

## Hybrid procedure based on Bounded Dynamic Programming and Linear Programming for solving a variant of the MMSP-W

J. Bautista\* A. Cano\* R. Alfaro\* C. Batalla\*

\* *Universitat Politècnica de Catalunya, Avda. Diagonal 647, 7th floor, 08028 Barcelona Spain (Tel: 34-934010769; e-mail: {joaquin.bautista; alberto.cano-perez; rocio.alfaro; cristina.batalla}@upc.edu).*

---

**Abstract:** In this paper, we propose a hybrid procedure based on Bounded Dynamic Programming (*BDP*) and linear programming to solve the Mixed-Model Sequencing Problem with Workload Minimization (*MMSP-W*), with serial workstations, free interruption of the operations and production mix restrictions. We performed a computational experiment with 225 instances from the literature. The results of our proposal are compared with those obtained through the Gurobi solver and previous procedures.

*Keywords:* Automobile industry; Manufacturing systems; JIT manufacturing; Sequences; Overload; Scheduling algorithms; Dynamic programming; Linear programming; Heuristics.

---

### 1. INTRODUCTION

In mixed-model manufacturing lines, which are common in Just-in-time (*JIT*) and Douki Seisan (*DS*) ideologies, several variants of one or more products can be handled. This flexibility determines the order in which the units are treated to drastically reduce intermediate stocks and to capitalize on the time available for manufacturing.

In these settings, we can choose two basic categories of objectives (Boysen et al., 2009): (A) Minimize the work overload or lost work that may appear due to production programs with mixed products; and (B) minimize the levels of stocks in the system in accordance with *JIT* manufacturing method.

For category A, in addition to a relative focus on maximizing the total amount of work completed (Yano and Rachamadugu, 1991), the excess effort that must be applied over time for certain operations can be modulated.

Using this perspective, sequencing problems can be grouped into three categories: (1) Mixed-model sequencing; (2) Car sequencing; and (3) Level scheduling.

Within this framework, Boysen et al. (2009) provided an up-to-date review of the literature. The present study can be placed in category A.1/3, and focuses on the minimization of the total work overload in mixed-model sequences maintaining the production mix in the working day.

Overload, or excess effort, is a measurement, in units of time, of work that cannot be completed at the standard work pace established, within the time granted to the workstations (cycle). This overload may arise when the processing time of a unit at a workstation is greater than the cycle time (Yano and Rachamadugu, 1991), although there may be a certain amount of play associated with extended cycles, which is called the length of the workstation or the time window.

When faced with a foreseeable workstation overload, at least three types of measures can be taken: (I) stop the line and complete the pending work using reinforcements (Okamura and Yamashina, 1979; Rabbani et al., 2011); (II) let the unit pass and finish the pending work in a final line at a later time. (Yano and Rachamadugu, 1991; Bolat, 2003; Tsai, 1995); and (III) increase productive activity above the standard, using the assistance of reinforcement operators (Cevikcan and Durmusoglu, 2011) or previously programmed robotized systems. The present study considered measures in categories II and III for handling work overloads.

On the other hand, the Level scheduling problems class focuses on obtaining regular sequences in production and consumption of components, among them are: (1) Product Rate Variation (*PRV*), which is used to preserve the production mix (Miltenburg, 1989) and (2) Output Rate Variation (*ORV*), based on the manner of sequencing the mixed products units, used at Toyota plants to maintain a constant consumption of components over time (Monden, 1983). The preservation of the production mix is a desirable property in *JIT* (Toyota) and *Douki Seisan* (Nissan) philosophies. We will introduce this property to the problem.

Mixed-Model Sequencing Problem with Workload Minimization (*MMSP-W*) is an NP-hard problem (Yano and Rachamadugu, 1991). Given the complexity of the problem, and the size of the case study presented in Bautista and Cano (2011) related to Nissan Barcelona powertrain plant, our objective is to find a computationally competitive procedure to solve the problem. For this reason, in this paper, we proposed a procedure based on Bounded Dynamic Programming (*BDP*). This procedure combines features of dynamic programming with features of branch and bound algorithms. The principles of the *BDP* have been described by Bautista et al. (1996).

Our proposal contains the following: (1) a model for the problem; (2) a new procedure based on dynamic

programming to solve this problem, *BDP-2*, that uses linear programming to obtain bounds for the problem; (3) a mathematical model to obtain the overload of a given subsequence to use it as a part of the lower bound of the problem; (4) reduction of the search space of the procedure through pseudo-dominances; and (5) a computational experiment with reference instances from the literature to compare the results offered by *BDP-2* with those offered by integer linear programming and previous procedures.

This paper is organized as follows. Section 2 presents a model for the *MMSP-W* with serial workstations, unrestricted interruption of the operations and production mix restrictions. Section 3 shows an illustrative example. Section 4 describes the basic elements and the application of the proposed *BDP* procedure. Section 5 describes the computational experiments with reference instances from the literature. Finally, Section 6 shows the conclusions of the study.

