

分销供应链协商调度多目标合作协同演化算法*

苏生¹, 于海杰², 吴正华¹, 姚远哲¹, 张良³

¹(电子科技大学 计算机科学与工程学院, 四川 成都 611731)

²(电子科技大学 经济与管理学院, 四川 成都 611731)

³(泸州老窖集团有限公司, 四川 泸州 646000)

通讯作者: 苏生, E-mail: susheng@uestc.edu.cn, http://www.ccse.uestc.edu.cn

摘要: 研究了在制造商占优并优先调度的分销供应链中, 多个分销商同时与制造商进行协商以改善自身调度的问题, 建立了基于补偿的多目标协商调度模型, 提出了同时实施分销商局部演化计算与制造商全局演化计算的新型多目标合作协同演化算法 GLCCEC. 提出了制造商全局精英解的跳跃渐变解组合策略及全局非支配解集实时更新策略, 设计了保持局部作业顺序约束下的分销商局部解全局化动态规划算法. 实验结果表明, GLCCEC 算法能够在不损害制造商调度的条件下有效改善每个分销商的调度, 所获得的非支配解集不仅目标值优于现有的 3 种主要合作协同演化算法 MOCCGA, NSCCGA, GBCCGA, 而且具有良好的解分散度.

关键词: 合作协同演化算法; 多目标; 分销供应链; 调度; 协商

中图法分类号: TP301 **文献标识码:** A

中文引用格式: 苏生, 于海杰, 吴正华, 姚远哲, 张良. 分销供应链协商调度多目标合作协同演化算法. 软件学报, 2013, 24(6): 1165-1176. <http://www.jos.org.cn/1000-9825/4288.htm>

英文引用格式: Su S, Yu HJ, Wu ZH, Yao YZ, Zhang L. Multi-Objective cooperative co-evolutionary algorithm for negotiated scheduling of distribution supply chain. Ruan Jian Xue Bao/Journal of Software, 2013, 24(6): 1165-1176 (in Chinese). <http://www.jos.org.cn/1000-9825/4288.htm>

Multi-Objective Cooperative Co-Evolutionary Algorithm for Negotiated Scheduling of Distribution Supply Chain

SU Sheng¹, YU Hai-Jie², WU Zheng-Hua¹, YAO Yuan-Zhe¹, ZHANG Liang³

¹(School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)

²(School of Economic and Management, University of Electronic Science and Technology of China, Chengdu 611731, China)

³(LuZhouLaoJiao Co., LTD, Luzhou 646000, China)

Corresponding author: SU Sheng, E-mail: susheng@uestc.edu.cn, <http://www.ccse.uestc.edu.cn>

Abstract: It is investigated that multiple distributors simultaneously negotiate with a manufacturer to improve themselves schedules on a distribution supply chain in which manufacturer has stronger power than distributors and does scheduling decision prior to distributors. A compensation based negotiation scheduling model is built. A novel multi-objective cooperative co-evolutionary algorithm (GLCCEC) that concurrently implements local evolutionary computing of distributors and global evolutionary computing of manufacturer is proposed. Global elite solution combination strategy with gradually gene skipping change and real time updating of global non-dominated solution set are designed for manufacturer. A dynamic programming algorithm with constraint of retaining sequence of local schedule is designed in order to get global solution from a local solution of distributor. Computational experiments show that GLCCEC algorithm can effectively improve schedule of each distributor with no deterioration of manufacturer's schedule. Moreover, the non-dominated solutions

* 基金项目: 国家自然科学基金(60904072); 国家教育部博士点新教师基金(20090185120002); 国家教育部人文社科青年基金(09YJC630018); 电子科技大学中央高校基本科研业务经费(103.1.2 E022050205)

收稿时间: 2012-04-28; 定稿时间: 2012-07-23

of GLCCEC not only are better than that of other best cooperative co-evolutionary algorithms: MOCCGA, NSCCGA, GBCCGA, but also has good spread in solution space.

Key words: cooperative co-evolutionary algorithm; multi-objective; distribution supply chain; scheduling; negotiation

分销商是为制造商销售产品的实体,制造商与一个或多个分销商形成分销供应链系统.制造商与分销商属于不同企业实体,它们之间相互独立,但在产品的生产与销售上紧密协作,然后分享从下游用户中获得的收益.作业调度是制造商与分销商均需要处理的日常决策,需要在两类实体之间良好配合.

在以制造商为中心的条件下,分销商在供应链上的影响力比制造商弱,分销商通常基于制造商的调度结果做出自己的调度决策.这种调度协作模式对制造商有利,但对分销商不利,同时也会损害供应链的整体运作性能.为此,一些研究从协调和联合角度研究了供应链上下游企业之间(包括制造商与分销商之间、供应商与制造商之间等)的协调调度.

Chen 等人考虑了一个组装供应链中供应商与制造商之间的协商调度问题^[1],分别考虑了供应商主导、制造商协商以及制造商主导、供应商协商两种协商环境,提出了基于补偿的协商机制和动态规划算法.研究结果显示,协商调度能够削减高达 82.37% 的调度成本.文献[2]也考虑了类似的补偿协商机制.Dawande 等人考虑了报纸生产厂家与其分销商之间的协调调度问题,以及两类相似生产厂家与其分销商之间协调调度问题^[3].作者在分析独立调度缺点的基础上研究了制造商与分销商之间调度的衔接机制,设计了相应的动态规划算法.实验结果表明,协调调度能够改善分销商百分之零点几至百分之几的调度成本,改善制造商平均 35% 左右的调度成本.

更多研究从集成角度研究制造商与分销商调度协作获得的好处,这包括生产与运输集成调度^[4-11]、生产与分销集成调度^[12-17].但斌等人研究了基于交货期窗口约束的多级供应链批量调度问题,并将批量调度和排序调度作为两个子问题处理^[18];姚建明等人用蚁群算法研究了大规模定制环境下的供应链动态调度问题^[19].

