Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 166 (2005) 406–429
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

O.R. Applications

Crop succession requirements in agricultural production planning

W.K. Klein Haneveld *, A.W. Stegeman [1]

Department of Econometrics and Operations Research, University of Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands

Received 14 July 2003; accepted 11 March 2004
Available online 19 May 2004

Abstract

A method is proposed to write crop succession requirements as linear constraints in an LP-based model for agricultural production planning. Crop succession information is given in the form of a set of inadmissible successions of crops. The decision variables represent the areas where a certain admissible sequence of crops is cultivated. The number of decision variables may be reduced by forming suitable combinations of crop sequences. For this purpose, an algorithm is presented. Also, multi-year linear programming models for farm production planning containing crop succession constraints are considered. It is shown that, under some regularity conditions, a stationary cropping plan is an optimal solution of such a model. Finally, it is discussed how to determine, given a collection of inadmissible sequences, crop sequences which are inadmissible but do not contain inadmissible subsequences. The length of the longest of these sequences determines the length of the crop sequences taken into account in the model.

Keywords: Agriculture; Crop rotation; Crop succession requirements; Production planning; Linear programming

1. Introduction

In agricultural production planning it may happen, and this is often the case, that succession of certain crops on the same piece of land is not allowed or not advisable. Otherwise soil fertility will decrease or the crops may become susceptible to diseases, plagues or weeds. On the other hand, a certain succession of crops may be recommended. We say that in these situations certain *crop succession requirements* have to be fulfilled. For example, in some regions cotton should not be grown after cotton because remaining seeds may cause pests; sorghum after sorghum may cause problems with the weed striga; potatoes should not succeed potatoes because of the occurrence of nematodes in the soil. And soya-beans is

* Corresponding author. Tel.: +31-503-633-786; fax: +31-503-633-720.
E-mail addresses: w.k.klein.haneveld@eco.rug.nl (W.K. Klein Haneveld), a.w.stegeman@ppsw.rug.nl (A.W. Stegeman).
[1] Address: Heijmans Institute, University of Groningen, Grote Kruisstraat 2/1, 9712 TS Groningen, The Netherlands.

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2004.03.009

recommended to alternate with grains in order to restore the fertility. Often, it is important to leave the land lying fallow so that natural vegetation can restore the fertility of the soil. Maize and other grains should, for instance, be followed by fallow. Or it might be advisable to let arable land alternate with grassland for cattle. In practice, the crop succession requirements usually determine *crop rotation cycles*. A crop rotation cycle is a sequence of crops (e.g. soya-beans, maize, fallow) which satisfies the crop succession requirements if applied cyclically on the same piece of land. A crop rotation cycle is usually implemented in such a way that each year there is approximately the same acreage for each of the crops in the cycle. Often only few crop rotation cycles are taken into consideration, advised by e.g. staff of agricultural experimental stations. A well-known example is the 8-course rotation cycle cotton–fallow–fallow–cotton–fallow–sorghum–fallow–fallow which has been applied for many years in the Gezira scheme in Sudan. Since the introduction of pesticides, insecticides and mechanized cultivation the Gezira rotation scheme has been replaced by the 4-course cycle cotton–wheat–sorghum (or groundnuts)–fallow.

It will be clear that crop succession requirements have important implications for production planning on the farm. Not only is it necessary to take the usage of the land in previous years into consideration, but it is also important to be sure that the production plan is such that also in the future good (i.e. high yielding) production plans are possible. The implementation of a fixed recommended crop rotation cycle may take into account these considerations, but as a consequence there is not much freedom in choosing production plans for future years.

In this paper we will focus instead on feasible production plans, i.e. those satisfying the crop succession requirements. In our approach we adopt a mathematical programming framework and show that crop succession requirements can be included as linear constraints in a model for agricultural production planning. The paper is intended to assist in the modeling of real-life situations in agricultural production planning, an example of which is the detailed study of farmers' strategies on the Central Plateau in Burkina Faso by Maatman [4]; see also Maatman et al. [3].

We assume the crop succession requirements are given in the form of crop sequences not allowed (i.e. not advisable) to be cultivated on the same Maximum elapsed time stopping criterionpiece of land. Whether a certain sequence is considered allowed or not may depend on several criteria (e.g. yield levels, revenues, level of soil erosion, occurrence of plant diseases), the relative importance of which may vary in different circumstances. In Section 2 we consider a planning period of T years. Decision variables are introduced which represent the area of land where a certain sequence of m crops is cultivated, the last crop of the sequence being grown in year t. Here, $m+1$ is the length of the longest inadmissible crop sequence not containing any inadmissible subsequence. It appears that the problem of determining whether the cropping plan of year $t-1$ is compatible with the cropping plan of year t, can be interpreted as a max-flow problem. In this way, the crop succession requirements may be written as linear constraints in the decision variables. In Section 3 we consider linear programming models for farm production planning, which contain crop succession constraints. It is shown that, under some regularity conditions, a multi-year LP-model may be replaced by a one-year model. In this case a stationary cropping plan, i.e. a cropping plan not depending on the year index t, is an optimal solution of the multi-year model. Also, it is shown that a stationary cropping plan is composed of several crop rotation cycles. In Section 4 we again consider a planning period of T years, as in Section 2. Here, we focus on reducing the number of decision variables. For n crops, the number of decision variables for year t equals n^m. We show that this number may be reduced by forming combinations of some of the crop sequences without losing crop succession information. An algorithm is presented which finds the crop sequences that may be combined. In Section 5 a conclusion is presented. In Appendix A to this paper, we discuss how to determine inadmissible crop sequences not containing inadmissible subsequences, when a collection of inadmissible sequences is given. The results of Sections 2.2 and 3 are due to Klein Haneveld [2] and were included in Schweigman [5].

2. Crop succession constraints

2.1. Definitions and assumptions

We consider a piece of land of A ha on which n different crops may be grown, where crop n is interpreted as fallow. We consider a planning period of T years, which are numbered $t = 1, 2, \ldots, T$, and assume that there is one growing season per year. It is assumed that crop succession requirements depend only on the types of crops that are grown and not on the methods of cultivation that are used. Moreover, we assume that the crop succession requirements are given in the form of crop sequences that are not allowed to be cultivated on the same piece of land. We introduce the following definitions. Consider a year t and a series of consequent years $t+1, t+2, \ldots, t+h$, with $h \geqslant 2$. Let $(j_1, j_2, \ldots, j_h)$, be a sequence of (not necessarily different) crops grown on the same piece of land, where j_s, $s = 1, 2, \ldots, h$, refer to the crops grown in year $t+s$. It is said to be an *inadmissible sequence* if one of the following three conditions holds:

(I1) Crop j_s is not allowed to succeed $(j_1, j_2, \ldots, j_{s-1})$, for some $s = 1, 2, \ldots, h$, on the same piece of land.
(I2) Condition (I1) does not hold, but the cultivation of $(x, j_1, j_2, \ldots, j_h)$ on the same piece of land is not allowed for all crops $x = 1, \ldots, n$.
(I3) Conditions (I1) and (I2) do not hold, but the cultivation of $(j_1, j_2, \ldots, j_h, y)$ on the same piece of land is not allowed for all crops $y = 1, \ldots, n$.

A sequence of crops that is not inadmissible is called *admissible*. We assume that the (in)admissibility of a crop sequence does not depend on the year t the sequence was started. Notice that it may happen that $(j_1, j_2, \ldots, j_h)$ is admissible, while $(j_0, j_1, j_2, \ldots, j_h)$ or $(j_1, j_2, \ldots, j_h, j_{h+1})$ are inadmissible, for certain crops j_0 and j_{h+1}. Hence, the admissibility of a certain sequence may depend on the crops preceding or succeeding the sequence. The sequence $(j_1, j_2, \ldots, j_h)$ is regarded as admissible if the succession of the crops $j_1, j_2, \ldots, j_h$ on the same piece of land does not violate the crop succession requirements and an extension of the sequence is possible in the past and the future. However, it is not likely that any sequence will be inadmissible due to (I2) or (I3), since an extension with fallow (i.e. crop n) is usually possible.

We say that the sequence $(i_1, i_2, \ldots, i_k)$ is a *subsequence* of $(j_1, j_2, \ldots, j_h)$ if, for some $l \in [1, h-k+1]$, $k \leqslant h$, there holds

$$(j_1, j_2, \ldots, j_h) = (j_1, \ldots, j_{l-1},\ i_1, \ldots, i_k,\ j_{l+k}, \ldots, j_h).$$

If a sequence is admissible, then all its subsequences are admissible too. Analogously, if a sequence $(j_1, j_2, \ldots, j_h)$ is inadmissible, then all crop sequences containing $(j_1, j_2, \ldots, j_h)$ are inadmissible too. Therefore, the crop succession information may be expressed by all inadmissible sequences with the shortest length h. We call such sequences *minimal inadmissible sequences*. Formally, they are defined as follows.

Definition 2.1. An inadmissible sequence $(j_1, j_2, \ldots, j_h)$ is called *minimal* if both the sequences $(j_2, j_3, \ldots, j_h)$ and $(j_1, j_2, \ldots, j_{h-1})$ are admissible.

As can be seen, a minimal inadmissible sequence does not contain any inadmissible subsequences. The set of all sequences $(j_1, j_2, \ldots, j_h)$ is denoted by $\{1, 2, \ldots, n\}^h$. In the sequel we will sometimes use the notation (i) for an arbitrary sequence $(i_1, i_2, \ldots, i_h)$ and (j) for an arbitrary sequence $(j_1, j_2, \ldots, j_h)$. We denote the length of the longest minimal inadmissible sequence by $m+1$, i.e.

$$m + 1 = \max\{h : \text{ there exists a minimal inadmissible sequence } (j) \in \{1, 2, \ldots, n\}^h\}. \quad (2.1)$$

Since we assume that all single crops are admissible, it follows that $m \geqslant 1$. We assume that all minimal inadmissible sequences are known. In Appendix A we will discuss how to determine minimal inadmissible sequences from a given collection of inadmissible sequences.

To be able to determine all feasible cropping plans for year t, it is sufficient to know:

(a) which crops have been grown in the years $t-m,\dots,t-1$ on each part of the available land, and
(b) all admissible sequences of length $m+1$.

Notice that by (2.1), (b) is equivalent to knowing all minimal inadmissible sequences.

Next, we define the set of all admissible sequences of length m

$$S = \{(j) \in \{1,2,\dots,n\}^m : \ (j) \text{ is admissible}\}.$$

For each year t, we define the following decision variables which indicate the extension of the area where a certain sequence $(j) \in S$ is applied.

$$X_t(j) = X_t(j_1, j_2, \dots, j_m): \ \text{size of the area (in ha) where in year } t-s \text{ crop } j_{m-s} \text{ is grown,}$$
$$s = 0,1,\dots,m-1, \ t = 1,2,\dots,T, \ (j) = (j_1,j_2,\dots,j_m) \in S. \tag{2.2}$$

For $(j) \notin S$, we set $X_t(j) = 0$. Notice that $X_t(j)$ only indicates the *size* of the area where crop sequence (j) is grown and not the *location* of this area on the piece of land under consideration. We will assume that this location is immaterial in the sense that it has no influence on the yield per ha and the required amount per ha of each input (see also Remark 3.4). Since we consider n crops, the number of decision variables may be as large as n^m for each t. In Section 4 we will reduce this number by combining several sequences in S.

