www.revistadyo.com

Solving the Car Resequencing Problem with mix banks
Resolviendo el problema de Resecuenciación de Coches con el uso de líneas de acumulo paralelas

Valero-Herrero, M.[1], Garcia-Sabater, J. P.[2], Vidal-Carreras, P. I.[2], Canos-Daros, L.[2]

[1] Ford España - Polígono Industrial Ford, s/n, Almussafes, 46440 Valencia
[2] Grupo ROGLE - Departamento de Organización de Empresas. Universitat Politècnica de València. Camino de Vera S/N, 46022 Valencia
mvaleroh@ford.com, jpgarcia@omp.upv.es, pivicar@omp.upv.es, loucada@omp.upv.es

Fecha de recepción: 22-10-2013
Fecha de aceptación: 4-11-2014

Resumen: En la industria del automóvil se utilizan líneas mixtas de montaje de modelos para obtener flujos de producción más eficientes. Aunque la secuencia de los productos en la línea se contempla en el comienzo del proceso, se suelen producir perturbaciones en la secuencia de producción. En ocasiones es necesaria una resecuenciación entendida como la creación de una nueva secuencia. Este trabajo se centra en la resecuenciación mediante el uso de líneas paralelas de almacenamiento (conocidas como mix-bank). Se proponen algoritmos para decidir en qué línea se debe introducir y de qué línea se debe extraer. Los algoritmos son verificados mediante modelos de simulación implementados en SIMIO. Se realiza un análisis para diferentes factores operacionales y se proponen reglas de gestión para su uso en casos reales.

Palabras Claves: Líneas mixtas de montaje, Secuenciación de Coches, Resecuenciación, Mix Bank, Algoritmo dinámico.

Abstract: Mixed-model assembly lines are used at the automotive industry to obtain more efficient production flows. The sequencing of the different units is done at the very beginning of the process, but frequently large perturbations of the original sequence can be observed. These perturbations might be intended or not but to some extent, a resequencing process –to create a new sequence with the available unts- is necessary. This paper focuses in the re-sequencing process by using parallel buffering lines (known as mix banks). Algorithms to decide the line of destination of incoming units and the line of extraction of units to the line are presented and evaluated. The algorithms and their performance have been verified using discrete event simulation software (SIMIO). An analysis considering different operations situations is done and some recommendations to real case users are presented.

Keywords: Mixed-model assembly line, Car sequencing, Resequencing, Mix bank, Dynamic algorithm.

1. Introduction and Brief Literature Review

Mixed-model assembly lines, which are used in car manufacturing, allow the efficient production of a quantity of different carlines and variants. The use of these lines requires overcoming the sequencing problem, to which the literature has paid considerable attention (Boysen et al., 2009). The market requirement of mass customized products eventually creates units that are richer or poorer in terms of options. The objective of the sequencing problem is to determine the succession order of units appearing on the line by considering different objectives such as components consumption levelling, levelling the appearance of variants on the line, equilibrating workload on stations and total line stoppage time (Xiaobo et al., 1999). Moreover a stable sequence will level the components consumption what is important for both internal (Valero-Herrero et al., 2012) and external logistics (Garcia-Sabater et al., 2008; Garcia-Sabater et al., 2010). For instance, non-regular consumption of components affects not only to the kanban supply system but also to the work schedule of the suppliers. In some cases, the authors have observed suppliers with an installed overcapacity of 25% to non smoothed sequences.

Mix banks, also called parallel line banks or selectivity banks in the literature, consist in several parallel lines, with a limited capacity, which are used to store products. Products are introduced into the lines with available capacity and products are extracted from those situated at the front (the first position) of each line. They can be described as a set of parallel and capacitated FIFO lanes. The literature contains different applications of parallel line buffers, which are situated before the painting section where the objective is to minimize changes in color (Cicirello y Smith, 2004; Moon et al., 2005a; Spieckermann et al., 2004). To the best of our knowledge there are very few works on resequencing by parallel lines before the assembly line. The objective of the majority of works involves reordering (Boysen and Zenker, 2013; Meissner, 2010); that is, reordering the sequence to obtain the initial sequence. Very few authors (Choi and Shin, 1997; Moon et al., 2005b; Valero-Herrero et al., 2011a; Valero-Herrero et al., 2011b) have set the objective in sequencing of minimizing the penalty of disregarding a set of defined constraints.

