applied sciences

MDPI

Article

Automatic Supervisory Controller for Deadlock Control in Reconfigurable Manufacturing Systems with Dynamic Changes

Husam Kaid [1,*], Abdulrahman Al-Ahmari [1], Zhiwu Li [2,3] and Reggie Davidrajuh [4]

1 Department of Industrial Engineering, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia; alahmari@ksu.edu.sa
2 Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau 999078, China; systemscontrol@gmail.com
3 School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China
4 Department of Electrical Engineering and Computer Science, Faculty of Science and Technology, University of Stavanger, 4036 Stavanger, Norway; reggie.davidrajuh@uis.no
* Correspondence: yemenhussam@yahoo.com

Received: 10 July 2020; Accepted: 27 July 2020; Published: 30 July 2020

check for updates

Abstract: In reconfigurable manufacturing systems (RMSs), the architecture of a system can be modified during its operation. This reconfiguration can be caused by many motivations: processing rework and failures, adding new products, adding new machines, etc. In RMSs, sharing of resources may lead to deadlocks, and some operations can therefore remain incomplete. The objective of this article is to develop a novel two-step solution for quick and accurate reconfiguration of supervisory controllers for deadlock control in RMSs with dynamic changes. In the first step, the net rewriting system (NRS) is used to design a reconfigurable Petri net model under dynamic configurations. The obtained model guarantees boundedness behavioral property but may lose the other properties of a Petri net model (i.e., liveness and reversibility). The second step develops an automatic deadlock prevention policy for the reconfigurable Petri net using the siphon control method based on a place invariant to solve the deadlock problem with dynamic structure changes in RMSs and achieve liveness and reversibility behavioral properties for the system. The proposed approach is tested using examples in the literature and the results highlight the ability of the automatic deadlock prevention policy to adapt to RMSs configuration changes.

Keywords: reconfigurable manufacturing system; Petri net; deadlock; siphon; supervisory controller

1. Introduction

A typical example of discrete event systems is an automated manufacturing system (AMS) [1,2]. It enables various product types to be entered at discrete times by sharing resources like machines, automatic controlled vehicles, automated tools, robots, and buffers at asynchronous or simultaneous operations. AMSs have to cope with unexpected and rapid market changes on a competitive global market. They must make rapid modifications to their software and hardware to meet these dynamic changes. This requirement cannot, however, be satisfied successfully with traditional automated manufacturing systems, which require large capital investments. Reconfigurable manufacturing systems have now been developed to deal with those drawbacks in traditional automated manufacturing systems [3–5]. Reconfigurable manufacturing systems are a new kind of production systems that are randomly and dynamically configured in real time. Such configurations involve processing rework and failures, adding new products and machines, and adding new handling device. In RMSs, a set of system resources can be used to process each component according to a specific process sequence.

This sharing of resources, however, may lead to deadlocks, and some operations can therefore remain incomplete. Therefore, dealing with deadlock problem is critical for RMSs.

Petri nets (PNs) are widely used for the scheduling, deadlock analysis and control in AMSs as graphical and mathematical modelling tools [6–14]. They can be used to describe characteristics and behaviors of AMSs such as synchronization, concurrency, conflict, causal dependence, and sequencing. Petri nets can be used for behavioral features, for example boundedness and liveness [15,16]. From a technical point of view, several policies based on Petri nets have been proposed. These policies are based on three strategies: (i) deadlock detection and recovery, (ii) deadlock avoidance, and (iii) deadlock prevention [15,17]. Most of these policies have proposed deadlock control in Petri nets through structural analysis [6,18] and reachability graph analysis [19–21]. In addition, three criteria to evaluate and construct an AMS supervisor have been proposed, namely behavioral permissiveness, computational complexity, and structural complexity [15,22].

Recently, several approaches have been adapted to deal with dynamic changes in manufacturing systems [7,23–36]. They primarily concentrate in two directions: direct and indirect. Direct approaches provide modification mechanisms or particular rules for system structure configurations, while indirect approaches typically import additional mechanisms for system reconfiguration specifications. The event–condition–action (ECA) paradigm is developed by Almeida et al. [30] for the design of reconfigurable logic controllers. Their research has demonstrated that the reconfiguration process is highly dependent on the modularity level of the logical control system and that not all "modular" structures can be reconfigured. For a class of discrete event systems (DESs), Sampath et al. [26] presented a reconfiguration approach for their control specifications, subject to linear constraint. This approach is suited to systems such as hospital management systems and can be reconfigured in non-real time. In order to evaluate and improve the performance of the control architecture, Dumitrache et al. [27] developed a real-time reconfigurable supervised control architecture for large manufacturing systems. A model-based control design for reconfigurable manufacturing systems is developed by Ohashi and Shin [28] through state transition diagrams and general graph representation taking into account configuration and reuse of design data. Kalita and Khargonekar [29] introduced a hierarchical structure and a framework for modeling, analysis, specification, and design of logic controllers for RMSs, which allows rapid reconfigurability and reusability of the controller during reconfiguration. In [23], reconfigurable manufacturing systems were used to replace the existing manufacturing systems to offer higher convertibility and flexibility such as dedicated production systems. Serial and parallel configurations, a rules-based matrix approach has been developed and implemented. In addition, a higher-level deadlock control method is presented for the serial and parallel configurations.

Net Rewriting Systems (NRS) are another graph-based reconfiguration mechanism [34]. In terms of pattern matching and dynamic structure replacements, the reconfiguration occurs. By the implementation of a Turing machine the expressive power was shown to be Turing equivalent. A subset of net rewriting systems, called reconfigurable nets, have also been provided with an algorithm to flatten a Petri net to standard. This subset only restricts NRS to those transformations that remain unchanged in the number of places and transitions, that is, only the flow relation can be changed. Flattening significantly increases the size of transitions by multiplying the number of reconfigurations by the amount of transitions. The NRS is used in logic controllers with improved net rewriting systems [35]. The improved NRS version restricts the rewriting rules to ensure important structural characteristics such as boundedness, liveness, and reversibility are not invalidated. In addition, in [24], an improved net rewriting system (INRS) was developed with the aim of reconfiguring an RMS supervisory controller based on PNs. Changes to an RMS modification were made to rewrite rules that were then applied in the initial PN controller. The INRS is first proposed as a reconfiguration basis. The structure of a Petri net model can be changed dynamically. Then, the study provided three representations of the RMS modification and suggested an INRS-based method to the design of the Petri net controller of an RMS. In this approach, the properties of behavioral, i.e., the boundedness, reversibility, and liveness of a modified system, were not verified or validated.

In [31], colored timed PNs (CTPN) were used in the modelling of RMSs and a mechanism to describe reconfigurability in the CTPN architecture was introduced that leads to a new architecture supporting the reconfiguration. This mechanism includes reconfigurable transitions, specific places, and inhibitor arcs. Wu and Zhou introduced intelligent token Petri net (ITPN) [25]. In their model, tokens representing job instances carry real-time knowledge about system states and changes, just like intelligent cards in practice such that dynamical changes of a system can be easily modeled. These formalisms can describe the reconfiguration behavior of the system. However, some of dynamic changes do not clearly define the modularity, which brings confusion to engineers in designing, understanding, and future redevelopment. Correctness of the system such as coherence of states before and after system reconfigurations is not considered. In addition, temporal constraints, which are of great significance in real-time systems are not mentioned. In [32], reconfigurable object nets (RONs) are used to model, simulate, and analyze RMSs. A formal method was proposed for fulfilling a new production requirement. The configuration consists of new extrusion and cutting machines. The reconfiguration is represented as graph transformations, RON tool was used to simulate the reconfigured systems and TINA [37] and PIPE [38] software tools were used to carry out the analysis.

The work of Silva et al. [36] explored the principles of the different approaches and takes from them the best practices. Configuration mechanisms were proposed using Holonic and multiagent system methods to allow a reconfigurable distributed production control system to systematically detect faults. To describe communication interfaces, the principle of service-oriented architecture was used. Hybrid top-down and bottom-up approaches were presented using Petri net models. In [33], object-oriented Petri nets (ORPNs) and π-calculus were used as two complementary formalisms. Initial RMSs structure and system behavior were modeled by ORPN while the π-calculus was used to describe RMSs' reconfiguration. To evaluate, check, and validate RMSs, Petri nets and π-calculus supporting tools were used. The reconfigurability mechanism and consistency of RMSs could be analyzed by π-calculus. In [7], a new model is proposed, namely the intelligent colored token Petri net (ICTPN), which simulated dynamic configurations of systems such as adding new machines, processing failures and rework, machine failures, processing routes changes, removing old machines, and adding new products. The primary idea is that smart colored tokens were part types which represented real-time knowledge of system status and configurations. This allowed for the effective modeling of dynamic system configurations. The proposed ICTPN could modularly model dynamic system changes to generate a very compact model. Moreover, when configurations appear, only the colored token of the part type, which is changed from the current model was changed. The resulting ICTPN model ensures that the behavioral properties such as deadlock-free, conservative, and reversible were guaranteed.

All of the above methods with PNs attempted to deal with dynamic configuration issues in manufacturing systems. However, most of them do not include an algorithm or mechanism for reconfiguration, could not guarantee the properties of behavioral Petri net (i.e., boundedness (or safeness), liveness, and reversibility), or could not ensure that the results of the reconfiguration are correct, accurate or valid. In addition, few techniques for rapid and valid reconfiguration of literature deadlock control supervisors were presented.

The objective of this article is to develop a novel two-step solution for quick and accurate reconfiguration of supervisory controllers for deadlock control in RMSs with dynamic changes. In the first step, the net rewriting system used in [34,39] was adapted to design a reconfigurable Petri net model under dynamic configurations. The obtained model guarantees boundedness behavioral property but may lose the other properties of a Petri net model (i.e., liveness and reversibility). This means that the reconfigured Petri net model has finite states, deadlocks, and does not behave cyclically. For this issue, the second step develops an automatic deadlock prevention policy for reconfigurable Petri net using the siphon control method based on place invariant to solve the deadlock problem with dynamic structure changes in RMSs and achieve liveness and reversibility behavioral properties for the system. Thus, the developed approach has the ability of adapting to RMS configuration changes.

The major applications of the developed approach are as follows:

1. Mass customization manufacturing can use the proposed approach to address its difficulties. For example, by trying to make products available rapidly to consumers, a high quality production of a wide variety of products can be maintained and achieve low costs in line with standard products.
2. Lean productivity concept can also use the proposed approach to enable a company to implement an RMS in order to improve the exploitation of the part of the resources for various family products and to minimize waste from the idle resource of an RMS.
3. Agile manufacturing can use the proposed approach to facilitate rapid products changeovers, rapid introduction of new products and unattended operation.
4. Flexible manufacturing systems can use the proposed approach to increase response to a variety of customers and markets. Moreover, scalability to the desired volume of products and convertibility to current systems, machines, robots, and controls are increased in accordance with the new production requirements.

This paper is organized as follows. Section 2 describes basic concepts of Petri nets, reconfigurable Petri nets. Section 3 presents the deadlock prevention policy for reconfigurable Petri net based on the concept of minimal siphons and place invariants. The behavioral and quantitative analysis of the proposed reconfigurable Petri net are presented in Section 4. A real-world case study is presented in Section 5 to demonstrate the application of the proposed approach. Conclusions and future research are presented in Section 6.

2. Preliminaries

2.1. S^3PR NET

Definition 1. *A simple sequential process (S^2P) is a Petri net model with $N = (\{p^0\} \cup P_A, T, F)$ if (1) N is a strongly connected state machine and (2) each circuit N contains place p^0, where p^0 is a process idle place, $P_A = \{p_1, p_2, \ldots, p_m\}$ is a set of operation places, $T = \{t_1, t_2, \ldots, t_n\}$ is a set of transitions, $P_B = P_A \cup \{p^0\}$, $P_B \cap T = \emptyset$, $P_B \cup T \neq \emptyset$, and $F: (P_B \times T) \cup (T \times P_B) \rightarrow \textbf{IN}$ is a set of weighted arcs called flow relations, where $\textbf{IN} = \{0, 1, 2, \ldots\}$.*

Definition 2. *A simple sequential process with resources (S^2PR) is a Petri net model with $N = (\{p^0\} \cup P_A \cup P_R, T, F)$ if*

1. *the subnet created by $Y = P_A \cup \{p^0\} \cup T$ is an S^2P;*
2. *$P_R \neq \emptyset$ and $(P_A \cup \{p^0\}) \cap P_R = \emptyset$, where P_R is called a set of resource places;*
3. *$P_C = P_A \cup \{p^0\} \cup P_R$, $F \subseteq (P_C \times T) \cup (T \times P_C)$ is flow relations;*
4. *$^{\bullet\bullet}(p^0) \cap P_R = (p^0)^{\bullet\bullet} \cap P_R \neq \emptyset$;*
5. *$\forall p \in P_A$, $\forall t \in {}^{\bullet}p$, $\forall t' \in p^{\bullet}$, $\exists r_p \in P_R$, ${}^{\bullet}t \cap P_R = t'^{\bullet} \cap P_R = \{r_p\}$;*
6. *$\forall r \in P_R$, $^{\bullet\bullet}r \cap P_A = r^{\bullet\bullet} \cap P_A \neq \emptyset$ and ${}^{\bullet}r \cap r^{\bullet} \neq \emptyset$;*

Definition 3. *Let $N = (\{p^0\} \cup P_A \cup P_R, T, F)$ be an S^2PR with M_o being an initial marking of net N. An S^2PR is called acceptably marked if (1) $M_o(p^0) \geq 1$, (2) $M_o(p) = 0$, $\forall p \in P_A$, and (3) $M_o(r) \geq 1$, $\forall r \in P_R$.*

Recursively, a system of S^2PR is called an S^3PR.

Definition 4. *A system of S^2PR, S^3PR, is defined recursively as follows:*

1. *An S^2PR is an S^3PR;*
1. *Let $N_i = (\{p^0{}_i\} \cup P_{Ai} \cup P_{Ri}, T_i, F_i)$, $i = \{1, 2\}$, be two S^3PRs such that $(\{p^0{}_1\} \cup P_{A1}) \cap (\{p^0{}_2\} \cup P_{A2}) = \emptyset$, $P_{R1} \cap P_{R2} = P_D$, $P_{A1} \cap P_{A2} \neq P_D$, and $T_1 \cap T_2 \neq \emptyset$; then, the net $N = (\{p^0\} \cup P_A \cup P_R, T, F)$ is an S^3PR resulting from the integration of N_1 and N_2 by the set of common P_D (denoted as $N_1 \circ N_2$) and expressed as: (1) $p^0 = \{p^0{}_1\} \cup \{p^0{}_2\}$, (2) $P_A = P_{A1} \cup P_{A2}$, (3) $P_R = P_{R1} \cup P_{R2}$, (4) $T = T_1 \cup T_2$, and (5) $F = F_1 \cup F_2$.*

The integration of n S^2PR N_1-N_n via P_D is expressed by $\otimes_{i=1}^{n} N_i$. $\overline{N_i}$ is used to indicate the S^2P from which the S^2PR N_i is built.

Definition 5. *Let $N_i = (\{p^0{}_i\} \cup P_{Ai} \cup P_{Ri}, T_i, F_i)$, $i = \{1, 2\}$, be two S^3PRs. M_o is an initial marking of N. (N, M_o) is called acceptably marked if (1) (N, M_o) is an acceptably marked S^2PR, and (2) $N = N_1 \circ N_2$, where (N_i, M_{io}) is called an acceptably marked S^3PR and*

1. $\forall i \in \{1,2\}$, $\forall p \in P_{Ai} \cup \{p^0{}_i\}$, $M_o(p) = M_{io}(p)$.
2. $\forall i \in \{1,2\}$, $\forall r \in P_{Ri} \backslash P_D$, $M_o(r) = M_{io}(r)$.
3. $\forall i \in \{1,2\}$, $\forall r \in P_D$, $M_o(p) = max\{M_{1o}(r), M_{2o}(r)\}$.

