

基于多Agent的虚拟企业任务调度模型及优化

赵 强^{1,2}, 肖人彬¹

(1. 华中科技大学 系统工程研究所, 湖北 武汉 430074; 2. 武汉科技大学 机械学院, 湖北 武汉 430081)

摘要: 采用多智能体技术构建虚拟企业任务调度模型, 并对基于该模型的任务调度运作过程进行说明. 针对调度优化问题, 以资源智能体承担的生产任务为研究对象, 综合考虑生产任务之间的时序逻辑关系、作业时间及资源自身已确定的生产任务等影响因素, 建立以生产延续时间最小为目标的优化模型, 给出粒子群优化求解算法. 应用实例及数字仿真验证了模型及优化算法有效性.

关键词: 虚拟企业; 任务调度; 多智能体系统; 粒子群优化

中图分类号: TP301 **文献标识码:** A

The multi-agent-based model for task scheduling in virtual enterprise and its optimization

ZHAO Qiang¹, XIAO Ren-bin²

(1. Institute of System Engineering, Huazhong University of Science and Technology, Wuhan Hubei 430074, China;

2. College of Machinery and Automation, Wuhan University of Science and Technology, Wuhan Hubei 430081, China)

Abstract: By means of multi-agent technology, the task-scheduling model in a virtual enterprise (VE) is presented, and the operation process of task-scheduling in VE based on the above model is explained. In the optimization of task-scheduling, the production tasks of each resource agent are taken as the research objective; and the model for task-scheduling in VE for minimizing the production duration is established by comprehensively considering the logical relation among subtasks, the operation time of each subtask, the assigned production activities of each resource agent, etc. The particle swarm optimization (PSO) algorithm is employed for the problem-solving. Finally, a real example and the numerical simulation show that this task-scheduling model in VE and the optimization algorithm are effective.

Key words: virtual enterprise; task scheduling; multi-agent system; particle swarm optimization

1 引言(Introduction)

虚拟企业(VE)具有结构灵活、快速响应等特性, 因而受到越来越多企业的重视. 当企业面临某些核心资源缺乏或能力限制, 而无法独立完成某生产任务时, 可以通过构建VE来完成该任务. 通常盟主企业首先将生产任务分解、归并为若干子任务, 然后采取招标或选择代理等方式选择适宜合作企业承担相应子任务, 最终通过合作企业共同努力完成整个生产任务. 任务调度关系到VE运作成败, 因此成为VE一项重要研究内容. 任务调度也是典型的NP难题^[1], 并且由于VE具有离散性、不确定性、动态性等特点, 使其任务调度变得更为复杂. 通常采用的禁忌搜索法、遗传算法等单一优化方法已不能较好解决VE任务调度问题. 将多智能体(multi-agent)技术与优化算法相结合为解决VE动态任务调度问题提

供了方法^[2].

目前采用多智能体技术研究VE构建、生产运作及协调管理的成果较多, 但很少研究涉及VE任务调度问题, 并且已有任务调度优化模型存在一定不足. 文[3]建立的VE任务调度模型仅考虑子任务的完成时段, 模型较简单, 并且给出的启发式算法仅能确定有效解存在性, 很难得到较优的调度方案. 本文认为VE任务调度优化问题类似于JobShop问题, 但该问题更复杂, 突出表现为生产任务之间约束关系复杂性, 如某些任务可能存在多个紧前或紧后任务, 同时需考虑成员地理位置分散的特点, 此外, 资源自身已确定的生产任务也是影响VE任务调度优化的重要因素, 而该因素往往被忽视, 如文[4]. 因此VE任务调度的模型及优化求解算法需进一步扩展, 才能更好运用于VE任务调度问题. 本文将构建VE任务调度

运作模型,并综合考虑生产任务之间的时序逻辑关系、作业时间、资源自身已确定的生产任务等影响因素,建立VE任务调度的优化模型,给出粒子群优化求解算法.结合多智能体技术与优化算法能够较为有效地解决VE任务调度问题.

2 任务调度模型(Model for task scheduling)

文[5]提出的调度模型包括任务智能体(task agent, TA)、调度智能体(scheduling agent, SA)及执行监控智能体(implement agent, IA).本文依据VE任务调度需要,新增管理智能体(management agent, MA)、任务分解智能体(decomposition agent, DA)及资源智能体(resource agent, RA),并建立产品、订单数据库与决策模型及算法知识库,VE任务调度运作模型如图1所示.

