

协同设计中的任务调度算法及实现^{*}

石 慧, 刘晓平

(合肥工业大学计算机与信息学院, 安徽 合肥 230009)

摘 要: 在协同设计子任务间形成的任务前趋图的基础上, 面向协同设计的实际需求, 采用了两个任务调度算法: 改进的AGA (Adaptive Genetic Algorithm, 自适应遗传算法) 及GASA (Genetic Simulated Annealing Algorithm, 遗传-模拟退火算法)。分别给出了算法的详细实现, 分析并比较了算法与其它任务调度算法的效能。最后, 针对较大规模的任务实例对这两个算法进行了横向比较, 分析了它们的时间效率和适用场合。

关键词: 任务调度; 自适应遗传算法; 遗传-模拟退火算法; 任务前趋图

中图分类号: TP391.7 **文献标识码:** A **文章编号:** 0529-6579 (2008) 06-0104-05

协同设计是由来自不同领域的多个设计小组共同协调完成同一个产品设计的过程。任务的分割及并发执行可以有效地提高设计效率, 协同设计的初衷也正基于此。由于设计产品的复杂性及不同设计组之间的交流, 分割的子任务之间存在相互依赖、相互制约的关系。任务规划的目的便是将产品设计包含的所有设计任务按照一定的原则分为若干个任务组, 并进一步将任务组分配给合适的设计组, 这是协同设计中的关键环节。因此, 合理的任务规划将有利于协同设计的顺利进行, 势必影响设计效率和质量。Agent 常被用于解决任务规划问题。文献[1]提出了协同装配任务规划系统, 借助于 Agent 解决任务分配问题。文献[2]针对协同工程设计问题提出了一个任务分配协议, 讨论了设计 Agent 如何选择任务, 任务如何挑选设计 Agent 等问题。文献[3]介绍了一种多 Agent 设计系统中的协作方法, 提出了支持动态任务分配的公告板机制以及设计过程中的冲突协调方法。公告板模型结合了黑板模型与合同网模型的优点。设计结构矩阵亦被用于任务规划问题^[4]。同时更多的文章采用了启发式算法来进行任务规划问题的研究。文献[5]引入了分组遗传算法。文献[6]考虑时间、任务顺序、任务协作等因素, 引入遗传算法解决任务分配问题。文献[7]主要采用遗传算法解决任务调度和分配问题, 设计了在任务调度和分配中的遗传算子。文献[8]仍旧采用遗传算法实现了并行设计中的任务调度, 但结果显示, 优化效果基本达不到最优。于是文献[9]用多步 Q 学习算法代替遗传

算法作为协同设计中的任务调度算法, 以同样的任务实例作为比照对象, 在优化效果上明显优于文献[8]中遗传算法, 但是对于大规模任务问题, 多步 Q 学习算法耗时耗空间。

本文则面向协同设计的实际需求, 改进了AGA 算法及 GASA 算法^[10-11], 将其作为协同设计中的任务调度算法, 描述了算法实现的具体步骤, 并采用文献[8]和文献[9]中实例与本文中算法进行纵向比较, 并采用复杂任务实例对本文中的两个算法进行横向比较, 分析其优劣。

1 问题描述及定义

在协同设计中, 任务间的相互依赖和约束是协同进程的重要方面, 每一个设计者可能会需要获取他人的设计结果才可顺利完成自身的任务, 但是设计者之间的合作并不总是协调一致的, 于是就需要在任务之间协调、规划以保证任务协作的效率。这种协调被理解为为了能够达到最终目标而对子任务间相互约束和依赖的控制。

我们采用有向图的方式来描述任务之间的约束及任务之间的驱动, 从而形成了任务约束图及任务前趋图^[12]。任务前趋图, 是一种反映了先后执行关系的抽象的图结构, 存储了任务节点以及它们之间的前趋后继关系。基于聚合与并序的转换算法的提出实现了由约束图向任务前趋图的转换。

定义 1 任务前趋图 $TPG = (PV, PE)$, PV 集的每一个任务节点 v_i 表示设计子任务, PE 集中的每一条有向边 e_{ij} 则表示任务之间的直接驱动关系,

^{*} 收稿日期: 2008-09-05

基金项目: 国家自然科学基金资助项目 (60573174); 国家自然科学基金资助项目 (60673028)

