

文章编号:1001-9081(2006)11-2631-04

基于蚁群算法的大规模定制供应链调度优化研究

孙 靖, 林 杰

(同济大学 经济与管理学院, 上海 200092)

(sunjing091@gmail.com)

摘 要:为解决大规模定制模式下客户订单分离点后的动态供应链调度问题,提出了包括供应商选择及企业合作时序安排的优化调度模型,设计了基于蚁群算法的求解过程。通过多组数据实验及结果比较分析,对模型算法的有效性、稳定性进行了验证。

关键词:大规模定制;客户订单分离点;供应链调度;优化模型;蚁群算法

中图分类号: TP18;F273 **文献标识码:** A

Study of supply chain optimization scheduling in mass customization based on ant colony algorithm

SUN Jing, LIN Jie

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

Abstract: To solve the problem of the dynamic supply chain optimization scheduling after customer order decoupling point in mass customization, an optimization scheduling model including supplier selection and inter-enterprise scheduling was proposed. Then an ant colony algorithm based solution was designed. Finally by practical manufacture data and methods comparison, the validity and stability of this algorithm were verified.

Key words: mass customization(MC); customer order decoupling point(CODP); supply chain scheduling; optimization model; ant colony algorithm

0 引言

大规模定制(Mass Customization, MC)是一种有机结合大规模生产和定制生产优势的新型生产模式,它在尽可能不牺牲企业生产效率的前提下,了解并满足消费者的个性需求,将成为未来的主流生产模式。目前关于 MC 的研究主要集中在产品族与产品平台管理^[1]、客户订单分离点(CODP)延迟策略^[2]以及一些供应链驱动机制、模型的概念性研究^[3,4]等方面,针对具体 MC 供应链调度的文献相对较少。

文献[5,6]针对 MC 生产环境下随机需求和随机资源约束等特点,研究了其供应链调度优化问题,但并未考虑 CODP 前后供应链调度需求差异,同时也没有给出优化模型的求解算法。CODP 是 MC 供应链中定制活动的开始点,是企业活动由基于库存的推动式生产转向基于客户需求拉动式生产的转折点。CODP 之前,核心制造商根据市场预测协调供应商对产品的标准组件进行库存生产,这实质上就是传统的大规模生产模式,已有大量成熟理论可供参考;CODP 之后,核心制造商根据客户需求选择标准组件进行装配、对定制组件进行生产或采购,这部分调度工作是决定生产成本及供应链响应速度的关键,也是本文研究重点。Chiung Moon 等^[7]基于企业之间充分合作,从产品总体拖期时间最小的角度研究了类似于 MC 供应链 CODP 之后的动态供应链调度问题,建立供应链优化模型,并采用遗传算法进行求解。但文中将供应链各单位的生产设备视如同一家企业进行综合调度,缺乏实际可操作性。Edgar Perea-Lopez 等^[8]采用混合整数规划方法解决此

类供应链优化问题,但随着问题规模的增大,混合整数规划将难以在可接受时间内求得最优,甚至是可行解。

CODP 之后的供应链调度属于敏捷协同制造问题。与 CODP 之前相比,CODP 之后的供应链调度表现出更强的动态及实时特性,调度方案优劣直接影响 MC 供应链的经济效益及客户满意度。目前关于此类动态调度问题的研究理论尚不成熟。针对这一不足,本文从 MC 供应链实际情况出发,建立供应链优化调度模型,并设计了基于蚁群算法的模型求解算法,最后通过实际生产数据和结果比较分析对模型算法的有效性、稳定性进行了验证。

1 问题描述及建模

本文以某核心制造商协调多个供应商共同完成若干定制任务的情况为例进行研究。核心制造商需要将时间域值 T_0 内接到的订单进行分类、合并,并对企业内部及合作供应商进行调度。调度过程包括供应商选择、各企业内任务排序及各子任务开工时间确定。以图 1 为例,假设核心制造商可以将某类定制订单分解为 6 个子任务,其中子任务 P1, P3, P6 由核心制造商完成,子任务 P2, P4, P5 是为产品提供相应的定制外购件。假设针对各类定制件,核心制造商已经建立了合作供应商库,并与库内企业保持良好的合作与信任关系。在接到用户订单后,核心制造商需要迅速从供应商库中选择适当的合作者,并就合作时序进行协商、调度。当考虑多个订单的综合调度时,各订单子任务之间约束关系复杂,任何选择及调度偏差都将导致订单生产时间、生产成本及库存成本的变化,