Definition 6. *Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, W, M_o)$ be an S^3PR, where $W: (P_C \times T) \cup (T \times P_C) \rightarrow$ **IN** is a mapping that assigns a weight to an arc and M_o: $P_C \rightarrow$ **IN** is the initial marking.*

Definition 7. *Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, W, M_o)$ be an S^3PR. N is said to be an ordinary net if $p \in P_C$, $t \in T$, $\forall(p, t) \in F$, and $W(p, t) = 1$.*

Definition 8. *Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, W, M_o)$ be an S^3PR. N is said to be a weighted net if $\exists p \in P_C$, $\exists t \in T$, $(p, t) \in F$, and $W(p, t) > 1$.*

Definition 9. *Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, W, M_o)$ be an S^3PR, where p and t are a place and a transition in N, respectively. The preset (postset) of p is the set of all input (output) transitions of p, i.e., ${}^\bullet p = \{t \in T \mid (t, p) \in F\}$($p^\bullet = \{t \in T \mid (p, t) \in F\}$). The preset (postset) of t is the set of all input (output) places of t, i.e., ${}^\bullet t = \{p \in P_C \mid (p, t) \in F\}$($t^\bullet = \{p \in P_C \mid (t, p) \in F\}$).*

Definition 10. *Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, W, M_o)$ be an S^3PR. N is self-loop free if for all $p, t \in P_C \cup T$; $W(p, t) > 0$ implies $W(t, p) = 0$ and has a self-loop if $W(t, p) > 0$.*

Definition 11. *Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, W, M_o)$ be an S^3PR and M be a marking of N, where M is a mapping M: $P_C \rightarrow$ **IN** and the pth element of M, expressed by M(p), is the number of tokens in place p.*

Definition 12. *Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, W, M_o)$ be an S^3PR. A transition $t \in T$ is enabled if $\forall p \in {}^\bullet t$, $M(p) \geq W(p, t)$.*

Definition 13. *Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, W, M_o)$ be an S^3PR. The marking M′ resulting from the firing of an enabled transition $t \in T$ at marking M is denoted by $M[t\rangle M'$ and expressed as follows:*

$$M'(p) = \begin{bmatrix} M(p) + W(p,\ t) & \text{if } p \in {}^\bullet t \backslash t^\bullet \\ M(p) - W(t,\ p) & \text{if } p \in t^\bullet \backslash {}^\bullet t \\ M(p) + W(t,\ p) - W(p,\ t) & \text{if } p \in t^\bullet \cap {}^\bullet t \\ M(p) & \text{otherwise} \end{bmatrix} \tag{1}$$

Definition 14. *Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, W, M_o)$ be an S^3PR. R(N, M) is a set of reachable markings from M in N, which is expressed by nodes and arcs; nodes represent markings that are labeled with M_i and arcs represent transition firings that are labeled with t. If t fires, then there is an arc from marking M_i to marking M_j and M_j is reached.*

Definition 15. *Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, W, M_o)$ be an S^3PR. A transition $t \in T$ is live at M_o if $\forall M \in R(N, M_o)$, $\exists M' \in R(N, M)$ such that $M'[t\rangle$ holds. (N, M_0) is dead at M_o if there does not exist $t \in T$ such that $M_o[t\rangle$ holds. (N, M_0) is weakly live or live-locked if $\forall M \in R(N, M_o)$, $\exists t \in T$, $M[t\rangle$ holds. (N, M_0) is quasi-live if $\forall t \in T$, $\exists M \in R(N, M_o)$ such that $M[t\rangle$ holds.*

Definition 16. *Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, W, M_o)$ be an S^3PR. [N] is said to be the incidence matrix of net N, where [N] is a $|P| \times |T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$. For a place p (transition t), its incidence vector, a row (column) in [N], is expressed as [N](p, .) ([N](., t).*

Definition 17. *Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, W, M_o)$ be an S^3PR. A marking M' is called reachable from M if there exists a sequence of transitions $\delta = t_1\ t_2\ t_3 \ldots\ t_n$ that can be fired, and markings $M_1, M_2, M_3, \ldots$, and M_{n-1} are such that $M[t_o\rangle M_1[t_1\rangle M_2[t_2\rangle M_3 \ldots M_n\ [t_n\rangle M'$ holds, expressed as $M[\delta\rangle M'$, satisfies the state equation $M' = M + [N]\ \vec{\delta}$. $\vec{\delta}: T \to \mathbf{IN}$ is called a firing count vector or a Parikh vector that maps t in T to the number of occurrences of t in δ.*

Definition 18. *Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, W, M_o)$ be an S^3PR. N is said to be bounded if there exists $q \in \mathbf{IN}$, $\forall M \in R(N, M_0)$, $\forall p \in P_C$, $M(p) \le q$. (N, M_0) is structurally bounded if it is bounded for any M_o.*

Definition 19. *LLet $N = (\{p^0\} \cup P_A \cup P_R, T, F, W, M_o)$ be an S^3PR. N is called safe if $\forall M \in R(N, M_0)$, $\forall p \in P_C$, $M(p) \le 1$. (N, M_0) is q -safe if it is q-bounded.*L

Consider the example of AMS illustrated in Figure 1a. The system has one robot R1 and one machine M1. Machine M1 processes one part at a time and robot R1 holds one part at a time. There are buffers for loading/unloading. Furthermore, one part type is considered to be processed in the system. The part operation sequence is illustrated in Figure 1b. Figure 2 shows the S^3PR net of the AMS example. It has six places and four transitions. The following sets of places can be used: $P^0 = \{p_1\}$, $P_R = \{p_5, p_6\}$, and $P_A = \{p_2, p_3, p_4\}$. There are five reachable markings on the Petri model. The initial marking is $M_o = (5, 0, 0, 0, 1, 1)^T$, which represents the different raw parts that are to be processed synchronously within the system, including preconditions, input signals, buffers and resource status, such as machines and robot. Places are generally used to represent the resource status, operations, and activities. The transitions are used to express control changes from one state to another. Directed arcs correspond to the material, resource, information flow, and control flow direction between states. Material, information, and resources are represented by tokens.

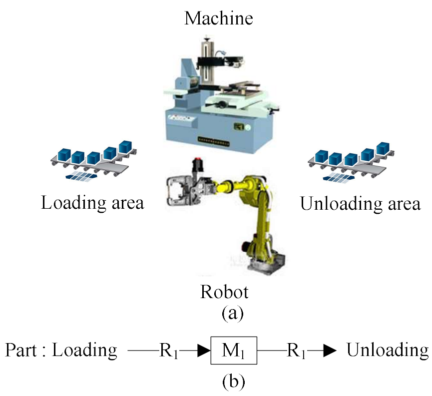

Figure 1. (**a**) Automated manufacturing system (AMS) example and (**b**) operation sequence.

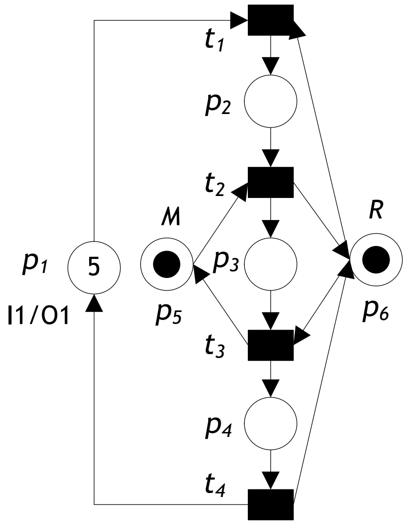

Figure 2. A system of (S^2P) (simple sequential process) (S^3PR) net of the AMS.

2.2. Reconfigurable S^3PR Net

This section presents definitions and theorems in the reconfigurable S^3PR nets, which are originally proposed by [34,35,39].

Definition 20. *Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, W, M_o, K)$ be a finite-capacity S^3PR, where p^0, P_A, P_R, T, F, W, and M_o are defined in Definitions 1–6. K: $P_C \rightarrow$ **IN** is the function of capacity that assigns to each place p the maximal number of tokens K(p).*

Definition 21. *Let (N_i, M_i) be two S^3PR nets with $N_i = (P_{Ci}, T_i, F_i, W_i, M_i, K_i)$, i= 1, 2. N_1 and N_2 are called morphism nets if there exists a bijection Ψ: $N_1 \rightarrow N_2$, $\Psi = (\Psi_{PC}: P_{C1} \rightarrow P_{C2}, \Psi_T: T_1 \rightarrow T_2)$ such that for all a, $b \in P_{C1} \cup T_1$, $F_1(a, b) \in N_1 = F_2(\Psi(a), \Psi(b)) \in N_2$, and for all $p \in P_{C1}$, $M_1(p) \leq M_2(\Psi_{PC}(p))$.*

Definition 22. *Let (N_i, M_i) be two S^3PR nets with $N_i = (P_{Ci}, T_i, F_i, W_i, M_i, K_i)$, i = 1, 2. N_1 is called the full subnet of N_2 if there exists an injection function that maps places to places and transitions to transitions, denoted by ξ: $N_1 \rightarrow N_2$, $\xi(P_{C1}) \subseteq P_{C2}$, and $\xi(T_1) \subseteq T_2$ such that for all a, $b \in P_{C1} \cup T_1$, $F_1(a, b) = F_2(\xi(a), \xi(b))$.*

In the algebraic, a rewriting rule is a transformation approach that can change and combine the Petri nets dynamically. The main idea is to define and change the system configurations as a graph rewriting rule.

Definition 23. *Let N_R be a reconfigurable S^3PR with $N_R = ((N, M_o), \mathcal{R})$, where(N, M_o) is an S^3PR net with $N = (P_C, T, F, W, M_o, K)$ and $\mathcal{R} = \{rr_1, rr_2, rr_3, \ldots, rr_m\}$ is called a set of rewriting rules or dynamic configurations if*

1. *For all $rr \in \mathcal{R}$, $rr = \{L, R, \varphi, {}^\bullet\varphi, \varphi^\bullet\}$;*
2. *$L = (P_{CL}, T_L, F_L, W_L, M_{oL}, K_L)$ is called the left-hand side;*
3. *$R = (P_{CR}, T_R, F_R, W_R, M_{oR}, K_R)$ is called the right-hand side;*
4. *$\varphi \subseteq (P_{CL} \times P_{CR}) \cup (T_L \times T_R)$ is said to be an interface transfer relation of r that relates places of L to places of R and transitions of L to transitions of R, $P_{CL}\,\varphi \subseteq P_{CR}$, $\varphi P_{CR} \subseteq P_{CL}$, $T_L\varphi \subseteq T_R$, and $\varphi T_R \subseteq P_L$;*
5. *${}^\bullet\varphi \subseteq \varphi$ is said to be an input interface transfer relation, expressed as ${}^\bullet\varphi = \{(\{L.p_i\}, \{R.p_i\})\}$ or $\{(\{L.t_i\}, \{R.t_i\})\}$, and L.* or R.* means to input nodes "*" in L or R;*
6. *$\varphi^\bullet \subseteq \varphi$ is named output interface transfer relation, $\varphi^\bullet = \{(\{L.p_j\}, \{R.p_j\})\}$ or $\{(\{L.t_j\}, \{R.t_j\})\}$, and L.* or R.* means to output nodes "*" in L or R;*
7. *for all rr_i, $rr_j \in \mathcal{R}$ $(i \neq j)$, $\xi(L_i) \cap \xi(L_j) \neq \emptyset$, a rewriting must be guaranteed without overlap; moreover, the order of rr_i, rr_j does not impact the result of the rewriting.*

Definition 24. *Let N_R be a reconfigurable S^3PR with N_R = ((N, M_o), $\mathcal{R}$). A new rewriting reconfigurable net N_R is an S^3PR net (N_R, M_R) with N_R = (P_C, T, F, W, M_R, K), and a net (N, M_o) is called the initial state of the rewriting net model.*

Definition 25. *Let N_R be a reconfigurable S^3PR with N_R = ((N, M_o), $\mathcal{R}$). A state graph in N_R is a labeled directed graph whose nodes are the marking of N_R, expressed as:*

1. *Transition firing: If Arcs labeled with t can fire in the net (N_1, M_1), leading to (N_2, M_2): (N_1, M_1) $\xrightarrow{t}$ (N_2, M_2) ⇔ (N_1 = N_2 and $M_1[t_2\rangle M_2$ in N_1).*
2. *Configuration changing: Arcs labeled with r = {L, R, φ, $^\bullet$φ, φ$^\bullet$} from state (N_1, M_1) to state (N_2, M_2) if there is ξ: L → N_1 so that, ∀a ∉ ξ(L) and b ∈ L if*
   - 2.1. *$a \in {}^\bullet\xi(b) \Rightarrow b \in {}^\bullet\varphi$ and $a \in \xi(b)^\bullet \Rightarrow b \in \varphi^\bullet$.*
   - 2.2. *N_1 = (P_{C1}, T_1, F_1, W_1, M_1, K_1) and N_2 = (P_{C2}, T_2, F_2, W_2, M_2, K_2) holds the following: $P_{C2} = P_{C1} - \xi(P_{C1L}) + P_{C1R}$ and $T_2 = T_1 - \xi(T_{1L}) + T_{1R}$. Note that −(+) means deleting(inserting) places or transitions from (to) N_1 and the places name of P_{C1R} and T_{1R} inserted into N_1 must be different to prevent clashes.*

Definition 26. *Let N_R be a reconfigurable S^3PR with N_R = ((N, M_o), $\mathcal{R}$). Let N_1 and N_2 be two states in N_R with N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1) and N_2 = (P_{C2}, T_2, F_2, W_2, M_{2o}, K_2). A net N_1 is the restriction of a net N_2 if $P_{C1} \subseteq P_{C2}$, $T_1 \subseteq T_2$, and $F_1 = F_2 \cap ((P_{C1} \times T_1) \cup (T_1 \times P_{C1}))$ and expressed by $N_1 \subseteq N_2$.*

Definition 27. *Let N_R be a reconfigurable S^3PR with N_R = ((N, M_o), $\mathcal{R}$). Let N_1 and N_2 be two states in N_R with N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1) and N_2 = (P_{C2}, T_2, F_2, W_2, M_{2o}, K_2). The set of weighted arcs (flow relation) F_2 is expressed as:*

$$F_2(a,\ b) = \begin{bmatrix} F_1(a,\ b) & \text{if } a \notin R \wedge b \notin R \\ F_R(a,\ b) & \text{if } a \in R \wedge b \in R \\ \sum\limits_{b_i \in \bullet\varphi b} F_1(a, \xi(y_i)) & \text{if } a \notin R \wedge b \in R \\ \sum\limits_{a_i \in \boldsymbol{\varphi}\bullet a} F_1(\xi(a_i), b) & \text{if } a \in R \wedge b \notin R \end{bmatrix} \quad (2)$$

Definition 28. *Let N_R be a reconfigurable S^3PR with N_R = ((N, M_o), $\mathcal{R}$). Let N_1 and N_2 be two states in N_R with N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1) and N_2 = (P_{C2}, T_2, F_2, W_2, M_{2o}, K_2). The marking of M′(p), p ∈ P_{C2}, is expressed as:*

$$M'(p) = \begin{bmatrix} M(p) & \text{if } p \notin R \\ \sum\limits_{p' \in \varphi p} M(\xi(p')) & \text{if } p \in R \end{bmatrix} \quad (3)$$

Theorem 1. *Let N_R be a reconfigurable S^3PR with N_R = ((N, M_o), $\mathcal{R}$). Let N_1 and N_2 be two states in N_R with N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1) and N_2 = (P_{C2}, T_2, F_2, W_2, M_{2o}, K_2), P_{C1}, $T_1 \neq \emptyset$ and $\mathcal{R}$ = {rr}, rr = {L, R, φ, $^\bullet$φ, φ$^\bullet$}. If L and R are a single place or single transition, then the obtained N_2 by rr is equal to N_1.*

Proof. Straightforward. □

Theorem 2. *Let N_R be a reconfigurable S^3PR with N_R = ((N, M_o), $\mathcal{R}$). Let N_1 and N_2 be two states in N_R with N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1) and N_2 = (P_{C2}, T_2, F_2, W_2, M_{2o}, K_2), P_{C1}, $T_1 \neq \emptyset$ and $\mathcal{R}$ = {rr}, rr = {L, R, φ, $^\bullet$φ, φ$^\bullet$}. If (N_1, M_1) is bounded, L is a single place or single transition and R is an S^3PR net, then the resulting (N_2, M_{2o}) net by rr is bounded.*

Proof. The rewriting of N_2 using *rr* is similar to replacing a place/transition by the S^3PR net. Therefore, the boundedness can be established by checking if the S^3PR net is well constructed and behaved. The resulting net (N_2, M_{2o}) maintains the boundedness because the S^3PR net is well constructed and behaved. □

Corollary 1. *Let N_R be a reconfigurable S^3PR with $N_R = ((N, M_o), \mathcal{R})$. Let N_1 and N_2 be two states in N_R with $N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1)$ and $N_2 = (P_{C2}, T_2, F_2, W_2, M_{2o}, K_2)$, $P_{C1}, T_1 \neq \emptyset$ and $\mathcal{R} = \{rr\}$, $rr = \{L, R, \varphi, {}^\bullet\varphi, \varphi^\bullet\}$. If (N_1, M_{1o}) is bounded, L is an S^3PR Petri net and R is a single place or single transition, then the resulting net (N_2, M_{2o}) by rr is bounded.*