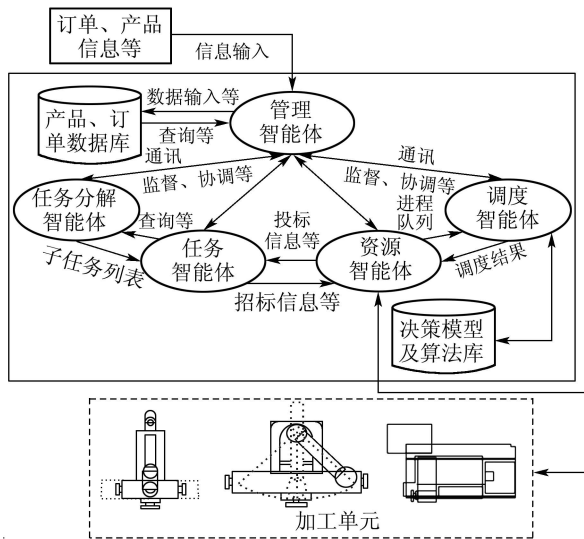


图1 基于多智能体的VE任务调度运作模型

Fig. 1 Mutil-agent based model for task scheduling in VE

上述模型中,MA负责其余智能体的监控及协调管理,并协调不同智能体之间信息交互,如DA进行任务分解时需参考RA部分信息,借助MA,DA与RA能实现信息交互;DA负责将MA传递的生产任务分解为若干具有时序约束关系的子任务,并将子任务列表传递给TA;TA通过招标等形式选择适宜的RA承担子任务,最终将所有子任务合理分配给RA,并监控其运行.若任务分配及子任务执行过程中出现意外情况,则TA通知MA进行协调解决;RA负责整个VE资源管理、配置及监控.当某企业加入到VE时,其所拥有的诸如原材料、设备、人员等资源也成为VE资源一部分,并由RA负责注册登记,记录其数量、位置、功能及状态等信息;当资源某项信息发生变化时,由RA通知MA以便及时更新数据库信息.SA负责调用相应决策模型及算法知识库对每个子任务进程进行合理规划,使其满足预订目标

要求(如生产延续时间最小、生产成本最低等),并将优化后的调度队列信息反馈给RA,再由RA执行相应子任务.通过智能体之间的协同合作,最终完成VE整个生产任务.

3 任务调度优化求解(Optimization on task scheduling)

调度优化是VE任务调度运作模型的关键环节之一,本文以生产延续时间最小为模型说明任务调度优化实现过程.

3.1 问题描述(Question description)

为完成某生产任务VE被组建.经任务分解、RA集选择及子任务分配,并获取相关信息(RA自身已确定的生产任务、完成子任务所需作业时间及子任务之间转运时间)后,SA优化目标为兼顾RA各自生产任务,规划VE任务调度方案使得生产延续时间最小.VE任务调度优化是一个循环反复的过程,当优化的调度方案无法满足预定完工期时,MA需要与RA进行协商,对子任务作业时间,乃至对RA已确定的生产任务进行修改,以满足完工期要求.为简化研究,假定在不影响RA各自生产任务的前提下,存在VE任务调度方案既能满足预定完工期,又能最小化生产延续时间.

3.2 优化模型(Optimization model)

为建立VE任务调度优化模型做如下假设:

- 1) 每个子任务执行顺序满足时序逻辑约束,且执行过程中不容许中断;
- 2) 考虑RA自身生产能力及条件限制,假定RA执行自身生产任务时,无力完成VE所分配的任何子任务,并且假定RA在执行VE所分配的子任务时,一次能且仅能执行一个子任务;
- 3) 生产任务的起始时间为0.

此外,模型中符号设定如下: R 为资源智能体RA集合, $R = \{r_1, \dots, r_i, \dots, r_m\}$, m 为RA总数; T 为经DA分解最终得到的子任务集合, $T = \{t_1, \dots, t_i, \dots, t_m\}$, n 为子任务总数; E_i 为 r_i 承担的子任务集合,并假定 $t_i \in E_i$; H_i 为 t_i 所有紧前子任务集合; B_i 为 t_i 所有紧后子任务集合,并假定 $t_j \in B_i$; z_{ij} 为子任务 t_i 到子任务 t_j 的转运时间; s_i 为子任务 t_i 的开始作业时间; q_i 为完成子任务 t_i 所需的作业时间; f_i 为完成子任务 t_i 所需的作业时间; $t_{i'}$ 为同由 r_i 完成,且位列于子任务 t_i 之前执行的子任务; P_i 为表示 r_i 已确定的生产任务集合,用时间段表示.

