

基于蚁群优化的时间 Petri 网及其在柔性制造系统调度优化中的应用

潘 理^{1,2}, 郑 红³, 郭观七^{1,2}, 刘显明⁴

(1. 湖南理工学院信息与通信工程学院, 湖南岳阳 414006; 2. 湖南理工学院复杂系统优化与控制湖南省普通
高等学校重点实验室, 湖南岳阳 414006; 3. 华东理工大学, 上海 200237; 4. 江西省电力公司, 江西南昌 330077)

摘 要: Petri 网的优化调度求解主要采用基于状态空间的智能搜索方法, 执行效率往往不高. 提出基于蚁群优化的时间 Petri 网, 通过在 Petri 网的变迁和进化规则中引入蚁群优化机制, 改变 Petri 网原有的非确定性选择行为, 使之具备蚁群寻优选择智能行为, 克服了需要基于状态空间进行启发式搜索的缺陷, 提高了调度优化的计算效率. 最后应用蚁群优化时间 Petri 网模型, 对柔性制造系统的区间 Job Shop 调度问题进行建模和优化求解. 实验结果进一步验证了所提模型和方法的有效性.

关键词: 调度优化; 蚁群优化; 时间 Petri 网; 柔性制造系统

中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2014)08-1531-07

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2014.08.011

Time Petri Net with Ant Colony Mechanism and Its Application in Scheduling Optimization of Flexible Manufacturing Systems

PAN Li^{1,2}, ZHENG Hong³, GUO Guan-qi^{1,2}, LIU Xian-ming⁴

(1. School of Information and Communication Engineering, Hunan Institute of Science and Technology, Yueyang, Hunan 414006, China;

2. Key Laboratory of Optimization and Control of Complex Systems (College of Hunan Province), Hunan Institute of Science and Technology, Yueyang, Hunan 414006, China; 3. Information Science and Engineering College, East China University of Science and Technology, Shanghai 200237, China;

4. Information and Communication Branch, Jiangxi Electric Power Company, Nanchang, Jiangxi 330077, China)

Abstract: Heuristic algorithms based on state search are usually used to solve scheduling problems of Petri nets. Suffering from the state space explosion, their executions are often not efficient for large-scale problems. This paper proposes a time Petri net with ant colony mechanism to address this problem. We introduce pheromones and foraging rules of ants into transitions and evolution rules of a time Petri net. The proposed model can work out scheduling problems during the run of a Petri net without building reachability tree. We use this model to describe and solve an interval job shop scheduling problem. The experimental results further verify the effectiveness of the proposed model.

Key words: scheduling optimization; ant colony optimization; time Petri net; flexible manufacturing system

1 引言

调度优化是提高系统效率的重要途径, 是系统设计者关注的焦点. 从 Petri 网的角度看, 调度优化是确定从初始标识到目标标识的最优变迁实施序列^[1]. 对柔性制造系统而言, 为了获得更好的系统性能, 常常需要在满足时限约束的条件下求解系统的优化调度^[1,2].

目前, 求解优化调度的主要方法是采用人工智能技

术对状态空间(或部分状态空间)进行搜索. 一种较好的解决方案是采用启发式搜索算法^[3~5], 它根据一定的启发函数产生部分可达图, 然后在可达图中选择最好的路径. 这一处理缩短了计算时间, 对于复杂系统来说是一种可行的方案. 但是, 启发式搜索方法仍需要构造可达空间进行分析, 因此面对大规模复杂系统, 往往执行效率不高. 另一类调度优化方法主要依靠遗传算法优良的搜索能力^[6~8]. 但是, 基于遗传算法的优化调度需要

进行染色体的编码解码和变异操作的可达性判断,增加了问题处理的复杂度。

本文提出基于蚁群优化的时间 Petri 网模型 (AntTPN),用于求解 Petri 网的优化调度问题.我们将蚂蚁信息素引入到 Petri 网的变迁,设置启发式因子与变迁动态实施间隔相关联,将蚁群寻优规则融合到 Petri 网的进化规则中.与近期相关研究工作相比^[9-12],主要有以下几处不同:(1)引入蚁群的 Petri 网类型不同.上述工作在随机 Petri 网、时延 Petri 网中引入蚁群思想^[9-11],本文将蚁群机制引入时间 Petri 网.(2)采用的蚁群算法思想不同.上述工作多使用蚂蚁系统(AS)算法思想^[9-11],本文则采用蚁群系统(ACS)算法思想,ACS 算法能保证解的收敛性^[13].(3)模型与蚁群机制的融合程度不同.上述工作或者利用蚁群算法对 Petri 网模型生成的可达树进行搜索^[10],两者融合度不高;或者仅考虑静态启发^[12],未能将 Petri 网的进化结果融合到寻优启发中,难以达到理想的启发效果.本文将变迁的动态实施间隔关联到启发式因子,使蚁群寻优与 Petri 网进化规则有效结合起来,在 Petri 网模型中实现静态建模与动态优化的统一。

AntTPN 方法在 Petri 网运行过程中进行优化求解.首先放置一群蚂蚁令牌在 Petri 网初始标识上,再根据蚁群 Petri 网的寻优规则,经过多次运行,便可逐步找到优化路径.与启发式 L_1 调度算法相比^[3],不需要构造庞大的搜索空间,可以节省大量空间成本;与遗传算法相比^[6],不需要进行染色体编码解码和变异操作的可达性判定工作,可以避免不必要的计算消耗。

