inequalities $x_{iM} \geq 0$ are then facet-defining for $P_{(SMKP)}$.

Similarly, besides X_{ij-1} , each of the n solutions verifies $x_{ij} \leq x_{ij-1}$ to equality, and is proven to be affinely independent. Inequalities $x_{ij} \leq x_{ij-1}$ are then facet-defining for $P_{(SMKP)}$. \square

Contrary to trivial inequalities with bound at 0 and the order constraints which always contain facet-defining inequalities, the bounds 1 require a minimum capacity C to contain facet-defining inequalities.

Property 3. Inequalities $x_{i1} \leq 1$ are facet-defining for $P_{(SMKP)}$ if $C \geq \sum_{j=1}^M w_j + w_1$.

Proof. Consider a group i' . For each item (i, j) , consider a solution X'_{ij} , similar to X_{ij} , with $x_{i'1} = 1$ if $i' \neq i$. As there are N items and M groups, solutions X'_{ij} yields a total of $N \times M = n$ different solutions, which are feasible if $C \geq \sum_{j=1}^M w_j + w_1$ and (fd) are verified. Each of the n solutions verifies $x_{i'1} \leq 1$ to equality, and is proven to be affinely independent in the same manner as solutions X_{ij} . Inequalities $x_{i'1} \leq 1$ are then facet-defining for $P_{(SMKP)}$ if $C \geq \sum_{j=1}^M w_j + w_1$.

In the case where $C < \sum_{j=1}^M w_j + w_1$, clearly $x_{iM} + x_{i'1} \leq 1$ is valid, for any $i \leq N$, $i' \leq N$ and $i \neq i'$ because of the symmetric weights. In which case inequality $x_{iM} \leq 1$ is dominated and cannot be facet-defining in this case.

Hence, inequalities $x_{iM} \leq 1$ are facet-defining if and only if $C \geq \sum_{j=1}^M w_j + w_1$. \square

3 Patterns inequalities

In this section we introduce new inequalities. We are interested in the faces defined by these inequalities, i.e., the set of points of the polytope $P_{(SMKP)}$ verifying these inequalities to equality. To handle the symmetries of the inequalities without the need to explicitly define all of them, we introduce a new structure called pattern.

3.1 Definitions

Definition 5 (Pattern). A *pattern* \mathcal{P} is a collection of N sets $S_i(\mathcal{P}) \subseteq \{1, \dots, M\}$, $i \leq N$.

A set $S_i(\mathcal{P})$ contains the indices j of the items in a same group. The sets of a pattern are not ordered, meaning that a pattern represents any permutation of an item set of the (SMKP).

As the aim is to produce inequalities from the patterns, we define the variable sets associated to a pattern.

Definition 6 (Variable set \mathcal{X} associated to \mathcal{P}). A variable set $\mathcal{X} \subseteq \mathcal{V}$ is associated to pattern \mathcal{P} and a permutation π of $\{1, \dots, N\}$ if:

$$x_{ij} \in \mathcal{X} \Leftrightarrow j \in S_{\pi(i)}(\mathcal{P}).$$

We denote $\chi(\mathcal{P})$ the set of all variable sets associated to \mathcal{P} . Note that $|\chi(\mathcal{P})|$ is in general exponential.

The cardinality of a pattern \mathcal{P} is the cardinality of any variable set associated to \mathcal{P} .

Definition 7 ($card(\mathcal{P})$). The cardinality of a pattern \mathcal{P} is $card(\mathcal{P}) = |\mathcal{X}|$ with $\mathcal{X} \in \chi(\mathcal{P})$.

The rank of a pattern \mathcal{P} is the valid upper bound for the sum of variables in any variable set associated to \mathcal{P} .

Definition 8 ($rank(\mathcal{P})$). The rank of a pattern \mathcal{P} is

$$rank(\mathcal{P}) = \max_{\mathcal{X} \in \chi(\mathcal{P})} \left\{ \max_{x_{ij} \in \mathcal{X}} \sum x_{ij} : \text{satisfying (2) - (4)} \right\}$$

The rank of a pattern can be computed with a shortest path algorithm [1] as described in **Section 4**.

With $rank(\mathcal{P})$ and $\chi(\mathcal{P})$, we can define the inequalities of a pattern \mathcal{P} as follows.

Definition 9 (Pattern inequalities). The *pattern inequalities* associated to a pattern \mathcal{P} are the following, for any $\mathcal{X} \in \chi(\mathcal{P})$:

$$\sum_{x_{ij} \in \mathcal{X}} x_{ij} \leq rank(\mathcal{P}) \quad (pi(\mathcal{X}))$$

By definition of the rank, and because the weights are symmetric, the pattern inequalities are valid. As for any set of variables there is a pattern, and vice-versa, these pattern inequalities cover all the binary inequalities of the (SMKP). Also, because $|\chi(\mathcal{P})|$ can be exponential, each pattern is associated up to an exponential number of pattern inequalities. As the number of patterns of an (SMKP) is also exponential, we need to define the conditions for a pattern to lead to tight pattern inequalities.

Definition 10 (Pattern-facet). A pattern \mathcal{P} is a *pattern-facet* if $\forall \mathcal{X} \in \chi(\mathcal{P})$, $(pi(\mathcal{X}))$ is facet-defining.

3.2 Necessary facet defining conditions

In this section we define three necessary conditions for a pattern to be pattern-facet. The first one is for a pattern to have at least one item in each of its sets.

Property 4 (Condition (i): no empty group). *If a pattern \mathcal{P} is pattern-facet, then \mathcal{P} verifies condition (i):*

$$\forall S_i(\mathcal{P}) \in \mathcal{P}:$$

$$|S_i(\mathcal{P})| \geq 1 \quad (i)$$

Proof. Let \mathcal{P} be a pattern of rank k . Let $\mathcal{X} \in \chi(\mathcal{P})$ with permutation π_{id} . Suppose $S_i(\mathcal{P})$ does not verify (i) for a given i , i.e., $S_i(\mathcal{P}) = \emptyset$. Let $\mathcal{X}' = \mathcal{X} \cup \{x_{iM}\}$. Because x_{iM} is the only variable of \mathcal{X}' for group i , $s_{iM}(\mathcal{X}') \geq s_{i'j'}(\mathcal{X}')$, $\forall x_{i'j'} \in \mathcal{X}'$. Then, the following inequality is valid:

$$\sum_{x_{i'j'} \in \mathcal{X}'} x_{i'j'} + x_{iM} \leq k$$

Indeed, when $x_{iM} = 0$, this inequality is valid by the rank of \mathcal{P} . When $x_{iM} = 1$, there cannot be k variables of \mathcal{X}' to 1. Otherwise as $s_{iM}(\mathcal{X}') \geq s_{i'j'}(\mathcal{X}')$, $\forall x_{i'j'} \in \mathcal{X}'$, one could set x_{iM} to 0, and any other variable of \mathcal{X}' to 1. This would reduce the total weight, leading to another solution. Such solution would have $k + 1$ variables of \mathcal{X} to 1, which contradicts the rank of \mathcal{P} . Therefore this inequality is valid.

This inequality dominates $(pi(\mathcal{X}))$. Indeed, one could sum it with $-x_{iM} \leq 0$ to obtain $(pi(\mathcal{X}))$.

Thus a pattern \mathcal{P} is pattern-facet only if \mathcal{P} verifies condition (i). \square

The idea of the following condition is that for any $\mathcal{X} \in \chi(\mathcal{P})$, there is a feasible solution with $(pi(\mathcal{X}))$ to equality, and $x_{iM} = 1$ for any group i .

Property 5 (Condition (ii): selection of item M). *Let k be the rank of \mathcal{P} and $\mathcal{X} \in \chi(\mathcal{P})$. If a pattern \mathcal{P} is pattern-facet, then \mathcal{P} verifies condition (ii) :*

$$\forall i \leq N, \exists \mathcal{Y} \subseteq \mathcal{V} \text{ a } k\text{-intersection of } \mathcal{X} \text{ with } x_{iM} \in \mathcal{Y}.$$

Proof. Let \mathcal{P} be a pattern of rank k , and $\mathcal{X} \in \chi(\mathcal{P})$. Suppose there is an $i \leq N$ such that (ii) is not verified for i . This means that there is no feasible solution with k variables of \mathcal{X} to 1, with $x_{iM} = 1$. Therefore, the following inequality is valid:

$$\sum_{x_{i'j'} \in \mathcal{X}} x_{i'j'} + x_{iM} \leq k$$

Indeed, when $x_{iM} = 0$, the inequality is valid by the rank of \mathcal{P} . When $x_{iM} = 1$, the inequality is valid as there cannot be more than $k - 1$ variables of \mathcal{X} to 1, which sums to a total of at most k .

This inequality dominates the inequality $(pi(\mathcal{X}))$. Indeed, one could sum it with $-x_{iM} \leq 0$ to obtain $(pi(\mathcal{X}))$.

Thus, a pattern \mathcal{P} is pattern-facet only if \mathcal{P} verifies condition (ii). □

The following condition is quite similar to condition (ii), but for any variable x_{ij-1} with $x_{ij} \in \mathcal{X}$, instead of any variable x_{iM} .

Property 6 (Condition (iii): independence of an item from its predecessor). *Let k be the rank of \mathcal{P} and $\mathcal{X} \in \chi(\mathcal{P})$. If a pattern \mathcal{P} is pattern-facet, then \mathcal{P} verifies condition (iii) :*

$$\forall x_{ij} \in \mathcal{X}, \exists \mathcal{Y} \subseteq \mathcal{V} \text{ a } k\text{-intersection of } \mathcal{X} \text{ with } x_{ij-1} \in \mathcal{Y} \text{ and } x_{ij'} \notin \mathcal{Y} \forall j' \geq j.$$

Proof. Let \mathcal{P} be a pattern of rank k . Let $\mathcal{X} \in \chi(\mathcal{P})$. Suppose for some (i, j) , $x_{ij} \in \mathcal{X}$ does not verify condition (iii). This means that there is no feasible solutions with a total of k variables of \mathcal{X} to 1, with $x_{ij-1} = 1$ and $x_{ij} = 0$. Therefore, the following inequality is valid:

$$\sum_{x_{i'j'} \in \mathcal{X}} x_{i'j'} + x_{ij-1} - x_{ij} \leq k$$

Indeed, when $x_{ij} = x_{ij-1} = 1$ or $x_{ij} = x_{ij-1} = 0$, this inequality is valid by the rank of \mathcal{P} . When $x_{ij-1} = 1$ and $x_{ij} = 0$, the inequality is valid as there cannot be more than $k - 1$ variables of \mathcal{X} to 1, which sums to at most k .

This inequality dominates the inequality $(pi(\mathcal{X}))$. Indeed, one could sum it with $-x_{ij-1} + x_{ij} \leq 0$ (equivalent to $x_{ij} \leq x_{ij-1}$) to obtain $(pi(\mathcal{X}))$.

Thus a pattern \mathcal{P} is pattern-facet only if \mathcal{P} verifies condition (iii). □

For a given pattern \mathcal{P} , conditions (i) can clearly be verified in linear time. Also, conditions (ii) and (iii) can be verified in polynomial time. More precisely, it requires to solve the shortest path algorithm described in **Section 4** at most once for each variable. As these conditions are necessary, we define a flexible pattern, which verifies all of these three conditions.

Definition 11 (Flexible pattern). A pattern \mathcal{P} is a flexible pattern if it verifies conditions (i), (ii) and (iii).

The conditions on a flexible pattern \mathcal{P} are not sufficient for \mathcal{P} to be pattern-facet. However, a minimum dimension can be guaranteed for the flexible patterns inequalities.

3.3 Lower bound on the dimension of the flexible patterns inequalities

In this section, we consider a flexible pattern \mathcal{P} and variable set $\mathcal{X} \in \chi(\mathcal{P})$. The idea of the following property is that for any x_{ij} there is a valid solution with $x_{ij} = 1$, $x_{ij+1} = 0$ and $(pi(\mathcal{X}))$ to equality. In a sense it is a generalization of condition (ii) defined only for variables x_{iM} and condition (iii) defined only for variables x_{ij} such that $x_{ij+1} \in \mathcal{X}$.

Property 7 (Generalization of (ii) and (iii) for any item of the (SMKP)). *Let \mathcal{P} be a flexible pattern and $\mathcal{X} \in \chi(\mathcal{P})$. For any item (i, j) , $\exists \mathcal{Y} \subseteq \mathcal{V}$ a k -intersection of \mathcal{X} with $x_{ij} \in \mathcal{Y}$ and $x_{ij'} \notin \mathcal{Y}$, $\forall j' > j$.*

The complete proof is in **A.1**, as it merely extends the proofs for conditions (ii) and (iii).

Conditions (i), (ii) and (iii) are necessary for a pattern \mathcal{P} to be pattern-facet. Moreover, with $\mathcal{X} \in \chi(\mathcal{P})$, the following theorem provides a lower bound on the number of linearly independent points verifying inequalities $(pi(\mathcal{X}))$ to equality when these three conditions are verified.

Theorem 3 ($n - \text{card}(\mathcal{P})$ linearly independent points). *Let \mathcal{P} be a flexible pattern. Let n be the number of variables of the (SMKP). There are at least $n - \text{card}(\mathcal{P})$ linearly independent points that verify the inequalities of \mathcal{P} to equality.*

Proof. Let \mathcal{P} be a pattern of rank k , and \mathcal{X} a variable set of \mathcal{P} . **Property 7** stipulates that if \mathcal{P} is a flexible pattern, then for any item (i, j) there is a set $\mathcal{Y}_{ij} \in \mathcal{V}$ a k -intersection of \mathcal{X} ; $x_{ij} \in \mathcal{Y}_{ij}$; $x_{ij'} \notin \mathcal{Y}_{ij}$, $\forall j' > j$ and $X_{\mathcal{Y}_{ij}}$ is feasible. Consider $X_{\mathcal{Y}_{ij}} \forall x_{ij} \notin \mathcal{X}$. Because $\text{card}(\mathcal{P}) = |\mathcal{X}|$, there are $n - \text{card}(\mathcal{P})$ solutions. We can prove that $X_{\mathcal{Y}_{ij}}$ is the only solution with $x_{ij} = 1$ and $x_{ij+1} = 0$. Let two distinct variables $x_{ij} \notin \mathcal{X}$ and $x_{i'j'} \notin \mathcal{X}$. We assumed without loss of generality in **Property 7** that $\mathcal{Y}_{i'j'} \setminus \{x_{i'j'}\} \subseteq \mathcal{X}$. As $x_{ij} \notin \mathcal{X}$, there would be a contradiction if $X_{\mathcal{Y}_{i'j'}}$ had $x_{ij} = 1$ and $x_{ij+1} = 0$.

Solutions $X_{\mathcal{Y}_{ij}} \forall x_{ij} \notin \mathcal{X}$ are linearly independent, and proven to be valid in **Property 7**. As \mathcal{Y}_{ij} is a k -intersection of \mathcal{X} , all these solutions also verify $(pi(\mathcal{X}))$ to equality. Hence, for a flexible pattern \mathcal{P} , there are $n - \text{card}(\mathcal{P})$ linearly independent points verifying $(pi(\mathcal{X}))$ to equality, $\forall \mathcal{X} \in \mathcal{P}$. \square

Theorem 3 provides a lower bound on the dimensions of the faces defined by flexible pattern inequalities. Recall that for a pattern to be a flexible pattern it solely requires to verify conditions (i), (ii) and (iii). It is shown in **Section 4** that verifying if these three conditions hold for a given pattern can be done in polynomial time, and **Theorem 3** is used in the experimental results in **Section 5**.

The following section provides properties complementary to **Theorem 3**.

3.4 Properties of the lower sub-patterns

In this section, we consider a flexible pattern \mathcal{P} of rank k and variable set $\mathcal{X} \in \chi(\mathcal{P})$. We extend the definition of a sub-set and a super-set to patterns.

Definition 12 (Sub-pattern). A pattern \mathcal{P}' is *sub-pattern* of \mathcal{P} if $\exists \pi$ a permutation such that $S_{\pi(i)}(\mathcal{P}') \subseteq S_i(\mathcal{P})$, $\forall i \leq N$.

If \mathcal{P} is sub-pattern of \mathcal{P}' , then \mathcal{P}' is *super-pattern* of \mathcal{P} .

We present a new set of sub-patterns for \mathcal{P} . In the following, properties are presented to show that these new patterns have large sub-patterns in common, inducing similarities. These similarities will be convenient to provide linearly independent points in the polytope $P_{(SMKP)}$.

Definition 13 (Lower sub-patterns \mathcal{Q}_i). For a given i , the *lower sub-pattern* \mathcal{Q}_i of \mathcal{P} is such that $S_i(\mathcal{Q}_i) = S_i(\mathcal{P})$, $\text{card}(\mathcal{Q}_i) = k$, minimizing the sum of the set weights $s(\mathcal{Y})$ with $\mathcal{Y} \in \chi(\mathcal{Q}_i)$.

Let \mathcal{Q}_i be a lower sub-pattern of \mathcal{P} and $\mathcal{Y} \in \chi(\mathcal{Q}_i)$ for the permutation π as \mathcal{X} . By construction, if $x_{ij} \in \mathcal{Y}$, then $\forall x_{ij'} \in \mathcal{X}$ with $j' \leq j$, $x_{ij'} \in \mathcal{Y}$, hence the name lower sub-patterns. Also, patterns \mathcal{Q}_i can be obtained via a shortest path algorithm defined in **Section 4**.

Lemma 1. \mathcal{Y} is a k -intersection of \mathcal{X} .

Proof. Let $\mathcal{X} \in \chi(\mathcal{P})$ and $\mathcal{Y} \in \chi(\mathcal{Q}_i)$ with the permutation π_{id} . For the sake of simplicity we consider permutation π_{id} but the proof can be done for any permutation π .