基于以上的假设及符号设定,则VE任务优化调度优化模型可如下描述:

$$\min F = \max f_i, i = 1, 2, \dots, n, \quad (1)$$

$$s_j \geq f_i + z_{ij}, j = 1, 2, \dots, n, \quad (2)$$

$$s_i \geq f_{i'} + z_{i'i} \times \delta, \quad (3)$$

$$f_i = s_i + q_i, \quad (4)$$

$$[s_i, f_i] \cap P_i = \phi. \quad (5)$$

其中
$$\delta = \begin{cases} 1, & t_{i'} \in H_i, \\ 0, & \text{其他}. \end{cases}$$

式(1)为目标函数, 优化VE任务调度方案力求生产延续时间最小; 约束条件(2)为子任务开始时间必须在其所有紧前子任务转运完成后才能开始; 约束条件(3)为子任务开始时间必须在同由一个RA完成的紧前一个子任务完成后才能开始; 约束条件(4)表示子任务开始时间与完成时间的关系; 约束条件(5)表示RA在执行VE任务时不能与自身已确定的生产任务相冲突。

4 PSO 算法优化求解(Optimization based on PSO algorithm)

PSO 算法^[6]具有收敛速度快、参数调整少、优化效果好等特点, 已被广泛用于连续或离散型问题的优化求解。采用PSO算法求解VE任务调度方案的关键环节为设定粒子位置矢量的表达、适应度函数及迭代更新公式, 以下对上述环节进行说明。

1) 约束矩阵建立。由于子任务之间存在时序逻辑关系, 因此只有建立子任务之间的时序逻辑约束矩阵才能有效保证优化后的VE任务调度方案合理性。本文用3个常量定义子任务之间的时序逻辑约束矩阵, 以子任务*i*与*j*为例说明。0表示子任务*i*与*j*之间不存在时序逻辑约束关系; 1表示子任务*i*与*j*之间存在时序逻辑约束关系, 且子任务*i*先于*j*开始作业; 2表示子任务*i*与*j*之间存在先后顺序关系, 且子任务*i*后于子任务*j*开始作业。

2) 粒子位置矢量表达。采用PSO算法进行优化求解需建立粒子位置矢量与问题解的联系。设定粒子群数为*m*, 维度为*n* (*n*为子任务数目), 本文用粒子位置矢量 $p_i = \{p_{i1}, \dots, p_{ij}, \dots, p_{in}\}$ (p_i 为1~*n*的一个序列)代表一种VE任务调度的候选方案。粒子位置矢量生产方法为随机给定粒子每一维度的数值, 其后按从小到大顺序, 重新将粒子位置矢量赋值为1~*n*。通过上述方法产生的VE任务调度方案还需用时序逻辑约束矩阵进行修正, 以保证其合理性。

3) 适应度函数设定。由于本文采用总生产延续时间最小做为VE任务调度的评价指标, 并且PSO算法通常进化的方向也为适应度函数减小的方向, 因此设定适应度函数如下。

$$\text{fitness} = \max f_i, i = 1, 2, \dots, n. \quad (6)$$

4) 迭代更新公式。设迭代次数*t*时, 粒子*i*位置矢量为 $p_i^{(t)} = \{p_{i1}^{(t)}, \dots, p_{ij}^{(t)}, \dots, p_{in}^{(t)}\}$, 速度矢量为 $v_i^{(t)} = \{v_{i1}^{(t)}, \dots, v_{ij}^{(t)}, \dots, v_{in}^{(t)}\}$, 个体最优位置矢量为 $p'_i = \{p'_{i1}, \dots, p'_{ij}, \dots, p'_{in}\}$, 种群最优位置矢量为 $g_i = \{g_{i1}, \dots, g_{ij}, \dots, g_{in}\}$, 求解过程中, 粒子速度矢量依据公式(7)和(8)更新。

$$v_{id}^{(t+1)} = \omega \cdot v_{id}^{(t)} + c_1 \cdot \text{rand}_1() \cdot [p'_{id} - p_{id}^{(t)}] + c_2 \cdot \text{rand}_2() \cdot [g_{id} - p_{id}^{(t)}], \quad (7)$$

$$p_{id}^{(t+1)} = p_{id}^{(t)} + v_{id}^{(t+1)}. \quad (8)$$

更新完毕后, 再采用粒子位置矢量表达中给出的赋值及修正方法产生新的粒子位置矢量, 进而产生新的VE任务调度方案。

5 实例及仿真(Example and simulation)

