

基于蚁群算法的MC供应链协调仿真研究

窦建华, 林杰

(同济大学经济与管理学院, 上海 200092)

摘要: 针对多企业MC供应链在信息完全共享情况下的协调问题, 建立了一个基于蚁群算法的MAS仿真系统。叙述了算法设计步骤及MAS和各Agent结构和功能。描述了MAS在JADE开发环境下实现的具体技术方法。给出了仿真结果, 对仿真结果进行了分析与对比。
关键词: 供应链协调; 蚁群算法; MAS; 仿真

Research on MC Supply Chain Coordination Simulation Based on Ant Colony Algorithm

DOU Jianhua, LIN Jie

(School of Economics and Management, Tongji University, Shanghai 200092)

【Abstract】 A MAS simulation system is built based on ant colony algorithm, aiming at the problem of MC supply chain coordination for multi-enterprise under the condition of complete shared information. The design steps of algorithm and the structures and functions of MAS and agents are described. The specific technology methods for implementing the MAS in the JADE environment are explained. The results of the simulation are analyzed and compared.

【Key words】 Supply chain coordination; Ant colony algorithm; MAS; Simulation

大规模定制(Mass Customization, MC)生产方式下, 供应链通过更好地协调, 可以使供应链各成员及整个供应链成本节约或利润增加^[1]。现有研究成果中, 文献[2]分析了集中决策下具有需求预测更新的订货模型; 文献[3]提出了大规模动态定制系统的概念及基于GDSS的供应链协调与管理系统; 文献[4]利用面向约束的近似搜索算法、综合模型分解算法和拍卖算法分别对不同协调机制下的供应链计划进行求解, 并进行了仿真。蚁群算法(Ant Colony Algorithm, ACA)的快速性、全局优化性以及鲁棒性使其在求解TSP、车间调度、约束优化等问题时都取得了较好的效果。本文利用蚁群算法与拥有强大的解决非结构化问题能力的多代理系统(Multi-Agent System, MAS)相结合, 解决供应链中各企业信息完全共享情况下的协调问题。

1 问题描述

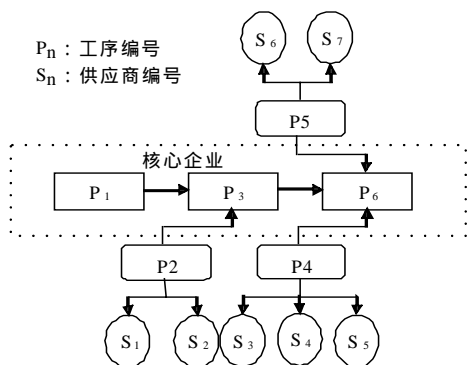


图1 产品工序约束

某MC供应链中, 有一核心企业C, 其生产的某种产品X需经过P₁至P₆ 6道工序完成。其中, P₁、P₃、P₆由核心企业完

成, P₂、P₄、P₅为定制件的生产, 由供应商完成。其工序约束关系如图1所示。

图中S₁至S₇分别为完成工序P₂、P₄、P₅时可选的供应商。鉴于定制件一般数量不大, 故假设同一订单的同种定制件仅由一家供应商提供。在核心企业完成P₃、P₆之前, 除了必须完成P₁、P₃之外, 还必须从S₁至S₇中选择3家分别完成P₂、P₄、P₅的生产。

核心企业接到订单后组织生产, 同时要求供应商在核心企业开始P₃、P₆生产时将货物按时按量送至生产线。供应商根据自身的剩余生产能力确定出最早交货时间, 同产品报价一起将信息反馈给核心企业。同时假设该核心企业不能并行处理不同订单的同一工序。

现核心企业接收到产品X的6个订单。每个订单都有最后交货时间的约束。核心企业与各供应商完成各工序所需时间, 如表1所示。各企业完成各工序所需成本如表2所示。

表1 工序完成时间及约束表

	P ₁	P ₂		P ₃	P ₄			P ₅		P ₆	约束
		S ₁	S ₂		S ₃	S ₄	S ₅	S ₆	S ₇		
订单1	6	7	5	9	5	8	9	9	16	12	30
订单2	2	2	3	3	3	3	3	4	4	4	47
订单3	6	6	7	9	3	8	5	15	11	12	54
订单4	8	7	9	12	9	7	6	21	11	16	71
订单5	4	5	5	6	7	4	5	6	12	8	89
订单6	10	10	9	15	7	9	8	14	17	20	90

基金项目: 上海市科委基金资助重点项目(04JC14073)

