MODELING SELECTIVITY BANKS FOR MIXED MODEL ASSEMBLY LINES

Alex Blatchford[a][1], Yakov Fradkin[a][2], Oleg Gusikhin[a][3], Ravi Lote[b][4], Marco Pucciano[c][5], Onur Ulgen[b,d][6]

[a] Ford Motor Company, USA
[b] PMC, USA
[c] University of Genova, Italy
[d] University of Michigan-Dearborn, USA

[1]ablatchf@ford.com, [2]yfradkin@ford.com, [3]ogusikhi@ford.com,
[4]rlote@pmcorp.com, [5]marco.pucciano@gmail.com, [6]ulgen@umich.edu

ABSTRACT

This paper presents a simulation study on the influence of different designs and operational policies of selectivity banks on the efficiency of automotive assembly sequencing. Specifically, we analyse whether or not a bypass lane could improve the performance of selectivity banks. A special purpose simulation framework has been developed and implemented using Witness simulation integrated with an Excel file containing the dataset and the decision making algorithms.

1. INTRODUCTION

A mixed-model automotive assembly plant is typically comprised of several different production areas including the body shop, paint shop and final assembly as shown in Figure 1. The plant's efficiency largely depends on the sequence of different types of vehicles moving down the line. The quality of the vehicle sequence can be defined by how well it satisfies the preferences or constraints and supply chain considerations posed by different production areas: see for example Fradkin (2006).

In general, different production areas have different preferences in terms of ideal sequences (Blatchford 2008). For example, paint operations prefer batching of similar colours to minimize changeovers (Spieckermann *et al.* 2004), while body shop and final assembly prefer smoothing of similar vehicle types to balance the labour and supply chain consumption rates.

To accommodate these conflicting requirements, many automakers have faced the problem of resequencing the vehicle flow by adding storage/ resequencing areas between the various assembly areas. Figure 1 shows these areas within the assembly plant, denoting them with ellipses. In particular:

- the White Body Storage (WBS) is located between the body shop and the paint shop;
- the Colour Rescheduling Storage (CRS) is located within the paint shop;
- the Painted Body Storage (PBS) is located between the paint shop and the trim shop.

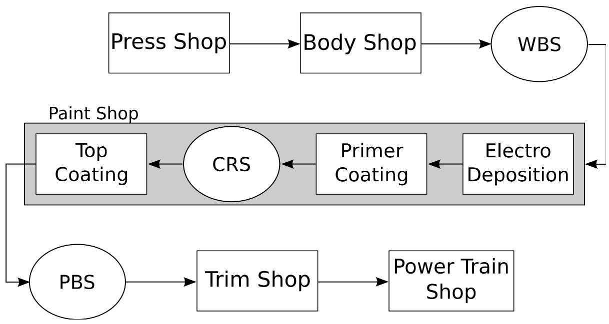

Figure 1: Storage Areas Scheme.

Those not only allow absorption of fluctuations in output from the previous phase and in input for the next one, but they can be also used as an opportunity to resequence the stream of items in order to better satisfy the constraints of the downstream process.

Under the term "constraints," we mean rules that are used in order to prevent certain sequences of models on the assembly line – as explained in Nagane (2002) – and therefore even prevent situations in which the lane could be imbalanced.

The resequencing activities could be useful not only between the production areas having different sequencing requirements, but also between the areas having the same constraints. For example, a defect could be discovered on a certain vehicle in the upstream area, requiring that the vehicle be put aside for the repair. This disrupts the initial, possibly perfect, sequence. To contain the effects of the disruption, some form of resequencing may be needed before feeding the vehicles to the downstream area.

Several authors such as Inman (2003), Inman *et al.* (1997), and Gusikhin *et al.* (2007) analysed the resequencing structures and algorithms. The commonly used resequencing structures are:

- automated storage and retrieval system (ASRS or AS/RS);
- pull of table;
- repair holding area;
- selectivity bank (SB).

Each of the above has advantages and disadvantages. For example, ASRS has the best resequencing capabilities (it allows random access) but is typically very expensive (both in initial investment and in work in progress costs) and requires a lot of physical space (which means that ASRS cannot be easily added to an existing plant). In comparison with ASRS, SB has marginal resequencing capabilities (described in Section 3) but is simpler and less expensive, which allows its use when ASRS is not a viable option.

