IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 10, 2021, accepted April 11, 2021, date of publication April 16, 2021, date of current version May 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3073639

Fault-Recovery and Robust Deadlock Control of Reconfigurable Multi-Unit Resource Allocation Systems Using Siphons

MAHMOUD SALAHELDIN ELSAYED[1,2], GAIYUN LIU[3], (Senior Member, IEEE), ALMETWALLY M. MOSTAFA[4,5], ABEER ALI ALNUAIM[6], AND PASSENT EL KAFRAWY[2,7], (Senior Member, IEEE)

[1]Department of Computer Science, Faculty of Computing and Information Technology, Northern Border University, Rafha 76321, Saudi Arabia
[2]Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Shibin El-Kom 32511, Egypt
[3]School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China
[4]Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11421, Saudi Arabia
[5]Systems and Computers Department, Faculty of Engineering, Al-Azhar University, Cairo 11371, Egypt
[6]Department of Natural and Engineering Sciences, College of Applied Studies and Community Service, King Saud University, Riyadh 11421, Saudi Arabia
[7]School of Information Technology and Computer Science, Nile University, Giza 12588, Egypt

Corresponding author: Mahmoud Salaheldin Elsayed (mahmoud.al-sayd@nbu.edu.sa)

This work was supported by the Deanship of Scientific Research at King Saud University under Grant RG-1441-425 and in part by the Natural Science Foundation of Shaanxi Province under Grant 2019JM-049.

ABSTRACT A multi-unit resource allocation system usually contains several processes and a number of resources with multiple units. Due to the competition for shared resources in these systems, deadlocks may occur. Recently, researchers have shown an increased awareness in deadlock control strategies for such a kind of systems without considering the dynamic changes such as processing failures and rework by using the Petri net paradigm. This article reports a new strategy for deadlock analysis and control in reconfigurable multi-unit resource systems (MRSs). We discuss a generalized class of Petri nets in which each stage of a process may require a number of units of different types of resources to perform a task. In this way, we can model more complex real systems. Thanks to a generalized class of Petri nets, i.e., the system of sequential systems with shared resources (S^4R), this article proposes an effective integrated strategy for designing robust supervisors for reconfigurable MRSs, and improves an S^4R model to achieve a new model, namely a system of sequential systems with shared resources and part-re-entry (S^4RP), which represents the procedure that a flawed product re-enters a system and is re-processed. We use a siphon-based max-controllability deadlock prevention policy (DPP) to supervise the evolution of the S^4RP, and present a comprehensive analysis to demonstrate that the controlled S^4RP is free of deadlocks. A net analysis tool (INA) is used to test and validate the resulting S^4RP.

INDEX TERMS Multi-unit resource system (MRS), Petri net, reconfiguration, robust deadlock control, max-controllability of siphon.

I. INTRODUCTION

From the viewpoint of resource capacity, there are two distinct types of resource allocation systems: single-unit resource systems (SRSs) and multi-unit resource systems (MRSs). The two kinds of systems usually consist of several processes and a limited number of resources, where the difference between them is the capacity of resources. In a single-unit resource system, each resource has only one unit of that kind to be allocated to distinctive processes; on the other hand, in an MRS, a resource may equip with any number of units. The possibility of a system being in a deadlock state significantly depends on the competition for shared resources among different processes [1].

A deadlock is actually a status in a system where two or more processes wait for resources occupied by others to finish, and neither ever does. In an SRS, if a cycle is formed in its resource allocation graph that represents relationships among the processes and resources, a deadlock necessarily occurs. The issue of deadlock is increasingly challenging in

The associate editor coordinating the review of this manuscript and approving it for publication was Yue Zhang.

an MRS since, for instance, the existence of a cycle in the resource allocation graph of an MRS does not necessarily imply the occurrence of a deadlock.

Many researchers and practitioners use Petri nets (PNs) for modelling and managing deadlocks in single or multi-unit resource systems [2]–[6]. PNs serve as a common formalism due to their inherent characteristics [7]–[17]. Deadlock control methods can be classified into three types: Deadlock detection and recovery [18], [19], deadlock avoidance [20], [21], and deadlock prevention [22], [23]. Deadlock prevention acts in an off-line way such that a group of constraints among the resources and processes in a system is imposed by an external agent, called a supervisor, thus making the resource requests that cause deadlocks be impossible. Deadlock avoidance offers an on-line computational mechanism. Whenever a deadlock state is recognized, a predefined mechanism helps the system return to a correct state. Deadlock prevention is usually thought of as being safer than other strategies [24]–[30]. For more details of Petri nets and their applications, readers are referred to [31]–[34].

The resources in manufacturing systems (a common example of MRSs) can have a multitude of forms, including a variety of equipment such as numerically controlled machine tools. The competitiveness of manufacturing companies depends not only by the high productivity of the manufacturing systems, but also by many factors from the market and customers. Reconfigurable manufacturing systems (RMSs) [35]–[37] offer the capability of an automatic and rapid response to the variation of the global market. Due to the complexity in its structure and configuration, the deadlock problem in an RMS remains a challenging issue for researchers and practitioners.

The existence of unreliable resources in an RMS is a significant factor contributing to deadlock occurrences. It is well-known that unreliable resources in RMSs may result in partial system collapse or even deadlocks due to the unavailability of some processing units in a resource [38]–[41]. Based on Petri nets, multiple strategies are introduced to make complex improvements in reconfigurable manufacturing systems [42]–[45]. However, most of them do not have a reconfiguration algorithm and cannot guarantee the behavioural properties of Petri nets (boundness, liveliness, and conservativeness).

To manage deadlocks within a reconfigurable MRS, we proposed in [46] a new net subclass, namely a simple sequential process with resources and part re-entry (S^2PRP), which captures the dynamics of machine processing and part re-entry to a system by deciding whether a final product satisfies the desired quality. If not, a flawed part has to be sent to the system again for further processing. However, in [46] the process can acquire only one unit of resource at a time, implying that the developed technique in [46] has a limited application scope.

This paper aims to mitigate the technical issue in [46] by expanding the reported strategy in it. We first create a new net subclass called a system of sequential systems with shared resources and part-re-entry (S^4RP). The major difference between the work in this paper and in [46] is that, this paper addresses a generalized class of Petri nets (S^4R) [47] in which each stage of a process may require multiple units of multiple types of resources for executing a task. The well-known Petri nets (PNs) S^3PR [48], [49] and WS^3PR [50] are both proper subclasses of S^4Rs.

In the literature, there are many net classes. Primarily, PNs can be either ordinary or generalized, depending on the weight of arcs. A PN is said to be ordinary when all arc-weights are one; a PN is said to be generalized if it has at least an arc with the weight being two or more. The structural approaches via siphon-control [11], [27], [51]–[53] are typical vehicles to derive deadlock prevention policies (DPPs) for generalized PNs. The occurrences of deadlocks in a generalized PN stem from the existence of siphons with insufficient tokens. It has been shown that deadlocks in a generalized PN could be managed or eliminated by the use of monitors (control places). However, the supervisors due to siphon control have several technical flaws such as structural complexity if the size of a plant is large. The notion of elementary siphons is introduced in [54], [55] and used for the design of liveness-enforcing-supervisors (LESs) with a simple structure.

For the modeling and analysis of manufacturing systems, a few PN subclasses have been reported, which are analyzed by means of siphons [56]–[62]. These findings could be traced to the seminal research in [48]. The proposed PN model (S^4RP) in our research is a failure-safe model. A methodology for fault recovery and re-entry of a faulty part is presented, where a single working process is not in deadlock, which guarantees that the S^4RP remains active even if an unreliable processing unit fails. Thus, the S^4RP is always live.

This study introduces a new control technique in order to derive an LES for RMSs. It reports an improved class of generalized PNs, called an S^4RP, which exposes how a failed part re-enters a system and is re-processed. We also employ a siphon-based max-controllability DPP to control the S^4RP. A complete analysis is conducted, showing that the controlled S^4RP is deadlock-free. Note that either inhibitor arcs or enumerating a reachability graph is necessary, which leads to less computational overheads and ensures that all predefined processing steps can be done continuously. We consider all unreliable resources in an MRS and the proposed method is applicable to a complex Petri net model for multiple unreliable resources.

This article is structured as follows. The deadlock problem in an MRS and deadlock control for an S^4R by max-controllability of siphons are reviewed in Section II. A novel generalized net subclass S^4RP is presented in Section III. The proposed model can well represent a part's failure and re-entry to a system when its quality test fails. Section IV introduces robust deadlock control for this net subclass via max-controllability of siphons, where an algorithm and a complete analysis for an illustrative example are given. Finally, conclusions are drawn in Section V. The basic definitions and properties of S^4R used throughout the paper can be found in [65], [66].

II. PRELIMINARIES

A. DEADLOCK IN AN MRS

A resource allocation graph (RAG) [63] is an effective tool to represent resource allocation relationships between processes and resources in an underlying system. An RAG is a digraph (V, E) with V and E being the sets of nodes and directed edges, respectively. Let $V = P \cup Q$ with $P \cap Q = \emptyset$, where P is a set of processes and Q is the set of resources. In this sense, an RAG is a bipartite graph. An edge $d_{ij} = (x_i, y_j)$ is a request edge if $x_i \in P$ and $y_j \in Q$. An edge $d_{ji} = (y_j, x_i)$ is a grant edge if $y_j \in Q$ and $x_i \in P$. In a path $(x_{i1}, y_{j1}), (y_{j1}, x_{i2}), \ldots, (x_{ik}, y_{jk}), \ldots, (y_{js}, x_{is+1})$, each edge is distinct.

The reachable set of a node collects the nodes such that there exists a path from the node to every element in the reachable set. A knot is a non-empty set K of nodes satisfying the fact that the reachable set of each node in K is exactly K [64].

In an MRS, a process may request multiple units of a type of resource, which can be represented in the underlying RAG by multiple edges, each of which is labeled with an integer denoting the number of units assigned. This alternative representation is said to be a weighted RAG (WRAG). Fig. 1 depicts System-on-a-Chip (SoC) as an example of an MRS with five processors and four multi-unit resources. In this example, a process runs on a processor. Fig. 1(b) visualizes a current resource allocation status in a form of WRAG.

Fig. 2 depicts an RAG with knot $K = \{p_1, p_2, p_3, q_1, q_2\}$ that represents a circular wait. A resource in an MRS can be requested and occupied as long as there are enough available units to be allocated. On the other hand, once a single-unit resource is occupied by a process, all the others that request it have to wait until it is released. We accordingly assume that (1) the capacity of a resource is fixed, i.e., it has a fixed number of processing units, and (2) a resource unit is granted without any time delay if it is available.

B. DEADLOCK CONTROL FOR S^4R

Here we first recall a DPP for an S^4R [47], derived from the notion of siphon's max-controllability. Details regarding generalized PNs are taken from [65]. Given a place p, the maximal weight of its output arcs is denoted by $max_{p}\bullet$.

Definition 1 ([65]): Given a marked S^4R net (N, M_0) and a siphon S in N, S is said to be max-marked at a marking $M \in R(N, M_0)$ if there exists a place $p \in S$ such that $M(p) \geqslant max_{p}\bullet$ holds. S is said to be max-controlled if it is max-marked at any marking $M \in R(N, M_0)$. If all strict minimal siphons (SMSs) are max-controlled, a marked S^4R is said to satisfy the max cs-property (controlled-siphon property).

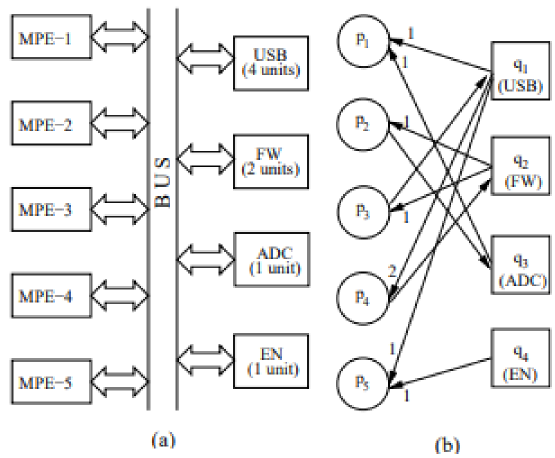

FIGURE 1. SoC example with its corresponding (WRAG).

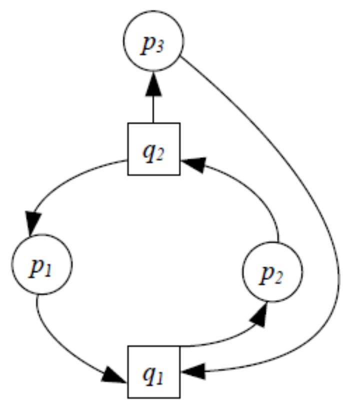

FIGURE 2. Resource allocation Graph (RAG) with a knot.

Theorem 1 ([65]): A marked S^4R net (N, M_0) is live if it satisfies the max cs-property.

Theorem 2 ([65]): Let (N, M_0) be a marked S^4R and S be a siphon. S is max-controlled if

$\exists$ P-invariant I, $\forall p \in \| I \|^- \cap S$, $max_{p}\bullet = 1$, $\| I \|^+ \subseteq S$, $\sum_{p \in P} I(p)M_0(p) > \sum_{p \in S} I(p)(max_{p}\bullet - 1)$.

