Solving the *Mixed Model Sequencing Problem with Workload Minimization* with Product Mix Preservation

Bautista J[1], Cano A[2], Alfaro R[3], Batalla C[4]

Abstract We propose a hybrid procedure based on Bounded Dynamic Programming assisted by linear programming to solve the Mixed-Model Sequencing Problem with Workload Minimization, with serial workstations, free interruption of the operations and with production mix restrictions. We performed a computational experiment with 7 instances from a case study related to the Nissan Powertrain plant located in Barcelona. The results of our proposal are compared with those obtained using a state-of-the-art Mathematical Programming Solver.

Keywords: Sequences, Overload, Dynamic Programming, Linear Programming

[1]Joaquín Bautista (✉ e-mail: joaquin.bautista@upc.edu)
Cátedra PROTHIUS. Dpto. de Organización de Empresas. Universitat Politècnica de Catalunya, Avda. Diagonal 647, 7th floor, 08028 Barcelona, Spain.

[2] Alberto Cano (✉ e-mail: alberto.cano-perez@upc.edu)
Cátedra PROTHIUS. Dpto. de Organización de Empresas. Universitat Politècnica de Catalunya, Avda. Diagonal 647, 7th floor, 08028 Barcelona, Spain.

[3] Rocío Alfaro (✉ e-mail: rocio.alfaro@upc.edu)
Cátedra PROTHIUS. Dpto. de Organización de Empresas. Universitat Politècnica de Catalunya, Avda. Diagonal 647, 7th floor, 08028 Barcelona, Spain.

[4] Cristina Batalla (✉ e-mail: cristina.batalla@upc.edu)
Cátedra PROTHIUS. Dpto. de Organización de Empresas. Universitat Politècnica de Catalunya, Avda. Diagonal 647, 7th floor, 08028 Barcelona, Spain.

* This work is supported by the Spanish Ministerio de Educación y Ciencia under Project DPI2010-16759 (PROTHIUS-III) including EDRF fundings.

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. The Mixed-Model Sequencing Problem with Workload Minimization (*MMSP-W*) is one of the problems that appear on those environments (Yano and Rachamadugu, 1991). This problem 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, it 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 minimize the total overload. A review of the literature about *MMSP-W* can be found at Bautista et al. (2012). Moreover, Boysen et al. (2009) provided an up-to-date review of the literature on sequencing mixed model assembly lines.

On the other hand, the Level scheduling problems class focuses on obtaining regular sequences in production and consumption of components. Among them, one of the related problems is the Product Rate Variation (*PRV*), which is used to preserve the production mix (Miltenburg, 1989).

Our proposal contains: (1) a model for the problem (§2); (2) a model to obtain the overload of a given subsequence to use it as a part of the lower bound of the problem (§3 and §4); (3) reduction of the search space of the procedure through theorems (§5); (4) a dynamic programming procedure to solve this problem that uses linear programming to obtain bounds (§6); and (5) a computational experiment with instances from a Nissan powertrain plant to compare the results offered by the *BDP* procedure with those offered by integer linear programming (§7).

2 Model for the Variant of the *MMSP-W*

For the *MMSP-W* with serial workstations, unrestricted interruption of the operations and production mix restrictions (*pmr*), we take as reference the *M4U3* model, proposed by Bautista et al. (2012). The proposed model *M4U3_pmr* is:

Table 1 Parameters and variables for the model *M4U3_pmr*.

Parameters	
I, K	Set of product types ($i = 1,...,\|I\|$) and set of workstations ($k = 1,...,\|K\|$).
d_i, $\dot{d}_i$	Programmed demand of product type *i* and ideal rate of production for product type *i*, $\dot{d}_i = d_i / T$ ($i = 1,\dots,\|I\|$).
b_k, l_k	Number of homogeneous processors at workstation *k*; and 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*.
$p_{i,k}$	Processing time required by a unit of type *i* at workstation *k* for each homogeneous processor (at normal activity).
t, T	Position index in the sequence ($t = 1,\dots,T$) and total demand. $\sum_{i=1}^{\|I\|} d_i = T$
c	Cycle time, the standard time assigned to workstations to process any product unit
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 0 otherwise.
$s_{k,t}$, $\hat{s}_{k,t}$	Start instant of the operation in t^{th} unit of the sequence of products at workstation *k* ($k = 1,...,\|K\|$) and positive difference between the start instant and the minimum start instant of the t^{th} operation at workstation *k*. $\hat{s}_{k,t} = [s_{k,t} - (t + k - 2) \cdot c]^+$ (with $[x]^+ = \max\{0, x\}$). $\hat{s}_{k,t} \geq 0$ ($\forall k, \forall t$), $\hat{s}_{1,1} = 0$.
$\rho_{k,t}$, $v_{k,t}$	Processing time required by the t^{th} unit of the sequence of products at workstation *k* and processing time applied to the t^{th} unit of the product sequence at station *k* for each homogeneous processor (at normal activity). $v_{k,t} \geq 0$ ($\forall k, \forall t$).
$w_{k,t}$	Overload generated for the t^{th} unit of the product sequence at workstation *k* for each homogeneous processor (at normal activity); measured in time. $w_{k,t} \geq 0$($\forall k, \forall t$).