The aim of this work is to develop methods and tools that could improve the selectivity bank's performance. Particularly we study whether or not the introduction of a bypass lane could improve SB's resequencing capabilities. We built a tool that could test whether adopting a bypass lane could be useful, given the characteristics of the plant and the vehicles that have to be assembled.

We used Modelling & Simulation (M&S) to achieve these goals. The benefits of M&S have been widely discussed in the literature. Karakal (1998) and Banks (1998) point out that the application of M&S is particularly useful in order to perform what-if analyses and analyse the behaviour of complex systems. The usefulness of the simulation approach to problems of this type has been also pointed out in Han *et al.* (2003) (in which the authors used a similar approach in order to reduce the number of colour changes within the paint shop), in Ulgen (1994), and in Park *et al.* (1998).

We used a simulator to represent the plant and study the behaviour of the system with and without (respectively WIB and WOB) a bypass lane. In this paper, we describe the theoretical advantage related to the usage of a bypass lane and discuss the results.

This paper is structured as follows: in Section 2 we introduce the selectivity banks and their characteristics; in Section 3 we describe the simulation model that we used to perform the analyses. The results and conclusions are given in Sections 4 and 5.

2. SELECTIVITY BANKS AND BYPASS LANE

Selectivity bank (SB) is a kind of a buffer that is commonly used for both storage and resequencing purposes, as shown in Narayanaswamy *et al.* (1997). As highlighted in Figure 2, it usually is a multilane structure in which each lane can carry any type of model and that is characterized by a certain length – i.e. the maximum number of vehicles that it can contain. In some applications, particular lanes may be dedicated to a certain single model (in order to be able to easily access it) or to a subset of models.

SB could be a traditional structure with a certain number of lanes (each one characterized by a certain length) – like the one in Figure 2 – but it could also include a bypass lane as in Figure 3 and/or a return lane like the one in Figure 4 (just the first two configurations have been considered in this work).

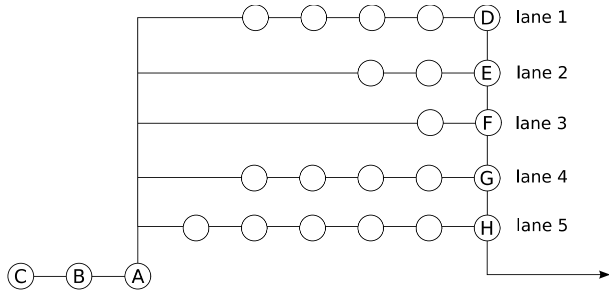

Figure 2: Example of Selectivity Bank Without Bypass Lane.

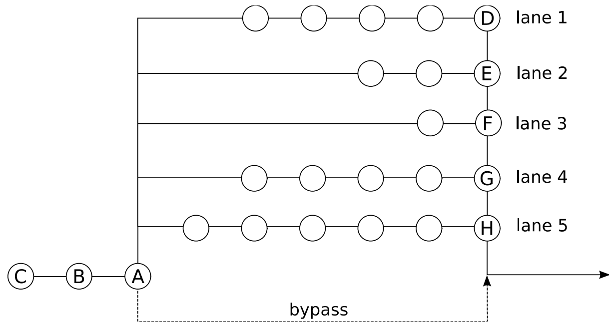

Figure 3: Example of Selectivity Bank With Bypass Lane.

The two major processes that have to be managed in SB are the input and the output ones. The input process determines in which lane a certain item has to be sent when it enters the SB, while the output process determines which vehicle should be sent downstream to the next phase. These two processes could be managed in different ways, typically using heuristics. Two important characteristics of SB are the maximum number of selection and retrieval points that determine the resequencing capabilities of this kind of buffer; in both cases they're equal to the number of lanes (bypass lane included).

3. THE SIMULATION MODEL

To test how different SB designs could affect system performance, we created a simulation model. Model building methodology included defining the problem, designing the study, designing the conceptual model, formulating inputs and assumptions, building and verifying the model, conducting experiments and documenting the results. We developed the following three components of the model:

- a simulator (realized using Witness);
- input data (in an Excel spreadsheet);
- decision making algorithms (realized in VBA within the Excel file).

