A Production Programming Model

Multistage with Intermediate Stocks

Gerson Ulbricht

Department of Mathematics
Federal Institute of Santa Catarina – IFSC, Jaraguá do Sul – 89254-430, Brazil

Neida M. Patias Volpi

Department of Mathematics
Federal University of Paraná – UFPR - 81531-990, Brazil

Copyright © 2015 Gerson Ulbricht and Neida M. Patias Volpi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

In this article is presented a mathematical model for production programming in two stages, the first one being composed by different machines that operate in a parallel way, and the second one, formed by a section for the items packaging. The objective of the model is to minimize production costs basing on the items arrangement, for both better machines choice, and for the correct production scheduling, since the time and the preparation cost of the machines for the manufacture of a new item is dependent on the sequence in which they are made. Due to the complexity of the mathematical model, where it isn’t possible to obtain a great solution in an acceptable time for the industry standards, the model is solved with a method of relaxation of integer variables called relax-and-fix, where with it’s obtained promising results.

Keywords: Production programming, multiple stages, relax-and-fix

1 Introduction

In the last decades, several studies have been realized in order to search for mathematical models that can help on decisions at the industrial environment, which involve the Planning and Production Control.
Recent studies concerning programming and lots sizing led to the GLSP formulation (General Lot Sizing and Scheduling Problem) presented by Fleischmann & Meyr [7], where each period is divided in subperiods in a way that in each of the subperiods it is allowed the production of the lot of only one kind of item, so that the obtainment of a feasible solution results in both sizing, and lots scheduling. Various extensions have been applied to GLSP, for instance, the fact of considering cost and time of setup dependent on the sequence in which the items will be produced (Clark & Clark [2]). An important extension refers to the case of using multiple machines, which was called GLSPPL - General Lot Sizing and Scheduling Problem for Parallel Production Lines (see Meyr [10]).

Some industrial processes are organized in a way that they occur in different stages or productive levels, what makes the models more complex to be formulated and computationally solved (Ferreira et al [4]).

A classic model was proposed by Drexl & Kimms [3], named Multi-level PLSP model, which includes productive multiple stages, considering that the production of a final item requires the production of intermediate items, which are produced in prior stages.

Some recent works that consider productive multiple stages have been published in recent years. Mohammadi et al [11], consider a multi-level flowshop production where all the machines are arranged in series. Ferreira et al [4] proposes an application on the soda production in a small factory considering two interdependent stages: syruping and package. The model considers parallel machines with setup dependent on the sequence. Transchel et al [18] presented a model based on GLSP, applied to the process of production in an industry, with two stages. The first stage is based in a problem of scheduling while on the second one the costs are accounted with setup stock and production. Ferreira et al [6], proposed single-stage mathematical formulations aimed to the resolution of the problem proposed by Ferreira et al [4], with the objective of reducing the dimensions of the problem and consequently the computational effort for the resolution. Seanneer&Meyr [16] presented a mathematical model called GLSPMS - General Lot-Sizing and Scheduling Problem for Multiple production Stages. The model proposed by these authors approaches the multistage production with the possibility of having parallel production lines by stage.

In this article is proposed a model based on GLSPMS that can attend the reality of business environment, as it will be explained in the next sections.

2 Problem Presentation

The mathematical model presented in this work was elaborated from studies in an electronic plates producer industry, with the objective of helping at the management of productive process.

The fabrication process, in this industry occurs in two stages. In the first one, machines that work in a parallel way are responsible for the placement of elec-
Electronic components in printed circuit plates. In the second stage, the produced items follow to a packaging sector, where each type of item has a designated time for the packaging process, which must rather be fulfilled within the daily 8h of operating of the enterprise. If the produced items in the first stage can’t be packaged within the daily working hours, there is the overtime generation ($e_{itt}$), or yet they can be in stock between the two stages, to be packaged in some posterior day to the fabrication day ($I_{it}^+$), since that to the end of the planning horizon that is 1 week long, all the demand be produced and packaged, what is indicated by $I_{it}^+ = 0$.

The picture 1 shows a summary of the operation of these 2 stages.

