Proceedings of DETC'98
1998 ASME Design Engineering Technical Conferences
September 13-16, 1998, Atlanta, Georgia, USA

DETC98/CIE-5691

ROBUSTNESS OPTIMIZATION OF FMS UNDER PRODUCTION PLAN VARIATIONS: PRELIMINARY RESULTS

Kazuhiro Saitou*
Assistant Professor
Department of Mechanical Engineering
and Applied Mechanics
University of Michigan
Ann Arbor, Michigan 48109-2125
Email: kazu@engin.umich.edu

Helge Qvam†
Graduate Student
Department of Mechanical Engineering
Delft University of Technology
Delft, The Netherlands
email: hqvam@hotmail.com

ABSTRACT

A method for robustness optimization of flexible manufacturing systems is presented which undergoes forecasted product plan variations. A configuration of an FMS is modeled by a colored Petri net and the associated transition firing sequence. The robustness optimization of the colored Petri net model is formulated as an multi-objective optimization problem which minimizes production costs under multiple production plans, and reconfiguration cost due to production plan changes. As a first attempt, machines with limited flexibility are considered, and a genetic algorithm, coupled with a simple FIFO dispatching rule, is used to simultaneously find an semi-optimal resource allocation and event-driven schedule of a colored Petri net. The resulting Petri nets are then compared with the Petri nets optimized for a particular production plan in order to address the effectiveness of the robustness optimization for simultaneous production of products with different similarities. The simulation results suggest that the robustness optimization should be considered when the products are moderately different in their manufacturing processes.

*Corresponding author
†Formerly a visiting student, Department of Mechanical Engineering and Applied Mechanics, University of Michigan.

INTRODUCTION

Flexible manufacturing systems (FMS's) are a class of manufacturing system which can be rapidly configured to produce multiple types of products. Recent increase in the use of FMS's has been driven by the need of agile manufacturing systems which can quickly adopt changes in production plans due to market demand fluctuation. While the increased flexibility of an FMS provides greater productivity under various production scenario, it imposes increased complexity in allocation of given resources to different processes required in making each product, and the scheduling of the sequence of activities to accomplish the best production efficiency (Lee, 1994). In order to quickly adapt fluctuating market demand, the resource allocation and scheduling, or configuration in short, of an FMS should not simply be optimized for the current production plan. Rather, it should ideally be *optimized for robustness* against the variation in production plans, so that the system can deal with the variation with minimal reconfiguration (*i.e.* reallocation and rescheduling) while achieving consistently efficient production under all production plans of interest. For this, a reliable forecast on the future change in production plan must be provided, which may or may not be available at a given time.

Assuming such forecasts are available, let us consider the scenario where the FMS simultaneously produces two kinds of products A and B, and the total number of production (sum of the numbers of A's and B's to be produced) per unit time (*eg.* a day) is kept constant with production plan variation (*i.e.* only a

fraction of the two products varies). When A and B are very similar[1], then, it is conjectured that one would not need to consider robustness optimization since the configuration optimized for the current production plan is robust enough such that little system reconfigurations are necessary to deal with production plan change (imagine the extreme of this case where A and B are identical). On the other hand, when products under simultaneous production are different (but not too different to impair the justification for simultaneous production), slight change in the production plan will heavily impact production efficiency, hence necessitating the system reconfiguration in order to achieve efficient production under the new production plan.

The above conjecture motivated us to develop a methodology for robustness optimization of FMS configuration which undergoes given product plan variations, and to study the effectiveness of the methodology for simultaneous production of products with different similarities. A configuration of an FMS is modeled by a colored Petri net and the associated transition firing sequence. The robustness optimization of the colored Petri net model is formulated as an multi-objective optimization problem which minimizes production costs under multiple production plans, and reconfiguration cost due to production plan changes. As an initial attempt, machines with limited flexibility are considered, and a genetic algorithm, coupled with a simple FIFO dispatching rule, is used to simultaneously find an semi-optimal resource (machine) allocation and event-driven schedule on a colored Petri net. The resulting Petri nets are then compared with the Petri nets optimized for a particular production plan in order to validate the above conjecture.

RELATED WORK

Petri nets (Petri, 1962) have been widely used for analysis and simulation of FMS due to their capability of modeling concurrency, synchronization and sequencing in discrete-event systems (Dubois, 1983; Narahari, 1985). Among the most recent is the work by Dhumal, Dhawan, Kona and Soni (Dhumal, 1996), where a Petri net model of a flexible forging cell was used to analyze the production performances under different production scenarios.

In addition to such use as an analysis tool, Petri net models are often used for FMS scheduling problems. Given a job specification (the amount of production and the sequence of operations needed for each job), and the corresponding resource allocation (the type and number of machines for each operation, and the processing time), one can construct a Petri net model of an FMS, where event-driven operation schedules of the modeled FMS are represented as the transition firing sequences of the Petri net. Due to the NP-completeness of the underlying job-shop scheduling problem (JSSP) (Garey, 1979), an optimal schedule is often found via heuristic search algorithms such as beam search (Shih, 1991), A* algorithm (Lee, 1994) and genetic algorithms (Chiu, 1997), coupled with discrete-event simulation of the operation of the Petri net model.

In general, the quality of the optimal schedule is influenced the quality of resource allocation (*i.e.* the topology of the Petri net model) for a given job specifications. This motivates the simultaneous optimization of resource allocation and scheduling, a generalization of JSSP known as generalized resource-constrained project scheduling problems (GRCPSP), which is also NP-complete (Garey, 1979). GRCPSP is typically formulated as mathematical programming problems and solved by heuristic search algorithms (Sprecher, 1994). The solution provides an optimal allocation of a given resources and time-driven operation schedules. Although event-driven schedules are often preferred for FMS scheduling due to their robustness (Lee, 1994), discrete-event based models such as Petri nets are rarely used for GRCPSP due to the computational time for the model simulation.

