

自动制造系统的稳健死锁避免策略

刘 伟, 卢雪芹

(山东科技大学 计算机科学与工程学院, 山东 青岛 266590)

摘要: 自动制造系统(AMSs)使用有限资源并行生产多种零件, 零件加工对共享资源的使用可能造成循环等待, 导致系统死锁。针对使用多种不可靠资源的一类 AMSs, 本研究扩展面向资源有色 Petri 网提出特殊资源标记图建模分析。为解决死锁和阻塞问题, 在每条路径设置本地管理器, 提出稳健死锁避免策略。该策略通过在线判断托肯能否触发所在路径的最近关键变迁和最近不可靠变迁, 得到不同状态下的稳健活性变迁。在此基础上, 提出判断路径使系统达到更多安全状态。本研究的稳健死锁避免策略具有多项式时间复杂度, 能够保证资源故障时不会阻塞不使用故障资源的零件持续生产, 从而实现系统的稳健死锁避免。

关键词: 自动制造系统; 死锁避免; 面向资源有色 Petri 网; 资源故障; 稳健死锁避免

中图分类号: TP311

文献标志码: A

Robust deadlock avoidance strategy for automated manufacturing systems

LIU Wei, LU Xueqin

(College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China)

Abstract: Automatic manufacturing systems (AMSs) use limited resources to produce multiple parts in parallel, and the use of shared resources by part processing may cause resource circular wait and system deadlock. For a class of AMSs using a variety of unreliable resources, this paper extends the modeling analysis of the special resource marked graphs for colored resource-oriented Petri nets. To solve the deadlock and blocking problems, this paper proposes a robust deadlock avoidance policy by setting up a local controller in each path. This strategy, by determining whether token can fire the nearest critical transition and the nearest unreliable transition of the path online, obtains the robust live transitions in different states. On this basis, the judgment path is proposed to make the system reach more safe states. With polynomial time complexity, the robust deadlock avoidance strategy can ensure that resource failures do not block the production of parts that use no faulty resources, thus realizing robust deadlock avoidance.

Key words: automated manufacturing systems; deadlock avoidance; colored resource-oriented Petri nets; resource failures; robust deadlock avoidance

为提高零件生产效率, 自动制造系统(automated manufacturing systems, AMSs)使用有限资源并行生产零件^[1-2]。然而, 零件使用共享资源时可能导致系统死锁, 并且资源故障可能导致系统阻塞, 使不需要故障资源的零件也无法继续加工^[3-4]。因此需设计稳健死锁控制策略, 阻止系统死锁和阻塞。本研究用“不可靠资源”表示可能故障的资源, 用“可靠资源”表示不会故障的资源。

AMSs 的死锁解决策略有死锁预防、死锁避免、死锁检测与恢复三种, 本研究采用死锁避免。对具有

收稿日期: 2024-05-27

基金项目: 教育部人文社科规划基金项目(23YJAZH084); 济宁市重点研发计划项目(2024NYNS013); 济宁市重点研发计划(软科学)项目(24JNZC001)

作者简介: 刘 伟(1977—), 男, 山东省泰安人, 教授, 博士生导师, 主要从事过程挖掘、Petri 网理论与应用等方面研究。

E-mail: liuwei_doctor@yeah.net

卢雪芹(1997—), 女, 山东省济宁人, 硕士研究生, 主要从事自动制造系统、Petri 网理论等研究。

装配操作和灵活路径的 AMSs, 研究人员使用 Petri 网(Petri nets, PNs)建模分析。如文献[5]使用 PNs 为一类 AMSs 设计稳健无死锁监控器避免死锁; 文献[6-9]扩展文献[5]中的关键库所, 放宽单一不可靠资源的限制, 解决具有多种不可靠资源的 AMSs 死锁问题, 文献[10]结合关键库所和不可靠资源库所, 基于资源容量约束策略, 解决同时需要可靠和不可靠资源的 AMSs 死锁和阻塞问题。上述方式基于面向进程 Petri 网(process-oriented Petri nets, POPNs)建模, 能清晰地描述零件的生产过程, 但其结构较为复杂。为了简化系统结构, 研究者提出面向资源有色 Petri 网(colored resource-oriented Petri nets, CROPNs)。如文献[11]根据托肯是否离开生产进程回路(production process circuit, PPC)对托肯分类, 通过控制变迁的触发顺序提出动态资源分配策略; 文献[12]用饱和 PPC 描述死锁, 基于在线监管器前瞻预测托肯未来的前进, 通过限制资源分配避免死锁; 文献[13]使用 ILSs(intrinsically live structures), 借助回路中有向弧的权值合理分配资源; 文献[14]研究了不良状态与强连通子网的关系, 为强连通子网添加控制器阻止死锁。

尽管现有死锁避免策略复杂度较低, 但这些策略拒绝了一些可达安全状态。为此, 本研究提出特殊资源标记图(special resource marked graphs, SRMGs), 利用其结构特征, 设计新的稳健死锁避免策略, 允许系统达到更多安全状态, 实现稳健死锁避免。本研究的主要贡献有: 提出三种特殊结构, 表示具有装配操作、灵活路径和多资源获取的一类 AMSs; 借助前瞻思想和一些特殊变迁, 提出一种稳健死锁避免策略, 能确定不同状态下的稳健活性变迁, 确保即使资源故障, 系统的剩余可用资源空间仍能满足其他零件的加工需求; 提出判断路径, 使得系统允许更多可达安全状态发生, 提高系统许可性。

1 CROPNs 建模

本节首先介绍 CROPNs 的相关概念, 然后提出具有三种特殊结构的新模型。

1.1 CROPNs 的相关概念

定义 1^[15] 满足下列条件的七元组 $N=(P, T, C, I, O, M, K)$ 称为一个 CROPN:

- 1) $P=P_0 \cup P_R, P_R=\{p_1, p_2, \dots, p_m\}$;
- 2) $T=\{t_1, t_2, \dots, t_n\}, P \cap T=\emptyset$ 且 $P \cup T \neq \emptyset$;
- 3) $C=C(t_u) \cup C(p)$;
- 4) $I:P \times T \rightarrow \mathbf{N}$;
- 5) $O:P \times T \rightarrow \mathbf{N}$;
- 6) $M:P \rightarrow \mathbf{N}$;
- 7) $K:P \rightarrow \mathbf{N}^+ = \mathbf{N} \setminus \{0\}, K(P_0)=\infty$ 。

