A strengthened formulation for the Simple Plant Location Problem with Order

Lázaro Cánovas\textsuperscript{1}, Sergio García\textsuperscript{1}, Martine Labbé\textsuperscript{2} and Alfredo Marín\textsuperscript{1}

\textsuperscript{1} Departamento de Estadística e Investigación Operativa
Universidad de Murcia, 30100 Murcia, Spain,
\textsuperscript{2} Département d’Informatique, Université Libre de Bruxelles, Belgium,
E-mails: lcanovas@um.es, sgarcia@um.es, mlabbe@ulb.ac.be, amarin@um.es

Abstract

The Simple Plant Location Problem is well known in the literature: it consists in deciding which facilities to open among a set of potential nodes and allocating a set of customers to the open facilities in such a way that the total cost is minimum, allocation process which is carried out by the locator. However, customers may be free to choose the facility they will be allocated to. This fact can be considered by including customers’ preferences in the model, obtaining the so-called Simple Plant Location Problem with Order (SPLPO). Some new valid inequalities which reinforce the SPLPO formulation are given here and their efficiency is shown with a computational study.

Keywords: Location - Integer Programming - Valid inequality - Preprocessing - Branch-and-bound
1. Introduction

The Simple Plant Location Problem (SPLP) is a well-known combinatorial optimization problem: some plants or facilities must be chosen among a set of candidates and each of the customers from a given set must be allocated to one of the open facilities, in such a way that the total cost (location plus allocation) is minimum. Many papers on this problem can be found in the literature: [2], [3] or [4] are just some of the most classical.

Two elements can be differentiated: one of location (i.e., the firm which decides which facilities to be opened) and another one of allocation (the process of deciding which customers attend which facility). In this classical approach, it is supposed that either both the locator and the allocator are the same or the allocator (i.e., the customers as a whole) knows the optimality criterium of the locator and agrees with it. For example, this happens when a firm delivers a product to the customers from the different facilities or when the customer is obliged to go to the prescribed facility with a law or fiat.

However, if, for example, the customers travel to the facilities to obtain their service, then they have no compulsion on what facility to attend to. This means that, once the facilities are open, what the locator wishes the customers to do may not coincide with how they behave. Different customers may have different preferences according to several reasons (mentality, habits, work, age, income...).

Therefore, a solution to this situation is to consider a formulation where customers’ preferences are taken into account, which was done in [5]. In the model proposed in this paper, each customer ranks the different potential facilities and, once it has been decided which ones will be opened, it attends
its most preferred one. The model postulates that the locator knows the preference orderings of the customers. Opposite to what happens with SPLP, which has been broadly studied in the literature, the authors know of only two references for the SPLPO: [5], where the problem is first introduced and a simple heuristic for very small problems is suggested, and [6], where a bilevel formulation for this problem is converted in a linear one in different ways, the stronger one being the same formulation that appears in the paper of Hanjoul and Peeters.

The rest of this paper is organized as follows. Section 2 contains the default formulation and shows different valid inequalities and properties for SPLPO. In Section 3, some of these properties are combined to obtain several preprocessing rules, which are applied to the model to obtain a reduced and tighter formulation. Finally, in Section 4 a computational study is shown and some conclusions are given.

2. Formulation and valid inequalities

A set $I$ of customers, $|I| = m$, and a set $J$ of facilities, $|J| = n$, are considered. We are also given costs $c_{ij}$ for supplying the whole demand of customer $i$ from facility $j$ and costs $f_j$ for opening a facility at node $j$.

**Definition 1**

Let $i \in I$ and $k, j \in J$. It is said that $k$ is $i$-worse than $j$ if customer $i$ prefers facility $j$ to facility $k$. This will be denoted by $k <_i j$.

Throughout this work, all of the preferences are supposed to be strict, that is, if $k =_i j$ then $k = j$. 
The following variables are used in the model:

\[ x_{ij} = \text{fraction of the demand of customer } i \text{ which is served from facility } j, \]

\[ y_j = \begin{cases} 
1 & \text{if facility } j \text{ is open,} \\
0 & \text{otherwise.} 
\end{cases} \]

Remark that, since no capacities are considered, variables \( x_{ij} \) will take binary values in the optimal solution.