作者简介: 石 慧 (1980 年生), 女, 硕士; E-mail: shihui@vcc.hfut.edu.cn

是一种前趋与后继关系，具有传递性。图1为设计实例所产生的任务前趋图示例。

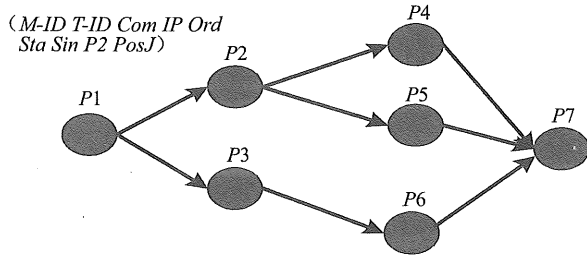


图1 任务前趋图

Fig. 1 Task precedence graph

$$PV = \{v_i\},$$

$$PE = \{e_{ij} = \text{Edge}(pv_i, pv_j) \mid pv_j \in \text{Succ}(pv_i)\}$$

$$i, j \in \{1, 2, 3 \dots n\}$$

$$pv_i = (M-ID, T-ID, Com, IP, Ord, Sta, Sin, Pos, J)$$

其中： $M-ID$ ——任务的ID号， $T-ID$ ——模板的ID号， Com ——该任务包含的设计组件， IP ——接收该任务的设计者， Ord ——该任务节点的执行次序， Sta ——任务的完成状态， Sin ——任务节点的入度即驱动该任务的任务节点数目， $PosJ$ ——该任务在并行任务间的位置。

$\text{Succ}(pv_i) : PV \rightarrow PV$ 指任务节点 pv_i 的直接后继节点集，即继 pv_i 之后立即执行的任务节点集。

$\text{After}(pv_i) : PV \rightarrow PV$ 指任务节点 pv_i 的直接或间接后继节点集，即继 pv_i 之后执行的任务节点集。

$\text{Pre}(pv_i) : PV \rightarrow PV$ 指任务节点 pv_i 的直接前趋节点集，即在 pv_i 之前执行的前趋任务节点集。

定义2 任务节点 pv_i 的深度值

$$H(pv_i) = \begin{cases} 0 & \text{Pre}(n_i) = \emptyset \\ 1 + \max(H(pv_j)) & \text{否则} \end{cases}$$

其中， $\text{Pre}(pv_i)$ 表示 pv_i 的前趋节点集合， $pv_j \in \text{Pre}(pv_i)$ 。

定义3 任务前趋图 TPG 中的任务与参与协同的设计者的任务匹配矩阵为 $NPM_{m \times n} = \{d_{ij} \mid 1 \leq i \leq m, 1 \leq j \leq n\}$ ，当 $d_{ij} = 1$ 表示任务 pv_j 调度给了设计者 i ，当 $d_{ij} = 0$ ，则表示任务 pv_j 没有调度给设计者 i 。定义 $NPM_{m \times n}$ 为一个调度策略记为 s ，为了确保任务的正常执行，在该策略中增加了两个约束条件：

(1) $\sum_{j=1}^n d_{ij} \geq 1, \sum_{i=1}^m d_{ij} = 1$ ，该约束条件表示每个设计者至少分配一个任务，并且一个任务只能调度给一个设计单元；

(2) 调度给同一个设计者的所有任务是按深

度值递增方式来排列的。

定义4 时间效率矩阵，由设计者提供的完成时间获得， $NPT_{m \times n} = \{t_{ij} \mid 1 \leq i \leq m, 1 \leq j \leq n\}$ ，描述了各设计者对各个子任务的执行效率，其中， t_{ij} 表示设计者 i 完成子任务 pv_j 所需时间。

本文算法的流程是在获得①描述子任务间驱动关系的任务前趋图与②设计者完成各个子任务所需时间的基础上，采用改进的进化算法求得最优的任务匹配矩阵，即所有任务完成所需的时间最短。

2 算法设计

在此，调度策略生成的目的是使得在该策略下设计者执行任务的时间最短，该类问题涉及到时间效率执行的优化，任务间的执行时间彼此影响^[6-7]。设计单元上某一个子任务的开始时间既受任务前趋图中其前趋任务集中子任务的最后结束时间影响，亦受其在该设计单元中的前趋任务的结束时间影响，取其中最大值作为该任务的开始时间。如果要求整个调度策略的时间，只要计算所有设计单元 NS_i 任务列表中的最后一个子任务结束时间的最大值即可。

任务进行初步的筛选，筛选后任务被执行的快慢往往是任务规划的最重要目标，于是可以将各个目标按重要度排序，并以时间列为最重要的影响因素。本文采用了两种改进的遗传算法来完成时间效率目标的规划，适应度函数、遗传交叉、变异算子均采用文献[8]中的设计方案。