In what follows, we write PI for a P-invariant. Let S be a siphon in an S^4R that is composed of n state machines. $\forall i \in \mathbb{N}_n = \{1, 2, \ldots, n\}$, $\forall S \in \Pi$, $\wp^s = \bigcup_{i=1}^{n} \wp_s^i$, let $\wp_s^i = [S^i] \cup \{p \in P_A^i \mid p < \overline{N_i}[S^i]\}$. For this S, a non-negative P-vector k_s for S is constructed, and it is assumed that $\forall i \in \mathbb{N}_l$, $[S]^i \neq \emptyset$; $\forall j \in \mathbb{N}_n \setminus \mathbb{N}_l$, $[S]^j = \emptyset$, where $\mathbb{N}_l \subseteq \mathbb{N}_n$. We define $B_s^i = \{p \mid p \in [S]^i, \nexists p' \in [S]^i, p <_{\overline{N_l}} p'\}$ [66].

Note that Π denotes the set of all SMSs in N, and we use $\overline{N_l}$ to represent the S^2P of the i-th S^2PR N_i.

Definition 2 ([66]): A P-vector k_s for a siphon S in an S^4R is constructed as follows:

$\forall p \notin \wp^s$, $k_s(p) := 0$
$\forall p \in [S]$, $k_s(p) := h_s(p)$
$i: = 1$
repeat
$\forall p \in B_s^i$, $\alpha_p := max\{h_s(p), h_s(p') \mid p' < \overline{N_l}p, p' \in [S]^i\}$

$\forall p_x \in \{p \mid p \in B_s^i\} \cup \{p' \mid p' < \overline{N_l} p, p' \in [S]^i\}, k_s(p_x) := \alpha_p$

$\forall p_y \in \{p'' \mid p'' < \overline{N_l} p, p \in B_s^i, p'' \in \wp^s \setminus [S]^i\}, k_s(p_y) := \alpha_p$

$\forall p_z \in \bigcap p_{w \in B_s^i} \{p \mid p \in \wp^s \setminus [S]^i, p < \overline{N_l} p_w\}, k_s(p_z) := max\{k_s(p) \mid p \in B_s^i\}$

$i := i + 1$

until $i \geqslant l + 1$

To demonstrate Definition 2, k_s for the SMS in the net model in Fig. 3 is computed, where $p_1^0 = p_7, p_2^0 = p_{11}, P_A^1 = \{p_1 - p_6\}, P_A^2 = \{p_8 - p_{10}\}, P_R^1 = \{p_{12} - p_{15}\}$, and $P_R^2 = \{p_{12} - p_{14}\}$. It has three SMSs:

$S_1 = \{p_3, p_6, p_9, p_{13}, p_{14}\}$,

$S_2 = \{p_2, p_5, p_{10}, p_{12}, p_{13}\}$, and

$S_3 = \{p_3, p_6, p_{10}, p_{12}, p_{13}, p_{14}\}$.

We have

$\wp_s^1 = \{p_1, p_2, p_5, p_8\}, \wp_s^2 = \{p_1, p_8, p_9\}$, and $\wp_s^3 = \{p_1, p_2, p_5, p_8, p_9\}$.

For S_1, $[S_1] = p_2 + p_5 + p_8, \forall p \notin \wp_s^1, k_s^1(p) = 0, B_s^1 = \{p_2, p_5, p_8\}$.

For $p_2 \in B_s^1$, we have

$\alpha_{p_2} := 1$,

$k_s^1(p_2) := 1$,

$k_s^1(p_1) = k_s^1(p_2) = k_s^1(p_5) = 1$.

For $p_8 \in B_s^1$, we have

$\alpha_{p_8} := 1$,

$k_s^1(p_8) := 1$,

In summary, we have $k_s^1(p_1) = k_s^1(p_2) = k_s^1(p_5) = k_s^1(p_8) = 1$.

For S_2, $[S_2] = 2p_1 + p_9, \forall p \notin \wp_s^2, k_s^2(p) = 0, B_s^2 = \{p_1, p_9\}$.

For $p_1 \in B_s^2$, we have

$\alpha_{p_1} := 2$,

$k_s^2(p_1) := 2$.

For $p_9 \in B_s^2$, we have

$\alpha_{p_9} := 1$,

$k_s^2(p_9) := 1$,

In summary, we have $k_s^2(p_8) = k_s^2(p_9) = 1$, and $k_s^2(p_1) = 2$.

For S_3, $[S_3] = 2p_1 + p_2 + p_5 + p_8 + p_9, \forall p \notin \wp_s^3, k_s^3(p) = 0, B_s^3 = \{p_2, p_5, p_9\}$.

For $p_2 \in B_s^3$, we have

$\alpha_{p_2} := max(1, 2) = 2$,

$k_s^3(p_2) := 2$,

$k_s^3(p_1) = k_s^3(p_2) = 2$.

For $p_5 \in B_s^3$, we have

$\alpha_{p_5} := max(1, 2) = 2$,

$k_s^3(p_5) := 2$.

For $p_9 \in B_s^3$, we have

$\alpha_{p_9} := 1$,

$k_s^3(p_9) := 1$,

$k_s^3(p_8) = k_s^3(p_9) = 1$. In summary, we have $k_s^3(p_8) = k_s^3(p_9) = 1$, and $k_s^3(p_1) = k_s^3(p_2) = k_s^3(p_5) = 2$.

Theorem 3 ([65]): Let S be an SMS in an S^{4}R (N, M_0) with $N = (P_A \cup P^0 \cup P_R, T, F, W)$. A monitor V_S is added to (N, M_0) such that $g_S = k_S + V_S$ is a PI of the resulting net (N_V, M_0^V), where $N_V = (P_A \cup P^0 \cup P_R \cup \{V_S\}, T, F \cup F_V, W \cup W_V)$; $\forall p \in P_A \cup P^0 \cup P_R, M_0^V(p) = M_0(p)$. Let $f_S = \sum_{r \in S^R} I_r - g_S$ and $M_0^V(V_S) = M_0(S) - \xi_S(\xi_S \in \mathbb{N}^+)$. Then, S is max-controlled if $\xi_S > \sum_{p \in S} f_S(p)(max_p^\bullet - 1)$ and $M_0^V(V_S) \geqslant max_{V_S}^\bullet$.

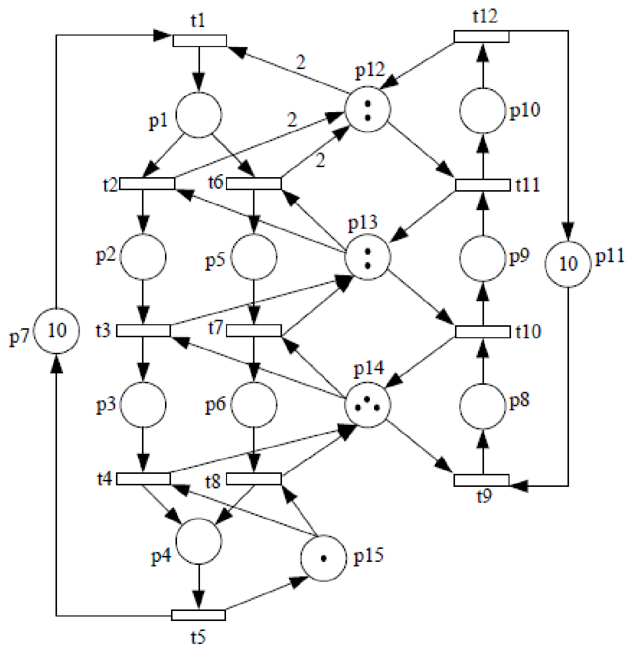

FIGURE 3. An S^{4}R (N, M_0).

Based on Theorem 3, for $S_1 = \{p_3, p_6, p_9, p_{13}, p_{14}\}$ in the S^{4}R net shown in Fig. 3, by Definition 2, we have $k_S^1 = p_1 + p_2 + p_5 + p_8$. As a result, $g_S^1 = k_S^1 + V_S^1 = p_1 + p_2 + p_5 + p_8 + V_S^1$. Notice that $\sum_{r \in S_1^R} I_r = I_{p_{13}} + I_{p_{14}} = p_2 + p_3 + p_5 + p_6 + p_8 + p_9 + p_{13} + p_{14}, f_{S_1} = \sum_{r \in S_1^R} I_r - g_S^1 = p_3 + p_6 + p_9 + p_{13} + p_{14} - p_1 - V_S^1$.

f_{S_1} is a PI of N_V. Let $\xi_S^1 = 2$. As a result, it holds $\xi_{S_1} > \sum_{p \in S_1} f_{S_1}(p)(max_{p^\bullet} - 1) = 0$. We conclude that S_1 is max-controlled by adding V_S^1 with $M_0^V(V_{S_1}) = 3$. Similarly, S_2 and S_3 are max-controlled by adding monitors V_S^2 and V_S^3, respectively, with $\xi_S^2 = 2$, $\xi_S^3 = 2$, $M_0^V(V_{S_2}) = 2$, and $M_0^V(V_{S_3}) = 5$. The resulting controlled net is shown in Fig. 4, which makes the net live.

Theorem 4 ([67]): (N_V, M_0^V) is obtained by adding monitors for all SMSs in an S^{4}R model (N, M_0) according to Theorem 3. Then, it is live.

III. S^{4}RP AS AN IMPROVED S^{4}R

A. S^{4}RP

Previous studies [68]–[77] have not treated some changes in an RMS caused by a variety of reasons, such as processing failures and equipment breakdowns. In an MRS, these changes may consist of newly added machines as well as the newly released product orders, or shifts in processing routes triggered by market competition in much detail. To

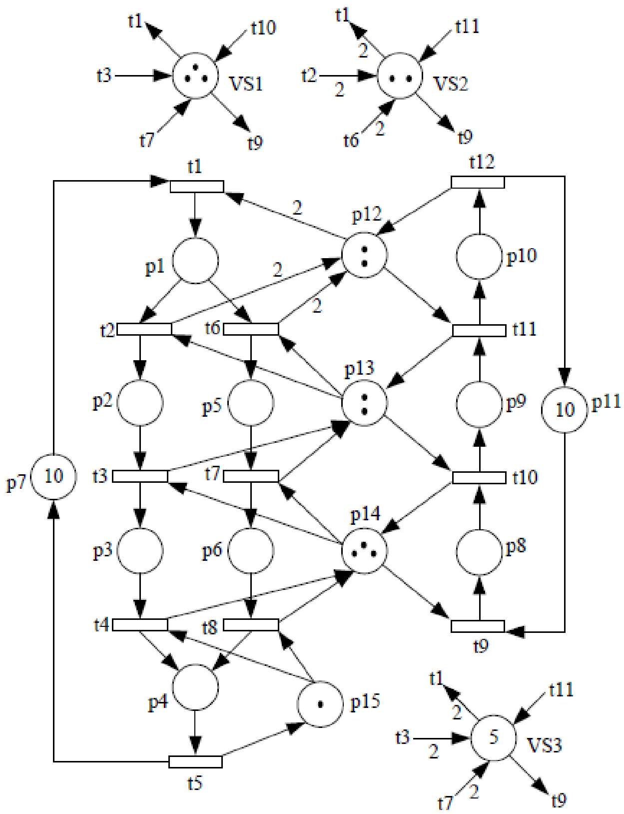

FIGURE 4. **The live S^4R (N_V, M_0^V) with three siphons being max-controlled by monitors.**

this end, an improved net model to S^2PR, namely a simple sequential process with resources and part-re-entry (S^2PRP), is proposed, which can deal with the dynamic changes in an RMS.

Since an S^4R is a generalized class of Petri net consisting of two or more WS^2PRs composed together through the shared resources, in this study, we extend our previous work [46] to an S^4R.

Definition 3 ([46]): Termination place (transition) p_t (t_t) is defined as the last place in a simple sequential process (S^2P) (the last transition) in an S^2P if $P_A \cap {}^\bullet t_t = \{p_t\}$.