收稿日期:2006-05-15;修订日期:2006-07-24 基金项目:上海市科委重点资助项目(04JC14073)

作者简介:孙靖(1977-),男,浙江上虞人,博士研究生,主要研究方向:供应链调度、决策支持系统;林杰(1967-),男,四川渠县人,教授,博士生导师,主要研究方向:决策支持系统、电子商务、供应链仿真与决策、群体决策支持系统。

影响供应链效率。

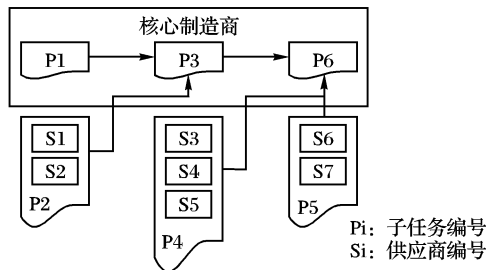


图1 订单分解结构示意图

1.1 研究假设

后续章节的研究基于以下假设：

- 1) 本文主要研究供应链企业间调度合作,而不涉及各企业内部调度流程,因此可将企业在不涉及外部合作情况下的全部工序(包括加工、运输和存储等)合并为一个子任务表示,不考虑各工序之间的调度关系。
- 2) 每个定制子任务只交由一个供应商完成,且一个企业不能并行处理不同订单的相同子任务。
- 3) 对于特定定制任务,核心制造商内各子任务加工成本、加工时间及合作供应商的采购成本、订货提前期固定且已知,后文中将统一用加工成本和加工时间表述。
- 4) 忽略企业之间的订货、运输时间及成本,仅考虑加工、库存成本及加工时间。
- 5) 核心企业的生产目标是在尽可能满足各订单交货时间约束的条件下,尽早完成全部订单任务,并降低生产及在制品库存成本。

1.2 调度模型建立

设核心制造商在时间域值 T_0 内接到 M 个订单,每个订单都可以分解为 $N_i (i = 1, 2, \dots, M)$ 个子任务,每个订单的最迟交货期为 $l_i (i = 1, 2, \dots, M)$ 。为便于描述进行如下定义：

定义1 子任务约束矩阵 K^i 为存储订单 i 各子任务之间时间约束关系的 N_i 阶方阵：

$$K^i = (k_{p,q}^i) = \begin{cases} 1, & \text{若订单 } i \text{ 中子任务 } p \text{ 为子任务 } q \text{ 的前置任务} \\ 0, & \text{其他} \end{cases}$$

定义2 方案描述矩阵 C^i 为一个结构矩阵,是描述订单 i 各子任务可选方案参数的集合：

$$C^i = (c_{j,m}^i) = \langle \text{Executor}, sTime, oTime, Cost, cStock \rangle_{j,m}^i$$

式中, $c_{j,m}^i$ 表示完成第 i 个订单第 j 个子任务的第 m 种方案; $Executor$ 表示 $c_{j,m}^i$ 的执行者; $sTime$ 表示 $c_{j,m}^i$ 的开工时间; $oTime$ 表示 $c_{j,m}^i$ 的加工时间; $Cost$ 表示 $c_{j,m}^i$ 的加工成本; $cStock$ 表示 $c_{j,m}^i$ 的单位时间库存成本。以上参数中,除 $sTime$ 外,在进行调度优化前均为已知。

则核心企业的优化调度模型为：

$$\begin{aligned} \min E = & w_1 \left(\sum_{i=1}^M \sum_{p=1}^{N_i} \sum_{q=1}^{N_i} (sTime_{q,m_{iq}}^i - sTime_{p,m_{ip}}^i - \right. \\ & \left. oTime_{q,m_{iq}}^i) k_{p,q}^i + cStock_{p,m_{ip}}^i + \sum_{i=1}^M \sum_{j=1}^{N_i} Cost_{j,m_{ij}}^i \right) + \\ & w_2 \max_i (sTime_{N_i, m_{iN_i}}^i + oTime_{N_i, m_{iN_i}}^i) + \\ & P \sum_{i=1}^M \max (sTime_{N_i, m_{iN_i}}^i + oTime_{N_i, m_{iN_i}}^i - l_i, 0) \end{aligned} \quad (1)$$

s. t.

