

一种基于 Petri 网的多目标无死锁蚁群调度算法

惠晓龙¹, 郜振鑫²

(1. 中国电子科技集团公司第 20 研究所 通信部, 陕西 西安 701168; 2. 西安交通大学 系统工程研究所, 陕西 西安 710068)

摘要 蚁群算法已被广泛应用于解决调度问题, 针对资源受限的项目调度问题, 死锁和多目标问题是重要的课题研究。文中针对具有死锁资源受限的多目标调度问题, 采用 Petri 网模型, 研究其蚁群调度算法, 以紧急项目的完成时间和所有项目的完成时间最短为优化目标, 并通过仿真实验验证了文中算法的有效性和实用性。

关键词 Petri 网; 蚁群算法; 调度

中图分类号 TN301.6 **文献标识码** A **文章编号** 1007-7820(2014)05-179-03

Research on Multi-objective and Deadlock-free Ant Colony Optimization Scheduling Algorithm with Petri Nets

HUI Xiaolong¹, GAO Zhenxin²

(1. Division of Communications, No. 20 Institute of China Electronic Technology Group Corporation, Xi'an 710068, China;

2. Systems Engineering Institute, Xi'an Jiaotong University, Xi'an 710068, China)

Abstract The ant colony algorithm has been widely used to solve scheduling problems. Deadlocks and multi-objective problems are important researches for Resource Constrained Project Scheduling Problem (RCPSPP). This paper studies the RCPSPP with deadlocks and multi-objective problems. Petri net model is established to study the ant colony scheduling algorithm. Optimization goal is the shortest time of finishing emergency project and all projects. Simulation results show the effectiveness and usefulness of the algorithm.

Keywords Petri nets; ant colony algorithm; scheduling

资源受限的项目调度问题(Resource Constrained Project Scheduling Problem, RCPSPP)属于 NP-hard 问题;广泛存在于制造系统等调度问题,它通过合理地分配资源,达到项目执行工期最短等优化目标^[1]。在合理分配资源的过程中,死锁是一个重要的研究课题,目前人们从控制的角度提出了多种死锁避免控制策略,但是这些策略无法优化系统的运行性能,因此系统的无死锁优化调度是一个值得研究的问题^[2-5]。传统的优化调度只有一个优化目标,而在实际项目中往往存在多个优化调度目标。

本文针对制造系统的赋时 Petri 网模型,研究以总工期最短和紧急项目工期最短为优化目标的无死锁调度问题,建立系统的多目标无死锁蚁群调度算法。蚁群算法采用最大最小蚂蚁系统,通过修改状态转移概率公式,并给出算法最优解的评价函数,使蚂蚁综合多个优化目标选择路径。单只蚂蚁选择下一步可执行的操作时,进行死锁判断,将死锁避免控制策略嵌入到算

法,实现无死锁调度。仿真结果验证了本文提出的多目标无死锁蚁群调度算法的可行性和有效性。

1 系统的 Petri 网模型

Petri 网^[2]结构是一个三元组 $N = (P, T, F)$, 其中 P 和 T 都是有限但互不相交的集合。 P 是位置(Place)的集合, T 是变迁(Transition)的集合, $F \subseteq (P \times T) \cup (T \times P)$ 是有向弧(Arc)的集合。

Petri 网 N 的一个标识或状态是一个映射 $M: P \rightarrow Z^+$, 其中 $Z^+ = \{0, 1, 2, \dots\}$ 。给定标记 Petri 网 $N = (P, T, F, M_0)$, 用 T^* 表示 T 的所有变迁序列的集合, $\forall a = t_0, t_1, t_2, \dots, t_n \in T^*$, 如果 $M_i t_i > M_{i+1}, i = 0, 1, 2, \dots, n$, 则称 a 为在 M_0 下的可行序列, 称 $M_i (i = 1, 2, \dots, n+1)$ 为从 M_0 的可达标识或状态。Petri 网 N 中的一条路径是一个序列 $\alpha_w = (u = x_1, x_2, \dots, x_{k+1} = v)$, 其中 $x_i \in P \cup T, (x_i, x_{i+1}) \in F, i = 1, \dots, k, k$ 是 α 的长度。

