

# A Hybrid Dynamic Programming for solving a Mixed-Model Sequencing Problem with Production Mix Restriction and free interruptions

Joaquín Bautista<sup>1</sup>, Alberto Cano, Rocío Alfaro-Pozo<sup>1</sup>

<sup>1</sup>Research Group OPE-PROTHIUS. Universitat Politècnica de Catalunya.  
ETSEIB· Avda. Diagonal, 647, 7th floor, 08028 Barcelona, Spain.  
{joaquin.bautista; rocio.alfaro}@upc.edu

**Abstract.** In this article, we propose a hybrid procedure based on bounded dynamic programming (BDP) assisted by linear programming (LP) to solve the mixed-model sequencing problem with workload minimization (MMSP-W) with serial workstations, free interruption of the operations and with production mix restrictions. We performed a computational experiment with 23 instances related to a case study of the Nissan powertrain plant located in Barcelona. The results of our proposal are compared with those obtained by Mixed Integer Linear Programming (MILP).

**Keywords:** Mixed model sequencing; Dynamic programming; Mixed integer linear programming; Hybrid metaheuristics; Industrial application.

## 1 Preliminaries

Product-oriented manufacturing systems are very common in production environments related to the automotive sector. In such systems, the manufacturing process of a product (engines, stamp forging, body welding, body painting and trim and chassis lines, for example) is conceived as a set of consecutive stages or manufacturing processes (due to the product orientation) that add value from raw materials to the final product (automobile).

This production type, which is product-oriented, culminates in flexible manufacturing systems composed by cells and modules or workstations arranged in series in assembly lines. In this last type of system, in addition to the line balancing problems, we can encounter the batch or product-unit sequencing problems, where the units are not completely identical, and their manufacture may require different consumption of components and different resource use at each manufacturing stage.

When the processing times of mixed products differ slightly at each stage, we are faced with problems similar to those that are known in the literature under the name of mixed product sequencing (homogeneous units). In these problems, the objective is to establish a production order of the products. Frequently, this order must be maintained from process to process whenever possible at all stages the manufacturing and

POSTPRINT: J. Bautista, A. Cano, R. Alfaro (2016) A Hybrid Dynamic Programming for solving a Mixed-Model Sequencing Problem with Production Mix Restriction and free interruptions. Progress in Artificial Intelligence, DOI 10.1007/s13748-016-0101-5

supply chain of the production systems governed by the Just in Time (JIT, Toyota) and Douki Seisan (DS, Nissan) philosophies.

Focusing on assembly lines, the order is conditioned by the line characteristics, the manufactured products and the most important elements of the production systems to establish optimization criteria. Among these elements we state the following: (1) component and product stocks, (2) human resources, and (3) special options within the products (e.g., sunroof, long body, or reinforced frame) that can generate bottlenecks in the assembly line.

Considering the stock as a relevant element of the system, a reasonable objective is to establish a product sequence that minimizes the stocks levels of products and components. To do this, we can either limit or minimize the variation of the production rates, as is the case in the Product Rate Variation Problem (PRVP), which was introduced by [1], or limit or minimize the variation of the product components rates, as is done in the problem proposed by [2], which is called the Output Rate Variation Problem (ORVP). In both cases, the objective is to keep these rates constants over time.

In contrast, if we consider human resources (HR) as the relevant element of the manufacturing system, then a reasonable objective is to minimize the work overloads that can appear when the mixed-product units treated by the line require different processing times at each stage or, more concretely, at each workstation. To achieve this, we can minimize the total work overload or maximize the total work completed, as in the Mixed-model Sequencing Problem with Work-overload minimization (MMSP-W), which was proposed by [3]. A recent work regarding this problem is [4].

Finally, if the bottlenecks generated by special options of some products are the relevant element of the manufacturing system, the reasonable objective is to minimize the number of sub-sequences of products with special options (units segments), which can be detrimental to the production line because more work or space (compared with the standard) is required consecutively at each workstation. One of these types of problems is the Car Sequencing Problem (CSP), which was originally proposed by [5], in which the constraint consists of sequencing a set of units with special options while respecting the number of allowed options within sub-sequences. Some works regarding the CSP include [6] and [7]. Among the variants of the CSP, we can find the following: (1) a version that considers the problem as an optimization problem rather than a constraint satisfaction problem [8] and (2) an extended version (named xCSP: extended CSP) that incorporates restrictions to allow a minimum number of products with special options in a sub-sequence of products [9].

Sometimes, as in real environments, the problems are treated as multi-objective problems. Several authors have used this perspective. For example [10] incorporated into the CSP conditions from the level scheduling and [11] incorporated conditions such as the minimization of the total utility work and idle costs into the mixed model assembly line (MMAL). Focusing on mixed model sequencing [12] incorporated the minimization of the utility work into the mixed model sequencing problem (MMSP). There also exist more recent works: for example [13] and [14] proposed incorporating conditions from the PRV into the MMSP-W. This objective can be achieved through regularizing the work or the work overload using pmr (product mix restrictions), as in the case of the MMSP-W-pmr, for example.

A survey of some of these sequencing problems can be found in [15].

This paper examines a variant of the MMSP-W, the MMSP-W-pmr. The original problem, MMSP-W, is an NP-hard problem [3] for which several alternative procedures have been proposed. These solutions include exact procedures based on branch-and-bound ([16]), dynamic programming ([3], [17], [18]), heuristic procedures based on local search ([19]), greedy algorithms with priority rules ([19], [20]), meta-heuristics ([21]) and beam search ([22]). Several studies have also considered the multi-criteria option ([23], [24], [25], [26]).

Given the complexity of the problem and the size of the case study related to Nissan Barcelona powertrain plant presented in [17], our objective is to find a computationally competitive procedure to solve the problem.

For this paper, we use a hybrid meta-heuristic based on bounded dynamic programming (BDP) assisted by linear programming. This procedure combines features of dynamic programming with features of branch-and-bound algorithms [27]. A review of hybrid meta-heuristics in combinatorial optimization can be found in [28].

Our proposal contains the following: (1) a model for the problem; (2) to solve this problem, procedures based on dynamic programming, which are referred to in this article as BDP-2/1 and BDP-2/2 (two versions), that use linear programming to obtain bounds for the problem; (3) a mathematical model to obtain the work overload of a given subsequence for use as 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 real instances from a case study of Nissan such that we can compare the results yielded by BDP-2 procedures with those offered by Mixed Integer Linear Programming (MILP).

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 presents an illustrative example. Section 4 describes the basic elements and the application of the proposed BDP procedure. Section 5 describes the computational experiment with a case study related to the Nissan powertrain plant. Finally, Section 6 presents the conclusions of the study.

## 2 Model for the problem

The MMSP-W The MMSP-W consists of sequencing  $T$  products, of which  $d_i$  are of type  $i$  ( $i = 1, \dots, n$ ). A unit of product type  $i$  requires from each processor (e.g., operator or robot) of workstation  $k$  ( $k = 1, \dots, m$ ) a processing time,  $p_{i,k}$ , assuming the processor works at its normal work pace or activity level. The standard time assigned to each processor to work on any product unit is the cycle time  $c$ . When a cycle ends at workstation  $k \in K$  (where  $K$  is the set of workstations), the processor can work on the product in progress for an additional positive time  $l_k - c$ , where  $l_k$  is the time window.

When it is not possible to complete all of the work required by the demand plan, work overload is generated. The objective of the problem is to maximize the total

POSTPRINT: J. Bautista, A. Cano, R. Alfaro (2016) A Hybrid Dynamic Programming for solving a Mixed-Model Sequencing Problem with Production Mix Restriction and free interruptions. Progress in Artificial Intelligence, DOI 10.1007/s13748-016-0101-5

work performed, which is equivalent to minimizing the total work overload generated (see Theorem 1 [17]).

