Dantzig-Wolfe Reformulation

\{x \in \mathbb{Z}_+^n \mid A x \leq b\} \cap \{x \in \mathbb{R}_+^n \mid C x \leq d\}

Author: Fabio Furini

• Book: Column Generation, Guy Desaulniers, Marius M. Solomon, Jacques Desrosiers, Springer US

Contents

1 Introduction 3

2 Dantzig-Wolfe Reformulation of Mixed Integer Linear Programs 3
  2.1 Convexification ................................. 6
  2.2 Discretization ................................. 7

3 DW reformulation for the Bin Packing Problem 12
  3.1 Column Generation for the BPP ............................. 16
  3.2 Stabilization Techniques for the BPP ............................. 18
4 Branch and Price Algorithm 22
  4.1 Branching rule for the Bin Packing Problem 22

5 DW reformulation for the Vertex Coloring Problem 25
  5.1 Column Generation for the VCP 26
1 Introduction

“Divide et Impera” in Mathematical Optimization

• Break up complex problems into a series of easier problems, solved in cascade!

• Decompose and Reformulate Mixed Integer Linear and Nonlinear Programs.

• Hard Combinatorial Optimization Problems, once decomposed and reformulated, become easier to tackle.

• Find effective decompositions and reformulations!
  – Dantzig-Wolfe Decomposition
  – Benders Decomposition
  – Lagrangian Relaxation
  – Dynamic Programming
  – ...

2 Dantzig-Wolfe Reformulation of Mixed Integer Linear Programs

Given a ILP with \( n \) integer variables – Convexification of \( Dx \leq e \):

\[
\begin{align*}
  Z(\text{ILP}) &= \max c^\top x & \quad (1) \\
  Ax &\leq b & \quad (2) \\
  Dx &\leq e & \quad (3) \\
  x &\in Z_+ & \quad (4)
\end{align*}
\]

Minkowski-Weyl Theorem. Every polyhedron can be represented:

• by outer descriptions (intersection of finitely many affine halfspaces)

• by inner descriptions (Minkowski sum of a polytope\(^1\) and a finitely generated cone\(^2\))

\(^1\) convex combination
\(^2\) conic combination of finitely many vectors
the polyhedron \( P = \{ x : Dx \leq e \} \) can be then expressed as:

\[
P = \left\{ x : x = \sum_{p \in EP} x_p \cdot \lambda_p + \sum_{r \in ER} x_r \cdot \lambda_r, \sum_{p \in EP} \lambda_p = 1, \lambda_p \geq 0, p \in EP, \lambda_r \geq 0, r \in ER \right\}
\]

where \( p \in EP \) are the extreme points and \( r \in ER \) are the extreme rays of \( P \) and \( x_p \) (\( x_r \)) are the values of the extreme points (extreme rays) \( \rightarrow \) one constraint per variable with the associated value in the extreme points (rays).

- In case of a bounded polyhedron (polytope) there are only extreme points (Minkowski sum of a polytope) \( \rightarrow \) we focus on this case but the theory is valid in general

the polyhedron \( P = \{ x : Dx \leq e \} \) can be then expressed as:

\[
P = \left\{ x : x = \sum_{p \in EP} x_p \cdot \lambda_p, \sum_{p \in EP} \lambda_p = 1, \lambda_p \geq 0, p \in EP \right\}
\]

where \( p \in EP \) are the extreme points of \( P \).

### Extended Formulation

\[
Q = \{ (x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^\ell : Ax + L\lambda \leq b \}
\]

is called an extended formulation of a polyhedron \( P \subseteq \mathbb{R}^n \) if

\[
P = \text{proj}_x(Q) := \{ x \in \mathbb{R}^n | \exists \lambda \in \mathbb{R}^\ell : (x, \lambda) \in Q \}
\]

\( Q \) is an extended formulation of the integer set \( X \) if

\[
X = \text{proj}_x(Q) \cup \mathbb{Z}^n_+
\]

The DW reformulation is an extended formulation. This is of interest in combinatorial optimization, since extended formulations are typically stronger than the original.

Example:

\[
A = \begin{bmatrix} -3 & 2 \end{bmatrix} \quad b = \begin{bmatrix} 4 \end{bmatrix} \quad D = \begin{bmatrix} 50 & 31 \end{bmatrix} \quad e = \begin{bmatrix} 250 \end{bmatrix} \quad c = \begin{bmatrix} 1 \\ 0.64 \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}
\]

\[
Z(\text{ILP}) = \max \quad x_1 + 0.64x_2 \\
-3x_1 + 2x_2 \leq 4 \\
50x_1 + 31x_2 \leq 250 \\
x_1, x_2 \in \mathbb{Z}_+
\]
In this simple case we can enumerate the \textbf{(integer)} three extreme points of

$$50x_1 + 31x_2 \leq 250, \quad x_1, x_2 \in \mathbb{Z}_+ \quad \rightarrow \quad p_1 = (0, 0), \quad p_2 = (5, 0), \quad p_3 = (0, 8),$$

(normally there are an exponential number of them) and the facets are

$$\frac{8}{5} x_1 + x_2 \leq 8, \quad x_1 \geq 0, \quad x_2 \geq 0$$

The DW-reformulation is then:

$$Z(DW) = \max \quad x_1 + 0.64x_2$$

$$- 3x_1 + 2x_2 \leq 4$$

$$0 \cdot \lambda_{p_1} + 5 \cdot \lambda_{p_2} + 0 \cdot \lambda_{p_3} = x_1$$

$$0 \cdot \lambda_{p_1} + 0 \cdot \lambda_{p_2} + 8 \cdot \lambda_{p_3} = x_2$$

$$\lambda_{p_1} + \lambda_{p_2} + \lambda_{p_3} = 1$$

$$x_1, x_2 \in \mathbb{Z}_+$$

$$\lambda_{p_1}, \lambda_{p_2}, \lambda_{p_3} \geq 0$$

Now by relaxing the integrality constraints on the variable, we obtain the following optimal solution value $Z^* = \frac{3932}{775} \approx 5.0735$ (stronger bound) and the corresponding optimal solution:
\[ x_1^* = \frac{60}{31} \approx 1.9355, \ x_2^* = \frac{152}{31} \approx 4.9032, \ \lambda_{p_1}^* = 0, \ \lambda_{p_2}^* = \frac{12}{31}, \ \lambda_{p_3}^* = \frac{19}{31} \]

