

混流装配生产线准时化物料补给调度方法

周炳海[†], 彭 涛

(同济大学 机械与能源工程学院, 上海 201804)

摘 要: 针对准时化顺序供应的混流装配生产线物料补给问题进行数学规划建模, 通过决策各次物料补给作业的发车时间和料箱标号使线边物料的库存成本最小. 为了解决这一复杂的混合优化问题, 给出了问题的性质分析, 并将该问题转化为求解最优料箱配送序列的组合优化问题. 在此基础上, 构建反向动态规划求解算法以获得该问题的最优解, 并证明该算法具有指数级别的时间复杂度. 为了求解中大规模调度问题, 构建了改进蜂群算法, 通过在邻域搜索部分融合基于分布估计算法的个体更新机制来强化基本蜂群算法的收敛性能. 最后通过仿真实验验证了所提出算法的可行性和有效性.

关键词: 混流装配线; 物料补给; 调度; 动态规划; 人工蜂群算法

中图分类号: TP391

文献标志码: A

Scheduling methods of just-in-time material replenishment in mixed-model assembly lines

ZHOU Bing-hai[†], PENG Tao

(School of Mechanical Engineering, Tongji University, Shanghai 201804, China)

Abstract: A mathematical programming model is constructed to tackle the just-in-time material replenishment problem in mixed-model assembly lines with the just-in-sequence supply policy. The bin and the departure time of each delivery have to be determined to minimize the line-side material inventory cost. Subsequently, the property is analyzed to convert this composite optimization problem to the discrete combinational optimization one, aiming to determine the optimal bin delivery sequence. A backwards dynamic programming algorithm is presented to solve the problem optimally, which is proved to be exponential-time complexity. Meanwhile, a modified artificial bee colony metaheuristic is established to cope with the medium-and large-scale instances. Based on the estimation of distribution method, a novel individuals update rule is developed to enhance the performance of the original metaheuristic. Finally, simulation results show the feasibility and effectiveness of the proposed methods.

Keywords: mixed-model assembly lines; material replenishment; scheduling; dynamic programming; artificial bee colony

0 引 言

混流装配生产线能够装配工艺相近、结构相似的多产品, 因此, 各装配工位所需物料的种类急剧增加, 且不同物料的需求量波动性较大^[1]. 据此可知, 进行准时化物料补给问题的研究对提高混流装配线的生产效率具有重要的理论和现实意义.

近年来, 许多国内外学者针对混流装配生产线的准时化物料补给问题行了较深入的研究. 一些学者研究了由中心仓库至装配工位的远距离物料补给问题, 这类研究通常以最小化物料运输成本和线边库

存成本为目标, 借助车辆路径规划模型对问题进行建模与优化^[2-4]. 同时也有学者研究物料超市至装配工位的物料补给问题, 物料超市设置在混流装配线附近, 暂存和分拣来自中心仓库的物料并对临近工位进行多批次、小批量的物料配送作业^[5]. 其中 Emde 等^[6]针对基于物料超市的循环配送问题进行了建模分析, 给出了多项式时间算法以优化线边库存水平; Fathi 等^[7]基于此构建了循环配送问题的多目标优化模型, 并提出了融合优先级规则的粒子群算法以确定最优的配送次数和配送量; Boysen 等^[8]针对点对点配送

收稿日期: 2016-06-20; 修回日期: 2016-09-08.

基金项目: 国家自然科学基金项目(71471135, 61273035).

作者简介: 周炳海(1965—), 男, 教授, 博士生导师, 从事制造系统的建模、调度等研究; 彭涛(1992—), 男, 博士生, 从事物流系统建模、调度的研究.

[†]通讯作者. E-mail: bhzhou@tongji.edu.cn

问题进行了建模,提出模拟退火算法来有效地控制线边库存量;文献[9]对混流生产线的料箱配送问题进行了建模,并给出了这类问题的复杂度分析.

本文针对文献[9]提出的一类料箱配送问题进行建模,通过引入准时化顺序供应(JIS)策略构建以最小化线边库存成本为目标的混流装配生产线准时化物料补给问题的数学规划模型,并提出了求解该模型的反向动态规划算法和改进蜂群算法.最后,通过仿真实验验证了所提出调度算法的可行性和有效性.