Corollary 2. *An S^3PR net (N_2, M_{2o}) can be a bounded net and a full subnet of (N_1, M_{1o}).*

Theorem 3. *Let N_R be a reconfigurable S^3PR with $N_R = ((N, M_o), \mathcal{R})$. Let N_1 and N_2 be two states in N_R with $N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1)$ and $N_2 = (P_{C2}, T_2, F_2, W_2, M_{2o}, K_2)$, $P_{C1}, T_1 \neq \emptyset$ and $\mathcal{R} = \{rr\}$, $rr = \{L, R, \varphi, {}^\bullet\varphi, \varphi^\bullet\}$. If (N_1, M_{1o}) is bounded, L is an S^3PR net and R is an S^3PR net, then the resulting net (N_2, M_{2o}) by rr is bounded.*

Proof. The rewriting of N_2 using *rr* is similar to replacing an S^3PR net by another S^3PR net. Therefore, the boundedness can be established by checking if the S^3PR net is well constructed and behaved. The resulting net (N_2, M_{2o}) maintains the boundedness because the S^3PR net is well constructed and behaved. □

Based on Definitions 20–28 and Theorems 1–3, the developed reconfiguration procedures for S^3PR net algorithm are constructed as follows:

Algorithm 1: *Reconfiguration procedures for S^3PR net*

Input: *An S^3PR net (N_o, M_o)*
Output: *A reconfigurable S^3PR net (N_R, M_{Ro})*
Initialization: *Generate dynamic configurations $\mathcal{R} = \{rr_1, rr_2, rr_3, \ldots, rr_m\}$ k=0.*
Step 1**: **while** $\mathcal{R} \neq \emptyset$ **do
    $k = k+1$
1.1. *Build $rr_k = \{L_k, R_k, \varphi_k, {}^\bullet\varphi_k, \varphi_k{}^\bullet\}$.*
1.2. *Build $L_k = (P_{CLk}, T_{Lk}, F_{Lk}, W_{Lk}, M_{Lko}, K_{Lk})$.*
1.3. *Build $R_k = (P_{CRk}, T_{Rk}, F_{Rk}, W_{Rk}, M_{Rko}, K_{Rk})$.*
1.4. *Build ${}^\bullet\varphi_k$ and $\varphi_k{}^\bullet$.*
1.5. *Build ξ_k: $N_{k-1} \rightarrow N_k$.*
1.6. *Apply rewriting rule rr_k: $N_k \xrightarrow{rr_k} N_{k-1}$.*
1.7. *Update the flow relation F_k as follows:*

$$F_k(a,b) = \begin{bmatrix} F_{k-1}(a,\ b) & \text{if } a \notin R_k \wedge b \notin R_k \\ F_{(k-1)R}(a,b) & \text{if } a \in R_k \wedge b \in R_k \\ \sum\limits_{b_i \in \bullet\varphi b} F_{k-1}(a, \xi(y_i)) & \text{if } a \notin R_k \wedge b \in R_k \\ \sum\limits_{a_i \in \boldsymbol{\varphi}\bullet a} F_{k-1}(\xi(a_i), b) & \text{if } a \in R_k \wedge b \notin R_k \end{bmatrix}$$

1.8. *Calculate the initial marking of N_k*

$$M_{ko}(p) = \begin{bmatrix} M_{(k-1)o}(p) & \text{if } p \in P_R, P_R \in R_k \\ 0 & \text{if } p \in P_A, P_A \in R_k \end{bmatrix}$$

1.9. *$\mathcal{R} = \mathcal{R} \backslash CR$. /* CR is covered rr_k*/*
    end while
***Step 2**: Output a reconfigurable S^3PR net (N_R, M_{Ro})*
***Step 3:** End*

To illustrate the proposed Algorithm 1, reconsider the initial S^{3}PR net (N_o, M_o) illustrated in Figure 2. Suppose that the first system configuration includes adding new machine. In this scenario, a new machine M2 is assigned to the system (N_o, M_o) to process a part after M, a robot is needed to load/unload a part to/from M2. To model the addition of new machine by using the synthesis procedure of Algorithm 1, we construct a configuration as a rewriting rule $\mathcal{R} = \{rr_1\}$ with $rr_1 = \{L_1, R_1, \varphi_1, {}^\bullet\varphi_1, \varphi_1{}^\bullet\}$, where L_1 and R_1 are illustrated in Figures 3a and 3b, respectively. We have ξ_1: $N_1 \rightarrow N_o$, $\varphi_1 = (\{p_1, p_6, p_7, p_8, p_9\},\{t_4, t_5, t_6\})$, ${}^\bullet\varphi_1 = (\{L_1.t_4\},\{R_1.t_4\})$, and $\varphi_1{}^\bullet = (\{L_1.p_1, L_1.p_6\},\{R_1.p_1\})$. Then the obtained reconfigurable S^{3}PR net (N_1, M_{1_0}) is illustrated in Figure 3c.

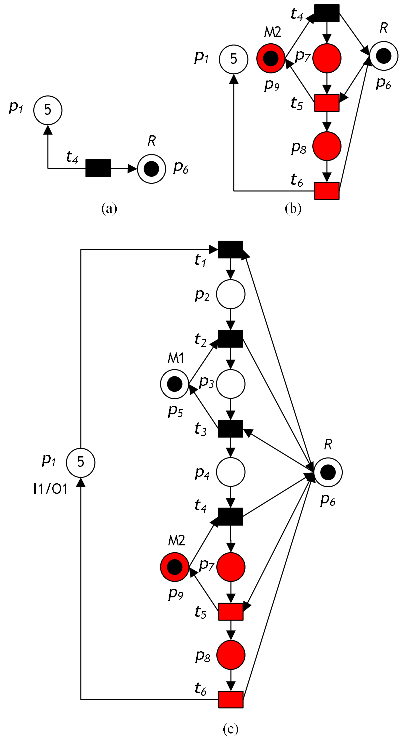

Figure 3. A reconfigured S^{3}PR net by addition of new machine. (**a**) Left hand side net *L*. (**b**) Right hand side net *R*. (**c**) A reconfigurable S^{3}PR net (N_1, M_{1_0}).

The second configuration includes adding a new product. In this scenario, a new product (part B) is assigned to a system, which indicates that a new operation sequence is assigned and the system requires an adjustment to its Petri net model structure. To model the addition of new product by using the synthesis procedure of Algorithm 1, we constructed a configuration as a rewriting rule $\mathcal{R} = \{rr_2\}$ with $rr_2 = \{L_2, R_1, \varphi_2, {}^\bullet\varphi_2, \varphi_2{}^\bullet\}$, where L_2 and R_2 are illustrated in Figures 4a and 4b, respectively. We have ξ_2: $N_2 \rightarrow N_1$, $\varphi_2 = (\{p_5, p_6, p_{10}, p_{11}, p_{12}, p_{13}\},\{t_7, t_8, t_9, t_{10}\})$, ${}^\bullet\varphi_2 = (\{L_2.p_5, L_2.p_6\},\{R_2.t_7\})$, and $\varphi_2{}^\bullet = (\{L_2.p_5, L_2.p_6\},\{R_2.t_{10}\})$. Then the obtained reconfigurable S^{3}PR net (N_2, M_{2_0}) is illustrated in Figure 4c.

The third system configuration involves rework. In this scenario, a part can be inspected after all operations have been completed. The system can proceed on the basis of the original sequence of operation if the configuration is carried out properly. Otherwise, rework is needed. By using Algorithm 1, the production operations of the reworked part can be exactly and easily modeled by considering rework operations as alternative sequences. Reconsider the reconfigurable S^{3}PR net (N_2, M_2) illustrated in Figure 4c. Suppose that an inspection machine M3 is added to a system and that part B is processed in M1. Then, part B is moved to M3 by Robot 1 to check if there are defects in part B. If part B performed properly, then it will leave the system by Robot 1. Otherwise, if part B has defects, rework is needed, and part B is moved to M1 by Robot 1. To model the rework operation by using the synthesis procedure of Algorithm 1, we construct a configuration as a rewriting rule $\mathcal{R} = \{rr_3\}$ with $rr_3 = \{L_3, R_3, \varphi_3, {}^\bullet\varphi_3, \varphi_3{}^\bullet\}$, where L_3 and R_3 are illustrated in Figure 5a,b, respectively. We have

ξ_3: $N_3 \rightarrow N_2$, $\varphi_3 = (\{p_5, p_6, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}\}, \{t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}\})$, ${}^{\bullet}\varphi_3 = (\{L_3.t_7\},\{R_3.t_7\})$, and $\varphi_3{}^{\bullet} = (\{L_3.t_{10}\},\{R_3.t_{14}\})$. Then the obtained reconfigurable S[3]PR net (N_3, M_{3o}) is illustrated in Figure 5c.

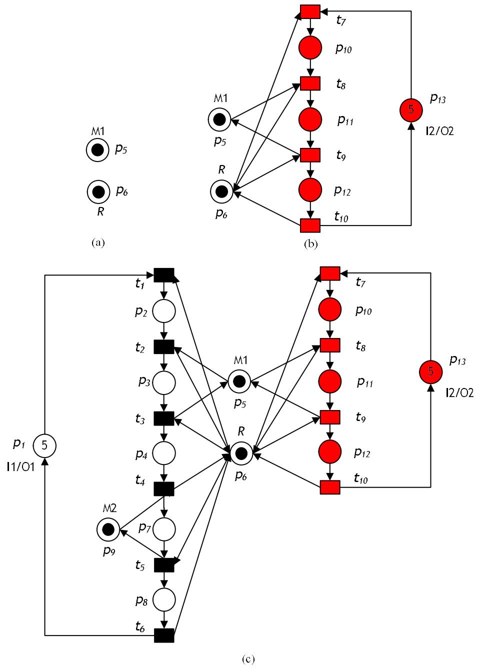

Figure 4. A reconfigured S[3]PR net by addition of new product. (**a**) Left hand side net *L*. (**b**) Right hand side net *R*. (**c**) A reconfigurable S[3]PR net (N_2, M_{2o}).

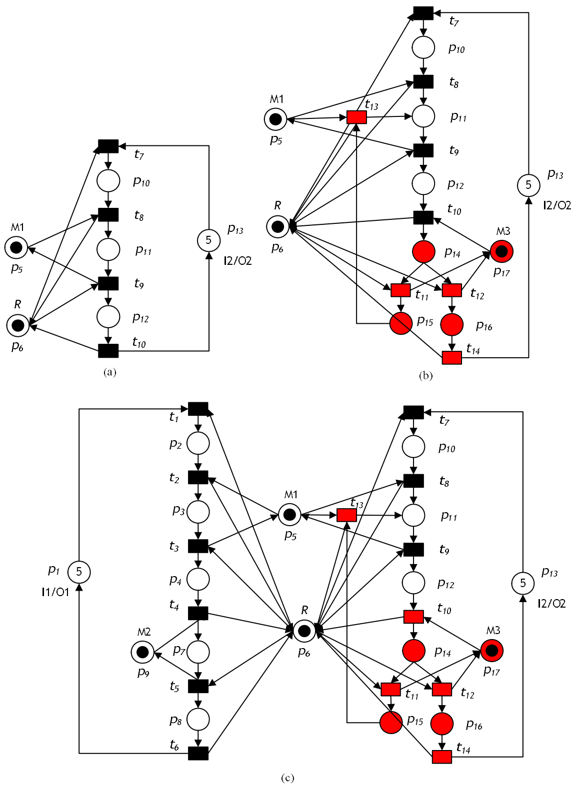

Figure 5. A reconfigured S[3]PR net by rework. (**a**) Left hand side net L. (**b**) Right hand side net R. (**c**) A reconfigurable S[3]PR net (N_3, M_{3o}).

Finally, a configuration includes adding a new robot. In this scenario, a new robot R2 is assigned to the system (N_3, M_{3o}) to load/unload a part A to/from M1 and M2. To model the addition of the new robot by using the synthesis procedure of Algorithm 1, we construct a configuration as a rewriting rule $\mathcal{R} = \{rr_4\}$ with $rr_4 = \{L_4, R_4, \varphi_4, {}^\bullet\varphi_4, \varphi_4{}^\bullet\}$, where L_4 and R_4 are illustrated in Figure 6a,b, respectively. We have ξ_4: $N_4 \rightarrow N_3$, $\varphi_4 = (\{p_1, p_2, p_3, p_4, p_{6\_1}, p_{6\_2}, p_7, p_8, p_{10}, p_{11}, p_{12}, p_{14}, p_{15}, p_{16}\},\{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}\})$, ${}^\bullet\varphi_4 = (\{L_4.t_1, L_4.t_7\},\{R_4.\ t_1, R_4.\ t_7\})$, and $\varphi_4{}^\bullet = (\{L_4.\ t_6, L_4.\ t_{14}\}, \{R_4.t_6, R_4.t_{14}\})$. Then the obtained reconfigurable S^3PR net (N_4, M_{4o}) is illustrated in Figure 6c.

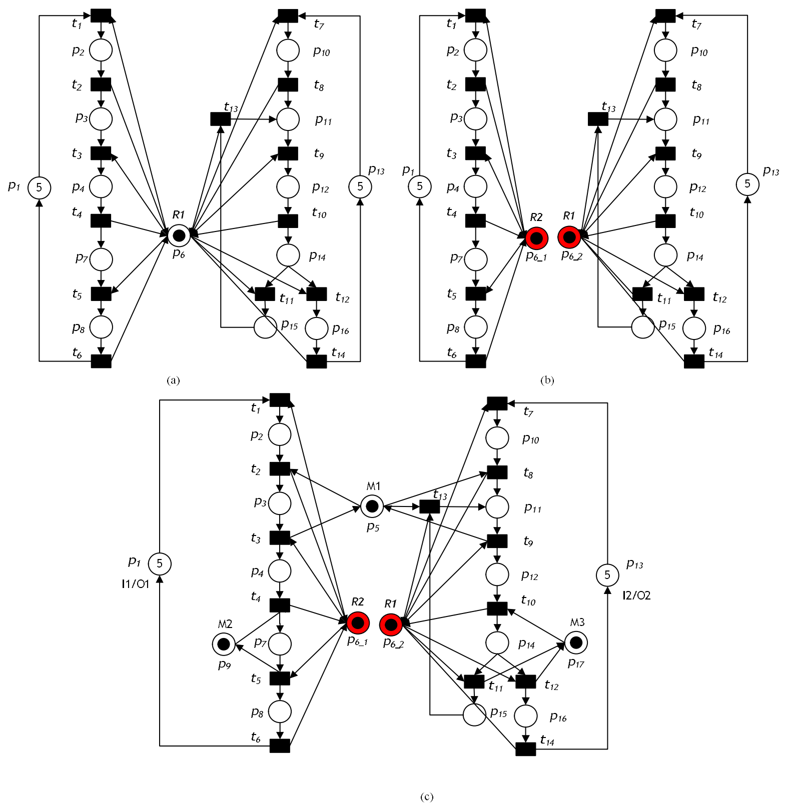

Figure 6. A reconfigured S^3PR net by addition of a new robot. (**a**) Left hand side net L. (**b**) Right hand side net R. (**c**) A reconfigurable S^3PR net (N_4, M_{4o}).

3. Deadlock Prevention Policy for Reconfigurable S^3PR Net Based on Siphons

This section presents definitions on siphons in reconfigurable S^3PR nets. Next, the siphon control method based on place invariants is introduced. Finally, a deadlock prevention algorithm is proposed to solve the deadlock problems in reconfigurable S^3PR nets.