It is required that $X_t(j) \geqslant 0$, $(j) \in S$, and

$$\sum_{(j)\in S} X_t(j) = A \tag{2.3}$$

for each $t = 1,2,\dots,T$. This implies that in each year t all land is either cultivated or lying fallow. Notice that the area where in year t crop i is grown, is given by

$$\sum_{(j)\in\{1,2,\dots,n\}^{m-1}} X_t((j),i).$$

The information under (a) above can be derived from the values of $X_{t-1}(i)$, $(i) \in S$, and can be used to determine feasible values for $X_t(j)$, $(j) \in S$. Next, we express the information under (b) by specifying, for each sequence $(i) \in S$, the crops which are allowed to be cultivated after (i) on the same piece of land. In fact, we consider *sequences* which are allowed to succeed one another. This notion is defined below. First, we need the following definition.

Definition 2.2. We say that the sequence $(j_1, j_2, \dots, j_m) \in S$ is a *compatible successor* of the sequence $(i_1, i_2, \dots, i_m) \in S$ if $j_1 = i_2, j_2 = i_3, \dots, j_{m-1} = i_m$.

Hence, (j) is a compatible successor of (i) if it is *logically possible* to cultivate the crops $j_1,\dots,j_m$ in the years $t-m+1,\dots,t$ on the same piece of land where in the years $t-m,\dots,t-1$ the crops $i_1,\dots,i_m$ are cultivated. It can be seen that this is equivalent to the requirement that $j_1 = i_2, j_2 = i_3, \dots, j_{m-1} = i_m$. Notice that for any sequence there are n compatible successors, since there are n possible values for j_m. For each sequence $(i) \in S$ we define a set of all $(j) \in S$ which are compatible successors of (i).

$$\text{COMP}(i) = \{(j) \in S : (j) \text{ is a compatible successor of } (i)\}.$$

If (j) is a compatible successor of (i) this does not imply that (j) is also *allowed to succeed* (i). We define the last notion as follows.

Definition 2.3. We say that the sequence $(j) = (j_1, j_2, \ldots, j_m) \in S$ is *allowed to succeed* the sequence $(i) = (i_1, i_2, \ldots, i_m) \in S$ if (j) is a compatible successor of (i) and the sequence $(i_1, i_2, \ldots, i_m, j_m)$ is admissible.

Notice that Definition 2.3 does not depend on the year t in which the last crop j_m is cultivated. For each sequence $(i) \in S$ we define a set of all $(j) \in S$ that are allowed to succeed (i).

$$\mathrm{SUC}(i) = \{(j) \in S : (j) \text{ is allowed to succeed } (i)\}. \tag{2.4}$$

Hence, $\mathrm{SUC}(i) \subseteq \mathrm{COMP}(i)$ for $(i) \in S$. Since $(i) \in S$ is admissible, it follows by (I2) and (I3) that $\mathrm{SUC}(i) \neq \emptyset$ and that there is a $(j) \in S$ such that $(i) \in \mathrm{SUC}(j)$.

The information under (b) above is now contained in the sets $\mathrm{SUC}(i)$, $(i) \in S$. This implies that the knowledge of $X_{t-1}(i)$, $(i) \in S$, and the sets $\mathrm{SUC}(i)$, $(i) \in S$, is sufficient to determine the feasible region of $X_t(j)$, $(j) \in S$.

2.2. Interpretation as a max-flow problem

Here, we focus on determining all feasible cropping plans $X_t(j)$, when the cropping plan $X_{t-1}(i)$ and the sets $\mathrm{SUC}(i)$ are known and (2.3) is satisfied for $t-1$ and t. By interpreting this problem as a max-flow problem, a system of linear equations in both $X_t(j)$ and $X_{t-1}(i)$ is obtained. These equations are necessary and sufficient conditions for the compatibility of $X_{t-1}(i)$ and $X_t(j)$. We call this system of equations the *crop succession constraints*. For the planning period of T years, the crop succession requirements are satisfied if and only if the crop succession constraints hold for $t = 2, 3, \ldots, T$.

Finding a feasible cropping plan $X_t(j)$ given the cropping plan $X_{t-1}(i)$ is equivalent to finding a feasible allocation of the crops in year t, given the allocations in years $t-m, \ldots, t-1$. This allocation problem may be formulated as follows. Let $X_{t-1}(i)$ and $X_t(j)$, $(i), (j) \in S$, be given. We define the transition variables

$Z_t((i),(j))$: size of the area (in ha) where in years $t-m, \ldots, t-1$ the crop sequence (i) has been grown and in years $t-m+1, \ldots, t$ the crop sequence (j) is grown, $(j) \in \mathrm{COMP}(i)$.

For $(j) \notin \mathrm{COMP}(i)$ we set $Z_t((i),(j)) = 0$. The variables $Z_t((i),(j))$ should satisfy

$$\sum_{(j)\in \mathrm{COMP}(i)} Z_t((i),(j)) = X_{t-1}(i) \quad \forall (i) \in S, \tag{2.5}$$

$$\sum_{(i)\in S:(j)\in \mathrm{COMP}(i)} Z_t((i),(j)) = X_t(j) \quad \forall (j) \in S \tag{2.6}$$

$$Z_t((i),(j)) \geqslant 0 \quad \forall (i),(j) \in S, (j) \in \mathrm{COMP}(i), \tag{2.7}$$

$$Z_t((i),(j)) = 0 \quad \text{if } (j) \notin \mathrm{SUC}(i). \tag{2.8}$$

We may now define the compatibility of the cropping plans $X_{t-1}(i)$, $(i) \in S$, and $X_t(j)$, $(j) \in S$, as follows.

Definition 2.4. We say that the cropping plans $X_{t-1}(i)$, $(i) \in S$, and $X_t(j)$, $(j) \in S$, are compatible if and only if (2.3) holds for $t-1$ and t and there exists a feasible solution $Z_t((i),(j))$ for the system (2.5)–(2.8).

Since (2.3) holds for $t-1$ and t, the sum of the right-hand side members of (2.5) equals A and the same is true of (2.6). The right-hand side members of (2.5) and (2.6) being given we recognize in (2.5)–(2.8) the constraints of a classical transportation problem in which total supply equals total demand and some of the variables $Z_t((i),(j))$ are restricted to zero values.

For our purposes, however, it is more convenient to interpret (2.5)–(2.8) as a max-flow problem. This is done as follows. Consider a network with a source node, supply nodes $(i) \in S$, demand nodes $(j) \in S$ and a sink node. There is a directed arc from the source to every supply node (i) with capacity $X_{t-1}(i)$. Analogously, there is a directed arc from every demand node (j) to the sink with capacity $X_t(j)$. Finally, there is a directed arc from a supply node (i) to a demand node (j) if and only if $(j) \in \mathrm{SUC}(i)$. Arcs from (i) to (j) have infinite capacity. In this way, $Z_t((i),(j))$ represents the flow from (i) to (j) and there exists a feasible solution to (2.5)–(2.8) if and only if there exists a flow of A units from the source to the sink.

By the max-flow min-cut theorem (see Theorem 1.1 in [1]) a flow of A units is also the maximum flow through the network. Indeed, since (2.3) holds for $t-1$ and t, the cuts {source}–{supply nodes, demand nodes, sink} and {source, supply nodes, demand nodes}–{sink} both have capacity A. Hence, by the max-flow min-cut theorem, there exists a flow of A units if and only if there are no cuts (separating the source and sink nodes) with capacity less than A. This observation is used to prove the following theorem.

We consider sets $I, J \subset S$ with the property that no $(j) \in J$ is allowed to succeed any $(i) \in I$, i.e.

$$(j) \notin \mathrm{SUC}(i) \quad \text{for all } (i) \in I \text{ and all } (j) \in J. \tag{2.9}$$

In Section 2.3 it will be discussed how such sets can be determined.

Theorem 2.5. *Assume that* (2.3) *holds for* $t-1$ *and* t. *Then the cropping plans* $X_{t-1}(i)$, $(i) \in S$, *and* $X_t(j)$, $(j) \in S$, *are compatible if and only if*

$$\sum_{(i)\in I} X_{t-1}(i) + \sum_{(j)\in J} X_t(j) \leqslant A, \quad \forall I, J \subset S \text{ satisfying (2.9)}. \tag{2.10}$$

Proof. It suffices to show that (2.10) is equivalent to the requirement that there are no cuts separating the source and sink nodes with capacity less than A. Denote the set of all supply nodes by U and the set of all demand nodes by V. Let $\{U_1, U_2\}$ and $\{V_1, V_2\}$ be disjoint partitions of U and V respectively. An arbitrary cut separating the source and sink nodes is of the form {source, U_1, V_1}–{U_2, V_2, sink}. The capacity of this cut is infinite if there is an arc from U_1 to V_2. If this is not the case, then the capacity of the cut equals

$$\sum_{(i)\in U_2} X_{t-1}(i) + \sum_{(j)\in V_1} X_t(j). \tag{2.11}$$

Hence, there are no cuts with capacity less than A if and only if (2.11) is greater than or equal to A for all cuts with no arcs from U_1 to V_2. Since (2.3) holds for $t-1$ and t, this is equivalent to

$$A - \sum_{(i)\in U_1} X_{t-1}(i) + A - \sum_{(j)\in V_2} X_t(j) \geqslant A. \tag{2.12}$$

The proof is completed by writing (2.12) in the form

$$\sum_{(i)\in U_1} X_{t-1}(i) + \sum_{(j)\in V_2} X_t(j) \leqslant A$$

and setting $I = U_1$ and $J = V_2$. $\square$

We call the system of equations (2.10) the *crop succession constraints*. As can be seen, the crop succession constraints are linear in $X_{t-1}(i)$ and $X_t(j)$. For the whole planning period of T years the crop succession requirements are satisfied if and only if

$$\sum_{(i)\in I} X_{t-1}(i) + \sum_{(j)\in J} X_t(j) \leqslant A, \quad t = 2, 3, \ldots, T \quad \forall I, J \subset S \quad \text{satisfying (2.9)}. \tag{2.13}$$

If necessary, the cropping plan from the year before $t = 1$ may be included by defining parameters $x_0(j)$, $(j) \in S$, analogous to the variables $X_t(j)$ and adding the constraint (2.13) for $t = 1$ with $X_0(j) = x_0(j)$.

The finite planning horizon of T years may result in a situation where the possible cropping plans for the year $T + 1$ result in low yields or revenues. To avoid this situation we may include in (2.13) also $t = T + 1$ and require for example that $X_1(j) = X_{T+1}(j)$. In this way it is guaranteed that the production plan $X_{T+1}(j)$ is a good starting point for the future. This approach will be discussed further in Section 3.1.

