第41卷 第3期 西 安 交 通 大 学 学 报 Vol.41 №3
2007年3月 JOURNAL OF XI′AN JIAOTONG UNIVERSITY Mar. 2007

多设计任务调度的非合作博弈研究

张国海[1,2]，江平宇[1,2]，周光辉[1,2]
(1.西安交通大学机械制造系统工程国家重点实验室，710049，西安；
2.西安交通大学机械工程学院，710049，西安)

摘要：从客户竞争需求的角度出发，以提交的设计任务各自的设计时间最短为调度目标，采用博弈理论，提出并构建了一种面向多设计任务的非合作博弈调度模型.在该调度模型中，设计任务被映射为博弈模型的局中人，与设计子任务集对应的可选设计节点映射为各设计任务的可行方案集，各设计任务的设计完成时间的倒数映射为收益函数，将多设计任务的调度转化为多设计任务调度模型的Nash均衡点来求解问题，并采用遗传算法进行了解算.同时，以6个设计任务验证了该任务调度模型及算法的可行性，为解决多设计任务调度问题提供了一种新的思路.

关键词：非合作博弈；Nash均衡点；遗传算法；任务调度
中图分类号：TP273 **文献标识码**：A **文章编号**：0253-987X(2007)03-0303-04

Non-Cooperation Game for Multiple Design Task Schedule

Zhang Guohai[1,2], Jiang Pingyu[1,2], Zhou Guanghui[1,2]
(1. State Key Laboratory for Manufacturing Systems Engineering, Xi′an Jiaotong University, Xi′an 710049, China;
2. School of Mechanical Engineering, Xi′an Jiaotong University, Xi′an 710049, China)

Abstract: As viewed from customer competition requirements, a non-cooperation game model for multiple design task schedule is presented for the goal of minimized time of each design task submitted by the corresponding customer. In the schedule model, the players correspond to the multiple design tasks, strategies to the optional design nodes related to design sub-task sets, and the payoff of each task to the reciprocal of its finishing time. Thus, the optimal scheduling results are determined by the nash equilibrium (NE) point of this game, the genetic algorithm is introduced to seek out the NE point. A numerical case study is given to demonstrate the feasibility of the methods. The presented model and algorithm provide a new way to solve the multiple design task schedule problem.

Keywords: non-cooperation game; nash equilibrium point; genetic algorithm; task schedule

产品设计在时间上表现为一系列设计任务的串行、并发和交叉耦合，目前对于产品设计调度研究的核心聚焦在调度的总体最优目标上[1-3]，而忽略了各设计任务之间的竞争性因素.因此，寻求一种新的设计任务调度策略和模型以达到各客户之间的利益均衡，已成为目前急需解决的课题之一.

据此，本文采用博弈论理论[4-5]提出并构建了一种具备完全信息的非合作博弈的多设计任务调度模型，将来源于不同客户的设计任务及其包含的设计子任务集对应的可选设计节点，以及各设计任务的设计完成时间的倒数，分别映射为博弈模型中的局中人、可行方案集和收益函数.对该调度模型的求解可归结为寻求非合作博弈模型的Nash均衡点，而对Nash均衡点的具体求解可通过引进遗传算

收稿日期：2006-07-04. 作者简介：张国海(1976～)，男，博士生；江平宇(联系人)，男，教授，博士生导师. 基金项目：教育部新世纪优秀人才支持计划资助项目(NCET-04-0928)；机械制造系统工程国家重点实验室开放课题研究基金资助项目；教育部流程综合工业自动化重点实验室开放课题研究基金资助项目.

法[6-7]进行解算.本文提出的任务调度模型与算法,为解决客户驱动的多设计任务调度提供了方案和实现途径.

1 多设计任务调度描述

多设计任务调度的数学描述如下:

(1) 设客户提交的设计任务数为 n,每个设计任务由一系列设计子任务组成;

(2) 设与任务相关的设计节点数为 m,设计节点 c_j 能够同时进行的设计任务数为 s_j,其中 $1\leqslant j\leqslant m$;

(3) 设每个设计任务的加工完成时间为 t_i,$i=1,\cdots,n$,调度目标是在各设计任务展开竞争的前提下使得 t_i 最小.