2 基于蚁群优化的时间 Petri 网

2.1 模型定义

假设 R 是实数集, R^+ 是非负实数集, N 是自然数集 $\{0, 1, 2, \dots\}$. 给定 $a, b \in R$ 且 $a \leq b$, 定义 $I = \{x \in R \mid a \leq x \leq b\}$ 为实数闭区间, 记 $I = [a, b]$. 用 $IR (IR^+)$ 表示所有(非负)实数闭区间的集合. 设 $I, I_1, I_2 \in IR$ 且 $I = [a, b]$, 定义闭区间的一元和二元运算如下: $\downarrow I = a$, $\uparrow I = b$; $I_1 + I_2 = [\downarrow I_1 + \downarrow I_2, \uparrow I_1 + \uparrow I_2]$; $I_1 - I_2 = [\downarrow I_1 - \uparrow I_2, \uparrow I_1 - \downarrow I_2]$; 若 $I_1 \cap I_2 \neq \emptyset$, 则 $I_1 \cap I_2 = [\max\{\downarrow I_1, \downarrow I_2\}, \min\{\uparrow I_1, \uparrow I_2\}]$.

在经典 Petri 网的基础上, 时间 Petri 网为每个变迁关联一个静态实施间隔, 用以表示该变迁最小和最大的持续使能时间。

定义 1^[14] 一个时间 Petri 网是一个六元组 $TPN = (P, T, Pre, Post, M_0, SI)$, 其中:

- (1) P 是有穷、非空的位置集;
- (2) T 是有穷、非空的变迁集;
- (3) $Pre: P \times T \rightarrow N$ 是向前关联矩阵;

(4) $Post: P \times T \rightarrow N$ 是向后关联矩阵;

(5) $M_0: P \rightarrow N$ 是初始标识;

(6) $SI: T \rightarrow IR^+$ 是变迁的静态实施间隔映射函数. 对 $\forall t \in T$, $SI(t)$ 表示变迁 t 的静态实施间隔。

$Pre(p, t) = i (i > 0)$ 当且仅当存在一条从库所 p 到变迁 t 的权值为 i 的弧; $Pre(p, t) = 0$ 当且仅当不存在从库所 p 到变迁 t 的弧. 用 $Pre(t) \in N^P$ 表示 t 的输入库所多重集. 类似地定义 $Post(p, t)$ 和 $Post(t)$. Petri 网的标识 M 是一个 $|P|$ 维向量, 其中第 i 个元素 $M(p_i)$ 表示库所 p_i 中的令牌数。

定义 2^[14] 如果 $Pre(t) \leq M$, 则称变迁 $t \in T$ 在标识 M 是使能的. 如果变迁 t 在标识 M 是使能的, 那么 t 可以实施并产生一个新的后继标识 M' , 且 $M' = M - Pre(t) + Post(t)$.

用 $En(M)$ 表示标识 M 的所有使能变迁的集合. 用 $Newly(M, t_f)$ 表示在标识 M 实施 t_f 后新使能的所有变迁的集合, 即 $Newly(M, t_f) = En(M - Pre(t_f) + Post(t_f)) \setminus En(M - Pre(t_f))$.

本质上, 蚂蚁的信息素可以放置在 Petri 网的变迁、库所、弧, 甚至是令牌上. 本文将信息素关联到变迁, 将启发式因子与变迁实施间隔相联系, 并将蚂蚁的选择规则、信息素更新规则、启发式因子更新规则入到 Petri 网进化规则中。

定义 3 一个蚁群优化时间 Petri 网是一个八元组 $AntTPN = (P, T, Pre, Post, M_0, SI; \tau, \eta)$, 其中:

(1) $(P, T, Pre, Post, M_0, SI)$ 是一个 TPN;

(2) $\tau: T \rightarrow R^+$ 是变迁的信息素映射函数, 对 $\forall t \in T$, $\tau(t)$ 表示放置在变迁 t 上信息素数量;

(3) $\eta: T \rightarrow R^+$ 是变迁的启发式因子映射函数, 对 $\forall t \in T$, $\eta(t)$ 表示变迁 t 的启发式因子。

初始情况下, 在每个变迁上放置相同数量的信息素. 这里取所有变迁的时间上界值之和的倒数, 然后分配到每个变迁上. 即对任一变迁 t_i , 它的初始信息素量 $\tau_0(t_i)$ 设为:

$$\tau_0(t_i) = \frac{1}{|T| \sum_{t \in T} \uparrow SI(t)}$$

设置变迁的启发式信息值与变迁的实施时间间隔相关联. 变迁 t_i 的初始启发式因子 $\eta(t_i)$ 可采用以下三种方式:

$$(1) \text{ 上界启发式: } \eta_0(t_i) = \frac{1}{\uparrow SI(t_i) + 1}$$

$$(2) \text{ 下界启发式: } \eta_0(t_i) = \frac{1}{\downarrow SI(t_i) + 1}$$

$$(3) \text{ 中值启发式: } \eta_0(t_i) = \frac{2}{\downarrow SI(t_i) + \uparrow SI(t_i) + 1}$$

注意, 分母加 1 是为了避免出现除数为 0 的情况。

2.2 进化规则

下面引入蚁群寻优规则到时间 Petri 网的状态类方法,形成基于蚁群优化的时间 Petri 网的状态类进化规则.