Definition 29. *Let N_R be a reconfigurable S^3PR net with $N_R = ((N, M_o), \mathcal{R})$. Let N_1 be a state of N_R with $N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1)$. A place vector of N_1 is expressed as a column vector I: $P_{C1} \rightarrow \mathbf{Z}$ indexed by P_{C1}, and a transition vector of N_1 is defined as a column vector J: $T_1 \rightarrow \mathbf{Z}$ indexed by T_1, where $\mathbf{Z} = \{\ \ldots, -2, -1, 0, 1, 2, \ldots\ \}$.*

Definition 30. *Let N_R be a reconfigurable S^3PR net with $N_R = ((N, M_o), \mathcal{R})$. Let N_1 be a state of N_R with $N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1)$. A place vector I of N_1 is expressed as a place invariant (PI) if $I^T.\ [N_1] = \mathbf{0}^T$ and $I \neq \mathbf{0}$, and a transition vector of N_1 is defined as a transition invariant (TI) if $[N_1].\ J = \mathbf{0}$ and $J \neq \mathbf{0}$.*

Definition 31. *Let N_R be a reconfigurable S^3PR net with N_R = ((N, M_o), $\mathcal{R}$). Let N_1 be a state of N_R with N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1). A place invariant I of N_1 is expressed as a place semi-flow if each element of I is non-negative. $\|I\|$ = {p |I(p) ≠ 0} is said to be the support of place invariant of I. $\|I\|^+$ = {p|I(p) > 0} is said to be the positive support of place invariant I. $\|I\|^-$ = {p |I(p) < 0} is said to be the negative support of place invariant I. I is a minimal place invariant if $\|I\|$ is not a superset of the support of any other one and its components are mutually prime.*

Definition 32. *Let N_R be a reconfigurable S^3PR net with N_R = ((N, M_o), $\mathcal{R}$). Let N_1 be a state of N_R with N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1). A transition invariant J of N_1 is expressed as a transition semi-flow if each element of J is non-negative. $\|J\|$ = {t |J(t) ≠ 0} is said to be the support of transition invariant of J. $\|J\|^+$ = {t|J(t) > 0} is said to be the positive support of transition invariant J. $\|J\|^-$ = {t |J(t) < 0} is said to be the negative support of transition invariant J. J is a minimal transition invariant, if $\|J\|$ is not a superset of the support of any other one, and its components are mutually prime.*

Definition 33. *Let N_R be a reconfigurable S^3PR net with N_R = ((N, M_o), $\mathcal{R}$). Let N_1 be a state of N_R with N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1). l_i is said to be the coefficients of place invariant I if for all $p_i \in P_{C1}$, $l_i = I(p_i)$.*

Definition 34. *Let N_R be a reconfigurable S^3PR net with N_R = ((N, M_o), $\mathcal{R}$). Let N_1 be a state of N_R with N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1). A non-empty set $S \subseteq P_{C1}$ is called a siphon in N_1 if ${}^\bullet S \subseteq S^\bullet$. $S \subseteq P_{C1}$ is called a trap in N_1 if $S^\bullet \subseteq {}^\bullet S$. $S \subseteq P_{C1}$ is called a minimal siphon (trap) if a siphon (trap) contains no other siphons. A minimal siphon S is called a strict minimal siphon if $S^\bullet \subsetneq {}^\bullet S$. Let Π = {S_1, S_2, S_3, … , S_k} be a set of strict minimal siphons of N_1. We have S= $S_A \cup S_R$, $S_R = S \cap P_R$, and $S_A = S \backslash S_R$, where S_A and S_R are sets of operations and resources places, respectively.*

Definition 35. *Let N_R be a reconfigurable S^3PR net with N_R = ((N, M_o), $\mathcal{R}$). Let N_1 be a state of N_R with N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1). A siphon S in N_1 is called marked at marking M if $\sum_{p\in S} M(p) \geq 1$, and otherwise is called unmarked at marking M.*

Definition 36. *Let N_R be a reconfigurable S^3PR net with N_R = ((N, M_o), $\mathcal{R}$). Let N_1 be a state of N_R with N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1). A siphon S in N_1 is called an emptiable siphon if there exists M ∈ R(N_1, M_{1o}) such that $\sum_{p\in S} M(p) = 0$, and otherwise is called non-emptiable siphon.*

Theorem 4. *Let N_R be a reconfigurable S^3PR net with N_R = ((N, M_o), $\mathcal{R}$). Let N_1 be a state of N_R with N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1) and Π the set of N_1 siphons. The net N_1 is deadlock-free if for all S ∈ Π, for all M ∈ R(N_1, M_{1o}), $\sum_{p\in S} M(p) \geq 1$.*

Proof. Let *S* be a siphon in N_1 and $p \in S$. *p* is marked at marking *M* and satisfies $\sum_{p\in S} M(p) \geq 1$. The net N_1 has at least one transition *t* enabled at any marking reachable from *M* and *S* is never be an unmarked, and it is therefore deadlock-free. □

Theorem 5. *Let N_R be a reconfigurable S^3PR net with N_R = ((N, M_o), $\mathcal{R}$). Let N_1 be a state of N_R with N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1) and Π the set of N_1 siphons. The net (N_1, M_{1o}) is in a deadlock state, i.e., M is a dead marking of N_1. Then, {$p \in P_{C1}$|M(p) = 0} is a siphon S.*

Proof. Since *M* is a dead marking, each *t* has an empty input place *p* at *M*, $\forall p \in {}^\bullet t$, $M(p) < W(p, t)$, and thus $S^\bullet$ includes each transition of N_1. In fact, we have ${}^\bullet S \subseteq S^\bullet$. Therefore, *S* is a siphon. Since the net has at least one transition $t \in T_1$, *S* is not an empty set. □

Corollary 3. *Let N_R be a reconfigurable S^3PR net with N_R = ((N, M_o), $\mathcal{R}$). Let N_1 be a state of N_R with N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1), a deadlocked N_1 net includes at least one unmarked siphon S.*

Corollary 4. *Let N_R be a reconfigurable S^3PR net with $N_R = ((N, M_o), \mathcal{R})$. Let N_1 be a state of N_R with $N_1 = (P_{C1}, T_1, F_1, W_1, M_{1o}, K_1)$, N_1 is a deadlocked net at marking M. Then, N_1 has at least one unmarked siphon S such that for all $p \in S$, there exists $t \in p^\bullet$ such that $W_1(p, t) > M(p)$.*

To develop a deadlock prevention policy for reconfigurable S^3PR net, we reviewed the approach of designing a control place (monitor) for a place invariant developed by Yamalidou et al. [40]. Then we develop a deadlock prevention policy for reconfigurable S^3PR net to achieve an optimal place invariant. Yamalidou et al. propose a computationally efficient method based on place invariants that enforces algebraic constraints on the elements of a marking of a net system by constructing control places. The control purpose is to ensure a siphon to be a marked siphon, i.e., ensure a siphon be non-emptiable at all elements of a marking.

Assume that a reconfigurable S^3PR net with $N_R = ((N, M_o), \mathcal{R})$ and N_k (state of N_R) with $N_k = (P_{Ck}, T_k, F_k, W_k, M_{ko}, K_k)$, $k = 1, 2, \ldots, |\mathcal{R}|$ is a net to be controlled, which includes n places and m transitions. Let $[N_k]$ be the incidence matrix of a plant reconfigurable S^3PR net. The control places can be represented by $[N_c]$ a matrix that shows the connection relationship between control places to transitions of the net N_k. The controlled net with incidence matrix $[N]$ comprises both the original reconfigurable S^3PR net and the monitors, i.e.,

$$[N] = \begin{bmatrix} N_k \\ N_c \end{bmatrix} \tag{4}$$

The control purpose is to impose a set of linear constraints to prevent unwanted markings being reached. The constraints are formulated in a matrix form:

$$\mathcal{L}.M \geq \mathcal{B} \tag{5}$$

where M denotes the marking vector of net N_k, $\mathcal{L}$ is an integer n_c x n matrix (n_c - the number of constraints), and $\mathcal{B}$ is an integer column vector. After the introduction of a non-negative slack variable that corresponds to the initial marking M_{ko} of N_k, constraint (5) can be reformulated as:

$$M_{co} = \mathcal{B} - \mathcal{L}.M_{ko}. \tag{6}$$

where M_{co} represents the initial marking of monitor c.

If $[N_k]$ is the incidence matrix, we have: $M_k = M_{ko} + [N_k].\vec{\delta}$. Therefore, $M_c = \mathcal{B} - \mathcal{L}.(M_{ko} + [N_k].\vec{\delta})$, which also can be reformulated as:

$$M_c = M_{co} + (-\mathcal{L}.[N_k].\vec{\delta}) \tag{7}$$

The place invariant computed by (5) must meet the place invariant equation $I^T[N] = \mathbf{0}^T$. Therefore, the monitor $[N_c]$ can be formulated as:

$$[N_c] = -\mathcal{L}.[N_k] \tag{8}$$

Consequently, M_c may be considered as a marking of some additional monitors, where the supervised reconfigurable S^3PR net has an incidence matrix $[N] = \begin{bmatrix} N_k \\ N_c \end{bmatrix}$, and a marking vector $M = \begin{bmatrix} M_k \\ M_c \end{bmatrix}$.

Theorem 6. *Let N_R be a reconfigurable S^3PR net with $N_R = ((N, M_o), \mathcal{R})$. Let N_k be a state of N_R with $N_k = (P_{Ck}, T_k, F_k, W_k, M_{ko}, K_k)$, incidence matrix $[N_k]$ and initial marking M_{ko} be given. A set of n_c linear constraints $\mathcal{L}.M_k \geq \mathcal{B}$ are to be imposed. If $\mathcal{B} - \mathcal{L}.M_k \geq 0$ then a Petri net controller with incidence matrix $[N_c] = -L.[N_k]$ and initial marking $M_{co} = \mathcal{B} - \mathcal{L}.M_{ko}$ enforces the constraint $\mathcal{L}.M_k \geq \mathcal{B}$ when included in the closed loop system $[N] = \begin{bmatrix} N_k \\ N_c \end{bmatrix}$. In addition, the controller is maximally permissive.*

Proof. See [40,41]. □

Now, we consider the place invariant approach to control the siphon. Let S be an unmarked siphon. The control purpose is to ensure that S is never unmarked through the system evolution (N, M_o) and eliminate markings that break the linear constraint (5) from the reachable markings.

Let $V_S \backslash S \in \Pi$ be the monitor resulting from controlling the siphon S. There are siphons S such that if $\sum_{p\in S} M_o(p) \geq 1$ for the initial marking M_o, then $\sum_{p\in S} M(p) \geq 1$ for all reachable markings M. Therefore, a siphon S does not require control. In order to reduce the supervisor's complexity, these siphons are identified and no monitors are added. Thus, we have two sets of constraints: $\mathcal{L}.M \geq \mathcal{B}$ and $\mathcal{L}_o.M \geq \mathcal{B}_o$ rather than a single set of constraints $\mathcal{L}.M \geq \mathcal{B}$. The deadlock prevention supervision of the original net needs enforcing $\mathcal{L}.M \geq \mathcal{B}$ and selecting an initial marking M_o such that $\mathcal{L}_o.M_o \geq \mathcal{B}_o$ and $\mathcal{L}.M_o \geq \mathcal{B}$. The constraints $\mathcal{L}_o.M \geq \mathcal{B}_o$ are the constraints that all reachable markings satisfy when the initial markings satisfy them. Therefore, there are two cases to control a siphon:

If $V_S{}^{\bullet} \subseteq {}^{\bullet}S$, then S does not require monitor and V_S is not assigned to a net N. Furthermore, $V_S{}^{\bullet} \subseteq {}^{\bullet}S$ if and only if S is a trap. Thus, when S is also a siphon, it is (trap) controlled for all initial markings M_o that satisfy $\sum_{p\in S} M_o(p) \geq 1$. Therefore, a siphon S is assigned to ($\mathcal{L}_o$; $\mathcal{B}_o$).

A. If $V_S{}^{\bullet} \subsetneq {}^{\bullet}S$, then S needs a monitor and V_S is assigned to N. Therefore, the S is assigned to ($\mathcal{L}$; $\mathcal{B}$).

Definition 37. *Let N_R be a reconfigurable S^3PR net with $N_R = ((N, M_o), \mathcal{R})$. Let N_k be a state in N_R with $N_k = (P_{Ck}, T_k, F_k, W_k, M_{ko}, K_k)$. A siphon S in N_k is called controlled if for all $M \in R(N_k, M_{ko})$, $\sum_{p\in S} M(p) \geq 1$ and satisfy $\mathcal{L}.M \geq \mathcal{B}$ and $\mathcal{L}_o.M \geq \mathcal{B}_o$.*

Definition 38. *Let N_R be a reconfigurable S^3PR net with $N_R = ((N, M_o), \mathcal{R})$. Let N_k be a state in N_R with $N_k = (P_{Ck}, T_k, F_k, W_k, M_{ko}, K_k)$. The deadlock controller for (N_k, M_{ok}) is expressed as $(V, M_{Vo}) = (P_V, T_V, F_V, M_{Vo})$, where (1) $P_V = \{V_S \backslash S \in \Pi\}$ is set of monitors. (2) $T_V = \{t \backslash t \in {}^{\bullet}V_S \cup V_S{}^{\bullet}\}$. (3) $F_V \subseteq (P_V \times T_V) \cup (T_V \times P_V)$ is called a flow relation of V. (4) for all $V_S \in P_V$, $M_{Vo}(V_s) = \mathcal{B} - \mathcal{L}.M_{ko}(V_S)$, where $M_{Vo}(V_S)$ is called an initial marking of a monitor. (N_{RC}, M_{RCo}) is said to be a controlled reconfigurable S^3PR net resulting from the integration of (N_k, M_{ko}) and (V, M_{Vo}), expressed as $(N_k, M_{ko}) \parallel (V, M_{Vo})$, where $N_{RC} = (P_{RC}, T_{RC}, F_{RC}, W_{RC}, M_{RCo}, K_{RC})$, $P_{RC} = P_{Ck} \cup P_V$, $T_{RC} = T_k \cup T_V$, F_{RC}: $(P_{RC} \times T_{RC}) \cup (T_{RC} \times P_{RC}) \rightarrow$ **IN** is called flow relations, W_{RC}: $(P_{RC} \times T_{RC}) \cup (T_{RC} \times P_{RC}) \rightarrow$ **IN** is a mapping that assigns a weight to an arc, M_{RCo}: $P_{RC} \rightarrow$ **IN** is the initial marking, and K_{RC}: $P_{RC} \rightarrow$ **IN** is the function of capacity that assigns to each place p the maximal number of tokens $K_{RC}(p)$.*

Based on the concept of place invariant and siphon control, the deadlock prevention algorithm for reconfigurable S^3PR net is developed as follows:

Algorithm 2: *Deadlock prevention algorithm for reconfigurable S^3PR net based on siphon control*

Input: *An S^3PR net (N_o, M_o)*
Output: *A controlled reconfigurable S^3PR net (N_{RC}, M_{RCo}).*
Initialization: *Generate dynamic configurations $\mathcal{R} = \{rr_1, rr_2, rr_3, \ldots, rr_m\}$ k=0, $P_V = \emptyset$, $T_V = \emptyset$, $F_V = \emptyset$, $(N_{RC}, M_{RCo}) = \emptyset$.*
Step 1**: **while** $\mathcal{R} \neq \emptyset$ **do
k=k+1
1.1. *Build (N_k, M_{ko}) by using Algorithm 1.*
1.2. *Compute minimal siphons Π for (N_k, M_{ko}).*
1.3. ***for** each $S \in \Pi$ **do***
if** $V_S{}^\bullet \subsetneq {}^\bullet S$, **then
a. Add S to $(\mathcal{L}; \mathcal{B})$.
b. $[N_{V_S}] = -\mathcal{L}.[N_k]$
c. $M_{Vo}(V_s) = \mathcal{B} - \mathcal{L}.M_{ko}$.
d. $P_V := P_V \cup \{Vs\}$
e. $T_V := T_V \cup \{t \backslash t \in {}^\bullet V_S \cup V_S{}^\bullet\}$.
f. $F_V := F_V \cup ((P_V \times T_V) \cup (T_V \times P_V))$
elseIf** $V_S{}^\bullet \subseteq {}^\bullet S$ and $\sum_{p \in S} M_o(p) \geq 1$, **then
Add S to $(\mathcal{L}_o; \mathcal{B}_o)$.
end if
end for
1.4. *$(N_{RC}, M_{RCo}) := (N_k, M_{ko}) \parallel (V_k, M_{Vko})$*
1.5. *$\mathcal{R} = \mathcal{R} \backslash CR$. /* CR is covered rr_k*/*
end while
Step 2: *Output a controlled reconfigurable S^3PR net (N_{RC}, M_{RCo}).*
Step 3: *End*

To illustrate the proposed Algorithm 2, reconsider the initial S^3PR net (N_o, M_o) illustrated in Figure 2. The initial net has four minimal siphons $S_1 = \{p_1, p_2, p_3, p_4\}$, $S_2 = \{p_3, p_5\}$, $S_3 = \{p_2, p_4, p_6\}$, and $S_4 = \{p_4, p_5, p_6\}$. The N_o incidence matrix is

$$[N_o] = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 \\ -1 & 1 & -1 & 1 \end{bmatrix} \tag{9}$$

while its initial marking is:

$$M_o = \begin{bmatrix} 5 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}^T \tag{10}$$

S_4 creates monitor V_{S1}, therefore one monitor V_{S1} is added, which enforces:

$$M(p_4) + M(p_5) + M(p_6) \geq 1 \tag{11}$$

The following place invariant is generated:

$$M(V_{S1}) = M(p_4) + M(p_5) + M(p_6) - 1 \tag{12}$$

The current matrices $\mathcal{L}$ and $\mathcal{B}$ represent the Equation (12).

$$\mathcal{L} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}, \mathcal{B} = [1] \tag{13}$$

while the others minimal siphons create constraints in $(\mathcal{L}_o; \mathcal{B}_o)$.

$$\mathcal{L}_o = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \mathcal{B}_o = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (14)$$

The controller net incidence matrix is calculated by Equation (8):

$$\left[N_{V_S}\right] = -\mathcal{L}.[N_o] = \begin{bmatrix} -1 & 0 & 1 & 0 \end{bmatrix} \quad (15)$$

The controller's initial place marking is calculated as:

$$M_o(V_{S1}) = M_o(p_4) + M_o(p_5) + M_o(p_6) - 1 = 1$$

The controlled net of (N_o, M_o) is illustrated in Figure 7. The place and arcs of the controller are shown with blue lines.

