

Petri 网融合蚁群算法的柔性装配系统调度

朱 伟, 徐克林, 蒋亚南

(同济大学, 机械工程学院, 上海 201804, zhwhliuhui@163.com)

摘 要: 为解决柔性装配系统的调度问题, 采用分层时延 Petri 网模型与蚁群优化算法相结合, 将柔性装配系统分解成诸多个装配体, 每个装配体再分解成多个柔性装配单元, 对装配单元分别建模并用蚁群算法对装配方案遍历优化, 最后确定装配系统的时延 Petri 网模型. 该方法融合了 Petri 网和蚁群算法各自的优点, 较好地解决了柔性装配系统中的装配建模和装配任务的分配优化问题. 实验证明该研究可行有效.

关键词: 分层模型; 时延 Petri 网; 蚁群优化算法; 柔性装配系统

中图分类号: TP205, TP301

文献标志码: A

文章编号: 0367-6234(2010)05-0832-04

Study of FAS scheduling based on fusion of Petri net and ant colony algorithm

ZHU Wei, XU Ke-lin, JIANG Ya-nan

(College of Mechanical and Electrical Engineering, Tongji university, Shanghai 201804, China, zhwhliuhui@163.com)

Abstract: To solve the scheduling problem in flexible assembly system (FAS), the method combining hierarchical timed-delay Petri net (HTPN) model with ant colony optimization algorithm (ACOA) is adopted. In this method, the FAS is divided into a number of subassemblies and then into many flexible assembly units. With building sub-model for each unit separately and assembly scheme being traversed, the final model is determined. The method has combined the strongpoint of ACOA and HTPN and solved the problems of assembly modeling and facilities assignments optimization of FAS. The experiments show that the method is effective and can be used to guide the assembly system design.

Key words: hierarchical model; timed-delay Petri net (TPN); ant colony optimization algorithm (ACOA); flexible assembly system (FAS)

装配系统的调度问题是一个 NP 难题^[1], 人们至今未曾有较完善的解决方案. Petri 网 (Petri Net, PN) 在柔性装配系统 (FAS) 的建模和设备调度中得到广泛应用^[2-3]. 特别是近年来, Petri 网和人工智能算法的融合已成为解决 FAS 调度问题的前沿思路^[4-6].

在前述研究的基础上, 本文用分层时延 Petri 网 (HTPN) 建立 FAS 的设备调度任务执行过程模型 (简称 HTPN 模型), 用蚁群优化算法优化任务调度, 以有效节约装配时间.

1 蚁群优化算法 (ACOA)

该蚁群优化算法 (ACOA) 以最大最小化蚁群

优化算法 (MMAS) 为基础^[7-9], 修改了全局更新规则. 在所有蚂蚁完成一次循环后, 按最优最差蚁群算法 (BWAS) 增加了对最差蚂蚁路径的信息素更新. 其数学模型如下:

$$h = \max_{u \in a_k} \{ [\tau(i, u)]^\alpha * [\eta(i, u)]^\beta \}, \quad (1)$$

$$p_{ih}^k(t) = \tau_{ih}^\alpha(t) \eta_{ih}^\beta(t) / \left(\sum_{u \in a_k} \tau_{iu}^\alpha(t) \eta_{iu}^\beta(t) \right), \quad (2)$$

$$\tau_{ih}(t+1) = (1 - \rho) * \tau_{ih}(t), \quad (3)$$

$$\tau_{ih} = \tau_{ih} + 1 / (f(s^{\text{best}})), \quad (4)$$

$$\tau_{ih} = \tau_{ih} - \varepsilon * f(s^{\text{worst}}) / f(s^{\text{best}}), \quad (5)$$

$$\tau_{ih}^*(t) = \tau_{ih}(t) + \delta * (\tau_{\max}(t) - \tau_{ih}(t)). \quad (6)$$

算法中各符号定义如下:

$u \in a_k$ 是蚂蚁 k 下一步允许选择的位置; i, h 分别表示蚂蚁当前位置和下一步要转移的位置; $p_{ih}^k(t)$ 表示 t 时刻, 蚂蚁 k 在位置 i 选择位置 h 的转移概率; ρ 是 $[0, 1]$ 区间内均匀分布的随机数, 由

$rand()$ 函数产生; $0 \leq Q \leq 1$ 是一给定参数; τ_{ih} 是边弧 (i, h) 上的信息素轨迹强度; η_{ih} 是能见度常量; $f(s^{best})$ 与 $f(s^{worst})$ 分别表示全局最优解与最差解; α, β, ρ 分别是信息素累积系数、启发系数与衰减系数; ε, δ 是调整信息素的常量因子. 下面介绍算法具体步骤.