With the objective of help in the planning and production control, it was proposed a mathematical model, which is presented in section 3 of this work, together with the description of its operation. The proposed model carries some features of models present in literature, for instance, the periods structure with limited capacity, divided in subperiods of variable size present in GLSPPL (Meyr [10]), and the existence of multiple stages presents in GLSPMS according to Seeanner & Meyr [16]. However, considering to attend the own features of the producing environment the model presents particularities that makes it different from the others present in literature.

3 Mathematical Model

In this topic is presented the mathematical model proposed, which is multi-machines, multi-items, and composed by 2 production stages type.

Below are presented the indexes, parameters and variables of the mathematical model.

Indexes:

\( i, j \): items;
\( k \): machines from stage I
\( t \): periods
\( s \): subperiods
Parameters:

- \( n \): number of items
- \( T \): number of periods in the planning horizon
- \( W \): total number of subperiods in all the planning horizon
- \( W_t \): set of subperiods contained on period \( t \)
- \( K \): number of machines belonging to stage \( I \)
- \( SP_k \): set of items that can be produced on machine \( k \) (stage \( I \))
- \( CP_{kt} \): capacity of producing available on machine \( k \) in period \( t \) (stage \( I \))
- \( cp_{ik} \): cost of the production to produce the item \( i \) on machine \( k \) (stage \( I \))
- \( ch_i \): cost of overtime to process the item \( i \), on stage \( II \)
- \( D_i \): demand of item \( i \) in planning horizon
- \( cs_{ijk} \): cost of setup to produce the item \( j \) right after the item \( I \) on machine \( k \) (stage \( I \))
- \( st_{ijk} \): time of setup for production of the item \( j \) right after the item \( i \) on machine \( k \) (stage \( I \))
- \( p_{ik} \): consumed time for production of a unit from item \( i \) on machine \( k \) (stage \( I \))
- \( lmin_k \): minimum lot of item \( i \) that can be produced on machine \( k \) (stage \( I \))
- \( x_{ik0} \): if the machine \( k \), from stage \( I \), is prepared to produce the item \( i \), in the beginning of the planning horizon; 0, otherwise

Variables:

- \( q_{iks} \): quantity of item \( i \) produced on machine \( k \) in subperiod \( s \) (stage \( I \))
- \( q_e_{it} \): quantity of item \( i \) sent to the packaging sector in period \( t \)
- \( I_i^+ \): quantity of item \( i \), kept in stock between stages, in period \( t \)
- \( x_{iks} \): equals 1, if the machine \( k \) is prepared to produce the item \( i \) in subperiods (stage \( I \))
- \( z_{iks} \): equals 1, when the item \( i \) is produced on machine \( k \) in subperiods (stage \( I \))
- \( y_{ijks} \): equals 1, if there is setup of the item \( i \) to the item \( j \) on machine \( k \) in subperiod \( s \) (stage \( I \))
- \( HT_{ks} \): stores the ending instant of lots in each machine \( k \) in subperiods (stage \( I \))
- \( Hm_{iks} \): assign the ending instant \( HT'_{ks} \) to each item \( i \) in subperiod \( s \) (stage \( I \))
- \( H_l_i \): ending instant of lot from item \( i \) in each period \( t \) (stage \( I \))
- \( Hte_{it} \): ending instant of lot from item \( i \) in each period \( t \) (stage \( I \)), added to the time that the item must remain on stage \( II \)
- \( e_{ikt} \): quantity in extra minutes on stage \( II \), for process finalization of the packaging of item \( i \) coming from each machine \( k \) in period \( t \).

Below is presented the mathematical model, followed by the description of the operation of objective function and of each one of the constraints.

\[
\min Z = \sum_{i \in SP_k} \sum_{j \in SP_k} \sum_{s=1}^W cs_{ijk} \cdot y_{ijks} + \sum_{i \in SP_k} \sum_{s=1}^W cp_{ik} \cdot q_{iks} + \sum_{i \in SP_k} \sum_{k=1}^K \sum_{t=1}^T ch_i \cdot e_{ikt} \tag{1}
\]
A production programming model multistage with intermediate stocks