## 2. MODEL FOR THE PROBLEM

The *MMSP-W* consists of sequencing  $T$  products, of which  $d_i$  are of type  $i$  ( $i = 1, \dots, |I|$ ). A unit of product type  $i$  requires to each processor (operator, robot, etc...) of the workstation  $k$  ( $k = 1, \dots, |K|$ ) a standard processing time,  $p_{i,k}$ . The standard time assigned to each processor to work on any product unit is the cycle time  $c$ . When a cycle ends at the workstation  $k$ , the processor can work on the product in progress in an additional positive time  $l_k - c$ , being  $l_k$  the time window.

When it is not possible to complete all of the work required by the demand plan, overload is generated. The objective of the problem is to maximize the total work performed, which is equivalent to minimizing the total overload generated (see Theorem 1 in Bautista and Cano, 2011).

For the *MMSP-W* with serial workstations, unrestricted interruption of the operations, production mix restrictions (*pmr*) and work overload minimization, we take as reference the *M4U3* model, proposed by Bautista et al. (2012a). The parameters and variables of the extended model *M4U3\_pmr* are presented below.

### Parameters

$K$	Set of workstations ( $k = 1, \dots,  K $ )
$b_k$	Number of homogeneous processors at workstation $k$
$I$	Set of product types ( $i = 1, \dots,  I $ )
$d_i$	Programmed demand of product type $i$
$p_{i,k}$	Processing time required by a unit of type $i$ at workstation $k$ for each homogeneous processor (at normal activity)
$T$	Total demand; obviously, $\sum_{i=1}^{ I } d_i = T$
$t$	Position index in the sequence ( $t = 1, \dots, T$ )
$c$	Cycle time, the standard time assigned to workstations to process any product unit
$l_k$	Time window, the maximum time that the workstation $k$ is allowed to work on any product unit, where $l_k - c > 0$ is the maximum time that the work in process is held at workstation $k$

### Variables

$x_{i,t}$	Binary variable equal to 1 if a product unit $i$ ( $i = 1, \dots,  I $ ) is assigned to the position $t$ ( $t = 1, \dots, T$ ) of the sequence, and to 0 otherwise
$s_{k,t}$	Start instant of the operation in $t^{\text{th}}$ unit of the sequence of products at workstation $k$ ( $k = 1, \dots,  K $ )

$\hat{s}_{k,t}$	Positive difference between the start instant and the minimum start instant of the $t^{\text{th}}$ operation at workstation $k$ . $\hat{s}_{k,t} = [s_{k,t} - (t+k-2)c]^+$ (with $[x]^+ = \max\{0, x\}$ )
$v_{k,t}$	Processing time applied by the $t^{\text{th}}$ unit of the sequence of products at workstation $k$ for each homogeneous processor (at normal activity).
$w_{k,t}$	Overload generated for the $t^{\text{th}}$ unit of the product sequence at workstation $k$ for each homogeneous processor (at normal activity); measured in time
$\rho_{k,t}$	Processing time required by the $t^{\text{th}}$ unit of the sequence of products at workstation $k$ for each homogeneous processor (at normal activity)

Model *M4U3\_pmr*:

$$\text{Min } W = \sum_{k=1}^{|K|} (b_k \sum_{t=1}^T w_{k,t}) \Leftrightarrow \text{Max } V = \sum_{k=1}^{|K|} (b_k \sum_{t=1}^T v_{k,t}) \quad (1)$$

Subject to:

$$\sum_{t=1}^T x_{i,t} = d_i \quad \forall i = 1, \dots, |I| \quad (2)$$

$$\sum_{i=1}^{|I|} x_{i,t} = 1 \quad \forall t = 1, \dots, T \quad (3)$$

$$v_{k,t} + w_{k,t} = \sum_{i=1}^{|I|} p_{i,k} x_{i,t} \quad \forall k = 1, \dots, |K|; \forall t = 1, \dots, T \quad (4)$$

$$\hat{s}_{k,t} \geq \hat{s}_{k,t-1} + v_{k,t-1} - c \quad \forall k = 1, \dots, |K|; \forall t = 2, \dots, T \quad (5)$$

$$\hat{s}_{k,t} \geq \hat{s}_{k-1,t} + v_{k-1,t} - c \quad \forall k = 2, \dots, |K|; \forall t = 1, \dots, T \quad (6)$$

$$\hat{s}_{k,t} + v_{k,t} \leq l_k \quad \forall k = 1, \dots, |K|; \forall t = 1, \dots, T \quad (7)$$

$$\hat{s}_{k,t} \geq 0 \quad \forall k = 1, \dots, |K|; \forall t = 1, \dots, T \quad (8)$$

$$v_{k,t} \geq 0 \quad \forall k = 1, \dots, |K|; \forall t = 1, \dots, T \quad (9)$$

$$w_{k,t} \geq 0 \quad \forall k = 1, \dots, |K|; \forall t = 1, \dots, T \quad (10)$$

$$x_{i,t} \in \{0, 1\} \quad \forall i = 1, \dots, |I|; \forall t = 1, \dots, T \quad (11)$$

$$\hat{s}_{11} = 0 \quad (12)$$

$$\sum_{\tau=1}^t x_{i,\tau} \geq \left\lceil t \cdot \frac{d_i}{T} \right\rceil \quad \forall i = 1, \dots, |I|; \forall t = 1, \dots, T \quad (13)$$

$$\sum_{\tau=1}^t x_{i,\tau} \leq \left\lfloor t \cdot \frac{d_i}{T} \right\rfloor \quad \forall i = 1, \dots, |I|; \forall t = 1, \dots, T \quad (14)$$

