Space minimization in agricultural production planning by column generation

J. Sadki [1], L. Alfandari [2], A. Plateau [3], A. Nagih [4]

1. LIPN, UMR-CNRS 7030, Université Paris 13,
99 Av. J-B Clément 93430 Villetaneuse
jalila.sadki@lipn.univ-paris13.fr
2. ESSEC Business School,
Av. Bernard Hirsch B.P. 50105 95021 Cergy-Pontoise Cedex
alfandari@essec.fr
3. CEDRIC, CNAM,
292 Rue Saint-Martin 75141 Paris Cedex 03
aplateau@cnam.fr
4. LITA, Université Paul Verlaine-Metz,
Ile du Saulcy 57045 Metz Cedex 1
anass.nagih@univ-metz.fr

Abstract

We deal in this paper with an agricultural production planning problem where crops must be scheduled on land plots so as to satisfy crop demands every period of time and to minimize the overall surface of land used for cultivation. This problem can be formulated as a covering integer program with a huge number of variables. A resolution scheme based on column generation is thus proposed, where the resulting pricing problem is efficiently solved by dynamic programming. The numerical experiments show that the method is all the more so efficient and robust as the planning horizon is long and plot sizes are small.

Key-words: Crop rotations, agricultural production planning, column generation, covering integer programming, dynamic programming.

1 Introduction

This paper deals with a class of production problems in agriculture, that of planning crops over a given time horizon and given land plots so that the total production of each crop every season meets the needs of the farmer. The difficulty of these problems lies in the decreasing yields of cultivated lands over time, which enforce to plan fallow periods over the time horizon and to alternate crops in the best possible way, which is not a trivial issue. Whereas typical crop-rotation problems focus on building rotations that maximize a profit or yield function as in [6, 4, 9, 3], the objective of the specific problem studied in this paper is to minimize space needed to meet the demand. This problem was originally presented in [1, 2] in a Madagascan context where the minimization of cultivated space helped to contribute in the long-term to the sustainable development of the primary forest around the farms. The model and software developed in [2] were part of a project of struggle against deforestation sponsored by the French-Madagascan Cooperation. Whereas the previous paper mainly focused on the formulation of the application and agricultural conclusions, this paper focuses on dedicated solving methods for a generic variant, where a single crop can be cultivated on a land plot every season when this plot is not left fallow. The proposed solving method is a column-generation method based on a Dantzig-Wolfe decomposition of the problem. A column generation approach was also applied in [3], where the objective is to maximize the total time during which the land is cultivated (space is fixed contrary to our problem). Let us remark that the space-minimization problem studied in this paper has the particularity that the objective value is a multiple of the number of selected columns in the master program, which is also the number of plots used for production. Another particularity of this paper is that the planning is studied over a much longer horizon, up to 30 years, because the aim, inherited from the Madagascan project, was not only to satisfy the farmer constraints but also to measure the long term impact on the primary forest. This critical parameter in the computational tests motivated the use of a column generation approach which is known to be a powerful tool for solving large size covering problems.

The paper is organized as follows. Section 2 introduces the problem statement and data as well as the covering integer program formulation of the generic crop rotation problem. Section 3 describes the pricing problem and the dynamic programming method dedicated to its resolution. Section 4 presents computational experiments for various lengths of the time horizon and various sizes of the plots, comparing the column generation approach to *direct solving* using CPLEX solver. The numerical results show that the column generation algorithm is all the more so efficient as the time horizon is long. Section 5 concludes the paper.

2 The covering master problem for column generation

The Minimum-Space Crop Rotation Planning (MSCRP) problem is that of constructing crop rotations minimizing required space area for covering seasonal crop demands. We consider in this paper the case where all plots have the same unitary surface area and yield characteristics, which happens when the planting is done in the same area, so the objective amounts to minimizing the number of plots used. MSCRP can be formulated by an arc-flow model, as done by Alfandari et al. in [2] for a variant of the problem presented in this paper. However, a natural alternative attempt to model it consists in considering the set of feasible rotations over the scheduling time horizon, and then selecting a minimum number of rotations that meets the seasonal crops demand. Note that there is an equivalence between the arc-flow model and the model presented in this paper using the classical Dantzig-Wolfe decomposition. Let us define the notations that will be used throughout the paper. We consider:

- $\mathcal{T} = \{1, .., T\}$ the set of time periods of the planning horizon
- $\mathcal{R}$ the set of feasible rotations of a plot over the planning horizon
- $\mathcal{C}$ the set of crops that can be cultivated over the time horizon
- $b_{jt}, j \in \mathcal{C}, t \in \mathcal{T}$ the demand (in tons) of crop j on season t
- $\rho^r_{jt}, j \in \mathcal{C}, t \in \mathcal{T}, r \in \mathcal{R}$ the unitary yield of crop j (in tons) provided by rotation r on season t, which can be 0 if crop j is not cultivated on that period t in rotation r.
- S the surface area of a plot

and let θ^r be a decision variable representing the number of plots on which rotation $r \in \mathcal{R}$ is planned. The Minimum-Space Crop Rotation Planning (MSCRP) problem can be formulated as a Covering Integer Program where the objective function minimizes the total surface area used, while the covering constraints ensure the crop demand satisfaction. It is given as follows for a given surface area S:

$$(MSCRP)_S \begin{cases} z(S) = \min & \sum\limits_{r \in \mathcal{R}} S\theta^r \\ \text{s.t} & \\ & \sum\limits_{r \in \mathcal{R}} S\rho^r_{jt}\theta^r \geq b_{jt} \quad \forall j \in \mathcal{C}, t \in \mathcal{T} \\ & \theta^r \in \mathbb{N} \qquad\qquad\quad \forall r \in \mathcal{R}. \end{cases} \tag{1}$$

As problem $(MSCRP)_S$ is untractable in this form because of the huge number of variables representing all feasible rotations, we apply the column generation approach suggested by Gilmore and Gomory [5]. It consists of generating new rotations (i.e. columns of the constraint matrix) only when needed instead of enumerating all feasible ones, in order to solve the continuous relaxation of the problem.

We denote the linear relaxation of $(MSCRP)_S$ as the Master Problem $(MP)_S$ and we note $\overline{z}(S)$ its optimal value:

$$(MP)_S \begin{cases} \overline{z}(S) = \min & \sum\limits_{r \in \mathcal{R}} S\theta^r \\ \text{s.t} & \\ & \sum\limits_{r \in \mathcal{R}} S\rho^r_{jt}\theta^r \geq b_{jt} \quad \forall j \in \mathcal{C}, t \in \mathcal{T} \\ & \theta^r \geq 0 \qquad\qquad\quad \forall r \in \mathcal{R} \end{cases} \tag{2}$$

The well-known iterative principle of a column generation approach can be summarized as follows. We start with a limited number of crop rotations and solve a master problem reduced to this subset of columns. Then, given the dual variables obtained by this LP solving, we check whether there is any new feasible crop rotation of negative reduced cost that could be added to the restricted master problem in order to improve the LP bound. If no such negative reduced cost column is found, then the current solution is optimal for $(MP)_S$, otherwise we add a subset of negative reduced-cost columns to the Master Problem and reiterate the process.
We keep the constant factor S in the objective because, as shown in the following proposition, this formulation will allow to solve the same pricing problem whatever the value of S.

Proposition 1. *The continuous relaxation value of the master problem $(MP)_S$ is a constant independent of S.*

Proof. Let μ_{jt} be the dual variables associated with the covering constraints, and let $(D-MP)_S$ denote the dual problem associated with $(MP)_S$.

$$(D-MP)_S \begin{cases} \overline{z}_D(S) = \max & \sum_{j\in\mathcal{C}}\sum_{t\in\mathcal{T}} b_{jt}\mu_{jt} \\ \text{s.t} & \\ & \sum_{j\in\mathcal{C}}\sum_{t\in\mathcal{T}} S\rho^r_{jt}\mu_{jt} \leq S \quad \forall r \in \mathcal{R} \\ & \mu_{jt} \geq 0 \qquad\qquad \forall j \in \mathcal{C}, t \in \mathcal{T} \end{cases} \tag{3}$$

Dividing by S in the $(D-MP)$ constraints, we obtain a dual problem that does not depend on S. As $\overline{z}_D(S) = \overline{z}(S)$, we get the claimed result. □

3 Pricing problem formulation

We can either generate new columns in the column generation process or verify optimality of a current solution by solving a pricing problem producing the minimum reduced cost rotation among all rotations of the original set $\mathcal{R}$. Because of identical yields and surface areas of plots, a single pricing problem is solved at every iteration of the column generation process. We first present its formulation and then its efficient solving by dynamic programming.

Let μ_{jt} be the dual variables associated with the covering constraints of the master problem $(MP)_S$, and let $\overline{c}^r$ be the reduced cost of rotation $r \in \mathcal{R}$. The pricing problem of minimizing the reduced cost of variables θ^r is given as follows:

$$\min_{r\in\mathcal{R}} \overline{c}^r \quad = S(1 - \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{C}} \rho^r_{jt}\mu_{jt}) \tag{4}$$