The formulation for the SPLPO which was proposed in [5] is the following:

\[
\begin{align*}
\text{Min.} & \quad \sum_{i,j} c_{ij} x_{ij} + \sum_k f_k y_k \\
\text{s.t.} & \quad \sum_j x_{ij} = 1 \quad \forall i \in I, \\
& \quad x_{ij} \leq y_j \quad \forall i \in I, j \in J, \\
& \quad \sum_{\{k: k \geq j\}} x_{ik} \geq y_j \quad \forall i \in I, j \in J, \quad (1) \\
& \quad x_{ij} \geq 0 \quad \forall i \in I, j \in J, \\
& \quad y_j \in \{0, 1\} \quad \forall j \in J.
\end{align*}
\]

As it can be seen, it is the SPLP formulation together with the inequalities family (1), the family which models the preferences.

Now, if the set

\[ W(i, j) = \{k \in J : k \leq i, j\}, \]

with \( i \in I, j \in J, \) is considered, then inequalities (1) can be rewritten as

\[ C2-1(i; j) \equiv \sum_{k \in W(i, j)} x_{ik} + y_j \leq 1. \]

(The notation \( C2-1(i; j) \) will be better understood under the frame of the more general inequalities \( C2-s \) described later in this section.)
Further notation:

$$W'(i, j) = W(i, j) \cup \{j\} = \{k \in J : k \leq j\},$$

$$\overline{W}(i, j) = J \setminus W(i, j) = \{k \in J : k \geq j\}.$$

Let $p_{ij}$ be the ranking preference of customer $i$ for facility $j$, that is, $p_{ij} = t$ means that $j$ is the $t^{th}$ $i$-best facility. Obviously, $1 \leq p_{ij} \leq n$.

Finally, we define

- $B_t(i) = j$ if $p_{ij} = t$,
- $W_t(i) = j$ if $p_{ij} = n - t + 1$.

Now, three new families of valid inequalities for the SPLPO will be shown.

2.1. Inequalities $\mathcal{C}1$

Proposition 2

Let $i_1, i_2 \in I$ and $j \in J$. It holds that

$$\mathcal{C}1(i_1, i_2; j) \equiv \sum_{k \in W(i_1, j)} x_{i_1k} \leq \sum_{k \in W(i_2, j)} x_{i_2k}. \tag{2}$$

Proof:

Assume that

$$\sum_{k \in W(i_1, j)} x_{i_1k} > \sum_{k \in W(i_1, j)} x_{i_2k}$$

for some $(x, y)$. It must necessarily be

$$\sum_{k \in W(i_1, j)} x_{i_1k} = 1, \quad \sum_{k \in W(i_2, j)} x_{i_2k} = 0.$$
Now

$$\sum_{k \in W(i_1, j)} x_{i_2k} = 0$$

means that it exists $\bar{k} \in W(i_1, j)$ such that $y_{\bar{k}} = 1$.

But this implies $\sum_{k \in W(i_1, j)} x_{i_1k} = 0$, which is a contradiction. $\square$

Now some immediate consequences.

**Corollary 3**

Let $i_1, i_2 \in I$ and $j \in J$. If

$$W(i_1, j) = W(i_2, j),$$

then

$$\sum_{k \in W(i_1, j)} x_{i_1k} = \sum_{k \in W(i_1, j)} x_{i_2k}. \quad (3)$$

**Corollary 4**

Let $i_1, i_2 \in I$ and $j \in J$. If

$$W(i_1, j_1) = W(i_2, j_2),$$

then

$$C I(i_1, i_2; j_1) \equiv C I(i_2, i_1; j_2),$$

that is, they are the same inequality.

**Corollary 5**

Let $i_1, i_2 \in I$ and $j \in J$. If

$$W(i_1, j) \subseteq W(i_2, j),$$

then it holds that

$$\sum_{k \in W(i_1, j)} x_{i_1k} \leq \sum_{k \in W(i_2, j)} x_{i_2k}. \quad (4)$$
Proof:

It is immediate that

\[ \sum_{k \in W(i_1, j)} x_{i_1 k} \leq \sum_{k \in W(i_2, j)} x_{i_2 k} \leq \sum_{k \in W(i_2, j)} x_{i_2 k}. \]

The first inequality is just \( C_1(i_1, i_2; j) \); the second one is consequence of being \( W(i_1, j) \subseteq W(i_2, j) \). \( \square \)