In the model, the equivalent objective functions (1) are represented by the total overload ( $W$ ) and total work performed ( $V$ ). Constraint (2) requires that the programmed demand to be satisfied. Constraint (3) indicates that only one product unit can be assigned to each position of the sequence. Constraint (4) establishes the relation between the processing times applied to each unit at each workstation and the overload generated in each unit at each workstation. Constraints (5)-(8) constitute the set of relative start instants of the operations at each station and the processing times applied to the products for each processor. Constraints (9) and (10) indicate that the processing times applied to the products and the generated overloads, respectively, are not negative. Constraint (11) requires the assigned variables to be binary. Constraint (12) fixes the start of operations. The constraints (13) and (14) are those that incorporate the preservation property of the production mix desired in *JIT* (Toyota) and *Douki Seisan* (Nissan) philosophies.

Also, in this work we will use to measure the non-regularity of a sequence the next quadratic function:

$$\Delta_Q(X) = \sum_{i=1}^T \sum_{t=1}^{|I|} \left( X_{i,t} - t \cdot \frac{d_i}{T} \right)^2 \quad (15)$$

where  $X_{i,t} = \sum_{\tau=1}^t x_{i,\tau}$  ( $\forall i = 1, \dots, |I|; \forall t = 1, \dots, T$ ) is the cumulative production.

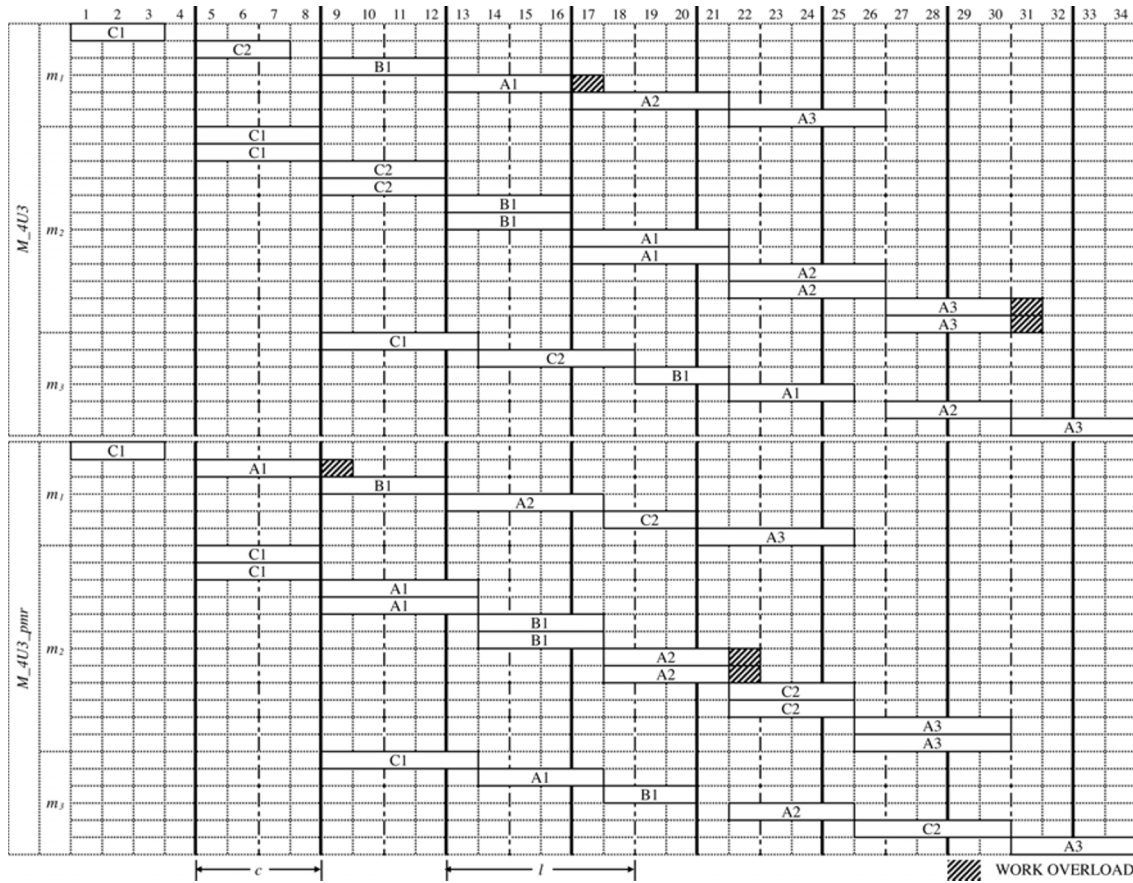


Fig. 1. Gantt chart for the optimum solutions for the example provided by  $M_{4U3}$  (top) and  $M_{4U3\_pmr}$  (bottom).