2.3. Redundant crop succession constraints

The number of crop succession constraints in (2.10) grows exponentially with n (although the number of admissible sequences in the set S may be substantially less than n^m). However, a lot of constraints in (2.10) may be redundant. Here, we discuss a way to identify combinations of I and J for which the constraint (2.10) is redundant. For $I \subset S$, let $J(I)$ be defined by

$$J(I) = \{(j) \in S : (j) \notin \mathrm{SUC}(i)\ \forall (i) \in I\}.$$

A first observation is that, for a fixed set $I \subset S$, we only need to consider the set $J(I)$ in (2.10), and not all sets $J \subset S$ which satisfy (2.9). This is due to the fact that, for a fixed set $I \subset S$, the term

$$\sum_{(j)\in J} X_t(j)$$

in (2.10), is maximal for $J = J(I)$. Hence, we replace (2.10) by

$$\sum_{(i)\in I} X_{t-1}(i) + \sum_{(j)\in J(I)} X_t(j) \leqslant A \quad \forall I \subset S. \tag{2.14}$$

However, if $J(I) = \emptyset$, then (2.14) is implied by (2.3). Therefore, we only need to consider sets I with $J(I) \neq \emptyset$. We define

$$S_1 = \{I \subset S : J(I) \neq \emptyset\}.$$

Suppose that $I, \tilde{I} \in S_1$ and $I \subset \tilde{I}$. Then $J(\tilde{I}) \subseteq J(I)$ and it may happen that $J(\tilde{I}) = J(I)$. In the latter case, the constraint (2.14) for I is implied by the one for $\tilde{I}$. Hence, we may only consider sets I in the following collection S_2:

$$S_2 = \{I \in S_1 : \text{ there is no } \tilde{I} \in S_1 \text{ such that } I \subset \tilde{I} \text{ and } J(\tilde{I}) = J(I)\}.$$

When we consider only sets $I \in S_2$ there may, however, still be redundant constraints in the system (2.14). The constraint for $I_1 \in S_2$ is redundant if there exists an $I_2 \in S_2$, such that

$$(I_1 \cup J(I_1)) \subset (I_2 \cup J(I_2)).$$

It may for example happen that $I_1 \subset I_2$ and $J(I_1) \subset I_2$. Or that $I_1 \subset J(I_2)$ and $J(I_1) \subset J(I_2)$. Since excluding all possibilities of redundancy is a rather tedious task, we will end our discussion at this point.

Notice that the roles of I and J are interchangeable, i.e. for $J \subset S$ we also could have defined a set $I(J) = \{(i) \in S : (j) \notin \mathrm{SUC}(i)\ \forall (j) \in J\}$ and similar collections S_1 and S_2 as above.

3. Stationary cropping plans in an LP-context

It is well-known that linear programming (LP) is a valuable tool for analysing production planning problems. Also production planning of a farm may be analysed with the use of LP, see e.g. Schweigman [5]. In this section, we consider a generic multi-year LP-model for farm production planning containing crop succession constraints. In Section 3.1 it is shown that, under some regularity conditions, the multi-year LP-model may be replaced by a one-year LP-model. In this case a stationary cropping plan, i.e. $X_t(j) = X(j)$ for all t, is an optimal solution of the multi-year LP-model. In Section 3.2 we show that a stationary cropping plan is composed of several crop rotation cycles. A method to determine these cycles is presented.

3.1. A stationary linear programming model

We formulate the following generic LP-model for agricultural production planning for a planning period of T years. The decision variables are $X_t(j)$, as defined in (2.2). The model is given by

$$\text{Maximize} \quad \sum_{t=1}^{T} \sum_{(j)\in S} c(j) X_t(j), \tag{3.1}$$

$$\text{subject to} \quad \sum_{(i)\in I} X_{t-1}(i) + \sum_{(j)\in J} X_t(j) \leqslant A, \quad t = 2, 3, \ldots, T+1$$

$$\forall I, J \subset S \quad \text{satisfying (2.9)}, \tag{3.2}$$

$$\sum_{(j)\in S} X_t(j) = A, \quad t = 1, 2, \ldots, T, \tag{3.3}$$

$$\sum_{(j)\in S} a(k, (j)) X_t(j) \leqslant b(k), \quad k = 1, 2, \ldots, K, \quad t = 1, 2, \ldots, T, \tag{3.4}$$

$$X_1(j) = X_{T+1}(j) \quad \forall (j) \in S. \tag{3.5}$$

As mentioned in Section 2.2 we also include the variables $X_{T+1}(j)$ in the crop succession constraints (3.2) and add (3.5) to ensure that the production plan $X_{T+1}(j)$ is a good starting point for the future.

The limited availability of land is described by (3.3). For other inputs, the constraints (3.4) are included. The parameters $a(k, (j))$ indicate, for the crop sequence (j), how much of input k is needed to cultivate one ha of crop j_m. The inputs may be labour, manure, money, machines, seeds, etc. The parameters $b(k)$ represent the available amount of input k. Also food requirements may be written as (3.4). The parameters $c(j)$ in the objective function represent, for example, the revenues per ha (i.e. the selling price per kg multiplied by the yield in kg) when crop sequence (j) has been grown. For more details on the interpretation of the constraints (3.4) and the objective function (3.1) we refer to Schweigman [5]; see also Maatman et al. [3] and Maatman [4].

Notice that the parameters $a(k, (j))$, $b(k)$ and $c(j)$ do not depend on t. Therefore, the model (3.1)–(3.5) may be called a *stationary* LP-model. We need the following definition.

Definition 3.1. We call a cropping plan $X_t(j)$, $t = 1, 2, \ldots, T$, $(j) \in S$, *stationary* if $X_t(j) = X(j)$, $t = 1, 2, \ldots, T$, $(j) \in S$.

It follows that if an optimal solution of (3.1)–(3.5) is stationary, it is an optimal solution of

$$\text{Maximize} \quad \sum_{(j)\in S} c(j)X(j), \tag{3.6}$$

$$\text{subject to} \quad \sum_{(i)\in I} X(i) + \sum_{(j)\in J} X(j) \leqslant A \quad \forall I, J \subset S \text{ satisfying (2.9)}, \tag{3.7}$$

$$\sum_{(j)\in S} X(j) = A, \tag{3.8}$$

$$\sum_{(j)\in S} a(k,(j))X(j) \leqslant b(k), \quad k = 1, 2, \ldots, K. \tag{3.9}$$

The following proposition shows that if (3.1)–(3.5) is feasible, then it has an optimal solution which is stationary. This result has an important practical implication: in an optimal strategy the production plan may be chosen the same in all years of the planning period.

Proposition 3.2

(P1) *Problem* (3.1)–(3.5) *is feasible if and only if problem* (3.6)–(3.9) *is feasible.*

(P2) *Suppose* (3.1)–(3.5) *is feasible. Let* $X^*(j)$, $(j) \in S$, *be an optimal solution of* (3.6)–(3.9). *Then* $X_t(j) = X^*(j)$, $t = 1, 2, \ldots, T$, $(j) \in S$, *is an optimal solution of* (3.1)–(3.5).

Proof. Let X be a feasible solution of (3.6)–(3.9). Then $X_t = X$, $t = 1, \ldots, T$, is a feasible solution of (3.1)–(3.5). The proof of (P1) is complete if we show that if (3.1)–(3.5) is feasible, then there exists a feasible solution to (3.1)–(3.5), which is stationary.

Suppose (3.1)–(3.5) is feasible. Since the feasible region is compact, it also has an optimal solution, say X_t^*, $t = 1, \ldots, T$. Define, for $h = 1, \ldots, T$,

$$X^{(h)} = (X_h^*, X_{h+1}^*, \ldots, X_T^*, X_1^*, \ldots, X_{h-1}^*).$$

Then each $X^{(h)}$ is a feasible solution to (3.1)–(3.5). Moreover, since $X^{(1)} = X^*$ and the objective function (3.1) has the same value for all $X^{(h)}$, $h = 1, \ldots, T$, the feasible solutions $X^{(h)}$ are also optimal. Define

$$\overline{X} = \frac{1}{T}\sum_{h=1}^{T} X^{(h)} = \left(\frac{1}{T}\sum_{t=1}^{T} X_t^*, \ldots, \frac{1}{T}\sum_{t=1}^{T} X_t^*\right) = (\overline{X}_1, \ldots, \overline{X}_T).$$

Since the feasible region of (3.1)–(3.5) is convex $\overline{X}$ is a feasible solution. This completes the proof of (P1), since $\overline{X}$ is a stationary solution. Notice that the objective function (3.1) has the same value for $\overline{X}$ as for X^*. Therefore, $\overline{X}$ is an optimal solution of (3.1)–(3.5) and $\overline{X}_1$ is an optimal solution of (3.6)–(3.9).

Next we prove (P2). Suppose (3.1)–(3.5) is feasible. By (P1) also (3.6)–(3.9) is feasible. Since the feasible region of (3.6)–(3.9) is compact it has an optimal solution, say X^0. The stationary solution $X_t = X^0$, $t = 1, \ldots, T$, is feasible for (3.1)–(3.5). Moreover, since $\overline{X}_1$ above has the same objective value (3.6) as X^0, the stationary solution $X_t = X^0$ has the same objective value (3.1) as $\overline{X}$. Hence, $X_t = X^0$ is an optimal solution of (3.1)–(3.5). This completes the proof of (P2). □

Remark 3.3. It is important to note that Proposition 3.2 does not hold if some of the parameters $a(k,(j))$, $b(k)$ and $c(j)$ in the model (3.1)–(3.5) depend on t. This implies that we assume that, for example, crop prices and yields per ha do not change during the T years of the planning period, which is rather unrealistic. If prices or yields change and the current stationary cropping plan $X(j)$ is not optimal anymore, we still need a multi-year LP-model to transform $X(j)$ into the optimal stationary solution. Also, we may not use the possibility of discounting future revenues.

Another situation in which Proposition 3.2 is not valid anymore is when we include the cropping plan from the year before $t = 1$. As explained in Section 2.2, this may be done by defining parameters $x_0(j)$, $(j) \in S$, analogous to the variables $X_t(j)$ and adding the constraint (3.2) for $t = 1$ with $X_0(j) = x_0(j)$.

Remark 3.4. The cropping plans $X_t(j)$, $t = 1, 2, \ldots, T$, $(j) \in S$, only indicate the *sizes* of the areas where in year t the crop sequence (j) is grown and not the *locations* of these areas on the A ha of land that is available. For $t = 2, 3, \ldots, T$, any feasible solution $Z_t((i), (j))$ of (2.5)–(2.8) describes a feasible allocation of the crops in year t. Moreover, for year $t = 1$ any ordering of $X_1(j)$, $(j) \in S$, is a feasible allocation.

We assume that the location of the areas $X_t(j)$ is immaterial in the sense that it has no influence on the required amount per ha of each input and on the yield per ha. However, the freedom of choosing a feasible $Z_t((i), (j))$ and the allocation in year $t = 1$ may still be used to obtain 'good' locations for the crops in each year.

Remark 3.5. The amounts of inputs used per ha and the yield per ha usually depend not only on the type of crop, but also on the method of cultivation. For example, the yield per ha of maize may be influenced by the amount of manure that is applied per ha. For this reason, it may be appropriate to define decision variables $X_t(f, (j))$ indicating the area where crop sequence (j) is grown and method of cultivation f is used. The model (3.1)–(3.5) may then be changed as follows. Since we assume that the crop succession requirements do not depend on the methods of cultivation that are used, we may set

$$X_t(j) = \sum_f X_t(f, (j)), \quad (j) \in S, \quad t = 1, 2, \ldots, T,$$

and leave the crop succession constraints (3.2) unchanged. Also (3.3) remains the same. The constraints (3.4) may be changed into

$$\sum_{(j) \in S} \sum_f a_t(k, f, (j)) X_t(f, (j)) \leqslant b_t(k), \quad k = 1, 2, \ldots, K, \quad t = 1, 2, \ldots, T.$$

In this way, it is expressed that the required amounts of inputs and the yield may also depend on the method of cultivation f. Analogously, the objective function (3.1) may be changed into

$$\text{Maximize} \sum_{t=1}^{T} \sum_{(j) \in S} \sum_f c_t(f, (j)) X_t(f, (j)).$$

Also, we may replace (3.5) by

$$X_1(f, (j)) - X_{T+1}(f, (j)) = 0 \quad \forall f, \quad \forall (j) \in S.$$

For this new model we then have an analogous result as Proposition 3.2 in terms of the decision variables $X_t(f, (j))$.

