TECHNICAL RESEARCH REPORT

SYSTEMS RESEARCH CENTER

Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University

Some Open Problems in the Design and Use of Modern Production Systems

by G. Harhalakis and J.M. Proth

TR 92-11

APPLIED STOCHASTIC MODELS AND DATA ANALYSIS, VOL. 7, 33–45 (1991)

SOME OPEN PROBLEMS IN THE DESIGN AND USE OF MODERN PRODUCTION SYSTEMS

G. HARHALAKIS

Department of Mechanical Engineering and Systems Research Center, University of Maryland, U.S.A.

AND

J. M. PROTH

INRIA–CESCOM, Taechnopôle Metz 2000, 57070 Metz Cedex and *Systems Research Center, University of Maryland, U.S.A.*

SUMMARY

During the past two decades, manufacturing systems have moved towards automation, integration and modularity. These trends will certainly continue in the future due to the constraints of the market and to evolution of resources and worker requirements. As a consequence, the design and use of manufacturing systems are increasingly expensive. Numerous methods and tools have been developed to face up to this situation, but some complementary aids could be provided for designers and manufacturing engineers. The goal of this paper is to present important open problems whose solutions could certainly significantly improve the design and use of modern production systems.

KEY WORDS Manufacturing systems layout Evaluation of manufacturing systems Scheduling Petri nets Data analysis

1. INTRODUCTION

The life cycle of a manufacturing system can be divided into four main stages, namely the preliminary design, the concrete design, the production and the dismantling stages.

The preliminary design starts as soon as the products to be manufactured are clearly specified. It includes the choice of the physical resources (machine, transportation systems, inventory devices, tools, etc.), the dimensioning of the system, the specification of the layout, the design (or the choice) of the management system and the evaluation of the whole system.

The design stage consists mainly of establishing and testing the manufacturing system, training the people involved in the system and doing a test production under real conditions.

The production stage includes tasks like planning and scheduling, quality control, inventory control, maintenance, handling and transportation.

The last stage of the life cycle groups the tasks which are necessary to dismantle or transform the manufacturing system when the market no longer asks for the initial products.

In this paper we emphasize possible improvements of some of the tasks described above. We first discuss the choice of the physical resources. We then revisit the layout problem to point out the parts of this problem which either remain unsolved or whose solutions are unsatisfactory. We then consider the tools available for the evaluation of the dynamics of the manufacturing systems and emphasize the improvements which still have to be made.

8755–0024/91/010033–13$06.50

Received 16 November 1990

Finally, we discuss the planning and scheduling problems and suggest some possible improvements in these fields.

2. CHOICE OF THE PHYSICAL RESOURCES

Starting from the physical resources available on the market, we have to select those which are able to meet the expected demand at the lowest cost. Usually, the cost under consideration includes the design cost and the running cost (for inventory, set-up, maintenance, etc.). In this section, we restrict ourselves to the basic problem of minimizing the purchasing cost of the machines (see [13]).

2.1. Problem formulation

Set $\mathbf{P} = \{P_1, P_2, \ldots, P_n\}$ be the set of specified products and $\{a_1, a_2, \ldots, a_n\}$ the related production rates ($\Sigma_{i=1}^{n} a_i = 1$). These production rates represent the forecast production assuming that the production is quite regular; otherwise, the set of ratios represents only a subset of the forecast production, and the same problem will be solved several times.

$\mathbf{M} = \{M_1, M_2, \ldots, M_m\}$ is the set of machines available on the market and $\{c_1, c_2, \ldots, c_m\}$ the related purchasing costs.

For any product P_1 $(i = 1, 2, \ldots, n)$, we denote by $u_i^1, u_i^2, \ldots, u_i^{r(i)}$ the set of alternative manufacturing processes. Remember that a manufacturing process u_i^k is the sequence of machines P_i has to visit in order to be manufactured and the time one unit of product has to spend on each machine. Such a sequence of machines, when considered without the related manufacturing times, is usually called routing.

In the following, a manufacturing process is then a sequence of pairs:

$$u_i^k = \left\{ (M_{i,1}^k, \mathbf{T}_{i,1}^k), \ldots, (M_{i,j}^k), \mathbf{T}_{i,j}^k), \ldots, (MJ_{i,s(k)}^k, \mathbf{T}_{i,s(i,k)}^k) \right\}$$

where $M_{i,j}^k \in \mathbf{M}$ is the jth machine visited by a unit of product P_i when using the kth manufacturing process available for this type of product (i.e. manufacturing process u_i^k); $\mathbf{T}_{i,j}^k$ stands for the processing time of one unit of product type P_i on machine $M_{i,j}^k$ when using u_i^k; $i = 1, 2, \ldots, n$; $k = 1, 2, \ldots, r(i)$; $j = 1, 2, \ldots, s(i,k)$.