$$sTime_{j,m_{ij}}^i - sTime_{j',m_{ij'}}^i \geq oTime_{j,m_{ij}}^i, \text{ 若 } k_{j',j}^i = 1 \quad (2)$$

$$sTime_{j,m_{ij}}^i - sTime_{j',m_{ij'}}^i \geq oTime_{j',m_{ij'}}^i \cdot \text{or. } sTime_{j',m_{ij'}}^i - sTime_{j,m_{ij}}^i \geq oTime_{j,m_{ij}}^i, \text{ 若 } Executor_{j,m_{ij}}^i = Executor_{j',m_{ij'}}^i \quad (3)$$

式中, w_1 与 w_2 为成本与时间的权重因子; P 为交货期延迟惩罚因子; m_{ij} 表示订单 i 的第 j 个子任务最终选定的方案编号; $sTime_{N_i, m_{iN_i}}^i$ 和 $oTime_{N_i, m_{iN_i}}^i$ 分别表示订单 i 最后一个子任务的开工及加工时间。

调度模型的第一部分为成本评价因子;第二部分为时间评价因子;第三部分为订单延期惩罚项。式(2)为同一订单各子任务之间的时间约束;式(3)表示企业资源约束。

2 基于蚁群算法的求解过程设计

典型车间调度问题 Flow-shop、Job-shop 的求解已被证明是 NP 难题,随着问题规模扩大,通过传统最优化及规划算法无法在可接受时间内求得最优,甚至是可行解。与之相比,MC 供应链调度更具柔性,不仅涉及合作供应商选择而且子任务间的约束关系更加复杂。本文根据供应链调度中合作企业选择及子任务排序与图论路径选择问题的结构相似性设计了用于求解 1.2 节模型的蚁群求解算法。

2.1 蚁群算法简介

蚁群算法是由意大利学者 M. Dorigo 等人于 1991 年提出的一种模拟蚂蚁觅食过程寻求最短路径的群体智能算法。在离开巢穴寻找食物的过程中,每只蚂蚁都会在身后留下一定量的信息素,用于提示回巢路径,并为其他蚂蚁指引食物方向。信息素是一种易挥发的化学物质,由于蚂蚁经过较短路径时可以在较短时间内走完全程,相同时间内较短路径上可以积累较多信息素,以后出发的蚂蚁将倾向于选择信息素比较浓的路径,大量蚂蚁经过一定时间后将集中选择较短的路径。模拟蚁群的这一群体行为可以解决众多寻优问题。

蚁群算法是一种启发式随机搜索方法。同其他启发式算法一样,通过由候选解组成的群体进化过程来寻找优解。该方法最初应用于著名的旅行商问题(TSP)取得较好结果。蚁群算法的正反馈、分布式以及贪婪式启发搜索等特性保证了问题求解的快速性、全局优化性以及鲁棒性和通用性,很快引起相关领域研究者的注意。在求解其他组合优化问题方面也取得了较好效果。其中生产调度方面,目前蚁群算法已在车间调度问题上有较多研究应用^[9, 10],而在供应链调度方面的应用相对较少。

2.2 相关定义

为便于 MC 供应链调度问题的蚁群算法求解,通过以下定义将 1.2 节模型映射到图中。

定义3 图为表示 MC 供应链调度问题的有向图：

$$G = (N, A)$$

式中, $N = (n_{j,m}^i)$ 是图中节点集合,每个节点对应方案描述矩阵 $C^i (i = 1, 2, \dots, M)$ 中的一个元素; A 是图中弧的集合,表示各节点间的关系。对于任意节点 $n_{j,m}^i$,由它出发的弧指向所有满足 $i' \neq i$. or. $(i' = i \text{ and. } k_{j',j}^i \neq 1)$ 的 $n_{j',m'}^{i'}$ 。

定义4 虚拟起始点 $sPoint$ 为图中一个仅与子任务约束矩阵 $K^i (i = 1, 2, \dots, M)$ 中全为“0”的列对应子任务节点相连的节点,虚拟起始点不占用任何操作时间及操作成本。

