engineering proceedings

MDPI

Proceeding Paper

Deadlock Prevention Policy for Flexible Manufacturing Systems: Petri Net-Based Approach Utilizing Iterative Synthesis and Places Invariant †

Shih-Chih Lee, Jui-Fu Cheng and Ter-Chan Row *

Department of Electrical and Electronic Engineering, Army Academy, Taoyuan 32092, Taiwan; sclee1028@gmail.com (S.-C.L.); tcypvs@gmail.com (J.-F.C.)

* Correspondence: lolororo1234@gmail.com; Tel.: +886-918214862

† Presented at the 2025 IEEE 5th International Conference on Electronic Communications, Internet of Things and Big Data, New Taipei, Taiwan, 25–27 April 2025.

Abstract

An iterative method was developed in this study within a Petri net system (PNS) for flexible manufacturing systems (FMSs) to eliminate deadlocks. The algorithm employs variant tokens, ranging from a few to many, in idle places to identify subnet deadlocks through reachability states. Additionally, it utilizes the place invariant (PI) to resolve deadlocks. An algorithm with a simple example was developed and applied using a complex scenario. The developed algorithm is user-friendly and effectively eliminates deadlocks in the PNS of FMSs, producing optimal results.

Keywords: petri net system; control place; places invariant; reachability state

1. Introduction

Deadlocks occur from simultaneous attempts to access and share a resource, creating an undesired situation in a Petri net of flexible manufacturing systems. The sole method to resolve deadlocks is to reassign the workflow of the shared resource. Three solutions are used to eliminate deadlocks: deadlock avoidance, deadlock detection and recovery, and deadlock prevention [1,2].

Deadlock avoidance involves implementing an online control method to identify the correct evolution of the Petri net system (PNS). The primary objective is to ensure that the PNS avoids deadlocks and operates smoothly. Consequently, this method does not eliminate all deadlocks in the PNS [3–5]. The second method is deadlock detection and recovery. When deadlocks happen, a deadlock-free state is detected, and the deadlock is omitted by relocating resources. Generally, this approach demands a large amount of data to guide the algorithm, making it complex, especially with shared resources [6–10]. The third method is the deadlock prevention policy. This method aims to prevent deadlocks statically for optimal reachability states. No dead markings are reached after implementing the deadlock prevention policy. We used the deadlock prevention policy to address deadlocks in the PNS to obtain optimal results [1,2,6,11]. We employed variant token numbers in specific locations to detect deadlocks and applied the PI to eliminate them through iterative computation. The algorithm yielded results consistent with those in Ref. [6].

The remainder of this article is organized as follows. The algorithm for the deadlock prevention policy is presented in Section 2. Section 3 provides two illustrative examples

check for updates

Academic Editors: Teen-Hang Meen, Shu-Han Liao and Cheng-Fu Yang

Published: 8 September 2025

Citation: Lee, S.-C.; Cheng, J.-F.; Row, T.-C. Deadlock Prevention Policy for Flexible Manufacturing Systems: Petri Net-Based Approach Utilizing Iterative Synthesis and Places Invariant. *Eng. Proc.* **2025**, *108*, 37. https://doi.org/10.3390/engproc2025108037

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

demonstrating the detailed application of the algorithm. Finally, the conclusion is presented in Section 4.

2. Iterative Synthesis and PI

We introduced the deadlock prevention policy in this study [12]. We employed iterative syntheses as a proactive approach to eliminate deadlocks within the PNS. This strategy aligns with the findings and recommendations presented in the referenced report, contributing to the enhancement of deadlock resilience in our system.

For two idle places (p_1, p_4), two resource places (p_7, p_8), and operation places (active places), the saturated line is (p_1, p_4) = (2, 2) [13] and the maximal reachability states are 8. In this case, the PNS has one deadlock. The PI is obtained by using the control place (CP) equation ($D_c = -L \cdot D_p$, $u_c = b - u_p$) where D_c is the PNS controller, L is an $n_c \times n$ matrix, D_p is the PNS of the process, u_c is the marking of the controller places which is an $n_c \times 1$ vector, b is an $n_c \times 1$ vector, and u_p is the marking vector of PNS modeling the process [12,14–16]. n_c is the number of constraints by the $u_i + u_j \leq 1$ where u_i and u_j are the marking places of p_i and p_j by the process PNS. One can refer to [12] for a more in-depth description.