Now, reconsider the reconfigured S[3]PR net by addition of new machine (N_1, M_{1o}) illustrated in Figure 3c. The reconfigured net has seven minimal siphons $S_1 = \{p_3, p_5\}$, $S_2 = \{p_7, p_9\}$, $S_3 = \{p_2, p_4, p_6, p_8\}$, $S_4 = \{p_4, p_5, p_6, p_8\}$, $S_5 = \{p_2, p_6, p_8, p_9\}$, $S_6 = \{p_5, p_6, p_8, p_9\}$, and $S_7 = \{p_1, p_2, p_3, p_4, p_7, p_8\}$.

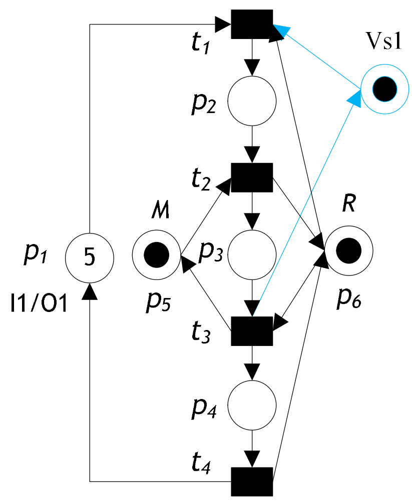

Figure 7. Controlled S[3]PR net by Algorithm 2.

The N_1 incidence matrix is:

$$[N_1] = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ -1 & 1 & -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 & 0 \end{bmatrix} \quad (16)$$

while its initial marking is:

$$M_{1O} = \begin{bmatrix} 5 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}^T \quad (17)$$

S_4, S_5, and S_6 create monitor V_{S1}, V_{S2}, and V_{S3}, respectively. Thus, three monitors are added, V_{S1}, V_{S2}, and V_{S3}, which enforce:

$$M(p_4) + M(p_5) + M(p_6) + M(p_8) \geq 1 \tag{18}$$

$$M(p_2) + M(p_6) + M(p_8) + M(p_9) \geq 1 \tag{19}$$

$$M(p_5) + M(p_6) + M(p_8) + M(p_9) \geq 1 \tag{20}$$

The following place invariants are accordingly generated:

$$M(V_{S1}) = M(p_4) + M(p_5) + M(p_6) + M(p_8) - 1 \tag{21}$$

$$M(V_{S2}) = M(p_2) + M(p_6) + M(p_8) + M(p_9) - 1 \tag{22}$$

$$M(V_{S3}) = M(p_5) + M(p_6) + M(p_8) + M(p_9) - 1 \tag{23}$$

The current matrices $\mathcal{L}$ and $\mathcal{B}$ represent the Equations (18)–(20).

$$\mathcal{L} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}, \mathcal{B} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{24}$$

while the other minimal siphons create constraints in $(\mathcal{L}_o; \mathcal{B}_o)$.

$$\mathcal{L}_o = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}, \mathcal{B}_o = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{25}$$

The controller's net incidence matrix is calculated by Equation (12);

$$\left[N_{V_S}\right] = -\mathcal{L}.[N_1] = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{26}$$

The initial marking controllers are calculated as:

$$M_o(V_{S1}) = M_o(p_4) + M_o(p_5) + M_o(p_6) + M_o(p_8) - 1 = 1$$

$$M_o(V_{S2}) = M_o(p_2) + M_o(p_6) + M_o(p_8) + M_o(p_9) - 1 = 1$$

$$M_o(V_{S3}) = M_o(p_5) + M_o(p_6) + M_o(p_8) + M_o(p_9) - 1 = 2$$

The controlled reconfigurable net of (N_1, M_{1o}) is illustrated in Figure 8. The place and arcs of the controllers are shown with blue lines.

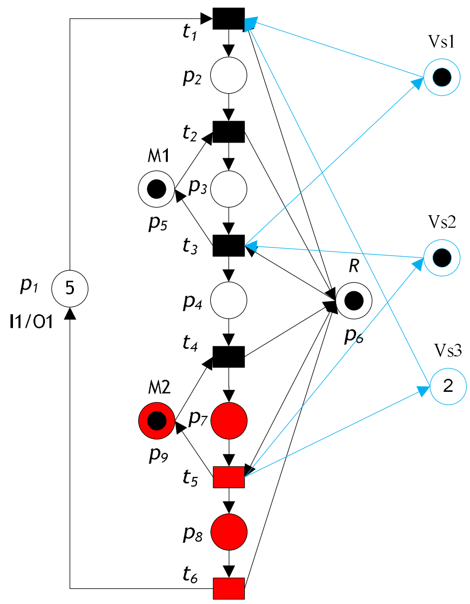

Figure 8. Controlled reconfigurable S[3]PR net by addition of new machine.

Then, reconsider the reconfigured S[3]PR net by addition of new product (N_2, M_{2o}) illustrated in Figure 4c. The reconfigured net has 11 minimal siphons $S_1 = \{p_7, p_9\}$, $S_2 = \{p_3, p_5, p_{11}\}$, $S_3 = \{p_{10}, p_{11}, p_{12}, p_{13}\}$, $S_4 = \{p_4, p_5, p_6, p_8, p_{12}\}$, $S_5 = \{p_5, p_6, p_8, p_9, p_{12}\}$, $S_6 = \{p_1, p_2, p_3, p_4, p_7, p_8\}$, $S_7 = \{p_2, p_4, p_6, p_8, p_{10}, p_{12}\}$, $S_8 = \{p_2, p_6, p_8, p_9, p_{10}, p_{12}\}$, $S_9 = \{p_4, p_5, p_6, p_8, p_{12}\}$, $S_{10} = \{p_5, p_6, p_8, p_9, p_{12}\}$, and $S_{11} = \{p_2, p_6, p_8, p_9, p_{10}, p_{12}\}$. The N_2 incidence matrix is

$$[N_2] = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

while its initial marking is:

$$M_{2O} = \begin{bmatrix} 5 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 5 \end{bmatrix}^{T} \quad (28)$$

S_4, S_5, and S_8 create monitors V_{S1}, V_{S2}, and V_{S3}, respectively. Thus, three monitors are added, V_{S1}, V_{S2}, and V_{S3}, which enforce:

$$M(p_4) + M(p_5) + M(p_6) + M(p_8) + M(p_{12}) \geq 1 \quad (29)$$

$$M(p_5) + M(p_6) + M(p_8) + M(p_9) + M(p_{12}) \geq 1 \quad (30)$$

$$M(p_2) + M(p_6) + M(p_8) + M(p_9) + M(p_{10}) + M(p_{12}) \geq 1 \quad (31)$$

The following place invariants are accordingly generated:

$$M(V_{S1}) = M(p_4) + M(p_5) + M(p_6) + M(p_8) + M(p_{12}) - 1 \quad (32)$$

$$M(V_{S2}) = M(p_5) + M(p_6) + M(p_8) + M(p_9) + M(p_{12}) - 1 \quad (33)$$

$$M(V_{S3}) = M(p_2) + M(p_6) + M(p_8) + M(p_9) + M(p_{10}) + M(p_{12}) - 1 \quad (34)$$

The current matrices $\mathcal{L}$ and $\mathcal{B}$ represent the Equations (29)–(31).

$$\mathcal{L} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}, \mathcal{B} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (35)$$

while the other minimal siphons create constraints in ($\mathcal{L}_o$; $\mathcal{B}_o$).

$$\mathcal{L}_o = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}, \mathcal{B}_o = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (36)$$

The controller's net incidence matrix is calculated by Equation (12);

$$\left[N_{V_S}\right] = -\mathcal{L}.[N_{2o}] = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (37)$$

The initial marking controllers are calculated as:

$$M_o(V_{S1}) = M_o(p_4) + M_o(p_5) + M_o(p_6) + M_o(p_8) + M_o(p_{12}) - 1 = 1$$

$$M_o(V_{S2}) = M_o(p_5) + M_o(p_6) + M_o(p_8) + M_o(p_9) + M_o(p_{12}) - 1 = 2$$

$$M_o(V_{S3}) = M_o(p_2) + M_o(p_6) + M_o(p_8) + M_o(p_9) + M_o(p_{10}) + M_o(p_{12}) - 1 = 1$$

The controlled reconfigurable net of (N_2, M_{2o}) is illustrated in Figure 9. The place and arcs of the controllers are shown with blue lines.

Then, reconsider the reconfigured S^3PR net by rework (N_3, M_{3o}) illustrated in Figure 5c. The reconfigured net has 13 minimal siphons $S_1 = \{p_4, p_5, p_6, p_8, p_{12}, p_{16}\}$, $S_2 = \{p_5, p_6, p_8, p_9, p_{12}, p_{16}\}$, $S_3 = \{p_4, p_5, p_6, p_8, p_{16}, p_{17}\}$, $S_4 = \{p_5, p_6, p_8, p_9, p_{16}, p_{17}\}$, $S_5 = \{p_2, p_6, p_8, p_9, p_{10}, p_{12}, p_{15}, p_{16}\}$, $S_6 = \{p_2, p_4, p_6, p_8\ p_{10}, p_{15}, p_{16}, p_{17}\}$, $S_7 = \{p_2, p_6, p_8, p_9, p_{10}, p_{15}, p_{16}, p_{17}\}$, $S_8 = \{p_2, p_4, p_6, p_8, p_{10}, p_{12}, p_{15}, p_{16}\}$, $S_9 = \{p_7, p_9\}$, $S_{10} = \{p_1, p_2, p_3, p_4, p_7, p_8\}$, $S_{11} = \{p_3, p_5, p_{11}\}$, $S_{12} = \{p_{14}, p_{17}\}$, and $S_{13} = \{p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}\}$. Siphons S_1–S_7, create monitors V_{S1}- V_{S7}, respectively. Thus, seven monitors are added, V_{S1}- V_{S7}, which enforce:

$$M(p_4) + M(p_5) + M(p_6) + M(p_8) + M(p_{12}) + M(p_{16}) \geq 1 \quad (38)$$

$$M(p_5) + M(p_6) + M(p_8) + M(p_9) + M(p_{12}) + M(p_{16}) \geq 1 \quad (39)$$

$$M(p_4) + M(p_5) + M(p_6) + M(p_8) + M(p_{16}) + M(p_{17}) \geq 1 \quad (40)$$

$$M(p_5) + M(p_6) + M(p_8) + M(p_9) + M(p_{16}) + M(p_{17}) \geq 1 \quad (41)$$

$$M(p_2) + M(p_6) + M(p_8) + M(p_9) + M(p_{10}) + M(p_{12}) + M(p_{15}) + M(p_{16}) \geq 1 \quad (42)$$

$$M(p_2) + M(p_4) + M(p_6) + M(p_8) + M(p_{10}) + M(p_{15}) + M(p_{16}) + M(p_{17}) \geq 1 \quad (43)$$

$$M(p_2) + M(p_6) + M(p_8) + M(p_9) + M(p_{10}) + M(p_{15}) + M(p_{16}) + M(p_{17}) \geq 1 \quad (44)$$

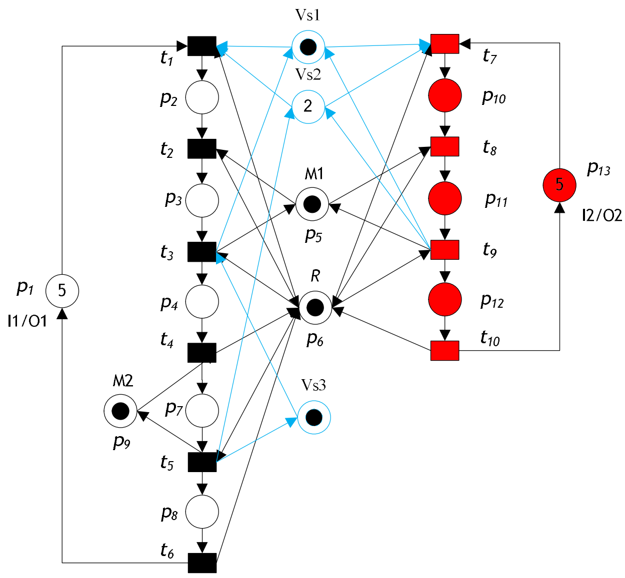

Figure 9. Controlled reconfigurable S^3PR net by addition of new product.

The following place invariants are accordingly generated:

$$M(V_{S1}) = M(p_4) + M(p_5) + M(p_6) + M(p_8) + M(p_{12}) + M(p_{16}) - 1 \quad (45)$$

$$M(V_{S2}) = M(p_5) + M(p_6) + M(p_8) + M(p_9) + M(p_{12}) + M(p_{16}) - 1 \quad (46)$$

$$M(V_{S3}) = M(p_4) + M(p_5) + M(p_6) + M(p_8) + M(p_{16}) + M(p_{17}) - 1 \quad (47)$$

$$M(V_{S4}) = M(p_5) + M(p_6) + M(p_8) + M(p_9) + M(p_{16}) + M(p_{17}) - 1 \quad (48)$$

$$M(V_{S5}) = M(p_2) + M(p_6) + M(p_8) + M(p_9) + M(p_{10}) + M(p_{12}) + M(p_{15}) + M(p_{16}) - 1 \quad (49)$$

$$M(V_{S6}) = M(p_2) + M(p_4) + M(p_6) + M(p_8) + M(p_{10}) + M(p_{15}) + M(p_{16}) + M(p_{17}) - 1 \quad (50)$$

$$M(V_{S7}) = M(p_2) + M(p_6) + M(p_8) + M(p_9) + M(p_{10}) + M(p_{15}) + M(p_{16}) + M(p_{17}) - 1 \quad (51)$$

The current matrices $\mathcal{L}$ and $\mathcal{B}$ represent the Equations (38)–(44).

$$\mathcal{L} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}, \mathcal{B} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (52)$$

The controller's net incidence matrix is calculated by Equation (12);

$$[N_{V_S}] = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \quad (53)$$

The initial marking controllers are calculated as:

$M_o(V_{S1}) = 1$, $M_o(V_{S2}) = 2$, $M_o(V_{S3}) = 2$, $M_o(V_{S4}) = 3$, $M_o(V_{S5}) = 1$, $M_o(V_{S6}) = 1$, and $M_o(V_{S7}) = 2$.

The controlled reconfigurable net of (N_3, M_{3o}) is illustrated in Figure 10. The place and arcs of the controller are shown with blue lines.

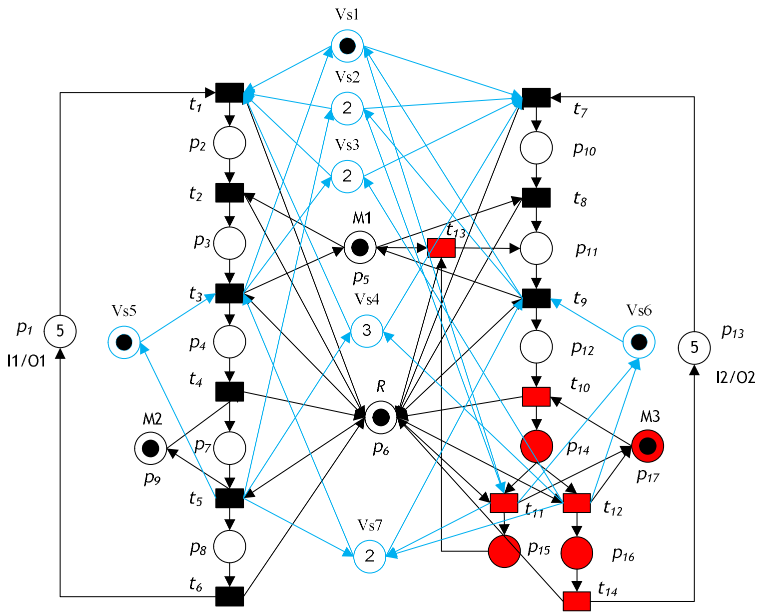

Figure 10. Controlled reconfigurable S^3PR net by rework.