Let us now model precisely a valid crop rotation of a plot. A rotation is defined as a sequence of cultivation and fallow periods over the time horizon. Let $P(j)$ be the set of crops that can precede crop $j \in \mathcal{C}$. We denote by (l,a) the state of a cultivated plot at period t, meaning that l fallow periods preceded a cultivation periods until current cultivated period t, where $1 \leq l \leq \bar{l}$ and $1 \leq a \leq \bar{a}$ (l will be called the fallow length and a will be called the cultivation age of the plot). $\bar{l}$ is the number of periods beyond which no yield improvement is achievable when going back to cultivation, and $\bar{a}$ is the number of cultivation periods beyond which the plot necessarily returns fallow because yield has decreased in too large amounts. If the plot has been left fallow at current period for l periods, l is also called the state of the plot. Taking into account the state of the plot enables to better model yields through plot history, as explained in [2]. Indeed, we denote by w^{la}_{ij} the yield of a plot in state (l,a) cultivated with crop j in current period preceded by crop $i \in P(j)$ (decreasing function of a and increasing function of l), and by w^{l1}_{fj} the yield of a plot that returns cultivated with crop j after l fallow periods (the yield is equal to zero when a plot is left fallow).
In order to represent sequences that will form a rotation, we define decision variables associated with a given period $t \in \{2,..T\}$, state (l,a), $1 \leq l \leq \bar{l}, 1 \leq a \leq \bar{a}$, crops $j \in \mathcal{C}$ and

$i \in P(j)$:

$$x^{la}_{ijt} = \begin{cases} 1 & \text{if plot is in state } (l,a)\text{, cultivated with crops } j \text{ and } i \text{ in periods } t \text{ and } t-1 \text{ respectively} \\ 0 & \text{otherwise} \end{cases}$$

we distinguish then the cases when a fallow of length l ($1 \leq l \leq \bar{l}$) precedes or follows a crop $j \in \mathcal{C}$:

$$x^{l}_{fjt} \quad = \quad \begin{cases} 1 & \text{if plot returns to cultivation with crop } j \in \mathcal{C} \text{ in period } t \in \mathcal{T} \text{ after } l \text{ fallow periods} \\ 0 & \text{otherwise} \end{cases}$$

$$x^{la}_{jft} \quad = \quad \begin{cases} 1 & \text{if plot is left fallow in period } t \ (t \geq 2) \text{ and was cultivated with crop } j \in \mathcal{C} \text{ in state} \\ & (l,a) \text{ in period } t-1 \\ 0 & \text{otherwise} \end{cases}$$

and finally the case of a fallow-fallow sequence without exceeding the maximal length $\bar{l}$ for $1 \leq l \leq \bar{l} - 1$:

$$x^{l}_{fft} \quad = \quad \begin{cases} 1 & \text{if, in period } t\text{, plot has been left fallow for } l+1 \text{ periods} \\ 0 & \text{otherwise} \end{cases}$$

and the particular case where the maximal fallow length is exceeded on a plot:

$$F^{\bar{l}}_{t} \quad = \quad \begin{cases} 1 & \text{if plot has been left fallow at least } \bar{l}+1 \text{ periods at period } t \\ 0 & \text{otherwise} \end{cases}$$

These last two series of variables differ from [2] because going back to cultivation after a long fallow period is not mandatory in this paper, and lead to the following flow conservation constraints to define feasible crop rotations.

$$1 = \sum_{j \in C} x^{\bar{l}}_{fj1} + F^{\bar{l}}_{1} \tag{5}$$

$$\sum_{i \in P(j)} x^{la}_{ijt} = \sum_{k: j \in P(k)} x^{l,a+1}_{jk,t+1} + x^{la}_{jf,t+1} \quad \forall t \in \{2,\ldots,T-1\}, j \in C, a \in \{2,\ldots,\bar{a}-1\}, l \in \{1,\ldots,\bar{l}\} \tag{6}$$

$$\sum_{i \in P(j)} x^{l\bar{a}}_{ijt} = x^{l\bar{a}}_{jf,t+1} \quad \forall t \in \{2,\ldots,T-1\}, j \in C, l \in \{1,..,\bar{l}\} \tag{7}$$

$$x^{l}_{fjt} = \sum_{k \in C: j \in P(k)} x^{l2}_{jk,t+1} + x^{l1}_{jf,t+1} \quad \forall t \in \{1,\ldots,T-1\}, l \in \{1,\ldots,\bar{l}\}, j \in C \tag{8}$$

$$\sum_{l \in \{1,\ldots,\bar{l}\}} \sum_{a \in \{1,\ldots,\bar{a}\}} \sum_{j \in C} x^{la}_{jft} = x^{1}_{ff,t+1} + \sum_{i \in C} x^{1}_{fi,t+1} \quad \forall t \in \{2,\ldots,T-1\} \tag{9}$$

$$x^{l-1}_{fft} = \sum_{j \in C} x^{l}_{fj,t+1} + x^{l}_{ff,t+1} \quad \forall t \in \{1,\ldots,T-1\}, l \in \{2,\ldots,\bar{l}-1\} \tag{10}$$

$$F^{\bar{l}}_{t} + x^{\bar{l}-1}_{fft} = \sum_{j \in C} x^{\bar{l}}_{fj,t+1} + F^{\bar{l}}_{t+1} \quad \forall t \in \{1,\ldots,T-1\} \tag{11}$$

Assuming that the plot is initially in a fallow state of length $\bar{l}$, constraint (5) establishes possible successions during the first period of the planning horizon. Constraints (6) are the flow conservation constraints when two crops follow one another without exceeding maximal cultivation age $\bar{a}$. When the maximal cultivation age is reached, then a fallow period is required to reset the plot; this is given by constraints (7). Constraints (8-9) are the flow conservation constraints when a fallow state precedes or follows a crop ensuring the possibility to move from a cultivation to a fallow period and vice versa. Finally, constraints (10-11) define the flow conservation when there is a succession of fallows resulting either from a fallow of maximal age or a fallow of lower age.