Notice that when $m = 1$, the decision variables indicate the size of the area where a *single crop* is grown. Hence, it is not possible to express a dependence of the yield of the current crop on the crop grown last year on the same piece of land. Therefore, in this case one may prefer to use the decision variables for the situation $m = 2$.

3.2. Crop rotation cycles

In this section we discuss how to determine, for a stationary cropping plan $X(j)$, the pieces of land where each crop should be cultivated each year. We show that a stationary cropping plan is composed of several crop rotation cycles. How to determine these rotation cycles and the sizes of the areas where they should be applied, is explained in the proof of Proposition 3.7. We define a crop rotation cycle in terms of sequences in S as follows.

Definition 3.6. A collection of sequences $C = \{(j)^1, (j)^2, \ldots, (j)^h\} \subseteq S$, $h \geqslant 1$, is called a *crop rotation cycle* if $(j)^k$ is allowed to succeed $(j)^{k-1}$, $k = 2, 3, \ldots, h$, and $(j)^1$ is allowed to succeed $(j)^h$.

Hence, a crop rotation cycle is defined as a collection of sequences in S, that is allowed to be applied cyclically on the same piece of land. If $h = 1$ in Definition 3.6, then the single sequence in C consists of m times the same crop. Hence, the same crop is grown permanently. If $h = 2$, then cycle C consists of two different crops which are alternated. For $h = 3$ there are two possibilities: either a sequence of three different crops is repeated, or periods of two years the same crop are alternated with one-year periods in which a different crop is grown. In general, the cycle C consists of at most h different crops.

In Definition 3.6, the number of sequences in C and the length in years of the crop rotation cycle defined by C, are both equal to h. For $h \leqslant m$, this may not seem obvious. However, in this case there holds for any sequence (j) in C,

$$j_k = j_{k+h}, \quad k = 1, \ldots, m - h.$$

The main result of this section is formulated in the following proposition.

Proposition 3.7. *Let* $X(j)$, $(j) \in S$, *satisfy* (3.7) *and* (3.8). *Define*

$$S_X = \{(j) \in S : X(j) > 0\}. \tag{3.10}$$

Then there exist crop rotation cycles $C_s \subseteq S_X$, $s = 1, 2, \ldots, v$, *which can be applied such that in every year the area where sequence* (j) *is grown, is equal to* $X(j)$, $(j) \in S_X$.

Proof. Let $X(j)$, $(j) \in S$, satisfy (3.7) and (3.8). Then, by Theorem 2.5, there exists a feasible solution $Z((i), (j))$, $(i), (j) \in S$, $(j) \in \mathrm{COMP}(i)$, to the following system:

$$\sum_{(j) \in \mathrm{COMP}(i)} Z((i), (j)) = X(i), \quad \forall (i) \in S, \tag{3.11}$$

$$\sum_{(i) \in S : (j) \in \mathrm{COMP}(i)} Z((i), (j)) = X(j), \quad \forall (j) \in S, \tag{3.12}$$

$$Z((i), (j)) \geqslant 0, \quad \forall (i), (j) \in S, \quad (j) \in \mathrm{COMP}(i), \tag{3.13}$$

$$Z((i), (j)) = 0 \quad \text{if } (j) \notin \mathrm{SUC}(i). \tag{3.14}$$

We choose a feasible solution Z of (3.11)–(3.14). Next, we construct a network of nodes $(j) \in S_X$. There is a directed arc from node (i) to node (j) if and only if $Z((i), (j)) > 0$. The value of $Z((i), (j))$ is interpreted as the flow from node (i) to node (j). The total flow in the network equals A, since by (3.8), (3.10), (3.11) and (3.12) there holds

$$\sum_{(i) \in S_X} \sum_{(j) \in S_X} Z((i), (j)) = A.$$

From (3.11) and (3.12) it can be concluded that for each node (j) in the network, the total flow into (j) is equal to the total flow out of (j). Because of this flow conservation property, each node (j) in the network is contained in a flow cycle. Suppose node (j) is contained in some flow cycle C_s. Let w_s be the minimum flow between two consecutive nodes in C_s, i.e.

$$w_s = \min\{Z((i),(k)) : \text{ cycle } C_s \text{ contains the arc } (i) \to (k)\}.$$

Because of the flow conservation property we know that $w_s > 0$. The flow in the whole network can be described by a finite number of flow cycles C_s, $s = 1, 2, \ldots, v$. Indeed, if a cycle C_1 is deleted from the network, i.e. the flow through the nodes in C_1 is decreased by w_1, then the flow conservation property still holds for all nodes in the network. Hence, we may delete another cycle C_2. In this way, the flow in the network becomes zero after deleting a finite number of cycles.

The flow cycles $C_1, \ldots, C_v$ determine the values of $Z((i),(j))$ completely, since

$$Z((i),(j)) = \sum_{s:\text{arc } (i)\to(j) \text{ in } C_s} w_s. \tag{3.15}$$

Let h_s be given by

$$h_s = \text{card}(C_s), \quad s = 1, 2, \ldots, v.$$

The procedure above splits up the transitions $Z((i),(j))$ into flows w_s in cycles C_s of lengths h_s. Hence, there holds

$$\sum_{s=1}^{v} h_s w_s = A. \tag{3.16}$$

From Definition 3.6 it is clear that a flow cycle C_s may be interpreted as a crop rotation cycle. Moreover, when each rotation cycle C_s is applied on an area of $h_s w_s$ ha, then a rotation scheme can be constructed where each year the area of crop sequence (j) is $X(j)$. This is done as follows.

In the first year, the total area of A ha is divided into v plots with sizes $h_s w_s$, $s = 1, 2, \ldots, v$. The plot for cycle C_s is divided into h_s parts of size w_s. On each part a different (j) in C_s is grown. In the following years, each part of the plot for C_s follows the schedule dictated by the rotation cycle C_s. By (3.16), it follows that all land is used. Moreover, by (3.15) and (3.12), the area for $(j) \in S_X$ is equal to

$$\sum_{(i)\in S_X} \sum_{s:\text{arc } (i)\to(j) \text{ in } C_s} w_s = \sum_{(i)\in S_X} Z((i),(j)) = X(j).$$

This completes the proof of Proposition 3.7. □

Remark 3.8. In the proof of Proposition 3.7 there may be many possible ways of choosing the cycles $C_1, \ldots, C_v$. For example, a different feasible solution Z of (3.11)–(3.14) may be taken. Also, the order of picking the cycles may be varied. Or we may choose to consider only cycles C_s which do not contain the same sequence (j) more than once. Another possibility is to decrease the flow in cycle C_s by a quantity less than w_s.

The cropping plan X is required to satisfy only (3.7) and (3.8). We may for example take X to be an optimal solution of (3.6)–(3.9). The transitions Z satisfying (3.11)–(3.14) may also be chosen by using preference criteria, e.g. some objective function. If X and Z are chosen optimal (in some sense), then $C_1, \ldots, C_v$ may be called *optimal crop rotation cycles*.

4. Preprocessing: Reducing the number of decision variables by combining crop sequences

Here we leave the LP-context of the previous section and reconsider the crop succession constraints (2.10). We show that some groups of admissible sequences of length m with the same last crop may be combined without losing crop succession information. As a result, the number of decision variables and the number of equations in (2.10) is reduced. It should be noted that our aim is to find groups of sequences that *can* be combined without losing crop succession information. This does not imply that they also *have to* be combined. For example, when using model (3.1)–(3.5) one may choose not to combine certain admissible sequences (i) and (j) when $c(i) \neq c(j)$ or $a(k,(i)) \neq a(k,(j))$ for some input k.

In Section 2.1 it is stated that to determine the feasible cropping plans for year t, it is sufficient to know (a) which crops have been grown in the years $t-m,\dots,t-1$ on each part of the available land, and (b) all admissible sequences of length $m+1$. Here we argue that, *given the crop succession information* (b), in order to satisfy the crop succession requirements it may not be necessary to know *exactly* which crops have been grown in the years $t-m,\dots,t-1$ on each part of the available land. In some cases, it may be sufficient to know that in a certain year one crop from a certain *set* of crops has been grown on a certain part of the available land. Hence, *given the crop succession information* (b), we do not need *all* information under (a) to determine the feasible cropping plans for year t.

Recall that we expressed (a) in terms of $X_{t-1}(i)$, $(i) \in S$, and (b) in terms of the sets $\mathrm{SUC}(i)$, $(i) \in S$. In this section, we combine certain groups of sequences in S into new sequences and thus obtain a new set of sequences S^*. A sequence $(j) \in S^*$ still has length m. The difference with sequences in S is that now the element j_k may represent a *set* of crops instead of only one crop. For each $(i) \in S^*$, the set of successors $(j) \in S^*$ is denoted by $\mathrm{SUC}^*(i)$. We define decision variables $X_t(j)$, $(j) \in S^*$, analogous to (2.2). Now $X_{t-1}(i)$, $(i) \in S^*$, do not contain all information under (a). However, the combinations may be chosen in such a way that the sets $\mathrm{SUC}^*(i)$, $(i) \in S^*$, still contain all information under (b). Moreover, given $X_{t-1}(i)$, $(i) \in S^*$, and $\mathrm{SUC}^*(i)$, $(i) \in S^*$, it is then still possible to determine all feasible cropping plans for year t. The combinations of sequences in S for which this holds are determined by the sets $\mathrm{SUC}(i)$, $(i) \in S$.

Forming combinations of groups of sequences in S as above, implies a reduction of the number of decision variables. For each t, the number of decision variables $X_t(j)$ in (2.2) equals $\mathrm{card}(S)$ and may be of the order n^m. In Section 4.1 we illustrate the process of combining sequences with a small-scale example. A larger example is formulated to express the need for a sequence-combining algorithm. Such an algorithm is presented in Section 4.2. In Section 4.3 we show that when this algorithm is used to combine sequences, the information under (b) above is preserved. Moreover, the cropping plans $X_{t-1}(i)$, $(i) \in S^*$, and $X_t(j)$, $(j) \in S^*$, are compatible if and only if crop succession constraints analogous to (2.10) are satisfied.