### 3. AN ILLUSTRATIVE EXAMPLE

To illustrate the model formulated above, we present the following example: There are six units of product ( $T=6$ ), of which three are type  $A$ , one is type  $B$  and two are type  $C$ , with a total work required  $V_0=104$ . The units are processed at three workstations ( $|K|=3$ ) with different numbers of processors ( $b_k$ ); the processing times for each processor (at normal activity) for each type of unit  $i$  ( $A, B, C$ ) at each workstation  $k$  ( $m_1, m_2, m_3$ ) are listed in Table 1.

Table 1. Processing times ( $p_{i,k}$ ), number of homogeneous processors ( $b_k$ ) and total work ( $V_0$ ) required by each type of unit at each workstation

	A ( $d_A=3$ )	B ( $d_B=1$ )	C ( $d_C=2$ )	$b_k$
$m_1$	5	4	3	1
$m_2$	5	4	4	2
$m_3$	4	3	5	1
<b>Total</b>	$19 (V_0(A)=57)$	$15 (V_0(B)=15)$	$16 (V_0(C)=32)$	$V_0=104$

Furthermore,  $c = 4$  (cycle time) and  $l_k = 6$  for  $k=1, \dots, 3$  (length of the workstation or time window).

Fig. 1 shows a Gantt diagram of the optimal solutions offered by models  $M4U3$  (top) and  $M_{4U3\_pmr}$  (bottom). The sequence of products that presents the minimum total overload for  $M4U3$  is  $C-C-B-A-A-A$ . The total work performed is  $V=101$ , and the overload, which is concentrated between workstations  $m_1$  and  $m_2$ , is  $W=3$  (the grey area in Fig.1). The non-regularity for  $M_{4U3}$  is 9.05. The sequence

of products that presents the minimum total overload for  $M4U3\_pmr$  is  $C-A-B-A-C-A$  (affected by the production mix restrictions). The total work performed is  $V=101$ , and the overload, which is concentrated between workstations  $m_1$  and  $m_2$ , is  $W=3$ . The non-regularity for  $M_{4U3\_pmr}$  is 2.05.

### 4. BDP FOR THE MMSP-W WITH PMR

This section presents the basic elements of the  $BDP$  procedure ( $BDP-2$ ) applied to the resolution on  $MMSP-W$  with serial workstations, unrestricted interruption of the operations and production mix restrictions.

#### 4.1 Bounds for the problem

Given a vertex of the stage  $t$ , reached through a partial sequence  $\pi(t) = \{\pi_1, \pi_2, \dots, \pi_t\}$ , the overall bound for  $W$  and a partial bound for the complement  $R(\pi(t))$  associated to the sequence or segment  $\pi(t)$  can be determined according to the schema presented in Fig. 2.

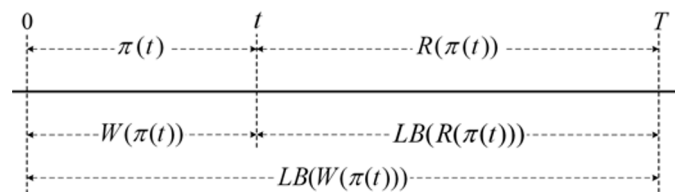


Fig. 2. Bound scheme for a partial sequence  $\pi(t)$ .

To obtain the overload associated to  $\pi(t)$ , in each stage of the procedure we use a mathematical model. Given the subsequence  $\pi(t) = (\pi_1, \pi_2, \dots, \pi_t)$  of products, the processing times for each workstation  $k$  and each cycle  $\tau$ ,  $p_{\pi_\tau k}$ , are known. We can define a mathematical model where the assignment variables has been removed:

$$\text{Min } W(\pi(t)) = \sum_{k=1}^{|K|} \left( b_k \sum_{\tau=1}^t w_{k,\tau} \right) \quad (16)$$