定义 4^[14] 一个状态类是一个二元组 $C = (M, D)$, 其中:

(1) $M: P \rightarrow N$ 是标识;

(2) $D: T \times T \rightarrow IR$ 是实施域, 这里 $T = T \cup \{t_*, t_0\}$, t_0 表示零变迁, 它的全局实施时间为 0; t_* 表示最近一次被实施的变迁.

$\forall t, t' \in En(M)$, $D(t', t)$ 表示变迁 t 相对于变迁 t' 的实施时间间隔, $D(t_*, t)$ 表示变迁 t 相对于状态类 C 的实施时间间隔, $D(t_0, t)$ 则表示变迁 t 的全局实施时间间隔, $D(t_0, t_*)$ 则表示状态类 C 的全局到达时间间隔.

初始状态类为 $C_0 = (M_0, D_0)$, 其中 M_0 是初始标识; $\forall t, t' \in En(M_0)$, $D(t, t) = 0$, $D_0(t', t) = SI(t) - SI(t')$, $D_0(t_0, t) = SI(t)$ 且 $t_* = t_0$.

定义 5(可实施条件) 一个变迁 t_f 在状态类 $C_k = (M_k, D_k)$ 是可实施的, 当且仅当同时满足下列条件:

(1) $t_f \in En(M_k)$;

(2) $\uparrow D_k(t_*, t_f) \geq 0$;

(3) $\forall t \in En(M - Pre(t_f))$, $\uparrow D_k(t_f, t) \geq 0$.

条件(1)保证 t_f 是使能的; 条件(2)保证 t_f 不是过期变迁, 即没有超越其时间上界; 条件(3)保证 t_f 在每个与 t_f 不冲突的变迁之前实施. 用 $Fr(C_k)$ 表示在 C_k 可实施的所有变迁的集合. 接下来, 使用蚁群选择规则从可实施变迁集 $Fr(C_k)$ 中选择一个候选变迁实施.

定义 6(选择规则) 在状态类 C_k , $\forall t_j \in Fr(C_k)$, 根据式(1)伪随机比例方法选择实施变迁:

$$t_j = \begin{cases} \arg \max_{t_i \in Fr(C_k)} \{ \tau_k(t) [\beta_k^{q_0}(t)] \}, & \text{if } q \leq q_0 \\ t_l, & \text{otherwise} \end{cases} \quad (1)$$

$$prob(t_l) = \frac{\tau_k(t_l) \beta_k^{q_0}(t_l)}{\sum_{t_i \in Fr(C_k)} \tau_k(t_i) \beta_k^{q_0}(t_i)} \quad (2)$$

这里, $\tau_k(t)$ 表示在状态类 C_k 变迁 t 上积累的信息素值, $\eta_k(t)$ 表示在状态类 C_k 变迁 t 的启发式信息值, β 是启发式信息的影响力参数, q 是均匀分布在区间 $[0, 1]$ 的一个随机变量, q_0 ($0 \leq q_0 \leq 1$) 是伪随机比例参数. 若 $q \leq q_0$, 则在 $Fr(C_k)$ 中选择 $\tau_k(t) \beta_k^{q_0}(t)$ 值最大的变迁实施; 反之, 根据式(2)给出的概率分布用赌轮法选择某个变迁 t_l 实施. 值得注意的是, 启发式信息参数 β 可以放置在指数或底数位置^[13, 15]. 通过实验发现, 对于本文问题, β 参数放置在底数位置时解的性能相对更好.

定义 7(实施规则) 假设从状态类 $C_k = (M_k, D_k)$

实施变迁 $t_f \in Fr(C_k)$ 后到达状态类 $C_{k+1} = (M_{k+1}, D_{k+1})$, 则状态类 C_{k+1} 可由下列实施规则确定:

(1) 标识 M_{k+1} :

$$M_{k+1} = M_k - Pre(t_f) + Post(t_f)$$

(2) 实施域 $D_{k+1} = \forall t_i, t_j \in En(M_{k+1}) \cup \{t_0, t_f\}$,

(I) $D_{k+1}(t_i, t_j) =$

$$\begin{cases} 0, & \text{if } t_i = t_j \\ -SI(t_i), & \text{if } t_i \in Newly(M_k, t_f) \\ \lfloor \downarrow D_k(t_i, t_f), DMIN(M, t_f) \rfloor, & \text{otherwise} \end{cases}$$

这里 $DMIN(M, t_f) = \min \{ \uparrow D_k(t_i, t_f), \uparrow D_k(t_i, t) \mid t \in En(M - Pre(t_f)) \}$.

(II) $D_{k+1}(t_i, t_j) =$

$$\begin{cases} 0, & \text{if } t_i = t_j \\ D_{k+1}(t_i, t_f) - D_{k+1}(t_j, t_f), & \text{if } t_i, t_j \in Newly(M_k, t_f) \\ D_k(t_i, t_j) \cap (D_{k+1}(t_i, t_f) - D_{k+1}(t_j, t_f)), & \text{otherwise} \end{cases}$$

规则(1)计算新标识 M_{k+1} . 规则(2 I) 计算变迁相对于当前状态类 C_{k+1} 的实施间隔; 规则(2 II) 计算任何两个变迁之间的相对实施间隔. 可以看出, 每次变迁实施, 变迁的实施间隔会发生动态变化. 如果仍然采用初始静态启发式因子进行启发, 难以达到理想的启发效果. 因此, 我们根据实施规则的计算结果, 更新关联在变迁上的启发式因子.