作者简介: 窦建华(1982 -), 男, 硕士生, 主研方向: 供应链仿真, 决策支持系统; 林杰, 研究员、博导

收稿日期: 2006-06-28 E-mail: jhdou@163.com

表 2 工序成本表

	P ₁	P ₂		P ₃	P ₄			P ₅		P ₆
		S ₁	S ₂		S ₃	S ₄	S ₅	S ₆	S ₇	
订单 1	300	32.76	31.88	9	42.29	44.74	42.17	18.75	21.71	90
订单 2	100	8.74	10.49	3	16.37	13.79	14.17	5.9	6.97	30
订单 3	300	30.42	31.16	9	43.51	38.87	45.45	15.94	22.6	90
订单 4	400	42.1	43.08	12	74	58.24	60.67	25.44	29.35	120
订单 5	200	21.15	20.53	6	28.78	26.88	28.94	11.01	15.13	60
订单 6	500	47.22	54.2	15	83.72	74.52	79.02	27.56	34.3	150

问题的关键在于合理地安排生产顺序和选择供应商,使得该供应链协调达到较满意的效果。

2 算法设计

鉴于蚁群算法在求解类似问题时的经验,将其加以改进便可很好地用于解决上述问题。算法设计如下:

步骤 1 将该问题映射到有向图,将每个订单中每个子任务的可选方案视为节点。节点之间的有向弧表示节点之间的信息素浓度,各弧的初始信息素浓度相等。

步骤 2 在虚拟起始点生成一定数量的一代蚂蚁。每只蚂蚁都根据图 1 所示工序约束关系判断可行节点,完成对图的遍历。对可行节点,蚂蚁 k 按照式(1)所示伪随机比例状态迁移规则移动:

$$s = \begin{cases} \arg \max_{i \in F^k(t)} (\tau_{ii}(t)), q < q_0 \\ s', q \geq q_0 \end{cases} \quad (1)$$

其中, $F^k(t)$ 表示 t 时刻蚂蚁 k 的可行节点域; $\tau_{ii}(t)$ 表示 t 时刻当前节点 i 指向邻域节点 j 弧上的信息素浓度; q 为 0~1 之间的随机数; q_0 为一个预设的 0~1 之间的数。蚂蚁以概率 q_0 移动到信息素浓度最大的相邻节点;以概率 $1-q_0$ 进行偏向性探索。此时, i 节点的蚂蚁以概率 $P_{is}^k(t)$ 移动到邻域节点 s'。

$$P_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha \sigma_j^\beta}{\sum_{l \in F^k(t)} (\tau_{il}(t))^\alpha \sigma_l^\beta}, j \in F^k(t) \\ 0, j \notin F^k(t) \end{cases} \quad (2)$$

式(2)中, σ_j 为节点 j 的启发信息, τ_{ij} 为信息素浓度与启发信息的权重因子。启发信息

$$\sigma_j = \max \left[\left(1 - \frac{t_j - \bar{t}}{\bar{t}} \right) \left(1 - \frac{c_j - \bar{c}}{\bar{c}} \right), 0.01 \right]$$

其中 t_j 、 c_j 表示节点 j 的加工时间和加工成本, \bar{t} 、 \bar{c} 表示与节点 j 对应相同订单及子任务的所有节点平均加工时间和加工成本。启发信息使得蚂蚁倾向于选择在处理相同任务时具有更少加工时间和加工成本的节点。

每只蚂蚁完成一次有向图遍历后将获得一个节点序列,此序列代表了每道工序选择的合作供应商以及各道工序之间的调度顺序。

步骤 3 在满足时间约束和资源约束(一个供应商不能同时处理 2 个子任务)的条件下,按照节点序列顺序推出各工序的最早开工时间及各订单的最早完工时间。为减少在制品库存,基于各订单的最早完工时间,逆向推出各工序的最晚开工时间作为其实际开工时间。

设 N: 每个订单都可以分解为工序的数量; $l_i (i=1,2,\dots,6)$: 每个订单的最迟交货期; $c_{i,j,m}$: 完成第 i 个订单第 j 道工序的第 m 种方案; $sTime : c_{i,j,m}$ 的开始时间; $oTime : c_{i,j,m}$ 的完成时间; $pCost : c_{i,j,m}$ 的生产成本; $cStock : c_{i,j,m}$ 在制品库存成本(此例所有在制品库存成本均计为该在制品生产成本的 0.5%)。将各道工序的开工时间、加工成本以及库存成本带入式(3),计

算出此协调方案的评价指标。

$$\begin{aligned} \min E = & \\ & w_1 \left(\sum_{i=1}^M \sum_{j=1}^N pCost_{i,j,m_j} + \sum_{i=1}^M \sum_{p=1}^N \sum_{q=1}^N (sTime_{i,q,m_{q_i}} - oTime_{i,p,m_p}) k_{pq} cStock_{i,p,m_p} \right) \\ & + w_2 \max_i (oTime_{i,N,m_N}) + P \sum_{i=1}^M \max(oTime_{i,N,m_N} - l_i, 0) \end{aligned} \quad (3)$$