Step1 初始化所有路径上的信息素为上限值(最大值) τ_{max} ;

Step2 以随机方式把蚂蚁分布在各路径初始端,每只蚂蚁按如下规则选择路径: If $\rho \leq Q$, 依据式(1)移至 h , else 以式(2)概率移至 h ;

Step3 连续执行 Step2 至一只蚂蚁找出一路径,记录该路径并换算为一组参数;

Step4 循环执行 Step2 和 Step3 直到所有蚂蚁都生成一条路径;

Step5 按式(3)更新所有路径上的信息素,评选出最优和最差蚂蚁,再分别按式(4)和(5)全局更新. 当更新某路径 τ_{ih} 时有 $\tau_{ih} < \tau_{min}$, 令 $\tau_{ih} = \tau_{min}$, 当 $\tau_{ih} > \tau_{max}$ 时, 令 $\tau_{ih} = \tau_{max}$;

Step6 循环执行 Step2 ~ Step5, 当迭代到规定次数或连续若干迭代而解没有变化时,所有路径上信息素按式(6)进行平滑处理;

Step7 循环迭代 Step2 ~ Step6, 直到达到规定迭代次数或连续若干迭代没有更好的解出现为止.

2 系统分层时延 Petri 网建模

2.1 时延 Petri 网基本原理简介^[10]

时延 Petri 网(TPN)可看作七元组, 设 $TPN_d = (P, T, I, O, M, M_0, d)$, 其中 P 为库所集; T 为变迁集; $I: P \times T \rightarrow N$ 是 P 到 T 的输入函数; $O: T \times P \rightarrow N$ 是 T 到 P 的输出函数; $N = (P, T, I, O, M_0)$ 是基本 Petri 网; M 是系统状态标志, 用各库所 P_i 包含的托肯数表示, 定义为行向量 $M = [M(P_1, \dots, P_m)]$; $i = 1, 2, \dots, m, m$ 为网图包含的库所节点数目; M_0 是系统初始状态标志. d 定义为 $d = T \rightarrow d(t_i)$, $d(t_i)$ 为完成变迁 t_i 所用的时间. 时延 Petri 网的运行规则如下:

①变迁只发生于相异库所之间; ②对任一变迁 t_i , 如果其输入库所 p_i 中的托肯数 $M(p_i) > 0$, 则该变迁是使能的; ③当条件满足后, 使能变迁激发, 可以输出多个托肯到不同库所去, 允许并发变迁; ④在变迁激发过程中, 有一段停留时间, 在这段时间内, 变迁无输出响应; ⑤一次变迁结束后, 系统形成新的状态标志 M' .

2.2 分层时延 Petri 网建模过程

分层建模思想是将一个装配系统自上而下分

解成诸多个装配体和装配单元, 对装配单元分别建模, 再自下而上整合成装配系统 Petri 网模型.

2.2.1 装配元的 Petri 网建模

本文将实现相异库中托肯的装配操作总称为一个装配元, 为便于说明, 记变迁 $t_i (i = 1, 2, 3, \dots)$ 对应的装配元为 $u_i (i = 1, 2, 3, \dots)$. 图 1 是装配体 A 的零件装配顺序 Petri 网图, 它给出一个满足工艺约束的零件装配顺序. 图 2 为装配元的 Petri 网模型.

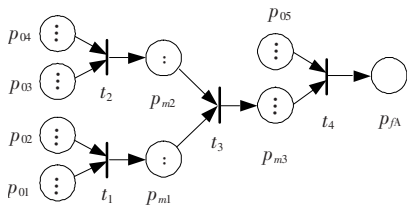


图1 装配体 A 的 Petri 网模型

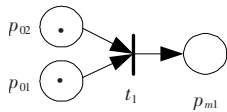


图2 装配元的 Petri 网模型

在图 1 中, 有库所、变迁、弧和托肯 4 种元素. 其中, 库所可分为 3 种: ① $p_{oi} (i = 1, 2, 3, \dots)$ 是初始装配库所, 库所里的托肯数表示可参与装配的零件的数量; ② $p_{mi} (i = 1, 2, 3, \dots)$ 是中间装配库所, 库所中的托肯数表示完成前序装配并等待后序作业的的零部件数量; ③ p_{fa} 为完成装配库所, 库所中的托肯数表示装配结束得到的装配体 A 的数量.