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 M\_4U3\_pmr model proposed by [13], whose parameters and variables are presented below.

Parameters	
$K$	Set of workstations ( $k = 1, \dots,  K $ ; $ K  = m$ )
$b_k$	Number of homogeneous processors at station $k$
$I$	Set of product types ( $i = 1, \dots,  I $ ; $ I  = n$ )
$d_i$	Programmed demand for product type $i$
$p_{i,k}$	Processing time required for a unit of type $i$ at station $k$ for each homogeneous processor (at its normal activity level)
$T$	Total demand; obviously, $\sum_{\forall i} d_i = T$
$t$	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: Maximum time that each processor at 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 $k$
$\dot{d}_i$	Ideal rate of production for product type $i$ : $\dot{d}_i = d_i/T$ ( $i = 1, \dots,  I $ ).
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
$s_{k,t}$	Starting instant for the $t^{\text{th}}$ 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 $k \in K$ , $\hat{s}_{k,t} = \max(0, s_{k,t} - (t + k - 2)c)$
$v_{k,t}$	Processing time applied to the $t^{\text{th}}$ unit of the product sequence at station $k$ for each homogeneous processor (at its normal activity level)
$w_{k,t}$	Work overload generated for the $t^{\text{th}}$ unit of the product sequence at station $k$ for each homogeneous processor (at its normal activity level); measured in units of time
$\rho_{k,t}$	Processing time required for the $t^{\text{th}}$ unit of the sequence of products at workstation $k$ for each homogeneous processor (at its normal activity level)

Under these conditions, we can define the following mathematical model.

Model M\_4U3\_pmr:

$$\min W = \sum_{k=1}^{|K|} \left( b_k \sum_{t=1}^T w_{k,t} \right) \Leftrightarrow \max V = \sum_{k=1}^{|K|} \left( b_k \sum_{t=1}^T v_{k,t} \right) \quad (1)$$

subject to:

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

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

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

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

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

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

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

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

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

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

$$\hat{s}_{1,1} = 0 \quad (12)$$

$$\sum_{\tau=1}^t x_{i,\tau} \geq \lfloor t \cdot \dot{d}_i \rfloor \quad i = 1, \dots, |I|; t = 1, \dots, T \quad (13)$$

$$\sum_{\tau=1}^t x_{i,\tau} \leq \lceil t \cdot \dot{d}_i \rceil \quad i = 1, \dots, |I|; t = 1, \dots, T \quad (14)$$

Where  $\lfloor x \rfloor = \text{int}(x)$  and  $\lceil x \rceil = -\text{int}(-x)$ ,  $\forall x$ .

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 work overload generated by each unit at each workstation. Constraints (5)-(8) constitute the set of relative starting 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 work overload, respectively, are nonnegative. Constraint (11) requires the assignment variables to be binary. Constraint (12) fixes the start of operations. Constraints (13) and

POSTPRINT: J. Bautista, A. Cano, R. Alfaro (2016) A Hybrid Dynamic Programming for solving a Mixed-Model Sequencing Problem with Production Mix Restriction and free interruptions. Progress in Artificial Intelligence, DOI 10.1007/s13748-016-0101-5

(14) are those that incorporate the preservation property of the production mix desired in the JIT (Toyota) and Douki Seisan (Nissan) philosophies.

Similar to [29], in this work, we measure the non-regularity of a sequence using the following quadratic function (15):

$$\Delta_Q(X) = \sum_{t=1}^T \sum_{i=1}^n (X_{i,t} - t \cdot \dot{d}_i)^2 \quad (15)$$

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.

### 3 An illustrative example

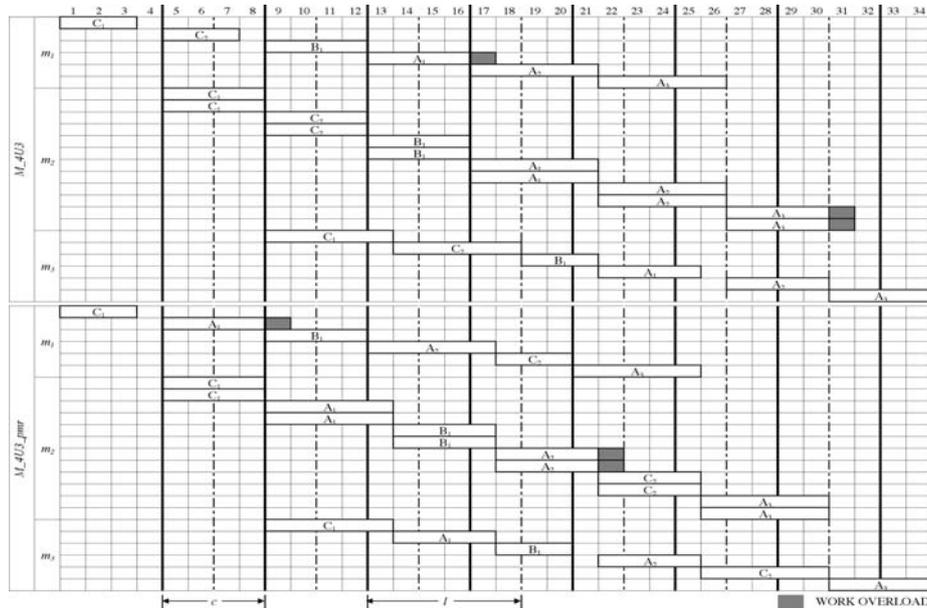
To illustrate the model formulated above, we present the following example: There are six units of product ( $T = 6$ ), of which three are type A, one is type B and two are type C, with a total work required  $V_0 = 104$ . The units are processed at three workstations ( $m = 3$ ) with different numbers of processors ( $b_k$ ); the processing times for each processor (at its normal activity level) for each type of unit  $i$  (A,B,C) at each workstation  $k$  ( $m_1$ ,  $m_2$ , and  $m_3$ ) are listed in Table 1. Furthermore,  $c = 4$  (cycle time) and  $l_k = 6 \forall k$  (length of workstation or time window).

**Table 1.** Construction phase Number of homogeneous processors ( $b_k$ ) at each station and processing times ( $p_{i,k}$ ) for each processor (at its normal activity level) required for each type of unit at each work station.

	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
Total	19 ( $V_0(A) = 57$ )	15 ( $V_0(B) = 15$ )	16 ( $V_0(C) = 32$ )	$V_0 = 104$

On the other hand, Fig 1 shows a Gantt chart of the optimal solutions offered by models M\_4U3 (top) and M\_4U3\_pmr (bottom). The sequence of products that yield the minimum total work overload for M\_4U3 is C-C-B-A-A-A. The total work performed is  $V=101$ , and the work 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 yields the minimum total work overload for M\_4U3\_pmr is C-A-B-A-C-A (the sequence is affected by the production mix restrictions). The total work performed is  $V = 101$ , and the work overload, which is concentrated between workstations  $m_1$  and  $m_2$ , is also  $W = 3$ , whereas the non-regularity for M\_4U3\_pmr is 2.05.

**Fig 1.** Gantt chart for the optimum solutions for the example provided by M\_4U3 (top) and M\_4U3\_pmr (bottom).



## 4 BDP Procedure to solve the problem

### 4.1 Introduction

Bounded Dynamic Programming (BDP) combines features of dynamic programming (determination of minimal-maximal paths in graphs) with features of branch and bound algorithms. The principles of BDP have been described by [27] to solve the Output Rate Variation Problem.

