

基于 Petri 网的柔性装配系统的鲁棒控制

池旭,贺晓

(中国石油大学(华东) 计算机科学与计算学院, 山东 青岛 266580)

摘要:该文集中讨论了拥有不可靠柔性装配系统(Flexible Assembly System, FAS)的鲁棒死锁控制问题,确保当故障资源恢复后系统应恢复正常生产而不会出现堵塞现象,即不仅可以在无故障资源的情况下使系统远离死锁,而且还能在有不可靠资源时保证系统的顺利生产。首先,建立拥有不可靠资源 FAS 的 Petri 网模型,模拟整个系统的运行情况和资源的故障与恢复;其次,根据系统的控制目标,通过提出的进程可被推进算法和改进银行家算法设计了一种满足鲁棒性的死锁避免策略;然后,利用两个例子验证所设计的鲁棒控制策略的有效性;最后,对比实验结果表明,所提出的死锁控制策略所得到的可达标识数为 30 710 个,显著多于其他论文的策略,在可允许性上具有一定优越性。

关键词:柔性装配系统;死锁避免;Petri 网;鲁棒控制;银行家算法

中图分类号:TP237

文献标识码:A

文章编号:1673-629X(2024)06-0010-06

doi:10.20165/j.cnki.ISSN1673-629X.2024.0081

Robust Control of Flexible Assembly Systems Based on Petri Nets

CHI Xu, HE Xiao

(School of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China)

Abstract: We concentrate on the robust deadlock control problem of flexible assembly system (FAS) with unreliable flexible assembly system to ensure that the system should resume normal production without blocking when the faulty resources are recovered, i. e., not only can the system be kept away from deadlock in the absence of faulty resources, but also ensure the smooth production of the system in the presence of unreliable resources. Firstly, the Petri net model with unreliable resources FAS is established to simulate the operation of the whole system and the failure and recovery of resources. Secondly, according to the control objective of the system, a deadlock avoidance strategy satisfying the robustness is designed by the proposed process advanceable algorithm and the improved banker's algorithm, and then two examples are used to validate the validity of the designed robust control strategy. Finally, comparative experimental results show that the number of reachable identifiers obtained by the deadlock control strategy proposed is 30 710, which is significantly more than that of strategies of other papers, and is superior in terms of permissibility.

Key words: flexible assembly system; deadlock avoidance; Petri net; robust control; banker's algorithm

0 引言

在自动化制造系统(Automated Manufacturing System, AMS)中存在死锁问题,死锁可能导致系统无限期地阻塞。在实际生产过程中,无法保证资源完全可靠。具体而言,系统运行过程中,由于传感器故障、工具损坏、电气故障等不可控因素,一些资源可能会失去原有功能无法正常工作。这种随机发生的故障会导致系统死锁,使生产过程永久停滞,给自动化生产造成重大损失。建立能处理具有不可靠资源的鲁棒控制策略已成为制造业研究的重要问题之一。

目前,对具有不可靠资源的柔性制造系统的控制研究,大多采用 Petri 网进行建模。现行死锁政策主要

包括死锁避免、死锁预防和死锁检测与恢复。其中,死锁避免是指利用当前状态和未来过程资源的需求反馈信息,帮助系统远离死锁^[1-4];死锁预防预先建立离线控制策略,以使操作无死锁^[5-8];死锁检测与恢复中止一个或多个受害者进程以允许系统从任何检测到的死锁中恢复^[9]。该文从死锁避免角度设计策略来帮助 FAS 远离死锁。

文献[6]中,作者提出了一种新的死锁避免策略,以确保当所需的不可靠资源可用时,系统中剩余的所有工件都能顺利完成处理;文献[7]中,作者利用离散事件系统的监督控制理论,为顺序资源分配系统中的死锁避免提供了一种新的系统处理方法;文献[8]中,

收稿日期:2023-09-14

修回日期:2024-01-17

基金项目:山东省自然科学基金资助项目(ZR2020MF094);国家自然科学基金资助项目(61402216)

作者简介:池旭(1999-),女,硕士,CCF会员(J6268G),通信作者,研究方向为 Petri 理论及应用。

作者提出了一类 Petri 网对 AMS 进行建模,设计了一种用于确定是否可以终止主动过程的算法,并引入了一种新的银行家算法来开发具有基本鲁棒特性的死锁避免策略。文献[10]从结构复杂性、行为允许性和计算复杂性等方面对目前鲁棒控制策略进行了比较和分析。但上述工作没有考虑多个工件装配在一起生产产品的装配操作下的死锁情况。

柔性装配系统(Flexible Assembly System, FAS)中的鲁棒死锁问题受到关注。在这类系统中,死锁不仅与资源短缺和失效有关,还可能由于其他工件装配的等待过程所造成。因此,该系统的死锁控制问题比没有装配操作的 AMS 系统更困难^[11-12]。文献[11]开发了一种创新的分布式方法,该方法可以在不需要外部和额外信息的情况下动态和前瞻性地在线实施。文献[12]中,作者提出了一种鲁棒监督器,该监督器控制资源分配并选择柔性路径,使得需要故障资源的停滞工件不会阻碍不一定需要这些资源的工件的移动。

