Received December 6, 2018, accepted December 13, 2018, date of publication December 24, 2018, date of current version February 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2889305

An Efficient Deadlock Recovery Policy for Flexible Manufacturing Systems Modeled With Petri Nets

YUNYUN DONG[1,2], YUFENG CHEN[1,2], (Senior Member, IEEE), SHAOYONG LI[3], MOHAMMED A. EL-MELIGY[4], AND MOHAMED SHARAF[5]

[1]School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China
[2]Fujian Engineering Research Center of Motor Control and System Optimal Schedule, Huaqiao University, Xiamen 361021, China
[3]School of Civil Engineering, Lanzhou University of Technology, Lanzhou 730050, China
[4]Advanced Manufacturing Institute, King Saud University, Riyadh 11421, Saudi Arabia
[5]School of Industrial Engineering, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia

Corresponding author: YuFeng Chen (chyf01@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61673309, Grant 61472295, Grant 61672400, and Grant 61873342, in part by the Fundamental Research Funds for the Central Universities under Grant JB190415, and in part by the Deanship of Scientific Research at King Saud University under Grant RG-1440-048.

ABSTRACT This paper focuses on solving deadlock problems in flexible manufacturing systems modeled with Petri nets by adding a set of recovery transitions. Different from the traditional deadlock control policies that add control places to a net model, this paper employs transitions to recover all the deadlock markings to be legal ones. A live net system can be obtained with all reachable markings. First, according to reachability graph analysis, a set of recovery transitions is obtained for each deadlock marking to be recovered. Second, we present a vector intersection approach to compute a recovery transition to recover multiple deadlock markings. Then, an iterative method is developed to find a set of recovery transitions to recover all deadlock markings. At each iteration step, a recovery transition is found to recover more than one deadlock markings. This iterative method cannot obtain the minimal number of recovery transitions in theory, but it can usually lead to a minimal one supported by extensive experimental studies. Finally, several widely used examples are provided to demonstrate the presented approach. The experimental results show that the reported deadlock recovery technique is effective and efficient.

INDEX TERMS Flexible manufacturing system, Petri net, deadlock, vector intersection.

I. INTRODUCTION

Flexible manufacturing systems (FMSs) [6], [15] are designed to complete different kinds of jobs by using limited and shared resources such as machines and robots. Deadlocks are highly undesirable situations caused by the competition for the shared resources in different processes, making the whole system or a part of it blocked and some production processes discontinuous. In FMSs, once a deadlock occurs, it usually leads to considerable and inexcusable loss such as long downtime and waste of resources. Hence, deadlock problems must be analyzed and resolved in these systems.

Deadlocks in FMSs can be dealt with by several tools: Graph theory [10], [11], [25], [47], automata [34], [35], [39], [45], [46], and Petri nets [11], [21], [28], [40], [44]. As a mathematical and graphical modeling tool, Petri nets are widely used to model and analyze the behavior of FMSs [6], [15], [22], [32], [36], [38]. They can detect deadlocks and develop a policy to prevent deadlocks. Many researchers prefer to use Petri nets to deal with deadlock problems [11], [11], [13]. Generally, there are three approaches for deadlock resolution: Deadlock avoidance [1], [14], [16], [25], [31], deadlock prevention [4], [7], [12], [14], [19], [57]–[63] and deadlock detection and recovery [20], [29], [33]. This work belongs to the last category, i.e., deadlock detection and recovery. The main idea of this approach is adding a set of recovery transitions such that all deadlock markings are recovered to legal ones.

In order to prevent the undesired states of a system from being reached, a variety of policies are developed by adding a set of control places to a Petri net model to be controlled [50], [54], [56]. Different from the traditional deadlock prevent policies, this work attempts to find a set of control transitions [53], [55], aiming to recover the deadlock states of a Petri net to be legal ones.

Generally, two analysis techniques are considered to deal with deadlock problems in Petri nets: Structural analysis [1], [17], [19], [49] and reachability graph analysis [30], [38], [41], [51], [52]. Structural analysis usually exploits special structural objects of a Petri net, siphons, place invariants, and resource-transition circuits, for examples, since some substructures have a close relationship with deadlocks. However, the optimality of permissive behavior cannot be guaranteed as partial legal markings are always prohibited. Reachability graph analysis is very specific and visual since it can completely show the behavioral evolution of a Petri net model. From the reachability graph, we can find the deadlock markings, bad markings, dangerous markings, and good markings [9], [30], [40], [48]. For the deadlock control purpose, the bad and deadlock markings should not appear in the controlled system.

Uzam and Zhou [38] and Uzam *et al.* [42] define an important class of markings, namely, the first-met bad markings (FBMs). First, they classify a reachability graph into two groups: A live-zone (LZ) and a deadlock-zone (DZ). The LZ contains all legal markings and the DZ contains all deadlock markings and bad markings. An illegal marking is called an FBM if it represents the first entry from the LZ to the DZ. Obviously, if no FBM is reached in the controlled system, the bad or deadlock markings cannot be reached anymore. Then, Chen *et al.* [2] and Chen and Li [3] improve Uzam and Zhou's work by proposing a vector covering approach to reduce the number of legal markings and FBMs that need to be considered. More importantly, an optimal supervisor can be designed by making all legal markings reachable but all FBMs unreachable if such a supervisor exists.

In [18], a transition-controlled deadlock recovery policy is proposed for a subclass of Petri nets, namely systems of simple sequential processes with resources (S^3PR for short) [4], [41], [45]. Different from deadlock prevention policies, the transition-controlled approach computes a set of transitions for a Petri net model. In an FMS, deadlocks are caused by the competition for shared resources when some processes keep waiting indefinitely for the other processes to release resources. Once a deadlock occurs, the system should be recovered to be some safe states. In this sense, recovery transitions can be considered as the recovery actions. By firing these transitions, all deadlock markings can be transformed to legal markings, resulting in a live net model. However, the work has at least two disadvantages. One is that it is applied to S^3PR only, a subclass of Petri nets. The other is that it usually obtains too many transitions, which means that it suffers from the structural complexity problem.

In order to overcome the structural complexity problem, Chen *et al.* [8] present an improved transition-based deadlock recovery policy. This study proposes two approaches to design recovery transitions. The first one is an iterative approach. At each iterative step, an integer linear programming problem (ILPP) is formulated to design a recovery transition, aiming to maximize the number of deadlock markings recovered by the obtained transition. The second one is a non-iterative approach that can find all recovery transitions at a time. The objective function of the ILPP aims to minimize the number of selected recovery transitions and the constraints can ensure that each deadlock marking is recovered by at least one selected recovery transition. Then, a minimal number of recovery transitions are obtained by solving one ILPP only. Although this policy can be applied to all classes of FMS-oriented Petri net models, it suffers from the computation complexity problem, since there are too many constraints and variables in the formulated ILPPs, leading to a long calculation time.

In this paper, we propose a transition-based deadlock recovery policy without solving ILPPs. First, we compute the reachability graph of a Petri net model and find all legal and deadlock markings. Then, a vector covering approach is introduced, which can greatly reduce the number of legal markings that are required to be considered. As a result, the computational overhead can be greatly reduced. A vector intersection approach is designed to compute all recovery transitions that can be fired at deadlock markings and their firing leads the system to legal markings. In fact, it is an iterative process. At each iteration step, we acquire a recovery transition to recover as many deadlock markings as possible. At the first step, we compute the shared recovery transitions called intersection of the first two deadlock markings. If the intersection is empty, it means that the two deadlock markings have no shared recovery transition, i.e., there is no transition to recover the two deadlock markings at the same time. Then, we compute the intersection of all possible recovery transitions of the first and third deadlock markings. If the intersection is not empty, it means that the recovery transition can simultaneously recover the first two deadlock markings. Then, we compute the intersection of the obtained intersection and all possible recovery transitions of the third deadlock marking, and so on. After each deadlock has been calculated once, we compute the intersection of the remaining deadlock markings according to the above method. The process ends when all deadlock markings are recovered. Finally, a small number of recovery transitions are obtained.

For economy of space, some basics of Petri nets [24] and the vector covering approach in [2] are outlined in [9]. The rest of the paper is organized as follows. In Section 2, reachability graph analysis and a vector covering approach are recalled. In Section 3, we present the derivation process called vector intersection to find a set of recovery transitions. A simple example is also presented to illustrate the proposed method. Some widely used examples from the literature are provided to demonstrate the performance of the proposed method in Section 4. Finally, Section 5 concludes this paper.

II. TRANSITION-CONTROLLED DEADLOCK RECOVERY

A. DEADLOCK RECOVERY BY TRANSITIONS

In this section a concrete design method of recovery transitions is shown, which leads all of the deadlock markings to legal ones. As mentioned previously, the basics of Petri net are referred to [10], where $\mathcal{M}_L$ and $\mathcal{M}_D$ denote the sets of legal and deadlock markings, respectively.