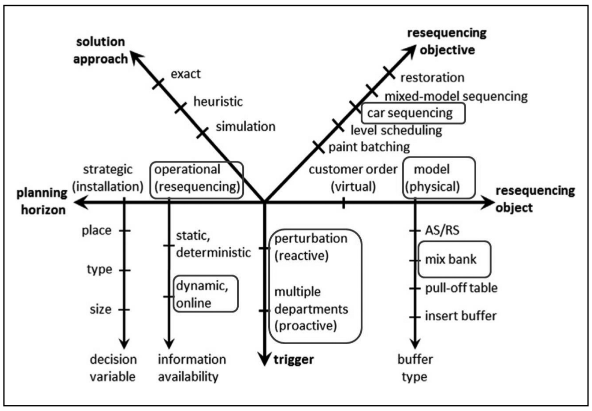
Resequencing in such buffers with several lines in which to store products has two decision problems: buffer input and buffer output. To support this statement two references are used. Moon (2005b) selects the cars which exceed the maximum allowed time at the front of the line, and the maximum storage time allowed. Of these, looks for a unit not disregarding constraints. Of those that do not, selects the one with the lowest smooth ratio. Choi and Shin (1997) employ lines that are assigned to certain products. If the product has no assigned line or if the corresponding line is full, the product is introduced into the line whose last product is the least similar. The goal is to minimize the number of constraints.

As suggested by Boysen et al. in (2010), the problem considered in practical applications tends to be a resequencing problem in which reordering the appearance sequence is sought rather than the sequencing problem as set out in most works in the literature. On automobile assembly lines, resequencing is necessary for two main reasons. Firstly, the different sections making up the process have different sequencing objectives (Poler et al., 1999) and therefore they might be interested in using a different sequence (Ding and Sun, 2007). Secondly, a series of unforeseen events may occur amending the sequence initially considered to be built. For instance, the initial sequence has been calculated upstream in the system considering that every component will be available. But an unforeseen stockout might prevent to assembly specific units, and these units should be held, wherever possible generating the need for a new sequence from each decision point. Other physical causes might provoke a non intended sequence disruption as the presence of regulation buffers or the existence of parallel lines where units are extracted to and later reintroduced in the main stream (García-Sabater et al., 1999).

According to Boysen et al (2012) and depicted in Figure 1, the problem that we are focusing is a Car Sequencing Problem. Any real problem considers different objectives such as levelling components consumption levelling or the appearance of carlines or variants, limiting workload on stations and total line stoppage time. But, in a Car Sequencing Problem all these issues are summarised in terms of constraints or rules. A given sequence will be better than other if the cumulated penalty due to disregarding constraint of the first adds up less than the second.

Figure 1

Graph depicting the resequencing problems classification (Boysen et al., 2012)



The problem is dynamic and online since we will only able to sequence units to the assembly line that are physically at the front of the buffer. The resequencing process is launched when the line request for a new product (following a pull strategy). And we add another trigger for the entry algorithm, which will be triggered when a new unit is incoming to the buffer system.

The problem is operational and it can also be classified simultaneously as reactive and proactive; proactive because it is performed in a buffer placed between two sections requiring different sequences; reactive because resequencing is required due to perturbations caused by breakdowns in previous sections, shortages of material, urgent orders and detected defects. And the resequencing process is done by managing a mix bank. Although it could be done also in a logical manner (by logically swapping units if their characteristics allow it) this problem is not considering it.

This work addresses the selection of units to be extracted is done by computing the quality of the expected output sequence and therefore buffer input and output algorithms are proposed. A simulation model implemented in SIMIO validates the algorithms. The buffer that is the object of this study is a parallel buffer line, which is situated between the trim assembly section and the final assembly section. This buffer separates and therefore gives flexibility to both sections. A set of input and output rules have been deployed, and a set of units and constraints will serve to run the simulation. A set of physical configurations will be considered in order to generate a set of instances that allow to deep in the understanding of the problem. An analysis is performed for the different operational factors such as stoppages or holdups. Different scenarios have been simulated with stoppages in previous lines or with hold ups of specific variants, as the most usual perturbing situations, which might actually happen.

The remainder of the paper is arranged as follow. After presented above the literature on the use of parallel lines buffers, Section 2 describes of how the buffer operates and the model for resequencing by using the algorithms implemented to manage buffer inputs and outputs. Section 3 presents the simulation model implemented in SIMIO. Finally, Section 4 provides the results of the experimentation done for the model performance study and the conclusions drawn from this work are presented in Section 5.