4.1. A small-scale example

Consider $n = 3$ crops and suppose that crop 1 may succeed crop 1 provided that crop 2 is grown on the land two years before. In this case, $(1,1,1)$ and $(3,1,1)$ are the minimal inadmissible sequences. Hence, $m = 2$ and

$$S = \{(1,1),(1,2),(1,3),(2,1),(2,2),(2,3),(3,1),(3,2),(3,3)\}.$$

The sets $\mathrm{SUC}(i)$, $(i) \in S$, are given by

$$\mathrm{SUC}(2,1) = \{(1,1),(1,2),(1,3)\},$$
$$\mathrm{SUC}(1,1) = \mathrm{SUC}(3,1) = \{(1,2),(1,3)\},$$
$$\mathrm{SUC}(1,2) = \mathrm{SUC}(2,2) = \mathrm{SUC}(3,2) = \{(2,1),(2,2),(2,3)\},$$
$$\mathrm{SUC}(1,3) = \mathrm{SUC}(2,3) = \mathrm{SUC}(3,3) = \{(3,1),(3,2),(3,3)\}.$$

As we see, $\mathrm{SUC}(1,1) = \mathrm{SUC}(3,1)$. This implies that if in year t crop 1 is grown on a certain piece of land, the possibilities for year $t+1$ do not depend on whether the crop grown in year $t-1$ is crop 1 or crop 3. Hence, intuitively, we may combine the sequences $(1,1)$ and $(3,1)$ into the new sequence $(1 \text{ or } 3, 1)$. The set $\mathrm{SUC}(1 \text{ or } 3, 1)$ is then equal to $\mathrm{SUC}(1,1)$. The set $\mathrm{SUC}(2,1)$ contains $(1,1)$ but not $(3,1)$. In this case, we just replace $(1,1)$ by the new sequence $(1 \text{ or } 3, 1)$. Logically, this is not a problem, since a succession of

$(2,1)$ by $(1 \text{ or } 3,1)$ implies that $(2,1,1)$ is grown. The same is true for $\mathrm{SUC}(1,3)$, which contains $(3,1)$ but not $(1,1)$. The sets $\mathrm{SUC}(i)$ change into

$$\mathrm{SUC}(2,1) = \{(1 \text{ or } 3,1),(1,2),(1,3)\},$$
$$\mathrm{SUC}(1 \text{ or } 3,1) = \{(1,2),(1,3)\},$$
$$\mathrm{SUC}(1,2) = \mathrm{SUC}(2,2) = \mathrm{SUC}(3,2) = \{(2,1),(2,2),(2,3)\},$$
$$\mathrm{SUC}(1,3) = \mathrm{SUC}(2,3) = \mathrm{SUC}(3,3) = \{(1 \text{ or } 3,1),(3,2),(3,3)\}.$$

Since $\mathrm{SUC}(1,2) = \mathrm{SUC}(2,2) = \mathrm{SUC}(3,2)$, we may also combine $(1,2)$, $(2,2)$ and $(3,2)$ into the sequence $(x,2)$, where x denotes all crops. The same holds for $(1,3)$, $(2,3)$ and $(3,3)$. We combine them into $(x,3)$. Using the same arguments as above, there holds

$$\mathrm{SUC}^*(2,1) = \{(1 \text{ or } 3,1),(x,2),(x,3)\},$$
$$\mathrm{SUC}^*(1 \text{ or } 3,1) = \{(x,2),(x,3)\},$$
$$\mathrm{SUC}^*(x,2) = \{(2,1),(x,2),(x,3)\},$$
$$\mathrm{SUC}^*(x,3) = \{(1 \text{ or } 3,1),(x,2),(x,3)\}.$$

We changed the notation from $\mathrm{SUC}(i)$ to $\mathrm{SUC}^*(i)$, since all $\mathrm{SUC}^*(i)$ are different and no more combinations can be formed in the same way as above. Hence, the final set of sequences is

$$S^* = \{(2,1),(1 \text{ or } 3,1),(x,2),(x,3)\}.$$

Notice that if $(x,3)$ is succeeded by $(x,2)$, then either $(1,3,2)$, $(2,3,2)$ or $(3,3,2)$ is grown. From the sets $\mathrm{SUC}^*(i)$ it can be concluded that $(1,1,1)$ and $(3,1,1)$ are the only inadmissible sequences. Hence, after combining the sequences as we did, we still know all admissible sequences of length $m+1=3$.

For $(i) \in S^*$ and $t = 1,2,\ldots,T$, we define a corresponding variable $X_t(i)$ as follows:

$X_t(2,1)$: area where in year t crop 1 is grown and in year $t-1$ crop 1 was grown.
$X_t(1 \text{ or } 3,1)$: area where in year t crop 1 is grown and in year $t-1$ crop 1 or crop 3 was grown.
$X_t(x,2)$: area where in year t crop 2 is grown.
$X_t(x,3)$: area where in year t crop 3 is grown.

By forming the combinations as above we have reduced the number of decision variables from 9 to 4. With respect to the compatibility of $X_{t-1}(i)$, $(i) \in S^*$, and $X_t(j)$, $(j) \in S^*$, the analogue of Theorem 2.5 holds. This result is formulated in Theorem 4.7.

In the small-scale example above the combinations that may be formed can be determined by hand. However, for an example with a larger number of crops n or a larger value of m it may be more convenient to have a computer program which determines the sequences that may be combined. Such a larger example is the following.

Example 4.1. Suppose we have the crops cotton (crop 1), sorghum (crop 2), soya-beans (crop 3) and fallow land (crop 4). We assume that the following crop succession requirements have to be fulfilled.

(R1) After cotton and sorghum, there should be at least a one-year fallow.
(R2) Cotton may succeed cotton only after a two-year fallow in between.
(R3) Sorghum may succeed sorghum only after a two-year fallow in between.
(R4) Cotton may succeed sorghum only after a two-year fallow in between.
(R5) Soya-beans may be succeeded only by sorghum and fallow.
(R6) At least three out of every five years the land should lie fallow.

Determining all minimal inadmissible sequences for (R1)–(R6) is a rather complex task. However, in Appendix A an algorithm is presented which determines the minimal inadmissible sequences from a given collection of inadmissible sequences. Using this algorithm, we obtain the minimal inadmissible sequences for (R1)–(R6) and the value of $m = 4$. Starting with the set S of admissible sequences of length 4 and using the same reasoning as above, the following set S^* is obtained:

$$\begin{array}{llllllll} (3,2,4,4) & (3,4,1,4) & (3,4,3,4) & (4,3,2,4) & (4,3,4,1) & (4,3,4,3) & (4,4,1,4) & (4,4,2,4), \\ (4,4,3,2) & (4,4,3,4) & (4,4,4,3) & (t,4,2,4) & (s,4,3,4) & (v,4,4,1) & (v,4,4,2) & (u,4,4,3), \\ (v,4,4,4) & (4,t,4,2) & (4,s,4,3) & (4,u,4,4), & & & & \end{array}$$

where s denotes crop 1 or crop 2, t denotes crop 1 or crop 3, u denotes crops 1, 2 or 3 and v denotes all four crops.

4.2. An algorithm

Here, we present an algorithm that may be used to combine sequences as in Section 4.1. After the combinations are formed, a sequence $(i) \in S^*$ has m elements and each element is either a *single crop* or a *combined crop*. For each crop, the area for the current year must be known exactly. Therefore, for $(i) \in S^*$, we want i_m to be a single crop. The elements $i_1, \ldots, i_{m-1}$ may be combined crops. Our algorithm consists of an Initialisation Step followed by Step q, $q = 1, 2, \ldots, m-1$. In each Step q, we look for sequences (i) and (j) with identical sets of successors and which only differ in the qth element, i.e.

$$(i_1, \ldots, i_{q-1}) = (j_1, \ldots, j_{q-1}) = (r), \quad i_q \neq j_q, \quad (i_{q+1}, \ldots, i_m) = (j_{q+1}, \ldots, j_m) = (k) \tag{4.1}$$

for some (r) and (k). For fixed (r) and (k), there are at most n sequences satisfying this property. They are then combined into the new sequence $((r), y, (k))$, where y is the combined crop consisting of all qth (single) crops of the sequences that are combined. When no more sequences can be combined in this way, the new set of admissible sequences $S^{(q)}$ is determined. Step q ends by determining the new sets of successors $\mathrm{SUC}^{(q)}(i)$, $(i) \in S^{(q)}$. A detailed description of the Initialisation Step and a general Step q are given below.

Initialisation Step
Set $S^{(0)} = S$ and $\mathrm{SUC}^{(0)}(i) = \mathrm{SUC}(i)$, $(i) \in S^{(0)}$, and go to Step 1.

Step q ($q = 1, \ldots, m-1$)

- For $(r) = (r_1, \ldots, r_{q-1})$ and $(k) \in \{1, 2, \ldots, n\}^{m-q}$, define $C^{(q)}_{(r)(k)} \subset S^{(q-1)}$ by

  $C^{(q)}_{(r)(k)} = \{(i) \in S^{(q-1)} :$ there is a $(j) \in S^{(q-1)}$, such that $\mathrm{SUC}^{(q-1)}(j) = \mathrm{SUC}^{(q-1)}(i)$ and (4.1) is satisfied$\}$.
- If $C^{(q)}_{(r)(k)} = \emptyset$ for all (r) and (k), then set $S^* = S^{(q-1)}$ and $\mathrm{SUC}^*(i) = \mathrm{SUC}^{(q-1)}(i)$, $(i) \in S^*$, and STOP.
- For any $C^{(q)}_{(r)(k)} \neq \emptyset$, combine all sequences in $C^{(q)}_{(r)(k)}$ into the new sequence $((r), y, (k))$, where y is the combined crop consisting of all qth (single) crops of the sequences in $C^{(q)}_{(r)(k)}$.
- Define the new set of sequences $S^{(q)}$ by deleting sequences $(i) \in S^{(q-1)}$ which were combined above and adding the new formed combinations.
- Construct the sets $\mathrm{SUC}^{(q)}(i)$, $(i) \in S^{(q)}$, as follows. Consider a pair $(i), (j) \in S^{(q)}$.
  - (C1) If $(i), (j) \in S^{(q)} \cap S^{(q-1)}$, then $(j) \in \mathrm{SUC}^{(q)}(i)$ if and only if $(j) \in \mathrm{SUC}^{(q-1)}(i)$.
  - (C2) If $(i) \in S^{(q)} \cap S^{(q-1)}$ and $(j) \in S^{(q)} \setminus S^{(q-1)}$, (j) was formed from $C^{(q)}_{(r)(k)}$, then $(j) \in \mathrm{SUC}^{(q)}(i)$ if and only if there is a $(t) \in C^{(q)}_{(r)(k)}$ with $(t) \in \mathrm{SUC}^{(q-1)}(i)$.
  - (C3) If $(i) \in S^{(q)} \setminus S^{(q-1)}$, (i) was formed from $C^{(q)}_{(p)(l)}$, and $(j) \in S^{(q)} \cap S^{(q-1)}$, then $(j) \in \mathrm{SUC}^{(q)}(i)$ if and only if $(j) \in \mathrm{SUC}^{(q-1)}(s)$, for any (all) $(s) \in C^{(q)}_{(p)(l)}$.

(C4) If $(i) \in S^{(q)} \setminus S^{(q-1)}$, (i) was formed from $C^{(q)}_{(p)(l)}$, and $(j) \in S^{(q)} \setminus S^{(q-1)}$, (j) was formed from $C^{(q)}_{(r)(k)}$, then $(j) \in \mathrm{SUC}^{(q)}(i)$ if and only if there is a $(t) \in C^{(q)}_{(r)(k)}$ with $(t) \in \mathrm{SUC}^{(q-1)}(s)$, for any (all) $(s) \in C^{(q)}_{(p)(l)}$.

- If $q \leqslant m-2$, then go to Step $q+1$. Else, set $S^* = S^{(q)}$ and $\mathrm{SUC}^*(i) = \mathrm{SUC}^{(q)}(i)$, $(i) \in S^*$, and STOP.

The construction of the sets $\mathrm{SUC}^{(q)}(i)$, $(i) \in S^{(q)}$, may need some explanation. In (C1), both (i) and (j) are not combined in Step q and nothing changes. In (C2), sequence (i) is not combined, while (j) is formed in Step q. In this case, (j) is allowed to succeed (i) if and only if some sequence (t), which is combined into (j), is allowed to succeed (i). In Lemma 4.9, we will show that there is at most one such sequence (t). In (C3), sequence (i) is formed in Step q, while (j) is not combined. Here, (j) is allowed to succeed (i) if and only if (j) is allowed to succeed some sequence (s), which is combined into (i). Notice that if such a sequence (s) exists, then (j) is allowed to succeed all (s) which are combined into (i), since, for these sequences (s), the sets $\mathrm{SUC}^{(q-1)}(s)$ are identical. In (C4), both (i) and (j) are formed in Step q. From (C2) and (C3) it follows that (j) is allowed to succeed (i) if and only if some sequence (t), which is combined into (j), is allowed to succeed some sequence (s), which is combined into (i). As in (C2), for each (s), there is at most one such sequence (t).