Definition 1: *Let M_d be a deadlock marking of a net system (N, M_0) with $N = (P, T, F, W)$ and $t_r \notin T$ be an external transition. If t_r is enabled at M_d and its firing at M_d leads to a legal marking of (N, M_0), t_r is said to be a recovery transition of M_d, i.e., there exits a legal marking $M_l \in \mathcal{M}_L$ such that $M_d[t_r\rangle M_l$. The set of recovery transitions is denoted as T_r, i.e., $T_r = \{t_r \mid \exists M_l \in \mathcal{M}_L, \exists M_d \in \mathcal{M}_D, M_d[t_r\rangle M_l\}$.*

Definition 2: *Let (N, M_0) be a net model system, and (N_r, M_0) be the resulting net model system after adding a set of recovery transitions T_r, where $N_r = (P, T \cup T_r, F_r, W_r)$ with $F_r \subseteq (P \times T \cup T_r) \cup (T \cup T_r \times P)$ (the flow relation of N_r) and W_r: $(P \times T \cup T_r) \cup (T \cup T_r \times P) \rightarrow \mathbb{N}$ (the weight of the arcs in N_r). The set of recovery transitions T_r is said to be behaviorally optimal if $R(N_r, M_0) = R(N, M_0)$.*

A recovery transition is defined as a transition t_r with the incidence vector $[N_r](P, t_r) = [x_1, x_2, x_3, \ldots, x_n]^T$, a column in $[N_r]$, where n denotes the number of places in a net system with $n = |P|$, which is enabled at M_d and its firing yields a legal marking. For a given system, the number of deadlock markings is fixed and our purpose is to recover all deadlock markings. However, the consideration of all legal markings undoubtedly increases the computational cost. According to [9, Corollary 1], if all markings in the minimal covering set $\mathcal{M}_L^*$ of legal markings can be reached, all legal markings can also be reached. Therefore, we can recover a deadlock marking M_d into a marking of the minimal covering set of legal markings $\mathcal{M}_L^*$ instead of all legal markings $\mathcal{M}_L$.

According to Definition 1, a deadlock marking M_d can be guided to a legal making M' after firing a recovery transition t_r, that is to say, there exists a marking $M \in \mathcal{M}_L^*$ with $M_d + [N_r]\overrightarrow{t_r} = M' \leq_A M$ [9]. $M' \leq_A M$ means that the marking M' is A-covered by marking M if for all places $p \in P_A$, $M(p) \geq M'(p)$. For the sake of simplicity, the markings in $\mathcal{M}_L^*$ are considered instead of all legal markings $\mathcal{M}_L$.

Generally, the places in a Petri net model of an FMS are classified into three parts: Idle, activity (operation), and resource places. Their sets are denoted as P^0, P_A, and P_R ($P = P^0 \cup P_A \cup P_R$), respectively, and their basis of classification is outlined in [9].

Assume that there is a net system (N_r, M_0) and a subvector of $[N_r](P, t_r)$ with restriction to $P_A \subseteq P$, denoted by $[N_r](P_A, t_r)$. The recovery transition t_r is used to transform a deadlock marking M_d into a marking A-covered by some markings in $\mathcal{M}_L^*$. That is to say, if the recovery transition t_r is enabled at a deadlock marking M_d and its firing leads to a marking M_{l1}, i.e., $M_d[t_r\rangle M_{l1}$, there exists a marking $M_{l2} \in \mathcal{M}_L^*$, $M_{l1} \leq_A M_{l2}$. Hence,

$$\exists M_l \in \mathcal{M}_L^*,\ 0 \leq M_d(p_i) + x_i \leq M_l(p_i),\ \forall p_i \in P_A \quad (1)$$

We can derive Eq. (2) by Eq. (1)

$$\exists M_l \in \mathcal{M}_L^*,\ -M_d(p_i) \leq x_i \leq M_l(p_i) - M_d(p_i),\ \forall p_i \in P_A \quad (2)$$

The theoretical method of calculating x_i's for all places $p_i \in P_A$ is represented by Eq. (2). In order to obtain the complete incidence relation of a recovery transition t_r, we need to compute x_i's for all places $p_i \in P \setminus P_A$. The analysis and derivation process is shown in below.

Theorem 1 [24]: *A Petri net is conservative iff there exists a P-vector I of positive integers such that $I^T[N] = \mathbf{0}^T$.*

There is a property in [24]: If a Petri net is structurally bounded and structurally live, then it is both conservative and consistent. Combined with Theorem 1, it indicates that conservativeness is a necessary condition for structural boundedness and liveness of a Petri net. That is to say, if a net model is not conservative, it is either not structurally bounded or not structurally live. As a result, the addition of recovery transitions must guarantee the conservativeness of a Petri net. Meanwhile, there are two assumptions as follows in the considered class of Petri net model.

Assumption 1: *Each idle place $p_{id} \in P^0$ is associated with a minimal P-semiflow $I_{p_{id}}$, such that for all places $p \in \|I_{p_{id}}\| \setminus \{p_{id}\}$, $p \in P_A$ holds, where P_A is the set of operation places in the Petri net model.*

Assumption 2: *Each resource place $p_r \in P_R$ is associated with a minimal P-semiflow I_{p_r}, such that for all places $p \in \|I_{p_r}\| \setminus \{p_r\}$, $p \in P_A$ holds.*

In summary, the above two assumptions are used to compute x_i's for all places $p_i \in P \setminus P_A$. Since the set of places is $P = P^0 \cup P_A \cup P_R$, we have $p_i \in P^0$ or $p_i \in P_R$ if $p_i \in P \setminus P_A$. Therefore, the places associated with x_i's can be partitioned into two subsets $P_x^1 \subseteq P^0$ and $P_x^2 \subseteq P_R$. In fact, the calculation method for each recovery transition is the same. To ease the description, the following derivation is exemplified by the case of adding only one recovery transition t_r to a net model.

First, we only need to consider x_i's for all places $p_{id} \in P^0$. According to Assumption 1, a minimal P-semiflow $I_{p_{id}}$ is obtained with $\|I_{p_{id}}\| \subseteq P_A \cup \{p_{id}\}$. Hence, it is easy to find ${I_{p_{id}}}^T = [{I_{p_{id}}}^T(P_A), {I_{p_{id}}}^T(p_{id}), {I_{p_{id}}}^T(P^0 \setminus \{p_{id}\})]$ where ${I_{p_{id}}}^T(P^0 \setminus \{p_{id}\}) = \mathbf{0}$. In order to guarantee the conservativeness of the Petri net model after adding a recovery transition t_r, we have ${I_{p_{id}}}^T \cdot [N_r](P, t_r) = 0$, i.e., ${I_{p_{id}}}^T \cdot [N_r](P_A \cup \{p_{id}\} \cup (P^0 \setminus \{p_{id}\}), t_r) = 0$. Since ${I_{p_{id}}}^T(P^0 \setminus \{p_{id}\}) = \mathbf{0}$, we have ${I_{p_{id}}}^T(P_A) \cdot [N_r](P_A, t_r) + {I_{p_{id}}}^T(p_{id}) \cdot [N_r](p_{id}, t_r) = 0$. Hence, $[N_r](p_{id}, t_r) = -{I_{p_{id}}}^T(P_A) \cdot [N_r](P_A, t_r)/{I_{p_{id}}}^T(p_{id})$. That is to say, $x_{id} = -\sum_{p_i \in P_A} I_{p_{id}}(p_i) \cdot x_i / I_{p_{id}}(p_{id})$ for all places $p_{id} \in P^0$.

Second, we consider x_i's for all places $p_r \in P_R$. According to Assumption 2, we can obtain a minimal P-semiflow I_{p_r} with $\|I_{p_r}\| = P_A \cup \{p_r\}$. Let ${I_{p_r}}^T = [{I_{p_r}}^T(P_A), {I_{p_r}}^T(p_r), {I_{p_r}}^T(P_R \setminus \{p_r\})]$. Similarly to the computation of x_i's for all places $p_{id} \in P^0$, we have $[N_r](P_r, t_r) = -{I_{p_r}}^T(P_A) \cdot [N_r](P_A, t_r)/{I_{p_r}}^T(p_r)$, i.e., $x_r = -\sum_{p_i \in P_A} I_{p_r}(p_i) \cdot x_i / I_{p_r}(p_r)$ for all places $p_r \in P_R$.

Eventually, we can find x_i's for all places $p_i \in P$. That is to say, the complete incidence of a recovery transition can be obtained.

B. THE DERIVATION OF RECOVERY TRANSITIONS

This section introduces the specific calculation process of x_i's for all places $p_i \in P_A$. To describe expediently, we consider operation places only since the computation for idle and resource places has been introduced in Section II-A.

Definition 3: *Let M_d be a deadlock marking, t_r be a recovery transition with $[N_r](P_A, t_r)$ with $[N_r](p_i, t_r) = x_i$ for all $p_i \in P_A$, and $\mathcal{M}_L^*$ be the minimal covering set of legal markings. A vector L_b with $L_b(i) = -M_d(p_i)$ ($p_i \in P_A$) and a set $\mathcal{U}_b = \{U_b \mid U_b(p_i) = M_l(p_i) - M_d(p_i), p_i \in P_A, M_l \in \mathcal{M}_L^*\}$ are called the lower bound and the set of upper bounds of x_i's ($p_i \in P_A$), respectively.*

According to Eq. (2) and Definition 3, we have $L_b(i) \leq x_i \leq U_b(i)$, $U_b \in \mathcal{U}_b$. Hence, any incidence vector satisfying this constraint represents a recovery transition, which leads the corresponding deadlock to a legal marking. Furthermore, if the equation relationship holds, there is a solution x_i for Eq. (2) with $x_i = -M_d(p_i)$ for all $p_i \in P_A$, which indicates that we can find a recovery transition for any deadlock marking such that $[N_r](P_A, t_r) = L_b = -M_d(P_A)$.

