

Adaptive job-shop scheduling with routing and sequencing flexibility using expert knowledge and Coloured Petri Nets

Horst Ey, Dirk Sackmann, Martin Mutz
University of Applied Sciences Braunschweig/Wolfenbüttel

Jürgen Sauer
University Oldenburg

Abstract

Petri Nets are known to be efficient for modeling manufacturing systems, because they have a graphical representation and a well-defined semantics allowing formal analysis. Considering conflicts as routing- and sequencing alternatives, we propose a knowledge based algorithm for on-line scheduling, that guides the search for a near optimal schedule in the state space efficiently and limits the state space explosion problem. Taking into account that expert knowledge is formulated mostly in natural language, the inference process is modeled by an approximate reasoning scheme consistent with possibility theory. For refining initial knowledge, a concept is presented that combines reinforcement learning techniques with possibilistic clustering methods. Finally, our approach is validated by a numerical example showing especially, that the use of expert knowledge heuristically guides the search for a near optimal solution of the scheduling problem.

Keywords: scheduling, Coloured Petri Nets, reachability graph, approximate reasoning, reinforcement learning, possibilistic clustering, rule induction

1. INTRODUCTION

In order to match nowadays market requirements, manufacturing systems have to become more flexible and efficient. To achieve these objectives, the systems need not only the introduction of automated and flexible machines, but also flexible scheduling systems. Scheduling in multi-stage manufacturing systems is confronted with problems concerning simultaneous routing- and sequencing decisions of the jobs that have to be processed. In a state space planning approach we decompose the problem into a routing and a sequencing problem. The first problem is selecting a suitable machine out of a set of different machines in each stage of the manufacturing system. Second the sequencing of jobs has to be solved when an appropriate machine has been selected. Formally the scheduling task is a combinatorial optimization problem that requires the selection of an action a_t in each state of the system s_t . The scheduling task consists of determining one path $a_0, \dots, a_t, \dots, a_{T-1}$ in a discrete and deterministic state space that leads to a terminal state s_T where an

appropriate objective function is optimized. In order to reduce the problem of exploding state spaces and for accelerating the search for (sub)optimal paths in the state space, a knowledge based approach is proposed, that guides the search in the state space. Taking into account that available expert knowledge is formulated mostly using linguistic expressions with fuzzy concepts, an approximate reasoning scheme is proposed to guide the action selection procedure in each system state. In the approach a reward r_{s_T} is considered that evaluates the terminal state s_T as the only available information from the environment. One part of the problem is, that for the evaluation of each decision along the sequence of selected actions $a_0, \dots, a_t, \dots, a_{T-1}$ the reward r_{s_T} has to be distributed to each action a_t . Solving this kind of credit assignment problem allows us to evaluate each action of the state space. This data is used finally as input data to derive new expert knowledge empirically. In section 2 we first describe Coloured Petri Nets as a modeling approach that is known to be well suited to model dynamics and to identify decision spaces for the proposed scheduling approach. In the following we show how to analyze the nets corresponding reachability graph partially taking fuzzy expert knowledge into account. Section 3 introduces an off-line learning method to modify the knowledge that is used on-line to derive a schedule from a partial state space. For this reason a reinforcement learning algorithm has been developed, making use of initially formulated expert knowledge. The application of the learning algorithm enables evaluating a subset of actions of the reachability graph in the learning step. The set of evaluated actions is used in the last step for deriving new knowledge by the use of a fuzzy clustering method. In section 4 we present numerical results for the application of the proposed reinforcement learning algorithm. Section 5 summarizes our work.

2. PETRI NET BASED CONTROL OF FLEXIBLE MANUFACTURING SYSTEMS USING EXPERT KNOWLEDGE

For modeling flexible manufacturing systems as discrete event dynamic systems for scheduling purposes, Petri Nets turned out to be efficient [1], [2], because they enable to describe both the parallelism of partial activities or operations and conflict situations in it. Discussing first the

aspects of modeling relevant parts of a manufacturing system with Coloured Petri Nets, we then propose an architecture including an algorithm for the analysis of partial reachability graphs using expert knowledge.