In the above work, the search is directed towards the discovery of the schedule (and the resource allocation in the case of GRCPSP) optimized for a *fixed* production plan, which could potentially be sensitive to a small perturbation in the current production plan. In continuous mathematical programming, this issue is addressed as sensitivity analyses, where the sensitivity of the optimum to small parameter perturbation is computed, in most cases, in terms of Lagrange multipliers. Several method has been proposed to find an optimal (or suboptimal) solution of nonlinear programming problems which is less sensitive to parameter perturbations (d'Entremont, 1988; Parkinson, 1990; Sundaresan, 1993). Since these methods are essentially an application of Taguchi's robust parameter design (Taguchi, 1978; Taguchi, 1987) to nonlinear programming, they are designed for continuous optimization problems, and hence do not directly apply to problems involving discrete design parameters, such as the FMS scheduling problems using Petri nets discussed above.

PROBLEM FORMULATION

Colored Petri net model of manufacturing systems

Colored Petri nets (Alla, 1985; David, 1992) are an extension of ordinary Petri nets where a place can contain multiple tokens distinguished by a "color" associated with each token. This extension allows colored Petri nets to model manufacturing systems capable of simultaneous production of multiple products in a graphically elegant manner by associating types of products with colors of tokens. As an ordinary Petri net, a colored Petri net is a directed graph consisting of two types of node, *places* and *transitions*. Two nodes are connected by an directed edge which connects either a place to a transition or a transition to a place (see Figures 1–3).

Formally, a colored Petri net R is a six-tuple:

[1] In which sense is left undefined here. This issue will be revisited in the discussion section.

$$R = \langle P, T, Pre, Post, M_0, C \rangle \quad (1)$$

where P is a set of places, T is a set of transitions and C is a set of colors. *Pre* and *Post* are functions of the type $P \times T \times C \mapsto \mathbf{Z}^{|C|}$ and $M_0 : P \mapsto \mathbf{Z}^{|C|}$ is the initial marking, where $\mathbf{Z}$ is a set of integers. A place $p \in P$ is graphically represented by a circle, and a transition $t \in T$ is represented by a bar. A place can contain one or more tokens (with possibly different colors). The number and colors of tokens at a place $p \in P$ is called *marking* of the place denoted as $M(p)$, where $M : P \mapsto \mathbf{Z}^{|C|}$, and represented graphically as colored dots in a circle[2]. Let places p, q and a transition t are connected by edges (p,t) and (t,q). The place p is called an *input place* of the transition t, and the place q is called an *output place* of the transition t. Marking of places change according to the following rules:

1. For each input place p of a transition t, if $M(p) \geq Pre(p,t,c)$ for a color, t is called *enabled* with respect to the color c.
2. If a transition t is enabled with respect to a color c, it can *fire*.
3. If a transition t enabled with respect to a color c fires, $M(p)$ changes to $M(p) - Pre(p,t,c)$, and for each output place q of t, $M(q)$ changes to $M(q) + Post(q,t,c)$.

In addition, capacities to places are often imposed in FMS modeling, represented by a capacity function $Cap : P \mapsto \mathbf{Z}_+$ and marking change occurs only if the total number of tokens of the resulting marking does not exceed the capacity of the place. A sequence of marking changes in all places of a colored Petri net is called *evolution of marking*. The evolution of marking in a colored Petri net from the initial marking represents the sequence of event occurrences in the modeled discrete-event system.

Figures 1–3 illustrate the evolution of marking in a simple colored Petri net which models a production facility consisting of one buffer p_1 and two machines p_2 and p_3. The production facility is to produce two types of products $< a >$ and $< b >$ which both need just one operation to finish. The machine p_2 is capable of performing this operation on both products $< a >$ and $< b >$, while the machine p_3 can only perform the operation on product $< b >$. In this colored Petri net, $P = \{p_1, p_2, p_3\}$, $T = \{t_1, t_2, t_3, t_4\}$, $C = \{< a >, < b >\}$, $M_0(p_1) = (1,2)$, and $M(p_2) = M(p_3) = (0,0)$. It is assumed that $Cap(p_1)$ is infinity (unlimited capacity buffer), and $Cap(p_2) = Cap(p_3) = 1$ (machines can process one product at a time). The functions *Pre* and *Post* are defined in terms of "shorthand" function $f, g : C \mapsto C$ such that

$$Pre(p1, t1, < a >) = f(< a >) =< a >= (1,0)$$

[2]In most literature, however, a token is represented as $< c >$, where c is a symbol representing the color of the token, as they are not normally printed in color.

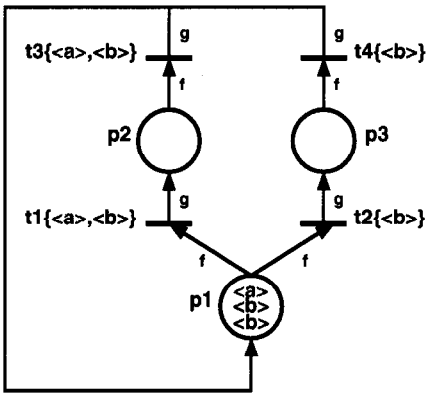

Figure 1. colored Petri net which models a production facility with one buffer p_1 and two machines p_2 and p_3, with initial marking.

$$Pre(p1, t1, < b >) = f(< b >) =< b >= (0,1)$$
$$Pre(p1, t2, < b >) = f(< b >) =< b >= (0,1)$$
$$Pre(p2, t3, < a >) = f(< a >) =< a >= (1,0)$$
$$Pre(p2, t3, < b >) = f(< b >) =< b >= (0,1)$$
$$Pre(p3, t4, < b >) = f(< b >) =< b >= (0,1)$$
$$Post(p1, t3, < a >) = g(< a >) =< a >= (1,0)$$
$$Post(p1, t3, < b >) = g(< b >) =< b >= (0,1)$$
$$Post(p1, t4, < b >) = g(< b >) =< b >= (0,1)$$
$$Post(p2, t1, < a >) = g(< a >) =< a >= (1,0)$$
$$Post(p2, t1, < b >) = g(< b >) =< b >= (0,1)$$
$$Post(p3, t2, < b >) = g(< b >) =< b >= (0,1)$$

For other points (p,t,c) not defined above, $Pre(p,t,c)$ and $Post(p,t,c)$ are undefined. The colors listed next to each transition is *firing colors* of the transition, with respect of which the transition can be enabled if appears in the input place.