Proposition 1: *Let $M_l \in \mathcal{M}_L^*$ be a legal marking and M_d be a deadlock marking. An upper bound U_b with $U_b(i) = M_l(p_i) - M_d(p_i)$ ($p_i \in P_A$) can be excluded from the set of upper bounds $\mathcal{U}_b$ if there exists a place $p_i \in P_A$ satisfying $U_b(p_i) < L_b(p_i)$.*

Proof: Eq. (2) shows that for all places $p_i \in P_A$, $L_b(p_i) \leq x_i \leq U_b(p_i)$. If there exists a place $p_i \in P_A$ such that $U_b(p_i) < L_b(p_i)$, i.e., the lower bound is greater than the upper bound. It means that there is no vector of x_i's satisfying Eq. (2). In other words, the upper bound is invalid since it cannot provide any feasible solution for x_i's. The conclusion holds. ■

Assume that, for a certain net, we have the lower bound $L_b = [1, 0, 1, 0, -1]^T$ and an upper bound $U_b = [1, 1, 0, 1, 1]^T$ with $U \in \mathcal{U}_b$ for a deadlock marking. There is no x_3 satisfying $1 \leq x_3 \leq 0$ since $U_b(3) < L_b(3)$. That is to say, $L_b(i) \leq x_i \leq U_b(i)$ is an invalid constraint. Therefore, the upper bound U_b should be removed from the set of upper bounds $\mathcal{U}_b$.

For deadlock recovery purpose, a deadlock marking is recovered to any of legal markings. That is to say, there are $|\mathcal{M}_L^*|$ upper bounds and one lower bound for each deadlock marking M_d. The time complexity of the calculation increases as the number of upper bounds increases. Proposition 1 can greatly reduce the number of upper bounds.

For a deadlock marking M_d, all x_i's satisfying the upper and lower bounds are valid. As a result, there may exist more than one recovery transition for each deadlock marking M_d. It is guaranteed that we can design at least one recovery transition for each deadlock marking. For a net model with $|\mathcal{M}_D|$ deadlock markings, it needs $|\mathcal{M}_D|$ recovery transitions at most to recover all deadlock markings. Actually, there may exist common recovery transitions sets from some of the deadlock markings which can be called intersections. Mathematically, in general, for a given intersection of two sets $\mathcal{A}$ and $\mathcal{B}$, it means that all elements in this intersection belong to both $\mathcal{A}$ and $\mathcal{B}$. That is to say, a transition may recover multiple deadlock markings. By considering the structural complexity of the controlled net model, we aim to find a small number of recovery transitions to recover all deadlock markings. A method for computing the shared recovery transitions is presented in what follows.

Definition 4: *Let Y and X be two $m \times 1$ vectors. Write $X \geq Y$ if each element in vector X is greater than or equal to the corresponding element in vector Y, i.e., $X(i) \geq Y(i)$. Similarly, $X \leq Y$ if $X(i) \leq Y(i)$, for all i.*

To make it easier to express, there exists a set of vectors $[L, U]$, denoted as $\{X \mid L \leq X \leq U\}$, if each element X in it satisfies $X \geq L$ and $X \leq U$. For example, given two vectors $L = [1, 0, 0, 1]^T$ and $U = [2, 0, 1, 1]^T$, the set $[L, U]$ contains four vectors with $X_1 = [1, 0, 0, 1]^T$, $X_2 = [1, 0, 1, 1]^T$, $X_3 = [2, 0, 0, 1]^T$, and $X_4 = [2, 0, 1, 1]^T$.

Definition 5: *Let $[L, U_1]$ and $[L, U_2]$ be two sets of vectors. $[L, U_1] \cup [L, U_2] = \{X \mid L \leq X \leq U_1 \vee L \leq X \leq U_2\}$, which is denoted as $[L, \{U_1, U_2\}]$ or $[L, \mathcal{U}]$ where $\mathcal{U} = \{U_1, U_2\}$.*

Definition 6: *Let L_1 and L_2 be two $m \times 1$ vectors, $\mathcal{U}_1$ and $\mathcal{U}_2$ be two sets of $m \times 1$ vectors with $\mathcal{U}_1 = \{U_{11}, U_{12}, \ldots, U_{1k}\}$ and $\mathcal{U}_2 = \{U_{21}, U_{22}, \ldots, U_{2j}\}$. Then, the intersection of $[L_1, \mathcal{U}_1]$ and $[L_1, \mathcal{U}_2]$, denoted as $[L_1, \mathcal{U}_1] \cap [L_2, \mathcal{U}_2]$ is defined as $\{X \mid X \in [L_1, \mathcal{U}_1] \wedge X \in [L_2, \mathcal{U}_2]\}$.*

Theorem 2: *Let L_1 and L_2 be two $m \times 1$ vectors, $\mathcal{U}_1$ and $\mathcal{U}_2$ be two sets of $m \times 1$ vectors with $\mathcal{U}_1 = \{U_{11}, U_{12}, \ldots, U_{1k}\}$ and $\mathcal{U}_2 = \{U_{21}, U_{22}, \ldots, U_{2j}\}$. Let $L = max\{L_1(i), L_2(i)\}$ and $\mathcal{U} = \{V_{kj} \mid V_{kj}(i) = min\{U_{1k}(i), U_{2j}(i)\}, U_{1k} \in \mathcal{U}_1, U_{2j} \in \mathcal{U}_2\}$. Then $[L_1, \mathcal{U}_1] \cap [L_2, \mathcal{U}_2] = [L, \mathcal{U}]$.*

Proof: First, we prove that $[L_1, \mathcal{U}_1] \cap [L_2, \mathcal{U}_2] \subseteq [L, \mathcal{U}]$ is true. Let Y be an $m \times 1$ vector in $[L_1, \mathcal{U}_1] \cap [L_2, \mathcal{U}_2]$. We have $Y \in [L_1, \mathcal{U}_1]$ and $Y \in [L_2, \mathcal{U}_2]$, i.e., there exist two vectors $U_{1k} \in \mathcal{U}_1$ such that $L_1 \leq Y \leq U_{1k}$ and $U_{2j} \in \mathcal{U}_2$ such that $L_2 \leq Y \leq U_{2j}$. Since $L_1 \leq Y$ and $L_2 \leq Y$, we have $max\{L_1(i), L_2(i)\} \leq Y(i)$, i.e., $L \leq Y$. Since $Y \leq U_{1k}$ and $Y \leq U_{2j}$, we have $Y(i) \leq min\{U_{1k}(i), U_{2j}(i)\}$, i.e., there exists a vector $V_{kj} \in \mathcal{U}$ such that $Y \leq V_{kj}$. Hence, we obtain $Y \in [L, \mathcal{U}]$. That is to say, $[L_1, \mathcal{U}_1] \cap [L_2, \mathcal{U}_2] \subseteq [L, \mathcal{U}]$.

Second, we certify that $[L, \mathcal{U}] \subseteq [L_1, \mathcal{U}_1] \cap [L_2, \mathcal{U}_2]$ is true. Let X be an $m \times 1$ vector in $[L, \mathcal{U}]$. Hence, there exists a vector $V_{kj} \in \mathcal{U}$ such that $L \leq X \leq V_{kj}$. Since $L(i) = \max\{L_1(i), L_2(i)\}$, we have $L_1 \leq X$ and $L_2 \leq X$. By $V_{kj} = min\{U_{1k}(i), U_{2j}(i)\}$, we have $X \leq U_{1k}$ and $X \leq U_{2j}$. According to $U_{1k} \in \mathcal{U}_1$, $X \in [L_1, \mathcal{U}_1]$ is obtained. Similarly, by $U_{2j} \in \mathcal{U}_2$, we have $X \in [L_2, \mathcal{U}_2]$. Then, it is concluded that $X \in [L_1, \mathcal{U}_1] \cap [L_2, \mathcal{U}_2]$. That is to say, $[L, \mathcal{U}] \subseteq [L_1, \mathcal{U}_1] \cap [L_2, \mathcal{U}_2]$.

Finally, by $[L_1, \mathcal{U}_1] \cap [L_2, \mathcal{U}_2] \subseteq [L, \mathcal{U}]$ and $[L, \mathcal{U}] \subseteq [L_1, \mathcal{U}_1] \cap [L_2, \mathcal{U}_2]$, we have $[L_1, \mathcal{U}_1] \cap [L_2, \mathcal{U}_2] = [L, \mathcal{U}]$. The conclusion holds. ■

Let us present an example to illustrate Theorem 2. Let $L_1 = [-1, 0, -1, 1]^T$ and $L_2 = [0, 1, 0, -1]^T$. Then, we have $L = max\{L_1, L_2\} = [0, 1, 0, 1]^T$. let $U_{11} = [2, 1, 0, 1]^T$, $U_{12} = [1, 1, 2, 1]^T$, $U_{21} = [1, 2, 1, 1]^T$, and $U_{22} = [1, 1, 3, 1]^T$. We have $min\{U_{11}, U_{21}\} = [1, 1, 0, 1]^T$.

