ELSEVIER

European Journal of Operational Research 97 (1997) 348-362

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

Scheduling of flexible flow lines in an automobile assembly plant

A. Agnetis [a,*], A. Pacifici [b], F. Rossi [b], M. Lucertini [b], S. Nicoletti [b], F. Nicolò [c], G. Oriolo [a], D. Pacciarelli [a], E. Pesaro [d]

[a] *Università "La Sapienza", Dip. di Informatica e Sistemistica, Via Buonarroti 12, 00185 Roma, Italy*
[b] *Università "Tor Vergata", Centro Vito Volterra, Via della Ricerca Scientifica, 00133 Roma, Italy*
[c] *Terza Università di Roma, Dip. di Meccanica e Automatica, Via Segre, 2, 00154 Roma, Italy*
[d] *FIAT S.p.A., Divisione Sistemi Informativi, 80038 Torino, Italy*

Abstract

This paper deals with the material flow management in a large-scale manufacturing process, namely the assembly of automobiles in a highly automated plant in Italy. After a detailed description of the plant from the viewpoint of material flow issues, the modeling process and the methodologies employed to address the problems are illustrated. The decision models were validated by means of simulations of the real plant in several different production scenarios (varying demand volume and mix, resource availability etc.).

Keywords: Automobile industry; Dispatching rules; Flow control; Manufacturing processes; Simulation

1. Introduction

This paper describes the results of a study that was undertaken to analyze the production process and, possibly, improve its performance, in a plant for automobile assembly. This study has been carried out by researchers from both academia and from ELASIS, a FIAT-group company for research and development. All the information and the data contained in this paper refer to the 1992 production environment, when two types of cars were produced, each in several different versions. The overall objective of the research was to recognize the areas of intervention in order to enhance productivity of the plant, which was actually 1 200-1 300 cars per day, whereas the plant was originally designed for a production rate of 1 800 cars per day.

From an analysis in the field and a preliminary simulation of the whole system, some factors emerged that negatively affected the system productivity. Among these factors, were an exceedingly high number of Automated Guided Vehicles (in the following, AGVs) in some AGV circuits, a frequent breakdown of certain material handling devices, and a recurrent lack of raw material at some areas (caused by a poor communication link between different plants). However, the first results of the research program pointed out that the major role was played by the way the material flow moved through the different areas of the plant. Significant improvements could be attained by adopting a different philosophy in the management of the operations at the shop floor level. This is the aspect we focus on in this paper.

[*] Corresponding author.

PII S0377-2217(96)00203-2

We considered the modeling of the flow management problem arising in the plant, and we propose some dynamic dispatching rules. The option for the dispatching rules is dictated by the size of the system on hand, which makes the decision problems too complex to be handled by an off-line approach. It must be underscored that the implementation of some flow management policies requires a structural change in the information system (namely, the way in which components are addressed by the supervisor) and in the layout. The cost of these changes will be considered versus the advantages of the new approach. However, some decision rules do not imply major changes in the information system.

The main characteristics of the plant and of the assembly process are described in Section 2. The relevant decisions for flow management are introduced in Section 3. Section 4 deals with the proposed dispatching rules in detail. The simulation experiments are described in Section 5. A financial evaluation of the profitability of the new approach is presented in Section 6. Finally, some conclusions are drawn.

2. The plant and the process

In this section, the physical and information structure of the plant considered is described, highlighting the most relevant aspects of resource management.

The plant consists of four sub-plants, respectively devoted to steel molding, welding, painting and assembly. This study concerns the assembly phase, that is actually the bottleneck of the whole manufacturing process. The assembly sub-plant has a nominal production capacity of 1 800 cars/day, while the capacity of other three sub-plants exceeds 2 000 cars/day. The actual daily production is only 1 300 cars/day. Hence, the assembly sub-plant is the bottleneck of the whole factory.

Fig. 1 depicts the layout of the assembly sub-plant. The car bodies move along four parallel lines, which altogether form the *bodyshell path.* The four lines may have different speeds. The components that must be assembled on the bodyshell are manufactured in *side areas*, located beside the bodyshell path. At special *marriage points* on the bodyshell path, a component prepared in the side area is assembled with the bodyshell. In some particular points along the bodyshell path, worker teams carry out certain manual operations, to complete the automated assembly operations performed by robots at the marriage points.

Bodyshell path and side areas are interfaced by means of special buffering facilities called *automotive buffers* (Fig. 1). These allow for random access to the prepared components currently in the buffer, i.e., finite components may leave the buffer (and hence the side area) in any sequence. Unlike more traditional assembly lines, the orders to the side areas are released on the basis of *real* orders, and not forecasted demand. Production control is *integrated* and *real-time*, in the sense of the definitions given by Bedworth and Bailey (1987).

Most of the operations carried out in both the side areas and the bodyshell path are performed by human operators. The length of each operation has been measured by researchers of ELASIS, in different time intervals and with different workers. These measurements showed that these values can be considered sensibly deterministic. Moreover, once the number of active servers at each stage has been fixed, it does not vary randomly, since an operator can be replaced at any time by another one.

The part flow control as well as the management of the transportation system is fully automated. When a bodyshell passes through a control gate (sensor) located on the bodyshell path, the information system releases an order to all the side areas. The order contains a number identifying the individual car being produced, and all the information concerning the specific component to be prepared in each side area. Only components having the same identification number as the bodyshell can be assembled at the marriage point. The information system does not allow modification of this correspondence between a bodyshell and its components.

When a bodyshell enters the marriage section, the corresponding unique component is "called" from the automotive buffer of that area. If the component is present in the buffer, the assembly operation can be performed with no delay. Otherwise, the appointment is missed and this causes the corresponding bodyshell to wait. As long as such a tardiness value is low, the only effect is to temporarily stop the bodyshell path, with no major consequences. If it is high, the bodyshell must be moved out of the main line, since otherwise this would block the whole line upstream. This maneuver is extremely costly since the bodyshell that has temporarily been moved out will have to be reintroduced in the line later on, with additional burden of labour and increase in flow management complexity. Actually, appointments between bodyshells and subassemblies are occasionally missed. Thus, the first objective to be pursued is the meeting of these appointments, i.e., we would like that every time the bodyshell calls a component at a marriage point, this component is already available in the automotive buffer of the corresponding side area.

