European Journal of Operational Research 278 (2019) 120–135

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Improved mathematical model and bounds for the crop rotation scheduling problem with adjacency constraints

Geraldo Regis Mauri

Federal University of Espírito Santo - UFES, Alegre - ES, 29500-000 Brazil

ARTICLE INFO

Article history:
Received 19 September 2018
Accepted 15 April 2019
Available online 18 April 2019

Keywords:
Crop rotation scheduling
Combinatorial optimization
Column generation
Lagrangian relaxation
Lagrangian decomposition

ABSTRACT

The Crop Rotation Scheduling Problem (CRSP) consists of alternating crops in neighboring plots during a period of time in order to find a planting schedule that satisfies a particular objective subject to some constraints such as the non-simultaneous cultivation of crops from the same botanical family in neighboring plots. In this work, some assumptions are proposed to improve a mathematical model presented in the literature, making it more general and easier to be solved by a commercial solver. In addition, five different relaxation approaches are proposed to find bounds and solutions for the CRSP. A detailed set of instances is also proposed, and a column generation procedure presented in the literature is implemented in order to perform a fair comparison of results. Computational experiments were performed indicating the improvements provided by the new model and the capability of some relaxation methods to generate high-quality solutions and bounds for the CRSP.

1. Introduction

Brazil is a country where agricultural production is mainly based on monoculture, which is also used in many other countries around the world where agriculture is more intensive. Although monoculture is still employed, it is an unsustainable technique that leads to several problems such as the overuse of toxic pesticides and synthetic fertilisers as well as reduced soil fertility.

Crop rotation appears as a sustainable alternative to monoculture preserving productive resources through the diversification of crops available to be planted. As pointed out by Jaehn (2016), crop rotation can be seen as a sustainable operation in agriculture due to its economic and environmental perspectives.

Brankatschk and Finkbeiner (2015) discussed the effects of crop rotation caused by changes in physical, chemical and biological properties of the soil over time. We can highlight some positive effects such as the reduction of infestation by parasites, the improvement in nutrient availability in the soil and the lower economic and climatic risks due to diversified production.

In a general way, crop rotation can be processed by scheduling crops planting in plots during a period of time, thus arising a problem known as Crop Rotation Scheduling Problem (CRSP). Several different approaches concerning CRSP can be found in the literature. One of the first works was presented by Clarke (1989) who reported some combinatorial aspects for maximizing the return from land use.

Some works like Filippi, Mansini, and Stevanato (2017) consider monetary objectives and non-perishable crops while others like Alfandari, Plateau, and Schepler (2015) and Santos, Michelon, Arenales, and Santos (2011) also deal with environmental aspects and perishable crops.

Alfandari et al. (2015) proved the NP-hardness of the CRSP with the objective of minimizing the area required to cover seasonal crops demands. The authors proposed a 0–1 linear programming model and a branch-and-price-and-cut algorithm to solve it. A branch-and-price-and-cut algorithm was also presented by Santos, Munari, Costa, and Santos (2015) to solve a mixed integer model that also determines the size of the plots.

Santos, Costa, Arenales, and Santos (2010) presented a 0–1 linear programming model for the CRSP aiming to maximize the plots occupation considering demand constraints. The authors proposed a column generation procedure to solve the model. A similar procedure was also employed by Santos et al. (2011) to solve another model for the CRSP without demand constraints. The authors were the first to introduce adjacency constraints for the CRSP preventing crops of the same botanical family to be cultivated at the same time in neighboring plots.

A model adapted from that proposed by Santos et al. (2011) was presented by Aliano Filho, Florentino, and Pato (2014). Demand constraints were introduced as well as an objective function which deals with maximizing the profit of planted crops. Metaheuristics were employed to solve the problem.

E-mail address: geraldo.mauri@ufes.br

https://doi.org/10.1016/j.ejor.2019.04.016

Despite several approaches to the CRSP, we must stress that only a few authors have dealt with adjacency and succession constraints for crops of the same family as well as the planting of green manuring crops and fallow periods. Santos et al. (2011) and Aliano Filho et al. (2014) dealt with that focusing on maximizing plots occupation and the profit, respectively.

Crop prices can vary weekly or even daily making it difficult to plan a rotation accurately. Then, in this work we handle both objectives, i.e. we maximize plots occupation and profit. The first objective is more suitable to deal with real situations since it is not dependent on the market price variability. We just try to keep the plots occupied throughout the rotation period. However, the second objective was considered in this work in order to evaluate the performance of the presented methods when applied to a more complex problem in which plots of different sizes and estimated prices for the crops must be considered.

Either way, both objectives are capable of handling different real cases aiming for solutions that are both sustainable and profitable. So, to deal with these objectives we propose some improvements in the mathematical model presented by Santos et al. (2011) and a detailed set of instances based on real-world values. In order to find bounds and solutions for the improved model, five different relaxation approaches are proposed: *Lagrangian Relaxation*, *Lagrangian Relaxation with Clusters* (LC), *Lagrangian Decomposition* (LD), *Column Generation based on* LC and *Column Generation based on* LD. Finally, we implemented the column generation procedure presented by Santos et al. (2011) in order to perform a fair comparison of results.

The remainder of this paper is organized as follows. Section 2 describes the mathematical model for the CRSP with the improvements proposed in this work. The different methods for relaxing the CRSP are presented in Section 3 followed by the algorithms coded to solve them in Section 4. Section 5 presents the instances proposed in this work and the procedure performed to generate them. Finally, computational experiments are reported in Section 6, followed by conclusions in Section 7.

2. Mathematical model for the CRSP

The mathematical model used in this work is presented below. It is based on those presented by Santos et al. (2011) and Aliano Filho et al. (2014). However, some assumptions have been considered to improve it and make it more general, covering different situations.

Let M be the number of periods in each rotation (in a predefined unit of time) considering K plots distributed into a cropping area. Each plot k ($k = 1..K$) has a set A_k of neighboring (adjacent) plots and an area r_k (in a predefined unit of space), defining the exact arrangement (layout) of the plots into the cropping area.

N different crops classified into F botanical families are available to be planted, considering that each botanical family f ($f = 1..F$) is composed of a disjoint set B_f of crops. The crops are divided into two independent sets C and D ($N = |C \cup D|$), considering C a set of trade crops (with commercial value) and D a set of crops which can be used for green manuring.

Each crop i ($i = 1..N$) has a profit p_i (expected profit per unit area), a production time t_i (including the time to prepare the soil until harvesting), and a set of periods $I_i \subseteq \{1..M\}$ in which crop i can be planted. The set I_i is defined according to the annual planting season for each crop i.

Crops for green manuring have no profit, i.e. $p_i = 0\ \forall i \in D$, and for convenience of notation, the fallow period is represented as a "virtual crop" n ($n = N + 1$) with $p_n = 0$ and $I_n = \{1..M\}$.

Two different values for the profit for planting crop i in plot k (c_{ik}) are considered in this work, as detailed in Section 2.1, and the decision variables x_{ijk} are defined in (7), where $x_{ijk} = 1$ if crop i is planted during period j in plot k, 0 otherwise, $i = 1..n$, $j \in I_i$ and $k = 1..K$. The CRSP can be modelled as follows:

$$z_{CRSP} = \max \sum_{k=1}^{K} \sum_{i \in C} \sum_{j \in I_i} c_{ik} x_{ijk} \tag{1}$$

Subject to:

$$\sum_{i=1}^{n} \sum_{q=0}^{t_i - 1} x_{i(j-q)k} \le 1 \qquad j = 1..M,\ k = 1..K \tag{2}$$

$$\sum_{i \in D} \sum_{j \in I_i} x_{ijk} \ge 1 \qquad k = 1..K \tag{3}$$

$$\sum_{j=1}^{M} x_{njk} \ge 1 \qquad k = 1..K \tag{4}$$

$$\sum_{i \in B_f} \sum_{q=0}^{t_i + t_n - 1} x_{i(j-q)k} \le 1 \qquad f = 1..F,\ j = 1..M,\ k = 1..K \tag{5}$$

$$\sum_{i \in B_f} \sum_{q=0}^{t_i - 1} \left[m_k x_{i(j-q)k} + \sum_{u \in A_k, u > k} x_{i(j-q)u} \right] \le m_k \qquad f = 1..F,\ j = 1..M,\ k = 1..K \tag{6}$$

$$x_{ijk} \in \{0, 1\} \qquad i = 1..n,\ j \in I_i,\ k = 1..K \tag{7}$$

Considering that a rotation is a cycle of M periods, the term $j - q \le 0$ in constraints (2), (5) and (6) indicates that a crop can be planted in the final periods and its harvesting can be made in the initial periods, e.g. considering a one-year rotation, a crop can be planted in November and harvested in February. So, in these cases the term $j - q$ must be replaced by $j - q + M$. This is also considered by Santos et al. (2011) and Aliano Filho et al. (2014). In addition, only $j - q$ (or $j - q + M$) $\in I_i$ for the respective crop i ($i = 1..n$) must be considered in these constraints, i.e. any crop i must be planted during its planting season I_i. These conditions must also be considered for the methods described in Section 3.

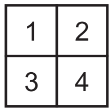
2.1. Objective function (1)

Two distinct objective functions for the CRSP are considered in this work. The first objective considers $c_{ik} = t_i$ and it was presented by Santos et al. (2011) for maximizing the total occupation of the plots with trade crops. The second objective considers $c_{ik} = r_k p_i$ and consists of maximizing the profit of the trade crops planted in the plots, and it is similar to that presented by Aliano Filho et al. (2014) disregarding the required demand.

2.2. Constraints (2), (3), (4) and (5)

Constraints (2) ensure that two crops cannot occupy the same place at the same time. These constraints are identical to those presented by Santos et al. (2011) and Aliano Filho et al. (2014).

Constraints (3) and (4) ensure that each plot has at least one green manuring crop and one fallow period, respectively. These constraints are identical to the ones presented by Aliano Filho et al. (2014). Santos et al. (2011) consider that these crops and fallow must be planted only once in each plot. But, both approaches are suitable, since fallow and green manuring crops have not commercial value (profit) and are not considered in (1), they will tend to be planted only once in each lot. Even so, they may be planted more than once without making a solution unfeasible.



$$\overbrace{\begin{aligned} m_1x_{ij1}+x_{ij2}+x_{ij3}&\le m_1\\ m_2x_{ij2}+x_{ij1}+x_{ij4}&\le m_2\\ m_3x_{ij3}+x_{ij1}+x_{ij4}&\le m_3\\ m_4x_{ij4}+x_{ij2}+x_{ij3}&\le m_4\end{aligned}}^{\text{without } u>k} \qquad \overbrace{\begin{aligned} m_1x_{ij1}+x_{ij2}+x_{ij3}&\le m_1\\ m_2x_{ij2}+x_{ij4}&\le m_2\\ m_3x_{ij3}+x_{ij4}&\le m_3\end{aligned}}^{\text{with } u>k}$$

Fig. 1. Example of a cropping area and adjacency constraints.

Constraints (5) guarantee that crops of the same family (including green manuring crops, but excluding fallow) cannot be planted in sequence on the same plot. These constraints are similar to those presented by Santos et al. (2011) and Aliano Filho et al. (2014). However, they are modified by adding the term $t_n - 1$ on the superior limit for the second summation aiming to ensure a minimum period of time (equivalent to a fallow) between two crops of the same family. Otherwise, when the fallow time t_n is greater than one unit of time, two crops of the same family could be planted with an interval not sufficient for a fallow.