The iterative number token of the idle places from the small to the saturated line [13] was used to search the deadlocks. Then, the PI was used to obtain CPs (D_c) and select dead marking minus 1. The algorithm is presented in Figure 1. The maximal reachability states (RSs) are 8, and the saturated line is (p_1, p_4) = (2, 2). When (p_1, p_4) is equal to (1, 1), a dead marking ($M_4 = p_2 + p_5$) is obtained. The operation places are omitted. According to the dead marking, D_p is obtained using Equation (1). D_c ($D_c = -L \cdot D_p$) when $D_c = t_2 + t_5 - t_1 - t_4$. The dead marking coefficient sum (DMCS) is 2 in this case. By subtracting 1 from the DMCS, the marking of the controller places (u_c = 1) is obtained.

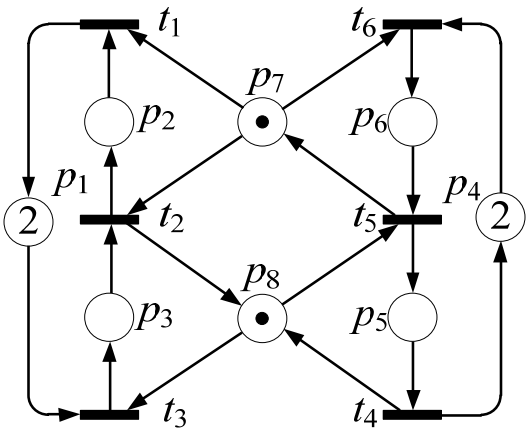


Figure 1. Simple PNS [14].

Figure 2 shows the result of the CP in time PNA (TINA) [17].

$$D_p = \begin{array}{c} \\ p_2 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_4 & t_5 \end{array} \\ \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \end{array} \quad (1)$$

In the above algorithm, the following parameters are defined.

- Input: PNM prone to deadlocks
- Output: PNM without deadlocks
- Saturated line for PNS
- Subtracting 1 from the saturated line as the token number in idle places

- Searching the dead markings
- $D_c = -L \cdot D_p$, $u_c = b - u_p$ to obtain CPs
- Zero deadlocks
- Obtaining the live PNS
- End

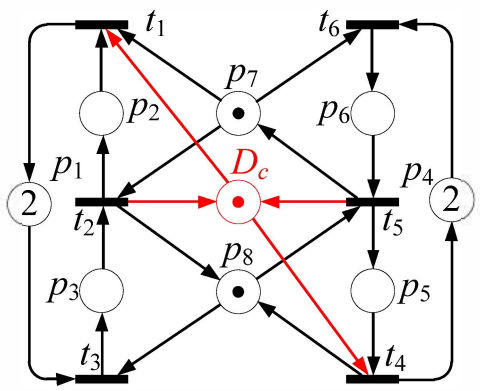


Figure 2. Simple Petri net with CP.

3. FMS

An FMS of the factory is shown in Figure 3a and its sequence production is illustrated in Figure 3b [6,11]. The PNS is illustrated in Figure 4. In the figure, p_{21} and p_{22} are I/O (input and output), p_{12}, p_{13}, and p_{15} are machines 1–3, p_{11}, and p_{14} are robots 1–2. The initial reachability states are 380; this PNS has a live lock, so it is not live and not reversible.