$$Min \quad W = \sum_{k=1}^{|K|} \left(b_k \sum_{t=1}^{T} w_{k,t} \right) \Leftrightarrow Max \quad V = \sum_{k=1}^{|K|} \left(b_k \sum_{t=1}^{T} v_{k,t} \right) \tag{1.1}$$

subject to:

Constraints (13) – (23) from Bautista et al. (2012)

$$\sum_{\tau=1}^{t} x_{i,\tau} \geq \lfloor t \cdot \dot{d}_i \rfloor \qquad \forall i = 1,\dots,|I|; \ \forall t = 1,\dots,T \tag{1.2}$$

$$\sum_{\tau=1}^{t} x_{i,\tau} \leq \lceil t \cdot \dot{d}_i \rceil \qquad \forall i = 1,\dots,|I|; \ \forall t = 1,\dots,T \tag{1.3}$$

Objective function (1.1) and constraints (13) to (23) corresponds to the mathematical program *M4U3* proposed in Bautista et al. (2012), while the constraints (1.2) and (1.3) 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 $\Delta_Q(X) = \sum_{t=1}^{T} \sum_{i=1}^{|I|} \left(\sum_{\tau=1}^{t} x_{i,\tau} - t \cdot \dot{d}_i \right)^2$.

3 Graph Associated to 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 $\{(t,j), \vec{q}(t,j), \pi(t,j), W(\pi(t,j)), LB\_R(t,j), \Delta_Q(X(\pi(t,j)))\}$, where:

- $\vec{q}(t,j) = (q_1(t,j), \dots, q_{|I|}(t,j))$: vector of satisfied demand.
- $\pi(t,j) = (\pi_1(t,j), \pi_2(t,j), \dots, \pi_t(t,j))$: partial sequence of *t* units of product associated to the vertex $J(t,j)$.
- $W(\pi(t,j))$: partial overload generated by the sequence $\pi(t,j)$.
- $LB\_R(t,j)$: lower bound of the overload generated by the unsequenced products, $d_i - q_i(t,j)$ $(i = 1, \dots, |I|)$.
- $\Delta_Q(X(\pi(t,j)))$: non-regularity of production generated by the sequence $\pi(t,j)$.

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

$$\sum_{i=1}^{|I|} q_i(t,j) = t \tag{1.4}$$

$$\lfloor t \cdot \dot{d}_i \rfloor \le q_i(t,j) \le \lceil t \cdot \dot{d}_i \rceil, \forall i \in I \tag{1.5}$$

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 (1.4) and (1.5). However, it is easy to reduce the cardinal that $J(t,j)$ may present a priori, establishing the following definition of pseudo-dominance ($\prec$): 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 $\pi(t,j_1) \prec \pi(t,j_2)$ if:

$$\pi(t,j_1) \prec \pi(t,j_2) \Leftrightarrow \left\{ \begin{array}{c} [\vec{q}(t,j_1) = \vec{q}(t,j_2)] \ \wedge [LB\_W(t,j_1) \le LB\_W(t,j_2)] \wedge \\ [\Delta_Q(X(\pi(t,j_1))) \le \Delta_Q(X(\pi(t,j_2)))] \end{array} \right\} \tag{1.6}$$

The reduction of $J(t)$ through the pseudo-dominances defined in (1.6) cannot guarantee the optimality of the solutions.

4 Bounds for the Problem

Given a vertex of the stage t, reached through a partial sequence $\pi(t,j)=\{\pi_1(t,j),\pi_2(t,j),...,\pi_t(t,j)\}$, the overall bound for W and a partial bound for the complement $R(t,j)$ associated to the sequence or segment $\pi(t,j)$ can be determined according to the schema presented in Fig. 1.