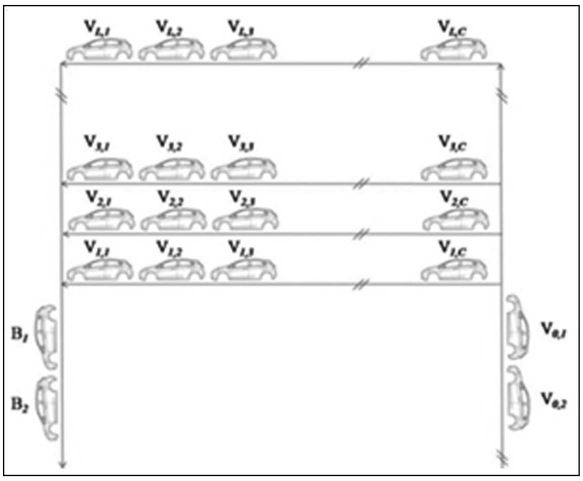
2. Model for resequencing CSP in a mix bank

2.1. A detailed description of the problem

The case hereinafter presented is based in a real situation between the trim section of the assembly line (the lines where the interior of the car is placed) and the chassis section of the assembly line (the lines where bumpers, engines, wheels and other external subassemblies are added). The buffer that is the object of this study is a parallel line buffer (Figure 2) with L lines. This buffer is used to compensate the stoppages that arise in different assembly line areas.

Figure 2

A parallel L line buffer



Cars arrive to the buffer trough the entry section, and then the PLC should send the unit to a specific line. The unit goes to this line and advances until the first available position. The unit will move as far as their precedent units are sequenced, and eventually it will arrive to the first position from which it could be sequenced. As the buffer has L parallel lines, the next unit to the line will be extracted from those at the first position of the line. These are the only possible candidates to be chosen, and on each step one of them should be scheduled.

2.2. Notation

The notation employed in developing the algorithms is presented in Table 1.

2.3. Input algorithm

The input algorithm is executed when a unit $V_{0,k}$ arrives to the entry of the buffer. To place products in the buffer, the space available on each line and a criterion relating the line content with the product to be inserted are taken into account. A weight per line, is obtained in accordance with the decision rule considered. Finally, the line with a heavier weight will be assigned to the product.

Hereinafter details of each rule are provided. Combinations of them are possible, although they have not been analyzed on this paper. If a line is empty the unit goes to it on any case. If there is no empty line, the algorithm will choose:

$$l \;\Big|\; w_i^m = \max_i (w_i^m) \qquad [1]$$

In order to evaluate w_i^m the following six rules have been deployed.

- Rule 1: The line of destination is chosen randomly.

$$w_i^1 = rnd(\ ) \qquad [2]$$

- Rule 2: To the emptier line. The weight assigned to each line is inversely proportional to the quantity of products already on this line (if $q_i = 0$ then this is the line chosen).

$$w_i^2 = \frac{1}{q_i} \qquad [3]$$

- Rule 3: To the more similar line comparing number of identical characteristics. The product to be

Table 1
Notation

Index	
i (L)	The index referring to buffer lines.
j (C)	The index referring to the position within each line, with the first position being closest to the output.
k (NR)	The index referring to product properties.
m (M)	The index referring to input algorithm.
Parameters	
$V_{0,k}$	Value of property k of the product situated at the first position before the buffer. It takes value 1 if it has this property, and 0 otherwise.
$V_{i,j,k}$	Value of property k at position j of buffer line i. It takes value 1 if it has this property, and 0 otherwise.
$B_{j,k}$	Value of property k at position j of the buy-off, a line before the final assembly line. It takes value 1 if it has this property, and 0 otherwise.
L	Number of buffer parallel lines.
C	Capacity of each buffer line.
R	Number of constraints.
M	Number of input algorithms.
P	Depth. Number of units used to make comparisons.
Z_k	Weight associated with disregarding constraint k.
M_k:N_k	Sequencing rule. No more Mk products of the consecutive N_k with option k.
M_iX_k	Proportion of products whose k property is equal to 1. It is calculated in relation to the total products. Data defined in the products input
Variables	
q_i	Quantity of products on one line.
w_i^m	A value that quantifies the weight of each line to assign a product to a line according to the input algorithm m.
l	Line assigned.

inserted is compared with the last one on the line for each common property.

$$w_i^3 = \sum_k \left| V_{0k} - V_{iq_i k} \right| \qquad [4]$$

- Rule 4: To the more similar line comparing number of identical characteristics weighted by the relevance of each characteristic.

$$w_i^4 = \sum_k Z_k \left| V_{0k} - V_{iq_i k} \right| \qquad [5]$$

- Rule 5: To the less similar line comparing number of identical characteristics. The product to be inserted is compared with the last one on the line for each common property.

$$w_i^5 = \sum_k \left| V_{0k} - V_{iq_i k} \right| \qquad [6]$$

- Rule 6: To the more similar line comparing number of identical characteristics weighted by the relevance of each characteristic.

$$w_i^6 = \sum_k Z_k \left| V_{0k} - V_{iq_i k} \right| \qquad [7]$$

In Table 2 the rule used in each of the input algorithms is shown.