Finally, reconsider the reconfigured S^3PR net by addition of a new robot (N_4, M_{4o}) illustrated in Figure 6c. The reconfigured net has 17 minimal siphons, ten of which S_1–S_{10} that create monitors V_{S1}-V_{S10}, respectively, which are siphons $S_1 = \{p_2, p_6, p_{6\text{-}1}, p_8, p_9\}$, $S_2 = \{p_4, p_5, p_{6\text{-}1}, p_8, p_{11}\}$, $S_3 = \{p_5, p_{6\text{-}1}, p_8, p_9, p_{11}\}$, $S_4 = \{p_4, p_5, p_{6\text{-}1}, p_{6\text{-}2}, p_8, p_{12}, p_{16}\}$, $S_5 = \{p_4, p_5, p_{6\text{-}1}, p_{6\text{-}2}, p_8, p_{16}, p_{17}\}$, $S_6 = \{p_5, p_{6\text{-}1}, p_{6\text{-}2}, p_8\ p_9, p_{12}, p_{16}\}$, $S_7 = \{p_5, p_{6\text{-}1}, p_{6\text{-}2}, p_8, p_9, p_{16}, p_{17}\}$, $S_8 = \{p_3, p_5, p_{6\text{-}2}, p_{12}, p_{16}\}$, $S_9 = \{p_3, p_5, p_{6\text{-}2}, p_{16}, p_{17},\}$, and $S_{10} = \{p_{6\text{-}2}, p_{10}, p_{15}, p_{16}, p_{17}\}$. The current matrices $\mathcal{L}$ and $\mathcal{B}$ are expressed as:

$$\mathcal{L} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}, \mathcal{B} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (54)$$

The controller's net incidence matrix is calculated by Equation (12);

$$[N_{V_S}] = \begin{bmatrix} 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \quad (55)$$

The initial marking controllers are calculated as $M_o(V_{S1}) = 1$, $M_o(V_{S2}) = 1$, $M_o(V_{S3}) = 2$, $M_o(V_{S4}) = 2$, $M_o(V_{S5}) = 3$, $M_o(V_{S6}) = 3$, $M_o(V_{S7}) = 4$, $M_o(V_{S8}) = 1$, $M_o(V_{S9}) = 2$, and $M_o(V_{S10}) = 1$.

The controlled reconfigurable net of (N_4, M_{4o}) is illustrated in Figure 11. The place and arcs of the controller are shown with blue lines.

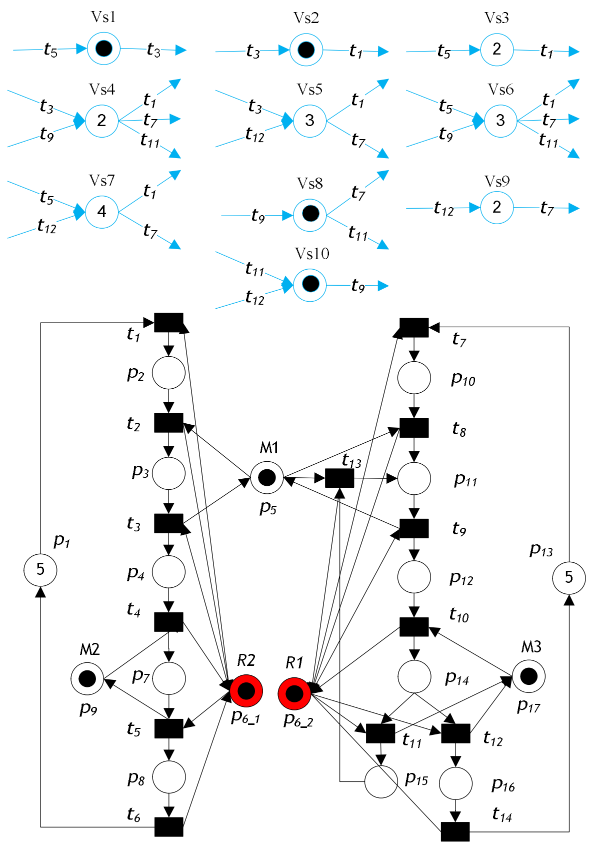

Figure 11. Controlled reconfigurable S^3PR net by rework.

4. Behavioral and Quantitative Analysis of Reconfigurable S^3PR Net

4.1. Liveness

Liveness is one of the most important issues in reconfigurable manufacturing systems with dynamic changes. Conversely, in these systems, deadlock is usually unwanted. When a system is not live, tasks could never be performed because of local or global deadlocks. Liveness of a transition means that, irrespective of the current state of the net, it can always eventually fire.

Theorem 7. *The controlled reconfigurable S^3PR net (N_{RC}, M_{RCo}) with N_{RC} = (P_{RC}, T_{RC}, F_{RC}, W_{RC}, M_{RCo}, K_{RC}) is live.*

Proof. All transitions T_{RC} in (N_{RC}, M_{RCo}) must be proven to be live. There is no unmarked siphon, $p \in S$. p is marked at marking M and satisfies $\sum_{p \in S} M(p) \geq 1$, since all $t \in T_{RC}$ are live. For all $t \in T_{RC}$, if for all $p \in {}^{\bullet}t$, $M_{RCo}(p) > 0$, then t can fire in any case. Therefore, the controlled reconfigurable S^3PR net (N_{RC}, M_{RCo}) is live. □

To demonstrate the liveness of a reconfigurable S^3PR net, consider the model illustrated in Figure 9. Its reachability graph with all model markings is illustrated in Figure 12 and it is apparent that all transitions are live, which means that the system is live.

4.2. Boundedness

The boundedness is associated with a place, indicating that the number of tokens in a place never exceeds a certain number. This means that there is no overflow in a place.

Theorem 8. *Let a reconfigurable S^3PR net (N_{RC}, M_{RCo}) with N_{RC} = (P_{RC}, T_{RC}, F_{RC}, W_{RC}, M_{RCo}, K_{RC}) be a controlled net. Then (N_{RC}, M_{RCo}) is bounded.*

Proof. Theorem 7 proves that the net (N_{RC}, M_{RCo}) is live. Therefore, the boundedness can be established by checking if the net (N_{RC}, M_{RCo}) is well constructed, behaved, and controlled. The resulting net (N_{RC}, M_{RCo}) maintains the boundedness as the net is well constructed, behaved and has a finite reachability set. □

To demonstrate the boundedness of a controlled reconfigurable S^3PR net, consider the net illustrated in Figure 9. Its reachability graph is illustrated in Figure 12. It is obvious that markings reachable from initial marking are five-bounded, which indicates that the system is bounded.

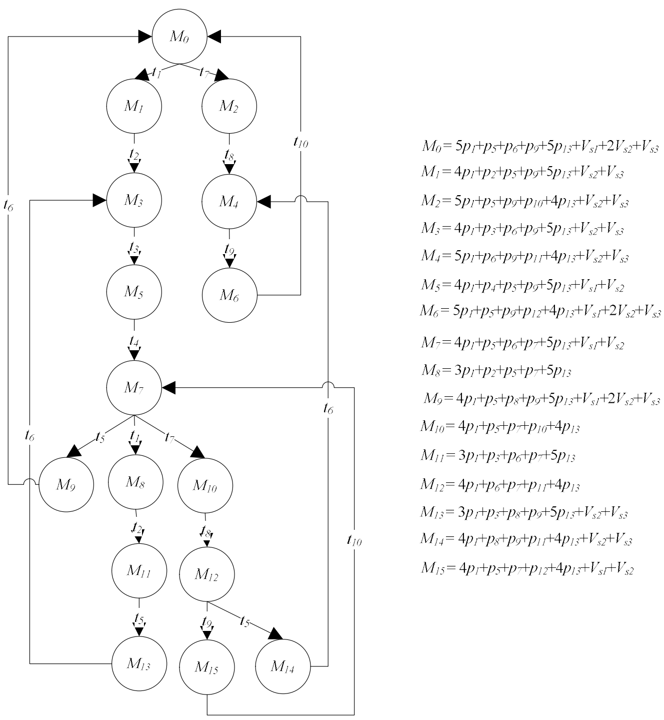

Figure 12. Reachable markings of a controlled reconfigurable S^3PR net, as illustrated in Figure 8.

4.3. Reversibility

Reversibility means that a system can always return to its initial marking. A controlled reconfigurable S^3PR Petri net model (N_{RC}, M_{RCo}) is reversible if for each marking $M \in R(N_{RC}, M_{RCo})$, initial marking M_{RCo} is reachable from M.

Theorem 9. *Let a reconfigurable S^3PR net (N_{RC}, M_{RCo}) with $N_{RC} = (P_{RC}, T_{RC}, F_{RC}, W_{RC}, M_{RCo}, K_{RC})$ be a live and controlled net. N_{RC} is reversible if for each marking $M \in R(N_{RC}, M_{RCo})$, initial marking M_{RCo} is reachable from M, M and M_{RCo} satisfying all place invariants and M marks each trap of N_{RC}.*

Proof. Suppose that M is reachable. Then there exists a finite transition sequence $\delta = t_1\ t_2\ t_3 \ldots\ t_n$ that can be fired, and markings $M_1, M_2, M_3, \ldots$, *and* M_{n-1} are such that $M_{RCo}[t_1\rangle M_1[t_2\rangle M_2[t_3\rangle M_3 \ldots M_{n-1}[t_n\rangle M$, expressed as $M_{RCo}[\delta\rangle M$, agrees with the state equation $M = M_{RCo} + [N_{RC}]\ \vec{\delta}$. In addition, M and M_{RCo} satisfy all place invariants, $I^T.M = I^T.M_{RCo}$. Therefore, we can say that M_{RCo} is the home marking of the net (N_{RC}, M_{RCo}), M is reachable from M_{RCo}, and we get $M_{RCo} \xrightarrow{\vec{\delta}} M$. Thus, the reconfigurable S^3PR net (N_{RC}, M_{RCo}) is reversible. □

To demonstrate the reversibility of a controlled reconfigurable S^3PR net, consider the model illustrated in Figure 8. Its reachability graph is illustrated in Figure 13. In the net shown in Figure 8, there are seven minimal place invariants: $I_1 = p_3 + p_5$, $I_2 = p_2 + p_3 + p_{10}$, $I_3 = p_7 + p_9$, $I_4 = p_4 + p_7 + p_{11}$, $I_5 = p_2 + p_3 + p_4 + p_7 + p_{12}$, $I_6 = p_2 + p_4 + p_6 + p_8$, $I_7 = p_1 + p_2 + p_3 + p_4 + p_7 + p_8$, since $\forall$i $\in$ {1,2,3,4,5,6,7}, $I_i^T.\ [N_{RC}] = 0^T$. $M_6 \in R(N_{RC}, M_{RCo})$, $I_1^T.M_6 = I_1^T.M_{RCo} = M_6(p_3) + M_6(p_5) = M_{RCo}(p_3) + M_{RCo}(p_5) = 1$. The net has a unique T-invariant $J = t_1 + t_2 + t_3 + t_4 + t_5 + t_6$ and the transition sequence $\delta = t_1t_2t_3t_4t_5t_6$ is firable. As a result, $M_{RCo}[t_1\rangle M_1[t_2\rangle M_2[t_3\rangle M_3[t_4\rangle M4[t_5\rangle M6[t_6\rangle M_{RCo}$. Therefore, the reconfigurable S^3PR net (N_{RC}, M_{RCo}) is live, bounded, and reversible.

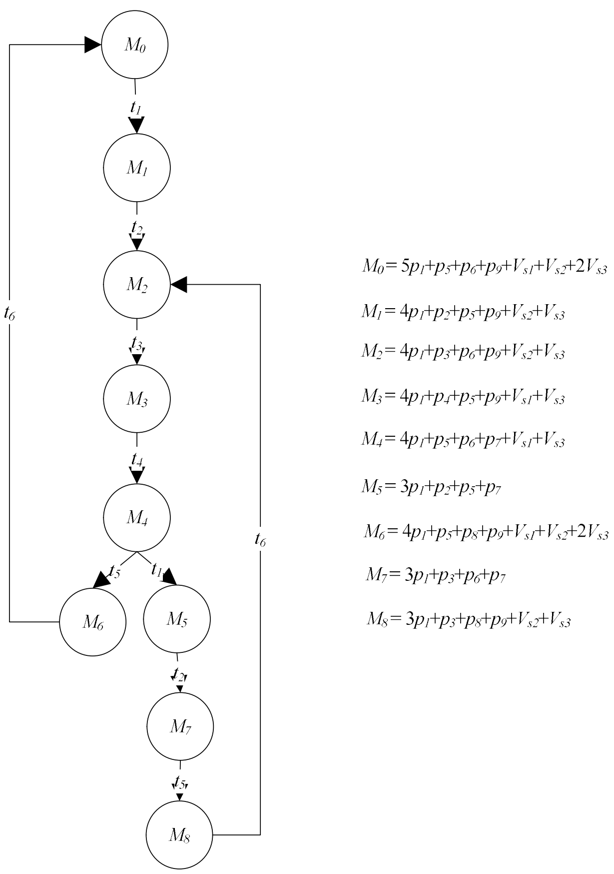

Figure 13. Reachable markings of a controlled reconfigurable S^3PR net, as illustrated in Figure 7.

4.4. Computational Complexity

Algorithm 1 is used to design a reconfigurable S^3PR net with $N_R = ((N, M_o), \mathcal{R})$. In addition, Algorithm 2 computes the control places to a reconfigurable S^3PR net with $N_R = ((N, M_o), \mathcal{R})$.

Theorem 10. *Given a reconfigurable S^3PR net with $N_R = ((N, M_o), \mathcal{R})$, where N_R with $N_k = (P_{Ck}, T_k, F_k, W_k, M_{ko}, K_k)$, the time complexity of Algorithm 1 is polynomial.*

Proof. Let N_k be states in N_R with $N_k = (P_{Ck}, T_k, F_k, W_k, M_{ko}, K_k)$, $\mathcal{R} = \{rr_1, rr_2, rr_3, \ldots, rr_k\}$, $rr_k = \{L_k, R_k, \varphi_k, {}^\bullet\varphi_k, \varphi_k{}^\bullet\}$, and net (N_R, M_{Ro}) be the obtained reconfigurable S^3PR net. Let x be the cardinality of $\mathcal{R}$, i.e., $|\mathcal{R}| = x$. The "While" loop is executed x times to design state N_k in a reconfigurable S^3PR net (N_R, M_{Ro}). Therefore, in the worst case, the computational complexity of algorithm 1 is $O(x)$. Thus, the computational complexity of the Algorithm 1 has polynomial time complexity. □

Theorem 11. *Given a reconfigurable S^3PR net with $N_R = ((N, M_o), \mathcal{R})$, where N_R with $N_k = (P_{Ck}, T_k, F_k, W_k, M_{ko}, K_k)$, the time complexity of Algorithm 2 is polynomial.*

Proof. Algorithm 2 is used to design a control place V_S to each minimal siphon S, $V_S{}^\bullet \subsetneq {}^\bullet S$ in each state N_k in a reconfigurable S^3PR net (N_R, M_{Ro}) to achieve the liveness of net (N_R, M_{Ro}). Obviously, each V_S is associated with the minimal siphon S in net (N_k, M_{ko}). Let x be the cardinality of $\mathcal{R}$, i.e., $|\mathcal{R}| = x$. Let y be the number of minimal siphons S (denoted as S') that requires V_S i.e., $|S'| = y$. The "While" loop is executed x times to design state N_k in reconfigurable S^3PR net (N_R, M_{Ro}). The "FOR loop" loop is executed y times to design V_S for the S' in (N_k, M_{ko}). Therefore, the computational complexity of Algorithm 2 is $O(xy)$. Thus, the computational complexity of the Algorithm 2 has polynomial time complexity. □

4.5. GPENSIM Code and Validation

We coded the developed approach using the GPenSIM tool [6,42] to verify and validate it and compared the developed code with the studies by Ezpeleta et al. [43], Li and Zhou [44], and Kaid et al. [6]. There were three files generated: (1) the Petri net definition file (PDF) that represents the static model by stating the sets of places, transitions, and arcs, (2) the common processor file (COMMON_PRE file) that represents the conditions for activation of the enabling fire transitions, and (3) the main simulation file (MSF) that calculates the results of the simulation. The developed approach was implemented on MATLAB R2015a. A PC with Windows 10, 64-bit and Intel(R) Core (TM) i7-4702MQ CPU @ 2.20 GHz, 16 GB RAM.