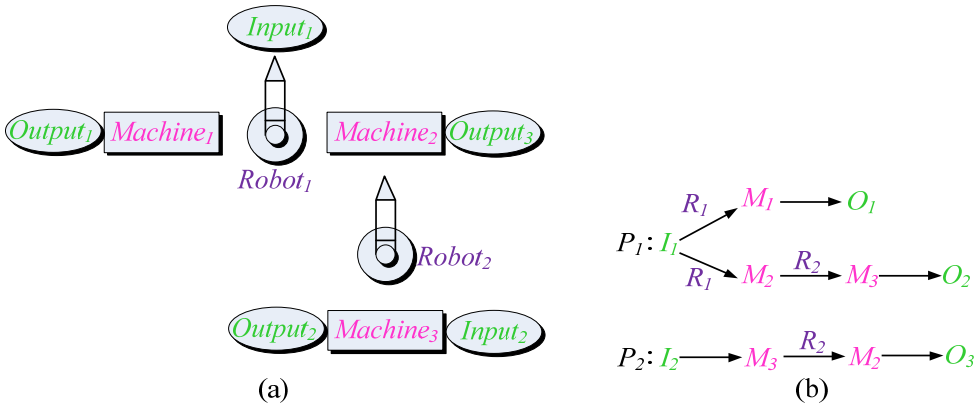


Figure 3. (**a**) Layout of FMS and (**b**) sequences of production [6,11].

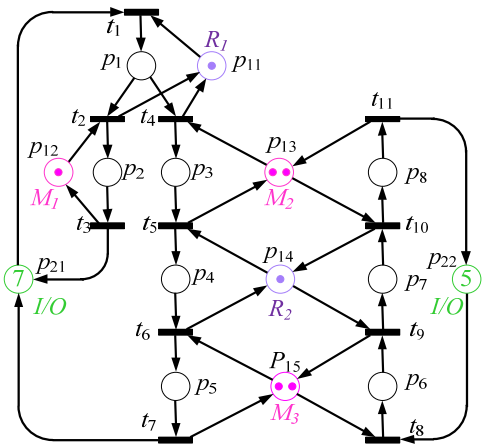


Figure 4. PNS of FMS [3,4].

In the developed method, (p_{21}, p_{22}) = (1, 2) and the dead marking was (p_{11} + p_{12} + 2p_{13} + p_4 + 2p_6). From the deadlock, we only considered the operation places (p_4 + 2p_6) and omitted other places. The total token number in operation places was 3. To limit the step not to enter p_4 + 2p_6, d the dead marking must be avoided. This means that u_4 + u_6 ≤ 2 and the control token is 2 [12]. D_{p1} cis calculated as follows.

$$D_{p1} = \begin{matrix} \\ p_4 \\ p_6 \end{matrix} \begin{matrix} \begin{matrix} t_5 & t_6 & t_8 & t_9 \end{matrix} \\ \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \end{matrix} \quad (2)$$

CP 1 is obtained as $D_{c1} = -t_5 + t_6 - t_8 + t_9$ by using $D_{c1} = -L \cdot \boldsymbol{D_{p1}}$. When ($p_{21}$, p_{22}) = (2, 1), the dead marking is 2cp_1 + p_{11} + p_{12} + 2p_{15} + 2p_3 + p_7. When u_3 + u_7 ≤ 2 and $D_{c2} = -t_4 + t_5 - t_9 + t_{10}$, D_{p2} is calculated as follows.

$$D_{p2} = \begin{matrix} \\ p_3 \\ p_7 \end{matrix} \begin{matrix} \begin{matrix} t_4 & t_5 & t_9 & t_{10} \end{matrix} \\ \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \end{matrix} \quad (3)$$

When (p_{21}, p_{22}) = (2, 2), the dead marking is p_{11} + p_{12} + p_{14} + 2p_3 + 2p_6. When u_3 + u_6 ≤ 3 and $D_{c3} = -t_4 + t_5 - t_8 + t_9$, D_{p3} is calculated as follows.

$$D_{p3} = \begin{matrix} \\ p_3 \\ p_6 \end{matrix} \begin{matrix} \begin{matrix} t_4 & t_5 & t_8 & t_9 \end{matrix} \\ \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \end{matrix} \quad (4)$$

CPs are presented in Table 1 and Figure 5.

Table 1. Three CPs.