Subject to:

$$\rho_{k\tau} = p_{\pi_\tau k} \quad k = 1, \dots, |K|; \tau = 1, \dots, t \quad (17)$$

$$\rho_{k\tau} - w_{k\tau} \geq 0 \quad k = 1, \dots, |K|; \tau = 1, \dots, t \quad (18)$$

$$\hat{s}_{k\tau} \geq \hat{s}_{k\tau-1} + \rho_{k\tau-1} - w_{k\tau-1} - c \quad k = 1, \dots, |K|; \tau = 2, \dots, t \quad (19)$$

$$\hat{s}_{k\tau} \geq \hat{s}_{k-1\tau} + \rho_{k-1\tau} - w_{k-1\tau} - c \quad k = 2, \dots, |K|; \tau = 1, \dots, t \quad (20)$$

$$\hat{s}_{k\tau} + \rho_{k\tau} - w_{k\tau} \leq l_k \quad k = 1, \dots, |K|; \tau = 1, \dots, t \quad (21)$$

$$\hat{f}_k \geq \hat{s}_{k1} + \rho_{k1} - w_{k1} \quad k = 1, \dots, |K| \quad (22)$$

$$\hat{f}_k \geq c \quad k = 1, \dots, |K| \quad (23)$$

$$\hat{s}_{k\tau} \geq 0 \quad k = 1, \dots, |K|; \tau = 1, \dots, t \quad (24)$$

$$w_{k\tau} \geq 0 \quad k = 1, \dots, |K|; \tau = 1, \dots, t \quad (25)$$

$$\hat{s}_{11} = 0 \quad (26)$$

where  $\hat{f}_k$  represents the relative finishing instants at workstation  $k$  of the  $t$  products sequenced. The result of the proposed mathematical model corresponds to  $W(\pi(t))$ .

To obtain a bound of the overload, ( $LB(R(\pi(t)))$ ), associated to the complement  $R(\pi(t))$ , we use the combination of three lower bounds.

Let  $\bar{d}_i$  as the pending demand of the product  $i$  when  $t$  units of product has been sequenced. We can define the available time at workstation  $k$ , for each homogeneous processor at normal activity, as:

$$TD_k = (T - t - 1) \cdot c + l_k \quad k = 1, \dots, |K| \quad (27)$$

And the pending work at workstation  $k$  as:

$$TP_k = \sum_{i=1}^{|I|} p_{i,k} \cdot \bar{d}_i \quad k = 1, \dots, |K| \quad (28)$$

Using (27) and (28) we can define  $LB1(t)$  as:

$$LB1(t) = \sum_{k=1}^{|K|} b_k \cdot [TP_k - TD_k]^+ \quad (29)$$

For the second bound, we define a bound for each product ( $i = 1, \dots, |I|$ ):

$$LB2(i) = \left[ \sum_{k=1}^{|K|} b_k p_{i,k} - c \sum_{k=1}^{|K|} b_k - b_{|K|} (l_{|K|} - c) \right]^+ \quad (30)$$

Using (30) we can define  $LB2(t)$  as:

$$LB2(t) = \sum_{i=1}^{|I|} LB2(i) \cdot \bar{d}_i \quad (31)$$

To define the third bound, we use a mathematical model. Let  $w_{k,i}$  as the overloads that a product  $i$  can generate, in the best situations, at workstation  $k$ . We can define:

$$\text{MIN } LW(i) = \sum_{k=1}^{|K|} b_k \cdot w_{k,i} \quad (32)$$

Subject to:

$$\hat{s}_{k,i} \geq \hat{s}_{k-1,i} + p_{k-1,i} - w_{k-1,i} - c \quad k = 2, \dots, |K| \quad (33)$$

$$\hat{s}_{k,i} + p_{k,i} - w_{k,i} \leq l_k \quad k = 1, \dots, |K| \quad (34)$$

$$p_{k,i} - w_{k,i} \geq 0 \quad k = 1, \dots, |K| \quad (35)$$

$$\hat{s}_{k,i} \geq 0 \quad k = 1, \dots, |K| \quad (36)$$

$$w_{k,i} \geq 0 \quad k = 1, \dots, |K| \quad (37)$$

$$\hat{s}_{1,i} = 0 \quad (38)$$

Once determined  $LW(i)$ , we can determine  $LB3(t)$  as:

$$LB3(t) = \sum_{i=1}^{|I|} LW(i) \cdot \bar{d}_i \quad (39)$$

And finally, to determine  $LB(R(\pi(t)))$ , we use:

$$LB(R(\pi(t))) = \max\{LB1(t), LB2(t), LB3(t)\} \quad (40)$$

Then, the total bound for the problem will be:

$$LB(W(\pi(t))) = W(\pi(t)) + LB(R(\pi(t))) \quad (41)$$

#### 4.2 Graph associated with the problem

Similar to Bautista and Cano (2011) we can build a linked graph without loops or direct cycles of  $T + 1$  stages. The set of vertices in level  $t$  ( $t = 0, \dots, T$ ) will be noted as  $J(t)$ .  $J(t, j)$  ( $j = 1, \dots, |J(t)|$ ) being a vertex of level  $t$ , which is defined by the tuple  $(\bar{q}(t, j), \pi(t, j), LB(W(\pi(t, j))), \Delta_Q(\bar{q}(t, j)))$ , where:

- $\bar{q}(t, j) = (q_1(t, j), \dots, q_{|I|}(t, j))$  is the vector of demand satisfied.
- $\pi(t, j)$  is the sequence of  $t$  units of product associated to the vertex.
- $LB(W(\pi(t, j)))$  is a lower bound of the total overload, generated by the sequence  $\pi(t, j)$ , obtained using  $W(\pi(t))$  and  $LB(R(\pi(t)))$ .
- $\Delta_Q(\bar{q}(t, j))$  is the non-regularity  $\Delta_Q(X)$  of the sequence of products  $\pi(t, j)$ , calculated as follows:

$$\Delta_Q(\bar{q}(t, j)) = \sum_{\tau=1}^t \sum_{i=1}^{|I|} \left( X_{i,\tau} - \tau \cdot \frac{d_i}{T} \right)^2 \quad (42)$$

where  $X_{i,\tau}$  is the cumulative production.