At the start of the production cycle, the machines are not working and the unfinished products, one $< a >$ and two $< b >$'s, are located in the buffer p_1, which is given as the initial marking $M_0(p_1) = (1,2)$. Since $M_0(p_1) > Pre(p_1, t_1, < a >), Pre(p_1, t_1, < b >)$ and $M_0 > Pre(p1, t2, < b >)$, transition t_1 is enabled with respect to both $< a >$ and $< b >$, and t_2 is enabled with respect to $< b >$. Let us assume t_2 fires at the next step. Then, $M(p_1)$ changes from $(1,2)$ to $(1,1)$ as $Pre(p_1, t_2, < b >) =< b >= (0,1)$, and hence one of two token $< b >$'s is removed from p_1. Also $M(p_3)$ changes from $(0,0)$ to $(0,1)$ as $Post(p_3, t_2, < b >) =< b >$, and hence the token $< b >$ removed form p_1 appears in p_3. At this point, transitions t_1, t_2 and t_4 are enabled. Let us assume t_1 fires at the next step. The transition t_1 has the choice of firing either $< a >$ or $< b >$. Suppose t_1 fires $< a >$. Proceeding similar to the previous firing, the token $< a >$ is removed from p_1 and appears in p_2 (Figure 2). This represents

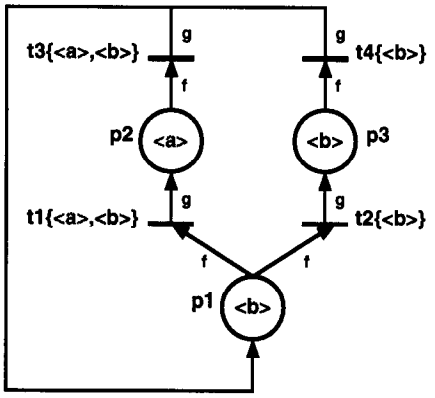

Figure 2. state of the colored Petri net after firing of t_2 with $< b >$ and t_1 with $< a >$.

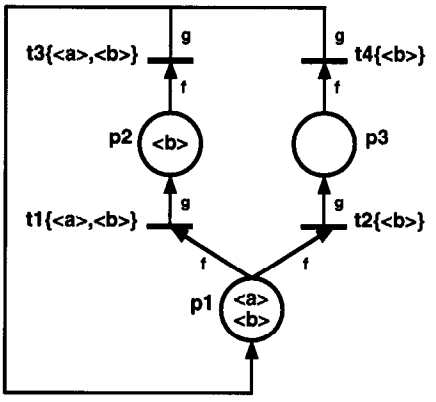

Figure 3. state of the colored Petri net after firing of t_2 with $< b >$, t_1 with $< a >$, t_3 with $< a >$, t_1 with $< b >$ and t_4 with $< b >$.

the system state of the machine p_2 processing the product $< a >$ and the machine p_3 processing the product $< b >$.

Although all transitions are enabled at this point, only t_3 or t_4 can fire since firing t_1 and t_2 would result in exceeding the capacity of places p_2 and p_3. Let us assume t_3 is fired and this moves $< a >$ in p_2 back to p_1. This means the machine p_2 has now finished processing the product $< a >$ and send it back to the buffer p_1. Then, t_1, t_2, and t_4 are enabled but only t_1 or t_4 can fire. The subsequent firing of t_1 with $< b >$ followed by the firing of t_4 with $< b >$ would move $< b >$ in p_1 to p_2, and $< b >$ in p_3 to p_1 (Figure 3). This represents that the machine p_2 is now processing the product $< b >$ while machine p_3 has finished processing the product $< b >$ and send it back to buffer.

As illustrated above, a sequence of transition firing of a colored Petri net can be interpreted as an even-driven schedule of the modeled manufacturing systems. Therefore, choosing a transition firing sequence in the above example would result in a different evolution of markings, *i.e.* different schedule would yield a different system behavior. In general, topology of a colored Petri net model is determined by a job specifications (the amount of production and the sequence of operations needed for each job), and the corresponding resource allocation (the type and number of machines for each operation, and the processing time). In the above example, the job specification was one $< a >$ and two $< b >$'s per cycle and one operation for both $< a >$ and $< b >$, and the resource allocation was one machine for both $< a >$ and $< b >$, one machine for $< b >$, and one buffer for $< a >$ and $< b >$, all with zero processing time. A different job specifications and resource allocation would yield a colored Petri net model with different topology from the above example.

Robustness optimization of FMS configurations

We consider a scenario where an FMS simultaneously produces multiple types of products which share common manufacturing operations. It is assumed that the production plan of the FMS given in terms of the numbers of each types of the products to be produced during a given period of time. Let n be the number of types of the products. Then, the production plan can be represented as $\rho \in \mathbf{Z}^n$. Suppose the total number of production (sum of the numbers of n product types to be produced) per unit time is kept constant to, say N, and hence the production plan changes are only due to the changes in the fraction of the product types. Let the fraction be α_i, where $0 \leq \alpha_i \leq 1$ for $i = 1, 2, ..., n$ and $\sum_{i=1}^{n} \alpha_i = 1$, or collectively be an n dimensional vector $\mathbf{a}$. Given N, therefore, a production plan can be uniquely specified as a function of the fraction vector $\mathbf{a}$, which we shall call $\rho(\mathbf{a})$.