2.3. Adjacency constraints (6)

Adjacency constraints (6) are global constraints linking several plots k together, i.e. they prevent crops of the same botanical family to be cultivated at the same time in neighboring plots. In these constraints, m_k is the optimal value for a classical graph problem of finding the maximum independent set of nodes for each plot k $(k = 1..K)$ as shown below:

$$m_k = \max \sum_{u \in A_k, u>k} y_u$$

$$\text{s.t.}$$

$$y_u + y_v \le 1 \qquad u = 1..K, v \in A_u, v > u$$

$$y_u \in \{0, 1\} \qquad u = 1..K$$

Constraints (6) and the model above are similar to those from Santos et al. (2011), but two modifications are made: the superior limit for the second summation in (6) has been replaced from t_i to $t_i - 1$; and the condition $u > k$ has been added at the limit for the internal summation in (6). The same condition has been considered for calculating m_k in the model above.

The first modification has been made to adjust the constraints to their purpose ensuring that two crops of the same family can not be cultivated in adjacent plots simultaneously. On the other hand, the use of t_i requires at least one period of time between two crops of the same family to be cultivated in adjacent plots, but this period is not necessary in practice.

The condition $u > k$ has been used to eliminate redundant adjacency constraints. For simplification, let i be the only crop of any family f and $I_i = \{1\}$. Given any two adjacent plots u and k, $u > k$, $A_k = \{u\}$, $A_u = \{k\}$, the resulting adjacency constraints for family f in these plots without using the condition $u > k$ can be written as follows:

$$m_k x_{i1k} + x_{i1u} \le m_k$$

$$m_u x_{i1u} + x_{i1k} \le m_u$$

It is easy to note that the above constraints are redundant as $m_u = m_k = 1$, and using the condition $u > k$ the second constraint will be eliminated. By analogy, the same can be verified for any pair of plots for each family f $(f = 1..F)$ and period j $(j = 1..M)$. The number of constraints eliminated will depend on the arrangement of plots into the cropping area and on the number of botanical families and periods. Besides reducing the number of constraints, the number of variables in each adjacency constraint can also be reduced considerably depending on the crops available for planting.

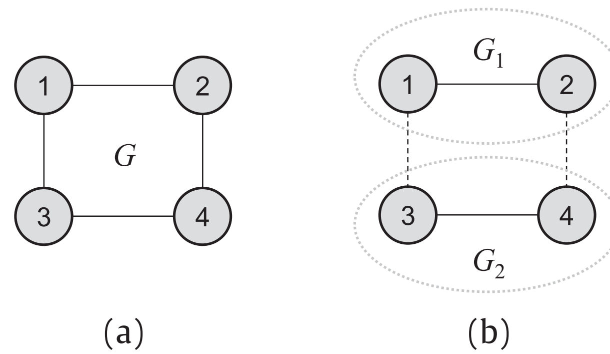


Fig. 2. Partitioning of graph G associated with Fig. 1.

Fig. 1 illustrates an example of a cropping area with four plots and the resulting adjacency constraints considering only one crop i for a period j. We can observe that one constraint and two variables were eliminated. So, applying the same condition $(u > k)$ for every family and period, we can get a significant reduction in the number of adjacency constraints.

A numerical comparison between the model with the adjacency constraints as proposed in this work and as presented by Santos et al. (2011) is reported in Section 6.1.

The following adjacency constraints (8) are presented by Aliano Filho et al. (2014). We can observe that they are similar to those described in (6), including the use of term $t_i - 1$ instead of t_i. However, the limit m_k is replaced by the number of plots K and the condition $u > k$ is not considered in the third summation.

$$\sum_{i \in B_f} \sum_{q=0}^{t_i-1} \sum_{u \in A_k} x_{i(j-q)u} \le K\left(1 - \sum_{i \in B_f} \sum_{q=0}^{t_i-1} x_{i(j-q)k}\right) \quad \begin{matrix} f = 1..F,\ j = 1..M, \\ k = 1..K \end{matrix} \tag{8}$$

Although these constraints do not need to solve the maximum independent set problem stated before, the resulting mathematical model becomes weaker in obtaining both bounds and solutions for the CRSP. In addition, m_k was obtained in very low computational time (<0.02 seconds) for the experiments reported in this work. A numerical comparison with the model using constraints (8) with and without condition $u > k$ is also presented in Section 6.1.

3. Methods for relaxing the CRSP

As observed by Santos et al. (2011), all K plots could be solved separately when removing linking (adjacency) constraints (6). Based on this, in this work we propose some solution strategies related to partially relax these constraints. The main idea is to partition the plots in clusters and to relax only the constraints associated with adjacency among different clusters.

A cropping area can be represented by a connected planar graph $G(V, E)$ in which the set of vertices V correspond to the plots and the set of edges is composed of pairs of vertices $(k, u) \in E$ if plots k and u $(u > k)$ are adjacent.

Fig. 2(a) depicts the graph G related to the cropping area presented previously in Fig. 1. The graph G can be partitioned into L

($L \leq K$) subgraphs $G_l(V_l, E_l)$, $l = 1..L$, with $V = V_1 \cup V_2 \cup \ldots \cup V_L$, $V_i \cap V_j = \emptyset \ \forall \ i, j = 1..L, \ i \neq j$; and E_l is composed of every edge (k, u) with $u > k$ and $k \in V_l$, $l = 1..L$. In addition, each set V_l ($l = 1..L$) can be divided into two disjoint subsets V_l' and V_l'' ($V_l = V_l' \cup V_l''$, $V_l' \cap V_l'' = \emptyset$). For all $k \in V_l$ ($l = 1..L$), $k \in V_l''$ if $\exists u \in A_k$, $u > k \,|\, u \notin V_l$, and $k \in V_l'$ otherwise, i.e. V_l' is composed of plots k whose all adjacent plots $u > k$ are in the same subgraph, and V_l'' is composed of plots for which there is at least one adjacent plot that is not in the same subgraph.

Fig. 2(b) illustrates a partitioning of graph G into two subgraphs G_1 and G_2. We can observe two edges (dashed) linking the subgraphs, and then $V_1 = \{1, 2\}$, $V_2 = \{3, 4\}$, $E_1 = \{(1, 2), (1, 3), (2, 4)\}$, $E_2 = \{(3, 4)\}$, $V_1' = \{\emptyset\}$, $V_1'' = \{1, 2\}$, $V_2' = \{3, 4\}$ and $V_2'' = \{\emptyset\}$.

The model (1)–(7) considering the cropping area partitioned into L ($L \leq K$) subareas can be rewritten as follows:

$$z^L_{CRSP} = \max \sum_{l=1}^{L} \left(\sum_{k \in V_l} \sum_{i \in C} \sum_{j \in I_i} c_{ik} x_{ijk} \right) \tag{9}$$

Subject to:

$$\sum_{i=1}^{n} \sum_{q=0}^{t_i - 1} x_{i(j-q)k} \leq 1 \qquad j = 1..M, \quad k \in V_l, l = 1..L \tag{10}$$

$$\sum_{i \in D} \sum_{j \in I_i} x_{ijk} \geq 1 \qquad k \in V_l, l = 1..L \tag{11}$$

$$\sum_{j=1}^{M} x_{njk} \geq 1 \qquad k \in V_l, l = 1..L \tag{12}$$

$$\sum_{i \in B_f} \sum_{q=0}^{t_i + t_n - 1} x_{i(j-q)k} \leq 1 \qquad f = 1..F, \ j = 1..M, \ k \in V_l, l = 1..L \tag{13}$$

$$\sum_{i \in B_f} \sum_{q=0}^{t_i - 1} \left[m_k x_{i(j-q)k} + \sum_{u \in A_k, u > k} x_{i(j-q)u} \right] \leq m_k \qquad f = 1..F, \ j = 1..M, \ k \in V_l', l = 1..L \tag{14}$$

$$\sum_{i \in B_f} \sum_{q=0}^{t_i - 1} \left[m_k x_{i(j-q)k} + \sum_{u \in A_k, u > k} x_{i(j-q)u} \right] \leq m_k \qquad f = 1..F, \ j = 1..M, \ k \in V_l'', l = 1..L \tag{15}$$

$$x_{ijk} \in \{0, 1\} \qquad i = 1..n, j \in I_i, \ k \in V_l, l = 1..L \tag{16}$$

It is easy to notice that (9)–(14) and (16) are independent for each subgraph (subarea) l ($l = 1..L$). However, constraints (15) consider the subgraphs with at least one "external" edge $(k, u) \,|\, k \in V_l$ and $u \notin V_l$, i.e. $k \in V_l''$, $l = 1..L$. Thus, a graph partitioning with few edges linking the subgraphs will reduce the number of constraints (15), resulting in more independent subgraphs and consequently a stronger relaxation for the model. Some alternatives to address this situation are presented in the following.

3.1. Lagrangian relaxation with clusters

Given any problem represented by a graph, the Lagrangian relaxation with clusters (LC) consists of relaxing constraints in a Lagrangian way after partitioning the graph into clusters of vertices (Ribeiro & Lorena, 2008).

The main difference between the LC and the "traditional" Lagrangian relaxation (LR), presented in Section 3.5, is related to the partitioning of the problem into independent subproblems, i.e. LR handles with only one problem while LC partitions the problem into some subproblems which can be solved separately.

The LC can be applied directly to model (9)–(16), since the graph G is partitioned into L subgraphs which can be considered as clusters. In this model, only constraints (15) connect the subproblems and then they can be considered as the "complicating" constraints and can be relaxed in the Lagrangian way with multipliers $\lambda_{fjk} \geq 0$ ($f = 1..F$, $j = 1..M$ and $k \in V_l''$, $l = 1..L$). The LC model for the CRSP is described below.

$$\bar{z}_{LC} = \max \sum_{l=1}^{L} \left(\sum_{k \in V_l} \sum_{i \in C} \sum_{j \in I_i} c_{ik} x_{ijk} \right) + \sum_{f=1}^{F} \sum_{j=1}^{M} \sum_{l=1}^{L} \sum_{k \in V_l''} \left\{ \lambda_{fjk} \left(m_k - \sum_{i \in B_f} \sum_{q=0}^{t_i - 1} \left[m_k x_{i(j-q)k} + \sum_{u \in A_k, u > k} x_{i(j-q)u} \right] \right) \right\} \tag{17}$$

Subject to: (10) – (14) and (16).

Now, the previous model can be easily divided into L ($L \leq K$) independent subproblems with each subproblem l ($l = 1..L$) defined as follows:

$$\bar{z}^l_{LC} = \max \sum_{k \in V_l} \sum_{i \in C} \sum_{j \in I_i} c_{ik} x_{ijk} - \sum_{f=1}^{F} \sum_{j=1}^{M} \sum_{k \in V_l''} \left(\lambda_{fjk} m_k \sum_{i \in B_f} \sum_{q=0}^{t_i - 1} x_{i(j-q)k} \right) - \sum_{f=1}^{F} \sum_{j=1}^{M} \sum_{\substack{k \in V_h'' \\ h \neq l}} \left(\lambda_{fjk} \sum_{i \in B_f} \sum_{q=0}^{t_i - 1} \sum_{\substack{u \in A_k \\ u > k \\ u \in V_l}} x_{i(j-q)u} \right) + \sum_{f=1}^{F} \sum_{j=1}^{M} \sum_{k \in V_l''} \left(\lambda_{fjk} m_k \right) \tag{18}$$

Subject to: (10) – (14) and (16) for only l.

Finally, the Lagrangian relaxation with clusters (LC) for the CRSP partitioned into L subproblems is redefined in (19), and its corresponding Lagrangian dual is defined in (20).

$$\bar{z}_{LC} = \sum_{l=1}^{L} \bar{z}^l_{LC} \tag{19}$$

$$w_{LC} = \min \left\{ \bar{z}_{LC} : \lambda_{fjk} \geq 0, f = 1..F, j = 1..M, k \in V_l'', l = 1..L \right\} \tag{20}$$

The Lagrangian dual (20) can be solved by a subgradient algorithm (see Section 4.1) that will manage the subproblems and update the Lagrangian multipliers. The value w_{LC} obtained by the Lagrangian dual is an upper bound for the CRSP and it may not be related to a feasible solution. So, a Lagrangian heuristic must be employed to make a solution feasible. For notation, z_{LC} is used hereafter to denote the value of a feasible solution obtained by the LC.