1 数学建模

1.1 问题描述

在混流装配生产线中,在制品依次进入各个工位进行装配作业,来自中心仓库的零部件在物料超市进行分拣和重新装箱.从物料超市至装配工位的物料补给任务由一辆AGV小车完成,每次配送作业包含一个料箱,需要合理决策每次配送作业的料箱标号以及发车时间来实现准时化的装配线物料供应调度,具体过程如图1所示.

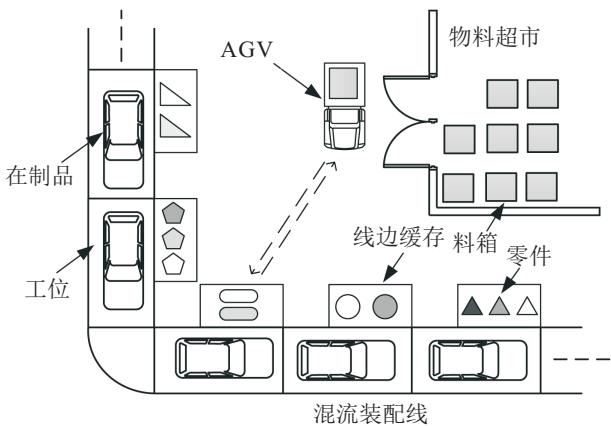


图1 混流装配线物料配送

为了有效地描述混流装配生产线的准时化物料补给问题,作如下假设:

- 1) 系统采用节拍作为基本时间单位,各工位的作业耗时均为一个节拍时间;
- 2) 每个工位装配一类物料,且各工位在装配作业开始时刻均需消耗一单元线边物料;
- 3) 产品装配顺序已知,各工位所需物料采用准时化顺序供应的方式依次装箱;
- 4) 每个料箱包含一种物料,该料箱的目标工位、交货期及包含的物料数量均已知;
- 5) 料箱的交货期设置为其对应装配子序列最早在该料箱的目标工位开始装配的时刻;
- 6) 超市与各工位间的补给时间已知(包括料装载或卸载时间);

7) 各类物料在单位时间内的线边库存成本费用已知,且线边库存费用与物料在装配线的驻留时间成正比.

由以上假设可知,各类物料与相应的装配工位实现了一一对应.

1.2 数学模型

为清晰地描述该调度问题,定义了以下3类符号:

1) 问题参数: B 为料箱集合(标号 b); M 为工位集合(标号 m); l_m 为超市与工位 m 间的配送时间; β_m 为工位 m 处的线边库存成本系数; $m(b)$ 为 B 中第 b 个料箱的目标工位; $t(b)$ 为 B 中第 b 个料箱的交货期; $n(b)$ 为 B 中第 b 个料箱所装的物料量.

2) 决策变量 $\pi = \{(b_i, \tau_i)\}$, τ_i 和 b_i 分别表示第 i 次配送作业的发车时间和料箱标号.

3) 目标函数 $C(\pi)$,为方案 π 产生的库存成本费用.

基于问题描述及符号定义,对混流装配线准时化物料补给问题建模如下:

$$\begin{aligned} \min C(\pi) = & \sum_{i=1}^{|\pi|} \sum_{j=1}^{n(b_i)} \beta_{m(b_i)} (t(b_i) + j - 1 - (\tau_i + l_{m(b_i)})). \\ \text{s.t. } & \tau_i + 2l_{m(b_i)} \leq \tau_{i+1}, \forall i \in \{1, 2, \dots, |\pi|\}; \quad (1) \\ & \tau_i + l_{m(b_i)} \leq t(b_i), \forall i \in \{1, 2, \dots, |\pi|\}. \quad (2) \end{aligned}$$

该模型通过决策各次物料补给作业的料箱标号和发车时间以最小化物料的线边库存成本总和,其中 $t(b_i) + j - 1 - (\tau_i + l_{m(b_i)})$ 表示第 i 次配送的料箱 b_i 中第 j 个物料的线边驻留时间;约束(1)表示每次物料补给仅包含一个料箱,即完成当前配送作业后才能开始下一次配送;约束(2)表示各料箱必须在交货期前到达对应的工位,即装配线不允许发生缺货.

