

文章编号: 1001-0920(2013)02-0217-06

## 一类求解订单分配和排序问题的集成优化算法

蒋大奎, 李波, 谭佳音

(天津大学 管理与经济学部, 天津 300072)

**摘要:** 研究大规模订单分配和排序问题. 从整体的角度协同优化多工厂供应链排序中的订单分配、生产调度和分批运输调度 3 个阶段; 以总订货提前期与总成本的加权和最小化为目标, 建立了问题的数学模型; 基于解的最优化条件, 设计了禁忌搜索算法与动态规划方法相结合的集成优化算法; 在算法中设计了自适应选择机制, 以提高算法的搜索效率. 数值实验结果验证了采用所提出的集成优化算法求解规模较大问题的有效性.

**关键词:** 供应链排序; 多工厂; 禁忌搜索; 动态规划; 自适应选择

**中图分类号:** TP29

**文献标志码:** A

## Integrated optimization approach for order assignment and scheduling problem

JIANG Da-kui, LI Bo, TAN Jia-yin

(College of Management and Economics, Tianjin University, Tianjin 300072, China. Correspondent: JIANG Da-kui, E-mail: jiangdakui@tju.edu.cn)

**Abstract:** The large scale order assignment and scheduling problem is studied. Decisions of different functions such as order assignment, production scheduling and batch delivery scheduling are integrated as a multi-plant supply chain scheduling for the simultaneous optimization. A mathematical model for the problem is built to minimize a weight sum of the total lead time and the total cost. Moreover, based on the optimality properties satisfied by the problem, an integrated approach integrating tabu search and dynamic programming method is developed to solve the problem. Meanwhile, an adaptive selection is designed to improve the efficiency of the algorithm. Experimental results show that the integrated approach is effective for solving the relative large problem.

**Key words:** supply chain scheduling; multiple plants; tabu search; dynamic programming; adaptive selection

### 0 引言

订单分配和排序问题<sup>[1]</sup>是一类较新的多工厂供应链排序问题<sup>[2]</sup>, 它以系统的角度集成研究订单分配、生产调度和分批运输调度 3 个阶段, 在具体运作层面使用确定性模型研究供应链问题, 是排序理论在供应链管理中的应用.

Chen<sup>[3]</sup>对供应链排序问题的研究现状进行了归纳总结, 从其结论中可以看出, 目前关于单工厂供应链排序问题的研究较多, 而针对订单分配和排序问题的研究则较少. 在订单分配和排序问题的研究中, Chen<sup>[1]</sup>针对时间敏感性产品, 以客户满意度和总成本加权和最小化为目标研究了多个海外工厂与一个本土分销中心组成的全球供应链排序问题, 分别以最长订货提前期和总订货提前期作为客户满意度衡量指

标, 证明了问题是 NP-hard 问题, 并给出了求解问题的启发式算法; Li 等<sup>[4]</sup>针对两个工厂、订单有两个组件的情况, 以总订货提前期和总成本加权和最小化为目标设计了启发式算法; 柏孟卓等<sup>[5-6]</sup>分别以总订货提前期、最大延迟时间和延误订单数量作为客户满意度衡量指标, 给出了求解问题的动态规划方法. 然而, 当问题规模较大时, 以上求解方法存在求解效率低或解值差等缺陷. 近几年出现了一些采用元启发式算法求解大规模生产配送协同优化问题的研究成果. 吴学静等<sup>[7-8]</sup>针对有交货时间限制的多产品订单情况, 分别采用粒子群算法和禁忌搜索算法对问题进行求解; 蒋大奎等<sup>[9]</sup>以最长订货提前期与总成本加权和最小化为目标, 设计了求解问题的混合禁忌搜索算法. 对于考虑总订货提前期的大规模订单分配和排序问题,

收稿日期: 2011-10-27; 修回日期: 2012-01-16.

基金项目: 新世纪优秀人才支持计划项目(NCET-06-0236); 高等学校博士学科点专项科研基金项目(20100032110034).

作者简介: 蒋大奎(1981—), 男, 博士生, 从事生产调度、物流调度、智能优化算法等研究; 李波(1967—), 女, 教授, 博士生导师, 从事物流与供应链管理等研究.

目前尚未见到元启发式算法研究成果的相关报道. 为此, 本文以总订货提前期与总成本加权和最小化为目标, 设计了一种禁忌搜索与动态规划方法相结合的集成优化算法, 并通过仿真实验验证了该算法的性能和有效性.