本文采用位置赋时 Petri 网模拟系统的特征,模型中一个标记必须在位置中经过一定时延,记作 $d(p)$, 才能进入下一个位置。

采用文献[2, 4]给出的制造系统模型,如图 1 所示。系统中有 4 类机器(m_1, m_2, m_3 和 m_4)和 3 类机器人(r_1, r_2 和 r_3), 把资源集合表示为 $r = \{m_1, m_2, m_3,$

收稿日期: 2014-02-17

作者简介: 惠晓龙(1988—), 男, 硕士, 工程师。研究方向: 智能优化算法和通信。E-mail: 191735880@qq.com。郜振鑫(1987—), 男, 博士研究生。研究方向: 制造系统的控制, 优化调度等。

$m_4, r_1, r_2, r_3\}$, $C(m_i) = 2, i = 1, 2, 3, 4, C(r_i) = 1, i = 1, 2, 3$ 。系统需要加工处理3类工件,加工 q_1 类工件的资源顺序是 $r_2 m_2 r_2$,加工 q_3 类工件的资源顺序是 $r_3 m_4 r_2 m_3 r_1$,加工 q_2 类工件有两条路径,其资源顺序分别是 $r_1 m_1 r_2 m_2 r_3$ 和 $r_1 m_3 r_2 m_4 r_3$ 。 t_{ij} 和 p_{ij} 中的字母 i 代表第 i 条路径, q_1 类工件的加工路径为 $p_{11} t_{11} p_{11} t_{12} p_{12} t_{13} p_{13} t_{14} p_{14}$, q_2 类工件的加工路径有两条分别为 $p_{21} t_{21} p_{21} t_{22} p_{22} t_{23} p_{23} t_{24} p_{24} t_{25} p_{25} t_{26} p_{26}$ 和 $p_{21} t_{21} p_{21} t_{32} p_{32} t_{33} p_{33} t_{34} p_{34} t_{35} p_{35} t_{36} p_{36}$, q_3 类工件的加工路径为 $p_{41} t_{41} p_{41} t_{42} p_{42} t_{43} p_{43} t_{44} p_{44} t_{45} p_{45} t_{46} p_{46}$ 。3类工件需要加工的个数分别为8,12和8。

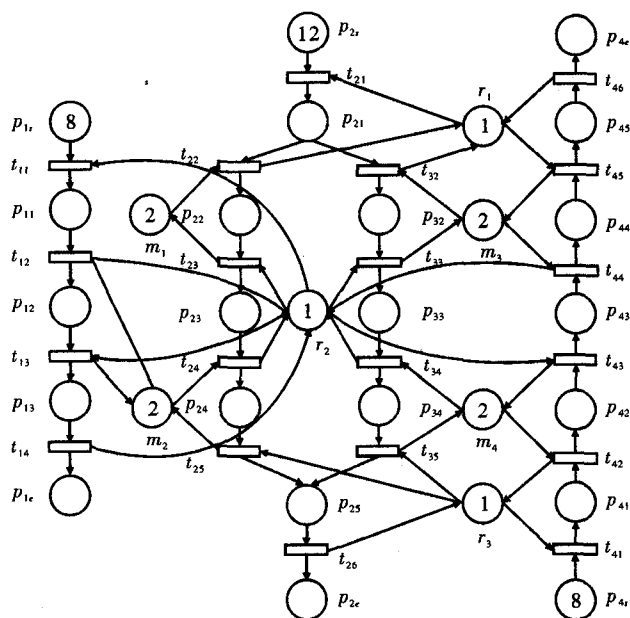


图1 系统的Petri网模型

2 多目标无死锁蚁群调度算法

2.1 算法简介

本文的蚁群算法采用最大最小蚂蚁系统(Max - Min Ant System, MMAS),其主要思想是:仅更新每一代中最好的蚂蚁个体所走的路径上的信息量,加快收敛速度;并将各条路径上的信息量限制在区间 $[\tau_{\min}, \tau_{\max}]$ 内,避免某条路径上的信息量远大于其他路径上的信息量,使蚂蚁的搜索行为集中到同一条路径上^[6]。