In the algorithm, the order of the steps is Step 1, Step 2,..., Step $m-1$. In Step q combinations are formed of sequences in the sets $C^{(q)}_{(r)(k)}$. Since these sequences satisfy (4.1), there holds that for any $(j) \in S^{(q)}$ all elements in $(j_{q+1},\dots,j_m)$ are single crops.

It may seem that the order of the steps may be changed. However, this is not true. In Lemma 4.2 we show that if, after steps $1,\dots,q-1$, we execute Step p with $p \in \{q+1,\dots,m-1\}$, then no sequences will be combined.

The algorithm stops if no sequences are combined in the current step. This is correct, since in Lemma 4.3 it is shown that if in Step $q-1$ no sequences are combined, then also in Step q no sequences will be combined.

Lemma 4.2. *Suppose the steps $1,\dots,q-1$ have been executed in this order. If next* Step *p is executed, with $p \in \{q+1,\dots,m-1\}$, then no sequences will be combined.*

Proof. Suppose to the contrary that in Step p some sequences are combined, i.e. for some $(i),(j) \in S^{(q-1)}$ with

$$(i_1,\dots,i_{p-1}) = (j_1,\dots,j_{p-1}) = (r), \quad i_p \neq j_p, \quad (i_{p+1},\dots,i_m) = (j_{p+1},\dots,j_m) = (k) \tag{4.2}$$

for some (r) and (k), there holds $\mathrm{SUC}^{(q-1)}(i) = \mathrm{SUC}^{(q-1)}(j)$. For any sequence $(x) \in S^{(q-1)}$ the elements $x_q,\dots,x_m$ are single crops. Hence, for any $(t) \in \mathrm{SUC}^{(q-1)}(i)$, there holds

$$j_s = i_s = t_{s-1}, \quad s = q+1,\dots,m.$$

But, since $p \in \{q+1,\dots,m-1\}$, this implies $i_p = j_p$ which contradicts (4.2). Therefore, in Step p no sequences will be combined. □

Lemma 4.3. *If in* Step *$q-1$ no sequences are combined, then also in* Step *q no sequences will be combined.*

Proof. The proof is similar to the proof of Lemma 4.2. Suppose to the contrary that in Step q some sequences are combined, i.e. for some $(i),(j) \in S^{(q-1)}$ satisfying (4.1) for some (r) and (k), there holds $\mathrm{SUC}^{(q-1)}(i) = \mathrm{SUC}^{(q-1)}(j)$. Since no sequences are combined in Step $q-1$, for any $(x) \in S^{(q-1)}$ the elements $x_{q-1},\dots,x_m$ are single crops. Hence, for any $(t) \in \mathrm{SUC}^{(q-1)}(i)$, there holds

$$j_s = i_s = t_{s-1}, \quad s = q,\dots,m.$$

But this implies $i_q = j_q$ which contradicts (4.1). Therefore, in Step q no sequences will be combined. □

Remark 4.4. An analogous statement as in Lemma 4.3 does not hold for an individual $(i) \in S^{(q-1)}$, i.e. when (i) is not combined in Step $q-1$, it may still be combined in Step q if in Step $q-1$ some other sequences are combined. An example is as follows. Consider $n = 2$ crops and suppose that all crop succession information is given by the inadmissibility of $(2,2,2,2)$ and $(2,1,2,2)$. Then these sequences are also minimal inadmissible and $m = 3$. If we execute our algorithm, then $S^* = \{(x,1,1),(x,2,1),(1,1,2),(1,2,2),(2,x,2)\}$, where x denotes all crops. Hence, $(2,1,2)$ and $(2,2,2)$ are not combined in Step 1, while in Step 2 they are combined into $(2,x,2)$.

Remark 4.5. We have programmed the algorithm using the software package Matlab 5. Our computational experience with the program learns that on a Pentium 4 PC, the algorithm terminates within approximately 15 minutes for values of n and m with $n^m \leqslant 3200$ approximately. To our opinion, this covers most practical situations.

4.3. No loss of crop succession information

Here, we show that the information under (b), i.e. all admissible sequences of length $m+1$, is preserved when sequences are combined in the algorithm of Section 4.2. First, we introduce some notation. Let

$$B^{(q)} = \{(i_1,\ldots,i_{m+1}) : (i_1,\ldots,i_m) \in S^{(q)}, i_{m+1} \in \{1,2,\ldots,n\}, \text{ such that there is a } (j) \in \mathrm{SUC}^{(q)}(i_1,\ldots,i_m) \text{ with } j_m = i_{m+1}\}.$$

Then $B^{(q)}$ contains all admissible sequences of length $m+1$ (containing both combined and single crops as elements), based on $S^{(q)}$ and the sets $\mathrm{SUC}^{(q)}(i)$, $(i) \in S^{(q)}$. For any sequence (i) of length m, possibly containing combined crops, let

$$\mathrm{COMB}(i) = \{(s) \in \{1,2,\ldots,n\}^m : (s) \text{ is contained in } (i)\}.$$

For $(i) \in S^{(0)}$ we have $\mathrm{COMB}(i) = (i)$. From the definition of the algorithm in Section 4.2 it follows that $\mathrm{COMB}(i) \subseteq S^{(0)}$, $(i) \in S^{(q)}$. Moreover, for any $q \in \{0,1,\ldots,m-1\}$, the sets $\mathrm{COMB}(i)$, $(i) \in S^{(q)}$, constitute a disjoint partition of $S^{(0)}$.

To express the information in $B^{(q)}$ in terms of $S^{(0)}$ and $\mathrm{SUC}^{(0)}(i)$, $(i) \in S^{(0)}$, we define

$$B_0^{(q)} = \{(s_1,\ldots,s_{m+1}) \in \{1,2,\ldots,n\}^{m+1} : \text{ there is an } (i) \in B^{(q)} \text{ such that } (s_1,\ldots,s_m) \in \mathrm{COMB}(i_1,\ldots,i_m) \text{ and } i_{m+1} = s_{m+1}\}.$$

Notice that $B_0^{(0)} = B^{(0)}$ contains all admissible sequences of length $m+1$ (containing only single crops as elements), i.e. all information under (b). We have the following result.

Proposition 4.6. *Suppose that in* Step q *some sequences are combined. Then there holds*

$$B_0^{(q)} = B_0^{(0)}.$$

The proof of Proposition 4.6 is presented below. Proposition 4.6 shows that, by combining sequences in the way the algorithm prescribes, all information under (b) is preserved. As a consequence, the results of Theorem 2.5, Section 2.3, Proposition 3.2 and 3.7 are also valid for the decision variables $X_t(j)$, $(j) \in S^*$ and the sets $\mathrm{SUC}^*(i)$, $(i) \in S^*$. The analogue of Theorem 2.5 is as follows.

We consider sets $I, J \subset S^*$ satisfying

$$(j) \notin \mathrm{SUC}^*(i) \text{ for all } (i) \in I \text{ and all } (j) \in J. \tag{4.3}$$

Theorem 4.7. *Assume that* (2.3) *holds for* $t-1$ *and* t. *Then the cropping plans* $X_{t-1}(i)$, $(i) \in S^*$, *and* $X_t(j)$, $(j) \in S^*$, *are compatible if and only if*

$$\sum_{(i)\in I} X_{t-1}(i) + \sum_{(j)\in J} X_t(j) \leqslant A, \quad \forall I, J \subset S \text{ satisfying (4.3).} \tag{4.4}$$

For the whole planning period of T years the crop succession requirements are satisfied if and only if (4.4) holds for $t = 2, 3, \ldots, T$. Since the last crop of any sequence in S^* is a single crop, the knowledge of $X_t(j)$, $t = 1, 2, \ldots, T$, $(j) \in S^*$, implies that for each year $1, 2, \ldots, T$ the area for each crop $1, 2, \ldots, n$ is known exactly.

Before we prove Proposition 4.6, we need the following definition and lemma.

Definition 4.8. Let (i) and (j) be combined sequences of length m. We say that (j) is a *compatible successor* of (i) if there is an $(s) \in \mathrm{COMB}(i)$ and a $(t) \in \mathrm{COMB}(j)$ such that $t_1 = s_2, t_2 = s_3, \ldots, t_{m-1} = s_m$.

Notice that a sequence $(i) \in S^{(q)}$, $q \geqslant 1$, may have more than n compatible successors $(j) \in S^{(q)}$. An example can be found in Remark 4.4, where $n = 2$ and $(2, x, 2) \in S^*$ has three compatible successors: $(x, 2, 1)$, $(1, 2, 2)$ and $(2, x, 2)$.

Lemma 4.9. *Suppose* $(i) \in S^{(q-1)}$ *and* $C^{(q)}_{(r)(k)} \neq \emptyset$. *Then there is at most one* $(t) \in C^{(q)}_{(r)(k)}$ *that is a compatible successor of* (i).

Proof. The proof is similar to the proofs of Lemmas 4.2 and 4.3. Let $(s), (t) \in C^{(q)}_{(r)(k)}$, with (t) a compatible successor of (i). Suppose to the contrary that also (s) is a compatible successor of (i). For any sequence $(x) \in S^{(q-1)}$ the elements $x_q, \ldots, x_m$ are single crops. Hence, there holds

$$s_{k-1} = t_{k-1} = i_k, \quad k = q+1, \ldots, m.$$

But this implies $s_q = t_q$ which contradicts the fact that $(s), (t) \in C^{(q)}_{(r)(k)}$. Therefore, if $C^{(q)}_{(r)(k)} \neq \emptyset$, then, for any $(i) \in S^{(q-1)}$, there is at most one $(t) \in C^{(q)}_{(r)(k)}$ that is a compatible successor of (i). □

Proof of Proposition 4.6. We will show that, if in Step q some sequences are combined, then $B_0^{(q)} = B_0^{(q-1)}$. Since $q \in \{1, 2, \ldots, m-1\}$ is arbitrary, this completes the proof.

Consider an arbitrary sequence $(k_1, \ldots, k_{m+1}) \in \{1, 2, \ldots, n\}^{m+1}$ and suppose that

$$\begin{aligned}
&(k_1, \ldots, k_m) \in \mathrm{COMB}(s), \quad (s) \in S^{(q-1)},\\
&(k_2, \ldots, k_{m+1}) \in \mathrm{COMB}(t), \quad (t) \in S^{(q-1)},\\
&(k_1, \ldots, k_m) \in \mathrm{COMB}(i), \quad (i) \in S^{(q)},\\
&(k_2, \ldots, k_{m+1}) \in \mathrm{COMB}(j), \quad (j) \in S^{(q)}.
\end{aligned}$$

Notice that $(s), (t), (i)$ and (j) above are uniquely determined. If in Step q the sequence (i) is formed from some group of sequences, then this group contains (s). The same is true for (j) and (t). For (i) and (j) we distinguish the situations (C1)–(C4) of Section 4.2.

(C1) $(i) = (s)$ and $(j) = (t)$.
(C2) $(i) = (s)$ and (j) is formed from a group containing (t).
(C3) (i) is formed from a group containing (s) and $(j) = (t)$.
(C4) (i) is formed from a group containing (s) and (j) is formed from a group containing (t).