## 1 问题描述

本文所研究的问题(以下简称为 P)可描述如下: 某制造商在不同地理位置有  $m$  个工厂和单个客户, 工厂集合  $M = \{1, 2, \dots, m\}$ , 每个工厂的生产环境为单台机器; 制造商需要生产和发送来自客户的  $n$  个订单, 订单集合  $N = \{1, 2, \dots, n\}$ ; 每个订单只需在一个工厂不间断生产一次; 订单  $j$  在工厂  $i$  的生产时间为  $p_{ij}$ , 生产成本为  $c_{ij}$ ,  $i \in M, j \in N$ ; 订单在工厂生产完成后按批次发送给客户, 每批次最多发送  $b$  个订单; 工厂  $i$  发送一个批次的运输时间为  $t_i$ , 运输成本为  $f_i$ ; 制造商需要协同优化订单分配、生产调度和分批运输调度, 通过制定集成调度方案(以下简称为方案)使总订货提前期与总成本的加权和最小.

为方便表述, 本文采用文献[9]的 3 参数法  $(\delta, \sigma, \varphi)$  表示一个方案. 其中: 参数  $\delta$  表示一个订单分配子方案;  $\sigma$  表示  $\delta$  给定下的一个生产调度子方案;  $\varphi$  表示  $\delta$  给定下的一个分批运输调度子方案. 记方案集合为  $\Omega$ , 生产调度子方案集合为  $\Theta$ , 采用生产调度子方案  $\sigma$  的方案集合为  $\Omega_\sigma$ .

对于一个方案  $(\delta, \sigma, \varphi)$ , 本文给出以下定义:

$D_{\text{total}}(\delta, \sigma, \varphi)$ : 采用方案  $(\delta, \sigma, \varphi)$  时的总订货提前期;

$C(\delta, \sigma, \varphi)$ : 采用方案  $(\delta, \sigma, \varphi)$  时的生产和运输总成本.

对于给定的权值  $\alpha \in [0, 1]$ , 问题 P 的数学模型可表示为

$$F = \min\{\alpha D_{\text{total}}(\delta, \sigma, \varphi) + (1 - \alpha)C(\delta, \sigma, \varphi) | (\delta, \sigma, \varphi) \in \Omega\}. \quad (1)$$

当生产调度子方案  $\sigma$  给定时, 问题 P 的数学模型可表示为

$$F(\sigma) = \min\{\alpha D_{\text{total}}(\delta, \sigma, \varphi) + (1 - \alpha)C(\delta, \sigma, \varphi) | (\delta, \sigma, \varphi) \in \Omega_\sigma\}. \quad (2)$$

因此, 式(1)也可表示为

$$F = \min\{F(\sigma) | \sigma \in \Theta\}. \quad (3)$$

问题 P 的最优解具有以下性质<sup>[1,3]</sup>.

**定理 1** 问题 P 存在具有如下最优化性质的最优方案:

- 1) 同一个工厂生产任何两个订单之间没有空闲;
- 2) 每一批订单的发送均发生在该批订单中的某

一个订单的生产完成时间;

- 3) 每个工厂生产的所有订单按生产顺序进行发送.

**定理 2** 对于问题 P, 当订单分配子方案给定时, 存在一个最优生产调度子方案: 分配给每个工厂的所有订单按最短生产时间优先(SPT)规则安排生产.

## 2 集成优化算法设计

禁忌搜索算法(TS)作为一类搜索效率较高的元启发式算法, 在求解背包问题<sup>[10]</sup>和生产调度<sup>[11]</sup>等经典组合优化问题上具有较好的寻优能力和搜索效率. 文献[1]设计了一种动态规划方法, 在订单分配和生产调度结果给定条件下, 该算法能够在多项式时间内求得总订货提前期与运输总成本加权和最小的方案. 为此, 本文设计了 TS 与动态规划法相结合的集成优化算法. 该算法将 TS 作为集成优化算法的外层, 以选择订单分配子方案和生产调度子方案; 动态规划方法作为集成优化算法的内层, 用以求解订单分配子方案和生产调度子方案给定条件下的最优方案. 以上两种方法的有机结合使得集成优化算法既具有全局收敛性, 又具有较高的求解质量和效率. 集成优化算法的基本操作包括动态规划方法、编码结构、生成初始解、自适应选择机制和邻域操作等. 下面对各个操作进行详细说明.