Flowing through a side area, each component undergoes a series of operations. Each operation stage consists of several identical servers in parallel. The component may be worked on by any of the

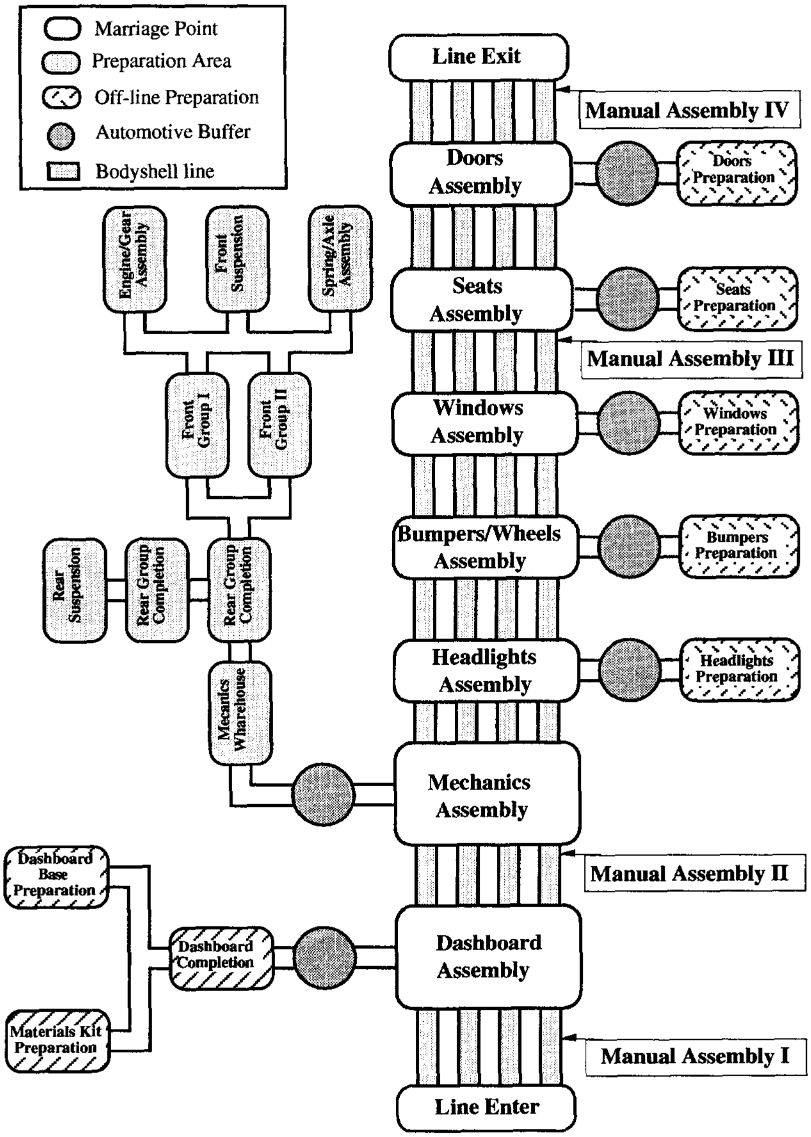

Fig. 1. Physical layout of the assembly sub-plant.

servers of each stage. Since many and complex operation types are required at each stage, the operations are carried out manually by human operators. The handling of the component is performed by AGVs. Each AGV carries only one component. Upon receipt of an order, a raw component is loaded on an AGV. The component remains on the AGV during all the operations. It is unloaded only after the last operation is completed, and routed to the automotive buffer. After a finite component is unloaded, the empty AGV goes back to the starting point, to receive a new raw component.

In this paper, we are concerned only with the operations in two of the side areas, shown in Fig. 2 and Fig. 4, namely the *dashboard area* and the *mechanic area*. This choice is due to the following two reasons: (i) in these two areas, the throughput time variance of components of the same type is high, whereas in the other side areas the throughput time is sensibly deterministic; (ii) the physical layout of these two areas is such that scheduling problems arise, whose solution significantly affects the performance of the system.

The other side areas have a simpler structure, in which no meaningful decision problems exist. In fact, in the other areas the preparation of the components takes a much smaller time than the time the bodyshell takes to reach the marriage point.

2.1. Dashboard area

The layout of the dashboard area is shown in Fig. 2.

The area is divided into two *AGV circuits*, called respectively *dashboard completion* and *dashboard testing*. The former consists of two load stations, two stages and four transfer cranes interfacing the circuit with dashboard testing. The latter includes four transfer cranes interfacing the circuit with dashboard completion, four stages and two unload stations interfaced with the automotive buffer of the area. When a new order is issued, a dashboard base and a preparation kit are extracted from the upstream buffer and a raw component is mounted on the first available AGV at the loading station. Upon completion of the operations in the first circuit, each dashboard is transferred from the first circuit to the second by means of a crane. When a component completes the operations in the second circuit, it is unloaded from the AGV and routed to the automotive buffer. In both circuits, each stage performs a specific operation. A fixed number of identical servers operate in each stage.