The vertex  $J(t, j)$  has the following properties:

$$\sum_{i=1}^{|I|} q_i(t, j) = t \quad (43)$$

$$\left\lfloor t \cdot \frac{d_i}{T} \right\rfloor \leq q_i(t, j) \leq \left\lceil t \cdot \frac{d_i}{T} \right\rceil \quad \forall i \in I \quad (44)$$

In short, a vertex  $J(t, j)$  will be represented as follows:

$$J(t, j) = \left\{ (t, j), \bar{q}(t, j), \pi(t, j), LB(W(\pi(t, j))), \Delta_Q(\bar{q}(t, j)) \right\} \quad (45)$$

At level 0 of the graph, there is only one  $J(0)$  vertex. Initially, we may consider that at level  $t$ ,  $J(t)$  contains the vertices associated to all of the sub-sequences that can be built with  $t$  products that satisfy properties (43) and (44). However, it is easy to reduce the cardinal that  $J(t)$  may present a priori, establishing the following definition of pseudo-dominance: given the sequences  $\pi(t, j_1)$  and  $\pi(t, j_2)$  associated to the vertices  $J(t, j_1)$  and  $J(t, j_2)$ , then  $\pi(t, j_1)$  pseudo-dominates  $\pi(t, j_2)$  if:

$$\pi(t, j_1) \prec \pi(t, j_2) \Leftrightarrow \left\{ \begin{array}{l} [\bar{q}(t, j_1) = \bar{q}(t, j_2)] \text{ and} \\ [LB(W(\pi(t, j_1))) \leq LB(W(\pi(t, j_2)))] \text{ and} \\ [\Delta_Q(\bar{q}(t, j_1)) \leq \Delta_Q(\bar{q}(t, j_2))] \end{array} \right\} \quad (46)$$

The reduction of  $J(t)$  through the pseudo-dominances defined in (46) cannot guarantee the optimality of the solutions.

### 4.3 The use of BDP

For this study, we used a procedure based on *BDP*. This procedure combines features of dynamic programming (determination of extreme paths in graphs) with features of branch and bound algorithms. The principles of *BDP* have been described by Bautista et al. (1996). Previous work on similar approaches has been done by Bautista and Cano (2011). The procedure is described below (see details on Bautista and Cano, 2011):

---

#### BDP-2 – MMSPW

---

*Input:*  $T, |I|, |K|, d_i(\forall i), l_k, b_k(\forall k), p_{ik}(\forall i, \forall k), c, Z_0, H$

*Output:* list of sequences obtained by *BDP*

```

0 Initialization:  $t = 0; LBZ_{\min} = \infty$ 
1 Generate_model();
2 While ( $t < T$ ) do
3      $t = t + 1$ 
4     Add_constraints( $t$ )
5     While (list of consolidated vertices in stage  $t-1$  not empty) do
6         Select_vertex ( $t$ )
7         Develop_vertex ( $t$ )
8         Filter_vertices ( $Z_0, H, LBZ_{\min}$ )
9     end while
10 End_stage ()
11 end while
end BDP - MMSPW

```

---

In the procedure appear the following functions:

- **Generate\_model ():** Generates the initial model to obtain  $W(\pi(t))$ , for  $t=0$ .
- **Add\_constraints ( $t$ ):** adds, to the existing model, the new constraints associated to the new stage  $t$ .
- **Select\_vertex ( $t$ ):** selects, following a non-decreasing ordering of the  $LB(W(\pi(t-1, j)))$  values, one of the vertices consolidated in stage  $t-1$ .
- **Develop\_vertex ( $t$ ):** develops the selected vertex in previous function adding a new product unit with pending demand. The vertices that do not satisfy the properties (43) and (44) are not generated.
- **Filter\_vertices ( $Z_0, H, LBZ_{\min}$ ):** chooses, from all the vertices developed in the previous function, a maximum number  $H$  of the most promising vertices (according to the lowest values of the lower bound  $LB(W(\pi(t, j)))$ ), and removing those vertices in which their lower bound is greater than  $Z_0$  (known initial solution) and those pseudo-dominated as defined in (46).
- **End\_stage ():** consolidates the most promising vertices in stage  $t$  ( $H$  vertices as maximum).

