ON-LINE JOB-SHOP SCHEDULING OF A MANUFACTURING SYSTEM BASED ON A VIRTUAL SUPERVISOR CONCEPT [1]

Jesús Trujillo * **Zbigniew J. Pasek** ** **Enrique Baeyens** *

* *Departamento de Ingeniería de Sistemas y Automática, Universidad de Valladolid. Paseo del Cauce, s/n. Valladolid. SPAIN E-47011*
** *NSF Engineering Research Center for Reconfigurable Machining Systems, University of Michigan, Ann Arbor, MI, USA*

Abstract: 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 estimates of processing times. This paper presents an approach for job-shop scheduling with uncertain arrival times. The approach exploits Virtual Supervisor (VS) concept, which provides 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 obtain a dual solution for on-line implementation.

Keywords: Manufacturing systems, factory automation, supervisory control, logic controllers, optimization.

1. INTRODUCTION

In reconfigurable manufacturing systems (RMS), scheduling decisions and exception handling policies 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.) (Trujillo *et al.*, 2003; Caramanis and Osman, 1999).

2. PROBLEM FORMULATION

The generic job-shop problem is extremely complex (Caramanis and Osman, 1999). A complete solution algorithm for solving it does not exist. The problem consists of 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

[1] 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)

machines is denoted by η_{nrs}. Part j $(1 \leq j \leq J)$ has arrival time Γ_j, due date D_j, and priority (weight) W_j. In reconfigurable manufacturing systems some machines can change their configuration, allowing redundant production lines (Koren *et al.*, 1999). 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 for a specified processing time t_{jir} and the operation may start only after its immediate preceding operation 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. Such parameters are modeled 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 1 for easy reference.

Table 1. 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

(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.,

$$\Gamma_j \leq B_{j1},\ j = 1, \cdots, J \tag{1}$$

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},

$$k_{ji} + O_{ji} \leq B_{j,i+1},$$
$$j = 1, \cdots, J,\ i = 1, \cdots, I_{j-1} \tag{2}$$

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.*,

$$k_{ji} = B_{ji} + T_{ji} + t_{jir} \leq B_{j,i+1},$$
$$j = 1, \cdots, J,\ i = 1, \cdots, I_j \tag{3}$$

(4) *Replacement machine requirements:* For any r-type can be substituted by an s-type machine, the completion time of part k_{ji} plus deadtime O_{ji} is less that beginning time B_{j1} plus arrival time Γ_{ji}. Substitution is only carried out when priority is maximal.

$$\text{If } W_{ji} \geq W_{\max},\ \text{and}$$
$$k_{ji} + O_{ji} \leq B_{j,i+1} + \Gamma_{ji},$$
$$j = 1, \cdots, J,\ i = 1, \cdots, I_{j-1} \tag{4}$$

(5) *Machine capacity constraints:* 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),

$$\sum_{ji} \theta_{jinrs} \leq \eta_{nrs},\ n = 0, \cdots, N-1,$$
$$r \in R \tag{5}$$

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 constraints (4) for all possible instances of random events is very difficult because of complexity. The feasible model is a schedule satisfying (1)-(6)

$$E\left[\sum_{ji} \theta_{jinrs}\right] \leq \eta_{nrs} \leq L_{np},$$
$$\text{on redundant line case,}$$
$$n = 0, \cdots, N-1,\ r \in R, s \in S \tag{6}$$

(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

$$\min_{\{B_{ji},r_{ji}\}} I,$$

$$\text{where } I = E\left[\sum_{j=1}^{J}(W_jT_j^2 + \varpi_i E_i^2)\right] \quad (7)$$

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).