According to the actual routing policy, when an AGV moves from one stage to the next, it heads to the server of the next stage that has performed the least number of operations so far. This rule selects

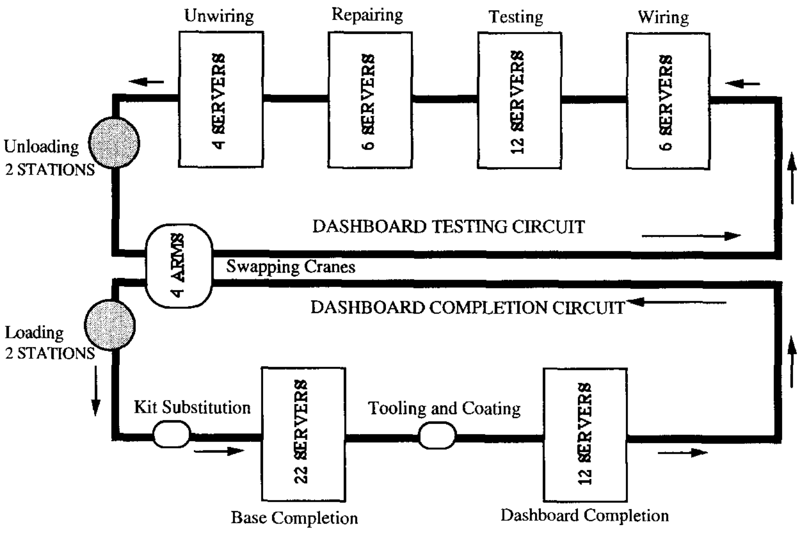

Fig. 2. Dashboard area layout.

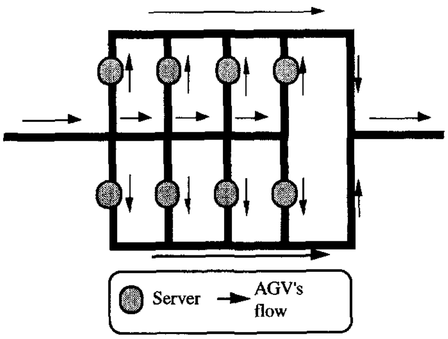

Fig. 3. Single stage layout.

the next server only on the basis of the workload situation at each server and not on the basis of the total operation time left for the dashboard being produced. In Section 4 we propose a rule that attempts to overcome this drawback.

Because of the grid layout of each stage (Fig. 3), the total distance traveled by each AGV throughout the system is the same, regardless of which servers the AGV has used. As a consequence, the total travel time is identical and deterministic for all the AGVs.

In the completion and testing circuits, respectively, 47 and 34 AGVs are at the moment available. In the actual management of the plant, all of the available AGVs and servers are employed, regardless of the production mix and volume.

In Table 1 we report the operation times required at each stage for different dashboard types as well as their production mix, in the standard 1992 production environment.

2.2. Mechanic area

The layout of the mechanic area is shown in Fig. 4. We distinguish seven sub-areas, namely Rear Group Completion (RGC), Front Group Completion (FGC), Mechanic Group Completion (MGC), Rear Suspension (RS), Engine/Gear Assembly (E/GA), Spring/Axle Assembly (S/AA), and Front Suspension Completion (FSC). Three of them (E/GA, RGC and MGC, the shaded areas in Fig. 4) are AGV circuits, the others are simple transfer lines. In the transfer lines, the output and input sequences of the parts are identical.

Table 1
Operation times and production mix for the Dashboard Area

Dashboard type num.	Production mix %	Operation times (minutes)						
		stage 1	stage 2	transfer	stage 3	stage 4	stage 5	stage 6
1	6.35	9.125	5.060	0.900	6.844	6.000	3.650	2.324
2	13.03	9.765	5.060	0.900	6.884	6.000	3.650	2.454
3	2.52	10.355	4.929	0.900	7.770	6.000	3.650	2.770
4	4.42	10.355	5.129	0.900	7.770	6.000	3.650	2.770
5	17.46	11.293	5.290	0.900	7.770	6.000	3.650	2.930
6	10.63	9.427	9.427	0.900	7.770	6.000	3.650	2.724
7	4.48	11.293	11.293	0.900	7.770	6.000	3.650	2.930
8	0.85	9.427	9.427	0.900	8.566	6.000	3.650	3.200
9	9.66	9.414	9.414	0.900	8.566	6.000	3.650	3.119
10	1.53	11.516	5.520	0.900	6.884	6.000	3.650	1.539
11	11.67	11.557	5.572	0.900	6.884	6.000	3.650	1.539
12	2.58	11.947	5.441	0.900	6.884	6.000	3.650	1.539
13	2.01	12.898	5.441	0.900	7.770	6.000	3.650	1.642
14	5.39	12.898	5.441	0.900	7.770	6.000	3.650	1.642
15	0.24	11.531	5.441	0.900	7.770	6.000	3.650	1.642
16	1.61	12.775	5.441	0.900	7.770	6.000	3.650	1.642
17	2.16	12.030	5.441	0.900	7.770	6.000	3.650	1.642
18	3.41	12.030	5.441	0.900	8.566	6.000	3.650	1.642

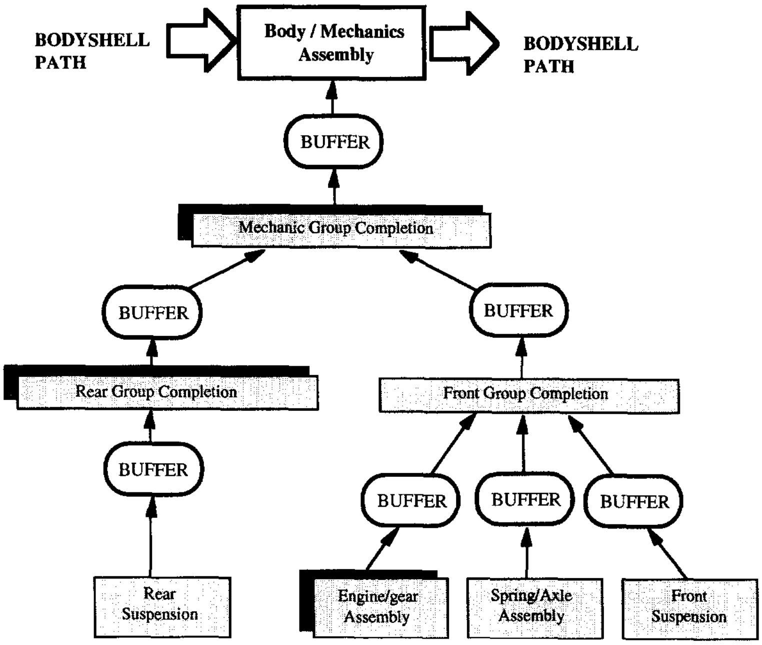

Fig. 4. The Mechanic Area.