针对制造系统的调度模型提出蚁群算法的数学模型,问题描述如下:针对一个可标识的自动制造系统 (N, M_0) ,加工完一定任务的工件总共需要的操作数目为 $n = |O|$,所有加工操作的集合表示为 $\{O_1, O_2, \dots, O_n\}$,操作 O_i 的加工时间为 $t(O_i)$ ($i = 1, 2, \dots, n$),算法的任务就是寻找一个有序的加工路径或者操作序列,完成所有的加工任务,优化目标使系统的加工时间和紧急工件的加工完成时间最短。算法对系统的工件、资源和操作均按顺序采用10进制进行编码。

2.2 多目标无死锁蚁群调度算法

算法中蚁群数量为 m ,蚂蚁 k ($k = 1, 2, \dots, m$)在选择下一步操作时,根据文献[5]中提出的具有多项式复杂性的死锁避免控制策略(DAP),将不会造成死锁的操作放入集合 allowed_k 中,并从中选择下一步遍历操作,从而实现无死锁调度。在遍历的过程中,根据各操作之间遗留的信息素来决定下一个操作,同时考虑操作的启发式信息和紧急工件的优先级信息。 P_{ij}^k 表示第 k 只蚂蚁由操作 O_i 转移到操作 O_j 的状态转移概率,由式(1)确定

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_j]^\beta \cdot [\zeta_j]^\gamma}{\sum_{s \in \text{allowed}_k} [\tau_{is}]^\alpha \cdot [\eta_s]^\beta \cdot [\zeta_s]^\gamma}, & j \in \text{allowed}_k \\ 0, & \text{其他} \end{cases} \quad (1)$$

其中, τ_{ij} 表示操作 O_i 和 O_j 之间的信息素,初始时刻为常数; η_j 表示在搜索过程中选择操作 O_j 的期望值,根据先验知识可知,本文 η_j 为操作 O_j 的最早开始加工时间的倒数; ζ_j 表示操作 O_j 对应工件的优先级,初始给定为常数。

α 表示信息素启发式因子,反映的是蚁群在遍历过程中所残留信息素的相对重要程度; β 表示期望启发式因子,反映了对下一个遍历操作所期望的相对重要程度; γ 表示优先级启发式因子,反映优先级的相对重要程度。如式(1)所示,蚂蚁优先选择信息素浓度高、能最早开始加工并且优先级高的操作路径。

算法每一次迭代只更新最优解路径上的信息素,因此给出了评价最优解的函数。假设系统中只含有一类紧急工件,其他工件都是普通工件,用式(2)的大小作为评价最优解 α 的标准,其中 ω_1 和 ω_2 分别是系统加工时间 t_1 和紧急工件加工时间 t_2 的权值。 ζ 表示紧急工件的优先级,finishtime表示紧急工件的加工完成时间。

$$\text{best}(\alpha) = \omega_1 \cdot \frac{1}{t_1} + \zeta \cdot \omega_2 \cdot \frac{1}{t_2} \quad (2)$$

为防止算法过早收敛陷入局部最优解,当算法收敛时重置信息素模型,重启调度。用收敛因子 c_f 描述算法的收敛程度;在初始时刻 $c_f = 0$,当所有的信息素的值都等于 τ_{\max} 或 τ_{\min} 时, $c_f = 1.0$,由式(3)求得。

$$c_f = 2 \cdot \left(\frac{\sum \max(\tau_{\max} - \tau_{ij}, \tau_{ij} - \tau_{\min})}{|\tau| \cdot (\tau_{\max} - \tau_{\min})} - 0.5 \right) \quad (3)$$

