Production Planning and Interval Goal Programming. Modelling feasibility.[1]

B. Vitoriano[2]
Instituto de Investigación Tecnológica, Universidad Pontificia Comillas, Alberto Aguilera 23, 28015 Madrid, Spain (bvitoriano@upco.es) Fax: +34 91 542 31 76

M. T. Ortuño
Facultad de Matemáticas, Universidad Complutense de Madrid, Avda. Complutense s/n, 28040 Madrid, Spain

[1]Research partially supported by the grant BFM2002-0281 from the MCYT, Spain
[2]Corresponding author

Production Planning and Interval Goal Programming. Modelling feasibility.

Abstract

Production Planning entails many decision making processes aimed at improving productivity and achieving the maximum profit from scarce resources. Assuming that there is a set of tasks to be carried out within a given time horizon, and each task can be performed in different ways depending on the resources used to develop it, the problem addressed in this work consists of determining how and when to carry out each task, in such a way that the tasks are scheduled in sequence at the minimum cost, taking into account any precedence relationships among them, the time window constraints for performing the tasks and the resources availability.
This paper focuses on developing a scheduling model which avoids infeasibilities. To achieve this, a family of *incompatibility conditions* is introduced to avoid a certain type of simultaneous usage of resources, and interval goal programming is used to model time windows in a more flexible way. The model has been applied to agricultural production planning, showing the advantages of this approach.

Keywords:Goal Programming, Production Planning, Integer Programming

1. Introduction

Production planning is the activity of establishing production goals over some future time period, called the *planning horizon*. The objective of production planning is to plan the optimal use of resources to meet stated production requirements.

Production Planning entails many decision making processes aimed at improving productivity and achieving the maximum profit from scarce resources. In this work a special class of planning problem is addressed. Consider a set of tasks to be carried out within a given time horizon, and consider that each task can be performed in several ways depending on the resources used to develop it. It would be the case, for example, of an activity which can be developed manually, semi-manually or using machinery of diverse kinds. The problem addressed in this work consists of determining *how* and *when* to carry out each task, in such a way that the tasks are scheduled in sequence at the minimum cost, taking into account any precedence relationships among them, the existence of time windows and the resources availability.

This paper focuses on developing a model enabling production planning in a medium/ long-term planning horizon. The proposed model concentrates on determining the minimum requirements in terms of machinery, labour, etc. for viable operation. This implies the need of finding an implementable time planning of the activities which use these resources. There are then two types of decisions to be made, namely, strategic decisions concerning *which* resources to acquire (via purchasing, hiring, etc.) and operational decisions about *when* and *how* to use them. Due to medium/long-term decisions to be made, any randomness inherent to duration of tasks is not considered explicitly, but included in a general way when the time windows are defined. Thus, special care will be taken in the definition and treatment of time windows along the paper. See

in Darby–Dowman et al. [4], for example, an stochastic approach for a more general kind of planning problems.

2. Problem Description

Let us consider a general production process in which several activities have to be carried out within a given time horizon, and consider that each one can be performed in several ways depending on the resources used to develop it. The following elements will be taken into account:

- **Tasks:** a task is each of the activities to be developed. Each task has an ideal time window assigned to it, and developing it out of this window either causes infeasibility or it is penalized by production loss. Before an specific task can begin being developed, other tasks must have been completed, and so relations of precedence are established among tasks. Moreover, depending on what resources are available, various modes may be chosen to perform each of the tasks.
- **Modes:** they are different ways of performing tasks. Since they are characterized by the resources needed to develop the task in that way, the estimated duration and costs of the task will be different depending on the mode chosen, as a result of the use of those specific resources.
- **Resources:** the different elements needed to carry out the tasks. They can be material or human, and the resources may or may not be present in the facility, the possibility of obtaining them by purchase, rental etc. being taken into account.

  Each of the resources has an associated variable cost and fixed cost. The variable cost derives from use, and it is calculated in terms of the time during which it will be used, the task to be done, wear and tear etc. The cost of a mode for developing a task includes the variable cost of the resources assigned to that mode.