Subject to:

\[ \sum_{i \in S} \sum_{k \in W_{i}} p_{ik} \cdot q_{iks} + \sum_{i \in S} \sum_{k \in W_{i}} \sum_{j \in S} \sum_{t \in W_{j}} s_{tijk} \cdot y_{ijks} \leq CP_{kt} \quad k = 1, ..., K, t = 1, ..., T \]  \hspace{1cm} (2)

\[ p_{ik} \cdot q_{iks} \leq CP_{kt} \cdot x_{iks} \quad i \in S, k = 1, ..., K, s \in W_{i} \text{ where } t = 1, ..., T \]  \hspace{1cm} (3)

\[ q_{iks} \geq l_{min} \cdot (x_{iks} - x_{ik,s-1}) \quad i = 1, ..., n, k = 1, ..., K, s = 1, ..., W \]  \hspace{1cm} (4)

\[ \sum_{i \in S} x_{iks} = 1 \quad k = 1, ..., K, s = 1, ..., W \]  \hspace{1cm} (5)

\[ y_{ijks} \geq x_{ik,s-1} + x_{jks} - 1 \quad i, j = 1, ..., n, s = 1, ..., W, k = 1, ..., K \]  \hspace{1cm} (6)

\[ l_{min} \cdot z_{iks} \leq q_{iks} \leq D_{i} \cdot z_{iks} \quad i = 1, ..., n, s = 1, ..., W, k = 1, ..., K \]  \hspace{1cm} (7)

\[ \sum_{s \in W_{i}} z_{iks} \leq 1 \quad i = 1, ..., n, k = 1, ..., K, t = 1, ..., T \]  \hspace{1cm} (8)

\[ \sum_{i \in S} z_{ik(s-1)} \geq \sum_{i \in S} z_{iks} \quad s \in W_{i} \text{ where } t = 1, ..., T \]  \hspace{1cm} (9)

\[ HT_{ks} = HT_{k(s-1)} + \sum_{i \in S} p_{ik} \cdot q_{iks} + \sum_{i \in S} \sum_{j \in S} \sum_{t \in W_{j}} s_{tijk} \cdot y_{ijks} \quad s \in W_{i} \text{ where } t = 1, ..., T \]  \hspace{1cm} (10)

\[ Hm_{iks} = HT_{ks} \cdot z_{iks} \quad i = 1, ..., n, s = 1, ..., W, k = 1, ..., K \]  \hspace{1cm} (11)

\[ H_{it} \geq \sum_{s \in W_{i}} Hm_{iks} \quad i = 1, ..., n, t = 1, ..., T, k = 1, ..., K \]  \hspace{1cm} (12)

\[ q_{e_{it}} = \sum_{k=1}^{K} q_{iks} + I_{ik}^{t} - I_{ik}^{t} \quad i = 1, ..., n, t = 1, ..., T \]  \hspace{1cm} (13)

\[ H_{te_{it}} = H_{it} + t_{e_{i}} \quad i = 1, ..., n, t = 1, ..., T \]  \hspace{1cm} (14)

\[ \sum_{t=1}^{T} q_{e_{it}} = D_{i} \quad i = 1, ..., n \]  \hspace{1cm} (15)

\[ e_{ikt} \geq H_{te_{it}} - CP_{kt} \quad i = 1, ..., n, k = 1, ..., K, t = 1, ..., T \]  \hspace{1cm} (16)

\[ q_{iks}, I_{ik}^{t} \in Z^{+} \forall i, t, s \]

\[ x_{iks}, z_{iks} \in \{0,1\} \forall i, k, s \]

\[ y_{ijks}, HT_{ks}, Hm_{iks}, q_{e_{it}}, H_{it}, H_{te_{it}}, e_{ikt} \geq 0 \forall i, j, t, k, s \]

\[ x_{ik0} = 1 \forall i, k \]  \hspace{1cm} (17)
The objective function (1) represents the minimization of setup costs and production of machines from the first productive stage, as well as the cost of using overtime caused by the process of items packaging in the second stage.

The set of constraints (2) ensures that the consumed time in the first stage for items production, added to the necessary time for machines preparation, doesn’t exceed the available time, in each machine of each period \( t \).

The set of constraints (3) indicates that there will be only production of item \( i \) in subperiods if the machine \( k \), from the first stage, is prepared for the production of the respective item \( x_{iks} = 1 \).

The set of constraints (4) allows to impose a condition of minimum lot of production for each type of item in a subperiod from the first stage, which is only activated when occurs the machine preparation for a new item.