### 2.1 动态规划方法

当订单分配子方案  $\delta$  和生产调度子方案  $\sigma$  给定时, 式(2)可通过动态规划方法进行求解. 下面给出动态规划方法.

记按照订单分配子方案  $\delta$  分配给工厂  $i$  的订单集合为  $N_i(\delta)$ ,  $N_i(\delta)$  中的订单数量为  $n_i(\delta)$ ; 按照生产调度子方案  $\sigma$ , 工厂  $i$  第  $l$  个位序生产的订单为  $[il]$ .

依次计算每个工厂  $i$  完成  $N_i(\delta)$  中所有订单的最优目标值  $F_i(n_i(\delta))$ .

$F_i(n_i(\delta))$  的计算方法为

$$F_i(n_i(\delta)) = V_i(n_i(\delta)) + (1 - \alpha) \sum_{l=1}^{n_i(\delta)} c_{i[il]},$$

其中  $V_i(n_i(\delta))$  通过以下动态规划方法求得<sup>[11]</sup>:

$V_i(l)$  表示工厂  $i$  完成前  $l$  个订单的总订货提前期与运输总成本加权和最小值;

初始条件为  $V_i(0) = 0$ ;

递推关系

For  $l = 1, 2, \dots, n_i(\delta)$ ,

$$V_i(l) = \min \left\{ V_i(l-h) + \alpha h \left( \sum_{k=1}^l p_{i[ik]} + t_i \right) + (1 - \alpha)f_i | h = 1, 2, \dots, \min(b, l) \right\};$$

最优值为  $V_i(n_i(\delta))$ .

在递推关系中, 算法通过枚举最后一个批次发送的订单数量  $h$  得到每个状态下的最优分批运输子方案. 对于每个工厂  $i$ , 上述迭代过程的时间复杂度为  $O(bn_i(\delta))$ .

综上,  $F(\sigma)$  的值可通过下式得到:

$$F(\sigma) = \sum_{i \in M} F_i(n_i(\delta)).$$

## 2.2 编码结构及生成初始解

对于某具有  $m$  个工厂、 $n$  个订单的问题, 共有  $m^n$  个订单分配子方案. 对于每个订单分配子方案, 可通过定理 2 得到一个最优的生产调度子方案. 当订单分配子方案和生产调度子方案选定后, 可通过上节给出的动态规划方法得到最优方案. 因此, 本文以生产调度子方案作为算法的解, 采用基于序数编码的向量组编码结构<sup>[9]</sup>对其进行描述, 式(2)作为算法的适应度函数.

在向量组编码结构中, 向量组的向量个数为工厂数量  $m$ , 向量  $V_i$  表示工厂  $i$ , 向量  $V_i$  中每个元素为一个订单, 订单在向量中的顺序为订单的生产顺序. 以 2 个工厂 4 个订单问题为例, 若订单 1 和订单 2 在工厂 1 的生产顺序为  $1 \rightarrow 2$ , 订单 3 和订单 4 在工厂 2 的生产顺序为  $3 \rightarrow 4$ , 则向量组如图 1(a) 所示. 与图 1(b) 所示的基于二进制编码的矩阵编码结构<sup>[12]</sup>相比, 采用向量组编码结构可使编码空间中的个体数量由  $((m \times n \times n)!)/(n \times (m \times n \times n - n)!)$  个减少到  $((m + n - 1)!)/(m - 1)!$  个, 且编码和解码过程更加简单.

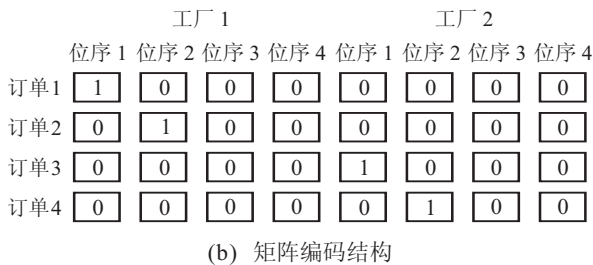
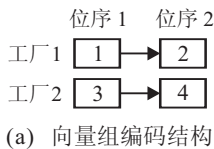


图 1 两种编码结构示意图

禁忌搜索算法对初始解具有较强的依赖性, 好的初始解有利于提高搜索效率. 本文利用定理 2 生成一个较优的初始解, 生成初始解的步骤如下.