1.3 问题性质

给定 $\pi = \{(b_i, \tau_i)\}$, b_i 为离散变量, τ_i 为连续变量,故该问题为复杂的混合优化问题.为了有效地求解该问题,给出了如下性质.给定配送序列 $\{b_i\}$,则该序列下各次物料补给的最优发车时间由下式唯一确定:

$$\tau_i^{\text{opt}} = \begin{cases} \tau_i = t(b_i) - l_{m(b_i)}, & i = |\pi|; \\ \min(\tau_{i+1} - 2l_{m(b_i)}, t(b_i) - l_{m(b_i)}), & i < |\pi|. \end{cases} \quad (3)$$

由式(3)可知, $\forall i \in \{1, 2, \dots, |\pi|\}$, τ_i^{opt} 满足约束(1)和(2);基于料箱配送序列 $\{b_i\}$ 构建配送方案 $\pi' = \{(\tau'_i, b_i)\}$ 和 $\pi'' = \{(\tau''_i, b_i)\}$, 满足 $\tau'_i < \tau_i^{\text{opt}}$ 和 $\tau_i'' =$

τ_i^{opt} , 则 $\Delta\tau_i = \tau'_i - \tau''_i < 0$, 故两调度方案的目标函数之差为

$$C(\pi'') - C(\pi') = \sum_{i=1}^{|\pi|} \left(\Delta\tau_i \cdot \sum_{j=1}^{n(b_i)} \beta_{m(b_i)} \right) < 0. \quad (4)$$

式(3)在满足装配线不允许缺货这一约束的同时, 尽可能使得配送序列中各料箱在相应的交货期时间到达其目标工位, 该性质可有效地将原问题转化为确定最优料箱配送序列的组合优化问题.

2 反向动态规划算法

为了有效地解决混流装配线准时化物料补给问题, 给出了反向动态规划算法(BDP)以获得该问题的全局最优解.

2.1 BDP算法

给定序列 $\{b_i\}$, 由式(3)可知最佳发车时间 $\tau_i^{\text{opt}} (\forall i)$ 由该次物料补给的交货期 $t(b_i)$ 及第 $i+1$ 次配补货作业的最佳发车时间 τ_{i+1}^{opt} 唯一确定. 故作反向决策, 将整个决策过程分为 $|B|$ 个阶段, 从后向前依次确定每次配送作业的料箱标号以获得最优序列.

给定 $S \subseteq B$ 表示后 $|S|$ 项配送作业的料箱集合, $f(S)$ 表示集合 S 以最优序列产生的最小线边库存费用, 则BDP算法的递归表达式为

$$f(S) = \begin{cases} g(k), & |S| = 1; \\ \min_{k \in S} \{f(S \setminus \{k\}) + g(k)\}, & |S| \geq 2. \end{cases} \quad (5)$$

其中: $g(k)$ 表示每个阶段的指标函数, 即倒数第 $|S|$ 次配送作业为料箱 $k (k \in S)$ 时, 该料箱所产生的线边库存成本费用. $g(k)$ 计算如下:

$$g(k) = \sum_{j=1}^{n(p_k)} \beta_{m(p_k)} (t(p_k) + j - 1 - \tau_k); \quad (6)$$

$$\tau_k = \begin{cases} t(p_k) - l_{m(p_k)}, & |S| = 1; \\ \min(\tau_a - 2l_{m(p_k)}, t(p_k) - l_{m(p_k)}), & |S| \geq 2. \end{cases} \quad (7)$$

其中 $a \in S \setminus \{k\}$, 表示 $S \setminus \{k\}$ 为最佳排序时倒数第 $|S| - 1$ 次配送的料箱标号. 完成 $|B|$ 个阶段的决策之后可获得该模型的最优目标函数值, 通过反向搜索可获得最优的料箱配送序列及发车时间.

为了有效说明BDP算法的求解过程, 给出如下实例分析. 其中, 工位集合为 $M = \{1, 2\}$, 料箱集合为 $B = \{1, 2, 3, 4\}$. 表1给出了算例的工位参数, 包括超市与各工位间的物料补给时间 l_m 和各物料线边库存成本系数 β_m . 表2给出了料箱的参数, 包括各料箱的目标工位 $m(b)$ 、交货期 $t(b)$ 和料箱中的物料量 $n(b)$. BDP算法的求解过程如图2所示.

