

University of Windsor

Scholarship at UWindsor

Industrial and Manufacturing Systems
Engineering Publications

Department of Industrial and Manufacturing
Systems Engineering

2006

On-line scheduling method of manufacturing system based on VS algorithm for reference pattern

J. A. Trujillo

Clemente Cárdenas

Zbigniew Pasek
University of Windsor

Enrique Baeyens

Follow this and additional works at: <https://scholar.uwindsor.ca/industrialengpub>



Part of the [Industrial Engineering Commons](#)

Recommended Citation

Trujillo, J. A.; Cárdenas, Clemente; Pasek, Zbigniew; and Baeyens, Enrique. (2006). On-line scheduling method of manufacturing system based on VS algorithm for reference pattern. *proceeding WAC, Budapest, Hungary*, 1-8.

<https://scholar.uwindsor.ca/industrialengpub/10>

This Article is brought to you for free and open access by the Department of Industrial and Manufacturing Systems Engineering at Scholarship at UWindsor. It has been accepted for inclusion in Industrial and Manufacturing Systems Engineering Publications by an authorized administrator of Scholarship at UWindsor. For more information, please contact scholarship@uwindsor.ca.

On-line Scheduling Method of manufacturing system based on VS algorithm for reference pattern

Jesús Trujillo and Clemente Cárdenas and Zbigniew J. Pasek and Enrique Baeyens

Abstract—In this paper, a Scheduling method is developed to provide planning for Manufacturing plants with multiple coordinating cells. The controls for reconfigurable manufacturing systems have to be capable not only of identifying exceptions on-line, but also simultaneously developing on-line strategies for unpredictable customer orders or inaccurate estimate of processing times. The approach exploits a complementary algorithm for (VS) reference pattern and Virtual Supervisor (VS), which has access to all system information during program execution and thus can readily monitor the overall system performance. The goal is to minimize expected part tardiness and earliness cost. A solution methodology based on a combined Lagrangian relaxation, VS-Patterns, Maxwell equations and temporal difference is developed to reduce the computation requirements for large problems. Sequences pattern shows that near optimal schedules can be obtained a dual solution for on-line implementation.

I. INTRODUCTION

Many manufacturing systems are organized in cells, and product flows across cells for processing. These cells require be reconfigured such as a reconfigurable manufacturing system (RMS), scheduling decisions and exception handling polices become more complex since multiple reconfiguration strategies have to be considered. The reconfigurability feature turns out to be a new technological factor enabling novel strategies for handling out-of-order events of the production process (machine breakdowns, job priority changes, unexpected job arrivals or cancellations, etc.) [7], [1]. This paper presents a new case based similarity assessment approach which addresses these problems.

II. PROBLEM FORMULATION

The generic job-shop problem is extremely complex [1], and a complete solution algorithm for solving it does not exist. The problem involves N discrete time units, ranging from 0 to $N - 1$, R machine types and J parts to be processed. Let the indexes r and s denote the type of machine. The available number of r -type machines ($1 \leq r \leq R$) at time n is given and denoted by η_{nr} . The number of r -type machines that could be substituted by s -type machines is denoted by $\eta_{nr s}$. Part j ($1 \leq j \leq J$) has arrival

J. Trujillo is with the department of Ingeniería de Sistemas y Automática, Universidad de Valladolid, Paseo del Cauce s/n, Valladolid E-47011, SPAIN jestru@eis.uva.es

C. Cárdenas is with CARTIF manufacturing, automatization, robotics and information technologies Center, Universidad de Valladolid, Paseo del Cauce s/n, Valladolid E-47011, SPAIN enrbae@eis.uva.es

Z. Pasek is with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA zbigniew@umich.edu

E. Baeyens is with the department of Ingeniería de Sistemas y Automática, Universidad de Valladolid, Paseo del Cauce s/n, Valladolid E-47011, SPAIN enrbae@eis.uva.es

time Γ_j , due date D_j , and priority (weight) W_j . In RMS some machines can change their configuration [3]. Let P denote the number of redundant lines. The available number of p -type ($1 \leq p \leq P$) redundant lines at time n is given and denoted by L_{np} . Processing part j requires a set of I_j operations for completion without assembly requirements. Let $\{j, i\}$ denotes operation i ($1 \leq i \leq I$) on part j . The first operation on part j , $\{j, 1\}$, can only be started after the arrival of an order or when the raw materials are available. Operation $\{j, i\}$ has to be performed on a machine type r for a specified processing time t_{jir} and the operation may start only after operation immediately it has been completed. For some parts, the arrival time Γ_j , processing time t_{jir} , due date D_j , and priority W_j , are not exactly known in advance. These parameters are modelled as independent random variables with known discrete probability distributions. The machine availability is assumed to be deterministic. The objective is to maximize on-time delivery of parts and to reduce work in process (WIP) inventory. The problem is characterized as follows with a list of symbols provided in Table I for easy reference.