定义 1 中, P 和 T 是两个不相交的集合, 其中 P 为库所集, P_0 为开始库所, P_R 为资源库所集, m 为资源库所数量; T 为变迁集, n 为变迁数量; $C(t_u)$ 表示 t_u 的颜色, $C(p)$ 表示 p 中托肯的颜色, 其中 $p \in P$; I 为输入函数; O 为输出函数; M 为系统当前状态, M_0 是初始状态; K 为库所容量。

定义 2^[15] 设 $N=(P, T, C, I, O, M, K)$ 为一个 CROPN, 其中 $\cdot p = \{t:t \in T, O(p, t) > 0\}$, $p \cdot = \{t:t \in T, I(p, t) > 0\}$, $\cdot t = \{p:p \in P, I(p, t) > 0\}$, $t \cdot = \{p:p \in P, O(p, t) > 0\}$, 称 $\cdot p$ 为 p 的输入变迁, $p \cdot$ 为 p 的输出变迁, $\cdot t$ 为 t 的输入库所, $t \cdot$ 为 t 的输出库所。

定义 3^[15] 设 $N=(P, T, C, I, O, M, K)$ 为一个 CROPN, N 满足变迁触发规则:

- 1) 对于 $t_u \in T$, 若满足式(1)和式(2), 称 t_u 为使能变迁, 记为 $M[t_u]$ 。

$$M(p_l, C(t_u)) \geq I(p_l, t_u), \tag{1}$$

$$K(p_l) \geq M(p_l) - I(p_l, t_u)(C(t_u)) + O(p_l, t_u)(C(t_u)). \tag{2}$$

- 2) 若 $M[t_u]M'$, 则对 $p_l \in P$,

$$M'(p_l) = M(p_l) - I(p_l, t_u)(C(t_u)) + O(p_l, t_u)(C(t_u)). \tag{3}$$

在定义 3 中, $M[t_u]M'$ 表示状态 M 下, 变迁 t_u 触发后, 产生状态 M' 。 $M(p_l)$ 表示可达状态 M 下库所 p_l 中的托肯数。将 t_u 的输入库所中 $C(t_u)$ 颜色的托肯称为 t_u 的相关托肯, $M(p_l, C(t_u))$ 表示 M 处 p_l 中 t_u 的相关托肯数。

1.2 SRMGs 模型

本节扩展现有 CROPNs 模型,提出特殊资源标记图(special resource marked graphs,SRMGs)。在 SRMGs 中,同一种零件加工过程中的所有变迁使用同一种颜色,区分不同零件的加工过程。

定义 4 设 $N=(X, P, T, C, I, O, M, K)$ 为一个 SRMG,与普通 CROPNs 相比,SRMGs 满足:

- 1) $X = \cup X_i, i \in \mathbf{N}_s^+ = \mathbf{N}_s \setminus \{0\} = \{1, 2, \dots, s\}$;
- 2) $P = P_0 \cup P_R, P_R = P_{Rr} \cup P_{Ru} = P_{Rs} \cup P_{Rus}, \forall p_a, p_b \in P, |C(p_a \cdot) \cap c_i| \geq 1, |C(\cdot p_b) \cap c_i| \geq 1$;
- 3) $T = \{t_1, t_2, \dots, t_n\}$ 。 $\forall t_c, t_d \in T, |t_c \cdot| \geq 1, |\cdot t_d| \geq 1$;
- 4) $C = C(t_u) \cup C(p_i), C(t_u) = C(X_i) = c_i$ 。

在 SRMGs 模型中, X 为零件类型集; P_{Rr} 为可靠资源库所集, P_{Ru} 为不可靠资源库所集, P_{Rs} 为共享资源库所集, P_{Rus} 为非共享资源库所集。 c_i 表示颜色,由于 $|t_c \cdot| \geq 1, |\cdot t_d| \geq 1$,当 $|t_c \cdot| > 1$ 且 $|\cdot t_d| > 1$ 时,若 $t_c \cdot \neq \cdot t_d$,该结构为装配操作;若 $t_c \cdot = \cdot t_d$,该结构表示多类型资源获取阶段。

定义 5 满足以下条件的 $N_i = \{X_i, P_i \cup P_0, T_i, C_i, I_i, O_i, M_i, K_i\}$ 称作 N 的一个子网:

- 1) $X_i \subseteq X, X_i^j \subseteq X_i$;
- 2) $P_i \subseteq P, P_{ij} \subseteq P_i, P_{ikl} \subseteq P_i$;
- 3) $T_i \subseteq T$;
- 4) $C_i \subseteq C$;
- 5) $I_i \subseteq (P_i \times T_i) \cup (T_i \times P_i) \cap I$;
- 6) $O_i \subseteq (P_i \times T_i) \cup (T_i \times P_i) \cap O$ 。

对于 N 的一个子网, X_i 为第 i 种零件, X_i^j 为 X_i 的第 j 种子零件。 P_i 为 X_i 的资源库所集。 P_{ij} 为 X_i 的第 j 个操作所需资源, P_{ikl} 为 X_i 的第 k 个灵活子路径中第 l 个操作所需资源。 P_{it}^k 为 X_i 的第 k 条装配子路径中第 l 个操作所需资源。 T_i 为 X_i 的变迁集, T_i^j 为 X_i^j 的变迁集。

本研究提出的 SRMGs 模型具有 3 种特殊结构:资源标记图(resource marked graph,RMG)块,装配标记图(assemble marked graph,AMG)块和路径选择图(path choose graph,PCG)块。

定义 6 如果一个结构中的变迁和库所满足:① $\cdot t_b = \emptyset, |t_b \cdot| \geq 2$;② $t_e \cdot = \emptyset, |\cdot t_e| \geq 2$;③ $t_b \cdot = \cdot t_e$,则称该结构为资源标记图块,即 RMG 块。

RMG 块表示多类型资源获取阶段,记为 $R(t_b, t_e)$ 。 T_b 表示多类型资源获取变迁集, $T_b = \cup t_b, t_b$ 表示多类型资源获取变迁, t_e 表示多类型资源释放变迁,如图 1 所示。