**Step 1:** 将集合  $N$  中的所有订单随机分配给各个工厂, 生成一个订单分配子方案  $\delta$ ;

**Step 2:** 对分配给每个工厂  $i$  的所有订单按 SPT 规则排序, 生成  $\delta$  给定下的生产调度子方案  $\sigma$ ;

**Step 3:** 按照  $\sigma$  生成向量组, 并采用上节动态规划方法求解.

## 2.3 邻域操作

根据向量组编码结构特点, 并使邻域操作结果仍然满足定理 2, 本文设计了两种邻域操作, 即插入操作和突变操作.

插入操作是从向量组的任意非空向量  $V_x$  中随机取出一个订单  $j$ ; 然后将订单  $j$  插入向量  $V_y, x \neq y$ , 并使插入订单  $j$  后的向量  $V_y$  中订单排序仍然符合 SPT 规则. 具体操作步骤如下.

**Step 1:** 在向量组中随机选择一个非空向量  $V_x$ , 并在  $V_x$  中随机取出一个订单  $j$ .

**Step 2:** 在向量组中随机选择一个向量  $V_y, x \neq y$ ; 记  $V_y$  中的元素数量为  $Q_y$ , 记  $V_y$  中第  $l$  个位置的订单为  $[y]_l$ . 分别按如下 4 种情况考虑:

- 1) 若  $Q_y = 0$ , 则订单  $j$  直接插入向量  $V_y$ ;
- 2) 若  $Q_y \neq 0$  且  $p_{yj} \leq p_{y1}$ , 则将订单  $j$  插入向量  $V_y$  的第 1 个位置;
- 3) 若  $Q_y \neq 0$  且  $p_{yj} \geq p_{y[Q_y]}$ , 则将订单  $j$  插入向量  $V_y$  的尾部.
- 4) 对于其余情况, 用二分法查找  $l$ , 使得  $p_{y[y(l-1)]} \leq p_{yj} \leq p_{y[y]_l}$ , 并将订单  $j$  插入向量  $V_y$  的第  $l$  个位置.

经过插入操作后, 将采用订单分配子方案  $\delta$  的生产调度子方案  $\sigma$  改变为采用订单分配子方案  $\delta'$  的生产调度子方案  $\sigma'$ , 生产调度子方案  $\sigma'$  的适应值由下式计算得出:

$$F(\sigma') = F(\sigma) - F_x(n_x(\delta)) - F_y(n_y(\delta)) + F_x(n_x(\delta')) + F_y(n_y(\delta')). \quad (4)$$

突变操作是从向量组中的多个非空向量中随机选取多个订单, 并将选中的订单随机插入其他向量, 使得插入订单后向量组中各个向量的订单排序仍然符合 SPT 规则. 通过突变操作生成的生产调度子方案  $\sigma'$  的适应值由式(2)计算得出.

## 2.4 自适应选择机制

在邻域操作的设计上, 本文不仅设计了爬山能力强的插入操作, 还设计了跳跃能力强的突变操作, 从而提高算法的全局寻优能力和局部搜索效率. 在算法迭代的初始阶段, 由于初始解的生成具有一定的随机性, 算法以较高的概率执行突变操作, 在较广的区域内搜索较优解; 当算法迭代到一定阶段时, 当前最优解处于适应值较高的区域, 考虑到突变操作会破坏当前解的结构, 应以较高的概率执行插入操作, 以提高局部寻优效率; 在之后的迭代过程中, 当最优解难以得到进一步优化时, 需再次调高执行突变操作的概率. 为此, 本文设计了自适应选择机制, 针对迭代次数  $t$  和

连续迭代最优解未改变次数  $t'$ , 以概率  $p$  自适应地选择突变操作. 概率  $p$  的计算公式如下:

$$p = \begin{cases} \exp \frac{\gamma_1 t' - 1}{\gamma_2 t}, & \gamma_1 t' \leq 1; \\ p_1, & \gamma_1 t' > 1. \end{cases} \quad (5)$$

其中:  $p_1$  为常量,  $0 \leq p_1 \leq 1$ ;  $\gamma_1$  为  $t'$  对  $p$  的影响系数;  $\gamma_2$  为  $t$  对  $p$  的影响系数.