A. Modeling of Manufacturing Systems with Coloured Petri Nets

Coloured Petri Nets are especially suited for modeling manufacturing systems, because they have a graphical representation and a well-defined semantics allowing formal analysis. They differ from low-level nets mainly in the point, that each token can carry complex information or data. JENSEN [5] defines a Coloured Petri Net as 8-tuple:

$$CPN = (\Sigma, P, T, A, N, C, G, E, I)$$

State transitions might occur when one or more transitions are enabled in any marking. Therefore the variables of the transitions have to be bound. A binding b assigns a colour to each variable of a transition. A binding element is a pair (t, b) where t is a transition while b is a binding for the variables of t . The distribution of tokens in a state s_i on the places is called a marking M_i .

A binding element is enabled, iff enough tokens with corresponding token colours in all input-places of the transition according to the arc expressions are existing.

$$E(p, t) < b \leq M(p)$$

Usually in each system state more than one enabled binding element exists. Concurrently enabled transitions may fire and can be combined to a step that may occur. Transitions, that are in conflict, require a decision, which of the enabled transitions should occur. The interpretation of conflicts as decision problems in manufacturing systems is a common approach in literature [1], [4], [11], [12]. For the selection in case of a conflict XIONG et al. [15] propose conflict sets that are defined as sets of enabled binding elements for a marking M , if for every pair of transitions in the set, firing of one transition disables another. The problem of their approach is, that it is only suitable for the treatment of sequencing decisions. For combined routing- and sequencing decisions we propose an enhanced definition that will be used in a new approach for state space analysis. In order to select binding elements it is necessary to investigate them by defining conflict sets in a different way. Therefore we have to project the binding elements (t, b) into a cartesian product of the interesting attributes $T \times \{job\}$, where $\{job\}$ denotes the set of all jobs modeled in the CP-net and T denotes the set of transitions.

Definition: Let B be the set of enabled binding elements in the current system state for transitions that share at least one input place and let further $D \subseteq B \times B$ be a relation such that $\langle (t, b)_i, (t, b)_j \rangle$ is in D iff the corresponding binding elements intersect in the following way:

$$\{t_i\} \cap \{t_j\} \neq \emptyset \vee \{job_i\} \cap \{job_j\} \neq \emptyset$$

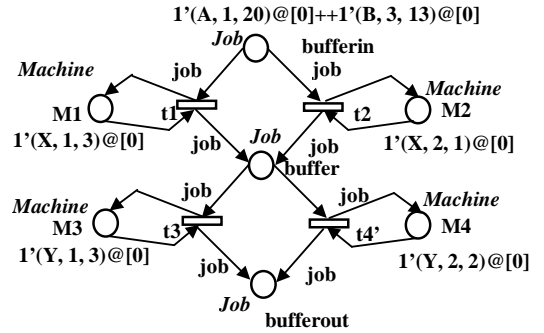
In order to control dependently enabled binding elements it is useful to derive conflict sets from the dependency graph. By doing so every pair of a conflict set is an element of D . As a consequence conflict sets exist in any state of the system when a sequencing- and/or a routing decision is required. For a sequencing decision we get the following structure of the conflict sets:

$$\{t_i(job_1), \dots, t_i(job_n)\}.$$

Routing decisions lead to the following structure of conflict sets:

$$\{t_i(job_1), \dots, t_m(job_i)\}.$$

Figure 1 depicts a Coloured Petri Net of a two-stage manufacturing system with routing- and sequencing flexibility in its initial state. In this state the existing conflict sets affect routing decisions for jobs A and B and a sequencing decision, if they are routed on the same machine.



```

color Jobtype = with A | B timed;
color Jobimportance = int;
color Jobslack = int;
color Job = product Jobtype * Jobimportance *
Jobslack timed;
color Machinetype = with X | Y;
color Machinparameter1 = int;
color Machineparameter2 = int;
color Machine = product Machinetype *
Machineparameter1 * Machineparameter2;
var jobtype:Jobtype;
var jobimportance:Jobimportance;
var jobslack:Jobslack;