2.3 算法描述

将 1.2 节模型映射到图中后,运用类似 TSP 的搜索方法

对问题进行求解。具体算法步骤如下:

步骤 1 初始化图中各弧的信息素浓度。各弧初始信息素浓度相同。

步骤 2 在虚拟起始点 $sPoint$ 生成一代人工蚁,将图中所有节点(虚拟起始点除外)加入各人工蚁的未访问节点表。各人工蚁按照一定规则分别构造出一条遍历所有子任务节点的路径。

步骤 3 路径构造的每一步,人工蚁 k (当前所在节点 n)均按照以下规则选择移动到邻域节点 s 。

对于未访问节点表中的各节点 l (对应 $c_{j,m}^i$),判断在其订单位子任务约束矩阵 K^i 中第 j 列为“1”的元素横坐标子任务是否均已完成?如均已完成则将此节点加入人工蚁 k 在 t 时刻的可行节点表 $F^k(t)$,否则为禁忌节点。

在可行节点中,按照以下伪随机比例状态迁移规则进行移动选择:

$$s = \begin{cases} \arg \max_{l \in F^k(t)} (\tau_{nl}(t)), & q < q_0 \\ s', & q \geq q_0 \end{cases} \quad (4)$$

式中, $\tau_{nl}(t)$ 表示 t 时刻当前节点 n 指向邻域节点 l 弧上的信息素浓度; q 为 $0 \sim 1$ 之间的随机数; q_0 为一个预设的 $0 \sim 1$ 之间的数。在搜索过程中,人工蚁以概率 q_0 移动到信息素浓度最大的邻域节点,即完全按照信息素的指引来选择路径,这种方式称为知识利用;与之对应,人工蚁以概率 $1 - q_0$ 进行偏向性探索。此时,人工蚁 k 以概率 $P_{ns'}^k(t)$ 移动到邻域节点 s' 。

$$P_{ns'}^k(t) = \begin{cases} \frac{\tau_{ns'}(t)}{\sum_{l \in F^k(t)} \tau_{nl}(t)}, & s' \in F^k(t) \\ 0, & s' \notin F^k(t) \end{cases} \quad (5)$$

式中, $\tau_{ns'}(t)$ 是弧 $arc(n, s')$ 上的信息素浓度。

步骤 4 人工蚁移动到一个新的节点 s (对应 $c_{j,m}^{i'}$)后,则将其未访问节点表中与订单 i' 子任务 j' 对应的节点全部删除。检查未访问节点表是否为空?为空则说明此人工蚁完成了遍历,记录路径;否则转步骤 3。

步骤 5 人工蚁 k 完成遍历后,按照以下规则计算路径质量。

1) 在满足时间约束和资源约束的条件下按照节点路径顺序,采用顺推法确定各子任务的最早开工时间及各订单的最早完工时间。按照人工蚁的遍历顺序依次取出子任务。对于新取出的子任务 $c_{j,m}^i$,查找在其订单位子任务约束矩阵 K^i 中第 j 列为“1”元素的横坐标子任务完成时间,取其最大值 $TpMax$;查找与此子任务由同一个供应商或核心企业工序完成,且遍历排序在它之前的所有子任务完成时间,取其最大值 $RpMax$ 。取 $\max(TpMax, RpMax, 0)$ 作为子任务 $c_{j,m}^i$ 的开始时间。

2) 为减少在制品库存,基于各订单的最早完工时间,按照路径序列的逆序,采用逆推法确定各子任务的最晚开工时间。按照人工蚁的遍历逆序依次取出子任务。对于每个订单的最后一个子任务直接取 a 中算出的最早完工时间逆推开工时间;其他新取出的子任务 $c_{j,m}^i$,查找在其订单位子任务约束矩阵 K^i 中第 j 行为“1”元素的列坐标子任务开工时间,取其最小值 $TpMin$;查找与此子任务由同一个供应商或核心企业工序完成,且遍历排序在它之后的所有子任务开工时间,取其最小值 $RpMin$ 。取 $\min(TpMin, RpMin)$ 作为子任务 $c_{j,m}^i$ 的完工时间并逆推开工时间。

3) 将各参数代入式(1),求出人工蚁 k 遍历路径的评价指标

E_k 。

步骤 6 信息素更新及路径存储。每一代人工蚁全部完成遍历及路径评价后,按照以下规则更新图 G 中各弧 A 的信息素浓度:

$$\tau_{ij}(t + \Delta t) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t + \Delta t) \quad (6)$$