式中, k_{pq} 为子任务约束变量,若子任务 p 为 q 的前置任务,则 $k_{pq}=1$, 否则 $k_{pq}=0$; w_1 与 w_2 为时间与成本的权重因子; P 为交货期延迟惩罚系数; m_{ij} 表示订单 i 的第 j 个子任务最终选定的方案编号; $oTime_{i,N,m_N}$ 表示订单 i 最后一道工序的完成时间。

步骤 4 更新节点间的信息素浓度:

$$\tau_{ij}(t + \Delta t) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t + \Delta t) \quad (4)$$

式(4)中,第 1 部分为每代蚂蚁完成一次遍历,信息素挥发后的剩余量;第 2 部分为信息素奖励函数,按式(5)给出:

$$\Delta\tau_{ij}(t + \Delta t) = \begin{cases} \frac{Q}{\min E_k}, \text{方案 } ij \text{ 属于历史最优方案 } k \\ 0, \text{其他} \end{cases} \quad (5)$$

其中 Q 为预设参数。

步骤 5 比较并储存该代蚂蚁的最优方案。判断是否达到结束条件,若没有则转到步骤 2 进行迭代,否则退出算法。

3 仿真设计

3.1 MAS 结构

仿真系统设计直接影响算法的效率和效果,本文选用 JADE(Java Agent Development Framework)作为开发环境。

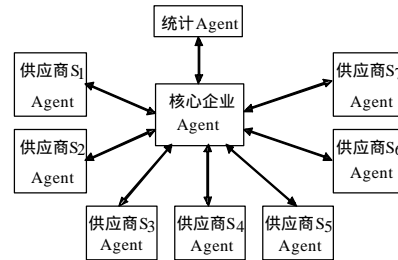


图 2 MAS 结构

如图 2 所示,系统分为 3 个部分。第 1 部分为核心企业 Agent,其主要功能有 5 个:(1)生成一定数量的“蚂蚁”;(2)初始化和更新信息素浓度;(3)控制蚂蚁完成路径遍历;(4)计算评价函数;(5)与统计 Agent 之间的参数传递。第 2 部分为 7 个供应商 Agent,其主要功能是与核心企业 Agent 交互信息,告知核心企业其自身生产能力。第 3 部分为统计 Agent,其功能有 3 个:(1)统计所有代蚂蚁的最优评价函数以及最优路径;(2)控制算法结束;(3)打印输出。

3.2 核心企业 Agent

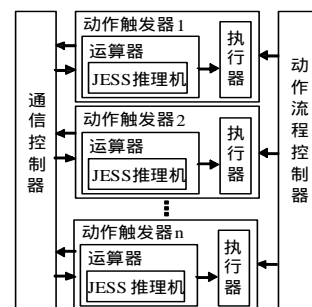


图 3 核心企业 Agent 结构

核心企业 Agent 由动作触发器、动作流程控制器、通信

控制器 3 部分组成。其中 动作触发器又由执行器和包含 JESS 推理机的运算器组成(图 3)。

3.2.1 动作触发器

动作触发器写入了核心企业 Agent 的各种动作, 每一个动作都对应一个动作触发器, 如初始化和更新信息素浓度、路径遍历、供应商的选择等。每个动作触发器中又包含了一个运算器和一个执行器。

运算器写入了该动作推理和运算的方法。一些运算器中嵌入了 JESS(Java Expert System Shell)推理机。JESS 是一款优秀的专家系统开发平台, 可进行正反推理, 且其“以空间换时间”的特点使其具有很高的执行效率。它采用产生式规则作为基本的知识表达模式, 在执行某些逻辑性较强、规则较多的动作时可将 JESS 推理机集成入动作触发器, 充当类似于知识库的角色。如本例在完成路径遍历动作时, 将订单分解结构与约束关系写入 JESS 推理机, 集成入路径遍历动作触发器, 蚂蚁便可进行可行节点的判断。JESS 与 JADE 都是基于 Java 的平台, 可将 JESS 代码直接嵌入到 JADE 中。下面一段代码给出了 JESS 定义规则及其与 JADE 之间的数据传递方法。

```
Rete r=new Rete(); //创建 Rete 类的一个对象 r
r.store("STR",str); //将 JADE 中字符串 str 传递给 JESS 中变量
//STR
String rule="(defrule rp1"+"(task (p1 0))"+"=>"+"(bind ?p1 2))";
//定义规则
r.executeCommand("(bind ?str (fetch STR))"); //JESS 得到 STR 字
//字符串
...
String Str_new=v.stringValue(r.getGlobalContext()); //将 JESS 中
//字符串 STR 传递给 JADE 中变量 str
r.run();
```