```

Figure 1: Coloured Petri Net representation of a manufacturing system with routing- and sequencing flexibility

In the next section we especially use the concept of conflicts to derive a schedule constructing the reachability graph of the corresponding Coloured Petri Net partially.

B. Analysis of Reachability Graphs for Scheduling Purposes Using Expert Knowledge

In order to use Petri Nets as a process model for control purposes, the analysis of their corresponding reachability

graphs has turned out to be a suitable analysis technique. The problem in this context, consists of selecting a (sub)optimal path of the reachability graph. A reachability graph's path is derived as a sequence of selected actions from conflict sets. Traditionally the analysis of the reachability graph is a two-phased process. The first step consists of the construction of the complete reachability graph. In the second step the graph is analyzed. Existing approaches for the exploration of reachability graphs for scheduling purposes might be divided into optimizing [18] and heuristic approaches [8], [15]. In order to tackle the problem of state space explosion, an algorithm is suggested, that uses expert knowledge in the construction phase to build up only the most interesting parts of the graph for analysis purposes. The proposed basic algorithm gives up the bipartition of the construction and analysis part and combines both of them. It consists of the following steps:

- Step 1: *Initialize the current marking $M_t \leftarrow M_0$;*
 Step 2: *Compute conflict sets using the list of enabled binding elements in M_t ; If the set of enabled binding elements is empty, then terminate;*
 Step 3: *Select one binding element out of each routing conflict set;*
 Step 4: *Select one binding element out of each sequencing conflict set;*
 Step 5: *Perform the resulting step resulting in M_{t+1} ;*
 Step 6: $M_t \leftarrow M_{t+1}$;
 Step 7: *Go to Step 2;*

Performing a run with the algorithm results in a path of the underlying net's reachability graph. The key for identifying (sub)optimal paths of the reachability graph are steps 3 and 4, where action selection takes place. Our approach is to use expert knowledge in the action selection step of the algorithm. Considering that expert knowledge is formulated mostly in natural language, an approach is proposed, that is suitable to make inferences using fuzzy propositions in the knowledge base. In practice, operators decide about control strategies according to their experiences using rules of the following form, whose antecedents and consequences contain fuzzy concepts rather than crisp values:

If x is A then y is B

where A and B are fuzzy concepts. Such an inference can not be made adequately by the methods based on classical two valued logic. The classical inference method Modus Ponens is generalized using fuzzy concepts:

antecedent 1: x is A	(A')
antecedent 2: <i>If x is A then y is B</i> ($A \rightarrow B$)	
consequence: y is B'	(B')

The generalized Modus Ponens approximates the inference step in the sense, that a fact A' may only match the antecedent of the rule ($A \rightarrow B$) partially. Depending on the matching result, a consequence is inferred that equals more or less the rules consequence B . For the support of action

selection a fuzzy knowledge based system is used to model linguistic expressions. Fuzzy sets are interpreted in our approach as possibility distributions that restrict the values a variable may assume [16]. The formulation of a mathematical model for the generalized Modus Ponens depends on the semantic interpretation of the implication operator \rightarrow in ($A \rightarrow B$). Using Modus Ponens as inference mechanism the implication operator is interpreted connectively [13]. In this interpretation we look for the truth value of the elementary propositions A and B when the truth value of the compound proposition is known, especially when $A \neq A'$. Due to MAGREZ and SMETS the certainty of a proposition B , $N(B)$ only depends on $N(A)$ and $N(A \rightarrow B)$ [10].