4 Solving the pricing problem by dynamic programming

As established before, the objective of the pricing problem is to determine a rotation with minimum reduced cost according to the dual values provided by LP solving of the current Master Problem. The yield ρ_{jt}^r of a given crop $j \in \mathcal{C}$ at period $t \in \mathcal{T}$ in rotation $r \in \mathcal{R}$ can be written as $\rho_{jt}^r = \sum_{l=1}^{\bar{l}} (\sum_{i \in P(j)} \sum_{a=2}^{\bar{a}} w_{ij}^{la} x_{ijt}^{la} + w_{fj}^{l1} x_{fjt}^{l})$, where variables $x_{ijt}^{la}, x_{fjt}^{l}$ satisfy constraints (5-11). Thereby, the objective of the pricing problem (4) aiming at minimizing the reduced cost of rotations is given as follows:

$$min \quad S - \sum_{t \in \mathcal{T}} \sum_{j \in C} \sum_{l=1}^{\bar{l}} S \mu_{jt} (w_{fj}^{l1} x_{fjt}^{l} + \sum_{i \in P(j)} \sum_{a=2}^{\bar{a}} w_{ij}^{la} x_{ijt}^{la})$$

which is equivalent to

$$min \quad 1 - \sum_{t \in \mathcal{T}} \sum_{j \in C} \sum_{l=1}^{\bar{l}} \mu_{jt} (w_{fj}^{l1} x_{fjt}^{l} + \sum_{i \in P(j)} \sum_{a=2}^{\bar{a}} w_{ij}^{la} x_{ijt}^{la}) \tag{12}$$

The following proposition provides a dynamic programming method for solving the pricing subproblem of finding the minimum reduced cost column at each iteration of the column generation scheme.

Proposition 2. *Let π_{jlat} and π_{flt} denote the values computed by the following recurrence formula for $t = 1, .., T$:*

$$\pi_{jlat} = \begin{cases} \pi_{fl,t-1} - \mu_{jt} w_{fj}^{l1} & \text{if} \quad a = 1 \\ \min_{i \in P(j)} \{\pi_{i,l,a-1,t-1} - \mu_{jt} w_{ij}^{la}\} & \text{if} \quad 2 \leq a \leq \bar{a} \end{cases}$$

$$\pi_{flt} = \begin{cases} \min_{j \in \mathcal{C}, l' \leq \bar{l}, a \leq \bar{a},} \{\pi_{jl'a,t-1}\} & \text{if} \quad l = 1 \\ \pi_{f,l-1,t-1} & \text{if} \quad 2 \leq l < \bar{l} \\ \min(\pi_{f,\bar{l}-1,t-1}, \pi_{f,\bar{l},t-1}) & \text{if} \quad l = \bar{l} \end{cases}$$

with initial conditions:

$$\pi_{f\bar{l}0} = 1, \quad \pi_{fl0} = \pi_{jla0} = \infty \quad \forall l \in \{1, .., \bar{l} - 1\}, a \in \{1, .., \bar{a}\}, j \in \mathcal{C}$$

Then,

$$\pi_T^* = \min(\min_{j,l,a} \pi_{jlaT}, \min_{l} \pi_{flT})$$

will be the value of a minimum reduced cost column. The corresponding column can be found by backtracking from $t = T$ to $t = 0$.

Proof. We show that the recurrence formula of Proposition 2 amount to solve a shortest path in an acyclic graph G where paths represent all feasible columns and the value of a path is the reduced cost of a column. The acyclic graph is a connected graph $G = \cup_{t=1}^{T} G^t$, where $G^t = (V^{t-1} \cup V^t, A^t)$ is a bipartite graph associated with each pair of periods $(t-1, t)$, such that:

$$V^t = \{(j, l, a, t)_{j \in \mathcal{C}, l \in \{1,..,\bar{l}\}, a \in \{1,..,\bar{a}\}}\} \cup \{(f, l, t)_{l \in \{1,..,\bar{l}\}}\}$$

where

- (j, l, a, t), $j \in \mathcal{C}, l \in \{1, .., \bar{l}\}, a \in \{1, .., \bar{a}\}$ represent cultivation state nodes, meaning that plot is in state (j, l, a) at period t
- (f, l, t), $l \in \{1, .., \bar{l}\}$ represent fallow state nodes, meaning that plot has been left fallow for l periods.