The physical structure of each AGV circuit is very similar to the structure of the circuits in the dashboard area. The dispatching rule actually followed by the AGVs is again the Round Robin rule. When an order is issued from the bodyshell path, in the leaves of the tree (i.e., RS, E/GA, S/AA and FSC) a new sub-component starts being produced. The sub-components leaving each sub-area are pushed into automotive output buffers (Fig. 4). When all of the three subcomponents of the same engine are available, such a triplet enters the downstream sub-area (FGC) and is machined. Actually, the three sub-areas do not have the same throughput time. Precisely, it has been observed that when the assembled block engine/gear enters the output buffer of the sub-area E/GA, the other two sub-components are already present in the other two output buffers. In other words, the productivity of E/GA is lower than the other two (S/AA and FSC). Similarly, when the completed front group enters the output buffer of FCG, the corresponding rear group is already available. Therefore, in the following, we are interested in focusing only on the AGV circuits E/GA and MGC.

In particular, the most time-consuming phase is E/GA. In Table 2 we report the operation times required at each stage for different engine/gear types

Table 2
Operation times and production mix for the Engine/Gear Assembly Area

Engine type no.	Production mix %	Operation times				
		stage 1	stage 2	stage 3	stage 4	stage 5
1	84.61	0.574	0.9	2.658	3.545	2.525
2	1.11	0.574	0.9	2.543	3.311	8.669
3	1.87	0.574	0.9	2.658	3.311	8.514
4	1.56	0.574	0.9	2.658	3.311	8.663
5	1.96	0.574	0.9	2.658	3.311	12.052
6	3.83	0.574	0.9	2.658	3.311	13.260
7	5.06	0.574	0.9	2.443	3.311	4.238

as well as their production mix, in the standard 1992 production environment. Notice that the demand of engine type 1 covers the large majority of the overall demand.

3. The decision process

In this section we specify the kind of decisions analyzed in this paper. The decision process concerns the management of the material flow through the plant.

In principle, one can distinguish two kinds of decision: (i) deciding the sequencing in which the bodyshells enter the plant (and therefore, the sequencing in which orders to the side areas are released); (ii) deciding the routing and dispatching of the components inside each side area.

Notice that, due to the pipeline structure of the assembly lines (Fig. 1), bodyshells cannot pass each other, i.e., the sequence in which the bodyshells arrive to the marriage points is identical to the sequence in which orders were released. We must underscore that such a sequence is given and cannot be altered. In fact, the sequence is actually decided by an expert system aiming at balancing the workload of the working teams along the bodyshell (in charge of completing the automated assembly operations at the marriage points), on the ground of an everchanging demand, communicated by the commercial division of the factory. Thus, point (i) above is not really a matter of decision, and therefore our field of action is restricted to the material flow management inside dashboard and mechanic areas (point (ii) above).

Let us next describe the decision process inside each side area. Each AGV circuit can be regarded as a flow shop with multiple processors (FSMP), consisting of b stages M_1, $M_2, \ldots$, M_b, each with a number of active servers. Let m_k denote the number of servers at stage M_k ($k = 1, \ldots, b$). A set of components to be produced is given, characterized by the following quantities:

r_i: release time of component i, i.e., the time at which the i-th order is issued;

d_i: due time of component i, i.e., the time when the bodyshell is ready for assembly with the finite component at the corresponding assembly point;

p_{ik}: time required for processing component i on a server of stage M_k.

Notice that the *allowance* $a_i = (d_i - r_i)$ is the amount of time available for a component to be produced. If the flow time of a component through the side area is less than or equal to a_i the assembly of the component with the bodyshell can occur with no delay. During normal operation, the speeds of the four lines of the bodyshell path are identical. As a consequence, the values a_i are the same for all the components. In what follows, C_i is the completion time, $F_i = C_i - r_i$ the flow time, and $T_i = \max\{0, C_i - d_i\}$ the tardiness of the i-th component.

The problem consists in routing the components on each stage of the flow shop in order to meet a number of production objectives. A major objective is to *minimize the number of tardy components.* In fact, the number of missed appointments has a direct impact on productivity (as explained in Section 1). In principle, one can always trivially achieve such an objective by increasing the distance between the control gate and the marriage point (i.e., issuing the order earlier) and therefore a_i. However, this has a negative consequence on the work-in-process (WIP) in the side areas. In fact, at any time, the number of components being processed in a side area is equal to the number of bodyshells between the control gate and the marriage point. For a given line speed, this number is obviously proportional to the abovementioned distance. A large WIP is undesirable for the classical reasons related to tied-up capital costs. In our case, a further negative effect of a large WIP might be automotive buffer overflow. Hence, we wish to have no tardy components, while *reducing the WIP as much as possible.* WIP is directly related to the *maximum flow time* of a component. In fact, the maximum flow time determines the minimum value of a_i.

The above model should consider other aspects which are typically neglected in classical scheduling literature: (i) the transfer of a component from one stage to the next requires a given transportation time; (ii) the WIP in a side area cannot exceed the number of available AGVs in the circuit; (iii) the intermediate storage capacity between two consecutive stages is very limited. (As shown in Fig. 3, each server can host at most one waiting component in front of it.) Even disregarding these features, for the number of stages and components of the problem on hand, finding an exact optimal solution to the FMSP with respect to any of the aforementioned objective functions is computationally infeasible. Moreover, we should remark that the number of components to be produced is known in advance only for a limited time horizon, i.e., orders arrive dynamically to the side area.