Note that, by assumption, $\mathbf{T}_{i,j}^k = 0$ if a product type P_i does not need to visit machine $M_{i,j}^k$ when using u_i^k.

Using the previous notations, the problem can be formulated as follows, assuming that the objective is to minimize the total purchasing cost.

$$\text{Minimize} \sum_{j=1}^{m} c_j x_j \tag{1}$$

such that

$$\sum_{k=1}^{r(i)} \varphi_i^k = 1 \quad \text{for } i = 1, 2, \ldots, n \tag{2}$$

$$\sum_{i=1}^{n} \sum_{k=1}^{r(i)} a_1 \varphi_i^k T_{i,j}^k \leqslant x_j \quad \text{for } j = 1, 2, \ldots, m \tag{3}$$

$$x_j \in \{0, 1\}; \quad \varphi_i^k \geqslant 0; \quad i = 1, 2, \ldots, n; \quad k = 1, 2, \ldots, r(i); \quad j = 1, 2, \ldots, m \tag{4}$$

where $x_j = 1$ (respectively 0) if machine M_j is chosen (respectively, not chosen); φ_i^k is the proportion of flow of products of type P_i using u_i^k.

Note that (1) is the objective which consists of minimizing the total purchasing cost. Constraints (2) are used to make sure that the production requirements for any product are satisfied. Finally, relations (3) are the capacity constraints.

2.2. Problem complexity

The previous problem has a very specific structure. Variables are of two types: some of them are binary (variables x_i) and others (variables φ_i^k) are real, positive and upper bounded; but the biggest difficulty comes from the size of the problem: hundreds of machines and product types are commonly involved in problems of this type, each product type having in turn numerous manufacturing processes. The number of manufacturing processes related to a given product type can exceed several factors of ten.

Considering the fact that only the x_i-values are of interest in this problem, an approach based on Farkas–Minkowsky's theorem has been proposed. Unfortunately, this approach generates an unpredictable number of so-called generators which, in turn, generate additional constraints: the size of the problem usually explodes as soon as the number of available machines is greater than ten.

To conclude, efficient heuristic algorithms are requested in the above simple case. The mathematical problem is much more complicated when the objective function takes into account the running costs, which are usually not linear.

3. MANUFACTURING LAYOUT PROBLEM

The manufacturing layout problem is composed of three sequential functions, namely:

(i) the manufacturing cell design;
(ii) the intra-cell location;
(iii) the cell-location

(see [13]).

3.1. Manufacturing cell design

This function aims at grouping the machines of a shop into cells in order to optimize one (or several) objective function(s) under various constraints.

The most popular of these problems consists of grouping the machines into cells in order to minimize the inter-cell traffic, the number of machines in the cells being upper bounded. This problem is solved by using a heuristic approach based on simulated annealing. The way the problem is stated as well as the algorithm used to reach a near-optimal solution that satisfies manufacturers, has been judged by the industrial applications performed by the authors of this paper in France and in the U.S.A.

Note that the simulated annealing approach is particularly adequate to take into account additional constraints like the requirement of putting given machines in the same cell or, on the contrary, the necessity of separating two machines in different cells.

Open problems in this field involve the search for other realistic criteria and constraints and the solution of the related problems.

3.2. Intra-cell location

Assuming that the machines included in the cell are known, the next step consists in structuring the cells, i.e. locating the machines inside the cells, choosing the transporting devices, setting the buffers, etc.

Some very general rules are known.

Assuming, for instance, that a cell contains few machines (four or five), that the product flows are low and that each part is quite light and small, we can imagine a structure like the one given in Figure 1, where the transportation device is a robot.

If we now assume that the flow of the products is low, that the production inside the cell is flow-shop like, and that products are heavy and voluminous, we can use a structure like the one given in Figure 2, where the transportation system is a conveyor.

3.3. Cell location

Having performed the intra-cell location step, we can consider that a cell is a rectangular-shaped surface with an input and an output position. These positions may be different (i.e. products enter and leave the cell through different points of the cell border).