算法实现步骤:(1)系统初始化。蚁群中蚂蚁的总数,最大循环迭代次数,信息素初始化 $\tau_{ij} = 0.5$ 。(2)单只蚂蚁调度。首先根据死锁避免控制策略,保证每只蚂蚁在选择下一个操作之前,保证操作不会造成死锁,然后根据状态转移概率公式选择下一步操作,直到遍

历完所有操作。(3)所有蚂蚁遍历完形成路径集合,从中选择最优路径更新信息素模型。(4)计算收敛因子,如果算法收敛后,重置信息素模型。(5)判断是否满足结束条件,即达到最大循环迭代次数,如果满足,则输出算法计算结果,算法结束。

3 算法仿真与结果分析

对图 1 所示的制造系统的调度问题,采用蚁群算法对其进行调度,采用文献[7]中给出的加工时间数据,加工时间如表 1 所示。

表 1 制造系统操作加工时间

q_1		q_2		q_3	
w_1		w_2		w_3	w_4
$O_{11}:8$		$O_{21}:4$		$O_{21}:4$	$O_{31}:5$
$O_{12}:34$		$O_{22}:32$		$O_{22}:23$	$O_{32}:22$
$O_{13}:5$		$O_{23}:8$		$O_{23}:6$	$O_{33}:4$
		$O_{24}:38$		$O_{24}:20$	$O_{34}:17$
		$O_{25}:5$		$O_{25}:5$	$O_{35}:6$

系统仿真参数设置如下。 q_1, q_2, q_3 这 3 类工件的优先级分别为(1,2,1), q_2 类工件为紧急工件, $\alpha=3$, $\beta=3$, $\gamma=1$, $\rho=0.01$, 蚁群数量为 10, 迭代次数 800, $\omega_1=0.4$, $\omega_2=0.6$ 。图 2 是迭代最优解变化情况, 图 3 是多目标全局最优解的变化情况, 求得所有工件完成时间 428, 紧急工件完成时间 259, 对应的最优调度路径为($t_{11}, t_{21}, t_{41}, t_{32}, t_{21}, t_{22}, t_{42}, t_{21}, t_{41}, t_{12}, t_{22}, t_{13}, t_{14}, t_{43}, t_{21}, t_{42}, t_{33}, t_{23}, t_{22}, t_{34}, t_{41}, t_{21}, t_{24}, t_{23}, t_{24}, t_{22}, t_{44}, t_{21}, t_{32}, t_{45}, t_{46}, t_{21}, t_{43}, t_{44}, t_{42}, t_{25}, t_{26}, t_{35}, t_{26}, t_{11}, t_{33}, t_{32}, t_{12}, t_{25}, t_{26}, t_{45}, t_{46}, t_{43}, t_{44}, t_{34}, t_{33}, t_{21}, t_{35}, t_{26}, t_{32}, t_{13}, t_{14}, t_{21}, t_{41}, t_{34}, t_{23}, t_{22}, t_{21}, t_{24}, t_{42}, t_{35}, t_{26}, t_{23}, t_{33}, t_{41}, t_{32}, t_{24}, t_{34}, t_{21}, t_{22}, t_{45}, t_{46}, t_{43}, t_{44}, t_{21}, t_{42}, t_{35}, t_{26}, t_{33}, t_{34}, t_{32}, t_{45}, t_{46}, t_{43}, t_{25}, t_{26}, t_{23}, t_{25}, t_{26}, t_{24}, t_{23}, t_{35}, t_{26}, t_{33}, t_{24}, t_{25}, t_{26}, t_{11}, t_{44}, t_{34}, t_{41}, t_{12}, t_{42}, t_{25}, t_{26}, t_{35}, t_{26}, t_{41}, t_{11}, t_{12}, t_{42}, t_{13}, t_{14}, t_{41}, t_{11}, t_{12}, t_{13}, t_{14}, t_{13}, t_{14}, t_{11}, t_{12}, t_{11}, t_{12}, t_{43}, t_{44}, t_{45}, t_{46}, t_{42}, t_{43}, t_{44}, t_{45}, t_{46}, t_{43}, t_{44}, t_{45}, t_{46}, t_{45}, t_{46}, t_{13}, t_{14}, t_{11}, t_{12}, t_{13}, t_{14}, t_{13}, t_{14}$)。