$$N(B) = f(N(A), N(A \rightarrow B))$$

Assuming a necessarily true compound proposition ($A \rightarrow B$), i.e. $N(A \rightarrow B) = 1$, and necessarily true propositions A' and B' , $N(A)$ and $N(B)$ have to be computed [10]. Thanks to the concept of parent possibility distributions, the necessity of A knowing A' for sure is computed by the following formula [13]:

$$N(A|A') = 1 - \sup_{u \in U} t(1 - \pi_A(u), \pi_{A'}(u)) = \delta$$

If it is required, that a proposition A should be impossible if the negation $\neg A$ is certain, it follows that $t(a, b) = T_m(a, b) = \max(0, a + b - 1)$. Inserting and using a t -norm for f and $N(A \rightarrow B) = 1$ it follows:

$$N(B) = t(N(A), N(A \rightarrow B)) = t(\delta, 1) = \delta$$

and therefore for a certain proposition B' :

$$N(B|B') = 1 - \sup_{v \in V} T_m(1 - \pi_B(v), \pi_{B'}(v)) = \delta$$

To find a possibility distribution $\pi_{B^*}(v)$ it can be shown that:

$$\pi_{B^*}(v) = 1 \wedge \pi_B(v) + 1 - \delta$$

For the enhancement to the case of existing rules with conjunctively or disjunctively compound rule antecedents see MAGREZ and SMETS [10]. In accordance with the principle of minimum specificity [17], [3], a necessity measure is computed that evaluates the certainty of the rules consequences that are specifying the routing- and sequencing suitability of the jobs.

The use of the above mentioned approximate reasoning approach in a blackboard architecture enables to divide the relevant expert knowledge into distributed knowledge sources. The algorithm is implemented by the following architecture:

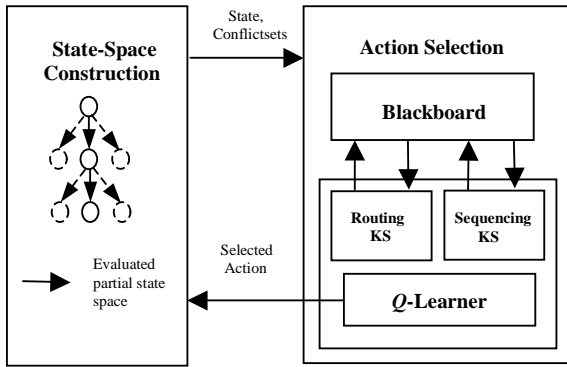


Figure 2: System architecture for knowledge based evaluated state space generation

The algorithm avoids constructing the complete reachability graph by path selection using the represented inference mechanism. The learn modul is described in detail in the next section. In order to evaluate the selected path and to derive new rules for the action selection by considering evaluated actions a necessary step consists of evaluating each action within a sequence of actions. In the next section we show how to evaluate each action considering only delayed rewards for the evaluation of terminal states and give an outline for the derivation of new knowledge based on the set of evaluated state-action pairs.

3. ADAPTION OF POLICY AND KNOWLEDGE BASE

In order to optimize the knowledge base for the proposed on-line scheduling algorithm, we suggest an off-line learning method for the adaption of the knowledge base consisting of a two step approach. First, we present a method for generating a set of evaluated state-action pairs. Subsequently, we give an outline for the derivation of a set of rules by learning new knowledge inductively, considering the set of evaluated state-action pairs as input data. Generating a set of evaluated state-action pairs in the first step means to learn an evaluated reachability graph at least partially. This labeled graph may be used for two purposes. On the one hand, it is possible to derive the optimal state-action sequence as an optimal policy from the evaluated graph. On the other hand the evaluated state-action pairs serve as input data for the adaption of the knowledge base. The problem is to compute an evaluation function which approximates the true state-action values $Q(s,a)$. Tracing the reachability graph, the only information to evaluate actions within the sequence $a_0, \dots, a_t, \dots, a_{T-1}$ consists of the signal r_{s_T} that evaluates the terminal state s_T . As a consequence, a learning algorithm has to explore the reachability graph's paths repeatedly and has to distribute the signals r_{s_T} backwards to the state-action pairs along the sequence of selected actions. If the complete state space is not available in advance, algorithms