For all these reasons, we propose a solution approach based on the testing and comparison of dispatching rules through simulation. Literature on routing and dispatching rules in manufacturing systems is huge and well-known. Among the others, we cite the comprehensive reviews by Panwalkar and Iskander (1977), and by Gupta et al. (1989). When the problem is deciding the input sequencing on a production line, a main issue can be balancing the workload among the processors over time. This has led to the goal-chasing sequencing rule by Miltenburg (1989) for a single machine, generalized to a multi-level system by Miltenburg and Sinnamon (1989). A specific study concerning sequencing issues in automobile assembly lines has been carried out by Burns and Daganzo (1987). However, their focus is on minimizing set-up costs and capacity costs, rather than optimizing throughput-related performance measures.

Typically, dispatching rules are simple priority indices associated with the jobs being scheduled. These indices may or may not depend on the actual state of the system (*static* vs. *dynamic* scheduling rules). Here we propose and compare several rules, tailored for this problem, and analyze them from the viewpoint of the information system already existing in the plant. To this aim, we present the rules in the perspective of the well-known concepts of *push* and *pull* production control.

4. Dispatching rules

In this section, we consider the dispatching rules employed in our study. Two basic types of rules have been considered: *Push* and *Pull* rules. (As it will be clear later on, here we use the terms push and pull in a slightly different sense than in classical production control.)

In a *push* dispatching rule, the assignment of operations to machines is driven by the flow of the components, i.e., when a component is completed on a server of stage M_k, it is assigned to a server of stage M_{k+1}. In other words, the component itself "pushes" production ahead (similarly to what occurs in push production control systems). Every time the processing of a component terminates at stage M_k, we decide which server, among those of stage M_{k+1}, the corresponding AGV will be sent to. The component will be loaded on the chosen server when all the components preceding it in the queue have been processed. The dispatching rule actually implemented in the plant is a simple push rule which can be described as follows:

Actual Rule (Round Robin). Choose the server which has performed the least number of operations so far.

Push rules simply account for the workload of the processors at the next stage, ignoring information concerning the components.

In a *pull* dispatching rule the assignment of operations to servers is driven by their activity: when a server completes an operation, it calls a component from the previous stage. In other words, the downstream stage "pulls" production throughout the system. Whenever a server of a stage is free, the "best" (according to some criterion) candidate among the components released by the previous stage is chosen. The pull rule we consider in this paper is the following:

Pull rule (Minimum Slack Time). Choose the part having minimum *slack time*, defined as the maximum time the part is allowed to wait in process without violating its due date.

Pull dispatching rules take into account the characteristics of the components but ignore the state of the other servers. Notice that a pull rule requires the presence of output buffers at the servers of each stage. These buffers are not present in the system right now, hence the implementation of a pull rule implies a modification of the layout. The only feasible modification to layout of the side area is to create room for one AGV waiting at the output of each server, i.e., to create an output buffer of capacity 1 for each server. In the experiments concerning the Pull rule (Section 5), we always assumed the presence of these buffers. Moreover, the supervisor must be able to acquire information concerning the parts in the output buffers of the previous stage. This requires hardware and software modifications in the information system.

In literature, dispatching rules are typically employed to select a part to be processed among a set of candidates. Therefore, we can say that these rules are employed in a "pull" fashion. We next describe in detail a new push rule which accounts not only for the state of servers at the next stage but also for the state of the processing of components at the current stage. In the following, we call such a rule *Matching* (this rule was already introduced by Agnetis et al., 1993). To this aim, let us define the following quantities:

r_i^k: *local release time* of component i at stage M_k, i.e., the time at which component i is completed at stage M_k.

TV: total AGV travel time between stages (this quantity does not depend on which servers are used at each stage, as explained in Section 2.1);

$ST_i^k(t)$: *slack time* of component i after the completion of operation at stage M_k, evaluated at time t. $ST_i^k(t)$ is the time the component can waste without violating its due date d_i. Its value is given by

$$ST_i^k(t) = d_i - TV - \sum_{h=k+1}^{b} p_{ih} - t$$

RT_j^k *ready time* of the jth server of stage M_k, i.e., the time at which the server is available for processing a new part (consequently, if at time t the server is idle, then $RT_j^k = t$).

The value of the slack time of a component *after* its processing at stage M_{k+1} depends on the server of stage M_{k+1} to which it is assigned. In particular, if we assign the component to a server j such that $RT_j^{k+1} \le r_i^k$, the slack time of component i is unchanged, whereas, if the component has to wait some time before starting processing at stage M_{k+1}, it decreases of the quantity $(RT_j^{k+1} - r_i^k)$. More precisely, if at time t we assign component i to the jth server of M_{k+1}, its slack time – evaluated at the completion of the operation at stage M_{k+1} – will be:

$$ST_i^k(t) - \max\left\{0, RT_j^{k+1} - r_i^k\right\}$$

Recalling that m_{k+1} is the number of servers at the stage M_{k+1}, suppose that at any time t there are $n^k(t)$ jobs under processing at stage M_k and $m_{k+1} > n^k(t)$. Rank the servers of M_{k+1} in nondecreasing order of ready times RT_j^{k+1}, where $j = 1, ..., m_{k+1}$, and the $n^k(t)$ components of M_k in nondecreasing order of r_i^k. For our objectives (meeting the due dates and minimizing $F_{\max}$), the slack time of no component must become too small, as the component flows throughout the system. The objective is therefore to maximize the minimum slack time of a component after processing on stage M_{k+1}. If $m_{k+1} > n^k(t)$, the $n^k(t)$ components at stage M_k are assigned to the first $n^k(t)$ servers of M_{k+1}. On the contrary, if $m_{k+1} < n^k(t)$, the first m_{k+1} jobs of M_k are assigned to the m_{k+1} servers of M_{k+1}. In the following, let n be the min $\{n^k(t), m_{k+1}\}$, and formulate an instance of the Bottleneck Assignment Problem (Gilmore et al. (1985)). In this problem, the decision variable x_{ij} is 1 if component i is assigned to the jth server of M_{k+1} and 0 otherwise. The Bottleneck Assignment Problem can be solved in polynomial time for any value of the coefficients on the objective function. In our case, the structure of these coefficients is such that the problem can be solved in time $O(n^2)$, since the coefficient matrix $ST_i^k(t) - \max\{0, RT_j^{k+1} - r_i^k\}$ is *graded* (Gilmore et al. (1985)). We recall that a matrix $C = \{c_{ij}\}$ is graded (across its rows) if $c_{ij} \ge c_{i,j+1}$ for all i, j. This is exactly our case, since, due to our ranking, $RT_j^{k+1} \le RT_{j+1}^{k+1}$, for all $j = 1, \cdots, m_{k+1} - 1$.