- 1) *Arrival time constraints*: the first operation of part n cannot be started until the arrival of an order or the appropriate raw material is available, i.e.,

$$\Gamma_j \leq B_{j1}, \quad j = 1, \dots, J \quad (1)$$

where B_{j1} is the beginning time of $(j, 1)$.

- 2) *Operation precedence constraints*: The operation precedence constraints state that operation $(i + 1)$ of part j cannot be started before the completion of operation I of part j plus a deadtime O_{ji} ,

$$k_{ji} + O_{ji} \leq B_{j,i+1}, \quad j = 1, \dots, J, \quad i = 1, \dots, I_{j-1} \quad (2)$$

where k_{ji} is the completion time of (j, i) , and $B_{j,i+1}$ is the beginning time of $(j, i + 1)$.

- 3) *Processing time requirements*: An operation i of part j must be assigned the required amount of processing time t_{jir} , i.e.,

$$k_{ji} = B_{ji} + T_{ji} + t_{jir} \leq B_{j,i+1}, \quad j = 1, \dots, J, \quad i = 1, \dots, I_j \quad (3)$$

- 4) *Replacement machine requirements*: For any r -type machine substitution by an s -type machine, the completion time of part k_{ji} plus deadtime O_{ji} is less than beginning time B_{j1} plus arrival time Γ_{ji} . Substitution

TABLE I
SYMBOLS FOR JOB-SHOP PROBLEM

Symbol	Description
n	Time index
k_j	Completion time
N	Total scheduling time
δ	Dual cost
r	Machine type index
D_j	Due date of part j
R	Set of machine types
E_j	Earliness part j
Γ_j	Arrival time of part j
T_j	Tardiness of part j
t_j	Processing time of part j
B_j	Initial time of part j
O_j	Time out of part j
s	Substitute machine
η_{nr}	Number of r -type machines
η_{nrs}	Number of r -type machines that can be substituted by s -type machines
P	Redundant line
ξ_j	Cumulative cost
W_j	Weight of tardiness
ϖ_j	Weight of earliness
π_j	Lagrangian Multiplier
L_{np}	Available number of p -type redundant lines
$ x $	Planning function
C_j	The samples from a known class set
$p(y = c_j)$	The probabilities
$\rho(x/c_j, w_j)$	The density function
w_j	The unknown values are parameters for c
$p^*(c_i)$	The sample fraction of c_i
m_i^*	The measures of samples
C_i^*	The covariance matrix of samples

is only carried out when priority is maximal.

$$\begin{aligned} &\text{If } W_{ji} \geq W_{\max}, \text{ and} \\ &k_{ji} + O_{ji} \leq B_{j,i+1} + \Gamma_{ji}, \\ &j = 1, \dots, J, \quad i = 1, \dots, I_{j-1} \end{aligned} \quad (4)$$

5) *Machine capacity constrains*: The number of operations assigned to an r -type machine at time n should be less than or equal to η_{nrs} (the number of machines available at that time),

$$\begin{aligned} \sum_{ji} \theta_{jinrs} &\leq \eta_{nrs}, \quad n = 0, \dots, N-1, \\ &r \in R \end{aligned} \quad (5)$$

where θ_{jinrs} is a boolean variable. It equals one if task $\{j, i\}$ is assigned to an r -type machine at time n , and zero otherwise. For random arrival processing times, handling machine capacity constrains (4) for all possible instances of random events is very difficult because of complexity. The feasible model is a schedule satisfying (1)-(6)

$$\begin{aligned} E \left[\sum_{ji} \theta_{jinrs} \right] &\leq \eta_{nrs} \leq L_{np}, \\ &\text{on redundant line case,} \\ n = 0, \dots, N-1, \quad r \in R, s \in S \end{aligned} \quad (6)$$

6) *Objective function*: The objective function is a weighted sum of penalties for parts tardiness T_j and raw materials earliness E_j . Therefore, the following optimization problem is formulated

$$\begin{aligned} &\min_{\{B_{ji}, r_{ji}\}} I, \\ &\text{where } I = E \left[\sum_{j=1}^J (W_j T_j^2 + \varpi_j E_j^2) \right] \end{aligned} \quad (7)$$

subject to constraints (1)-(6)

In the next section, a heuristic scheduling list is used to dynamically construct the schedule based on the optimization solution and the realization of random actions (events).