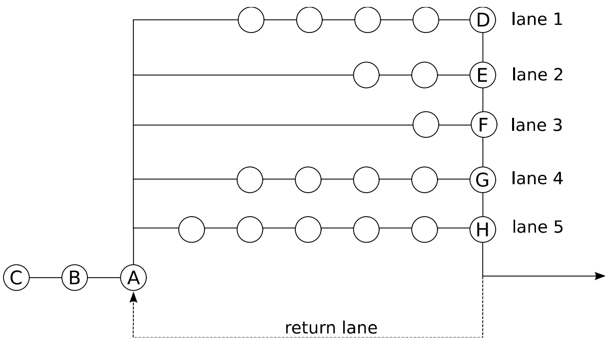

Figure 4: Example of selectivity bank with Return Lane.

One of the modelling decisions was how to represent the assembly line constraints. We reduced them to "no more than x out of y" constraints specifying the maximum allowed number (x) of vehicles that have a certain option out of (y) of consecutive items. Figure 5 shows how these constraints are stored in the Excel file.

We defined a metric to measure the performance of the system: with each constraint violation we associated a penalty (time lost, measured in seconds). According to this framework the performance of the system is measured as the total penalty that has to be sustained after having resequenced the whole input sequence. Note that these penalties could be different from plant to plant and therefore just a comparison between different designs of the same plant is meaningful. In other words, this definition doesn't allow us to compare performance of different plants.

The model's three components are linked together as shown in Figure 6: Witness drives the simulation and passes input data to the Excel spreadsheet when a decision making process is needed; within the Excel file the VBA algorithms use this information to decide to which lane a vehicle that is currently entering the SB should be sent and from which lane a vehicle should be retrieved for sending downstream of the SB. The decision is communicated to the simulator via the Excel spreadsheet.

In summary, the developed framework is based on the following assumptions:

- The possibility of implementing a bypass lane is considered (as previously mentioned).
- The lanes (but not the bypass lane) are shared between the various vehicle models.
- The constraints are considered as explained above.

Model verification was completed to make sure that the algorithm and programming is error free. The modelling analysts used various techniques to verify the model. These techniques included:

	Constraint	1	2	3	4
	x	2	3	2	2
	in y	10	4	6	6
(secs)	Penalty	60	30	280	60

Figure 5: How the Constraints are Stored in the Excel file.

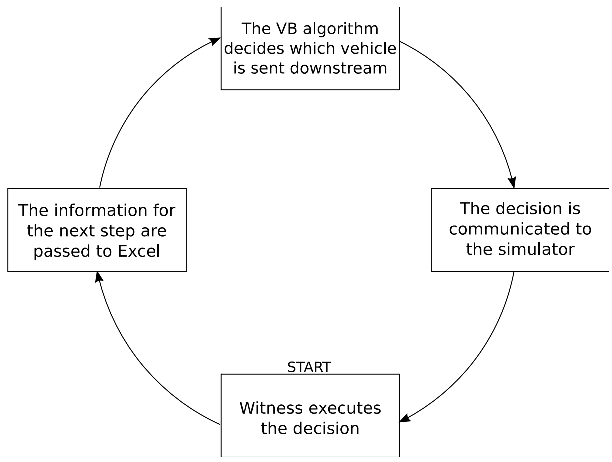

Figure 6: How the Three Components are Linked Together.

- Using modular programming concepts;
- Sending only one vehicle into the model and ensure that the flow and cycle time is correct;
- Undertaking "directional testing" (e.g., if a cycle time increases, throughput should decrease or remain the same);
- Executing deterministic runs: removing all randomness and making sure that the results match static analysis;
- Error-trapping the events that "can't" happen;
- Making sure that the time units and distance units are consistent throughout the model;
- Using simulation traces extensively.

The implementation was corrected based on the errors found in the underlying model, often resulting in retesting to ensure integrity of the programming done. However, since the model is built at the abstract level, validation was not possible.

While building and verifying the model, we used the help of subject matter experts from assembly plants, utilized input data coming from the real plants and performed sensitivity analyses.

Table 1: The considered selectivity buffer's design for the 1st Test Case.

#	Lanes	Length	Bypass Lane
1	5	8	No
2	10	8	No
3	5	8	Yes
4	10	8	Yes

Table 2: The considered selectivity buffers' design for the 2nd Test Case.

#	Lanes	Length	Bypass Lane
1	3	17	No
2	2	17	Yes
3	5	7	Yes
4	6	7	No

Table 3: Some Details of the 1st Test Case.