同时,对设计任务调度作如下假设:

(1) 如果某个设计节点能同时进行 s 个设计,则把该节点分解为具有相同条件的 s 个节点;

(2) 所有的设计任务机会均等,不存在优先权;

(3) 同一时间内一个设计节点只能进行一个任务的设计;

(4) 将各设计节点之间的信息传递、交互及协同时间作为设计准备时间计算,在本文中把该类时间直接作为设计任务的持续时间.

基于上述数学模型和假设条件,客户驱动的多任务设计调度的目标是:寻求在各设计任务展开竞争的前提下,使得每个设计任务均能达到最短完成时间的利益均衡调度结果.

2 面向任务调度的非合作博弈模型

将任务调度的数学模型映射为非合作博弈模型,该模型及相关符号定义如下.

设多设计任务调度非合作博弈模型为三元组 $G=(N,P_i,W_i)$,其中:①N 为局中人(设计任务)集,记为 $N=\{N_1,N_2,\cdots,N_n\}$,共 n 个设计任务,N_i 包含 J_i 个设计子任务,记 $\omega_{i,j}$ 为 N_i 的第 j 个子任务,则 $N_i=\{\omega_{i,1},\omega_{i,2},\cdots,\omega_{i,J_i}\}$;②$P_i$ 为 N_i 的可行方案(对应于 N_i 相关的可选设计节点)集,假设 N 所对应的设计节点集为 C,而 $C=\{c_1,c_2,\cdots,\underbrace{c_{j_1},\cdots c_{j_{s_j}}}_{s_j},\cdots c_m\}$,共 $\sum_{j=1}^{m}s_j$ 个设计节点,即

$$P_i\subseteq C,\ C\equiv P=P_1P_2\cdots P_n$$

③W_i 为 N_i 的收益函数,其定义为 N_i 的 $\omega_{i,j}$ 在 $c_{j_k}(k\leqslant s_j)$ 上的设计时间为 $T_{\mathrm{P}_{ij}}(c_{j_k})$,设 $T_{\mathrm{S}_{ij}}(c_{j_k})$ 为 N_i 的 $\omega_{i,j}$ 在 $c_{j_k}(k\leqslant s_j)$ 上的开始设计时间,则 N_i 的收益函数为

$$W_i(p)=\frac{1}{T_{\mathrm{S}_{ij}}(c_{j_k})+T_{\mathrm{P}_{ij}}(c_{j_k})} \tag{1}$$

$$p=(p_1,\cdots,p_i,\cdots,p_n),\ p_i\in P_i$$

同一个设计节点在某一时刻只能进行一个设计,即

$$T_{\mathrm{S}_{ij}}(c_k)\geqslant T_{\mathrm{S}_{il}}(c_k)+T_{\mathrm{P}_{il}}(c_k) \tag{2}$$

式中:$T_{\mathrm{S}_{ij}}(c_k)$ 表示在 c_k 上进行设计的 ω_{ij} 的开始时间;$T_{\mathrm{S}_{il}}(c_k)$ 表示 ω_{ij} 在 c_k 上的上一个子任务在 c_k 上的开始时间;$T_{\mathrm{P}_{il}}(c_k)$ 表示 ω_{il} 在 c_k 上的设计持续时间.设计紧前紧后关系约束

$$T_{\mathrm{S}_{ij}}\geqslant\max(T_{\mathrm{S}_{im}}+T_{\mathrm{P}_{im}}) \tag{3}$$

式中:$\max(T_{\mathrm{S}_{im}}+T_{\mathrm{P}_{im}})$ 表示完成 ω_{ij} 的紧前任务集的最大时间值.设计时间限制

$$\max(T_{\mathrm{S}_{ij}}+T_{\mathrm{P}_{ij}})\leqslant T_{\mathrm{l}_i} \tag{4}$$

式中:$T_{\mathrm{S}_{ij}}$ 是第 i 个设计任务的最后一个子任务 j 的开始时间;$T_{\mathrm{P}_{ij}}$ 是第 i 个设计任务的最后一个子任务 j 的设计时间;T_{l_i} 表示第 i 个设计任务的设计周期.