As \mathcal{P} verifies condition (ii), there exists a variable set that is a k -intersection of \mathcal{X} containing x_{iM} . By definition \mathcal{Q}_i minimizes the sum of its set weights and all variables of group i in \mathcal{X} are in \mathcal{Y} . Hence, if condition (ii) for variable x_{iM} cannot be verified with \mathcal{Y} , there is a contradiction as \mathcal{Q}_i cannot minimize the sum of its set weights. \square

Remark 1. Let $\mathcal{X} \in \chi(\mathcal{P})$ and $\mathcal{Y} \in \chi(\mathcal{Q}_i)$ with permutation π_{id} . As $\mathcal{Y} \cup \{x_{iM}\}$ is a k -intersection of \mathcal{X} , then \mathcal{Y} is also a k -intersection of \mathcal{X} . Indeed, if $x_{iM} \in \mathcal{X}$ then $x_{iM} \in \mathcal{Y}$ by definition of \mathcal{Q}_i .

For the next properties, we consider for the sake of simplicity that the sets of \mathcal{P} are ordered such that $|S_i(\mathcal{P})| \leq |S_{i+1}(\mathcal{P})|$. Therefore, $S_1(\mathcal{P})$ is the smallest set of \mathcal{P} , and \mathcal{Q}_1 is the lower sub-pattern of \mathcal{P} associated to $S_1(\mathcal{P})$. We define $U = |S_N(\mathcal{P})|$ and $V = |S_1(\mathcal{P})|$. Also, we define $S_i(\mathcal{P})(u)$ the u^{th} lowest index of $S_i(\mathcal{P})$ and $S_i(\mathcal{P})[u]$ the u^{th} highest index of $S_i(\mathcal{P})$.

Example 4. Let a pattern $\mathcal{P} = \{S_1 = \{4\}, S_2 = \{2, 5\}, S_3 = \{1, 2\}, S_4 = \{1, 3, 4\}\}$. In this case, $U = 3$ and $V = 1$. Also, $S_4(\mathcal{P})(1) = 1$, $S_4(\mathcal{P})(2) = 3$, $S_4(\mathcal{P})(3) = 4$, $S_4(\mathcal{P})[1] = 4$, $S_4(\mathcal{P})[2] = 3$ and $S_4(\mathcal{P})[3] = 1$.

The following property provides a lower bound on the size of each set of sub-pattern \mathcal{Q}_1 of \mathcal{P} .

Property 8 (Minimum size on the sets of \mathcal{Q}_1). *Lower sub-pattern \mathcal{Q}_1 is such that $\forall i \leq N$, $S_i(\mathcal{Q}_1)$ contains the $|S_i(\mathcal{P})| - V$ smallest indices of $S_i(\mathcal{P})$.*

Remark 2. It is equivalent to say that $S_i(\mathcal{Q}_1)$ contains the $|S_i(\mathcal{P})| - V$ smallest indices of $S_i(\mathcal{P})$ and $S_i(\mathcal{P})[V + 1] \in S_i(\mathcal{Q}_1)$. In the following, the latter notation will be used.

The idea of the proof is illustrated by **Example 5** and **Figure 3**. The complete proof is in **A.2**

Example 5. Let $(4, 5, w, v, C)$ be an instance of the (SMKP). Let a flexible pattern $\mathcal{P} = \{S_1(\mathcal{P}) = \{3\}, S_2(\mathcal{P}) = \{2, 4\}, S_3(\mathcal{P}) = \{1, 2\}, S_4(\mathcal{P}) = \{1, 2, 3\}\}$ of rank 5. Let $\mathcal{Q}_1 = \{S_1(\mathcal{Q}_1) = \{3\}, S_2(\mathcal{Q}_1) = \{2\}, S_3(\mathcal{Q}_1) = \{1, 2\}, S_4(\mathcal{Q}_1) = \{1\}\}$. In this case, $S_4(\mathcal{P})[V + 1] = 2 \notin S_4(\mathcal{Q}_1)$, hence \mathcal{Q}_1 does not verify **Property 8**. We show in the following that it leads to a contradiction with the rank of \mathcal{P} .

Let $\mathcal{X} = \{x_{13}, x_{22}, x_{24}, x_{31}, x_{32}, x_{41}, x_{42}, x_{43}\} \in \chi(\mathcal{P})$ and $\mathcal{Y} = \{x_{13}, x_{22}, x_{31}, x_{32}, x_{41}\} \in \chi(\mathcal{Q}_1)$ as illustrated in **Figure 3a** and **3b**. Let $\mathcal{Y}' = \mathcal{Y} \cup \{x_{14}\}$ as illustrated in **Figure 3c**. From **Lemma 1** solution $X_{\mathcal{Y}'}$ is valid. One can create a set $\mathcal{Y}'' = \mathcal{Y}' \setminus \{x_{13}, x_{14}\} \cup \{x_{42}, x_{43}\}$ as illustrated in **Figure 3d**. By removing $\{x_{13}, x_{14}\}$, there are no remaining variables in group 1, thus reducing the weight by $w_1 + w_2 + w_3 + w_4$. And as $x_{41} \in \mathcal{Y}'$, adding $\{x_{42}, x_{43}\}$ only increases the weight by $w_2 + w_3$. As the weights are non-negative, clearly $w_2 + w_3 \leq w_1 + w_2 + w_3 + w_4$, hence $X_{\mathcal{Y}''}$ is valid. However \mathcal{Y}'' is a $k + 1$ -intersection of \mathcal{X} . Indeed, since \mathcal{Q}_1 does not verify **Property 8**, only $V = 1$ variable of $\mathcal{Y}' \setminus \mathcal{Y}''$ is in \mathcal{X} , namely x_{13} and $V + 1 = 2$ variables of $\mathcal{Y}'' \setminus \mathcal{Y}'$ are in \mathcal{X} , namely x_{42} and x_{43} . As $X_{\mathcal{Y}''}$ is valid, there is a contradiction with the rank of \mathcal{P} .

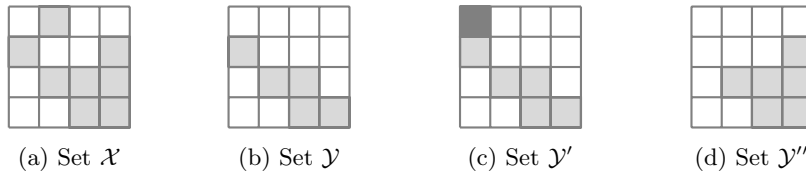


Figure 3: Illustration of **Example 5**

Property 9 provide dependencies between the indices of $S_1(\mathcal{P})$ and any set $S_i(\mathcal{P})$, based on **Property 8**.

Property 9 (Minimum indices for the smallest set). *Let $i \in \{2, \dots, N\}$. $\forall u \in [1, V]$, if $|S_i(\mathcal{P})| \geq u + V$ then $S_1(\mathcal{P})[u] \geq S_i(\mathcal{P})[u + V]$.*

Proof. The proof is divided in two possible cases, each being supported by a lemma. In the case $S_i(\mathcal{Q}_1) \subseteq S_i(\mathcal{P})$ with $S_i(\mathcal{P})[u] \in S_i(\mathcal{Q}_1)$, the proof is provided by **Lemma 2**. In the case $S_i(\mathcal{Q}_1) \subset S_i(\mathcal{P})$ with $S_i(\mathcal{P})[u] \notin S_i(\mathcal{Q}_1)$, the proof is provided by **Lemma 3**. Hence, the property is always verified. The two lemmas are in **A.3**. \square

The idea of **Property 10** is to give a minimum size on the sets of any lower sub-pattern \mathcal{Q}_i based on the results of **Property 9**. Note that, it is not a generalization of **Property 8**. Indeed, **Property 8** addresses only \mathcal{Q}_1 and not any \mathcal{Q}_i , but the minimum size provided is larger than the one in **Property 10**.

Property 10 (Minimum size of the sets of any \mathcal{Q}_i). *For every $i \leq N$, the sub-pattern \mathcal{Q}_i of \mathcal{P} is such that $\forall i' \leq N$ if $|S_{i'}(\mathcal{P})| \geq 2V$, $S_{i'}(\mathcal{P})[2V] \in S_{i'}(\mathcal{Q}_i)$*

Proof. The proof is divided in three possible cases, each being supported by a lemma. In the case $|S_1(\mathcal{Q}_1)| + |S_i(\mathcal{Q}_1)| = |S_1(\mathcal{Q}_i)| + |S_i(\mathcal{Q}_i)|$, the proof is provided by **Lemma 4**. In the case $|S_1(\mathcal{Q}_i)| = 0$, the proof is provided by **Lemma 5**. In the case $|S_1(\mathcal{Q}_i)| > 0$, the proof is provided by **Lemma 6**. Hence, the property is always verified. The three lemmas are in **A.4**. \square

The previous conditions are valid for any pattern. However, in the special case where pattern \mathcal{P} is with $|S_1(\mathcal{P})| = 1$, **Property 10** indicates that $S_{i'}(\mathcal{P})[2] \in S_{i'}(\mathcal{Q}_i)$, $\forall i' \leq N$, $\forall i \leq N$. Thus, the shape of all patterns \mathcal{Q}_i is very restricted, as each set $S_{i'}(\mathcal{Q}_i)$ has at most one missing index set in comparison to $S_{i'}(\mathcal{P})$.

Example 6. Let a flexible pattern $\mathcal{P} = \{S_1(\mathcal{P}) = \{3\}, S_2(\mathcal{P}) = \{1, 3, 4\}, S_3(\mathcal{P}) = \{1, 2, 4\}, S_4(\mathcal{P}) = \{1, 2, 3, 4\}\}$. Pattern \mathcal{P} verifies **Property 9**, as $S_1(\mathcal{P})[1] = 3$ is greater or equal to $S_2(\mathcal{P})[2] = 3$, $S_3(\mathcal{P})[2] = 2$ and $S_4(\mathcal{P})[2] = 3$. Consider $\text{rank}(\mathcal{P}) = 9$.

Let the lower sub-pattern $\mathcal{Q}_1 = \{S_1(\mathcal{Q}_1) = \{3\}, S_2(\mathcal{Q}_1) = \{1, 3, 4\}, S_3(\mathcal{Q}_1) = \{1, 2\}, S_4(\mathcal{Q}_1) = \{1, 2, 3\}\}$. Pattern \mathcal{Q}_1 verifies **Property 8**, as $V = 1$ and there is at most $V = 1$ missing index per set compared to \mathcal{P} .

Let the lower sub-pattern $\mathcal{Q}_2 = \{S_1(\mathcal{Q}_2) = \emptyset, S_2(\mathcal{Q}_2) = \{1, 3, 4\}, S_3(\mathcal{Q}_2) = \{1, 2, 3\}, S_4(\mathcal{Q}_2) = \{1, 2, 3\}\}$. Pattern \mathcal{Q}_2 verifies **Property 10**, as $V = 1$ and there are at most $2V - 1 = 1$ missing index per set compared to \mathcal{P} .

This restricted shape on the lower sub-pattern is used in the following section to prove necessary and sufficient conditions for patterns containing a set of cardinality 1.

3.5 Necessary and sufficient conditions for patterns with a set a cardinality 1

In this section, we focus on patterns with at least one set of cardinality 1, hence we define for this section \mathcal{P} a flexible pattern of rank k and with $V = 1$. It is proven with **Property 10** that for such pattern \mathcal{P} , its lower sub-patterns \mathcal{Q}_i have a restricted shape. Using this result, we will show that the lower sub-patterns share many elements in common.

As mentioned in **Lemma 4**, for a given i , multiple lower sub-patterns \mathcal{Q}_i with the exact same set weights can exist. From now on we only consider for each $i \leq N$ the unique \mathcal{Q}_i verifying the following tie-break rule:

Definition 14 (Tie-break rule). For any two indices $i' < i''$ different than i , if

$$\sum_{j=S_{i'}(\mathcal{P})[2]+1}^{S_{i'}(\mathcal{P})[1]} w_j = \sum_{j=S_{i''}(\mathcal{P})[2]+1}^{S_{i''}(\mathcal{P})[1]} w_j$$

then $S_{i''}(\mathcal{P})[1] \in S_{i''}(\mathcal{Q}_i)$ only if $S_{i'}(\mathcal{P})[1] \in S_{i'}(\mathcal{Q}_i)$.

From **Property 10**, at most one index is missing in a set of \mathcal{Q}_i compared to \mathcal{P} . Hence, with such rule there can only be one \mathcal{Q}_i for a given i .

Example 7. Let $(4, 4, [2, 1, 1, 1], v, C)$ be an instance of the (SMKP). Let a flexible pattern $\mathcal{P} = \{S_1(\mathcal{P}) = \{3\}, S_2(\mathcal{P}) = \{1, 3\}, S_3(\mathcal{P}) = \{2, 4\}, S_4(\mathcal{P}) = \{2, 4\}\}$ of rank 5. In this case, $w_2 + w_3 = w_3 + w_4 = 2$, meaning that there are 3 possible lower sub-pattern \mathcal{Q}_1 with the exact same weight: $\{\{3\}, \{1, 3\}, \{2\}, \{2\}\}; \{\{3\}, \{1\}, \{2, 4\}, \{2\}\}; \{\{3\}, \{1\}, \{2\}, \{2, 4\}\}$. The rule stipulates that $4 \in S_4(\mathcal{Q}_1)$ only if $4 \in S_3(\mathcal{Q}_1)$ and $3 \in S_2(\mathcal{Q}_1)$. Also, the rule stipulates that $4 \in S_3(\mathcal{Q}_1)$ only if $3 \in S_2(\mathcal{Q}_1)$. Only the first option for \mathcal{Q}_1 verifies the rule, and consequently is the only one considered.

To prove that the lower sub-patterns \mathcal{Q}_i share many elements, we provide a pattern \mathcal{C} , sub-pattern to all \mathcal{Q}_i . We then prove that \mathcal{C} is of cardinality $k - 1$.

Definition 15 (Common sub-pattern \mathcal{C} of all \mathcal{Q}_i). The pattern \mathcal{C} is the largest cardinality pattern that is sub-pattern to all the lower sub-patterns \mathcal{Q}_i of a pattern

Property 11 (Cardinality of \mathcal{C}). *The cardinality of \mathcal{C} is $\text{card}(\mathcal{C}) = k - 1$.*

An example of **Property 11** is provided in **Example 8**.

Proof. As \mathcal{P} is a flexible pattern, **Property 10** holds, hence $\forall i, \forall i', S_{i'}(\mathcal{P})[2] \in S_{i'}(\mathcal{Q}_i)$. In other words, there is at most one element of $S_{i'}(\mathcal{P})$ not in $S_{i'}(\mathcal{Q}_i)$. Let $k' = k - (\text{card}(\mathcal{P}) - N)$. Let \mathcal{L} be the set of the k' indices i' such that $S_{i'}(\mathcal{Q}_i) = S_{i'}(\mathcal{P})$. By definition of \mathcal{Q}_i , $i \in \mathcal{L}$. Because $S_{i'}(\mathcal{P})[2] \in S_{i'}(\mathcal{Q}_i)$, $\forall i'$, the $k' - 1$ sets $i' \neq i$ such that $S_{i'}(\mathcal{Q}_i) = S_{i'}(\mathcal{P})$ are the ones minimizing:

$$\sum_{j=S_{i'}(\mathcal{P})[2]+1}^{S_{i'}(\mathcal{P})[1]} w_j$$

Consequently, \mathcal{L} contains the $k' - 1$ indices minimizing this sum. Indeed, either i is in these $k' - 1$ indices, hence \mathcal{L} contains the k' indices minimizing this sum, or i is not in these $k' - 1$ indices, but by construction \mathcal{L} contains these $k' - 1$ indices.

As such, all patterns \mathcal{Q}_i are all super-pattern of common pattern \mathcal{C} , with $\text{card}(\mathcal{C}) = (\text{card}(\mathcal{P}) - N) + k' - 1 = k - 1$. \square

Example 8. Let $(3, 4, [2, 1, 3, 2], v, C)$ be an instance of the (SMKP). Let a flexible pattern $\mathcal{P} = \{S_1(\mathcal{P}) = \{3\}, S_2(\mathcal{P}) = \{1, 2\}, S_3(\mathcal{P}) = \{1, 3\}, S_4(\mathcal{P}) = \{1, 3, 4\}\}$ of rank 6. First we identify the lower sub-pattern \mathcal{Q}_1 . By definition, $S_1(\mathcal{Q}_1) = S_1(\mathcal{P}) = \{3\}$. From **Property 10**, as $V = 1$, $\forall i \leq 4$, $S_i(\mathcal{P})[2] \in S_i(\mathcal{Q}_1)$. In this case, $1 \in S_2(\mathcal{Q}_1)$, $1 \in S_3(\mathcal{Q}_1)$ and $1, 3 \in S_4(\mathcal{Q}_1)$. By definition, $\text{card}(\mathcal{Q}_1) = 6$, but only five elements have been identified yet. As $w_2 = 1 < w_4 = 2 < w_2 + w_3 = 4$ and \mathcal{Q}_1 minimize the sum of the set weights of $\mathcal{Y} \in \chi(\mathcal{Q}_1)$, then clearly $2 \in S_2(\mathcal{Q}_1)$. In this case:

$$\mathcal{Q}_1 = \{S_1(\mathcal{Q}_1) = \{3\}, S_2(\mathcal{Q}_1) = \{1, 2\}, S_3(\mathcal{Q}_1) = \{1\}, S_4(\mathcal{Q}_1) = \{1, 3\}\}$$

With a similar process, we also deduce:

$$\mathcal{Q}_2 = \{S_1(\mathcal{Q}_2) = \emptyset, S_2(\mathcal{Q}_2) = \{1, 2\}, S_3(\mathcal{Q}_2) = \{1\}, S_4(\mathcal{Q}_2) = \{1, 3, 4\}\}$$

$$\mathcal{Q}_3 = \{S_1(\mathcal{Q}_3) = \emptyset, S_2(\mathcal{Q}_3) = \{1, 2\}, S_3(\mathcal{Q}_3) = \{1, 3\}, S_4(\mathcal{Q}_3) = \{1, 3\}\}$$

$$\mathcal{Q}_4 = \{S_1(\mathcal{Q}_4) = \emptyset, S_2(\mathcal{Q}_4) = \{1, 2\}, S_3(\mathcal{Q}_4) = \{1\}, S_4(\mathcal{Q}_4) = \{1, 3, 4\}\}$$

There is $\mathcal{C} = \{S_1(\mathcal{C}) = \emptyset, S_2(\mathcal{C}) = \{1, 2\}, S_3(\mathcal{C}) = \{1\}, S_4(\mathcal{C}) = \{1, 3\}\}$ of cardinality 5 that is sub-pattern to all the aforementioned lower sub-patterns \mathcal{Q}_i . Note that only $S_2(\mathcal{C}) = S_2(\mathcal{P})$, which is because $s_{22}(\mathcal{X}) < s_{44}(\mathcal{X}) < s_{33}(\mathcal{X}) < s_{13}(\mathcal{X})$ and by definition the lower sub-patterns \mathcal{Q}_i minimize the set weights of $\mathcal{Y} \in \chi(\mathcal{Q}_i)$.