The set of constraints (5) determines that the machine is prepared to produce only one type of item per machine in each subperiod \( s \) from the first productive stage.

The set of constraints (6) indicates if there was changing of production of item \( i \) to the item \( j \) on machine \( k \) between the subperiods from the first stage.

The set of constraints (7) imposes the condition that when the item \( i \) is produced on machine \( k \) in subperiod \( s \) from the first stage, the variable \( z_{iks} \) takes up the value 1. Otherwise, \( z_{iks} = 0 \). For programming purposes, this constraint can be divided in two other constraints, the first one being \( q_{iks} \geq t_{\text{min}}_{ik} \cdot z_{iks} \) and the second one \( q_{iks} \leq D_i \cdot z_{iks} \).

The set of constraints (8) imposes that the lot of an item from the first stage, is produced all inside an only subperiod from the respective period to which it belongs to.

The set of constraints (9) orders the production in successive subperiods inside each period from the first stage, knowing that in case of existing more subperiods than items in determined period, the idle subperiods stay concentrated in the end of each period.

The set of constraints (10) is responsible for the calculation of the ending instant in the first stage, considering the sum of the lots production time and the machines preparation times (setup). The result obtained in this set of constraints can be interpreted as \( k \) vectors with \( s \) components each (a vector for each machine), where each element represents the ending instant in the first stage, of the items lots produced in each subperiod from the planning horizon, as shown in the expression (18).
A production programming model multistage with intermediate stocks

\[ H_T^{(k=1)_s} \]

Mach. 1:
\[
\begin{bmatrix}
ht_1 & ht_2 & \ldots & ht_w_1, \\
ht_1 & ht_2 & \ldots & ht_w_2, \\
\vdots & \vdots & \ddots & \vdots \\
ht_1 & ht_2 & \ldots & ht_w_t
\end{bmatrix}
\]

\[ HT: \]
\[
\begin{bmatrix}
\vdots \\
Period~1 & \ldots & Period~t
\end{bmatrix}
\]

Mach.\(k\):
\[
\begin{bmatrix}
ht_1 & ht_2 & \ldots & ht_w_1, \\
ht_1 & ht_2 & \ldots & ht_w_2, \\
\vdots & \vdots & \ddots & \vdots \\
ht_1 & ht_2 & \ldots & ht_w_t
\end{bmatrix}
\]

\[ H_T^{(k)_s} \]

In a way to assign this calculated ending instant, to the respective item, it is necessary the set of constraints (11), which is responsible for forming \(k\) dimension arrays \(i \times W, (s \in W)\), containing the ending instant in each machine for each item \(i\) produced in each subperiod \(s\) from the first stage. Considering \(z_{iks}\) as being \(k\) dimension arrays \(i \times W, (s \in W)\), which contain binary results where the value 1 represents that the item \(i\) was produced in determined subperiod, and where as an only item can be produced in a subperiod as guaranteed by the sets of constrains (5) and (8), through \(k\) arrays \(Hm_{ks}\) is possible to attribute to each item its ending instant, because the \(k\) vectors \(H_T^s\) just presented the ending instants of each item lots in each period, but without assigning this value to the item to which it represented.

An example considering the machine \(k = 1\), is presented by the expressions (19) to (21).

\[ HT^{(k=1)_s} \]

\[
\begin{bmatrix}
ht_1 & ht_2 & \ldots & ht_w_1, \\
ht_1 & ht_2 & \ldots & ht_w_2, \\
\vdots & \vdots & \ddots & \vdots \\
ht_1 & ht_2 & \ldots & ht_w_t
\end{bmatrix}
\]

Performing the multiplication element by element (example):

\[ z_{i(k=1)_s} \]

\[
\begin{bmatrix}
0 & 0 & 0 & 1 \\
0 & 1 & 0 & \ldots & 0 \\
1 & 0 & 0 & \ldots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & \ldots & 1
\end{bmatrix}
\]
Results in:

\[
\begin{bmatrix}
0 & 0 & \text{ht}_3 & 0 \\
0 & \text{ht}_2 & 0 & \cdots & 0 \\
\text{ht}_1 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & \text{ht}_{\text{w}_1}
\end{bmatrix}
\]  

(21)

It can be noted that the set of constraints (11) makes the nonlinear model. Linear models are in general less complex to be solved if compared to nonlinear models.