Moreover, they are easy to see the similarities between BDP and Beam Search (BS) approach ([22], [30]) before exploring a multistage graph in combinatorial problems. In fact BDP and BS combined elements of breadth-first and best-first searches and therefore its implementation is similar. The basic difference between the two procedures is that BDP accurate lower/upper bounds (in case of minimization / maximization) to solve the problem and can thereby ensure optimal solutions, while BS only requires heuristic rules to explore the graph. Obviously, the scheme BS can also incorporate the use of lower bounds (if minimization) or upper bounds (if maximization).

This section presents the basic elements of the BDP procedure applied to the MMSP-W with serial workstations, unrestricted interruption of the operations and production mix restrictions (here, we name this meta-heuristic procedures BDP-2).

## 4.2 Graph associated with the problem

Similar to ([18],[27]) can build a linked graph without loops or direct cycles of  $T + 1$  stages. The set of vertices at level  $t$  ( $t = 0, \dots, T$ ) is denoted as  $J(t)$  and a vertex of level  $t$  is denoted as  $J(t, j)$  ( $j = 1, \dots, |J(t)|$ ), which is defined by the tuple:

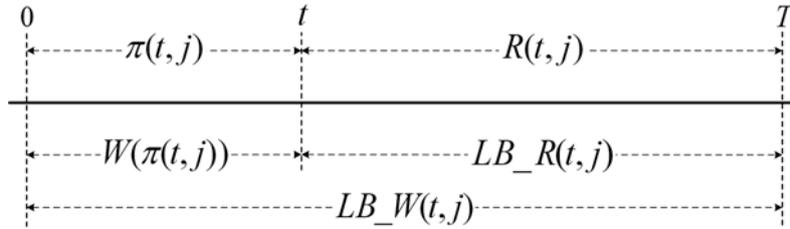
$$J(t, j) = \{(t, j), \vec{q}(t, j), \pi(t, j), W(\pi(t, j)), LB\_R(t, j), \Delta_Q(X(\pi(t, j)))\}$$

where:

- $\vec{q}(t, j) = (q_1(t, j), \dots, q_n(t, j))$  is the vector production associated with  $J(t, j)$
- $\pi(t, j) = (\pi_1(t, j), \dots, \pi_i(t, j))$  is the sequence associated with  $J(t, j)$
- $W(\pi(t, j))$  is the work overload generated by the sequence  $\pi(t, j)$
- $LB\_R(t, j)$  is a lower bound of the work overload associated with  $R(t, j)$
- $R(t, j)$  corresponds to complementary segment of units not sequenced associated with  $\pi(t, j)$ :  $d_i - q_i(t, j) \quad \forall i = 1, \dots, n$ .
- $\Delta_Q(X(\pi(t, j)))$  is the non-regularity of production generated by  $\pi(t, j)$

Obviously, to obtain a global bound ( $LB\_W(t, j)$ ) of the work overload associated with  $J(t, j)$ , we can set:  $LB\_W(t, j) = W(\pi(t, j)) + LB\_R(t, j)$ . Therefore, the overall bound on  $W$  can be determined according to the schema presented in Fig 2.

**Fig 2.** Bound scheme for a partial sequence  $\pi(t, j)$  at vertex  $J(t, j)$ .



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

$$\sum_{i=1}^n q_i(t, j) = t \quad (16)$$

$$\lfloor t \cdot \dot{d}_i \rfloor \leq q_i(t, j) \leq \lceil t \cdot \dot{d}_i \rceil \quad \forall i \in I \quad (17)$$

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 with all of the sub-sequences that can be built with  $t$  products that satisfy properties (16) and (17). However, it is

easy to a priori reduce the cardinality of  $J(t)$  by establishing the following definitions of pseudo-dominance (PSD):

**Definition 1.** PSD\_1: Given the sequences  $\pi(t, j_1)$  and  $\pi(t, j_2)$  associated with the vertices  $J(t, j_1)$  and  $J(t, j_2)$ , then  $\pi(t, j_1)$  pseudo-dominates  $\pi(t, j_2)$  if:

$$\pi(t, j_1) \prec \pi(t, j_2) \Leftrightarrow \left\{ \begin{array}{l} [\bar{q}(t, j_1) = \bar{q}(t, j_2)] \wedge \\ [LB\_W(t, j_1) \leq LB\_W(t, j_2)] \wedge \\ [\Delta_\rho(X(\pi(t, j_1))) \leq \Delta_\rho(X(\pi(t, j_2)))] \end{array} \right\} \quad (18)$$

**Definition 2.** PSD\_2: Given the sequences  $\pi(t, j_1)$  and  $\pi(t, j_2)$  associated with the vertices  $J(t, j_1)$  and  $J(t, j_2)$ , then  $\pi(t, j_1)$  pseudo-dominates  $\pi(t, j_2)$  if:

$$\pi(t, j_1) \prec \pi(t, j_2) \Leftrightarrow \left\{ \begin{array}{l} [\bar{q}(t, j_1) = \bar{q}(t, j_2)] \wedge \\ [LB\_W(t, j_1) < LB\_W(t, j_2)] \\ \vee \\ \left[ \begin{array}{l} (\bar{q}(t, j_1), LB\_W(t, j_1)) \equiv \\ (\bar{q}(t, j_2), LB\_W(t, j_2)) \end{array} \right] \wedge \\ [\Delta_\rho(X(\pi(t, j_1))) \leq \Delta_\rho(X(\pi(t, j_2)))] \end{array} \right\} \quad (19)$$

The reduction of  $J(t)$  through the pseudo-dominances defined in (18) or (19) cannot guarantee the optimality of the solutions.

### 4.3 Lower Bounds for the problem

Given the  $J(t, j)$  vertex, reached through the sequence  $\pi(t, j)$ , and a partial bound on the complementary segment  $R(t, j)$  associated with  $\pi(t, j)$ , the overall bound on  $W$  can be determined according:  $LB\_W(t, j) = W(\pi(t, j)) + LB\_R(t, j)$ .

To obtain the work overload associated with  $\pi(t, j)$ , in each stage of the procedure, we use linear programming.

Indeed, given a sequence  $\pi(t, j)$ , the processing times for each workstation  $k$  of the  $\tau^{th}$  ( $\tau = 1, \dots, t$ ) unit of  $\pi(t, j)$  are foreknown:  $\rho_{k,\tau} = P_{\pi_\tau(t,j),k}$ . Then we can define a linear program ( $LP\_W(\pi(t, j))$ ) in which the assignment variables have been removed. Here it is:

$$LP\_W(\pi(t, j)) : \min W(\pi(t, j)) = \sum_{k=1}^m \left( b_k \sum_{\tau=1}^t w_{k,\tau} \right) \quad (20)$$

subject to:

$$\rho_{k,\tau} = P_{\pi_\tau(t,j),k} \quad k = 1, \dots, m; \tau = 1, \dots, t \quad (21)$$

$$\rho_{k,\tau} - w_{k,\tau} \geq 0 \quad k = 1, \dots, m; \tau = 1, \dots, t \quad (22)$$

$$\hat{s}_{k,\tau} \geq \hat{s}_{k,\tau-1} + \rho_{k,\tau-1} - w_{k,\tau-1} - c \quad k = 1, \dots, m; \tau = 2, \dots, t \quad (23)$$

$$\hat{s}_{k,\tau} \geq \hat{s}_{k-1,\tau} + \rho_{k-1,\tau} - w_{k-1,\tau} - c \quad k = 2, \dots, m; \tau = 1, \dots, t \quad (24)$$

$$\hat{s}_{k,\tau} + \rho_{k,\tau} - w_{k,\tau} \leq l_k \quad k = 1, \dots, m; \tau = 1, \dots, t \quad (25)$$

$$\hat{s}_{k,\tau} \geq 0 \quad k = 1, \dots, m; \tau = 1, \dots, t \quad (26)$$

$$w_{k,\tau} \geq 0 \quad k = 1, \dots, m; \tau = 1, \dots, t \quad (27)$$

$$\hat{s}_{1,1} = 0 \quad (28)$$

The result of the proposed linear program corresponds to  $W(\pi(t, j))$ .