以设备制造为例^[4]验证上述VE任务调度模型及优化算法有效性。DA将设备制造过程分解为31个子任务(子任务作业次序的活动网络如图2所示)。

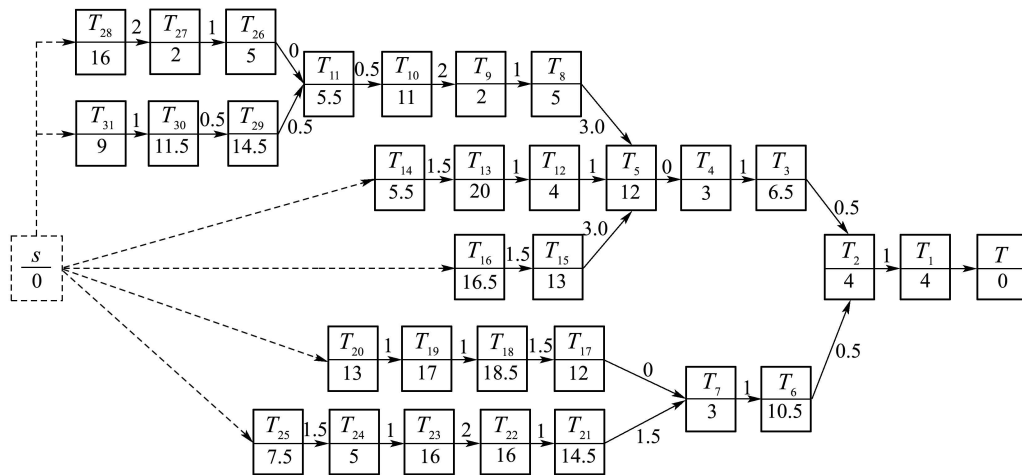


图2 子任务作业次序的活动网络

Fig. 2 Active network of subtask operation order

图中矩形框中的数字为子任务的作业时间(单位:天),箭头上标注为转运时间. TA选择6个RA完成所有子任务. RA任务分配方案如下: r_1 执行任务 $\{t_1, t_4, t_5, t_{22}, t_{27}, t_{31}\}$, r_2 执行任务 $\{t_6, t_{11}, t_{21}, t_{26}, t_{30}\}$, r_3 执行任务 $\{t_2, t_{10}, t_{14}, t_{25}, t_{29}\}$, r_4 执行任务 $\{t_7, t_{12}, t_{16}, t_{17}, t_{28}\}$, r_5 执行任务 $\{t_3, t_9, t_{13}, t_{19}, t_{24}\}$, r_6 执行任务 $\{t_8, t_{15}, t_{18}, t_{20}, t_{23}\}$. RA已确定的生产任务如下: r_1 已确定的生产任务为 $\{[12, 24], [56, 68]\}$, r_2 已确定的生产任务为 $\{[0, 8], [24, 35], [90, 102]\}$, r_3 已确定的生产任务为 $\{[38, 46], [60, 90]\}$, r_4 已确定的生产任务为 $\{[35, 55], [80, 100]\}$, r_5 已确定的生产任务为 $\{[0, 14], [64, 86]\}$, r_6 已确定的生产任务为 $\{[0, 2], [18, 23], [70, 90]\}$. 产品交货期为110天, 规划VE任务调度方案使其满足交货期要求.

设定粒子群数为10, 维度为31, 迭代次数为10, 并令 $\omega=0.729$, $c_1 = c_2 = 1.49445$. 采用PSO算法优化后设备制造任务调度方案如图3所示. 此调度方案下生产延续时间为102天, 可提前8天交货. 若不考虑成员企业自身生产任务, 采用PSO算法求得生产延续时间为99天, 与文献[10]优化结果相同, 但本文的PSO算法采用MATLAB6.5 编码, 在Pentium IV PC机上的平均求解时间为1.7 s, 明显优于文[10]给出的6~8 s的平均求解时间.