There are two different DW reformulations of an ILP:

\[
\begin{align*}
Z(\text{ILP}) &= \max c^\top x \\
\text{Ax} &\leq b \\
x &\in X
\end{align*}
\]  \hfill (5)

where the set of constraints to reformulate is:

\[
X = \{ x \in \mathbb{Z}^n_+ : \text{Dx} \leq e \}
\]

### 2.1 Convexification

Reformulation of \( \text{conv}(X) \):

\[
\begin{align*}
Z(\text{DW}_C) &= \max \sum_{p \in EP} c_p \cdot \lambda_p \\
\sum_{p \in EP} a_p \cdot \lambda_p &\leq b \\
\sum_{p \in EP} \lambda_p &= 1 \\
\lambda &\geq 0 \\
x &= \sum_{p \in EP} x_p \cdot \lambda_p \\
x &\in \mathbb{Z}^n_+
\end{align*}
\]  \hfill (8-13)

where

\[ c_p = c^\top x_p \] and \[ a_p = A_i x_p \]

(\( A_i \) is the row vector of the coefficients of the constraint \( i \), \( i \)-th row of the matrix \( A \)). In other words, the value of the extreme point \( p \) in the objective function and in the constraints.

The Convexification DW-reformulation of the example is then:
\[ Z(DW_C) = \max \begin{align*} & 0 \cdot \lambda_{p_1} + 5 \cdot \lambda_{p_2} + 5.12 \cdot \lambda_{p_3} \\ & 0 \cdot \lambda_{p_1} - 15 \cdot \lambda_{p_2} + 16 \cdot \lambda_{p_3} \leq 4 \\ & 0 \cdot \lambda_{p_1} + 5 \cdot \lambda_{p_2} + 0 \cdot \lambda_{p_3} = x_1 \\ & 0 \cdot \lambda_{p_1} + 0 \cdot \lambda_{p_2} + 8 \cdot \lambda_{p_3} = x_2 \\ & \lambda_{p_1} + \lambda_{p_2} + \lambda_{p_3} = 1 \\ & \lambda_{p_1}, \lambda_{p_2}, \lambda_{p_3} \geq 0 \end{align*} \]

2.2 Discretization

Reformulation of \( X \) itself for Binary Problems:

\[ x = \sum_{p \in EP} x_p \cdot \lambda_p, \quad \sum_{p \in EP} \lambda_p = 1, \quad \lambda \in \{0, 1\}^{|P|} \]

\[ Z(DW_D) = \max \sum_{p \in EP} c_p \cdot \lambda_p \quad (14) \]

\[ \sum_{p \in EP} a_p \cdot \lambda_p \leq b \quad (15) \]

\[ \sum_{p \in EP} \lambda_p = 1 \quad (16) \]

\[ \lambda \in \{0, 1\}^{|P|} \quad (17) \]

In case of integer variables the Discretization DW-reformulation provide a valid relaxation. In our example is then:

\[ Z(DW_D) = \max \begin{align*} & 0 \cdot \lambda_{p_1} + 5 \cdot \lambda_{p_2} + 5.12 \cdot \lambda_{p_3} \\ & 0 \cdot \lambda_{p_1} - 15 \cdot \lambda_{p_2} + 16 \cdot \lambda_{p_3} \leq 4 \\ & \lambda_{p_1} + \lambda_{p_2} + \lambda_{p_3} = 1 \\ & \lambda_{p_1}, \lambda_{p_2}, \lambda_{p_3} \geq 0 \end{align*} \]

Both reformulations must be solved via Column Generation and Branch-and-Price techniques!!!

How do we generate the extreme point? – Pricing (Column Generation)

- The Linear Programming Relaxation of the DW initialized with only a subset of the extreme points is called the Restricted Master Problem (RMP).
Now for the discretization DW, for a subset of extreme points \( p \in EP \) and writing constraints (15) as:

\[
\sum_{p \in EP} a^i_p \cdot \lambda_p \leq b_i \quad i = 1, \ldots, m
\]

the RMP becomes:

\[
\bar{Z}(LP(DW_D)) = \max \sum_{p \in EP} c_p \cdot \lambda_p
\]

(18)

\[
\sum_{p \in EP} a^i_p \cdot \lambda_p \leq b_i \quad i = 1, \ldots, m \quad (\alpha_i)
\]

(19)

\[
\sum_{p \in EP} \lambda_p = 1 \quad (\pi_0)
\]

(20)

\[
\lambda \geq 0 \quad p \in EP
\]

(21)

the dual of the RMP becomes:

\[
\bar{Z}(D(DW_D)) = \min \sum_{i=1}^{m} b_i \alpha_i + \pi_0
\]

\[
\sum_{i=1}^{m} a^i_p \alpha_i + \pi_0 \geq c_p \quad p \in EP
\]

\[
\alpha_i \geq 0 \quad i = 1, \ldots, m
\]

(22)

Since we are working with a subset of extreme points (primal variables), we need to check that all the dual constraints are satisfied (separation of the dual constraints). If we manage to found a violated constraints, we have found a new extreme points which needs to be added to the RMP (and then we reiterate).