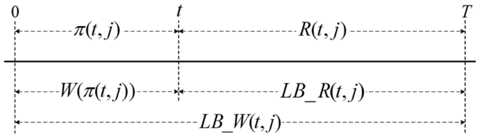

Fig. 1 Bound scheme for a partial sequence $\pi(t,j)$

To obtain the overloads associated to $\pi(t,j)$, in each stage of the procedure we use a mathematical model. Given the subsequence $\pi(t,j)$ of products, the processing times for each workstation k and each cycle τ, $p_{\pi_\tau(t,j),k}$, are known. We can define a mathematical model without assignment variables, $M\_W(\pi(t,j))$:

$$Min \quad W(\pi(t,j))=\sum_{k=1}^{|K|}\left(b_k\cdot\sum_{\tau=1}^{t}w_{k,\tau}\right) \qquad (1.7)$$

Subject to:

$$\rho_{k,\tau}=p_{\pi_\tau(t,j),k} \qquad k=1,...,|K|;\ \tau=1,...,t \qquad (1.8)$$

$$\rho_{k,\tau}-w_{k,\tau}\geq 0 \qquad k=1,...,|K|;\ \tau=1,...,t \qquad (1.9)$$

$$\hat{s}_{k,\tau}\geq\hat{s}_{k,\tau-1}+\rho_{k,\tau-1}-w_{k,\tau-1}-c \qquad k=1,...,|K|;\ \tau=2,...,t \qquad (1.10)$$

$$\hat{s}_{k,\tau}\geq\hat{s}_{k-1,\tau}+\rho_{k-1,\tau}-w_{k-1,\tau}-c \qquad k=2,...,|K|;\ \tau=1,...,t \qquad (1.11)$$

$$\hat{s}_{k,\tau}+\rho_{k,\tau}-w_{k,\tau}\leq l_k \qquad k=1,...,|K|;\ \tau=1,...,t \qquad (1.12)$$

The result of the proposed mathematical model corresponds to $W(\pi(t,j))$. Objective function (1.7) corresponds to the minimization of the partial work overload associated with the subsequence of t product units. Constraint (1.8) assigns the processing times of the t sequenced units. Constraint (1.9) prevents the workoverload to be greater than the processing times. Constraints (1.10)-(1.12) constitute the set of relative start instants of the operations at each workstation.

To obtain a bound of the overload associated to the complement $R(t,j)$, we use the combination of two lower bounds, the first one oriented towards stations (1.13):

$$LB1(t,j)=\sum_{k=1}^{|K|}b_k\cdot\left[TP_k(t,j)-TD_k(t,j)\right]^+ \qquad (1.13)$$

where $TP_k(t,j)=\sum_{i=1}^{|I|}p_{i,k}\cdot\left(d_i-q_i(t,j)\right)$ and $TD_k(t,j)=(T-t-1)\cdot c+l_k,\ k=1,...,|K|$.

And the other bound (1.14) oriented towards products:

$$LB2(t,j)=\sum_{i=1}^{|I|}\left(d_i-q_i(t,j)\right)\cdot LB2(i) \qquad (1.14)$$

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

To determine $LB\_R(t,j)$, we use $LB\_R(t,j)=\max\{LB1(t,j),LB2(t,j)\}$. Moreover, we can obtain a global lower bound of the total overload associated to vertex $J(t,j)$: $LB\_W(t,j)=W(\pi(t,j))+LB\_R(t,j)$.

5 Properties Derived of the Production Mix Restrictions – *pmr* –

Let $\pi=\{\pi_1,\pi_2,...,\pi_T\}$ as a sequence of products for the *MMSP-W pmr*, and $X_{i,t}$ represents the total number of units of product type *i* sequenced during the first *t* production cycles. In these conditions, the fulfillment of the *pmr* restrictions combined with the demand variety results in the following properties:

Theorem 1: If $\lfloor t\cdot\dot{d}_i\rfloor\le X_{i,t}\le\lceil t\cdot\dot{d}_i\rceil$, $\forall i\in I;t=1,...,T$ and $\pi_t=\{j\}$ with $2\le t\le T$, then its satisfied: if $\exists i\in I:\left(X_{i,t}>0\right)\wedge\left(d_i<d_j\right)\Rightarrow X_{i,t}\le X_{j,t}$, $\forall t=2,...,T$.