Similarly, $min\{U_{11}, U_{22}\} = [1, 1, 0, 1]^T$, $min\{U_{11}, U_{21}\} = [1, 1, 1, 1]^T$, and $min\{U_{11}, U_{21}\} = [1, 1, 2, 1]^T$. The set $\mathcal{U} = \{V_{kj} \mid V_{kj}(i) = min\{U_{1k}(i), U_{2j}(i)\}, U_{1k} \in \mathcal{U}_1, U_{2j} \in \mathcal{U}_2\}$ contains the above three vectors since there are two vectors that are identical.

According to Eq. (2), there exists more than one solution of x_i's satisfying the constraint. That is to say, a deadlock marking can be recovered to legal markings by firing multiple recovery transitions, and each of them has an incidence vector $[N_r](P_A, t_r)$ with $[N_r](p_i, t_r) = x_i$ for all $p_i \in P_A$. corresponding to it. Hence, there is a set of incidence vectors for one deadlock marking M_d, whose definition is as follows.

Definition 7: The set of incidence vectors for a deadlock marking M_d is defined as $\mathcal{T}_{M_d}(P_A) = [-M_d(P_A), \mathcal{U}_{M_d}(P_A)]$, where $\mathcal{U}_{M_d}(P_A) = \{X \mid X = M_l(P_A) - M_d(P_A), M_l \in \mathcal{M}_L^*\}$.

Theorem 3: Let M_{d_1} and M_{d_2} be two deadlock markings, whose set of incidence vectors are denoted as $\mathcal{T}_{M_{d_1}}(P_A) = [-M_{d_1}(P_A), \mathcal{U}_{M_{d_1}}(P_A)]$ and $\mathcal{T}_{M_{d_2}}(P_A) = [-M_{d_2}(P_A), \mathcal{U}_{M_{d_2}}(P_A)]$, respectively. A common recovery transition t_r with $[N_r](p, t_r) \in \mathcal{T}_{M_{d_1}} \cap \mathcal{T}_{M_{d_2}}$ can recover two deadlock markings M_{d_1} and M_{d_2}.

Proof: A recovery transition t_r with $[N_r](p, t_r) \in \mathcal{T}_{M_{d_1}} \cap \mathcal{T}_{M_{d_2}}$ means $[N_r](p, t_r) \in \mathcal{T}_{M_{d_1}}$ and $[N_r](p, t_r) \in \mathcal{T}_{M_{d_2}}$. That is to say, t_r can recover both deadlock markings M_{d_1} and M_{d_2}. The conclusion holds. ■

By Theorem 3 and Definition 6, $\mathcal{T}_{M_{d_1}} \cap \mathcal{T}_{M_{d_2}}$ is said to be the intersection of incidence vector recovery transitions for two deadlock markings M_{d_1} and M_{d_2} or intersection of recovery transitions for short.

In the following, an example is provided to demonstrate the proposed approach. Suppose that in a net model there are two deadlock markings $M_{d1} = [1, 1, 0, 1, 0, 0]^T$ and $M_{d2} = [1, 0, 0, 1, 1, 0]^T$, and two legal markings $M_{L1}^* = [1, 1, 1, 0, 0, 0]^T$ and $M_{L2}^* = [0, 0, 0, 1, 1, 1]^T$. According to Definition 3, there exist two lower bounds $L_1 = [-1, -1, 0, -1, 0, 0]^T$ and $L_2 = [-1, 0, 0, -1, -1, 0]^T$, and two sets of upper bounds $\mathcal{U}_1$ and $\mathcal{U}_2$ for deadlock markings M_{d1} and M_{d2}, respectively, where $\mathcal{U}_1 = \{U_{11}, U_{12}\}$ with $U_{11} = [0, 0, 1, -1, 0, 0]^T$ and $U_{12} = [-1, -1, 0, 0, 1, 1]^T$, and $\mathcal{U}_2 = \{U_{21}, U_{22}\}$ with $U_{21} = [0, 1, 1, -1, -1, 0]^T$ and $U_{22} = [-1, 0, 0, 0, 0, 1]^T$. First, according to Theorem 2, the lower bound of the intersection is L with $L(i) = max\{L_1(i), L_2(i)\} = [-1, 0, 0, -1, 0, 0]^T$. Next, we aim to reduce the number of the upper bounds. The upper bound U_{12} is excluded since $L(2) > U_{12}(2)$. Similarly, the upper bound U_{21} is also excluded since $L(5) > U_{21}(5)$. Finally, there is only one upper bound retained for each deadlock marking. The shared upper bound is $\mathcal{U} = \{V_{kj} \mid V_{kj}(i) = min\{U_{1k}(i), U_{2j}(i)\}, U_{1k} \in \mathcal{U}_1, U_{2j} \in \mathcal{U}_2\}$, i.e., $\mathcal{U} = \{V_{12} \mid V_{12}(i) = min\{U_{11}(i), U_{22}(i)\}\}$ with $V_{12} = [-1, 0, 0, -1, 0, 0]^T$. For this example, the lower bound and the upper bound are the same, i.e., $L = V_{12} \in \mathcal{U}$. Hence, there is only one vector left in $[L, \mathcal{U}]$, i.e., $[-1, 0, 0, -1, 0, 0]^T$. As a result, the recovery transition t_r of the two deadlock markings is $[N_r](P_A, t_r) = [-1, 0, 0, -1, 0, 0]^T$.

C. ITERATIVE INTERSECTION APPROACH FOR RECOVERY TRANSITIONS

In Section II-B, an approach is developed to compute a common recovery transition to recover two deadlock markings. In a more general case, we may need to design multiple recovery transitions to recover all deadlock markings. The set of recovery transitions is denoted as $T_r = \{t_{r1}, t_{r2}, t_{r3}, \ldots, t_{rm}\}$, where m indicates the number of recovery transitions added to a net model. In order to make the net model as structurally simple as possible, it is significant to minimize the number of recovery transitions to recover all deadlock markings. To this end, an iterative intersections method for recovery transitions is presented. Let $\Omega(tr_j)$ denote the set of deadlock markings that are recovered by transition t_{r_j}, i.e.,

$$\Omega(tr_j) = \{M_d \in \mathcal{M}_{\mathcal{D}} \mid M_d \propto t_{r_j}\} \tag{3}$$

Assume that there are n_d deadlock markings in a net model. According to Theorem 3, the intersection of recovery transitions for the first two deadlocks is computed. If the intersection is empty ($\mathcal{T}_{M_{d_1}} \cap \mathcal{T}_{M_{d_2}} = \emptyset$), record the second deadlock marking and then calculate the intersection of recovery transitions of the first and the third deadlock markings ($\mathcal{T}_{M_{d_1}} \cap \mathcal{T}_{M_{d_3}}$). Otherwise, we calculate the intersection of the obtained intersection and recovery transitions of the third deadlock marking (($\mathcal{T}_{M_{d_1}} \cap \mathcal{T}_{M_{d_2}}) \cap \mathcal{T}_{M_{d_3}}$) and so on. Repeat the above process until the last deadlock marking has been processed. As a result, one recovery transition is obtained and all deadlocks that have no shared intersection are recorded. Second, we compute the intersection of deadlock markings once again based on the recorded deadlock markings until the last recorded deadlock marking has been computed. Then, the second recovery transition is obtained and so on. Each step of the iteration can find a recovery transition. The iteration stops when all the deadlock markings are recovered. Finally, the number of recovery transitions is equal to that of iterations. There is a special case: Only one recorded deadlock marking is left, i.e., there is only one deadlock marking that has no shared recovery transition with that of all the other deadlock markings. In this case, any incidence vector $[N_r](P_A, t_r) \in \mathcal{T}_{M_d}$ for the deadlock marking is selected to obtain a recovery transition which can recover the left deadlock marking. In the worst case, n_d recovery transitions are found and the number of the intersections to be computed is $n_d(n_d - 1)/2$. This iterative method may not be able to obtain the minimal number of recovery transitions in theory, but it can usually lead to the minimal one in practice, which is supported by extensive experimental studies compared with the results of MNRTP [8].

A simple example is presented to demonstrate the proposed iterative method in detail. Assume that there are eight deadlock markings, and the results can be drawn through two-step iterations. Fig. 1 shows the specific iterative process by using the presented approach.