Given \( \alpha^*, \pi_0^* \), we are looking for an extreme point \( p^* \) such that:

\[
\sum_{i=1}^{m} a^i_{ps} \alpha^*_i + \pi_0^* < c_{ps}
\]

\[
c_{ps} - \sum_{i=1}^{m} a^i_{ps} \alpha^*_i > \pi_0^*
\]

Such extreme point (if exist), can be found by solving the following MIP (called pricing problem):
\[ Z(Pr) = \max \sum_{j=1}^{n} \left( c_j - \left( \sum_{i=1}^{m} \alpha_i^* a_{ij} \right) \right) x_j \]

\[ \sum_{j=1}^{n} d_{ij} x_j \leq e_i \quad i = 1, \ldots, m \]

\[ x_j \in \mathbb{Z}_+ \quad j = 1, \ldots, n \]

where \( d_{ij} \) is the coefficient in the row \( i \) and column \( j \) of the constraint matrix \( D \) and \( e_i \) is the \( i \)-th coefficient of the RHS vector \( e \). The variables \( x \) represent a feasible extreme point, i.e., an extreme point of

\[ X = \{ x \in \mathbb{Z}_+^n : Dx \leq e \} \]

If the optimal solution value \( Z(Pr) > \pi_0^* \), then the optimal solution \( x^* \) define the new extreme point to be added to the RMP (column generation). Otherwise it is solved to proven optimality. (Attention: the RMP optimal solution may be fractional → branching!!!)

The pricing problem can be equivalently written as:

\[ Z(Pr) = \max \{ c^\top x - \alpha^{\star \top} A x - \pi_0 : x \in X \} \]

In this case, if \( Z(Pr) > 0 \) a column is generated.

**EXAMPLE:** The RMP of the previous example with just \( \lambda_{p1} \) is:

\[ \bar{Z}(LP(DW_D)) = \max \quad 0 \cdot \lambda_{p1} \]

\[ 0 \cdot \lambda_{p1} \leq 4 \quad (\alpha) \]

\[ \lambda_{p1} = 1 \quad (\pi_0) \]

\[ \lambda_{p1} \geq 0 \]

the optimal solution value is 0 and \( \lambda_{p1}^* = 1 \) and the dual of the RMP is:

\[ \bar{Z}(D(DW_D)) = \min \quad 4\alpha + \pi_0 \]

\[ \pi_0 \geq 0 \quad (\lambda_{p1}) \]

\[ \alpha \geq 0 \]

the optimal solution value is 0 and \( \pi_0^* = 0, \alpha^* = 0 \)
The first Pricing problem reads as follow:

\[
Z(Pr) = \max \quad x_1 + 0.64x_2 \\
\quad 50x_1 + 31x_2 \leq 250 \\
\quad x_1, x_2 \in \mathbb{Z}_+
\]

the optimal solution is \(x^*_1 = 0, x^*_2 = 8\) of value 5.12 which is > 0. We have found a new extreme point, the new RMP \(\lambda_{p_1}\) and \(\lambda_{p_3}\) reads as follow:

\[
\bar{Z}(LP(DW_D)) = \max \quad 0 \cdot \lambda_{p_1} + 5.12\lambda_{p_3} \\
\quad 0 \cdot \lambda_{p_1} + 16\lambda_{p_3} \leq 4 \quad (\alpha) \\
\quad \lambda_{p_1} + \lambda_{p_3} = 1 \quad (\pi_0) \\
\quad \lambda_{p_1}, \lambda_{p_3} \geq 0
\]

the optimal solution value is 1.28 and the optimal solution is \(\lambda^*_p = 0.75, \lambda^*_p = 0.25\). The dual of the RMP is:

\[
\bar{Z}(D(DW_D)) = \min \quad 4\alpha + \pi_0 \\
\quad \pi_0 \geq 0 \quad (\lambda_{p_1}) \\
\quad 16\alpha + \pi_0 \geq 5.12 \quad (\lambda_{p_3}) \\
\quad \alpha \geq 0
\]

the optimal solution value is 1.28 and the optimal solution is \(\alpha^* = 0.32, \pi_0^* = 0\). The Pricing problem reads as follow:

\[
Z(Pr) = \max \quad 1.96x_1 \\
\quad 50x_1 + 31x_2 \leq 250 \\
\quad x_1, x_2 \in \mathbb{Z}_+
\]

the optimal solution is \(x^*_1 = 5, x^*_2 = 0\) of value 9.80 which is > 0 (\(\pi_0^*\)). We have found a new extreme point, the new RMP with \(\lambda_{p_1}, \lambda_{p_3}\) and \(\lambda_{p_2}\) reads as follow:

\[
\bar{Z}(LP(DW_D)) = \max \quad 0 \cdot \lambda_{p_1} + 5.12\lambda_{p_3} + 5\lambda_{p_2} \\
\quad 0 \cdot \lambda_{p_1} + 16\lambda_{p_3} - 15\lambda_{p_2} \leq 4 \quad (\alpha) \\
\quad \lambda_{p_1} + \lambda_{p_3} + \lambda_{p_2} = 1 \quad (\pi_0) \\
\quad \lambda_{p_1}, \lambda_{p_3}, \lambda_{p_2} \geq 0
\]
the optimal solution value is 5.0735 and the optimal solution is $\lambda_{p_0}^* = 0, \lambda_{p_3}^* = 0.6129, \lambda_{p_2}^* = 0.3871$. The dual of the RMP is:

$$\bar{Z}(D(D_{W})) = \min 4\alpha + \pi_0$$

$$\pi_0 \geq 0 \quad (\lambda_{p_1})$$

$$16\alpha + \pi_0 \geq 5.12 \quad (\lambda_{p_3})$$

$$-15\alpha + \pi_0 \geq 5 \quad (\lambda_{p_2})$$

$$\alpha \geq 0$$

the optimal solution value is 5.0735 and the optimal solution is $\alpha^* = 0.0039, \pi_0^* = 5.0581$. The Pricing problem reads as follow:

$$Z(Pr) = \max 1.0116 x_1 + 0.6323 x_2$$

$$50x_1 + 31x_2 \leq 250$$

$$x_1, x_2 \in \mathbb{Z}_+$$

the optimal solution is $x_1^* = 5, x_2^* = 0$ of value 5.0581 which is NOT $> 5.0581 (\pi_0^*)$. The RMP is solved to proven optimality.