式中,第一部分为每代人工蚁完成一次遍历,信息素挥发后的剩余量;第二部分为信息素奖励函数,按以下公式给出:

$$\Delta\tau_{ij}(t + \Delta t) = \begin{cases} \frac{Q}{E_o}, & \text{路径 } ij \text{ 属于历史最优路径 } O \\ 0, & \text{其他} \end{cases} \quad (7)$$

式中 Q 为预设参数, E_o 为历史最优路径 O 的评价指标。

比较各代最优路径,存储其中的最优值。

步骤 7 循环执行。如果已经收敛于最优值或者达到最大蚁群代数则退出循环,输出最优路径及各子任务开工时间,即最终调度方案;否则转步骤 2 循环执行。

2.4 算法改进

蚁群算法虽然在求解组合优化问题中具有众多优越特性,但随着问题规模增大会逐渐暴露出搜索时间长、易陷入局部最优解等缺陷^[11]。为解决这些问题,本文在 2.3 基本算法的基础上设计了以下优化方案:

优化方案 1 增加启发信息。为引导人工蚁在搜索过程中尽可能选择那些在处理相同任务时需要更少加工时间和加工成本的节点,将 2.3 节算法的步骤 3 中式(5) 改为:

$$P_{ns'}^k(t) = \begin{cases} \frac{(\tau_{ns'}(t))^\alpha \sigma_{s'}^\beta}{\sum_{l \in F^k(t)} (\tau_{nl}(t))^\alpha \sigma_l^\beta}, & s' \in F^k(t) \\ 0, & s' \notin F^k(t) \end{cases} \quad (8)$$

式中, σ_l 为节点 l 的启发信息, α, β 为信息素浓度与启发信息的权重因子。

启发信息 $\sigma_l = \max\left[\left(1 - \frac{t_l - \bar{t}}{t}\right)\left(1 - \frac{c_l - \bar{c}}{c}\right), 0.01\right]$, 其中 t_l, c_l 表示节点 l (对应 $c_{j,m}^i$) 的加工时间和加工成本, \bar{t}, \bar{c} 表示订单 i 子任务 j 所有可选方案的平均加工时间和加工成本。

优化方案 2 限制最小信息素浓度。为防止蚁群完全陷入局部最优解,对节点间的信息素浓度设定最小值限制。由于算法中节点间的信息素浓度实际上也有最大值限制($\tau_{max} = Q/E^*/\rho$),因此,人工蚁总有一定的几率跳出局部最优解。

优化方案 3 2.3 节算法的步骤 3 中知识利用与偏向性搜索的控制概率 q_0 对于算法搜索范围及收敛速度影响很大。为了在算法初期尽可能扩大搜索范围寻找优解,在算法后期加快收敛速度,重视局部寻优,将 q_0 设置为一个随蚁群代数增加而动态变化的量,从而优化寻优过程。

3 实践验证及分析

表 1 订单子任务加工时间表

子任务	P1	P2		P3	P4			P5		P6	Deadline	
		S1	S2		S3	S4	S5	S6	S7			
order1	3	3	5	1	2	2	3	3	5	2	40	
order2	6	9	8	2	2	3	4	6	7	4	60	
order3	6	5	5	2	4	5	4	10	14	4	75	
order4	9	12	10	3	5	4	5	9	10	6	75	
order5	9	8	10	3	4	9	8	13	9	6	60	
order6	18	16	20	6	11	17	8	3	3	15	12	50

为简化计算过程,便于调度结果优劣分析,本文选取实际 MC 供应链系统,抽取其中 6 个订单数据进行验证分析。这 6 个订单具有相同的订单分解结构(如图 1 所示),表 1 为订单子任务在各企业内的处理时间及最迟交货期,表 2 为各企业的生产成本,每单位时间的库存成本以在制品生产成本的 0.5% 计。