**Corollary 6**

Let \( i_1, i_2 \in I \) and \( j \in J \). If

\[ W(i_1, j) \subseteq W(i_2, j), \]

then \( C_2-1(i_1; j) \) is dominated by \( C_2-1(i_2; j) \)

Proof:

Using the previous corollary and \( C_2-1(i_2; j) \), we have that

\[ \sum_{k \in W(i_1, j)} x_{i_1 k} \leq \sum_{k \in W(i_2, j)} x_{i_2 k} \leq 1 - y_j. \]

\( \square \)

Therefore, it seems natural to replace \( C_2-1(i_1; j) \) with \( C_1(i_1, i_2; j) \) whenever \( W(i_1, j) \subseteq W(i_2, j) \). However, it is not worth the better linear relaxation bound to increasing considerably the number of non-zero elements in the model.

**2.2. Inequalities \( C2 \)**

The following inequalities dominate family \( C_2-1(i_1; j) \).
Proposition 7

Let $i_1, i_2 \in I$ and $j \in J$. It holds that

\[
C_{2-2}(i_1, i_2; j) \equiv \sum_{k \in W(i_1, j)} x_{i_1 k} + \sum_{k \in W(i_2, j) \setminus W(i_1, j)} x_{i_2 k} + y_j \leq 1.
\] (5)

Proof:

It is enough to prove that the inequality holds when $\sum_{k \in W(i_1, j)} x_{i_1 k} = 1$.

But this implies that $y_k = 0$ for $k \geq i_1, j$. Particularly,

\[
y_j = \sum_{k \in W(i_2, j) \setminus W(i_1, j)} x_{i_2 k} = 0.
\]

\[\square\]

A quick thought arises: to replace each $C_{2-1}(i_1; j)$ in the initial formulation with a $C_{2-2}(i_1, i_2; j)$ such that $W(i_2, j) \setminus W(i_1, j) \neq \emptyset$. Again, there is an improved lower bound in the linear relaxation, but there are also too much new non-zero elements.

The case $W(i_1, j) \cap W(i_2, j) = \emptyset$ will be of special interest in the preprocessing method described later.

Corollary 8

Let $i_1, i_2 \in I$ and $j \in J$. If

\[
W(i_1, j) \cap W(i_2, j) = \emptyset,
\]

then it holds

\[
\sum_{k \in W(i_1, j)} x_{i_1 k} + \sum_{k \in W(i_2, j)} x_{i_2 k} + y_j \leq 1.
\] (6)

It is easy to generalize inequalities $C_{2-2}$ to have stronger inequalities:
Corollary 9

Let $i_1, \ldots, i_s \in I$ and $j \in J$. It holds that

$$C_{\text{2-s}}(i_1, \ldots, i_s; j) \equiv \sum_{k \in W(i_1, j)} x_{i_1k} + \sum_{t=2}^{s} \sum_{k \in W(i_t, j) \setminus \bigcup_{q=1}^{t-1} W(i_q, j)} x_{itk} + y_j \leq 1. \quad (7)$$

Proof:

Consider Proposition 7 and proceed by induction. □

Next, it will be shown a relation between three families of valid inequalities. Let consider, respectively, inequalities $C_1(i_1, i_2; j)$, $C_{\text{2-1}}(i_2; j)$ and $C_{\text{2-2}}(i_1, i_2; j)$:

$$\sum_{k \in W(i_1, j)} x_{i_1k} \leq \sum_{k \in W(i_1, j)} x_{i_2k}, \quad (8)$$

$$y_j \leq \sum_{k \in W(i_2, j)} x_{i_2k}, \quad (9)$$

$$\sum_{k \in W(i_1, j)} x_{i_1k} + y_j \leq \sum_{k \in W(i_1, j) \cup W(i_2, j)} x_{i_2k}. \quad (10)$$

Remark now that if, for given $i_1$, $i_2$ and $j$, we add the two first inequalities, then we obtain

$$\sum_{k \in W(i_1, j)} x_{i_1k} + y_j \leq \sum_{k \in W(i_1, j)} x_{i_2k} + \sum_{k \in W(i_2, j)} x_{i_2k}. \quad (11)$$