The optimal RMP solution is fractional, branching is then necessary! In the discretization approach we do not have the original variables, it is NOT possible to branch directly on the variable of the extreme points (we cannot set a variable to zero, since it will be immediately regenerated!). For this reason we need problem specific branching rules (see next section).
3 DW reformulation for the Bin Packing Problem

Definition: Bin Packing Problem (BPP)  Given a infinite number of bins of capacity $W$ and a set $N = \{1, \ldots, n\}$ of items of integer sizes $w_1, \ldots, w_n$ ($0 \leq w_i \leq W$), the Bin Packing Problem (BPP) asks to pack the items into the bins so that the capacity of the bin is not exceeded and the number of bins used is minimized.

Kantorovich Model

- Items $\rightarrow j = 1, \ldots, n; \quad w_j$ (weight)
- Bins $\rightarrow i = 1, \ldots, m; \quad W$ (capacity)

Variables

$$x_{ij} = \begin{cases} 1 & \text{item } j \text{ goes to bin } i \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \ldots, m, j = 1, \ldots, n$$

$$y_i = \begin{cases} 1 & \text{if bin } i \text{ is used} \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \ldots, m$$

The compact formulation reads as follows:

$$\min \sum_{i=1}^{m} y_i \quad (22)$$

$$\sum_{i=1}^{m} x_{ij} = 1 \quad j = 1, \ldots, n \quad (23)$$

$$\sum_{j=1}^{n} w_j x_{ij} \leq W y_i \quad i = 1, \ldots, m \quad (24)$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \ldots, m, j = 1, \ldots, n \quad (25)$$

$$y_i \in \{0, 1\} \quad i = 1, \ldots, m \quad (26)$$

($m$ is a valid upper bound on the number of bins, e.g., the number of bins in a feasible solution)

Constraints (23) impose that each item is packed in one bin. Constraints (24) impose that the sum of the weight of the items does not exceed the capacity of the bin (if the bin is used, otherwise they impose that nothing can be packed in it).

We now want to reformulate the following set of constraints:
\[ P_i = \left\{ x_{ij} \in \{0, 1\}, y_j \in \{0, 1\}, \ j = 1, \ldots, n : \sum_{j=1}^{n} w_j x_{ij} \leq W y_i \right\} \quad i = 1, \ldots, m \]

**Extreme points** \( \rightarrow p \in EP \)

\[ \bar{x}_{ij}^p = \begin{cases} 1 & \text{if item } j \text{ is in bin } i \text{ in extreme point } p \\ 0 & \text{otherwise} \end{cases} \]

\[ \bar{y}_i^p = \begin{cases} 1 & \text{if bin } i \text{ is used in } p \\ 0 & \text{otherwise} \end{cases} \]

Relation between the original variables and the new ones:

\[ x_{ij} = \sum_{p \in EP} \bar{x}_{ij}^p \lambda_i^p \quad j = 1, \ldots, n, i = 1, \ldots, m \]

\[ y_i = \sum_{p \in EP} \bar{y}_i^p \lambda_i^p \quad i = 1, \ldots, m \]

Example of extreme points with 8 items (bin last position):

- **Items 1,3,8 and open bin**
  
  \[ [1, 0, 1, 0, 0, 0, 0, 1|1] \]

- **Closed bin**
  
  \[ [0, 0, 0, 0, 0, 0, 0, 0|0] \]

Now applying the Discretization DW to the Kantorovich formulation we obtain:

\[ \min \sum_{i=1}^{m} \sum_{p \in EP} \bar{y}_i^p \lambda_i^p \quad (27) \]

\[ \sum_{i=1}^{m} \sum_{p \in EP} \bar{x}_{ij}^p \lambda_i^p = 1 \quad j = 1, \ldots, n \quad (28) \]

\[ \sum_{p \in EP} \lambda_i^p = 1 \quad i = 1, \ldots, m \quad (29) \]

\[ \lambda_i^p \in \{0, 1\} \quad i = 1, \ldots, m, p \in EP \quad (30) \]

The extreme points in which \( \bar{y}_j^p = 0 \) can be removed obtaining:
\[
\min \sum_{i=1}^{m} \sum_{p \in EP: \bar{y}_{i}^{p} = 1} \lambda_{i}^{p}
\]
\[
\sum_{i=1}^{m} \sum_{p \in EP: \bar{x}_{i}^{p} = 1} \lambda_{i}^{p} = 1 \quad j = 1, \ldots, n
\]
\[
\sum_{p \in EP: \bar{y}_{i}^{p} = 1} \lambda_{i}^{p} \leq 1 \quad i = 1, \ldots, m
\]
\[
\lambda_{i}^{p} \in \{0, 1\} \quad i = 1, \ldots, m, p \in EP
\]

Now since all the bins are identical (same set of extreme points) and constraints (33) can be removed (due to the objective function and having \(m\) large enough to pack all the items), one can remove the index \(i\) from the extreme point, introducing the new variables:

\[
\lambda_{p} = \sum_{i=1}^{m} \lambda_{i}^{p}
\]

and thus obtaining:

\[
\min \sum_{p \in EP: \bar{y}_{i}^{p} = 1} \lambda_{p}
\]
\[
\sum_{p \in EP: \bar{x}_{j}^{p} = 1} \lambda_{p} = 1 \quad j = 1, \ldots, n
\]
\[
\lambda_{p} \in \{0, 1\} \quad p \in EP
\]