III. SOLUTION APPROACH

A. Gradient Projection Method

This numeric method for obtaining the minimum keep under to equality restrictions can be applied after introducing Lagrange multipliers to hold expected machine capacity constraints (6). The following problem is obtained

$$\begin{aligned} &\min_{\{B_{ji}, r_{ji}\}} \mathcal{L}, \text{ where} \\ &\mathcal{L} = E \left[\sum_j (W_j T_j^2 + \varpi_j E_j^2) \right] + \\ &\sum_{nrs} \pi_{nrs} \left\{ E \left[\sum_{ji} \theta_{jinrs} \right] - \theta_{nrs} \right\} \end{aligned} \quad (8)$$

By using the conditions imposed on capacity constraints in (5) and regrouping relevant terms, the problem can be decomposed into the following part-level subproblems:

$$\begin{aligned} &\min_{\{B_{ji}, r_{ji}\}} \mathcal{L}_j, \text{ where} \\ &\mathcal{L}_j = E \left[W_j T_j^2 + \varpi_j E_j^2 + \sum_{i=1}^{I_j} \sum_{n=B_{ji}}^{k_{ji}} \pi_{nrs} \right], \\ &j = 1, \dots, J \end{aligned} \quad (9)$$

subject to (1)-(6). Let denote the resulting minimal subproblem cost. The dual problem is then obtained as,

$$\begin{aligned} &\max_{\{\pi_{nrs}\}} \delta, \text{ where} \\ &\delta = \sum_j \mathcal{L}_j - \sum_{nrs} \pi_{nrs} \eta_{nrs} \end{aligned} \quad (10)$$

B. Temporal-Difference Method (TD)

TD method can learn directly from patterns (reference set structures) without a model of the environment [5]. This method updates estimates partially based on other learned references, without waiting for the final outcome. In this paper, backward stochastic dynamic programming is used on part subproblems (9) to manage uncertainties. In this procedure, each TD/DP (dynamic programming) stage corresponds to an operation. At each stage, the positions

are the possible operation beginning times. The subgradient component $E[\sum \theta_{jinrs} - \eta_{nrs}]$ which is required to update the multipliers, is calculated based on subproblem results. Next, the TD-DP procedure is illustrated for the deterministic case.

- 1) TD deterministic case: In this case, all parameters of part j are deterministic. The gradient-descent procedure was applied, although for effectiveness reasons it has been parametrically combined with conventional TD methods. The algorithm starts at the last stage having the following terminal cost:

$$\begin{aligned} \varsigma_{ji}(B_{ji}, r_{ji}, s_{ji}) = & \\ & W_j T_j^2 + \sum_{n=B_j I_j}^{k_j I_j} \pi_{nr_{jI_j} s_{jI_j}} \end{aligned} \quad (11)$$

The cumulative cost when moving backward is then obtained recursively as follows,

$$\begin{aligned} \varsigma_{ji}(B_{ji}, r_{ji}, s_{ji}) = & \\ & \min_{\{B_{j,i+1}, r_{j,i+1}, s_{j,i+1}\}} \varpi_j E_j^2 A_{ji} + \\ & \sum_{n=B_j I_j}^{k_j I_j} \pi_{nr_{jI_j} s_{jI_j}} + \\ & \varsigma_{j,i+1}(B_{j,i+1}, r_{j,i+1}, s_{j,i+1}) \end{aligned} \quad (12)$$

where A_{ji} is an integer variable that equals one if operation $\{j, i\}$ is the first operation of part j , and zero otherwise. The optimal \mathcal{L} is obtained as the minimal cumulative cost at the first stage, subject to the arrival time constraint. Finally the optimal beginning times and the corresponding machine types can be obtained by tracing the stages forward.

The TD algorithm for the uncertain case is similar to the deterministic case. The terminal cost for the stochastic case is given by (13), where the expectation is taken with respect to all possible processing times of the last operation and weights.

- 2) Solving subproblems with uncertain processing times: When the processing times t_{jnr} are random and other parameters of part j are deterministic, the terminal cost is the expected value of all these possible costs,

$$E \left[W_j T_j^2 + \sum_{n=B_j I_j}^{k_j I_j} \pi_{nr_{jI_j} s_{jI_j}} \right] \quad (13)$$

The associated cost is obtained as in (11). Thus, the cumulative costs of the positions are then the expected value of all the above costs. The intermediate position shows level corresponds to diagram VS, in (1). The level 5 in y_2 is refereed at position 0, level 4 at position 1, similar way still level 0 that is corresponding at position 5. Levels are representing to

the control actions and these are produced between positions shown at y_1 axis.

$$\begin{aligned} \varsigma_{ji}(B_{ji}, r_{ji}, s_{ji}) = & \\ & E \left[\varpi_j E_j^2 A_{ji} + \sum_{n=B_j I_j}^{k_j I_j} \pi_{nr_{jI_j} s_{jI_j}} + \varsigma_{j,i+1}^* \right] \end{aligned} \quad (14)$$

where

$$\begin{aligned} \varsigma_{j,i+1}^* = & \\ & \min_{\{B_{j,i+1}, r_{j,i+1}, s_{j,i+1}\}} \varsigma_{j,i+1}(B_{j,i+1}, r_{j,i+1}, s_{j,i+1}) \end{aligned} \quad (15)$$

This procedure continues until the cumulative costs for all the positions at the first stage are obtained.