Definition 4: A system of sequential systems with shared resources and part-re-entry (S^4RP), denoted as $(N, M_0) = (P, T, F, W, M_0, C, \Psi(p_a, p_r, T_e, F_e))$, is defined as follows:

1) $P = P_A \cup \{p_a\} \cup P^0 \cup P_R \cup \{p_r\}$, where $P_A = \bigcup_{j=1}^{n} P_A^j$ is called the set of operation places such that $P_A^i \cap P_A^j = \emptyset$, $\forall i \neq j$, and p_a is an additional operation place for the re-entry of failed parts and p_r is an additional resource for the failed operation. $P^0 = \bigcup_{i=1}^{n} \{p_i^0\}$ is called the set of idle places with $P^0 \cap (P_A \cup \{p_a\}) = \emptyset$, p_r is an additional resource place that represents a test machine TM, and $P_R = \{r_1, r_2, \ldots, r_m\}$ is called the set of resource places such that $((P^0 \cup \{P_A \cup \{p_a\}) \cap (P_R \cup \{p_r\}) = \emptyset$;
2) $T = \bigcup_{j=1}^{n} T_j$, and $\forall i \neq j$, $T_i \cap T_j = \emptyset$;
3) $W = W_A \cup W_R$, where $W_A : (P_A \cup \{p_a\} \cup P^0) \times T) \cup (T \times (P_A \cup \{p_a\} \cup P^0)) \longrightarrow \{0, 1\}$ such that $\forall j \neq i$, $((P_A^j \cup \{p_a^j, p_j^0\}) \times T_i) \cup (T_i \times (P_A^j \cup \{p_a^j, p_j^0\})) \longrightarrow \{0\}$, and $W_R : (P_R \cup \{p_r\} \times T) \cup (T \times P_R \cup \{p_r\}) \longrightarrow \mathbb{N}$;
4) $\forall j \in \mathbb{N}_n = \{1, 2, \ldots, n\}$, the subnet N_j derived from $P_A^j \cup \{p_j^0\} \cup T_j$ is a strongly connected state machine such that every circuit contains p_j^0;
5) $\forall r \in P_R$, there exists a unique PI I_r such that $\| I_r \| \cap P_R = \{r\}$, $\| I_r \| \cap P^0 = \emptyset$, $\| I_r \| \cap P_A \neq \emptyset$, and $I_r(r) = 1$. Furthermore, $P_A = (\cup_{r \in P_R} \| I_r \|) \setminus P_R$;
6) N is pure and strongly connected;
7) $M_0 : P \longrightarrow \mathbb{N}$ is the initial marking of N. $C : P \longrightarrow \mathbb{N}$ is a capacity function, where $C(p)$ indicates the bound of the place p.
8) $T_e = \{t_i, t_f\} \mid (T_e \cap {}^\bullet p_a = \{t_i\}) \wedge (T_e \cap p_a^\bullet = \{t_f\})$;
9) $({}^\bullet p_r = \{t_i, t_t\}) \wedge (p_r^\bullet = \{t \in T \mid t^\bullet = p_t\})$; and
10) $F_e \subseteq (P \cup \{p_a, p_r\} \times T \cup T_e) \cup (T \cup T_e \times P \cup \{p_a, p_r\})$.
11) $\forall p \in P_A$, $M_0(p) = 0$; $\forall r \in P_R$, $M_0(r) \geq max_{p \in \|I_r\|} I_r(p)$; $M_0(p_a) = 0$; $M_0(p_r) = 1$; $\forall p_j^0 \in P^0$, $M_0(p_j^0) \geq 1$.

Definition 5: Let $(N, M_0) = (P, T, F, W, M_0, C, \Psi(p_a, p_r, T_e, F_e))$ be an S^4RP. N is self-loop-free if for all $x, y \in P_A \cup \{p_a\} \cup P^0 \cup P_R \cup \{p_r\} \cup T$; $W(x, y) > 0$ means $W(y, x) = 0$.

Definition 6: Let $(N, M_0) = (P, T, F, W, M_0, C, \Psi(p_a, p_r, T_e, F_e))$ be an S^4RP. The marking M' reachable by firing an enabled transition $t \in T(M[t\rangle M')$ satisfies

$$M'(p) = \begin{cases} M(p) - W(p, t) + W(t, p), & \forall p \in ({}^\bullet t \cap t^\bullet); \\ M(p) - W(p, t), & \forall p \in ({}^\bullet t \setminus t^\bullet); \\ M(p) + W(t, p), & \forall p \in (t^\bullet \setminus {}^\bullet t); \\ M(p), & \text{otherwise} \end{cases}$$

Definition 7: Let $(N, M_0) = (P, T, F, W, M_0, C, \Psi(p_a, p_r, T_e, F_e))$ be an S^4RP. N is said to be ordinary if for all $(p, t) \in F$, $W(p, t) = 1$; otherwise it is generalized.

Definition 8: Let $(N, M_0) = (P, T, F, W, M_0, C, \Psi(p_a, p_r, T_e, F_e))$ be an S^4RP. The reachability set of (N, M_0) is denoted by $R(N, M_0)$. A transition $t \in T$ is live if for all $M \in R(N, M)$, there exists a reachable marking $M' \in R(N, M)$ such that $M'[t\rangle$ holds. (N, M_0) is said to be dead at M_0 if there is no $t \in T$ such that $M'[t\rangle$.

Definition 9: Let $(N, M_0) = (P, T, F, W, M_0, C, \Psi(p_a, p_r, T_e, F_e))$ be an S^4RP. A marking M_0 is said to be reversible if for each marking $M' \in R(N, M_0)$, M_0 is reachable from M'.

Definition 10: Let $(N, M_0) = (P, T, F, W, M_0, C, \Psi(p_a, p_r, T_e, F_e))$ be an S^4RP. A marking M' is said to be coverable if for each $p \in P$, there exists a marking $M'' \in R(N, M_0)$ such that $M''(p) \geqslant M'(p)$.

B. BUILDING AN S^4RP

To explain the construction of an S^4RP, let us model an RMS (with four robots R_1–R_4 and three machine tools M_1–M_3) whose layout is shown in Fig. 5 and production routines are in Fig. 6. Each robot (except for R_4) can hold one product at a time, while the capacity of R_4 is 3. The capacities of M_1, M_2, and M_3 are two, three, and three, respectively. There are three loading buffers I_1–I_3 and three unloading buffers O_1–O_3. The

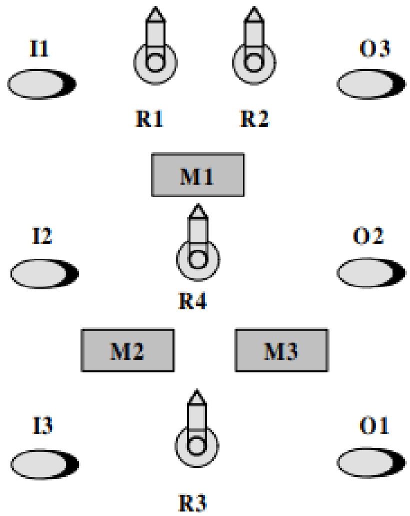

FIGURE 5. An RMS layout.

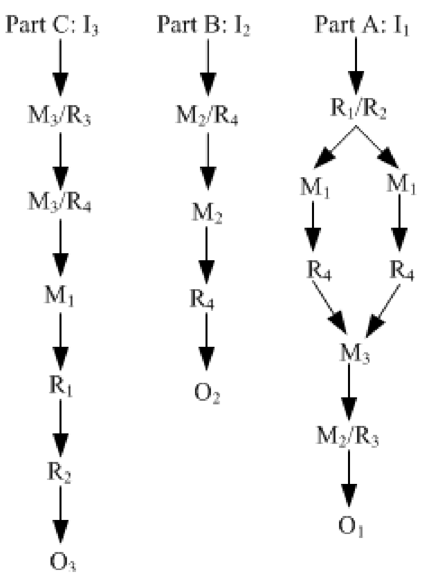

FIGURE 6. The production routings of an RMS.

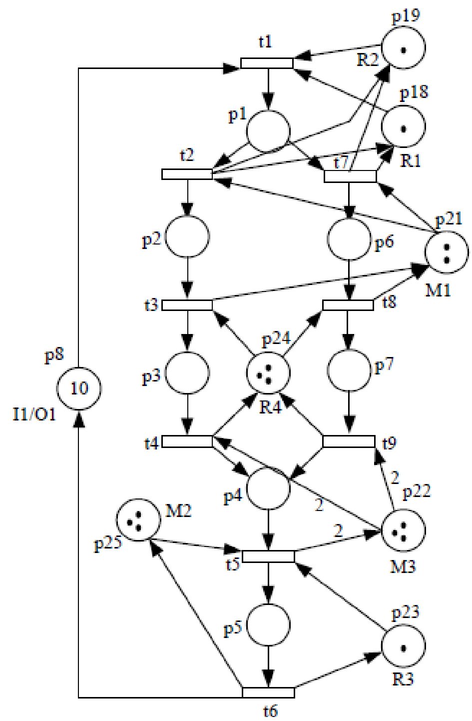

FIGURE 7. The petri net model (N, M_0) for working process 1.

system can produce three part types, namely A, B, and C. In Fig. 6, r/r' implies an alternative resource requirement for r and r' for a processing step. Fig. 7 represents the PN model (N, M_0) for working process of part A.

Now, we consider the case that the working process fails to produce a part A. According to the policy in [46], to take into account the processing failure and rework, a test machine TM is needed. Fig. 8 represents the Petri net model (N, M_0) that models the normal (as colored black) and re-entry operations (as colored red). Table 1 presents the physical meaning of the new improved part of the net model shown in Fig. 8.

The net model in Fig. 8 is a weighted simple sequential processes with resources and part-re-entry (WS^{2}PRP), where $p^0 = p_8$, $P_A = \{p_i \mid i = 2, 3, \ldots, 7\}$, $P_R = \{p_{18}, p_{19}, p_{21}, p_{22}, p_{23}, p_{24}, p_{25}\}$, $T = \{t_i \mid i = 1, 2, \ldots, 9\}$. Places $p_{18}, p_{19}, p_{21}, p_{22}, p_{23}, p_{24}$ and p_{25} represent $R_1, R_2, M_1, M_3, R_3, R_4$ and M_2, respectively. Before a system starts, no parts are in it. $M_0(p_8) = 10$ represents the number of instances that should be processed for part type $P_1 \longrightarrow A$, $p_a = p_{28}$, $p_r = p_{27}$, $p_t = p_{26}$, $t_t = t_{20}$, $T_e = \{t_i, t_f\}$ with $t_i = t_{21}$ and $t_f = t_{22}$, respectively; $F = \{(p_8, t_1), (t_1, p_1), (p_1, t_2), (t_2, p_2), (p_2, t_3), (t_3, p_3), (p_3, t_4), (t_4, p_4), (p_1, t_7), (t_7, p_6), (p_6, t_8), (t_8, p_7), (p_7, t_9), (t_9, p_4), (p_4, t_5), (t_5, p_5), (p_5, t_6), (t_6, p_8), (p_{18}, t_1), (p_{19}, t_1), (t_2, p_{18}), (t_2, p_{19}), (t_7, p_{18}), (t_7, p_{19}), (p_{21}, t_7), (p_{21}, t_2), (t_3, p_{21}), (t_8, p_{21}), (t_4, p_{24}), (t_9, p_{24}), (p_{24}, t_3), (p_{24}, t_8), (p_{22}, t_4), (p_{22}, t_9), (t_5, p_{22}), (p_{23}, t_5), (t_6, p_{23}), (p_{25}, t_5), (t_6, p_{25})\}$, $F_e = \{(t_6, p_{26}), (p_{26}, t_{20}), (p_{26}, t_{21}), (t_{21}, p_{28}), (p_{28}, t_{22}), (t_{22}, p_5), (t_{22}, p_{23}), (p_{23}, t_{21}), (p_{25}, t_{22}), (t_{20}, p_{27}), (t_{21}, p_{27}), (p_{27}, t_6)\}$, the weights of all arcs are equal to one, except that the weights of arcs in $\{(p_{22}, t_4), (p_{22}, t_9), (t_5, p_{22})\}$ are equal to two. Finally, the capacities of the resources are $C(R_1) = C(R_2) = C(R_3) = 1$, $C(R_4) = 3$, $C(M_1) = 2$, and $C(M_2) = C(M_3) = 3$.

According to Fig. 6, the Petri net model (N, M_0) for the working process 2 can be established as shown in Fig. 9(a) and Fig. 9(b) represents the PN model (N, M_0) which explains the reconfiguration of the working process when it fails to produce a qualified product B. The part with black arcs of the PN models the normal process, while the part with red arcs models the re-entry case. Table 2 demonstrates the implications of the new improved part of the net model in Fig. 9(b).

The net model shown in Fig. 9(b) is a weighted simple sequential processes with resources and part-re-entry

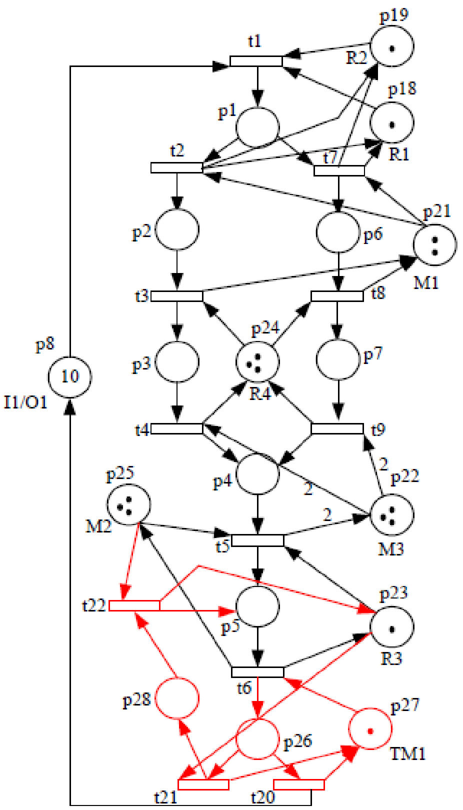

FIGURE 8. The Petri net model (N, M_0) explaining the re-entry part for working process 1.