式 (5) 中,  $p$  为关于  $t'$  的增函数, 关于  $t$  的减函数. 当  $t$  较小时, 即算法迭代初始阶段,  $p$  较大; 当  $t$  逐渐增大时,  $p$  逐渐减小, 即执行突变操作的概率降低; 当  $t'$  增大时,  $p$  也随之增大, 即执行突变操作的概率提高.

## 2.5 禁忌规则及终止条件

本文算法的禁忌规则如下: 如果向量  $V_i$  中的订单  $j$  通过插入操作移动到其他向量中, 则在接下来的  $\lambda$  代中不允许订单  $j$  被移回到向量  $V_i$ ; 如果经过邻域操作得到比当前解更好的解, 则不管该邻域操作是否被禁忌, 都采用该邻域作为当前邻域. 突变操作不受禁忌规则制约. 算法的终止条件为: 达到预先设定的迭代次数时, 算法终止.

## 2.6 算法框架

本文算法的框架如图 2 所示.

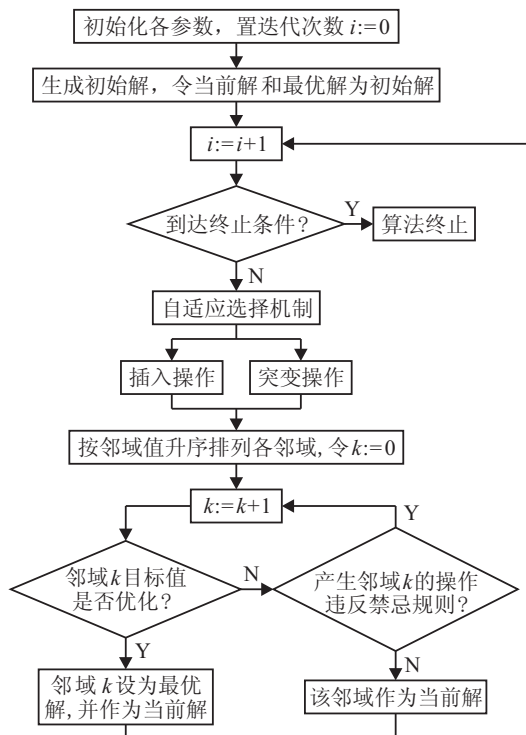


图 2 算法框架

## 3 仿真实验

本文算例选取工厂数量为  $m \in \{4, 6, 8, 10, 12\}$ , 订单数量为  $n \in \{100, 150, 200, 250, 300\}$ , 每个批次最多可发送的订单数量为  $b \in \{3, 6, 9, 12, 15\}$ . 为避免具体算例对算法性能可信度的影响, 本文通过随机方式

生成以下数据: 各工厂到客户的运输时间和运输成本均服从均匀分布  $U[100, 1000]$ , 订单在各工厂的生产时间服从  $U[10, 100]$ , 生产成本服从  $U[100, 500]$ .

### 3.1 不同算法比较

为了验证本文算法的有效性, 本节选择 13 组随机算例  $(m, n, b)$  对本文算法与文献 [1] 的 H1-IMP 启发式算法进行对比实验. 在 13 组随机算例中, 目标函数中的权值  $\alpha = 0.5$ . 本文集成优化算法通过 Visual C++2005 实现, 并使用 C++ 标准模板库. PC 为 AMD Athlon (tm) II X2 245, 4 GB RAM. 本文算法的参数如下: 禁忌表长度为 20, 候选解集为 40, 迭代次数为 500,  $p_1 = 0.2, \gamma_1 = 0.05, \gamma_2 = 0.3$ .

对 13 组随机算例分别运算 10 次, 比较结果如表 1 所示. 由表 1 可以看出: 与 H1-IMP 算法相比, 对于每组算例, 本文集成优化算法在解值平均损失约 0.25% 的情况下节省约 76% 的计算时间; 在大多数算例中, 本文算法能够得到与 H1-IMP 算法相同或更优的解. 由于初始解和邻域操作具有随机性, 本文算法对同一算例的多次运算结果具有一定差异. 对于 13 组算例, 本文均采用固定的禁忌表长度、候选解集和迭代次数, 因此当  $m$  和  $n$  增大时, 本文算法多次运算结果的标准差增大. 若增大禁忌表长度、候选解集或迭代次数, 本文算法则有机会得到更加优化的解或缩小标准差.