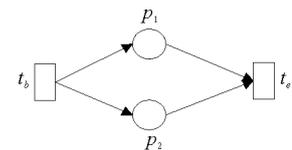


图 1 一个 RMG 块
Fig. 1 A RMG block

定义 7 如果一个结构中的变迁和库所满足:① $\cdot t_d = \emptyset, |t_d \cdot| \geq 2$;
② $t_a \cdot = \emptyset, |\cdot t_a| \geq 2$;③ 不包含回路,则称该结构为装配标记图块,即 AMG 块。

AMG 块表示一个装配操作,记为 $A(t_d, t_a)$, t_a 为装配变迁, t_d 为拆卸变迁, $\cdot t_a$ 为装配库所,如图 2 所示。

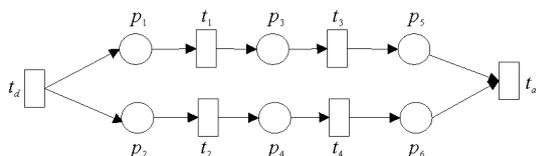


图 2 一个 AMG 块
Fig. 2 An AMG block

定义 8 如果一个结构中的变迁和库所满足:① $\cdot p_c = \emptyset, |p_c \cdot| \geq 2$;② $p_u \cdot = \emptyset, |\cdot p_u| \geq 2$;
③ $|C(p_c \cdot) \cap c_i| > 1, |C(\cdot p_u) \cap c_i| > 1$;④ 不包含回路,则称该结构为路径选择图块,即 PCG 块。

PCG 块表示零件的灵活路径选择,记为 $P(p_c, p_u)$,如图 3 所示。

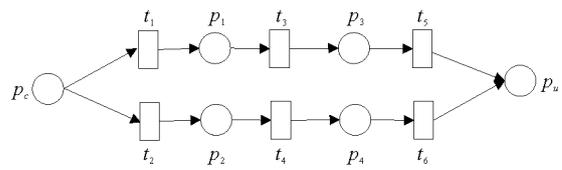


图 3 一个 PCG 块
Fig. 3 A PCG block

系统中的可达状态分为安全和不安全状态。无死锁不安全状态下任意使能变迁触发都会到达死锁。死锁、无死锁不安全和阻塞状态统称为不安全状态。称触发后不会到达不安全状态的使能变迁为稳健活性变迁,用 T_{RL} 表示 M 处的稳健活性变迁集。

2 死锁控制

本节基于分布式思想,为具有多个不可靠资源库所的 SRMGs 提出稳健死锁避免策略。

2.1 关键变迁

首先引入局部关键变迁集和全局关键变迁,根据其划分零件的加工路径,使系统仅通过检查部分加工路径便可以避免不安全状态。

定义 9 对于一个 SRMG,满足以下条件的变迁称为 X_i^k 的子关键变迁 T_{Si}^k :

- 1) $T_{Si1}^k = \{t : P_{il}^k \in P_{Rus}\}$;
- 2) $T_{Si2}^k = \{t : \forall z \geq l, P_{il}^k \geq P_{ik}^k\}$;
- 3) $T_{Si3}^k = \{t : l = e_{ik}\}$ 。

$T_{Si}^k = \{T_{Si1}^k, T_{Si2}^k, T_{Si3}^k\}; T_{Si} = \cup T_{Si}^k, T_S = \cup T_{Si}$ 。其中, $t = \cdot P_{il}^k \cap T_{Si}^k, e_{ik}$ 表示 X_i^k 加工操作的数量。由于装配变迁 t_a 的触发需要其装配库所中同时都有 c_i 颜色的托肯。下面引入局部关键变迁集,其由每条装配子路径中的子关键变迁组成。

定义 10 对于一个 SRMG,满足以下条件的变迁称为 X_i 的局部关键变迁集 T_{Li} :

- 1) $T_{Li1} = \{\{T_1, T_2, \dots, T_{z_i}\} | (T_1 \in T_i^1) \wedge \dots \wedge (T_{z_i} \in T_i^{z_i}) \wedge \{T_t \in T_{Si1}^k \wedge T_u \in T_{Si}^k\}\}$;
- 2) $T_{Li2} = \{\{T_1, T_2, \dots, T_{z_i}\} | (T_1 \in T_i^1) \wedge \dots \wedge (T_{z_i} \in T_i^{z_i}) \wedge \{T_t \in T_{Si2}^k \wedge T_u \in T_{Si}^k\}\}$;
- 3) $T_{Li3} = \{\{T_1, T_2, \dots, T_{z_i}\} | (T_1 \in T_i^1) \wedge \dots \wedge (T_{z_i} \in T_i^{z_i}) \wedge \{T_t \in T_{Si3}^k \wedge T_u \in T_{Si}^k\}\}$ 。

$T_{Li} = \{T_{Li1}, T_{Li2}, T_{Li3}\}, P_{Li} = T_{Li} \cdot \cap P_i$ 。 $T_L = \cup T_{Li}$ 。其中, z_i 表示 X_i 中装配子路径的数量, T_t 表示当前需要判断的使能变迁所在子路径的局部关键变迁, T_u 表示其余装配子路径中的局部关键变迁。

定义 11 对于一个 SRMG,满足以下条件的变迁称为 X_i 的全局关键变迁 T_{Gi} :

- 1) $T_{Gi1} = \{t : p_g \in P_{Rus} \cup P_0\}$;
- 2) $T_{Gi2} = \{t : p_g \geq p_{iz}, p_{iz} \in p_g^{(pi)}\}$ 。

$T_{Gi} = \{T_{Gi1}, T_{Gi2}\}, T_G = \cup T_{Gi}$ 。其中, $t = \cdot p_g \cap T_i, \forall p_g \in P_G, P_G = P_{ij} \cup P_{ikl} \cup P_{Li}$ 。 $p_g^{(pi)}$ 表示 p_i 的后续库所集,代表 X_i 的剩余阶段使用的资源库所。

2.2 不可靠变迁

为防止资源故障造成阻塞,考虑触发后占用不可靠资源空间的变迁。对于 AMG 块中的托肯,资源故障可能导致系统无法完成装配操作。因此需要考虑 AMG 块中一些触发后不占用不可靠资源空间的变迁。

定义 12 对于一个 SRMG,满足以下条件的变迁称为 X_i 的不可靠变迁 T_{Ui} :