将多设计任务调度问题转化为相应约束条件的 Nash 均衡点求解,即需满足

$$W_i(p_i^*,p_{-i}^*)\geqslant W_i(p_i,p_{-i}^*) \tag{5}$$

$$\forall p_i\in P_i,p_{-i}^*=(p_1^*,\cdots,p_{i-1}^*,p_{i+1}^*,\cdots,p_n^*)$$

$$i=1,2,\cdots,n$$

3 遗传算法的模型解算法设计

基于非合作博弈模型,针对 Nash 均衡点的求解,采用遗传算法实现对非合作博弈多设计任务调度模型的求解.

3.1 基因及染色体编码设计

针对多设计任务调度的数学模型,构建了设计任务、设计节点类型的基因,编码策略采用字符方式,对各基因的编码格式作如下描述.

(1) 设计任务编码

(“j”,“i”,“f”,“c”,“s”,“p”)

式中:“j”表示待设计任务的编号;“i”表示“j”的子任务编号;“f”表示“i”的紧前关系编号;“c”表示“i”的可用设计节点编号;“s”表示“i”在“c”上的开始设计时间;“p”表示“i”在“c”上的设计时间.

(2) 设计节点编码

(“c”,“m”)

式中:“m”表示该设计节点可同时进行设计的任务数.当 c_i 可同时进行的设计任务数 $m>1$ 时,将该节点分解为 m 个设计节点,用 $c_{i1},c_{i2},\cdots,c_{im}$ 表示.

根据上述基因编码规则,采用如图 1 所示的编

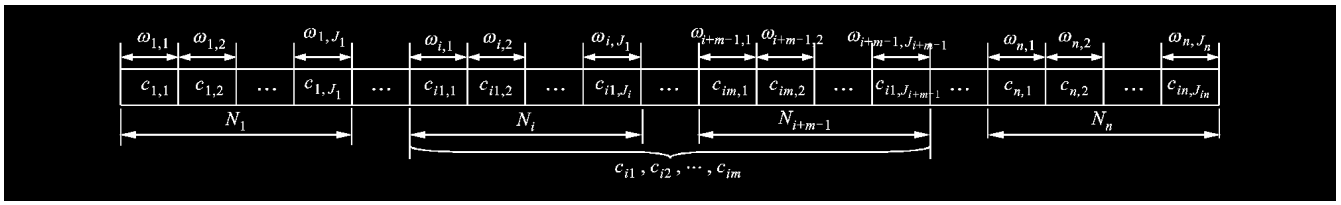


图 1　设计节点编码方式的染色体编码

码结构. 染色体为由设计节点编号构成的字符串，可分为 n 个子字符串，每个子字符串对应于一个设计任务子任务序列的可选设计节点集. 例如，N_1 包含的子任务序列的集合为 $N_1=\{\omega_{1,1},\omega_{1,2},\cdots,\omega_{1,J_1}\}$，所对应的可选设计节点序列为 $\{c_{1,1},c_{1,2},\cdots,c_{1,J_1}\subset C\}$. 因此，染色体的总长度为 $L=\sum_{1}^{n}J_i$.

3.2　适应度函数设计

基于非合作博弈的多设计任务调度的目标，是希望满足各设计任务的设计时间最小化，从而达到各设计任务间的利益均衡，因此要求适应度函数能够反映每个设计任务的竞争性设计需求. 据此，设计了如下的适应度函数

$$F^k(W_1^k,\cdots,W_i^k,\cdots,W_n^k)=\sum_{i=1}^{n}|W_i^k-\widehat{W}_i^{k-1}|=\sum_{i=1}^{n}\left|\frac{1}{T_{S_{ij}}^k+T_{p_{ij}}^k}+\frac{1}{T_{S_{ij}}^{k-1}+T_{p_{ij}}^{k-1}}\right| \tag{6}$$

式中：W_i^k 表示 N_i 在第 k 代的设计完成时间；$\widehat{W}_i^k$ 为 N_i 在第 $k-1$ 代的最短设计时间，其值来源于在 $k-1$ 代的最佳适应度函数 F_b^{k-1}. 当式(6)满足以下条件时，认为算法达到 Nash 均衡

$$F^k(W_1^k,\cdots,W_i^k,\cdots,W_n^k)<\xi \tag{7}$$

式中：ξ 为终止阈值，是确定求解 Nash 均衡点的条件.