of reinforcement learning are necessary for this kind of task. Reinforcement learning algorithms are aimed at the computation of an evaluation function $\hat{Q}(s,a)$, which approximates the true state-action values $Q(s,a)$. One key element of a successful reinforcement learning algorithm is an efficient exploration strategy. On the one hand, the reachability graph as a learning agent's environment must be explored sufficiently. On the other hand, experience gained during the learning process also has to be considered for action selection in order to minimize costs of learning. Empirical studies have proven the acceleration of the learning process using advices from an external observer e.g. [9]. Integrating expert knowledge into the exploration step of the reinforcement learning algorithm, the following enhancement of WATKINS' [14] Q -Learning algorithm is proposed:

- Step 1: For each s,a initialize table entry $\hat{Q}(s,a)$ to zero
- Step 2: Observe the current state s
- Step 3: Do forever:
 - Select an action a according to
$$V(s,a) = \tau \cdot N(s,a) + (1-\tau) \cdot \hat{Q}(s,a)$$
 \max_a
 - Receive immediate reward
 - Observe the new state s'
 - Update the table entry for $\hat{Q}(s,a)$ as follows:
$$\hat{Q}(s,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s',a')$$
 - $s \leftarrow s'$

The incorporation of expert knowledge occurs in the action selection step, which should ensure efficient exploration of the state space. To guide the exploration by considering available expert knowledge, the algorithm gives the possibility to stress the preferences of actions in a certain state, taking the evaluation of a state-action pair by the available expert knowledge using a weight τ into account. This is done by using the result $N(s,a)$ of the approximate reasoning model, that represents the certainty of the routing suitability of a job on a machine in a certain state s . A parameter τ guides the exploration step depending on the quality of available knowledge and depending on the learning progress. Assuming a bad approximation of the evaluation function $Q(s,a)$ in the beginning of learning, τ should be diminished during learning reducing the influence of the proposed actions by the approximate reasoning procedure. To gain the required signals r_{s_T} those evaluate terminal states of the state space an objective (reward) function has to be defined. The proposed architecture for the generation of a partial evaluated state space is depicted in figure 2.

On the one hand, it is possible to determine the optimal policy, that leads to the optimal terminal state with the approximated Q -function. On the other hand, this function might be used for the derivation of new knowledge taking the information of successful action selection into account. Finally an algorithm for the derivation of an adapted fuzzy knowledge base is outlined in this section. Besides other approaches e.g. genetic algorithms or decision trees we give a short outline for the derivation of an adapted fuzzy knowledge base from fuzzy clustering keeping the semantics of possibility theory.

The set of evaluated state-action pairs serves, in the step of learning fuzzy classifier rules inductively, as a set of numerical input-output sample vectors $X = \{\bar{x}_1, \dots, \bar{x}_i, \dots, \bar{x}_N\}$. According to the manufacturing problem, a k -dimensional feature vector \bar{x}_i , describing an evaluated binding element, consists of values of state variables and of the Q -value, that evaluates the corresponding decision. Learning a classifying function represented by a fuzzy rule base that induces a general hypothesis $f(\bar{x}_i)$ from the set of previously evaluated decisions, means to build up a fuzzy rule base that is able to classify not yet considered examples $\bar{x}_j \notin X$ for routing- and sequencing decisions. The data space is described by the universe $U = U_1 \times \dots \times U_k$. It is separated into an input space $U_1 \times \dots \times U_{k-1}$ and an output space U_k . Using evaluated state-action pairs from Q -Learning as input data, the elements of X are unclassified examples in the sense that there are no linguistic labels associated with them. For inducing a fuzzy rule base from X it is necessary to first classify each vector of X with its corresponding label applying a clustering algorithm. Using X as input data for the application of a possibilistic clustering algorithm [7] results in a possibilistic $C \times N$ partition matrix $U_{poss} = [u_{ij}]$. Here, u_{ij} is interpreted as grade of membership of a feature vector \bar{x}_j in a cluster β_i . Each cluster β_i models a relation between the linguistic variables of the input space and the linguistic variable of the output space. Projecting the membership values u_{ij} of a cluster β_i for all j to the one-dimensional coordinate axes results in k one-dimensional discrete possibility distributions that have to be labeled with linguistic terms [6]. Interpolating the discrete distributions leads to the required possibility distribution of the rule's input and output. Next we empirically judge the value of using our enhanced Q -Learning algorithm that takes advice in the exploration step.