2.4. Output algorithm

The output algorithm chooses the following unit to be sequenced when the chassis line request a given product. The algorithm returns the line from which the following product should be extracted. The set of consecutive units creates the sequence. Sequence quality is measured according to the total penalty obtained from disregarding constraints. The lower the summatory of penalties, the better this sequence is considered.

Table 2
Input Algorithm

Algorithm	*Rule applied*
A1	Random
A2	Emptier line
A3	More similar line
A4	More similar line comparing weight of identical characteristics
A5	Less similar line comparing number of identical characteristics
A6	Less similar line weighting comparing weight of identical characteristics

In order to define the unit to be chosen, the algorithm chooses the first unit of the best possible sequence of P units that can be created on the given moment that is requested. A sequence is possible if the units sequenced might be extracted when requested since they are at the first position of their line. The algorithm should explore the whole space of possible sequences of P consecutive elements that might be sequenced. The output algorithm has been run with different depths of analysis (from 1 to 5). For the sake of clarity the output algorithm that varying the value of P creates have been named as in Table 3.

Table 3
Output algorithms

Algorithm	*Depth applied*
B1	P = 1
B2	P = 2
B3	P = 3
B4	P = 4
B5	P = 5

The number of possible sequences is L^P and since exploring the whole set of possible sequence is prohibitive for any a very basic branch and bound algorithm has been developed that stops the search on a given branch if the penalty of the already created subsequence is worst than the best available sequence. Thus, after the assembly line requests a new unit, the output algorithm looks for the best sequence that can be created, and the very first unit is sequenced. Therefore the output algorithm will have to be performed for each unit to be sequenced.

3. Design of the Experiment

To understand the performance of the system and to validate the algorithms presented, an experiment has been design using SIMIO as discrete event simulation named. The buffer configuration considered is based on a buffer installed in the plant of an automotive manufacturer located in Spain. Two parallel assembly lines (so called the trim system) feed a single line (so called the chassis system) through a 4 lines mix bank. For the purpose of analyzing the sensitivity of the problem to the physical configuration we will run models with 2,4 6 and 8 lines. In order to define the products to be assembled, the characteristics affected by constraints are considered. Two sequences, each of around +1000 consecutive units have been created, using as the base the real sequence of units of a given day. The sequence has been used to feed the buffer that is the object of study. Therefore each sequence will repeat 2000 times the process of assigning line and sequencing units.

A set of 6 constraints has been created (Table 4). In order to create this set we have used the set of constraints that were performing on a given moment in the same day that was considered. Although they are real not necessarily are representative since users might change the constraints that are being considered. It is out of the scope of this paper to do a sensitivity analysis of the number and difficulty of the constraints, but the use of this set might give some insight. It has to be said that too many constraints might create an almost random sequence (if every unit disregards many constraints the choice of the best suitable unit is unclear), but a number of constraints too low will derive in a problem that is too easy to be solved.

Table 4
Spacing Constraints Scenario 1

Constraint k	*M*	*N*	*W*	*MIX*
k1	2	4	200	22%
k2	3	4	50	34%
k3	4	5	200	35%
k4	2	3	50	26%
k5	1	3	50	30%
k6	2	3	200	44%

Line 1 of Table 4 should be interpreted as follows: a given characteristic has a constraint related k1, the perfect sequence should not incorporate more than 2 units having the characteristic out of 4 consecutive units. If, for any reason, we have to disregard the constraint, the unit will have a penalty of 200 points.

The percentage of units on the overall schedule having this constraint is 22%. Characteristics 1 and 2 are only in units coming from one of the lines (so called Trim A), characteristics 3 and 4 are coming from the other line (so called Trim B) and characteristics 5 and 6 are present in units coming from both trims.

The practical experience shows that the real problem comes when special circumstances occur at the system. Among all the possibilities two are especially relevant: 1) that one of the lines has a short stop (and then the entry sequence is not as smoothed as desired; 2) the necessity of having to hold some units at the buffer due to specific stock outs or other concerns.