进一步, 考虑两种算法对参数  $m, n$  和  $b$  的时间敏感性. 如图 3(a) 所示, 当  $n$  和  $b$  不变  $m$  增大时, 本文算法的计算时间随  $m$  的增大而减少, H1-IMP 的计算时间却随  $m$  的增大而增加; 如图 3(b) 所示, 当  $m$  和  $b$  不变  $n$  增大时, 本文算法和 H1-IMP 的计算时间均增加, 但本文算法计算时间增幅较缓; 如图 3(c) 所示, 当  $m$  和  $n$  不变  $b$  增大时, 本文算法和 H1-IMP 的计算时间均增加. 由于本文算法的计算时间主要取决于动态规划方法的求解上, 动态规划方法的决策过程按订单单位序划分阶段, 每个阶段  $l$  的状态变量为  $\min\{b, l\}$ . 当工厂数量增大时, 分配给每个工厂的订单数量减少, 因此动态规划方法计算时间缩短; 当订单数量增大时, 分配给每个工厂的订单数量增大, 因此动态规划方法计算时间有所增加; 当每批次最多可发送的订单数量增大时, 动态规划方法的计算时间将有所增加.

### 3.2 自适应选择机制

下面对自适应选择的有效性进行仿真实验. 随机生成一组算例  $(4, 200, 12)$ , 权值  $\alpha = 0.5$ . 分别对采用自适应选择机制和固定概率选择机制的集成优化算法进行求解. 其中在固定概率选择机制的算法中, 分别考虑了插入优先  $(p=0.2)$  和突变优先  $(p=0.8)$  的两种情况. 分别对算例运算 10 次, 比较结果如表 2 所示.

表1 不同算法比较结果

算例	本文集成优化算法					H1-IMP		算法偏差率	时间偏差率
	最优解	最差解	标准差	平均解 $M_R$	平均计算时间 $T_1$	解值 $R$	计算时间 $T_2$		
(4, 200, 12)	130 468	130 608	44.853	130 530.1	18.405	130 411	37.297	0.091	-50.653
(6, 200, 12)	116 842	117 078.5	89.377 87	116 948.3	13.184	116 842	44.609	0.091	-70.445
(8, 200, 12)	119 918.5	120 305	158.057	120 135.3	10.798	119 918.5	63.468	0.181	-80.878
(10, 200, 12)	115 570	116 058	213.345	115 876	9.881	115 570	68.266	0.265	-85.525
(12, 200, 12)	85 672.5	86 284.5	165.416	86 075.75	9.431	85 762.5	98.25	0.365	-90.401
(8, 100, 12)	46 041.5	46 149	38.008	46 095.1	5.144	46 041.5	10.062	0.116	-48.877
(8, 150, 12)	77 675.5	77 909	66.581	77 820.05	7.692	77 675.5	27.578	0.186	-72.108
(8, 250, 12)	155 494	156 116.5	223.287	155 882	14.091	155 494	119.375	0.25	-88.196
(8, 300, 12)	196 639.5	196 956	103.122	196 787.8	18.045	196 190.5	239.906	0.304	-92.467
(8, 200, 3)	118 227	119 477.5	336.359 3	119 102.5	5.964	118 272	35.859	0.702	-83.368
(8, 200, 6)	104 835	105 432.5	225.979	105 112.1	8.669	104 835	49.734	0.264	-82.569
(8, 200, 9)	11 484.5	111 777	129.222	111 627.2	10.331	111 484.5	56.265	0.128	-81.639
(8, 200, 15)	122 505.5	122 807	116.723	122 695.1	12.566	122 548	73.75	0.12	-82.961

注：算法偏差率 =  $(M_R - R)/M_R \times 100\%$ ；时间偏差率 =  $(T_1 - T_2)/T_2 \times 100\%$ 。