3. SOLUTION APPROACH

3.1 Gradient Projection Method

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

$$\min_{\{B_{ji},r_{ji}\}} \mathcal{L}, \text{ where}$$

$$\mathcal{L} = E\left[\sum_j(W_jT_j^2 + \varpi_j E_j^2)\right] +$$

$$\sum_{nrs}\pi_{nrs}\left\{E\left[\sum_{ji}\theta_{jinrs}\right] - \theta_{nrs}\right\} \quad (8)$$

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

$$\min_{\{B_{ji},r_{ji}\}} \mathcal{L}_j, \text{ where}$$

$$\mathcal{L}_j = E\left[W_jT_j^2 + \varpi_j E_j^2 + \sum_{i=1}^{I_j}\sum_{n=B_{ji}}^{k_{ji}}\pi_{nrs}\right],$$

$$j = 1, \cdots, J \quad (9)$$

subject to (1)-(6). The high level dual problem is then obtained as,

$$\max_{\{\pi_{nrs}\}} \delta, \text{ where}$$

$$\delta = \sum_j \mathcal{L}_j - \sum_{nrs}\pi_{nrs}\eta_{nrs} \quad (10)$$

3.2 Temporal-Difference Method (TD)

TD method can learn directly from patterns (reference set structures) without a model of the environment (Sutton and Barto, 2002). 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 effectively reasons it has been parametrically combined with conventional TD methods. The algorithm starts at the last stage having the following terminal cost:

$$\varsigma_{ji}(B_{ji}, r_{ji}, s_{ji}) =$$

$$W_jT_j^2 + \sum_{n=B_{jI_j}}^{k_{jI_j}}\pi_{nr_{jI_j}s_{jI_j}} \quad (11)$$

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

$$\varsigma_{ji}(B_{ji}, r_{ji}, s_{ji}) =$$

$$\min_{\{B_{j,i+1},r_{j,i+1},s_{j,i+1}\}} \varpi_j E_j^2 A_{ji} +$$

$$\sum_{n=B_jI_j}^{k_jI_j}\pi_{nr_{jI_j}s_{jI_j}} +$$

$$\varsigma_{ji+1}(B_{j,i+1}, r_{j,i+1}, s_{j,i+1}) \quad (12)$$

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,

$$\varsigma_{ji}(B_{ji}, r_{ji}, s_{ji}) =$$

$$E\left[W_jT_j^2 + \sum_{n=B_{jI_j}}^{k_{jI_j}}\pi_{nr_{jI_j}s_{jI_j}}\right] \quad (13)$$

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.

$$\varsigma_{ji}(B_{ji}, r_{ji}, s_{ji}) = \\ E\left[\varpi_j E_j^2 A_{ji} + \sum_{n=B_j I_j}^{k_j I_j} \pi_{n r_{jI_j} s_{jI_j}} + \varsigma^*_{ji+1}\right] \tag{14}$$

where

$$\varsigma^*_{ji+1} = \\ \min_{\{B_{j,i+1}, r_{j,i+1}, s_{j,i+1}\}} \varsigma_{ji+1}(B_{j,i+1}, r_{j,i+1}, s_{j,i+1}) \tag{15}$$

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

3.3 The Dual Problem

The dual cost function in (10) is concave, piecewise linear, and consists of many phases (Mulvey and Ruszczynsky, 1995). Each phase corresponds to a possible scheduling policy of the problem. The number of possible scheduling policies strongly 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 repeats until some stopping criteria is 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. The block diagram of Fig. 1 is a pictorial representation of the algorithm.

4. SYSTEM OPERATION MONITORING VIA VIRTUAL SUPERVISOR

4.1 Monitoring methods

Virtual Supervisor through setup pattern (Trujillo and Pasek, 2003) and inductive Maxwel method (Trujillo *et al.*, 2003) are 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.

4.2 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 (Trujillo *et al.*, 2003). The potential induced by each machine depends on these 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.

$$R_{ec} = \frac{\sum_k \sum_j v_{kj}}{In}((1+\varsigma)(1+\zeta)(1+x))e^{-(x+1)^2} \\ k = 1, 2, \cdots, n,\ j = 1, 2, \cdots, 2n \tag{16}$$

The coefficient ς reflects the path criticity and its value depends on priority level. The inductor is a manufacturing piece affected by other coefficients:

$$v_0 = \frac{\sum_k \sum_j v_{kj}}{n}(1+W_j)(1+\varpi_j)(1+s_j)e^{-(x+1)} \\ k = 1, 2, \cdots, n\ j = 1, 2, \cdots, 2n \tag{17}$$

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.