定义 8(启发式因子更新规则) 假设从状态类 $C_k = (M_k, D_k)$ 实施 t_f 到达状态类 $C_{k+1} = (M_{k+1}, D_{k+1})$, $\forall t_i \in En(M_{k+1})$, 变迁 t_i 的启发式因子 $\eta_{k+1}(t_i)$ 可采用下列一种方式更新:

$$(1) \text{ 上界启发: } \eta_{k+1}(t_i) = \frac{1}{\uparrow D_{k+1}(t_*, t_i) + 1}$$

$$(2) \text{ 下界启发: } \eta_{k+1}(t_i) = \frac{1}{\downarrow D_{k+1}(t_*, t_i) + 1}$$

$$(3) \text{ 中值启发: } \eta_{k+1}(t_i) =$$

$$\frac{2}{\downarrow D_{k+1}(t_*, t_i) + \uparrow D_{k+1}(t_*, t_i) + 1}$$

变迁实施之后, 需立即更新该变迁的信息素, 称为局部更新. 局部更新会导致该变迁信息素的减少, 但不会低于 τ_0 . 局部更新规则可有效避免收敛到同一路径.

定义 9(局部信息素更新规则) 假设从状态类 $C_k = (M_k, D_k)$ 实施 t_f 到达状态类 $C_{k+1} = (M_{k+1}, D_{k+1})$, 按如下规则更新 t_f 的信息素:

$$\tau_{k+1}(t_f) = (1 - \zeta) \tau_k(t_f) + \zeta_0(t_f)$$

ζ ($0 < \zeta < 1$) 是一个参数. 全局更新在所有蚂蚁都完成它们的路径之后执行, 使蚂蚁的搜索主要集中在目前为止所找到的最好路径的邻域内.

定义 10(全局信息素更新规则) 当所有蚂蚁完成一次迭代, 选择从初始标识到目标标识的最优路径, 对

该路径上每个变迁 t_i 按如下规则更新信息素:

$$\tau'(t_i) = (1 - \rho)\tau(t_i) + \rho \frac{1}{L_{op}}$$

其中 $\rho(0 < \rho < 1)$ 是一个参数, L_{op} 是到目前为止最优路径的时间长度。

2.3 运行流程

接下来给出基于蚁群优化的时间 Petri 网的运行流程, 具体步骤如下:

(1) 初始化: 设置最大循环次数, 即 $N_c \leq n$, 初始值 $N_c = 0$; 设置最大蚂蚁数, 即 $K \leq m$, 初始值 $K = 0$; 初始化各变迁上的信息素量 τ_0 。

(2) 在初始标识中放置一组蚂蚁令牌, 每只蚂蚁执行从初始标识开始的一次运行。

(3) 对于当前运行的蚂蚁 K , 执行:

(3.1) 计算当前状态类的可实施变迁集;

(3.2) 根据选择规则确定具体的可实施变迁;

(3.3) 根据实施规则实施变迁, 到达下一状态类, 更新相关变迁的信息素值和启发式因子值;

(3.4) 若无可实施变迁或达到目标标识, 则转到(3)运行下一只蚂蚁, 否则转到(3.1)。

(4) 当所有蚂蚁运行完毕, 即一次迭代完成, 选择到目前为止的最优路径, 按全局更新规则更新该路径的信息素量。

(5) 若循环迭代次数已满, 则输出最优结果并结束, 否则转到第(2)步。

可以看出, 利用 AntTPN 进行优化调度求解需要时间 $O(K \cdot N \cdot L \cdot |T|^2)$, 这里 K 是蚂蚁数, N 是迭代次数, $|T|$ 是变迁数, L 是解的路径长度。由于只需要保持一个状态类(包括标识和实施域)和一条路径的存储空间, 所以 AntTPN 方法需空间 $O(|P| + |T|^2 + L)$, 这里 $|P|$ 是库所数。而启发式搜索方法需要保存全部(或部分)状态类树^[3], 需空间 $O(2^L \cdot |T|^2)$ 。因此我们的方法更节省空间成本。

如果利用遗传算法进行 Petri 网优化调度^[6], 其选择操作需时间 $O(Q \cdot L \cdot |T|^2)$, 交叉操作需 $O(P_c \cdot Q \cdot L)$, 变异操作需 $O(P_m \cdot 2^{L/2} \cdot |T|^2)$, 这里 Q 是种群数量, P_c 是交叉概率, P_m 是变异概率。因此该遗传算法的时间复杂度是 $O[N \cdot (Q \cdot L \cdot |T|^2 + P_m \cdot 2^{L/2} \cdot |T|^2)]$ 。尽管变异概率较小(一般取 0.01 ~ 0.2), 但变异操作的时间和空间成本不可忽视, 因为每次变异都需构造一棵以变异点基因座为根的可达树^[6]。若取 $N \geq 100$, 复杂度可写为 $O(Q \cdot N \cdot L \cdot |T|^2 + 2^{L/2} \cdot |T|^2)$, 其中第一项 $O(Q \cdot N \cdot L \cdot |T|^2)$ 略高于 AntTPN 方法 $O(K \cdot N \cdot L \cdot |T|^2)$, 因为种群数量 Q 通常大于蚂蚁数量 K ; 第二项是由变异操作可达性判断带来的额外计算成本。