Suppose $(k_1,\dots,k_{m+1}) \in B_0^{(q-1)}$. Then it follows that $(s_1,\dots,s_m,t_m) \in B^{(q-1)}$ and, hence, that $(t) \in \mathrm{SUC}^{(q-1)}(s)$. To obtain that $(k_1,\dots,k_{m+1}) \in B_0^{(q)}$, we need to show that $(j) \in \mathrm{SUC}^{(q)}(i)$. But, since $(t) \in \mathrm{SUC}^{(q-1)}(s)$, this follows for each of (C1)–(C4) from the definition of the algorithm in Section 4.2.

It remains to show that if $(k_1,\dots,k_{m+1}) \notin B_0^{(q-1)}$, then $(k_1,\dots,k_{m+1}) \notin B_0^{(q)}$. When $(k_1,\dots,k_m) \notin S^{(0)}$, this holds trivially. We will therefore assume that $(k_1,\dots,k_m) \in S^{(0)}$. Suppose that $(k_2,\dots,k_{m+1}) \notin S^{(0)}$. Let (i) be as above. If $(k_1,\dots,k_{m+1}) \in B_0^{(q)}$, then $(i_1,\dots,i_m,k_{m+1}) \in B^{(q)}$ and, hence, there exists a $(j) \in S^{(q)}$ with $(j) \in \mathrm{SUC}^{(q)}(i)$ and $j_m = k_{m+1}$. But this implies $(k_2,\dots,k_{m+1}) \in \mathrm{COMB}(j)$ which contradicts $(k_2,\dots,k_{m+1}) \notin S^{(0)}$. Therefore, if $(k_2,\dots,k_{m+1}) \notin S^{(0)}$, then $(k_1,\dots,k_{m+1}) \notin B_0^{(q)}$.

Next, we give the proof for the case $(k_2,\dots,k_{m+1}) \in S^{(0)}$. Let $(s),(t),(i)$ and (j) be as above. Since $(k_1,\dots,k_{m+1}) \notin B_0^{(q-1)}$, this implies that $(s_1,\dots,s_m,t_m) \notin B^{(q-1)}$ and, hence, that $(t) \notin \mathrm{SUC}^{(q-1)}(s)$. However, since $(k_1,\dots,k_m) \in \mathrm{COMB}(s)$ and $(k_2,\dots,k_{m+1}) \in \mathrm{COMB}(t)$, (t) is a compatible successor of (s). Analogously, (j) is a compatible successor of (i). The proof will be complete if we show that $(j) \notin \mathrm{SUC}^{(q)}(i)$. Again, we distinguish the cases (C1)–(C4) above.

For (C1) there is nothing to prove. For (C2), it follows from Lemma 4.9 that (t) is the only compatible successor of (s) in the group of sequences from which (j) is formed. Hence, from the defnition of the algorithm in Section 4.2 it follows that $(t) \notin \mathrm{SUC}^{(q-1)}(s)$ implies $(j) \notin \mathrm{SUC}^{(q)}(i)$. Next, we consider (C3). The sequences in the group from which (i) is formed all have the same set $\mathrm{SUC}^{(q-1)}$ as (s). Therefore, $(j) = (t) \notin \mathrm{SUC}^{(q-1)}(s)$ implies $(j) \notin \mathrm{SUC}^{(q)}(i)$. Combining the arguments for (C2) and (C3), it can be seen that also for (C4) there holds $(j) \notin \mathrm{SUC}^{(q)}(i)$. This completes the proof.

5. Conclusion

In the paper we have discussed crop succession requirements from a mathematical programming point of view. We assumed that all crop succession information is given in the form of crop sequences which are not allowed (i.e. not advisable) to be cultivated on the same piece of land. These sequences are called inadmissible. The inadmissibility of a sequence is assumed not to depend on the year it was started. A key role is played by so-called minimal inadmissible sequences, i.e inadmissible sequences not containing inadmissible subsequences. If $m+1$ is the length of the longest minimal inadmissible sequence, then we need to consider the cropping plans of the years $t-m,\dots,t-1$ in order to determine the feasible cropping plans for year t. In Appendix A a method was presented to determine the minimal inadmissible sequences from an arbitrary collection of inadmissible sequences.

We defined decision variables representing the size of the area where a certain admissible sequence of m crops is applied, the last crop being grown in year t. In Section 2.2 we showed that, for a piece of land of fixed size, the crop succession requirements can be written as linear constraints in the decision variables. The number of admissible sequences of length m (and, hence, the number of decision variables) may be reduced, without losing crop succession information, by forming suitable combinations of sequences. An algorithm which determines such combinations of sequences was presented in Section 4, together with a proof of its correctness.

In Section 3 we considered a generic LP-model for agricultural production planning with a finite planning horizon of T years, containing the linear crop succession constraints. We considered a stationary version of the model in which the parameters do not depend on the year index and the cropping plan for year $T+1$ is also taken into account and is required to be identical to the cropping plan of year 1. We showed that any optimal solution of this model has a stationary version with the same objective value. This implies that, when using this model, the optimal production strategy may be chosen the same zin all years of the planning period. In Section 3.2 we presented a method to determine the pieces of land where each crop should be cultivated each year, when such a stationary cropping plan is to be applied.

Acknowledgements

We would like to thank Caspar Schweigman for drawing our attention to the subject of crop succession requirements in agricultural production planning and for various discussions on the subject. Also, we thank two anonymous referees for valuable suggestions which have considerably improved the presentation of the paper.

Appendix A. Determining minimal inadmissible sequences

The notion of a minimal inadmissible sequence plays an important role in the analysis above. The length of the longest minimal inadmissible sequence indicates how many years we should look into the past when determining feasible cropping plans for next year. Above, we implicitly assumed that all minimal inadmissible sequences were known. In practice, however, it may happen that only a collection of inadmissible sequences is given. Here, we discuss how the minimal inadmissible sequences may be determined from such a collection, under the assumption that the given collection of inadmissible sequences represents all crop succession information. The following example shows that, in general, this may not be a trivial task. Recall that inadmissible sequences are defined by conditions (I1)–(I3) in Section 2.1.

Example A.1. Suppose we consider $n = 2$ crops and all crop succession information is represented by the inadmissibility of the sequences $(1, 1, 1, 1)$ and $(2, 1, 1)$. Then there is only one minimal inadmissible sequence, namely $(1, 1)$. Indeed, suppose we have cultivated $(1, 1)$ in years $t-1$ and t. Then we have grown crop 1 in year $t-2$, since $(2, 1, 1)$ is inadmissible. And also in year $t-3$ we must have grown crop 1. Otherwise we have a sequence $(2, 1, 1)$ for the years $t-3, t-2, t-1$. But now we have cultivated crop 1 for four successive years, which is also inadmissible. Therefore, $(1, 1)$ is inadmissible. Since $(1, 1, 1, 1)$ and $(2, 1, 1)$ both contain $(1, 1)$, they are not minimal inadmissible. The sequence $(1, 1)$, however, is minimal inadmissible, since the cultivation of crop 1 is not entirely prohibited.

The example above can still be worked out by hand. However, to determine the minimal inadmissible sequences for (R1)–(R6) in Example 4.1 is more difficult. In this case, it is more convenient to have a computer program which determines all minimal inadmissible sequences from a given collection of inadmissible sequences. Below, we present such an algorithm. We assume that an initial collection of inadmissible sequences is given by the sets M_h, $h \geqslant 1$, where

$$M_h = \{(i) \in \{1, 2, \ldots, n\}^h : \ (i) \text{ is inadmissible}\}.$$

Suppose the longest inadmissible sequence has length $H + 1$, i.e.

$$H + 1 = \max\{h : \ M_h \neq \emptyset\}. \tag{A.1}$$

As above, we assume that $H \geqslant 1$. If the collection $\cup_{h=1}^{H+1} M_h$ represents all crop succession information, then its set of minimal inadmissible sequences is unique and the longest minimal inadmissible sequence has a length of at most $H + 1$.

Our algorithm consists of three steps: Step M1, Step M2 and Step M3. Before we determine the minimal inadmissible sequences, we first determine all inadmissible sequences based on the sets M_h, $h = 1, 2, \ldots, H + 1$. This is done in Step M1 and Step M2.

Step M1: For $h = 1, 2, \ldots, H$, do the following.

1.1 For each $(i) \in M_h$, for each $g \in \{h + 1, \ldots, H + 1\}$, if there is a $(j) \in \{1, 2, \ldots, n\}^g$ with $(j) \notin M_g$ and (i) is a subsequence of (j), then add (j) to M_g.

In Step m1 we enlarge the collection of inadmissible sequences by including all sequences which contain an inadmissible subsequence in some set M_h. However, after Step m1 the collection of inadmissible sequences does not yet contain all inadmissible sequences that can be found, using the sets M_h. We also need to include sequences (not contained in any M_h) of which the inadmissibility follows from the sets M_h by logical reasoning. An example is the sequence $(1,1)$ in Example A.1. Such sequences are identified by making use of similar sets as SUC(i) in (2.4). For $(i) \in \{1,2,\ldots,n\}^h$, we define

$$\mathrm{SUC}_h(i) = \{(j) \in \{1,2,\ldots,n\}^h : (j) \notin M_h, (j_1,\ldots,j_{h-1}) = (i_2,\ldots,i_h) \text{ and } ((i),j_h) \notin M_{h+1}\}.$$

Hence, $(j) \in \mathrm{SUC}_h(i)$ if and only if (j) is allowed to succeed (i) according to M_h and M_{h+1}. Notice that if $\mathrm{SUC}_h(i) = \emptyset$, then (i) is inadmissible. Also, if $(i) \in \{1,2,\ldots,n\}^h$ is not contained in any set $\mathrm{SUC}_h(j)$, then (i) is inadmissible.

Step m2: For $h = H, H-1, \ldots, 1$, do the following.

2.1 For all sequences $(i) \in \{1,2,\ldots,n\}^h$ with $(i) \notin M_h$, determine the sets $\mathrm{SUC}_h(i)$ based on the information in M_{h+1}.

2.2 Let $D_h = \{(i) \in \{1,2,\ldots,n\}^h \setminus M_h : \mathrm{SUC}_h(i) = \emptyset \text{ or } (i) \notin \mathrm{SUC}_h(j),\ \forall (j) \notin M_h\}$. If $D_h \neq \emptyset$, then go to Step 2.3, else NEXT h.

2.3 Add all $(i) \in D_h$ to M_h. For each $(i) \in D_h$, for each $g \in \{h+1,\ldots,H+1\}$, if there is a $(j) \in \{1,2,\ldots,n\}^g$ with $(j) \notin M_g$ and (i) is a subsequence of (j), then add (j) to M_g. Start again at the beginning of Step m2.

If in Step 2.2 some inadmissible sequences are found, they are added to M_h. Moreover, all sequences containing these sequences are inadmissible too and are added to M_g, $g \geqslant h+1$. This may change the sets $\mathrm{SUC}_g(i)$, $g \geqslant h$. Therefore, when inadmissible sequences are found, we start again at the beginning of Step m2. It can be seen that after Step m2 the sets $M_1, M_2, \ldots, M_{H+1}$ contain all inadmissible sequences that can be found based on the initial collection of inadmissible sequences. However, the addition in Step 2.3 of sequences to M_g, $g \in \{h+1,\ldots,H\}$, is not necessary, since at this point they are already contained in the sets M_g, $g \in \{h+1,\ldots,H\}$. Moreover, also the addition of sequences to M_{H+1} is not necessary. In Proposition A.3 it is shown that we may replace Step 2.3 by

2.3* Add all $(i) \in D_h$ to M_h and go to Step 2.1.