FAS 死锁问题是一个 NP 难的问题,研究者们一直致力于尽可能花费更少代价以提高系统的容许性能。目前关于具有不可靠资源的 FAS 死锁问题研究并不充分,每种方法都具有一定的局限性。例如,文献[11]只允许柔性路径使用不可靠资源,限制了系统的通用性。文献[12]中子系统的本地控制器之间的协同控制使得策略更加复杂。

该文考虑了具有不可靠资源的 FAS 死锁问题,根据改进的银行家算法为系统设计鲁棒控制策略以避免死锁。得到的鲁棒控制策略不仅可以在无故障资源的情况下使系统远离死锁,而且还能在有不可靠资源时保证系统的顺利生产。最后通过两个例子和实验分析表明了所设计的鲁棒控制策略的有效性。

1 预备知识

1.1 Petri 网

定义 1: 一个 Petri 网是一个加权二部图 $N = (P, T, F, W)$, 其中 P 是有限库所集, T 是有限变迁集, 且 $P \cap T = \emptyset$, $P \cup T \neq \emptyset$ 。库所指向变迁或变迁指向库所的有向弧集合用 $F \subseteq (P \times T) \cup (T \times P)$ 进行表示。每条有向弧都有对应的权值 W , $W: F \rightarrow \mathbb{N}^+ = \{1, 2, \dots\}$ 是给有向弧分配权值的函数。

给定一个 Petri 网 $N = (P, T, F, W)$, $x \in P \cup T$ 表示 N 中的库所或变迁, 则 x 的前集可以表示为 $\cdot x = \{y \in P \cup T \mid (y, x) \in F\}$ 。相应地, x 的后集可以表示为 $x \cdot = \{y \in P \cup T \mid (x, y) \in F\}$ 。

映射 $M: P \rightarrow \mathbb{N} = \{0, 1, 2, \dots\}$ 是表示 Petri 网 $N = (P, T, F, W)$ 的标识, 对于 $\forall p \in P$, $M(p)$ 表示标识 M 下库所 p 中的 token 数量。为降低表示复杂度, 可将 M

表示为 $\sum_{p \in P} M(p) \cdot p$ 。

当 $\forall p \in \cdot t$ 时, 若 $M(p) \geq W(p, t)$ 成立, 则称变迁 t 在标识 M 下是使能的, 记为 $M[t]$ 。若对于任意 $p \in P$, 存在 $M'(p) = M(p) - W(p, t) + W(t, p)$, 则称标识 M 发生使能变迁 t 可以到达 M' , 记为 $M[t]M'$ 。如果存在一个使能的变迁序列 $\sigma = t_1 t_2 \dots t_k$, 满足条件 $M[t_1]M_1[t_2] \dots [t_k]M'$, 则称标识 M' 从标识 M 发生变迁序列 $\sigma = t_1 t_2 \dots t_k$ 可达, 记为 $M[\sigma]M'$ 。 $R(N, M_0)$ 代表从初始标记 M_0 可到达的所有标记的集合。对于一个变迁序列 σ , $|T|$ 维向量 $\bar{\sigma}$ 表示变迁在 σ 中出现的总次数, $|T|$ 表示变迁集合的基数或大小。

给定两个 Petri 网 $N_1 = (P_1, T_1, F_1, W_1)$ 和 $N_2 = (P_2, T_2, F_2, W_2)$, $P_1 \cap P_2 \neq \emptyset$, $T_1 \cap T_2 = \emptyset$, 通过 N_1 和 N_2 的相同库所共享合成一个 Petri 网 $N_1 \oplus N_2 = (P, T, F, W)$, 其中 $P = P_1 \cup P_2$, $T = T_1 \cup T_2$, $F = F_1 \cup F_2$, 对于 $\forall (x, y) \in (P \times T) \cup (T \times P)$, 若 $(x, y) \in F_1$, 则 $W(x, y) = W_1(x, y)$; 若 $(x, y) \in F_2$, 则 $W(x, y) = W_2(x, y)$; 否则 $W(x, y) = 0$ 。

1.2 拥有可靠资源的 FAS 的 Petri 网

定义 2^[13]: 第 i 种产品的生产流程由一个 Petri 网 $N_i = (P_i, T_i, F_i, W_i) = (P_{i0} \cup P_{is} \cup P_{ir}, T_{is}, F_i, W_i)$ 表示, $i \in \{1, 2, \dots, k\}$, $k > 0$ 。其中:

(1) $P_{i0} = \{p_{i0}\}$ 表示空闲库所的集合; $P_{is} = \{p_{i1}, p_{i2}, \dots, p_{im}\}$ 表示活动库所的集合, $m > 0$; $P_{ir} = \{r_{i1}, r_{i2}, \dots, r_{in}\}$ 表示资源库所的集合, $n > 0$; $P_{i0} \cap P_{is} = \emptyset$, $P_{is} \cap P_{ir} = \emptyset$, 且 $P_{ir} \cap P_{i0} = \emptyset$ 。