### 5. COMPUTATIONAL EXPERIMENT

For the test operations of the *BDP-2* procedure, 225 instances from the literature were used. These instances were built from 45 production programs and 5 processing times structures, composed by four product types ( $|I|=4$ ) and four workstations ( $|K|=4$ ). These instances can be found in Cano et al. (2010).

The solutions offered by the *BDP-2* procedure proposed were obtained under the following conditions and features: (1) *BDP-2* procedure programmed in C++, using gcc v4.2.1, running on an Apple Macintosh iMac computer with an Intel Core i7 2.93 GHz processor and 8 GB RAM using MAC OS

X 10.6.7 (not using any type of parallel code; therefore, the computer can be considered as a single 2.93 GHz processor); (2) four windows width ( $H$ ) were used, with values 1, 6, 16 and 32; (3) the initial solution  $Z_0$  for each window width was the solution obtained by *BDP-2* with the previous window width, except in the case  $H = 1$ , where  $Z_0$  was established as  $\infty$ ; and (4) to calculate the lower bounds,  $LB(W(\pi(t, j)))$ , of the overload associated to each vertex in the *BDP-2* procedure, the solver Gurobi v4.6.1 was used, solving the linear program associated to  $W(\pi(t))$ .

The results for the 225 instances were obtained using the following procedures solving the model *M\_4U3\_pmr*: (1) *Gurobi*: Solving the model using Solver Gurobi (Bautista et al., 2012a); (2) *BDP-1*: A procedure based on *BDP* (Bautista et al., 2012b); and (3) *BDP-2*: The *BDP* procedure described in this document.

To study the behavior of the three procedures to solve *M\_4U3\_pmr*, taking into account the set of instances  $E$  and designating as  $\hat{S}_G(\varepsilon)$  (optimal for  $W$ ),  $\hat{S}_1(\varepsilon)$  and  $\hat{S}_2(\varepsilon)$  the best solutions found for the instance  $\varepsilon \in E$ , through the procedures *Gurobi*, *BDP-1* and *BDP-2*, respectively, we define the following relative percentage deviation:

$$RPD_1(f, \varepsilon) = \frac{f(\hat{S}_G(\varepsilon)) - f(\hat{S}_2(\varepsilon))}{f(\hat{S}_G(\varepsilon))} \quad f \in \{W, \Delta_Q(X)\}; \varepsilon \in E \quad (47)$$

$$RPD_2(f, \varepsilon) = \frac{f(\hat{S}_1(\varepsilon)) - f(\hat{S}_2(\varepsilon))}{f(\hat{S}_1(\varepsilon))} \quad f \in \{W, \Delta_Q(X)\}; \varepsilon \in E \quad (48)$$

The results obtained by the experiment are collected in tables 2 and 3.

Table 2. Minimum, maximum and average *CPU* times needed to obtain the solutions for the 225 instances using *Gurobi*, *BDP-1* and *BDP-2*.

	<i>Gurobi</i>	<i>BDP-1</i>	<i>BDP-2</i>
<i>CPU</i> min	0.03	0.06	0.04
<i>CPU</i> max	110.53	2.72	0.61
$\overline{CPU}$	11.79	0.78	0.20

Table 3.  $RPD_1$  and  $RPD_2$  values by structures, blocks and average (225 instances), of the solutions, given by *Gurobi*, *BDP-1* and *BDP-2*.

	$W$		$\Delta_Q(X)$	
	$RPD_1$	$RPD_2$	$RPD_1$	$RPD_2$
<i>E1</i>	-3.15	-3.05	2.14	7.01
<i>E2</i>	-1.86	-1.84	10.84	9.60
<i>E3</i>	-0.49	-0.49	13.07	12.23
<i>E4</i>	-0.08	-0.08	4.23	3.85
<i>E5</i>	-1.65	-1.64	2.89	2.99
<i>B1</i>	-0.86	-0.86	-1.23	0.84
<i>B2</i>	-0.58	-0.58	1.80	3.77
<i>B3</i>	-1.62	-1.62	6.09	6.50
<i>B4</i>	-1.93	-1.81	1.71	3.14
<i>B5</i>	-1.63	-1.60	10.13	9.87
<i>Average</i>	-1.45	-1.42	6.63	7.13

As show in Table 2, we obtained a substantial improvement in the average *CPU* time of the new procedure *BDP-2* when is compared to the other two procedures. Specifically, *BDP-1*

requires 4 times longer than *BDP-2* to achieve the best solutions of the 225 instances. Also, *Gurobi* requires 59 times more average *CPU* time than *BDP-2*. For maximum *CPU* time, *BDP-2* improves 298 and 7 times compared to *Gurobi* and *BDP-1*, respectively. Regarding the minimum *CPU* times there is no difference between the three procedures.