Therefore, if $W(i_1, j) \cap \overline{W(i_2, j)} = \emptyset$, then inequality (11) will just be the addition of the two first ones, that is, if (8) and (9) are considered simultaneously, then (10) is redundant. But, if on the contrary, this intersection is not void, then a reduction on some coefficients will have been done. More precisely, coefficients of the terms belonging to $W(i_1, j) \cap \overline{W(i_2, j)}$, which have value 2, will have been reduced to 1, obtaining the tighter inequality $C_{\text{2-2}}(i_1, i_2; j)$ in this way.
2.3. Inequalities \( C3 \)

**Proposition 10**

Let \( i_1, i_2 \in I \) and \( j \in J \). If

\[
W(i_1, j) \subseteq W(i_2, j),
\]

then it holds that

\[
C3(i_1, i_2; j) \equiv x_{i_1j} \leq x_{i_2j}.
\] \hspace{1cm} (12)

**Proof:**

Assume that \( x_{i_1j} = 1 \) and \( x_{i_2j} = 0 \).

Since \( x_{i_1j} = 1 \), then \( y_j = 1 \) (because \( x_{i_1j} \leq y_j \)) and, from (1),

\[
y_k = 0 \ \forall k \in W'(i_1, j).
\]

Now, \( y_j = 1 \) and \( x_{i_2j} = 0 \) imply that it exists \( \bar{k} > i_2 j \) such that \( y_{\bar{k}} = 1 \).

Particularly,

\[
\bar{k} \in W'(i_2, j) \setminus W'(i_1, j) = \emptyset,
\]

which is a contradiction. \( \square \)

An immediate consequence is that, opposite to what happens in the SPLP, the SPLPO will have inequalities \( x_{ij} \leq y_j \) which are not facets. In fact, later in the preprocessing section, it will be shown that many of these inequalities can be replaced by dominant inequalities (12).

**Corollary 11**

Let \( i_1, i_2 \in I \) and \( j \in J \). If

\[
W(i_1, j) = W(i_2, j),
\]

then it holds that

\[
x_{i_1j} = x_{i_2j}.
\] \hspace{1cm} (13)
3. Preprocessed formulation

Some of the results from the previous section are used to strengthen and reduce the formulation. Two preprocessing steps can be considered.

3.1. Step 1

This first step is basically a strengthening of inequalities \( x_{ij} \leq y_j \).

Let \( j \in J \) and suppose

\[
W(i_1, j) = \cdots = W(i_s, j),
\]

with \( i_1 < \ldots < i_s \).

Now, Corollary 11 grants that

\[
x_{i_1j} = x_{ih} \quad \forall h, t \in 1, \ldots, s.
\]

In order to use a minimal formulation of this fact, we add constraints

\[
x_{i_1j} = x_{ih}, \quad j = 2, \ldots, s
\]

to the model.

Once these constraints have been added, inequalities

\[
x_{ih} \leq y_j, \quad j = 2, \ldots, s
\]

are removed because they are redundant.

Therefore, only one of the “copies” of a particular variable \( x_{ij} \) will be considered from now on. We say that “only the representantive of the set of copies is considered”.

Next, let \( i_1 \in I, j \in J \).
• If $W(i_1, j)$ is maximal, that is, there is no $i_2$ such that $W(i_1, j)$ is strictly contained in $W(i_2, j)$, then constraint
\[ x_{i_1j} \leq y_j \]
remains in the model.

• If $W(i_1, j)$ is not maximal, let $i_2$ be such that $W(i_1, j)$ is minimally contained in $W(i_2, j)$. Then inequality
\[ x_{i_1j} \leq x_{i_2j} \]
replaces $x_{i_1j} \leq y_j$.
Remark that there may be more than one such $i_2$; the one considered is the minimum lexicographical one. If all such inequalities were considered for every pair $(i_1, j)$ instead of just one, although the lower bound from the linear relaxation would be better, the model size would grow too much in size.

3.2. Step 2

Now, we deal with constraints $C2-1$.

First, two very special cases. Let $i \in I$, $j \in J$.

• If $p_{ij} = n$, then inequality $C2-1(i, j)$ becomes $y_j \leq 1$, which is removed.