2.1 改进的自适应 GA

遗传算法的参数中交叉概率 P_c 和变异概率 P_m 的选择是影响遗传算法行为和性能的关键所在，直接影响算法的收敛性，Srinivas 等提出一种自适应遗传算法 (AGA)， P_c 和 P_m 能够随着适应度自动改变。在自适应遗传算法中， P_c 和 P_m 按如下公式进行自适应调整：

$$P_c = \begin{cases} P_{m1} - \frac{(P_{c1} - P_{c2})(f' - f_{avg})}{f_{max} - f_{avg}} & f' \geq f_{avg} \\ P_{c1} & f' < f_{avg} \end{cases} \quad (1)$$

$$P_m = \begin{cases} P_{m1} - \frac{(P_{m1} - P_{m2})(f_{max} - f')}{f_{max} - f_{avg}} & f \geq f_{avg} \\ P_{m1} & f < f_{avg} \end{cases} \quad (2)$$

本文适应任务规划实际问题需要，对自适应 GA 进行了改进。

(1) 优化初始种群：任务前趋图中的并行任务，即深度值 $H(pv_i)$ 相同的任务，适宜在任务执行过程中并行执行，可缩短策略的执行时间。因此在初始化种群时，将并行任务分布分配，分给不同

的设计者, 相当于在种群个体注入了优化基因, 在选择、交叉和变异过程中保留该基因的概率也相应增大, 从而达到缩短执行时间的目的。

(2) 择优操作: 在每一代个体中选择适应度最高的符合策略中约束条件的个体直接复制到下一代中。

下面给出改进的自适应 GA 的算法描述:

Step1 读入任务前趋图, 计算图中各子任务的深度值 $H(pv_i)$

Step2 按并行任务分布分配的原则生成初始种群

Step3 如果不满足算法的终止条件 (如达到进化代数), 执行 Step4; 否则, 转 Step8

Step4 计算种群中每个个体的适应度

Step5 利用轮盘赌算法进行个体选择

Step6 以公式 (1), (2) 计算交叉概率和变异概率, 并以概率 P_c 和 P_m 执行交叉运算和变异运算

Step7 计算种群中的每个个体的调度时间 $t(s_i)$, 保存适应度最高的染色体, $t = t + 1$, 转 Step3

Step8 输出最好解, 算法结束

经过实验分析, 该算法适用于小规模的任务分配问题。对于文献 [8] 和文献 [9] 中的调度实例, 实验结果如表 1 所示。其中 $P_{c1} = 0.9$, $P_{c2} = 0.6$, $P_{m1} = 0.1$, $P_{m2} = 0.001$ 。

表 1 传统遗传算法与改进 AGA 比较表 (表中 “*” 表示无此参数数据或者不确定)

Tab. 1 The comparison between common GA and improved AGA (the “*” means no such parameters or uncertain)

初始种群规模	进化代数	计算次数	算法	最优值	平均值	获取最优值的最短时间	获取最优值的次数
100	200	100	遗传算法	19	20	*	*
			改进 AGA	8	14	0.015	55
100	1000	100	遗传算法	14	15	*	*
			改进 AGA	8	15	0.016	92
100	2000	100	遗传算法	14	15	*	*
			改进 AGA	8	15	0.015	87

2.2 改进的 GASA 算法

GASA 算法是遗传算法与模拟退火算法的结合。算法描述如下:

Step1 初始化算法参数, 包括种群数目 p 、初始温度 t_k 、退温速率 λ 等

Step2 初始化遗传算法种群

Step3 计算种群中所有个体的适应度, 确定最优适应度及最优个体

Step4 判断最优适应度 S 是否连续 t 步没有发生变化? 如果稳定, 执行 Step9, 否则, 继续执行

Step5 循环执行 k 次: 随机选择个体与最优个体执行外部交叉操作, 扩大种群数量, 同时, 筛选优秀个体, 更新最优适应度及最优个体

Step6 按变异概率 P_m 对个体执行变异操作, 筛选优秀个体, 更新最优适应度及最优个体

Step7 对遗传算法生成的种群执行模拟退火操作

Step7.1 判断所有个体是否都执行过退火操作? 如果是, 执行 Step8, 否则, 执行以下循环

Step7.2 取种群中未退火个体

Step7.3 判断最优适应度是否连续 s 步没有发生变化? 如果稳定, 执行 Step7.1

Step7.4 由 SA 状态产生函数产生新个体

Step7.5 如果适应度增加则更新个体, 否则, 以概率 $\min(1, \exp(s_1 - s_2)/t_k)$ 接受新个体

Step7.6 更新最优适应度及最优个体

Step8 进行退温操作, $t_k = \lambda \times t_{k-1}$, 返回 Step4

Step9 算法结束, 输出结果

其中, SA 算法中状态生成函数的步骤描述如下:

Step1 随机选择一个子任务, 即在任务匹配矩阵 $NPM_{m \times n}$ 中选择了—个基因列

Step2 计算该子任务分配给不同设计者的效率, 并将该任务分配与执行效率最高即消耗时间最短的设计者 (如果原方案已经符合该原则, 则无须改动)

Step3 如果此策略不符合分配策略的两个约束条件, 则转 Step1, 否则结束

通过采用文献 [8] 和文献 [9] 中的调度实例, 得实验结果如表 2 所示。其中 $t = 20$, $k = 4$, $s = 20$, $\lambda = 0.8$, $t_k = 5$ 。

显然, 对于上述小规模问题, 不论是改进的自适应 GA 算法还是 GASA 算法, 效果明显优于文献 [8] 中算法, 获得最优值的频率高的多。

表 2 传统遗传算法与改进的 GASA 比较表
Tab. 2 The comparison between common GA and improved GASA

算法	初始种群规模	进化代数	计算次数	最优值	平均值	获取最优值的最短时间	获取最优值的次数
遗传算法	100	200	100	19	20	*	*
	100	1000	100	14	15	*	*
	100	2000	100	14	15	*	*
改进的 GASA	100	*	100	8	8.01	6.282	99

3 算法比较

构造复杂实例验证算法效果，共 4 个设计者，10 个设计子任务，其任务前趋图如图 2 所示。时间效率矩阵如表 3 所示。

采用本文中的两类算法进行实现，其比较结果如表 4。

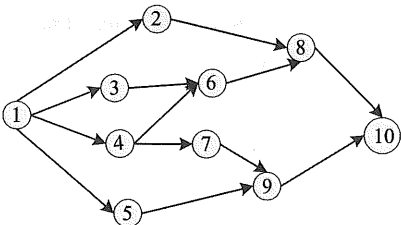


图 2 4 * 10 设计实例任务前趋图
Fig. 2 Task precedence graph of 4 * 10 design instance

表 3 4 * 10 设计实例时间效率矩阵
Tab. 3 Time-efficiency matrix of 4 * 10 design instance

	pv_1	pv_2	pv_3	pv_4	pv_5	pv_6	pv_7	pv_8	pv_9	pv_{10}	
designer ₁	2	4	2	5	4	3	5	8	6	2	Tmax = 62
designer ₂	4	1	7	2	3	5	3	6	4	7	
designer ₃	6	5	3	5	7	4	1	2	5	3	
designer ₄	1	6	1	2	2	3	5	8	2	4	

表 4 改进的 AGA 与改进的 GASA 比较表
Tab. 4 Comparison of improved AGA and improved GASA

算法	初始种群规模	进化代数	计算次数	最优值	平均值	获取最优值的最短时间	获取最优值的次数	平均计算一次所需时间
改进的自适应 GA	100	200	100	10	23.2	2.156	1	1.21
	100	300	100	10	23.4	0.938	2	3.55
	100	400	100	10	23.3	1.219	4	5.00
	100	500	100	10	23.3	0.360	4	5.99
	100	1000	100	10	23.0	0.016	9	12.11
	100	2000	100	10	23.0	0.094	7	24.18
改进的 GASA	100	*	100	10	10.8	13.11	27	5.12

对于复杂实例，改进的 GASA 算法得到最优值的次数明显高于自适应 GA 算法。文献 [9] 中的 Q 学习算法与多步 Q 学习算法，对于上述小规模实例均能达到最优，其中多步 Q 学习算法收敛速度更快。但是当面临规模大的问题时，Q 学习算法状态 - 动作空间亦很大，收敛速度明显下降，耗时多。

4 结 论

协同设计是多个设计者共同协作完成一件复杂设计任务的过程。各个子任务之间存在着约束关系，相互制约。对于任务进行合理排序和规划，能够促进协同设计的效率。本文为了提高任务分配及规划的质量，改进了自适应遗传算法和 GASA 算法，描述了算法实现的详细步骤，进行了算法比

较:首先分别用这两个算法实现引用任务实例,并将结果与文献中的算法结果进行纵向比较和分析;然后实现复杂任务实例,对自适应遗传算法和 GASA 算法进行横向比较,分析了它们的性能和优缺点。从优化结果来看,本文算法的优化效果明显好于传统的遗传算法,即既能获得最优结果,且获得最优结果的频率较高;从问题规模来看,本文算法对于解决较大规模问题的效果良好,克服了 Q 学习算法状态-动作空间大,收敛速度明显下降,耗时多的问题。诚然,本文仅针对时间要素对协同设计中的任务进行规划,并未考虑成本、质量、信用等多个要素,即仍属于单目标优化问题。