3 典型区间 Job Shop 调度问题建模与优化

Job Shop 调度问题是柔性制造系统中经典问题。该问题可描述为: 在 M 个机器上加工 N 个工件, 每个工件 J_i 由 n_i 个工序组成, n_i 个工序之间有工艺的先后约束, 工件的每道工序可由 M 台机器中的多台机器加工。用 O_{ijk} 表示第 i 个工件的第 j 道工序在机器 M_k 上加工。区间 Job Shop 调度问题使用时间间隔表示加工时间^[16]。调度的任务就是在 M 台机器上安排 N 个工件的加工任务, 在满足一些约束条件的情况下, 最优化给定的性能指标。

Makespan 是 Job Shop 调度问题中一个最常用的性能指标。最小化 Makespan 是指求解所有工件的最大完成时间的最小值。考虑一个柔性制造系统, 它由 3 台机器组成, 能够处理 3 种作业, 每种作业均需要 3 道工序完成。表 1 和表 2 分别描述了每种作业的资源需求和操作时间。

表 1 作业的资源需求

工序	作业 1	作业 2	作业 3
1	M_1/M_2	M_2/M_3	M_2/M_3
2	M_2/M_3	M_2	M_1/M_2
3	M_1	M_1/M_3	M_1/M_3

表 2 TPN 的库所和变迁描述

库所	描述	变迁	描述	时间间隔
p_1	机器 1	t_1	操作 $O_{1,1,1}$	[3,6]
p_2	机器 2	t_2	操作 $O_{1,1,2}$	[2,4]
p_3	机器 3	t_3	操作 $O_{1,2,2}$	[3,5]
p_4	工件 1 准备加工	t_4	操作 $O_{1,2,3}$	[2,10]
p_5	工件 1 完成工序 1	t_5	操作 $O_{1,3,1}$	[2,6]
p_6	工件 1 完成工序 2	t_6	操作 $O_{2,1,2}$	[1,4]
p_7	工件 1 完成工序 3	t_7	操作 $O_{2,1,3}$	[2,3]
p_8	工件 2 准备加工	t_8	操作 $O_{2,2,2}$	[2,5]
p_9	工件 2 完成工序 1	t_9	操作 $O_{2,3,1}$	[2,6]
p_{10}	工件 2 完成工序 2	t_{10}	操作 $O_{2,3,3}$	[3,4]
p_{11}	工件 2 完成工序 3	t_{11}	操作 $O_{3,1,2}$	[2,10]
p_{12}	工件 3 准备加工	t_{12}	操作 $O_{3,1,3}$	[1,11]
p_{13}	工件 3 完成工序 1	t_{13}	操作 $O_{3,2,1}$	[3,6]
p_{14}	工件 3 完成工序 2	t_{14}	操作 $O_{3,2,2}$	[5,5]
p_{15}	工件 3 完成工序 3	t_{15}	操作 $O_{3,3,1}$	[2,7]
		t_{16}	操作 $O_{3,3,3}$	[3,9]

根据上面的结构和需求描述, 建立该柔性制造系统的 TPN 模型, 如图 1 所示。其库所和变迁的详细说明见表 2。

模型建立之后, 我们运行 AntTPN。当所有蚂蚁完成

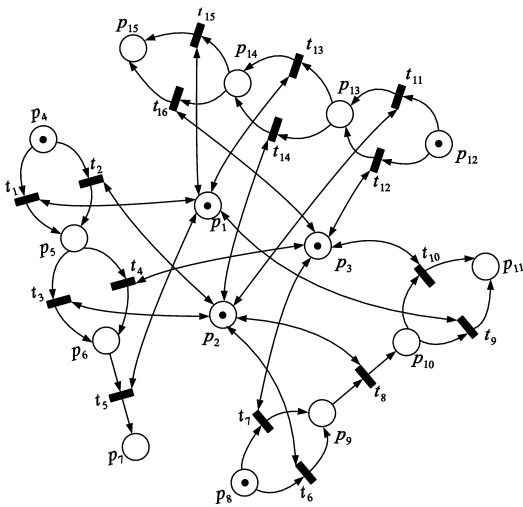


图1 柔性制造系统的TPN模型

它们的路径之后,选取最优蚂蚁路径进行全局信息素更新.运行若干次迭代之后,可获得系统的优化调度.调度的优化程度受 β, q_0, ρ 等参数影响.接下来讨论这些参数优化设置.

伪随机比例参数 q_0 确定使用选择 $\tau_k(t)\beta^{q_0(t)}$ 最大值或赌轮法的比例. $q_0 = 1$ 表明只使用 $\tau_k(t)\beta^{q_0(t)}$ 值最大的变迁. $q_0 = 0$ 表明只使用赌轮法选择规则.表 3 是运行 20 次的平均结果.可以看出,伪随机比例参数 $q_0 \in [0.4, 0.5]$ 能取得较好解.当 $q_0 = 0.4$ 时结果最优.

表 3 伪随机比例 $q_0(m = 10, n = 100, \beta = 10, \rho = 0.1, \text{下界启发})$

q_0	最优下界平均值	最优上界平均值	最优解的平均迭代层次
0.1	8.0	19.35	58.05
0.2	8.0	18.8	36.15
0.3	8.0	18.8	46.45
0.4	8.0	17.15	34.6
0.5	8.0	17.4	38.9
0.6	8.0	18.1	41.3
0.7	8.1	17.95	47.5
0.8	8.2	17.7	41.7
0.9	8.2	19.1	40.0

启发式因子参数 β 用于权衡启发式信息和信息素的相对影响力. $\beta = 1$ 意味着启发式信息没被蚂蚁利用. β 越大表明启发式信息在蚁群优化中影响度越大.表 4 实验结果表明,当 $\beta \in [5, 20]$ 能取得较好解,当 $\beta = 10$ 时结果最优.