• If $p_{ij} = 1$, then inequality $C2-1(i, j)$ becomes $y_j \geq x_{ij}$. This constraint, together with $x_{ij} \leq y_j$ are replaced by the immediate equality
\[ x_{ij} = y_j. \]

Only the representative of the set of copies is considered.
Next, there are a couple of results that apport additional information on how to handle inequalities $C_{2-1}$ when a set $W(i, j)$ has several copies in the sense that has been established before.

**Corollary 12**

Let $i_1, i_2 \in I$. If

$$B_1(i_1) = B_1(i_2) = j_1, \ldots, B_r(i_1) = B_r(i_2) = j_r,$$

then the inequalities considered up to now in the preprocessing method (either in the initial formulation or added when preprocessing) are enough for equality

$$\sum_{k \in W(i_1,j_r)} x_{i_1k} = \sum_{k \in W(i_1,j_r)} x_{i_2k}$$

to be implicitly established in the model.

**Proof:**

$B_t(i_1) = B_t(i_2) = j_t$ means that equality

$$x_{i_1j_t} = x_{i_2j_t}$$

has been included in the formulation due to the first preprocessing step, $t = 1, \ldots, r$.

Besides, $W(i_1, j_t) = W(i_2, j_t)$.

Since $\sum_{k \in J} x_{ihk} = 1$, $h = 1, 2$, are in the model, then

$$\sum_{t=1}^{r} x_{i_1jt} + \sum_{k \in W(i_1,j_r)} x_{i_1k} = \sum_{k \in J} x_{i_1k} = 1 =$$

$$= \sum_{k \in J} x_{i_1k} = \sum_{t=1}^{r} x_{i_2jt} + \sum_{k \in W(i_1,j_r)} x_{i_2k};$$

$$\sum_{k \in W(i_1,j_r)} x_{i_1k} = \sum_{k \in W(i_1,j_r)} x_{i_2k}$$

is already in the formulation. \qed
Corollary 13
Let \(i_1, i_2 \in I\) and \(j \in J\). Situation described in Corollary 12 happens whenever \(W(i_1, j) = W(i_2, j)\) and \(|W(i_1, j)| \geq n - 2\).

Proof:

If \(|W(i_1, j)| = n - 1\) then \(p_{i_1j} = p_{i_2j} = 1\) and \(B_1(i_1) = B_1(i_2) = j\).

If \(|W(i_1, j)| = n - 2\), then \(p_{i_1j} = p_{i_2j} = 2\) and \(B_2(i_1) = B_2(i_2) = j\).

Besides, \(\overline{W'(i_1, j)} = \overline{W'(i_2, j)} = \{h\}\) for a certain \(h\), which means that \(B_1(i_1) = B_1(i_2) = h\). □

Corollary 14
Let \(i_1, i_2 \in I\). If

\[
W_1(i_1) = W_1(i_2) = j_1, \ldots, W_r(i_1) = W_r(i_2) = j_r,
\]

then the inequalities considered up to now in the preprocessing method (either in the initial formulation or added when preprocessing) are enough for equality

\[
\sum_{k \in W(i_1,j_r)} x_{i_1k} = \sum_{k \in W'(i_1,j_r)} x_{i_2k}
\]
to be implicitly established in the model.

Proof:

\[
\sum_{k \in W(i_1,j_r)} x_{i_1k} = \sum_{t=1}^{r-1} x_{i_1j_t} = \sum_{t=1}^{r-1} x_{i_2j_t} = \sum_{k \in W(i_1,j_r)} x_{i_2k}.
\]

□

Corollary 15
Let \(i_1, i_2 \in I\) and \(j \in J\). Situation described in Corollary 14 happens whenever \(W(i_1, j) = W(i_2, j)\) and \(|W(i_1, j)| \leq 1\).
Thus, whenever \( W(i_1, j) = W(i_2, j) \), we do as follows:

- if \(|W(i_1, j)\) \(\in\) \(\{0, n - 1\}\), the case has already been considered previously;

- if \(|W(i_1, j)\) \(\in\) \(\{1, n - 2\}\), then only the representative of the set of copies is considered for the purpose of keeping constraint \( C2-1(i_1; j) \) in the formulation, that is, this constraint is removed for the copies because it is redundant.

Finally, we use inequalities \( C2-s \) to aggregate as much constraints as possible when the sets \( W(i_t, j) \) have intersection empty. The detailed explanation follows.