3.1.	% of vehicles	x	y
Option 1	6.9	2	10
Option 2	72.1	3	4
Option 3	20.9	2	6
Option 4	14.4	2	6
Option 5	1.6	1	8
Option 6	13.7	5	7
Option 7	3.1	2	10
Option 8	4.9	2	8
Option 9	39.1	3	4
Option 10	2.1	1	3
Option 11	21.9	2	5
Option 12	4.6	1	8
Option 13	14.4	2	6
Option 14	1.1	2	5

Table 4: Some Details of the 2nd Test Case.

3.2.	% of vehicles	x	y
Option 1	61.9	2	3
Option 2	7.8	1	2
Option 3	14.4	1	3
Option 4	7.6	1	3

4. RESULTS

We used two dataset from different plants, as shown in Tables 1 and 2. With the first dataset we've used a perfect (having no constraint violations) input sequence that has been disrupted considering four different levels of repair (5%, 10%, 15%, 20%). For the second dataset we've just considered a random input sequence (and therefore we didn't use any level of disruption/repair as in the previous case).

These different choices have been made in order to create a more realistic model of the considered plants.

The first test case has 14 constraints and an input sequence composed of 14,814 vehicles, while the second one has only 4 constraints and 8,809 vehicles. The characteristics of the input sequences of the considered test cases are shown in Tables 3 and 4; the second column in those tables shows the percentage of vehicles in the input sequence that had a particular option.

The first dataset is more complex and presents more difficult constraints than the second one. The second dataset is simpler: it only has four constraints, of which only the first one is tight, and the remaining three should be easily satisfied.

The obtained results for the 1st test case are shown in Figures 7, 8, and 9. These tests have been performed using two different heuristics in order to manage the SB, called policy 1 and 2. Policy 1 – that is a greedy algorithm – has been designed in order to consider just the explicit constraints of the downstream phase of the production plant and perform a local optimisation at every step; this means that at every iteration the algorithm evaluates which vehicle determines the minimum selection penalty.

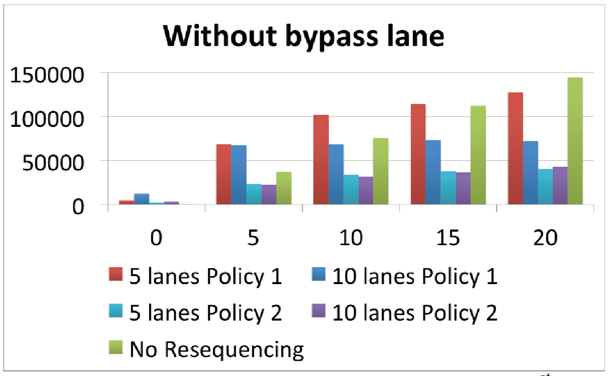

Figure 7: Results Without Bypass Lane for the 1st Test Case.

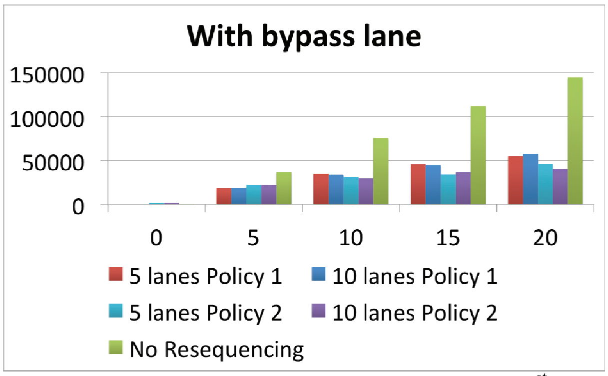

Figure 8: Results With Bypass Lane for the 1st Test Case.

Policy 2 is more complex than the previous one; it considers in particular not just the explicit constraints of the assembly area but spread the vehicles with certain characteristics within the production sequence.

This is realized considering an objective function (used in order to compare the various vehicles) that is composed of two parts:

- the total penalty related to the constraint violations;
- an additional term that is used in order to give priority to vehicles that present options that should be selected more frequently;

These two terms are then adequately weighted in order to give more or less importance to the first or to the second one using another parameter.

This more complex logic allows to better adjust the behaviour of the heuristic to the characteristics of the input sequence than Policy 1; therefore, it should be – in most cases – more efficient than this last one, but also substantially more difficult to implement in practice.