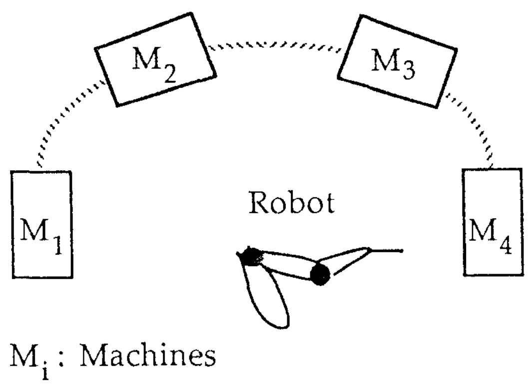


Figure 1. Intra-cell structure 1

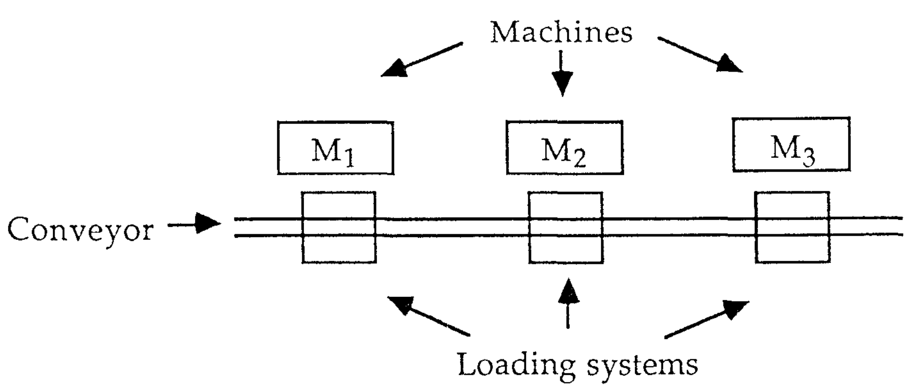


Figure 2. Intra-cell structure 2

The cell location step consists of locating the rectangular-shaped cells on the available surface in order to optimize some criteria, taking into account given constraints.

The most usual criterion is the sum of the terms (distance × flow) computed for every ordered pair (M_a, M_b) of cells, where:

(1) the distance is the distance between the output of machine M_a and the input of machine M_b;

(2) the flow is the rate of products circulating between the output of M_a and the input of M_b during significant period of time.

The constraints to take into account are of two types:

(a) strong constraints; for instance,
   (i) cells have to be located inside a limited surface (the surface being limited by walls, for instance);
   (ii) a given point on the available surface can only be taken up by one machine;

(b) weak constraints; for instance, a given set of machines should preferably be located in the same cell (because they are performing dirty tasks, or because they need the same technical skill to operate).

Simulated annealing is again a good technique for finding a near-optimal solution to the cell location problem. Nevertheless, the most difficult problem when using simulated annealing is the definition of the initial location of the cells on the given surface. Another problem is the fast computation of realistic paths between cells; despite the fact that algorithms (which are derived from graph theory) exist, it seems that drastic improvement is required in order to reduce the re-computation burden when removing or moving some of the cells.

4. EVALUATION OF THE BEHAVIOUR OF MANUFACTURING SYSTEMS

4.1. Overview

We assume that the characteristics of the resources are known. Once the layout of a manufacturing system is chosen, engineers have to find out what the future productivity of the system will be in relation to both the layout and the resource characteristics.

Very few tools are available to perform this evaluation. They are summarized in Figure 3.

As we can see, two families of tools are available for the evaluation of manufacturing systems, namely simulation techniques (see [9]) and mathematical tools (see [2, 5, 8, 11, 12]). We examine these approaches in the following sections.

4.2. Mathematical analysis

4.2.1. Petri nets: a promising approach

The most popular mathematical tools used to evaluate the dynamics of manufacturing systems are those derived from queueing theory and Petri nets. In our opinion, tools derived from queueing theory are irrelevant for this task mainly because the assumptions which have to be made to reach a promising model are contradictory with the regular running states of the system. For instance, good properties are available when we assume that a part arriving in a full buffer is lost! Furthermore, queueing theory is a statistically based approach, and the