This formulation is called the Set Partitioning Formulation for the BPP. By defining the following collection of all feasible packing patterns (subset of items respecting the bin capacity):

\[
\bar{P} = \left\{ S \subseteq N : \sum_{j \in S} w_{j} \leq W \right\}
\]

(this collection corresponds to the extreme points) and associating a variable to each packing pattern:

\[
\lambda_{S} = \begin{cases} 
1 & \text{if packing pattern } S \text{ is chosen as a bin} \\
0 & \text{otherwise}
\end{cases} \quad S \in \bar{P}
\]

the Set Partitioning formulation can be equivalently rewritten as:

\[
\min \sum_{S \in \bar{P}} \lambda_{S}
\]
\[
\sum_{S \in \bar{P}, j \in S} \lambda_{S} = 1 \quad j = 1, \ldots, n
\]
\[
\lambda_{S} \in \{0, 1\} \quad S \in \bar{P}
\]
Working only with maximal packing patterns:

\[ \mathcal{P} = \left\{ S \subseteq N : \sum_{j \in S} w_j \leq W, \sum_{j \in S \cup \{k\}} w_j > W, k \notin S \right\} \]

we obtain the Set Covering formulation of the BPP:

\[
\begin{align*}
\min & \sum_{S \in \mathcal{P}} \lambda_S \\
\text{s.t.} & \sum_{S \in \mathcal{P}, j \in S} \lambda_S \geq 1 & j = 1,\ldots,n \\
& \lambda_S \in \{0,1\} & S \in \mathcal{P}
\end{align*}
\]

The Set Covering Formulations is equivalent to the Set Partitioning Formulation since the following hold:

**Remark.** Any Set Partitioning solution corresponds to a Set Covering solution of the same value

We say that a Set Covering solution is minimal if no bin is entirely composed by items that are packed in other bins of the solution as well

**Remark.** Any minimal Set Covering solution corresponds to a Set Partitioning solution with the same value

**Example:** 4 items, \( W = 4, w_1 = 1, w_2 = 2, w_3 = 3, w_4 = 4 \):

\[ \mathcal{P} = S_1 = \{1\}, S_2 = \{2\}, S_3 = \{3\}, S_4 = \{4\}, S_5 = \{1,2\}, S_6 = \{1,3\} \]

Set Partitioning formulations:

\[
\begin{align*}
\min & \lambda_{S_1} + \lambda_{S_2} + \lambda_{S_3} + \lambda_{S_4} + \lambda_{S_5} + \lambda_{S_6} \\
& \lambda_{S_1} + \lambda_{S_5} + \lambda_{S_6} = 1 \\
& \lambda_{S_2} + \lambda_{S_5} = 1 \\
& \lambda_{S_3} + \lambda_{S_6} = 1 \\
& \lambda_{S_4} = 1 \\
& \lambda_{S_1}, \lambda_{S_2}, \lambda_{S_3}, \lambda_{S_4}, \lambda_{S_5}, \lambda_{S_6} \in \{0,1\}
\end{align*}
\]

\[ \mathcal{P} = S_4 = \{4\}, S_5 = \{1,2\}, S_6 = \{1,3\} \]

Set Covering formulations:
\[ \begin{align*}
\min & \quad \lambda_{S_4} + \lambda_{S_5} + \lambda_{S_6} \\
& \quad \lambda_{S_5} + \lambda_{S_6} \geq 1 \\
& \quad \lambda_{S_5} \geq 1 \\
& \quad \lambda_{S_6} \geq 1 \\
& \quad \lambda_{S_4} \geq 1 \\
& \quad \lambda_{S_4}, \lambda_{S_5}, \lambda_{S_6} \in \{0, 1\}
\end{align*} \]

### 3.1 Column Generation for the BPP

The set of feasible packing pattern is exponential in number and it cannot be enumerated in advance (exponential number of extreme points). For this reason we need to generate only the subset of packing pattern which are necessary to solve the model to proven optimality. We need then price out the variables (Column Generation).

- The dual of the RMP (with a subset of variables \( \tilde{P} \)) is for the Set Covering Formulation:

\[
\begin{align*}
\max & \quad \sum_{j=1}^{n} \pi_j \\
\sum_{j \in S} \pi_j & \leq 1 \quad S \in \tilde{P} \quad (45) \\
\pi_j & \geq 0 \quad j = 1, \ldots, n \quad (46)
\end{align*}
\]

- Given an optimal solution \((\lambda^*, \pi^*)\) of the restricted master problem (primal and dual), does it exist a set \(S^* \in \mathcal{P}\) (separation problem):

\[
\sum_{j \in S^*} \pi_j^* > 1 \quad (47)
\]

- the separation problem can be formulated by defining the following binary variables

\[
x_j = \begin{cases} 
1 & \text{if item } j \text{ belongs to } S^* \\
0 & \text{otherwise} 
\end{cases} \quad j = 1, \ldots, n
\]

and it corresponds to the following Knapsack Problem KP01:

\[
\begin{align*}
Z(Pr) &= \max \sum_{j=1}^{n} \pi_j^* x_j \\
\sum_{j=1}^{n} w_j x_j & \leq W \quad (49) \\
x_j & \in \{0, 1\} \quad j = 1, \ldots, n \quad (50)
\end{align*}
\]
If \( \pi_j > 0 \) for all items \( j \in N \) then the solution is maximal, otherwise we can make it maximal adding (if possible) items with \( \pi_j = 0 \).