2.2.2 装配元的时延 Petri 网建模

在时延 Petri 网模型中, 要具体描述装配体的各个装配元由那些设备来完成, 设备如何配置以及设备完成各装配元的装配任务分别需要的时间. 下面以装配体 A 为例, 说明时延 Petri 网建模过程.

Step 1 建立装配元延时变迁时间表, 如表 1 所示. 在表 1 中 u_1, u_2, u_3 和 u_4 是指与图 1 对应的

表 1 装配元延时变迁时间表 A

装配单元	单元延迟时间	装配设备延迟时间 t/min			
	t/min	e_1	e_2	e_3	e_4
u_1	t_1	$t_{13} = 9$	$t_{12} = 3 \quad t_{11} = 9$		$t_{14} = 6$
u_2	t_2	$t_{21} = 8$			
		$t_{22} = 4$			
u_3	t_3	$t_{31} = 5$		$t_{32} = 12$	
		$t_{41} = 6$		$t_{42} = 8$	
u_4	t_4	$t_{43} = 2$		$t_{44} = 9$	

装配元, e_1, e_2, e_3, e_4 表示 FAS 中的 4 台装配设备. 有 3 种设备配置方式可以分别独立完成装配元 u_1 , 分别是: ① e_3 用 9 个单位时间完成部分 u_1 , 再由 e_2 用 3 个单位时间最终完成 u_1 , 对应的延时变迁分别为 t_{11} 和 t_{12} ; ② e_1 用 9 个单位时间完成 u_1 , 对应的延时变迁为 t_{13} ; ③ e_4 用 6 个单位时间完成 u_1 , 对应的延时变迁为 t_{14} .

Step 2 建立装配元变迁的 TPN 模型, 如图 3(a) 所示. 在图 3(a) 中, 用 TPN 网表示装配元 u_1 的 3 种可供选择设备配置方案 (见表 1), $p_{e1}, p_{e2}, p_{e3}, p_{e4}$ 为对应 e_1, e_2, e_3, e_4 的设备库所, 它们中各自的托肯数量表示可供选择的设备数. $p_{s11}, p_{s12}, p_{s13}$ 是装配元 u_1 的 3 个设备方案选择库所, 这 3 个库所构成了装配元 u_1 的选择库所集. 其中, 托肯数为 1 的选择库所对应的设备方案被选择执行任务. 在完成一个装配元的过程中, 该装配元的设备选择库所中只能有一个托肯数被置 1, 其余必须始终为 0. 变迁 $t_{11}, t_{12}, t_{13}, t_{14}$ 为装配元 u_1 的 4 个延时变迁, 分别表示各台设备在参与装配元 u_1 时耗用的时间, 具体数值标注在 Petri 网图相应的延时变迁弧线旁.

Step 3 拓展实施装配顺序 Petri 网图中各装配元对应的变迁, 如图 3(c) 是对图 3(b) 进行拓展而得. 在图 3(c) 中, t_{1p} 是将装配元 u_1 对象准备变迁, p_{1r} 是 u_1 准备状态库所. p_{1r} 中的托肯数为 1 表示装配对象已准备好, 托肯数为 0 表示装配对象未准备好.

Step 4 联合图 3(a) 和图 3(c), 得到装配元 u_1 拓展了的 TPN 模型^[11], 如图 3(d) 所示.

2.2.3 整合所有装配元的 TPN 模型, 从而得到装配体的 TPN 模型.