虽然目前存在较多供应链调度的研究,但针对具有多个分销商的供应链,基于协商的供应链调度研究或者让每个分销商独立与制造商进行协商,导致协商效果欠佳;或者将多个分销商的协商收益作为一个整体与制造商协商,然后在分销商之间分享协商收益.然而在现实供应链中,多个分销商之间相互独立,每个分销商参与协商的目的是最大化自身的利益,且相互之间没有达到完全信任,不会将作业处理时间和成本等关键信息公布给其他分销商以及制造商,将多个分销商的协商收益作为一个整体与制造商协商更多是一种理想状态,而在协商过程中同时最大化每个分销商的收益更具有可操作性.为此,本文以同时最大化所有分销商协商收益为目标,研究分销商与制造商之间的多目标协商调度问题.此外,分销商与制造商之间也是非完全信任,不共享彼此的作业处理时间.

本文第 1 节描述问题.第 2 节建立多目标协商调度模型.第 3 节提出多目标合作协同演化算法.第 4 节为计算实验与分析.最后是结论.

1 问题描述

考虑由一个制造商和 m 个分销商组成的分销供应链,分销商 $i(i=1, \dots, m)$ 需要对 n_i 个作业(来源于订单)进行调度安排,以完成产品的处理与销售.任意作业相关产品首先由制造商生产,然后通过分销商销售.分销商 i 的第 $j(j=1, \dots, n_i)$ 个作业 $J_{i,j}$ 在制造商和分销商所占用的处理时间分别为 $p_{i,j,u}$ 和 $p_{i,j,d}$,其中, $p_{i,j,u}$ 为广义处理时间,包括作业 j 在制造商内部的加工时间以及将作业相关的产品运输到分销商所耗费的时间.作业 $J_{i,j}$ 的交货期为 $d_{i,j}$.

虽然制造商和分销商均是供应链上的成员,但他们相互独立,企业通常不愿意将内部的信息全部暴露给供应链上的其他伙伴成员.我们考虑制造商和分销商之间以及分销商之间均不共享作业处理时间信息的情况,即针对作业 $J_{i,j}$,制造商不知道分销商 $i(i=1, \dots, m)$ 的处理时间 $p_{i,j,d}$,分销商也不知道制造商的处理时间 $p_{i,j,u}$,但分销供应链上,企业之间允许在调度过程中进行某种合作.

假定制造商处于主导地位,分销商彼此独立,各分销商之间地位平等,当制造商为大品牌厂商时经常出现这

种情况.制造商首先进行自己的调度安排,然后分销商再根据制造商作业的完成情况(主要依据作业在制造商的完成时间)进行调度安排.制造商在这种调度模式中占优,其调度结果对制造商有利,但对分销商不利,同时对整个供应链调度的性能也不利.为此,本文在不违背制造商和分销商之间信息私有性的条件下,研究多个分销商如何与制造商协商以改善分销商调度及整个供应链调度,同时提供分销商协商调度算法.

假定所有作业均在零时刻释放(如在某个供应链计划周期的开始时刻),制造商以最小化作业的总完成时间为目标,分销商以最小化作业总权重延迟为目标,根据调度三要素 $a/b/c$ 法则(a, b 和 c 分别表示机器环境、作业特点和调度优化目标),制造商的调度问题可表示为 $1||\sum_{i=1}^m \sum_{j=1}^{n_i} C_{i,j}$, 分销商的调度问题可表示为 $1||r_{i,j} | \sum w_{i,j} T_{i,j}$, $1 \leq j \leq n_i$, 其中,作业 $J_{i,j}$ 的释放时间 $r_{i,j}$ 依赖于制造商的作业完成时间, $w_{i,j}$ 是作业 $J_{i,j}$ 的单位延迟惩罚权重.用 $s_{i,j,u}$ 和 $e_{i,j,u}$ 分别表示作业 $J_{i,j}$ 在制造商调度中的开始时间和结束时间, $s_{i,j,d}$ 和 $e_{i,j,d}$ 分别表示作业 $J_{i,j}$ 在分销商调度中的开始时间和结束时间, α 和 β_i 分别表示制造商的调度和分销商 i 的调度, α_i 表示分销商 i 的作业在制造商的调度, α_i 和 β_i 之间的衔接应合法,即 $s_{i,j,d} \geq e_{i,j,u}$.用 $O(\alpha_i)$ 和 $O(\beta_i)$ 分别表示调度 α_i 和 β_i 的目标值, μ_i 表示单位 $O(\alpha_i)$ 的成本, ν_i 表示单位 $O(\beta_i)$ 的成本.

2 多目标调度协商模型

订单作业先后经过制造商与分销商两个阶段进行处理,制造商对分销商的影响来源于制造商调度中作业的完成时间,主要体现为衔接约束 $s_{i,j,d} \geq e_{i,j,u}$, 即作业 $J_{i,j}$ 在分销商 i 调度中的开始时间必须等于或晚于其在制造商调度中的完成时间, $e_{i,j,u}$ 相当于作业 $J_{i,j}$ 在分销商调度问题中的释放时间 $r_{i,j}$. 如果分销商 i 要改善自己的调度, 则必须改变 $r_{i,j}$, 这意味着分销商 i 需要与制造商协商, 将作业 $J_{i,j}$ 在制造商调度中的完成时间 $e_{i,j,u}$ 变为一个对自己有利的新值. 而 $e_{i,j,u}$ 的改变意味着制造商调度的改变, 即制造商要采用一个比以前调度方案差的新调度. 为激励制造商采用新的调度方案, 分销商提供一个补偿 σ 给制造商, 制造商新调度的目标值在补偿后不会比原调度差. 不失一般性, 假定 σ 等于制造商新调度目标值与制造商原调度目标值之差(见公式(1)). 相应地, 分销商在新调度目标值的基础上进行制造商补偿操作后, 其结果应该比分销商原调度的目标值好, 否则, 分销商没有必要进行协商, 见公式(2).

$$\sigma = \sum_{k=1}^m u_k (O(\alpha'_k) - O(\alpha_k)) \tag{1}$$

$$\nu_i O(\beta'_i) + \sigma_i > \nu_i O(\beta_i) \tag{2}$$

分销商 i 给制造商的补偿 σ_i 应大于等于制造商采用新调度带来的利益损失, 分销商 i 会选择扣除 σ_i 后对分销商 i 最有利的调度. 由于存在 m 个地位平等的分销商, 每个分销商均与制造商协商, 导致制造商会面对 m 个新调度, 且这 m 个新调度经常会发生冲突. 为避免在分销商中造成不公平, 导致客户流失, 制造商的策略是在获得应有补偿后不偏向任何一个分销商. 如果一个新调度使某些分销商调度变优, 但却让另外一些分销商的调度变差, 则制造商不选择此新调度, 只有在每个分销商调度均不变差的条件下才被制造商采用. 所以, 此分销供应链面对的是一个多目标协商调度问题, 协商的目的是让制造商采用新调度, 每个分销商均以最大化调度改善为协商目标.