3.2. Column generation for LC

Mauri and Lorena (2012a) proposed a column generation approach as an alternative way to find the multipliers and to solve the dual for a Lagrangian relaxation with clusters (LC) applied to a quadratic programming problem.

The idea presented in that work can be adapted and applied to the LC described in the previous section. The column generation technique uses a coordinating problem (or restricted master problem – RMP) and subproblems for generating columns to the RMP. Dual variables from the RMP are used for guiding subproblems on the search for new columns.

Firstly, the model (9)–(16) can be reformulated by using the Dantzig-Wolfe decomposition. Since constraints (15) are the only ones connecting the subproblems they are used to define the RMP while the other constraints are handled in the subproblems.

Let S^l be a subset of all solutions for subproblem l. Each solution $s \in S^l$ is composed of binary variables $\hat{x}^s_{ijk}$ ($i = 1..n$, $j \in I_i$ and $k \in V_l$). In addition, let $o \in S^h$ be a solution for a subproblem h ($h \neq l$) and μ_{sl} be the decision variable for each solution $s \in S^l$. Therefore, the linear relaxation for the RMP is given as follows:

$$\bar{z}^{RMP}_{CGLC} = \max \sum_{l=1}^{L} \sum_{k \in V_l} \sum_{i \in C} \sum_{j \in I_i} \sum_{s \in S^l} c_{ik} \hat{x}^s_{ijk} \mu_{sl} \tag{21}$$

Subject to:

$$\sum_{i \in B_f} \sum_{q=0}^{t_i-1} \sum_{s \in S^l} \left[m_k \mu_{sl} \hat{x}^s_{i(j-q)k} + \sum_{\substack{u \in A_k \\ u > k \\ u \in V_h}} \mu_{oh} \hat{x}^o_{i(j-q)u} \right] \le m_k \quad \begin{array}{l} f = 1..F, \\ j = 1..M, \\ k \in V''_l, \\ l = 1..L \end{array} \tag{22}$$

$$\sum_{s \in S^l} \mu_{sl} = 1 \qquad l = 1..L \tag{23}$$

$$\mu_{sl} \ge 0 \qquad \begin{array}{l} l = 1..L, \\ s \in S^l \end{array} \tag{24}$$

From the RMP, dual variables ε_{fjk} ($f = 1..F$, $j = 1..M$ and $k \in V''_l, l = 1..L$) are associated with constraints (22) and φ_l ($l = 1..L$) with the convexity constraints (23). ε_{fjk} are used to replace the Lagrangian multipliers λ_{fjk} in (18) resulting in a model with value denoted as $\bar{z}^l_{CGLC}$, while φ_l are used to calculate the reduced cost for a column related to the subproblem l.

A new column for subproblem l is inserted into the RMP if its reduced cost ς_l is positive ($\varsigma_l = \bar{z}^l_{CGLC} - \varphi_l > 0$). So, the RMP guides the subproblems' solutions through the dual variables looking for a good solution for the CRSP.

In the column generation procedure, an optimal solution for the RMP is a valid upper bound for the CRSP if there is no column that improves $\bar{z}^{RMP}_{CGLC}$, i.e. if an optimal solution for the RMP is found. However, another upper bound can be calculated at each iteration of the procedure by using (25), that states a Lagrangian dual in which dual variables obtained from the RMP are used to replace the Lagrangian multipliers employed by the LC presented in Section 3.1.

$$w_{CGLC} = \sum_{l=1}^{L} \bar{z}^l_{CGLC} \tag{25}$$

In addition, $\bar{z}^{RMP}_{CGLC}$ is obtained by a linear relaxation and an optimal solution from the RMP may not be feasible for the CRSP. So, a feasible solution can be obtained by solving the RMP with all generated columns using integer variables μ_{sl}, i.e. replacing (24) by $\mu_{sl} \in \{0, 1\}$. A feasible solution obtained from the RMP is denoted hereafter as z_{CGLC}.

3.3. Lagrangian decomposition

The Lagrangian decomposition (LD) is a special case of Lagrangian relaxation that consists of partitioning a problem into subproblems creating copies of some decision variables (Guignard & Kim, 1987; Mauri, Ribeiro, & Lorena, 2010).

While constraints (15) are relaxed in the LC, they are kept inside subproblems in the LD by using copy variables. New constraints must be introduced to ensure the same value for copy and "original" variables. These constraints will be the only ones connecting the subproblems and then they are relaxed in a Lagrangian way.

Let $\tilde{x}^l_{i(j-q)u}$ be a copy of the variable $x_{i(j-q)u}$ into subproblem l ($u \notin V_l$). Constraints (15) can be redefined by (26) separating the internal summation from (15) into two summations in (26). Adjacent plots inside the same subproblem are considered in the first summation while copy variables are used to handle with plots which do not belong to the subproblem in the second summation.

Constraints (27) ensure that all copies have the same value as the original variables, and then constraints (15) in the model (9)–(16) can be replaced by constraints (26) and (27) resulting in an equivalent model.

$$\sum_{i \in B_f} \sum_{q=0}^{t_i-1} \left[m_k x_{i(j-q)k} + \sum_{\substack{u \in A_k \\ u > k \\ u \in V_l}} x_{i(j-q)u} + \sum_{\substack{u \in A_k \\ u > k \\ u \notin V_l}} \tilde{x}^l_{i(j-q)u} \right] \le m_k \quad \begin{array}{l} f = 1..F, \\ j = 1..M, \\ k \in V''_l, \\ l = 1..L \end{array} \tag{26}$$

$$\tilde{x}^l_{i(j-q)u} = x_{i(j-q)u} \qquad \begin{array}{l} f = 1..F, i \in B_f, j = 1..M, q = 0..(t_i - 1), \\ l = 1..L, k \in V''_l \mid u \in A_k, u > k, u \notin V_l \end{array} \tag{27}$$

Now, constraints (27) can be considered as the complicating constraints which must be relaxed with unrestricted multipliers $\lambda^l_{i(j-q)u}$ ($f = 1..F, i \in B_f, j = 1..M, q = 0..(t_i - 1), l = 1..L, k \in V''_l \mid u \in A_k, u > k, u \notin V_l$), resulting in the LD model described below.

$$\bar{z}_{LD} = \max \sum_{l=1}^{L} \left(\sum_{k \in V_l} \sum_{i \in C} \sum_{j \in I_i} c_{ik} x_{ijk} \right) + \sum_{f=1}^{F} \sum_{i \in B_f} \sum_{j=1}^{M} \sum_{q=0}^{t_i-1} \sum_{l=1}^{L} \sum_{k \in V''_l} \sum_{\substack{u \in A_k \\ u > k \\ u \notin V_l}} \left\{ \lambda^l_{i(j-q)u} \left(\tilde{x}^l_{i(j-q)u} - x_{i(j-q)u} \right) \right\} \tag{28}$$

Subject to:

(10)–(14), (16) and (26), for only l; and

$$\tilde{x}^l_{i(j-q)u} \in \{0, 1\} \qquad \begin{array}{l} f = 1..F, i \in B_f, j = 1..M, q = 0..(t_i - 1), \\ l = 1..L, k \in V''_l \mid u \in A_k, u > k, u \notin V_l \end{array} \tag{29}$$

Since constraints (27) are relaxed, constraints (29) must be inserted into the model to ensure correct values for the copy variables.

Similar to that presented in Section 3.1 for the LC, the previous model can be divided into L ($L \le K$) independent subproblems with each subproblem l ($l = 1..L$) defined as follows:

$$\bar{z}^l_{LD} = \max \sum_{k\in V_l}\sum_{i\in C}\sum_{j\in I_i} c_{ik}x_{ijk} + \sum_{f=1}^{F}\sum_{i\in B_f}\sum_{j=1}^{M}\sum_{q=0}^{t_i-1}\sum_{l=1}^{L}\sum_{k\in V_l''}\sum_{\substack{u\in A_k\\ u>k\\ u\notin V_l}} \lambda^l_{i(j-q)u}\bar{x}^l_{i(j-q)u} - \sum_{f=1}^{F}\sum_{i\in B_f}\sum_{j=1}^{M}\sum_{q=0}^{t_i-1}\sum_{l=1}^{L}\sum_{\substack{k\in V_h''\\ h\neq l}}\sum_{\substack{u\in A_k\\ u>k\\ u\in V_l}} \lambda^h_{i(j-q)u}x_{i(j-q)u} \quad (30)$$

Subject to: (10)–(14), (16), (26) and (29), for only l.

The Lagrangian decomposition (LD) for the CRSP partitioned into L subproblems is redefined in (31), and its corresponding Lagrangian dual is defined in (32).

$$\bar{z}_{LD} = \sum_{l=1}^{L}\bar{z}^l_{LD} \quad (31)$$

$$w_{LD} = \min\left\{\bar{z}_{LD} : \lambda^l_{i(j-q)u} \text{ unrestricted}, \begin{array}{l} f=1..F, i\in B_f, j=1..M,\\ q=0..(t_i-1), l=1..L,\\ k\in V_l'' \mid u\in A_k, u>k, u\notin V_l \end{array}\right\} \quad (32)$$

Based on that presented in Section 3.1 for the LC, the Lagrangian dual (32) can be solved by a subgradient algorithm resulting in the value w_{LD} that is an upper bound for the CRSP. z_{LD} is used afterwards to denote the value of a feasible solution obtained by the LD.

3.4. Column generation for LD

A column generation approach was employed by Mauri and Lorena (2012b) as an alternative way to find the multipliers and to solve the dual for a Lagrangian decomposition (LD) applied to solve a quadratic programming problem.

Adapting the method presented by those authors and considering that constraints (27) must be relaxed, as performed in the previous section for the LD, the model (9)–(14), (16), (26) and (27) can be reformulated by using the Dantzig-Wolfe decomposition considering constraints (27) to define the RMP.

Let S^l be a subset of all solutions for subproblem l. Each solution $s\in S^l$ is composed of original and copy variables $\hat{x}^s_{ijk}$ and $\hat{\bar{x}}^{ls}_{ijk}$, respectively. In addition, let $o\in S^h$ be a solution for a subproblem h ($h\neq l$). The linear relaxation for the RMP is given as follows:

$$\bar{z}^{RMP}_{CGLD} = \max \sum_{l=1}^{L}\sum_{k\in V_l}\sum_{i\in C}\sum_{j\in I_i}\sum_{s\in S^l} c_{ik}\hat{x}^s_{ijk}\mu_{sl} \quad (33)$$

Subject to:

$$\sum_{s\in S^l}\left[\mu_{sl}\hat{\bar{x}}^{ls}_{i(j-q)u} - \mu_{oh}\hat{x}^o_{i(j-q)u}\right] = 0 \quad \begin{array}{l} f=1..F, i\in B_f,\\ j=1..M, q=0..(t_i-1),\\ l=1..L, k\in V_l'' \mid u\in A_k,\\ u>k, u\in V_h, h\neq l \end{array} \quad (34)$$

$$\sum_{s\in S^l}\mu_{sl} = 1 \quad l=1..L \quad (35)$$

$$\mu_{sl}\geq 0 \quad \begin{array}{l} l=1..L,\\ s\in S^l \end{array} \quad (36)$$

Dual variables $\varepsilon^l_{i(j-q)u}$ ($f=1..F, i\in B_f, j=1..M, q=0..(t_i-1), l=1..L, k\in V_l'' \mid u\in A_k, u>k, u\notin V_l$) and φ_l ($l=1..L$) are associated with constraints (34) and (35) from the RMP, respectively. $\varepsilon^l_{i(j-q)u}$ are used to replace the Lagrangian multipliers $\lambda^l_{i(j-q)u}$ in (30) resulting in a value denoted as $\bar{z}^l_{CGLD}$, while φ_l are used to calculate the reduced cost for a column related to the subproblem l.