For the following results, we need to generalize the definition of lower sub-pattern.

Definition 16 (Generalized lower sub-patterns $\mathcal{Q}_i(u)$). For a given $i \leq N$ and $u \in \{0, \dots, |S_i(\mathcal{P})|\}$, the generalized lower sub-pattern $\mathcal{Q}_i(u)$ of \mathcal{P} is such that $\text{card}(\mathcal{Q}_i(u)) = k$ and $S_i(\mathcal{Q}_i(u))$ contains the u smallest indices of $S_i(\mathcal{P})$, minimizing the sum of set weights $s(\mathcal{Y})$ with $\mathcal{Y} \in \chi(\mathcal{Q}_i(u))$.

As for lower sub-patterns \mathcal{Q}_i , we can also find similarities between lower sub-patterns $\mathcal{Q}_i(u)$.

Property 12 (Common elements between \mathcal{C} and $\mathcal{Q}_i(u)$). For each $i \leq N$ and $u \in \{0, \dots, |S_i(\mathcal{P})| - 1\}$, lower sub-pattern $\mathcal{Q}_i(u)$ is such that $\forall i' \neq i, S_{i'}(\mathcal{C}) \subseteq S_{i'}(\mathcal{Q}_i(u))$.

Proof. By definition $\text{card}(\mathcal{Q}_i(u)) = k$ and $|S_i(\mathcal{Q}_i(u))| = u$. From **Property 11** $\text{card}(\mathcal{C}) = k - 1$, and from **Property 10** $\forall i' \leq N$ with $|S_{i'}(\mathcal{P})| \geq 2, S_{i'}(\mathcal{P})[2] \in S_{i'}(\mathcal{C})$. Let K be the difference between $\text{card}(\mathcal{Q}_i(u) \setminus S_i(\mathcal{Q}_i(u)))$ and $\text{card}(\mathcal{C} \setminus S_i(\mathcal{C}))$, i.e.,

$$\begin{aligned} K &= \text{card}(\mathcal{Q}_i(u) \setminus S_i(\mathcal{Q}_i(u))) - \text{card}(\mathcal{C} \setminus S_i(\mathcal{C})) \\ &= (k - u) - (k - 1 - |S_i(\mathcal{C})|) = 1 + |S_i(\mathcal{C})| - u \end{aligned}$$

Note that $u \leq |S_i(\mathcal{C})|$, meaning that $K \geq 1$. Because $\forall i' \leq N, S_{i'}(\mathcal{P})[2] \in S_{i'}(\mathcal{C})$, and both \mathcal{C} and $\mathcal{Q}_i(u)$ are sub-patterns of \mathcal{P} , then there are K sets where $|S_{i'}(\mathcal{Q}_i(u))| = |S_{i'}(\mathcal{C})| + 1 = |S_{i'}(\mathcal{P})|$.

Let i' be one of these K sets. By definition, $\text{card}(\mathcal{Q}_{i'}) = k$ and $\text{card}(\mathcal{C}) = k - 1$, meaning that $\text{card}(\mathcal{Q}_{i'} \setminus S_{i'}(\mathcal{Q}_{i'})) = \text{card}(\mathcal{C} \setminus S_{i'}(\mathcal{C}))$. By definition \mathcal{C} is sub-pattern of $\mathcal{Q}_{i'}$, hence $\mathcal{Q}_{i'} \setminus S_{i'}(\mathcal{Q}_{i'}) = \mathcal{C} \setminus S_{i'}(\mathcal{C})$. Consequently, $S_i(\mathcal{Q}_{i'}) = S_i(\mathcal{C})$ and we deduce:

$$\begin{aligned} &\text{card}(\mathcal{Q}_i(u) \setminus S_i(\mathcal{Q}_i(u))) - \text{card}(\mathcal{Q}_{i'} \setminus S_{i'}(\mathcal{Q}_{i'})) \\ &= (k - u) - (\text{card}(\mathcal{Q}_{i'}) - |S_{i'}(\mathcal{Q}_{i'})|) \\ &= (k - u) - (k - |S_i(\mathcal{C})|) = K - 1 \end{aligned}$$

From **Property 10** and because $V = 1$, there are $K - 1$ sets of $\mathcal{Q}_i(u)$ with one more element than the respective set of $\mathcal{Q}_{i'}$. However:

$$\begin{aligned} K - 1 &= 1 + |S_i(\mathcal{C})| - u - 1 \\ &= |S_i(\mathcal{C})| - u \\ &= |S_i(\mathcal{Q}_{i'})| - |S_i(\mathcal{Q}_i(u))| \end{aligned}$$

The difference between $|S_i(\mathcal{Q}_{i'})|$ and $|S_i(\mathcal{Q}_i(u))|$ is exactly $K - 1$.

Consequently, for set i and the $K - 1$ sets where $\mathcal{Q}_i(u)$ contains one more element than $\mathcal{Q}_{i'}$, patterns $\mathcal{Q}_i(u)$ and $\mathcal{Q}_{i'}$ have the same number of elements. As both patterns are of cardinality k , then for the sets complementary to i and the $K - 1$ sets where $\mathcal{Q}_i(u)$ contains one more element than $\mathcal{Q}_{i'}$, patterns $\mathcal{Q}_i(u)$ and $\mathcal{Q}_{i'}$ have the same number of elements. As both lower sub-patterns minimize their set weights, they are equal for these complementary sets.

Hence, for all sets besides i , $\mathcal{Q}_i(u)$ contains all elements of $\mathcal{Q}_{i'}$, which itself contains all elements of \mathcal{C} . Thus $\forall i' \neq i, S_{i'}(\mathcal{C}) \subseteq S_{i'}(\mathcal{Q}_i(u))$. \square

With all lower sub-patterns $\mathcal{Q}_i(u)$ defined, the following theorem shows that a flexible pattern with a set of cardinality 1 is pattern-facet.

Theorem 4. \mathcal{P} is pattern-facet if and only if \mathcal{P} is flexible pattern.

Proof. Recall that without loss of generality, pattern sets can be ordered such that $|S_i(\mathcal{P})| \leq |S_{i+1}(\mathcal{P})|$, meaning $|S_1(\mathcal{P})| = 1$. Let $\mathcal{X} \in \chi(\mathcal{P})$ with permutation π_{id} . Consider the following lower sub-patterns: $\mathcal{Q}_1, \mathcal{Q}_i \forall i \leq N$ such that $S_i(\mathcal{P})[1] \notin S_i(\mathcal{Q}_1); \mathcal{Q}_{i'}(u) \forall i' \leq N, \forall u \in \{0, \dots, |S_{i'}(\mathcal{C})| - 1\}$. For all mentioned sub-patterns, we consider their respective variable set with permutation π_{id} denoted $\mathcal{X}_1; \mathcal{X}_i; \mathcal{X}_{i'}(u)$ and their respective solution $X_{\mathcal{X}_1}; X_{\mathcal{X}_i}; X_{\mathcal{X}_{i'}(u)}$. This results in a total of $\text{card}(\mathcal{P})$ solutions. There is

one solution $X_{\mathcal{X}_1}$. As \mathcal{P} is of rank k , there are $\text{card}(\mathcal{P}) - k$ solutions $X_{\mathcal{X}_i}$. As $\text{card}(\mathcal{C}) = k - 1$, there are $k - 1$ solutions $X_{\mathcal{X}_{i'(u)}}$.

By definition of lower sub-patterns \mathcal{Q}_i and $\mathcal{Q}_i(u)$ and from **Property 7**, all mentioned variable sets are k -intersections of \mathcal{X} . Hence, all mentioned solutions are feasible and verify $(pi(\mathcal{X}))$ to equality.

Consider now the points associated to the afore-enumerated solutions. We can prove these points to be linearly independent. Start by considering first the point associated to $X_{\mathcal{X}_1}$. As it is the only point considered, it is necessarily linearly independent. From **Property 11**, \mathcal{C} is sub-pattern to all \mathcal{Q}_i and of cardinality $k - 1$. Consequently, \mathcal{Q}_i with $S_i(\mathcal{P})[1] \notin S_i(\mathcal{Q}_1)$ is the only lower sub-pattern, excluding the generalized lower sub-patterns, with $S_i(\mathcal{P})[1] \in S_i(\mathcal{Q}_i)$. It results that for each solution $X_{\mathcal{X}_i}$, the associated point is the only one with $x_{iS_i(\mathcal{P})[1]} = 1$, thus they are linearly independent. For each solution $X_{\mathcal{X}_{i'(u)}}$, starting with large u , the associated point is the first one with $x_{i'S_{i'}(\mathcal{P})[u]} = 0$, thus being linearly independent.

The enumerated solutions yield $\text{card}(\mathcal{P})$ linearly independent points. Moreover, from **Theorem 3**, $\forall x_{ij} \notin \mathcal{X}$, there is a feasible solution with $x_{ij} = 1$ and $x_{ij+1} = 0$ that verifies $(pi(\mathcal{X}))$ to equality. Sequentially adding these points associated to their corresponding solutions to our pool of $\text{card}(\mathcal{P})$ points still keeps them linearly independent, as there are the only ones with $x_{ij} = 1$ and $x_{ij+1} = 0$ with $x_{ij} \notin \mathcal{X}$. As there are $n - \text{card}(\mathcal{P})$ of these new points, there is a total of n linearly independent points.

Thus, a pattern \mathcal{P} with a set of cardinality 1 is pattern-facet if and only if it is flexible pattern. \square

Recall that for a pattern to be flexible pattern it solely requires to verify conditions (i) , (ii) and (iii) . Hence, a pattern with a set of cardinality 1 is pattern-facet if conditions (i) , (ii) and (iii) hold. It is shown in **Section 4** that verifying if these three conditions hold for a given pattern can be done in polynomial time. As for **Theorem 3**, the result of **Theorem 4** is used for the experimental results in **Section 5**, guaranteeing a flexible pattern to be pattern-facet if $\min_{i \leq N} |S_i(\mathcal{P})| = 1$, or a lower bound on the dimensions of the faces defined by the pattern inequalities of \mathcal{P} otherwise.

3.6 Conditions for any pattern

For a pattern \mathcal{P} such that $\min_{i \leq N} |S_i(\mathcal{P})| \geq 2$, the conditions (i) , (ii) and (iii) are necessary but not sufficient for \mathcal{P} to be pattern-facet. This is because the lower sub-patterns \mathcal{Q}_i of \mathcal{P} lose many of their structural properties in the general case. In the following, we present three new conditions, that will complement the aforementioned conditions. For this purpose, we present new patterns \mathcal{R}_u that will take the role of the lower sub-patterns \mathcal{Q}_i . The idea is to help constructing points with a common coefficients, which is convenient to provide independent points.

In the following, we consider \mathcal{P} a flexible pattern of rank k .

Definition 17 (Nested sub-patterns \mathcal{R}_u). Pattern $\{\mathcal{R}_u, 1 \leq u \leq U'\}$ are nested sub-patterns of \mathcal{P} if $\forall u$, $\text{card}(\mathcal{R}_u) = k - u$, \mathcal{R}_u is a lower sub-pattern of \mathcal{P} and \mathcal{R}_u sub-pattern of \mathcal{R}_{u-1} .

For the following, we define $U' = \min_{u \leq k} \{\max_{i \leq N} \{S_i(\mathcal{P}) - S_i(\mathcal{R}_u)\} \geq u\}$. We consider the following subset of nested sub-patterns $\{\mathcal{R}_u, 1 \leq u \leq U'\}$. We also consider $\mathcal{X} \in \chi(\mathcal{P})$ and $\mathcal{Y}_u \in \chi(\mathcal{R}_u)$, $\forall u \leq U'$ with permutation π_{id} .

In the following we define three new conditions. Condition (iv) indicates that there are k -intersections of \mathcal{X} with \mathcal{Y}_u and u variables in a same group. Condition (v) is similar, but with u variables in at least two different groups. Condition (vi) is a more constrained version of condition (iii) , but only for variables in $\mathcal{Y}_{U'}$.

Definition 18 (Condition (iv) : selection of items in the same group). For any $u \leq U'$ and $i \leq N$ such that $|S_i(\mathcal{P}) - S_i(\mathcal{R}_u)| \geq u$, there is a variables set \mathcal{Z} containing the u variables with the smallest indices of group i in $\mathcal{X} \setminus \mathcal{Y}_u$. The variable set $\mathcal{Z} \cup \mathcal{Y}_u$ is a k -intersection of \mathcal{X} .

Definition 19 (Condition (v): selection of items in different group). For any $u \in \{2, \dots, U'\}$, $\exists \mathcal{Z}_u \subset \mathcal{X}$ of cardinality u such that $\mathcal{Z}_u \cap \mathcal{Y}_1 = \emptyset$, \mathcal{Z}_u contains variables in at least two different groups and the variable set $\mathcal{Y}_u \cup \mathcal{Z}_u$ is a k -intersection of \mathcal{X} .

Definition 20 (Condition (vi): constrained independence of an item from its predecessor). For any $x_{ij} \in \mathcal{Y}_{U'}$, $\exists \mathcal{Z} \subseteq \mathcal{V}$ a k -intersection of \mathcal{X} such that $\forall x_{i'j'} \in \mathcal{Y}_{U'}$, if $i' = i$ and $j' \geq j$, then $x_{i'j'} \notin \mathcal{Z}$, otherwise $x_{i'j'} \in \mathcal{Z}$.

Note that contrary to conditions (i), (ii) and (iii), conditions (iv), (v) and (vi) do not apply on \mathcal{P} but, instead, on the nested sub-patterns.

Example 9. Let $(4, 4, [8, 4, 2, 3], v, 37)$ an instance of the (SMKP). Let $\mathcal{P} = \{S_1(\mathcal{P}) = \{1, 2\}, S_2(\mathcal{P}) = \{2, 3\}, S_3(\mathcal{P}) = \{2, 4\}, S_4(\mathcal{P}) = \{2, 4\}\}$ of rank 4. Let the nested sub-patterns $\mathcal{R}_1 = \{S_1(\mathcal{R}_1) = \{1\}, S_2(\mathcal{R}_1) = \{2, 3\}, S_3(\mathcal{R}_1) = \emptyset, S_4(\mathcal{R}_1) = \emptyset\}$ and $\mathcal{R}_2 = \{S_1(\mathcal{R}_2) = \{1\}, S_2(\mathcal{R}_2) = \{2\}, S_3(\mathcal{R}_2) = \emptyset, S_4(\mathcal{R}_2) = \emptyset\}$. Let $\mathcal{X} = \{x_{11}, x_{12}, x_{22}, x_{23}, x_{32}, x_{34}, x_{42}, x_{44}\}$, $\mathcal{Y}_1 = \{x_{11}, x_{22}, x_{23}\}$ and $\mathcal{Y}_2 = \{x_{11}, x_{22}\}$.

Condition (iv) is verified. For \mathcal{Y}_1 the following variable sets are 4-intersections of \mathcal{X} : $\{x_{11}, x_{12}, x_{22}, x_{23}\}$, $\{x_{11}, x_{22}, x_{23}, x_{32}\}$, $\{x_{11}, x_{22}, x_{23}, x_{42}\}$. For \mathcal{Y}_2 the following variable sets are 4-intersections of \mathcal{X} : $\{x_{11}, x_{22}, x_{32}, x_{34}\}$, $\{x_{11}, x_{22}, x_{42}, x_{44}\}$.

Condition (v) is verified. For \mathcal{Y}_2 the following variable set is a 4-intersection of \mathcal{X} : $\{x_{11}, x_{12}, x_{22}, x_{32}\}$, with x_{12} and x_{32} being in different groups, while not being in \mathcal{Y}_1 .

Condition (vi) is verified. For \mathcal{Y}_2 , the following variable sets are 4-intersections of \mathcal{X} : $\{x_{11}, x_{12}, x_{32}, x_{34}\}$ and $\{x_{22}, x_{23}, x_{32}, x_{34}\}$. The first one contains all variables of \mathcal{Y}_2 but x_{22} , and the second one contains all variables of \mathcal{Y}_2 but x_{11} .

In this example, the lower sub-pattern \mathcal{R}_1 is the unique one of cardinality 3 minimizing the set weights of \mathcal{Y}_1 . However, \mathcal{R}_2 is not the one of cardinality 2 minimizing the set weights of \mathcal{Y}_2 . In fact, $\mathcal{R}'_2 = \{S_1(\mathcal{R}'_2) = \{1, 2\}, S_2(\mathcal{R}'_2) = \emptyset, S_3(\mathcal{R}'_2) = \emptyset, S_4(\mathcal{R}'_2) = \emptyset\}$ is of cardinality 2 and minimizes the set weights of $\mathcal{Y}'_2 \in \chi(\mathcal{R}'_2)$. Because \mathcal{R}'_2 is not sub-pattern to \mathcal{R}_1 , they cannot be nested sub-patterns.

For a set of nested sub-patterns \mathcal{R}_u , conditions (iv) can be verified in polynomial time. Indeed, the k -intersection are explicitly defined, it only requires to compute the sum of its set weights. Also, verifying conditions (v) and (vi) requires to find a k -intersection, as for conditions (ii) and (iii). Hence, it can be verified in polynomial time, in a similar fashion as for conditions (ii) and (iii), using a variant of the shortest path algorithm. However, conditions (iv), (v) and (vi) apply on a set of nested sub-patterns, instead of on a single pattern \mathcal{P} . As shown in **Example 9**, the lower sub-patterns of \mathcal{P} minimizing the sum of the set weights of their respective variable sets may not be nested. Consequently, the difficulty is that one may need to enumerate all lower sub-patterns of \mathcal{P} to find a set of nested sub-patterns verifying conditions (iv), (v) and (vi).

The following theorem shows that conditions (i) to (vi) are sufficient to prove any pattern to be pattern-facet.