表1 算例的工位参数

| m | l_m | β_m |
|-----|-------|-----------|
| 1 | 1.02 | 0.73 |
| 2 | 0.64 | 0.59 |

表2 算例的料箱参数

| b | $m(b)$ | $n(b)$ | $t(b)$ |
|-----|--------|--------|--------|
| 1 | 1 | 2 | 0 |
| 2 | 1 | 2 | 2 |
| 3 | 2 | 2 | 1 |
| 4 | 2 | 2 | 3 |

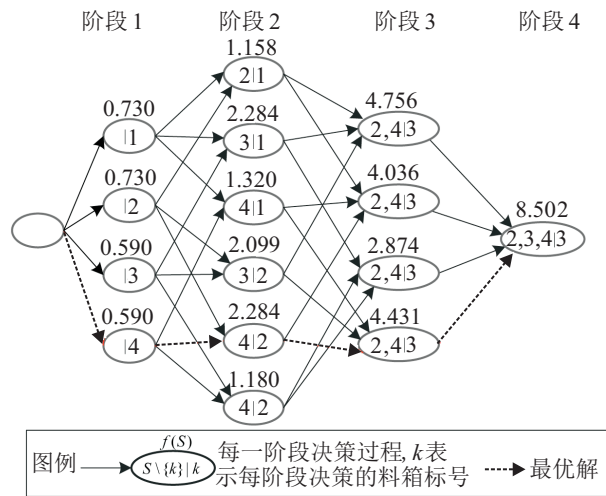


图2 BDP算法求解过程

由图2可知, 该算例的决策过程分为4个阶段, 第1阶段决策第4次配送作业的料箱标号, 第2阶段决策第3次配送作业的料箱标号, 依次类推. 完成4个阶段的决策之后获得最优目标函数值, 即为8.502. 通过反向搜索可获得最优决策序列和发车时间序列分别为 $\{4, 2, 3, 1\}$ 和 $\{2.36, 0.32, -0.96, -3.00\}$, 故最优解为 $\{(1, -3.00), (3, -0.96), (2, 0.32), (4, 2.36)\}$.

2.2 BDP算法时间复杂度分析

给定料箱集合 B , 则该问题的解空间为 $|B|!$. 由式(6)和(7)可知, 计算每个料箱产生的库存成本费用耗时 $O(n(b))$. 令 $\ell = \max\{n(b) | b \in B\}$, 利用全枚举方法获得所有解的库存费耗时 $O(h(\ell, B))$, 其中 $h(\ell, B) = \ell(|B| - 1) \cdot |B|!$. 完成全枚举后, 利用快速排序法获得全局最优解的时间复杂度为 $O(h(\ell, B) \log_2 h(\ell, B))$.

若采用BDP算法来获得该问题的全局最优解, 则完成各阶段决策以获得最小库存费用的时间复杂度为 $O(\mu(\ell, B))$, 其中 $\mu(\ell, B) = \ell \sum_{k=1}^{|B|} k C_{|B|}^k = \ell |B| 2^{|B|}$. 完成 $|B|$ 阶段决策后, 反向搜索获得最优序列的计算复杂度为 $O(|B|^2)$.

由此可知,相比于全枚举搜索,BDP算法能够从阶乘级别的计算复杂度降低到指数级别的计算复杂度,有效地减少了计算冗余,得到小规模问题的精确解.然而,随着问题规模($|B|$)的扩大,BDP算法所需计算时间急剧增加,降低了算法的执行效率.为此需构建启发式算法以求解中大规模问题.

3 改进蜂群算法

人工蜂群算法(ABC)凭借其控制参数少、收敛速度快、求解精度高等优点,已经成功运用于数值优化^[10]、流水车间调度^[11]等方面的研究.然而,ABC算法存在早熟收敛、易陷入局部最优等缺陷.针对以上问题,学者们构建了多种改进策略以提升ABC算法的收敛性能.尽管如此,如何在提高ABC算法搜索能力的同时避免陷入局部最优仍然是一个难题.

为此,本文构建了求解中大规模物料补给问题的改进蜂群算法(MABC).MABC算法在邻域搜索部分融合了分布估计算法^[12](EDA)的设计思想,从统计学习的角度出发,通过构造优秀解分布信息的概率模型,指导算法的进化方向,进而提升基本ABC算法的搜索性能并防止陷入局部最优.

MABC算法的核心内容包括编码及种群初始化、概率模型初始化及其更新机制、基于EDA的个体更新机制3个部分.

3.1 编码及种群初始化

给定料箱集合 B ,该问题的编码可表示为料箱标号 $1, 2, \dots, |B|$ 的一个排列表示,图3给出了调度方案 $A_1 = \{1, 3, 2, 4\}$ 、 $A_2 = \{1, 2, 3, 4\}$ 的编码示意图.