The linearization process of a constraint composed by a binary variable and another continuous one, according to Bisschop (2000, p. 82) is presented below.

Two variables being \( x_1 \) and \( x_2 \), being \( x_1 \) binary and \( x_2 \) the continuous variable limited by \( 0 \leq x_2 \leq M \) where \( M \) is a big number. The expression \( y = x_1 \cdot x_2 \) is equivalent to:

\begin{align*}
\text{i)} & \quad y \leq M \cdot x_1 \\
\text{ii)} & \quad y \leq x_2 \\
\text{iii)} & \quad y \geq x_2 - M(1 - x_1) \\
\text{iv)} & \quad y \geq 0
\end{align*}  

(22)

It can be noted that if \( x_1 = 0 \), this implies in \( y = 0 \). If \( x_1 = 1 \), then the expression \( y \geq x_2 - M(1 - x_1) \) results in \( y \geq x_2 \), that combined with \( y \leq x_2 \), implies \( y = x_2 \) and this way there are linear expressions equivalent to the nonlinear expression.

In the case of the set of constraints (11), there is:

\[ Hm_{i(\text{k}=\text{s})} = HT_{\text{ks}}, z_{i\text{ks}} \quad i = 1, \ldots, n \quad k = 1, \ldots, K \quad s = 1, \ldots, W \]  

(23)

It can be noted in the model that the set of variables \( HT_{\text{ks}} \) is real positive, in other words, continuous, while the set of variables \( z_{i\text{ks}} \) is binary.

Applying the linearization process in the set of constraints (11), the nonlinear constraint \( Hm_{i\text{ks}} = HT_{\text{ks}}, z_{i\text{ks}} \), was substituted by 4 sets of constraints below, where \( i = 1, \ldots, n; \quad k = 1, \ldots, K \) e \( s = 1, \ldots, W \):

\begin{align*}
\text{i)} & \quad Hm_{i\text{ks}} \leq M \cdot z_{i\text{ks}} \\
\text{ii)} & \quad Hm_{i\text{ks}} \leq HT_{\text{ks}} \\
\text{iii)} & \quad Hm_{i\text{ks}} \geq HT_{\text{ks}} - M \cdot (1 - z_{i\text{ks}}) \\
\text{iv)} & \quad Hm_{i\text{ks}} \geq 0
\end{align*}  

(24)

The set of constraints (12) calculates the ending instant of each item, in each period from the first stage, in a way that if a same item is produced in different machines as calculated by the set of constraints (11), predominates the largest ending
A production programming model multistage with intermediate stocks

instant. It can be noted that through the constraint (11), are obtained values that can be organized in \( k \) dimension arrays \( i \times W (s \in W) \), as illustrated in the example (25), for a determined machine \( k = 1 \).

\[
Hm_{i(k=1)s} = \begin{bmatrix}
0 & 0 & h_{t_3} & 0 \\
0 & h_{t_2} & 0 & \cdots & 0 \\
h_{t_1} & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & h_{t_{w_1}} \\
\end{bmatrix}
\]

(25)

Thus, in the set of constraints (12), the summation \( \sum_{s \in W} Hm_{iks} \), causes them to be generated \( k \) dimension arrays \( i \times t \), that represent the ending instant in each machine from stage I. This way, each line from the \( H_{it} \) array represents the item \( i \) and each column the period \( t \). Applying it on the example showed before, for each machine is obtained an array as shown below.

\[
\begin{bmatrix}
h_{t_3} & h_{t_2} & \cdots & h_{t_1} \\
h_{t_2} & h_{t_1} & \cdots & h_{t_2} \\
h_{t_1} & h_{t_3} & \cdots & h_{t_3} \\
\vdots & \vdots & \ddots & \ddots \\
h_{t_{w_1}} & h_{t_{w_2}} & \cdots & h_{t_{w_t}} \\
\end{bmatrix}
\]

(machine\( k=1 \))

(26)

Due to the fact that an item can be processed in several machines in parallel, the approach made in this mathematical model allows the items to be forwarded to the second stage only when the complete lot production of the respective item is finished in the first stage. This way, through the constraint (12), the variables \( H_{it} \) always take up a higher or equal value to the ending instants of each machine from the first stage, but limited to the minimization of overtime through the objective function, obtaining then, the \( H \) array, which indicates the instant that the processing of each lot from item \( i \) in each period \( t \) was finished in the first productive stage.