To obtain a bound on the work overload associated with the complement  $R(t, j)$ , we use the combination of three lower bounds.

Given a workstation  $k$  and a vertex  $J(t, j)$ , the available time to complete the pending operations, for each homogeneous processor at its normal activity level, is:

$$TD_k(t, j) = (T - t - 1) \cdot c + l_k \quad k = 1, \dots, m \quad (29)$$

whereas the required time to complete these operations is:

$$TP_k(t, j) = \sum_{i=1}^n p_{i,k} \cdot (d_i - q_i(t, j)) \quad k = 1, \dots, m \quad (30)$$

Using (29) and (30), we can define a lower bound on the work overload of  $R(t, j)$ :

$$LB1(t, j) = \sum_{k=1}^m b_k \max(0, TP_k(t, j) - TD_k(t, j)) \quad (31)$$

However, if we consider the minimum work overload that a product of type  $i$  can generate, we have:

$$LB2(i) = \max\left(0, \sum_{k=1}^m b_k (p_{i,k} - c) - b_{|K|} (l_{|K|} - c)\right) \quad i = 1, \dots, n \quad (32)$$

Thus, a bound on the work overload of  $R(t, j)$  is the following:

$$LB2(t, j) = \sum_{i=1}^n (d_i - q_i(t, j)) \cdot LB2(i) \quad (33)$$

A more refined bound on the minimum work overload that a unit of product type  $i$  can generate can be obtained using the following mathematical model:

$$LP\_LB3(i) : \min LB3(i) = \sum_{k=1}^m b_k \cdot w_{k,i} \quad (34)$$

subject to:

$$\hat{s}_{k,i} \geq \hat{s}_{k-1,i} + p_{k-1,i} - w_{k-1,i} - c \quad k = 2, \dots, |K| \quad (35)$$

$$\hat{s}_{k,i} + p_{k,i} - w_{k,i} \leq l_k \quad k = 1, \dots, |K| \quad (36)$$

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

$$\hat{s}_{k,i} \geq 0 \quad k = 1, \dots, |K| \quad (38)$$

$$w_{k,i} \geq 0 \quad k = 1, \dots, |K| \quad (39)$$

$$\hat{s}_{1,i} = 0 \quad (40)$$

Using the optimal solutions,  $LB3^*(i)$ , of the previous mathematical model, we can determine the following bound on the work overload of  $R(t, j)$ :

$$LB3(t, j) = \sum_{i=1}^n (d_i - q_i(t, j)) \cdot LB3^*(i) \quad (41)$$

To determine  $LB\_R(t, j)$ , we use:

$$LB\_R(t, j) = \max\{LB1(t, j), LB2(t, j), LB3(t, j)\} \quad (42)$$

Finally, we can obtain a lower bound on the total work overload associated with vertex  $J(t, j)$ :

$$LB\_W(t, j) = W(\pi(t, j)) + LB\_R(t, j) \quad (43)$$

#### 4.4 Properties from the production mix restrictions (pmr)

In this section, we will study the properties of the product sequences that are derived from the incorporation of the restrictions to preserve the production mix (pmr) in the MMSP-W.

First, we must define how to measure the non-regularity of the production ( $\Delta_Q(X)$ ) at each vertex of the graph associated with the problem. In effect, given a vertex  $J(t, j)$ , associated with sequence  $\pi(t, j)$ , let  $X_{i,\tau}(\pi(t, j))$  ( $i = 1, \dots, n$ ),  $\tau = 1, \dots, t$ ) be the number of units of product type  $i$  sequenced at the first  $\tau$  positions of the sequence  $\pi(t, j)$ , that is:

$$X_{i,\tau}(\pi(t, j)) = |\{\pi_h(t, j) \in \pi(t, j) : (\pi_h(t, j) = \{i\}) \wedge (1 \leq h \leq \tau)\}| \quad (44)$$

Using the previous definition, let us define the non-regularity of the production associated with the sequence  $\pi(t, j)$  of vertex  $J(t, j)$  as:

$$\Delta_Q(X(\pi(t, j))) = \sum_{\tau=1}^t \sum_{i=1}^n (X_{i,\tau}(\pi(t, j)) - \tau \cdot \dot{d}_i)^2 \quad (45)$$

The restrictions to preserve the production mix can be expressed as follows:

$$\lfloor t \cdot \dot{d}_i \rfloor \leq X_{i,t} \leq \lceil t \cdot \dot{d}_i \rceil \quad i = 1, \dots, n; t = 1, \dots, T \quad (46)$$

where  $X_{i,t}$  is a variable that represents the total number of units of product type  $i$  sequenced during the first  $t$  production cycles.

The imposition of these restrictions on the sequences leads to a set of properties which are defined in [29]. These properties are the following:

**Theorem 2:** If  $\lfloor t \cdot \dot{d}_i \rfloor \leq X_{i,t} \leq \lceil t \cdot \dot{d}_i \rceil \quad \forall (i,t)$ , then:  $X_{i,t} - X_{j,t} \leq \lceil t \cdot \dot{d}_i \rceil - \lfloor t \cdot \dot{d}_j \rfloor$   
 $\forall \{i, j\} \subseteq I, \forall t$ .

**Corollary 2:** If  $d_i \leq d_j$ , then  $X_{i,t} - X_{j,t} \leq 1 \quad \forall \{i, j\} \subseteq I, \forall t$ .

**Theorem 3:** If  $\lfloor t \cdot \dot{d}_i \rfloor \leq X_{i,t} \leq \lceil t \cdot \dot{d}_i \rceil \quad \forall (i,t)$ , then:  $X_{i,t} - X_{j,t} \geq \lfloor t \cdot \dot{d}_i \rfloor - \lceil t \cdot \dot{d}_j \rceil$   
 $\forall \{i, j\} \subseteq I, \forall t$ .

**Corollary 3:** If  $d_i \geq d_j$ , then  $X_{i,t} - X_{j,t} \leq -1 \quad \forall \{i, j\} \subseteq I, \forall t$ .

**Corollary 4:** If  $d_i = d_j$ , then  $|X_{i,t} - X_{j,t}| \leq 1 \quad \forall \{i, j\} \subseteq I, \forall t$ .

In addition, the fulfilment of the pmr restrictions combined with the demand variety, results in the following property:

**Theorem 4:** If  $\lfloor t \cdot \dot{d}_i \rfloor \leq X_{i,t} \leq \lceil t \cdot \dot{d}_i \rceil \quad \forall (i,t)$ , given the sequence  $\pi = (\pi_1, \dots, \pi_T)$ , where  $\pi_t = \{j\}$  with  $2 \leq t \leq T$ , the following is satisfied:

If  $\exists i \in I : (X_{i,t} > 0) \wedge (d_i \leq d_j) \Rightarrow X_{i,t} \leq X_{j,t} \quad \forall t = 2, \dots, T$ .

**Proof:** If we suppose that  $\exists i \in I : (X_{i,t} > 0) \wedge (d_i \leq d_j)$  such that  $X_{i,t} > X_{j,t}$ , then  $X_{i,t} - X_{j,t} \geq 1$ . In contrast, if  $\pi_t = \{j\}$  then  $X_{j,t} = X_{j,t-1} + 1$  and  $X_{i,t} = X_{i,t-1}$ .