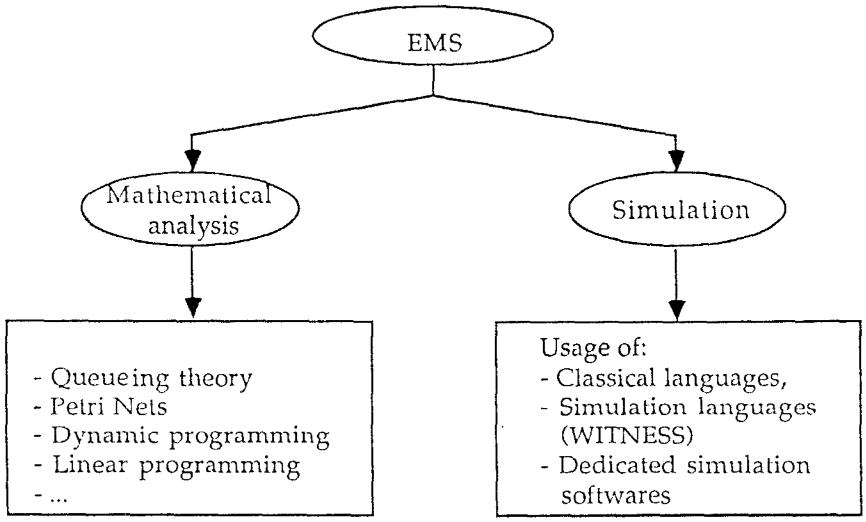


Figure 3. EMS: evaluation of manufacturing systems

flow of products in some manufacturing systems is not high enough to be handled in a statistical way.

For these reasons, we consider that queueing theory is irrelevant for evaluating manufacturing systems and we focus on Petri nets, which are able to model high or low flows as well as transient states.

Roughly speaking, the less sophisticated Petri nets have the most powerful properties. As a consequence, they will be used to model manufacturing systems when one aims at deriving properties related to the behaviour of types of manufacturing systems. Among those simple Petri nets are decision-free nets and, as a subset, event graphs. We define this type of net and present their properties in the following.

4.2.2. Event graphs: definition and properties

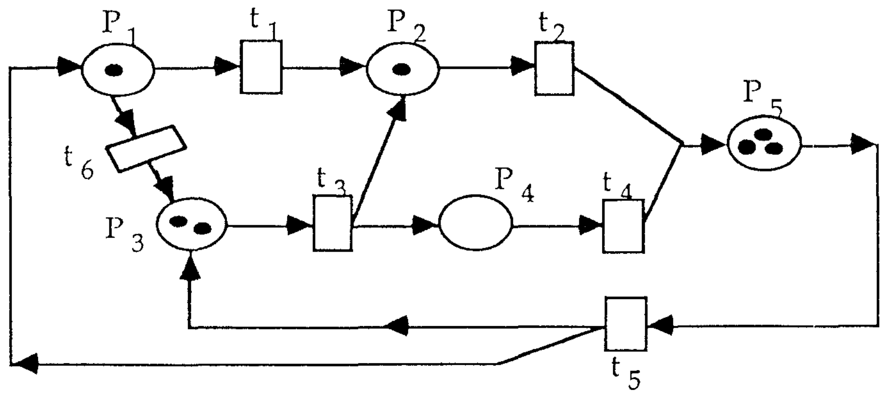
A Petri net is a bipartite directed graph made with two types of nodes called places and transitions. Directed arcs join places to transitions or transitions to places.

The input places of a transition are the places which are the origins of the arcs whose end is the transition considered. Similarly, the output places of a transition are the places at the end of the arcs whose origin is the transition considered. The definitions obtained by permutating 'place' and 'transition' in the previous definitions also hold.

A Petri net is represented in Figure 4. Places are represented by circles and transitions by bars. Tokens, which are the dynamic part of Petri nets, are represented by black dots. A Petri net along with its tokens is a 'marked Petri net'.

Formally, a Petri net is specified as a triplet $N=(P, T, F)$ where P is the set of places, T the set of transitions and $F \subset (P \times T) \cup (T \times P)$ is the set of directed arcs.

The 'marking' of a Petri net N is a function M: $P \rightarrow \{0, 1, 2, \ldots,\}$ which assigns a non-negative number of tokens to each place. A transition is 'enabled' if each of its input places contains at least one token. The 'firing' of a transition consists of removing one token from each of its input places and adding one token to each of its output places. In some Petri nets, it may arise that several transitions compete for firing. For instance, in the net of Figure 4, we cannot fire t_6 when firing t_1 nor fire t_1 when firing t_6 because there is only one token in P_1:



t_1, t_2, t_3, t_4, t_5 : Transitions

P_1, P_2, P_3, P_4, P_5 : Places

Figure 4. Marked Petri net

such a Petri net is 'not decision free' because we have to make a choice during the evolution of the net.

In a 'timed Petri net', each transition needs a given time to fire. In such a Petri net, a firing is 'initiated' by removing one token from each of the related input places; the token remains in the transition during the firing time, finally, the firing 'terminates' by adding one token in each of the output places. Note that firing times are either deterministic or random.

An 'event graph' is a Petri net such that each place has exactly one input transition and one output transition. The Petri net presented in Figure 4 is not an event graph because P_1 has two output transitions (i.e. t_1 and t_6).

An 'elementary circuit' is a directed path that goes from one node back to the same node while never going through any node more than once.

In an event graph, elementary circuits have the following important property.