The reduced cost ς_l ($\varsigma_l = \bar{z}^l_{CGLD} - \varphi_l > 0$) indicates if the column related to the subproblem l ($l=1..L$) must be inserted into RMP, while $\bar{z}^{RMP}_{CGLD}$ will denote a valid upper bound for the CRSP at the end of column generation procedure.

The Lagrangian dual (37) can be calculated at each iteration of the procedure by replacing the Lagrangian multipliers (employed by the LD presented in the previous section) to the dual variables obtained from the RMP.

$$w_{CGLD} = \sum_{l=1}^{L}\bar{z}^l_{CGLD} \quad (37)$$

Finally, the Lagrangian dual and a feasible solution for the CRSP can be obtained in the same way as presented in Section 3.2, i.e. by using (37) and solving the RMP with integer variables μ_{sl} (replacing (36) by $\mu_{sl}\in\{0,1\}$), respectively. z_{CGLD} is used hereafter to denote the value of a feasible solution obtained from the RMP.

3.5. Lagrangian relaxation without clusters

As previously stated, constraints (6) can be considered as the complicating constraints for the CRSP. In its "traditional" way, the Lagrangian relaxation (LR) does not consider the partitioning of the problem into subproblems.

Then, constraints (6) can be relaxed with multipliers $\lambda_{fjk}\geq 0$ ($f=1..F$, $j=1..M$ and $k=1..K$) resulting in an "easier" problem as presented below.

$$\bar{z}_{LR} = \max \sum_{k=1}^{K}\sum_{i\in C}\sum_{j\in I_i} c_{ik}x_{ijk} + \sum_{f=1}^{F}\sum_{j=1}^{M}\sum_{k=1}^{K}\left\{\lambda_{fjk}\left(m_k - \sum_{i\in B_f}\sum_{q=0}^{t_i-1}\left[m_k x_{i(j-q)k} + \sum_{u\in A_k, u>k} x_{i(j-q)u}\right]\right)\right\} \quad (38)$$

Subject to: (2)–(5) and (7).

Upper bounds for the CRSP and values for λ_{fjk} can be obtained through the Lagrangian dual presented in (39), which can be solved by the subgradient algorithm described in Section 4.1.

$$w_{LR} = \min\left\{\bar{z}_{LR} : \lambda_{fjk}\geq 0, f=1..F, j=1..M, k=1..K\right\} \quad (39)$$

For notation, z_{LR} is used afterwards to denote the value of a feasible solution from the LR obtained by the Lagrangian heuristic in the subgradient algorithm.

3.6. Column generation presented by Santos et al. (2011)

The authors reformulated the problem by using the Dantzig-Wolfe decomposition and solved it by a column generation procedure. Constraints (6) are considered to define the RMP.

In a summarized way, the column generation proposed by these authors (denoted here as CGSA) is similar to that described in Section 3.2 when considering each plot k ($k=1..K$) into an independent subproblem l ($l=1..L$), i.e. $L=K$.

Then, for each subproblem l ($l=1..L$), $V_l'=\emptyset$ and $V_l''=V_l$, i.e. there will be no constraints (14) and all constraints (15) will be relaxed. For notation, $\bar{z}^{RMP}_{CGSA}$, $\bar{z}^l_{CGSA}$, w_{CGSA} and z_{CGSA} are used to denote the values of the relaxed RMP, subproblem l ($l=1..L$), Lagrangian dual, and feasible solution obtained from the RMP.

4. Algorithms to solve the relaxations

Defining the subproblems is the first step in applying the methods presented in the previous section (except the LR), i.e. how to partition the problem.

Thus, given a number L ($L \leq K$) the graph G must be partitioned in order to minimize the number of edges linking the subgraphs, i.e. reducing the number of relaxed constraints as stated before in Section 3. The graph partitioning library METIS (Karypis & Kumar, 1998) is used in this work due to its good results reported in previous researches, as can be seen in Mauri and Lorena (2012a,b) and Ribeiro and Lorena (2008).

The subproblems (subgraphs) are then sorted according to the number of adjacencies related to plots that are not in the same subproblem. The LR does not consider subproblems ($L = 1$) and then the plots are sorted according to their number of adjacencies. At this point, the first subproblem (or plot for the LR) is the one with more "external edges". As pointed out by Santos et al. (2011) this procedure is important for the heuristics employed in the algorithms presented below.

After defining the subproblems, the relaxation models presented in Section 3 can be obtained, and the following algorithms can be used to solve them.

4.1. Subgradient algorithm

The subgradient algorithm presented in Algorithm 1 was used to solve the relaxation models described in Sections 3.1, 3.3 and 3.5. It is based on a classical algorithm as described in Mauri and Lorena (2011).

Firstly, all Lagrangian multipliers are set to zero and lower and upper bounds and α are initialized. At each iteration t the multipliers λ^t are defined according to the relaxation method, i.e. λ_{fjk} for *LC* and *LR* and $\lambda^l_{i(j-q)u}$ for *LD*.

Inside the loop, the relaxed models defined by (19), (31) or (38) are solved by a commercial solver and a feasible solution is generated for *LC*, *LD* or *LR*, respectively, i.e. $\bar{z} = (\bar{z}_{LC}$ or $\bar{z}_{LD}$ or $\bar{z}_{LR})$ (line 8) and $z = (z_{LC}$ or z_{LD} or $z_{LR})$ (line 9). The bounds are updated and α is reduced by half when the upper bound does not improve after some iterations.

Algorithm 1: Subgradient algorithm.

```
1  t ← 0;
2  Set all multipliers λ^t to 0;
3  ub ← ∞; lb ← −∞; α ← 2;
4  repeat
5  |  for l ← 1 to L do
6  |  |  Solve the subproblem l (or problem for LR) finding z̄^l;
7  |  end
8  |  z̄ ← Σ_{l=1}^{L} z̄^l;
9  |  Apply a Lagrangian heuristic finding a feasible solution with value
   |  z;
10 |  ub ← min(ub, z̄);
11 |  lb ← max(lb, z);
12 |  α ← α/2 if ub has not improved in the last 15 iterations;
13 |  Calculate the subgradient vector π;
14 |  Calculate the step θ ← α(ub−lb)/||π||^2;
15 |  Update the Lagrangian multipliers
   |  λ^{t+1} ← { max(0, λ^t − θπ)   for LC and LR
   |            { λ^t − θπ           for LD          ;
16 |  Update the subproblems (or problem for LR) using the new
   |  multipliers λ^{t+1};
17 |  t ← t + 1;
18 until (ub = lb or α < 0.005 or ||π||^2 = 0 or time ≥ 1 hour);
19 w ← ub; z ← lb;
20 return w and z;
```

The subgradients are calculated according to the relaxed constraints in each method: (15) for *LC*, (27) for *LD* or (6) for *LR*. Then, the step is calculated and the multipliers and subproblems are updated. The stopping criteria (line 18) are based on those presented in Mauri and Lorena (2011).

Finally, the best bounds obtained are returned, i.e. the Lagrangian dual (20), (32) or (39) is assigned to $w = (w_{LC}$ or w_{LD} or $w_{LR})$ and the value of the best feasible solution is assigned to $z = (z_{LC}$ or z_{LD} or $z_{LR})$.

The Lagrangian heuristic employed at each iteration of the subgradient algorithm (line 9) is based on the *plot-to-plot* greedy heuristic proposed by Santos et al. (2011).

After solving the subproblems separately (lines 5 to 7 in Algorithm 1), we verify whether the solution obtained for the second subproblem satisfies the adjacency constraints related to the solution obtained for the the first subproblem. If so, the solutions for the first and second subproblems are feasible. Otherwise, the second subproblem must be solved again by the solver, after setting to zero the decision variables that violate the adjacency constraints related to the first subproblem.

A solution for the third subproblem must consider the solutions obtained for the first and second subproblems, and so on, i.e. a solution for each subproblem must be found by setting some variables to zero in order to guarantee that the adjacency constraints related to the previous subproblems are satisfied.

A feasible solution for the CRSP with value z ($z = z^1 + z^2 + \cdots + z^L$) is obtained if there is a solution for all subproblems. Otherwise, if a solution for some subproblem cannot be found, the heuristic must be terminated returning a value zero ($z = 0$), because a feasible solution for the CRSP cannot be obtained starting from the solution for the first subproblem.

As observed by Santos et al. (2011) this heuristic is able to provide good results quickly while each subproblem is composed of only one plot ($L = K$) and solved by a commercial solver. From the experiments performed in this work (Section 6.3) it is possible to note that the heuristic also provides good results when the subproblems are defined with more than one plot.

4.2. Column generation algorithm

The column generation approaches stated in Sections 3.2, 3.4 and 3.6 were solved through Algorithm 2 , which is based on those presented by Santos et al. (2011) and Mauri and Lorena (2012a).

A feasible solution for the CRSP considering only green manuring crops and fallow for all plots can be quickly obtained by commercial solvers. So, similar to that proposed by Santos et al. (2011), this solution can be used to generate the initial columns for the RMP (line 1).

The linear relaxation for the RMP is solved finding $\bar{z}^{RMP}_{CG} = (\bar{z}^{RMP}_{CGLC}$ or $\bar{z}^{RMP}_{CGLD}$ or $\bar{z}^{RMP}_{CGSA})$ for the methods described in Section 3.2, 3.4 or 3.6.

Dual variables from the RMP are used to update the subproblems which are solved separately. If the reduced cost for each subproblem l ($l = 1..L$) is positive a column related to it is inserted into the RMP.

The procedure described in lines 10–15 are proposed by Santos et al. (2011) who demonstrated that it improves the best feasible solution found by the algorithm. This procedure is based on the Lagrangian heuristic presented in the previous section.

An upper bound $w_{CG} = (w_{CGLC}$ or w_{CGLD} or $w_{CGSA})$ is calculated in line 18 where $\bar{z}^l_{CG} = (\bar{z}^l_{CGLC}$ or $\bar{z}^l_{CGLD}$ or $\bar{z}^l_{CGSA})$, for $l = 1..L$.

The stopping criteria are described in line 19. Santos et al. (2011) used only the first stopping criterion, i.e. when no column has a positive reduced cost (an optimal solution for the relaxed RMP is found). However, the second stop criterion was used

Algorithm 2: Column generation algorithm.

Generate an initial set of columns to define the RMP;
repeat
  Solve the RMP finding $\bar{z}_{CG}^{RMP}$;
  Update the subproblems using the dual variables ε from the RMP;
  for $l \leftarrow 1$ **to** L **do**
    Solve the subproblem l finding $\bar{z}_{CG}^{l}$;
    Calculate the reduced cost $\varsigma_l \leftarrow \bar{z}_{CG}^{l} - \varphi_l$;
    if $\varsigma_l > 0$ **then**
      Insert the column (solution) for subproblem l in the RMP;
      if $l \geq 2$ **then**
        Try to find a feasible solution for subproblem l without violating the adjacency constraints related to each previous subproblem o ($\forall\, o < l$);
        if *there is a feasible solution for* l **then**
          Insert the new column (new solution) for subproblem l in the RMP;
        end
      end
    end
  end
  Calculate $w_{CG} \leftarrow \bar{z}_{CG}^{1} + \bar{z}_{CG}^{2} + \cdots + \bar{z}_{CG}^{L}$;
until ($\varsigma_l \leq 0\, \forall\, l$ **or** $w_{CG} = \bar{z}_{CG}^{RMP}$ **or** *time* $\geq$ 1 hour);
Solve the RMP with integer variables finding a feasible solution for the CRSP with value z_{CG};
return z_{CG}, $\bar{z}_{CG}^{RMP}$ and w_{CG};

by Mauri and Lorena (2012a,b) in order to reduce the processing time of the algorithm. The last criterion is employed to provide a fair comparison with the subgradient algorithm presented in Section 4.1.