Let $\rho(\mathbf{a}_0)$ be the current production plan. We assume the forecasts on production plan changes within the timeframe of interest are available as a sequence of m production plans $\rho(\mathbf{a}_1), \rho(\mathbf{a}_2), \ldots, \rho(\mathbf{a}_m)$. Our objective is to optimize the robustness of the current configuration (resource allocation and schedule) of the FMS against the given variation in production plans. Namely, we want to minimize reconfiguration while achieving consistently efficient production under all of m production plans foreclosed. Let $\mathbf{x}_0$ be the current configuration of the FMS, and $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$ be the future configurations corresponding the m production plans forecasted. Then, the problem can be formulated as the simultaneous minimization of the following $2m+1$ functions:

$$\textit{production-cost}(\mathbf{x}_j, \rho(\mathbf{a}_j)) \quad j = 0, 1, \ldots, m \tag{2}$$

$$\textit{reconfig-cost}(\mathbf{x}_j, \mathbf{x}_{j+1}) \quad j = 0, 1, \ldots, m-1 \tag{3}$$

where $\textit{production-cost}(\mathbf{x}_j, \rho(\mathbf{a}_j))$ is the production cost of the FMS with the configuration $\mathbf{x_j}$ under the production plan $\rho(\mathbf{a}_j)$, and $\textit{reconfig-cost}(\mathbf{x}_j m \mathbf{x}_{j+1})$ is the reconfiguration cost from the configuration $\mathbf{x}_j$ to the configuration $\mathbf{x}_{j+1}$.

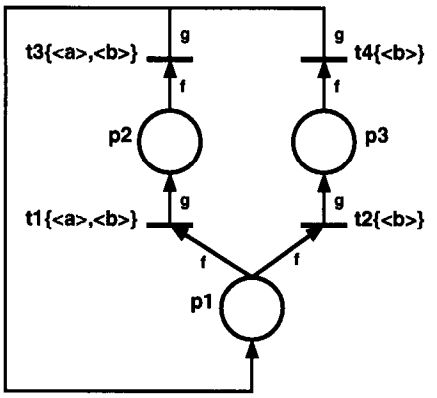

Figure 4. colored Petri net before reconfiguration.

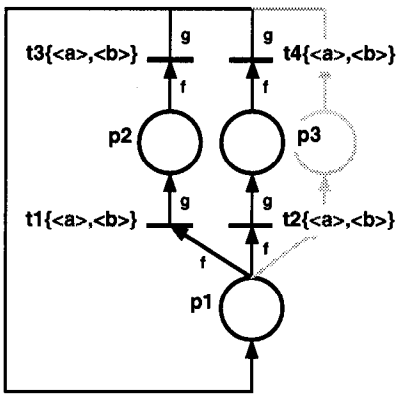

Figure 5. colored Petri net after reconfiguration.

Given the configuration $\mathbf{x}_j$ and the production plan $\rho(\mathbf{a}_j)$, the production cost can be evaluated using discrete-event simulations based on a colored Petri net of an FMS. The production cost is estimated as the combination of cost of the production time and facility:

$$\textit{production-cost}(\mathbf{x}_j, \rho(\mathbf{a}_j)) = r(\mathbf{x}_j, \rho(\mathbf{a}_j)) \times \left(\sum_k c_k M_k(\mathbf{x_j}) \right) \tag{4}$$

where $r(\mathbf{x}_j, \rho(\mathbf{a}_j))$ is the number of rounds to accomplish the production plan $\rho(\mathbf{a}_j))$, and c_k and $M_k(\mathbf{x}_j)$ are the running cost and number of the machine type k, respectively. Reconfiguration cost from one configuration to the other is estimated as the number of rerouting required to accomplish the new configuration, *i.e.* the number of routings (connection between two places) in the colored Petri net which need to be changed due to the change in the resource allocation. The reconfiguration cost associated with the change in schedules is not considered here since dynamic scheduling with dispatching rules is used as discussed in the following section[3]. Namely:

$$\begin{aligned} &\textit{reconfig-cost}(\mathbf{x}_i, \mathbf{x}_j) \\ &\quad = \text{number of routings differences from } \mathbf{x}_i \text{ to } \mathbf{x}_j \end{aligned} \tag{5}$$

For instance, the change from the colored Petri nets in Figure 4 to the one in Figure 5 is 2 since two routings between p_1 and p_3 must change since the machine type of p3 changed from a specialized machine only capable of producing product $< b >$, to a flexible machine capable of producing product $< a >$ and product $< b >$.

[3]Incorporation of differences in schedule should be a part of future work.

In the following sections, we discuss simple case studies for $n = 2$ and $m = 1$, namely when two types of products A and B, are to be produced, and only one forecast on the production plan is available. We have, therefore, the fraction vector with dimension 2, and 3 functions to be minimized. In this case, the fraction vector can be expressed as using one parameter α as $(\alpha N, (1-\alpha)N)$, where $0 \leq \alpha \leq 1$.

Optimization using a genetic algorithm and dispatching rules

The robustness optimization of FMS configurations discussed in the previous section requires simultaneous optimization of resource allocation and scheduling. Due to the high complexity of the underlying optimization problem (GRCPSP), a hybrid scheme is adopted where a genetic algorithm is used for resource allocation, and dispatching rules are used for dynamic scheduling of the colored Petri net models of a FMS. Although the resulting configuration may not be guaranteed to be optimal, this hybrid scheme allows very fast evaluation of large number of feasible configurations.

Genetic algorithm (GA) is an optimization technique in which points in design space are analogous to organisms subject to a process of natural selection, or "survival of the fittest (Holland, 1975; Goldberg. 1989). GAs model reproduction in population of encoded representation of points (typically strings of bits) in design space – called genetic "chromosomes" – over generations. In a given generation, the quality of a chromosome, *i.e.* a bit string representation of a point in design space, is measured based in a fitness function, and highly-fit chromosomes have higher chances to be selected for reproduction. Two "parent" chromosomes selected for reproduction are mated through genetic crossover, resulting in two offsprings which are likely to inherit good "genes" from their parents. Many generations of such selection and mating will produce a highly-fit population of chromosomes, *i.e.* better designs.