4.3 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. $v_{k1} >> v_{k2} >> \cdots >> v_{kn}$, $\{k1, \cdots, kn\} \subseteq [\sum_n E_n, \sum_n X_n, \sum_n V_n]$ where (E_n, X_n, V_n are pattern events, states, and times respectively) (Trujillo *et al.*, 2003; Trujillo and Pasek, 2003),

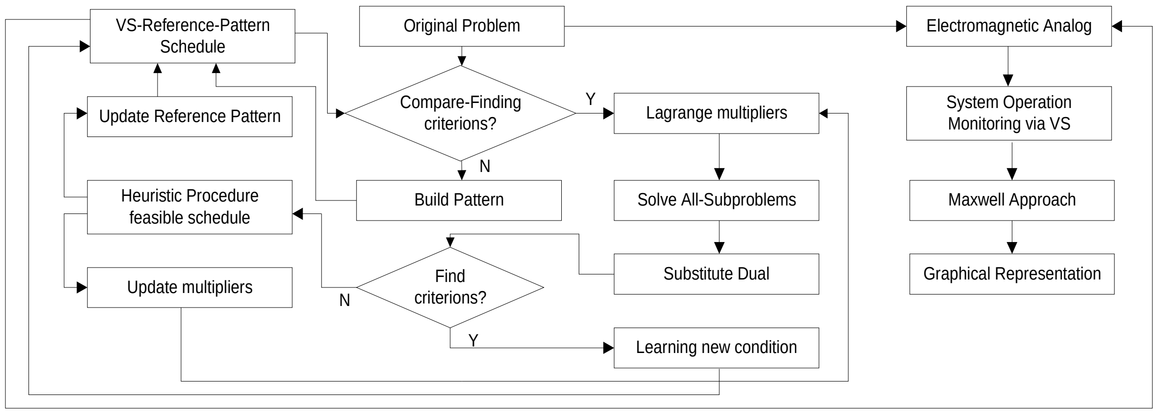

Fig. 1. Overall system for Job-Shop Scheduling via VS

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.

5. EXAMPLE. INDUCTIVE METHOD

The following example shows how the proposed methodology can be applied, how it allows to detect exceptions and to establish handling procedures. The Virtual Supervisor can recognize an exception, and propose a new job-shop scheduling by getting all necessary information from predictive state space, where such solution can be tested virtually. The routing and operation processing times for a six-job, job-shop scheduling problem are shown Gantt diagram in Fig. 2 depicted the total time for this schedule is 60 time units.

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.

5.1 Interpretation of results

The Virtual Supervisor can build in advance the estimated situation in a virtual space, where it is recognized by induction how each machine is working, and preview the future situation for an eventual 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, *i.e.* $M2$, $M5$, $M9$ that become overloaded. The priority for each process, part, and what machine could absorb the overload generated by $M3$ in a new job-shop scheduling is obtained from pattern analysis and TD algorithm. Thus Fig. 3 shows the evolution to balance the line for the complete system. The VS obtained a real-time predictive state space so that the controller can get the information required to perform an exception decision.

6. CONCLUSIONS

A novel methodology that balances modeling 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.

REFERENCES

Caramanis, M.C. and M.A. Osman (1999). Modeling load-dependent lead time constraints for efficient supply chain coordination: Convexity