- If the optimal solution value \( Z(Pr) \) of the KP01: has value \( \leq 1 \), then all the dual constraints are satisfied by \( \pi^* \).

- Otherwise, given the subset

\[
S^* := \{ j = 1, \ldots, n : x^*_j = 1 \}
\]

any maximal set \( S' \in \mathcal{P} \) s.t. \( S^* \subset S' \) corresponds to the dual constraint most violated by \( \pi^* \).

- **CG Example:** with \( n = 5 \), bin capacity \( W = 10 \) and item weights \( w = (7, 5, 4, 4, 3) \)

  - Initialize with the (heuristic) initial feasible solution in which each item is packed separately into a (different) bin

  - The associated variables for the LP (after having made the sets maximal) are \( \lambda_S \) for

\[
S \in \{\{1, 5\}, \{2, 5\}, \{3, 5\}, \{4, 5\}\}
\]

  - The column generation (CG) algorithm evolves as follow

1. the nonzero (positive) components of \( \lambda^* \) and \( \pi^* \) are \( \lambda^*_S = 1 \) for \( S \in \{\{1, 5\}, \{2, 5\}, \{3, 5\}, \{4, 5\}\} \) and \( \pi^*_j = 1 \) for \( j \in \{1, 2, 3, 4\} \); the constraint associated with \( S^* = \{2, 3\} \) is violated and the corresponding variable \( \lambda_{\{2,3\}} \) is added

2. the nonzero (positive) components of \( \lambda^* \) and \( \pi^* \) are \( \lambda^*_S = 1 \) for \( S \in \{\{1, 5\}, \{2, 3\}, \{4, 5\}\} \) and \( \pi^*_j = 1 \) for \( j \in \{1, 2, 4\} \); the constraint associated with \( S^* = \{2, 4\} \) is violated and the corresponding variable \( \lambda_{\{2,4\}} \) is added

3. the nonzero (positive) components of \( \lambda^* \) and \( \pi^* \) are \( \lambda^*_S = 1 \) for \( S \in \{\{1, 5\}, \{2, 3\}, \{4, 5\}\} \) and \( \pi^*_j = 1 \) for \( j \in \{1, 3, 4\} \); the constraint associated with \( S^* = \{3, 4\} \) is violated and the corresponding variable \( \lambda_{\{3,4\}} \) is added

4. the nonzero (positive) components of \( \lambda^* \) and \( \pi^* \) are \( \lambda^*_S = 1 \) and \( \lambda^*_{\{1,5\}} = \frac{1}{2} \) for \( S \in \{\{2, 3\} \{2, 4\} \{3, 4\}\} \) and \( \pi^*_1 = 1 \) and \( \pi^*_j = \frac{1}{2} \) for \( j \in \{2, 3, 4\} \); and no constraint is violated, thus \( \lambda^* \) and \( \pi^* \) are also the optimal primal and dual optimal solutions of the full LP, respectively

**Farley bound.** Once the optimal value \( z^* \) is computed, one can obtain a dual feasible solution and accordingly a lower bound for the problem as follows:

\[
\sum_{j \in S^*} \pi^*_j = Z(Pr) \quad (69), \quad \frac{\sum_{j \in S} \pi^*_j}{Z(Pr)} \leq 1, S \in \mathcal{P}, \quad LB_F = \frac{\sum_{j \in N} \pi^*_j}{Z(Pr)} \quad (71)
\]
Tailing off effect. Column generation is known to suffer from tailing off, i.e., there is only incremental progress per iteration the closer we get to the optimum, in particular for large and degenerate problems. There are several partial explanations, but a main reason lies in the unstable behavior of the dual variables.

3.2 Stabilization Techniques for the BPP

Proximity of a Stability Center using a piecewise linear penalty function. The Tailing off effect is partially due to the oscillation of dual variables values:
Primal of the Set Covering formulation of the BPP:

\[
\min \sum_{S \in \tilde{\mathcal{P}}} \lambda_S + \sum_{j=1}^{n} (\delta^+ \cdot y_j^+ - \delta^- \cdot y_j^-) 
\]
\[
\sum_{S \in \tilde{\mathcal{P}} : j \in S} \lambda_S + y_j^+ - y_j^- \geq 1 \quad j = 1, \ldots, n
\]
\[
y_j^+ \leq \epsilon^+ \quad j = 1, \ldots, n
\]
\[
y_j^- \leq \epsilon^- \quad j = 1, \ldots, n
\]
\[
\lambda_S \geq 0 \quad S \in \tilde{\mathcal{P}}
\]
\[
y_j^+, y_j^- \geq 0 \quad j = 1, \ldots, n
\]

Dual of the Set Covering formulation of the BPP:

\[
\max \sum_{j=1}^{n} (\pi_j - \epsilon^+ \cdot w_j^+ - \epsilon^- \cdot w_j^-)
\]
\[
\sum_{j \in S} \pi_j \leq 1 \quad S \in \tilde{\mathcal{P}}
\]
\[
\pi_j \leq \delta^+ + w_j^+ \quad j = 1, \ldots, n
\]
\[
\pi_j \geq \delta^- - w_j^- \quad j = 1, \ldots, n
\]
\[
\pi_j \geq 0 \quad j = 1, \ldots, n
\]
\[
w_j^+, w_j^- \geq 0 \quad j = 1, \ldots, n
\]
the box \([-\delta, \delta]\) is centered on a **stability centre** \(\hat{\pi}\) and it **penalizes** by \(\epsilon_-\) and \(\epsilon_+\) the deviation from it (outside from the box):

\[
[\hat{\pi}_j - \delta, \hat{\pi}_j + \delta] \quad j = 1, \ldots, n
\]

in the last iteration we set \(y_- = y_+ = 0\) (i.e. \(\epsilon_- = \epsilon_+ = 0\) and \(-\infty < \pi < +\infty\)) in order to remove the stabilization term from the problem.