Theorem 1

The number of tokens in any elementary circuit is invariant by transition firing.

In other words, if we consider an event graph with an initial marking M_0 and if we perform a sequence of transition firings, the number of tokens in any elementary circuit remains the same as for the marking M_0.

Let $\mu(\gamma)$ be the sum of the transition firing times in the elementary circuit γ. We define the cycle time $C(\gamma)$ of γ as

$$C(\gamma) = \mu(\gamma)/M(\gamma) \tag{5}$$

where $M(\gamma)$ is the total number of tokens in γ.

If the transition firing times are constant, the $C(\gamma)$ does not depend on the state of the system.

Let us denote by Γ the set of elementary circuits in the event graph $\gamma^* \in \Gamma$ is a critical circuit if

$$C(\gamma^*) = \max_{\gamma \in \Gamma} C(\gamma)$$

Assuming that the event graph is strongly connected and functioning in such a way that transitions fire as soon as they are enabled, the firing rate of any transition in the steady state is given by $\lambda = 1/C(\gamma^*)$.

That is to say that, for any transition t, a token enters (or leaves) t every λ units of time on the average. Thus, the critical circuit determines the speed of the tokens in the graph: the greater $C(\gamma^*)$, the more slowly the tokens move in the system.

An important result is given in the following theorem.

Theorem 2

An event graph is deadlock-free if and only if there is at least one token in every circuit.

4.2.3. FMSs modelling

The previous properties are used to evaluate flexible manufacturing systems (FMSs) in the steady state when using a particular type of control.

Assume that a given job-shop can manufacture a set $P_1, P_2, \ldots, P_n$ of product types.

For every P_i $(i = 1, 2, \ldots, n)$, the 'manufacturing process' is known. The manufacturing process is the sequence of machines to be visited by a part of P_i in order to be manufactured. We also know the time a part will spend at each machine. Thus, the manufacturing process of part P_i is as follows:

$$R_i\colon \{(M_{i,1}, \theta_{i,1}), (M_{i,2}, \theta_{i,2}), \ldots, (M_{i,k_i}, \theta_{i,k_i})\}$$

where, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, k$, M_{ij} is a machine of the job-shop; θ_{ij} is the time a part of type P_i spends on the jth machine of its manufacturing process to be manufactured; and k_i is the length of the manufacturing process in number of manufacturing steps.

We assign to each product type P_i a ratio q_i which represents the proportion of parts of type P_i the system has to manufacture in the steady state.

The control of the system is given by providing a sequence of product types to each machine. Let S_j be the sequence attached to machine M_j $(j = 1, 2, \ldots, m)$, where $M_1, M_2, \ldots, M_m$ are the machines of the FMS. The sequence S_j gives the order in which the types of products are manufactured by machine M_j. Note that each sequence S_j may contain the same type of product several times and that the proportions of the product type P_i in S_j fits with its ratio q_i. In other words, if $q_{i1} = \alpha q_{i2}$, then $n_{i1,j} = \alpha n_{i2,j}$ where $n_{i1,j}$ (respectively $n_{i2,j}$) is the number of times P_{i1} (respectively P_{i2}) appears in S_j, assuming that M_j appears in both R_{i1} and R_{i2}. Furthermore, if M_j does not appear in R_i, then P_i will not appear in S_j.

Let us consider the following small example to illustrate how to model an FMS using event graph. An FMS is composed of three machines (or cells) denoted by M_1, M_2 and M_3. This FMS can manufacture three types of products called P_1, P_2 and P_3 whose manufacturing processes are as follows:

$$\begin{aligned} &R_1\colon M_1(2);\ M_2(4);\ M_3(2)\\ &R_2\colon M_3(4);\ M_1(2)\\ &R_3\colon M_2(7);\ M_1(2);\ M_3(5) \end{aligned}$$

Quantities in parentheses are the manufacturing times.

We assume that the different types of product have to be manufactured, respectively, in the proportions 25%, 25% and 50%. that is to say

$$q_1 = q_2 = 0{\cdot}25 \quad \text{and} \quad q_3 = 0{\cdot}5$$

A sequence of product types fitting the previous ratios is $S = (P_1, P_2, P_3, P_3)$. Note that several sequences (i.e. all the sequences derived from S by permutating its elements) fit the given ratios.

The following sequences are chosen from machines M_1, M_2 and M_3 respectively:

$$S_1 = S = (P_1, P_2, P_3, P_3)$$
$$S_2 = (P_1, P_3, P_3)$$
$$S_3 = (P_1, P_3, P_3, P_2)$$

It is easy to verify that these sequences satisfy the desired conditions.