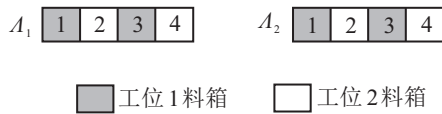


图3 编码方式

给定种群规模 U ,MABC算法通过生成 U 个随机排列构建初始种群.

3.2 概率模型初始化及其更新机制

给定料箱集合 B ,则该问题解空间分布的概率模型可表示为矩阵 $P = (p_{jk})_{|B| \times |B|}$,其中 p_{jk} 表示第 k 次配送作业的料箱编号为 j 的概率.初始化阶段, $\forall j, k \in \{1, 2, \dots, |B|\}$,令 $p_{jk} = 1/|B|$.

MABC算法采用基于群体的增量学习算法(PBIL)更新该概率模型 P .给定参数 $\xi \in (0, 1)$,令 $E = \lceil \xi|U| \rceil$,当前种群完成更新操作后,采样新种群中的优势个体.给定学习系数 ω ,利用Hebb学习规则更新概率模型 P ,公式如下:

$$p_{jk}^{g+1} = (1 - \omega)p_{jk}^g + \omega \frac{1}{E} \sum_{e=1}^E I_e(j, k). \quad (8)$$

其中: g 为算法的当前迭代次数; $I_e(j, k)$ 为示性函数,定义如下:

$$I_e(j, k) = \begin{cases} 1, & \text{个体 } e \text{ 基因座 } k \text{ 处值为料箱 } j; \\ 0, & \text{否则.} \end{cases} \quad (9)$$

3.3 基于EDA的个体更新机制

在个体 Λ 的更新阶段,给定 $\eta \in (0, 1)$,令 $\lambda = \lceil \eta|B| \rceil$,通过继承 Λ 中的后 λ 个料箱配送子序列以保留自身较好的基因.然后,结合概率矩阵 P 更新个体 Λ 中前 $|B| - \lambda$ 个基因.令 K 为个体前 $|B| - \lambda$ 个基因座上料箱标号所构成的集合,依次更新基因座1至基因座 $|B| - \lambda$ 处的料箱标号,具体步骤如下.

Step 1: 令 $j = 1$,转Step 2;

Step 2: 若 $j > |B| - \lambda$,则转Step 3,否则,转Step 5;

Step 3: 以概率 $p_{jk} / \sum_{k \in K} p_{jk}$ 生成料箱标号 k^* ($k^* \in K$),更新基因座 j 上的基因,转Step 4;

Step 4: 令 $K \leftarrow K \setminus \{k^*\}$, $j = j + 1$,转Step 2;

Step 5: 结合个体 Λ 中后 λ 个基因生成新个体 Λ' .

针对 Λ 及其变异个体 Λ' ,结合式(3)求解两调度方案的目标函数值.采用贪婪选择,保留两个体中目标函数值较小的个体进行后续搜索.

3.4 MABC算法流程

基于上述分析,给出求解该问题的MABC算法流程,具体步骤如下.

Step 1: 初始化MABC算法各项参数,包括种群规模 U 、调节参数 L 、最大迭代次数 G_1 和算法停滞代数 G_2 等,转Step 2;

Step 2: 采用随机初始化操作构建初始种群,并生成初始概率矩阵 P ,转Step 3;

Step 3: 利用基于EDA的个体更新机制优化当前种群中的每个个体,转Step 4;

Step 4: 通过轮盘赌算子选择待优化的个体,利用基于EDA的个体更新机制优化该个体,转Step 5;

Step 5: 随机生成新的个体以替换当前种群中 L 次尚未更新的个体,转Step 6;

Step 6: 选择当前种群中 E 个优势个体更新概率矩阵 P ,转Step 7;

Step 7: 若MABC算法满足终止条件(最大迭代次数或算法停滞代数),则结束循环并输出最优解,否则,转Step 3.

MABC的关键步骤包括Step 3中的雇佣蜂搜索、Step 4中基于轮盘赌算子的观察蜂搜索、Step 5中的

侦查蜂搜索和Step 6中概率模型的更新操作.