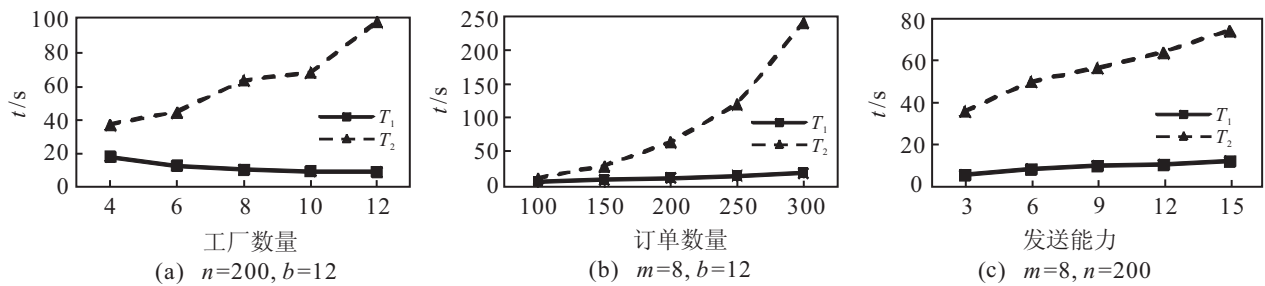


图3 两种算法对不同参数的敏感性分析

表2 不同选择机制比较结果

选择机制	最优解	最差解	标准差	平均解	平均时间
自适应选择	142 034.5	143 121	319.268	142 641.6	21.3
插入优先	142 375.5	143 148.5	255.657	142 816.5	16.742
突变优先	166 611.5	168 598.5	686.107	167 463.1	86.03

由表2可知，采用自适应选择机制和插入优先机制的算法能够得到较为满意的解，且插入优先机制的算法的平均计算时间最短。由于采用自适应选择机制的算法在算法迭代初期以较高的概率执行突变操作，其计算时间较插入优先机制的算法稍长。

下面对采用3种机制的算法的收敛性进行比较。由图4可以看出：与采用插入优先机制的算法相比，采用自适应选择机制和突变优先机制的算法在算法迭代初期具有较强的搜索效率；经过一段迭代过程后，突变优先机制的算法搜索效率大幅度下降，而自适应选择机制仍然能够保持较好的搜索效率。

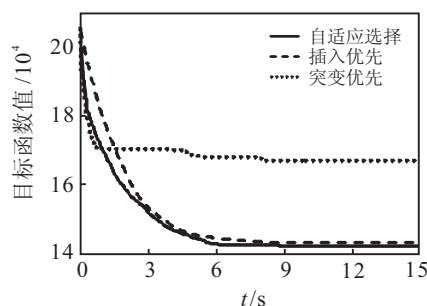


图4 3种机制收敛性对比

### 3.3 不同决策偏好比较

本节考虑决策者对总订货提前期和总成本的不同偏好，即选择不同的权值  $\alpha \in \{0.2, 0.5, 0.8\}$ 。  $\alpha$  越小，表示决策者越偏好于缩短总订货提前期；  $\alpha$  越大，表示决策者越偏好于降低总成本。对于第4.1节的6组随机算例，本节对每个权值分别运行10次，并选取最优解，结果如表3所示。

表3 不同决策偏好

算例	$\alpha$	总订货提前期	总成本	解值
(4, 200, 12)	0.2	225 349	53 974	88 249
	0.5	190 715	70 107	130 411
	0.8	180 862	88 612	162 412
(12, 200, 12)	0.2	143 436	44 007	63 892.8
	0.5	112 230	59 295	85 762.5
	0.8	99 183	83 056	95 957.6
(8, 100, 12)	0.2	75 995	24 750	34 999
	0.5	59 743	32 340	46 041.5
	0.8	53 539	44 571	51 745.4
(8, 300, 12)	0.2	349 763	71 575	127 212.6
	0.5	296 019	96 362	196 190.5
	0.8	276 737	131 707	247 731
(8, 200, 3)	0.2	194 578	56 165	83 847.6
	0.5	169 492	66 962	118 227
	0.8	161 410	80 574	145 242.8
(8, 200, 15)	0.2	217 446	45 930	80 233.2
	0.5	183 938	61 067	122 502.5
	0.8	171 343	81 981	153 470.6

由表3可以看出：对于6组算例，总订货提前期随

$\alpha$  的增大而减小, 即如果决策者对缩短总订货提前期的偏好增强, 则最优方案的总订货提前期缩短, 总成本增加; 而总成本随  $\alpha$  的增大而增大, 即如果决策者对降低总成本的偏好增强, 则最优方案的总成本减少, 总订货提前期增加. 结果符合设计算法时的要求, 即当决策者对客户服务水平较为关注时, 可减小  $\alpha$  值; 当决策者更关心成本时, 可增大  $\alpha$  值.