- 1) $T_{Ui1} = \{t : t = \cdot p_t \cap T_i \cap T_b, p_t \cap P_{Ru} \neq \emptyset, p_t \cap P_{Rr} \cap P_{Rs} \neq \emptyset\}$;
- 2) $T_{Ui2} = \{t : t = (\cdot p_t \cap T_i) \setminus T_{Ui1}, p_t \cap P_{Ru} \neq \emptyset\}$;
- 3) $T_{Ui3} = \{t : t = \cdot P_{ie_i}^k \cap T_i^k, \exists p_u \in P_{Ru} \cap P_i^k, k \in \mathbf{N}_{z_i}^+\}$ 。

$T_{Ui} = \{T_{Ui1}, T_{Ui2}, T_{Ui3}\}, T_U = \cup T_{Ui}, P_{U1} = T_{U1} \cdot = \cup T_{Ui1} \cdot$, 其中 $\forall p_t \in P_T, P_T = P_{ij} \cup P_{ikl} \cup P_{il}^k$ 。

若托肯不在 AMS 块中,且其最近不可靠变迁位于 AMG 块,需要在每条装配子路径中都找到一个距离该托肯最近的不可靠变迁,作为该托肯的最近不可靠变迁集。

定义 13 对于一个 SRMG,对于 $p_k \in t_\omega \cdot \cap P_{Rr} \cap P_{Rs}$,称 p_k 为危险库所。其中, $t_\omega \in T_{Ui1}$ 表示一个危险变迁,触发 t_ω 后同时占据可靠和不可靠资源。假设触发 t_ω 后,托肯占据的不可靠资源故障,需要该可靠资源空间的零件加工可能受到影响,导致系统阻塞。

定义 14 对于一个 SRMG, $\forall p \in P_R, R_e(M(p)) = K(p) - M(p), R_a(M(p)) = R_e(M(p)) - 1$ 。其中, $R_e(M(p))$ 表示库所 p 的剩余资源空间, $R_a(M(p))$ 表示库所 p 的剩余可用资源空间,即触发 p_k 的

输出变迁后的剩余资源空间。因此对于危险库所 p_k , 只要 $R_a(M(p_k)) > 0$, 无论是否发生资源故障, 触发 t_w 都不会影响其他零件的加工。

2.3 稳健死锁避免策略

本节提出稳健死锁避免策略, 借助关键和不可靠变迁, 判断使能变迁是否为稳健活性变迁, 托肯触发稳健活性变迁能保证零件无死锁的进行下一步加工。

定义 15 设 α 为 SRMG 的一条加工路径, 称满足以下条件的路径 α_s 为 α 的子集:

- 1) $\tau(\alpha_s) = \tau(\alpha(t_s, t_s'))$;
- 2) $\epsilon(\alpha_s) = \epsilon(\alpha(p_s, p_s'))$ 。

其中, $\tau(\alpha)$ 为 α 的变迁序列, $\epsilon(\alpha)$ 为 α 的库所序列。 $\tau(\alpha(t, t'))$ 表示初始变迁是 t 、终止变迁是 t' 的 α 的变迁序列; $\epsilon(\alpha(p, p'))$ 表示初始库所是 p 、终止库所是 p' 的 α 的库所序列。

算法 1 用来判断 AMG 块中的使能变迁是否为稳健活性变迁。将使能变迁 t_{enh} 的输入库所 p_l 中与 t_{enh} 相关的托肯称为活动托肯。如果 t_{enh} 有多个输入库所, 将这些库所中与 t_{enh} 相关的托肯看作一个整体, 统称为活动托肯, 用 l_o 表示。 T_i^u 表示使能变迁所在装配子路径中的所有变迁集, T_i^u 表示其余装配子路径中的变迁集, $u \in \mathbf{N}_{z_i}^+ \setminus t$ 。对于 P_i^u 的装配库所中 c_i 颜色的托肯, 将其优先视为相关托肯。 l_{or} 表示 l_o 的相关托肯。 T_{Gin} 表示 l_o 的最近全局关键变迁, T_{Lin} 表示活动托肯的最近局部关键变迁集, $T_{\text{en}} = \{T_{\text{Gin}} \cup T_{\text{Lin}}\}$ 表示 l_o 的最近关键变迁。 T_{Un} 表示 l_o 的最近不可靠变迁。当判断 t_{enh} 时, 首先找到其对应的 l_o , 确定 l_o 是否在 P_{Ru} 中。如果不在, 进一步判断其剩余路径中是否存在不可靠变迁。如果没有, 仅需判断 l_{or} 和 l_o 能否触发最近关键变迁。否则, 还需要判断 l_o 能否触发最近不可靠变迁。对于 t_w , 由于资源故障时, 其输入资源库所中的托肯无法正常前进, 只有当 $R_a(M(t_w')) > 0$ 时, 才允许 l_o 前进, 即认为 t_{enh} 是稳健活性的。此外, 如果 l_o 在 P_{Ru} 中, 当 P_{Ru} 没有故障时, 与上述判断相同。否则, l_o 无法前进一步, t_{enh} 不是稳健活性的。

算法 2 用来计算 M 处的稳健活性变迁。对于某个使能变迁, 算法 2 首先判断其 l_o 在 AMG 块还是 PCG 块中。对于 AMG 块中的 l_o , 直接调用算法 1 判断; 对于 PCG 块中的 l_o , 活动托肯首先选择一条灵活子路径, 然后确定该子路径中的最近关键变迁和最近不可靠变迁, 其最多需检查所有的 y_i 条灵活子路径。

定理 1 设 N 为一个 SRMG, 对于安全状态 $M \in R(N, M_0)$, 算法 2 所求 $t_v \in T_{\text{RL}}, M[t_v] > M', M'$ 是安全状态。

证明: t_v 对应的活动托肯有两种可能: ①活动托肯不在不可靠资源库所中; ②活动托肯占据的不可靠资源库所没有故障。这两种情况下 t_v 的判别情况一致: 剩余路径中不存在不可靠变迁, 此时剩余资源空间足够托肯触发最近关键变迁, 因此触发 t_v 不会导致死锁。剩余路径中存在不可靠变迁, 当剩余资源空间足够托肯触发最近关键变迁和最近不可靠变迁, 以及 $R_a(M(t_w')) > 0$ 时, 才允许触发 t_v ; 当没有资源

算法 1 判断 AMG 块中的使能变迁是否为稳健活性变迁

输入: 可达安全状态 $M \in R(N, M_0)$, 使能变迁 $t_{\text{enh}} \in T_{\text{EN}}$;