在实验中,信息素更新规则的参数 ζ, ρ 取 ACS 蚁群算法的推荐值 0.1.表 5 是三种启发式因子的测试结果,其中最佳下界是 8,最佳上界是 12,最佳中值是 10.5.

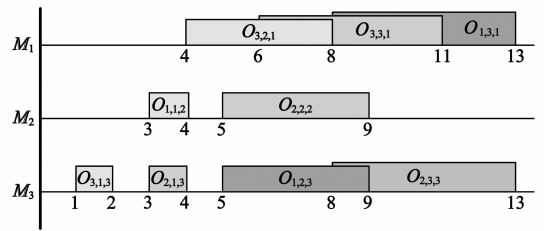
表 4 启发式因子参数 $\beta(m = 10, n = 100, q_0 = 0.4, \rho = 0.1, \text{下界启发})$

β	最优下界平均值	最优上界平均值	最优解的平均迭代层次
1	8.75	17.0	58.05
2	8.05	17.75	45.75
3	8.05	18.2	56.35
4	8.05	18.1	41.95
5	8.0	17.85	43.95
6	8.0	17.6	46.35
8	8.0	17.65	44.65
10	8.0	16.85	33.4
20	8.0	17.6	47.65
50	8.05	18.05	37.45

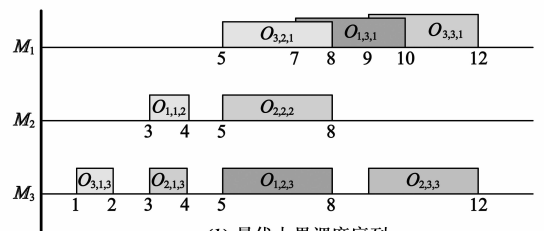
表 5 启发式因子 $\eta(m = 10, n = 100, \beta = 10, q_0 = 0.2, \rho = 0.1)$

η	最优值	最优值次数	平均值
下界启发	8(下界值)	20	8(下界值)
上界启发	12(上界值)	11	12.65(上界值)
中值启发	10.5(中值)	1	11.15(中值)

由于区间 Job Shop 调度问题的操作时间是一个时间区间.因此, Makespan 的时间值也会落在一个区间中.采用下界启发因子,求出具有最优下界值的最优调度是 $t_{12}t_7t_2t_{13}t_4t_8t_{15}t_5t_{10}$,完工时间 $[8, 13]$,其调度序列见图 2(a),每一区段代表一个操作的完成时间范围.例如, $O_{2,2,2}$ 区段代表操作 t_8 的最早完成时间是 5,最晚完成时间是 9.可以看出,所有操作都完成的最小时间是 8.同样,采用上界启发因子,求出最优上界值调度为 $t_{12}t_7t_2t_4t_8t_{13}t_5t_{15}t_{10}$,完工时间 $[9, 12]$,其调度序列见图 2(b).这两个调度均为最优中值调度.



(a) 最优下界调度序列



(b) 最优上界调度序列

图2 区间 Job Shop 调度问题的优化调度

4 实验对比

本节通过实验对比分析 AntTPN 方法和基于 Petri

网的启发式搜索算法及遗传算法. 实验环境为 CPU 2.6GHz, Memory 8G 的计算机, 实验平台是 Matlab 7.0.

首先对比 AntTPN 方法和启发式搜索方法 (L_1 调度算法)^[3]. 实验案例为文献[3]中的例 3. 一个柔性制造系统具有 3 台机器, 能处理 5 种作业, 每种作业需要加工 10 个产品 (lot = 10), 每种作业需要 4 道工序才能完成. 表 6 列出了该系统的作业资源需求以及操作时间.

表 6 作业资源需求及操作时间 (lot = 10)

作业 1		作业 2		作业 3		作业 4		作业 5	
1	M_1/M_3	M_1/M_2	$M_1/M_2/M_3$	M_2/M_3	M_1/M_3	M_1/M_3			
2	M_2	M_3	M_2/M_3	M_1/M_3	M_2/M_3				
3	M_1/M_3	M_1/M_2	M_1/M_3	M_2/M_3	M_1/M_2				
4	M_1/M_2	M_1/M_3	M_1/M_2	$M_1/M_2/M_3$	$M_1/M_2/M_3$				
操作	时间	操作	时间	操作	时间	操作	时间	操作	时间
$O_{1,1,1}$	7	$O_{2,3,2}$	4	$O_{3,4,2}$	3	$O_{5,1,3}$	15		
$O_{1,1,3}$	4	$O_{2,4,1}$	8	$O_{4,1,2}$	9	$O_{5,2,2}$	7		
$O_{1,2,2}$	3	$O_{2,4,3}$	4	$O_{4,1,3}$	5	$O_{5,2,3}$	14		
$O_{1,3,1}$	3	$O_{3,1,1}$	10	$O_{4,2,1}$	6	$O_{5,3,1}$	5		
$O_{1,3,3}$	6	$O_{3,1,2}$	15	$O_{4,2,3}$	2	$O_{5,3,2}$	8		
$O_{1,4,1}$	2	$O_{3,1,3}$	8	$O_{4,3,2}$	7	$O_{5,4,1}$	4		
$O_{1,4,2}$	4	$O_{3,2,2}$	2	$O_{4,3,3}$	12	$O_{5,4,2}$	6		
$O_{2,1,1}$	8	$O_{3,2,3}$	6	$O_{4,4,1}$	9	$O_{5,4,3}$	8		
$O_{2,1,2}$	12	$O_{3,3,1}$	2	$O_{4,4,2}$	6				
$O_{2,2,3}$	4	$O_{3,3,3}$	4	$O_{4,4,3}$	3				
$O_{2,3,1}$	7	$O_{3,4,1}$	6	$O_{5,1,1}$	10				