As expected with a higher number of lanes, the results are better; this happens both with and without a bypass lane. This is related to the characteristics of SBs and in particular to their resequencing capabilities (described in Section 2). The interesting thing we noticed is a big difference in terms of performance when implementing a bypass lane while using the first policy.

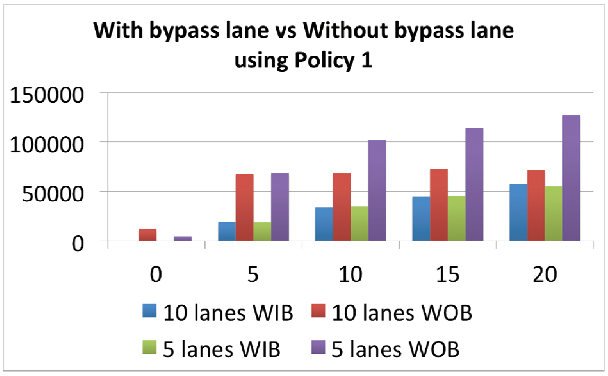

Figure 9: Comparison of the Results With and Without Bypass Lane for the 1st Test Case using the 1st Policy.

As can be seen in Figure 9, when using Policy 1, the SB with a bypass lane outperforms the SB without a bypass lane at any considered disruption level. However, this observation does not hold true when using a more efficient Policy 2, as can be seen in Figure 10. Based on previous results, we performed some other tests, in particular in order to compare whether using a bypass lane is more – or less – efficient than adding another traditional resequencing lane.

Using the first test case, we compared an 11 by 8 SB configuration without a bypass lane *vs.* a 10 by 8 SB configuration with a bypass lane. The results are reported in Figure 11.

The obtained performance measures indicate that adding a bypass lane results in performance improvement that is almost equal to the one that could be obtained with an additional resequencing lane.

We performed similar comparison using the second test case, as shown in Figure 12, under both Policies 1 and 2. This test case had a random input sequence, thus we didn't need to consider various disruption levels.

As seen in Figure 12, implementing a bypass lane doesn't seem to improve the performance of the buffer.

5. CONCLUSIONS

We developed a simulation network consisting of a Witness driver and an Excel file containing the dataset and the VBA decision making algorithms.

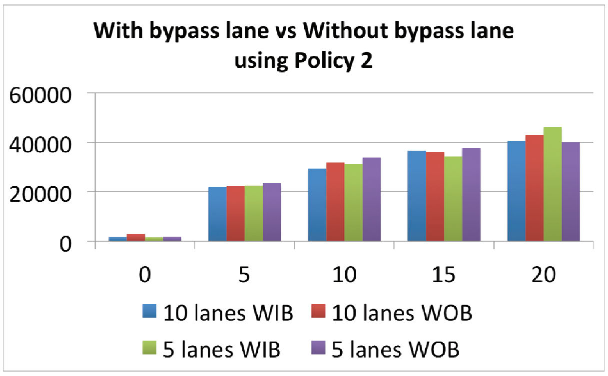

Figure 10: Comparison of the results with and without bypass lane for the first test case using the second policy.

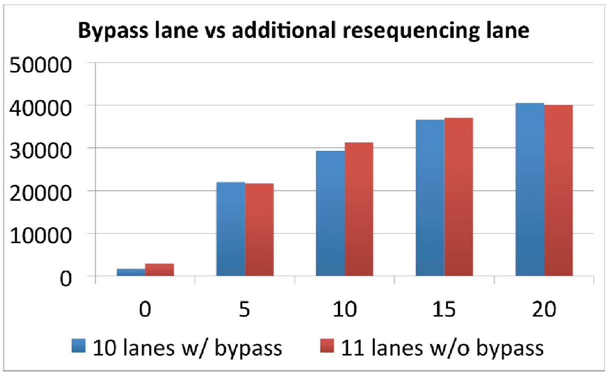

Figure 11: Comparison of the results with a bypass lane *vs.* the ones with an additional resequencing lane.

This framework allows us to simulate what happens within a selectivity bank, typically used in assembly lines both for storage and resequencing activities. It could be used for two main purposes:

- As a decision support tool that could be used during the design phase of a new plant;
- To perform what-if analyses both regarding the usage of different managing policies for the bank and for its layout.