由于所有分销商均参与协商, 在最终选择的新调度(如果存在)中, 一个分销商或者从协商中获益, 或者保持收益不变. 如果让某一个分销商负担所有的补偿 σ , 则对这个分销商不公平. 为此, 需要在分销商中分摊补偿 σ , 以下是分摊机制:

分销商 i 提供给制造商的补偿 σ_i 与其从新调度中获得的收益成正比, 即收益越大, 补偿越多, 所有分销商补偿之和等于 σ . 用 δ_i 表示分销商 i 原调度成本与新调度成本之差, β'_i 表示在分销商协商调度模式下分销商 i 的新调度.

δ_i 可表示为公式(3), σ_i 用公式(4)表示:

$$\delta_i = \nu_i \max(0, (O(\beta_i) - O(\beta'_i))) \tag{3}$$

$$\sigma_i = \frac{\delta_i}{\sum_{i=1}^m \delta_i} \sigma \quad (4)$$

分销供应链多目标协商调度模型为

$$\begin{aligned} & \text{Min } v_1 O(\beta'_1) + \sigma_1 \\ & \vdots \\ & \text{Min } v_m O(\beta'_m) + \sigma_m \\ \text{s.t. } & \sigma = \sum_{k=1}^m u_k (O(\alpha'_k) - O(\alpha_k)), v_i O(\beta'_i) + \sigma_i > v_i O(\beta_i), \sigma_i = \frac{\delta_i}{\sum_{i=1}^m \delta_i} \sigma. \end{aligned}$$

3 多目标合作协同演化调度算法

分销供应链存在多个分销商,由于信息私有性,不能将所有分销商结合成一个整体与制造商协商,每个分销商均需独立与制造商进行协商.即,每个分销商是一个独立的协商主体,但在协商过程中可以获得制造商和其他分销商的配合.分销商提出的协商内容体现为局部新调度,它相当于协商提议.处理协商提议的传统方法主要分为提议评估策略与提议生成策略,其中:提议评估策略用来对收到的提议进行评估,判断是否接受对方给出的提议;提议生成策略根据收到的提议生成反提议.根据第2节的协商模型,分销商和制造商之间提议的评估需要分销商和制造商协作才能完成,提议评估策略与提议生成策略均不适合,且整个分销供应链协商模型是为一个具有信息私有性约束的多目标优化问题.为此,本文提出了求解分销供应链协商调度的合作协同演化算法,在此算法中,所有分销商同时启动与制造商的协商进程,在求解协商调度解的过程中,所有分销商的协商进程在保持信息私有性的同时相互合作,以求在短时间内获得好的分销供应链协商调度解.

3.1 算法框架

在提出的合作协同演化协商算法中,每个分销商维护一个演化种群,实施演化计算.种群中的解为此分销商的局部调度,但每个解的目标应包含此解在每个分销商上的目标函数值,以评估此解在协商过程中的优劣.分销商将此调度作为协商内容发送给制造商.为评估此局部新调度,制造商需利用一定规则形成涉及所有分销商所有作业的全局新调度,并根据协商模型进行多目标评估.根据评估结果,如果此解满足协商模型中的约束条件,则认为此局部新调度可行,且将多目标值返回给分销商;否则,认为此局部新调度不可行,多目标值设为无限大.

在协商过程中,制造商拥有一个非支配解集,解集中的每个解是一个分销供应链全局调度解,它具有所有分销商所有作业在制造商的排列顺序、开始处理时间和结束处理时间,同时有每个分销商上所有作业的排列顺序,但不包含这些作业在各自分销商的开始处理时间和结束处理时间,其目标值包含补偿后每个分销商的调度成本以及制造商调度成本.制造商的非支配解集随着演化过程不断更新.为充分利用制造商所拥有的全局非支配解,制造商也实施基于非支配解集的演化计算,不断更新非支配解集,也同时影响分销商的局部演化过程.我们将这种全局解与局部解均实施演化计算的算法称为 GLCCEA(global local cooperative co-evolutionary algorithm)算法,以区别于现有合作协同演化算法.图1给出了合作协同演化算法框架示意图.

协商初始时,每个分销商随机产生自身的局部调度作业顺序传送给制造商,制造商结合分销商的作业顺序形成一个全局调度作业顺序,并将此解作为非支配解,将每个分销商的调度成本及制造商调度成本设为无穷大.

3.2 分销商局部演化算法

每个分销商维护一个由局部调度解构成的种群,并在这个种群上执行演化算法.一个局部调度是此分销商所有作业形成的一个排列.虽然一个分销商的局部调度只涉及此分销商的作业,但它能够从制造商处获得基于此局部调度的全局调度多目标值,对分销商局部调度的评价也是基于多目标值,而不是针对此分销商的单目标,采用求解多目标优化问题的演化算法求解分销商局部调度问题.

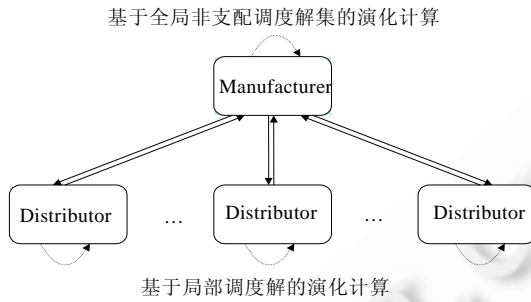


Fig.1 Cooperative co-evolutionary algorithm framework between manufacturer and distributors

图 1 制造商和分销商之间的合作协同演化算法框架

分销商演化计算的基本操作为普通遗传算法中的选择、交叉与变异.一个解的适应度为由当前种群及父种群组成的组合种群中支配此解的其他解个数,适应度越大,表明支配此解的其他解个数越多.如果一个解的适应度等于 0,则表明此解是组合种群中的非支配解.