1. Let \( j \in J \) and let

\[
T = \{i_1 \in I : \exists i_2 \in I / W(i_1, j) \cap W(i_2, j) = \emptyset, W(i_2, j) \neq \emptyset\}.
\]

2. We consider the graph obtained from associating a node to each element of \( T \) and an undirected arc to each pair \((i_1, i_2) \in T^2\) such that

\[
W(i_1, j) \cap W(i_2, j) = \emptyset.
\]

3. We search a clique \( \{i_1, \ldots, i_s\} \) in the graph and replace inequalities

\[
\sum_{k \in W(i_t, j)} x_{ik} + y_j \leq 1, \ t = 1, \ldots, r,
\]

with

\[
\sum_{t=1}^{r} \sum_{k \in W(i_t, j)} x_{ik} + y_j \leq 1.
\]
4. Nodes $i_1, \ldots, i_r$ are removed from the graph and the process is repeated with the remaining nodes until no arcs can be drawn.

Besides, if $|W(i_1, j)| > n/2$, then

$$\sum_{k \in W(i,j)} x_{ik}$$

is replaced with

$$\sum_{k \in W(i,j)} x_{ik}$$

to reduce even more the number of non-zero elements in the problem matrix.

4. Computational study and conclusions

A computational experience was carried out in a machine AMD 1800 MHz, 512 MB RAM under Windows XP as operative system. The solver used was Xpress-MP in its version 15.25.2 for the optimizer and 1.15.04 for the IVE environment ([7]).

The data sets were partially taken from Beasley’s OR-Library ([1]). Values $c_{ij}$ and $f_j$ were taken from the data files cap131, cap132, cap133, cap 134, capa, capb and capc for the uncapacitated warehouse location problem. In the case of the three last files, which have 1000 customers and 100 facilities, just the first 75 customers and 50 facilities were considered; for these problems the $f_k$ were reduced proportionally to the product of the new values for $m$ and $n$, that is, they were multiplied by 0.0375. As a tool to generate the preferences of the customers, a random method of triangular distributions based on the costs $c_{ij}$ was used. Four instances were generated for each file. Remark that $I \neq J$ in problems 131.1 - 134.4 despite the fact that $m = n$. 16
The headers of the tables have the following meanings:

- **RR (Redundant Rows)**: Percentage of rows in the preprocessed formulation compared to the number of rows in the non-preprocessed one;

- **NZE (Non-Zero Elements)**: Percentage of non-zero elements in the preprocessed formulation compared to the number of non-zero elements in the non-preprocessed one;

- **RI (Reinforced Inequalities)**: Percentage of inequalities \( x_{ij} \leq y_j \) from the non-preprocessed formulation which are replaced with a tighter \( C3 \) inequality;

- **Method 1**: Xpress-MP solver and default formulation.

- **Method 2**: Xpress-MP solver and preprocessed formulation embedded in a branch-and-bound algorithm. Preprocessing times are included in the CPU time for solving the problem. The nearest variable to 0.5 is chosen for branching and a best-first criterium is used for choosing nodes.

- **Time**: CPU time in seconds.

- **Nodes**: Number of nodes of the branching tree.

- **Gap**: \( 100(1 - z_L/z_I) \), where \( z_L \) and \( z_I \) are, respectively, the optimal values of the linear relaxation and the integer problem.

As it can be seen in Tables 1 and 2, thanks to the preprocessing there is an interesting reduction in the problem size: 8% of the rows can be removed and what is better, near 43% of the non-zero elements. By looking at constraints \( x_{ij} \leq y_j \), it can be seen that due to the particular structure of the
problem, 83.5% of them in problems of size 50x50 and 91.4% in problems of 75x50 are replaced with tighter inequalities.

With regard to Tables 3 and 4, the following facts can be remarked:

• Computational times are clearly reduced thanks to preprocessing.

• The number of nodes in the branching tree is also reduced. Even for a few problems where the preprocessed method produces more nodes, computational times are much better.

• The gap of the default formulation is big for the small problems and huge for the big ones (near 30%). Nevertheless, the strengthened formulation reduces this gap a bit: 1.68% for the first ones and 4.16% for the second one.

Therefore, since there are significative reductions in all the computational times, we can conclude that the preprocessing rules described in the previous section are very efficient.