Theorem 5 (Sufficient conditions for any pattern-facet). *If nested sub-patterns $\{\mathcal{R}_u, 1 \leq u \leq U'\}$ of \mathcal{P} verify conditions (iv), (v) and (vi), then \mathcal{P} is pattern-facet.*

The complete proof is in **Appendix A.5**, relying on **Lemma 7** defined in the same appendix.

Even though conditions (iv) (v) and (vi) can provide pattern-facets, we will only be using flexible patterns for the experimental results. The rationale behind is that as shown in **Example: 9**, one may need to enumerate all sub-patterns of \mathcal{P} to find nested patterns verifying conditions (iv), (v) and (vi). Another reason is that from **Theorems 3** and **4**, a flexible pattern \mathcal{P} is pattern-facet if it has a set of cardinality one. Otherwise it has a lower bound on the dimensions of the faces defined by the pattern inequalities of \mathcal{P} . As shown in **Section 5**, using flexible pattern, with or without the cuts of CPLEX, within a Branch and Cut framework can drastically reduce the number of nodes explored and the computational time required to solve instances of the (SMKP). Furthermore, experimental results in **Section 5** show that for some instances, a few flexible patterns are generated. Consequently, adding conditions (iv) (v) and (vi) would further reduce the number of patterns generated, making it more difficult to measure their impact.

4 Algorithms

In this section, we define the two-phase B&C scheme. The first phase of this scheme generates flexible patterns as a pre-processing. The second phase separates the associated pattern inequalities within a B&C framework. The flexible-pattern generating algorithm is described in **Section 4.2**, based on two algorithms defined in **Section 4.1**. The separation algorithm is described in **Section 4.3**, producing the most violated inequality for a given pattern in polynomial time. The two-phase B&C scheme is described in **Section 4.4**

4.1 Graph model associated to variable sets

The shortest path problem and many of its variants are known to be easy to solve [1] [5]. In the case of the (SMKP), for a given pattern \mathcal{P} , it is possible to define a graph associated to $\mathcal{X} \in \chi(\mathcal{P})$, for which solving a variant of the shortest path problem gives the rank of \mathcal{P} . This graph will be used to compute the rank of a pattern and to verify conditions (ii) and (iii). Such graph could also be used to compute the sub-patterns \mathcal{Q}_i or to verify conditions (v) and (vi).

Consider a pattern \mathcal{P} and a variable set $\mathcal{X} \in \chi(\mathcal{P})$. Let $G = (W, A)$ be the graph defined as follows. The variable set \mathcal{X} is associated to the vertex set W , a source vertex p and a sink vertex q are added to W . For convenience purposes, each vertex in W is denoted by the corresponding variable in \mathcal{X} and the vertices associated to $x_{ij} \in \mathcal{X}$ are renumbered with the same order in a compact sequence in \mathcal{X} . More precisely consider the following renumbering, denoted by parentheses on the indices: For any $x_{ij} \in \mathcal{X}$, we consider $x_{(i)(j)} = x_{ij}$ with $(i) = i$ and (j) the number of variables $x_{ij'} \in \mathcal{X}$ with $j' \leq j$. To each couple $(x_{(i)(j)}, x_{(i)(j+1)}) \in \mathcal{X}^2$ corresponds an arc in A . To each $x_{(i)(j)} \in \mathcal{X}$ corresponds an arc in A from $x_{(i)(j)}$ to $x_{(i')(1)}$, $i' > i$. To each $x_{(i)(1)} \in \mathcal{X}$ corresponds an arc from source vertex p to $x_{(i)(1)}$ in A . Similarly, for each vertex $x_{(i)(j)}$, it corresponds an arc from $x_{(i)(j)}$ to sink vertex q . Finally the weight of an arc heading to $x_{(i)(j)}$ is $s_{ij}(\mathcal{X})$ and to sink vertex q is 0.

Example 10. Let $(3, 3, [3, 2, 2], v, C)$ be an instance of the (SMKP). For $\mathcal{X} = \{x_{11}, x_{12}, x_{21}, x_{22}, x_{23}, x_{31}, x_{33}\}$, the vertex set is $W = \{p, q, x_{(1)(1)}, x_{(1)(2)}, x_{(2)(1)}, x_{(2)(2)}, x_{(2)(3)}, x_{(3)(1)}, x_{(3)(2)}\}$. **Figure 4** illustrates the graph G associated to \mathcal{X} .

With graph G defined with respect to \mathcal{X} , the following property makes the link between a path in G and a variable set $\mathcal{Y} \subseteq \mathcal{X}$.

Property 13. *Finding a shortest path in G featuring exactly $k + 1$ arcs is equivalent to finding $\mathcal{Y} \subseteq \mathcal{X}$, with \mathcal{Y} featuring the following properties: 1) $|\mathcal{Y}| = k$, 2) $\forall x_{ij} \in \mathcal{Y}$, if $x_{ij'} \in \mathcal{X}$, and $j' < j$, then $x_{ij'} \in \mathcal{Y}$ 3) \mathcal{Y} minimizes its set weights $s_{ij}(\mathcal{Y})$.*

Proof. 1) The path between p and q with exactly $k + 1$ arcs is a path with $k + 2$ vertices. Note that such path necessarily exists with $k \in \{1, \dots, \text{card}(\mathcal{P})\}$. Indeed, by construction of G , there is a path going through all nodes $x_{(i)(j)}$ and there is an arc from any $x_{(i)(j)}$ to q . Also any arc $(x_{(i)(j)}, x_{(i')(j')})$ is with $(i') \geq i$ or $(j') > (j)$, thus there are no cycles in G . Hence, the $k + 2$ vertices are necessarily different. As p and q do not correspond to any variables of \mathcal{X} , a path with exactly $k + 1$ arcs represents a set \mathcal{Y} with exactly k variables of \mathcal{X}

2) By construction of G , the only arc towards $x_{(i)(j)}$ is from $x_{(i)(j-1)}$ with $(j) > 1$. Consequently, $\forall x_{ij} \in \mathcal{Y}$, if $x_{ij'} \in \mathcal{X}$, and $j' < j$, then $x_{ij'} \in \mathcal{Y}$.

3) By construction of G , for each $x_{ij} \in \mathcal{X}$, every arc towards x_{ij} has a weight $s_{ij}(\mathcal{X})$. As such, the weight of a path with a set $\mathcal{Y} \subseteq \mathcal{X}$ of variables, has a weight equal to $\sum_{x_{ij} \in \mathcal{Y}} s_{ij}(\mathcal{X})$. Hence, set \mathcal{Y} corresponds to a shortest path minimizing the sum of its set weights. \square

Computing the rank Clearly, if the sum of the set weights of \mathcal{Y} is smaller than or equal to C , then \mathcal{Y} is a k -intersection of \mathcal{X} . By definition, if a k -intersection of \mathcal{X} exists, then there is a feasible solution with k variables of \mathcal{X} to 1. Hence, the idea to compute the rank of a pattern \mathcal{P} is to find a

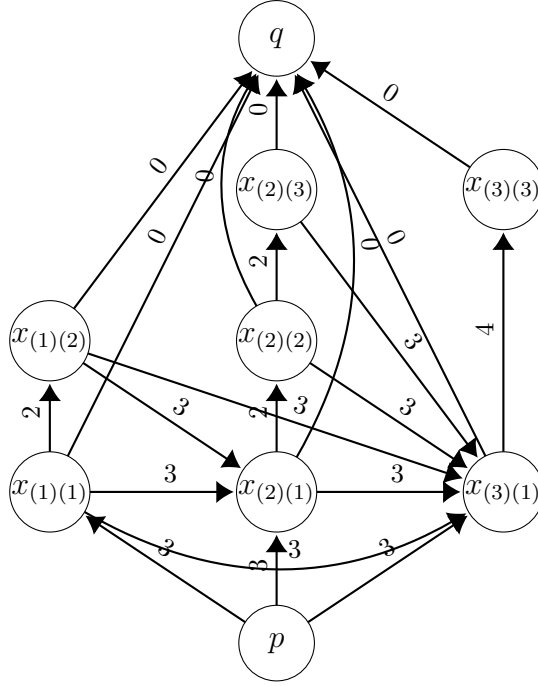


Figure 4: G for **Example 10**

maximum cardinality path in G associated to $\mathcal{X} \in \chi(\mathcal{P})$, with its total weight being smaller than or equal to C . From **Property 13**, if the maximum cardinality path found is with $k + 1$ arcs, then the rank of \mathcal{P} is k . Such rank computing algorithm is described in **Algorithm 1**.

The procedure `COMPUTERANK`(\mathcal{P}) in **Algorithm 1** returns the rank of \mathcal{P} . Indeed, this algorithm is similar to the Bellman-Ford algorithm [1], but with a different stopping condition. Because all weights are non-negative, if **Algorithm 1** stops for a given $nbArcs$, then there cannot be a path of $nbArcs$ arcs with a total weight smaller than or equal to C . Hence, the maximum cardinality path found is with $nbArcs - 1$ arcs, containing $nbArcs - 2$ vertices different than p and q . Consequently, the rank of \mathcal{P} is $k = nbArcs - 2$.

Verifying conditions (ii) and (iii) Let \mathcal{P} be a pattern of rank k and $\mathcal{X} \in \chi(\mathcal{P})$ with permutation π_{id} . To verify if condition (ii) holds for a group i , one need to find a set \mathcal{Y} , a k -intersection of \mathcal{X} with $x_{iM} \in \mathcal{Y}$. To verify if condition (iii) holds for x_{ij} , one need to find a set \mathcal{Y} , a k -intersection of \mathcal{X} with $x_{ij-1} \in \mathcal{Y}$ and $x_{ij} \notin \mathcal{Y}$. Let u be the number of variables $x_{ij'} \in \mathcal{X}$, with $j' \leq M$ for condition (ii), and with $j' < j$ for condition (iii). Finding such k -intersection is to find a shortest path of length $k - u + 1$ in G , without any variable of group i . To do so, one can use a variant of the shortest path algorithm as described in **Algorithm 2**.

The procedure `VERIFYCONDITION`(\mathcal{P}, i, j) in **Algorithm 2**, in the case $j = M + 1$, returns true only if (ii) is verified for i , and in the case $j \in S_i(\mathcal{P})$, returns true only if (iii) is verified for i and j . Indeed, this algorithm is similar to the Bellman-Ford algorithm [1], but with a different stopping condition. The path found is a shortest path with $k - u + 1$ arcs, thus with $k - u + 2$ vertices and with $k - u$ vertices different than p and q . By construction of \mathcal{X}' , this path is without any variables of group i . Consequently if the total weight of the generated shortest path plus $\sum_{j'=1}^{j-1} w_{j'}$ is smaller than or equal to C , then there is a k -intersection of \mathcal{X} , containing all variables in the generated shortest path and the variables $x_{ij'} \in \mathcal{V}$, $j' < j$. In the case $j \leq M$, then the resulting k -intersection contains

Algorithm 1 Computing the rank of a pattern

```
procedure COMPUTERANK(pattern  $\mathcal{P}$ ):  
  build  $G = (W, A)$  associated to  $\mathcal{X} \in \chi(\mathcal{P})$  for a permutation  $\pi$   
   $dist[p] \leftarrow 0$   
   $dist[w] \leftarrow +\infty, \forall w \in W \setminus \{p\}$   
   $nbArcs \leftarrow 0$   
  while  $\min(dist) \leq C$  do  
     $nDist[w] \leftarrow \infty, \forall w \in W$   
    for  $a \in A$  do  
       $nDist[a.head] \leftarrow \min(nDist[a.head], dist[a.tail] + a.weight)$   
    end for  
     $dist \leftarrow nDist$   
     $nbArcs \leftarrow nbArcs + 1$   
  end while  
   $k \leftarrow nbArcs - 2$   
  return  $k$   
end procedure
```

the variable x_{ij-1} but not x_{ij} , which corresponds to condition (iii). In the case $j = M + 1$, then the resulting k -intersection contains x_{iM} , which corresponds to the condition (ii).

Remark 3. Both algorithms described run in polynomial time with respect to $|W|$, with $|W| = card(\mathcal{P}) + 2$ in **Algorithm 1**, and $|W| = card(\mathcal{P}) + 2 - |S_i(\mathcal{P})|$ in **Algorithm 2**. In the case of **Algorithm 1**, the **while** loop can occur $|W|$ time, and in the case of **Algorithm 2**, the **while** loop can occur $rank(\mathcal{P}) \leq |W|$ time. In both cases, in the **while** loop we iterate over the arcs of A , meaning $|A| \leq |W|^2$ operations. Hence, both algorithms have a worst case time complexity of $\mathcal{O}(|W|^3)$. It is worth mentioning that because of the structure of G , $|A|$ is usually much smaller than $|W|^2$, and the **while** loop rarely iterates $|W|$ times. Hence, these algorithms usually requires much less than $|W|^3$ operations.

Theorem 6. *Verifying condition (i), (ii) and (iii) is in polynomial time.*

Proof. Condition (i) requires to verify if each set is non-empty, which is linear with respect to the number of sets N .

Condition (ii) requires to run **Algorithm 2** for each x_{iM} , $i \leq N$, which is an algorithm of complexity $\mathcal{O}(|W|^3)$, $|W| \leq n$ a total of $N \leq n$ times. Condition (ii) can be verified in $\mathcal{O}(n^4)$ time.

Condition (iii) requires to run **Algorithm 2** for each $x_{ij} \in \mathcal{X}$, which is an algorithm of complexity $\mathcal{O}(|W|^3)$, $|W| \leq n$ a total of $|\mathcal{X}| \leq n$ times. Condition (iii) can be verified in $\mathcal{O}(n^4)$ time. \square

Remark 4. It is worth mentioning that the time complexity $\mathcal{O}(n^4)$ is rarely reached. Indeed, only one pattern has a cardinality equals to n : the one containing N times the set $\{1, \dots, M\}$. All other patterns have a smaller cardinality, which reduces the required time to verify (ii) and (iii) in two ways. The first one is that **Algorithm 2** is needed $card(\mathcal{P})$ times for (iii), and $N \leq card(\mathcal{P})$ times for (ii). The second one is that the complexity of **Algorithm 2** depends on $|W|$, which is $card(\mathcal{P}) + 2 - |S_i(\mathcal{P})|$ as mentioned in **Remark 3**. Then likewise **Algorithm 2** usually requires much less than $|W|^3$ operations.

Algorithm 2 can also be extended to gather further information on a pattern as described in the following. These extensions are not used for the experimental results in **Section 5**. Only **Algorithm 1** and **Algorithm 2** are required to apply **Theorem 3** and **Theorem 4**

Algorithm 2 Verifying conditions (ii) and (iii)

```

procedure VERIFYCONDITION(pattern  $\mathcal{P}$ ,  $i \in [1; N]$ ,  $j \in [1; M + 1]$ ):
     $\triangleright j \leq M$  for condition (ii),  $j = M + 1$  for condition (iii)
     $u \leftarrow |\{j' \in S_i(\mathcal{P}) : j' < j\}|$ 
     $\mathcal{X} \leftarrow$  set in  $\chi(\mathcal{P})$  for permutation  $\pi_{id}$ 
     $\mathcal{X}' \leftarrow \mathcal{X} \setminus \{x_{ij'} : j' \in S_i(\mathcal{P})\}$ 
    build  $G = (W, A)$  associated to  $\mathcal{X}'$ 
     $dist[p] \leftarrow 0$ 
     $dist[w] \leftarrow +\infty, \forall w \in W \setminus \{p\}$ 
     $nbArcs \leftarrow 0$ 
    while  $nbArcs \leq rank(\mathcal{P}) - u$  do
         $nDist[w] \leftarrow \infty, \forall w \in W$ 
        for  $a \in A$  do
             $nDist[a.head] \leftarrow \min(nDist[a.head], dist[a.tail] + a.weight)$ 
        end for
         $dist \leftarrow nDist$ 
         $nbArcs \leftarrow nbArcs + 1$ 
    end while
    return  $dist[q] + \sum_{j'=1}^{j-1} w_{j'} \leq C$ 
end procedure

```

Computing generalized lower sub-patterns $\mathcal{Q}_i(u)$ To find a flexible pattern, there is no need to compute the lower sub-patterns $\mathcal{Q}_i(u)$. However, it is worth mentioning that finding $\mathcal{Q}_i(u)$ can also be done with a variant of the shortest path algorithm. More precisely, finding $\mathcal{Q}_i(u)$ can be obtained via **Algorithm 2**, for x_{ij+1} with $j = S_i(\mathcal{Q}_i(u))(u)$. The only modification is to add an extra step to memorize the paths, and to return the path to node q .

Verifying conditions (v) and (vi) Let \mathcal{P} be a pattern and $\{\mathcal{R}_u, 1 \leq u \leq U'\}$ the nested sub-patterns of \mathcal{P} . To verify if conditions (v) or (vi) hold for its nested sub-patterns \mathcal{R}_u , one needs to find k -intersections. Finding a k -intersection is also what is required to verify conditions (ii) and (iii). Hence, a similar algorithm as **Algorithm 2** can be used to verify these conditions.

4.2 Pattern generation

The pattern generation procedure heavily relies on conditions (ii) and (iii) and also on the rank. The complete procedure is described in **Algorithm 3**, using **Algorithm 1** and **Algorithm 2**.