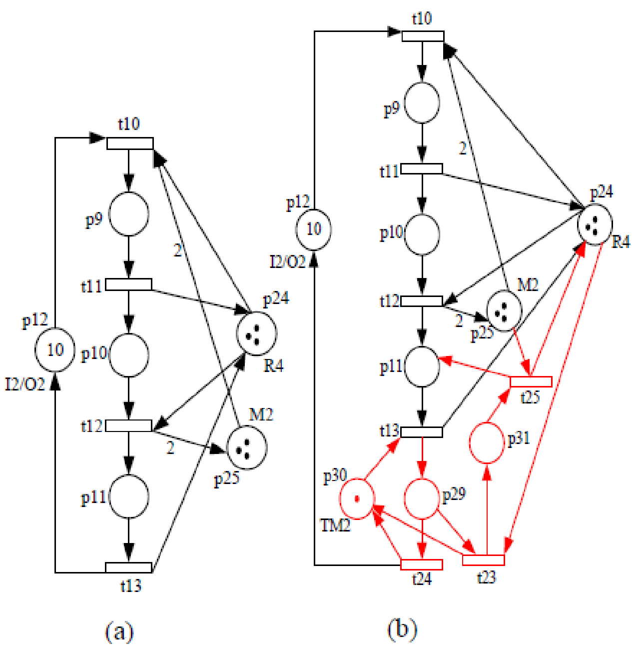

FIGURE 9. (a) The Petri net model (N, M_0) for working process 2, and (b) The petri net model (N, M_0) explaining the re-entry behaviour for working process 2.

(WS^{2}PRP), where $p^0 = p_{12}$, $P_A = \{p_9, p_{10}, p_{11}\}$, $P_R = \{p_{24}, p_{25}\}$, and $T = \{t_{10}, t_{11}, t_{12}, t_{13}\}$. Places p_{24} and p_{25} represent R_4 and M_2, respectively. At an initial stage, no parts

TABLE 1. Physical meaning of the re-entry part in the net model shown in Figure 8.

p/t	Meaning
t_6	When firing, robot R_3 is released and a finished part is dropped into place p_{26} for test.
p_{26}	The final part is downloaded for testing by the test machine p_{27}.
t_{20}	Its firing means that the test succeeds, and a final part is downloaded into output buffer p_8.
t_{21}	Its firing means that a failed part re-enters the system and is re-uploaded by robot R_3.
p_{28}	A failed part is downloaded.
t_{22}	When firing, a failed part is re-manufactured by machine M_2 and uploaded into p_5.
p_{18}	Robot 1 (R_1) with one unit available.
p_{19}	Robot 2 (R_2) with one unit available.
p_{21}	Machine 1 (M_1) with two units available.
p_{22}	Machine 3 (M_3) with three units available.
p_{23}	Robot 3 (R_3) with one unit available.
p_{24}	Robot 4 (R_4) with three units available.
p_{25}	Machine 2 (M_2) with three units available.
p_{27}	Test machine 1 TM_1 available.

TABLE 2. Physical meaning of the re-entry part in the net model shown in Figure 9(b).

p/t	Meaning
t_{13}	When firing, robot R_4 is released and a finished part is dropped into place p_{29} for test.
p_{29}	A final part is unloaded for testing by the test machine p_{30}.
t_{24}	Its firing means that the test succeeds, and a final part is loaded into output buffer p_{12}.
t_{23}	Its firing means that a failed part re-enters the system and is re-uploaded by robot R_4.
p_{31}	A failed part is unloaded.
t_{25}	When firing, a failed part is re-manufactured by machine M_2 and loaded into p_{11}.
p_{24}	Robot 4 (R_4) with three units available.
p_{30}	Test machine 2 TM_2 available.

are present in a system. $M_0(p_{12}) = 10$ represents the number of instances to be processed for part type $P_2 \longrightarrow B$, $p_a = p_{31}$, $p_r = p_{30}$, $p_t = p_{29}$, $t_t = t_{24}$, $T_e = \{t_i, t_f\}$ with $t_i = t_{23}$ and $t_f = t_{25}$, respectively; $F = \{(p_{12}, t_{10}), (t_{10}, p_9), (p_9, t_{11}), (t_{11}, p_{10}), (p_{10}, t_{12}), (t_{12}, p_{11}), (p_{11}, t_{13}), (t_{13}, p_{12}), (p_{24}, t_{10}), (p_{24}, t_{12}), (p_{25}, t_{10}), (t_{11}, p_{24}), (t_{12}, p_{25}), (t_{13}, p_{24})\}$, $F_e = \{(t_{13}, p_{29}), (p_{29}, t_{24}), (p_{29}, t_{23}), (t_{23}, p_{30}), (p_{31}, t_{25}), (t_{25}, p_{11}), (t_{25}, p_{24}), (p_{24}, t_{23}), (p_{25}, t_{25}), (t_{24}, p_{30}), (t_{23}, p_{31}), (p_{30}, t_{13})\}$, the weights of all arcs are equal to one, except that the weights of arcs $\{(p_{25}, t_{10}), (t_{12}, p_{25})\}$ are equal to two. Finally, the capacity of the resources is $C(R_4) = C(M_2) = 3$.

According to the production cycles shown in Fig. 6, the Petri net model (N, M_0) for the working process 3 can be constructed, as visualized in Fig. 10 and Fig. 11 represents the Petri net model (N, M_0) that explains the reconfiguration of the working process when it fails to produce a qualified product C. The part with black arcs of the PN models the normal process, while the part with red arcs models the re-entry case. Table 3 presents the physical meaning of the new improved part of the net model shown in Fig. 11.

The net model shown in Fig. 11 is a weighted S^{2}PRP, where $p^0 = p_{20}$, $P_A = \{p_i \mid i = 13, 14, \ldots, 17\}$, $P_R = \{p_{18}, p_{19}, p_{21}, p_{22}, p_{23}, p_{24}\}$, and $T = \{t_i \mid i = 14, 15, \ldots, 19\}$. Places p_{18}, p_{19}, p_{21}, p_{22}, p_{23} and p_{24} represent R_1, R_2, M_1, M_3, R_3, and R_4, respectively. Suppose that no parts are processed at the initial state. Thus, $M_0(p_{20}) = 10$ means the maximal number of instances to be processed for part type $P_3 \longrightarrow C$, $p_a = p_{34}$, $p_r = p_{33}$, $p_t = p_{32}$, $t_t = t_{26}$, $T_e = \{t_i, t_f\}$ with $t_i = t_{27}$ and $t_f = t_{28}$, respectively; $F = \{(p_{20}, t_{14}), (t_{14}, p_{13}), (p_{13}, t_{15}), (t_{15}, p_{14}), (p_{14}, t_{16}), (t_{16}, p_{15}), (p_{15}, t_{17}), (t_{17}, p_{16}), (p_{16}, t_{18}), (t_{18}, p_{17}), (p_{17}, t_{19}), (t_{19}, p_{19}), (p_{19}, t_{18}), (t_{18}, p_{18}), (p_{18}, t_{17}), (t_{17}, p_{21}), (p_{21}, t_{16}), (t_{16}, p_{24}), (t_{16}, p_{22}), (p_{24}, t_{15}), (p_{22}, t_{14}), (t_{15}, p_{23}), (p_{23}, t_{14})\}$,

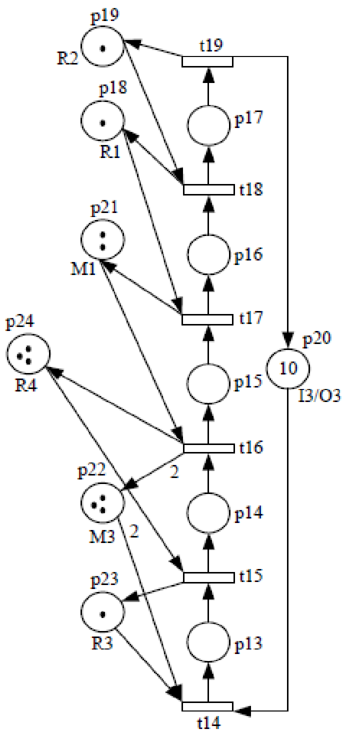

FIGURE 10. A PN (N, M_0) for working process 3.

TABLE 3. Physical meaning of the re-entry part in the net model shown in Figure 11.

p/t	Meaning
t_{19}	When firing, robot R_2 is released and a finished part is dropped into place p_{32} for test.
p_{32}	A final part is downloaded for testing by the test machine p_{33}.
t_{26}	Its firing means that the test succeeds, and a final part is loaded into output buffer p_{20}.
t_{27}	Its firing means that a failed part re-enters the system and is re-uploaded by robot R_2.
p_{34}	A failed part is unloaded.
t_{28}	When firing, a failed part is re-manufactured by machine M_1 and uploaded into p_{15}.
p_{18}	Robot 1 (R_1) with one unit available.
p_{19}	Robot 2 (R_2) with one unit available.
p_{21}	Machine 1 (M_1) with two units available.
p_{22}	Machine 3 (M_3) with three units available.
p_{23}	Robot 3 (R_3) with one unit available.
p_{24}	Robot 4 (R_4) with three units available.
p_{33}	Test machine 3 TM_3 available.

$F_e = \{(t_{19}, p_{32}), (p_{32}, t_{26}), (t_{26}, p_{33}), (p_{33}, t_{19}), (p_{32}, t_{27}), (t_{27}, p_{33}), (t_{27}, p_{34}), (p_{34}, t_{28}), (t_{28}, p_{15}), (t_{28}, p_{19}), (p_{19}, t_{27}), (p_{21}, t_{28})\}$, the weights of all arcs are equal to one, except that the weights of arcs $\{(p_{22}, t_{14}), (t_{16}, p_{22})\}$ are equal to two. Finally, the capacities of resources are $C(R_1) = C(R_2) = C(R_3) = 1$, $C(R_4) = 3$, $C(M_1) = 2$, and $C(M_3) = 3$.

According to the concept of composition [66], [67], we can compose N_1, N_2, and N_3 shown in Figs. 7, 9(a), and 10 respectively to obtain an S^{4}R that represents an RMS layout shown in Fig. 5. In the same way, we can compose N_1, N_2, and N_3 as shown in Figs. 8, 9(b), and 11 respectively to obtain an S^{4}RP that includes the case of processing failures and rework. Fig. 12 shows the S^{4}R and Fig. 13 illustrates its corresponding S^{4}RP.

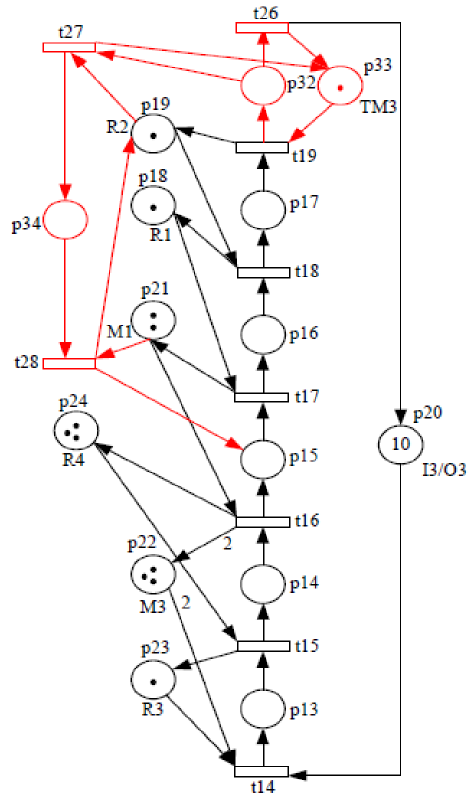

FIGURE 11. A PN (N, M_0) that explains the re-entry part for working process 3.

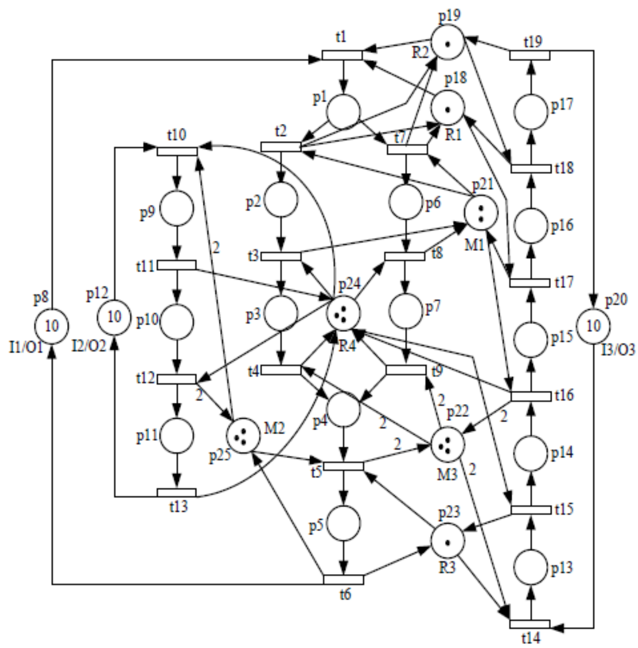

FIGURE 12. The S^{4}R that represents an RMS layout shown in Fig. 5.