(2) $T_{is} = \{t_{i1}, t_{i2}, \dots, t_{ih}\}$, $h > 0$ 。

(3) $F_i \subseteq (P_i \times T_i) \cup (T_i \times P_i)$ 。

(4) 若 $(x, y) \notin F_i$, 则 $W_i(x, y) = 0$; 否则 $W_i(x, y) \geq 1$ 。

定义 3^[13]: 通过共享资源将 $N_i (i = 1, 2, \dots, k)$ 合成所得 FAS 系统 Petri 网: $N = (P, T, F, W) = (P_0 \cup P_s \cup P_r, T_s, W) = \bigoplus_{i \in \{1, 2, \dots, k\}} N_i$, $k > 0$ 。其中 $P_0 = \bigcup_{i=1}^k P_{i0}$, $P_s = \bigcup_{i=1}^k P_{is}$, $P_r = \bigcup_{i=1}^k P_{ir}$, $T_s = \bigcup_{i=1}^k T_{is}$, $F = \bigcup_{i=1}^k F_i$ 。对于 $\forall (x, y) \in F$, 如果 $\forall (x, y) \in F_i$, 可得 $W(x, y) = W_i(x, y)$, 否则 $W(x, y) = 0$ 。

资源 r 的容量 $C(r)$, 即 r 中包含 $C(r)$ 个相同加工单元。对于任意 $p \in P_s$, $|P_r|$ 维向量 $\mathfrak{R}(p)$ 表示 p 代表的活动库所使用的资源。对于任意 $r \in P_r$, $\mathfrak{R}(p)(r)$ 代表与操作库所 p 相关的一个活动进程占用的 r 资源的副本数目。 $\forall p \in P_s$, $\forall t_1, t_2 \in p \cdot$, $W(p, t_1) = W(p, t_2)$ 。为了简便, 令 $W(p)$ 表示从 p 到 $p \cdot$ 中任何一个变迁的弧的权重。对 $\forall t \in T$, 若 $|{}^{(o)}t| > 1$, 则 t 为装配变迁。每个库所 $p \in {}^{(o)}t$ 称为装

配变迁 t 的装配活动库所。

定义 4^[13]: 对于一个 FAS Petri 网 $N = (P, T, F, W)$, 若满足以下条件, 则 M_0 是一个 N 可接受的初始标识:

- (1) $M_0(p) > 0, \forall p \in P_{i0}$;
- (2) $M_0(p) = 0, \forall p \in P_S$;
- (3) $M_0(r) \geq \mathfrak{R}(p)(r) \cdot W(p), \forall r \in P_R, \forall p \in P_S$;
- (4) $\forall t \in T$, 若 $|t \cap P| \geq 2$, 可得 $M_0(r) \geq \sum_{p \in P \cap t} (\mathfrak{R}(p)(r) \cdot W(p)), \forall r \in P_R, \forall p \in P_S$;
- (5) $\forall t \in T$, 若 $|t \cap P| \geq 2$, 可得 $M_0(r) \geq \sum_{p \in P \cap t} (\mathfrak{R}(p)(r) \cdot W(p)), \forall r \in P_R, \forall p \in P_S$ 。

图 1 是能生产一个产品的 FAS, 图 2 是建模的 Petri 网模型, 可接受初始标识为 $M_0 = 4p_0 + r_1 + r_2 + r_3 + r_4 + r_5 + r_6 + r_7$ 。其中, $P_0 = \{p_0\}, P_S = \{p_1 - p_8\}, P_R = \{r_1 - r_7\}, T_S = \{t_1 - t_9\}$; $C(r_1) = C(r_2) = C(r_3) = C(r_5) = C(r_6) = C(r_7) = 1, C(r_4) = 2$; $\mathfrak{R}(p_1) = [1, 0, 0, 0, 0, 0, 0, 0], \mathfrak{R}(p_2) = [0, 0, 0, 1, 0, 0, 0, 0], \mathfrak{R}(p_3) = [0, 0, 1, 0, 0, 0, 0, 0], \mathfrak{R}(p_4) = [0, 0, 0, 0, 1, 0, 0, 0], \mathfrak{R}(p_5) = [0, 1, 0, 0, 0, 0, 0, 0], \mathfrak{R}(p_6) = [0, 0, 0, 1, 0, 0, 0, 0], \mathfrak{R}(p_7) = [0, 1, 0, 0, 0, 1, 0, 0], \mathfrak{R}(p_8) = [0, 0, 0, 0, 0, 0, 0, 1]$; t_6 是装配变迁, p_4 和 p_7 是装配活动库所。