(p_{21}, p_{22})	Dead Marking	Tokens	CP
(1, 2)	$p_{11} + p_{12} + 2p_{13} + p_4 + 2p_6$	2	$D_{c1} = -t_5 + t_6 - t_8 + t_9$
(2, 1)	$2cp_1 + p_{11} + p_{12} + 2p_{15} + 2p_3 + p_7$	2	$D_{c2} = -t_4 + t_5 - t_9 + t_{10}$
(2, 2)	$p_{11} + p_{12} + p_{14} + 2p_3 + 2p_6$	3	$D_{c3} = -t_4 + t_5 - t_8 + t_9$

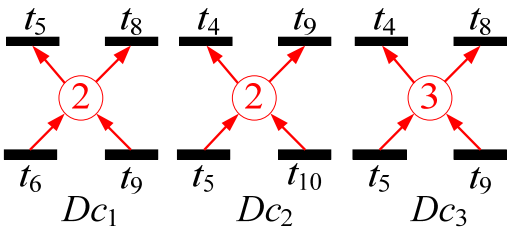


Figure 5. CPs.

Table 2 presents a comparison of the CP number, the number of tokens in each CP, and the reachability states (RSs) with those in Ref. [6].

Table 2. Comparison of results.

Item	Reference	This Study
CP number	3	3
CP token	7	7
RS	336	336

4. Conclusions

We identified deadlocks through the repeated analysis of small blocks and the application of the PI method. The developed algorithm in this study was experimentally proven to be feasible [18–22]. Optimal results were obtained by employing the theory of regions [23] and dividing the PNS structure into subnets to identify CPs. By utilizing simple variant tokens in idle places, we constructed the subnet PNS, thereby simplifying deadlocks in the reachability graph. This approach enables easy access to CPs and successfully mitigates the state exploration problem [18–22]. In future research, it is necessary to extend the applicability of the algorithm to larger PNSs and obtain optimal results.

Author Contributions: Conceptualization, S.-C.L. and T.-C.R.; methodology, S.-C.L. and T.-C.R.; software, T.-C.R.; validation, S.-C.L., J.-F.C. and T.-C.R.; formal analysis, S.-C.L. and T.-C.R.; investigation, S.-C.L. and T.-C.R.; resources, S.-C.L., J.-F.C. and T.-C.R.; data curation, S.-C.L. and T.-C.R.; writing—original draft preparation, S.-C.L. and T.-C.R.; writing—review and editing, S.-C.L. and T.-C.R.; visualization, J.-F.C.; supervision, S.-C.L. and T.-C.R.; project administration, S.-C.L. and J.-F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: This study was exempt from ethical review and approval.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data provided in this study are available from the corresponding author upon request.

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

References

1. Uzam, M. An Optimal Deadlock Prevention Policy for Flexible Manufacturing Systems Using Petri Net Models with Resources and the Theory of Regions. *IEEE Int. J. Adv. Manuf. Technol.* **2002**, *19*, 192–208. [CrossRef]
2. Uzam, M. The use of Petri net reduction approach for an optimal deadlock prevention policy for flexible manufacturing systems. *IEEE Int. J. Adv. Manuf. Technol.* **2003**, *23*, 204–219. [CrossRef]
3. Hsieh, F.S.; Chang, S.C. Dispatching-driven deadlock avoidance controller synthesis for flexible manufacturing systems. *IEEE Trans. Robot. Automat.* **1994**, *10*, 196–209. [CrossRef]
4. Fanti, M.P.; Maione, B.; Mascolo, S.; Turchiano, A. Event-based feedback control for deadlock avoidance in flexible production systems. *IEEE Trans. Robot. Automat.* **1997**, *13*, 347–363. [CrossRef]
5. Yim, D.-S.; Kim, J.-I.; Woo, H.-S. Avoidance of deadlocks in flexible manufacturing systems using a capacity-designated directed graph. *Int. J. Prod. Res.* **1997**, *35*, 2459–2476. [CrossRef]
6. Uzam, M.; Zhou, M.C. An Iterative Synthesis Approach to Petri Net-Based Deadlock Prevention Policy for Flexible Manufacturing Systems. *IEEE Trans. Syst. Man Cybern. A Syst. Hum.* **2007**, *37*, 362–371. [CrossRef]
7. Row, T.C.; Pan, Y.L. An Optimal Deadlock Recovery Algorithm for Special and Complex Flexible Manufacturing Systems-S^4PR. *IEEE Access* **2021**, *9*, 111083–111094. [CrossRef]
8. Row, T.C.; Syu, W.M.; Pan, Y.L.; Wang, C.C. One Novel and Optimal Deadlock Recovery Policy for Flexible Manufacturing Systems Using Iterative Control Transitions Strategy. *Math. Probl. Eng.* **2019**, *2019*, 1–12. [CrossRef]
9. Wysk, R.A.; Yang, N.S.; Joshi, S. Resolution of deadlocks in flexible manufacturing systems: Avoidance and recovery approaches. *J. Manuf. Syst.* **1994**, *13*, 128–138. [CrossRef]
10. Cho, H.; Kumaran, T.K.; Wysk, R.A. Graph-theoretic deadlock detection and resolution for flexible manufacturing systems. *IEEE Trans. Robot. Automat.* **1995**, *11*, 413–421.
11. 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]
12. Yamalidou, K.; Moody, J.; Lemmon, M.; Antsaklis, P. Feedback control of Petri nets based on place invariants. *Automatica* **1996**, *32*, 15–28. [CrossRef]
13. Row, T.C.; Lee, S.C.; Pan, Y.L. Identifying the Saturated Line Based on the Number of Idle Places: Achieving Precise Maximal Permissiveness without Deadlocks Using Control Transitions or CPs. *Processes* **2023**, *11*, 2325. [CrossRef]