  The fixed cost is additional to the cost arising from use, and does not depend on how long or how many times it is used. This cost may include storage costs, insurance etc., or, if the resource is not actually present, the cost of purchasing, renting etc.

The problem must be modelled in mathematical terms in order to meet the following objectives:

- To minimize the variable and fixed costs associated with each of the resources.
- To build the flexibility associated with the way the tasks are performed into the model, taking into account whether the resources can feasibly be assigned to the various tasks.
- To build the flexibility associated with the time windows, penalizing production loss.

The tasks to be done, the precedence relationships among them, the possible resources and modes for carrying out the tasks, and their cost and duration can all be regarded as fully specified.

Thus the problem addressed is to construct a model which selects the mode each task is to be carried out and the starting time for each task, in order to minimize costs, in accordance with task-precedence relationships, and taking time windows into account. Special care will be taken in order to decide whether the violation of a time window condition yields to problem infeasibility or to production loss, in which case this loss should be minimized. Alongside this, the optimum sizing of the facility's resources is to be decided.

There are then two types of infeasibilities the model should take care of:

1. *Strong infeasibility.* The model *must* prevent a resource to be used simultaneously in two different tasks.

2. *Weak infeasibility.* A task *should* be done inside its time window. If it is not so, it may cause a loss of productivity, which should be minimized. If the infeasibility is greater than a certain parameter, then it will become strong, and the solution should not be accepted.

This paper focuses on developing a scheduling model which treats infeasibilities efficiently. To achieve this, a family of *incompatibility conditions* will be introduced to avoid strong infeasibility, and interval goal programming will be used to deal with weak infeasibility and model time windows in a more flexible way.

3. Elements of the model

3.1. Data Representation

Parameters:

The parameters which define the size of the problem are the following:

n,	number of tasks,
nr,	number of resources subject to planning,
m_i,	number of modes associated with task i.

Index sets:

i,	task, $i = 1, \ldots, n$,
j,	mode associated with task $i, j = 1, \ldots, m_i$,
k,	resource subject to planning, $k = 1, \ldots, nr$.

Problem data:

d_{ij}, length of task i when done following mode j,

c_{ij}, cost of task i when done following mode j.
Note: c_{ij} includes variable cost of use of resources.

K_{ij}, set of resources assigned to task i when done following mode j.
Let $a_{ijk} = \begin{cases} 1, & \text{if } k \in K_{ij}, \\ 0, & \text{otherwise,} \end{cases}$

J_i^k, set of modes assigned to task i that need resource k ($j \in J_i^k \iff k \in K_{ij}$)

cf_k, fixed cost of resource k.

$G = (N, P)$, Directed graph of direct precedences
-N, vertices = set of tasks.
-P, arcs (i_1, i_2), such that task i_2 cannot begin before task i_1 is finished.

Additional Problem Data:

Time Windows of task i: For each task, an ideal time window $[a_{i0}, b_{i0}]$ with no related cost and a general time window $[a_{ih_i}, b_{ih_i}]$ is defined. Outside the ideal time window a penalty function is defined as follows (see figure 1):

Start time	*Marginal penalty*	*Final time*	*Marginal penalty*
$a_{i1} - a_{i0}$	α_{i1}	$b_{i0} - b_{i1}$	β_{i1}
$a_{i2} - a_{i1}$	α_{i2}	$b_{i1} - b_{i2}$	β_{i2}
$\cdots$	$\cdots$	$\cdots$	$\cdots$
$a_{ih_i} - a_{i,h_i-1}$	α_{ih_i}	$b_{i,h_i-1} - b_{ih_i}$	β_{ih_i}
Below a_{ih_i}	∞	Over b_{ih_i}	∞