表 2 订单子任务加工成本表

子任务	P1	P2		P3	P4			P5		P6
		S1	S2		S3	S4	S5	S6	S7	
order1	123.67	13.26	14.17	21.54	7.42	5.81	6.95	29.61	34.81	30.83
order2	247.34	28.01	32.83	43.08	12.13	11.44	14.17	72.99	58.40	61.66
order3	247.34	26.62	33.46	43.08	12.89	11.57	15.47	62.40	59.20	61.66
order4	371.01	39.23	43.28	64.62	19.65	17.20	21.23	100.15	91.56	92.49
order5	371.01	39.69	50.64	64.62	19.21	17.75	23.38	108.39	88.45	92.49
order6	742.02	79.39	89.94	129.24	44.16	41.63	45.79	219.18	165.58	184.98

采用 2.3 节算法,参数选择按照文献[12]的方法通过多次试验获得。选取成本权重因子 $w_1 = 0.015$, 时间权重因子 $w_2 = 0.985$, 交货期延迟惩罚因子 $P = 10$; 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 $\rho = 0.15$, 最优路径奖励因子 $Q = 25$; 人工蚂蚁按最大信息素浓度移动概率 $q_0 = \begin{cases} (\text{代数} - 1) \times 0.005, & \text{蚁群代数} \leq 160 \\ 0.8, & \text{蚁群代数} > 160 \end{cases}$, 进行偏向性探索时 $\alpha = 1, \beta = 0.5$; 每代人工蚂蚁为 75 只, 共迭代 600 代。图 2 所示为某次调度优化过程中前 200 代的各代最优路径评价指标变化曲线。此次调度计算的最优路径出现在第 529 代, 其评价指标为 115.182, 全部订单在 58 个单位时间内完成, 总生产成本为 3840.26 个单位成本, 库存成本为 29.90 个单位成本, 优化调度方案如图 3 所示。

蚁群算法属于启发式随机搜索算法,即使对于相同的订单数据,每次运算结果也不尽相同,算法稳定性是算法优劣的重要评价指标。蚁群算法的参数设置对于收敛速度及结果稳定性至关重要。经过 50 次试验,在上述参数情况下,蚁群一般在 100 代以内就可产生能够接受的较优解,在 100~200 代之间评价指标逐步收敛,200 代以后主要通过局部搜索优化生产及库存成本,生产时间再次大幅度优化的可能性不大。上例中,29 代即出现了较优解:评价指标 115.741,全部订单在第 58 个单位时间完成,总生产成本为 3865.79 个单位成本,库存成本为 41.65 个单位成本;200 代内的最优解为:评价指标 115.530,全部订单在第 58 个单位时间完成,总生产成本为 3861.62 个单位成本,库存成本为 32.13 个单位成本。50 次试验中有 41 次全部订单完成时间收敛于 58 个单位时间,9 次订单完成时间收敛于 59 个单位时间,平均最优评价指标为 115.4973,算法具有较好的稳定性。