图 1 FAS 示例图

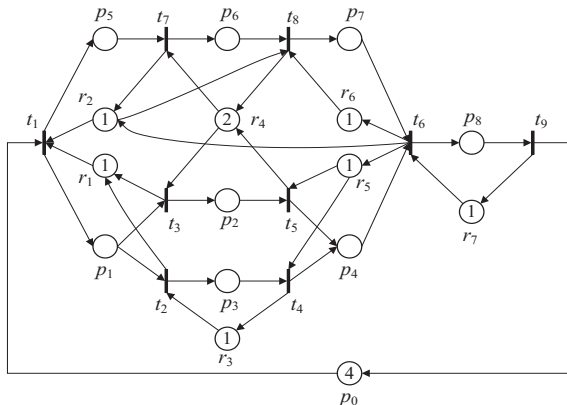


图 2 拥有可靠资源的 FAS 的 Petri 网

2 具有不可靠资源 FAS 的 Petri 网

定义 5: $N = (P, T, F, W) = (P_0 \cup P_S \cup P_R, T_S,$

$F, W)$ 是一个 FAS Petri 网, $P_R^u \subseteq P_R$ 是 N 的不可靠资源集合。对任意 $r \in P_R^u, r$ 的不可靠位置集合表示为 $U(r) = H(r) \cup \{r\} \subseteq P_S \cup P_R$, 其中 $H(r) = \{p \in P_S \mid \mathfrak{R}(p)(r) \geq 1\}$ 是使用资源 r 的活动库所的集合。

定义 6: 对 $r \in P_R^u$ 和一个不可靠库所 $z \in U(r) = H(r) \cup \{r\}$, z 的恢复子网是一个 Petri 网 $N_r(z) = (\{z, p_u(z)\}, \{t_\alpha(z), t_\beta(z)\}, F_r(z), W_r(z))$, 其中, $F_r(z) = \{(z, t_\alpha(z)), (t_\alpha(z), p_u(z)), (p_u(z), t_\beta(z)), (t_\beta(z), z)\}$, 且对 $\forall (x, y) \in F_r(z), W_r(z)(x, y) = 1$ 。

在上述定义 6 中, $t_\alpha(z)$ 称为 z 的故障变迁, $t_\beta(z)$ 称为 z 的恢复变迁, $p_u(z)$ 称为 z 的等待恢复库所。变迁 $t_\alpha(z)$ 的发生表示对应的不可靠资源故障。若资源处于空闲状态, 则进程无法使用此故障资源; 若资源处于工作状态, 则使用这个资源的进程将会阻塞。变迁 $t_\beta(z)$ 表示对应的不可靠资源已经修复并恢复正常工作。对 $\forall r \in P_R^u, \forall z \in U(r)$, 可以通过下面的定义 6 中所示的恢复子网对不可靠资源的故障和恢复进行建模。图 3 为一个恢复子网。

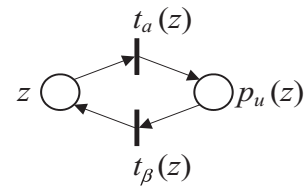


图 3 恢复子网

定义 7: 给定一个 FAS Petri 网 $N = (P, T, F, W) = (P_0 \cup P_S \cup P_R, T_S, F, W)$ 和不可靠资源集合 $P_R^u \subseteq P_R$ 。一个具有不可靠资源 FAS Petri 网 $N_u = (P_u, T_u, F_u, W_u) = N \oplus (\bigoplus_{r \in P_R^u, z \in U(r)} N_r(z))$, 其中:

- (1) 对于 $r \in P_R^u, z \in U(r), N_r(z) = (\{z, p_u(z)\}, \{t_\alpha(z), t_\beta(z)\}, F_r(z), W_r(z))$;
- (2) $P_u = P_0 \cup P_S \cup P_R \cup (\bigcup_{r \in P_R^u, z \in U(r)} \{p_u(z)\})$;
- (3) $T_u = T_S \cup (\bigcup_{r \in P_R^u, z \in U(r)} \{t_\alpha(z), t_\beta(z)\})$;
- (4) $F_u = F \cup (\bigcup_{r \in P_R^u, z \in U(r)} F_r(z))$;
- (5) 若 $(x, y) \in \bigcup_{r \in P_R^u, z \in U(r)} F_r(z)$, 则 $W_u(x, y) = 1$; 否则 $W_u(x, y) = W(x, y)$ 。

根据定义 7, 一个具有不可靠资源 FAS Petri 网是一个 FAS Petri 网和一组恢复子网通过它们在 $\bigcup_{r \in P_R^u} U(r)$ 中的公共库所共享合成。标识 M_{i0} 是其的可接受的初始标识, $M_{i0}(p) = 0, p \in \bigcup_{r \in P_R^u, z \in U(r)} \{p_u(z)\}$; 否则, $M_{i0}(p) = M_0(p)$ 。对 $t \in T, {}^{(r)}t$ 表示输入资源库所集合, $t^{(r)}$ 表示输出资源库所集合, ${}^{(o)}t$ 表示输入活动库所集合, $t^{(o)}$ 表示输出活动库所集合, $t = {}^{(r)}t \cup {}^{(o)}t$ 且 $t = t^{(r)} \cup t^{(o)}$ 。对 $M_u \in R(N_u, M_{i0}), t \in T_0 \cup T_S,$

对于 $\forall p \in {}^{(o)}t$, 若 $M_u(p) \geq W_u(p, t)$, 则 t 在 M_u 下过程使能; 对于 $\forall p \in {}^{(r)}t$, 若 $M_u(p) \geq W_u(p, t)$, 则 t 在 M_u 下资源使能。 $E(M_u) \subseteq T_S$ 表示 M_u 下既过程使能又资源使能的变迁的集合。在 M 下所有活动进程的多集表示为 $A(M)$ 。系统中的进程由 a_p 表示, $\pi(M): A(M) \rightarrow P_S$ 是从活跃进程与其相应活动库所的映射。