($a_{ih_i} \leq \cdots \leq a_{i1} \leq a_{i0} \leq b_{i0} \leq b_{i1} \leq \cdots \leq b_{ih_i}$, $\alpha_{ih_i} \geq \cdots \geq \alpha_{i2} \geq \alpha_{i1}$ and $\beta_{i1} \leq \beta_{i2} \leq \cdots \leq \beta_{ih_i}$)

$$c_i^- = \begin{cases} 0 & T_i \geq a_0 \\ \alpha_1(a_0 - T_i) & a_1 \leq T_i < a_0 \\ \alpha_1(a_0 - T_i) + (\alpha_2 - \alpha_1)(a_1 - T_i) & a_2 \leq T_i < a_1 \\ \alpha_1(a_0 - T_i) + (\alpha_2 - \alpha_1)(a_1 - T_i) + (\alpha_3 - \alpha_2)(a_2 - T_i) & a_3 \leq T_i < a_2 \\ \vdots & \vdots \end{cases}$$

$$c_i^+ = \begin{cases} 0 & TF_i < b_0 \\ \beta_1(TF_i - b_0) & b_0 \leq TF_i < b_1 \\ \beta_1(TF_i - b_0) + (\beta_2 - \beta_1)(TF_i - b_1) & b_1 \leq TF_i < b_2 \\ \beta_1(TF_i - b_0) + (\beta_2 - \beta_1)(TF_i - b_1) + (\beta_3 - \beta_2)(TF_i - b_2) & b_2 \leq TF_i < b_3 \\ \vdots & \vdots \end{cases}$$

where T_i and TF_i are the start and finish time for task i, respectively. This formulation is a suitable approach for most of the convex penalty functions, even if nonlinear penalties are considered.

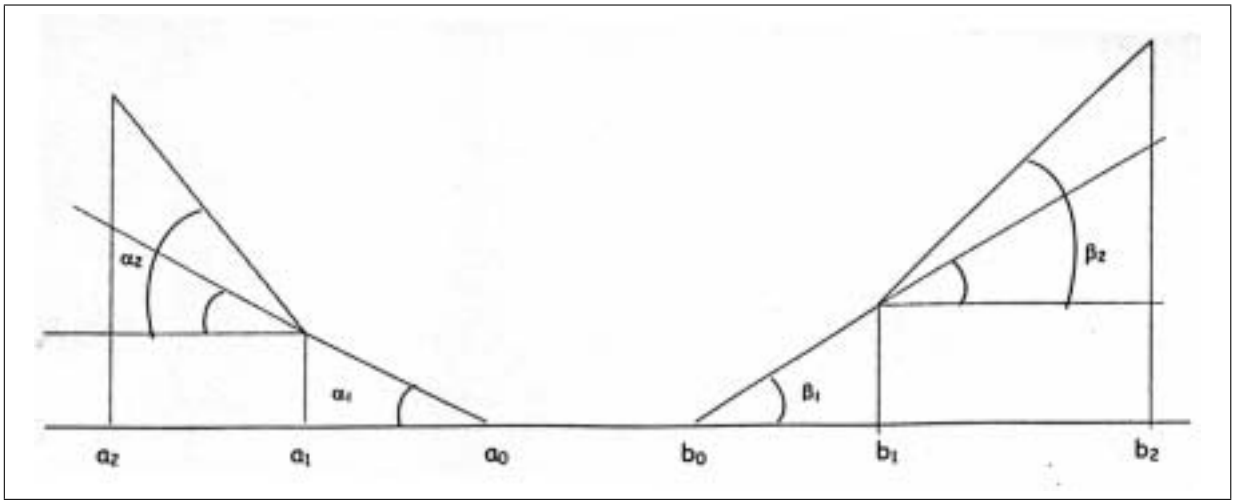


Figure 1: marginal penalties for a generic task

3.2. Variables

T_i,	instant in which task i is started.	$\forall i = 1, \ldots, n$
n_{il}	unwanted lower deviational variables in time windows.	$\forall i = 1, \ldots, n, \quad l = 1, \ldots, h_i$
n'_{il}	lower deviational variables in time windows.	$\forall i = 1, \ldots, n, \quad l = 1, \ldots, h_i$
p_{il}	unwanted upper deviational variables in time windows.	$\forall i = 1, \ldots, n, \quad l = 1, \ldots, h_i$
p'_{il}	upper deviational variables in time windows.	$\forall i = 1, \ldots, n, \quad l = 1, \ldots, h_i$
$x_{ij} = \begin{cases} 1, \\ 0, \end{cases}$	if task i is performed under task mode j, otherwise,	$\forall i = 1, \ldots, n, \quad j = 1, \ldots, m_i,$
$y_k = \begin{cases} 1, \\ 0, \end{cases}$	if resource k is used, otherwise,	$\forall k = 1, \ldots, nr.$
$\delta_{i_1 i_2} = \begin{cases} 1, \\ 0, \end{cases}$	if task i_2 precedes task i_1, otherwise,	$\forall i_1, i_2$ incompatible tasks*.