为验证PSO算法求解大规模任务调度问题有效性, 采用模拟实验方法进行测试. 模拟实验中子任务总数分别为40, 50, 60, 80, 子任务作业时间及转运时间、子任务分配方案、作业次序的活动网络均随机产生. 针对不同子任务数目的VE任务调度问题, 模拟实验重复进行5次. 试验过程中, 均设定粒子群数为20, 迭代次数为30. 实验数据整理如表1所示. 模拟实验验证了PSO算法求解大规模任务调度问题的有效性.

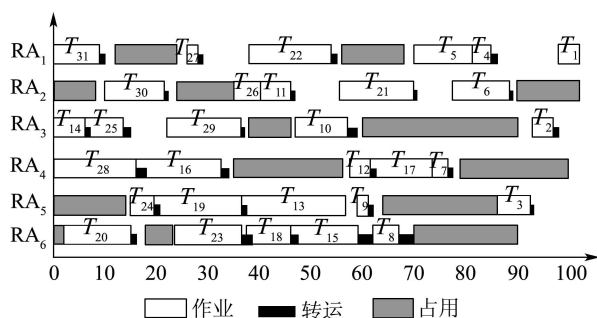


图3 设备制造任务调度方案

Fig. 3 Task scheduling plan on equipment manufacture

表1 数字仿真的测试结果

Table 1 The result of numerical experiment

序号	子任务总数	RA总数	计算时间/s
1	40	6	5.3120
2	50	6	6.6560
3	60	8	8.1720
4	80	10	11.2810

6 结论(Conclusion)

采用多智能体技术构建了VE任务调度运行模型. 该模型运作过程为首先由MA将总的生产任务传递给DA, 由其将总的生产任务分解为若干具有时序约束关系的子任务; 其次, 由TA选择RA承担子任务, 并将所有子任务进行合理分配; 最后, 由SA进行优化调度, 并将调度结果传递给RA执行作业. 针对调度优化调度问题, 以RA承担的生产任务为对象, 综合考虑生产任务之间的时序逻辑关系、作业时间、已确定生产任务等因素, 建立了以生产延续时间最小为目标的优化模型. 为求解该模型, 构造了时序逻辑约束矩阵, 有效解决了任务之间复杂的约束关系, 并采用基于任务编号编码的PSO算法进行优化求解. 实例及数字仿真验证了VE任务调度模型及优化算法有效性.

参考文献(References):

- [1] LIN W, BYRNES C I. Control of discrete-time nonlinear systems[J]. *IEEE Transaction on Automatic Control*, 1996, 41(4): 494 – 510.
- [2] RABELO R J, CAMARINHA MATOS L M, AFSARMANESH H. Multi-agent-based agile scheduling[J]. *Robotics and Autonomous Systems*, 1999, 27(1/2): 15 – 28.
- [3] 苏平, 伍乃骐. 一种可重构制造系统的生产计划方法[J]. 计算机集成制造系统, 2003, 9(3): 189 – 193.
(SU Ping, WU Naiqi. A methos for production scheduling in reconfigurable system[J]. *CIMS*, 2003, 9(3): 189 – 193.)
- [4] 高阳, 江资斌. 用混合遗传算法求解虚拟企业生产计划[J]. 控制与决策, 2007, 22(8): 931 – 934.
(GAO Yang, JIANG Zibin. Using HGA to slove production planning in VE[J]. *Control and Decision*, 2007, 22(8): 931 – 934.)
- [5] GRAHAM J R. *Real-time scheduling in distributed multi-agent system*[D]. Newark : University of Delaware, 2001.
- [6] KENNEDY J EBERHART R C. Particle swarm optimization[C]// *Proceedings of IEEE International Conference on Neural Networks*. Piscataway, NJ: IEEE, 1995: 1942 – 1948.

作者简介:

赵强 (1976—), 男, 博士研究生, 目前从事虚拟企业、群集智能等研究, E-mail: zhaoqiang88@gmail.com;

肖人彬 (1965—), 男, 教授, 博士生导师, 目前从事系统方法论、系统复杂性等研究.