Dispatching rules are local rules which specifies priorities in the dispatching of products to machines while production is in progress. Dispatching rules has been traditionally used for scheduling, due to its simplicity and reliability. A number of dispatching rules such as FIFO (First-In-First-Out), SIO (Shortest Imminent Operation time) and SRPT (Shortest Remaining Processing Time) have been successfully applied to FMS scheduling (Choi, 1988). Although the schedules created by off-line heuristic algorithms often outperform the ones by dispatching rules, they allow very fast and dynamic creation of near-optimal schedules of FMS which do not exhibit very high nonlinearity in routing. Also, the schedules created by dispatching rules tends to be robust against the sudden change in resource allocation (e.g. machine breakdown), since the schedules are dynamically created during the operation, rather than determined off-line. Although in practice a combination of several dispatching rules are used, a simple FIFO rule is selected in the following examples for the purpose of a initial demonstration. Namely, a FIFO rule is used for dynamic scheduling of a colored Petri net whose resource allocation is specified by a parameter, or a "chromosome," of a genetic algorithm.

We assume a limited flexibility of machines in which one type of machines can perform only one type of manufacturing operation, possibly for different product types. This assumption is made to reduce the need of cyclic routing in a colored Petri net, in order to increase a chance that the schedules obtained by a FIFO dispatching rule are near-optimal. Based on the above assumption, each machine for each operation is marked by a number indicating which product type the machine can process, with 0 being the both product types A and B, 1 being only the product type A, 2 being only the product type B, and 3 being no product type, *i.e.* the machine is turned off. This corresponds to two bits in binary number per machine is assigned on a genetic chromosome. In the following examples, the maximum of three machines per machine type are assumed to be available. Therefore, the total number of bits to specify a resource allocation of a colored Petri net is 3*(bits per machine)* $\times$ 3*(machines per machine type)* $\times\ l$, where l is the number of machine type needed to produce all product types. Since $m+1$ configurations must be specified for robustness optimization, the length of chromosomes is $18*l$, in the case of $m=1$.

SIMULATION RESULTS

This section describes simple case studies of the robustness optimization of FMS as described in the previous section applied to the example production scenario with $n=2$ and $m=1$. In other words, two product types A and B are to be produced, and only one forecast on the production plan is available.

Assumptions

It is assumed that both product types require three manufacturing operations with the same sequence, and a product type is characterized by its job list, a list of the processing time needed for each operation. For instance, if the product type require milling, turning, and drilling in this order, the job list $(10,15,5)$ indicates the product type takes 10 unit time for milling, 15 unit time for turning, and 5 unit time for drilling in order to be finished. Since one type of machines can perform only one type of manufacturing operation, the number of machine types l is 3. Therefore, the length of chromosomes for robustness optimization is 36. A production plan is represented by the fraction of product A out of the total number of production N per unit time, which is set to 20 for all examples. In addition, running cost of all machine types are assumed to be the same.

The actual objective function (to be minimized) for robustness optimization is a multiplication of *recofig-cost* with adjustments and *production-cost*:

$$f(\mathbf{x}_0,\mathbf{x}_1) = \{\textit{reconfig-cost}(\mathbf{x}_0,\mathbf{x}_1)/10+1\} \times\{r(\mathbf{x}_0,\rho(\alpha_0))\cdot M(\mathbf{x}_0)+r(\mathbf{x}_1,\rho(\alpha_1))\cdot M(\mathbf{x}_1)\} \quad (6)$$

where $\mathbf{x}_0$ and $\mathbf{x}_1$ are the configurations for the current production plan $\rho(\alpha_0)$ and the forecasted production plan $\rho(\alpha_1)$, respectively. $M(\mathbf{x}_0)$ and $M(\mathbf{x}_1)$ are the total numbers of machines in the configurations $\mathbf{x}_0$ and $\mathbf{x}_1$, respectively. Unless otherwise specified, the current production plans is $(18,2)$, *i.e.* $alpha=0.9$, and the forecasted production plan is $(2,18)$, *i.e.* $\alpha=0.1$. Rather surprisingly, it is observed that *reconfig-cost*$(\mathbf{x}_0,\mathbf{x}_1)$ becomes zero, *i.e.* $\mathbf{x}_0=\mathbf{x}_1$ as a result of robustness optimization in *all* of the following example. In other words, the robustness optimization always resulted in *one* configuration which can efficiently produce A and B under *both* current and forecasted production plan. In order to study the effectiveness of the robustness optimization, in each example this resulting configuration is compared with two configurations: the one optimized *only* for the current production plan, and the one optimized *only* for the forecasted production plan. The comparison is done by plotting the production cost of each configurations under the production plans in the range from $\alpha=0$ (only A produced) to $\alpha=1$ (only B produced). We will refer to this plot as the production cost-alpha plot.

The simulation modules were written in C++ with the optimize being the GALib developed at the MIT CADLAB. The simulations were run on a Sun Ultraspark Workstation, which in some cases took two hours to complete. For all results presented below, the population size is 100, the number of generations is 50, the probability of crossover is 0.7 and the probability of mutation is 0.08.

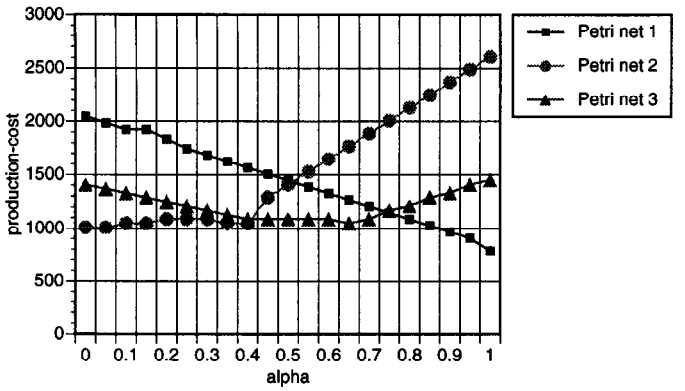

Figure 6. production cost-alpha plot for $A:(5,15,10)$ and $B:(15,15,5)$.