\(\hat{\pi}\) and parameters \(\delta\) and \(\epsilon\) are **updated** dynamically. **Strategies** to perform the update can be different.

the center of stability can be set to the best dual feasible solution found so far given by the Farley bound.

---

**Smoothing techniques.** The current dual solution \(\pi\) is corrected using informations of the previous dual solution. Consider that a set of column generation iterations has been performed, at the iteration \(t\) then we set:

\[
\tilde{\pi}_j^t = \alpha \cdot \tilde{\pi}_j^{t-i} + (1 - \alpha) \cdot \pi_j^t \quad j \in N
\]

where \(\tilde{\pi}_j^t\) is a weighted sum of the values of the \(\pi\) in previous iterations (with a discount factor modelling the obsolescence of old iterations):

\[
\tilde{\pi}_j^t = \sum_{\tau=0}^{t} (1 - \alpha) \cdot \alpha^{t-\tau} \cdot \pi_j^\tau
\]

Wentges proposed the following different rule:

\[
\tilde{\pi}_j^t = \alpha \cdot \hat{\pi}_j + (1 - \alpha) \cdot \pi_j^t \quad j \in N
\]

that is equivalent to:

\[
\tilde{\pi}_j^t = \hat{\pi}_j + (1 - \alpha) \cdot (\pi_j^t - \hat{\pi}_j) \quad j \in N
\]
which amounts to taking a step of size \((1 - \alpha)\) from \(\hat{\pi}\) in the direction of \(\pi^t\). The pricing problem is then solved using the smoothed new dual variables values. This can result in a mispricing (the modified pricing might not yield a negative reduced cost column even when one exists). For this reason it is necessary to perform an additional round with the real dual variables value.
4 Branch and Price Algorithm

Column generation methods: necessary when the number of variables of a Linear Program is exponential

Branch-and-Price Framework

- Generate a subset of the variables (feasible solution)

1 Solve the LP and get the current solution $\lambda^*, \pi^*$

2 If some constraints in the Dual are violated by $\lambda^*, \pi^*$, add the columns to the LP and go to 1

3 $\lambda^*$ is optimal and feasible for the LP (possibly fractional)

- Branch on a feasible disjunction for the problem and go to 1

4.1 Branching rule for the Bin Packing Problem

Basic idea (Ryan/Foster branching): at each node of the branching tree we select two items $j$ and $j' \in N$ such that:

$$\sum_{S \in \mathcal{P}, j, j' \in S} \lambda^*_S = \alpha, \alpha \text{ is fractional}$$

Then two branching nodes are created as follows:

1) items $j$ and $j'$ in the same bin

2) items $j$ and $j'$ in different bins

This branching rule is complete

Remark. Since the master constraint matrix $A$ is a 0-1 matrix, if a basic solution to $A\lambda^* = 1$ is fractional, i.e., at least one of the components of $x^*$ is fractional, then there exist two rows (items) $r$ and $s$ of the master problem such that:

$$0 < \sum_{S \in \mathcal{P}, r, s \in S} \lambda^*_S < 1$$

Even for the Set Covering formulation we have for any basic solution $A\lambda^* = 1$ (due to the objective function and after copy removal)

Proof. 
- Consider a fractional column $\lambda_{S_1}$.

- Let a row $r$ be any row covered by $S_1$. Since $\sum_{S \in \mathcal{P}, r \in S} \lambda^*_S = 1$ there must exist at least one other basic column $\lambda_{S_2}$ with $0 < \lambda_{S_2} < 1$ and $r \in S_2$. 

22
• Since there are no duplicate columns in the basis, there must exist a row \( s \) such that either \( s \in S_1 \) or \( s \in S_2 \) but not both. Then:

\[
1 = \sum_{S \in \mathcal{P}} \lambda_{S_1}^* > \sum_{S \in \mathcal{P}, r \in S, s \in S} \lambda_{S}^* > 0
\]

where the inequality follows from the fact that the last summation includes either \( \lambda_{S_1} \) or \( \lambda_{S_2} \), but not both.

How does the branching work in practice?

• consider 5 items and the following patterns:

\[
S_1 = \{1, 3\}, \ S_2 = \{1, 4\}, \ S_3 = \{2, 4\}, \ S_4 = \{2, 5\}, \ S_5 = \{3, 5\}
\]

• Current optimal basic solution \( \lambda^* \)

\[
\begin{align*}
\lambda_{S_1}^* + \lambda_{S_2}^* &= 1 \text{ (item 1)} \\
\lambda_{S_3}^* + \lambda_{S_4}^* &= 1 \text{ (item 2)} \\
\lambda_{S_1}^* + \lambda_{S_3}^* + \lambda_{S_5}^* &= 1 \text{ (item 3)} \\
\lambda_{S_2}^* + \lambda_{S_3}^* + \lambda_{S_5}^* &= 1 \text{ (item 4)} \\
\lambda_{S_4}^* + \lambda_{S_5}^* &= 1 \text{ (item 5)}
\end{align*}
\]

• For example, it is possible to branch considering the column containing both items 1 and 3

\[
\sum_{S \in \mathcal{P}: 1 \in S, 3 \in S} \lambda_{S}^* = 0.5
\]

• subset of the columns infeasible after branching (removed)

• Additional constraints to the pricing KP01 problem to generate columns with the desired proprieties

1) Together branching – create a super item merging \( j \) and \( j' \)

2) Separate branching – add an incompatibility constraint to the KP01

Together Pricer

\[
\begin{align*}
\text{max} & \quad \sum_{j=1}^{n} \pi_j^* x_j \\
\sum_{j=1}^{n} w_j x_j & \leq W \\
x_j = x_{j'} & \quad j = 1, \ldots, n
\end{align*}
\]
Separate Pricer

\[
\begin{align*}
\max & \quad \sum_{j=1}^{n} \pi_j^* x_j \\
\sum_{j=1}^{n} w_j x_j & \leq W \\
x_j + x_{j'} & \leq 1 \\
x_j & \in \{0, 1\}, \quad j = 1, \ldots, n
\end{align*}
\]

The additional constraints are one for each branching decision.
5 DW reformulation for the Vertex Coloring Problem

Definition: Vertex Coloring Problem (VCP) Given a simple graph \( G = (V,E) \), with \( n = |V| \) and \( m = |E| \), assign a color to each vertex in such a way that colors on adjacent vertices are different and the number of colors used is minimized.