The set of constraints (13) determines that the items produced in all the machines during the subperiods \( s \) belonging to the period \( t \), or in stock of the previous periods in the first productive stage, are sent to the packaging sector, in other words, to stage II.

The set of constraints (14) calculates the ending instant of processing in stage I, added from the time in which the items take to be packaged on stage II.

The set of constraints (15) imposes that the amount of quantity of items sent to stage II during all the periods be equal to the expected demand for the planning horizon.
The set of constraints (16) calculates the quantity of necessary overtime on stage II, in order to conclude the items production. It is important to point out that for the first stage it wasn’t considered the use of overtime, letting the production conditioned to the capacity of each period. The assignment of overtime on stage II, is due to the fact that most items are not available for processing in the beginning of the second stage, fact that doesn’t occur in the first stage, where all the items are available to be processed in the beginning of each period.

The set of constraints (17) indicates the domain and initial condition of the variables.

4 Resolution Strategies

According to Ferreira et al [5], “[...] Full mixed optimized models are difficult to be solved, even those that only consider the lots sizing”. According to Meyr [10], for some problems, even the fact of finding a feasible solution to begin the search for better solutions, is a hard task.

According to Toledo [17], “An increasing number of researchers have appealed to heuristics and metaheuristics to deal with this complexity of finding great solutions or almost great ones in a reasonable computational time”. A lot of researchers have currently worked with heuristics that consist in integer variables relaxation, mainly applied to production programming problems ((Kawamura [8], Ferreira et al, [4], Ribeiro, [14]), herein among the most used relaxation strategies stand out the relax-and-fix (Merce and Fontan [9], Pochet and Wolsey [13]) and the Fix-and-optimize (Sahling et al [15]).

The heuristic relax-and-fix is based on the partition of integer variables of the problem, creating $P$ distinct sets, $Q_i$, $i=1,...,P$, knowing that the number $P$ of sets determines the number of iterations of the heuristic. According to Moraes [12], “in an iteration $n$, only the variables of the set $Q_n$ are defined as integer and the other variables are relaxed or fixed”. After the resolution of an iteration $n$, the obtained results for the integer variables are fixed and the variables of a new partition $P$ are defined as integer. The process is ended when all the partitions are solved or the subproblem created is unfeasible (Ribeiro [14]).

5 Model Resolution

For the resolution of the model were used two strategies relax-and-fix based on the relaxation of the integer variables in each period. In the first strategy, called relax-and-fix forward (RFF), The sequence of relaxation of the variables and resolution was initiated in chronological order starting up from the subproblem corresponding to the first period of the planning horizon and finishing it in the last iteration, in the subproblem corresponding to the last period. Now the second strategy, called relax-and-fix backward (RFB), the sequence of subproblems to be solved was initiated in reverse order starting from the subproblem corresponding to the last period and finishing it with the subproblem corresponding to the first period.
To perform the tests, it was created 3 problems of different sizes, where all consider a planning horizon composed of 5 periods, each period being corresponding to a day of the week. Each period was divided in subperiods equal to the quantity of items to be produced in a way to enable the production of all items concomitantly. Table 1 shows the dimension of the generated problems.

<table>
<thead>
<tr>
<th>Description</th>
<th>Problem A</th>
<th>Problem B</th>
<th>Problem C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraints</td>
<td>6,680</td>
<td>23,090</td>
<td>41,406</td>
</tr>
<tr>
<td>Total of variables</td>
<td>4,100</td>
<td>14,874</td>
<td>26,993</td>
</tr>
<tr>
<td>Binary variables</td>
<td>823</td>
<td>2,531</td>
<td>3,948</td>
</tr>
<tr>
<td>Integer variables</td>
<td>420</td>
<td>1,277</td>
<td>1,988</td>
</tr>
<tr>
<td>Real variables (not integer)</td>
<td>2,857</td>
<td>11,066</td>
<td>21,057</td>
</tr>
<tr>
<td>Coefficient different from zero</td>
<td>66,360</td>
<td>359,887</td>
<td>996,195</td>
</tr>
</tbody>
</table>

### 6 Found Results for the Generated Instances

Considering the resolution of the 3 problems, being each one of them with the use of the 2 strategies \textit{relax-and-fix}, there are 6 proposed instances, which were implemented using the computational package IBM ILOG CPLEX Optimization Studio 12.6 with interface OPL using a computer with Intel Core i5 processor (2.4GHz) and 8Gb of RAM memory.