Simulation leads to a better time performance in the designed model including total throughput time (total time in system), total throughput, and utilization of the robots and machines. Consider the model illustrated in Figure 8. The simulation was undertaken for 480 min. The results summarized in Table 1 were obtained after simulation in MATLAB. Table 1 shows the results for the time performance criteria mentioned above. All methods achieve approximately the same values for the utilization of resources as illustrated in Figure 14. In addition, the proposed method, as illustrated in Figure 14, can achieve approximately the same values with other techniques for throughput. In term of throughput time of Part A, the proposed method can achieve approximately the same values with other techniques as illustrated in Figure 14. Therefore, the proposed method is valid, sufficiently accurate results can be obtained and other cases can be applied.

Table 1. Time performance comparison with the existing methods.

Performance	Ezpeleta et al. [43]	Li and Zhou [44]	Kaid et al. [6]	The Proposed Method
M1 utilization (%)	29.05	29.05	29.60	29.05
M2 utilization (%)	29.61	29.61	30.50	29.61
R1 utilization (%)	48.04	48.04	47.56	48.04
Throughput (parts)	34	34	34	34
Throughput time (min/part)	14.12	14.12	14.12	14.12

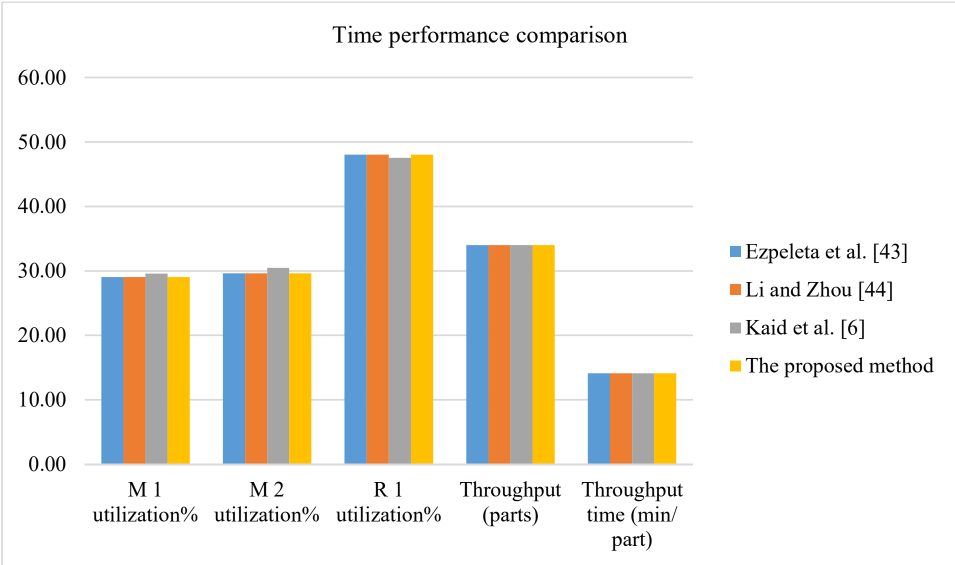

Figure 14. Comparison of the proposed method with the existing methods.

5. Numerical Example

In this section, an example is used to present the application of the proposed approach. Consider an AMS example illustrated in Figure 15a. Its Petri net model is given in [6,7,15,22,45,46]. The system consists of four machines M1–M4 for processing parts; two robots R1 and R2 for loading and unloading parts. Each machine (robot) can process (hold) one part at a time. There are two input buffers I1 and I2 and two output buffers O1 and O2. Two raw part types, A and B, are considered to be processed in the system. Figure 15b shows the operation sequences of the two raw part types. The S^3PR net of this AMS example is illustrated in Figure 16. It comprises 19 places and 14 transitions. The places can be defined as the following set partitions: $P_A = \{p_2, p_3, \ldots, p_{12}\}$, $P_R = \{p_{13}, p_{14}, \ldots, p_{18}\}$, and $P^0 = \{p_1, p_{19}\}$. The S^3PR net contains 282 reachable markings.

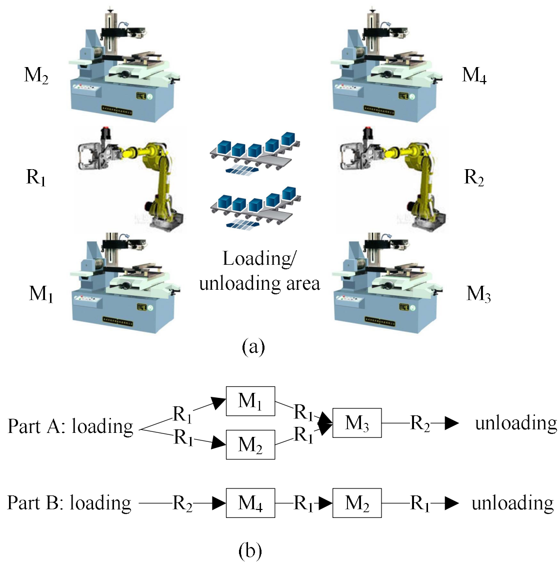

Figure 15. (**a**) An AMS example and (**b**) production sequence.

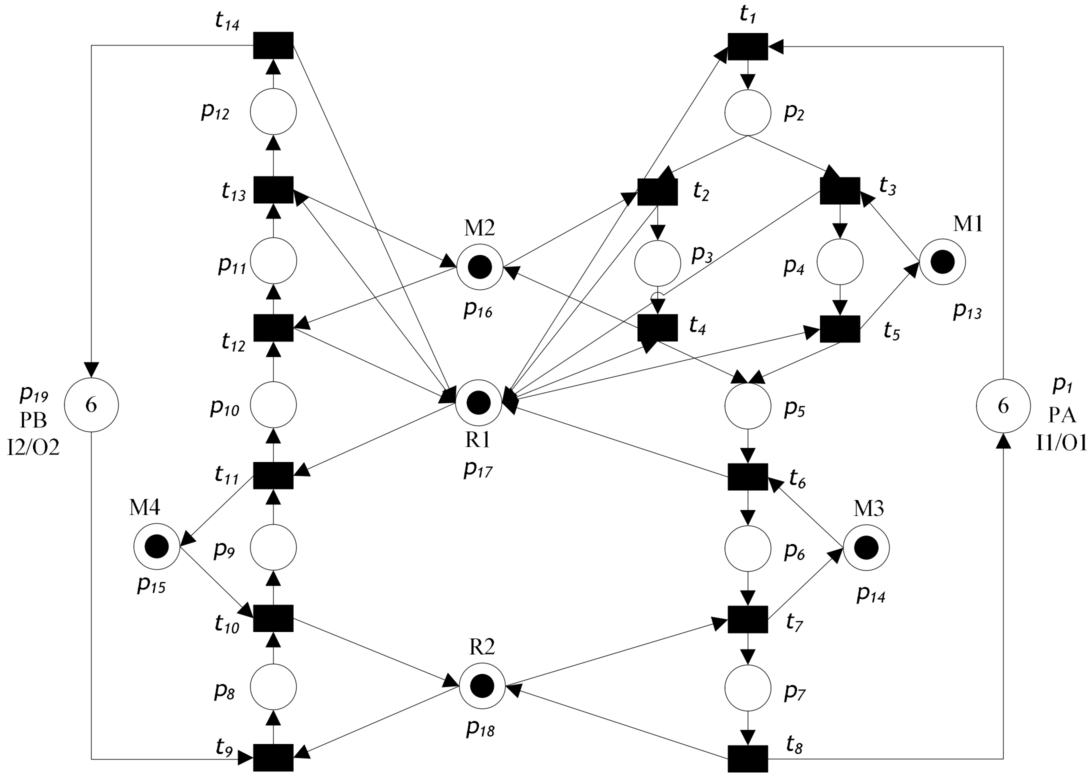

Figure 16. S^3PR net (N_o, M_o) of the AMS illustrated in Figure 13a.

Suppose that the first configuration of the system involves removing old machine. In this case, an old machine M1 is removed from the system (N_o, M_o). To model the removed machine by using the synthesis procedure of Algorithm 1, we construct a configuration as a rewriting rule $\mathcal{R}= \{rr_1\}$ with $rr_1 = \{L_1, R_1, \varphi_1, {}^\bullet\varphi_1, \varphi_1{}^\bullet\}$, where L_1 and R_1 are illustrated in Figures 17a and 17b, respectively. In addition, we have ξ_1: $N_1 \rightarrow N_o$, $\varphi_1= (\{p_2, p_3, p_4, p_5, p_{13}, p_{16}, p_{17}\},\{t_1, t_2, t_3, t_4, t_5, t_6\})$, ${}^\bullet\varphi_1= (\{L_1.t_1\},\{R_1.t_1\})$, and $\varphi_1{}^\bullet = (\{L_1.t_6\},\{R_1.t_6\})$. The second configuration includes adding new product. If a new product (part C) is assigned to a system, which indicates that a new operation sequence is assigned and the system requires an adjustment to its Petri net model structure. To model the addition of new product by using the synthesis procedure of Algorithm 1, we construct a configuration as a rewriting rule $\mathcal{R} = \{rr_2\}$ with $rr_2 = \{L_2, R_1, \varphi_2, {}^\bullet\varphi_2, \varphi_2{}^\bullet\}$, where L_2 and R_2 are illustrated in Figures 18a and 18b, respectively. Moreover, we have ξ_2: $N_2 \rightarrow N_1$, $\varphi_2= (\{p_{15}, p_{17}, p_{20}, p_{21}, p_{22}, p_{23}\},\{t_{15}, t_{16}, t_{17}, t_{18}\})$, ${}^\bullet\varphi_2= (\{L_2.p_{15}, L_2.p_{17}\},\{R_2.t_{15}\})$, and $\varphi_2{}^\bullet = (\{L_2.p_{15}, L_2.p_{17}\},\{R_2.t_{18}\})$.

The third system configuration involves rework. In this scenario, a part can be inspected after all operations have been completed. By using the proposed Algorithm 1, the production operations of the reworked part can be exactly and easily modeled by considering rework operations as alternative sequences. Suppose that an inspection machine M5 is added to a system and that part A is processed in M1 and M3. Then, part A is moved to an M5 by Robot 2 to check if there are defects in part A. If part A performs properly, then it will leave the system by Robot 2. Otherwise, if part A has defects, rework is needed, and part A is moved to M3 by Robot 2. To model the rework operation by using the synthesis procedure of Algorithm 1, we construct a configuration as a rewriting rule $\mathcal{R} = \{rr_3\}$ with $rr_3 = \{L_3, R_3, \varphi_3, {}^\bullet\varphi_3, \varphi_3{}^\bullet\}$, where L_3 and R_3 are illustrated in Figures 19a and 19b, respectively, ξ_3: $N_3 \rightarrow N_2$, $\varphi_3= (\{p_6, p_7, p_{14}, p_{18}, p_{24}, p_{25}, p_{26}, p_{27}\},\{t_6, t_7, t_8, t_{19}, t_{20}, t_{21}, t_{22}\})$, ${}^\bullet\varphi_3= (\{L_3.t_6\},\{R_3.t_6\})$, and $\varphi_3{}^\bullet = (\{L_3.t_8\},\{R_3.t_{21}\})$. The Specifications of S^3PR net illustrated in Figure 16 under changeable control specifications are shown in Table 2. In addition, the required monitors using Algorithm 2 of the system illustrated in Figure 16 under changeable control specifications are shown in Table 3.

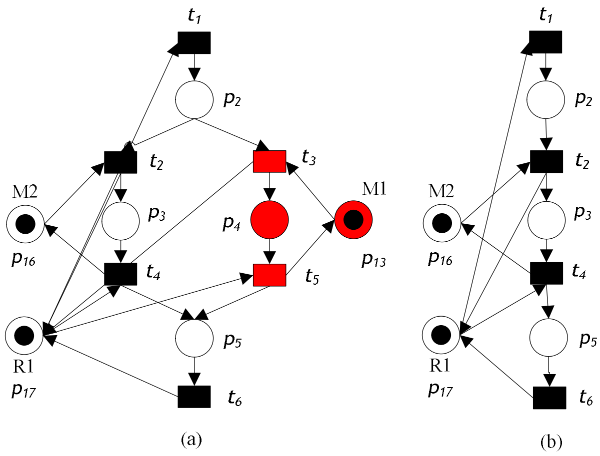

Figure 17. A reconfigured S^3PR net by removing a machine. (**a**) Left hand side net *L*. (**b**) Right hand side net *R*.

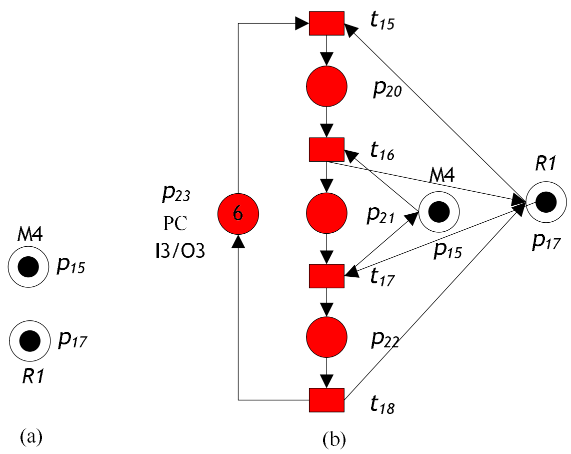

Figure 18. A reconfigured S^3PR net by adding a product. (**a**) Left hand side net *L*. (**b**) Right hand side net *R*.

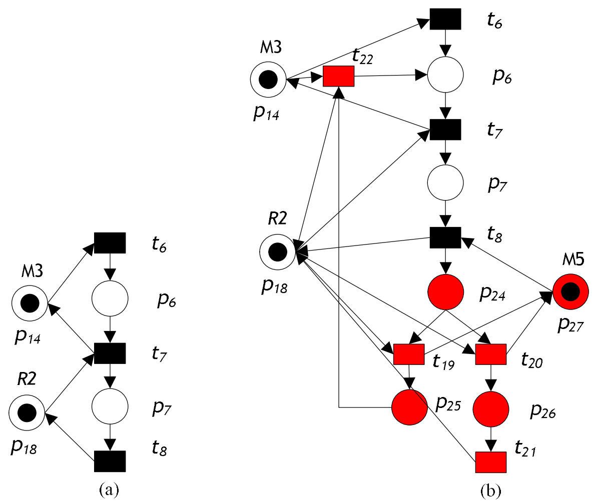

Figure 19. A reconfigured S^3PR net by rework. (**a**) Left hand side net *L*. (**b**) Right hand side net *R*.

Table 2. The Specifications of S^{3}PR net illustrated in Figure 16 under configurations.

Parameter	Configuration			
	An Initial S^{3}PR Net	Removal of an Old Machine	Addition of a New Product	Rework
No. of monitors	5	3	5	10
No. of arcs	21	12	26	48
Liveness	Live	Live	Live	Live
Boundedness	Bounded	Bounded	Bounded	Bounded
Reversibility	Reversible	Reversible	Reversible	Reversible

Table 3. Required monitors using Algorithm 2 of the system illustrated in Figure 16 under configurations.

Configuration	i	Siphon	$^\bullet V_{Si}$	$V_{Si}{}^\bullet$	M_{RCo} (V_{si})
An initial S^{3}PR net	1	S_1	t_7,t_{13}	t_1,t_9	5
	2	S_2	t_4,t_5,t_{13}	t_1,t_{11}	2
	3	S_3	t_7,t_{13}	t_1,t_9	4
	4	S_4	t_7,t_{11}	t_1,t_9	3
	5	S_5	t_4,t_{13}	t_2,t_{11}	1
Removal of an old machine	1	S_1	t_4,t_{13}	t_1,t_9	1
	2	S_2	t_7,t_{11}	t_4,t_9	3
	3	S_3	t_7,t_{13}	t_1,t_9	4
Addition of a new product	1	S_1	t_{11},t_{17}	t_{10},t_{15}	1
	2	S_2	t_4,t_{13},t_{17}	t_1,t_{10},t_{15}	2
	3	S_3	t_4,t_{13}	t_1,t_9	1
	4	S_4	t_7,t_{11},t_{17}	t_4,t_9,t_{15}	3
	5	S_5	t_7,t_{13},t_{17}	t_1,t_9,t_{15}	4
Rework	1	S_1	t_7	t_6,t_{19}	1
	2	S_2	t_{20}	t_6	2
	3	S_3	t_6,t_{20}	t_7	1
	4	S_4	t_7,t_{11},t_{17}	t_4,t_9,t_{15},t_{19}	3
	5	S_5	t_7,t_{13},t_{17}	t_1,t_9,t_{15},t_{19}	4
	6	S_6	t_{11},t_{17},t_{20}	t_4,t_9,t_{15}	4
	7	S_7	t_{13},t_{17},t_{20}	t_1,t_9,t_{15}	5
	8	S_8	t_{11},t_{17}	t_8,t_{15}	1
	9	S_9	t_4,t_{13},t_{17}	t_1,t_{10},t_{15}	2
	10	S_{10}	t_4,t_{13}	t_1,t_9	1

The controlled net after adding above changeable control specifications is illustrated in Figure 20. The place and arcs of the controller are illustrated with blue lines.