Compact Model

- Vertices \( v \in V \);
- Colors \( c = 1, \ldots, k \); \( k \) can be set to \( |V| = n \)

Variables

\[
x_{cv} = \begin{cases} 
1 & \text{vertex } v \text{ takes color } i \\
0 & \text{otherwise}
\end{cases}, \quad c = 1, \ldots, k, v \in V
\]

\[
y_c = \begin{cases} 
1 & \text{color } c \text{ is used} \\
0 & \text{otherwise}
\end{cases}, \quad c = 1, \ldots, k
\]

The compact formulation reads as follows:

\[
\min \sum_{c=1}^{k} y_c 
\]

\[
\sum_{c=1}^{k} x_{cv} = 1, \quad v \in V \quad (64)
\]

\[
x_{cu} + x_{cv} \leq y_i, \quad uv \in E, c = 1, \ldots, k \quad (65)
\]

\[
x_{ij} \in \{0, 1\}, \quad i = 1, \ldots, m, j = 1, \ldots, n \quad (66)
\]

\[
y_i \in \{0, 1\}, \quad i = 1, \ldots, m \quad (67)
\]

\((k \text{ is a valid upper bound on the number of colors, e.g., the number of colors in a feasible solution})\)

Constraints (64) impose that each vertex is colored. Constraints (65) impose that vertices linked by an edge receive different colors (if the color is used, otherwise they impose that none of them can take that color).
**Set Partitioning Formulation**  With similar procedure to the one used for the BPP we can obtain the following model. We also observe that, a *coloring* $C$ of $G$ is a partition of $V$ into $k$ non-empty *stable sets*:

$$C = \{V_1, \ldots, V_k\},$$

where all vertices belonging to $V_i$ are colored with the same color $c$ ($c = 1, \ldots, k$). A *stable set* is a subset of fully disconnected vertices.

The *chromatic number* of $G$, denoted by $\chi(G)$, is the minimum number of stable sets (or equivalently colors) in a coloring of $G$ and the *Vertex Coloring Problem* (VCP) is the problem of determining the chromatic number of the graph $G$.

By defining the following collection of all *stable sets* (subset of vertices inducing a graph with no edges):

$$\mathcal{P} = \{S \subseteq V : u \in S, v \in S, uv \notin E\}$$

(this collection corresponds to the extreme points) and associating a variable to each stable set:

$$\lambda_S = \begin{cases} 1 & \text{if stable set } S \text{ is chosen as a color} \\ 0 & \text{otherwise} \end{cases} \quad S \in \mathcal{P}$$

the Set Partitioning formulation read as follows:

$$\begin{align*}
\min & \quad \sum_{S \in \mathcal{P}} \lambda_S \\
\text{s.t.} & \quad \sum_{S \in \mathcal{P}, v \in S} \lambda_S = 1 \\
& \quad \lambda_S \in \{0, 1\} \\
& \quad S \in \mathcal{P}
\end{align*}$$

(68)

(69)

(70)

### 5.1 Column Generation for the VCP

The collection of stable sets is exponential in number and it cannot be enumerated in advance (exponential number of extreme points). For this reason we need to generate only the subset of stable sets which are necessary to solve the model to proven optimality. We need then price out the variables (*Column Generation*).

- The dual of the RMP (with a subset of variables $\mathcal{P}$) is for the Set Covering Formulation:

$$\begin{align*}
\max & \quad \sum_{v \in V} \pi_v \\
\text{s.t.} & \quad \sum_{v \in S} \pi_v \leq 1 \\
& \quad \pi_j \geq 0 \\
& \quad j = 1, \ldots, n
\end{align*}$$

(71)

(72)

(73)
• Given an optimal solution \((\lambda^*, \pi^*)\) of the restricted master problem (primal and dual), does it exist a set \(S^* \in \mathcal{P}\) (separation problem):

\[
\sum_{v \in S^*} \pi_v^* > 1
\]  

(74)

• the separation problem can be formulated by defining the following binary variables

\[
x_v = \begin{cases} 
1 & \text{if item } v \text{ belongs to } S^* \\
0 & \text{otherwise}
\end{cases} \quad v \in V
\]

and it corresponds to the following Maximum Weight Stable Set Problem MWSS:

\[
Z(Pr) = \max \sum_{v \in V} \pi_v^* x_v
\]  

(75)

\[
x_u + x_v \leq 1 \quad uv \in E
\]

(76)

\[
x_v \in \{0, 1\} \quad v \in V
\]  

(77)

If \(\pi_v > 0\) for all items \(j \in N\) then the solution is maximal, otherwise we can make it maximal adding (if possible) vertices with \(\pi_v = 0\).

– If the optimal solution value \(Z(Pr)\) of the MWSS: has value \(\leq 1\), then all the dual constraints are satisfied by \(\pi^*\).

– Otherwise, given the subset

\[
S^* := \{v \in V : x_v^* = 1\}
\]

any maximal set \(S' \in \mathcal{P}\) s.t. \(S^* \subset S'\) corresponds to the dual constraint most violated by \(\pi^*\).