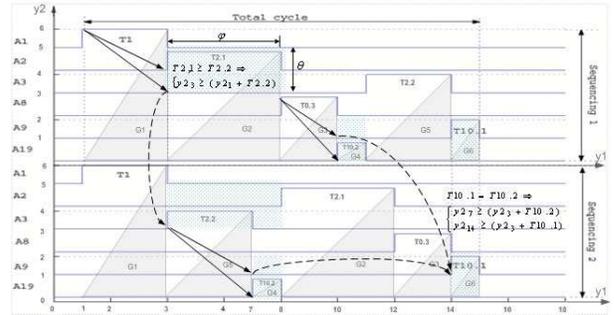


Fig. 1. Dynamic programming TD. Regular transitions between nodes

- 3) Obtaining subgradients: The subgradient,

$$E \left[\left[\sum_{ji} \theta_{jinrs} \right] - \eta_{nrs} \right]$$

The multipliers have to be updated in short periods, where quantifiable changes are produced under known periods. An important step in obtaining the subgradient is calculation of the expected machine utilization $E[\theta_{jinrs}]$ for stage i of piece j . If it can be obtained, every intermediate positions are located in optimal way TD for piece j , and the probabilities that they are chosen as beginning times are determined. An intermediate position node, in optimal way at piece j , the machine utilization θ_{jinrs} can be calculated for the optimal machine type and each processing possible time. The machine utilization $E[\theta_{jinrs}]$ associated with the node, then it will be the probability, where this node will be the expected value of all these θ_{jinrs} multiplied by probability when the intermediate position is chosen.

Finally, expected machine utilization $E[\theta_{jinrs}]$ for this stage j is the addition of machine utilization $E[\theta_{jinrs}]$ for all these intermediate positions. This procedure is illustrated as follows: Let $P(1)$ show the all intermediates positions set on the optimal path corresponding to stage 1. For any intermediate position $p(1) \in P(1)$, given $T_{\Gamma}(p_1)$ denote the all

possible arriving time set having position (p_1), and thus optimal beginning time.

The probability $\rho(p_1)$, where intermediate position be chosen as beginning time is the sum of probabilities ρ_Γ associated with all these possible arrival times, as follows:

$$\rho_{p_1} = \sum_{\Gamma \in T_\Gamma(p_1)} \rho_\Gamma, \quad p_1 \in P(1) \quad (16)$$

When moving from stage i to stage $i + 1$ ($1 \leq i \leq I_j - 1$), let $P(i + 1)$ denotes the set of all intermediate positions on the optimal path ($i + 1$). For any intermediate position $\rho_{i+1} \in P(i + 1)$, let $PT(\rho_{i+1})$ show all pairs of $\rho_i \in P_i$ and possible processing times t having ρ_{i+1} as the optimal beginning time for operation $i + 1$. The probability $\rho_{p_{i+1}}$, where the intermediate position p_{i+1} is chosen as beginning time from all probabilities sum $\rho_{p_{i+1}}\rho_t$ associated with these pairs (p_i, t), is as follows:

$$\rho_{p_{i+1}} = \sum \rho_{p_i}\rho_t, \quad p_{i+1} \in P(i + 1) \quad (17)$$

Finally the subgradient is calculated by:

$$E \left[\sum_{ij} \theta_{ijnrs} \right] - \eta_{nrs} = \sum_{ij} E[\theta_{ijnrs}] - \eta_{nrs} \quad (18)$$

The calculation complexity $E[\sum_i \theta_{ijnrs}]$ is $O(C \sum_i |T_{jins}|)$ because intermediate position numbers at a stage is the sum C . When this method is used to calculate subgradient for deterministic case. The complexity is equal to the deterministic method [1].

C. Overload problem

All terms are taking in \mathcal{L} and it binds to overload resource, this is used such as leader guide of sequences to next overload problems:

$$\min_{\{f_{nr}\}}, \text{ where } F_{nr} = (W_r^f \cdot f_r^2 n - \pi_{rn} \cdot f_{rn}), \quad f_{rn} \geq 0. \quad (19)$$

The solution of these subproblems is given by expression:

$$f_{nrs} = \left(\frac{\pi_{nrs} / (2 \cdot W_r^f)}{m} \right) \cdot m \quad (20)$$

Where m is smallest unit to be adjusted f_{nrs} and $|x|$ is the planning function.

D. The Dual Problem

The dual cost function in (10) is concave, piece-wise linear, and consists of many phases [4]. Each phase corresponds to a possible scheduling policy of the problem. The number of possible scheduling policies significantly increases with the problem size. The reasons are the combinatorial nature of discrete optimization and the presence of uncertain factors.