Finally, the RMP is solved with integer variables resulting in a feasible solution for the CRSP with value $z_{CG} = (z_{CGLC}$ or z_{CGLD} or $z_{CGSA})$, which is returned along with the best upper bounds.

5. Input data for the CRSP

Among the works found in the literature, Santos et al. (2011) and Aliano Filho et al. (2014) report in details the main data used in their experiments. Then, the data presented in these works were used as a basis to draw the instances proposed here. The data presented in both works were based on real-world values.

Santos et al. (2011) used different instances with 7 to 28 crops ($7 \leq N \leq 28$), 3–10 botanical families ($3 \leq F \leq 10$) and 1 to 4 green manuring crops ($1 \leq |D| \leq 4$). The authors presented areas with up to 10 ($K \leq 10$) plots for a rotation of 2 years considering periods of 10 days ($M = 72$; each year was rounded to 360 days) and fallow of 30 days ($t_n = 3$; three units of time).

Aliano Filho et al. (2014) considered 29 crops ($N = 29$), 11 botanical families ($F = 11$) and 4 green manuring crops ($|D| = 4$). Different instances were generated considering areas with 10–20 ($10 \leq K \leq 20$) plots for a rotation of 2 years with periods and fallow of 1 month ($M = 24$ and $t_n = 1$; one unit of time).

5.1. Defining the crops

Firstly, we put all the crops presented in the aforementioned works together into a single set. Duplicate crops were removed and the production time for each crop was set to the value presented by Santos et al. (2011) when different values were found. For example, the production time for Lettuce presented by Santos et al. (2011) and Aliano Filho et al. (2014) was 50 and 60 days, respectively. Probably the value presented by Aliano Filho et al. (2014) has been rounded up to fit in periods of 1 month (30 days). The profit values were estimated from the data presented by Aliano Filho et al. (2014) and other works reported in the literature. That done, there were 35 crops left (31 trade crops and 4 green manuring crops) among 11 botanical families.

The resulting crops were filtered again to remove "dominated" crops. For each botanical family f ($f = 1..F$), if there are two crops i and i' ($i, i' \in \{1..N\} \,|\, i, i' \in B_f$) such that $I_{i'} \subseteq I_i$, $t_{i'} \geq t_i$ and $p_{i'} \leq p_i$, i' can be considered as a dominated crop because it can always be replaced by crop i resulting in a better or equal profit. Finally, there were 24 trade crops and 3 green manuring crops (total of 27 crops without fallow) distributed into 11 botanical families, as shown in Table 1.

5.2. Cropping areas

Both works of Santos et al. (2011) and Aliano Filho et al. (2014) did not present the exact layout of the cropping areas, although they presented some features about them. Considering that, we propose different cropping areas with four different sizes (6, 9, 15 and 20), each with two layout: *checkerboard* and *shuffled*.

The checkerboard layout represents a desirable cropping area on real-world due to favouring crop handling, mainly through agricultural machinery. On the other hand, the shuffled layout dealt with some situations commonly found in Brazil where uneven relief and natural barriers prevent planting in an organized area as a checkerboard.

Fig. 3 depicts the generated layouts. The checkerboard layout was set in a trivial way while the shuffled layout was randomly defined. We can observe that the shuffled layout increases the adjacencies, i.e. the number of edges in the respective graphs G increases by 25.5% on average.

The shape of the plots in Fig. 3 is used to depict adjacency but is not representative of the actual area. The area of each plot was randomly defined in a range [1,20] of 0.25ha each unit, resulting in plots with 0.25 to 5.00ha and thus in cropping areas with distinct sizes as shown in Table 2.

5.3. Instances

Based on the data described above, a set of instances was generated by using the full set of 27 crops (plus fallow) for rotations of 1 and 2 years with periods of 1 month and 10 days ($M = 12$, 24, 36 and 72). So, there are 16 different combinations of M and K ($K = 6$, 9, 15 and 20) for each layout of cropping area, resulting in 32 different instances.

It is important to highlight that the number of crops was similar to those from the larger instances presented by Santos et al. (2011) and Aliano Filho et al. (2014). In addition, odd and even numbers of plots with different layouts were also considered, resulting in areas like the more complex and larger ones presented in these works.

When periods of 1 month (30 days) were considered the production time for all crops were rounded up ($t_i = \lceil \frac{t_i}{30} \rceil$, $i = 1..n$).

6. Computational experiments

All the computational experiments presented in this work were performed on a Workstation Dell Precision T5810 with an Intel Xeon® E5–1650 v3 (6C – 3.5GHz) processor with 64MB of RAM memory. The mathematical models were solved by CPLEX 12.6 64bits for Windows accessed by its callable library through C++ language. Default settings of CPLEX were used for all methods and algorithms, which were also coded in C++ language.

Except for LR and CGSA described in Sections 3.5 and 3.6, the plots were divided into subproblems by using the partitioning library Metis 5.1.0 64bits (Karypis & Kumar, 1998) which was also accessed through C++ language.

Table 1
Description of the crops: $i = 1..n,\ f = 1..F \mid i \in B_f,\ I_i,\ t_i$ and p_i.

Crop		Botanical family		Planting season Begin	End	Production time (days)	Profit (R$/ha)
1	Yams	1	Aracea	September	December	250	460
2	Lettuce	2	Compositae	All year		50	300
3	Watercress	3	Brassicaceae	All year		120	500
4	Cole	3	Brassicaceae	March	June	90	300
5	Cauliflower	3	Brassicaceae	March	June	140	900
6	Chinese cabbage	3	Brassicaceae	March	June	100	420
7	Beet	4	Chenopodiaceae	March	July	70	810
8	Spinach	4	Chenopodiaceae	February	July	50	600
9	Pumpkin	5	Cucurbitaceae	September	January	150	480
10	Watermelon	5	Cucurbitaceae	August	October	90	900
11	Cucumber	5	Cucurbitaceae	All year		140	450
12	Garlic	6	Aliacea	March	April	180	810
13	Onion	6	Aliacea	March	June	120	430
14	Oatmeal	7	Gramineae	March	May	180	350
15	Corn	7	Gramineae	August	April	90	350
16	Okra	8	Malvaceae	August	March	230	710
17	Potato	9	Solanaceae	August	October	90	240
18	Tomato	9	Solanaceae	All year		150	810
19	Carrot	10	Umbelliferae	March	July	140	620
20	Peruvian carrot	10	Umbelliferae	March	August	300	1000
21	Parsley	10	Umbelliferae	September	March	190	400
22	Pea	11	Leguminosae	March	April	90	830
23	Snap bean	11	Leguminosae	August	April	110	750
24	Bean	11	Leguminosae	August	September	90	720
25	Vetch*	11	Leguminosae	March	June	140	0
26	Jack bean*	11	Leguminosae	September	December	80	0
27	Black velvet bean*	11	Leguminosae	September	January	110	0
28	Fallow	–	–	All year		30	0

* green manuring crops

Table 2
Area (ha) of the plots.

Plot	Area	Plot	Area	Plot	Area	Plot	Area
1	1.50	6	2.75	11	0.25	16	0.25
2	2.00	7	5.00	12	0.75	17	2.75
3	2.25	8	1.50	13	1.00	18	3.25
4	0.75	9	4.00	14	0.50	19	5.00
5	0.50	10	4.00	15	3.50	20	3.00

6.1. Analysis of adjacency constraints

An initial analysis on the model (1)–(7) regarding to adjacency constraints was performed. Table 3 shows some features about the model using or not the condition $u > k$ as proposed in Section 2.3. The first column identifies the different combinations (instances) of layout for the cropping area, rotation period (M) and number of plots (K) presented in the following columns. The number of decision variables for each instance is presented in column x_{ijk}, followed by the number of other constraints.

The next columns report the number of adjacency constraints (6) and the total number of decision variables inside them considering the condition $u > k$ as proposed in this work (New) or not (Old), as presented by Santos et al. (2011) and Aliano Filho et al. (2014). As can be seen in Section 2.3 the number of adjacency constraints (6) and (8) and their variables are the same.

The number of decision variables and other constraints are not dependent on the layout of the cropping area and then the values in the first half of columns x_{ijk} and (2)–(5) in Table 3 are the same as in the second half. The same is also true for the number of adjacency constraints when condition $u > k$ is not considered (Old), but the number of decision variables inside them always depends on the layout of the cropping area.

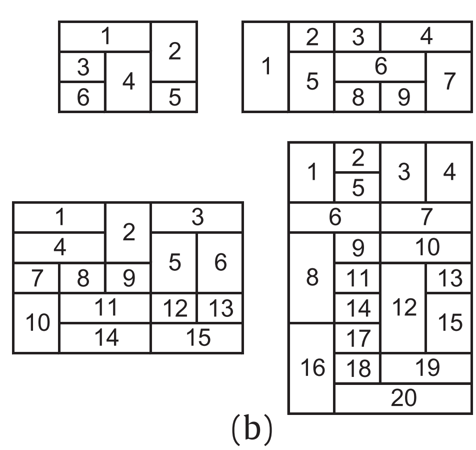


Fig. 3. Cropping areas with checkerboard (a) and shuffled (b) layouts.

Table 3
Dimensions of the model (1)–(7) considering old and new adjacency constraints.

Id	Layout	M	K	x_{ijk}	(2)–(5)	Old		New	
						(6)	Var.	(6)	Var.
1	CHECKERBOARD	12	6	1008	834	738	14,340	615	8604
2			9	1512	1251	1107	23,661	984	14,340
3			15	2520	2085	1845	42,303	1722	25,812
4			20	3360	2780	2460	58,794	2337	35,850
5		24	6	2016	1656	1476	28,680	1230	17,208
6			9	3024	2484	2214	47,322	1968	28,680
7			15	5040	4140	3690	84,606	3444	51,624
8			20	6720	5520	4920	117,588	4674	71,700
9		36	6	3024	2490	2226	122,520	1855	73,512
10			9	4536	3735	3339	202,158	2968	122,520
11			15	7560	6225	5565	361,434	5194	220,536
12			20	10,080	8300	7420	502,332	7049	306,300
13		72	6	6048	4968	4452	245,040	3710	147,024
14			9	9072	7452	6678	404,316	5936	245,040
15			15	15,120	12,420	11,130	722,868	10,388	441,072
16			20	20,160	16,560	14,840	1,004,664	14,098	612,600
17	SHUFFLED	12	6	1008	834	738	17208	492	9321
18			9	1512	1251	1107	27,963	984	16,491
19			15	2520	2085	1845	49,473	1722	29,397
20			20	3360	2780	2460	70,266	2337	41,586
21		24	6	2016	1656	1476	34,416	984	18,642
22			9	3024	2484	2214	55,926	1968	32,982
23			15	5040	4140	3690	98,946	3444	58,794
24			20	6720	5520	4920	140,532	4674	83,172
25		36	6	3024	2490	2226	147,024	1484	79,638
26			9	4536	3735	3339	238,914	2968	140,898
27			15	7560	6225	5565	422,694	5194	251,166
28			20	10,080	8300	7420	600,348	7049	355,308
29		72	6	6048	4968	4452	294,048	2968	159,276
30			9	9072	7452	6678	477828	5936	281796
31			15	15,120	12,420	11,130	845,388	10,388	502,332
32			20	20,160	16,560	14,840	1200,696	14,098	710,616
	Average			6300.00	5181.25	4631.25	272,009.25	4214.44	162,307.41

The number of adjacency constraints becomes dependent on the layout of the cropping area when using the condition $u > k$ as proposed in this work (New). Considering the areas presented in Section 5.2, a difference occurs only for the cropping area with 6 plots, for which plot 5 has one adjacent following plot (plot 6) in the checkerboard layout and none in the shuffled (see Fig. 3), i.e. $A_5 = \{2, 4, 6\}$ and $A_5 = \{2, 4\}$ for checkerboard and shuffled layouts, but when condition $u > k$ is applied plot 5 has no adjacent plot for shuffled layout, resulting in fewer adjacency constraints (see for example the instances 1 and 17, 5 and 21 and so on).