Conclusions

This paper has shown how to preprocess and strengthen the SPLPO formulation to solve it more efficiently: the approach here described reduces computational times considerably and improves the gap between the linear and integer solutions. For this purpose, a detailed analysis of the problem has been the key to obtain several new families of valid inequalities.

It is interesting to remark that, opposite to what happens in the SPLP, where all constraints $x_{ij} \leq y_j$ are facets, most of these inequalities can be improved for the SPLPO.
<table>
<thead>
<tr>
<th>Problem</th>
<th>RR</th>
<th>NZE</th>
<th>RI</th>
</tr>
</thead>
<tbody>
<tr>
<td>131_1</td>
<td>92.22</td>
<td>57.27</td>
<td>85.80</td>
</tr>
<tr>
<td>131_2</td>
<td>91.96</td>
<td>57.25</td>
<td>82.92</td>
</tr>
<tr>
<td>131_3</td>
<td>92.16</td>
<td>57.26</td>
<td>83.48</td>
</tr>
<tr>
<td>131_4</td>
<td>92.51</td>
<td>57.29</td>
<td>86.68</td>
</tr>
<tr>
<td>132_1</td>
<td>92.30</td>
<td>57.28</td>
<td>84.04</td>
</tr>
<tr>
<td>132_2</td>
<td>92.20</td>
<td>57.27</td>
<td>86.00</td>
</tr>
<tr>
<td>132_3</td>
<td>92.10</td>
<td>57.26</td>
<td>81.80</td>
</tr>
<tr>
<td>132_4</td>
<td>91.96</td>
<td>57.25</td>
<td>82.88</td>
</tr>
<tr>
<td>133_1</td>
<td>92.10</td>
<td>57.26</td>
<td>82.48</td>
</tr>
<tr>
<td>133_2</td>
<td>91.66</td>
<td>57.23</td>
<td>84.04</td>
</tr>
<tr>
<td>133_3</td>
<td>92.08</td>
<td>57.26</td>
<td>79.08</td>
</tr>
<tr>
<td>133_4</td>
<td>91.62</td>
<td>57.23</td>
<td>83.20</td>
</tr>
<tr>
<td>134_1</td>
<td>92.30</td>
<td>57.28</td>
<td>85.68</td>
</tr>
<tr>
<td>134_2</td>
<td>92.04</td>
<td>57.26</td>
<td>81.96</td>
</tr>
<tr>
<td>134_3</td>
<td>92.63</td>
<td>57.30</td>
<td>84.04</td>
</tr>
<tr>
<td>134_4</td>
<td>92.34</td>
<td>57.27</td>
<td>81.44</td>
</tr>
</tbody>
</table>

Table 1: Preprocessing reductions ($m = 50$, $n = 50$)

Nevertheless, the big gaps prove the problem to be very difficult, which should encourage further research on this problem.

**Acknowledgements**

Research of Lázaro Cánovas, Sergio García and Alfredo Marín has been funded by Plan Nacional de Investigación Científica, Desarrollo e Innovación
<table>
<thead>
<tr>
<th>Problem</th>
<th>RR</th>
<th>NZE</th>
<th>RI</th>
</tr>
</thead>
<tbody>
<tr>
<td>a75_50_1</td>
<td>90.77</td>
<td>57.16</td>
<td>92.29</td>
</tr>
<tr>
<td>a75_50_2</td>
<td>90.73</td>
<td>57.16</td>
<td>89.28</td>
</tr>
<tr>
<td>a75_50_3</td>
<td>90.67</td>
<td>57.16</td>
<td>91.76</td>
</tr>
<tr>
<td>a75_50_4</td>
<td>90.75</td>
<td>57.16</td>
<td>90.88</td>
</tr>
<tr>
<td>b75_50_1</td>
<td>90.94</td>
<td>57.18</td>
<td>92.11</td>
</tr>
<tr>
<td>b75_50_2</td>
<td>90.72</td>
<td>57.16</td>
<td>92.08</td>
</tr>
<tr>
<td>b75_50_3</td>
<td>90.77</td>
<td>57.17</td>
<td>93.28</td>
</tr>
<tr>
<td>b75_50_4</td>
<td>90.92</td>
<td>57.18</td>
<td>89.04</td>
</tr>
<tr>
<td>c75_50_1</td>
<td>90.84</td>
<td>57.17</td>
<td>92.37</td>
</tr>
<tr>
<td>c75_50_2</td>
<td>90.84</td>
<td>57.17</td>
<td>90.48</td>
</tr>
<tr>
<td>c75_50_3</td>
<td>90.75</td>
<td>57.16</td>
<td>90.83</td>
</tr>
<tr>
<td>c75_50_4</td>
<td>90.71</td>
<td>57.16</td>
<td>92.43</td>
</tr>
</tbody>
</table>

Table 2: Preprocessing reductions ($m = 75$, $n = 50$)

Tecnológica (I+D+I) together with the European Regional Development Funds (ERDF), project TIC2003-05982-C05-03.