and

$$A^t = \{(u, v) \in V^{t-1} \times V^t : \text{ state } u \text{ can precede state } v\}$$

represents feasible state sequences between each pair of nodes from $t-1$ to t according to the flow conservation constraints (5-11). Values λ_{uv}^t on arcs $(u, v) \in A^t$ are equal to the opposite of the yield associated to state v after state u, multiplied by the dual variable associated with state v, i.e.

$$\lambda_{uv}^t = \begin{cases} -\mu_{jt} w_{ij}^{la} & if \quad u = (i, l, a-1, t-1), \quad v = (j, l, a, t), \\ -\mu_{jt} w_{fj}^{l1} & if \quad u = (f, l, t-1), \quad v = (j, l, 1, t), \\ 0 & if \quad u = (j, l, a, t-1), \quad v = (f, 1, t), \\ 0 & if \quad u = (f, l, t-1), \quad v = (f, l+1, t), \\ 0 & if \quad u = (f, \bar{l}, t-1), \quad v = (f, \bar{l}, t) \end{cases} \tag{13}$$

Figure 1 illustrates an example of a bipartite state subgraph G^t defined on $C = \{Rice, Potatos\}$ with $(\bar{l}, \bar{a}) = (2, 2)$.

By construction, there is a 1-1 correspondence between paths in graph G and feasible columns of the Master Program, i.e. each path in G corresponds to a crop rotation on the plot, and vice versa. Also by construction, the value of a path in G is the reduced cost of the corresponding column. As graph G is acyclic, it can be solved by the Bellman-Ford algorithm computing for $t = 1, \ldots, T$

$$\pi_v = \min_{u:(u,v) \in A^t} (\pi_u + \lambda_{uv}^t) \quad \forall v \in V^t \tag{14}$$

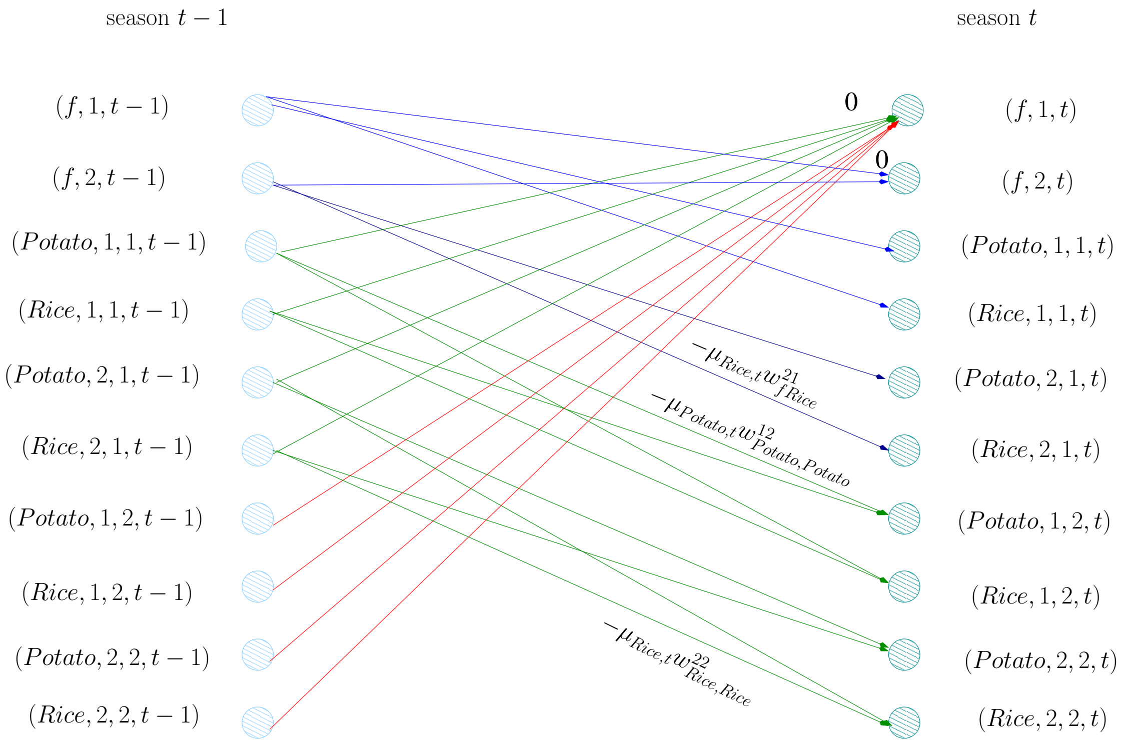

Figure 1: Bipartite subgraph $G^t = (V^{t-1} \cup V^t, A^t)$ with $(\bar{l}, \bar{a}) = (2, 2)$

with initial conditions $\pi_{f,\bar{l},0} = 1$, $\pi_v = \infty$ for $v \neq (f, \bar{l}, 0)$, and output $\min_{v \in V^T} \pi_v$. As $v \in V^t$ describes all possible states (j, l, a, t) and (f, l, t), the above Bellman-Ford recurrence formula are equivalent to the dynamic programming formula of Proposition 2. The backtracking process to find the optimal column is the same as backtracking from $t = T$ to $t = 0$ in graph G to find the path associated with the optimal value. $\square$

The next section is devoted to computational results comparing the column generation scheme to a *direct solving* of the arc-flow model by a commercial solver.