As can be observed from Table 3, both the number of adjacency constraints and decision variables inside them are reduced by 9% (416.81 constraints) and 40.3% (109,701.84 variables) on average when using the condition $u > k$ as proposed in this work.

CPLEX was applied to solve the CRSP through the model considering the old and new adjacency constraints (6) and (8). A time limit of 1 hour (3600s) was applied for each instance and model. Fig. 4 presents the average values for the best feasible solution (z_{CRSP}), best node/upper bound (w_{CRSP}) and linear relaxation ($\bar{z}_{CRSP}$) considering the first objective ($c_{ik} = t_i$).

We can observe in Fig. 4(a) that values for both w_{CRSP} and $\bar{z}_{CRSP}$ were optimal when using any adjacency constraints, as well as the value z_{CRSP} for both old and new constraints (6). However, all optimal solutions for the CRSP could not be found when using constraints (8). Considering the results for the shuffled layout depicted in Fig. 4(b), we can observe that CPLEX was able to find the optimal solutions only when using the new constraints (6). The upper bounds (w_{CRSP}) were the same again, but the values of the linear relaxation have worsened, especially when both old and new constraints (8) were used.

A comparison of the values found by CPLEX considering the second objective ($c_{ik} = r_k p_i$) described in Section 2.1 is presented in Fig. 5. We can note that the problem becomes more difficult to be solved by CPLEX. Even so, the best values were still obtained when using the new constraints (6), except for the linear relaxation applied to the shuffled layout for which the results of old constraints (6) were slightly better.

Table 4 reports a comparison of the results obtained by using the old and new constraints (6). The columns Opt indicate the number of optimal solutions proven by CPLEX over the total of 16 (16 instances for each layout). The average gaps (%) and computational times (seconds) are presented in columns Gap and Time, respectively. Finally, the columns $\overline{\text{Time}}$ indicate the computational time (seconds) to solve the linear relaxation for the CRSP.

As can be observed from Table 4, all optimal solutions were obtained by using new constraints (6) for the first objective while the optimal were not found for two instances when using the old constraints (6). For the second objective, the model using new constraints (6) was able to provide two more optimal solutions. In addition, we can observe average improvements of 27.2%, 22.6% and 38.6% in gap and computational times to solve model (1)–(7) and its linear relaxation, respectively.

A similar comparison considering the old and new constraints (8) are presented in Table 5. Average improvements of 8.1%, 21.2% and 46.4% can be observed in gap and computation times when using new constraints (8). For the first objective we can see that optimal solutions for two instances were not provided using both old and new constraints (8), while two more optimal solutions were provided when using new constraints (8) for the second objective.

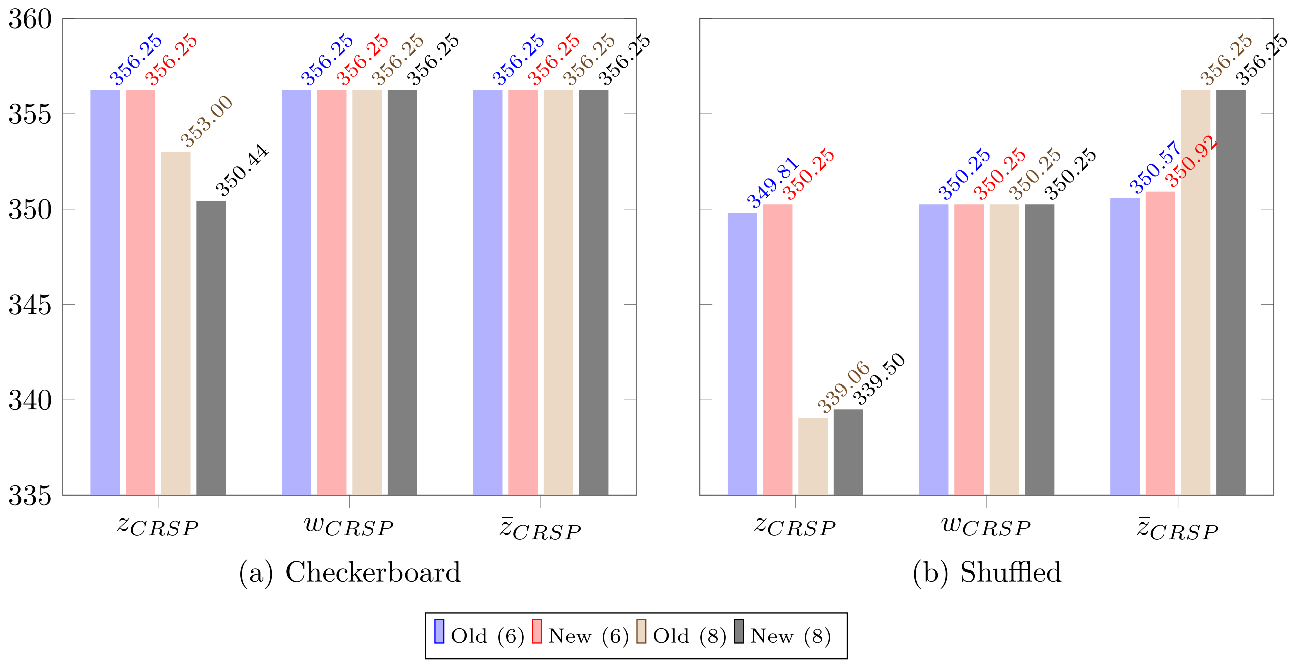


Fig. 4. Average values from CPLEX considering $c_{ik} = t_i$.

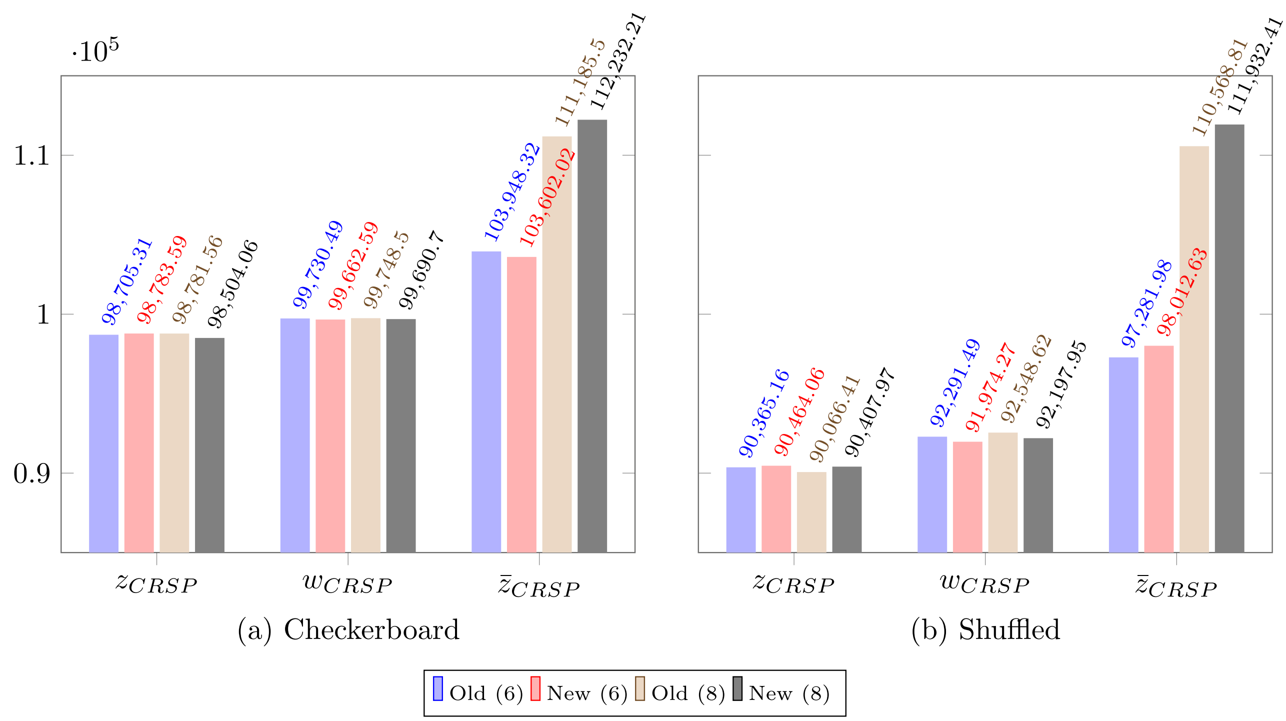


Fig. 5. Average values from CPLEX considering $c_{ik} = r_k p_i$.

Table 4
CPLEX results considering the adjacency constraints (6).

Objective	Layout	Old (6)				New (6)			
		Opt	Gap	Time	$\overline{\text{Time}}$	Opt	Gap	Time	$\overline{\text{Time}}$
$c_{ik} = t_i$	Chec.	16	0.00	40.81	3.14	16	0.00	43.56	0.67
	Shuf.	14	0.04	531.03	2.31	16	0.00	112.09	1.04
$c_{ik} = r_k p_i$	Chec.	12	0.54	959.20	18.19	12	0.45	932.56	13.15
	Shuf.	10	1.40	1486.96	21.87	12	0.99	1248.68	13.10
Average		–	0.50	754.50	11.38	–	0.36	584.22	6.99

Table 5
CPLEX results considering the adjacency constraints (8).

Objective	Layout	Old (8)				New (8)			
		Opt	Gap	Time	$\overline{\text{Time}}$	Opt	Gap	Time	$\overline{\text{Time}}$
$c_{ik} = t_i$	Chec.	15	0.28	257.05	0.28	15	0.52	252.20	0.19
	Shuf.	15	1.08	607.49	0.36	15	1.03	244.93	0.22
$c_{ik} = r_k p_i$	Chec.	12	0.52	1007.75	1.43	12	0.62	929.48	0.62
	Shuf.	10	1.77	1566.22	1.63	12	1.20	1281.76	0.96
Average		–	0.91	859.63	0.92	–	0.84	677.09	0.49