采用 Fonseca,Fleming 提出的适应度共享策略保持种群多样性^[20],在解空间,根据个体的相似度在种群中形成若干小生境.个体 i 与 j 之间的相似度 d_{ij} 表示为这个解的多目标值在 m 维空间上的 Euclidean 距离.

$$d_{ij} = \sqrt{\sum_{k=1}^m (o_{ki} - o_{kj})^2} \tag{5}$$

其中, o_{ki} 和 o_{kj} 分别为解 i 和 j 的第 $k(k=1, \dots, m)$ 个目标值, m 为分销商的个数,也为目标的个数.根据 d_{ij} ,个体 i 与 j 之间的共享函数 $sh(d_{ij})$ 定义为

$$sh(d_{ij}) = \begin{cases} 1 - (d_{ij} / \sigma_{share})^\alpha, & d_{ij} < \sigma_{share} \\ 0, & \text{else} \end{cases} \tag{6}$$

α 是控制共享函数性质的参数,通常取 $\alpha=1$. σ_{share} 为控制允许共享度的参数,通过以下多项式方程求解获得取值.

$$N\sigma_{share}^{m-1} - \frac{\prod_{i=1}^m (M_i - m_i + \sigma_{share}) - \prod_{i=1}^m (M_i - m_i)}{\sigma_{share}} = 0 \tag{7}$$

其中, N 为种群规模.根据 $sh(d_{ij})$,修正个体 i 的适应度 f_i 为 f_{mi} .

$$f_{mi} = \frac{f_i}{\sum_{j=1}^N sh(d_{ij})} \tag{8}$$

3.3 制造商全局演化算法

制造商维护一个全局非支配解集,每个全局非支配解包含所有作业在制造商的调度信息,分销商的作业处理顺序与其在制造商的一样,但制造商没有任何作业在其所属分销商的开始处理时间和结束处理时间信息,其原因是制造商与分销商之间并不共享作业的处理时间信息,而作业顺序在分销商的开始处理时间和结束处理时间依赖于作业的处理时间信息.由于不知道作业在分销商的处理时间,故需要分销商计算出作业顺序对应的目标值后传递给制造商,然后由制造商根据公式(4)计算每个分销商的补偿,并将补偿后的目标值作为相应分销商的目标值.由于全局非支配解是分销商演化过程中的精英解,故制造商在全局非支配解集上实施的演化计算主要应用基于精英解的勘探策略,包括跳变解组合策略与邻域搜索策略,实施操作的对象是制造商的作业排列.

3.3.1 跳跃渐变解组合

用链表 L 存储非支配解集,三元组 (i,j,d) 为距离标识,表示 L 中解 i 与 j 之间的距离 d_{ij} 及解在 L 中的序号,链表 P 根据 d_{ij} 从大到小的顺序存储 L 中所有解之间的距离标识,为全局非支配解集中的任意解 $k(k=1, \dots, N)$ 建立一个链表 Q_k ,将此解与其他元素的距离标识存储在链表 Q_k 中,链表中元素的位置根据 d_{ij} 从小到大的顺序进行

排列.

算法 1. SkipCombine.

输入:非支配解集 L ;

输出:更新后的非支配解集 L' .

- (1) 令 $P'=P, R=\emptyset$;
- (2) 选择 P' 的第 1 个元素 $\langle s, r, d \rangle$;
- (3) 生成新解集合
 - (3.1) 令位置 $z=0$, 并用 $g_z(s)$ 表示解 s 在位置 z 上的基因值;
 - (3.2) 从位置 z 开始扫描解 s 与 r 的染色体直到位置 $x, g_x(s) \neq g_x(d)$;
 - (3.3) 继续扫描解 s 与 r 中位置 x 后的基因直到位置 $y+1, g_{y+1}(s) = g_{y+1}(d)$;
 - (3.4) 将 s 中从 x 到 y 的基因片段 $s_{x \rightarrow y}$ 替换为 r 中对应基因片段 $r_{x \rightarrow y}$;
 - (3.5) 将 $s_{x \rightarrow y}$ 中的每个基因调整到 $r_{x \rightarrow y}$ 对应基因在 s 中的原始位置;
 - (3.6) 记更改基因后的解为 s' , 计算其多目标值;
 - (3.7) 根据支配情况用 s' 更新解集 R ;
 - (3.8) If $y < \sum_{i=1}^m n_i$ Then $z=y$, 转步骤(3.2);
- (4) 从 P' 中删除含有解 $s, r, l (l \in L, d_{ls} < \delta$ 或 $d_{lr} < \delta)$ 的距离标识, δ 为解距离半径阈值;
- (5) If $|L| > 1$ Then 返回步骤(2);
- (6) 根据支配情况用集合 R 更新解集 L .

算法 1 的基本思想是,从全局非支配解集中选择相似度最小的两个个体进行跳变解组合,然后排除那些与被组合个体距离半径小于 δ 的个体,并重复个体选择、跳变解组合与排除操作,直到全局非支配解集中剩余的个体数小于 2,然后用跳变解组合形成的新解集合更新原有的全局非支配解集.跳变解组合操作的目的是最大程度利用非支配解代表的模式,并通过渐进组合形成若干新解,以期在全局非支配解空间附近发现更好解.图 2 给出跳跃渐变解组合操作的示意,其中, $d_{ij} > d_{sr} > d_{uv}$. 从解 i 与 j 、 s 与 r 以及 u 与 v 之间的路径上,通过跳跃渐变解组合形成多个新解,这些解同时拥有路径两端点解的部分基因.

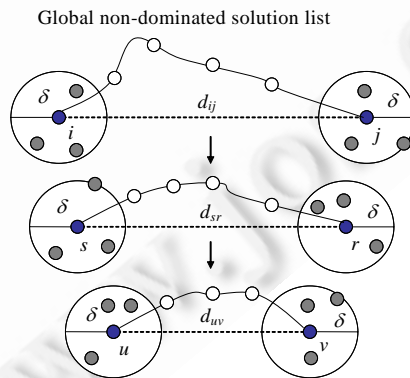


Fig.2 Jumping driven combination of solutions

图 2 跳跃渐变解组合示意

3.3.2 制造商作业交换邻域搜索

跳跃渐变解组合操作主要是利用从一个全局非支配解到另一个全局非支配解进行跳跃渐变形成一个新解集合,并用这个新解集合更新当前全局非支配解集.此外,为探索单个全局非支配解附近的解空间,利用交换邻域搜索来获得更好解.根据全局非支配解的染色体格式,选择在制造商作业排列上进行 ζ 步随机作业交换,一次作业交换形成一个新解,并计算此新解的多目标值.在完成 ζ 步随机作业交换完成后,用形成的新解集合更新现有

的全局非支配解集.