5 Computational experiments

To evaluate the contribution of the column generation approach, we have modeled the explicit arc-flow model whose constraints on plots are constraints (5)-(11) with our problem specifications. In the following sections, *direct solving* means that the solution of the explicit arc-flow model is computed by the MIP-solver of cplex.
We have generated $(MSCRP)_S$ instances in the same way as in [2].

Regarding the column generation strategies, we opted to initialize the first restricted master problem with a heuristic set of columns provided by a greedy algorithm and to add, at each iteration, a set of columns of negative reduced cost obtained from solving the pricing problem by the dynamic programming method described above. Finally, to get integer solutions, we use an enumeration heuristic based on a depth-first search, within a specified time limit, based on the last master problem. We call this resolution scheme the Integer Column Generation Heuristic (ICGH).

The comparative tests are performed for time horizons $T = 20, 30, 40$, and 60 seasons (resp. 10, 15, 20, and 30 years) that correspond to a long term strategic planning in a context of sustainable development. On the other hand, for each class of instances of a fixed time horizon we evaluate the impact of varying the area S which gives another point of view that is more relevant for farmers than the long-term planning aspect.
All the computations reported in this section have been carried out on a DELL Latitude D610 Personal Computer with Pentium M-1.73 Ghz processor and 512 Mo RAM. The code has been written in C++ and the solver is Cplex 12.0.

About agricultural data, we considered a set $\mathcal{C}$ of two crops such as each one can precede the other, and we fixed the maximal cultivation and fallow ages to $\overline{a} = 4$, $\overline{l} = 5$ respectively as in [2].

The first observation that can be made concerns the continuous relaxation value. As expected the value of the column generation relaxation, which is a lagrangian relaxation, is, in most cases, tighter (very slightly, however) than the classical continuous relaxation of *direct solving* and is obtained in very short time as shown in Table 1 where average CPU indicates the average running time over 10 instances of different S for each class of a fixed T. In addition, as the problem size increases (eg. T = 60 seasons) *direct solving* by Cplex is even unable to find a feasible solution for 3 instances within the allowed time of 3600s CPU.

T	Continuous resolution of:			
	Master problem by CG		Arc-flow model by Cplex	
	Optimal value	average CPU	Optimal value	average CPU
20	18.57	1.88	18.54	82.91
30	19.07	7.23	19.05	247.21
40	19.23	18.46	19.22	563.86
60	19.32	99.37	19.32	***

Table 1: Continuous resolution of Master problem vs. Arc-flow model

Table 2 presents the results of the Integer Column Generation Heuristic (ICGH) (i.e column generation followed by the enumeration heuristic) compared to *direct solving*. We compare the two methods in terms of: number of used plots (which, multiplied by corresponding surface S equals the objective value $z(S)$) and computational time CPU. We also report the time to the best integer solution found by both methods.
We can see that, for most instances, (ICGH) provides integer solutions of high quality. More precisely, over the 40 instances we have that:

- for 95% of instances integer solutions found by (ICGH) are either optimal, or lower or equal to the best integer bound reached by *direct solving* on its time limit

		Integer Column Generation Heuristic				Direct Solving		
T	S	# plots	CG CPU	Total CPU	CPU-2-best	# plots	CPU	CPU-2-best
20	0.1	**187**	1.34	>3600	1.34	188	>3600	3500
	0.2	94	1.31	>3600	1.31	94	>3600	2813
	0.5	38	1.51	1.52	1.52	38	477,46	477,46
	0.7	28	1.56	>3600	1.56	28	>3600	619,64
	1	20	3.66	>3600	3.66	20	>3600	270.23
	2	11	2.44	334.44	2.44	11	>3600	97.11
	4	6	1.84	2.11	2.11	6	2648	40.31
	6	4	1.65	1.69	1.69	4	11	11.07
	8	3	1.88	4.25	4.25	3	94	94.04
	10	3	1.64	1.89	1.89	3	3.95	3,95
30	0.1	**192**	6.14	>3600	6.14	**	>3600	>3600
	0.2	**97**	8.01	>3600	8.01	**	>3600	>3600
	0.5	40	7.11	>3600	7.11	40	>3600	2282,64
	0.7	28	7.48	7.58	7.58	28	951,57	951,57
	1	**20**	9.30	9.41	9.41	21	>3600	755,29
	2	11	6.78	6.94	6.94	11	>3600	359,63
	4	6	1.96	2.02	2.02	6	62.4	62,4
	6	4	7.46	7.5	7.5	4	31.78	1.78
	8	4	8.09	34.84	8.09	**3**	550.85	550.85
	10	4	10.06	78	10.06	**3**	569.9	569.9
40	0.1	**194**	19.51	>3600	19.51	**	>3600	>3600
	0.2	**97**	16.07	60.43	60.43	**	>3600	>3600
	0.5	40	16.3	601	16.3	**	>3600	>3600
	0.7	28	22.23	46.43	46.43	28	3469,1	3469,1
	1	20	15.37	15.9	15.9	20	1479.3	1479.3
	2	**11**	13.3	216.8	216.8	12	>3600	954.42
	4	**6**	13.98	61	61	7	>3600	279.3
	6	4	24.74	31.16	31.16	4	1476.48	1476.48
	8	4	22.15	103.83	22.15	4	>3600	140.13
	10	4	21	2048	21	4	>3600	71.32
60	0.1	**195**	102.5	>3600	102.5	**	>3600	>3600
	0.2	**98**	128.8	>3600	128.8	**	>3600	>3600
	0.5	**40**	108.8	>3600	118.8	**	>3600	>3600
	0.7	**28**	123.14	123.87	123.87	**	>3600	>3600
	1	**20**	57.69	70.83	70.83	**	>3600	>3600
	2	**11**	67.21	>3600	615.21	13	>3600	2243.66
	4	**6**	60.21	640	640	8	>3600	1348.34
	6	4	119.4	363.92	363.92	4	1458.6	1458.6
	8	4	102	>3600	102	4	>3600	183.49
	10	4	124	3400	124	4	>3600	190.47