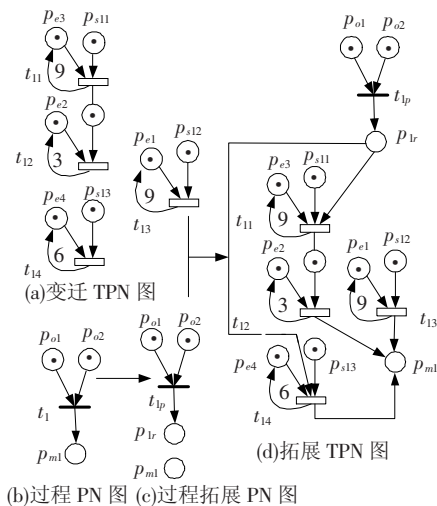


图 3 装配元延时 Petri 网模型

3 HTPN 和蚁群优化接口算法 (HTPN – ACOA)

软件流程如图 4 所示.

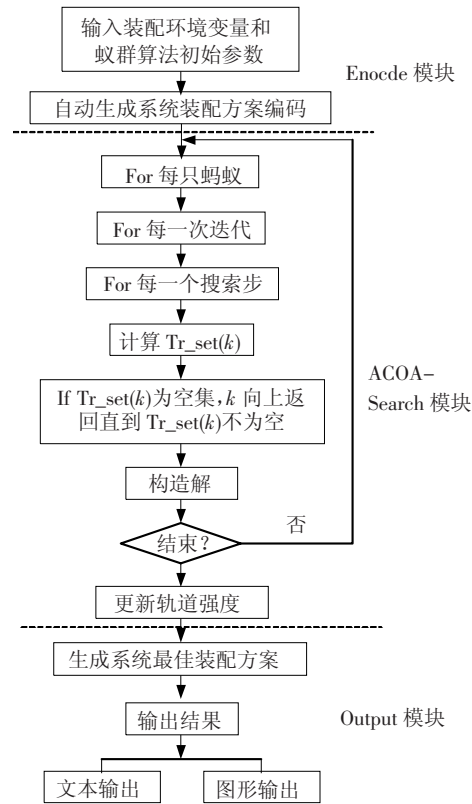


图 4 算法流程图

3.1 Encode 模块

软件的 **Encode** 模块用于自动生成系统装配方案编码库, 该模块的数据输入窗体用以输入系统装配环境变量和蚁群优化算法参数的初始值. 其中装配环境变量主要有装配设备代码、装配顺序及各变迁的时延数值等.

3.2 ACOA-Search 模块

该模块功能是用 ACOA 算法完全遍历 **Encode** 模块中生成的装配方案空间, 寻找满足要求的最短时间装配方案. 其中, $Tr_set(k)$ 是第 k 个搜索步可能实施的时延变迁集合^[12].

3.3 Output 模块

Output 模块将优化结果以不同的形式输出, 包括文本输出和图形输出两种形式. 其中, 文本输出是将计算结果输出到文本文件, 包括装配件的装配顺序, 设备占用情况, 对装配单元每个装配操作的开始时间、结束时间等, 对文本文件可进行保存、打开、打印、编辑等操作. 图形输出则是以甘特图的形式输出优化调度结果.

4 案例与分析

一柔性装配车间由 4 台工业机器人、2 个转盘

式输送平台、2 台自动小车和 4 台计算机组成,上述装配设备分别用 e_1, e_2, e_3, e_4 表示. 两台工业机器人用以完成拧、插、压等不同的装配操作,自动小车可以完成部分部件的装卸,各计算机之间用以太网连接,计算机与各操作设备之间通过 RS232 协议通讯用串口连接. 软件系统在 dell 1 000 MB RAM/2 GHz PC 机上运行,各装配设备相互分工,协同工作. 装配体 A 的分层 TPN 图及其装配元时延变迁时间关系表分别如图 3 和表 1 所示,装配体 B 的装配元时延变迁时间表如表 2 所示.

本例中,算法中各参数取值如下: $\alpha, \beta \in [1, 3]; \rho \in [0.5, 0.8]; Q \in [1, 9\,999]$. 在 HT-PN-ACOA 软件输入窗体中输入装配环境参数和 ACOA 算法中的参数初始值,在 PC 机上运行算法,30 次实验中有共有 26 次装配时间为 33 min, 3 次装配时间为 38 min, 1 次装配时间为 41 min. 它所对应装配的设备分配甘特图如图 5 所示. 从表 3 可以看出,本文算法在计算效率和优化程度上均比文献[11]和[13]提出的算法有所提高.