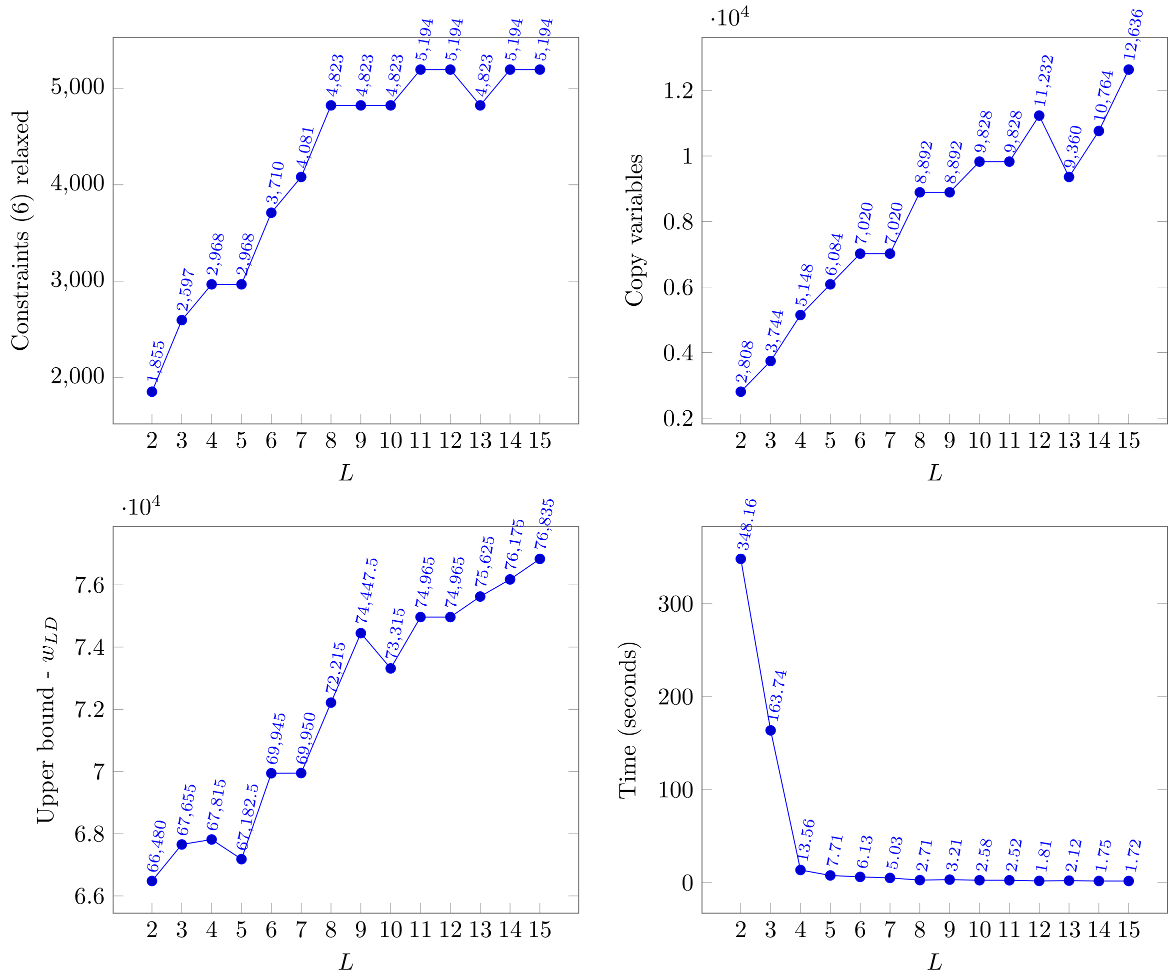


Fig. 6. Sensitivity analysis of L for instance 27 (shuffled, $K = 15$ and $M = 36$).

Comparing the results presented in Tables 4 and 5 we can observe that the computational times to solve the linear relaxation were significantly lower when using constraints (8) instead of (6). Nevertheless, the values obtained for the linear relaxation ($\bar{z}_{CRSP}$) were notably worse, as illustrated in Figs. 4 and 5. The values of gap and computational time were also worse when using constraints (8).

The results reported so far demonstrate that the new constraints (6) as proposed in this work are able to provide an improved mathematical model enabling the CPLEX to obtain better bounds and solutions for the CRSP.

6.2. Defining subproblems

Given an instance with K plots, the first decision to be made to apply the LC, CGLC, LD and CGLD concerns the definition of the number of subproblems L ($L \leq K$). As stated before, the library METIS (Karypis & Kumar, 1998) was used to partition the graph related to the cropping area minimizing the number of edges linking the subgraphs, i.e. reducing the number of relaxed constraints for the CRSP.

Fig. 6 illustrates a standard behavior of the LD by varying the value of L. In general, this behavior can also be extended to LC, CGLC and CGLD. Instance 27 (shuffled, $K = 15$ and $M = 36$) was chosen randomly as an example, and the values of upper bound and time were obtained by solving the subproblems only once. We can observe that larger values for L increase the number of relaxed constraints and copy variables providing weaker relaxations (worse upper bounds), but generating small subproblems that are faster to be solved by CPLEX. On the other hand, a lower L results in stronger relaxations, but with subproblems that are more difficult to be solved within a reasonable time.

It is interesting to note that there is no strictly linear relationship between L and the values obtained because they are strongly

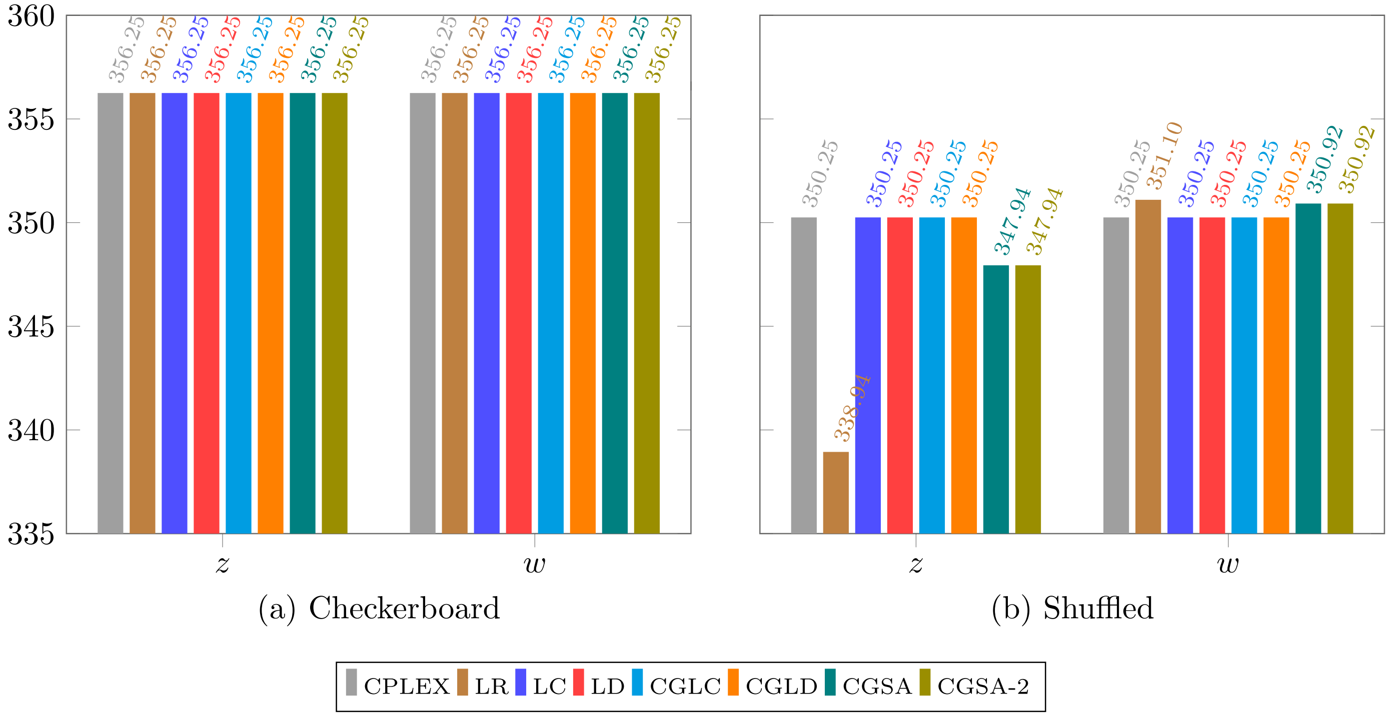


Fig. 7. Average results considering $c_{ik} = t_i$.

Table 6
Average computational times (seconds) considering $c_{ik} = t_i$.

Layout	CPLEX	LR	LC	LD	CGLC	CGLD	CGSA	CGSA-2
Checkerboard	43.56	3.09	4.41	4.90	5.40	6.00	5.39	10.69
Shuffled	112.09	657.80	7.36	8.25	11.18	13.49	10.72	17.95
Average	77.83	330.44	5.89	6.58	8.29	9.74	8.05	14.32

dependent on the cropping area arrangement and the way it is partitioned. Therefore, the most appropriate value of L should be chosen for each instance, but that would be a hard and impractical work.

Then, after some preliminary experiments it was possible to realize that a suitable value for L to handle with all instances considering both the first and second objectives can be be defined through the expression $\lfloor \frac{K}{L} \rfloor = 3$, i.e. trying to define each subproblem with approximately 3 plots. Thus, all the experiments presented hereafter consider $L = 2$, 3, 5 and 6 for $K = 6$, 9, 15 and 20, respectively.

6.3. Relaxation methods

All methods presented in Section 3 were applied to solve the 32 instances (see Table 3) for each objective ($c_{ik} = t_i$ and $c_{ik} = r_k p_i$), resulting in 64 experiments for each method.

Two different approaches for the column generation proposed by Santos et al. (2011) were performed. The first (CGSA) adopts exactly the same stopping criteria as those used by CGLC and CGLD (see line 19 in the Algorithm 2). Nevertheless, a second approach (CGSA-2) removing the criterion $w_{CG} = \bar{z}^{RMP}_{CG}$ (line 19 in Algorithm 2) was also evaluated aiming to reproduce exactly the same method as that proposed by Santos et al. (2011).

6.3.1. Maximizing plots occupation ($c_{ik} = t_i$)

A comparison of the average results obtained by all methods considering the first objective ($c_{ik} = t_i$) is illustrated in Fig. 7, where z and w indicate, respectively, the value of the best feasible solution and the upper bound obtained by each method and by CPLEX when solving the model (1)–(7).

We can observe in Fig. 7(a) that optimal solutions were found and proven by all methods for all instances. The same can also be observed in Fig. 7(b) for CPLEX, LC, LD, CGLC and CGLD. However, considering the shuffled layout, we can see that LR, CGSA and CGSA-2 were not able to close the gaps. The LR found only 7 (over 16) optimal solutions (average gap of 5%) while both CGSA and CGSA-2 found 11 (average gap of 0.7%). All the adjacency constraints are relaxed in LR, CGSA and CGSA-2 and consequently weaker relaxations are provided. In addition, it is interesting to highlight that the feasible solutions and upper bounds found by CGSA and CGSA-2 were the same for all instances. For all column generation based relaxations the value of the relaxed RMP was the same as the upper bound for all instances, i.e. $\bar{z}^{RMP}_{CGLC} = w_{CGLC}$, $\bar{z}^{RMP}_{CGLD} = w_{CGLD}$, $\bar{z}^{RMP}_{CGSA} = w_{CGSA}$ and $\bar{z}^{RMP}_{CGSA-2} = w_{CGSA-2}$.

Table 6 reports the average computational times. As can be observed, LC, LD, CGLC and CGLD required approximately 10% of the computational time spent by CPLEX, even though they also provide the optimal solutions and upper bounds for all instances.

The computational time of LC, LD, CGLC and CGLD are practically the same. Since all these methods converged completely (without reaching the time limit of 1 hour) for all instances, the minor differences between LC and LD (or CGLC and CGLD) can be attributed to the computational time to define the copy variables employed by the LD. On the other hand, the increase in computational time of CGLC and CGLD when compared to LC and LD can be related to the time required to create the RMP and to solve it in each iteration and at the end (with integer variables) of the column generation algorithm.