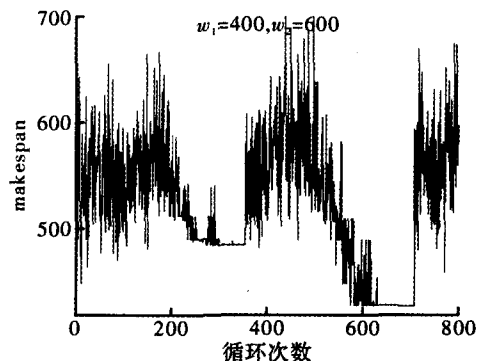


图 2 蚁群算法迭代最优解

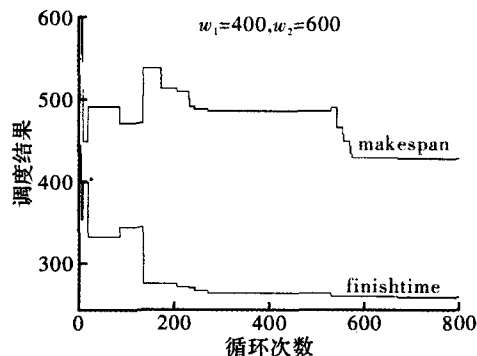


图 3 蚁群算法多目标调度最优解

从仿真结果可以看出, 蚁群算法在迭代次数为约 360 和 705 时收敛, 重置信息素模型重启调度, 以免算法陷入局部最优解。算法根据优化目标权值的不同, 做出权衡, 选择最优解, 从图中可以看出, 算法较好地给出了调度的最优解, 是可行、有效的。

4 结束语

本文针对资源受限项目中的制造系统调度问题, 采用其 Petri 网模型, 将死锁避免控制策略嵌入到蚁群算法中, 解决系统具有多个加工优化目标的问题, 给出系统的多目标无死锁蚁群调度算法。蚁群算法采用最大最小蚂蚁系统, 通过修改状态转移概率公式, 使蚂蚁具有较大的概率优先选择紧急工件的加工操作, 算法每次只更新最优调度解对应的信息素, 给出评价函数来评价调度解的优劣。通过仿真实例, 证明了算法的可行性和有效性。

参考文献

- [1] 庞南生, 孟俊姣. 多目标资源受限项目鲁棒调度研究[J]. 运筹与管理, 2012(3): 27-32.
- [2] EZPELETA J, COLOM J M, MARTINEZ J. A petri net based deadlock prevention policy for flexible manufacturing systems [J]. IEEE Transactions on Robotics and Automation, 1995, 11(2): 173-184.
- [3] REVELIOTIS S A, LAWLEY M A, FERREIRA P M. Polynomial-complexity deadlock avoidance policies for sequential resource allocation systems [J]. IEEE Transactions on Automatic Control, 1997, 42(10): 1344-1357.
- [4] XING K Y, ZHOU M C, LIU H X, et al. Optimal petri-net-based polynomial-complexity deadlock-avoidance policies for automated manufacturing systems [J]. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 2009, 39(1): 188-199.
- [5] 邢科义, 田锋, 杨小军, 等. 具有多项式时间复杂性的避免制造系统死锁控制策略[J]. 自动化学报, 2007, 33(8): 893-896.
- [6] STÜTZLE T, HOOS H H. Max-min ant system [J]. Future Generation Computer Systems, 2000, 16(9): 889-914.
- [7] XING K Y, HAN L B, ZHOU M C. Deadlock-free genetic scheduling algorithm for automated manufacturing systems based on deadlock control policy [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2012, 42(3): 603-615.