假设图2中 $P_R^u = \{r_6\}$ 。显然, $U(r_6) = \{r_6, p_7\}$ 。如图4所示, 图2的FAS的Petri网和它的恢复子网共享合成的具有不可靠资源FAS Petri网。其初始标识为 $M_{u0} = 4p_0 + r_1 + r_2 + r_3 + r_4 + r_5 + r_6 + r_7$ 。

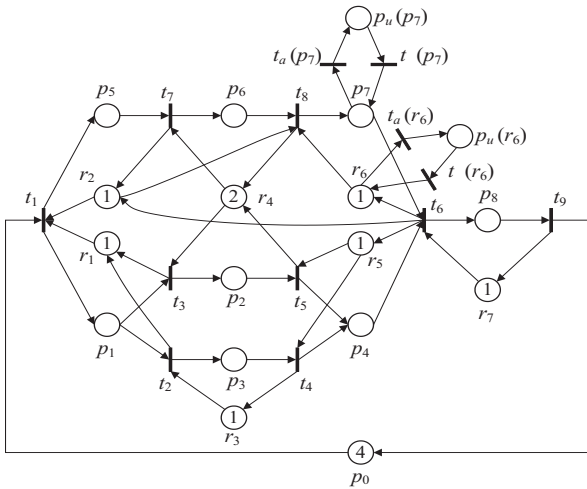


图4 一个具有不可靠资源FAS Petri网

3 系统鲁棒控制策略

本章节的控制目标是在无故障资源的情况下使系统远离死锁, 所有故障资源恢复后, 系统不能处于死锁状态而应恢复正常生产。因此, 该文提出了两个算法, 对系统鲁棒性控制策略进行求解。

3.1 进程可被推进算法

给定一个路径 $\alpha = p_1 t_1 p_2 t_2 \dots p_i t_i p_{i+1}$, $i > 0$, $\tau(\alpha) = t_1 t_2 \dots t_i$, $\varphi(\alpha) = \{p_1, p_2, \dots, p_i\}$ 。对任意 $p \in P_S$, $\mathcal{L}(p)$ 是以库所 p 开始的所有路径的集合, $\mathcal{Q}(p)$ 是以库所 p 结束的所有路径的集合。 $\mathcal{K}(p) = \bigcup_{\alpha \in \mathcal{Q}(p)} \varphi(\alpha)$ 表示活动库所的集合, 其中的每个活动库所都有通往活动场库所 p 的路径。

进程可被推进算法 (Processes Can be Advanced algorithm, PCA) 用于确定在 M_u 标识下活动进程是否可以从 p_a 推进到 p_b ($p_a, p_b \in P_0 \cup P_S$)。首先, 如果 p_a 是一个装配活动库所, 则需要遍历所有的此装配活动的所有分支, 将进程尽可能推进到相应的装配活动库所 (第3~9行), 以便后续装配活动的推进。然后, 找到最长的可发生的路径 $\tau(\alpha)$, 使之发生更新标识, 并且递归地调用 PCA 算法并输入 $((N_u, M_{u0}), M_u, p_l, p_b)$, 其中 p_l 是 α 中最后一个库所 (第11~13行)。

最终判断活动进程是否可以从 p_a 推进到 p_b 。如算法1所示:

算法1: 进程可被推进算法 (PCA)

输入: 具有不可靠资源 FAS 标记 Petri 网 (N_u, M_{u0}) , 标识 M_u , 库所 p_a , 库所 p_b

输出: (True/False, 标识 M_u)

1. If p_a 和 p_b 代表同一个库所

2. Return (True, M_u);

3. If p_a 是一个装配活动库所

4. For $p_c \in {}^{(p_a)}t$ && $p_c \in P_S$

5. For $p \in \mathcal{K}(p_c)$

6. If $M_u(p) > 0$ Then

7. (flag, M'_u) = PCA $((N_u, M_{u0}), M_u, p, p_c)$;

8. If flag == true Then

9. $M_u = M'_u$; break;

10. If 找到最长的 $\alpha \in \mathcal{L}(p_a)$ 并且满足 $M_u[\tau(\alpha)]$

11. Then $M_u[\tau(\alpha)] > M'_u$; $M_u = M'_u$;

12. Return PCA $((N_u, M_{u0}), M_u, p_l, p_b)$; /* p_l 是 α 中最后一个库所 */

13. Return (False, M_u);

3.2 改进鲁棒银行家算法

本节, 开发了一种称为改进鲁棒银行家算法 (Modified Robust Banker's Algorithm, MRBA) 来释放相关资源, 得到 M_u 下阻塞进程多集 $A_B(M_u)$ 。首先, 假设在算法执行期间不会再发生故障和恢复。初始化 $A_B(M_u)$ 为 M_u 下所有的活动进程。其次, 通过算法 PCA 判断进程是否可终止, 将这些进程占有的资源释放以供后续其他进程使用。最后, 所有可终止的进程都已终止, $A_B(M_u)$ 中所有元素均为阻塞进程。具体算法如下:

算法2: 改进鲁棒银行家算法 (MRBA)

输入: 具有不可靠资源 FAS 标记 Petri 网 (N_u, M_{u0}) , 标识 M_u , 当前可用资源向量 AVAILABLE

输出: 阻塞进程 $A_B(M_u)$

1. 初始化 $A_B(M_u)$ 为 M_u 下所有的活动进程;

2. While $A_B(M_u)$ 存在一个活动进程 a 能利用当前可用资源终止 do

3. 在 $A_B(M_u)$ 找到一个活动进程 a , 使算法 PCA $((N_u, M_{u0}), M_u, \pi(M_u)(a), P_0(a))$ 的第一个返回值是 True, 其中 $\pi(M_u)(a)$ 代表 M_u 下进程 a 所在的活动库所;

4. AVAILABLE = AVAILABLE + $\mathfrak{R}(\pi(M_u)(a))$; /* 向量 $\mathfrak{R}(\pi(M_u)(a))$ 表示进程 a 使用的资源 */

5. $A_B(M_u) = A_B(M_u) / a$; /* 从集合 $A_B(M_u)$ 中删除元素 a */

3.3 鲁棒控制策略

鲁棒控制策略定义如下:

定义8: 一个具有不可靠资源 FAS 标记 Petri 网 $(N_u, M_{u0}) = (P_u, T_u, F_u, W_u, M_{u0})$ 的鲁棒控制策略是一个映射 $\rho: R(N_u, M_{u0}) \rightarrow 2^{T_u}$ 。对于任意的 $M_u \in$

$R(N_u, M_{u0}), \rho(M_u) = \{t \in T_u \mid t \text{ 都被允许在标记 } M_u \text{ 和 } \rho \text{ 下发生}\}$ 。 $\rho(M_u)$ 具体被定义为:

$$(1) \forall t \in \bigcup_{r \in P_r^*, z \in U(r)} \{t_\alpha(z), t_\beta(z)\}, t \in \rho(M_u);$$

(2) $\forall t \in T_s$, 当 $t \in E(M_u)$ 且 $M_u[t]M_{u1}$ 时, 可得 $A_B(M_{u1}) = \text{MRBA}((N_u, M_{u0}), M_{u1}), a \in A(M_{u1}), \pi(M_{u1})(a) = t^{(a)}$, 若 $a \notin A_B(M_{u1})$, 则 $t \in \rho(M_u)$ 。

对 $\forall t \in \bigcup_{r \in P_r^*, z \in U(r)} \{t_\alpha(z), t_\beta(z)\}$, 资源的故障和恢复是不可控制的, 因此故障和恢复变迁属于鲁棒控制策略 ρ 。对于 $\forall t \in T_s$, 如果满足以下两个条件, 则 $t \in \rho(M_u)$: (1) $t \in E(M_u)$, 即 t 在 M_u 下既过程使能又资源使能; (2) 对于给定的变迁 t 以及由它发生到达的标识 M_{u1} , 变迁 t 的后集操作库所中的进程不是多重集 $A_B(M_{u1})$ 的元素, $A_B(M_{u1})$ 由算法 MRBA 计算得到。为实现其控制目标, 鲁棒控制策略 ρ 将使用算法 MRBA, 而算法 MRBA 使用算法 PCA。算法 PCA 以递归方式执行。

4 举例说明与实验分析

4.1 举例说明

例 1: 考虑图 4 模型且初始标识 $M_{u0} = 4p_0 + r_1 + r_2 + r_3 + 2r_4 + r_5 + r_6 + r_7$ 。 $E(M_u) = \{t_1\}$ 。 $M_{u0}[t_1]M_{u1} = 3p_0 + p_1 + p_5 + r_3 + 2r_4 + r_5 + r_6 + r_7$, $\text{AVAILABLE} = [0, 0, 1, 2, 1, 1, 1]$ 。 $E(M_{u1}) = \{t_1\}$ 。执行算法 MRBA $((N_u, M_{u0}), M_{u1}, \text{AVAILABLE})$ 。初始化 $A_B(M_u) = \{a_{p_1}, a_{p_2}\}$ 次判断进程 a_{p_1}, a_{p_2} 是否可终止。得到 PCA $((N_u, M_{u0}), M_{u1}, p_1, p_0)$ 的第一个返回值为 True, 则说明 a_{p_1} 可终止。释放其占有的资源 $\text{AVAILABLE} = [0, 0, 1, 2, 1, 1, 1] + [1, 0, 0, 0, 0, 0, 0] = [1, 0, 1, 2, 1, 1, 1]$ 。 $A_B(M_{u1}) = \{a_{p_1}, a_{p_2}\} / a_{p_1} = a_{p_2}$ 。同理可得进程 a_{p_2} 可终止, $A_B(M_u) = \emptyset$ 。 $t_1^{(a)} = \{p_1, p_5\}$, $\pi(M_{u1})(a_{p_1}) = p_1 \notin A_B(M_{u1})$, $\pi(M_{u1})(a_{p_2}) = p_5 \notin A_B(M_{u1})$, 则 $t_1 \in \rho(M_{u0})$ 。 $\{t_\alpha(r_6), t_\alpha(r_7)\} \subseteq \bigcup_{r \in P_r^*, z \in U(r)} \{t_\alpha(z), t_\beta(z)\}$, 综上 $\rho(M_{u0}) = \{t_1, t_\alpha(r_6), t_\alpha(r_7)\}$ 。