Proof: If we suppose $\exists i\in I:\left(X_{i,t}>0\right)\wedge\left(d_i<d_j\right)$ such as $X_{i,t}>X_{j,t}$; then we have $X_{i,t}-X_{j,t}\ge 1$. On the other side, given $\pi_t=\{j\}$, then must be satisfied: $X_{j,t}=X_{j,t-1}+1$ and $X_{i,t}=X_{i,t-1}$, and we can write: $X_{i,t}-X_{j,t}=X_{i,t-1}-X_{j,t-1}-1\ge 1\Rightarrow \Rightarrow X_{i,t-1}-X_{j,t-1}\ge 2$. Furthermore, given that $X_{i,t-1}-X_{j,t-1}\le\lceil(t-1)\cdot\dot{d}_i\rceil-\lfloor(t-1)\cdot\dot{d}_j\rfloor\le$ $\le\lceil(t-1)\cdot\dot{d}_j\rceil-\lfloor(t-1)\cdot\dot{d}_j\rfloor$, then we have: $\lceil(t-1)\cdot\dot{d}_j\rceil-\lfloor(t-1)\cdot\dot{d}_j\rfloor\ge X_{i,t-1}-X_{j,t-1}\ge 2$, that is absurd, so the hypothesis $X_{i,t}>X_{j,t}$ is false and, consequently, must be fulfilled $X_{i,t}\le X_{j,t}$, $\forall t=2,...,T$ and $\forall i\in I:X_{i,t}>0$, when $\pi_t=\{j\}$.

Theorem 2: If $\lfloor t\cdot\dot{d}_i\rfloor\le X_{i,t}\le\lceil t\cdot\dot{d}_i\rceil$, $\forall i\in I;t=1,...,T$ and $d_i=d_j$, then $\left|X_{i,t}-X_{j,t}\right|\le 1$ $\forall\{i,j\}\subseteq I;t=1,...,T$.

Proof: On one hand: $X_{i,t}-X_{j,t}\le\lceil t\cdot\dot{d}_i\rceil-\lfloor t\cdot\dot{d}_j\rfloor=\lceil t\cdot\dot{d}_i\rceil-\lfloor t\cdot\dot{d}_i\rfloor\le 1$. On the other: $X_{i,t}-X_{j,t}\ge\lfloor t\cdot\dot{d}_i\rfloor-\lceil t\cdot\dot{d}_j\rceil=\lfloor t\cdot\dot{d}_i\rfloor-\lceil t\cdot\dot{d}_i\rceil\ge -1$.

Then: $-1\le X_{i,t}-X_{j,t}\le 1\Rightarrow\left|X_{i,t}-X_{j,t}\right|\le 1$, $\forall\{i,j\}\subseteq I;t=1,...,T$.

6 The Use of *BDP*

The *BDP* procedure combines features of dynamic programming with features of branch and bound algorithms. The procedure has the following stages:

1. The initial model to obtain $W(\pi(t,j))$ is generated, for *t*=0.
2. The new constraints associated to the new stage *t* are added to the existing model, in order to generate the model used to obtain the partial work overload associated to the subsequence.
3. One of the vertices consolidated in stage $t-1$ is selected, following a nondecreasing ordering of the $LB\_W(t,j)$ values.
4. The selected vertex is developed by adding a new product unit with pending demand. The vertices that do not satisfy the properties (1.4) and (1.5) or the theorems 1 and 2 are not generated. The bound $LB\_W(t,j)$ is obtained.
5. The vertices are filtered. From all the vertices developed in the previous function, a maximum number H of the most promising vertices (according to the lowest values of $LB\_W(t,j)$) are chosen. Those vertices in which their lower bound is greater than Z_0 (known initial solution) or those pseudo-dominated as defined in (1.6) are removed.
6. Finally, the most promising vertices in stage *t* (H vertices as maximum) are consolidated.

The procedure is described more in detail in Bautista and Cano (2011).

7 Computational Experiment

To analyze the validity of the *BDP* procedure for industrial applications, an assembly line from the Powertrain plant of Nissan Spanish Industrial Operations (NSIO) in Barcelona, Spain, was investigated. The line consists on 21 modules or workstations distributed serially in which nine types of engines ($p_1,...,p_9$) were assembled. The data associated to the demand plans and the processing times for each of the nine types of engines can be found at Bautista and Cano (2011). For this manuscript, we have selected 7 representative demand plans (instances 1, 2, 3, 6, 9, 12 and 18), corresponding each one to a representative situation of the demand.