Preliminary examples

There are three preliminary examples where the job lists of A and B are different (Example 1), very different (Example 2), and totally different (Example 3). These examples are studied to investigate the effect of product similarities on the results of the robustness optimization.

Example 1 ¡ The product type A has a job list of $(5,15,10)$ and the product type B has a job list $(15,15,5)$. These two job lists share identical process time of the second operation, and the processing times for the first and the third operation are not too different. Figure 6 shows the production cost-alpha plot of this case. In Figure 6, the colored Petri net optimized for robustness ("dual plan" configuration) was called Petri net 3, and the colored Petri nets optimized only for current production plan, and only for forecasted production plan ("single plan" configurations) are called Petri net 1 and Petri net 2 in Figure 6, respectively. In the range $\alpha \in (0.4, 0.75)$, Petri net 3 gives lower production cost compared to Petri net 1 and Petri net 2. For other values of α, Petri net 1 and Petri net 2 give the better results than Petri net 3. Petri net 1 and Petri net 2, however, have a tendency to reach high levels of production cost as α varies from the original value to which they are optimized, possibly by the creation of production bottleneck. This is observed clearly in the case of Petri net 2 around $\alpha = 0.4$. Overall, Petri net 3, optimized for robustness, is in fact robust in that it exhibits consistently low production cost over the range of α between 0 and 1.

Figures 7, 8 and 9 show Petri net 1, Petri net 2 and Petri net 3, respectively. As seen in these figures, the resulting colored Petri nets are quite different. The differences between these colored Petri nets (*PND*: Petri Net Difference) can be measured same way as *reconfig-cost* defined earlier, which gives $PND(1,2) = 10$, $PND(1,3) = 8$ and $PND(2,3) = 10$.

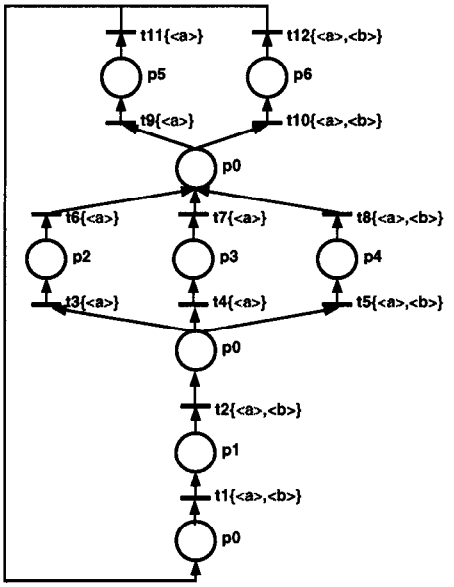

Figure 7. colored Petri net optimized for current production plan (Petri net 1). p_0: buffer (unlimited capacity), p_1: flexible milling machine, p_2, p_3: turning machine for A, p_4: flexible turning machine, p_5: drilling machine for A, p_6: flexible drilling machine.

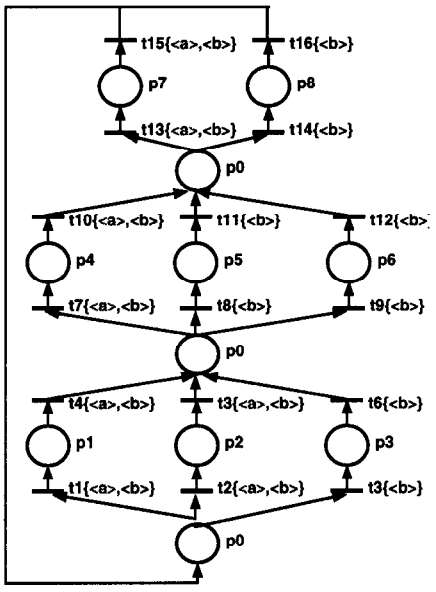

Figure 8. colored Petri net optimized for forecasted production plan (Petri net 2). p_0: buffer (unlimited capacity), p_1, p_2: flexible milling machine, p_3: milling machine for B, p_4: flexible turning machine, p_5, p_6: turning machine for B, p_7: flexible drilling machine, p_8: drilling machine for B.

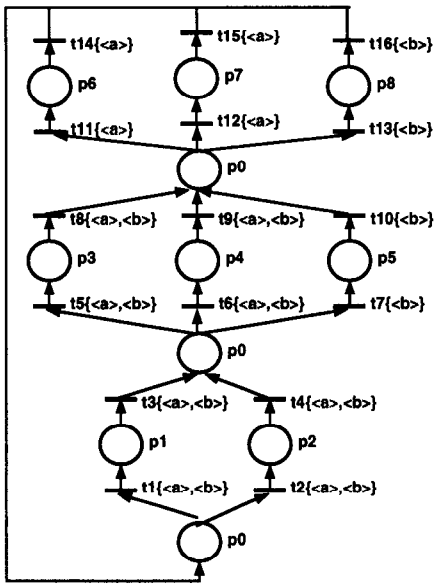

Figure 9. colored Petri net 3 optimized for robustness (Petri net 3). p_0: buffer (unlimited capacity), p_1: flexible milling machine, p_2: milling machine for B, p_3, p_4: flexible turning machine, p_5: turning machine for B, p_6, p_7: drilling machine for A, p_8: drilling machine for B.

Example 2 In this case product A has a job list $(10,5,1)$ and product B has a job list $(1,5,10)$. Similar to Example 1, these job lists share the same process time for the second operation. Also, the total process time of all three operations are identical. However, the fraction of the second process time to the total process time is much smaller than the job lists in Example 1, reducing the "sameness" of the two job lists. Also, the first and the third process time are very different. Compared with Example 1, therefore,.the two products are more different. Although the production cost-alpha plot in Figure 10 shows general trends similar to Figure 6 three curves in Figure 10 are shows more rapid increase in production cost per unit change in α, *i.e.* the curves are *steeper*. In particular, Petri net 3, optimized for robustness, is less robust in this case in that it does not indicate the consistent production cost as shown in the previous example.