例 2: 考虑图 4 模型且标识 $M_{u2} = 2p_0 + p_2 + p_4 + 2p_6 + r_1 + r_2 + r_3 + r_6 + r_7$, 此时空闲的 r_6 资源故障, $M_{u2}[t_\alpha(r_6)]M_{u3} = 2p_0 + p_2 + p_4 + 2p_6 + r_1 + r_2 + r_3 + p_u(r_6) + r_7$ 。 $M_{u3}[t_1]M_{u4} = p_0 + p_1 + p_5 + p_2 + p_4 + 2p_6 + r_3 + p_u(r_6) + r_7$ 。 $\text{AVAILABLE} = [0, 0, 1, 0, 0, 0, 1]$ 。 $E(M_{u4}) = \{t_1\}$ 。执行算法 MRBA $((N_u, M_{u0}), M_{u4}, \text{AVAILABLE})$, 得到 $A_B(M_{u4}) = \{a_{p_1}, a_{p_2}, a_{p_4}, a_{p_5}, 2a_{p_6}\}$ 。 $t_1^{(a)} = \{p_1, p_5\}$, $\pi(M_{u4})(a_{p_1}) = p_1 \in A_B(M_{u4})$, $\pi(M_{u4})(a_{p_5}) = p_5 \in A_B(M_{u4})$, 则 $t_1 \notin \rho(M_{u0})$ 。 $\{t_\beta(r_6), t_\alpha(r_7)\} \subseteq \bigcup_{r \in P_r^*, z \in U(r)} \{t_\alpha(z), t_\beta(z)\}$, 综上 $\rho(M_{u0}) = \{t_\beta(r_6), t_\alpha(r_7)\}$ 。

4.2 分析对比

图 5 是文献[14]给出的一个经典 FAS 模型, 初始标识 $M_{u0} = 8p_1 + 8p_{19} + r_1 + 2r_2 + r_3 + 2r_4 + r_5 + 2r_6 + r_7 + 2r_8 + 2r_9 + 2r_{10}$ 。将所提出的方案与文献[13-15]中的方案进行了比较。为了简单起见, 用 L_{19}, H_{15} 和 W_{08} 分别表示它们。由于文献[13-15]中的方案不可处理拥有不可靠资源的系统, 所以下实验假设系统无不可靠资源。实验结果如图 6 所示。 H_{15} 可达标记的数量为 66 个, W_{08} 可达标记的数量为 444 个, L_{19} 可达标记的数量为 5 098 个, 文中方案可到达标记的数量为 30 710 个。文中方案在不拥有不可靠资源的前提下得到的可达标记的数量最多, 并且可以处理不可靠资源下的 FAS 死锁问题。

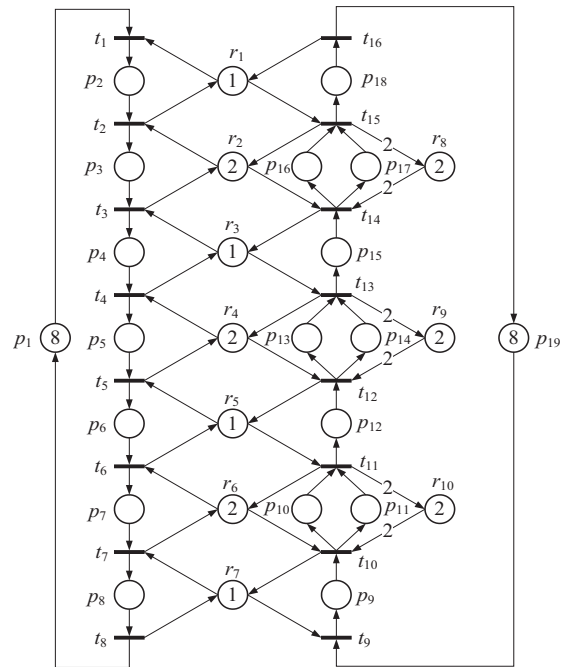


图 5 文献[14]的 FAS 模型

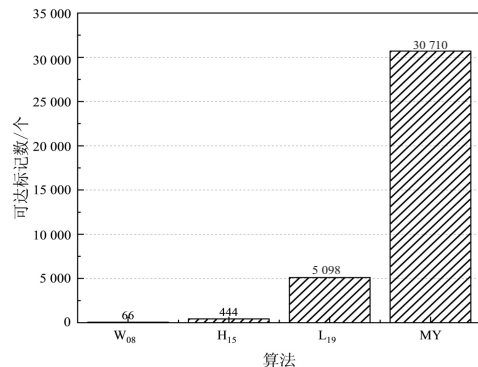


图 6 可达标记数分析图

此外, 与自动制造系统的鲁棒死锁工作^[16-17]进行了系统和方法的对比。从研究系统角度分析, 文献[16]研究资源共享的顺序系统, 文献[17]考虑的是柔性制造系统, 而文中研究的系统是具有装配操作的 FAS, 应用范围更广。从研究方法角度分析, 文献[16]