* Note: See in section 4 conditions for two tasks to be incompatible.

3.3. Objective functions

The main objective of our model is to achieve the desired planning at minimum cost. This cost to be minimized is given by the sum of variable costs related to the execution of each task

plus fixed costs incurred when using each of the resources. This objective is modeled as follows

$$\text{mín} \sum_{i=1}^{n} \sum_{j=1}^{m_i} c_{ij} x_{ij} + \sum_{k=1}^{nr} c f_k y_k$$

This will be the first objective of our model.

A second objective will be minimizing production loss. To do so, the model should try to achieve each task inside its ideal time window, or, otherwise, minimize the deviation of this window.

Charnes and Collomb [1] and Ignizio [5] introduced Goal Interval Programming (GIP). Later, this approach was developed by approximating the piecewise linear function that represents the penalty scale as a continuous function (see Charnes et al. [2] and Charnes and Cooper [3]). Kvanli [6] suggested an extension of GIP based upon penalty functions. Such an extension attempts to minimise the weighted sum of unwanted deviational variables from a specified set of intervals. Several authors have proposed technical improvements and refinements of Kvanli's aproach. See in Vitoriano and Romero [9] a new GP variant. Following the Jones and Tamiz [7] formulation, we model penalties of time windows in the following way

$$\sum_{i=1}^{n} \sum_{l=1}^{h_i} ((\alpha_{il} - \alpha_{i,l-1}) n_{il} + (\beta_{il} - \beta_{i,l-1}) p_{il})$$

$(\alpha_{i0} = \beta_{i0} = 0)$

In order to aggregate these two objectives, the weights assigned to deviational variables must be interpreted in economic terms, and they are then estimated as the penalty production loss involved by the time in which each task is developed.

So the aggregate objective function obtained is formulated as follows

$$\text{mín} \sum_{i=1}^{n} \sum_{j=1}^{m_i} c_{ij} x_{ij} + \sum_{k=1}^{nr} c f_k y_k + \sum_{i=1}^{n} \sum_{l=1}^{h_i} ((\alpha_{il} - \alpha_{i,l-1}) n_{il} + (\beta_{il} - \beta_{i,l-1}) p_{il})$$

$(\alpha_{i0} = \beta_{i0} = 0)$

4. Constraints

The constraints of the model are as follows:

- **Assignment constraints:** Each task is to be performed just once:

$$\sum_{j=1}^{m_i} x_{ij} = 1 \quad \forall i = 1, \ldots, n$$

- **Precedence forcing constraints:** If task i_1 must take precedence over task i_2, then task i_2 cannot be started until task i_1 is finished:

$$T_{i_1} + \sum_{j=1}^{m_{i_1}} d_{i_1 j} x_{i_1 j} \leq T_{i_2}, \quad \forall (i_1, i_2) \in \mathcal{P}.$$

- **Fixed cost constraints:** They ensure that a resource will not be used if its associated fixed cost is not incurred.

$$x_{ij} \leq y_k, \quad \forall i = 1, \ldots, n,\ j = 1, \ldots, m_i,\ k \in \mathcal{K}_{ij}.$$

- **Time windows constraints:** The resultant conditions for the left side penalty function, according to the Jones and Tamiz formulation, are:

$$\begin{aligned} T_i + n_{i1} - p'_{i1} &= a_{i0} \\ T_i + n_{i2} - p'_{i2} &= a_{i1} \\ &\ldots \qquad\qquad \forall i = 1, \ldots, n \\ T_i + n_{ih_i} - p'_{ih_i} &= a_{i,h_i-1} \\ T_i &\geq a_{ih_i} \end{aligned}$$