4. EXPERIMENTAL RESULTS

We choose a simple Coloured Petri Net structure (figure 1) and its corresponding reachability graph as a testbed in order to present initial results transparently. To determine the job's suitability to be processed next on a possible machine we take the following attributes into account: For considering the priority of a job the job attributes slack per step and job importance are used. In order to describe the machine's suitability from a job's point of view we consider its intensity and its tendency towards breaking down as parameters. To model initially formulated expert knowledge we use a hierarchical rule base in order to compute the certainty of those propositions that determine the suitability of an attachment of the suggested machine/job pair. In order to demonstrate the influence of predefined expert knowledge we consider the average rewards depending on the number of runs. The following figure depicts the results of the proposed informed Q -Learning algorithm ($\tau = \frac{1}{x}$, where x denotes the number, a state has been visited by the agent) in contrast to a traditional Q -Learning algorithm that does not take advice in the exploration step ($\tau = 0$).

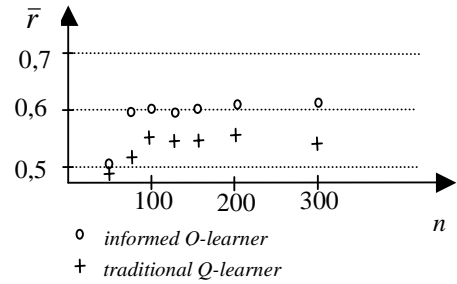


Figure 3: Average rewards as a function of amount of training episodes

It is shown that taking advice outperforms traditional Q -Learning in the way that using the proposed informed Q -Learner average rewards \bar{r} overcome the average rewards using a traditional Q -Learning algorithm. Dependently on the number of runs the informed Q -Learner explores fewer states N to gain higher average rewards.

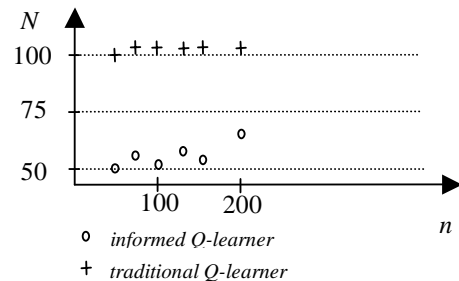


Figure 4: Number of inspected states as a function of training episodes

Advice taking in the exploration step results in a faster construction of the promising part of the corresponding reachability graph. This result is especially useful for the consideration of larger reachability graphs when learning time is expensive.

5. CONCLUSIONS AND FUTURE TRENDS

With this paper we have developed a Petri Net based approach that is suitable for analyzing the corresponding reachability graph to derive a schedule for a modeled manufacturing system. Expert knowledge efficiently guides the search for a schedule in a partial reachability graph. Empirically we have shown the benefits of intelligently integrating expert knowledge and outlined an adequate approach to induce fuzzy knowledge from examples that has to be validated empirically next.