3.3　算法的流程设计

算法的流程：① 随机产生设计任务调度的初始种群；② 计算个体适应度值，并判断是否达到收敛条件或最大进化代数，若达到收敛条件则输出最佳个体及代表各设计任务的最优方案，并结束算法，若已达到最大进化代数，则表示此次调度失败，否则转③；③ 根据个体适应度值、交叉和变异概率，进行选择、交叉和变异等操作，以产生新一代种群并返回②.

3.4　算法的进化操作

算法的进化操作由选择、交叉和变异 3 种进化算子共同支撑. 对于选择算子，采用轮盘赌方法，根据式(6)可以得出适应度越小的个体，其生存空间越大. 将式(6)作以下变换，以适应轮盘赌方法的选择概率计算方法

$$F'^k(W_1^k,\cdots,W_i^k,\cdots,W_n^k)=1/F^k(W_1^k,\cdots,W_i^k,\cdots,W_n^k) \tag{8}$$

对于交叉算子，本文算法采用的两点交叉方法如图 2 所示.

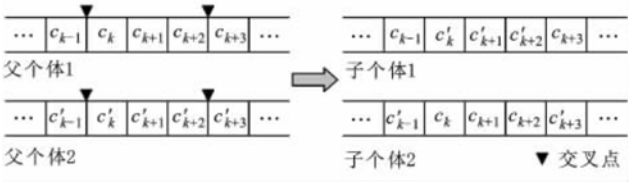


图 2　两点交叉逻辑

对于变异算子，因只有在预定的设计节点集内选择才有意义，所以算法控制在对应的设计节点集内进行选择变异，这样可以提高算法的收敛速度.

4　算例验证与分析

采用下列算例对多设计任务调度模型与调度算法进行验证和分析.

4.1　初始条件与参数

采用某企业的 6 个设计任务作为验证算例，如表 1 所示. 假设具有 5 个设计节点，其中节点 3 能够

表 1　设计任务序列

任务名称	子任务					
	1	2	3	4	5	6
分度机构	箱体	凸轮	输入轴	输出轴	分度盘	滚子等
泵体	轴	滑阀	阀芯	安全阀	进出油口	后盖
转向油泵	叶片	左配流盘	右配流盘	定子	转子	壳体
汽车驱动轴	半轴	轴承座	轮毂	环套	联结头	圆盘及附件
变速操纵机构	变速鼓	档位开关组件	定位板组件	换档臂组件	离合器凸轮板组件	离合器挺杆组件
发动机	机体组件	活塞连杆组件	曲轴飞轮组件	气门组件	气门传动组件	气门驱动组件

同时进行2个不同任务的设计.表2列出了6个设计任务的基本信息,表3为遗传算法设计的初始输入参数.

表2 设计任务的基本信息

设计任务	设计子任务					
	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6
N_1	{ } (3,5) [5,7]	{1} (1,3,4) [5,5,6]	{2} (3) [5]	{2} (1,5) [4,5]	{3,4} (2,4) [3,4]	{5} (1,3,5) [4,5,5]
N_2	{ } (2,3) [4,5]	{1} (3,5) [6,7]	{1} (1, 4) [4,3]	{3} (3,4) [4,6]	{2,4} (3) [5]	{5} (4,5) [4,3]
N_3	{ } (1,4) [4,5]	{1} (5) [5]	{2} (3,5) [4,4]	{1} (2,4) [4,5]	{4} (1) [5]	{3,5} (2, 3) [5,4]
N_4	{ } (1) [4]	{1} (3,5) [4,3]	{1,2} (2, 3,4) [5,5,4]	{2} (3) [5]	{3,4} (2,5) [3,5]	{5} (5) [4]
N_5	{ } (1,2) [5,6]	{1} (1, 2,5) [6,5,7]	{1} (2) [6]	{3} (4) [5]	{2,4} (1,4) [4,3]	{4,5} (2, 3,5) [3,5,4]
N_6	{ } (4,5) [6,5]	{1} (3,4) [7,6]	{1} (1, 3) [5,6]	{2,3} (2,5) [5,4]	{3} (1,2,3) [4,5,5]	{4,5} (3,5) [4,3]