输出: F ;

- 1) 初始化: $F = \text{false}$;
- 2) **if** l_o 不在 P_{Ru} 中
- 3) **if** l_o 的剩余路径中没有 T_U
- 4) **if** l_o 和 l_{or} 的剩余路径中存在 T_{L_i}
- 5) **if** l_o 和 l_{or} 能触发 T_{Lin}
- 6) $F = \text{true}$;
- 7) **else**
- 8) **if** l_o 能触发 T_{Gin}
- 9) $F = \text{true}$;
- 10) **else**
- 11) **if** l_o 和 l_{or} 的剩余路径中存在 T_{L_i}
- 12) **if** l_o 和 l_{or} 能触发 T_{Lin} , 且 l_o 能触发 T_{Un}
- 13) **if** l_o 的剩余路径中没有 t_w
- 14) $F = \text{true}$;
- 15) **if** l_o 的剩余路径中有 t_w
- 16) **if** $R_a(M(t_w')) > 0$
- 17) $F = \text{true}$;
- 18) **else**
- 19) **if** l_o 和 l_{or} 能触发 T_{Gin} , 且 l_o 能触发 T_{Un}
- 20) 执行步骤 13)~17);
- 21) **if** l_o 占据的不可靠资源没有故障
- 22) 执行步骤 3)~20)。

故障时,系统不会死锁。此外,即使资源故障,危险库所也留下了足够的资源空间避免系统阻塞。因此 t_v 的触发不会导致系统到达不安全状态。

3 系统可达安全状态优化

对于一些不能触发最近关键变迁或最近不可靠变迁的活动托肯,其相关变迁的触发可能不会导致系统死锁。因此,下面提出可执行路径优化所提稳健死锁避免策略,使系统可以到达更多可达安全状态。

定义 16 设 α_1 和 α_2 为 SRMG 的加工路径, $\varepsilon(\alpha_1(p_x, p_y)), \varepsilon(\alpha_2(p_y, p_x))$, 对于

- 1) $\nexists p_e \in \alpha_1 \cup \alpha_2, p_e \in P_R \setminus (p_x \cup p_y)$;
- 2) 有且仅有一个 $p_z \in P_R \setminus (p_x \cup p_y), p_z \in \alpha_1$ 且 $p_z \in \alpha_2, \nexists p_e \in \alpha_1 \cup \alpha_2, p_e \in P_R \setminus (p_x \cup p_y \cup p_z)$ 。

称满足条件 1) 或条件 2) 的 α_1 和 α_2 互为资源循环等待路径, 否则, 称 α_1 和 α_2 互为资源等待路径。

定义 17 设 α_p 和 α_c 为 SRMG 的加工路径, 在标记 M :

- 1) $\varepsilon(\alpha_p) = \varepsilon(\alpha_p(\cdot t_{enh}, \{T_{cn}, T_{Un}\}))$, $\tau(\alpha_p) = \tau(\alpha_p(t_{enh}, \{T_{cn}, T_{Un}\}))$, α_p 为 t_{enh} 的判断路径;
- 2) $\varepsilon(\alpha_c) = \varepsilon(\alpha_c(p_l, p_x))$, $\tau(\alpha_c) = \tau(\alpha_c(t_l, t_x))$, $\forall p \in \varepsilon(\alpha_c) \setminus (p_l \cup p_x), R_e(M(p)) > 0, M(p_l)$ 中有 $C(t_l)$ 颜色的托肯, $p_l \in \varepsilon(\alpha_p) \cap \varepsilon(\alpha_c), R_e(M(p_l)) = 0$, 且 $p_x \in \varepsilon(\alpha_p(\cdot t_{enh}, p_l)) \cap \varepsilon(\alpha_c), \alpha_c$ 为 α_p 中 p_l 的可执行路径。
- 3) $\varepsilon(\alpha_{pc}) = \varepsilon(\alpha_p(p_x, p_l))$ 。

用 α_c 表示 α_c 的集合, α_{pc} 表示 α_p 的一个子集。算法 3 计算 M 处优化的稳健活性变迁。对于使能变迁 t_{enh} , 首先调用算法 2, 找到能触发最近关键变迁和最近不可靠变迁的活动托肯, 其对应的使能变迁是稳健活性变迁。否则, 通过 t_{enh} 的判断路径 α_p , 找到 α_p 中剩余资源空间为 0 的所有 p_l 。如果 p_l 具有可执行路径 α_c , 分情况讨论。对每个 p_l 分别进行判断, 只有都满足条件时, t_{enh} 才为活性变迁。否则, 暂时不能触发 t_{enh} , 以避免死锁。

定理 2 设 N 为一个 SRMG, 对于可达安全状态 M , 算法 3 所求 $t_v \in T_{RL}, M[t_v \rangle M', M'$ 是安全状态。

证明: 如果活动托肯能触发其最近关键变迁和最近不可靠变迁, 根据定理 1 可知, M' 是安全状态。否则, 分两种情况分析。

- 1) α_p 中没有 t_ω 。此时 α_p 中至少存在一个 $p_l, R_e(M(p_l)) = 0$ 。只有当 $R_e(M(p_x)) \neq 0$ 时, p_x 与 p_l 才不会出现资源的循环等待。只有当 $R_e(M(p_x)) \neq 0$ 或 $R_e(M(p_z)) > 1$ 时, p_x, p_z 与 p_l 才不会到达无死锁不安全状态。如果 α_{pc} 和 α_c 是资源等待路径, α_c 中的 $(p_l, C(t_l))$ 能无死锁的前进。如果 p_l 没有可执行路径, 当 $R_e(M(t_l(p_l^{(p)}))) > 0$ 时, 触发 t_v 不会导致死锁。

算法 2 求 M 处的 T_{RL}

输入: 可达安全状态 $M \in R(N, M_0), T_{EN}$;

输出: T_{RL} ;