In **Algorithm 3**, the lower bound on k ensures that the rank is high enough to create a flexible pattern. Indeed, because of condition (i), too small of a rank will lead to patterns that cannot be flexible patterns. The algorithm starts by initializing a pattern \mathcal{P} containing N times the set $\{M\}$. By construction, \mathcal{P} has at least one index per set, hence condition (i) is verified. In order for the pattern inequalities of \mathcal{P} not to be trivial, it requires $card(\mathcal{P}) > rank(\mathcal{P})$. As the \mathcal{P} is initialized with cardinality N , in the case $rank(\mathcal{P}) \geq N$ one need to add elements to \mathcal{P} . For the purpose of this algorithm, we start with elements of higher indices. To do so, we select a random i and add $j - 1$ to $S_i(\mathcal{P})$, with $j = S_i(\mathcal{P})(1)$ the smallest index in $S_i(\mathcal{P})$. The sets are randomly selected in order to make it possible to generate different patterns for a same k . The core of the procedure is to modify \mathcal{P} so that it verifies condition (ii) and (iii). We randomly select i and j such that $j \notin S_i(\mathcal{P})$ and $j + 1 \in S_i(\mathcal{P})$ or $j = M$ and call the procedure $VERIFYCONDITION(\mathcal{P}, i, j)$. If $VERIFYCONDITION(\mathcal{P}, i, j)$ returns false, then in the case $j < M$ we replace $j + 1$ by j in $S_i(\mathcal{P})$, and in the case $j = M$ we add M to $S_i(\mathcal{P})$. The elements are also chosen randomly, so that two same patterns for the same rank k can

Algorithm 3 Generating a flexible pattern

```
procedure GENERATEPATTERN( instance  $(N, M, w, v, C)$  of the (SMKP))
   $lowerBound \leftarrow \lceil C / \sum_{j=1}^M w_j \rceil$ 
   $k \leftarrow$  random integer in  $[lowerBound; n]$ 
   $\mathcal{P} \leftarrow N$  sets  $\{M\}$ 
  while  $card(\mathcal{P}) < k$  do
     $i \leftarrow$  random integer in  $[1; N]$  such that  $|S_i(\mathcal{P})| < M$ 
     $j \leftarrow S_i(\mathcal{P})(1)$ 
     $S_i(\mathcal{P}) \leftarrow S_i(\mathcal{P}) \cup \{j - 1\}$ 
  end while
  while (ii) and (iii) not verified for  $\mathcal{P}$  do
     $\mathcal{X} \leftarrow$  set in  $\chi(\mathcal{P})$  for permutation  $\pi_{id}$ 
     $x_{ij} \leftarrow$  variable chosen randomly in  $\mathcal{X}$  such that  $x_{ij+1} \in \mathcal{X}$  or  $j = M$ 
    if  $j = M$  and !VERIFYCONDITION( $\mathcal{P}, i, j$ ) then ▷ (see Algorithm 2)
       $S_i(\mathcal{P}) = S_i(\mathcal{P}) \cup \{M\}$ 
    end if
    if  $j < M$  and !VERIFYCONDITION( $\mathcal{P}, i, j$ ) then ▷ (see Algorithm 2)
       $S_i(\mathcal{P}) = S_i(\mathcal{P}) \cup \{j\} \setminus \{j + 1\}$ 
    end if
  end while
  if COMPUTERANK( $\mathcal{P}$ )= $k$  then ▷ (see Algorithm 1)
    return  $\mathcal{P}$ 
  else
    pattern discarded
  end if
end procedure
```

yield two different flexible patterns through this procedure. Finally, because of the modifications of \mathcal{P} , it is possible for it to have $\text{rank}(\mathcal{P}) > k$, hence we need to compute it. If the rank is different than k , then \mathcal{P} may not be flexible pattern, as verifying the conditions (ii) and (iii) depends on the rank k . One would need to restart the whole procedure to verify if (ii) and (iii) indeed hold for \mathcal{P} . We chose to discard \mathcal{P} in this case, in order to avoid to long computational time for each call to the procedure $\text{GENERATEPATTERN}((N, M, w, v, C))$.

Procedure $\text{GENERATEPATTERN}((N, M, w, v, C))$ only returns a pattern if it is a flexible pattern. As \mathcal{P} is discarded if $\text{rank}(\mathcal{P}) \neq k$, it ensures the rank of \mathcal{P} to be exactly k . Similarly, pattern \mathcal{P} verifies conditions (ii) and (iii). Indeed, condition (ii) is verified $\forall i \in \{1, \dots, N\}$ and (iii) is verified $\forall i \in \{1, \dots, N\}$ and $j \in S_i(\mathcal{P})$. Also, if a condition (ii) is verified for a given i , it is still verified if \mathcal{P} is modified during the core of procedure $\text{GENERATEPATTERN}((N, M, w, v, C))$. Similarly, if condition (iii) is verified for a given i and $j \in S_i(\mathcal{P})$, it is still verified if \mathcal{P} is modified during the core of procedure $\text{GENERATEPATTERN}((N, M, w, v, C))$. Indeed, as either one element is added, or an index j' is replaced by $j' - 1$ in a set $S_{i'}(\mathcal{P})$, then the same k -intersection satisfying (ii) or (iii) can be found, or a k -intersection with lighter total set weights. As \mathcal{P} is initialized with one element per set, and no element can be removed, condition (i) is clearly verified.

Note that $\text{GENERATEPATTERN}((N, M, w, v, C))$ only needs the instance, more precisely the knapsack bound and the item weight from the instance, hence it can be only used in pre-processing. Indeed, only the constraints are needed to compute a pattern verifying (i), (ii) and (iii). Also, procedure $\text{GENERATEPATTERN}((N, M, w, v, C))$ returns at most a single pattern, which may not be enough to make a large difference in a B&C framework. Consequently, we call this procedure multiple times as the pre-processing step of the two phase scheme, in order to generate various patterns. Note that each call is independent to the previous ones, meaning that multiple calls can be done in parallel on different threads. The set of patterns obtained with this procedure is then used within a B&C framework, with the separation algorithm described in the following section.

4.3 Separation algorithm for the (SMKP)

For the separation algorithm, we have two pieces of information: the fractional point \tilde{X} and the set of generated patterns. Recall that for a given pattern \mathcal{P} , variables sets $\mathcal{X} \in \chi(\mathcal{P})$, and their associated inequalities $(pi(\mathcal{X}))$ exist for any permutation π . The aim of the separation algorithm is to find, for a given pattern, the permutation π maximizing the left hand side of $(pi(\mathcal{X}))$ for $\mathcal{X} \in \chi(\mathcal{P})$ with permutation π . However, the number of pattern-inequalities is exponential, as any permutation of groups is possible (see **Section 3**). Given a pattern \mathcal{P} , in order to find the best permutation π , we can solve a Maximum Matching Problem (MMP).

Definition 21 (Maximum Matching Problem). Let \mathcal{H} be a weighted bipartite graph. The (MMP) is to find the set of edges E , such that at most one edge of E is incident to each vertex of \mathcal{H} , while maximizing the sum of the weight of E .

Property 14. Finding a permutation π maximizing the violation of $(pi(\mathcal{X}))$ with $\mathcal{X} \in \chi(\mathcal{P})$ for permutation π is equivalent to solving the (MMP).

Proof. The aim is to find the permutation π of the sets of \mathcal{P} , maximizing the left hand side of the inequality $(pi(\mathcal{X}))$ with $\mathcal{X} \in \chi(\mathcal{P})$. To do so, we can build a bipartite graph $\mathcal{H} = (\mathcal{H}_1, \mathcal{H}_2, E)$. Each vertex in \mathcal{H}_1 corresponds to a group $i \in \{1, \dots, N\}$ of solution \tilde{X} and each vertex in \mathcal{H}_2 corresponds to a set $S_{i'}(\mathcal{P})$, $i' \in \{1, \dots, N\}$. In the set of edges E there is an edge (i, i') for each $i \in \mathcal{H}_1$, $i' \in \mathcal{H}_2$. The edge (i, i') has a weight equal to $\sum_{j \in S_{i'}(\mathcal{P})} \tilde{x}_{ij}$.

Solving this (MMP) to optimality yields a matching maximizing the weight of the considered edges. One can deduce a permutation from this matching: $\pi(i) = i'$ if edge (i, i') is in the matching. As the (MMP) is solved to optimality, permutation π is such that the left hand side of $(pi(\mathcal{X}))$ is maximized with $\mathcal{X} \in \chi(\mathcal{P})$. \square

Example 11. Let $(4, 4, w, v, C)$ be an instance of the (SMKP). Let $\tilde{X} = [[1, 1, 0, 0], [1, 0.7, 0.7, 0], [0.6, 0, 0, 0], [1, 0, 0, 0]]$ be a fractional point and let $\mathcal{P} = \{S_1(\mathcal{P}) = \{1, 3\}, S_2(\mathcal{P}) = \{1, 2\}, S_3(\mathcal{P}) = \{1\}, S_4(\mathcal{P}) = \{1\}\}$ be a pattern with $\text{rank}(\mathcal{P}) = 5$. **Table 1** represents the weight matrix of \mathcal{H} , with $\tilde{X}[i] \in \mathcal{H}_1$ the vertex corresponding to the group i of \tilde{X} . In this example, the optimal solution to

	$\tilde{X}[1]$	$\tilde{X}[2]$	$\tilde{X}[3]$	$\tilde{X}[4]$
$S_1(\mathcal{P})$	1	1.7	0.6	1
$S_2(\mathcal{P})$	2	1.7	0.6	1
$S_3(\mathcal{P})$	1	1	0.6	1
$S_4(\mathcal{P})$	1	1	0.6	1

Table 1: Weight matrix

the (MMP) is: $(\tilde{X}[1], S_2(\mathcal{P}))$, $(\tilde{X}[2], S_1(\mathcal{P}))$, $(\tilde{X}[3], S_3(\mathcal{P}))$, $(\tilde{X}[4], S_4(\mathcal{P}))$. This solution has a value $2 + 1.7 + 0.6 + 1 = 5.3$. The inequality with $\mathcal{X} \in \chi(\mathcal{P})$, ordered with respect to the permutation corresponding to the solution of the (MMP) is:

$$x_{11} + x_{12} + x_{21} + x_{23} + x_{31} + x_{41} \leq 5$$

This inequality cuts the fractional point \tilde{X} as the left hand side of the inequality equals 5.3.

To solve the (MMP), one can use the Hungarian algorithm [10]. Initially the complexity of this algorithm was $\mathcal{O}((|\mathcal{H}_1| + |\mathcal{H}_2|)^4)$, and a more recent version [6] is of complexity $\mathcal{O}((|\mathcal{H}_1| + |\mathcal{H}_2|)^3)$. In the scope of the separation algorithm, $|\mathcal{H}_1| = |\mathcal{H}_2| = N$. Thus, the separation algorithm is polynomial for a pattern, as it is of complexity $\mathcal{O}((2N)^3)$. However the number of patterns can be very large. As the Hungarian algorithm solves the (MMP) to optimality, the inequality built for a given \mathcal{P} is the one with the permutation of the groups maximizing the left hand side of the inequality. Hence, for a given \mathcal{P} , solving the (MMP) produces the most violated inequality associated to \mathcal{P} if such inequality exists.

4.4 Two-phase B&C scheme

The first phase is to generate flexible patterns using **Algorithm 3** as pre-processing. Because of the symmetries of the (SMKP), each flexible pattern encodes an exponential number of pattern inequalities. From **Theorem 3**, there is a lower bound on the dimensions of the faces defined by these inequalities, and they are facet-defining in the case of a flexible pattern with a set of cardinality 1.

The second phase is to use inequalities associated to generated patterns within a B&C framework. However, adding all these inequalities would be counter-productive, as only enumerating them would take too long, hence the need of a separation algorithm. As the patterns have already been generated, the separation algorithm only requires to find the permutation π leading to the most violated inequality for each pattern. Such permutation can be found for a given pattern by solving the (MMP), with a polynomial time algorithm.

5 Experimental results

Results are computed on a single thread of an Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz (Cascade Lake) processor, with 2 CPUs of 24 cores, with Linux as operating system. All algorithms are developed with C++. Version 12.8 of CPLEX is used.

5.1 Instance description

A set of 15 (SMKP) instances are as follows: instances 1 to 5 with $N = 20$ $M = 5$; instances 6 to 10 with $N = 30$ $M = 5$; instances 11 to 15 with $N = 20$ $M = 10$. These 15 instances are selected amongst a pool of hundreds of instances as difficult enough to see the impact of the cuts. The selection criteria is for these instances to take at least 60 seconds to be solved by CPLEX, with all the default options of CPLEX enabled but the cuts. As such, there is enough room in terms of time and number of nodes to see the impact of the pattern inequalities and the CPLEX cuts.

5.2 Pattern generation

As the pattern generating process is random based, for each instance, 10 sets of patterns (P_i for $i \in \{0, \dots, 9\}$) are generated with different seeds, with a time limit of 60 seconds for each generation. Also, an eleventh set P10 is generated with a time limit of 3600 seconds. Pattern sets P0 to P9 represent a more practical case, where a limited time is attributed to the pattern generation. Pattern set P10 represents an ideal case, where a larger number of patterns are generated, to have more room to see the effects of the pattern inequalities to solve the (SMKP). Even if the pattern generation procedure could be parallelized, we used a single thread.

The following metrics are used in **Table 2** to compare the pattern generations for each instance:

- *#iter*: the number of iterations of the generation process
- *#find*: the number of iteration where a flexible pattern is found
- *#patt*: the number of different flexible patterns found
- *%facet*: the proportion of them guaranteed to be pattern-facet, i.e., with a set of cardinality 1

Firstly, one can see that the generation becomes slower on larger instances, thus with the same time limit, fewer iterations can be performed, namely thousands of iterations for instances 1 to 5, whereas around one thousand for instances 6 to 10, and around 300 for instances 11 to 15. Consequently, fewer patterns are found for larger instances. Note that many iterations of the pattern generation procedure fail to produce a flexible pattern in a majority of cases. Indeed, for instances 4, 7, 9, 11, 12, 13, 14 and 15, the ratio *#find*/*#iter* is very small. In fact, around 90% of the iterations do not generate a flexible pattern. Generally speaking, the larger the instances, the higher the proportion of failed iterations.

Moreover, when *#find* is large, most of the flexible patterns are generated multiple times, as *#patt* is much smaller, namely for instances 1, 2, 3, 5, 6 and 8. As the procedure is random based, when *#find* is large, there is a higher chance to generate duplicates. It is especially visible as P10 rarely generated more than 10 times the number of patterns compared to P0 to P9, even if the total time allocated to P10 is 60 times longer.

Results in the following show that *%facet* does not seem to be related to the computational time or the number of pattern cuts. In the case where the pattern cuts lead to a short computational time, it could mean that many of the flexible patterns without a set of cardinality 1 are also pattern-facet. In the case where pattern cuts do not reduce the computational time, it could mean that only inequalities with integer coefficients would be efficient.

5.3 Separation of the pattern inequalities

The separation of the (SMKP) is done within a user cut callback from CPLEX. For a given pattern set P_i , it solves the (MMP) for each pattern in P_i (see **Section 4.3**), hence creating one inequality per pattern, and retains the most violated inequality. Preliminary results shows that this strategy is, on average, more efficient than retaining the first violated inequality, or all the violated inequalities. In the separation, we only consider the inequalities that are violated by at least 0.4 by the fractional point. As shown in the next section, the pattern inequalities are very tight. Hence, this violation

inst	set	#iter	#find	#patt	%facet	inst	set	#iter	#find	#patt	%facet	inst	set	#iter	#find	#patt	%facet
1	P0	5099	3308	263	13	6	P0	1095	515	94	18	11	P0	336	35	25	68
	P1	4075	2591	245	13		P1	1049	480	104	17		P1	330	23	17	59
	P2	3490	2218	210	13		P2	1116	531	110	15		P2	329	20	16	50
	P3	3723	2341	231	15		P3	1131	519	103	19		P3	333	24	17	76
	P4	3655	2379	229	14		P4	1057	508	103	16		P4	324	30	23	65
	P5	3711	2353	234	13		P5	1103	516	98	19		P5	321	15	14	86
	P6	3701	2348	213	15		P6	1071	522	108	15		P6	339	27	19	84
	P7	3628	2337	233	14		P7	1083	531	109	17		P7	316	13	11	90
	P8	3524	2277	225	14		P8	1117	510	108	16		P8	322	28	21	71
	P9	3524	2244	221	14		P9	1102	538	117	17		P9	307	21	17	76
P10	264935	169772	540	15	P10	55745	26346	278	16	P10	18201	1397	265	47			
2	P0	8101	4240	46	85	7	P0	899	23	6	100	12	P0	338	16	11	91
	P1	7877	4117	44	36		P1	890	22	9	100		P1	305	20	16	87
	P2	7999	4250	45	38		P2	858	20	5	100		P2	296	13	8	87
	P3	7812	4081	45	33		P3	880	16	7	100		P3	319	15	10	90
	P4	7674	3978	44	39		P4	851	15	5	100		P4	338	11	7	96
	P5	7664	4025	52	37		P5	883	29	8	100		P5	318	11	6	67
	P6	7658	4046	47	36		P6	870	12	3	100		P6	326	12	11	91
	P7	7643	3911	44	36		P7	875	19	8	100		P7	301	16	11	100
	P8	7641	4004	47	36		P8	925	23	6	100		P8	281	14	10	100
	P9	7681	4053	47	36		P9	894	20	6	100		P9	277	12	6	100
P10	486567	253969	69	37	P10	51518	1174	33	94	P10	17520	755	146	79			
3	P0	8437	7513	52	15	8	P0	1160	824	185	10	13	P0	321	21	14	71
	P1	7951	7041	53	17		P1	1176	845	178	11		P1	318	15	11	64
	P2	8075	7162	53	15		P2	1165	830	185	12		P2	321	20	10	90
	P3	7710	6850	51	16		P3	1133	828	175	11		P3	319	22	17	71
	P4	7946	6990	51	16		P4	1152	829	181	10		P4	323	19	16	62
	P5	7736	6841	52	17		P5	1183	861	181	12		P5	345	21	18	83
	P6	7797	6855	53	15		P6	1169	840	185	11		P6	290	14	11	73
	P7	7766	6896	52	17		P7	1194	857	176	11		P7	277	13	11	73
	P8	7739	6874	54	19		P8	1154	834	174	10		P8	310	22	17	82
	P9	7547	6647	51	18		P9	1174	873	171	13		P9	341	26	19	68
P10	497691	440097	71	20	P10	60280	43425	521	8	P10	17754	1063	187	54			
4	P0	4065	374	71	85	9	P0	806	82	38	100	14	P0	329	10	9	89
	P1	3892	358	69	85		P1	808	77	42	98		P1	334	12	11	100
	P2	4009	391	67	90		P2	771	67	31	97		P2	311	9	7	100
	P3	4074	387	72	85		P3	780	60	26	100		P3	334	6	5	100
	P4	4083	391	72	85		P4	844	90	35	97		P4	349	12	9	100
	P5	4028	363	75	92		P5	795	61	34	91		P5	323	10	7	100
	P6	3941	360	76	87		P6	810	58	37	95		P6	325	10	9	89
	P7	3974	365	76	83		P7	808	68	34	94		P7	311	10	8	75
	P8	4013	370	69	91		P8	796	61	35	97		P8	348	8	8	87
	P9	4071	351	67	91		P9	794	58	32	94		P9	334	12	12	100
P10	215019	19706	312	69	P10	41518	3368	251	83	P10	18661	609	114	86			
5	P0	4327	1222	215	23	10	P0	809	88	51	88	15	P0	308	16	14	79
	P1	3397	950	194	26		P1	781	91	46	91		P1	309	14	12	83
	P2	3212	903	177	32		P2	755	81	46	93		P2	293	13	12	100
	P3	3246	936	202	26		P3	772	89	46	96		P3	288	14	10	90
	P4	4288	1262	220	27		P4	754	82	41	95		P4	317	17	14	100
	P5	3743	1095	211	26		P5	769	101	56	98		P5	304	16	14	79
	P6	3111	897	175	26		P6	791	84	47	94		P6	314	10	10	70
	P7	3160	901	184	30		P7	750	74	46	93		P7	292	17	13	77
	P8	3227	872	191	28		P8	780	99	50	92		P8	292	17	14	83
	P9	3213	876	194	31		P9	775	88	44	91		P9	296	7	7	86
P10	197506	55493	868	18	P10	40633	4825	430	84	P10	17376	763	147	73			

Table 2: Patterns generations

bound serves to filter the less efficient violated inequalities. A too high violation bound would be too selective, and a too small number of inequalities would be added despite the dedicated time to generate them.