References


<table>
<thead>
<tr>
<th>Problem</th>
<th>Method 1</th>
<th>Method 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Nodes</td>
</tr>
<tr>
<td>131.1</td>
<td>25</td>
<td>109</td>
</tr>
<tr>
<td>131.2</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>131.3</td>
<td>11</td>
<td>32</td>
</tr>
<tr>
<td>131.4</td>
<td>15</td>
<td>52</td>
</tr>
<tr>
<td>132.1</td>
<td>32</td>
<td>115</td>
</tr>
<tr>
<td>132.2</td>
<td>39</td>
<td>197</td>
</tr>
<tr>
<td>132.3</td>
<td>40</td>
<td>137</td>
</tr>
<tr>
<td>132.4</td>
<td>10</td>
<td>25</td>
</tr>
<tr>
<td>133.1</td>
<td>12</td>
<td>35</td>
</tr>
<tr>
<td>133.2</td>
<td>7</td>
<td>25</td>
</tr>
<tr>
<td>133.3</td>
<td>33</td>
<td>140</td>
</tr>
<tr>
<td>133.4</td>
<td>14</td>
<td>60</td>
</tr>
<tr>
<td>134.1</td>
<td>32</td>
<td>112</td>
</tr>
<tr>
<td>134.2</td>
<td>11</td>
<td>39</td>
</tr>
<tr>
<td>134.3</td>
<td>38</td>
<td>168</td>
</tr>
<tr>
<td>134.4</td>
<td>38</td>
<td>231</td>
</tr>
</tbody>
</table>

Table 3: Computational times ($m = 50$, $n = 50$)
<table>
<thead>
<tr>
<th>Problem</th>
<th>Method 1</th>
<th></th>
<th>Method 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Nodes</td>
<td>Gap</td>
<td>Time</td>
</tr>
<tr>
<td>a75_50_1</td>
<td>1329</td>
<td>3084</td>
<td>27.67</td>
<td>667</td>
</tr>
<tr>
<td>a75_50_2</td>
<td>1457</td>
<td>3998</td>
<td>27.22</td>
<td>775</td>
</tr>
<tr>
<td>a75_50_3</td>
<td>1038</td>
<td>1778</td>
<td>27.14</td>
<td>848</td>
</tr>
<tr>
<td>a75_50_4</td>
<td>1062</td>
<td>2359</td>
<td>24.29</td>
<td>581</td>
</tr>
<tr>
<td>b75_50_1</td>
<td>1022</td>
<td>2674</td>
<td>28.11</td>
<td>433</td>
</tr>
<tr>
<td>b75_50_2</td>
<td>1734</td>
<td>5254</td>
<td>31.06</td>
<td>909</td>
</tr>
<tr>
<td>b75_50_3</td>
<td>884</td>
<td>3015</td>
<td>28.22</td>
<td>502</td>
</tr>
<tr>
<td>b75_50_4</td>
<td>383</td>
<td>966</td>
<td>27.28</td>
<td>256</td>
</tr>
<tr>
<td>c75_50_1</td>
<td>976</td>
<td>4015</td>
<td>31.85</td>
<td>783</td>
</tr>
<tr>
<td>c75_50_2</td>
<td>626</td>
<td>2199</td>
<td>29.04</td>
<td>427</td>
</tr>
<tr>
<td>c75_50_3</td>
<td>296</td>
<td>1078</td>
<td>28.47</td>
<td>131</td>
</tr>
<tr>
<td>c75_50_4</td>
<td>1072</td>
<td>3953</td>
<td>30.47</td>
<td>746</td>
</tr>
</tbody>
</table>

Table 4: Computational times ($m = 75$, $n = 50$)