表 2 装配元延时变迁时间表 B

装配单元	单元延迟时间 t/min	装配设备延迟时间 t/min			
		e_1	e_2	e_3	e_4
u_5	t_5	$t_{51} = 5$		$t_{52} = 3$	
			$t_{53} = 4$		$t_{54} = 6$
u_6	t_6	$t_{61} = 4$			$t_{62} = 6$
			$t_{63} = 3$		$t_{64} = 4$
u_7	t_7	$t_{71} = 4$		$t_{72} = 4$	
		$t_{73} = 5$		$t_{74} = 12$	
u_8	t_8	$t_{75} = 3$	$t_{76} = 4$		
			$t_{81} = 2$		$t_{82} = 9$

表 3 算法优化程度比较

算法	装配时间/min	装配时间 产生次数	运行时间/min
本文算法	33	26	4.5
	38	3	
	41	1	
文献[11]算法	37	29	4.7
	39	1	
文献[13]算法	37	7	4.7
	40	23	

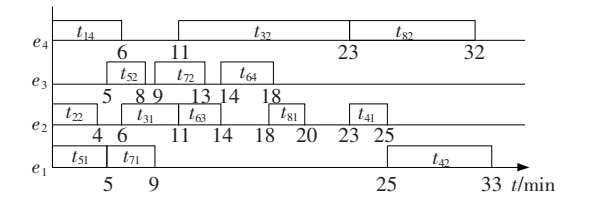


图 5 优化调度方案甘特图

5 结 论

1) 基于 Petri 网和蚁群优化算法融合的调度

优化方法实现了并发/异步功能,并具有路径柔性化,防止调度死锁和有效节约装配时间的优点.

2) 与已有的 Petri 网结合 AI 求解调度优化问题的研究相比较,本方案算法简单,易于实现,且在优化程度和运算速度方面存在优势.

参考文献:

[1] CARLOS R, JOAO R, ZITA V. On the complexity of precedence graphs for assembly and task planning[J]. Computers in Industry, 1998, 36(1/2): 101 - 111.

[2] 高峰,王明哲. 装配生产线 Petri 网模型优化算法研究[J]. 机械与电子, 2004(5): 13 - 15.

[3] 张伟军,贾传圣,杨汝清. 基于延时 Petri 网模型的机器人装配任务规划[J]. 中国机械工程, 2002, 13(14): 1219 - 1222.

[4] YUH, REYES A, CANG S, *et al.* Combined petri net modeling and AI-based heuristic hybrid search for flexible manufacturing systems-part I, Petri net modeling and heuristic hybrid search[J]. Computers and Industrial Engineering, 2003, 44(4): 527 - 543.

[5] REYES A, YU H, KELLEHER G, *et al.* Integrating petri nets and hybrid heuristic search for the scheduling of FMS [J]. Computers in Industry, 2002, 47: 123 - 138.

[6] MEJIA G. Timed Petri net modeling and optimization with heuristic search for flexible manufacturing workstations[C]// Proceedings of the 9th IEEE International Conference on Emerging Technologies and Factory Automation. Lisbon, Portugal: [s. n.], 2003: 211 - 217.

[7] DORIGO M, DICARO G. The ant colony optimization: a new meta-heuristic [C]//Proceedings of the IEEE international conference on evolutionary computation. Piscataway, USA: IEEE Press, 1999: 1470 - 1477.

[8] DORIGOM, DICARO G, GAMBARDELLA LM. Ant algorithms for discrete optimization [J]. Artificial Life, 1999, 5(2): 137 - 172.

[9] 李士勇. 蚁群算法及其应用[M]. 哈尔滨: 哈尔滨工业大学出版社, 2004: 22 - 52.

[10] GUL A A, FIORELLA D C, GRZEGORZ R. Concurrent object-oriented programming and Petri net [M]. Berlin: Germany Springer, 2001: 197 - 254.

[11] 刘维来,孔凡让,刘志刚,等. 基于遗传算法和延时 Petri 网的柔性装配系统的设备调度方法[J]. 计算机集成制造系统, 2006, 12(8): 1247 - 1248.

[12] 邵志芳,刘仲英,钱省三. 整合 Petri 网和蚁群算法用于柔情制造系统高度优化研究[J]. 计算机应用, 2006, 26(11): 2753 - 2754.

[13] VALLE C D, CAMACHO E F. Automatic assembly task assignment for a multiple-robot environment [J]. Control Engineering Practice, 1996, 14(7): 915 - 921.

(编辑 杨波)