IV. ROBUST DEADLOCK CONTROL FOR S^{4}RP

Based on the classical deadlock prevention policy in [66], this section proposes a supervisor for an S^{4}RP and a procedure to compute a robust LES and the controlled system (N_V, M_0^V) for S^{4}RPs is developed. Here we make use of the notion of elementary siphons proposed in [65], [66] to simplify

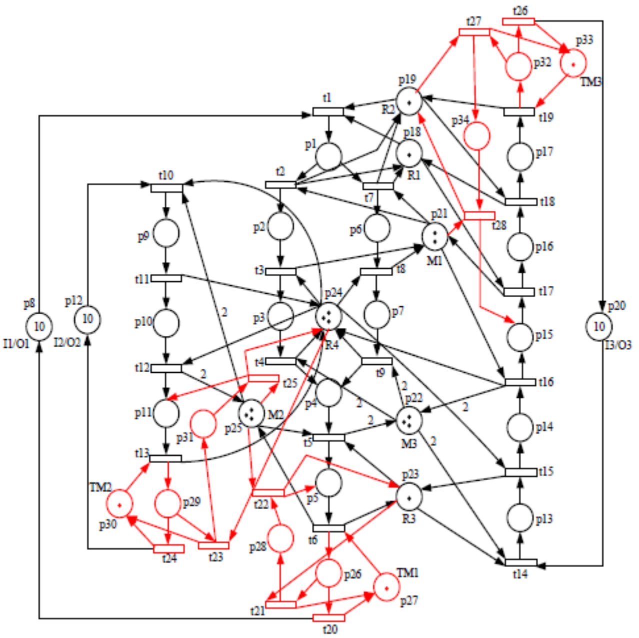

FIGURE 13. **The S^4RP that presents the case of processing failures and rework.**

the structural complexity of the supervisor. For economy of space, the related definitions regarding elementary siphons are not presented.

Theorem 5: Let $(N, M_0) = (P_A \cup P^0 \cup P_R, T, F, W, M_0, C, \Psi(p_a, p_r, T_e, F_e))$ be an S^4RP and N_{ES} the number of its elementary siphons. Then, $N_{ES} \leqslant |P_A|$.

Proof: A similar proof can be found in [65]. □

In an S^4RP, the control of a dependent siphon could be achieved through that of its elementary siphons. By Definition 4 and Theorem 3, we can formulate a DPP (as shown in Algorithm 1) for S^4RP.

Assumption: Transitions in $T_e = \{t_i, t_f\}$ in an S^4RP $(N, M_0) = (P, T, F, W, M_0, C, \Psi(p_a, p_r, T_e, F_e))$ are reliable (i.e., transitions that do not fail to fire whenever enabled).

Theorem 6: An S^4RP $(N, M_0) = (P, T, F, W, M_0, C, \Psi(p_a, p_r, T_e, F_e))$ with an unreliable element that causes failed parts can be recovered by re-entering the failed parts and becomes live by applying Algorithm 1.

Proof: When a working process fails to produce a predefined product, the processing step modelled by P_A^i cannot be successfully finished such that the token in P_A^i is removed. Meanwhile, no token is in P_R^i, indicating the availability of the considered product. Thus, the transitions pertaining to the process that makes the considered product are disabled. At this state when this considered product fails to pass the test, the terminal place p_t receives a token, implying that the product is under testing by the test machine p_r. Then, transition $t_i \in T_e$ is enabled. Enabling $t_i \in T_e$ implies that the flawed product re-enters the system and is moved into the operation place p_a. Thus, $t_f \in T_e$ is enabled and the flawed product re-enters the system and is re-produced by

Algorithm 1 Computation of a robust LES for S^4RP

Input: S^4RP $(N, M_0) = (P, T, F, W, M_0, C, \Psi(p_a, p_r, T_e, F_e))$.
Output: Robust controlled system (N_V, M_0^V) with its liveness ensured or "Undecided".
1. **for** $i \leftarrow 1$ to n **do**
1.1 Find the possible routes RO_i
1.2 Find the termination place p_t by Definition 3
1.3 Find the termination transition t_t such that $P_A \cap {}^\bullet t_t = \{p_t\}$
1.4 Introduce a resource place p_r representing the test machine TM
1.5 Build the operation place p_a that receives a failed part
1.6 Build a set of extended transitions T_e and its corresponding extended arcs F_e such that
1.6.1 $T_e = \{t_i, t_f\} \mid (T_e \cap^\bullet p_a = \{t_i\}) \wedge (T_e \cap p_a^\bullet = \{t_f\})$
1.6.2 $({}^\bullet p_r = \{t_i, t_t\}) \wedge (p_r^\bullet = \{t \in T \mid t^\bullet = p_t\})$
1.6.3 $F_e \subseteq (P \cup \{p_a, p_r\} \times T \cup T_e) \cup (T \cup T_e \times P \cup \{p_a, p_r\})$.
1.7 Establish a WS^2PRP that models RO
1.8 **end for**
2. Construct an S^4RP that models all possible routes RO_i that produce different part types by Definition 4
3. Compute $\prod$ for the obtained S^4RP
4. Compute $\prod_E$
5. flag: = 0
6. **for** $(i = 1; i <|\prod_E| + 1; i++)$ **do**
6.1 Compute k_S^i for S_i according to Definition 2
6.2 Add V_S^i to (N, M_0) to make S_i max-controlled based on Theorem 3 with $M_0^V(V_S^i) = M_0(S_i) - \xi_S^i$, where $\xi_S^i > \sum_{p \in Si} fs_{S_i}(p)(max_{p^\bullet} - 1)$
6.3 **if** $M_0^V(V_S) < max_{v_{s_i}^\bullet}$ **then**
6.4 flag: = 1
6.5 **end if**
6.6 **end for**
7. **if** flag = 1 **then**
7.1 **Output** "Undecided"
7.2 **else**
7.3 **Output** a robust controlled system (N_V, M_0^V)
7.4 **end if**

the machine. We infer that the PN S^4RP always stays active when the unreliable element fails. Thus, the PN S^4RP is live under the controlled fault recovery and re-entry process stated in Algorithm 1. □

Now we explain, by Algorithm 1, how to design a robust LES by using the S^4RP $(N, M_0) = (P, T, F, W, M_0, C, \Psi(p_a, p_r, T_e, F_e))$ represented in Fig. 13 as an example.

Thanks to INA [78], 22 strict minimal siphons shown in Table 4 can be computed. It is clear that the set of elementary siphons $\Pi_E = \{S_i \mid i = 1, 2, \ldots, 6\}$, and we compute V_S^i for each siphon according to Definition 4 and Theorem 3.

1. For elementary siphon S_1.
$S_1 = \{p_5, p_9, p_{11}, p_{14}, p_{22}, p_{24}, p_{25}\}$,
$I_{p_{22}} = 2p_4 + 2p_{13} + 2p_{14} + p_{22}$,
$I_{p_{24}} = p_3 + p_7 + p_9 + p_{11} + p_{14} + p_{31} + p_{24}$,
$I_{p_{25}} = p_5 + 2p_9 + 2p_{10} + p_{11} + p_{25}$,
$\sum_{r\in S_1^R} I_r = p_3 + 2p_4 + p_5 + p_7 + 3p_9 + 2p_{10} + 2p_{11} + 2p_{13} + 3p_{14} + p_{31} + p_{22} + p_{24} + p_{25}$.
$[S_1] = p_3 + 2p_4 + p_7 + 2p_{10} + 2p_{13} + p_{31}$,
$\wp_s = \wp_s^1 \cup \wp_s^2 \cup \wp_s^3 = \{p_1, p_2, p_3, p_4, p_6, p_7\} \cup \{p_9, p_{10}, p_{11}, p_{29}, p_{31}\} \cup \{p_{13}\}$,
$B_S^i = \{p_4, p_{31}, p_{13}\}$,
$h_s(p_3) := 1, h_s(p_4) := 2, h_s(p_7) := 1, h_s(p_{10}) := 2, h_s(p_{13}) := 2, h_s(p_{31}) := 1$.
For $p_4 \in B_S^i$, we have
$\alpha_{p_4} := max\{2, 1\} = 2$,
$k_s(p_1) = k_s(p_2) = k_s(p_3) = k_s(p_4) = k_s(p_6) = k_s(p_7) = 2$.
For $p_{31} \in B_S^i$, we have
$\alpha_{p_{31}} := max\{1, 2\} = 2$,
$k_s(p_9) = k_s(p_{10}) = k_s(p_{11}) = k_s(p_{29}) = k_s(p_{31}) = 2$.
For $p_{13} \in B_S^i$, we have
$\alpha_{p_{13}} := 2$,
$k_s(p_{13}) := 2$,
$k_{s_1} = 2p_1 + 2p_2 + 2p_3 + 2p_4 + 2p_6 + 2p_7 + 2p_9 + 2p_{10} + 2p_{11} + 2p_{29} + 2p_{31} + 2p_{13}$,
$g_{s_1} = 2p_1 + 2p_2 + 2p_3 + 2p_4 + 2p_6 + 2p_7 + 2p_9 + 2p_{10} + 2p_{11} + 2p_{29} + 2p_{31} + 2p_{13} + Vs_1$,
$f_{s_1} = \sum_{r\in S_1^R} I_r - g_{s_1} = p_5 + p_9 + p_{11} + 3p_{14} + p_{22} + p_{24} + p_{25} - V_{s_1} - 2p_1 - 2p_2 - p_3 - 2p_6 - p_7$,
$V_{s_1} = p_5 + p_9 + p_{11} + 3p_{14}$,
$M_0^v(V_{s_1}) = 9 - \xi_{s_1}$,
$2 < \xi_{s_1} \leq 8$.
2. For elementary siphon S_2.
$S_2 = \{p_5, p_9, p_{11}, p_{14}, p_{22}, p_{23}, p_{24}, p_{28}, p_{31}\}$,
$I_{p_{22}} = 2p_4 + 2p_{13} + 2p_{14} + p_{22}$,
$I_{p_{23}} = p_5 + p_{13} + p_{23} + p_{28}$,
$I_{p_{24}} = p_3 + p_7 + p_9 + p_{11} + p_{14} + p_{31} + p_{24}$,
$\sum_{r\in S_2^R} I_r = p_3 + 2p_4 + p_5 + p_7 + p_9 + p_{11} + 3p_{13} + 3p_{14} + p_{31} + p_{22} + p_{23} + p_{24}$.
$[S_2] = p_3 + 2p_4 + p_7 + 3p_{13}$,
$\wp_s = \wp_s^1 \cup \wp_s^2 = \{p_1, p_2, p_3, p_4, p_6, p_7\} \cup \{p_{13}\}$,
$B_S^i = \{p_4, p_{13}\}$,
$h_s(p_3) := 1, h_s(p_4) := 2, h_s(p_7) := 1, h_s(p_{13}) := 3$.
For $p_4 \in B_S^i$, we have
$\alpha_{p_4} := max\{2, 1\} = 2$,
$k_s(p_1) = k_s(p_2) = k_s(p_3) = k_s(p_4) = k_s(p_6) = k_s(p_7) = 2$.
For $p_{13} \in B_S^i$, we have
$\alpha_{p_{13}} := 3$,
$k_s(p_{13}) := 3$,
$k_{s_2} = 2p_1 + 2p_2 + 2p_3 + 2p_4 + 2p_6 + 2p_7 + 3p_{13}$,
$g_{s_2} = 2p_1 + 2p_2 + 2p_3 + 2p_4 + 2p_6 + 2p_7 + 3p_{13} + V_{s_2}$,
$f_{s_2} = \sum_{r\in S_2^R} I_r - g_{s_2} = p_5 + p_9 + p_{11} + 3p_{14} + p_{31} + p_{22} + p_{23} + p_{24} - V_{s_2} - 2p_1 - 2p_2 - p_3 - 2p_6 - p_7$,
$V_{s_2} = p_5 + p_9 + p_{11} + 3p_{14} + p_{31}$,
$M_0^v(Vs_2) = 7 - \xi_{s_2}$,
$1 < \xi_{s_2} \leq 6$.
3. For elementary siphon S_3.
$S_3 = \{p_4, p_9, p_{11}, p_{14}, p_{22}, p_{24}, p_{31}\}$, $I_{p_{22}} = 2p_4 + 2p_{13} + 2p_{14} + p_{22}$, $I_{p_{24}} = p_3 + p_7 + p_9 + p_{11} + p_{14} + p_{31} + p_{24}$,
$\sum_{r\in S_3^R} I_r = p_3 + 2p_4 + p_7 + p_9 + p_{11} + 2p_{13} + 3p_{14} + p_{31} + p_{22} + p_{24}$.
$[S_3] = p_3 + p_7 + 2p_{13}$,
$\wp_s = \wp_s^1 \cup \wp_s^2 = \{p_1, p_2, p_3, p_6, p_7\} \cup \{p_{13}\}$,
$B_S^i = \{p_3, p_7, p_{13}\}$,
$h_s(p_3) := 1, h_s(p_7) := 1, h_s(p_{13}) := 2$.
For $p_3 \in B_S^i$, we have
$\alpha_{p_3} := max\{1, 1\} = 1$,
$k_s(p_1) = k_s(p_2) = k_s(p_3) = k_s(p_6) = k_s(p_7) = 1$.
For $p_{13} \in B_S^i$, we have
$\alpha_{p_{13}} := 2$,
$k_s(p_{13}) := 2$,
$k_{s_3} = p_1 + p_2 + p_3 + p_6 + p_7 + 2p_{13}$,
$g_{s_3} = p_1 + p_2 + p_3 + p_6 + p_7 + 2p_{13} + V_{s_3}$,
$f_{s_3} = \sum_{r\in S_3^R} I_r - g_{s_3} = 2p_4 + p_9 + p_{11} + 3p_{14} + p_{31} + p_{22} + p_{24} - V_{s_3} - p_1 - p_2 - p_6$,
$Vs_3 = 2p_4 + p_9 + p_{11} + 3p_{14} + p_{31}$,
$M_0^v(Vs_3) = 6 - \xi s_3$,
$1 < \xi s_3 \leq 5$.
4. For elementary siphon S_4.
$S_4 = \{p_3, p_7, p_9, p_{11}, p_{15}, p_{21}, p_{24}, p_{31}\}$,
$I_{p_{21}} = p_2 + p_6 + p_{15} + p_{21}$,
$I_{p_{24}} = p_3 + p_7 + p_9 + p_{11} + p_{14} + p_{31} + p_{24}$,
$\sum_{r\in S_4^R} I_r = p_2 + p_3 + p_6 + p_7 + p_9 + p_{11} + p_{14} + p_{15} + p_{31} + p_{21} + p_{24}$.
$[S_4] = p_2 + p_6 + p_{14}$,
$\wp_s = \wp_s^1 \cup \wp_s^2 = \{p_1, p_2, p_6\} \cup \{p_{13}, p_{14}\}$,
$B_S^i = \{p_2, p_6, p_{14}\}$,
$h_s(p_2) := 1, h_s(p_6) := 1, h_s(p_{14}) := 1$.
For $p_2 \in B_S^i$, we have
$\alpha_{p_2} := max\{1, 1\} = 1$,
$k_s(p_1) = k_s(p_2) = k_s(p_6) = 1$.
For $p_{14} \in B_S^i$, we have
$\alpha_{p_{14}} := 1$,
$k_s(p_{13}) = k_s(p_{14}) = 1$,
$k_{s_4} = p_1 + p_2 + p_6 + p_{13} + p_{14}$,
$g_{s_4} = p_1 + p_2 + p_6 + p_{13} + p_{14} + Vs_4$,
$f_{s_4} = \sum_{r\in S_4^R} I_r - g_{s_4} = p_3 + p_7 + p_9 + p_{11} + p_{15} + p_{31} + p_{21} + p_{24} - Vs_4 - p_1 - p_{13}$,
$Vs_4 = p_3 + p_7 + p_9 + p_{11} + p_{15} + p_{31}$,
$M_0^v(Vs_4) = 5 - \xi_{s_4}$,
$0 < \xi_{s_4} \leq 4$.
5. For elementary siphon S_5.
$S_5 = \{p_2, p_6, p_{16}, p_{18}, p_{21}\}$,
$I_{p_{18}} = p_1 + p_{16} + p_{18}$,
$I_{p_{21}} = p_2 + p_6 + p_{15} + p_{21}$,
$\sum_{r\in S_5^R} I_r = p_1 + p_2 + p_6 + p_{15} + p_{16} + p_{18} + p_{21}$.
$[S_5] = p_1 + p_{15}$,
$\wp_s = \wp_s^1 \cup \wp_s^2 = \{p_1\} \cup \{p_{13}, p_{14}, p_{15}\}$,
$B_S^i = \{p_1, p_{15}\}$,
$h_s(p_1) := 1, h_s(p_{15}) := 1$.