L_1 调度算法是一种基于 A^* 的 Petri 网启发式搜索算法. 设置 L_1 调度算法的 h 函数 $h(M) = \text{dep}(M)$, 这里 $\text{dep}(M)$ 表示标识 M 在可达图中的深度, 即从初始标识到达标识 M 的实施变迁数目^[3]. 设置 AntTPN 运行参数为: 蚂蚁数 $m = 10$, 迭代次数 $n = 100$, 启发式因子 $\beta = 10$, 伪随机比例 $q_0 = 0.2$, 信息素更新参数 $\rho = 0.1$.

表 7 AntTPN 方法与 L_1 调度算法实验对比

	L_1 调度算法	AntTPN 方法
迭代次数	758	45.2
最终标识深度	400	200
最优调度	[426, 426]	[392, 392]

表 7 实验数据是运行 20 次的平均结果. AntTPN 方法 10 只蚂蚁平均迭代 45.2 次可找到优化解 [392, 392], 优于 L_1 调度算法获得的解. 因此, 相比于 L_1 调度算法, AntTPN 方法在解的质量、搜索迭代层次上具有优势.

下面对比分析 AntTPN 方法和基于遗传算法的 Petri 网调度方法^[6]. 实验案例是文献[6]的例子. IC 生产线的光刻单元的调度. 该系统有四台机器, 能处理两种作业, 每种作业有 3 道工序, 需加工 3 个产品. 调度过程需

要考虑对机器进行定期维护, 每台机器每加工两次必须维修一次, 维修时间为 5. 表 8 给出该系统的作业需求和加工时间.

表 8 作业资源需求及操作时间 (lot = 3)

		作业 1		作业 2	
1		M_1/M_2		$M_1/M_3/M_4$	
2		$M_2/M_3/M_4$		$M_1/M_2/M_3$	
3		$M_1/M_2/M_4$		M_4	
操作	时间	操作	时间	操作	时间
$O_{1,1,1}$	15	$O_{1,3,1}$	20	$O_{2,1,4}$	25
$O_{1,1,2}$	20	$O_{1,3,2}$	20	$O_{2,2,1}$	25
$O_{1,2,2}$	20	$O_{1,3,4}$	25	$O_{2,2,2}$	25
$O_{1,2,3}$	25	$O_{2,1,1}$	15	$O_{2,2,3}$	30
$O_{1,2,4}$	30	$O_{2,1,3}$	20	$O_{2,3,4}$	30

文献[6]设置交叉概率 $P_c = 0.9$, 变异概率 $P_m = 0.2$, 初始种群 $Q = 100$, 迭代次数 $N = 100$, 代沟 $G = 0.85$. 设置 AntTPN 运行参数为: 蚂蚁数 $m = 20$, 迭代次数 $n = 100$, 启发式因子 $\beta = 10$, 伪随机比例 $q_0 = 0.4$, 信息素更新参数 $\rho = 0.1$.

表 9 给出平均运行 20 次的实验结果. 虽然基于遗传算法的 Petri 网调度方法在大多数情况下能获得最优解. 但 AntTPN 方法运行 20 次全部获得最优解, 而且平均运行时间略好于遗传算法.

表 9 AntTPN 方法与遗传算法实验对比

	遗传算法	AntTPN 方法
最优调度	[135, 135] (17 次)	[135, 135] (20 次)
最优调度序列	[140, 140] (2 次) [145, 145] (1 次) $O_{2,1,1} O_{2,1,3} O_{1,1,1} M_1 O_{2,1,3} O_{2,2,2} M_3 O_{1,1,1} O_{2,2,2}$ $O_{1,1,1} O_{2,3,4} M_1 M_2 O_{1,2,3} O_{1,3,1} O_{1,2,2} O_{2,3,4} O_{2,2,3}$ $M_4 M_3 O_{1,3,1} O_{1,2,2} M_1 M_2 O_{2,3,4} O_{1,3,2} M_4$ (M_i 表示第 i 台机器的维修操作)	
平均运行时间	216.53s	143.04s

5 结论

提出基于蚁群优化的时间 Petri 网模型 (AntTPN), 通过引入蚂蚁信息素到 Petri 网结构, 引入蚁群寻优机制到 Petri 网进化规则, 实现了 Petri 网调度问题的静态建模和动态优化的统一求解. AntTPN 改变了原有 Petri 网模型的选择行为, 使之具备了智能寻优的选择特点, 因此, 模型的优化求解无需构造状态空间进行搜索. 与现有可达性分析方法、 L_1 启发式调度算法、遗传算法相比, AntTPN 方法在 Petri 网调度解的质量、运行时间和空

间上具有一定优势. 我们实现了基于 Matlab 平台的 AntTPN 分析工具, 它具备图形建模和调度优化功能. AntTPN 工具网址为: <http://61.187.92.238:8105/html/kexueyanjiu/2013/0614/1053.html>.