参考文献:

- [1] DONG Tianyang, TONG Ruofeng, ZHANG Ling, et al. Approach to cooperative assembly task planning for multiple manipulators[C]//The 8th International Conference on Computer Supported Cooperative Work in Design, 2004(1):202-208.
- [2] KWANG M, SHIU S C, BUN M L. Simulation of a multi-agent protocol for task allocation in cooperative design[C]//1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999(3):95-100.
- [3] 刘弘.一种支持动态任务分配的协同设计方法[J].软件学报,2001,12(12):1830-1836.
LIU Hong. A cooperative design approach supporting dynamic task assignment[J]. Journal of Software, 2001, 12(12):1830-1836.
- [4] 苏财茂,柯映林.面向协同设计的任务规划与解耦策略[J].计算机集成制造系统,2006,12(1):21-26.
SU Caimao, KE Yinglin. Task planning and decoupling strategy for collaborative design[J]. Computer Integrated Manufacturing Systems, 2006, 12(1):21-26.
- [5] 杨友东,高曙明,张书亭,等.面向自顶向下协同装配设计的任务规划研究[J].计算机集成制造系统,2007,13(7):1268-1281.
YANG Youdong, GAO Shuming, ZHANG Shuting, et al. Task planning for top-down collaborative assembly design[J]. Computer Integrated Manufacturing Systems, 2007, 13(7):1268-1281.
- [6] SHIMA T, RASMUSSEN S J, SPARKS A G. UAV cooperative multiple task assignments using genetic algorithms[C]//2005 American Control Conference, 2005(5):2989-2994.
- [7] 钟求喜,陈火旺.任务分配与调度中遗传算子的设计[J].国防科技大学学报,2000,22(3):34-38.
ZHONG Qiuxi, CHEN Huwang. Genetic operators in task matching and scheduling[J]. Journal of National University of Defense Technology, 2000, 22(3):34-38.
- [8] 殷国富,罗阳,龙红能,等.并行设计子任务调度的遗传算法原理与实现方法[J].计算机辅助设计与图形学学报,2004,16(8):1122-1126.
YIN Guofu, LUO Yang, LONG Hongneng, et al. Genetic algorithms for subtask scheduling in concurrent design[J]. Journal of Computer-aided Design and Computer Graphics, 2004, 16(8):1122-1126.
- [9] 陈圣磊,吴慧中,肖亮,等.协同设计任务调度的多步 Q 学习算法[J].计算机辅助设计与图形学学报,2001,19(3):398-408.
CHEN Shenglei, WU Huizhong, XIAO Liang, et al. Multi-step QLearning algorithm for task scheduling in collaborative design[J]. Journal of Computer-aided Design and Computer Graphics, 2001, 19(3):398-408.
- [10] 夏锐,肖明清,程进军.基于混合遗传退火算法的并行测试任务调度优化[J].系统仿真学报,2007,19(15):3564-3567.
XIA Rui, XIAO Mingqing, CHENG Jinjun. Optimization for the parallel test scheduling based on hybrid GASA[J]. Journal of System Simulation, 2007, 19(15):3564-3567.
- [11] 王凌,郑大钟.一种 GASA 混合优化策略[J].控制理论与应用,2001,18(4):552-554.
WANG Lin, ZHENG Dazhong. A kind of GASA hybrid optimization strategy[J]. Control Theory and Application, 2001, 18(4):552-554.
- [12] LIU Xiaoping, SHI Hui, LU Qiang, et al. Visual task-driven based on task precedence graph for collaborative design[C]//11th International Conference on Computer Supported Cooperative Work in Design, 2007:246-251.

Task Scheduling Algorithms in Cooperative Design

SHI Hui, LIU Xiao-ping

(School of Computer & Information, Hefei University of Technology, Hefei 230009, China)

Abstract: A reasonable task scheduling algorithm can effectively improve the quality and efficiency of Collaborative Design. For the actual demand of Collaborative Design, two task scheduling algorithms are adopted based on the task precedence graph formed by subtasks. One is improved Adaptive Genetic Algorithm, the other is Genetic Simulated Annealing Algorithm. The detailed steps are given, the performance of the algorithms are also analyzed with comparison with other task scheduling algorithm. Finally, these two algorithms are compared horizontally for relatively large-scale examples, and the time efficiency and application occasions are analyzed as well.

Key words: task scheduling; AGA; GASA; task precedence graph