The stoppage of a line will be simulated of 10 cycles for every 100 cycles. The stoppages will be always from the same line, because it is assumed to be a worst case situation. The hold of a given characteristic is related mainly when a stock out is noticed. Although longer hold ups might exist, only short holds are created by creating 50 cycles of holdup of specific variants every 200 cycles and the characteristic 6 is the one affected by the hold. By mixing up the special event simulation 4 scenarios are to be analyzed ("no stop, no hold"; "stop, no hold"; "no stop, hold", "stop, hold").

The simulation has been done using SIMIO as a discrete event simulator. On Figure 3 a snapshot of the simulation model is presented. In http://goo.gl/RMtJKX it might be seen a short video of the simulation running.

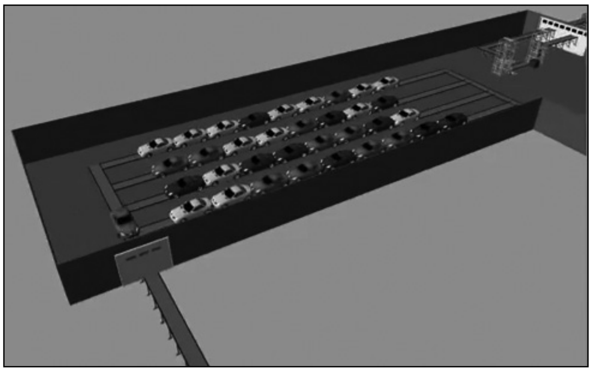


Figure 3
Snapshot of the simulation model

4. Results analysis

The quality of the sequence has been measured using the mean of penalty points per unit. Although the lower bound is zero (since every constraint is individually possible) reality is that the combination of constraints generates that no disregarding any constraint is a very difficult issue when no every unit is available on any given time.

Figure 4 compares the performance of the input algorithms and the depth of analysis of the output algorithms. Although the previous comment of the outperformance for the "less similar" rule is still valid, what can be said from the analysis of the results shown that this advantage is clear only as less myopic is the algorithm.

In order to eliminate side effects the figure 5 only represents the input algorithm A6. It clearly shows that the improvement of the system when reducing the myopia of the output algorithm is enlarged as larger is the number of lines in parallel.

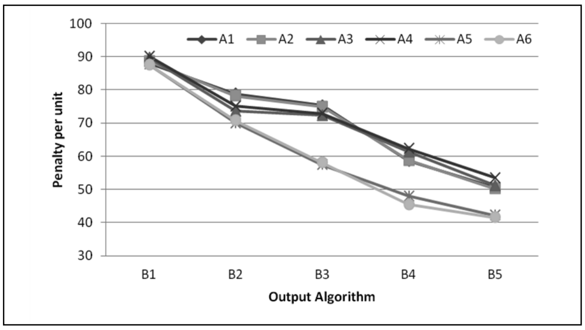


Figure 4
Penalty per Unit (Average) vs Output Algorithm for different Input Algorithm

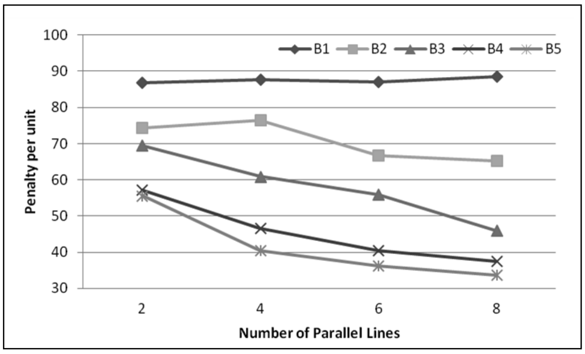


Figure 5
Penalty per Unit vs Number of Parallel Lines for different Output Algorithm for the Input Algorithm A6

Figure 6 illustrates the simulation results for the hold and stop instances separately. The horizontal axis shows the input algorithms and presents results of having different number of lines.

On regular conditions the input algorithms that outperform are those that select the line looking for the less similar unit. It should be said that on reality the "more similar" options is the one used. It is very interesting to observe that the holds reduces significantly the performance of the system only for the "less similar" alternative. Of course, having more lines to select is better, but it can be said that, although the improvement from 2 to 4 is evident, and also from 4 to 6. The improvement, with the number of constraints that we are considering is less evident. Other experiments show that with a larger number of constraints, more lines is better.

An interesting effect appears when comparing the performance of the different input algorithms against different parameters for the output algorithm when analyzing the different disturbing events (Figure 7). The hold situation eludes the improvement that can be seen by choosing a different input algorithm.