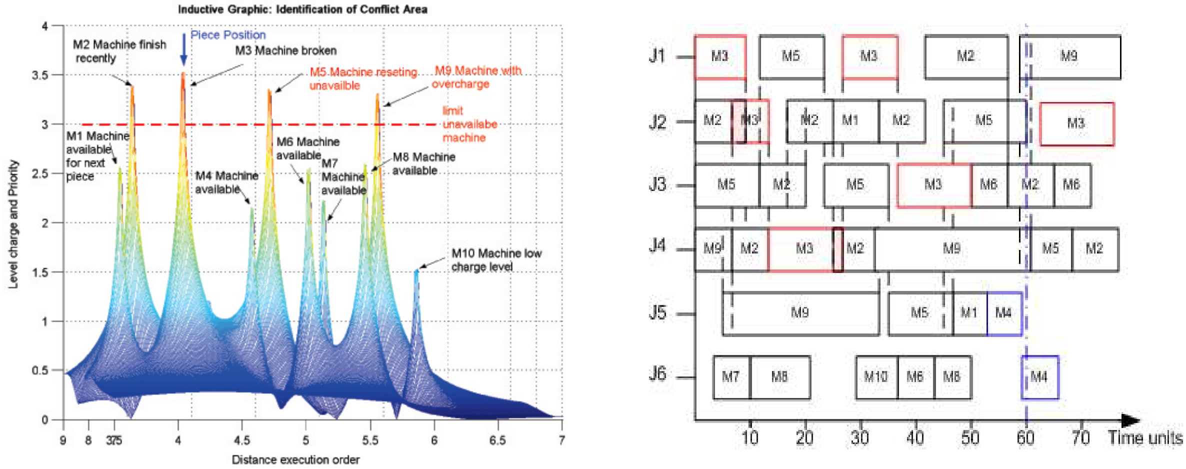

Fig. 2. Area conflict priority ordered by induction. $M1$–$M10$ from left to right consecutively. The Gantt chart for this scenario

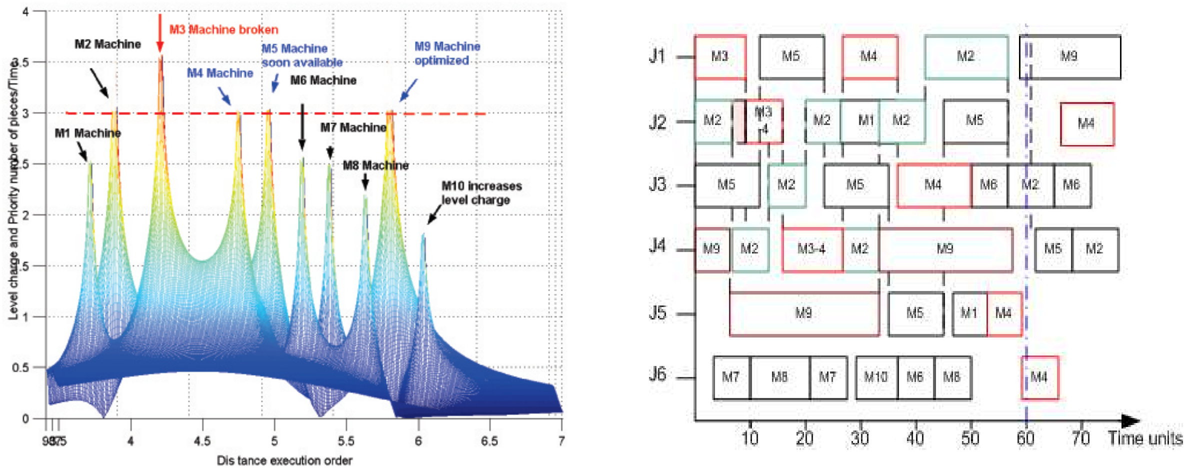

Fig. 3. Machine line ordered in balanced line. The Gantt chart for a feasible solution to a job-shop scheduling

issues. In: *Proceedings of the IEEE Conference on Decision and Control*. Phoenix, Arizona.

Koren, Y., U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy and H. Van Brussel (1999). Reconfigurable manufacturing systems. In: *Annals of the CIRP*. Vol. 48/2. pp. 527–540.

Mulvey, J.M. and A. Ruszczynsky (1995). A new scenario decomposition method for large-scale stochastic optimization. *IEEE Trans. Robot. Automat.* **43**(3), 477–490.

Sutton, R.S. and G. Barto (2002). *Reinforcement Learning: An Introduction*. The MIT Press. Cambridge, MA.

Trujillo, J. and Z.J. Pasek (2003). On-line des control behavior verification via virtual supervisor. In: *IEEE SMC Conference*. Washington DC.

Trujillo, J., Z.J. Pasek and E. Baeyens (2003). On-line tracking of manufacturing system performance via virtual supervisor. In: *2nd CIRP International Conference on Reconfigurable Manufacturing*. Ann Arbor, MI.