A conjugate gradient method is used to iteratively solve the high level dual problem (10), but using subgradients instead of gradients. Through a given set of multipliers, subproblems are solved to obtain the optimal subproblem solutions, and multipliers are then updated based on degrees of constraint violation using the conjugate subgradient method. This iterative procedure is repeated until some stopping criteria are met. Computation of the objective function (7) for a single dual solution involves simulation and is very time consuming. The idea of optimization is employed to perform short simulation runs on selected candidate dual solutions to determine the ranking of their expected costs. A winner (substituted) of the short tryout is then the dual solution selected to generate pattern schedules, and feasible simulation runs are then accomplished to obtain performance statistics. Figure 2 shows a block diagram of the algorithm.

E. Building Pattern Schedules

In our work several heuristic methods were developed in view of the existence of resource overload. The method extends the approach of [10] by adding overload problem solution to the original resource capacities. These methods are carried out alternatively during the multiplier update iterations, and the schedule with the lowest cost is recorded as a pattern schedule. The relative difference between this pattern cost i and the maximum dual value D is the relative gap $(J - D)/D \times 100\%$, and it quantifies the quality of the schedule obtained. The scheduling must be usually performed periodically at the beginning of a shift based on a *snapshot* of the factory status. Scheduling update might a be needed after the arrival of major orders or breakdown of critical resources. Since the status of the factory may not change drastically, rescheduling can then be initialized by using multipliers patterns (obtained from previous schedules). This initialization provides a better starting point for the optimization process, and significantly reduces the computational requirements as will be illustrated in next section.

IV. SYSTEM OPERATION MONITORING VIA VIRTUAL SUPERVISOR

A. Monitoring methods

VS through setup pattern [6] and inductive Maxwell method [7] is capable of monitoring the manufacturing process at the plant, machine, and device level. It updates the equivalent model at each clock interval. Thus, the manufacturing plant is checked on line against the model generated in a virtual space, where it is also compared with the reference setup pattern sequence.

B. Real time processing

The process works as follows: each machine, process and parts are assigned a level of resistance using coefficients and previously described conditions, see the details in [7]. The potential induced by each machine depends on these

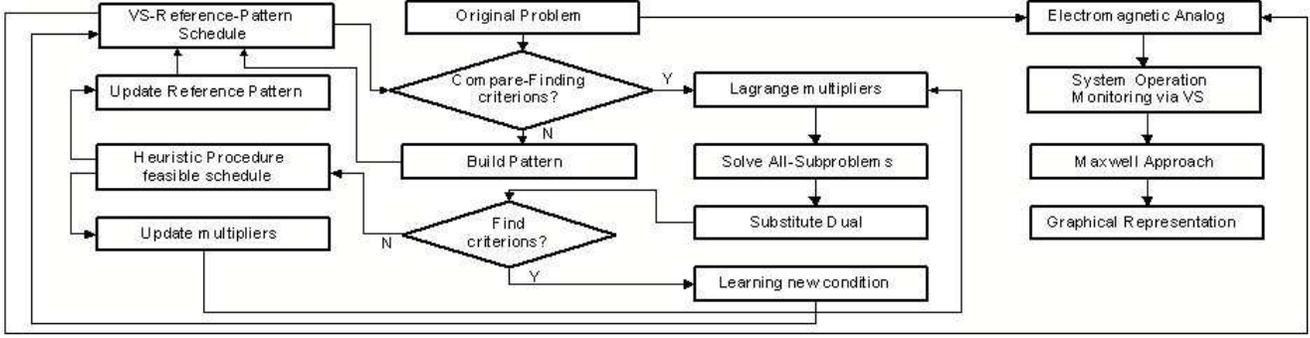


Fig. 2. Overall system for Job-Shop Scheduling via VS

coefficients and conditions, e.g. for a machine on a path of critical flow, since a critical path has high priority, the induced potential will be higher.

$$R_{ec} = \frac{\sum_k \sum_j v_{kj}}{In} ((1 + \varsigma)(1 + \zeta)(1 + x)) e^{-(x+1)^2} \quad (21)$$

$$k = 1, 2, \dots, n, \quad j = 1, 2, \dots, 2n$$

The coefficient ς reflects the path criticality and its value depends on priority level. The inductor is a manufactured workpiece affected by other coefficients:

$$v_0 = \frac{\sum_k \sum_j v_{kj}}{n} (1 + W_j)(1 + \varpi_j)(1 + s_j) e^{-(x+1)} \quad (22)$$

$$k = 1, 2, \dots, n \quad j = 1, 2, \dots, 2n$$

where W_j and ϖ_j are the priority or weight of tardiness and earliness penalty for part j , respectively, and s_j is the index of the possible substituted machines.