为充分探索全局非支配解邻域空间,对非支配解集中的每个解实施交换邻域搜索.需要注意的是,虽然每次作业交换仅改变两个作业的位置,但位于交换作业之后的那些作业在制造商的完成时间可能会改变,导致其在相应分销商的释放时间也会改变,故需要将改变了作业顺序和完成时间的分销商局部作业排列发送到相应分销商,分销商计算后再将其局部目标值返回给制造商,制造商再根据补偿模型计算新解的多目标值.

3.3.3 全局非支配解集实时更新策略

制造商并不独立产生解,所维护的非支配解集中每个解均来自于分销商的协作需求.由于一个分销商在协商过程中需要不断产生新的局部调度与制造商协商,并验证协商结果,但在产生自己的局部调度后,分销商无法独立确定此调度中作业的释放时间,进而无法确定作业的开始时间和结束时间,也无法知道这个局部调度的协商效果.为此,分销商需要将此作业排列传递给制造商,制造商利用特定算法(后文中的 SCS 算法)形成一个全局解,并获得这个全局解的多目标值,这表明每收到一个分销商局部作业排列,就获得一个多目标解.为此,我们采用非支配解集及时更新策略,即每评价一个解就更新一次全局非支配解集.演化计算开始时非支配解集为空,当向此解集插入一个解 s 时,遵循算法 InsertSol.

算法 2. InsertSol.

输入:解 s ;

输出:更新后的 L, P, Q_k .

$dom=false$

For \forall 解 $w(w \in L)$ Do

 If 解 s 支配解 w Then

 遍历 P 和 $Q_k(k=1, \dots, M)$, 从 P 和 Q_k 中删除解 w 的序号;

 For $\forall v(v \in L, I_v > I_w)$, 更改 v 在 P 和 Q_k 中的序号 End For

 End If

 If 解 w 支配解 s Then

$dom=true$;

 break;

 End If

End For

If $dom=false$ Then

 计算 $d_{s,k}(k \in L, I_k \neq I_s)$;

 根据 $d_{s,k}$, 将 I_s 插入到 P 和 Q_k 中;

End If

3.4 合作机制

3.4.1 分销商局部调度的全局化策略

在制造商收到分销商 $i(i=1, \dots, m)$ 的局部作业调度后,制造商为每个分销商 $j(j \neq i, j=1, \dots, m)$ 从当前全局非支配解集中随机选择一个非支配解,并将此非支配解中分销商 j 的作业顺序抽出来与分销商 i 的局部作业调度组合,最后形成一个包含所有分销商作业的完整调度.由于是分销商给出的顺序,故制造商调度时需要保持分销商的局部作业顺序.从非支配解中获得的其他分销商局部调度具有代表性,故也应该在组合时保留.为此,需要找到一个算法在不改变所有分销商局部作业顺序的条件下组合所有分销商局部作业顺序,同时满足制造商最小化作业总完成时间的目标.下面给出制造商合并两个作业顺序的动态规划算法 SCS,调用 $m-1$ 次 SCS 算法即可获得保持局部作业顺序约束下的所有分销商作业顺序的组合调度.

要求:给定两个需合并的队列 $R=\{r_1, \dots, r_n\}$ 和 $S=\{s_1, \dots, s_n\}$, R 和 S 有一定顺序,且其中的每个元素为一个作业,有一定的处理时间,要求在保持 R 和 S 原有顺序的条件下将 R 和 S 合并,并使合并后的队列处理时间之和最小.

算法 3. SCS.

在队列 S 的 s_i 元素之前添加一个虚拟元素 s_0 , 此元素不参与调度. 令队列 S 中 s_i 与 s_{i+1} ($i=0, \dots, n$) 之间的位置为 p_i ($i=0, \dots, n$), 这些位置形成位置队列 $P=\{p_0, \dots, p_n\}$. 将 R 合并到 S , 即是用 R 中的元素填充到位置队列 P 中.

令将 R 中元素 $\{r_i, \dots, r_n\}$ 填充到队列 $\{p_j, \dots, p_n\}$ 中与 S 形成的合并队列成本为 $C_{i,j}$, 则问题完整解的成本为 $C_{1,0}$.

状态: $C_{i,j}$.

状态转移函数

$$C_{i,j} = \min(d_{i,j} + C_{i+1,j}, C_{i,j+1}), 1 \leq i \leq m-1 \quad (9)$$

根据状态转移函数, R 中的每个元素均有可能放在任何一个位置, 且一个位置可能放多个元素, 但要保持原有顺序, 考虑第 j 个位置被占用和没有被占用两种情况:

- Case 1: 第 j 个位置被占用. 如果第 i 个元素放在第 j 个位置, R 队列被削减了一个元素, 而第 $i+1 \sim m$ 个元素仍然可能放在第 j 个位置, 这包含在子问题 $C_{i+1,j}$ 中;
- Case 2: 第 j 个位置没有被占用. 即 R 中没有任何元素位于第 j 个位置, 整个 R 队列都从第 $j+1$ 个位置开始, 这等价于子问题 $C_{i,j+1}$.

边界条件:

if $i > m$, then $C_{i,j} = 0$

if $j > n$, then $C_{i,j} = \infty$

$C_{i,m} = d_{i,n}$

$C_{i,n} = d_{i,n}$ 蕴含 $C_{k,n} = d_{k,n}$ ($i < k \leq m$)

在 SCS 算法中, 共有 $m(n+1)$ 个状态, 每个状态转移的状态数为 2, 每次状态转移的时间包含计算 $d_{i,j}$ 的值和两个 $d_{i,j}$ 值的比较, $d_{i,j}$ 值的计算时间来源于把 S 中位于 j 位置后的作业时间与 R 的第 i 个作业时间相加所耗费的时间, 由于 $0 \leq j \leq n$, 故任何一次相加的作业个数不会超过 $n+1$ 个; 由于 S 中的每个元素只会被加一次, 故平均相加作业个数不超过 2. 故一次状态转移的平均时间为 $O(2)$, 总的时间复杂度上界为 $O(m(n+1))$.