In the first iterative step, the intersection of recovery transitions for the first two deadlock markings is not an empty set, but it has no common subset with the set of recovery

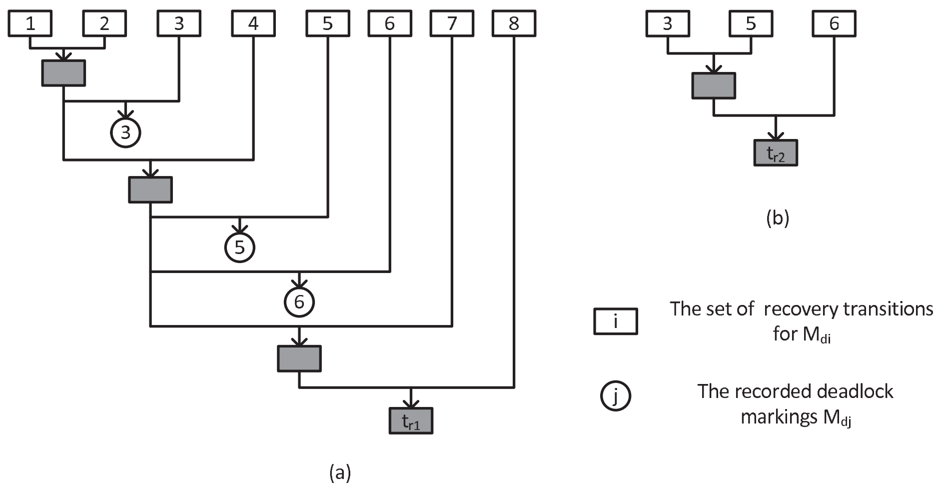


FIGURE 1. **The process of the iterative method.**

transitions for the third deadlock marking. Then, the third deadlock marking is recorded. We compute the common set from the obtained set and the set of recovery transitions for the fourth deadlock marking and so on. The iterative step ends when all the deadlock markings are computed. After that, one recovery transition t_{r1} is found to recover the five deadlock markings ($M_{d1}, M_{d2}, M_{d4}, M_{d7}, M_{d8}$) to be legal markings.

In the second iterative step, we calculate the intersection of the three recorded deadlock markings (M_{d3}, M_{d5}, M_{d6}) separately, using the same method as the first iterative step. As a result, the three deadlock markings can be recovered by a shared recovery transition t_{r2}.

Above all, each iterative step can find one recovery transition to recover multiple deadlock markings, and the number of iterations equals to that of the obtained recovery transitions. To this end, the resulting net model is live after adding two recovery transitions only.

III. DEADLOCK RECOVERY POLICY AND ILLUSTRATIVE EXAMPLE

This section presents a deadlock recovery policy to design a small number of recovery transitions whose set is denoted as $T_r = \{t_{r1}, t_{r2}, t_{r3}, \ldots, t_{rm}\}$. A simple example is also provided to demonstrate the presented approach in detail.

A deadlock recovery policy is presented in Algorithm 1, which is an iterative process. At each iteration step, we can calculate one recovery transition to recover multiple deadlock markings. Then, at the end of each iteration, the recovered deadlock markings are removed from the set of deadlock markings. The iteration process ends when the set of deadlock markings is empty. Finally, a small number of recovery transitions is found to recover all of the deadlock markings.

In what follows, a simple example is presented to demonstrate the proposed approach in detail. Fig. 2 shows a Petri net model of an FMS. All places are divided into three parts: $P^0 = \{p_1, p_8\}$, $P_R = \{p_9 - p_{11}\}$, and $P_A = \{p_2 - p_7\}$. According to the reachability analysis, the Petri net model has 20 reachable markings, 15 of which are legal markings, and 2 of which are

---

Algorithm 1 Computation of Recovery Transitions

Input: A Petri net model (N, M_0) of an FMS with $N = (P^0 \cup P_A \cup P_R, T, F, W)$.

Output: A live Petri net system (N_r, M_0).

1) Compute the set of legal markings $\mathcal{M}_L$ and the set of deadlock markings $\mathcal{M}_D$ of (N, M_0).
2) Compute the minimal covering set of legal markings $\mathcal{M}_L^* \subseteq \mathcal{M}_L$.
3) $T_r := \emptyset$. /* T_r is used to denote the set of recovery transitions*/
4) **while** $|\mathcal{M}_{D_r}| > 1$ **do** /* $\mathcal{M}_{D_r}$ is used to denote the set of recorded deadlock markings */
   Calculate the intersections of $\{\mathcal{T}_{M_{d_1}}, \mathcal{T}_{M_{d_2}}, \ldots, \mathcal{T}_{M_{d_m}}\}$ in turn.
   Let x_i's ($\forall p_i \in P_A$) be the solution.
   Compute recovery transition t_{r_j}.
   $T_r := T_r \cup \{t_{r_j}\}$ and $\mathcal{M}_{D_r} := \mathcal{M}_D - \Omega(tr_j)$.
   endwhile
5) **if** $|\mathcal{M}_{D_r}| = 1$ **then**
   $T_r := T_r \cup \{t_{r_j}\}$
   end if
6) Add all recovery transitions in T_r to (N, M_0) and denote the resulting net system as (N_r, M_0).
7) Output (N_r, M_0).
8) End.

---

deadlock markings with $\mathcal{M}_D = \{p_2+p_3+p_5, p_2+p_5+p_6\}$. The minimal covering set of legal markings contains two elements, i.e., $\mathcal{M}_L^* = \{p_2 + p_3 + p_4, p_5 + p_6 + p_7\}$.

Let t_r be a recovery transition with $[N_r](P, t_r) = [x_{r1}, x_{r2}, x_{r3}, \ldots, x_{r11}]$. In fact, we can calculate it in three steps. The first is the computation of x_i's (for all $p_i \in P_A$). Only one recovery transition can be obtained by using Algorithms 1, expressed as $[N_r](P_A, t_r) = [x_2, x_3, \ldots, x_7]^T = [-1, 0, 0, -1, 0, 0]^T$. Next, according to Assumptions 1 and 2, we can compute x_i's (for all $p_{id} \in P^0$)

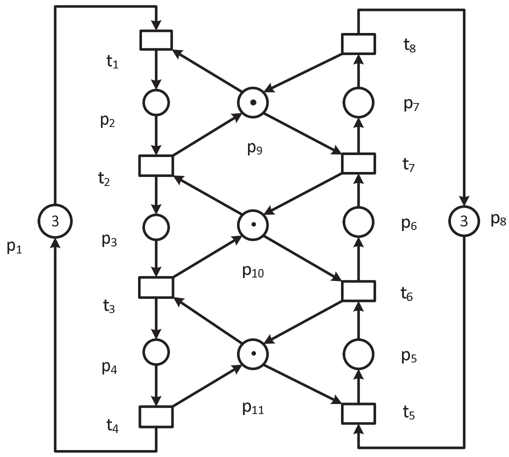


FIGURE 2. Petri net model of an FMS.

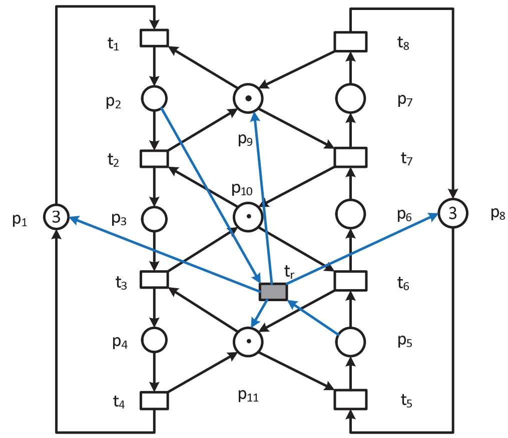


FIGURE 3. Petri net model with a recovery transition.

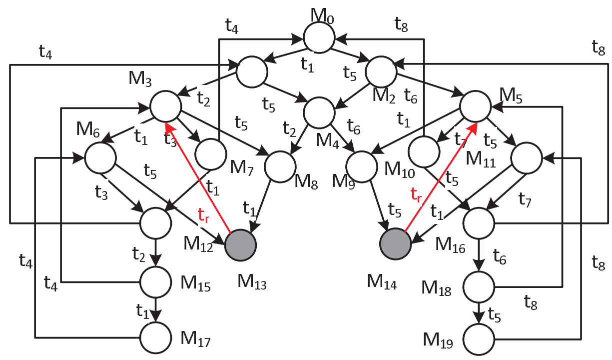


FIGURE 4. Reachability graph with a recovery transition.

and x_i's (for all $p_r \in P_R$), i.e., $[N_r](P^0, t_r) = [x_1, x_8]^T$ and $[N_r](P_R, t_r) = [x_9, x_{10}, x_{11}]^T$ by $[N_r](P_A, t_r)$. There are five place invariants for idle and resource places, as shown below:

$$I_1 = [1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0]^T$$
$$I_8 = [0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0]^T$$
$$I_9 = [0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0]^T$$
$$I_{10} = [0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0]^T$$
$$I_{11} = [0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1]^T$$