Thus, we can write  $X_{i,t} - X_{j,t} = X_{i,t-1} - X_{j,t-1} - 1 \geq 1 \Rightarrow X_{i,t-1} - X_{j,t-1} \geq 2$ .

Furthermore, given that:

$X_{i,t-1} - X_{j,t-1} \leq \lceil (t-1) \cdot \dot{d}_i \rceil - \lfloor (t-1) \cdot \dot{d}_j \rfloor \leq \lceil (t-1) \cdot \dot{d}_j \rceil - \lfloor (t-1) \cdot \dot{d}_j \rfloor$ , we have  $\lceil (t-1) \cdot \dot{d}_j \rceil - \lfloor (t-1) \cdot \dot{d}_j \rfloor \geq X_{i,t-1} - X_{j,t-1} \geq 2$ , which is absurd.

Thus, the hypothesis  $X_{i,t} > X_{j,t}$  is false and consequently, the following must be fulfilled:  $X_{i,t} \leq X_{j,t} \quad \forall t = 2, \dots, T$  and  $\forall i \in I : X_{i,t} > 0$ , when  $\pi_t = \{j\}$ .

**Corollary 5:** If  $\lfloor t \cdot \dot{d}_i \rfloor \leq X_{i,t} \leq \lceil t \cdot \dot{d}_i \rceil \quad \forall (i,t)$ , given the sequence  $\pi = (\pi_1, \dots, \pi_T)$ , where  $\pi_t = \{j\}$  with  $2 \leq t \leq T$ , and if  $\exists i \in I : X_{i,t} > X_{j,t} \Rightarrow d_i > d_j$ .

Evidently, from Theorem 3,  $d_i \leq d_j \Rightarrow X_{i,t} \leq X_{j,t}$ , which negates the hypothesis ( $X_{i,t} > X_{j,t}$ ); therefore, it must be that  $d_i > d_j$ .

#### 4.5 Rules to discard vertices from Graph associated with the problem

At stage  $t$ , let  $X_i(\pi(t-1, h))$  be the satisfied demand for product type  $i \in I$  associated with the sequence  $\pi(t-1, h)$  of the vertex  $J(t-1, h)$ .

Assuming that an extension of the vertex  $J(t-1, h)$  is built by adding at stage  $t$  a product type  $j$  to the sequence, let  $J(t, h')$  be the resulting vertex for the partial sequence  $\pi(t, h') = \pi(t-1, h) \cup \{j\}$ . The satisfied demands must fulfil:

$$\begin{aligned} X_i(\pi(t, h')) &= X_i(\pi(t-1, h)) \quad \forall i \neq j \\ X_j(\pi(t, h')) &= X_j(\pi(t-1, h)) + 1 \quad \text{with } \pi_t(t, h') = \{j\} \end{aligned}$$

Under these conditions, the vertex  $J(t, h')$  can be discarded from the exploring process if any one the following rules is satisfied:

Block 1 (pmr constraints):

$$\begin{aligned} \forall j \in I, \text{ if } \exists j: [X_j(\pi(t, h')) < \lfloor t \cdot \dot{d}_j \rfloor] \vee [X_j(\pi(t, h')) > \lceil t \cdot \dot{d}_j \rceil] \\ \rightarrow \text{Discard } J(t, h') \end{aligned}$$

Block 2 (Theorems 2 and 3):

$$\begin{aligned} \forall i \neq j \in I, \text{ if } \exists i: [X_i(\pi(t, h')) - X_j(\pi(t, h')) > \lceil t \cdot \dot{d}_i \rceil - \lfloor t \cdot \dot{d}_j \rfloor] \vee \\ [X_i(\pi(t, h')) - X_j(\pi(t, h')) < \lfloor t \cdot \dot{d}_i \rfloor - \lceil t \cdot \dot{d}_j \rceil] \rightarrow \text{Discard } J(t, h') \end{aligned}$$

Block 3 (Corollaries 2, 3 and 4):

$$\begin{aligned} \forall i \neq j \in I, \text{ if } [[d_i < d_j] \wedge [X_i(\pi(t, h')) - X_j(\pi(t, h')) > 1]] \vee \\ \text{if } [[d_i = d_j] \wedge [X_i(\pi(t, h')) - X_j(\pi(t, h')) > 1]] \vee \\ \text{if } [[d_i > d_j] \wedge [X_j(\pi(t, h')) - X_i(\pi(t, h')) > 1]] \rightarrow \text{Discard } J(t, h') \end{aligned}$$

Block 4 (Theorem 4):

$$\begin{aligned} \text{Given the partial sequence } \pi(t, h') \text{ associated with vertex } J(t, h'), \text{ with} \\ \pi_t(t, h') = \{j\} \\ \forall i \in I: X_i(\pi(t, h')) \geq 1; \text{ if } [[d_j \geq d_i] \wedge (X_j(\pi(t, h')) < X_i(\pi(t, h')))] \\ \rightarrow \text{Discard } J(t, h') \end{aligned}$$

#### 4.6 Use of BDP to solve the problem

To solve the Mixed-Model Sequencing Problem with work overload minimization, production mix preservation and free interruption of the operations, we propose the following meta-heuristic:

---

*BDP-2 – MMSP-W*

---

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

*Output:* list of sequences obtained by BDP

---

---

```

0  Initialization:  $t = 0$  ,  $LBZ_{\min} \rightarrow \infty$ 
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, the following functions appear:

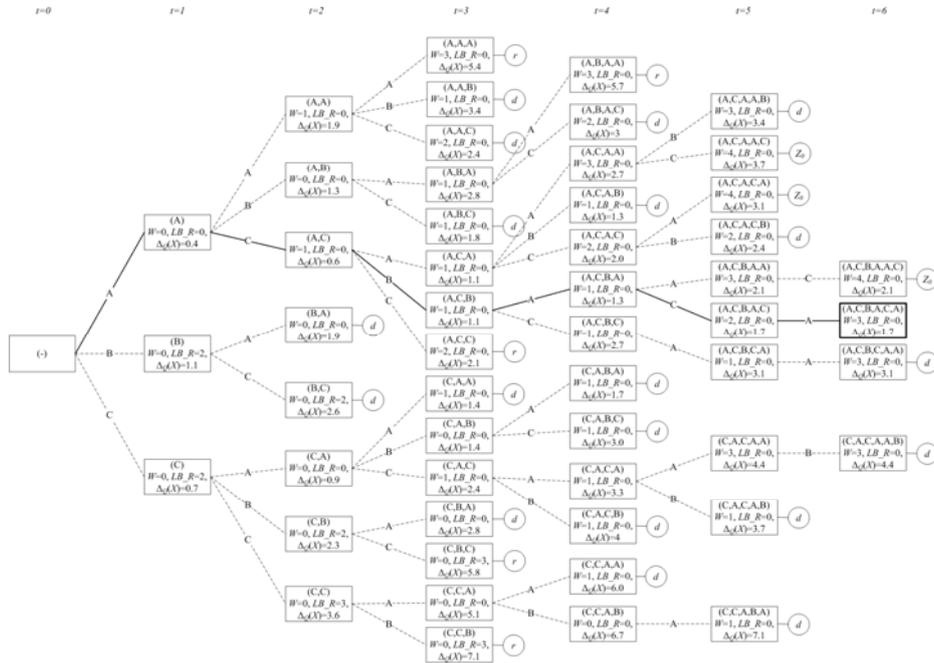
- Generate\_model(): this function generates the initial model  $LP\_W(\pi(t, j))$  to obtain the optimal solution  $W^*(\pi(t, j))$  for  $t = 0$ .
- Add\_constraints( $t$ ): this function adds the new constraints associated with the new stage ( $t$ ) to the existing model.
- Select\_vertex( $t$ ): this function selects one of the vertices consolidated in stage  $t - 1$  following a non decreasing ordering of the  $LB\_W(t, j)$  values.
- Develop\_vertex( $t$ ): this function develops the vertex selected in the previous function by adding a new product unit with pending demand. Vertices that do not satisfy properties (16) and (17) are not generated. This is performed by incorporating the rules contained in the blocks to discard vertices Block 1 and Block 3 into this phase.
- Filter\_vertices( $Z_0, H, LBZ_{\min}$ ): this function 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  $LB\_W(t, j)$ ) and removes those vertices for which the lower bound is greater than  $\bar{Z}_0$  (known initial solution) and those that are pseudo-dominated, as defined in (18) or (19).
- End\_stage(): this function consolidates the most promising vertices in stage  $t$  ( $H$  vertices are the maximum number of vertices selected).

#### 4.7 Example of the Graph reduction

Figure 3 represents the vertex exploration of the graph associated with the problem to solve the illustrative example through the BDP procedure described in this paper; here, we do not perform the elimination of vertices allowed by the incorporation of the pmr restrictions into the MMSP-W. In the example, an initial solution  $Z_0 = 4$  and a window width  $H = 6$  have been used.

Figure 4 represents the same exploration when the rules, to discard vertices associated with Block-1 and Block-3 to assure the fulfilment of the pmr restrictions, are incorporated to the BDP procedure. For this graph,  $Z_0 = 4$  and  $H = 6$  have also been used (although  $H = 3$  is sufficient).

**Fig 3.** Original graph for the example using  $Z_0 = 4$ , pseudo-dominance-1 and  $H = 6$ . Note the similarity between BDP and Beam Search when exploring the multistage graph



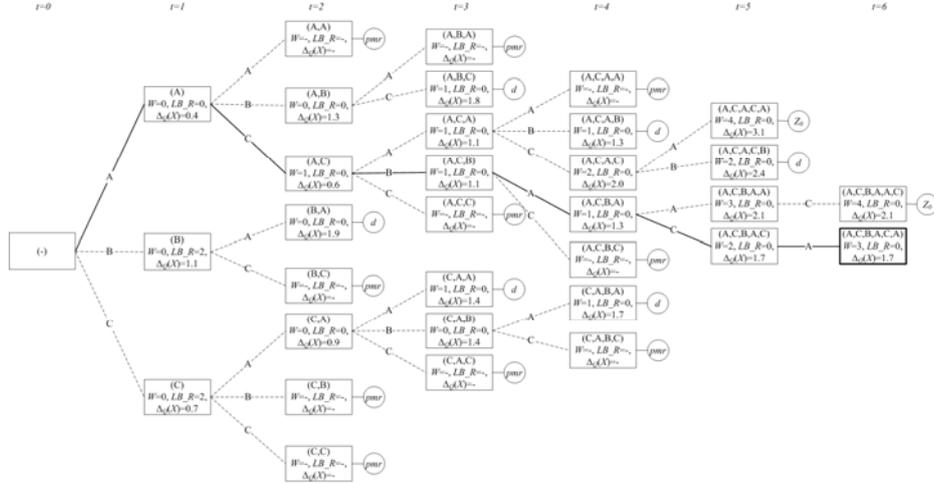
In the figures, we can see the vertices' elimination states:

- **Dominated vertex ( $d$ ).** For the example in Figure 3, the representative vertex of the partial sequence (B,C), with  $\Delta_q(X) = 2.6$ , is dominated by vertex (C,B), with  $\Delta_q(X) = 2.3$ . Additionally, in the same figure, we can see that vertex (B,A) is dominated by (A,B).
- **Removed vertex ( $r$ ).** The limitation of the window width to  $H = 6$  contributes to selecting the most promising vertices (best value for  $LB\_W$ ) to be developed at each stage  $t$ . For example, at stage  $t = 3$  of figure 3, the vertices that correspond to the partial sequences (A,A,A), (A,C,C), (C,B,C) and (C,C,B) are removed and only six vertices are developed to reach stage  $t = 4$ .
- **Discarded vertex ( $Z_0$ ).** The discarded vertices are those for which their development cannot finish at solution that is better than the best known solution  $Z_0$ . For example, the sequence (A,C,B,A,A,C) in figure 4 does not improve the best known

solution  $Z_0 = 4$ .

- Breaker vertex (pmr). This is a vertex for which the sequence does not satisfy the restrictions, in our case, the pmr restrictions. For example, in figure 4, the partial sequences (A,A), (B,C), (C,B) and (C,C) do not satisfy the pmr restrictions.

**Fig 4.** Graph using the pmr restrictions,  $Z_0 = 4$ , pseudo-dominance-1 and  $H = 6$ .



## 5 Computational experience. Case study

The computational experience is aimed to assess the BDP-2's performances (BDP-2/1 and BDP-2/2) against MILP procedure (*Mixed Integer Linear Programming*) in regard with the quality of solutions and CPU times. For the test, we use a case study from the Nissan's plant in Barcelona, which consists of an assembly line of nine types of engines that are grouped into three families (SUVs, vans and trucks). In the said line there are 42 operators when the cycle time is around three minutes.

Similar to [31] the features of the study case are:

- Number of workstations:  $m = 21$ .
- Number of products types:  $n = 9$ .
- Cycle time:  $c = 175s$ , and time window:  $l_k = 195s \forall k \in K$
- Number of homogeneous processors (with 2 operators):  $b_k = 1 \forall k \in K$
- Processing times  $p_{i,k} \forall i \in I, \forall k \in K$  by product and workstation. They are between 89 and 185 seconds at normal activity (see [17]: Table 5).
- Number of demand plans (set E):  $|E| = 23$  ( $\varepsilon = 1, \dots, 23$ ). All of them with the same daily demand (see [17]: Table 6, Block I, NISSAN-9ENG).
- Daily demand:  $T \equiv D_\varepsilon = 270$  units  $\forall \varepsilon = 1, \dots, 23$ .

On the other hand, the features of the procedures are:

- MILP-pmr: Mixed Integer Linear Program M\_4U3\_pmr proposed by [13]: Implementation of the model on Gurobi solver, v4.5.0, and running on an iMac (Intel Core i7 2.93 GHz, 8 GB de RAM). Maximum CPU time for each demand plan ( $\varepsilon \in E$ ) equal to 7200 s. (23 solver's executions). The model incorporates production mix preservation and free interruption of operations. Average CPU time for each demand plan ( $\varepsilon \in E$ ) equal to 6605 s.
- BDP-2/1: Bounded Dynamic Programming Algorithm (BDP) assisted by Linear Programming (LP): The code is compiled and run on iMac computer (Intel Core 2 Duo 2.33 GHz, 3 GB de RAM). Maximum number of transitions from each vertex, equal to the number of products types  $n=9$ . Four windows widths,  $\vec{H} = (1,36,81,126)$ , for the 23 demand plans (92 algorithm's executions). Initial solution,  $Z_0$ , for  $H_z$  ( $z=2,3,4$ ) equal to the best solution obtained with  $H_{z-1}$ , except for  $H_1 \rightarrow 1$ , where  $Z_0 \rightarrow \infty$ . Algorithm with pseudo-dominance rules PSD\_1 (formula 18). Algorithm with production mix preservation and with free interruption of operations assisted by Linear Programming. Average CPU time for each demand plan ( $\varepsilon \in E$ ) equal to 2946.1 s.