3.4.2 计算分销商局部调度的多目标值

分销商提出的局部新调度需要制造商的协作才能获得全局多目标值, 且这种协作必须遵循分销商与制造商之间的信息私有性. 以两个分销商为例, 制造商在收到分销商 1 传来的局部作业顺序 A 后, 从当前非支配解集中随机选择一个非支配解 R , 利用 SCS 算法将顺序 A 与解 R 中分销商 2 的作业顺序 B 组合, 形成一个完整解 C . 由于制造商不知道分销商的作业处理时间, 故制造商再将 C 中具有开始时间和完成时间的分销商作业顺序 A 和 B 分别发回给分销商 1 和分销商 2, 它们在计算出各自作业顺序的目标值后发回给制造商, 制造商组合完成后再将多目标值给分销商 1, 同时利用此完整解更新自己的非支配解集.

4 仿真实验

为验证多目标协商调度的有效性, 进行两方面实验: 一方面是利用本文设计的 GLCCEA 算法进行多分销商调度, 然后比较协商调度与独立调度的优劣; 另一方面是将 GLCCEA 与现有的合作协同演化算法进行求解比较, 验证 GLCCEA 算法的优劣.

虽然理论上分销商的数据可以无限多, 但超过 3 个分销商就不好图形化表示算法最终获得的非支配解集情况, 且求解时间将大幅增加. 为此, 我们仅考虑具有两个和 3 个分销商的情况. 随机生成若干算例进行实验, 针对两个分销商, 则考虑每个分销商的作业数分别为 10, 20, 40, 60, 80, 100, 150, 200; 针对 3 个分销商, 则考虑每个分销商的作业数分别为 10, 20, 40, 60, 80, 100. 不同的分销商数量和作业数代表了算例规模从小到大的变化. 制造商和分销商作业处理时间是 1~100 的随机数, 一个作业的交货期是此分销商上所有作业的处理时间之和与一个随机数 δ 的乘积, δ 的取值范围为 1.5~2.5. 单位延迟惩罚时间为 1~3 之间的整数.

算法用 Java 语言编写, 分销商演化种群规模为 100. 针对每个算例, 算法重复运行 10 遍, 然后将获得的 10 个非支配解集进行合并, 获得一个最终的非支配解集作为此算法求解此算例的结果解集. 制造商和分销商单位成

本 μ_i 和 v_i 均设为0.5.

图3给出了分销商利用GLCCGA算法获得的协商调度改善度 $\eta_i(i=1, \dots, m)$, η_i 的值由公式(10)求得.图3(a)和图3(b)分别是分销商数目为2和3的条件下各个分销商的调度改善度(d1 win%表示第1个分销商的改善度、d2 win%表示第2个分销商的改善度、d3 win%表示第3个分销商的改善度)以及所有分销商的平均改善度(sum win%).横坐标用 x_y 表示,其中, x 表示作业数量, y 表示分销商数量.纵坐标表示 η_i 的值.

$$\eta_i = \frac{O(\beta_i) - O(\beta'_i)}{O(\beta_i)} \times 100 \tag{10}$$

如图3所示,GLCCGA算法在不损害制造商调度的条件下能够有效改善每个分销商的调度:对于两个分销商的情况,最小改善度为32.11%,最大改善度为82.07%;对于3个分销商的情况,最小改善度为5.93%,最大改善度为76.01%,有力地表明了GLCCGA算法的分销商协商调度能力.

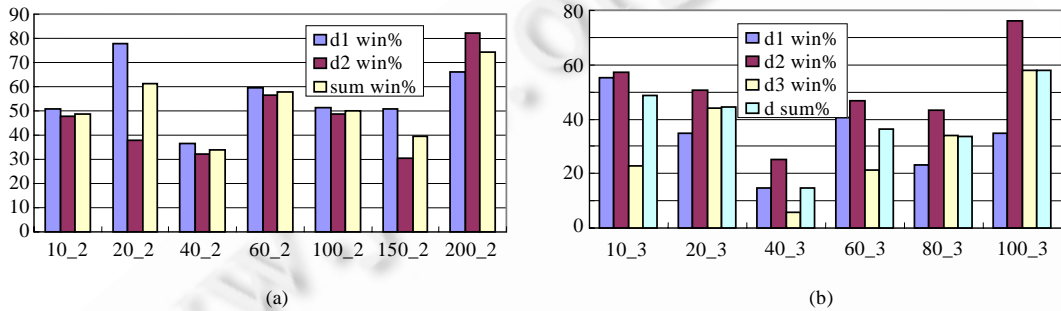


Fig.3 Gains of negotiated scheduling of GLCCGA on different instances

图3 GLCCGA在不同算例上的协商调度收益

为进一步验证GLCCGA的多目标优化能力,将GLCCGA在算例上获得非支配解集与现有的3种主要合作协同演化算法MOCCGA^[21],NSCCGA^[22],GBCCGA^[23]进行比较.公平起见,4种算法在每个算例上均运行相同时间,然后比较在每个算例上4种算法获得最终非支配解集.

通常,比较多目标算法的性能指标包含两种:一种是获得的非支配解在目标值上的大小,另一种是非支配解在解空间上的分散度情况.为此,图4和图5分别给出了4种算法在两个和3个分销商条件下获得最终非支配解集.

图4的图4(a)~图4(f)分别给出了作业数等于10,20,40,60,80,100时,4种算法获得的非支配解集.纵横坐标分别代表一个分销商的局部调度目标值.从图4可以看出,在每种算例上,GLCCGA算法获得的非支配解均明显优于其他3种算法获得的非支配解,GLCCGA算法的非支配解集能够完全支配其他3种算法的非支配解集.此外,GLCCGA算法获得的非支配解个数也多于其他算法,并具有很好的分散度,非支配解在图中表现为伸展的弧形.

图5(a)~图5(d)分别给出了作业数=100时,4种算法获得的非支配解集.一个坐标轴代表一个分销商的局部调度目标值.从图5可以看出,在每种算例上,GLCCGA算法获得的非支配解不仅优于其他3种算法获得的非支配解(注意,与GLCCGA相关子图的坐标轴范围与其他算法的不同),且其数量和分散度也较好.