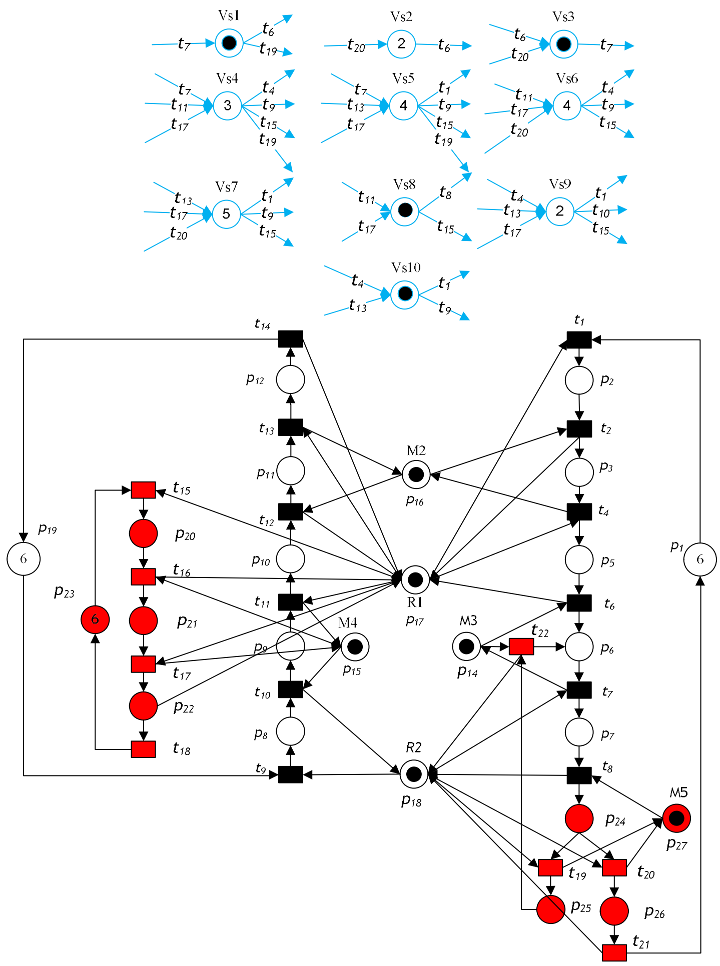

Figure 20. Controlled reconfigurable S^3PR net after adding changeable control specifications.

6. Conclusions

This paper develops a novel two-step solution for quick and accurate reconfiguration of supervisory controllers for deadlock control in RMSs with dynamic changes. In the first step, the net rewriting system is used to design a reconfigurable PN model under dynamic configurations. The obtained model guarantees boundedness behavioral property but may not guarantee the other properties of a Petri net model (i.e., liveness and reversibility). The second step proposes an automatic deadlock prevention policy for reconfigurable Petri net using the siphon control method based on a place invariant to solve the deadlock problem with dynamic structure changes in RMSs and guarantee the liveness and reversibility properties for the system. The proposed method is validated using the GPenSIM tool and compared with existing methods in the literature to highlight its ability of adapting to RMS configuration changes.

The major advantages of the developed approach are as follows: (1) It does not need to compute reachability graphs as illustrated in Algorithm 2, Section 3, and has low-overhead computation as proved in Theorems 10 and 11, Section 4.4. (2) It can automatically and dynamically modify the structure of a Petri net model without affecting its behavioral properties, i.e., liveness, boundedness, and reversibility as illustrated in Algorithm 2, Section 3. (3) It allows rapid reconfigurability and reusability of the controller during reconfiguration as shown in Algorithm 2, Section 3. (4) It can easily handle any dynamical changes in RMSs compared with the studies in Badouel et al. [39], Llorens and Oliver [34], Wu and Zhou [25], and Kaid et al. [7] as shown in Algorithm 2, Section 3. (5) The GPenSIM code is developed for designing, simulation, validation, and performance analysis of deadlock problems with dynamic structure changes in RMSs and the correctness of the proposed approach is proven and compared with the studies in Ezpeleta et al. [43], Li and Zhou [44], and Kaid et al. [6] as shown in Section 4.5. (6) Based on Theorems 10 and 11, the computational complexity of the proposed approach has polynomial time complexity. Therefore, it has low computational complexity and can be applicable to other types of complex systems such as mass customization manufacturing, lean productivity, agile manufacturing, and flexible manufacturing systems. (7) It can consider systems with sequential and complex resource requirements, meaning that a set of system resources can be used and shared to process each component according to sequential processes that depend on the step-by-step discrete execution and multiple processes that depend on the execution at the same time as shown in numerical example.

The limitation of the developed approach is that the obtained models lack an appropriate conversion approach from the PN model into control languages for application. Thus, our future research will examine the developed approach to have an automatic method to examine the applicability of the obtained models for real world manufacturing systems.

Author Contributions: Conceptualization, H.K. and A.A.-A.; software, H.K. and R.D.; resources, H.K., A.A.-A. and R.D.; formal analysis, H.K. and A.A.-A.; investigation, H.K., A.A.-A. and Z.L.; validation, H.K., A.A.-A., Z.L. and R.D.; writing—original draft preparation, H.K., A.A.-A. and Z.L.; writing—review and editing, H.K., A.A.-A., Z.L. and R.D.; visualization, H.K., A.A.-A. and Z.L.; supervision, A.A.-A., and Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by King Saud University through Researchers Supporting Project Number (RSP-2020/62).

Acknowledgments: The authors would like to thank King Saud University for funding and supporting this research through Researchers Supporting Project Number (RSP-2020/62).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hu, Y.; Ma, Z.; Li, Z. Design of Supervisors for Active Diagnosis in Discrete Event Systems. *IEEE Trans. Autom. Control* **2020**. [CrossRef]
2. Wang, D.; Wang, X.; Li, Z. Nonblocking Supervisory Control of State-Tree Structures with Conditional-Preemption Matrices. *IEEE Trans. Ind. Inform.* **2019**, *16*, 3744–3766. [CrossRef]
3. Mehrabi, M.G.; Ulsoy, A.G.; Koren, Y. Reconfigurable manufacturing systems: Key to future manufacturing. *J. Intell. Manuf.* **2000**, *11*, 403–419. [CrossRef]
4. Katz, R. Design principles of reconfigurable machines. *Int. J. Adv. Manuf. Technol.* **2007**, *34*, 430–439. [CrossRef]
5. Patel, R.; Gojiya, A.; Deb, D. Failure Reconfiguration of Pumps in Two Reservoirs Connected to Overhead Tank. In *Innovations in Infrastructure*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 81–92.
6. Kaid, H.; Al-Ahmari, A.; Li, Z.; Davidrajuh, R. Single controller-based colored Petri nets for deadlock control in automated manufacturing systems. *Processes* **2020**, *8*, 21. [CrossRef]
7. Kaid, H.; Al-Ahmari, A.; Li, Z.; Davidrajuh, R. Intelligent colored token Petri nets for modeling, control, and validation of dynamic changes in reconfigurable manufacturing systems. *Processes* **2020**, *8*, 358. [CrossRef]
8. Li, Z.; Zhou, M. *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
9. Zan, X.; Wu, Z.; Guo, C.; Yu, Z. A Pareto-based genetic algorithm for multi-objective scheduling of automated manufacturing systems. *Adv. Mech. Eng.* **2020**, *12*, 1687814019885294. [CrossRef]
10. Li, L.; Basile, F.; Li, Z. An approach to improve permissiveness of supervisors for GMECs in time Petri net systems. *IEEE Trans. Autom. Control* **2019**, *65*, 237–251. [CrossRef]
11. Liu, Y.; Cai, K.; Li, Z. On scalable supervisory control of multi-agent discrete-event systems. *Automatica* **2019**, *108*, 108460. [CrossRef]
12. Chen, Q.; Yin, L.; Wu, N.; El-Meligy, M.A.; Sharaf, M.A.F.; Li, Z. Diagnosability of vector discrete-event systems using predicates. *IEEE Access* **2019**, *7*, 147143–147155. [CrossRef]

13. Kaid, H.; Al-Ahmari, A.; Nasr, E.A.; Al-Shayea, A.; Kamrani, A.K.; Noman, M.A.; Mahmoud, H.A. Petri Net Model Based on Neural Network for Deadlock Control and Fault Detection and Treatment in Automated Manufacturing Systems. *IEEE Access* **2020**, *8*, 103219–103235. [CrossRef]
14. Zhao, M. An integrated control method for designing non-blocking supervisors using Petri nets. *Adv. Mech. Eng.* **2017**, *9*, 1687814017700829. [CrossRef]
15. Chen, Y.; Li, Z.; Khalgui, M.; Mosbahi, O. Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems. *Autom. Sci. Eng. IEEE Trans.* **2011**, *8*, 374–393. [CrossRef]
16. Al-Ahmari, A.; Kaid, H.; Li, Z.; Davidrajuh, R. Strict Minimal Siphon-Based Colored Petri Net Supervisor Synthesis for Automated Manufacturing Systems with Unreliable Resources. *IEEE Access* **2020**. [CrossRef]
17. Kaid, H.; Al-Ahmari, A.; El-Tamimi, A.M.; Abouel Nasr, E.; Li, Z. Design and implementation of deadlock control for automated manufacturing systems. *South Afr. J. Ind. Eng.* **2019**, *30*, 1–23. [CrossRef]
18. Chao, D.Y. Improvement of suboptimal siphon-and FBM-based control model of a well-known S^3PR. *IEEE Trans. Autom. Sci. Eng.* **2011**, *8*, 404–411. [CrossRef]
19. Ghaffari, A.; Rezg, N.; Xie, X. Design of a live and maximally permissive Petri net controller using the theory of regions. *IEEE Trans. Robot. Autom.* **2003**, *19*, 137–141. [CrossRef]
20. Uzam, M. The use of the Petri net reduction approach for an optimal deadlock prevention policy for flexible manufacturing systems. *Int. J. Adv. Manuf. Technol.* **2004**, *23*, 204–219. [CrossRef]
21. Sun, D.; Chen, Y.; El-Meligy, M.A.; Sharaf, M.A.F.; Wu, N.; Li, Z. On algebraic identification of critical states for deadlock control in automated manufacturing systems modeled with Petri nets. *IEEE Access* **2019**, *7*, 121332–121349. [CrossRef]
22. Nasr, E.A.; El-Tamimi, A.M.; Al-Ahmari, A.; Kaid, H. Comparison and Evaluation of Deadlock Prevention Methods for Different Size Automated Manufacturing Systems. *Math. Probl. Eng.* **2015**, *501*, 1–19. [CrossRef]
23. Lee, S.; Tilbury, D.M. Deadlock-free resource allocation control for a reconfigurable manufacturing system with serial and parallel configuration. *IEEE Trans. Syst. ManCybern. Part C (Appl. Rev.)* **2007**, *37*, 1373–1381. [CrossRef]
24. Li, J.; Dai, X.; Meng, Z. Automatic reconfiguration of petri net controllers for reconfigurable manufacturing systems with an improved net rewriting system-based approach. *IEEE Trans. Autom. Sci. Eng.* **2009**, *6*, 156–167. [CrossRef]
25. Wu, N.; Zhou, M. Intelligent token Petri nets for modelling and control of reconfigurable automated manufacturing systems with dynamical changes. *Trans. Inst. Meas. Control* **2011**, *33*, 9–29.
26. Sampath, R.; Darabi, H.; Buy, U.; Liu, J. Control reconfiguration of discrete event systems with dynamic control specifications. *IEEE Trans. Autom. Sci. Eng.* **2008**, *5*, 84–100. [CrossRef]
27. Dumitrache, I.; Caramihai, S.; Stanescu, A. Intelligent agent-based control systems in manufacturing. In Proceedings of the 2000 IEEE International Symposium on Intelligent Control. Held jointly with the 8th IEEE Mediterranean Conference on Control and Automation (Cat. No. 00CH37147), Rio Patras, Greece, 19–19 July 2000; pp. 369–374.
28. Ohashi, K.; Shin, K.G. Model-based control for reconfigurable manufacturing systems. In Proceedings of the 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164), Seoul, Korea, 21–26 May 2001; pp. 553–558.
29. Kalita, D.; Khargonekar, P.P. Formal verification for analysis and design of logic controllers for reconfigurable machining systems. *IEEE Trans. Robot. Autom.* **2002**, *18*, 463–474. [CrossRef]
30. Almeida, E.E.; Luntz, J.E.; Tilbury, D.M. Event-condition-action systems for reconfigurable logic control. *IEEE Trans. Autom. Sci. Eng.* **2007**, *4*, 167–181. [CrossRef]
31. Zhang, L.; Rodrigues, B. Modelling reconfigurable manufacturing systems with coloured timed Petri nets. *Int. J. Prod. Res.* **2009**, *47*, 4569–4591. [CrossRef]
32. Kahloul, L.; Bourekkache, S.; Djouani, K.; Chaoui, A.; Kazar, O. Using high level Petri nets in the modelling, simulation and verification of reconfigurable manufacturing systems. *Int. J. Softw. Eng. Knowl. Eng.* **2014**, *24*, 419–443. [CrossRef]
33. Yu, Z.; Guo, F.; Ouyang, J.; Zhou, L. Object-oriented Petri nets and π-calculus-based modeling and analysis of reconfigurable manufacturing systems. *Adv. Mech. Eng.* **2016**, *8*, 1687814016677698. [CrossRef]
34. Llorens, M.; Oliver, J. Structural and dynamic changes in concurrent systems: Reconfigurable Petri nets. *IEEE Trans. Comput.* **2004**, *53*, 1147–1158. [CrossRef]

35. Li, J.; Dai, X.; Meng, Z. Improved net rewriting systems-based rapid reconfiguration of Petri net logic controllers. In Proceedings of the 31st Annual Conference of IEEE Industrial Electronics Society, Raleigh, NC, USA, 6–10 November 2005.
36. da Silva, R.M.; Benítez-Pina, I.F.; Blos, M.F.; Santos Filho, D.J.; Miyagi, P.E. Modeling of reconfigurable distributed manufacturing control systems. *IFAC-Pap.* **2015**, *48*, 1284–1289. [CrossRef]
37. Berthomieu, B.; Ribet, P.-O.; Vernadat, F. The tool TINA–construction of abstract state spaces for Petri nets and time Petri nets. *Int. J. Prod. Res.* **2004**, *42*, 2741–2756. [CrossRef]
38. Bonet, P.; Catalina, M.L.; Puigjaner, R. A Petri Net tool for performance modeling. In Proceedings of the 23rd Latin American Conference on Informatics (CLEI 2007), San Jose, Costa Rica, October 2007.
39. Badouel, E.; Llorens, M.; Oliver, J. Modeling concurrent systems: Reconfigurable nets. In Proceedings of the PDPTA, Las Vegas, NV, USA, 23–26 June 2003; pp. 1568–1574.
40. Yamalidou, K.; Moody, J.; Lemmon, M.; Antsaklis, P. Feedback control of Petri nets based on place invariants. *Automatica* **1996**, *32*, 15–28. [CrossRef]
41. Moody, J.O.; Antsaklis, P.J. *Supervisory Control of Discrete Event Systems Using Petri Nets*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 8.
42. Davidrajuh, R. *Modeling Discrete-Event Systems with GPenSIM: An Introduction*; Springer: Berlin/Heidelberg, Germany, 2018.
43. Ezpeleta, J.; Colom, J.M.; Martinez, J. A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Trans. Robot. Autom.* **1995**, *11*, 173–184. [CrossRef]
44. Li, Z.; Zhou, M. Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2004**, *34*, 38–51. [CrossRef]
45. Piroddi, L.; Cordone, R.; Fumagalli, I. Selective siphon control for deadlock prevention in Petri nets. *IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum.* **2008**, *38*, 1337–1348. [CrossRef]
46. Chen, Y.; Li, Z.; Zhou, M. Behaviorally optimal and structurally simple liveness-enforcing supervisors of flexible manufacturing systems. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2012**, *42*, 615–629. [CrossRef]