On reality according to Figure 7, when manually handled, the operators tend to group units by similarity of the most difficult constraints (the input algorithm A4) and they chose the next unit by looking only to the first unit available on the line. Although the experimentation shows that is not, by far, the best solution, results show that is very reasonable. If you are no able to compute in-depth sequences (and they are usually not able), since they have to sequence a unit every 30 seconds, to have different units on the same line, will create confusion, and it is not going to give you any real advantage.

And even if they try to be "less myopic" they know that when something wrong happens (a massive hold) the benefit of being more clever on the decision making disappears. And, given that, most people will try to manage for the "worst case situation" using less efficient algorithms has a performance that is not worse than the efficient algorithm.

5. Conclusions

The automotive industry has utilized mixed-model assembly lines where a wide variety of models is produced on the same assembly line. On this type of lines, sequencing systems dynamics substantially affects productivity on the line. Despite the importance of resequencing dynamics, the literature in this domain is scarce.

This work presents a problem and tries to analyze it by the use of simulation. A resequencing model for

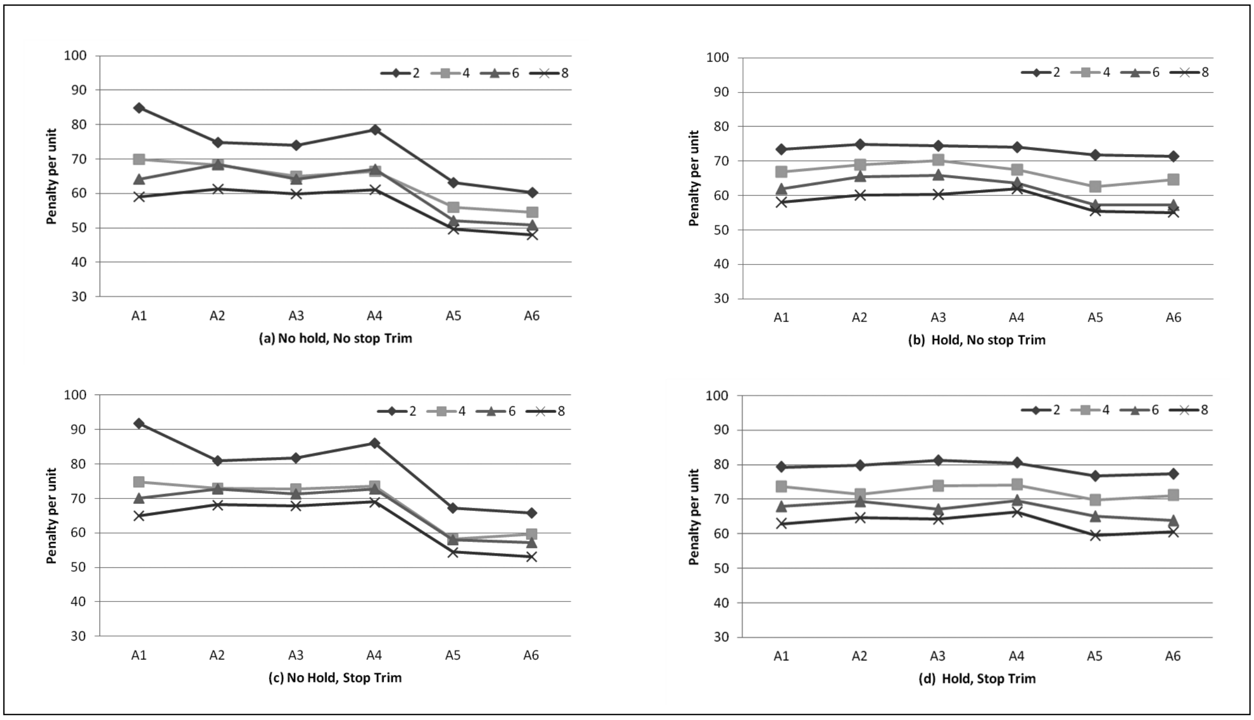


Figure 6

Penalty per Unit (Average) vs Input Algorithm with number of parallel lines (2,4,6,8)