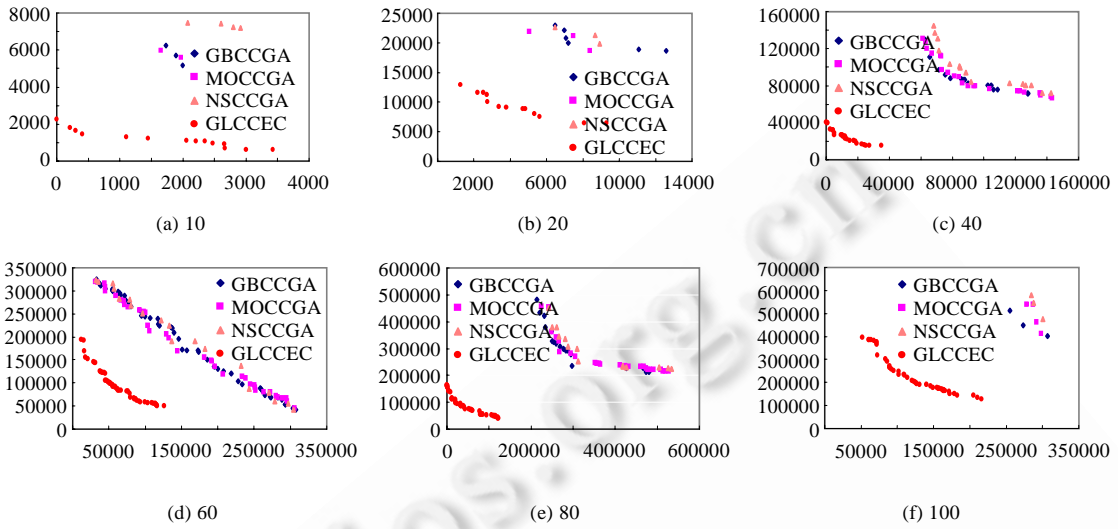


Fig.4 Nondominated sets of four algorithms for supply chain with two distributors

图 4 4 种算法在 2 个分销商时获得的非支配解集

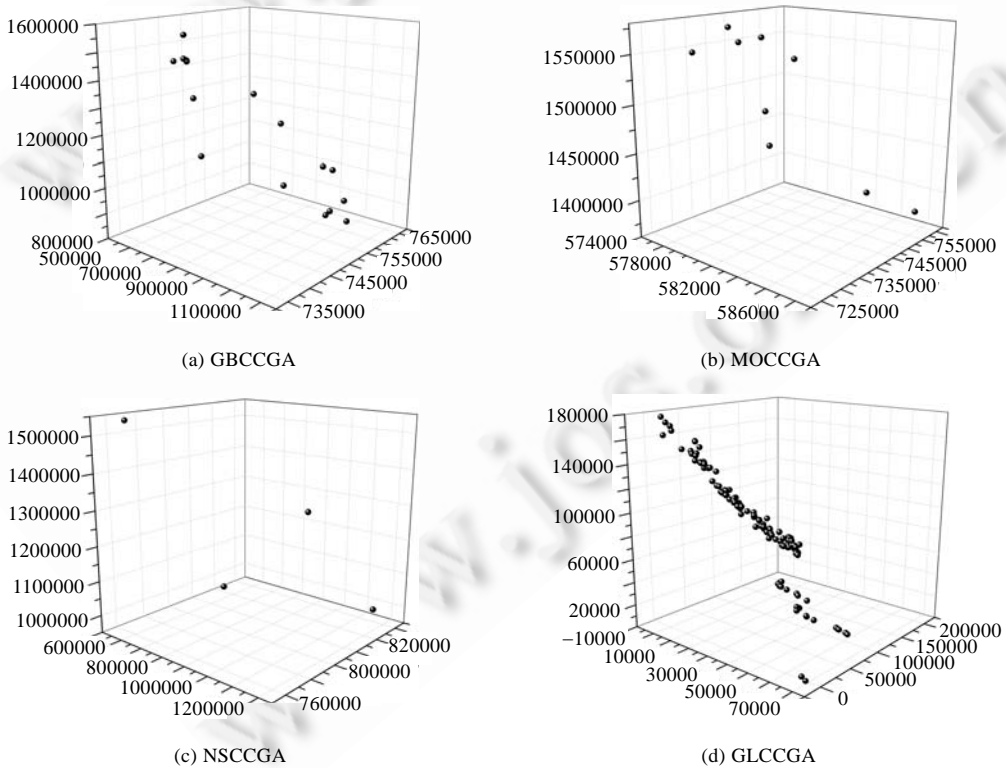


Fig.5 Nondominated sets of four algorithms for supply chain with three distributors

图 5 4 种算法在 3 个分销商时获得的非支配解集

5 结 论

本文研究了在一个由多个分销商与制造商组成的分销供应链上分销商的协商调度问题,所提出的多目标合作协同演化算法 GLCCEC 有效解决了在分销商与制造商之间、分销商之间不分享作业处理时间的非完全信任条件下进行协商调度的问题,能够在不恶化制造商的调度的条件下大幅改善每个分销商的调度.GLCCEC 算法一方面体现了分销商与制造商之间的协作,另一方面体现分销商与制造商作为企业实体的自主性和自私性,这符合供应链运作特征,对分销供应链调度具有实际应用价值。

根据分销供应链协商调度问题特征所设计的制造商精英解跳跃渐变组合以及非支配解集实时更新策略,是不同于其他合作协同演化算法的明显特征,对 GLCCEC 的多目标优化能力有很好的促进作用.GLCCEC 算法的思想也可以用于解决其他多目标优化问题,在未来的研究中我们将进一步验证。

由于非完全信任的问题,每个分销商的每个局部调度需要通过 SCS 算法在保持局部调度作业顺序的条件下形成全局调度,以获得局部调度的多目标值.而 SCS 算法比较耗时,这导致 GLCCEC 算法也相对耗时.为此,如何提高 GLCCEC 的计算效率,也是值得研究的地方。

References:

- [1] Chen ZL, Hall NG. Supply chain scheduling: Conflict and cooperation in assembly systems. *Operations Research*, 2007,55: 1072–1089. [doi: 10.1287/opre.1070.0412]
- [2] Dawande M, Geismar HN, Hall NG, Sriskandarajah C. Supply chain scheduling: distribution systems. *Production and Operations Management*, 2006,15:243–261. [doi: 10.1111/j.1937-5956.2006.tb00243.x]
- [3] Manoj UV, Gupta JND, Gupta SK, Sriskandarajah C. Supply chain scheduling: Just-in-Time environment. *Annals of Operations Research*, 2008,161:53–86. [doi: 10.1007/s10479-007-0290-1]
- [4] Hall NG, Potts CN. The coordination of scheduling and batch deliveries. *Annals of Operations Research*, 2005,135:41–64. [doi: 10.1007/s10479-005-6234-8]
- [5] Chen ZL, Vairaktarakis GL. Integrated scheduling of production and distribution. *Management Science*, 2005,51:614–628. [doi: 10.1287/mnsc.1040.0325]
- [6] Pundoor G, Chen ZL. Scheduling a production-distribution system to optimize the tradeoff between delivery tardiness and distribution cost. *Naval Research Logistics*, 2005,52:571–589. [doi: 10.1002/nav.20100]
- [7] Agnetis A, Hall NG, Pacciarelli D. Supply chain scheduling: Sequence coordination. *Discrete Applied Mathematics*, 2006,154: 2044–2063. [doi: 10.1016/j.dam.2005.04.019]
- [8] Gong H, Tang LX. The coordination of two parallel machines scheduling and batch deliveries. In: *Proc. of the COCOON 2008*. LNCS 5092, 2008. 670–677. [doi: 10.1007/978-3-540-69733-6_66]
- [9] Selvarajah E, Steiner G. Approximation algorithms for the supplier's supply chain scheduling problem to minimize delivery and inventory holding costs. *Operations Research*, 2009,57:426–438. [doi: 10.1287/opre.1080.0622]
- [10] Zegordi SH, Abadi INK, Nia MAB. A novel genetic algorithm for solving production and transportation scheduling in a two-stage supply chain. *Computers & Industrial Engineering*, 2010,58:373–381. [doi: 10.1016/j.cie.2009.06.012]
- [11] Li KP, Ma SH. Study of synchronized scheduling of parallel machine manufacturing and air transportation within ATO supply chain. *Systems Engineering-Theory & Practice*, 2007,12:9–51 (in Chinese with English abstract).
- [12] Sawik T. Coordinated supply chain scheduling. *Int'l Journal of Production Economics*, 2009,120:437–451. [doi: 10.1016/j.ijpe.2008.08.059]
- [13] Yimer AD, Demirli K. A genetic approach to two-phase optimization of dynamic supply chain scheduling. *Computers & Industrial Engineering*, 2010,58:411–422. [doi: 10.1016/j.cie.2009.01.010]
- [14] Yao JM, Liu LW. Optimization analysis of supply chain scheduling in mass customization. *Int'l Journal of Production Economics*, 2009, 117:197–211. [doi: 10.1016/j.ijpe.2008.10.008]
- [15] Bhatnagar R, Mehta P, Teo CC. Coordination of planning and scheduling decisions in global supply chains with dual supply modes. *Int'l Journal of Production Economics*, 2011,131:473–482. [doi: 10.1016/j.ijpe.2011.01.011]

- [16] Yeung WK, Choi TM, Cheng TCE. Optimal scheduling of a single-supplier single-manufacturer supply chain with common due windows. *IEEE Trans. on Automatic Control*, 2010,55:2767–2777. [doi: 10.1109/TAC.2010.2049766]
- [17] Chen ZL, Hall NG. Supply chain scheduling: Conflict and cooperation in assembly systems. *Operations Research*, 2007,55: 1072–1089. [doi: 10.1287/opre.1070.0412]
- [18] Dan B, Xiao J, Liu XH, Zhang XM, Li ZW, Tang HL. Batch scheduling of a multi-stage supply chain with due windows. *Computer Integrated Manufacturing Systems*, 2007,13(2):113–613 (in Chinese with English abstract).
- [19] Yao JM, Liu LW, Pu Y, Zhang XM. Analysis on ants optimization algorithm for supply chain dynamic scheduling in mass customization. *Journal of Management Sciences in China*, 2007,10(3):7–41 (in Chinese with English abstract).
- [20] Fonseca CM, Fleming PJ. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In: *Proc. of the 5th Int'l Conf. on Genetic Algorithms*. 1993. 416–423.
- [21] Keeratitittumrong N, Chaiyaratana N, Varavithya V. Multi-Objective co-operative co-evolutionary genetic algorithm. In: *Guervós JJM, Adamidis PA, Beyer HG, Fernández-Villacañas JL, Schwefel HP, eds. Proc. of the PPSN 2002. LNCS 2439, Heidelberg: Springer-Verlag, 2002. 288–297. [doi: 10.1007/3-540-45712-7_28]*
- [22] Iorio AW, Li XD. A cooperative coevolutionary multiobjective algorithm using nondominated sorting. In: *Deb K, et al., eds. Proc. of the GECCO 2004. LNCS 3102, Heidelberg: Springer-Verlag, 2004. 537–548. [doi: 10.1007/978-3-540-24854-5_56]*
- [23] Fard SM, Hamzeh A, Ziarati K. A grid based cooperative co-evolutionary multi-objective algorithm. In: *Deng H, et al., eds. Proc. of the AICI 2009. LNAI 5855, 2009. 167–175. [doi: 10.1007/978-3-642-05253-8_19]*

附中文参考文献:

- [11] 李昆鹏, 马士华. ATO 供应链中航空运输及并行机生产协调调度问题. *系统工程理论与实践*, 2007, 12: 9–51.
- [18] 但斌, 肖剑, 刘晓红, 张旭梅, 李志威, 汤华丽. 基于交货期窗口约束的多级供应链批量调度问题研究. *计算机集成制造系统*, 2007, 13(2): 113–613.
- [19] 姚建明, 刘丽文, 蒲云, 张秀敏. MC 模式下供应链动态调度的蚁群寻优分析. *管理科学学报*, 2007, 10(3): 7–41.



苏生(1978—),男,四川南充人,博士,副教授,主要研究领域为复杂系统智能调度,演化算法,互联网搜索引擎.

E-mail: susheng@uestc.edu.cn



于海杰(1977—),女,博士,讲师,主要研究领域为供应链管理,复杂系统调度.

E-mail: yuhaijie@uestc.edu.cn



吴正华(1971—),男,博士,副教授,主要研究领域为人机交互技术,脑机接口技术.

E-mail: wzhxwz@sina.com



姚远哲(1973—),男,博士,讲师,主要研究领域为分布式计算.

E-mail: yaoyuanzhe@gmail.com



张良(1965—),男,教授级高工,博士生导师,主要研究领域为供应链管理.

E-mail: zhangl@lzlj.com.cn