In Table 3 we can see an improvement of the regularity of the production obtained by the *BDP-2* procedure. Specifically, *BDP-2* reduces, on average, the non-regularity function,  $\Delta_Q(X)$ , in a 6.63% versus *Gurobi* results, and a 7.13% versus *BDP-1*. However, on average the overload,  $W$ , worsens using *BDP-2* in 1.45% and in 1.42% versus *Gurobi* and *BDP-1*, respectively. These improvements in production regularity and worsens in overload also occur monotonically for the instances grouped by structures and blocks. The more significant improvements in regularity occur in structure *E3* and block *B5*, while the worsening in work overloads, while being much more balanced between the blocks and structures, are more relevant in *E1* and *B4*.

## 6. CONCLUSIONS

We have proposed a new procedure based on the *BDP*, the *BDP-2*, for the *MMSP-W* problem that minimizes the total overload or maximizes the total work completed, taking into account serial workstations, parallel processors, free interruption of the operations and restrictions to preserve the production mix in the manufacturing sequence.

The proposed procedure uses global bounds based on linear programming. A mathematical program that minimizes the work overload given a subsequence of operations for any instant  $t$  has been formulated. In addition, the proposed procedure incorporates pseudo-dominances between partial solutions to limit the search space. These pseudo-dominances take into account the preservation of the production mix in the partial solutions.

The results obtained for the 225 instances using the proposed procedure, *BDP-2*, are compared with those offered (for model *M\_4U3\_pmr*) by the *Gurobi* solver (Bautista et al., 2012a) and another procedure *BDP-1* (Bautista et al., 2012b). *BDP-2* improves, on average, the *CPU* time and regularity of production given by *Gurobi* (98% reduction in *CPU* time and 6.63% in non-regularity) and *BDP-1* (80% in *CPU* time and 7.13% in non-regularity). However, *BDP-2* worsens on average overload in less than 1.5% compared to the other two procedures.

As future work we will: (1) set properties to be satisfied by sequences related to the preservation of the production mix in order to reduce the search space of solutions with the *BDP* procedure; and (2) design and apply new procedures using the properties described in point (1) to a set of instances associated to a industrial case study in the automotive sector.

## ACKNOWLEDGMENT

The authors greatly appreciate the collaboration of Nissan Spanish Industrial Operations (NSIO). This work was funded

by project PROTHIUS-III, DPI2010-16759, including EDRF funding from the Spanish government.

## REFERENCES

- Bautista, J., Companys, R., and Corominas, A. (1996) Heuristics and exact algorithms for solving the Monden problem. *European Journal of Operational Research*, 88(1), 101-113.
- Bautista, J., and Cano, A. (2011) Solving mixed model sequencing problem in assembly lines with serial workstations with work overload minimisation and interruption rules. *European Journal of Operational Research*, 210(3), 495-513.
- Bautista, J., Cano, A., and Alfaro, R. (2012a) Modeling and solving a variant of the mixed-model sequencing problem with work overload minimisation and regularity constraints. An application in Nissan's Barcelona Plant. *Expert Systems with Applications*, 39(12), 11001-11010.
- Bautista, J., Cano, A., and Alfaro, R. (2012b) Modeling and Solving a Variant of MMSP-W Problem with Production Mix Restrictions. *Industrial Engineering: Overcoming the Crisis. Book of Abstracts. ISBN: 978-84-938642-4-8, pp 231, VI International Conference on Industrial Engineering and Industrial Management*.
- Bolat, A. (2003) A mathematical model for sequencing mixed models with due dates. *International Journal of Production Research*, 41(5), 897-918.
- Boysen, N., Fliedner, M., and Scholl, A. (2009) Sequencing mixed-model assembly lines: survey, classification and model critique. *European Journal of Operational Research*, 192(2), 349-373.
- Cano-Belmán, J., Ríos-Mercado, R.Z. and Bautista, J. (2010) A scatter search based hyper-heuristic for sequencing a mixed-model assembly line. *Journal of Heuristics*, 16 (6), 749-770.
- Cevikcan, E. and Durmusoglu, M.B. (2011) Minimising utility work and utility worker transfers for a mixed-model assembly line. *International Journal of Production Research*, 49 (24), 7293-7314.
- Miltenburg, J. (1989) Scheduling Mixed-Model Assembly Lines for Just-In-Time Production Systems. *Management Science*, 35(2), 192-207.
- Monden, Y. (1983) Toyota Production System. Industrial Engineering and Management Press, Norcross, Georgia.
- Okamura, K., and Yamashina, H. (1979) A heuristic algorithm for the assembly line model-mix sequencing problem to minimize the risk of stopping the conveyor. *International Journal of Production Research*, 17(3), 233-247.
- Rabbani, M., Radmehr, F. and Manavizadeh, N. (2011) Considering the conveyer stoppages in sequencing mixed-model assembly lines by a new fuzzy programming approach. *International Journal of Advanced Manufacturing Technology*, 54 (5-8), 775-788.
- Tsai, L.H. (1995) Mixed-model sequencing to minimize utility work and the risk of conveyor stoppage. *Management Science*, 41(3), 485-495.
- Yano, C.A., and Rachamadugu, R. (1991) Sequencing to minimize work overload in assembly lines with product options. *Management Science*, 37(5), 572-586.