Similarly, the resultant conditions for the right side penalty function are:

$$\begin{aligned} T_i + \sum_{j=1}^{m_i} d_{ij} x_{ij} + n'_{i1} - p_{i1} &= b_{i0} \\ T_i + \sum_{j=1}^{m_i} d_{ij} x_{ij} + n'_{i2} - p_{i2} &= b_{i1} \\ &\ldots \qquad\qquad \forall i = 1, \ldots, n \\ T_i + \sum_{j=1}^{m_i} d_{ij} x_{ij} + n'_{ih_i} - p_{ih_i} &= b_{i,h_i-1} \\ T_i + \sum_{j=1}^{m_i} d_{ij} x_{ij} &\leq b_{ih_i} \end{aligned}$$

- **Binary character of integer variables:**

$$\begin{aligned} y_k &\in \{0, 1\}, \quad \forall k = 1, \ldots, nr, \\ x_{ij} &\in \{0, 1\}, \quad \forall i = 1, \ldots, n\ j = 1, \ldots, m_i. \end{aligned}$$

- **Feasible resource-assignment constraints:**

  In order to avoid simultaneous use of resources, arrangements must be made to prevent two modes of different tasks overlapping in time when those two modes are selected and both use the same resource. For tasks that might be involved in such a situation, a special class of constraints, which we will call *incompatibility conditions* will be defined.

  Given two different tasks, i_1 and i_2, the following conditions when verified simultaneously define the incompatibility between modes j_1 of i_1 and j_2 of i_2:

  Inc-1 The intersection between the general time windows of tasks i_1 and i_2 is not empty:

  $$[a_{i_1 h_{i_1}}, b_{i_1 h_{i_1}}] \cap [a_{i_2 h_{i_2}}, b_{i_2 h_{i_2}}] \neq \emptyset.$$

  Inc-2 There are no precedence relationships between tasks (i_1, i_2), $(i_2, i_1) \notin \overline{\mathcal{P}}$ (where $\overline{\mathcal{P}} = \mathcal{P}$ extended by transitivity: $\overline{\mathcal{P}} = \{(i_1, i_2) \mid$ there is a path from i_1 to i_2 in $G = (\mathcal{N}, \mathcal{P})\}$).

  Inc-3 Modes j_1 and j_2 share at least one resource:

  $$\mathcal{K}_{i_1 j_1} \cap \mathcal{K}_{i_2 j_2} \neq \emptyset.$$

  In the event of tasks i_1 and i_2 being performed under modes j_1 and j_2 respectively ($x_{i_1 j_1} = 1$ and $x_{i_2 j_2} = 1$), then they must not coincide in time, and it must therefore be required that one begins when the other is completed:

  $$x_{i_1 j_1} + x_{i_2 j_2} = 2 \Rightarrow \begin{cases} T_{i_1} + d_{i_1 j_1} \leq T_{i_2} \\ \text{or} \\ T_{i_2} + d_{i_2 j_2} \leq T_{i_1}. \end{cases} \tag{4.1}$$

  The disjunction shown on the right side of the implication can be modelled as a conjunction using the classical methods of integer programming (see Nemhauser and Wolsey [8] among others).

  $$x_{i_1 j_1} + x_{i_2 j_2} = 2 \Rightarrow \begin{cases} T_{i_1} + d_{i_1 j_1} - T_{i_2} \leq M_{i_1 j_1 i_2 j_2}(1 - \delta_{i_1 j_1 i_2 j_2}) \\ T_{i_2} + d_{i_2 j_2} - T_{i_1} \leq M_{i_2 j_2 i_1 j_1} \delta_{i_1 j_1 i_2 j_2}. \end{cases}$$

  Once the disjunction has been entered into the mathematical model, the implication must be included giving the following expression:

  $$T_{i_1} + d_{i_1 j_1} - T_{i_2} - M_{i_1 j_1 i_2 j_2}(1 - \delta_{i_1 j_1 i_2 j_2}) \leq M'_{i_1 j_1 i_2 j_2}(2 - x_{i_1 j_1} - x_{i_2 j_2}), \tag{4.2}$$

  $$T_{i_2} + d_{i_2 j_2} - T_{i_1} - M_{i_2 j_2 i_1 j_1} \delta_{i_1 j_1 i_2 j_2} \leq M'_{i_2 j_2 i_1 j_1}(2 - x_{i_1 j_1} - x_{i_2 j_2}). \tag{4.3}$$

  Generalised incompatibilities

  The incompatibilities defined previously refer to a pair of modes (j_1, j_2) for performing two tasks (i_1, i_2). However, it may happen that the same resource is needed not just by modes j_1 and j_2 but also by several modes associated with tasks i_1 and i_2.