6. REFERENCES

- [1] Al-Jaar, R.Y., Desrochers, A.A.: Petri Nets in Automation and Manufacturing, in: Saridis, G.N. (ed.): *Advances in Automation and Robotics*, Vol. 2, Greenwich, Conn., 1990, pp. 153-225
- [2] Alla, H., Ladet, P.: Colored Petri Nets: A Tool for Modeling, Validation, and Simulation of FMS, in: Kusiak, A. (ed.): *Flexible Manufacturing Systems: Methods and Studies*, New York, 1986, pp. 271-281
- [3] Dubois, D., Prade, H.: Fuzzy Sets in Approximate Reasoning, Part 1: Inference with Possibility Distributions, in: *Fuzzy Sets and Systems*, Vol. 40, 1991, pp. 143-202
- [4] Hatono, I., Yamagata, K., Tamura, H.: Modeling and On-Line Scheduling of Flexible Manufacturing Systems Using Stochastic Petri Nets, in: *IEEE Transactions on Software Engineering*, Vol. 17, No. 2, February 1991, pp. 126-132
- [5] Jensen, K.: *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*, Vol. 1, Second Edition, Berlin et al., 1997
- [6] Klawonn, F., Kruse, R.: Constructing a Fuzzy Controller from Data, in: *Fuzzy Sets and Systems*, Vol. 85, 1997, pp. 177-193
- [7] Krishnapuram, R., Keller, J.: A Possibilistic Approach to Clustering, *IEEE Transactions on Fuzzy Systems*, Vol. 1, No. 2, May 1993, pp. 98-110
- [8] Lee, D.Y., DiCesare, F.: FMS Scheduling Using Petri Nets and Heuristic Search, in: *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, May 1992, pp. 1057-1062
- [9] Maclin, R., Shavlik, J.W.: Incorporating Advice into Agents that learn from Reinforcements, in: *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Vol. 1, Seattle, USA, Juli-August, 1994, pp. 694-699
- [10] Magrez, P., Smets, P.: Fuzzy Modus Ponens: A New Model Suitable for Applications in Knowledge-Based Systems, in: *International Journal of Intelligent Systems*, Vol. 4, 1989, pp. 181-200
- [11] Martinez, J., Muro, P., Silva, M.: Modeling, Validation and Software Implementation of Production Systems Using High Level Petri Nets, *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, Raleigh, N.C., April 1987, pp. 1180-1185
- [12] Sackmann, D., Ey, H., Bastian, A.: Decision Support in FMSs Using a Fuzzy-Petrimet Approach, in: *Proceedings of the European Symposium on Applications of Intelligent Technologies*, Aachen, Germany, September 9-10, 1997, pp. 73-80
- [13] Smets, P.: Implication and Modus Ponens in Fuzzy Logic, in: *Technical Report No. TR/IRIDIA/90-18*, Universite Libre De Bruxelles, 1990
- [14] Watkins, C.J.: *Learning from Delayed Rewards*, PhD Thesis, King's College, Cambridge University, 1989
- [15] Xiong, H.H., Zhou, M.C., Manikopoulos, C.: Scheduling Flexible Manufacturing Systems Based on Timed Petri Nets and Fuzzy Dispatching Rules, in: *Proceedings of the IEEE Symposium on Technologies and Factory Automation*, Vol. 3, October 1995, pp. 309-315
- [16] Yager, R.R.: An Approach to Inference in Approximate Reasoning, in: *International Journal of Man-Machine Studies*, Vol. 13, 1980, pp. 323-338
- [17] Zadeh, L.A.: Fuzzy Sets as a Basis for a Theory of Possibility, in: *Fuzzy Sets and Systems*, Vol. 1, 1978, pp. 3-28
- [18] Zhou, M.C., Chiu, H.-S., Xiong, H.H.: Petri Net Scheduling of FMS Using Branch and Bound Method, in: *Proceedings of the IEEE International Conference on Industrial Electronics*, Orlando, November 1995, pp. 211-216