Here, we take x_1 as an example. We have $I_1(p_1) = 1$, and $I_1(P_A) = [1, 1, 1, 0, 0, 0]^T$. Hence, $x_1 = -I_1(P_A)^T \cdot [N_r](P_A, t_r)/(I_1(p_1)) = -[1, 1, 1, 0, 0, 0] \cdot [-1, 0, 0, -1, 0, 0]^T = 1$. Similarly, x_8, x_9, x_{10}, and x_{11} are computed as follows:

$$\begin{aligned} x_8 &= -I_8(P_A)^T \cdot [N_r](P_A, t_r)/(I_8(p_8)) \\ &= -[0, 0, 0, 1, 1, 1] \cdot [-1, 0, 0, -1, 0, 0]^T = 1 \\ x_9 &= -I_9(P_A)^T \cdot [N_r](P_A, t_r)/(I_9(p_9)) \\ &= -[1, 0, 0, 0, 0, 1] \cdot [-1, 0, 0, -1, 0, 0]^T = 1 \\ x_{10} &= -I_{10}(P_A)^T \cdot [N_r](P_A, t_r)/(I_{10}(p_{10})) \\ &= -[0, 1, 0, 0, 1, 0] \cdot [-1, 0, 0, -1, 0, 0]^T = 0 \\ x_{11} &= -I_{11}(P_A)^T \cdot [N_r](P_A, t_r)/(I_{11}(p_{11})) \\ &= -[0, 0, 1, 1, 0, 0] \cdot [-1, 0, 0, -1, 0, 0]^T = 1 \end{aligned}$$

Finally, the complete recovery transition t_r is represented as $[N_r](P, t_r) = [1, -1, 0, 0, -1, 0, 0, 1, 1, 0, 1]^T$. The resulting net model is shown in Fig. 3 after adding recovery transition t_r, and the reachability graph with t_r is also shown in Fig. 4. The two deadlock markings M_{13} and M_{14} are guided to the legal markings M_3 and M_5, respectively, by firing the obtained recovery transition t_r. Finally, there is no deadlock marking in the resulting net model system (N_r, M_0).

IV. EXPERIMENTAL RESULTS

A widely used FMS example is adopted to demonstrate the proposed deadlock recovery policy, as shown in Fig. 5 (see [2], [3], [5], [23], [26], [27]). There are 19 places and 14 transitions with 282 reachable markings, 205 and 26 of which are legal and deadlock markings, respectively. The minimal covering set of legal markings $\mathcal{M}_L^*$ contains 26 markings. All places can be divided into three parts: $P^0 = \{p_1, p_8\}$, $P_R = \{p_{14} - p_{19}\}$, and $P_A = \{p_2 - p_7, p_9 - p_{13}\}$.

TABLE 1. Experiment results for the net model in Fig. 6.

j	$\lvert\Omega(t_{rj})\rvert$	${}^\bullet t_{rj}$	$t_{rj}^\bullet$	T
1	7	p_3, p_9, p_{10}	$p_1, 2p_8, p_{15}, p_{17}, p_{19}$	
2	3	p_5, p_6, p_9, p_{10}	$2p_1, 2p_8, p_{16}, p_{17}, p_{18}, p_{19}$	$< 2s$
3	6	p_9, p_{10}, p_{12}	$3p_8, p_{15}, p_{17}, p_{19}$	

By applying Algorithm 1, the obtained results are shown in Table 1, where j is the number of iterations, $|\Omega(t_{rj})|$ is the number of deadlock markings recovered by t_{rj}, ${}^\bullet t_{rj}$ and $t_{rj}^\bullet$ are the preset and postset of t_{rj}, and the last column T shows the time of computation for Algorithm 1. As a result, three recovery transitions are obtained to recover all deadlock markings, i.e., the net model is live with 282 reachable markings after adding three recovery transitions, which is shown in Fig. 6.

Huang *et al.* [18] develop a transition based deadlock recovery policy for a subclass of Petri nets called S^3PR. In terms of structural complexity, the final results of this work include seven recovery transitions, which is four more than our work. Chen *et al.* [8] propose two transition-based

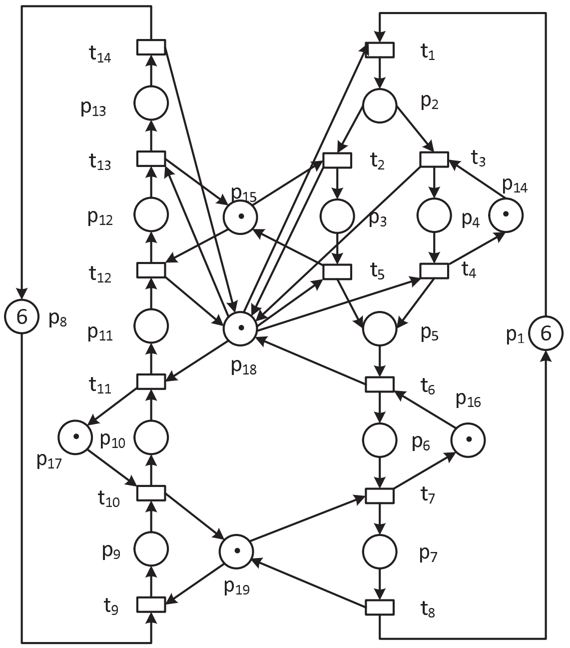


FIGURE 5. Petri net model of an FMS.

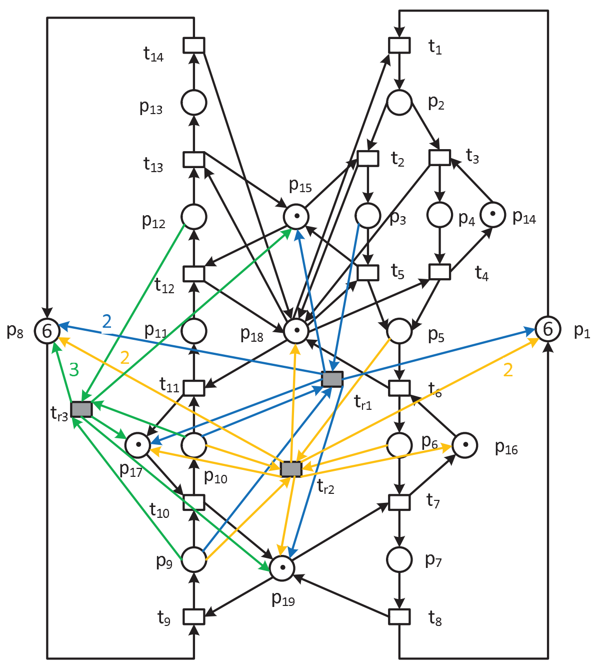


FIGURE 6. Petri net model with recovery transitions.

deadlock recovery policies by solving ILPPs. The former is called the maximal number of recovered deadlock markings problem (MNRDMP), and the latter is the minimal number of recovery transitions problem (MNRTP). Both ILPPs obtain three recovery transitions. In fact, as the size of the net model increases, the numbers of constraints and variables will increase dramatically, and the calculation time increases accordingly. The proposed policy in Algorithm 1 is an iterative process which costs less time to obtain the

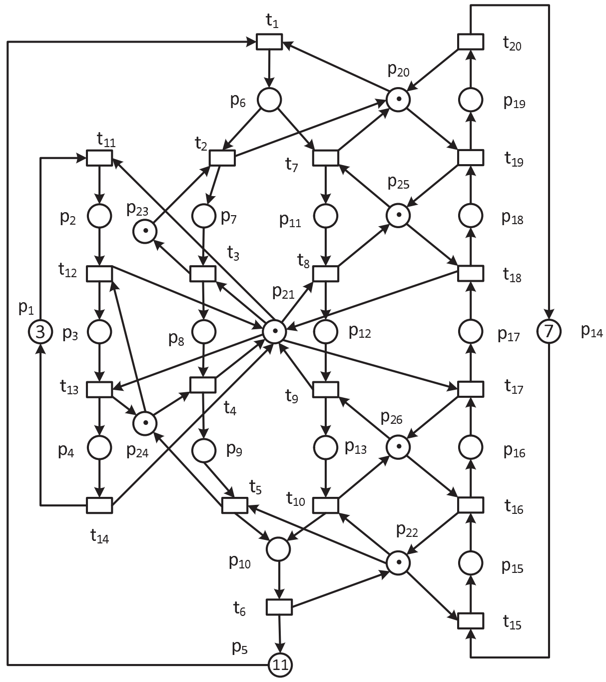


FIGURE 7. A Petri net model of an FMS.

TABLE 2. Comparison of some transition-controlled deadlock recovery policies.

Parameters	MNRDMP [8]	MNRTP [8]	Alg. 1
No. transitions	3	3	3
No. arcs	25	27	25
No. states	282	282	282
Time	30 s	45 h	<2 s

TABLE 3. Constraints and variables in [8].

methods	no.constraints	no.variables
MNRDMP	41496	1336
MNRTP	166104	5348

result than solving ILPPs. Table 2 gives the comparison of these deadlock recovery solutions. The last row shows the computational time of each method. It is indicated that the proposed method is the most efficient since it takes much less time to find the results. More importantly, it can usually find the minimal number of recovery transitions compared with MNRTP [8] and lead to the same number of reachable markings.