在很多实际应用中, 往往涉及多个性能指标的优化, 例如, 执行时间、服务开销等. 进一步工作将考虑引入多种群蚁群机制, 研究基于 Petri 网的多目标优化问题.

参考文献

- [1] 江志斌. Petri 网及其在制造系统建模与控制中的应用 [M]. 北京: 机械工业出版社, 2004.
Jiang Z B. Petri Nets and Their Applications to Modeling and Control of Manufacturing Systems [M]. Beijing: China Machine Press, 2004. (in Chinese)
- [2] 薛雷, 郝跃. 面向集成电路制造的基于 Petri 网的生产调度 [J]. 电子学报, 2001, 29 (8): 1064 – 1067.
Xue Lei, Hao Yue. Petri net based scheduling for integrated circuits manufacturing [J]. Acta Electronica Sinica, 2001, 29 (8): 1064 – 1067. (in Chinese)
- [3] Lee D Y, DiCesare F. Scheduling flexible manufacturing systems using Petri nets and heuristic search [J]. IEEE Trans Robotics and Automation, 1994, 10(2): 123 – 132.
- [4] Wu N Q, Zhou M C, Chu F. A Petri net-based heuristic algorithm for realizability of target refining schedule for oil refinery [J]. IEEE Trans Automation Science and Engineering, 2008, 5 (4): 661 – 676.
- [5] Lee J, Lee J S. Heuristic search for scheduling flexible manufacturing systems using lower bound reachability matrix [J]. Computers and Industrial Engineering, 2010, 59 (59): 799 – 806.
- [6] 郝东, 蒋昌俊, 林琳. 基于 Petri 网与 GA 算法的 FMS 调度优化 [J]. 计算机学报, 2005, 28(2): 201 – 208.
Hao D, Jiang C J, Lin L. Petri net based modeling and GA based scheduling for FMS [J]. Chinese Journal of Computers, 2010, 28(2): 340 – 344. (in Chinese)
- [7] 曹政才, 余红霞, 乔非. 基于 Petri 网与遗传算法的半导体生产线建模与优化调度 [J]. 电子学报, 2010, 38 (2): 340 – 344.
Cao Zheng-cai, Yu Hong-xia, Qiao Fei. Petri-net and GA-based approach to modeling and optimize for semiconductor Wafer fabrication [J]. Acta Electronica Sinica, 2010, 38 (2): 340 – 344. (in Chinese)
- [8] Chiang T C, Huang A C, Fu L C. Modeling, scheduling, and performance evaluation for wafer fabrication; a queuing colored Petri-net and GA-based approach [J]. IEEE Transactions on Automation Science and Engineering, 2006, 3(3): 330 – 338.
- [9] Ciufudean C, Graur A, et al. A new formalism for failure diag-

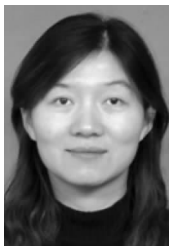
nosis: Ant colony decision Petri nets [J]. Journal of Software, 2007, 2(1): 39 – 46.

- [10] 邵志芳, 刘仲英. 半导体制造系统基于资源的 Petri 网调度优化 [J]. 同济大学学报, 2008, 36(5): 674 – 679.
Shao Z F, Liu Z Y. Resource-based Petri-net scheduling optimization for Wafer fabrication systems [J]. Journal of Tongji University, 2008, 36(5): 674 – 679. (in Chinese)
- [11] 黄光球, 张斌. 基于变迁可靠性的随机 Petri 网最优路径序列寻找 [J]. 系统工程与电子技术, 2009, 31 (1): 187 – 194.
Huang G Q, Zhang B. Optimum route sequence search in stochastic Petri net based on reliability of transition [J]. Systems Engineering and Electronics, 2009, 31(1): 187 – 194. (in Chinese)
- [12] Liu Xian-ming, Pan Li, Zheng Hong. Schedule optimization of time Petri nets based on ant colony systems [J]. Applied Mechanics and Materials, 2013, (263 – 266): 1733 – 1739.
- [13] Dorigo M, Stutzle T. Ant Colony Optimization [M]. Cambridge, MA: MIT Press, 2004.
- [14] 潘理, 丁志军, 郭观七. 混合语义时间 Petri 网模型 [J]. 软件学报, 2011, 22(6): 1199 – 1209.
Pan L, Ding Z J, Guo G Q. Time Petri net model with mixed semantics [J]. Journal of Software, 2011, 22(6): 1199 – 1209. (in Chinese)
- [15] Chen W N, Zhang J. An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements [J]. IEEE Trans Syst, Man, Cybern C, Appl Rev, 2009, 39(1): 29 – 43.
- [16] Lei D M. Interval job shop scheduling problems [J]. Intl Journal Adv Manuf Technol, 2012, 60(60): 291 – 301.

作者简介



潘 理 男, 1975 年生, 湖南平江人. 博士, 湖南理工学院副教授. 研究方向为 Petri 网、工作流、计算智能等.
E-mail: panli.hnist@gmail.com



郑 红 女, 1973 年生. 博士, 华东理工大学副教授, 美国加州大学河滨分校访问学者. 研究方向为普适计算、Petri 网应用.