The previous system is modelled as shown in Figure 5. Each element of S is represented by an elementary circuit called 'process circuit'. For instance $(V_1, t_1, V_2, t_2, V_3, t_3, V_1)$ is the process circuit relating to product type P_1.

Transitions relating to the same machine are linked by an elementary circuit called a 'command circuit'.

For instance, $(V_{15}, t_1, V_{12}, t_5, V_{13}, t_7, V_{14}, t_{10}, V_{15})$ is the command circuit related to M_1; $(V_{16}, t_2, V_{17}, t_6, V_{18}, t_9, V_{16})$ is the command circuit related to M_2; $(V_{19}, t_3, V_{20}, t_{11}, V_{21}, t_8, V_{22}, t_4, V_{19})$ is the command circuit related to M_3.

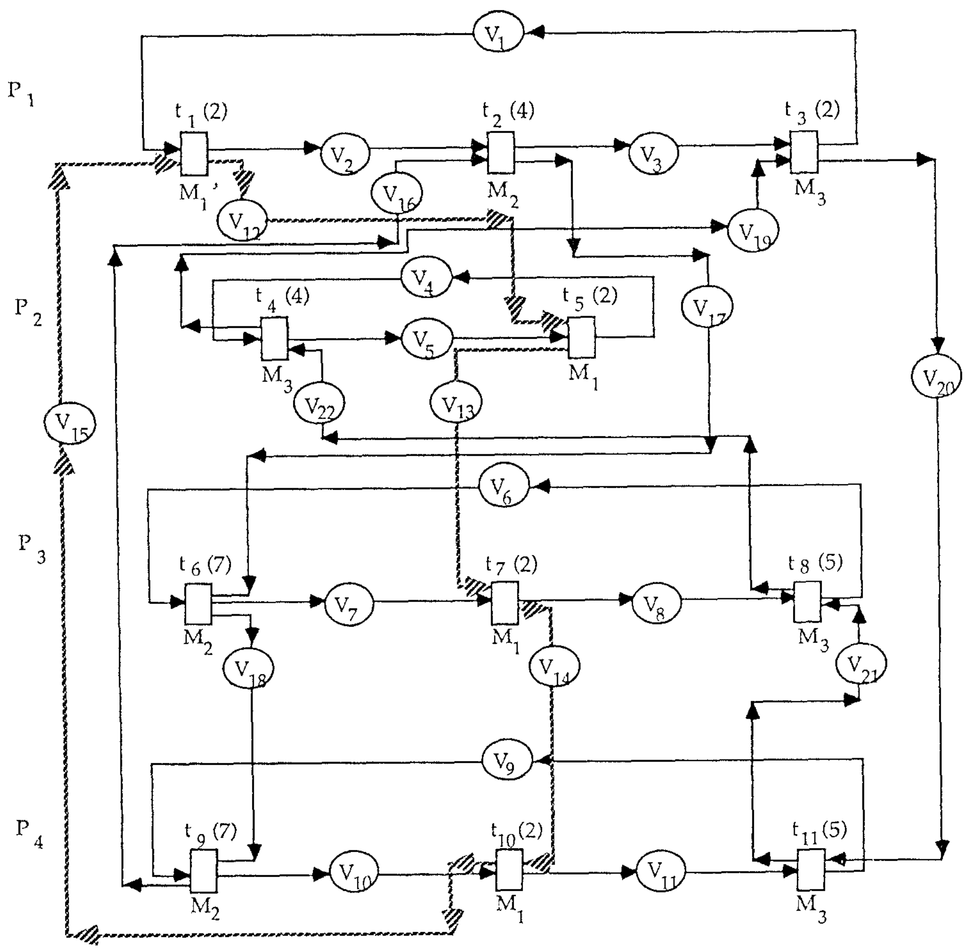


Figure 5. Event graph model

A command circuit models the sequence related to the machine: the order of the transitions in the command circuit corresponds to the order of the product types in the sequence relating to the machine under consideration.

We also see that the event graph obtained contains elementary circuits made with parts of command circuits and parts of process circuits. They are called 'mixed circuits'. For instance, in the model given in Figure 5:

$$(t_1, V_2, t_2, V_{17}, t_6, V_7, t_7, V_8, t_8, V_{22}, t_4, V_{19}, t_3, V_1, t_1)$$

is a mixed circuit. The number of mixed circuits in an event graph model cannot be forecast. This is the major problem we have to face when analysing such a model.

Each command circuit contains only one token. This token makes sure that the production is made with respect to the input sequence and that at most one product uses a machine at each time. Thus, such a token represents information.