$$\max_{x_{ij}} \left\{ \min_i \sum_{j=1}^{n} x_{ij}\left[ST_i^k(t) - \max\left\{0, RT_j^{k+1} - r_i^k\right\}\right]\right\} \qquad (1)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad j = 1, ..., n$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, ..., n$$

$$x_{ij} = \{0, 1\} \quad i = 1, ..., n,\ j = 1, ..., n$$

The solution to problem (1) is employed to make real-time dispatching decisions in the following way. Whenever a component i ends its processing on

Table 3
Dispatching decision rules

	Push rules		Pull rules
	Round Robin	Matching	Minimum Slack Time
Input	The last loaded server h of stage M_{k+1}	Ready time for each server of the stage M_{k+1}. Slack time and release time of each component being processed at stage M_k	Slack time of each component being processed at stage M_k
Decision	Select server h of stage M_{k+1}	Matching between the components of stage M_k and the servers of stage M_{k+1}. Select the server found by solving the matching instance	Select the component at stage M_k with minimum slack time
Decision time	When a component ends its processing at stage M_k		When a server k of the stage M_{k+1} is ready to process another component
Output	Number of the selected server at stage M_{k+1}		Number of the selected component at stage M_{k+1}
Actions	Queue the finished component at stage M_k to the selected server at stage M_{k+1}		The finished component of stage M_k waits in its output buffer until it is selected by a server h of the next stage. When selected, the component goes to the server h

stage M_k, on the basis of the actual ready times, local release times and slack times, an instance of (1) is defined and solved. Then, i is routed to the server j of M_{k+1}, such that $x_{ij}^* = 1$. The other components (which are yet to be completed by M_k), in general will *not* be routed to the server specified by the solution $\{x^*\}$ of problem (1). Rather, the destination of the next component leaving stage M_k will be computed by solving an *updated* instance of (1), when it is its turn to leave stage M_k. In other words, the dispatching rule is implemented in a rolling horizon fashion.

We must remark that the Matching rule does not require layout and/or major information system changes (as better explained in Section 6).

Table 3 summarizes the different dispatching decision rules considered in this paper.

5. Simulation experiments and results

A simulation model of the whole assembly subplant has been built using WITNESS. Individual simulation models have been devised for the side areas and interfaced with the model of the bodyshell path.

Several simulation experiments have been carried out, for different values of those parameters that can be decided by the plant management every working day. Some of these parameters concern the whole plant, namely:

- *Nominal production rate and mix*: the number of cars of each type to be produced by the whole plant during a working day (consisting of 1 125 minutes).
- *Line rate*: the number of cars entering each line of the bodyshell path during a working day. Obviously, the sum of the four line rates gives the production rate.

Other parameters are specific of the Dashboard area:

- *AGV number*: the number of Automated Guided Vehicles present in each circuit.
- *Active servers*: the number of servers activated at each stage of each AGV circuit. Maximum values for these quantities are indicated in Fig. 2.

The following quantities are the output parameters of the experiments:

- *Mean cycle time (MCT)*: the mean time (in minutes) between the exit of two successive finite components from the AGV circuit. The inverse of this value is the *throughput* of the AGV circuit (number of parts processed per minute).
- *Flow time:* the length of the time interval (in minutes) from the release time of a component to its availability in the automotive buffer of the Dashboard area. In particular, as noticed in Section 3, minimizing maximum flow time allows a reduction of WIP in the AGV area and also along the bodyshell path.
- *Number of tardy jobs:* the number of components arriving at the marriage point of the Dashboard area *after* the respective due time.
- *Efficiency:* the ratio between the number of cars actually produced and the nominal production rate.

The *processing times* are those given in Table 2. As *production mix*, the standard production mix for the 1992 production environment has been used (Table 1).

Simulations were performed comparing the three different dispatching rules (*Round Robin*, *Pull* and *Matching*) in different processing conditions. In any case, no exceptional event (such as breakdowns) has been considered. We investigated the following scenarios:

i) *Balanced lines.* The four lines of the bodyshell path have equal rates. The processing times are deterministic and are given in Table 1. The experiments concerned five different production rates, namely 1 100, 1 300, 1 450, 1 600 and 1 800 cars/day. Moreover, different numbers of AGVs and active servers have been used.

ii) *Unbalanced lines.* The four lines of the bodyshell path have different rates. The production rate was fixed to 1 450 cars/day.

First of all, a large set of simulation experiments on the mechanic area has shown that the behaviour of such an area with respect to the relevant output parameters is remarkably insensitive to different values of input parameters. This can be easily grasped by glancing through Table 4. These results concern simulations at 1 800 cars per day (hence many more than the actual production), with balanced lines, and the actual AGV sizing of the mechanic area.

In other words, only the Dashboard area affects the whole plant productivity in a meaningful way.