Thus if resource k can be used to perform tasks i_1 and i_2, then all the modes that use this resource will be incompatible and cannot be selected simultaneously for performing those tasks. The following conditions, when verified simultaneously, define the incompatibility between tasks i_1 and i_2 in the use of resource k:

I-1 The intersection between the general time windows of tasks i_1 and i_2 is not empty:

$$[a_{i_1h_{i_1}}, b_{i_1h_{i_1}}] \cap [a_{i_2h_{i_2}}, b_{i_2h_{i_2}}] \neq \emptyset$$

I-2 There are no precedence relationships between tasks (i_1, i_2), $(i_2, i_1) \notin \overline{\mathcal{P}}$ (defined in [Inc-2]).

I-3 Neither of mode sets $\mathcal{J}_{i_1}^k$ and $\mathcal{J}_{i_2}^k$ (defined in 3.1) is empty.

In the event of task i_1 being performed under one of the modes of $\mathcal{J}_{i_1}^k$, and task i_2 is performed under one of the modes of $\mathcal{J}_{i_2}^k$, then they cannot coincide in time, and therefore one beginning when the other has ended must be required:

$$\sum_{j\in\mathcal{J}_{i_1}^k} x_{i_1j} + \sum_{j\in\mathcal{J}_{i_2}^k} x_{i_2j} = 2 \Rightarrow \begin{cases} T_{i_1} + \displaystyle\sum_{j\in\mathcal{J}_{i_1}^k} d_{i_1j}x_{i_1j} \leq T_{i_2} \\ \text{or} \\ T_{i_2} + \displaystyle\sum_{j\in\mathcal{J}_{i_2}^k} d_{i_2j}x_{i_2j} \leq T_{i_1}. \end{cases} \tag{4.4}$$

Similarly to the process followed above, condition (4.4) is equivalent to conditions (4.5)and (4.6)

$$T_{i_1} + \sum_{j\in\mathcal{J}_{i_1}^k} d_{i_1j}x_{i_1j} - T_{i_2} - M_{i_1i_2k}(1-\delta_{i_1i_2k}) \leq M'_{i_1i_2k}\left(2 - \sum_{j\in\mathcal{J}_{i_1}^k} x_{i_1j} - \sum_{j\in\mathcal{J}_{i_2}^k} x_{i_2j}\right), \tag{4.5}$$

$$T_{i_2} + \sum_{j\in\mathcal{J}_{i_2}^k} d_{i_2j}x_{i_2j} - T_{i_1} - M_{i_2i_1k}\delta_{i_1i_2k} \leq M'_{i_2i_1k}\left(2 - \sum_{j\in\mathcal{J}_{i_1}^k} x_{i_1j} - \sum_{j\in\mathcal{J}_{i_2}^k} x_{i_2j}\right). \tag{4.6}$$

This new mathematical formulation of the incompatibilities needs a lower number of variables 0-1 and constraints.

Constants $M_{i_1i_2k}$ and $M_{i_2i_1k}$ must be large enough, and it is easily obtained that $b_{i_1h_{i_1}} - a_{i_2h_{i_2}}$ and $b_{i_2h_{i_2}} - a_{i_1h_{i_1}}$, respectively, are possible values for these constants. These values are also possible values for constants $M'_{i_1i_2k}$ and $M'_{i_2i_1k}$ (see Vitoriano et al. [10] for an application of this new type of constraints and a more detailed description).