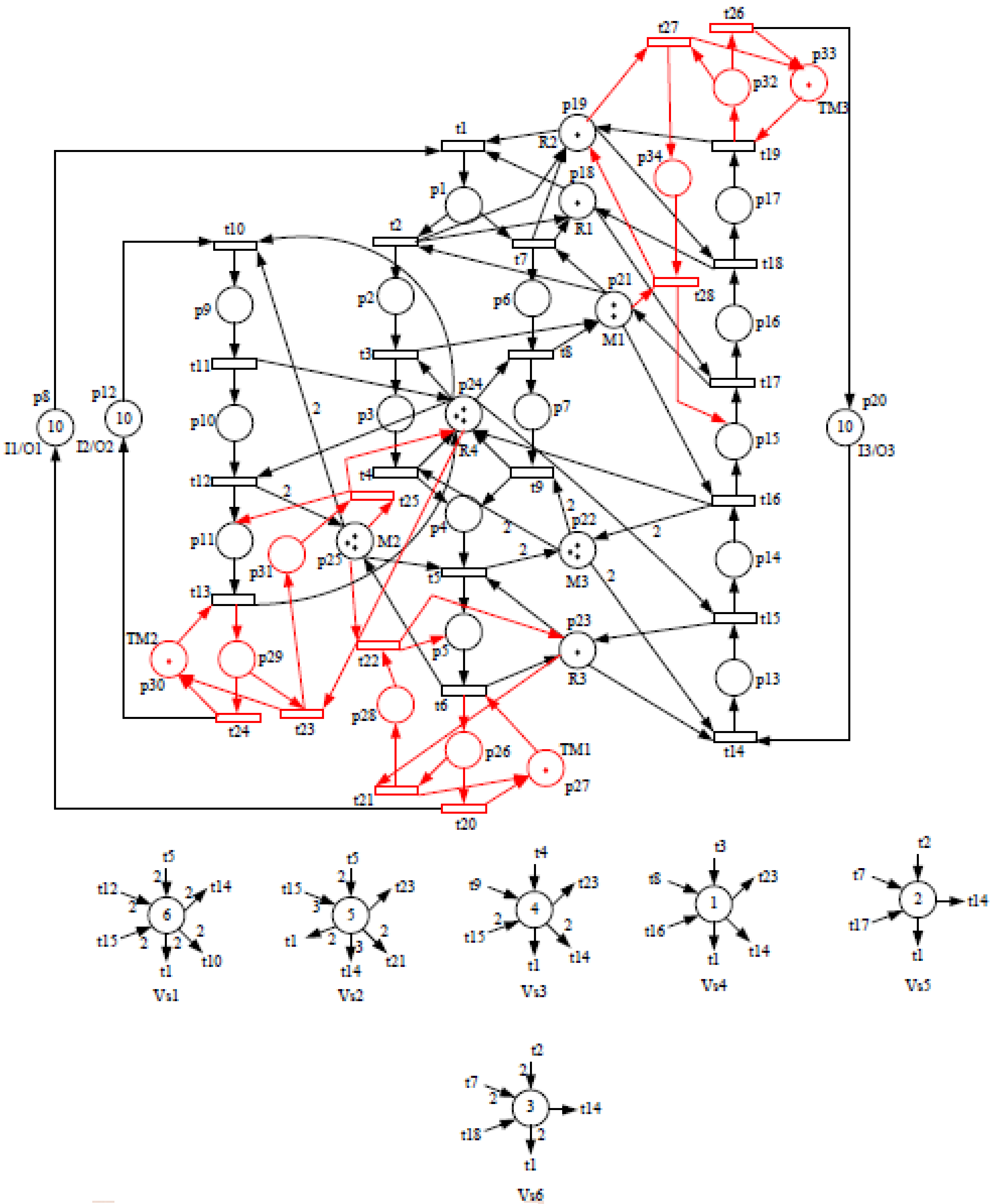

FIGURE 14. The controlled and live S^4RP with processing failures and rework.

For $p_1 \in B_S^i$, we have
$\alpha_{p_1} := max\{1, 1\} = 1$,
$k_s(p_1) := 1$.
For $p_{15} \in B_S^i$, we have
$\alpha_{p_{15}} := 1$,
$k_s(p_{13}) = k_s(p_{14}) = k_s(p_{15}) = 1$,
$k_{s_5} = p_1 + p_{13} + p_{14} + p_{15}$,
$g_{s_5} = p_1 + p_{13} + p_{14} + p_{15} + Vs_5$,

TABLE 4. SMSs in Figure 13.

S_i	SMS
S_1	$\{p_5, p_9, p_{11}, p_{14}, p_{22}, p_{24}, p_{25}\}$.
S_2	$\{p_5, p_9, p_{11}, p_{14}, p_{22}, p_{23}, p_{24}, p_{28}, p_{31}\}$.
S_3	$\{p_4, p_9, p_{11}, p_{14}, p_{22}, p_{24}, p_{31}\}$.
S_4	$\{p_3, p_7, p_9, p_{11}, p_{15}, p_{21}, p_{24}, p_{31}\}$.
S_5	$\{p_2, p_6, p_{16}, p_{18}, p_{21}\}$.
S_6	$\{p_2, p_6, p_{17}, p_{18}, p_{19}, p_{21}\}$.
S_7	$\{p_5, p_9, p_{11}, p_{15}, p_{21}, p_{22}, p_{23}, p_{24}, p_{28}, p_{31}\}$.
S_8	$\{p_5, p_9, p_{11}, p_{16}, p_{18}, p_{21}, p_{22}, p_{24}, p_{25}\}$
S_9	$\{p_5, p_9, p_{10}, p_{25}\}$.
S_{10}	$\{p_4, p_9, p_{11}, p_{17}, p_{18}, p_{19}, p_{21}, p_{22}, p_{24}, p_{31}\}$.
S_{11}	$\{p_4, p_9, p_{11}, p_{16}, p_{18}, p_{21}, p_{22}, p_{24}, p_{31}\}$.
S_{12}	$\{p_4, p_9, p_{11}, p_{15}, p_{21}, p_{22}, p_{24}, p_{31}\}$.
S_{13}	$\{p_5, p_9, p_{11}, p_{15}, p_{21}, p_{22}, p_{24}, p_{25}\}$.
S_{14}	$\{p_3, p_7, p_9, p_{11}, p_{17}, p_{18}, p_{19}, p_{21}, p_{24}, p_{31}\}$.
S_{15}	$\{p_3, p_5, p_7, p_9, p_{11}, p_{17}, p_{18}, p_{19}, p_{21}, p_{24}, p_{25}\}$.
S_{16}	$\{p_3, p_7, p_9, p_{11}, p_{16}, p_{18}, p_{21}, p_{24}, p_{31}\}$.
S_{17}	$\{p_3, p_5, p_7, p_9, p_{11}, p_{16}, p_{18}, p_{21}, p_{24}, p_{25}\}$.
S_{18}	$\{p_5, p_9, p_{11}, p_{17}, p_{18}, p_{19}, p_{21}, p_{22}, p_{24}, p_{25}\}$.
S_{19}	$\{p_3, p_5, p_7, p_9, p_{11}, p_{15}, p_{21}, p_{24}, p_{25}\}$.
S_{20}	$\{p_5, p_9, p_{11}, p_{17}, p_{18}, p_{19}, p_{21}, p_{22}, p_{23}, p_{24}, p_{28}, p_{31}\}$.
S_{21}	$\{p_3, p_5, p_7, p_9, p_{11}, p_{14}, p_{24}, p_{25}\}$.
S_{22}	$\{p_5, p_9, p_{11}, p_{16}, p_{18}, p_{21}, p_{22}, p_{23}, p_{24}, p_{28}, p_{31}\}$.

TABLE 5. Elementary siphons and monitors in Figure 13. Note that we use d_i to denote $M_V0(V_Si)$.

S_i	Elementary Siphon	${}^\bullet V_{Si}$	$V_{S_i}^\bullet$	d_i
S_1	$\{p_5, p_9, p_{11}, p_{14}, p_{22}, p_{24}, p_{25}\}$	$\{2t_5, 2t_{12}, 2t_{15}\}$	$\{2t_1, 2t_{10}, 2t_{14}\}$	6
S_2	$\{p_5, p_9, p_{11}, p_{14}, p_{22}, p_{23}, p_{24}, p_{28}, p_{31}\}$	$\{2t_5, 3t_{15}\}$	$\{2t_1, 3t_{14}, 2t_{21}, t_{23}\}$	5
S_3	$\{p_4, p_9, p_{11}, p_{14}, p_{22}, p_{24}, p_{31}\}$	$\{t_4, t_9, 2t_{15}\}$	$\{t_1, 2t_{14}, t_{23}\}$	4
S_4	$\{p_3, p_7, p_9, p_{11}, p_{15}, p_{21}, p_{24}, p_{31}\}$	$\{t_3, t_8, t_{16}\}$	$\{t_1, t_{14}, t_{23}\}$	1
S_5	$\{p_2, p_6, p_{16}, p_{18}, p_{21}\}$	$\{t_2, t_7, t_{17}\}$	$\{t_1, t_{14}\}$	2
S_6	$\{p_2, p_6, p_{17}, p_{18}, p_{19}, p_{21}\}$	$\{2t_2, 2t_7, t_{18}\}$	$\{2t_1, t_{14}\}$	3

$f_{s_5} = \sum_{r\in S_5^R} I_r - gs_5 = p_2 + p_6 + p_{16} + p_{18} + p_{21} - Vs_5 - p_{13} - p_{14}$,

$Vs_5 = p_2 + p_6 + p_{16}$,

$M_0^v(Vs_5) = 3 - \xi_{s_5}$,

$0 < \xi_{s_5} \leq 2$.