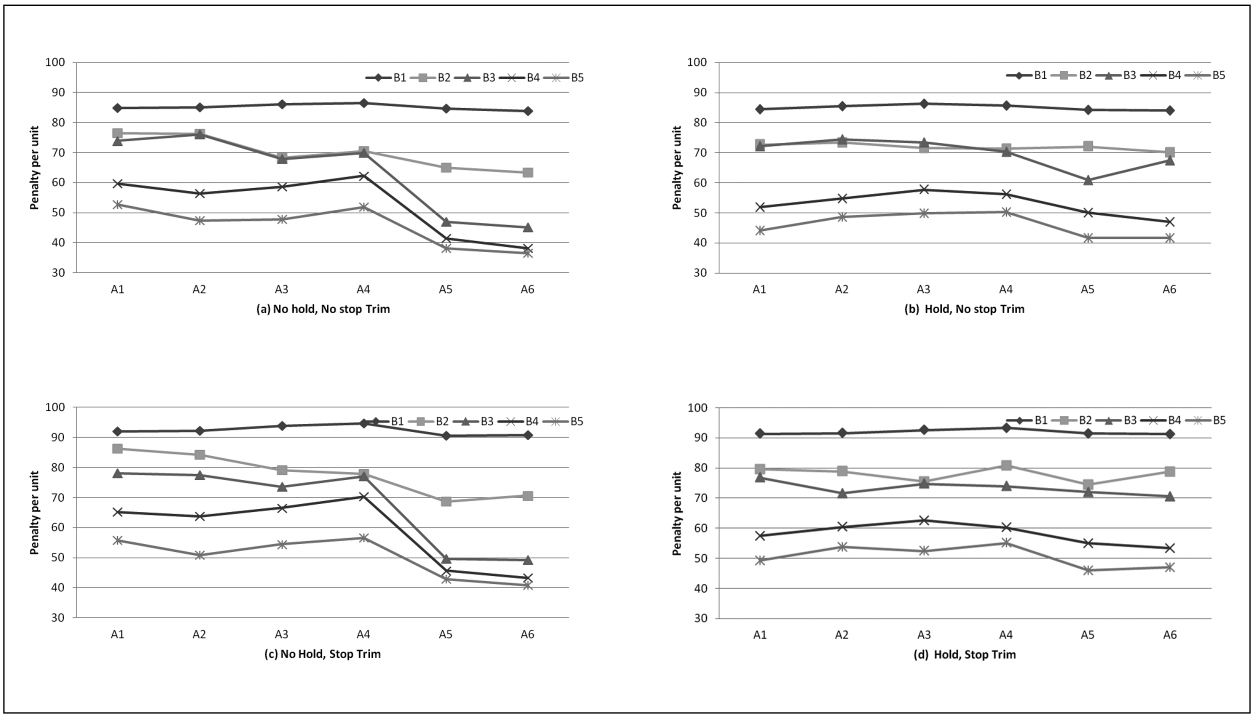


Figure 7

Behaviour of input and output algorithms with disturbing events

a parallel line buffer has been developed and the basics of this system have been developed. Aspects such as size of the buffer are usually defined on real situations, but in some cases investments might be necessary since there is a direct relation between size of the buffer and quality of the sequence.

The control algorithms of the system act both at the entry and at the exit of the system. Usually managers tend to consider only the output side, but the results show that there is a way of improvement at the entry. The results obtained show that, depending on the given situation, the choice of the best input algorithm may vary, and the proposed techniques are fairly simple but better than those regularly used. Therefore it could be interesting to understand how to collect information about events that are to happen before or after the buffer.

Another important point raised is model performance when changing the operational conditions of the line. Different scenarios have been simulated with stoppages in previous lines, a situation which actually occurs quite often. Different results might be obtained for each algorithm in accordance with the variety of the models in the buffer, and this variation is due to the stoppages occurring upstream of the buffer.

There are real advantages, in terms of quality of sequence, to increase the intelligence of the buffer management and usually these improvements are cheaper than enlarging the buffer (which is the other alternative). But when changing the way of working (for instance changing the manual manager by a automatic system), it will be interesting to know that the actual way of manually handling the operation has a logic based on the worst case situation. Changing from a manual management system will require analyzing the worst cases that the users have found, because there is logic behind this way of working.

Future work will have to thoroughly analyze the behavior of the algorithms with different sets of constraints, and with different sets of units. More risky scenarios might be analyzed and they will originate insight for better input algorithms. Improving the so-called output algorithm to allow larger depth of analysis is also a very relevant issue to be analyzed.

Finally, the development of a dynamic model is proposed to amend the algorithm applied, according to the situation in the buffer, caused by stoppages or other unforeseen events. Defining a quantitative characterization of buffer situations will help to understand when switch from one algorithm to another.

Acknowledgements

The work described in this paper has been supported by the project "CORSARI MAGIC DPI2010-18243" by the Ministerio de Ciencia e Innovación del Gobierno de España within the Program de "Proyectos de Investigación Fundamental No Orientada".

This paper also has been written with financial support from the Project "Path Dependence y toma de decisiones para selección de herramientas y prácticas de Lean Manufacturing" (PAID-06-12-SP20120717) of the Universitat Politècnica de València.