5. Mathematical model

The resulting model is as follows:

$$\text{mín} \sum_{i=1}^{n}\sum_{j=1}^{m_i} c_{ij}x_{ij} + \sum_{k=1}^{nr} cf_k y_k + \sum_{i=1}^{n}\sum_{l=1}^{h_i}((\alpha_{il} - \alpha_{i,l-1})n_{il} + (\beta_{il} - \beta_{i,l-1})p_{il}) \tag{5.1}$$

subject to

$$\sum_{j=1}^{m_i} x_{ij} = 1, \quad \forall i = 1\ldots, n, \tag{5.2}$$

$$T_{i_1} + \sum_{j=1}^{m_{i_1}} d_{i_1 j}x_{i_1 j} \leq T_{i_2}, \quad \forall (i_1, i_2) \in \mathcal{P}, \tag{5.3}$$

$$x_{ij} \leq y_k, \quad \forall i = 1, \ldots, n,\ j = 1, \ldots, m_i,\ k \in \mathcal{K}_{ij}, \tag{5.4}$$

$$T_i + n_{il} - p'_{il} = a_{i,l-1}, \quad \forall i = 1, \ldots, n \quad l = 1, \ldots, h_i \tag{5.5}$$

$$T_i + \sum_{j=1}^{m_i} d_{ij}x_{ij} + n'_{il} - p_{il} = b_{i,l-1} \quad \forall i = 1, \ldots, n \quad l = 1, \ldots, h_i \tag{5.6}$$

$$T_i + \sum_{j=1}^{m_i} d_{ij}x_{ij} \leq b_{ih_i} \quad \forall i = 1, \ldots, n \tag{5.7}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i = 1, \ldots, n,\ j = 1, \ldots, m_i, \tag{5.8}$$

$$y_k \in \{0, 1\}, \quad \forall k = 1, \ldots, nr, \tag{5.9}$$

$$T_i \geq a_{ih_i}, \quad \forall i = 1, \ldots, n \tag{5.10}$$

and the incompatibility constraints for each pair of tasks (i_1, i_2) and each resource k verifying conditions (I-1), (I-2) and (I-3)

$$T_{i_1} + \sum_{j \in \mathcal{J}_{i_1}^k} d_{i_1 j}x_{i_1 j} - T_{i_2} \leq (b_{i_1 h_{i_1}} - a_{i_2 h_{i_2}})\left(3 - \sum_{j \in \mathcal{J}_{i_1}^k} x_{i_1 j} - \sum_{j \in \mathcal{J}_{i_2}^k} x_{i_2 j} - \delta_{i_1 i_2 k}\right), \tag{5.11}$$

$$T_{i_2} + \sum_{j \in \mathcal{J}_{i_2}^k} d_{i_2 j}x_{i_2 j} - T_{i_1} \leq (b_{i_2 h_{i_2}} - a_{i_1 h_{i_1}})\left(2 - \sum_{j \in \mathcal{J}_{i_1}^k} x_{i_1 j} - \sum_{j \in \mathcal{J}_{i_2}^k} x_{i_2 j} + \delta_{i_1 i_2 k}\right), \tag{5.12}$$

$$\delta_{i_1 i_2 k} \in \{0, 1\}. \tag{5.13}$$

6. A case study

The model proposed to deal with infeasibilities has been applied to several real cases, focused on agricultural planning. Vitoriano et al. [10] present a real-life problem in which crop production planning is to be decided at the beginning of the agricultural year. Several crops have to be carried out at the same time, and the tasks to be developed share a set of resources.

In that paper a model to obtain an optimal planning is proposed, and several examples and their associated computational experience are presented. The problem is that, as the number of crops involved increases, the number of infeasible cases also increases, or the optimal solution proposed involves the acquisition of new resources, generally at a very high cost. Moreover, if the time windows was extended to achieve feasible solutions, it was not possible to distinguish inside the most efficient period to perform the tasks.

One of these cases is presented below.

The case consists of 5 crops which have to be developed at the same time, with 67 tasks in total to be carried out and 45 precedence relationships between them. The model has a high level of flexibility in the way of developing the tasks. 27 resources have been considered for planning which, combined in different forms to develop the 67 tasks, lead to 556 possible modes.