At time 0, tokens will be located in V_{15}, V_{16} and V_{19} to fit S_1, S_2 and S_3. The other places of the command circuits will remain empty.

Other tokens will be located in the places belonging to the process circuits at time 0. Each of those tokens will represent a transportation resource (assuming that each transportation resource carries one product).

The goal of the study is to maximize the productivity of the system using as few transportation resources as possible.

According to the fact that the model is a strongly connected event graph, the following properties are derived from the results presented in the previous sections.

(a) An FMS is deadlock-free if there is at least one token in each elementary circuit of its model (including mixed circuits).

(b) If we put enough tokens in the places belonging to the process circuits, we can reach a situation where the critical circuit is a command circuit. In such a situation, the machine corresponding to the critical circuit is fully utilized and the productivity of the system is maximal. It has been proved that such a situation is attained when we put one token in each place of the process circuit.

4.2.4. Stating the problem

Let $\gamma_1, \gamma_2, \ldots, \gamma_q$ be the elementary circuits, $V_1, V_2, \ldots, V_h$ the places and

$$A = [a_{ij}]; \quad i = 1, 2, 3, \ldots, q; \quad j = 1, 2, \ 3, \ldots, h$$

the matrix defined as follows:

$$a_{ij} = \begin{cases} 1 & \text{if } V_j \in \gamma_i \\ 0 & \text{otherwise} \end{cases}$$

Futhermore, x_j, $j = 1, 2, \ldots, h$ is the number of tokens in V_j in the initial marking M_0.

Γ_c being the set of command circuits, we define

$$m^* = \operatorname*{Max}_{\gamma \in \Gamma_c} \mu(\gamma)$$

Let γ^* be a command circuit such that

$$m^* = \mu(\gamma^*)$$

In order that γ^* be a critical circuit, we must put at least $n(\gamma) = \lceil \mu(\gamma)/m^* \rceil$ tokens in each elementary circuit γ, $\lceil z \rceil$ being the smallest integer greater than or equal to z.

The following notation is also used:

$$N = [n(\gamma_1), n(\gamma_2), \ldots, n(\gamma_q)]$$
$$X = [x_1, x_2, \ldots, x_h]$$

Finally, the problem to be solved is an integer LP-problem:

$$\text{Min} \sum_{j=1}^{k} x_j \tag{6}$$

such that

$$AX^{\mathrm{T}} \geqslant N \tag{7}$$

$$x_j \geqslant 0 \quad \text{and integer for } j = 1, 2, \ldots, h \tag{8}$$

$$x_j \text{ is known (i.e. equal to 0 or 1) if } V_j \text{ belongs to a command circuit} \tag{9}$$

The most difficult problem concerns the number of constraints (7), owing to the unpredictable number of mixed elementary circuits.

4.2.5. Open problems

The previous approach to maximizing the productivity of the job-shop with a minimal number of transportation resources cannot be used for real-life problems because of the size of the resulting problem. Thus, an open problem could consist in finding a new and manageable approach to reach the same goal.

The case for which the manufacturing times are random has been studied extensively and some upper and lower bounds have already been proposed for the average cycle time. Nevertheless, it seems that these results should be refined and that some fast algorithms should be provided to reach this goal.

Finally, we think that local manufacturing rules such as LIFO, FIFO, etc. applied to job-shops have to be revisited using Petri net based models.

All the previous aspects have also to be considered for assembly systems.

5. PLANNING AND SCHEDULING PROBLEMS

5.1. Introductory remarks

Planning and scheduling problems have been recognized for their complexity see [1, 3, 4, 7]. The goal of a planning system is to provide, at a given horizon H, the list of tasks to be performed in each elementary period h. The goal of the scheduling system is then to assign the resources to the tasks and to define the beginning times of these tasks in order to optimize some criteria. These decisions are made according to various constraints. They are usually restricted to the first elementary period $[0, h]$, and the whole system works on a rolling horizon basis, as shown in Figure 6.

Results for production planning have not met their expectations. Furthermore, scheduling problems have been studied extensively, but, owing to their complexity, the most interesting results have been obtained for small systems which are very far from realistic situations.