The difference in the Petri nets are of less interest in this example because the places with the highest degree of utilization are different for each product and therefore harder to compare: $PND(1,2) = 10$, $PND(1,3) = 8$, $PND(2,3) = 8$.

Example 3 The job list of product A is $(10,0,0)$ and the job list of product B is $(0,5,5)$. These two products are so different that they share no common operation. In this case, the robustness optimization seems to have very few value as shown in the curve of Petri net 3 in Figure 11. Although the production cost of net 3 is consistent, the overall production cost is no better

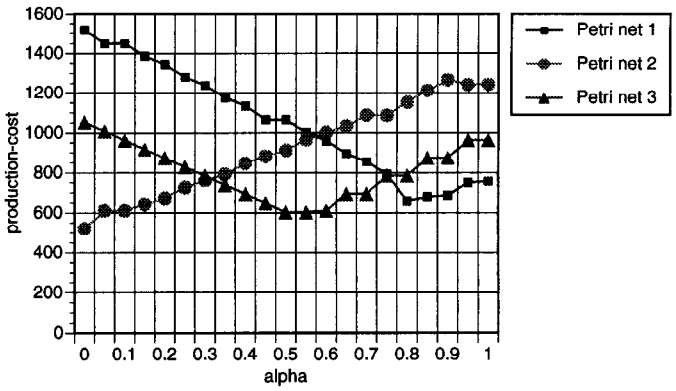

Figure 10. production cost-alpha plot for $A:(10,5,1)$ and $B:(1,5,10)$.

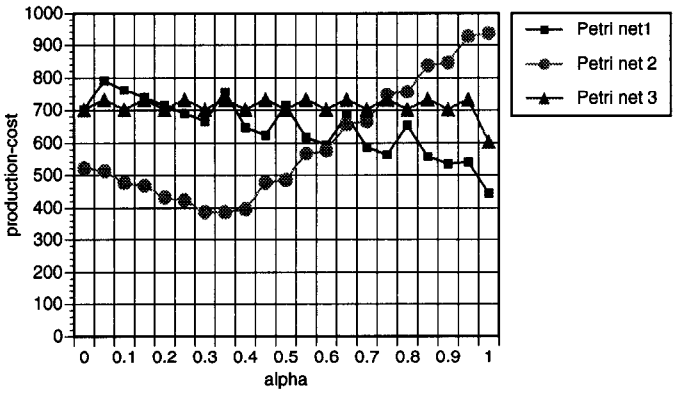

Figure 11. production cost-alpha plot for $A:(10,0,0)$ and $B:(0,5,5)$.

than Petri net 1 and Petri net 2 for almost entire range of α. This implies that the proposed method robustness optimization is less valid in the cases where the products to be produced are totally different.

In this example, the differences in the Pert nets where $PND(1,2) = 12$, $PND(1,3) = 8$, $PND(2,3) = 10$. These numbers clearly indicate Pert nets optimized for products not sharing any operations must be very different. Observing $PND(1,3) = 8$, it is expected that the performance of these nets would be quite different. Figure 11, however, indicates that this is not the case, and adds to the case that the differences in the Petri nets is of less importance to the performance of the system than the difference of the nets around the places with the highest degree of utilization. Around the places with highest degree of utilization $PND(1,2) = 4$, $PND(1,3) = 2$ and $PND(2,3) = 4$ which could help to explain both the relative similarity of Petri net 1 and 3 and their common dissimilarity with Petri net 2.

Forging Cell Examples

The above procedures are applied to more realistic examples of forging cell operation adopted from (Dhumal, 1996). The forging cell consists of shearing machines, furnaces, forging press and material handling systems (buffers and robot arms), which can be configured to produce two product types. The both product types require shearing, heating and forging in this sequence and these operations are done only by shearing machines, furnaces and forging press, respectively. The reconfiguration time of flexible machines (machines that can process both product types) are set to 100, about 50% of the total production time for one product.

The production plans of two product types in Case 4 ($\alpha = 0.3$) and Case 1 ($\alpha = 0.65$) of Configuration II, as appeared in (Dhumal, 1996), are used as the current and forecasted production plan, respectively. The job list of product A is $(12, 140, 16)$ and the job list of B is $(16, 195, 22)$. The both job list have the process time of the second operation nearly ten times larger than the process times of the first and the third processes, which dominates the total process time. Also, all operations of the product B take longer than the ones of the Product A. Therefore, a production facility which can produce B optimally could also produce A fairly efficiently. Such a facility, however, is not necessarily optimal for producing A, in which case there exists a "robust" facility which can produce both types of products with consistent efficiency over the wide range of α. This was clearly shown in the production cost-alpha plot in Figure 12. Petri net 2 (optimizes for $\alpha = 0.3$, *i.e.* 30% A and 70% B), performs worse than Petri net 3 (optimized for both $\alpha = 0.65$ and $\alpha = 0.3$) in the range $\alpha \in (0.5, 1)$. Although outperformed by Petri net 2 around $\alpha = 0.3$, Petri net 3 shows very robust performance over all range of α.

The difference in the Petri nets were $PND(1,2) = PND(1,3) = PND(2,3) = 2$. Although the overall differences in these Petri nets are small, all the differences are situated around the places with the highest utilization, where even a small difference have severe impact on the overall performance.