14. Li, Z.; Wu, N.; Zhou, M. Deadlock Control of Automated Manufacturing Systems Based on Petri Nets—A Literature Review. *IEEE Trans. Syst. Man Cybern. Part C* **2011**, *42*, 437–462.
15. Row, T.C.; Pan, Y.L. Incomparable Single Controller for Solving Deadlock Problem of Flexible Manufacturing Systems. *IEEE Access* **2023**, *11*, 45270–45278. [CrossRef]
16. Chen, Y.F.; Li, Z.W.; Zhou, M.C. Behaviorally optimal and structurally simple liveness-enforcing supervisors of flexible manufacturing systems. *IEEE Trans. Syst. Man, Cybern. A Syst. Humans Syst. Hum.* **2012**, *42*, 615–628. [CrossRef]
17. TINA (Time Petri Net Analyzer) Is a Toolbox for the Editing and Analysis of Petri Nets, with Possibly Inhibitor and Read Arcs, Time Petri Nets, with Possibly Priorities and Stopwatches, and an Extension of Time Petri Nets with Data Handling Called Time Transition Systems. 2020. Available online: http://projects.laas.fr/tina/home.php (accessed on 9 June 2020).
18. Notomi, M.; Murata, T. Hierarchical reachability graph of bounded Petri nets for concurrent-software analysis. *IEEE Trans. Softw. Eng.* **1994**, *20*, 325–336. [CrossRef]
19. McMillan, K.L. A technique for state space search based on unfolding. *Form. Methods Syst. Des.* **1995**, *6*, 45–65. [CrossRef]
20. Esparza, J.; Römer, S.; Volger, W. *An Improvement of McMillan's Unfolding Algorithm. Tools and Algorithms for the Construction and Analysis of Systems, Vol. 1055 of Lecture Notes in Computer Science*; Springer: Passau, Germany, 1996; pp. 87–106.
21. Heiner, M. Verification and optimization of control programs by Petri nets without state explosion. In Proceedings of the 2nd International Workshop on Manufacturing and Petri Nets, a Workshop Within the 28th International Conference on Applications and Theory of Petri Nets, Tolouse, France, 23 June 1997.
22. Taubin, A.; Kondratvey, A.; Kishinevsky, M. Deadlock prevention using Petri nets and their unfoldings. *Int. J. Adv. Manuf. Technol.* **1998**, *14*, 750–759. [CrossRef]
23. Badouel, E.; Darondeau, P. *Theory of Regions*; Reisig, W., Rozenberg, G., Eds.; Springer: Berlin, Germany, 1998; Volume 1491, pp. 529–586.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.