3.5 MABC算法复杂度分析

对MABC算法迭代过程中各项操作所需的时间复杂度分析如下:

- 1) 各个体的适应度值计算耗时为 $O(\ell|B|)$, 其中 $\ell = \max\{n(b)|b \in B\}$;
- 2) 利用基于EDA的个体更新机制生成新个体的时间复杂度为 $O((|B| - \lambda)^2)$;
- 3) 利用快速排序法对 U 个个体进行适应度值排序耗时为 $O(U \cdot \log_2 U)$;
- 4) 基于 E 个优势个体,更新概率矩阵 P 的时间复杂度为 $O(|B| \cdot (|B| + E))$;
- 5) 利用轮盘赌算子选择优势个体的时间复杂度为 $O(|B|)$.

据此可知,MABC算法的时间复杂度相对较低,能够有效应对中大规模问题.

4 实验与分析

引用文献[6-8]中的相关参数,设计了混流装配生产线准时化物料补给问题的仿真实验,并给出实验分析以验证本文构建算法的有效性.

4.1 算例构建

为了测试本文提出的算法性能,分别构建了小规模情形和中大规模情形的算例,步骤如下:

- 1) 令工位集合 M 为 $\{1, 2, \dots, 5\}$,小规模情形下料箱数量 $|B| \in \{10, 12, \dots, 20\}$,中大规模情形下料箱数量 $|B| \in \{30, 40, \dots, 80\}$.
- 2) $\forall m \in M$,令 $l_m = \text{rand}(0.5, 1.5)$, $\beta_m = \text{rand}(0.5, 2)$,随机生成配送时间和库存成本系数两类参数.
- 3) 将 $|B|$ 个料箱随机分配给 $|M|$ 个工位,且保证每个工位至少有一个料箱,进而获得每个料箱的目标工位值 $m(b)$.
- 4) 记工位 m 的料箱集合 B_m ,假设 $|B|$ 个料箱的物料用于装配 V 个在制品,因而 $\forall m \in M$,各工位的物料需求总量为 V ;令 $V = |B|$,将 V 个物料随机给 $|B_m|$ 个料箱,进而获得每个料箱所装的零件数量 $n(b)$.
- 5) 基于假设1)和假设2),装配序列中第 j 个在制品在工位 m 处开始装配的时间为 $T(j, m) = j + m - 2$,结合假设5),令 $B'_m = \{k|k \in B_m, k < b\}$,可知料箱 b 的交货期为

$$t(b) = T(\ell(b), m(b)) = \ell(b) + m(b) - 2, \quad (10)$$

$$\ell(b) = 1 + \sum_{k \in B'_m} n(k). \quad (11)$$

基于Matlab2013a仿真平台,在主频为2.4 GHz、内存4 GB、Intel(R) Core(TM) i5-2430M CPU的便携式计算机上进行实验,实验结果和分析如下.

4.2 BDP算法测试结果分析

考虑到反向动态规划算法具有指数级别的时间复杂度,故利用该算法求解小规模算例.BDP算法的运行时限设置为1800 s,测试结果如表3所示.其中, $|B|$ 、 C^* 和CPU分别表示各算例的问题规模、最优目标函数值和算法运行时间.此外,符号—表示BDP算法无法在规定时间内获得问题的最优解.

表3 BDP算法试验结果

| 算例 | $ B $ | C^* | CPU/s |
|----|-------|--------|--------|
| 1 | 10 | 167.8 | 0.35 |
| 2 | 12 | 233.7 | 1.87 |
| 3 | 14 | 288.23 | 11.6 |
| 4 | 16 | 351.68 | 68.36 |
| 5 | 18 | 433.96 | 654.88 |
| 6 | 20 | — | — |

由表3可知,当问题规模不超过18时,BDP算法能够在规定的时限内获得问题的最优解;同时,算法运行时间随着问题的规模增加急剧上升,当问题规模为18时,BDP算法运行时间达到654.88 s.此外,当问题规模达到20时,BDP算法无法在规定的时限(1800 s)内获得问题的最优解.因而,有必要构建求解该问题的启发式方法.

4.3 MABC参数设置分析

MABC算法包含3个重要参数,即优势个体参数 ξ 、继承长度参数 η 和学习速率参数 ω .为了获得MABC算法的最优搜索性能,设计了三因素、四水平的正交实验,选择 $L_{16}(4^3)$ 的正交表,实验参数配置如表4所示.