- BDP-2/2: Bounded Dynamic Programming Algorithm (BDP) assisted by Linear Programming (LP): The code is compiled and run on iMac computer (Intel Core 2 Duo 2.33 GHz, 3 GB de RAM). Maximum number of transitions from each vertex, equal to the number of products types  $n=9$ . Four windows widths,  $\vec{H} = (1,36,81,126)$ , for the 23 demand plans (92 algorithm's executions). Initial solution,  $Z_0$ , for  $H_z$  ( $z=1,2,3,4$ ) equal to the best solution obtained with  $H_{z-1}$ , except for  $H_1 \rightarrow 1$ , where  $Z_0 \rightarrow \infty$ . Algorithm with pseudo-dominance rules PSD\_2 (formula 19). Algorithm with production mix preservation and with free interruption of operations assisted by Linear Programming. Average CPU time for each demand plan ( $\varepsilon \in E$ ) equal to 2080.5 s.

Table 2 shows results about CPU times for MILP-pmr procedures and BDP-2/1 and BDP-2/2 algorithm's executions.

**Table 2.** CPU times for the case study of the Block I · NISSAN-9ENG (powertrain plant).

CPU (seconds)	MILP-pmr	BDP-2/1 (PSD_1)			
		H=1	H=36	H=81	H=126
CPU min	149.4	0.1	427.6	717.9	883.0
CPU max	7200.0	35.0	526.2	1119.9	1701.1
Average	6605.1	3.2	484.5	992.9	1465.5
CPU (seconds)	MILP-pmr	BDP-2/2 (PSD_2)			
		H=1	H=36	H=81	H=126
CPU min	149.4	0.1	369.0	524.1	535.0
CPU max	7200.0	35.9	509.8	972.2	1134.2
Average	6605.1	3.2	450.1	772.4	854.8

Regarding CPU times (see Table 2), BDP-2/1 and BDP-2/2 using a window width of  $H=126$  (the largest used in this experiment) improved the CPU times required

POSTPRINT: J. Bautista, A. Cano, R. Alfaro (2016) A Hybrid Dynamic Programming for solving a Mixed-Model Sequencing Problem with Production Mix Restriction and free interruptions. Progress in Artificial Intelligence, DOI 10.1007/s13748-016-0101-5

compared with that required by MILP-pmr (Gurobi solver) by 4.5 and 7.7 times, on average.

Table 3 presents the best values for  $W$  and  $\Delta_Q(X)$  of the 23 instances for the problem reached by MILP-pmr and the BDP-2/1 and BDP-2/2 procedures for the four window widths ( $H=1,36,81,126$ ). Moreover a relative deviation is used to calculate the relative gain of BDP-2 over MILP-pmr, that is the  $rg(\cdot)$ :

$$rg(f, \varepsilon) = \frac{f(\hat{S}_{MILP}(\varepsilon)) - f(\hat{S}_{BDP2}(\varepsilon))}{f(\hat{S}_{MILP}(\varepsilon))} \quad f \in \{W, \Delta_Q(X)\}; \varepsilon \in E \quad (47)$$

Where  $\hat{S}_{MILP}(\varepsilon)$  is the best solution found for the instance  $\varepsilon \in E$  using MILP-pmr and  $\hat{S}_{BDP2}(\varepsilon)$  is the best solution found for the instance  $\varepsilon \in E$  using the BDP-2 procedure, for each of its two variants (BDP-2/1 and 2/2).

**Table 3.** For each demand plan  $\varepsilon \in E$ : work overload ( $W_{MILP}, W_{BDP2}$ ) and non-regularity of production ( $\Delta_{Q,MILP}, \Delta_{Q,BDP2}$ ) according procedure MILP-pmr or BDP-2, relative gain  $rg(f, \varepsilon)$  of BDP-2 over MILP-pmr, and winner procedure.

$\varepsilon \in E$	$W_{MILP}$	$W_{BDP2}$	$rg(W, \varepsilon)$	Winner	$\Delta_{Q,MILP}$	$\Delta_{Q,BDP2}$	$rg(\Delta_Q, \varepsilon)$
1	186	166	0.11	BDP-2	400.0	400.0	0.00
2	383	318	0.17	BDP-2	423.5	327.9	0.23
3	423	444	-0.05	MILP	408.5	340.7	0.17
4	307	305	0.01	BDP-2	421.3	333.6	0.21
5	661	633	0.04	BDP-2	394.7	352.1	0.11
6	478	428	0.10	BDP-2	420.0	324.3	0.23
7	731	740	-0.01	MILP	396.0	388.1	0.02
8	160	112	0.30	BDP-2	448.1	347.6	0.22
9	751	739	0.02	BDP-2	411.2	360.7	0.12
10	<b>1208</b>	1209	0.00	MILP	381.1	330.8	0.13
11	122	92	0.25	BDP-2	447.3	384.1	0.14
12	287	293	-0.02	MILP	410.2	385.5	0.06
13	336	277	0.18	BDP-2	436.4	334.5	0.23
14	423	381	0.10	BDP-2	414.9	353.9	0.15
15	442	422	0.05	BDP-2	445.2	378.1	0.15
16	251	216	0.14	BDP-2	404.9	340.0	0.16
17	488	466	0.05	BDP-2	415.3	370.2	0.11
18	619	610	0.01	BDP-2	419.6	336.3	0.20
19	<b>945</b>	949	0.00	MILP	412.3	336.7	0.18
20	150	129	0.14	BDP-2	393.6	342.6	0.13
21	561	565	-0.01	MILP	404.2	354.8	0.12
22	984	991	-0.01	MILP	395.8	317.7	0.20
23	121	111	0.08	BDP-2	385.6	309.2	0.20
<i>Aver.</i>			<b>0.07</b>				<b>0.15</b>

The highlighted results are the following:

- MILP-pmr confirmed as optimal solutions corresponding to demand plans #10 and #19, while BDP-2 has not been able to guarantee optimal solutions.
- BDP-2 is the winner procedure for  $W$  with 16 best solutions of 23 instances
- BDP-2 is the winner procedure for  $\Delta_Q(X)$  with 22 best solutions of 23 instances (the procedures are tied for demand plan #1).
- On average, BDP-2 enhances the best solutions for  $W$  and  $\Delta_Q(X)$ , which were obtained with MILP-pmr. The values for the improvements for  $W$  and  $\Delta_Q(X)$ , obtained using BDP-2 were 7.1% and 15.1%, respectively.
- On average, BDP-2 and MILP-pmr used 5026.6 and 6605.1s., respectively, to validate the best solutions for each demand plan  $\varepsilon \in E$ .
- Observing the results of this computational experiment, we can conclude that BDP-2 is more competitive, on average, than MILP-pmr procedure.

## 6 Conclusions

We have proposed a hybrid procedure based on the BDP, the BDP-2 (two versions), for the MMSP-W problem that minimizes the total work overload or maximizes the total completed work and considers serial workstations, parallel processors, free interruption of the operations and restrictions to preserve the production mix in the manufacturing sequence.

The proposed procedure use global bounds based on Linear Programming (LP). A Linear Program that minimizes the work overload given a subsequence of operations at any instant has been formulated. In addition, the proposed procedure incorporates pseudo-dominances between partial solutions to limit the search space. These pseudo-dominances consider the preservation of the production mix in the partial solutions. From both versions of these pseudo-dominances, PSD\_1 and PSD\_2, we proposed two versions of the BDP-2 procedure (BDP-2/1 and BDP-2/2).

We performed a computational experiment corresponded to a case study of a Nissan powertrain plant in Barcelona. 23 instances corresponding to different demand plans, one production day and two shifts were considered. Both BDP-2 procedures were competitive in terms of CPU times and in terms of the results for and compared with MILP-pmr [13], because we always found an improvement, on average, for all indicators; however, MILP-pmr confirmed as optimal solutions for two demand plans, while BDP-2 has not been able to guarantee optimal solutions.

BDP-2 procedure is the winner in number of best solutions for both work overload values as preservation of the production mix. BDP-2 enhances the best solutions for work overload and production mix preservation, which were obtained with MILP-pmr; the values of these improvements are 7.1% a and 15.1%, respectively. Besides, BDP-2 reduced by 23.9% the average CPU time used by MILP-pmr.