5.1. Balanced lines

In these experiments, the four lines of the bodyshell path move at the same speed. For what concerns the Dashboard area, the maximum productivity allowed by the bottleneck stage (wiring - stage #3) is about 1 450 cars per day. Thus, simulations were executed setting the production rates at the values of 1 100, 1 300, 1 450, and using the processing times and production mix of Tab.1. Moreover, in order to observe the system behaviour at a higher throughput (1 600 and 1 800 cars per day), we performed an additional set of experiments reducing the processing times at stage #3. We tested two different situations concerning system sizing: *nominal sizing* and *oversizing*. In the former case, the number of servers open at each stage is shown in Table 5. These values were obtained multiplying the average

Table 4
Mechanic area simulation results at 1 800 cars/day

Rule	Mean cycle time	Mean flow time	Max flow time	Allowance	Tardy jobs
Push	0.62	15.77	28.73	35	0
Pull	0.62	16.85	20.54	35	0
Matching	0.62	17.22	30.19	35	0

Table 5
Nominal system sizing in the dashboard area

	Production rates			
	1 100	1 300	1 450	1 800
Stage	number of servers on each stage			
1	12	14	15	19
2	6	7	7	9
3	4	4	4	4
4	4	5	5	5
5	6	7	8	10
6	6	6	6	6
7	3	3	3	4
Sub-area	number of AGVs in each sub-area			
1	28	30	32	42
2	25	28	28	33

processing time of each stage by the production rate. This can be seen as a rough application of Little's law, if we view the mean throughput time as the average processing time. In the oversized case, these numbers have been increased by 20%. This environment has been analyzed because management suggested increasing the number of AGVs in order to better meet the appointments.

Tables 6 and 7 show the results for the Dashboard area. In the first column the nominal production rates (*cars / day*) are indicated. In the second column the different dispatching rules are reported. The other columns report the values of the output parameters. As one can see from the last column, as long as the nominal production rate does not exceed 1 300 cars/day, the number of cars produced is always over 95% the nominal value, whichever the dispatching rule and the number of AGVs. For these values of production rate, there are no tardy jobs. In the nominal sizing environment, the comparison among the different rules is straightforward: a slack-based criterion (Pull, Matching) is better than the actual rule. In particular, for low production rates (1 100 cars/day) the best rule is Matching, whereas Pull is more efficient at medium production rates (1 300-1 450cars/day). For higher production rates, Pull still results in no tardy jobs, whereas Push and Matching do. Hence Matching is not robust with respect to traffic problems. With respect to mean flow time, a slack-based rule shows slightly worse results than the actual rule. This is reasonable, since slack-based rules tend to "squash" the value of

Table 6
Dashboard area simulation results: balanced lines, nominal sizing

Production rates	Rule	Mean cycle time	Mean flow time	Max flow time	Tardy jobs	Mean efficiency
1 100	Push	1.02	44.36	61.87	0	1.000
	Pull	1.03	43.32	54.01	0	0.992
	Matching	1.03	45.00	53.37	0	0.992
1 300	Push	0.84	43.53	64.08	0	1.000
	Pull	0.87	42.41	53.06	0	0.994
	Matching	0.87	43.93	55.21	0	0.994
1 450	Push	0.77	42.82	59.25	0	1.000
	Pull	0.78	44.88	53.42	0	0.994
	Matching	0.79	48.32	58.79	0	0.982
1 600	Push	0.7	42.04	59.32	6	1.000
	Pull	0.72	44.06	52.57	0	0.976
	Matching	0.72	45.52	58.81	0	0.976
1 800 (mod. pr. times)	Push	0.62	41.45	59.48	11	1.000
	Pull	0.63	43.35	51.09	0	0.991
	Matching	0.63	46.69	58.89	14	0.991

Table 7
Dashboard area simulation results: balanced lines, oversizing

Production rates	Rule	Mean cycle time	Mean flow time	Max flow time	Tardy jobs	Mean efficiency
1100	Push	1.02	45.11	62.65	0	1.000
	Pull	1.03	44.12	55.08	0	0.992
	Matching	1.03	46.30	54.39	0	0.992
1300	Push	0.87	46.54	66.64	0	0.994
	Pull	0.90	43.12	57.09	0	0.961
	Matching	0.91	48.76	68.84	0	0.950
1450	Push	0.79	50.22	63.74	0	0.982
	Pull	0.79	51.66	61.70	0	0.982
	Matching	0.84	63.71	84.31	15	0.923
1600	Push	0.75	52.06	70.81	2	0.937
	Pull	0.75	53.16	62.1	1	0.937
	Matching	0.79	70.8	87.82	19	0.890
1800 (mod. pr. times)	Push	0.65	53.5	81.23	11	0.961
	Pull	0.66	52.9	69.51	2	0.946
	Matching	0.72	75.32	90.73	34	0.868

mean flow time towards its maximum. This is also confirmed by a reduced variance of the flow time.

Similar considerations hold for the oversized environment, with the difference that now the overall performance of the system is worse. However, even in this case Pull is more robust than the other two rules.

Notice that, as we discussed in Section 3, the work-in-process (WIP) is related to the maximum flow time of a component. In particular, the maximum *WIP reduction* (ρ) achievable could be easily computed by means of the following equation:

$$\rho = \frac{(a_i - F_{\max})}{MCT}.$$

Table 8
Allowances for different production rates

Production rates	Nominal cycle time	a_i
1 800	0.62	52
1 600	0.70	59
1 450	0.77	64
1 300	0.84	69
1 100	1.02	76

The different values for a_i are given in Table 8.

Table 9 shows the maximum WIP reduction for different values of production rate and different rules.