- 1) 初始化: $T_{RL} = \emptyset, F = \text{false}$;
- 2) **for** $h = 1; h \leq |T_{EN}|; h++$
- 3) **if** l_o 在 AMG 块中
- 4) $F = \text{算法 1}(M, t_{enh})$;
- 5) **if** $F = \text{true}$
- 6) $T_{RL} = T_{RL} \cup t_{enh}$;
- 7) **if** l_o 的剩余路径中有 PCG 块
- 8) **if** l_o 不在 P_{Ru} 中
- 9) **for** $k = 1; k \leq y_i; k++$ // l_o 选择第 k 条灵活子路径
- 10) **if** l_o 的剩余路径中没有 T_U
- 11) **if** l_o 能触发 T_{Gin}
- 12) $T_{RL} = T_{RL} \cup t_{enh}$;
- 13) **else**
- 14) **if** l_o 能触发 T_{Gin} 和 T_{Un}
- 15) **if** l_o 的剩余路径中没有 t_ω
- 16) $T_{RL} = T_{RL} \cup t_{enh}$;
- 17) **if** l_o 的剩余路径中有 t_ω
- 18) **if** $R_a(M(t_\omega)) > 0$
- 19) $T_{RL} = T_{RL} \cup t_{enh}$;
- 20) **if** l_o 占据的不可靠资源没有故障
- 21) 执行步骤 9)-19);
- 22) **else**
- 23) **if** l_o 此时不在 P_{Ru} 中
- 24) 执行步骤 10)-21)。

算法 3 求 M 处优化的 T_{RL} 输入: 一个可达安全状态 $M \in R(N, M_0), T_{EN}$;输出: T_{RL} ;

```

1) 初始化:  $T_{RL} = \emptyset, F = \text{false}$ ;
2) for  $h = 1; h \leq |T_{EN}|; h++$ 
3)    $F = \text{算法 1}(M, t_{enh})$ ;
4)   if  $F = \text{true}$ 
5)      $T_{RL} = T_{RL} \cup t_{enh}$ ;
6)   else
7)     if  $t_{en}$  不在  $P_{Ru}$  中或  $t_{en}$  占据的不可靠资源没有故障
8)       if  $t_{enh} \notin T_U$ 
9)         if  $\nexists t_{\omega}, t_{\omega} \in \alpha_p$ 
10)          while  $\forall R_e(M(p_l)) = 0, p_l \in \varepsilon(\alpha_p)$ 
11)             $F = \text{false}$ ;
12)            if  $M$  处  $\alpha_p$  中的  $p_l$  具有可执行路径
13)              for  $c = 1; c \leq |\alpha_c|; c++$  //  $\forall \alpha_c \in \alpha_c$ 
14)                if  $\alpha_{pc}$  和  $\alpha_c$  互为资源循环等待路径
15)                  if  $\alpha_{pc}$  和  $\alpha_c$  符合定义 16 中的条件 1
16)                    if  $R_e(M'(p_x)) \neq 0$  //  $M[t_{enh}]M'$ 
17)                       $F = \text{true}; \text{continue}$ ;
18)                    if  $\alpha_{pc}$  和  $\alpha_c$  符合定义 16 中的条件 2
19)                      if  $R_e(M'(p_x)) \neq 0$  或  $R_e(M'(p_z)) > 1$ 
20)                         $F = \text{true}; \text{continue}$ ;
21)                    if  $\alpha_{pc}$  和  $\alpha_c$  互为资源等待路径
22)                       $F = \text{true}; \text{continue}$ ;
23)                    if  $p_l$  没有可执行路径
24)                      if  $\exists t \in p_l \cdot$ 
25)                        if  $\forall p_l^{(p)}, R_e(M_t(p_l^{(p)})) > 0$  //  $M[t]M_t$ 
26)                           $F = \text{true}; \text{continue}$ ;
27)                    if  $F = \text{false}$ 
28)                      break;
29)                  else //  $\exists t_{\omega} \in \alpha_p$ 
30)                    for  $p_l \in \varepsilon(\alpha_p)$ 
31)                      if  $R_e(M(p_l)) = 0$  或  $p_l = t_{\omega} \cdot$  且  $R_s(M(p_l)) = 0$ 
32)                        执行步骤 11-28;
33)                    if  $t_{enh} \in T_U$  且  $t_{enh} \in R_s(M(t_{enh} \cdot)) > 0$ 
34)                      执行步骤 9)~32)。

```

2) α_p 中有 t_{ω} 。对于 α_p 中剩余资源空间或剩余可用资源空间为 0 的库所, 需要寻找其可执行路径 α_c 。由于算法 3 考虑了危险库所的剩余可用资源空间, 其能避免系统阻塞。因此, M' 是安全状态。

4 模拟实验

本节将提出的策略与现有常用死锁避免策略进行对比, 证明研究优势。

4.1 模型的结构分析

示例 1: 给定一个 AMS, 其零件加工过程为 $X_1: P_0 \rightarrow t_{11} \rightarrow p_6 \rightarrow t_{12} \rightarrow p_2 \cup p_3 \rightarrow t_{13} \rightarrow p_1 \rightarrow t_{14} \rightarrow P_0, X_2: P_0 \rightarrow t_{21} \rightarrow p_1 \rightarrow t_{22} \rightarrow (p_5 \rightarrow t_{23} \rightarrow p_2) \cup (p_4 \rightarrow t_{24} \rightarrow p_6) \rightarrow t_{25} \rightarrow p_7 \cup p_8 \rightarrow t_{26} \rightarrow P_0, X_3: P_0 \rightarrow t_{31} \rightarrow p_9 \rightarrow (t_{32} \rightarrow p_6 \rightarrow t_{34} \rightarrow p_4 \rightarrow t_{36}) \cup (t_{33} \rightarrow p_8 \rightarrow t_{35} \rightarrow p_7 \rightarrow t_{37}) \rightarrow p_5 \rightarrow t_{38}$ 。 $\forall i \in \mathbb{N}_9^+ \setminus \{6\}, M_0(r_i) = 2, M_0(r_6) = 3$, 资源 r_3 和

r_4 可能出现故障。

对于示例 1,文献[9]和[10]中基于 POPNs 建立的模型如图 4 所示。使用本研究策略建立的模型如图 5,其中不同颜色的变迁用于区分不同零件的变迁。初始时零件在 P_0 中,用托肯表示。例如, P_0 中第一种类型的零件颜色为 c_1 ,托肯触发 c_1 颜色的 t_{11} 进入 p_6 进行第一阶段的加工。加工完成后该托肯触发下一个 c_1 颜色的 t_{12} 继续加工。依次进行,即可完成一个零件的加工。