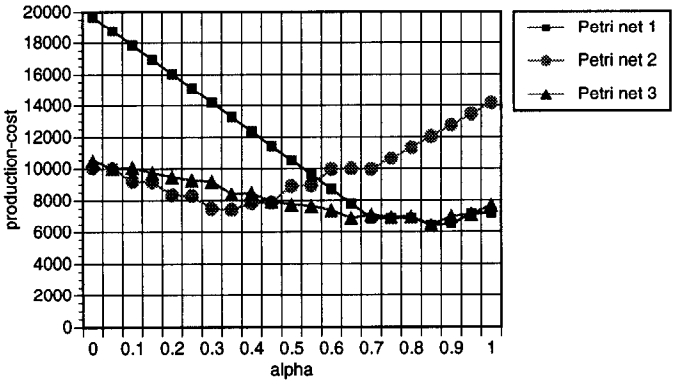

Figure 12. production cost-alpha plot for A : $(12, 140, 16)$ and B : $(16, 195, 22)$. Current and forecasted production plans are $\alpha = 0.65$ and $\alpha = 0.3$, respectively.

DISCUSSION AND FUTURE WORK

This paper presented a method for robustness optimization of flexible manufacturing systems which undergoes forecasted product plan variations. A configuration of an FMS is modeled by a colored Petri net and the associated transition firing sequence. The robustness optimization of the colored Petri net model is then formulated as an multi-objective optimization problem which minimizes production costs under multiple production plans forecasted, and reconfiguration cost due to production plan changes. As a first attempt, machines with limited flexibility are considered, and a genetic algorithm, coupled with a simple FIFO dispatching rule, is used to simultaneously find an semi-optimal resource allocation and event-driven schedule of a colored Petri net. The resulting Petri net is then compared with the Petri nets optimized for a current production plan, and for a forecasted production plan, in order to address the effectiveness of the robustness optimization for simultaneous production of products with different similarities. The simulation results suggest that the robustness optimization should be considered when the products are moderately different in their manufacturing processes.

There are number of assumptions made in this paper which prohibits the over generalization of the above interpretation of the simulation results. They include the assumption of one machine type for one operation, the assumption of identical running costs for all machine in a type, the assumption of the same process sequences for all product types, the assumption of the same process time for all machine types. Relaxation of some or all of these assumptions would results in colored Petri net model with much higher routing nonlinearity, and would require the adoption of more sophisticated dispatching rules, or off-line scheduling algorithm, to guarantee the quality of optimization. Addressing the effect of product similarity on the robustness optimization with such generalization is part of the future work. It is also of great interests to compare the various "product similarity indicator" in terms of their impact on the robustness optimization.

In addition, current formulation of the robustness optimization is based on existence of perfectly reliable forecast of production plans. Although acceptable for this initial attempt, this assumption must be relaxed to incorporate uncertainty in the forecast. One way to do it would be to associate probabilities to each of discrete forecast, and model the transition among the forested production plans as finite Markov process. Incorporation of stochastic nature of forecast, without much increase in computational expenses, would be an important extension.

ACKNOWLEDGMENT

This work was carried out using computational facilities at the Computational Design Laboratory, Department of Mechanical Engineering and Applied Mechanics, the University of Michigan. This source of support is gratefully acknowledged.

REFERENCES

H. Alla, P. Ladet, J. Martinez, and M. Silva-Suarez. Modeling and validation of complex systems by colored petri nets: Application to a flexible manufacturing systems. In *Lecture Notes in Computer Science*, volume 188, pages 1–14. 1985.

Y.-F. Chiu and L.-C. Fu. A GA embedded dynamic search algorithm over a petri net model for an fms scheduling. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 513–518, 1997.

R. Choi and M. Malstrom. Evaluation of traditional work scheduling rules in a flexible manufacturing system with a physical simulator,". *Journal of Manufacturing Systems*, 7(1):27–45, 1988.

R. David and H. Alla. *Perti Net and Grafcet: Tools for Modeling Discrete Event Systems*. Prentice Hall, 1992.

K. d'Entremont and K. Ragsdell. Design for latitude using TOPT. In *ASME Advances in Design Automation*, volume DE-Vol. 14, pages 265–272, 1988.

A. Dhumal, R. Dhawan, A. Kona., and A. Soni. Reconfigurable system analysis for agile manufacturing. In *Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Irvine, California, August 18–22 1996.

D. Dubois and E. Stecke. Using petri nets to represent production processes. In *Proceedings of the 22nd IEEE Conference on Decision and Control*, pages 1062–1067, San Antonio, TX, 1983.

M. R. Garey and D. S. Johnson. *Computer and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

J. H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI, 1975.

D. Lee and F. DiCesare. Scheduling flexible manufacturing systems using petri nets and heuristic search. *IEEE Transaction on Robotics and Automation*, 10:123–132, 1994.

Y. Narahari. and N. Yiswanadham. A petri net approach to the modeling and analysis of flexible manufacturing systems. *Annals of Operations Research*, 3:449–472, 1985.

A. Parkinson, C. Sorensen, J. Free, and B. Canfield. A petri net approach to the modeling and analysis of flexible manufacturing systems. In *Proceedings of the 1990 ASME Design Automation Conference*, volume 2, pages 121–128, Chicago, Illinois, September 1990.

C. Petri. *Kommunikation mit Automaten*. PhD thesis, Universitat Bonn, Bonn, West Germany, 1962.

H. Shih and T. Sekiguchi. A timed petri net and beam search based on-line fms scheduling system with routing flexibility. In *Proceeding of the 1991 IEEE International Conference on Robotics and Automation*, pages 2548–2553, 1991.

A. Sprecher. *Resource-Constrained Project Scheduling*. Springer-Verlag, 1994.

S. Sundaresan, K. Ishii, and D. Houser. A robustness optimization procedure with variations on design variables and constraints. In *ASME Advances in Design Automation*, volume DE-Vol. 65-1, pages 379–386, 1993.

G. Taguchi. Off-line and on-line quality control systems. In *Proceedings of the International Conference on Quality Control*, Tokyo, Japan, 1978.

G. Taguchi. Systems of experimental design. In Don Clausing, editor, *American Supplier Institute*, Deaborn, Michigan, 1987.