In fact, the work in [8] is inapplicable to some complicated Petri net models with too many deadlock markings and legal markings since there are a large number of constraints and variables in the ILPPs. Fig. 7 shows a Petri net model with 1650 reachable markings, 998 and 24 of which are legal and deadlock markings, respectively. The minimal covering set of legal markings $\mathcal{M}_L^*$ contains 54 markings. Table 3 shows the number of constraints and variables in the ILPPs proposed in [8] for this example. Obviously, lots of constraints and vari-

TABLE 4. Results of Fig. 7 by Algorithm 1.

j	$\|\Omega(t_{rj})\|$	$^\bullet t_{rj}$	$t_{rj}^\bullet$	T
1	8	$p_6, p_7, p_{11}, p_{15}, p_{16}$	$3p_5, 2p_{14}, p_{20}, p_{22}, p_{23}, p_{25}, p_{26}$	
2	5	$p_3, p_6, p_7, p_{11}, p_{15}$	$p_1, 3p_5, p_{14}, p_{20}, p_{22}, p_{23}, p_{24}, p_{25}$	$3s$
3	9	p_6, p_7, p_{15}	$2p_5, p_{14}, p_{20}, p_{22}, p_{23}$	
4	2	$p_3, p_6, p_7, p_{15}, p_{16}, p_{18}$	$p_1, 2p_5, 3p_{14}, p_{20}, p_{22}, p_{23}, p_{24}, p_{25}, p_{26}$	

TABLE 5. Results of modified net model by Algorithm 1.

j	$\|\Omega(t_{rj})\|$	$^\bullet t_{rj}$	$t_{rj}^\bullet$	T
1	64	$p_6, 2p_7, p_{11}, p_{15}, p_{16}$	$4p_5, 2p_{14}, p_{20}, p_{22}, 2p_{23}, p_{25}, p_{26}$	
2	22	$p_3, p_6, 2p_7, p_{11}, p_{15}$	$p_1, 4p_5, p_{14}, p_{20}, p_{22}, 2p_{23}, p_{24}, p_{25}$	$02:04:30$
3	30	$p_6, 2p_7, p_{15}$	$3p_5, p_{14}, p_{20}, p_{22}, 2p_{23}$	
4	4	$2p_3, p_6, 2p_7, p_{15}, p_{16}, 2p_{18}$	$2p_1, 3p_5, 4p_{14}, p_{20}, p_{22}, 2p_{23}, 2p_{24}, 2p_{25}, p_{26}$	

ables lead to a rather complicated ILPPs, and it is practically impossible to find a solution in a reasonable time. However, our work is more efficient and structurally simpler. According to Algorithm 1, the results are shown in Table 4. It takes only three seconds to obtain four recovery transitions to recover all deadlock markings.

If we modify the number of tokens in the last four places ($p_{23}, p_{24}, p_{25}, p_{26}$) in Fig. 7, with the initial marking $M_0 = [3, 0, 0, 0, 11, 0, 0, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 2, 2]^T$ [8], [26], [43], the net model is more complicated with 26750 reachable markings, where the number of deadlock markings is 120 and the minimal covering set of legal markings $\mathcal{M}_L^*$ contains 393 markings. The results of the proposed method are shown in Table 5.

For this modified net model, four recovery transitions can be obtained by using Algorithm 1 in about two hours. However, the work in [8] cannot find a solution for this example since there are too many constraints and variables in the designed ILPPs. Combined with the above examples, it is verified that our work is more efficient.

V. CONCLUSIONS

This work deals with the deadlock problems in a Petri net modeling FMSs by adding recovery transitions, which can make all deadlock markings reachable to the legal markings. A vector intersection approach is proposed to compute a recovery transition to recover multiple deadlocks. Then, an iterative algorithm is presented to find a small number of recovery transitions to recover all deadlock markings. Compared with the previous work, the proposed approach is more efficient since it does not need to solve ILPPs. More importantly, the experimental results show that it usually finds the minimal number of the recovery transitions compared with the method of MNRTP [8]. In the future, we will improve the algorithm so as to guarantee the structural minimality. On the other hand, we aim to avoid computing the reachability graph of a net model to improve the efficiency of the proposed method.

REFERENCES

[1] Z. Li, G. Liu, H.-M. Hanisch, and M. Zhou, "Deadlock prevention based on structure reuse of Petri net supervisors for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 1, pp. 178–191, Jan. 2012.

[2] Y. Chen, Z. Li, M. Khalgui, and O. Mosbahi, "Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 2, pp. 374–393, Apr. 2011.

[3] Y. Chen and Z. Li, "Design of a maximally permissive liveness-enforcing supervisor with a compressed supervisory structure for flexible manufacturing systems," *Automatica*, vol. 47, no. 5, pp. 1028–1034, May 2011.

[4] Y. Chen and Z. Li, "On structural minimality of optimal supervisors for flexible manufacturing systems," *Automatica*, vol. 48, no. 10, pp. 2647–2656, Oct. 2012.

[5] Y. Chen, Z. Li, and A. Al-Ahmari, "Nonpure Petri net supervisors for optimal deadlock control of flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 43, no. 2, pp. 252–265, Mar. 2013.

[6] Y. Chen, Z. Li, and M. Zhou, "Optimal supervisory control of flexible manufacturing systems by Petri nets: A set classification approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 2, pp. 549–563, Apr. 2014.

[7] Y. Chen, Z. Li, K. Barkaoui, and M. Uzam, "New Petri net structure and its application to optimal supervisory control: Interval inhibitor arcs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 10, pp. 1384–1400, Oct. 2014.

[8] Y. Chen, Z. Li, A. Al-Ahmari, N. Q. Wu, and T. Qu, "Deadlock recovery for flexible manufacturing systems modeled with Petri nets," *Inf. Sci.*, vol. 381, pp. 290–303, Mar. 2017.

[9] Y. F. Chen and Z. W. Li, "Basics of Petri nets in the logic control of resource allocation systems," School Electro-Mech. Eng., Xidian Univ., Xi'an, China, Tech. Note SCATN01, 2017. [Online]. Available: https://sites.google.com/site/zhiwulisys/notes

[10] M. P. Fanti and M. Zhou, "Deadlock control methods in automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 34, no. 1, pp. 5–22, Jan. 2004.

[11] G. Liu, P. Li, Z. Li, and N. Wu, "Robust deadlock control for automated manufacturing systems with unreliable resources based on Petri net reachability graphs," *IEEE Trans. Syst., Man, Cybern. Syst.*, to be published, doi: 10.1109/TSMC.2018.2815618.

[12] J. Ezpeleta, F. Tricas, F. Garcia-Valles, and J. M. Colom, "A Banker's solution for deadlock avoidance in FMS with flexible routing and multiresource states," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 621–625, Aug. 2002.

[13] J. Ezpeleta, J. M. Colom, and J. Martinez, "A Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 173–184, Apr. 1995.

[14] J. Ezpeleta and L. Recalde, "A deadlock avoidance approach for non-sequential resource allocation systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 34, no. 1, pp. 93–101, Jan. 2004.

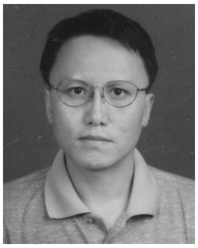
[15] M. Zhou, F. DiCesare, and A. A. Desrochers, "A hybrid methodology for synthesis of Petri net models for manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 350–361, Jun. 1992.

[16] F.-S. Hsieh, "Fault-tolerant deadlock avoidance algorithm for assembly processes," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 34, no. 1, pp. 65–79, Jan. 2004.