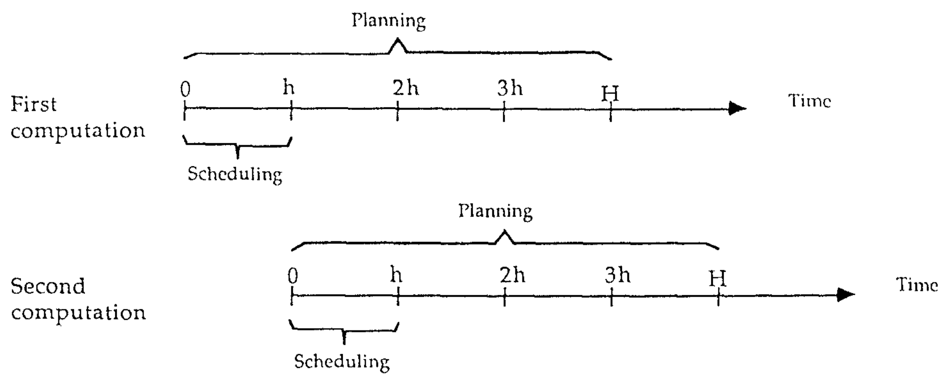


Figure 6. First two steps of the rolling-horizon process

5.2. Open problems

A hierarchical approach appears to be of great interest to overcome the complexity of real-time decision making in manufacturing systems. A (planning + scheduling) system is a two-level hierarchy. Several types of two-level and three-level hierarchy have been proposed so far, but these structures have to be justified on a sound basis. It seems to us that the key problem is to find a process which starts from the manufacturing system requirements or objectives and which provides:

(1) the number of levels of the hierarchy;
(2) at each level:
  (a) the horizon and the elementary period;
  (b) the model;
  (c) the criteria;
  (d) the constraints.

The most important problem in this field is then more to find a methodology to design a structure than to solve the problems (i.e. to optimize the decisions) at each level of the hierarchy.

Concerning the difficult problem of the lowest level of the hierarchy (i.e. the scheduling problem), the following two promising ways have been identified.

(i) For deterministic problems, study the properties of local manufacturing rules;
(ii) For stochastic problems, compare the efficiency of local manufacturing rules 'on the average'. This goal is less ambitious than the previous one, but we think that it should lead to interesting results.

6. CONCLUSION

Some open problems have been introduced in this paper. Of course, we do not claim that we have proposed an extensive list of unsolved problems, but this is the list of problems we consider as the most urgent to be tackled owing to new trends in manufacturing. To summarize: design, evaluation and hierarchical management of manufacturing systems will require substantial R&D effort in the near future.

REFERENCES

1. G. Bitran and A. Hax, 'On the design of hierarchical production planning systems', *Decision Scis*. **8** (1). 28–55 (1977).
2. F. Commoner, A. Holt, S. Evens and A. Pnuelia, 'Marked directed graphs', *J. Comput. System Scis.*, **5** (5), 511–523 (1971).
3. L. F. Gelders and L. N. Wassenhove, '*Production planning: a review*', *Eur. J. Ops. Res.* **7**, 101–110 (1981).
4. A. C. Hax and H. C. Meal, 'Hierarchical integration of production planning and scheduling', in M. A. Geisler (ed.), *Management Sciences*, Vol. 1, *Logistics*, North-Holland, Amsterdam, 1975, pp. 53–69.
5. H. P. Hillion and M. Proth, 'Performance evaluation of job-shop systems using timed event-graphs', *I.E.E.E. Trans. Autom. Control*, **34** (1) 3–9 (1989).
6. G. Hutchinson and D. Sinha, 'Quantification of the value of flexibility', *J. Manufg Systems*, **8** (1), 47–56 (1989).
7. S. Jahn, H. Kalb and U. Fischer, 'A concept for decentralization: production management—workstations and their tools', *IFIP Working Conference*, March 28–29, Munich, North-Holland, Amsterdam, 1985, pp. 160–167.
8. J. Martinez, P. Muro and M. Silva, 'Modelling, validation and software implementation of production systems using high level Petri nets', *Proc. I.E.E.E. Conf. on Robotics and Automation*, Raleigh, NC, April 1987, pp. 307–314.
9. A. A. B. Pritsker and C. D. P. Pedgen, *Introduction to Simulation and SLAM*, Halsted Press/John Wiley and Sons, Systems Publishing Corporation, West Lafayette, Indiana.
10. J. M. Proth, 'Flexibility, automation and modularity', in Madon, G. Singh (ed.), *Encyclopedia of Systems and Control*, Pergamon Press, London, 1990.
11. J. M. Proth and H. P. Hillion, *Mathematical Tools in Production Management*, Plenum Press, New York, 1990.
12. C. V. Ramamoorthy and G. S. Ho, 'Performance evaluation of asynchronous concurrent systems using Petri nets', *I.E.E.E. Trans. Softw. Enging*, **6** (5), 440–449 (1980).
13. Stark Draper Laboratory, *The Flexible Manufacturing Systems Handbook*, The Charles Stark Draper Laboratory, Cambridge, MA, 1982.