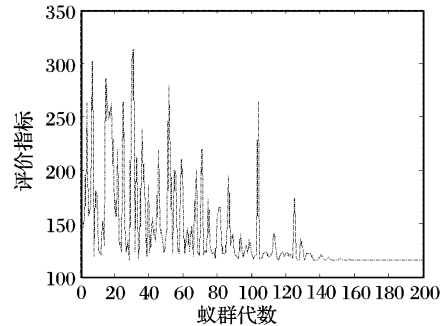


图 2 蚁群各代最优路径评价指标变化曲线

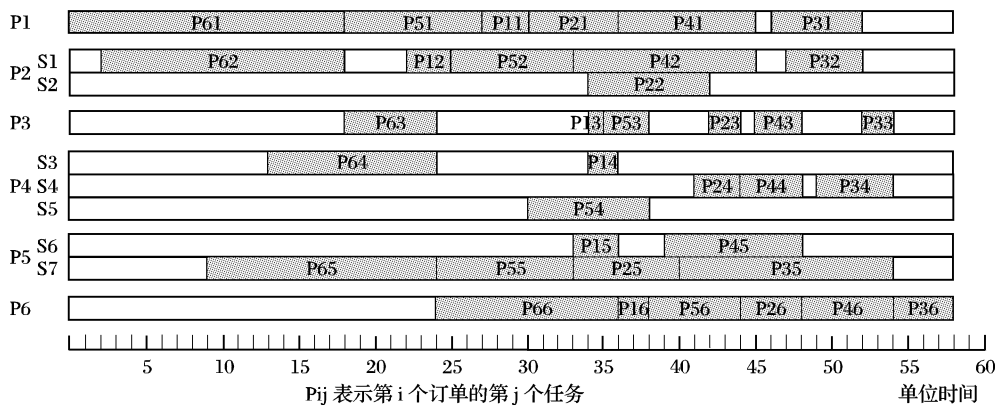


图 3 优化调度方案

表 3 算法调度结果比较表

序号	交货期	本文算法					专家系统				
		完成时间	加工成本	库存成本	延迟天数	总体评价	总体评价	完成时间	加工成本	库存成本	延迟天数
1	[90 90 90 90 90]	58	3837.58	24.34	0	115.059	58	3820.27	18.18	0	114.707
2	[40 60 75 75 60 50]	58	3840.26	29.90	0	115.182	59	3817.98	47.28	0	116.094
3	[35 42 54 75 62 58]	61	3830.45	27.07	0	117.948	63	3817.08	20.32	0	119.616
4	[20 40 40 60 65 90]	63	3834.55	13.03	0	119.769	69	3817.98	23.32	0	125.585
5	[20 18 58 40 36 54]	69	3842.09	10.06	15	275.747	67	3817.98	9.64	0	303.409

为检验 2.3 节算法对调度方案的优化程度以及对订单任务的适应能力,即对各类订单结构及约束的普遍有效性,本文还采用了多组订单数据,分别进行调度运算,并与基于先交货先生产的专家系统调度方案进行比较。篇幅所限,此处仅列出上例订单数据在不同交货期限限制条件下的比较结果(见表

3),此结果在类似比较研究中很具代表性。由表 3 可见,专家系统在加工成本优化方面具有较好的表现特性,但在时间特性优化方面则明显劣于本文算法,在库存优化方面两种调度算法各有优劣。从总体评价来看,专家系统仅在第一个交货 (下转第 2638 页)

我们从查准率和查全率两个方面对三种方法作比较。在实验过程中,该用户分别输入10种不同的检索词进行检索,查全率计算如表2所示。我们取根据每一种检索词检索得到的前15个页面来计算查准率,实验数据如表3。

通过表2可以看出,利用ontology对用户查询进行语义扩展后查全率明显提高。由表3可以看出,利用用户描述文件对检索结果重排序后查准率明显提高。综上所述,本文方法查全率与查准率均明显高于方法一,与方法二比较,查全率有显著提高,但查准率略有下降,这主要是由于分析数据过少造成误差。

5 结语

本文在语义分析和个性化检索研究的基础上提出了利用用户ontology实现个性化检索的方法,并对用户ontology的构建过程作了详细说明,最后用实验说明了本文方法的有效性。

虽然本文从语义角度对个性化检索作了有益的探索,但是仍存在很多不足之处,有待于进一步探索研究:1)由于用户行为是随机的,2.1中的假设2并不总是成立,这就有可能将用户不感兴趣的网页也作为感兴趣的网页,从而使构建的用户描述文件和用户ontology不够准确,进而造成系统查准率降低,检索结果不符合用户要求。2)文章只是讨论了基本语义关系的提取。对于复杂的语义关系,如何制定相应的提取规则,才能使提取出的语义关系完整而准确未作讨论。3)由于技术水平限制,实验设计不太完善,导致实验结果存在较大误差。

参考文献:

- [1] 中国搜索[EB/OL]. <http://www.zhongsou.com>, 2006.
- [2] WU Y-H, CHEN Y-C, CHEN ALP. Enabling personalized recommendation on the web based on user interests and behaviors[A]. KLAS W, ed. Proceedings of the 11th International Workshop on

Research Issues in Data Engineering[C]. Los Alamitos, CA: IEEE CS Press, 2001. 17-24.