6. For elementary siphon S_6.

$S_6 = \{p_2, p_6, p_{17}, p_{18}, p_{19}, p_{21}\}$,

$I_{p_{18}} = p_1 + p_{16} + p_{18}$,

$I_{p_{19}} = p_1 + p_{17} + p_{19}$,

$I_{p_{21}} = p_2 + p_6 + p_{15} + p_{21}$,

$\sum_{r\in S_6^R} I_r = 2p_1 + p_2 + p_6 + p_{15} + p_{16} + p_{17} + p_{18} + p_{19} + p_{21}$.

$[S_6] = 2p_1 + p_{15} + p_{16}$,

$\wp_s = \wp_s^1 \cup \wp_s^2 = \{p1\} \cup \{p_{13}, p_{14}, p_{15}, p_{16}\}$,

$B_S^i = \{p_1, p_{16}\}$,

$h_s(p_1) := 2, h_s(p_{16}) := 1$.

For $p_1 \in B_S^i$, we have

$\alpha_{p_1} := 2$,

$k_s(p_1) := 2$.

For $p_{16} \in B_S^i$, we have

$\alpha_{p_{16}} := 1$,

$k_s(p_{13}) = k_s(p_{14}) = k_s(p_{15}) = k_s(p_{16}) = 1$,

$k_{s_6} = 2p_1 + p_{13} + p_{14} + p_{15} + p_{16}$,

$g_{s_6} = 2p_1 + p_{13} + p_{14} + p_{15} + p_{16} + V_{s_6}$,

$f_{s_6} = \sum_{r\in S_6^R} I_r - g_{s_6} = p_2 + p_6 + p_{17} + p_{18} + p_{19} + p_{21} - V_{s_6} - p_{13} - p_{14}$,

$V_{s_6} = p_2 + p_6 + p_{17}$,

$M_0^v(Vs_6) = 4 - \xi_{s_6}$,

$0 < \xi_{s_6} \leq 3$.

Table 5 shows the elementary siphons in the S4RP in Fig. 13. Fig. 14 represents the live and controlled S4RP $(N, M_0) = (P, T, F, W, M_0, C, \Psi(p_a, p_r, T_e, F_e))$ for the Net shown in Fig. 13.

V. CONCLUSION

The main goal of the current study was to design a new DPP for preventing the occurrences of deadlocks in a reconfigurable MRS by excogitating a novel PN subclass, S4RP for short, which addresses the case of processing failures and rework, and represents the process that a flawed part re-enters the underlying system to be re-processed. We apply a siphon-based max-controllability DPP to make the proposed S4RP live through adding external control elements. We also use INA–a PN analyzer to validate the results.

The main results of the proposed strategy consist of the following contributions: (1) it can be extended to a generalized PN subclasses, such as ES3PR, S*PR, S2LSPR, S3PGR2 and S3PMR; (2) the proposed structural analysis techniques for S4RP do not need to generate reachability graphs; (3) it can manage failures efficiently in an MRS; (4) the developed S4RP can dynamically modify the arrangement of the PN without breaking its liveness property; (5) we include a representation of the configuration of reconfigurable manufacturing systems and provide an S4RP model for the architecture of the supervisor; and (6) an S4RP is applicable to a system with complicated resource requirements.

This research develops a generalized class of Petri nets, namely S4R that can simulate reconfigurable MRS. This means that the process can request and release more than one unit of single or multiple resources at a time. Moreover, elementary siphons are used for the case of re-entry parts in an MRS, which leads to expensive computing overheads.

In the future study, the behavioural permissiveness and structural complexity of a supervisor will be optimized and reduced. We will also focus on expanding and improving the proposed method to a system with uncontrollable and unobservable events.

REFERENCES

[1] E. G. Coffman, M. Elphick, and A. Shoshani, "System deadlocks," *ACM Comput. Surv.*, vol. 3, no. 2, pp. 67–78, Jun. 1971.

[2] J. Luo, Z. Liu, and M. Zhou, "A Petri net based deadlock avoidance policy for flexible manufacturing systems with assembly operations and multiple resource acquisition," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3379–3387, Jun. 2019.

[3] J. Sha, Y. Du, and L. Qi, "A user requirement oriented Web service discovery approach based on logic and threshold Petri net," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 6, pp. 1528–1542, Nov. 2019.

[4] J. Zhou, J. Wang, and J. Wang, "A simulation engine for stochastic timed Petri nets and application to emergency healthcare systems," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 4, pp. 969–980, Jul. 2019, doi: 10.1109/JAS.2019.1911576.

[5] F. Yang, N. Wu, Y. Qiao, and R. Su, "Polynomial approach to optimal one-wafer cyclic scheduling of treelike hybrid multi-cluster tools via Petri nets," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 1, pp. 270–280, Jan. 2018, doi: 10.1109/JAS.2017.7510772.

[6] M. Zhou and F. DiCesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. London, U.K.: Kluwer Academic Publishers, 1993.

[7] H.-C. Liu, J.-X. You, Z. Li, and G. Tian, "Fuzzy Petri nets for knowledge representation and reasoning: A literature review," *Eng. Appl. Artif. Intell.*, vol. 60, pp. 45–56, Apr. 2017, doi: 10.1016/j.engappai.2017.01.012.

[8] D. Y. Chao, T.-Y. Chen, J.-T. Chen, and K.-C. Wu, "A best deadlock control for S^3PMR to reach all states," *Asian J. Control*, vol. 14, no. 1, pp. 278–283, Jan. 2012, doi: 10.1002/asjc.370.

[9] X. Guan, W. Wu, and S. Wang, "New controllability condition for siphons in WS^3PR nets," *Asian J. Control*, vol. 17, no. 3, pp. 908–916, May 2015, doi: 10.1002/asjc.939.

[10] H. Hu, M. Zhou, and Z. Li, "Algebraic synthesis of timed supervisor for automated manufacturing systems using Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 549–557, Jul. 2010, doi: 10.1109/TASE.2009.2037825.

[11] Y.-S. Huang, T.-H. Chung, and P.-J. Su, "Synthesis of deadlock prevention policy using Petri nets reachability graph technique," *Asian J. Control*, vol. 12, no. 3, pp. 336–346, Feb. 2010, doi: 10.1002/asjc.188.

[12] G. Liu, "Complexity of the deadlock problem for Petri nets modeling resource allocation systems," *Inf. Sci.*, vol. 363, pp. 190–197, Oct. 2016.

[13] G. Liu and K. Barkaoui, "Necessary and sufficient liveness condition of GS^3PR Petri nets," *Int. J. Syst. Sci.*, vol. 46, no. 7, pp. 1147–1160, 2015.

[14] L. Yang, Z. Li, and A. Giua, "Containment of rumor spread in complex social networks," *Inf. Sci.*, vol. 506, pp. 113–130, Jan. 2020.

[15] X. Li, E. S. Abouel Nasr, A. Mohammed El-Tamimi, and Z. Li, "Adaptive consensus of two coupled heterogeneous networked systems with bidirectional actions," *IEEE Access*, vol. 8, pp. 35832–35841, 2020, doi: 10.1109/ACCESS.2020.2970866.

[16] X. Zan, Z. P. Wu, C. Guo, and Z. H. Yu, "A Pareto-based genetic algorithm for multi-objective scheduling of automated manufacturing systems," *Adv. Mech. Eng.*, vol. 12, no. 1, pp. 1–15, 2020, doi: 10.1177/1687814019885294.

[17] H. Kaid, A. Al-Ahmari, Z. Li, and R. Davidrajuh, "Intelligent colored token Petri nets for modeling, control, and validation of dynamic changes in reconfigurable manufacturing systems," *Processes*, vol. 8, no. 3, p. 358, Mar. 2020, doi: 10.3390/pr8030358.

[18] M. P. Fanti, G. Maione, and B. Turchiano, "Deadlock detection and recovery in flexible production systems with multiple capacity resources," in *Proc. 8th Medit. Electrotechnical Conf. Ind. Appl. Power Syst., Comput. Sci. Telecommun. (MELECON)*, vol. 1, May 1996, pp. 237–241, doi: 10.1109/MELCON.1996.550998.

[19] 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, doi: 10.1080/00207549408957073.

[20] N. Wu, M. Zhou, and Z. Li, "Resource-oriented Petri net for deadlock avoidance in flexible assembly systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 1, pp. 56–69, Jan. 2008, doi: 10.1109/TSMCA.2007.909542.

[21] K. Xing, M. Chu Zhou, H. Liu, and F. Tian, "Optimal Petri-net-based polynomial-complexity deadlock-avoidance policies for automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 1, pp. 188–199, Jan. 2009, doi: 10.1109/TSMCA.2008.2007947.

[22] 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, doi: 10.1109/TSMCA.2009.2013189.

[23] M. Yan and C. Zhong, "A deadlock prevention policy for a class of Petri net models of flexible manufacturing systems," *Int. J. Prod. Res.*, vol. 49, no. 24, pp. 7379–7396, Dec. 2011, doi: 10.1080/00207543.2010.513347.

[24] Y. Chen, Z. Li, K. Barkaoui, N. Wu, and M. Zhou, "Compact supervisory control of discrete event systems by Petri nets with data inhibitor arcs," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 2, pp. 364–379, Feb. 2017, doi: 10.1109/TSMC.2016.2521833.

[25] 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, doi: 10.1109/TASE.2013.2241762.

[26] D. Liu, Z. Li, and M. Zhou, "Hybrid liveness-enforcing policy for generalized Petri net models of flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 43, no. 1, pp. 85–97, Jan. 2013, doi: 10.1109/TSMCA.2012.2192266.

[27] Y. F. Hou, A. M. Al-Ahmari, and Z. W. Li, "Optimal controllability of 2-composed siphons in a class of Petri nets," *Electron. Lett.*, vol. 48, no. 24, pp. 1535–1537, Nov. 2012, doi: 10.1049/el.2012.2336.

[28] Z. Li and M. Shpitalni, "Smart deadlock prevention policy for flexible manufacturing systems using Petri nets," *IET Control Theory Appl.*, vol. 3, no. 3, pp. 362–374, Mar. 2009, doi: 10.1049/iet-cta:20070399.

[29] Z. Li, M. Uzam, and M. Zhou, "Deadlock control of concurrent manufacturing processes sharing finite resources," *Int. J. Adv. Manuf. Technol.*, vol. 38, nos. 7–8, pp. 787–800, Sep. 2008.

[30] Y.-S. Huang and Y.-L. Pan, "An improved maximally permissive deadlock prevention policy based on the theory of regions and reduction approach," *IET Control Theory Appl.*, vol. 5, no. 9, pp. 1069–1078, Jun. 2011, doi: 10.1049/iet-cta.2010.0371.

[31] M. Zhou, and K. Venkatesh, *Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach*. Singapore: World Scientific, 1998.

[32] T. Murata, "Petri nets: Properties, analysis and application," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.

[33] N. Wu and M. Zhou, "Schedulability analysis and optimal scheduling of dual-arm cluster tools with residency time constraint and activity time variation," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 1, pp. 203–209, Jan. 2012, doi: 10.1109/TASE.2011.2160452.

[34] N. Q. Wu and M. Zhou, "Modeling, analysis and control of dual-arm cluster tools with residency time constraint and activity time variation based on Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 2, pp. 446–454, Apr. 2012, doi: 10.1109/TASE.2011.2178023.

[35] M. G. Mehrabi, A. G. Ulsoy, and Y. Koren, "Reconfigurable manufacturing systems: Key to future manufacturing," *J. Intell. Manuf.*, vol. 11, pp. 403–419, Aug. 2000, doi: 10.1023/A:1008930403506.

[36] R. Katz, "Design principles of reconfigurable machines," *Int. J. Adv. Manuf. Technol.*, vol. 34, nos. 5–6, pp. 430–439, 2007.

[37] R. Patel, A. Gojiya, and D. Deb, "Failure reconfiguration of pumps in two reservoirs connected to overhead tank," in *Innovations in Infrastructure*. Berlin, Germany: Springer, 2019, pp. 81–92.

[38] Y. Wu, K. Xing, J. Luo, and Y. Feng, "Robust deadlock control for automated manufacturing systems with an unreliable resource," *Inf. Sci.*, vols. 346–347, pp. 17–28, Jun. 2016, doi: 10.1016/j.ins.2016.01.049.

[39] G. Y. Liu, Z. W. Li, K. Barkaoui, and A. M. Al-Ahmari, "Robustness of deadlock control for a class of Petri nets with unreliable resources," *Inf. Sci.*, vol. 235, pp. 259–279, Jun. 2013, doi: 10.1016/j.ins.2013.01.003.

[40] 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.*, vol. 49, no. 7, pp. 1371–1385, Jul. 2019, doi: 10.1109/TSMC.2018.2815618.

[41] G. Liu, P. Li, N. Wu, and L. Yin, "Two-step approach to robust deadlock control in automated manufacturing systems with multiple resource failures," *J. Chin. Inst. Eng.*, vol. 41, no. 6, pp. 452–462, Oct. 2018, doi: 10.1080/02533839.2018.1498023.