提出了鲁棒性增强算法和基于关键区域的稳健性算法进行系统死锁避免,文献[17]建立了一种鲁棒控制器来预防系统出现死锁,而文中采用改进的银行家算法提出了一种在线的鲁棒死锁避免机制。这种机制无需对系统进行事先优化或处理,避免系统出现死锁情况,保证了在资源发生故障时以及资源故障恢复后系统正常生产。

5 结束语

该文讨论了具有不可靠资源的柔性装配系统(Flexible Assembly System, FAS)在生产过程中可能发生的死锁和堵塞现象,并创新性地提出了适用于具有不可靠资源和装配操作FAS的改进银行家算法。利用Petri网为不可靠资源和系统流程建模,并结合提出的进程可被推进算法和改进的银行家算法为系统建立了一种新的鲁棒控制策略,保证系统在资源发生故障时以及资源故障恢复后,系统能正常生产不会出现阻塞的情况。实验结果表明,该策略计算得到的系统的可达状态显著提高,证实了算法的有效性。

此外,基于银行家算法的基本逻辑,所提出的策略在一般情况下是次优的,提高策略的行为允许性是未来进一步的研究方向之一。

参考文献:

- [1] FENG Y, XING K, ZHOU M, et al. Robust deadlock prevention for automated manufacturing systems with unreliable resources by using general Petri nets[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, 50(10): 3515–3527.
- [2] LIU G, LI P, LI Z, et al. Robust deadlock control for automated manufacturing systems with unreliable resources based on Petri net reachability graphs[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019, 49(7): 1371–1385.
- [3] LIU G, ZHANG L, CHANG L, et al. Robust deadlock control for automated manufacturing systems based on elementary siphon theory[J]. Information Sciences, 2020, 510: 165–182.
- [4] 关旋旋. 基于Petri网的死锁避免方法研究[J]. 自动化应用, 2022(6): 4–7.
- [5] 刘慧霞, 吴维敏, 苏宏业, 等. 信标基底的可选择性在制造系统死锁控制中的应用[J]. 控制理论与应用, 2016, 33(10): 1345–1351.
- [6] LUO J, LIU Z, WANG S, et al. Robust deadlock avoidance policy for automated manufacturing system with multiple unreliable resources [J]. IEEE/CAA Journal of Automatica Sinica, 2020, 7(3): 812–821.
- [7] REVELIOTIS S, FEI Z. Robust deadlock avoidance for sequential resource allocation systems with resource outages [J]. IEEE Transactions on Automation Science and Engineering, 2017, 14(4): 1695–1711.
- [8] YUE H, XING K, HU H, et al. Petri-net-based robust supervisory control of automated manufacturing systems[J]. Control Engineering Practice, 2016, 54: 176–189.
- [9] LU Y, CHEN Y F, LI Z W, et al. An efficient method of deadlock detection and recovery for flexible manufacturing systems by resource flow graphs[J]. IEEE Transactions on Automation Science and Engineering, 2021, 19(3): 1707–1718.
- [10] DU N, HU H, ZHOU M. A survey on robust deadlock control policies for automated manufacturing systems with unreliable resources[J]. IEEE Transactions on Automation Science and Engineering, 2019, 17(1): 389–406.
- [11] DU N, HU H, ZHOU M. Robust deadlock avoidance and control of automated manufacturing systems with assembly operations using Petri nets[J]. IEEE Transactions on Automation Science and Engineering, 2020, 17(4): 1961–1975.
- [12] DU N, HU H S. Robust deadlock avoidance control for AMSs with assembly operations embedded in flexible routes using Petri nets [J]. IET Control Theory & Applications, 2019, 13(11): 1579–1590.
- [13] LUO J, LIU Z, ZHOU M. A Petri net based deadlock avoidance policy for flexible manufacturing systems with assembly operations and multiple resource acquisition[J]. IEEE Transactions on Industrial Informatics, 2018, 15(6): 3379–3387.
- [14] HU H, ZHOU M. A Petri net-based discrete-event control of automated manufacturing systems with assembly operations [J]. IEEE Trans on Control Systems Technology, 2015, 23(2): 513–524.
- [15] WU N, ZHOU M, LI Z. Resource-oriented Petri net for deadlock avoidance in flexible assembly systems [J]. IEEE Trans on Systems Man and Cybernetics Part a-Systems and Humans, 2008, 38(1): 56–69.
- [16] 王小俊. 自动制造系统的稳健性监督控制策略研究[D]. 西安: 西安电子科技大学, 2021.
- [17] 刘慧霞, 李俊红, 王红梅, 等. 具有不可靠资源柔性制造系统的鲁棒控制器设计[J]. 控制与决策, 2022, 37(8): 2040–2048.