#### 4 结 论

本文针对规模较大的订单分配和排序问题, 以总订货提前期和总成本加权和最小化为目标建立了问题的数学模型, 并设计了基于禁忌搜索和动态规划方法的集成优化算法. 在算法中加入了向量组编码结构和自适应选择机制, 以提高搜索效率. 仿真实验结果表明, 本文算法可以有效地解决规模较大的订单分配和排序问题.

本文研究的问题中没有考虑工厂有多台机器、多客户等情况, 考虑这些情况的研究将更具实际应用价值.

#### 参考文献(References)

- [1] Chen Zhilong, Pundoor G. Order assignment and scheduling in a supply chain[J]. *Operations Research*, 2006, 54(3): 555-572.
- [2] Hall N G, Potts C N. Supply chain scheduling: Batching and delivery[J]. *Operations Research*, 2003, 51(4): 566-584.
- [3] Chen Zhilong. Integrated production and outbound distribution scheduling: Review and extensions[J]. *Operations Research*, 2010, 58(1): 130-148.
- [4] Li Chunglun, Ou Jinwen. Coordinated scheduling of customer orders with decentralized machine locations[J]. *IIE Transactions*, 2007, 39(9): 899-909.
- [5] 柏孟卓, 陈峰, 唐国春. 供应链管理中生产和运输集成的排序问题[J]. *工业工程与管理*, 2007, 12(5): 47-50.  
(Bai M Z, Chen F, Tang G C. Integrated production and distribution scheduling in supply chain management[J]. *Industrial Engineering and Management*, 2007, 12(5): 47-50.)
- [6] 柏孟卓, 唐国春. 与交货期有关的供应链排序问题[J]. *运筹学学报*, 2009, 13(1): 113-119.  
(Bai M Z, Tang G C. Integrated production and distribution scheduling in supply chain management[J]. *OR Transactions*, 2009, 13(1): 113-119.)
- [7] 吴学静, 周泓, 梁春华. 基于协同进化粒子群的多层供应链协同优化[J]. *计算机集成制造系统*, 2010, 16(1): 127-132.  
(Wu X J, Zhou H, Liang C H. Collaborative optimization of multi-echelon supply chain based on co-evolutionary particle swarm optimization[J]. *Computer Integrated Manufacturing Systems*, 2010, 16(1): 127-132.)
- [8] 梁春华, 周泓, 赵健. 一种求解供应链环境下生产-配送问题的集成优化方法[J]. *控制与决策*, 2011, 26(1): 27-31.  
(Liang C H, Zhou H, Zhao J. Integrated optimization approach for production-distribution planning in supply chain[J]. *Control and Decision*, 2011, 26(1): 27-31.)
- [9] 蒋大奎, 李波. 基于混合禁忌搜索算法的供应链排序问题[J]. *机械工程学报*, 2011, 47(20): 53-59.  
(Jiang D K, Li B. Supply chain scheduling based on hybrid taboo search algorithm[J]. *Chinese J of Mechanical Engineering*, 2011, 47(20): 53-59.)
- [10] Lokketangen A, Glover F. Solving zero-one mixed integer programming problem using tabu search[J]. *European J of Operational Research*, 1998, 106(2/3): 624-658.
- [11] Zhang Chaoyong, Li Peigen, Guan Zailin, et al. A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem[J]. *Computers & Operations Research*, 2007, 34(11): 3229-3242.
- [12] Yang Fengcheng, Chen Kuentai, Wang Mingtzong, et al. Mathematical modeling of multi-plant order allocation problem and solving by genetic algorithm with matrix representation[J]. *Int J of Advanced Manufacturing Technology*, 2010, 51(9): 1251-1259.

## 下 期 要 目

- 多机械臂协调控制研究综述 . . . . . 甘亚辉, 戴先中  
 基于前景理论的多目标灰靶决策方法 . . . . . 刘 勇, 等  
 连续 Markov 跳变系统最优控制 . . . . . 胡诗国, 等  
 基于特征结构配置的二阶线性系统鲁棒容错控制设计 . . . . . 王国胜, 等  
 基于反馈增益的 AUV 稳定神经网络反步变深控制 . . . . . 陈子印, 等  
 基于和谐管理理论的免疫信息网络优化算法 . . . . . 孙奕菲, 等