We evaluated two input sequences, and we've tested them with different designs (in particular with or without using a bypass lane) and different sequencing policies. We investigated in particular whether using a bypass lane could improve SB's performance both in comparison with layouts with the same number of resequencing lanes and with others characterized by the same total number of lanes.

The obtained results seem to indicate that different policies could significantly affect system performances. In particular the more complex Policy 2 (described in Section 4) yields better results than its counterpart (Policy 1).

Further considerations regarding the obtained results are that:

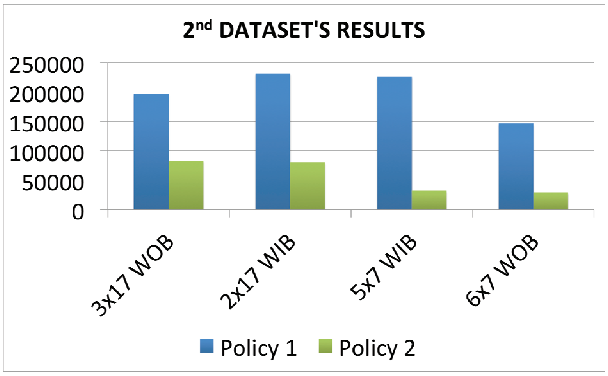

Figure 12: Comparison of the results with and without bypass lane for the second test case.

- a higher number of lanes tends to reduce the performance advantage of the more complex policies;
- the usage of a bypass lane tends to reduce the difference in terms of performance between policies 1 and 2 in certain conditions (in particular, using the more complex dataset).

Regarding the comparison between different SB designs, the results indicate that the usage of a bypass lane does not significantly affect system performance as compared with traditional designs with the same total number of lanes. Nevertheless, this could be considered an interesting result because it indicates that even if the current plant layout doesn't allow to increase the number of traditional resequencing lanes it is sufficient to implement a bypass lane (that could be for example extremely short) in order to obtain the same performance improvement.

Future work could allow to use this framework to evaluate buffer performance depending on the input mix complexity as well as to consider more than just one buffer of the plant. It could allow plant managers to use this tool to formulate recommendations regarding:

- possible improvements of the buffer's design;
- possible new policies used in order to manage resequencing activities;
- best combination of the previous elements considering a certain – fixed – budget dedicated to plant modifications.

REFERENCES

Banks, J., 1998. *Handbook of Simulation*. John Wiley & Sons, Inc.

Blatchford, A., 2008. *Quantifying Product Mix on Flow Lines with High Product Variety*, MSc Thesis, University of Cardiff.

Fradkin, Y., 2006. *Automotive Assembly Line Sequencing Application*. Ford Technical Report SRR-2006-0142, Ford Motor Company, Dearborn, USA.

Gusikhin, O., Caprihan, R., and Stecke, K., 2007. Least in-sequence probability heuristic for mixed-volume production lines. *International Journal of Production Research*, 46(3):647–673.

Han, Y., Zhou, C., Bras, B., McGinnis, L., Carmichael, C., and Newcomb, P., 2003. Paint Line Color Change Reduction in Automobile Assembly Through Simulation. *In Proceedings of the 2003 Winter Simulation Conference.*

Inman, R., 2003. ASRS sizing for recreating automotive assembly sequences. *International Journal of Production Research*, 41:847–863.

Inman, R., Bhaskaran, S., and Blumenfeld, D., 1997. In-plant material buffer sizes for pull system and level-material-shipping environments in the automotive industry. *International Journal of Production Research*, 35:1213–1228.

Karakal, S.C., 1998. A novel approach to simulation modeling. *Computers & Industrial Engineering*.

Nagane, S., 2002. *Simulation Study of Multilane Selectivity Bank in Automotive Industry*, MSc Thesis, University of Kentucky.

Narayanaswamy, R., Jayaraman, A., and Gunal, A.K., 1997. *A Sortation System Model*.

Park, Y.H., Matson, J.E., and Miller, D.M., 2008. Simulation and Analysis of the Mercedes-Benz All Activity Vehicle (AAV) Production Facility. *In Proceedings of the 1998 Winter Simulation Conference*.

Spieckermann, S., Gutenschwager, K., and Voß, S., 2004. A sequential ordering problem in automotive paint shops. *International Journal of Production Research*.

Ulgen, O., 1994. The Role of Simulation In Design And Operation Of Body And Paint Shops In Vehicle Assembly Plants. *In Proceedings of The European Simulation Symposium*.