Table 2: Comparison between *direct solving* with Cplex and integer column generation heuristic ICGH
** a feasible integer solution was not obtained within one hour

- for 37.5% of instances they are strictly better than the best integer bounds obtained by *direct solving*
- for only 5% of instances, they are worse than *direct solving* with a gap of 1 unit corresponding to only one supplementary plot

Furthermore, *direct solving* failed to find integer solutions in 25% of the cases.

Concerning the running time of the two methods, *direct solving* is clearly more time-consuming than column generation. However, in the integer column generation heuristic, the enumeration heuristic dedicated to get integer solutions is still time consuming (see Total CPU), but as reported in column CPU-2-best, in most cases, this time is spent to prove optimality of the obtained integer solutions that are often found at the beginning of the enumeration process.
On the other hand, *direct solving* is even unable to find feasible solutions when the surface value decreases and the time horizon increases, while column generation had the same trend for any value of S but increases only with T with a very low computational time. Plots of figures $2-3$ express statistical performance in terms of total running time to the best integer of each resolution method in function of the surface area in figure 2 and in function of the running time in figure 3. In order to have an homogeneous comparison, this average is computed over all T for a given S for figure 2, and it is computed over all S for a given T for figure 3.

We clearly see in figure 2 that the behavior of *direct solving* depends on the value of S, whereas (ICGH) is insensitive to this parameter. Also, we observe in figure 3 the dramatic increase of the running time while increasing the size of the problem, when column generation increases in a monotonic way with a small slope.

In conclusion, these tests show that our method based on column generation outperforms *direct solving* on all the following performance criteria: robustness, solution quality and running time.

6 Conclusion

In this paper, we presented a heuristic resolution scheme based on column generation for Minimum-Space Crop Rotation Planning. The master problem is modeled as a covering integer program and the pricing problem is efficiently solved by dynamic programming. Numerical tests demonstrate that this resolution scheme can be used on problems with up to 60 seasons, as well as problems of small values of surface area where *direct solving* failed to find solutions in the time limit. The method is quite robust then for long-term strategic planning in the framework of an environmental evaluation, and for small division of land as this is often the case in developing countries.