The solutions offered by the *BDP* procedure proposed were obtained under the following conditions and features: (1) *BDP* procedure programmed in C++ (gcc v4.2.1), running on an Apple iMac (Intel Core i7 2.93 GHz, 8 GB RAM, MAC OS X 10.6.7, no parallel code); (2) four windows width were used (*H*=1, 36, 81, 126); (3) the initial solution Z_0 for each window width was the solution obtained by *BDP* with the previous window width, except in the case $H=1$, where Z_0 was established as ∞; and (4) to calculate the lower bounds, $LB\_W(t,j)$, of the overload associated to each vertex in the *BDP* procedure, the solver *Gurobi* v4.6.1 was used, solving the linear program $M\_W(\pi(t,j))$.

The best results for the 7 instances were obtained using the solver *Gurobi* (Bautista et al., 2012) and the *BDP* procedure described in this document. To study the behavior of both procedures we use the following relative percentage deviation:

$$RPD(f)=\frac{f\left(Solution_{Gurobi}\right)-f\left(Solution_{BDP}\right)}{f\left(Solution_{Gurobi}\right)} \qquad f\in\left\{W,\Delta_Q(X)\right\} \tag{1.15}$$

The minimum, maximum and average *CPU* times used by *Gurobi* and *BDP* (*H*=1, 36, 81, 126) are collected in table 2.

Table 2 Minimum, maximum and average *CPU* times needed by *Gurobi* and *BDP* to obtain the solutions for the 7 instances.

	Gurobi	*BDP*			
		H=1	*H*=36	*H*=81	*H*=126
CPU min	7200.0	0.1	428.6	717.9	883.0
CPU max	7200.0	35.0	504.1	1055.9	1572.5
$\overline{CPU}$	7200.0	5.3	471.6	967.0	1411.7

Table 3 collects the values for *W* and $\Delta_Q(X)$ of the solutions obtained using *Gurobi* and *BDP* (*H*=1, 36, 81, 126). *BDP* outperforms to *Gurobi* respect to the value for *W*, with the exception of instances 3 and 12. According to the values for $\Delta_Q(X)$, *BDP* never was worst than *Gurobi*. The average improvements were 4.3% and 14.3% in *W* and $\Delta_Q(X)$, respectively.

Table 3 Values for *W* and $\Delta_Q(X)$ obtained by *Gurobi* and *BDP* (*H*=1,…,126). Column "*Best*" corresponds to the *RPD* obtained for *W* and $\Delta_Q(X)$ from the best solution of the *BDP*.

	Gurobi		*BDP*									
			H=1		*H=36*		*H=81*		*H=126*		*Best*	
	W	$\Delta_Q(X)$	*W*	$\Delta_Q(X)$	*W*	$\Delta_Q(X)$	*W*	$\Delta_Q(X)$	*W*	$\Delta_Q(X)$	*RPD W*	*RPD* $\Delta_Q(X)$
1	186	400.0	368	400.0	166	400.0	-	-	-	-	10.8	0.0
2	383	423.5	-	-	404	393.5	358	369.5	318	327.9	17.0	22.6
3	423	408.5	-	-	444	340.7	-	-	-	-	-5.0	16.6
6	478	420.0	-	-	467	385.0	451	344.0	447	324.3	6.5	22.8
9	751	411.2	-	-	773	440.7	739	360.7	-	-	1.6	12.3
12	287	410.2	-	-	334	425.1	293	385.5	-	-	-2.1	6.0
18	619	419.6	-	-	648	424.7	620	381.1	610	336.3	1.5	19.9

8 Conclusions

We presented a hybrid procedure based on *BDP* assisted by linear programming (used to obtain bounds to solve the *MMSP-W* with *pmr*). This procedure is used to solve 7 instances from a case study of the Nissan Powertrain plant, improving the results on *CPU* times and objective function when compared with those obtained previously using the *Gurobi* solver. Future research will focus on using faster metaheuristics to solve the problem and to establish new models that take into account operations with variable processing time.

9 References

Bautista, J, 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, Alfaro, R (2012) 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.

Boysen, N, Fliedner, M, Scholl, A (2009) Sequencing mixed-model assembly lines: survey, classification and model critique. European Journal of Operational Research 192(2), 349-373

Miltenburg, J (1989) Scheduling Mixed-Model Assembly Lines for Just-In-Time Production Systems. Management Science 35(2), 192-207.

Yano, C A, Rachamadugu, R (1991) Sequencing to minimize work overload in assembly lines with product options. Management Science 37 (5), 572-586.