- [3] 赵仲孟,袁薇,何世丽,等.个性化搜索引擎中用户模型智能调整算法的研究[J].计算机工程与应用,2005,41(24):184-187.
- [4] CHANDRASEKARAN B, JOSEPHSON JR, BENJAMINS VR. What are ontologies, and why do we need them?[J]. IEEE Intelligent Systems, 1999, 14(1): 20-26.
- [5] KAYED A, COLOMB RM. Extracting ontological concepts for tendering conceptual structures[J]. Data and Knowledge Engineering, 2002, 40(1): 71-89.
- [6] STOJANOVIC N. On the query refinement in the ontology-based searching for information[J]. Information Systems, 2005, 30(7): 543-563.
- [7] 王文杰,叶世伟.人工智能原理与应用[M].北京:人民邮电出版社,2004.226-257.
- [8] KOO SO, YEON LIM S, LEE SJ. Building an ontology based on hub words for information retrieval[A]. Proceedings of the IEEE/WIC International Conference on Web Intelligence[C]. Halifax, Canada: IEEE Computer Society, 2003. 466-469.
- [9] NOY NF, SINTEK M, DECKER S, et al. Creating semantic web contents with protege 2000[J]. IEEE Intelligent Systems, 2001, 16(2): 60-71.
- [10] CRUBEZY M, O'CONNOR M, PINCUS Z, et al. Ontology-centered syndromic surveillance for bioterrorism[J]. IEEE Intelligent Systems, 2005, 20(5): 26-35.
- [11] SWRL Editor FAQ[EB/OL]. <http://protege.stanford.edu/plug-ins/owl/swrl/>, 2006.
- [12] GNU Wget[EB/OL]. <http://www.gnu.org/software/wget/wget.html>, 2006.
- [13] 天网2006[EB/OL]. <http://e.pku.edu.cn>, 2006.

(上接第2634页)

期限制情况下优于本文算法,此时所有订单只要在90个单位时间内交货即可,对于此算例而言相当于没有严格的交货期限制。此时两种算法的完成时间相同,而专家系统在成本方面略优于本文算法。随着交货期限制条件变严,专家系统距离最优解(此类调度问题的最优解很难精确获得,这里指的最优解是实验过程中采用不同算法,通过多次运算获得的相对最优解)的差距越来越明显,而本文算法则表现出较好的寻优和适应能力。多组订单数据的比较结果均与上述情况类似,且订单数量越多,本文算法所表现的优势就越大。

4 结语

大规模定制CODP之后的供应链调度是整个供应链效率的关键。本文在分析此类供应链调度特点的基础上,建立了符合实际的优化调度数学模型,并设计了用于模型求解的蚁群算法。试验数据证明算法模型不仅调度优化效果良好,同时具有很好的适用性和稳定性。

参考文献:

- [1] JIAO J, TSENG MM, DUFFY VG, et al. Product family modeling for mass customization[J]. Computers & Industrial Engineering, 1998, 35(3/4): 495-498.
- [2] SUA JCP, CHANGB Y-L, FERGUSON M. Evaluation of postponement structures to accommodate mass customization[J]. Journal of Operations Management, 2005, 23(3/4): 305-318.

- [3] GHIASSIA M, SPERAB C. Defining the Internet-based supply chain system for mass customized markets[J]. Computers & Industrial Engineering, 2003, 45(1): 17-41.
- [4] GU XJ, QI GN, YANG ZX, et al. Research of the optimization methods for mass customization[J]. Journal of Materials Processing Technology, 2002, 129(1-3): 507-512.
- [5] 姚建明,周国华.大规模定制模式下供应链计划调度优化分析[J].管理科学学报,2003,6(5):58-64.
- [6] 姚建明,蒲云.基于动态生产能力约束的MC模式下供应链调度优化[J].系统工程,2005,23(2):25-30.
- [7] MOON C, KIM J, HUR S. Integrated process planning and scheduling with minimizing total tardiness in multi-plants supply chain[J]. Computers and Industrial Engineering, 2002, 43(1): 331-349.
- [8] PEREA-LOPEZ E, YDSTIE BE, GROSSMANN IE. A model predictive control strategy for supply chain optimization[J]. Computers and Chemical Engineering, 2003, 27(8/9): 1201-1218.
- [9] 姜桦,李莉,乔非,等.蚁群算法在生产调度中的应用[J].计算机工程,2005,31(5):76-78,101.
- [10] 胡燕海,马登哲,叶飞帆.制造系统通用作业计划与蚁群算法优化[J].计算机集成制造系统,2005,11(1):104-108.
- [11] 吴启迪,汪雷.智能蚁群算法及应用[M].上海:上海科技教育出版社,2004.57-93,101-118.
- [12] 叶志伟,郑肇葆.蚁群算法中参数 α 、 β 、 r 设置的研究——以TSP问题为例[J].武汉大学学报(信息科学版),2004,29(7):597-601.