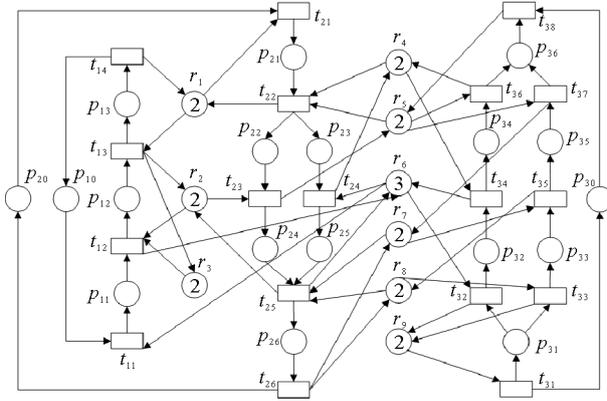


图 4 POPNs 模型

Fig. 4 POPNs model

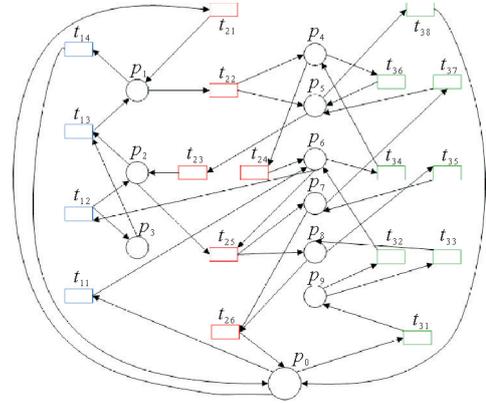


图 5 SRMGs 模型

Fig. 5 SRMGs model

对比图 4 和图 5 发现,相比 POPNs 模型,本研究的 SRMGs 模型结构简单。在 POPNs 中,用活动库所表示零件的加工步骤,模型结构随着零件的加工步骤变复杂。而本研究使用带颜色托肯的移动描述零件的加工过程,减少了活动库所的使用。

4.2 许可性分析

文献[9]和[10]会拒绝一些无法触发最近关键库所的变迁,本研究借助可执行路径使系统能接受更多的可行状态。对于示例 1 的标记 $M, M(p_1, c_2) = M(p_5, c_2) = M(p_2, c_2) = 1, M(p_6, c_2) = M(p_8, c_3) = 2$, 此时 $T_{EN} = \{t_{11}, t_{12}, t_{21}, t_{22}, t_{23}, t_{24}, t_{31}, t_{35}\}$ 。

对于 $t_{11}, T_{Gin}(t_{11}) = t_{13}, T_{Un}(t_{11}) = t_{12}, l_o$ 不能触发 t_{12} 。对于 $\alpha_p = P_0 \rightarrow t_{11} \rightarrow p_6 \rightarrow t_{12} \rightarrow p_2 \cup p_3 \rightarrow t_{13} \rightarrow p_1, \alpha_p$ 中只有 $R_a(M(p_2)) = 0$ 。存在 $\alpha_c = p_2 \rightarrow t_{25} \rightarrow p_7 \cup p_8 \rightarrow t_{26} \rightarrow P_0, p_2$ 中具有 c_2 颜色的托肯,且 $R_c(M(p_7)) > 0, R_c(M(p_8)) > 0$ 。对于 $\alpha_{pc} = P_0 \rightarrow t_{11} \rightarrow p_6 \rightarrow t_{12} \rightarrow p_2 \cup p_3, \alpha_{pc}$ 和 α_c 互为资源等待路径, $t_{11} \in T_{RL}$ 。

由于 t_{12} 是不可靠变迁,且 $R_a(M(p_2)) = 0, t_{12} \notin T_{RL}$ 。不断循环这一过程,得到 $T_{RL} = \{t_{11}, t_{21}, t_{22}, t_{23}, t_{24}, t_{31}, t_{35}\}$ 。基于文献[9]和[10]的策略得到 $T_{RL} = \{t_{31}, t_{35}\}$ 。本研究控制策略下系统的许可性相较文献[9]和[10]有所提升。

4.3 复杂度分析

文献[9]的复杂度为 $O(|T_{EN}| \cdot |P_{Ru}| + |P_{Rr}| \cdot |T_{EN}| \cdot (|N(t_{enh})| + |T_{Ni}|))$,文献[10]的复杂度为 $O(|T_{EN}| \cdot |P_{Ru}| + |P_{Rr}| \cdot |T_{EN}| \cdot (|T_{max}| + |T_{Ni}|))$,本研究的复杂度为 $O(|T_{EN}| \cdot |P_{Ru}| + (|P_{Rr}| + s) \cdot |T_{EN}| \cdot (|N(t_{enh})| + |T_{Ni}|))$ 。其中, $N(t_{enh})$ 表示 t_{enh} 到最近关键变迁的距离, $T_{max} = \max\{|T_i|\}$, T_{Ni} 为活动变迁的数量, s 为系统中零件的数量。文献[9]的复杂度与活动托肯到最近变迁的距离相关,文献[10]与路径中最大变迁数相关,本研究与零件类型数和活动托肯到最近变迁的距离相关。

图 6 显示了本研究与文献[9-10]的计算复杂度对比,图 7 展示了可达状态对比,表 1 展示了结构复杂度对比,其中 m 表示资源数, n 表示加工过程数。

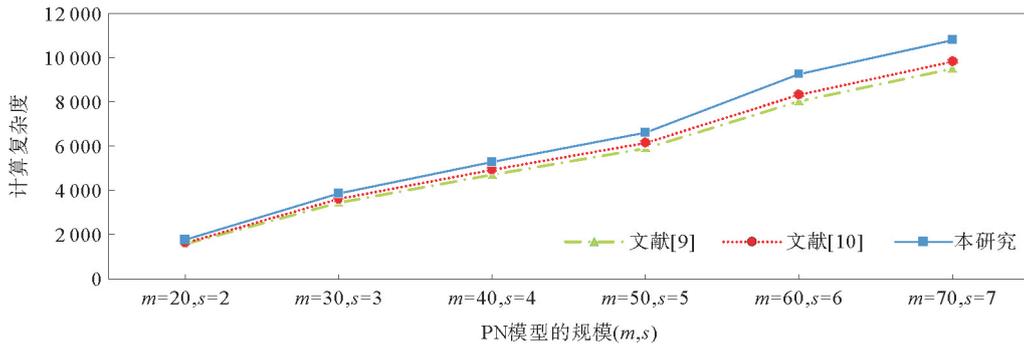


图 6 本研究策略与文献[9-10]计算复杂度变化曲线

Fig. 6 Computational complexity change curve of this paper and literature[9-10]