[17] Y.-S. Huang, M. Jeng, X. Xie, and D.-H. Chung, "Siphon-based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 36, no. 6, pp. 1248–1256, Nov. 2006.
[18] Y.-S. Huang, Y.-L. Pan, and P.-J. Su, "Transition-based deadlock detection and recovery policy for FMSs using graph technique," *ACM Trans. Embedded Comput. Syst.*, vol. 12, no. 1, pp. 11.1–11.13, Jan. 2013.
[19] M. D. Jeng and X. L. Xie, "Deadlock detection and prevention of automated manufacturing systems using Petri nets and siphons," in *Deadlock Resolution Computer-Integrated System*. New York, NY, USA: CRC Press, 2004, pp. 233–281.
[20] T. K. Kumaran, W. Chang, H. Cho, and R. A. Wysk, "A structured approach to deadlock detection, avoidance and resolution in flexible manufacturing systems," *Int. J. Prod. Res.*, vol. 32, no. 10, pp. 2361–2379, Oct. 1994.
[21] M. Zhou and N. Wu, *System Modeling and Control with Resource-Oriented Petri Nets*. Boca Raton, FL, USA: CRC Press, 2009.
[22] M. C. Zhou and F. DiCesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Berlin, Germany: Springer, 2012.
[23] Z. Ma, Y. Tong, Z. Li, and A. Giua, "Basis marking representation of Petri net reachability spaces and its application to the reachability problem," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1078–1093, Mar. 2017.
[24] T. Murata, "Petri nets: Properties, analysis and application," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.
[25] J. Park and S. A. Reveliotis, "Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings," *IEEE Trans. Autom. Control*, vol. 46, no. 10, pp. 1572–1583, Oct. 2001.
[26] L. Piroddi, R. Cordone, and I. Fumagalli, "Selective siphon control for deadlock prevention in Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 6, pp. 1337–1348, Nov. 2008.
[27] L. Piroddi, R. Cordone, and I. Fumagalli, "Combined siphon and marking generation for deadlock prevention in Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 3, pp. 650–661, May 2009.
[28] F. Tricas and J. Martinez, "An extension of the liveness theory for concurrent sequential processes competing for shared resources," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Oct. 1995, pp. 3035–3040.
[29] V. D. Gligor and S. H. Shattuck, "On deadlock detection in distributed systems," *IEEE Trans. Softw. Eng.*, vol. 6, no. 5, pp. 435–440, Sep. 1980.
[30] A. Ghaffari, N. Rezg, and X. Xie, "Design of a live and maximally permissive Petri net controller using the theory of regions," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 137–142, Feb. 2003.
[31] Z. Ma, Z. Li, and A. Giua, "Characterization of admissible marking sets in Petri nets with conflicts and synchronizations," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1329–1341, Mar. 2017.
[32] N. Wu, M. Zhou, and Z. Li, "Short-term scheduling of crude-oil operations: Enhancement of crude-oil operations scheduling using a Petri net-based control-theoretic approach," *IEEE Robot. Autom. Mag.*, vol. 22, no. 2, pp. 64–76, Jun. 2015.
[33] R. A. Wysk, N.-S. Yang, and S. Joshi, "Resolution of deadlocks in flexible manufacturing systems: Avoidance and recovery approaches," *J. Manuf. Syst.*, vol. 13, no. 2, pp. 128–138, 1994.
[34] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, Jul. 1987.
[35] P. J. G. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proc. IEEE*, vol. 77, no. 1, pp. 81–89, Jan. 1989.
[36] J. Zhang, M. Khalgui, Z. Li, G. Frey, O. Mosbahi, and H. B. Salah, "Reconfigurable coordination of distributed discrete event control systems," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 323–330, Jan. 2015.
[37] J. Ye, Z. Li, and A. Giua, "Decentralized supervision of Petri nets with a coordinator," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 6, pp. 955–966, Jun. 2015.
[38] M. Uzam and M. C. Zhou, "An improved iterative synthesis method for liveness enforcing supervisors of flexible manufacturing systems," *Int. J. Prod. Res.*, vol. 44, no. 10, pp. 1987–2030, May 2006.
[39] X. Wang, I. Khemaissia, M. Khalgui, and Z. Li, O. Mosbahi, and M. C. Zhou, "Dynamic low-power reconfiguration of real-time systems with periodic and probabilistic tasks," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 1, pp. 258–271, Jan. 2015.
[40] G. Zhu, Z. Li, and N. Wu, "Model-based fault identification of discrete event systems using partially observed Petri nets," *Automatica*, vol. 96, pp. 201–212, Oct. 2018.
[41] X. Cong, M. P. Fanti, A. M. Mangini, and Z. W. Li, "On-line verification of current-state opacity by Petri nets and integer linear programming," *Automatica*, vol. 94, pp. 205–213, Aug. 2018.
[42] M. Uzam, Z. Li, G. Gelen, and R. S. Zakariyya, "A divide-and-conquer-method for the synthesis of liveness enforcing supervisors for flexible manufacturing systems," *J. Intell. Manuf.*, vol. 27, no. 5, pp. 1111–1129, Oct. 2016.
[43] Y. Chen, Z. Li, K. Barkaoui, and A. Giua, "On the enforcement of a class of nonlinear constraints on Petri nets," *Automatica*, vol. 55, pp. 116–124, May 2015.
[44] X. Y. Cong, M. P. Fanti, A. M. Mangini, and Z. W. Li, "Decentralized diagnosis by Petri nets and integer linear programming," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 10, pp. 1689–1700, Oct. 2018.
[45] H. Zhang, L. Feng, N. XWu, and Z. Li, "Integration of learning-based testing and supervisory control for requirements conformance of black-box reactive systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 2–15, Jan. 2018.
[46] H. Zhang, L. Feng, and Z. Li, "A learning-based synthesis approach to the supremal nonblocking supervisor of discrete-event systems," *IEEE Trans. Autom. Control*, vol. 63, no. 10, pp. 3345–3360, Oct. 2018.
[47] G. Zhu, Z. Li, N. Wu, and A. Al-Ahmari, "Fault identification of discrete event systems modeled by Petri nets with unobservable transitions," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 2, pp. 333–345, 2019.
[48] S. G. Wang, D. You, and C. Seatzu, "A novel approach for constraint transformation in Petri nets with uncontrollable transitions," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 8, pp. 1403–1410, Aug. 2018.
[49] D. You, S. Wang, W. Dai, W. Wu, and Y. Jia, "An approach for enumerating minimal siphons in a subclass of Petri nets," *IEEE Access*, vol. 6, pp. 4255–4265, 2018.
[50] C. Gu, Z. Li, N. Wu, M. Khalgui, T. Qu, and A. Al-Ahmari, "Improved multi-step look-ahead control policies for automated manufacturing systems," *IEEE Access*, vol. 6, no. 1, pp. 68824–68838, 2018.
[51] Z. He, Z. Li, and A. Giua, "Cycle time optimization of deterministic timed weighted marked graphs by transformation," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 4, pp. 1318–1330, Jul. 2017.
[52] Z. He, Z. W. Li, and A. Giua, "Performance optimization for timed weighted marked graphs under infinite server semantics," *IEEE Trans. Autom. Control*, vol. 63, no. 8, pp. 2573–2580, Aug. 2018.
[53] N. Ran, S. Wang, and W. Wu, "Event feedback supervision for a class of Petri nets with unobservable transitions," *IEEE Access*, vol. 6, pp. 6920–6926, 2018.
[54] X. Zhang and M. Uzam, "Transition-Based deadlock control policy using reachability graph for flexible manufacturing systems," *Adv. Mech. Eng.*, vol. 8, no. 2, pp. 1–9, Feb. 2016.
[55] M. Bashir, D. Liu, M. Uzam, N. Q. Wu, A. Al-Ahmari, and Z. Li, "Optimal enforcement of liveness to flexible manufacturing systems modeled with Petri nets via transition-based controllers," *Adv. Mech. Eng.*, vol. 10, no. 1, pp. 1–19, 2018.
[56] C. Gu, X. Wang, and Z. W. Li, "Synthesis of supervisory control with partial observation on normal state tree structures," *IEEE Trans. Autom. Sci. Eng.*, to be published, doi: 10.1109/TASE.2018.2880178.
[57] Z. W. Li and M. C. Zhou, "Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 34, no. 1, pp. 38–51, 2004.
[58] Z. W. Li and M. C. Zhou, "Two-stage method for synthesizing liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *IEEE Trans. Ind. Informat.*, vol. 2, no. 4, pp. 313–325, 2006.
[59] Z. W. Li and M. C. Zhou, "Control of elementary and dependent siphons in Petri nets and their application," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 1, pp. 133–148, 2008.
[60] Z. W. Li, M. C. Zhou, and M. D. Jeng, "A maximally permissive deadlock prevention policy for FMS based on Petri net siphon control and the theory of regions," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 1, pp. 182–188, 2008.
[61] Z. W. Li and M. Zhao, "On controllability of dependent siphons for deadlock prevention in generalized Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 2, pp. 369–384, 2008.
[62] Z. W. Li and M. C. Zhou, *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*. Berlin, Germany: Springer, 2009.
[63] Y. Qiao, N. Q. Wu, F. J. Yang, M. C. Zhou, and Q. H. Zhu, "Wafer sojourn time fluctuation analysis of time-constrained dual-arm cluster tools with wafer revisiting and activity time variation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 4, pp. 622–636, Apr. 2018.

YUNYUN DONG received the B.S. degree in automation from the Xi'an University of Architecture and Technology, Xi'an, China, in 2016. She is currently pursuing the M.S. degree in control engineering with Xidian University. Her research interests include Petri nets and deadlock control of discrete event systems.

YUFENG CHEN (M'13–SM'17) received the B.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2006 and 2011, respectively. He has authored or co-authored over 20 publications. Along with Z. Li, he has co-authored the book *Optimal Supervisory Control of Automated Manufacturing Systems* (CRC Press, Taylor & Francis Group, 2013). He is with the School of Electro-Mechanical Engineering, Xidian University. His research interests include Petri net theory and applications, supervisory control of discrete event systems, and scheduling flexible manufacturing systems.



SHAOYONG LI received the Ph.D. degree from Xidian University, Xi'an, China, in 2012. He is currently a Professor as well as a Master's Supervisor with the School of Civil Engineering, Lanzhou University of Technology. His research interests include control engineering and application of supervisory control of discrete event systems and deadlock prevention algorithm using Petri net theory.

MOHAMMED A. EL-MELIGY received the B.S. degree in information technology from the Menoufia University of Egypt, in 2005. He has been a Software Engineer with King Saud University, Riyadh, Saudi Arabia, since 2009. His research interests include Petri nets, supervisory control of discrete event systems, database software, and network administration.

MOHAMED SHARAF received the Ph.D. degree in industrial engineering from Chiba University, Japan. He is the Head of the Development and Quality Unit, College of Engineering, King Saud University. He has published more than 30 papers in the areas of spare parts control, quality management, maintenance, six sigma methodology, and academic accreditation.