表4 实验参数配置

| 水平 | 因素 | | |
|----|-------|--------|----------|
| | ξ | η | ω |
| 1 | 0.2 | 0.2 | 0.1 |
| 2 | 0.3 | 0.3 | 0.2 |
| 3 | 0.4 | 0.4 | 0.3 |
| 4 | 0.5 | 0.5 | 0.4 |

如表5所示,给出了正交实验方案.测试算例的问题规模设置为20,MABC算法运行20次,选用优化结果的均值(\bar{C})作为评价指标.MABC算法其他参数设置如下:种群规模 $U = 30$,调节参数 $L = 50$,最大迭代次数 $G_1 = 800$,算法停止代数 $G_2 = 200$,表6给出了仿真结果的统计分析.

表5 正交实验方案及仿真结果

| 实验号 | 因素 | | | \bar{C} |
|-----|-------|--------|----------|-----------|
| | ξ | η | ω | |
| 1 | 1 | 1 | 1 | 543.34 |
| 2 | 1 | 2 | 2 | 538.62 |
| 3 | 1 | 3 | 3 | 510.75 |
| 4 | 1 | 4 | 4 | 520.04 |
| 5 | 2 | 1 | 2 | 519.99 |
| 6 | 2 | 2 | 1 | 545.73 |
| 7 | 2 | 3 | 4 | 539.38 |
| 8 | 2 | 4 | 3 | 517.08 |
| 9 | 3 | 1 | 3 | 518.81 |
| 10 | 3 | 2 | 4 | 513.45 |
| 11 | 3 | 3 | 1 | 536.58 |
| 12 | 3 | 4 | 2 | 542.18 |
| 13 | 4 | 1 | 4 | 518.35 |
| 14 | 4 | 2 | 3 | 540.3 |
| 15 | 4 | 3 | 2 | 545.56 |
| 16 | 4 | 4 | 1 | 541.81 |

表6 仿真结果统计分析

| 水平 | 因素 | | |
|------|--------|--------|----------|
| | ξ | η | ω |
| 1 | 528.19 | 525.12 | 541.86 |
| 2 | 530.54 | 534.52 | 536.59 |
| 3 | 527.75 | 533.07 | 521.74 |
| 4 | 536.5 | 530.28 | 522.8 |
| 极差 | 8.75 | 9.4 | 20.13 |
| 主次排序 | 3 | 2 | 1 |
| 最优组合 | 3 | 1 | 3 |

由表6可知,参数 ω 对MABC算法搜索性能的影响最大.其次,继承长度参数 η 对MABC算法的搜索性能具有重要影响. η 值过大将导致种群相似度过大,反之,则使得算法无法有效利用前一代种群的优秀基因.此外,优势个体参数 ξ 对算法收敛性能的影响相对较小,但仍需要进行合理设置,以获得最佳的寻优能力.

综上所述,MABC算法的3个参数设置如下:优势个体参数 $\xi = 0.4$,学习速率参数 $\omega = 0.2$,继承长度参数 $\eta = 0.2$.

4.4 MABC测试结果分析

首先,给出MABC算法求解小规模算例的仿真结果分析.针对每个算例,MABC算法运行20次,实验结果如表7所示.

表7 小规模算例实验结果

| 算例 | $ B $ | \bar{C} | Δ | σ | CPU/s |
|----|-------|-----------|----------|----------|-------|
| 1 | 10 | 167.8 | 0 | 0 | 2.79 |
| 2 | 12 | 234.05 | 0.35 | 0.85 | 2.87 |
| 3 | 14 | 288.94 | 0.71 | 1.24 | 2.97 |
| 4 | 16 | 352.52 | 0.84 | 1.87 | 3.23 |
| 5 | 18 | 435.13 | 1.17 | 2.80 | 3.81 |

由表7可知,MABC算法能够有效解决小规模问题.各算例优化结果的均值(\bar{C})与最优值(表3中的 C^* 值)的绝对偏差(Δ 值)在区间 $[0, 1.17]$ 内,各算例的优化结果的标准差(σ 值)在区间 $[0, 2.80]$ 内.与此同时,MABC算法的运行时间相对较短,且随着问题规模的扩大,MABC算法运行耗时增长较为缓慢.当问题规模为18时,MABC算法的平均运行时间为3.81 s,而BDP算法的运行时间为654.88 s.