In Step m3 we determine the minimal inadmissible sequences by deleting inadmissible sequences that contain an inadmissible subsequence.

Step m3: For $h = 1, 2, \ldots, H$, do the following.

3.1 For each $(i) \in M_h$, for each $g \in \{h+1,\ldots,H+1\}$, if there is a $(j) \in M_g$ that contains (i) as a subsequence, then delete (j) from M_g.

In this way, we end up with all minimal inadmissible sequences in the sets $M_1, M_2, \ldots, M_{H+1}$. There holds

$$m + 1 = \max\{h : M_h \neq \emptyset\}.$$

Notice that m may be smaller than H, as can be seen from Example A.1.

There are several ways in which the algorithm above can be made faster. A first observation is that the sequences which are added to the sets M_h in Step m1 and Step 2.3 (for $g \geqslant h+1$) are not minimal inadmissible, since they contain an inadmissible subsequence. Hence, in Step m3, these sequences may be deleted without examination. Another change by which the procedure may become faster, is by starting with Step m3, followed by a recalculation of H in (A.1). Since Step m3 excludes sequences from the initial collection which are not minimal inadmissible, after Step M3 the value of H may have decreased. As a result, the algorithm consisting of Step M3, Step M1, Step M2 and again Step M3, has more steps but may be faster.

Remark A.2. We have programmed a version of the algorithm above using the software package Matlab 5. We start with Step M3, followed by Step M1, Step M2 and again Step M3, and use Step 2.3* instead of Step 2.3. Our computational experience with the program learns that on a Pentium 4 PC, the algorithm terminates within approximately 20 minutes for values of n and H with $n^H \leqslant 3400$ approximately. To our opinion, this covers most practical situations.

Proposition A.3. *Suppose an initial collection M_h, $h \geqslant 1$, is given and* Step M1 *has been executed. Suppose that in* Step 2.2 *$D_h \neq \emptyset$ for some h. Then there holds for any $(i) \in D_h$*

(F1) *If $h \leqslant H-1$, then after* Step M2 *has been executed for $h+1$, the sets M_g, $g \in \{h+1,\dots,H\}$, contain all sequences having (i) as a subsequence.*
(F2) *In* Step 2.3 *it is not necessary to add sequences to M_{H+1}, containing (i) as a subsequence.*

Proof. First, we prove (F1). Suppose Step M1 has been executed. Observe that there holds for $h = 1,2,\dots,H$,

$$(s) \in M_h \Rightarrow \text{ all } (t) \in \{1,2,\dots,n\}^g \text{ containing } (s) \text{ are in } M_g,\ g = h+1,\dots,H+1. \tag{A.2}$$

Suppose that $D_h \neq \emptyset$ in Step 2.2 for some $h \in \{1,2,\dots,H-1\}$. Let $(i) \in D_h$. Suppose that $\mathrm{SUC}_h(i) = \emptyset$. Then after Step M2 has been executed for $h+1$, there holds for all $x \in \{1,2,\dots,n\}$,

(E1) If $(i_2,\dots,i_h,x) \notin M_h$, then $((i),x) \in M_{h+1}$,
(E2) If $((i),x) \notin M_{h+1}$, then $(i_2,\dots,i_h,x) \in M_h$.

Since the set M_h is still the same as after Step M1 and no sequences have been deleted from M_{h+1}, it follows from (A.2) that (E2) is impossible. Therefore, if $\mathrm{SUC}_h(i) = \emptyset$, then after Step M1

$$((i),x)) \in M_{h+1}, \quad \forall x \in \{1,2,\dots,n\}. \tag{A.3}$$

Notice that this implies that any sequence $(y,(i))$, $y \in \{1,2,\dots,n\}$, not contained in M_{h+1} has an empty set $\mathrm{SUC}_{h+1}(y,(i))$. However, such sequences $(y,(i))$ do not exist, since they were added to M_{h+1} when Step M2 was executed for $h+1$. Therefore, after Step M2 has been executed for $h+1$, there holds

$$(y,(i)) \in M_{h+1}, \quad \forall y \in \{1,2,\dots,n\}. \tag{A.4}$$

Hence, it follows from (A.3) and (A.4) that after step M2 has been executed for $h+1$, all sequences containing (i) are contained in M_{h+1}. Since (A.3) holds after Step M1, it follows from (A.2) that after Step M1 all sequences of the form $(y,(i),x)$ and $((i),x,y)$, $x,y \in \{1,2,\dots,n\}$, are contained in M_{h+2}. But this implies that sequences of the form $(x,y,(i))$, $x,y \in \{1,2,\dots,n\}$, have an empty set $\mathrm{SUC}_{h+2}(x,y,(i))$. Hence, all sequences of the form $(x,y,(i))$ were added to M_{h+2} when Step M2 was executed for $h+2$. Therefore, after Step M2 has been executed for $h+1$, all sequences containing (i) are contained in both M_{h+1} and M_{h+2}. Analogously, it can be shown that after Step M2 has been executed for $h+1$, all sequences containing (i) are contained in M_g for $g = h+1,\dots,H$.

Suppose next that $(i) \in D_h$ due to $(i) \notin \mathrm{SUC}_h(j)$ for all $(j) \notin M_h$. Then after Step M2 has been executed for $h+1$, there holds for all $x \in \{1,2,\dots,n\}$,

(E3) If $(x,i_1,\dots,i_{h-1}) \notin M_h$, then $(x,(i)) \in M_{h+1}$,
(E4) If $(x,(i)) \notin M_{h+1}$, then $(x,i_1,\dots,i_{h-1}) \in M_h$.

Since the set M_h is still the same as after Step M1 and no sequences have been deleted from M_{h+1}, it follows from (A.2) that (E4) is impossible. Therefore, if $(i) \notin \text{SUC}_h(j)$ for all $(j) \notin M_h$, then after Step M1,

$$(x,(i))) \in M_{h+1}, \quad \forall x \in \{1,2,\ldots,n\}. \tag{A.5}$$

Notice that this implies that any sequence $((i),y)$, $y \in \{1,2,\ldots,n\}$, not contained in M_{h+1} is not contained in any set SUC_{h+1}. However, such sequences $((i),y)$ do not exist, since they were added to M_{h+1} when Step M2 was executed for $h+1$. Therefore, after Step M2 has been executed for $h+1$, there holds

$$((i),y) \in M_{h+1}, \quad \forall y \in \{1,2,\ldots,n\}. \tag{A.6}$$

Hence, it follows from (A.5) and (A.6) that after Step M2 has been executed for $h+1$, all sequences containing (i) are contained in M_{h+1}. Since (A.5) holds after Step M1, it follows from (A.2) that after Step M1 all sequences of the form $(y,x,(i))$ and $(x,(i),y)$, $x,y \in \{1,2,\ldots,n\}$, are contained in M_{h+2}. But this implies that sequences of the form $((i),x,y)$, $x,y \in \{1,2,\ldots,n\}$, are not contained in any set SUC_{h+2}. Hence, all sequences of the form $((i),x,y)$ were added to M_{h+2} when Step M2 was executed for $h+2$. Therefore, after Step M2 has been executed for $h+1$, all sequences containing (i) are contained in both M_{h+1} and M_{h+2}. Analogously, it can be shown that after Step M2 has been executed for $h+1$, all sequences containing (i) are contained in M_g for $g = h+1,\ldots,H$.

Notice that when h is the first (i.e. largest) h for which $D_h \neq \emptyset$ in Step 2.2, an analogous proof as above shows that after Step M1 all sequences containing an $(i) \in D_h$ are contained in the sets M_g, $g = h+1,\ldots,H$. This completes the proof of (F1).

Next, we prove (F2). First, we consider the case $h = H$. Let $(i) \in D_H$. In Step 2.3 sequences containing (i) as a subsequence are added to M_{H+1}. As a result, the sets SUC_H may change. However, the same changes in SUC_H are obtained when (i) is added to M_H. This can be seen as follows. The sequences added to M_{H+1} are of the form $(x,(i))$ or $((i),x)$, for $x \in \{1,2,\ldots,n\}$. Therefore, as a result, only the sets $\text{SUC}_H(j)$ may change for which either (j) is a compatible successor of (i), i.e. $(j_1,\ldots,j_{H-1}) = (i_2,\ldots,i_H)$, or (i) is a compatible successor of (j), i.e. $(i_1,\ldots,i_{H-1}) = (j_2,\ldots,j_H)$. But these sets $\text{SUC}_H(j)$ change in the same way when (i) is added to M_H. This completes the proof for the case $h = H$.

Next, we consider the case $h \leqslant H-1$. Let $(i) \in D_h$. From the proof of (F1) it follows that all sequences (k) of length H containing (i) are contained in M_H after Step M2 has been executed for H. Sequences of length $H+1$ containing (i) are of the form $(x,(k))$ or $((k),x)$, for $x \in \{1,2,\ldots,n\}$, where (k) has length H and contains (i). From the proof of (F2) for $h = H$ it follows that adding such sequences to M_{H+1} will not result in more inadmissible sequences of length H. This completes the proof of (F2). □

Proposition A.3 shows that Step 2.3 may be replaced by Step 2.3*. One may wonder whether it is necessary to update the sets SUC_h after the sequences in D_h have been added to M_h in Step 2.3*. In other words, may we replace Step 2.3* by

Add all $(i) \in D_h$ to M_h. NEXT h.

The following example shows that this is not the case.

Example A.4. Consider $n = 3$ crops and start with $M_4 = \{(2,1,3,2),(2,1,3,1)\}$ and $M_3 = \{(3,3,1),(3,3,2),(3,3,3)\}$. Then $H = 3$ and after Step M1 we have

$$M_4 = \{(2,1,3,2),(2,1,3,1),(3,3,s,t),(u,3,3,v), s,t,u,v = 1,2,3\}.$$

Next, we execute Step M2. In Step 2.2 we have $\text{SUC}_3(1,3,3) = \text{SUC}_3(2,3,3) = \emptyset$. Hence, $(1,3,3)$ and $(2,3,3)$ are added to M_3. When we go to Step 2.1 and change the sets SUC_3, we find that $\text{SUC}_3(2,1,3) = \emptyset$. Therefore, also $(2,1,3)$ is added to M_3. For $h = 2$, it is found that $\text{SUC}_2(3,3) = \emptyset$ and $(3,3)$ is added to M_2. After Step M3, we obtain $M_4 = \emptyset$, $M_3 = \{(2,1,3)\}$ and $M_2 = \{(3,3)\}$.

References

[1] L.R. Ford Jr., D.R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, NJ, 1962.
[2] W.K. Klein Haneveld, Crop rotation and linear programming, Technical Paper, Department of Econometrics, University of Groningen, 1983.
[3] A. Maatman, C. Schweigman, A. Ruys, Synopsis of a linear programming study of farmers' strategies on the Central Plateau in Burkina Faso, Centre for Development Studies Research Report, University of Groningen, 1996.
[4] A. Maatman, Si le fleuve se tord, que le crocodile se tord, Une Analyse des Systèmes Agraires de la Région Nord-Ouest du Burkina Faso à l'Aide des Modèles de Programmation Mathématique, Ph.D. Thesis, Faculty of Economics, University of Groningen, 2000.
[5] C. Schweigman, Operations Research Problems in Agriculture in Developing Countries, Khartoum University Press, Khartoum, Sudan, 1985.