当动作参数计算完成后将由执行器去触发并执行该动作, 同时执行器也担负着控制动作节拍的任务。

3.2.2 流程控制器

流程控制器是核心企业 Agent 的调度中心。在 Agent 中, 它就是一个 setup()方法, 包含了动作触发器触发条件和触发顺序。

```
protected void setup()
{
    if(i==0)
    {
        addBehaviour(new ini());
        addBehaviour(new count());
    }
}
```

以上代码给出了一个流程:判断变量 i 是否为 0, 若为 0, 触发动作 ini; 若不为 0, 向下执行, 触发动作 count。

3.2.3 通信控制器

通信控制器用于控制在不同的 Agent 之间进行的参数传递。JADE 通过标准 ACL 语言格式进行消息传递。它提供了 get()和 set()方法来处理消息中的发送者、接收者、通信目标、通信内容等部分。下面给出的代码表示将变量 a 的值从当前 Agent 传递给另一台机器 Bill 上名为 statant 的 Agent。

```
ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
AID statant=new AID();
statant.addAddresses("http://Bill:7778/acc");
statant.setName("statant@Bill:1099/JADE");
msg.setProtocol("jade.mtp.http.MessageTransportProtocol");
msg.setContent(a);
```

```
send(msg);
```

3.3 供应商 Agent

由于本例只研究核心企业可以获得供应商全部生产信息的情况, 因此 7 个供应商具有相同的较为简单的结构, 即生产信息表和通信控制器。生产信息表中存放的是表 1、表 2 中列举的供应商的生产信息数据, 通信控制器定义了通信参数和通信节拍。

3.4 统计 Agent

统计 Agent 由统计器、通信控制器和打印机 3 部分组成。统计器接收比较每代蚂蚁的评价函数, 记录较优者及其路径, 并据此计算出最优路径上所需更新的信息素浓度。同时统计器内还写入了算法退出条件。通信控制器定义了通信参数和通信节拍。打印机负责输出控制。

4 仿真结果及分析

4.1 参数设置与仿真结果

某次试验每代蚂蚁个数为 100 只, 选取成本权重因子 $w_1=0.029$, 时间权重因子 $w_2=0.971$, 初始信息素浓度为 0.2, 最小信息素浓度为 0.05, 信息素挥发率 $\rho=0.15$, $Q=15$; 蚂蚁按最大信息素浓度移动概率 $q_0 = \begin{cases} (代数-1) \times 0.005, & \text{蚂蚁代数} \leq 80 \\ 0.8, & \text{蚂蚁代数} > 80 \end{cases}$; 共迭代 500 代。

在此次试验中, 最优方案出现在第 377 代。最优评价指标 minE 为 169.832; 完成时间为 87 个时间单位; 库存成本为 5.03 个价格单位; 总生产成本为 2 938.250 个价格单位。图 4 为此次仿真过程中前 200 代蚂蚁选择的最优方案的评价指标曲线。

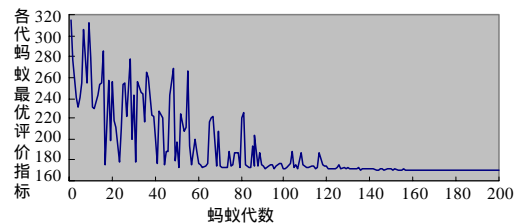


图 4 前 200 代最优评价指标曲线

4.2 仿真结果分析

蚁群算法的特点, 使其对于相同的数据, 每次运算结果也不尽相同, 对不同的参数设置更是如此。我们认为 minE 小于 171, 完成时间为 87 个时间单位即为可以接受的较优方案。在对本例进行的 30 次仿真中, 有 27 次在 100 代以内产生了较优方案, 其余 3 次均在 150 代以内产生了较优方案。在 160 代左右, 评价指标逐步收敛, 时间指标很难再进行优化, 此后主要是在生产成本和库存成本之间进行优化搜索。在 30 次仿真中, 最终所有的 minE 均收敛在 169.820 到 169.850 之间。

在进行此试验的同时还利用了其他方法对此例进行了仿真和运算。从表 3 看出, 本文的仿真方法具有运算速度快、运算结果优化、稳定的特点。

表 3 不同算法结果对比表

	语言	运行最优结果		100 代内出现较优解次数	出现较优解平均时间
		时间	综合指标		
蚁群	Java	87	169.832	27/30	448m
遗传	c	87	169.836	21/30	502m
专家系统		88	172.686		

(下转第 192 页)