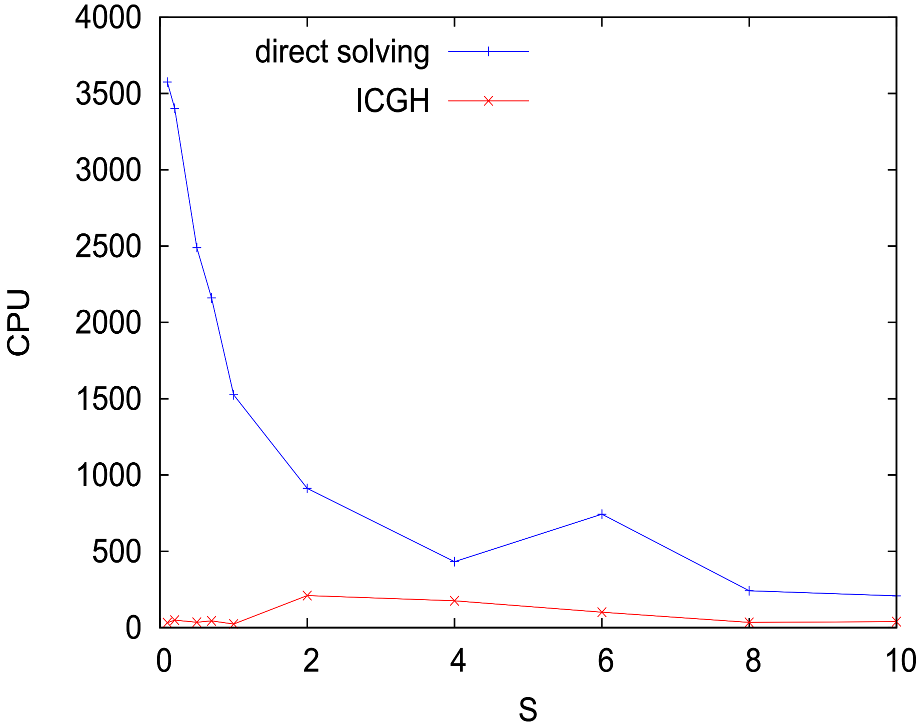

Figure 2: Evolution of CPU-time with surface $S = 0.1, 0.2, .., 10$

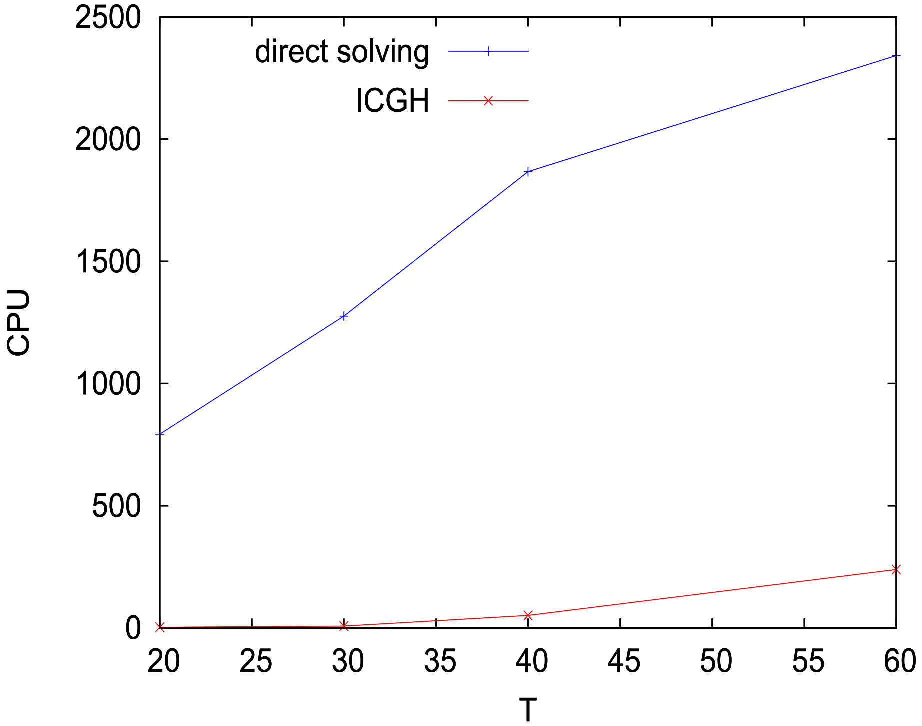

Figure 3: Evolution of CPU-time with horizon length $T = 20, 30, 40, 60$

References

[1] J.L. Lemalade, L. Alfandari , A. Nagih and G. Plateau. A MIP flow model for crop rotation planning in a sustainable development context. EURO XXI Conference, Reykjavik, July 3-6, 2006.

[2] L. Alfandari, J.L Lemalade, A. Nagih, G. Plateau. A MIP flow model for crop-rotation planning in a context of forest sustainable developement. *Annals of Operational Research*, DOI: 10.1007/s10479-009-0553-0, 2009.

[3] L.M.R. dos Santos, P. Michelon, M.N Arenales, R.H.S. Santos. Crop rotation scheduling with adjacency constraints. *Annals of Operational Research*, DOI: 10.1007/s10479-008-0478-z, 2008.

[4] W.K. Klein Haneveld, A.W. Stegman. Crop succession requirements in agricultural production planning. *European Journal of Operational Research*, 166(2) : 406–429, 2005.

[5] P. C. Gilmore, R. E. Gomory. A Linear Programming Approach to the Cutting-Stock Problem. *Operations Research*, 9 :6, 849–859, 1961.

[6] N.K. Detlefsen, A.L. Jensen. Modelling optimal crop sequences using network flows. *Agricultural Systems*, 94-2: 566-572, 2007.

[7] S. Dogliotti, W.A.H. Rossing, M.K. van Ittersum. ROTAT, a tool for systematically generating crop rotations. *European Journal of Agronomy*, 19 : 239–250, 2003.

[8] C Durr, E. Goles, I. Rapaport, E. Rémila. Tiling with bars under tomographic constraints. *Theoretical Computer Science*, 290 :3, 1317–1329, 2003.

[9] T. El-Nazer, B.A. McCarl. The choice of crop-rotation: a modelling approach and case study. *Amer. J. Agri. Economics*, 68, 127–136, 1986.