表3 遗传算法的基本输入参数设计

种群/个	交叉概率	变异概率	ξ	最大迭代次数/次
50	0.8	0.02	0.000 1	5 000

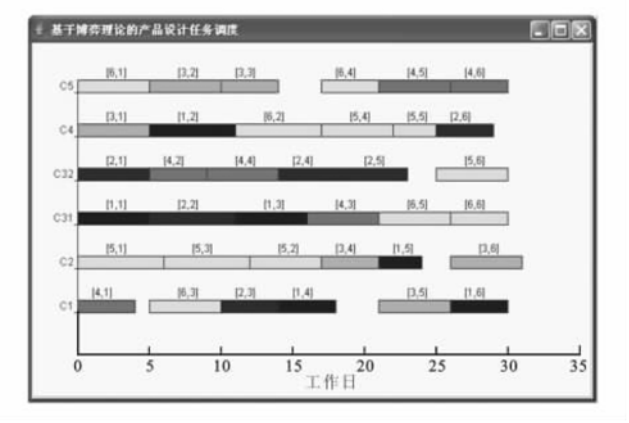
表2大括弧中的数字代表该设计子任务的紧前设计任务,如果为空表示该任务必须首先进行设计;圆括弧中的数字代表某设计子任务的可选设计节点;方括弧中的数字代表子任务在对应设计节点上的设计时间.同时,产品设计任务调度还要综合考虑设计任务周期.

4.2 验证结果与分析

采用遗传算法对模型进行了调度验证,其结果如图3所示.表4列出了在Nash均衡点处各设计任务的完成时间及其预计的设计周期.

表4 Nash均衡点各设计任务的完成时间

设计任务	完成时间/工作日	设计周期/工作日
N_1	30.0	35
N_2	29.5	32
N_3	30.5	36
N_4	30.0	35
N_5	30.0	34
N_6	30.0	36



方括号中的第1个数字表示设计任务,第2个数字表示该设计任务的子任务,如[2,3]表示设计任务 N_2 的第3个子任务,并以此类推

图3 多设计任务的调度甘特图

5 结 论

本文在考虑客户竞争性需求的基础上,对关联于各客户的多设计任务的调度问题进行了研究,主要工作内容及结果如下.

(1)以完成各设计任务的最短时间为目标,对多设计任务调度问题进行了数学描述.

(2)采用博弈理论,构建了适应各目标条件的任务调度非合作博弈模型,将任务调度问题转化为寻求该博弈模型的Nash均衡点问题.

(3)采用遗传算法对Nash均衡点进行了求解.

(4)以具体算例对这些方法和模型进行了验证.结果表明,采用博弈理论构建的多设计任务调度模型及基于遗传算法的求解方法,能够很好地解决非合作博弈多设计任务的调度问题.

参考文献：

[1] Luh P B, Liu Feng, Moser B. Scheduling of design projects with uncertain number of iterations[J]. European Journal of Operational Research, 1999,113:575-592.

[2] 黄洪钟，李丽，刘伟. 并行工程中设计任务的动态调度[J]. 机械工程学报，2002,38(增刊):164-167.
Huang Hongzhong, Li Li, Liu Wei. Dynamic scheduling of design tasks in concurrent engineering[J]. Chinese Journal of Mechanical Engineering, 2002, 38(Supp): 164-167.

[3] 任东锋，方宗德. 并行设计中任务调度问题的研究[J]. 计算机集成制造系统，2005,11(1):32-38.
Ren Dongfeng, Fang Zongde. Research on task scheduling in concurrent design[J]. Computer Integrated Manufacturing Systems, 2005,11(1): 32-38.

[4] 侯定丕. 博弈论导论[M]. 合肥：中国科学技术大学出版社，2004: 48-96.

[5] Son Y S, Baldick R. Hybrid coevolutionary programming for Nash equilibrium search in games with local optima [J]. IEEE Transactions on Evolutionary Computation,2004,8(4):305-315.

[6] 玄光男，程润伟. 遗传算法与工程优化[M]. 于歆杰，周根贵，译. 北京：清华大学出版社，2004:178-208.

[7] Azaron A, Perkgoz C, Sakawa M. A genetic algorithm approach for the time-cost trade-off in PERT networks[J]. Applied Mathematics and Computation, 2005(168):1317-1339.

（编辑　管咏梅）