This process is repeated at each node, as long as a violated inequality is found. This means that the separation of the (SMKP) is made at least once per node. Recall that it solves a (MMP) for each pattern, and from **Table 2** the number of patterns can reach hundreds. This results to the separation taking up to 90% of the total computational time. To avoid such cases, we only use the separation algorithm while less than 1000 cuts have been added, or less than 10,000 nodes have been explored. These limits cover the case where lots of violated inequalities are found, but also the opposite case

where very little inequalities are found.

5.4 Resolution of the (SMKP)

The B&C framework is limited to a single thread, and a maximum of 3600 seconds of computational time. Multiple combinations are considered in order to compare the introduced inequalities. For this purpose, we define *default CPLEX*, CPLEX with all default options enabled, and *no-cut CPLEX*, which is default CPLEX with the cuts disabled. Three combinations are considered:

- Cplx: default CPLEX
- Psep: no-cut CPLEX with pattern inequalities separation.
- Cplx+Psep: default CPLEX with pattern inequalities separation.

As P10 is always outperformed by P0 to P9, results for P10 are not included in the following. This is due to the fact that P10 contains to many patterns, which increases the number of time the (MMP) is solved, without necessarily producing more efficient cuts. For all these combinations, the results for pattern sets P0 to P9 are quite similar. Hence, the following metrics, used in the **Table 3** to compare the aforementioned combinations, are averaged for P0 to P9 besides #s.

- #s: the number of pattern sets for which the instance is solved with respect to the total number of pattern sets
- C-cuts: the number of CPLEX cuts added
- P-cuts: the number of pattern cuts added
- Pvv: the average violation value of the pattern cuts added
- r-value: the linear relaxation value at the root node
- user-time: the proportion of the computational time dedicated to the separation
- #nodes-s: the number of nodes explored when the instance is solved
- #nodes-us: the number of nodes explored when the instance is unsolved
- gap: the gap when the instance is not solved
- time: the total computational time

The relaxation value r-value resulting from using Psep is often higher than the one obtained using Cplx, instances 1 and 3 being the only two exceptions. This could mean that the pattern inequalities are the most efficient when some branching have already occurred.

Besides instances 8 and 9, using Psep instead of Cplx leads to much smaller computational times. For instance 1, only 77 nodes are required with Psep, while Cplx needs 580 seconds and 3.6 million nodes. The difference is even more visible for instance 3, where Psep solves the (SMKP) in 0 seconds at the root node, while Cplx needs 275 seconds and 1.8 million nodes. In general, Psep is very efficient because of the violation value of the added cuts. Indeed, metric Pvv is very high, over 0.5 in most case, and over 0.9 for instances 3 and 7.

For instances 10 and 13, Psep yields smaller computational times than Cplx even if no inequality is added. This means that the cuts added by CPLEX for these instances are slowing down the LP without significantly cutting the fractional points. This is visible on the number of nodes developed, Psep explores 28.5 million nodes in 2400 seconds, while Cplx explores 8 million nodes in 3600 seconds.

For the majority of the instances, Cplx+Psep is the second best combination. For instance 1, Cplex+Psep requires 124 nodes and 2.5 seconds, while Psep requires 77 nodes and 2.0 seconds, and Cplex requires 3.6 nodes and 580 seconds. It is worth mentioning that for instances 6 and 11, Cplex+Psep seems to be complementary, as they yield the smallest computational time. More specifically, for instance 6 Cplex+Psep can solve the instance at the root node in 0 seconds, with only 4 CPLEX cuts and 1 patern cut, while Cplx did not solve the instance in 3600 seconds with 600 CPLEX cuts, and Psep required 73 seconds with 700 pattern cuts. Also, Cplx+Psep always leads to the smallest root node relaxation value r-value. These results show that the pattern cuts added are different than the CPLEX cuts, and in addition the pattern cuts can help CPLEX finding more efficient inequalities.

inst	cut type	#s	C-cuts	P-cuts	Pvv	r-value	user-time	#nodes-s	#nodes-us	gap	time
1	Cplx	1/1	345	-	-	8125.4	-	3625008	-	-	583.7
	Cplx+Psep	10/10	3	147	0.81	8067.1	96.3 %	124	-	-	2.5
	Psep	10/10	0	155	0.83	8087.8	96.3 %	77	-	-	2.0
2	Cplx	1/1	305	-	-	8498.9	-	1048228	-	-	221.6
	Cplx+Psep	10/10	3	1000	0.56	8426.7	59.0 %	9522	-	-	14.9
	Psep	10/10	0	1000	0.55	8502.4	64.1 %	5730	-	-	12.8
3	Cplx	1/1	405	-	-	5234.6	-	1849854	-	-	275.6
	Cplx+Psep	10/10	5	3	0.76	4795.7	13.9 %	0	-	-	0.0
	Psep	10/10	0	7	0.94	4829.7	27.7 %	0	-	-	0.0
4	Cplx	1/1	405	-	-	6575.9	-	5442220	-	-	2287.3
	Cplx+Psep	10/10	43	782	0.60	6572.3	2.2 %	3367226	-	-	1529.7
	Psep	10/10	0	698	0.63	6576.2	2.8 %	3858694	-	-	1563.4
5	Cplx	1/1	405	-	-	14288.9	-	1816805	-	-	468.2
	Cplx+Psep	10/10	63	567	0.60	14282.0	18.9 %	1246269	-	-	560.6
	Psep	10/10	0	369	0.58	14302.3	35.8 %	1360370	-	-	456.3
6	Cplx	0/1	605	-	-	6672.8	-	-	6230692	0.18 %	-
	Cplx+Psep	10/10	4	1	0.66	6597.6	42.4 %	0	-	-	0.0
	Psep	10/10	0	698	0.57	6677.3	78.0 %	15041	-	-	73.3
7	Cplx	1/1	605	-	-	42275.2	-	1791840	-	-	1495.0
	Cplx+Psep	10/10	337	26	0.59	42273.4	1.2 %	1052887	-	-	671.8
	Psep	10/10	0	30	0.93	42301.2	4.3 %	1131631	-	-	151.3
8	Cplx	1/1	227	-	-	7961.6	-	31200	-	-	7.1
	Cplx+Psep	10/10	31	943	0.50	7961.6	4.9 %	3116656	-	-	2169.9
	Psep	0/10	0	459	0.46	8051.7	4.4 %	-	10232105	0.52 %	-
9	Cplx	1/1	218	-	-	28987.0	-	67547	-	-	15.9
	Cplx+Psep	10/10	363	0	-	28987.0	6.8 %	2256372	-	-	822.4
	Psep	10/10	0	0	-	29030.7	13.3 %	4911185	-	-	406.9
10	Cplx	0/1	605	-	-	28438.1	-	-	7965193	0.21 %	-
	Cplx+Psep	0/10	363	0	-	28438.1	2.4 %	-	8530504	0.26 %	-
	Psep	10/10	0	0	-	28456.6	4.0 %	28518001	-	-	2407.1
11	Cplx	1/1	905	-	-	39387.8	-	1147464	-	-	1323.3
	Cplx+Psep	10/10	362	240	0.53	39281.3	30.3 %	129380	-	-	139.0
	Psep	10/10	0	430	0.63	39391.8	4.5 %	1028021	-	-	377.6
12	Cplx	0/1	770	-	-	43984.0	-	-	3217619	0.05 %	-
	Cplx+Psep	10/10	540	3	0.49	43984.0	0.6 %	3154490	-	-	2374.2
	Psep	10/10	0	2	0.71	44055.1	3.0 %	4509387	-	-	424.0
13	Cplx	1/1	605	-	-	35842.2	-	2358605	-	-	1231.4
	Cplx+Psep	10/10	363	0	-	35842.2	1.6 %	2086498	-	-	1030.5
	Psep	10/10	0	0	-	35856.8	6.0 %	2173793	-	-	240.4
14	Cplx	0/1	560	-	-	27403.7	-	-	7614965	0.13 %	-
	Cplx+Psep	0/10	352	11	0.43	27403.7	0.7 %	-	9892733	0.12 %	-
	Psep	10/10	0	31	0.41	27411.0	1.5 %	15305740	-	-	1485.3
15	Cplx	1/1	519	-	-	41001.8	-	760888	-	-	490.7
	Cplx+Psep	10/10	205	638	0.54	41001.7	1.8 %	804107	-	-	631.2
	Psep	10/10	0	602	0.55	41003.1	4.5 %	792874	-	-	289.5

Table 3: Resolution of the instances 1 to 15 of the (SMKP)

6 Conclusion

In this paper, the (SMKP) is considered as a new variant of the knapsack problem. Two main contributions, namely the polyedral study and the two-phase B&C scheme, are proposed, revolving around the patterns. We derived pattern inequalities as a new class of valid inequalities embedding symmetries with respect to the groups. Necessary facet-defining conditions are defined for these inequalities, which are also sufficient in the case of a pattern with a set of cardinality 1. An algorithm is presented to generate a set of patterns verifying such conditions as pre-processing and first phase of the scheme. A separation algorithm based on the generated patterns is presented and reduces to

match the patterns with the fractional point in order to produce a violated inequality. This algorithm is used within a B&C framework in the second phase of the scheme. Experimental results demonstrate the efficiency of the inequalities and algorithms presented, as they significantly speed up the solving process of difficult instances compared to default CPLEX.

An extension of this work would be to study the integer inequalities for the (SMKP). The proposed patterns can also be extended to other problems, such as the use case of the Hydro Unit Commitment problem whose core structure corresponds to the (SMKP). A promising perspective would be to see how a two phase B&C scheme could be generalized to other problems facing numerous symmetries.

References

- [1] Richard Bellman. “On a routing problem”. In: *Quarterly of applied mathematics* 16.1 (1958), pp. 87–90.
- [2] E.Andrew Boyd. “Polyhedral results for the precedence-constrained knapsack problem”. In: *Discrete Applied Mathematics* 41.3 (1993), pp. 185–201. ISSN: 0166-218X.
- [3] Sebastian Ceria et al. “Cutting planes for integer programs with general integer variables”. In: *Mathematical programming* 81.2 (1998), pp. 201–214.
- [4] Keely L Croxton, Bernard Gendron, and Thomas L Magnanti. “A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems”. In: *Management Science* 49.9 (2003), pp. 1268–1273.
- [5] Edsger W Dijkstra. “A note on two problems in connexion with graphs”. In: *Numerische mathematik* 1.1 (1959), pp. 269–271.
- [6] Jack Edmonds and Richard M Karp. “Theoretical improvements in algorithmic efficiency for network flow problems”. In: *Journal of the ACM (JACM)* 19.2 (1972), pp. 248–264.
- [7] Carlos E Ferreira, Alexander Martin, and Robert Weismantel. “Solving multiple knapsack problems by cutting planes”. In: *SIAM Journal on Optimization* 6.3 (1996), pp. 858–877.
- [8] Grace Hechme-Doukopoulos et al. “The short-term electricity production management problem at EDF”. In: *Optima Newsletter* 84 (Oct. 2010), pp. 2–6.
- [9] Christopher Hojny et al. “Knapsack polytopes: a survey”. In: *Annals of Operations Research* 292.1 (2020), pp. 469–517.
- [10] Harold W Kuhn. “The Hungarian method for the assignment problem”. In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.
- [11] Robert LMJ van de Leensel, CPM Van Hoesel, and JJ Van de Klundert. “Lifting valid inequalities for the precedence constrained knapsack problem”. In: *Mathematical programming* 86.1 (1999), pp. 161–185.
- [12] Kyungchul Park and Sungsoo Park. “Lifting cover inequalities for the precedence-constrained knapsack problem”. In: *Discrete Applied Mathematics* 72.3 (1997), pp. 219–241.
- [13] Yves Pochet and Laurence A Wolsey. “Integer knapsack and flow covers with divisible coefficients: polyhedra, optimization and separation”. In: *Discrete Applied Mathematics* 59.1 (1995), pp. 57–74.
- [14] Mariem Ben Salem et al. “Optimization algorithms for the disjunctively constrained knapsack problem”. In: *Soft Computing* 22.6 (2018), pp. 2025–2043.
- [15] Moshe Zukerman et al. “A polynomially solvable special case of the unbounded knapsack problem”. In: *Operations Research Letters* 29.1 (2001), pp. 13–16.

A Proofs and lemmas

A.1 Proof of Property 7

Proof. Let (i, j) be an item such that $\exists j' > j, x_{ij'} \in \mathcal{X}$. Assume without loss of generality that j' is such that there are no $x_{ij''} \in \mathcal{X}, j < j'' < j'$. As \mathcal{P} is pattern-facet, condition (iii) holds and the property is verified for item $(i, j' - 1)$. Let \mathcal{Y}' be the set of variable verifying condition (iii) for item $(i, j' - 1)$. Let \mathcal{Z} be the subset of \mathcal{Y}' with all variables $x_{ij''} \in \mathcal{Y}', j'' \geq j$. One can build $\mathcal{Y} = \mathcal{Y}' \setminus \mathcal{Z} \cup \{x_{ij}\}$. We can prove that the set of variable \mathcal{Y} verifies the property for item (i, j) . By construction, $x_{ij} \in \mathcal{Y}$ and $x_{ij''} \notin \mathcal{Y}, \forall j'' > j$. As there are no $x_{ij''} \in \mathcal{X}, j < j'' < j'$ and \mathcal{Y}' is a k -intersection of \mathcal{X} by condition (iii), then $|\mathcal{Y} \cap \mathcal{X}| = k$. Also, as the weights are non-negative, and solution $X_{\mathcal{Y}'}$ is valid by condition (iii), then $X_{\mathcal{Y}}$ is also valid. Consequently, \mathcal{Y} is a k -intersection of \mathcal{X} and the property is verified for x_{ij} .

Let (i, j) be an item such that $\forall j' > j, x_{ij'} \notin \mathcal{X}$. As \mathcal{P} is pattern-facet, condition (ii) hold and the property is verified for item (i, M) . The proof is the same as in the first case, with \mathcal{Y}' the set of variable verifying condition (ii) for item (i, M) .

Thus, for any (i, j) , there exist a feasible solution with $x_{ij} = 1, x_{ij+1} = 0$ and a total of k variables of \mathcal{X} to 1. \square

A.2 Proof of Property 8

Proof of Property 8. Let \mathcal{P} be a flexible pattern. Suppose that there is $i \leq N$ such that $S_i(\mathcal{Q}_1)$ does not contain the $|S_i(\mathcal{P})| - V$ smallest indices of $S_i(\mathcal{P})$. By definition of \mathcal{Q}_1 , $S_1(\mathcal{Q}_1) = S_1(\mathcal{P})$, meaning the property is trivially verified for $i = 1$. In the following we consider $i > 1$. Let $\mathcal{X} \in \chi(\mathcal{P})$ with permutation π_{id} . Let $\mathcal{Y} \in \chi(\mathcal{Q}_1)$ with permutation π_{id} , to which we add x_{1M} if $x_{1M} \notin \mathcal{X}$. The solution $X_{\mathcal{Y}}$ is valid as proven in **Lemma 1**. Let \mathcal{Y}' be the variables set \mathcal{Y} to which we remove all variables of group 1, and to which we add all variables of group i in $\mathcal{X} \setminus \mathcal{Y}$. Because of the symmetric weights, selecting every item $(1, j), j \leq M$ is at least as heavy as selecting items $(i, j), j \leq \max(S_i(\mathcal{P}))$. Consequently:

$$\sum_{x_{i'j'} \in \mathcal{Y}'} s_{i'j'}(\mathcal{Y}') \leq \sum_{x_{i'j'} \in \mathcal{Y}} s_{i'j'}(\mathcal{Y})$$

As solution $X_{\mathcal{Y}}$ is valid, solution $X_{\mathcal{Y}'}$ must also be valid. By construction, there are V variables of group 1 in $\mathcal{Y} \setminus \mathcal{Y}'$. Also, as $|S_i(\mathcal{Q}_1)| < |S_i(\mathcal{P})| - V$ by hypothesis, there are at least $V + 1$ variables of group i in $\mathcal{Y}' \setminus \mathcal{Y}$. Besides groups 1 and i , sets \mathcal{Y} and \mathcal{Y}' are identical, hence $|\mathcal{Y}'| > |\mathcal{Y}|$. As \mathcal{Y} is a k -intersection of \mathcal{X} , then \mathcal{Y}' is at least a $k + 1$ -intersection of \mathcal{X} , which contradicts the rank k of \mathcal{P} .

Thus, \mathcal{Q}_1 must contain the $|S_i(\mathcal{P})| - V$ smallest indices of $S_i(\mathcal{P}), \forall i \leq N$. \square

A.3 Lemmas for Property 9

Lemma 2. *Let \mathcal{P} be a flexible pattern. Let $i \in \{2, \dots, N\}$. For any $u \in [1, V]$, if $S_i(\mathcal{P})[u] \in S_i(\mathcal{Q}_1)$ and $|S_i(\mathcal{P})| \geq u + V$ then $S_1(\mathcal{P})[u] \geq S_i(\mathcal{P})[u + V]$.*

The idea of the following proof is illustrated by **Example 12** and **Figure 5**.