C. On Estimating maximum similarity criteria

Our similarity approach exploits the case library of patterns that are ready to be used. Such a method is appealing for a number of reasons. It simplifies knowledge schedules for similarity criteria, because similarity judgments are based on experience with adaptation rather than a priori analysis that may be hard to connect to actual performance on specific schedules or process sequences. It provides a finer-grained method for estimating adaptation costs, reflecting knowledge of individual prior problem. It also provide a simple way to refine similarity criteria as new schedules are learned. The method begins performing next considerations:

- i. It is considered a set $X = x_1, \dots, x_n$ of n samples
- ii. The samples are from a known class set, C_j
- iii. The probabilities prior $p(y = c_j)$ of pertinence to a specific class are known, $j = 1, \dots, c$
- iv. The density function forms are a conditional probability $\rho(x/c_j, w_j)$ are known, $j = 1, \dots, N$
- v. The just unknown values are parameters w_j for c arrays.

The patterns are obtained selecting a class c_j with probability $p(y = c_j)$, and subsequently it is selected a x as much as probability rule $p(x/c_j, w_j)$ [2]. The probability density function for the samples came given as follows:

$$\rho(x/w) = \sum_{j=1}^c \rho(x/c_j, w_j) p(y = c_j) \quad (23)$$

where $w = (w_1, w_2, \dots, w_c)$. A density function of this form is denoted as *mixture density*. The conditional probability density $\rho(x/c_j, w_j)$ previously are the components density and probabilities $p(y = c_j)$ are *mixture parameters*. These parameters can be included between unknown parameters. The unknown parameters are representing the unknown schedules.

Let *maximum similarity*, where it is supposed that there is a set $X = x_1, x_2, \dots, x_n$ of n no market samples and the density is considered as mixture Eq. 23. The observed samples similarity is by definition

$$p(X/w) = \prod_{k=1}^n \rho(x_k, w) \quad (24)$$

The maximum similarity estimated w^* is the value of w that maximizes $p(X/w)$. The main concepts are contained in [11] and supposing that $p(X/w)$ is a differential function in w ,

The maximum similarity estimated w^* is value w that maximizes $p(X/w)$. Considering $p(X/w)$ is a differentiable function to w [11],

$$l = \sum_{k=1}^n \ln \rho(x_k/w) \quad (25)$$

and

$$\nabla_{w_i} l = \sum_{k=1}^n \frac{1}{\rho(x_k/w)} \nabla_{w_i} \left[\sum_{j=1}^c \rho(x_k/c_j, w_j) p(c_j) \right] \quad (26)$$

where l represents similarity logarithm and $\nabla_{w_i} l$ gradient of l respect to w_i ; by notational simplicity has been substituted expression $p(y = c_j)$ by $p(c_j)$.

Let's suppose that elements w_i y w_j are functionally independent if $i \neq j$, and the probability would be introduced afterward,

$$p(c_i/x_k, w) = \frac{\rho(x_k/c_i, w_i) p(c_j)}{\rho(x_k/w)} \quad (27)$$

with this expression, the logarithm gradient can be writing as follows:

$$\nabla_{w_i} l = \sum_{k=1}^n p(c_i/x_k, w) \nabla_{w_i} \ln \rho(x_k/c_i, w_i) \quad (28)$$

since the gradient must be zero in w_i that this maximize l , the maximum similarity estimated w_i^* must satisfy the conditions:

$$\sum_{k=1}^n p(c_i/x_k, w^*) \nabla_{w_i} \ln \rho(x_k/c_i, w_i^*) = 0, \quad i = 1, \dots, c \quad (29)$$

Back form between the solutions to these equations by w_i^* where is founded the maximum similarity solution.

The obtained results can be generalized to include the probabilities prior $p(c_j)$ as unknown parameters. In this case the search of maximum value of $p(X/w)$ is extended to w and $p(c_j)$ with restrictions,

$$p(c_i) \geq 0 \quad \sum_{i=1}^c p(c_i) = l \quad i = 1, \dots, c \quad (30)$$

Let $p^*(c_i)$ and w_i^* the maximum similarity estimations by $P(c_j)$ and w_i . If the similarity function is differentiable and if $p^*(c_i) \neq 0$ for any i , then $p^*(c_i)$ and w_i^* must satisfy,

$$p^*(c_j) = \frac{1}{n} \sum_{k=1}^n p^*(c_j/x_k, w^*) \quad (31)$$

$$\sum_{k=1}^n p^*(c_i/x_k, w^*) \nabla_{w_i} \ln \rho(x_k/c_i, w^*) = 0 \quad (32)$$

where

$$p^*(c_i/x_k, w^*) = \frac{\rho(x_k/c_i, w_i^*) P^*(c_i)}{\sum_{j=1}^c \rho(x_k/c_j, w_j^*) P^*(c_j)} \quad (33)$$