POSTPRINT: J. Bautista, A. Cano, R. Alfaro (2016) A Hybrid Dynamic Programming for solving a Mixed-Model Sequencing Problem with Production Mix Restriction and free interruptions. Progress in Artificial Intelligence, DOI 10.1007/s13748-016-0101-5

At the present time, BDP-2 is the state of art procedure to solve the Mixed-Model Sequencing Problem with work overload minimization, production mix preservation and free interruption of the operations.

**Acknowledgements.** This work was funded by the Ministerio de Economía y Competitividad (Spanish Government) through the FHI-SELM2 (TIN2014-57497-P) project.

## 7 References

1. Miltenburg, J.: Scheduling Mixed-Model Assembly Lines for Just-In-Time Production Systems. *Management Science* 35(2), 192–207 (1989)
2. Monden, Y.: *Toyota Production System - An Integrated Approach to Just-In-Time*. Springer US (1994)
3. Yano, C.A., Rachamadugu, R.: Sequencing to Minimize Work Overload in Assembly Lines with Product Options. *Management Science* 37(5), 572–586 (1991)
4. Bautista, J., Alfaro, R., Batalla, C.: Modeling and solving the mixed-model sequencing problem to improve productivity. *International Journal of Production Economics* 161, 83–95 (2015)
5. Parrello, B.D., Kabat, W.C., Wos, L.: Job-shop scheduling using automated reasoning: A case study of the car-sequencing problem. *Journal of Automated Reasoning* 2(1), 1–42 (1986)
6. Golle, U., Rothlauf, F., Boysen, N.: Iterative beam search for car sequencing. *Annals of Operations Research* 226(1), 239–254 (2014)
7. Morin, S., Gagné, C., Gravel, M.: Ant colony optimization with a specialized pheromone trail for the car-sequencing problem. *European Journal of Operational Research* 197(3), 1185–1191 (2009)
8. Bautista, J., Pereira, J., Adenso-Díaz, B.: A beam search approach for the optimization version of the car sequencing problem. *Annals of Operations Research* 159(1), 233–244 (2008)
9. Bautista, J., Pereira, J., Adenso-Díaz, B.: A GRASP approach for the extended car sequencing problem. *Journal of Scheduling* 11(1), 3–16 (2008)
10. Drexl, A., Kimms, A., Matthießen, L.: Algorithms for the car sequencing and the level scheduling problem. *Journal of Scheduling* 9(2), 153–176 (2006)
11. Fattahi, P., Salehi, M.: Sequencing the mixed-model assembly line to minimize the total utility and idle costs with variable launching interval. *International Journal of Advanced Manufacturing Technology* 45(9–10), 987–998 (2009)
12. Tsai, L.H.: Mixed-model sequencing to minimize utility work and the risk of conveyor stoppage. *Management Science* 41(3), 485–495 (1995)
13. Bautista, J., Cano, A., Alfaro, R.: 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 (2012)

POSTPRINT: J. Bautista, A. Cano, R. Alfaro (2016) A Hybrid Dynamic Programming for solving a Mixed-Model Sequencing Problem with Production Mix Restriction and free interruptions. *Progress in Artificial Intelligence*, DOI 10.1007/s13748-016-0101-5

14. Manavizadeh, N., Tavakoli, L., Rabbani, M., Jolai, F.: A multi-objective mixed-model assembly line sequencing problem in order to minimize total costs in a Make-To-Order environment, considering order priority. *Journal of Manufacturing Systems* 39(1), 124–137 (2013)
15. Boysen, N., Fliedner, M., Scholl, A.: Sequencing mixed-model assembly lines: survey, classification and model critique. *European Journal of Operational Research*, 192(2), 349–373 (2009)
16. Bolat, A.: A mathematical model for sequencing mixed models with due dates. *International Journal of Production Research* 41(5), 897–918 (2003)
17. Bautista, J., Cano, A.: 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 (2011)
18. Bautista, J., Cano, A., Alfaro, R.: Modeling and Solving a Variant of MMSP-W Problem with Production Mix Restrictions. In Prado-Prado, J.C. and García-Arca, J. (Eds.): *Annals of Industrial Engineering 2012*, DOI: 10.1007/978-1-4471-5349-8\_28. Springer-Verlag London (2014)
19. Bautista, J., Cano, J.: Minimizing work overload in mixed-model assembly lines. *International Journal of Production Economics* 112(1), 177–191 (2008)
20. Bolat, A., Yano, C.A.: Scheduling algorithms to minimize utility work at a single station on paced assembly line. *Production Planning and Control* 3(4), 393–405 (1992)
21. Scholl, A., Klein, R., Domschke, W.: Pattern based vocabulary building for effectively sequencing mixed-model assembly lines. *Journal of Heuristics* 4(4) 359–381 (1998)
22. Erel, E., Gocgun, Y., Sabuncuoglu, I.: Mixed-model assembly line sequencing using beam search. *International Journal of Production Research* 45(22), 5265–5284 (2007)
23. Aigbedo, H., Monden, Y.: A parametric procedure for multicriterion sequence scheduling for just-in-time mixed-model assembly lines. *International Journal of Production Research* 35(9), 2543–2564 (1997)
24. Kotani, S., Ito, T., Ohno, K.: Sequencing problem for a mixed-model assembly line in the Toyota production system. *International Journal of Production Research* 42(23), 4955–4974 (2004)
25. Ding, F.Y., Zhu, J., Sun, H.: Comparing two weighted approaches for sequencing mixed-model assembly lines with multiple objectives. *International Journal of Production Economics* 102(1), 108–131 (2006)
26. Rahimi-Vahed, A., Mirzaei, A.H.: A hybrid multi-objective shuffled frog-leaping algorithms for a mixed-model assembly line sequencing problem. *Computers and Industrial Engineering* 53 (4), 642–666 (2007)
27. Bautista, J., Companys, R., Corominas, A.: Heuristics and exact algorithms for solving the Monden problem. *European Journal of Operational Research* 88(1), 101–113 (1996)

POSTPRINT: J. Bautista, A. Cano, R. Alfaro (2016) A Hybrid Dynamic Programming for solving a Mixed-Model Sequencing Problem with Production Mix Restriction and free interruptions. *Progress in Artificial Intelligence*, DOI 10.1007/s13748-016-0101-5

28. Blum, C., Puchinger, J., Raidl, G.R., Roli, A.: Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing Journal* 11(6), 4135-4151 (2011)
29. Bautista, J., Cano, A., Alfaro, R., Batalla, C.: Impact of the Production Mix Preservation on the ORV Problem. In Bielza, C. et al. (Eds.): CAEPIA 2013, LNAI 8109, pp. 250–259, DOI: 10.1007/978-3-642-40643-0\_26. Springer-Verlag Berlin Heidelberg (2013)
30. Blum, C.: Beam-ACO-hybridizing ant colony optimization with beam search: an application to open shop scheduling, *Computers & Operations Research* 32(6), 1565-1591 (2005)
31. Bautista, J., Alfaro-Pozo, R., Batalla-García, C.: GRASP for sequencing mixed models in an assembly line with work overload, useless time and production regularity. *Progress in Artificial Intelligence* 5(1), pp. 27-33 (2016)