For each one of the generated problems (A, B and C), the model was executed in a nonrelaxed way, for a time of 2h (7200 minutes). The results found are shown in table 2, which also presents the gap \((gap_1)\), as expression (27), obtained in relation to the inferior limiting, in other words, in relation to the value of the objective function with all the integer variables relaxed.

\[
gap_1 = \frac{(\text{limited time solution}) - (\text{inferior limiting})}{\text{inferior limiting}} \times 100
\]

<table>
<thead>
<tr>
<th>Problem</th>
<th>(\text{gap}_1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>29.13%</td>
</tr>
<tr>
<td>B</td>
<td>14.42%</td>
</tr>
<tr>
<td>C</td>
<td>26.84%</td>
</tr>
</tbody>
</table>

In a way to make the first verifications about the use of \textit{relax-and-fix}, the 6 generated instances were then executed with the use of this heuristic.

The time summation for the resolution of all the generated suproblems with the heuristic \textit{relax-and-fix} was limited in 2h, knowing that only in instances 3
and 6 this limiting was reached. It was started from this idea for such choice, in a way to make that the total time of execution of relax-and-fix was less or equal to the execution time of the nonrelaxed model, to evaluate, in the generated instances, the efficiency of the heuristic. To measure this efficiency, two gaps were calculated, as expressions (28) and (29).

\[
gap_2 = \left( \frac{\text{solution relax and fix}}{\text{Inferior limiting}} \right) \cdot 100
\] (28)

\[
gap_3 = \left( \frac{\text{solution relax and fix}}{\text{limited time solution}} \right) \cdot 100
\] (29)

Table 3 shows the results found after the computational tests made for each one of the proposed instances.

<table>
<thead>
<tr>
<th>Instance</th>
<th>( \text{gap}_2 )</th>
<th>( \text{gap}_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>17.96%</td>
<td>-8.65%</td>
</tr>
<tr>
<td>2</td>
<td>23.77%</td>
<td>8.17%</td>
</tr>
<tr>
<td>3</td>
<td>58.46%</td>
<td>24.93%</td>
</tr>
<tr>
<td>4</td>
<td>15.78%</td>
<td>-10.34%</td>
</tr>
<tr>
<td>5</td>
<td>12.68%</td>
<td>-1.52%</td>
</tr>
<tr>
<td>6</td>
<td>29.40%</td>
<td>2.02%</td>
</tr>
</tbody>
</table>

In the tests performed so far, it can be noted, due to the gaps found in relation to the obtained solutions for the nonrelaxed problem executed for a limited time of 2h (\( \text{gap}_3 \)), that the heuristic relax-and-fix was able to generate good solutions in practically all instances (except for instance 3). The negative value for the \( \text{gap}_3 \) means that the relax-and-fix found a better solution than the nonrelaxed model, executed for a time of 2h.

7 Conclusions

The main objective of this work was to develop a multistage mathematical model, adapted to the presented problematic, wherein, by the features of the developed model, it is possible to realize its generality, and due to this fact, it is expected to contribute for the development of new research in a way that the presented model can serve as general basis, being adaptable for use in other productive environments.

Regarding to the resolution of the model strategies, the relax-and-fix proved to be a promising strategy of resolution, for its both easy implementation and for the good results presented when compared to the inferior limitings.
This way it is expected, that this mathematical model can serve as a basis for industrial applications, allowing the reduction on production costs, in a way to bring about as consequence benefits to the environment, due to lower consumption of productive resources used to produce the established demands.

Acknowledgements. To the Federal Institute of Santa Catarina and to the Federal University of Paraná - Brazil, for the granted support for the research developing.

References


Received: February 27, 2015; Published: March 20, 2015