There are two different case [8] with a normal density. In the first case is considered that unknown parameters are the average arrays. In the second case are all parameters, and the prior probabilities $p(c_j)$ are unknown. The measures array m_i and the covariance matrix C_j . In this last case the maximum similarity estimations can be as follows:

$$p^*(c_i) = \frac{1}{n} \sum_{k=1}^n p^*(c_i/x_k, w^*) \quad (34)$$

$$m_i^* = \frac{\sum_{k=1}^n p^*(c_i/x_k, w^*) x_k}{\sum_{k=1}^n p^*(c_i/x_k, w^*)} \quad (35)$$

$$C_i^* = \frac{\sum_{k=1}^n p^*(c_i/x_k, w^*) (x_k - m_i^*) (x_k - m_i^*)^t}{\sum_{k=1}^n p^*(c_i/x_k, w^*)} \quad (36)$$

where

$$p^*(c_i/x_k, w^*) = \frac{\rho(x_k/c_i, w_i^*) p^*(c_i)}{\sum_{k=1}^c \rho(x_k/c_j, w_j^*) p^*(c_i)} \quad (37)$$

This expression is really complex, although its main part is fairly elementary. It can be found also in this second case an important reference in [9], [11]. In the extreme case $p^*(c_i/x_k, w^*)$ take the value one when x_k is of class c_i and zero to another case. The equation $p^*(c_i)$ is the sample fraction of c_i , m_i^* is the measures of these samples and C_i^* is the covariance matrix of samples. On general form, $p^*(c_i/x_k, w^*)$ takes values between zero to one and all samples take any sequence on the estimation.

This method can identify the maximum similarity of schedule, and taking the pattern, thus the process will be a feasible and optimized process. Manufacturing system represents cyclical process that allows their identification and comparing can be obtained their schedule rapidly. The schedule can be subdivide by others smaller structures such as modular blocs. This blocs adequately composed can be form a feasible and optimized schedule.

D. Learning of new conditions

The induction method combined with the reference pattern contains enough information to deal with the conflict. The new sequence will have a new order imposed by the position determined by induction. This way the potential provides the order and priority magnitudes required by the controller to drive the control action. $v_{k1} \gg v_{k2} \gg \dots \gg v_{kn}$, $\{k1, \dots, kn\} \subseteq [\sum_n E_n, \sum_n X_n, \sum_n V_n]$ where (E_n , X_n , V_n are pattern events, states, and times respectively) [7], [6], where k_n becomes established by sequence imposed by the value of ν_{kn} . The exception S_{ne} is integrated as a new reference pattern after being optimized and verified $S_{ne} \subseteq \sum_n H_n$, where $\sum_n H_n$ is a setup state-event-time pattern. The relation of pattern references contains the list of operations, and the process, part and machine can be obtained from it. Thus, the algorithm can take the order and identify the exact position required by VS to simulate on line the scenario for job-shop scheduling.

V. EXAMPLE. INDUCTIVE METHOD

This example is to demonstrate using the value of resource overload, and to see that low overload can be achieved by properly selecting the overload penalty coefficients. There are eleven cells each one unit of a key resource. Twelve products each with three operation are to be scheduled, and the processing is required to be finished before time unit 60. For all the products, due dates and weights are 1 and 11 respectively and there is no earliness or lead time penalty.

There is no optimal schedule if resource overload is not allowed. By allowing overload, optimal schedules are generated in less than 60s for different values of overload penalty coefficients w_k . The optimal cost and overload of the schedules are shown in sequence by patterns VS in Figure 3. Very large penalty coefficients, which are similar to the no overload allowed case, result in small overload but large tardiness. If coefficient are very small, the schedule is similar to that in MRP *material requirement planing*

with infinite capacity, resulting in large overload and small optimal cost J . There are several possible sequence can achieve optimal schedule. These schedule has a corresponds pattern plans for each cell. Induction method allows decide which is the possible interaction with all plans possible. Better pattern plan dependent of realized sequence. This form multipliers could be selecting which pattern family is optimal and induction method is allowing with of all pattern is the optimal schedule. In figure Figure 4 show the machine with maximum priority before the sequence task be selected.

In this chart a series of operations marked with red line form a critical path, where the machine $M3$ breaks down after operation 2 in $J2$ for half a time unit. (The repair time for the machine is known in advance). The on-line supervisor is showing the scenario represented in Fig. 2 Using the proposed method is possible to automatically recognize what machines produce a conflict and obtain the priority order of a new scheduling.