Example 12. Let $(4, 5, w, v, C)$ be an instance of the (SMKP). Let a flexible pattern $\mathcal{P} = \{S_1(\mathcal{P}) = \{1, 2\}, S_2(\mathcal{P}) = \{1, 3\}, S_3(\mathcal{P}) = \{1, 2, 3\}, S_4(\mathcal{P}) = \{2, 3, 4, 5\}\}$ of rank 7 and with $V = 2$. Suppose $\mathcal{Q}_1 = \{\{1, 2\}, \{1\}, \emptyset, \{2, 3, 4, 5\}\}$. In this case \mathcal{P} does not follow **Lemma 2**. Indeed, with $u = 2$, there is $S_1(\mathcal{P})[2] = 1 < S_4(\mathcal{P})[2 + 2] = 2$.

Let $\mathcal{X} = \{x_{11}, x_{12}, x_{21}, x_{23}, x_{31}, x_{32}, x_{33}, x_{42}, x_{43}, x_{44}, x_{45}\} \in \chi(\mathcal{P})$ as represented in **Figure 5a** and $\mathcal{Y} = \{x_{11}, x_{12}, x_{31}, x_{42}, x_{43}, x_{44}, x_{45}\} \in \chi(\mathcal{Q}_1)$. As \mathcal{P} is a flexible pattern, **Lemma 1** proves that solution $X_{\mathcal{Y} \cup \{x_{15}\}}$, as represented in **Figure 5b**, is valid. Let $\mathcal{Z} = \{x_{21}, x_{32}\}$, by construction

$|\mathcal{Z}| = u = 2$. Clearly, solution $X_{\mathcal{Y} \setminus \{x_{12}\} \cup \mathcal{Z}}$, as represented in **Figure 5c**, is unfeasible by the rank of \mathcal{P} . Hence, $w_2 + w_1 > w_2 + w_3 + w_4 + w_5$.

As \mathcal{P} is pattern-facet, \mathcal{P} verifies **Property 7**. Hence there is a set \mathcal{Y}' such that $x_{41} \in \mathcal{Y}'$, $x_{4j} \notin \mathcal{Y}' \forall j \in [2, \dots, 5]$ and $|\mathcal{Y}' \cup \mathcal{X}| = 7$ and $X_{\mathcal{Y}'}$ is a valid solution. Suppose in this example $\mathcal{Y}' = \{x_{11}, x_{12}, x_{21}, x_{23}, x_{31}, x_{32}, x_{33}, x_{41}\}$ represented in **Figure 5d**.

Note that $S_4(\mathcal{P})[u+V] - 1 = S_4(\mathcal{P})[4] - 1 = 1$, and by construction, $x_{4j} \notin \mathcal{Y}' \forall j \in [2, \dots, 5]$. Hence, there are $u+V = 4$ variables, namely $x_{21}, x_{23}, x_{32}, x_{33}$, in $\mathcal{Y}' \setminus \mathcal{Y}$. **Property 8** indicates that $\forall i \leq N$, $S_i(\mathcal{P})[V+1] \in S_i(\mathcal{Q}_1)$. Thus, these 4 variables in $\mathcal{Y}' \setminus \mathcal{Y}$ are splitted into a set $\mathcal{O}_2 \subset \mathcal{X}$ of 1 to $V = 2$ variables in a group, and a set $\mathcal{O}_1 \subset \mathcal{X}$ of $u = 2$ to $u+V-1 = 3$ variables in the other groups. In this case, we arbitrary chose $\mathcal{O}_2 = \{x_{32}, x_{33}\}$ and $\mathcal{O}_1 = \{x_{21}, x_{23}\}$. Because both \mathcal{Z} and \mathcal{O}_1 are subsets of $\mathcal{X} \setminus \mathcal{Y}$ and $|\mathcal{Z}| \leq |\mathcal{O}_1|$, then $w_1 + w_2 + w_3 \geq w_1 + w_2 > w_2 + w_3 + w_4 + w_5$. Hence, with $\mathcal{Y}'' = \mathcal{Y}' \setminus \mathcal{O}_1 \cup \{x_{42}, x_{43}, x_{44}, x_{45}\}$ represented in **Figure 5e**, as solution $X_{\mathcal{X}'}$ is valid, then solution $X_{\mathcal{Y}''}$ is also valid. However \mathcal{Y}'' is an 8-intersection of \mathcal{X} , which contradicts the rank of \mathcal{P} .

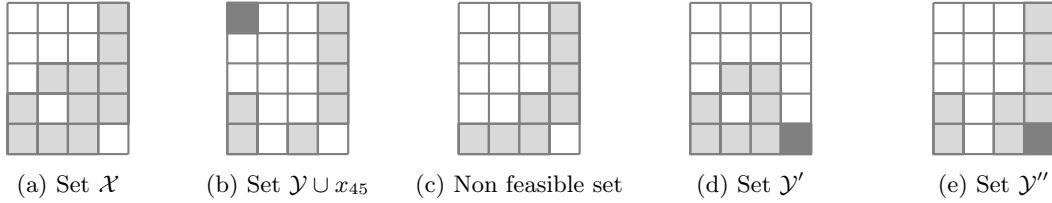


Figure 5: Illustration of **Example 12**

Proof. Let \mathcal{P} be a flexible pattern of rank k . Suppose that there is $u \leq V$ and $i > 1$ such that $S_1(\mathcal{P})[u] \in S_i(\mathcal{Q}_1)$ and $|S_i(\mathcal{P})| \geq u+V$ but $S_1(\mathcal{P})[u] < S_i(\mathcal{P})[u+V]$.

Let $\mathcal{X} \in \chi(\mathcal{P})$ (resp. $\mathcal{Y} \in \chi(\mathcal{Q}_1)$) with permutation π_{id} . By definition of \mathcal{Q}_1 , solution $X_{\mathcal{Y} \cup \{x_{1M}\}}$ is feasible. However, for a variable set \mathcal{Z} , $\mathcal{Z} \subseteq \mathcal{X} \setminus \mathcal{Y}$, it is not possible to create a feasible solution from $X_{\mathcal{Y}}$ by setting every variable of \mathcal{Z} to 1, and $|\mathcal{Z}| - 1$ variables of \mathcal{Y} to 0. Otherwise there would be $k+1$ variables of \mathcal{X} to 1, which contradicts the rank of \mathcal{P} . With $|\mathcal{Z}| = u$, as $X_{\mathcal{Y} \cup \{x_{1M}\}}$ is a feasible solution, we deduce the following:

$$\sum_{x_{i'j'} \in \mathcal{Z}} s_{i'j'}(\mathcal{X}) > \sum_{j=S_1(\mathcal{P})[u]+1}^M w_j$$

Because \mathcal{P} is flexible, **Property 7** stands, it is possible for $j = S_i(\mathcal{P})[u+V]$ to have a set $\mathcal{Y}' \in \mathcal{V}$ a k -intersection of \mathcal{X} with $x_{ij-1} \in \mathcal{Y}'$ and $x_{ij'} \notin \mathcal{Y}'$, $j' \geq j$, such that $X_{\mathcal{Y}'}$ is valid.

In the case $S_i(\mathcal{Q}_1) = S_i(\mathcal{P})$, clearly there are $u+V$ more variables of group i in $\mathcal{Y} \setminus \mathcal{Y}'$. However, as $|\mathcal{Y}| = |\mathcal{Y}'| = k$, there are $u+V$ variables of groups different than i in $\mathcal{Y}' \setminus \mathcal{Y}$. **Property 8** indicates that $\forall i' \leq N$, $|S_{i'}(\mathcal{Q}_1)| \geq |S_{i'}(\mathcal{P})| - V$. Thus, these $u+V$ variables in $\mathcal{Y}' \setminus \mathcal{Y}$ are splitted into a set $\mathcal{O}_2 \subset \mathcal{X}$ of 1 to V variables in a group, and a set $\mathcal{O}_1 \subset \mathcal{X}$ of u to $u+V-1$ variables in the other groups.

In the case $S_i(\mathcal{Q}_1) \subset S_i(\mathcal{P})$ with $S_i(\mathcal{P})[u] \in S_i(\mathcal{Q}_1)$ there are between $V+1$ and $u+V-1$ variables of group i in $\mathcal{Y} \setminus \mathcal{Y}'$ and reciprocally between $V+1$ and $u+V-1$ variables of groups different than i in $\mathcal{Y}' \setminus \mathcal{Y}$. In this case we consider $\mathcal{O}_1 \subset \mathcal{X}$ to be the set of $V+1$ to $u+V-1$ variables if $\mathcal{Y}' \setminus \mathcal{Y}$, and $\mathcal{O}_2 = \emptyset$.

We know that $\mathcal{Z} \in \mathcal{X}$ and $\mathcal{Z} \cup \mathcal{Y} = \emptyset$ and similarly, $\mathcal{O}_1 \in \mathcal{X}$ and $\mathcal{O}_1 \cup \mathcal{Y} = \emptyset$. Also, we know that $|\mathcal{O}_1| \geq u = |\mathcal{Z}|$. We deduce that:

$$\sum_{x_{i'j'} \in \mathcal{O}_1} s_{i'j'}(\mathcal{X}) \geq \sum_{x_{i'j'} \in \mathcal{Z}} s_{i'j'}(\mathcal{X}) > \sum_{j=S_1(\mathcal{P})[u]+1}^M w_j$$

By construction, $S_i(\mathcal{P})[1] \leq M$ and by hypothesis, $S_i(\mathcal{P})[u+V] > S_1(\mathcal{P})[u]$, which means that:

$$\sum_{j \in S_i(\mathcal{P}): j \geq S_i(\mathcal{P})[u+V]} s_{ij}(\mathcal{X}) = \sum_{j=S_j(\mathcal{P})[u+V]}^{S_i(\mathcal{P})[1]} w_j \leq \sum_{j=S_1(\mathcal{P})[u]+1}^M w_j$$

The $u+V$ variables of group i that are in \mathcal{Y} but not in \mathcal{Y}' are lighter than the ones in \mathcal{Z} , thus lighter than variables of \mathcal{O}_1 . Hence we build $\mathcal{Y}'' = \mathcal{Y}' \setminus \mathcal{O}_1$ to which we add all the $u+V$ variables of group i in $\mathcal{X} \setminus \mathcal{Y}'$. As $X_{\mathcal{Y}'}$ is valid, then $X_{\mathcal{Y}''}$ is also valid. However $|\mathcal{O}_1| \leq u+V-1$, meaning that the new solution has at least $k+1$ variables of \mathcal{X} to 1, which contradicts the rank of \mathcal{P} . \square

Lemma 3. *Let \mathcal{P} be a flexible pattern. Let $i \in \{2, \dots, N\}$. For any $u \in [1, V]$, if $S_i(\mathcal{P})[u] \notin S_i(\mathcal{Q}_1)$ and $|S_i(\mathcal{P})| \geq u+V$ then $S_1(\mathcal{P})[u] \geq S_i(\mathcal{P})[u+V]$.*

Proof. Suppose that there is $u \leq V$ and $i > 1$ such that $S_i(\mathcal{P})[u] \notin S_i(\mathcal{Q}_1)$ and $|S_i(\mathcal{P})| \geq u+V$ but $S_1(\mathcal{P})[u] < S_i(\mathcal{P})[u+V]$.

Let $\mathcal{X} \in \chi(\mathcal{P})$ (resp. $\mathcal{Y} \in \chi(\mathcal{Q}_1)$) with permutation π_{id} . Because \mathcal{P} is flexible, **Lemma 1** proves that solution $X_{\mathcal{Y} \cup \{x_{1M}\}}$ is valid. By construction, $S_i(\mathcal{P})[1] \leq M$ and $S_i(\mathcal{P})[V+1] \geq S_i(\mathcal{P})[u+V]$ and by hypothesis $S_i(\mathcal{P})[u+V] > S_1(\mathcal{P})[u]$, which means that:

$$\sum_{j=S_i(\mathcal{P})[V+1]+1}^{S_i(\mathcal{P})[1]} w_j \leq \sum_{j=S_1(\mathcal{P})[u]+1}^M w_j$$

Hence, one can build $\mathcal{Y}' = \mathcal{Y} \cup \{x_{1M}\}$ to which we remove all variables $x_{1j} \in \mathcal{X}$, $j > S_1(\mathcal{P})[u]$ and to which we add all variables for group i : $x_{ij} \in \mathcal{X} \setminus \mathcal{Y}$, with solution $X_{\mathcal{Y}'}$ being valid. However, there are $u-1$ variables $x_{1j} \in \mathcal{X}$ such that $j > S_1(\mathcal{P})[u]$, and at least u variables $x_{ij} \in \mathcal{X} \setminus \mathcal{Y}$ as $S_i(\mathcal{P})[u] \notin S_i(\mathcal{Q}_1)$. As \mathcal{Y} is a k -intersection of \mathcal{X} , then \mathcal{Y}' is at least a $k+1$ -intersection of \mathcal{X} which contradicts the rank of \mathcal{P} . \square

A.4 Lemmas for Property 10

Lemma 4. *Let \mathcal{P} be a flexible pattern of rank k . Let \mathcal{Q}_i be a lower sub-patterns of \mathcal{P} with $i > 1$. If $|S_1(\mathcal{Q}_1)| + |S_i(\mathcal{Q}_1)| = |S_1(\mathcal{Q}_i)| + |S_i(\mathcal{Q}_i)|$ and $|S_i(\mathcal{P})| \geq 2V$, then it is possible for \mathcal{Q}_i to be such that $\forall i' \leq N$ with $|S_{i'}(\mathcal{P})| \geq 2V$, $S_{i'}(\mathcal{P})[2V] \in S_{i'}(\mathcal{Q}_i)$*

Proof. We are in the case $|S_1(\mathcal{Q}_1)| + |S_i(\mathcal{Q}_1)| = |S_1(\mathcal{Q}_i)| + |S_i(\mathcal{Q}_i)|$. Let $\mathcal{L} = \{1, \dots, N\} \setminus \{1, i\}$. By definition, $\text{card}(\mathcal{Q}_1) = \text{card}(\mathcal{Q}_i) = k$ hence we deduce the following equation:

$$\sum_{i' \in \mathcal{L}} |S_{i'}(\mathcal{Q}_1)| = \sum_{i' \in \mathcal{L}} |S_{i'}(\mathcal{Q}_i)|$$

By definition \mathcal{Q}_1 and \mathcal{Q}_i minimize the weight of their respective variable sets. Hence they have both the exact same weight for their variables sets restricted to indices in \mathcal{L} , otherwise it is clear that one of them do not minimize the weight of its variable sets. Consequently, one can modify the sets of index in \mathcal{L} in \mathcal{Q}_i by the ones in \mathcal{Q}_1 .

Property 8 indicates that $S_{i'}(\mathcal{P})[V+1] \in S_{i'}(\mathcal{Q}_1)$. As \mathcal{P} is flexible pattern, $V \geq 1$ by condition (i), and $V+1 \geq 2V$. Clearly the Lemma is verified for \mathcal{Q}_1 .

As $S_{i'}(\mathcal{Q}_1) = S_{i'}(\mathcal{Q}_i)$ for any $i' \in \mathcal{L}$, the Lemma is also verified for any $S_{i'}(\mathcal{Q}_i)$ with $i' \in \mathcal{L}$. By definition of \mathcal{Q}_i , $S_i(\mathcal{Q}_i) = S_i(\mathcal{P})$, trivially verifying the Lemma. As $|S_1(\mathcal{P})| = V < 2V$, the Lemma does not concern $S_1(\mathcal{Q}_i)$. Thus, the Lemma is verified for any set $S_{i'}(\mathcal{Q}_i)$ for which $|S_{i'}(\mathcal{Q}_i)| \geq 2V$. \square

Lemma 5. *Let \mathcal{P} be a flexible pattern of rank k . Let \mathcal{Q}_i be a lower sub-patterns of \mathcal{P} with $i > 1$. If $|S_1(\mathcal{Q}_i)| = 0$, then $\forall i' \leq N$ with $|S_{i'}(\mathcal{P})| \geq 2V$, $S_{i'}(\mathcal{P})[2V] \in S_{i'}(\mathcal{Q}_i)$*

The idea of the following proof is illustrated by **Example 13** and **Figure 6**.

Example 13. Let $(5, 3, w, v, C)$ be an instance of the (SMKP). Let a flexible pattern $\mathcal{P} = \{S_1 = \{3\}, S_2 = \{1, 2, 3\}, S_3 = \{1, 2, 3\}, S_4 = \{1, 2, 3\}, S_5 = \{1, 2, 3\}\}$ of rank 10. The smallest set of \mathcal{P} is $S_1(\mathcal{P})$ hence $V = 1$. Suppose $\mathcal{Q}_1 = \{S_1 = \{3\}, S_2 = \{1, 2, 3\}, S_3 = \{1, 2\}, S_4 = \{1, 2\}, S_5 = \{1, 2\}\}$ and $\mathcal{Q}_2 = \{S_1 = \emptyset, S_2 = \{1, 2, 3\}, S_3 = \{1\}, S_4 = \{1, 2, 3\}, S_5 = \{1, 2, 3\}\}$. We are in the case of **Lemma 5** as $|S_1(\mathcal{Q}_2)| = 0$, but not in the case of **Lemma 4**, as $|S_1(\mathcal{Q}_1)| + |S_2(\mathcal{Q}_1)| = 4 \neq |S_1(\mathcal{Q}_2)| + |S_2(\mathcal{Q}_2)| = 3$. However, **Lemma 5** is not verified, as $S_3(\mathcal{P})[2V] = S_3(\mathcal{P})[2] = 2 \notin S_3(\mathcal{Q}_2)$.