Each task has an associated ideal time window but, for the flexible model proposed, a deviation of 20 days left or right is accepted as feasible, incurring in a production loss. Penalties per day outside the ideal time window have been estimated as 5 % of the average cost of the task.

Two models have been formulated. Model 1 allows no flexibility concerning time windows; model 2 includes the flexibility described above. Models dimensions are shown in tables 1 and 2. Note that the number of incompatibilities constraints is the same in both models, but it is not general, because extending time widows could give an increase in the number of incompatibilities.

Variables	Binary	Continuous	SOS1
Model 1	273 + 27	67	556
Model 2	273 + 27	335	556

Table 1: Number of variables

Constraints	Fixed Cost	Precedence	Incompatibilities	Time windows
Model 1	1753	45	2×968	67
Model 2	1753	45	2×968	3×67

Table 2: Number of constraints

The problems have been solved using CPLEX in a Pentium III with an Intel processor to 1Ghz. In this case model 1, with rigid time windows, is infeasible, and to be able to develop it new resources, with a high cost associated, will be needed. However model 2, with flexible time windows, is feasible, and it is possible to obtain a solution with a tolerance of 2 % in 224 seconds.

If the problem is modified by decreasing the tasks lengths conveniently, both models achieve the same optimal solution in few seconds.

7. Conclusions

The production process entails taking a large number of decisions aimed at improving productivity and achieving the best yield from available resources, which are normally limited. Assuming that there is a certain path of tasks to be carried out within a period, and that each task can be done in different ways, the problem consists on choosing how and when to carry out each one, in such a way that the tasks are scheduled in sequence at the lowest possible cost, taking account of any relations of precedence among them, and in such a way that each is done within its time window and with the resources being assigned in a feasible way. In case a task can not be done within its ideal time window, a production loss is incurred, and this loss has to be minimized.

As noted, this is a process that takes in conditions concerning the sequencing of tasks, time windows for performing them, production loss variables and conditions, variable and fixed costs, and a flexibility in the manner of performing the tasks that results in the inclusion of assignment conditions and conditions that have been called 'feasible resource-assignment conditions'.

In this paper, a linear programming model has been presented to achieve the two proposed objectives, aggregating them by assigning weights to deviational variables over both goals which are estimated like penalties for production loss. A case study about agricultural planning is commented, showing the power of the model to achieve feasible and not so expensive solutions.

References

.1. Charnes A. and Collomb. Optimal economic stabilization policy: Linear goal-interval programming models. *Socio-Econ Plan Science* **6**, 1972, p. 431-435.

.2. Charnes A. et al. A goal interval programming model for resource allocation in a marine environmental protection program. *J Environ Econ and Mgmt* **3**, 1976, p. 347-362.

.3. Charnes A. and Cooper W.W. Goal programming and multiple objective optimization. Part I. *European Journal of Operational Research* **1**, 1977, p. 39-54.

.4. Darby–Dowman, K., Barker, S., Audsley, E. and Parsons, D.. A Two Stage Stochastic Programming with Recourse Model for Determining Robust Planting Plans in Horticulture. *Journal of the Operational Research Society* **51**, 2000, p. 83-89.

.5. Ignizio J.P. . *Interval Goal Programming and Applications.* Working Paper, Pennsylvania State University, 1974

.6. Jones, D.F. and Tamiz, M.. Expanding the Flexibility of Goal Programming via Preference Modelling Techniques. *Omega, Int. J. Management Science* **23**, 1995, p. 41-48. 5

.7. Kvanli, A.H., Financial Planning Using Goal Programming, OMEGA, Vol. 8, No. 2, 1980

.8. Nemhauser, and Wolsey, L. *Integer and Combinatorial Optimization.* Wiley, 1989.

.9. Vitoriano, B. and Romero, C. Extended interval goal programming. *Journal of the Operational Research Society* **50**, 1999, 1-4.

.10. Vitoriano, B., Ortuño, M.T., Recio, B., Rubio, F., and Alonso, A. Two alternative models for farm

management: discrete versus continuous time horizon. *European Journal of Operational Research* **144**, 2003, 613-628