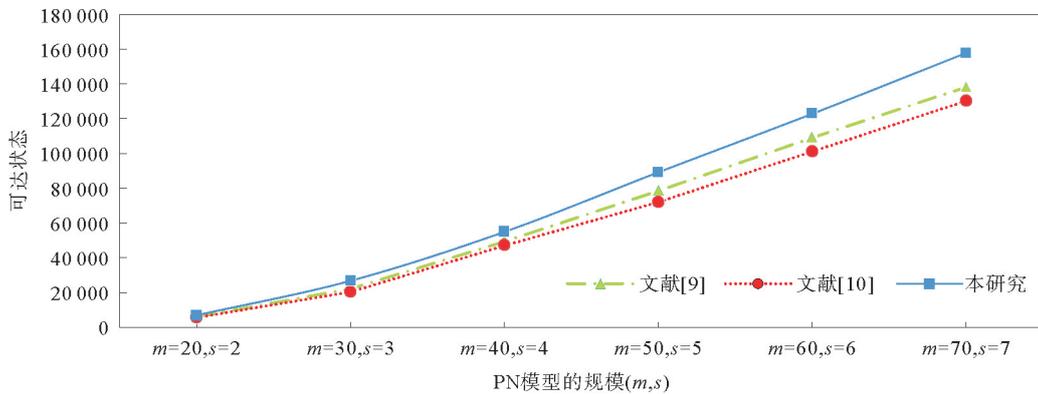


图 7 本研究策略与文献[9-10]可达状态变化曲线

Fig. 7 Achievable marking change curve of this paper and literature[9-10]

事实上,计算复杂度和结构复杂度越低越好,可达状态越多越好。文献[9]虽然复杂度略低,但其借助了外部缓冲区,结构相比文献[10]与本研究所提策略复杂。本研究所提策略的复杂度虽比文献[10]略高,但简化了结构,允许系统到达更多安全状态。与文献[9-10]相比,本研究提出的策略更适用于解决大型 AMSs 的死锁和阻塞问题。

表 1 本研究策略与文献[9-10]结构复杂度比较

Table 1 Comparison of structural complexity between this paper and literature[9-10]

项目	文献[9]	文献[10]	本研究
库所数	$m+n+s$	$m+n$	m
变迁数	$n+2s$	n	n

5 结论

针对一类具有多种不可靠资源的 AMSs,提出稳健死锁避免策略。该策略前瞻预测活动托肯能否触发最近关键变迁和最近不可靠变迁,为零件动态分配资源,使所有零件交替完成加工。此外,为避免系统阻塞,提出剩余可用资源空间,确保即使资源故障,现有资源空间仍能满足其它零件加工需求。为提高系统许可性,基于判断路径进行优化,允许系统到达更多安全状态。与现有策略相比,本策略灵活性较好,复杂度较低。未来考虑将本策略扩展到结构更复杂的 AMSs,并通过改进算法增加系统许可性。

参考文献:

[1] 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, 2022, 19(3): 1707-1718.

[2] FAN X, HU H S, YANG B Y, et al. Event circuit structures for deadlock avoidance in flexible manufacturing systems [J]. IEEE Transactions on Automation Science and Engineering, 2023, 20(1): 597-610.

[3] 白凤. 自动制造系统中的关键安全距离分析与应用[D]. 西安: 西安电子科技大学, 2022.

- BAI Feng. Analysis and application of critical safety distance in automatic manufacturing systems[D]. Xi'an:Xi'an University of Electronic Science and Technology,2022.
- [4] 刘伟,史晓浩,孙红伟. 基于逻辑混合 Petri 网的混合系统建模与分析[J]. 山东科技大学学报(自然科学版),2021,40(4):65-75.
- LIU Wei,SHI Xiaohao,SUN Hongwei. Modeling and analysis of hybrid systems based on logic hybrid Petri nets[J]. Journal of Shandong University of Science and Technology (Natural Science),2021,40(4):65-75.
- [5] DU N,HU H S. Robust control of backward conflict free systems with resources using Petri nets[C]//2016 IEEE International Conference on Automation Science and Engineering. Worth,2016:1034-1041.
- [6] DU N,HU H S. Robust control of automated manufacturing systems with flexible routes and assembly operations using Petri nets[C]//2016 IEEE 55th Conference on Decision and Control. Las Vegas,2016:4785-4790.
- [7] DU N,HU H,LIU Y. Robust control of automated manufacturing systems with assembly operations using Petri nets[C]//2016 IEEE International Conference on Robotics and Automation. Stockholm,2016:3632-3638.
- [8] DU N,HU H,ZHOU Y,et al. Robust control of automated manufacturing systems with complex structures using Petri nets[C]//2017 13th IEEE Conference on Automation Science and Engineering. Xi'an,2017:364-369.
- [9] WANG X,HU H S. A robust control approach to automated manufacturing systems allowing multitype and multi-quantity of resources with Petri nets[J]. IEEE Transactions on Systems, Man, and Cybernetics; Systems,2020,50(10):3499-3514.
- [10] DU N,HU H S,ZHOU M C. 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.
- [11] WU N Q. Necessary and sufficient conditions for deadlock-free operation in flexible manufacturing systems using a colored Petri net model[J]. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews),1999,29(2):192-204.
- [12] ZHAO Y M,CHAI X J,ZHAO L Z. An efficient deadlock avoidance policy for FMS using ROPN[J]. Advanced Materials Research,2014,998-999:751-754.
- [13] LIU D,HOU Y F,BARKAOUIK,et al. Intrinsically live structures in process and resource-oriented Petri nets modeling automated manufacturing systems[C]//IEEE International Conference on Netting, Sensing and Control. IEEE,2014:578-583.
- [14] CHEN H F,WU N Q,ZHOU M C. Resource-oriented Petri net-based approach to deadlock prevention of AMSs[C]//2015 IEEE International Conference on Systems, Man, and Cybernetics. Hong Kong,2015:515-520.
- [15] ZHOU M C,WU N Q. System modeling and control with resource-oriented Petri nets[M]. Boca Raton: CRC Press,2018.

(责任编辑:傅游)