A. Interpretation of results

VS can build in advance the estimated situation in a virtual space, where it recognizes by induction how each machine is working, and it can preview what the situation is for a given workpiece arrival at specific machines. The Gantt charts of the resulting schedules are shown in Fig. 3 with a 15% lower expected cost than that of the conflict scenario in Fig. 2. The reason can be explained as follows. In the conflict scenario the delay produced at machine $M3$ provokes future delays in the operations where this machine has part process, affecting to $M2$, $M5$, $M9$ shows become overloaded. The priority for each process, part, and what machine could absorb the overload created by $M3$ in a new job-shop scheduling is obtained from patterns and TD algorithm. Thus Fig. 3 shows the tending evolution to balance the line for overall vs system. Obtaining a real on line predictive state space, where the controller can get the information required to perform an exception decision.

VI. CONCLUSIONS AND FUTURE WORKS

A novel methodology that balances modelling accuracy and solution methodology complexity is presented. Satisfaction of arrival time constraints and operation precedence are effectively managed. Simulated testing results demonstrate that the method can be substantially better than those used today, and near optimal schedules are generated for problems of practical size. The handling of unpredictable machine breakdowns is also an important issue, this falls directly into the current framework. These strategies allow to observe performance results during simulation and, automatically terminate a simulation when accurate results are obtained.

VII. ACKNOWLEDGMENTS

This work has been supported in part by the Programa de Apoyo a Proyectos de Investigación de la Junta de Castilla

y León, (grant VA057/01) and University of Michigan NSF Engineering Research Center for Reconfigurable Manufacturing Systems (NSF grant EEC-9529125)

REFERENCES

- [1] M.C. Caramanis and M.A. Osman, "Modeling Load-Dependent Lead Time Constraints for Efficient supply chain coordination: Convexity Issues", *Proceedings of the IEEE Conference on Decision and Control*, Phoenix, Arizona, 1999.
- [2] K. Fukunaga, "Introduction to statistical pattern", *Academic Press*, Boston, USA, 1990.
- [3] Y. Koren and U. Heisel and F. Jovane and T. Moriwaki, G. Pritschow, G. Ulsoy and H. Van Brussel, "Reconfigurable manufacturing systems", In: *Annals of the CIRP*, Vol. 48/2, pp. 527-540, 1999.
- [4] J.M. Mulvey and A. Ruszczyński, "A new scenario decomposition method for large-scale stochastic optimization", *IEEE Trans. Robot. Automat.* **43**(3), 477-490, 1995.
- [5] R.S. Sutton and G. Barto, "Reinforcement Learning: An Introduction", *The MIT Press*, Cambridge, MA, 2002.
- [6] J. Trujillo and Z.J. Pasek, "On-line des control behavior verification via virtual supervisor", In: *IEEE SMC Conference*, Washington DC, 2003.
- [7] J. Trujillo and Z.J. Pasek and E. Baeyens, "On-line tracking of manufacturing system performance via virtual supervisor", In: *2nd CIRP International Conference on Reconfigurable Manufacturing*, Ann Arbor, MI, 2003.
- [8] R.O. Duda and P.E. Hart and D.G. Stork, "Pattern Classification", Wiley and Sons, Canada, 2001.
- [9] S.Theodoris and K. Koutroumbas, "Pattern recognition", *Elsavier Science*, San Diego CA, USA, 2003.
- [10] J. Trujillo and Z.J. Pasek and E. Baeyens, "On-line job-shop scheduling of a manufacturing system based on a virtual supervisor concept", In: *16th IFAC World Congress*, Prague, 2005.
- [11] J. Trujillo, "Finite position machine for Logic Control: Pattern composition in Reconfigurable manufacturing system", In: *Doctoral Thesis, Universidad de Valladolid*, Spain, 2004.

Fig. 3. Sequences patterns for workcell schedule

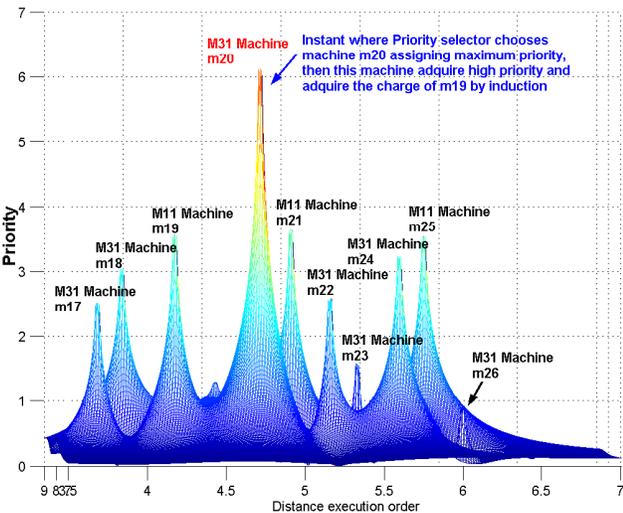


Fig. 4. The selector of priority select machine m20 with maximum priority