Let $\mathcal{Y} = \{x_{13}, x_{21}, x_{22}, x_{23}, x_{31}, x_{32}, x_{41}, x_{42}, x_{51}, x_{52}\} \in \chi(\mathcal{Q}_1)$ and $\mathcal{Y}' = \{x_{21}, x_{22}, x_{23}, x_{31}, x_{41}, x_{42}, x_{43}, x_{51}, x_{52}, x_{53}\} \in \chi(\mathcal{Q}_2)$. As \mathcal{P} is pattern-flexible, **Lemma 1** proves that solution $X_{\mathcal{Y} \cup \{x_{14}\}}$ is valid. Also, \mathcal{Q}_2 minimizes the sum of the weights, $\{x_{43}, x_{53}\} \in \mathcal{Y}'$ and $\{x_{32}, x_{33}\} \notin \mathcal{Y}'$ hence $w_3 + w_3 \leq w_2 + w_3$. Clearly, $w_3 + w_3 \leq w_2 + w_3 \leq w_1 + w_2 + w_3 + w_4$. With $\mathcal{O} = \mathcal{Y}' \setminus \mathcal{Y} = \{x_{43}, x_{53}\}$, the set $\mathcal{Y}'' = \mathcal{Y} \cup \{x_{14}\} \setminus \{x_{13}, x_{14}\} \cup \mathcal{O}$ yields a feasible solution $X_{\mathcal{Y}''}$. However, \mathcal{Y}'' is a 9-intersection of \mathcal{X} , which contradicts the rank of \mathcal{P} .

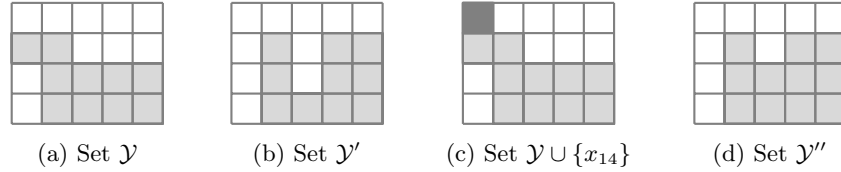


Figure 6: Illustration of **Example 13**

Proof. Because of the shape of \mathcal{Q}_1 (see **Property 8**), $|S_1(\mathcal{Q}_1)| = V$ meaning that if $|S_i(\mathcal{Q}_1)| = |S_i(\mathcal{P})| - V$, then $|S_i(\mathcal{Q}_1)| + |S_1(\mathcal{Q}_1)| = |S_i(\mathcal{P})|$. Because we are in the case $|S_1(\mathcal{Q}_i)| = 0$, then $|S_i(\mathcal{Q}_i)| + |S_1(\mathcal{Q}_i)| = |S_i(\mathcal{Q}_1)| + |S_1(\mathcal{Q}_1)|$ which is verified by **Lemma 4**.

Hence, we consider $|S_i(\mathcal{Q}_1)| \geq |S_i(\mathcal{P})| - V + 1$. Consequently, $|S_i(\mathcal{Q}_1)| + |S_1(\mathcal{Q}_1)| \geq |S_i(\mathcal{Q}_i)| + |S_1(\mathcal{Q}_i)| + 1$. Suppose there is $i' \neq 1, i' \neq i$ such that $S_{i'}(\mathcal{P})[2V] \notin S_{i'}(\mathcal{Q}_i)$. With $v \in [0, |S_{i'}(\mathcal{P})| - 2V]$, we have $|S_{i'}(\mathcal{Q}_i)| = |S_{i'}(\mathcal{P})| - 2V - v$, i.e., $S_{i'}(\mathcal{P})[2V + v] \notin S_{i'}(\mathcal{Q}_i)$. We know by the shape of \mathcal{Q}_1 that $|S_{i'}(\mathcal{Q}_1)| \geq |S_{i'}(\mathcal{P})| - V$ and thus $|S_{i'}(\mathcal{Q}_1)| \geq |S_{i'}(\mathcal{Q}_i)| + V + v$. This leads to the following result:

$$|S_1(\mathcal{Q}_1)| + |S_i(\mathcal{Q}_1)| + |S_{i'}(\mathcal{Q}_1)| - |S_1(\mathcal{Q}_i)| - |S_i(\mathcal{Q}_i)| - |S_{i'}(\mathcal{Q}_i)| \geq V + v + 1$$

There must be a set of groups L , $1 \notin L$, $i \notin L$, $i' \notin L$ such that:

$$V + v + 1 \leq \sum_{l \in L} |S_l(\mathcal{Q}_i)| - |S_l(\mathcal{Q}_1)| \leq 2V + v$$

If not, then there is a set l such that $|S_l(\mathcal{Q}_i)| - |S_l(\mathcal{Q}_1)| > V + 1$, which contradicts the shape of \mathcal{Q}_1 (see **Property 8**) which is that $|S_l(\mathcal{Q}_1)| \geq |S_l(\mathcal{P})| - V$.

Let $\mathcal{X} \in \chi(\mathcal{P})$ (resp. $\mathcal{Y} \in \chi(\mathcal{Q}_1)$, $\mathcal{Y}' \in \chi(\mathcal{Q}_i)$) with permutation π_{id} . Let \mathcal{O} be the set of variables $x_{lj} \in \mathcal{Y}' \setminus \mathcal{Y}$ with $l \in L$. Note that $V + v + 1 \leq |\mathcal{O}| \leq 2V + v$. Also, $S_{i'}(\mathcal{P})[2V + v] \notin S_{i'}(\mathcal{Q}_i)$, \mathcal{O} is sub-pattern of \mathcal{Q}_i and by definition \mathcal{Q}_i minimizes the sum of the weights and \mathcal{O} is a sub-pattern of \mathcal{Q}_i , hence we deduce:

$$\sum_{x_{lj} \in \mathcal{O}} s_{lj} \leq \sum_{j=S_{i'}(\mathcal{P})[2V+v+1]+1}^{S_{i'}(\mathcal{P})[1]} w_j \leq \sum_{j=S_{i'}(\mathcal{P})[2V+v+1]+1}^M w_j \leq \sum_{j=1}^M w_j$$

As \mathcal{P} is flexible pattern, **Lemma 1** provides a feasible solution $X_{\mathcal{Y} \cup \{x_{1M}\}}$. One could create a new feasible solution from $X_{\mathcal{Y} \cup \{x_{1M}\}}$ by setting all variables of group 1 to 0, and all variables of \mathcal{O} to one, creating a lighter solution, thus valid. However, $|\mathcal{O}| \geq V + v + 1 \geq V + 1 > |S_1(\mathcal{Q}_1)| = V$. The new solution has at least $k + 1$ variables of \mathcal{X} to 1, which contradicts the rank of \mathcal{P} . \square

Lemma 6. *Let \mathcal{P} be a flexible pattern of rank k . Let \mathcal{Q}_i be a lower sub-patterns of \mathcal{P} with $i > 1$. If $|S_1(\mathcal{Q}_i)| > 0$, then $\forall i' \leq N$ with $|S_{i'}(\mathcal{P})| \geq 2V$, $S_{i'}(\mathcal{P})[2V] \in S_{i'}(\mathcal{Q}_i)$*

Proof. We consider the case where $|S_1(\mathcal{Q}_1)| + |S_i(\mathcal{Q}_1)| \neq |S_1(\mathcal{Q}_i)| + |S_i(\mathcal{Q}_i)|$, as otherwise it is the case of **Lemma 4**. Suppose there is $i' \neq 1, i' \neq i$ such that $S_{i'}(\mathcal{P})[2V] \notin S_{i'}(\mathcal{Q}_i)$. Let $v \in [0, |S_{i'}(\mathcal{P})| - 2V]$ such that $|S_{i'}(\mathcal{Q}_i)| = |S_{i'}(\mathcal{P})| - 2V - v$. Let u be the smallest value such that $S_1(\mathcal{P})[u] \in S_1(\mathcal{Q}_i)$, i.e., $u = V - |S_1(\mathcal{Q}_i)| + 1$. Let \mathcal{X} (resp. \mathcal{Y}) be the set of variables of \mathcal{P} (resp. \mathcal{Q}_i) with permutation π_{id} . We know that $X_{\mathcal{Y}}$ is a feasible solution. We also know by **Property 9** that a flexible pattern \mathcal{P} is such that $S_1(\mathcal{P})[u] \geq S_{i'}(\mathcal{P})[u + V]$, which means that:

$$\sum_{j' \leq S_{i'}(\mathcal{P})[u+V]} s_{i'j'}(\mathcal{X}) \leq \sum_{j \leq S_1(\mathcal{P})[u]} s_{1j}(\mathcal{X})$$

Hence, one can build $\mathcal{Y}' = \mathcal{Y}$ from which we remove all variables $x_{1j} \in \mathcal{Y}$ and to which we add all $x_{i'j'} \in \mathcal{X}, j' \leq S_{i'}(\mathcal{P})[u + V]$. Clearly, as $X_{\mathcal{Y}}$ is a valid solution, so is $X_{\mathcal{Y}'}$. In group i' , by hypothesis there are $|S_{i'}(\mathcal{P})| - 2V - v$ variables in \mathcal{Y} , and by construction $|S_{i'}(\mathcal{P})| - (V + u) + 1$ in \mathcal{Y}' . Hence in group i' , there are $V + v - u + 1$ variables in $\mathcal{Y}' \setminus \mathcal{Y}$. In group 1, by hypothesis there are $V - u + 1$ variables in \mathcal{Y} , and by construction 0 variables in \mathcal{Y}' . For groups different than i' and 1 variables sets \mathcal{Y} and \mathcal{Y}' are identical by construction. We deduce $|\mathcal{Y}'| - |\mathcal{Y}| = V + v - u + 1 - (V - u + 1) = v \geq 0$.

In the case $v \geq 1$, as \mathcal{Y} is a k -intersection of \mathcal{X} , then \mathcal{Y}' is at least a $k + 1$ -intersection of \mathcal{X} , which contradicts the rank of \mathcal{P} .

In the case $v = 0$, by construction \mathcal{Y} and \mathcal{Y}' contain the same variables for group i . Also by construction, the weight of \mathcal{Y}' is lighter or equals the weight of \mathcal{Y} . Clearly, there is a \mathcal{Q}'_i such that $\mathcal{Y}' \in \chi(\mathcal{Q}'_i)$. However, by construction $S_1(\mathcal{Q}'_i)$ is empty, which cannot be possible as proven by **Lemma 5**. \square

A.5 Proof of Theorem 5

The following Lemma aims to use condition (iv) to provide feasible solutions for any variable in $\mathcal{X} \in \chi(\mathcal{P})$. The idea is for these solutions to have at least all variables of $\mathcal{Y}_{U'}$ to 1, with $\mathcal{Y}_{U'} \in \chi(\mathcal{R}_{U'})$. As these solutions have many variables to 1 in common, this will be convenient to prove them to be linearly independent in **Theorem 5**.

Lemma 7. *If the nested sub-patterns $\{\mathcal{R}_u, 1 \leq u \leq U'\}$ verify (iv), then for any $x_{ij} \in \mathcal{X} \setminus \mathcal{Y}_1, \exists u \leq U'$ such that there is a set \mathcal{Z} containing u variables $x_{ij'} \in \mathcal{X} \setminus \mathcal{Y}_u, j' \leq j$ and $\mathcal{Z} \cup \mathcal{Y}_u$ is a k -intersection of \mathcal{X} .*

Proof. Let \mathcal{P} be a pattern verifying (iv). Let $\mathcal{X} \in \chi(\mathcal{P})$ and $\mathcal{Y}_u \in \chi(\mathcal{R}_u), \forall u \leq U'$.

Firstly, by definition of \mathcal{R}_1 , there is a feasible solution with all variables of \mathcal{Y}_1 to 1, and $x_{ij} = 1$, with $x_{ij} \in \mathcal{X} \setminus \mathcal{Y}_u, x_{ij'} \notin \mathcal{X} \setminus \mathcal{Y}_u, j' < j$. Hence the Lemma is verified $\forall x_{ij} \in \mathcal{X} \setminus \mathcal{Y}_1$ such that $x_{ij'} \notin \mathcal{X} \setminus \mathcal{Y}_1, j' < j$.

Secondly, we can define a recursive rule. Let x_{ij} be a variable such that $x_{ij} \in \mathcal{X} \setminus \mathcal{Y}_1$. Suppose for $x_{ij'}$ the Lemma is verified for \mathcal{R}_u , with j' such that $x_{ij''} \notin \mathcal{X}, j' < j'' < j$. In other word, there is a feasible solution with $x_{ij'}$ to 1 and all variables of \mathcal{Y}_u to 1. Note that $x_{ij} = 0$ in this solution. By definition of the nested sub-pattern \mathcal{R}_{u+1} , there is a feasible solution with all variables of \mathcal{Y}_{u+1} and the $u + 1$ variables $x_{ij'} \in \mathcal{X} \setminus \mathcal{Y}_u, j' \leq j$. We distinguish two cases:

The first case is $|S_i(\mathcal{R}_{u+1})| = |S_i(\mathcal{R}_u)|$. By hypothesis, there is a \mathcal{Z} containing u variables of group i from $\mathcal{X} \setminus \mathcal{Y}_u$, with $x_{ij'} \in \mathcal{Z}, x_{ij} \notin \mathcal{Z}$ such that $\mathcal{Z} \cup \mathcal{Y}_u$ is a k -intersection of \mathcal{X} . As (iv) holds, then clearly there is \mathcal{Z}' containing $u + 1$ variables of group i from $\mathcal{X} \setminus \mathcal{Y}_{u+1}$, such that $\mathcal{Z}' \cup \mathcal{Y}_{u+1}$ is a k -intersection of \mathcal{X} . As $x_{ij''} \notin \mathcal{X}, j' < j'' < j$, then $x_{ij} \in \mathcal{Z}'$ and $x_{ij''} \notin \mathcal{Z}$ with $j'' \geq j$. Hence the Lemma is verified for x_{ij} .

The second case is $|S_i(\mathcal{R}_{u+1})| = |S_i(\mathcal{R}_u)| - 1$. In which case, with \mathcal{Z}' containing $u + 1$ variables of group i from $\mathcal{X} \setminus \mathcal{Y}_{u+1}, \mathcal{Z} \cup \mathcal{Y}_u = \mathcal{Z}' \cup \mathcal{Y}_{u+1}$. Consequently $x_{ij} \notin \mathcal{Z}$.

From these two cases, we deduce that if the Lemma is verified for x_{ij} , it is verified $\forall x_{ij'} \in \mathcal{X} \setminus \mathcal{Y}_1$, with $j' \leq j$.

Finally, by definition of $\mathcal{R}_{U'}$, there is at most U' variables in $S_i(\mathcal{P}) \setminus S_i(\mathcal{R}_{U'})$. Hence this Lemma is necessarily verified $\forall x_{ij} \in \mathcal{X}$ such that $x_{ij'} \notin \mathcal{X}$, $j' > j$.

Because of the initialisation with \mathcal{Y}_1 , the recursive rule between \mathcal{Y}_u and \mathcal{Y}_{u+1} , and because $\forall x_{ij} \in \mathcal{X}$, $x_{ij'} \notin \mathcal{X}$, $j' > j$ the Lemma is verified, then the Lemma is verified $\forall x_{ij} \in \mathcal{X} \setminus \mathcal{R}_1$. \square

Proof. Proof of **Theorem 5** Let $\mathcal{X} \in \chi(\mathcal{P})$ and $\mathcal{Y}_u \in \chi(\mathcal{R}_u)$ with permutation π_{id} .

The points will be enumerated iteratively.

First, from **Lemma 7**, we know that $\forall x_{ij} \in \mathcal{X} \setminus \mathcal{Y}_1$, $x_{ij'} \notin \mathcal{X} \setminus \mathcal{Y}_1$, $j' < j$, there is a feasible solution with $x_{ij} = 1$ and with $x_{ij+1} = 0$, with all variables of \mathcal{Y}_1 to 1. These solutions are all linearly independent, as for each x_{ij} considered, it is the only solution with $x_{ij} = 1$.

As condition (v) holds, there is a feasible solution with all variables of \mathcal{Y}_2 to 1, and two variables $x_{ij}, x_{i'j'} \notin \mathcal{Y}_1$, $i \neq i'$ to 1. This solution is linearly independent to the previously mentioned, as it is the only one with a variable of \mathcal{Y}_1 to 0.

From **Lemma 7**, we know that there is a feasible solution with \mathcal{Y}_2 and two variables $x_{ij}, x_{ij'} \in \mathcal{X} \setminus \mathcal{Y}_2$, $j' \neq j$. For each solution where both x_{ij} and $x_{ij'}$ are not in \mathcal{Y}_1 is linearly independent to the others, as it is the only one with $x_{ij'} = 1$.

One can keep enumerating points with the same process. With **Lemma 7**, $|\mathcal{X}| - |\mathcal{Y}_1|$ linearly independent points are generated. With condition (v), one linearly independent point is generated $\forall 2 \leq u < U'$, which is a total of $U' - 1$ linearly independent points.

With condition (vi), for each $x_{ij} \in \mathcal{Y}_{U'}$, there is a set \mathcal{Z} containing all variables $x_{ij'} \in \mathcal{Y}_{U'}$, $j' < j$ and all variables $x_{i'j'} \in \mathcal{Y}_{U'}$, $i' \neq i$, without x_{ij} and with \mathcal{Z} being a k -intersection of \mathcal{X} . Hence, starting with greater j , each new solution is the first one with $x_{ij} = 0$, being linearly independent to the others. With condition (vi), one new feasible solution is created $\forall x_{ij} \in \mathcal{Y}_{U'}$, meaning $|\mathcal{Y}_{U'}|$ new linearly independent points.

A total of $|\mathcal{X}| - |\mathcal{Y}_1| + U' - 1 + |\mathcal{Y}_{U'}| = \text{card}(\mathcal{P}) - (k-1) + U' - 1 + (k-U')$ linearly independent points are generated, i.e., $\text{card}(\mathcal{P})$ linearly independent points. Moreover, from **Theorem 3**, $\forall x_{ij} \notin \mathcal{X}$, there is a feasible solution with $x_{ij} = 1$ and $x_{ij+1} = 0$ that verifies $(pi(\mathcal{X}))$ to equality. Sequentially adding these points associated to their corresponding solutions to our pool of $\text{card}(\mathcal{P})$ points still keeps them linearly independent, as there are the only ones with $x_{ij} = 1$ and $x_{ij+1} = 0$ with $x_{ij} \notin \mathcal{X}$. As there are $n - \text{card}(\mathcal{P})$ of these new points, there is a total of n linearly independent points. \square