From the results illustrated in Table 9 the following considerations arise:

i) For production rates between 1 100 and 1 300, WIP can be reduced by means of "slack based" rules, mantaining a tolerance on the value of the

Table 9
Reduction of WIP

Production rates	Rule	ρ Nominal sizing	ρ Overizing
1 100	Push	14.41	13.61
	Pull	22.42	21.33
	Matching	23.08	22.04
1 300	Push	4.13	2.48
	Pull	13.38	10.01
	Matching	11.58	-
1 450	Push	3.65	-
	Pull	8.05	1.71
	Matching	4.01	-
1 600	Push	-	-
	Pull	4.50	-
	Matching	-	-
1 800 (modified proc. times)	Push	-	-
	Pull	-	-
	Matching	-	-

Table 10
Line rates on the bodyshell path

Line	Rates	Nominal cycle time	a_i
1	400	0.7	59
2	362	0.77	64
3	362	0.77	64
4	326	0.84	69
tot	1450	0.77	-

allowance. Observe that also the actual Push rule allows a reduction, but with a very narrow tolerance.

ii) When the production rate increases or the system is oversized, only the Pull rule would allow a reduction in WIP, however the tolerance becomes negligible.

In the situations in which there are tardy jobs, obviously the allowance cannot be reduced.

5.2. Unbalanced lines

In the following, we describe a set of simulation experiments in which the production rates of the four lines of the bodyshell path have different values, as shown in Table 10. The sizing of the AGVs is nominal.

In this case, the allowance a_i is not the same for all dashboards but depends on the line which the dashboard order is issued from. The output results are reported in Table 11, where the maximum flow time corresponding to the four lines is indicated.

"Slack-based" rules appear to be more efficient than the actual rule, even though the respective maximum flow times on lines 2, 3 and 4, are greater than the value obtained by the actual Push rule. This behaviour is due to the fact that Push can not take into account the different urgencies of the parts. There is no use, for instance, to complete a dashboard in 59.78 minutes if the corresponding bodyshell gets to the marriage point only in 69 minutes.

6. Implementation issues and conclusions

In this section we give a brief evaluation of the costs entailed by the adoption of a new dispatching philosophy.

The implementation of the Matching rule only requires minor modifications of the information system. In fact, the implementation of the Matching rule concerns the possibility for a part to select the downstream server. The information system selects the server when a part is still on an upstream server. In other words, no major hardware modification of the information system is needed. Moreover, up to medium production rates, the Matching rule outperforms the actual Push rule from the viewpoint of WIP reduction, and performs similarly to the Pull rule at the actual production rate (1200-1300 cars per day). The decrease of WIP can be evaluated in terms of tied-up capital. Estimating that the value of one dashboard is around $600, one can easily infer the savings obtained by the WIP reduction illustrated in Table 9.

However, on a long-term basis, when a major plant reorganization is involved, the Pull rule is more attractive, because of the higher production rates the rule allows one to attain. In particular, the following issues must be taken into account.

i) *Information system.* In order to implement the Pull decision rule, the information system - equipped with suitable hardware flow sensors - must collect data, elaborate them and distribute the result of the decision rule to the end effectors of the flow management control units. (Push and Matching produc-

Table 11
Dashboard area simulation results: line rates of Table 10

Rule	MCT	Max flow time: line 1	Max flow time: line 2	Max flow time: line 3	Max flow time: line 4	Tardy jobs
Push	0.77	59.12	59.35	60.12	59.78	2
Pull	0.79	55.24	60.45	60.29	63.41	0
Matching	0.79	59.39	60.89	61.37	64.32	0

tion rules do not require the parts to be addressed individually.) Software and hardware implementation costs can be expressed as about $6.75 per car being produced.

ii) *Layout.* The plant layout must allow the material flows required by the decision rules. These costs cannot be determined exactly, but still have to be considered.

As we have seen in Section 5, the Pull rule allows an increase in the production capacity of the plant. The length of the period during which the plant reaches its nominal capacity is four years. (Consider that in one year there are 230 working days.) Moreover, during the first two years we can assume a linear growth of the production capacity, until the level 1 600 is reached. This value is then constant during the third and fourth year. Labor for developing the new system represents a minor cost and can be disregarded. The overall costs consist of a fixed investment cost to be paid at the beginning of the first year and a structural cost which is spanned over the four years. Reaggregating the contributions over the four years, we can assume that production uniformly increases by 300 cars/day, and hence get $ 300 × 230 × 4 × 6.75) = $ 1 863 000.

In order to get the same increase in production rate by enhancing the capacity of the plant (i.e., increasing the number of servers, modifying the process plan etc.), a different cost must be computed. Precisely, the cost of increasing the production capacity of one car/day is $100. Hence, the overall costs can be evaluated as $(300)(230)(4)(100) = $27,600,000.

The comparison among the last two figures indicates that the Pull rule is definitely the winner, and this is the one which is going to be implemented in the plant.

References

Agnetis A., Oriolo G., Pacciarelli D., Rossi F. (1993), A dynamic lookahead dispatching rule in a flexible flow line for automobile assembly, *Proceedings of the 12th Triennal World Congress of the International Federation of Automatic Control*, Sydney, Australia, 18-23 July 1993.

Bedworth D.D., and Bailey J.E. (1987), *Integrated Production Control Systems*, Wiley and Sons Inc., New York.

Burns L.D., and Daganzo C.F. (1987), Assembly line job sequencing principles, *International Journal of Production Research*, **25**(1), 71-99.

Gilmore P.C., Lawler E.L., Shmoys D.B. (1985), Well-solved special cases, in: Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B. (eds.), *The Traveling Salesman Problem*, Wiley and Sons, UK.

Gupta Y.P., Gupta M.C., Bector C.R. (1989), A review of scheduling rules in flexible manufacturing systems, *International Journal of Computer Integrated Manufacturing*, **2**(6), 356-377.

Miltenburg G.J. (1989), A theoretical basis for scheduling mixed-model assembly lines for just-in-time production systems, *Management Science*, **35**, 192-207.

Miltenburg J., and Sinnamon G. (1989), Scheduling mixed-model multi-level just-in-time production systems, *International Journal of Production Research*, **27**(9), 1487-1509.

Panwalkar S.S., Iskander W. (1977), A survey of scheduling rules, *Operations Research*, **25**(1), 45-62.