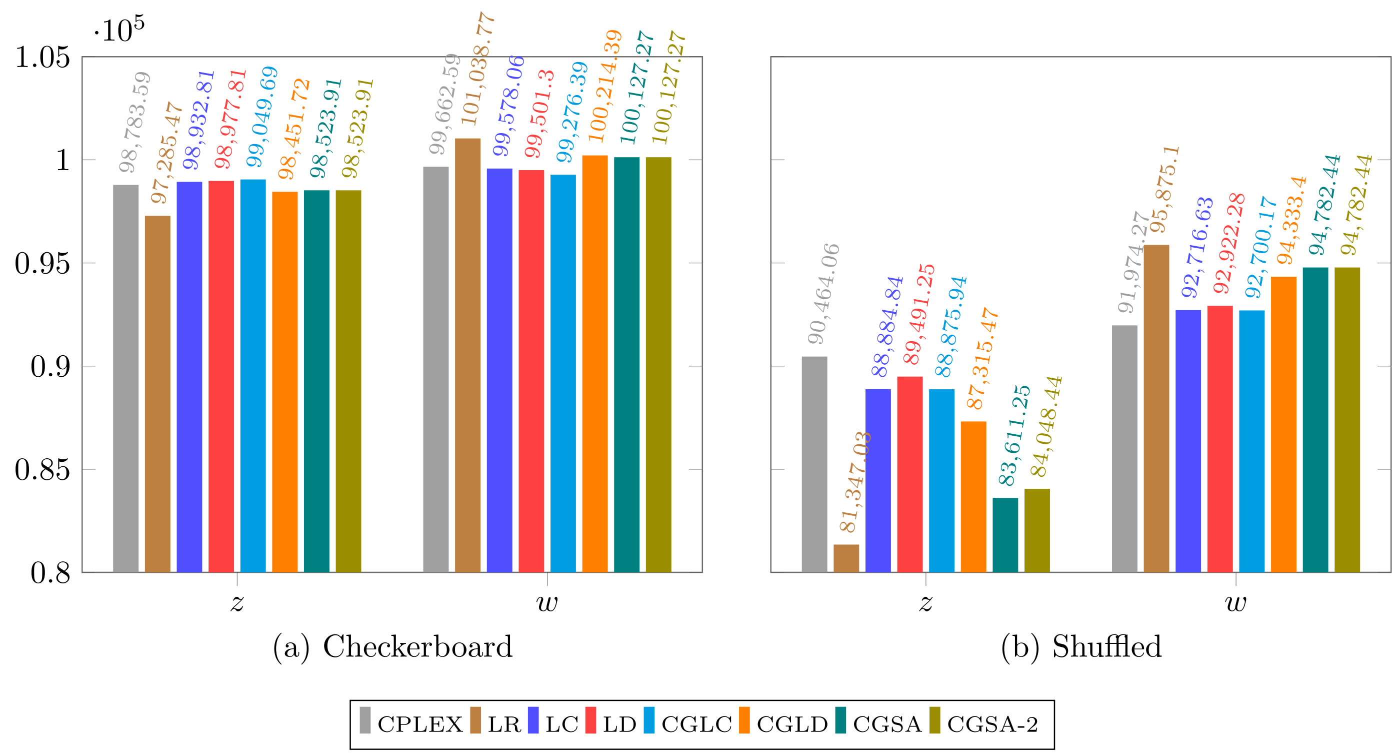


Fig. 8. Average results considering $c_{ik} = r_k p_i$.

Table 7
Average computational times (seconds) considering $c_{ik} = r_k p_i$.

Layout	CPLEX	LR	LC	LD	CGLC	CGLD	CGSA	CGSA-2
Checkerboard	932.56	2675.70	2940.03	2050.26	532.04	1898.91	152.11	2031.61
Shuffled	1248.68	2168.21	3001.87	2896.22	894.58	3171.62	665.07	3189.07
Average	1090.62	2421.95	2970.95	2473.24	713.31	2535.26	408.59	2610.34

The LR was slightly fastest for the checkerboard layout for which it converged completely, because it does not need to partition the problem generating and handling with subproblems and RMP. However, the LR was the slowest method for the shuffled layout since the gaps were not closed considering the stopping criteria described in Algorithm 1.

The CGSA-2 required almost twice the computational time spent by the CGSA to find the same solutions and upper bounds. Thus, it can be claimed that the stopping criterion $w_{CG} = \bar{z}_{CG}^{RMP}$ was suitable for reducing the computational time without compromising the quality of the solutions and bounds.

6.3.2. *Profit maximization* ($c_{ik} = r_k p_i$)

A comparison of the average results considering the second objective ($c_{ik} = r_k p_i$) is presented in Fig. 8. For both checkerboard and shuffled layouts we can observe that the optimal solutions for all instances were not proven by any method. Even so, the results of all methods were close for the checkerboard layout, but CGLC, LD and LC presented slightly better values. On the other hand, for the shuffled layout the best results were obtained by CPLEX, followed by LD, LC and CGLC.

Overall, two main points justify these results: the strength of the relaxations and the performance of the Lagrangian heuristic. The average computational times and gaps are reported in Tables 7 and 8, respectively, followed by a joint analysis with Fig. 8.

The column generation algorithm was able to improve the computational time by 76% in relation to the subgradient algorithm used in LC, i.e. CGLC in relation to LC. In addition, the gap was considerably reduced for the checkerboard layout. The same did not happen for the LD in relation to CGLD due to the heuristic employed to find feasible columns did not handle efficiently with the copy variables inside the column generation algorithm. Despite this, we can observe that LD outperforms LC because the use of copy variables provides a stronger relaxation while maintaining all adjacency constraints inside the subproblems.

Since all the adjacency constraints are relaxed in LR, CGSA and CGSA-2, weaker relaxations are provided and consequently worse solutions and bounds were obtained. In addition, the CGSA-2 increases the average time of the CGSA by more than 500% despite the same results had been obtained for the checkerboard layout and a minor improvement in the feasible solutions (z) had been provided for the shuffled layout (only 3 instances). Again, the stopping criterion $w_{CG} = \bar{z}_{CG}^{RMP}$ was proven.

The Lagrangian heuristic used in the subgradient algorithm for LR, LC and LD and the similar procedure employed to find feasible columns for CGLC, CGLD, CGSA and CGSA-2 were based on those proposed by Santos et al. (2011) for handling the CRSP considering $c_{ik} = t_i$. However, the objective $c_{ik} = r_k p_i$ considers the area of the plots (i.e. different plots) and estimated prices for each crop, and consequently makes the problem more difficult to be solved by CPLEX increasing the computational time.

Thus, both computation times required to solve the subproblems and quality of their solutions are compromised. Therefore, both column generation and subgradient algorithms were not able to converge completely for all instances within the time limit of 1 hour for none of the relaxation methods.

Table 9 reports the number of best solutions (12 optimal proven by CPLEX for each layout - see Table 4) obtained by each method.

Table 8
Average gaps (%) considering $c_{ik} = r_k p_i$.

Layout	CPLEX	LR	LC	LD	CGLC	CGLD	CGSA	CGSA-2
Checkerboard	0.45	2.53	0.46	0.34	0.16	1.30	0.95	0.95
Shuffled	0.99	20.87	2.99	2.56	2.90	6.47	13.83	13.57
Average	0.72	11.70	1.73	1.45	1.53	3.89	7.39	7.26

Table 9
Number of best known solutions obtained considering $c_{ik} = r_k p_i$.

Layout	CPLEX	LR	LC	LD	CGLC	CGLD	CGSA	CGSA-2
Checkerboard	14	9	11	11	14	10	10	10
Shuffled	15	0	6	8	7	6	0	0

Table 10
Best feasible solutions for instances 12, 14, 15 and 16.

Id	CPLEX	LR	LC	LD	CGLC	CGLD	CGSA/CGSA-2
12	103900.00	103900.00	103900.00	103900.00	103900.00	103900.00*	103900.00
14	115130.00	111350.00	114812.50	114812.50	115130.00	112820.00	115130.00
15	168482.50	162285.00	168900.00	169542.50	168900.00	168900.00	168900.00
16	243552.50	233187.50	246605.00	246605.00	247865.00*	244812.50	241185.00

* optimal values proven

As discussed earlier, CPLEX presented the best results for the instances with shuffled layout. The same is also true for the number of best solutions obtained (see Table 9). For the instances with checkerboard layout the best average values (see Fig. 8) were presented by CGLC, LD and LC, although the CPLEX has obtained more best solutions than LC and LD.

A complementary analysis is given below regarding the instances for which CPLEX cannot prove optimality. These instances are a total of 8 (over 32) with $M = 36$ for $K = 20$ and $M = 72$ for $K = 9$, 15 and 20, for both layouts, i.e. instances 12, 14, 15, 16, 28, 30, 31 and 32. Table 10 presents values of the best feasible solutions obtained for the instances with checkerboard layout. We can observe that two solutions were proven to be optimal by the relaxation methods. The CGSA and CGSA-2 obtained the same solutions for these instances.

Finally, for the instances with shuffled layout (28, 30, 31 and 32), no optimal solution has been proven by either CPLEX or any relaxation method.

7. Conclusions

We presented an improved mathematical model for the CRSP considering cropping areas divided into plots. For some instances, the new model enabled CPLEX to find optimal solutions that were not obtained through the model previously reported in the literature. Substantial improvements were also provided in the average gaps, bounds and computational times.

Five different relaxation approaches were also proposed and compared to a method presented in the literature and to the direct solving of the complete model (1)–(7). When considering the first objective (maximize the plots occupation), some relaxation methods were capable of finding all optimal solutions requiring approximately 10% of the computational time spent when solving the complete model, while all optimal solutions were not obtained by the method presented in the literature.

For the second objective (maximize profit), some relaxation methods outperformed the method presented in the literature, and the direct solving was outperformed again by some relaxation methods when considering the checkerboard layout for the cropping areas. For the shuffled layout, the best average results were obtained when solving the complete model, but the optimal solutions for some instances were proven only by some relaxation methods.

Finally, a detailed set of instances based on real-world values was also proposed considering the number of crops and plots similar to the more complex and larger instances reported in the literature. The details provided in this work are enough so that the instances can be used in further researches.

Acknowledgments

This research was partly supported by the CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico (processes 454569/2014-9 and 301725/2016-0) and the FAPES - Fundação de Amparo à Pesquisa e Inovação do Espírito Santo (process 73290475/2016). Thanks are due to the referees for their valuable suggestions.

References

Alfandari, L., Plateau, A., & Schepler, X. (2015). A branch-and-price-and-cut approach for sustainable crop rotation planning. *European Journal of Operational Research, 241*, 872–879.
Aliano Filho, A., Florentino, H. O., & Pato, M. V. (2014). Metaheuristics for a crop rotation problem. *International Journal of Metaheuristics, 3*(3), 199–222.
Brankatschk, G., & Finkbeiner, M. (2015). Modeling crop rotation in agricultural LCAs. *Agricultural Systems, 138*, 66–76.
Clarke, H. R. (1989). Combinatorial aspects of cropping pattern selection in agriculture. *European Journal of Operational Research, 40*, 70–77.
Filippi, C., Mansini, R., & Stevanato, E. (2017). Mixed integer linear programming models for optimal crop selection. *Computers & Operations Research, 81*, 26–39.
Guignard, M., & Kim, S. (1987). Lagrangean decomposition: a model yielding stronger Lagrangean bounds. *Mathematical Programming, 39*(2), 215–228.
Jaehn, F. (2016). Sustainable operations. *European Journal of Operational Research, 253*, 243–264.
Karypis, G., & Kumar, V. A. (1998). Multilevel *k*-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing, 48*, 96–129.
Mauri, G. R., & Lorena, L. A. N. (2011). Lagrangean decompositions for the unconstrained binary quadratic programming problem. *International Transactions in Operational Research, 18*(2), 257–270.
Mauri, G. R., & Lorena, L. A. N. (2012a). A column generation approach for the unconstrained binary quadratic programming problem. *European Journal of Operational Research, 217*(1), 69–74.
Mauri, G. R., & Lorena, L. A. N. (2012b). Improving a lagrangian decomposition for the unconstrained binary quadratic programming problem. *Computers & Operations Research, 39*(7), 1577–1581.

Mauri, G. R., Ribeiro, G. M., & Lorena, L. A. N. (2010). A new mathematical model and a Lagrangean decomposition for the point-feature cartographic label placement problem. *Computers & Operations Research, 37*, 2164–2172.

Ribeiro, G. M., & Lorena, L. A. N. (2008). Lagrangean relaxation with clusters for point-feature cartographic label placement problems. *Computers & Operations Research, 35*, 2129–2140.

Santos, L. M. R., Costa, A. M., Arenales, M. N., & Santos, R. H. S. (2010). Sustainable vegetable crop supply problem. *European Journal of Operational Research, 204*(3), 639–647.

Santos, L. M. R., Michelon, P., Arenales, M. N., & Santos, R. H. S. (2011). Crop rotation scheduling with adjacency constraints. *Annals of Operations Research, 190*(1), 165–180.

Santos, L. M. R., Munari, P., Costa, A. M., & Santos, R. H. S. (2015). A branch-price-and-cut method for the vegetable crop rotation scheduling problem with minimal plot sizes. *European Journal of Operational Research, 245*(2), 581–590.