[42] S. Lee and D. M. Tilbury, "Deadlock-free resource allocation control for a reconfigurable manufacturing system with serial and parallel configuration," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1373–1381, Nov. 2007, doi: 10.1109/TSMCC.2007.905843.

[43] J. Li, X. Dai, and Z. Meng, "Automatic reconfiguration of Petri net controllers for reconfigurable manufacturing systems with an improved net rewriting system-based approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 1, pp. 156–167, Jan. 2009, doi: 10.1109/TASE.2008.2006857.

[44] B. Tarnauca, D. Puiu, V. Comnac, and C. Suciu, "Modelling a flexible manufacturing system using reconfigurable finite capacity Petri nets," in *Proc. 13th Int. Conf. Optim. Electr. Electron. Equip. (OPTIM)*, Brasov, Romania, May 2012, pp. 1079–1084, doi: 10.1109/OPTIM.2012.6231954.

[45] R. M. D. Silva, I. F. Benítez-Pina, M. F. Blos, D. J. S. Filho, and P. E. Miyagi, "Modeling of reconfigurable distributed manufacturing control systems," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 1284–1289, 2015.

[46] M. S. Elsayed, P. E. Kafrawy, and N. Wu, "Modeling and deadlock control of reconfigurable multi-unit resource systems," *IEEE Access*, vol. 8, pp. 133605–133621, 2020, doi: 10.1109/ACCESS.2020.3010237.

[47] G. Liu, Z. Li, and C. Zhong, "New controllability condition for siphons in a class of generalised Petri nets," *IET Control Theory Appl.*, vol. 4, no. 5, pp. 854–864, May 2010, doi: 10.1049/iet-cta.2009.0264.

[48] 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, doi: 10.1109/70.370500.

[49] X. Guo, S. Wang, D. You, Z. Li, and X. Jiang, "A siphon-based deadlock prevention strategy for S^3PR," *IEEE Access*, vol. 7, pp. 86863–86873, 2019, doi: 10.1109/ACCESS.2019.2920677.

[50] Y. Feng, K. Xing, M. Zhou, and H. Liu, "Liveness analysis and deadlock control for automated manufacturing systems with multiple resource requirements," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 2, pp. 525–538, Feb. 2020, doi: 10.1109/TSMC.2017.2767902.

[51] Y. F. Hou, Z. W. Li, L. Hong, and A. M. Al-Ahmari, "Optimal controllability of 3-composed siphons in a class of Petri nets," *Electron. Lett.*, vol. 49, no. 11, pp. 697–699, May 2013, doi: 10.1049/el.2012.4512.

[52] Y. Hou, Z. Li, and A. Al-Ahmari, "Extended elementary siphons and their application to liveness-enforcement of generalized Petri nets," *Assian J. Control*, vol. 16, no. 6, pp. 1–22, 2014, doi: 10.1002/ASJC.801.

[53] Y. Hou, Z. Li, M. Zhao, and D. Liu, "Extended elementary siphon-based deadlock prevention policy for a class of generalized Petri nets," *Int. J. Comput. Integ. Manuf.*, vol. 27, no. 1, pp. 85–102, 2014.

[54] Z. Li and M. 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, Jan. 2004, doi: 10.1109/TSMCA.2003.820576.

[55] Z. Li and M. Zhou, "Clarifications on the definitions of elementary siphons in Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 36, no. 6, pp. 1227–1229, Nov. 2006, doi: 10.1109/TSMCA.2006.878966.

[56] M. Zhou and K. Venkatesh, *Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach*. Singapore: World Scientific, 1998.

[57] F. T. S. Chan and R. Swarnkar, "Ant colony optimization approach to a fuzzy goal programming model for a machine tool selection and operation allocation problem in an FMS," *Robot. Comput.-Integr. Manuf.*, vol. 22, no. 4, pp. 353–362, Aug. 2006.

[58] D. G. Mogale, A. Dolgui, R. Kandhway, S. K. Kumar, and M. K. Tiwari, "A multi-period inventory transportation model for tactical planning of food grain supply chain," *Comput. Ind. Eng.*, vol. 110, pp. 379–394, Aug. 2017.

[59] D. G. Mogale, S. K. Kumar, and M. K. Tiwari, "Two stage Indian food grain supply chain network transportation-allocation model," *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 1767–1772, 2016.

[60] G. Tian, H. Zhang, Y. Feng, H. Jia, C. Zhang, Z. Jiang, Z. Li, and P. Li, "Operation patterns analysis of automotive components remanufacturing industry development in China," *J. Cleaner Prod.*, vol. 164, pp. 1363–1375, Oct. 2017, doi: 10.1016/j.jclepro.2017.07.028.

[61] G. Tian, H. Zhang, M. Zhou, and Z. Li, "AHP, gray correlation, and TOPSIS combined approach to green performance evaluation of design alternatives," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 7, pp. 1093–1105, Jul. 2018, doi: 10.1109/TSMC.2016.2640179.

[62] M. B. Yildirim, T. Cakar, U. Doguc, and J. C. Meza, "Machine number, priority rule, and due date determination in flexible manufacturing systems using artificial neural networks," *Comput. Ind. Eng.*, vol. 50, nos. 1–2, pp. 185–194, May 2006.

[63] X. Xiao and J. J. Lee, "A parallel multi-unit resource deadlock detection algorithm with $O(\log_2(\min(m,n)))$ overall run-time compexity," *J. Parallel Distrib. Comput.*, vol. 71, no. 7, pp. 938–954, Jul. 2011.

[64] R. C. Holt, "Some deadlock properties of computer systems," *ACM Comput. Surv.*, vol. 4, no. 3, pp. 179–196, Sep. 1972.

[65] K. Barkaoui and J. F. Pradat-Peyre, "On liveness and controlled siphons in Petri nets," in *Proc. 17th Int. Conf. Appl. Theory Petri Nets*, in Lecture Notes in Computer Science, vol. 1091. Berlin, Germany: Springer, 1996, pp. 57–72.

[66] Z. Li and M. Zhou, *Deadlock Resolution in automated Manufacturing Syatems: A Novel Petri Net Approach*. London, U.K.: Springer, 2009.

[67] Z. Li, J. Zhang, and M. Zhao, "Liveness-enforcing supervisor design for a class of generalised Petri net models of flexible manufacturing systems," *IET Control Theory Appl.*, vol. 1, no. 4, pp. 955–967, Jul. 2007.

[68] X. Cong, C. Gu, M. Uzam, Y. Chen, A. M. Al-Ahmari, N. Wu, M. Zhou, and Z. Li, "Design of optimal Petri net supervisors for flexible manufacturing systems via weighted inhibitor arcs," *Asian J. Control*, vol. 20, no. 1, pp. 511–530, Jan. 2018, doi: 10.1002/asjc.1583.

[69] Q. Zhuang, W. Dai, S. Wang, J. Du, and Q. Tian, "An MIP-based deadlock prevention policy for siphon control," *IEEE Access*, vol. 7, pp. 153782–153790, 2019, doi: 10.1109/ACCESS.2019.2939855.

[70] H. Kaid, A. Al-Ahmari, A. M. El-Tamimi, E. A. Nasr, and Z. Li, "Design and implementation of deadlock control for automated manufacturing systems," *South Afr. J. Ind. Eng.*, vol. 30, no. 1, pp. 1–23, May 2019.

[71] D. Sun, Y. Chen, M. A. El-Meligy, M. A. F. Sharaf, N. Wu, and Z. Li, "On algebraic identification of critical states for deadlock control in automated manufacturing systems modeled with Petri nets," *IEEE Access*, vol. 7, pp. 121332–121349, 2019, doi: 10.1109/ACCESS.2019.2936052.

[72] A. Al-Ahmari, H. Kaid, Z. Li, and R. Davidrajuh, "Strict minimal siphon-based colored Petri net supervisor synthesis for automated manufacturing systems with unreliable resources," *IEEE Access*, vol. 8, pp. 22411–22424, 2020, doi: 10.1109/ACCESS.2020.2968469.

[73] L. Li, F. Basile, and Z. Li, "An approach to improve permissiveness of supervisors for GMECs in time Petri net systems," *IEEE Trans. Autom. Control*, vol. 65, no. 1, pp. 237–251, Jan. 2020, doi: 10.1109/TAC.2019.2914895.

[74] Y. Liu, K. Cai, and Z. Li, "On scalable supervisory control of multi-agent discrete-event systems," *Automatica*, vol. 108, Oct. 2019, Art. no. 108460, doi: 10.1016/j.automatica.2019.06.012.

[75] Q. Chen, L. Yin, N. Wu, M. A. El-Meligy, M. A. F. Sharaf, and Z. Li, "Diagnosability of vector discrete-event systems using predicates," *IEEE Access*, vol. 7, pp. 147143–147155, 2019, doi: 10.1109/ACCESS.2019.2945132.

[76] D. Wang, X. Wang, and Z. Li, "Nonblocking supervisory control of state-tree structures with conditional-preemption matrices," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 3744–3756, Jun. 2020, doi: 10.1109/TII.2019.2939628.

[77] Y. Hu, Z. Ma, and Z. Li, "Design of supervisors for active diagnosis in discrete event systems," *IEEE Trans. Autom. Control*, vol. 65, no. 12, pp. 5159–5172, Dec. 2020, doi: 10.1109/TAC.2020.2970011.

[78] P. H. Starke. (2003). *INA: Integrated Net Analyzer*. [Online]. Available: http://www2.informatik.hu-berlin.de/~starke/ina.html

MAHMOUD SALAHELDIN ELSAYED received the B.S. degree in mathematics and computer science and the M.S. degree in computer science from the Faculty of Science, Menoufia University, Menoufia, Egypt, in 2000 and 2005, respectively, where he is currently pursuing the Ph.D. degree.

From 2006 to 2008, he was with the Teachers College, Arar, Saudi Arabia. Since 2008, he has been a Lecturer of computer science with the Department of Computer Science, Faculty of Computing and Information Technology, Northern Border University, Rafha, Saudi Arabia. He has published many papers, including international journals and conference ones, which all focus on deadlock control policies for single and multi-unit resource systems. His research interests include distributed database systems, Petri nets, resource allocation systems, and control of automated manufacturing, discrete-event, and hybrid systems. He received the Excellent Paper Award from the International Institute of Engineers and Researchers (IIER), in 2016, and the book on *Deadlock Analysis and Resolution in Distributed Systems* from the LAP LAMBERT Academic Publishing, Germany, in 2017.

GAIYUN LIU (Senior Member, IEEE) received the B.S. degree in measurement and control technology and instrumentation and the Ph.D. degree in mechanical engineering from Xidian University, Xi'an, China, in 2006 and 2011, respectively.

In 2011, she joined Xidian University, where she is currently an Associate Professor with the School of Electro-Mechanical Engineering. She has authored or coauthored 26 publications. Her research interests include Petri net theory and applications, supervisory control of discrete event systems, and robust supervisory control of automated manufacturing systems.

ALMETWALLY M. MOSTAFA received the B.Sc. degree in computers and systems engineering and the M.Sc. degree from Al-Azhar University, Egypt, and the Ph.D. degree in computers and systems from a channel program between the Université catholique de Louvain, Belgium, and Al-Azhar University. Before he joined King Saud University (KSU), he worked two years at the Centre d'Excellence en Technologies de l'Information et de la Communication (CETIC), Belgium. He is currently an Associate Professor with the Department of Information Systems, College of Computers and Systems (IS/CCS), KSU. His research interests include distributed systems protocols, cloud–mobile-pervasive computing, fault-tolerance, and security of information systems.

ABEER ALI ALNUAIM received the M.Sc. degree in advanced computing and the Ph.D. degree in computer science in U.K. She held a number of academic and administrative roles. She is currently an Assistant Professor with the Department of Computer Science and Engineering, King Saud University. Her research interests include soft computing, distributed systems, HCI, and user experience.

PASSENT EL KAFRAWY (Senior Member, IEEE) received the bachelor's degree from the Department of Computer Science and Engineering, American University in Cairo, the master's degree from the Faculty of Science, Menoufia University, and the Ph.D. degree in Computer Science and Engineering, especially in computational geometry and artificial intelligence from the University of Connecticut, USA, in 2006.

In 2007, she worked as an Assistant Professor with the Department of Mathematics and Computer Science, Faculty of Science, Menoufia University. In 2011, she joined The American University in Cairo, as an Adjunct Professor with the Department of Computer Science and Engineering. In 2013, she was appointed as an Associate Professor. Then, she taught at the Eastern State University of Connecticut for one year. Since 2018, she has been a Professor with the School of Information Technology and Computer Science, Nile University. She is supervising several research studies between the Ph.D. and M.Sc. degrees in the field of natural language processing, semantic knowledge, bioinformatics, big data analytics, and knowledge mining and acquisition. She has joined the IBRO School for neurodegenerative physician training organized by ENND and personalized medicine workshop organized by AUC. She has over 45 publications and edited three books. Her research interests include software engineering, bioinformatics, big data analytics, machine learning, and cloud computing.

Dr. El Kafrawy is a member of the Egyptian Society of Language Engineering and the Editor In Chief of the *Egyptian Journal of Language Engineering*.