The authors wish to thank the comments of the two anonymous reviewers that have helped to improved the previous version of the paper.

References

BOYSEN, N., FLIEDNER, M., SCHOLL, A. (2009). «Sequencing mixed-model assembly lines: Survey, classification and model critique». *European Journal of Operational Research,* 192(2), pp. 349-373.

BOYSEN, N., GOLLE, U., ROTHLAUF, F. (2010). «The Car Resequencing Problem». *Working Papers in Information Systems and Business Administration,* Johannes Gutenberg-University Mainz.

BOYSEN, N., SCHOLL, A., WOPPERER, N. (2012). «Resequencing of mixed-model assembly lines: Survey and research agenda». *European Journal of Operational Research,* 216(3), pp. 594-604.

BOYSEN, N., ZENKER, M. (2013). «A decomposition approach for the car resequencing problem with selectivity banks». *Journal Computers and Operations Research,* 40(1), pp. 98-108.

CHOI, W., SHIN, H. (1997). «A real-time sequence control system for the level production of the automobile assembly line». *Computers & industrial engineering,* 33(3-4), pp. 769-772.

CICIRELLO, V. A., SMITH, S. F. (2004). «Wasp-like Agents for Distributed Factory Coordination». *Autonomous Agents and Multi-Agent Systems,* 8(3), pp. 237-266.

DING, F.H., SUNG, H. (2007). «Sequence alteration and restoration related to sequenced parts delivery on an automobile mixed-model assembly line with multiple departments». *International Journal of Production Research,* 42(8), pp. 1525-1543.

GARCÍA-SABATER, J., RODRIGUEZ, A., POLER, R., LARIO, F. C. (1999). «Cumplimiento de la secuencia prevista y uso de almacenes reguladores». *III Jornadas de Ingeniería de Organización.* ISBN 84-95355.

GARCIA-SABATER, J. P., GARCIA-SABATER, J. J., MARIN-GARCIA, J. A. (2008). «Optimización de los recursos humanos a través de las TICs. Resultados de la reorganización laboral de un proveedor logístico». *Economía y industrial,* 370, pp. 143-151.

GARCIA-SABATER, J. J., GARCIA-SABATER, J.P., VIDAL-CARRERAS, P. I., MAHEUT, J. (2010). «Simulación de la entrega secuenciada de asientos en un proveedor de primer nivel del sector del automóvil». Proceedings of 4th International Conference on Industrial Engineering and Industrial Management, pp. 268-275.

MEISSNER, S. (2010). «Controlling just-in-sequence flow-production». *Logistics Research,* 2(1), pp. 45-53.

MOON, D. H., KIM, H. S., SONG, C. (2005a). «A Simulation Study for Implementing Color Rescheduling Storage in an Automotive Factory». *Simulation,* 81(9), pp. 625-635.

MOON, D. H., SONG, C., HA, J. H. (2005b). «A Dynamic Algorithm for the Control of Automotive Painted Body Storage». *Simulation,* 81(11), pp. 773-787.

POLER, R., GARCIA-SABATER, J. P., RODRIGUEZ, A., LARIO, F. C. (1999). «Un algoritmo greedy para la agrupación de colores en la secuencia maestra de una empresa de automoción». *III Jornadas de Ingeniería de Organización,* ISBN 84-95355.

SPIECKERMANN, S., GUTENSCHWAGER, K., VOSS, S. (2004). «A sequential ordering problem in automotive paint shops». *International Journal of Production Research,* 42(9), pp. 1865-1878.

VALERO-HERRERO, M., GARCIA-SABATER, J. P., CORONADO-HERNANDEZ, J. R., MAHEUT, J. (2011a). «Planteamiento dinámico del problema de secuenciación en líneas de montaje con mezcla de modelos». *Proceedings of 5th International Conference on Industrial Engineering and Industrial Management,* pp 288-296.

VALERO-HERRERO, M., GARCIA-SABATER, J. P., MAHEUT, J. (2011b). «An approach to the real circumstances of the car sequencing problem». *Proceedings of the 41st International Conference on Computers & Industrial Engineering,* pp. 295-300.

VALERO-HERRERO, M., GARCIA-SABATER, J. P., MORELL-MONZO, V. (2012). «La alimentación a una línea de montaje: Procedimientos y Costes». *Proceedings of 5th International Conference on Industrial Engineering and Industrial Management,* pp. 774-780.

XIAOBO, Z., ZHOU, Z., ASRES, A. (1999). «A note on Toyota's goal of sequencing mixed models on an assembly line». *Computers & Industrial Engineering,* 36(1), pp. 57-65.