为了进一步验证MABC算法搜索能力,利用MABC算法求解中大规模算例,并将优化的结果与文献[13]、文献[14]中的两种离散蜂群算法(ABC(M)和ABC(A))进行比较.ABC(M)和ABC(A)算法的参数设置及收敛条件与MABC算法相同,3种算法优化结果的均值(\bar{C} 值)和标准差(σ 值)如表8所示.

表8 中大规模算例实验结果

| 算例 | $ B $ | 算法 | \bar{C} | σ |
|----|-------|--------|-----------|----------|
| 1 | 30 | ABC(M) | 1 166.83 | 28.45 |
| | | ABC(A) | 1 193.6 | 20.3 |
| | | MABC | 1 126.58 | 14.16 |
| 2 | 40 | ABC(M) | 2 138.13 | 32.9 |
| | | ABC(A) | 2 188.56 | 37.86 |
| | | MABC | 2 052.64 | 25.73 |
| 3 | 50 | ABC(M) | 3 404.13 | 57.96 |
| | | ABC(A) | 3 487.44 | 43.04 |
| | | MABC | 3 234.03 | 28.47 |
| 4 | 60 | ABC(M) | 5 175.04 | 61.85 |
| | | ABC(A) | 5 363.87 | 59.39 |
| | | MABC | 4 884.53 | 36.09 |
| 5 | 70 | ABC(M) | 6 660.5 | 85.28 |
| | | ABC(A) | 6 934.88 | 71.16 |
| | | MABC | 6 267.68 | 42.21 |
| 6 | 80 | ABC(M) | 8 117.64 | 106.7 |
| | | ABC(A) | 8 547.88 | 90.3 |
| | | MABC | 7 619.02 | 52.16 |

由表8可知,与ABC(M)和ABC(A)算法的优化结果相比,MABC算法的表现相对较好,寻优性能更为稳定.基于EDA的个体更新机制有效地强化了ABC算法的搜索深度,避免陷入局部最优.

此外,给出了MABC算法与ABC(A)算法在求解中大规模算例时的算法运行时间曲线,见图4.

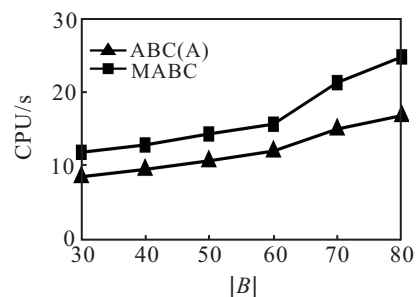


图4 运行时间曲线

由于MABC算法需要利用优势个体构建解空间的概率模型,MABC算法运行的时间较ABC(A)算法多,图4的算法运行时间曲线佐证了这一点.但总体而言,MABC算法的运行时间与ABC(A)算法相近,能够在较短的时间内解决中大规模准时化物料补给问题,且解的质量较优.

5 结 论

1) 研究准时化顺序供应的混流装配生产线物料补给问题,以降低物料的线边库存成本为目标进行了数学规划建模与分析,并将原先复杂的混合优化问题转换为决策料箱最优配送序列的组合优化问题.

2) 构建反向动态规划算法以获得小规模问题的全局最优解,并证明了该算法具有指数级别时间复杂度.同时构建了改进蜂群算法以求解中大规模问题,该改进算法通过融合基于EDA的个体更新机制有效提升了基本ABC算法的寻优能力.

综上所述,通过合理安排物料补给作业能够降低线边物料的库存费用,实现混流装配线的准时化物料供应,并保障生产作业的顺利运转.后续研究可以考虑带生产线停机等不确定因素的动态物料补给问题.

参考文献(References)

- [1] 彭涛,周炳海. 车辆装配线线边物料准时化配送算法[J]. 控制理论与应用, 2016, 33(6): 779-786.
(Peng T, Zhou B H. Just-in-time distribution algorithm of line-side parts for automobile assembly lines[J]. Control Theory & Applications, 2016, 33(6): 779-786.)
- [2] Golz J, Gujjula R, Günther H O, et al. Part feeding at high-variant mixed-model assembly lines[J]. Flexible Services & Manufacturing J, 2012, 24(2): 119-141.
- [3] Choi W, Lee Y. A dynamic part-feeding system for an automotive assembly line[J]. Computers & Industrial Engineering, 2002, 43(1/2): 123-134.
- [4] Souza M C D, Carvalho C R V D, Brizon W B. Packing items to feed assembly lines[J]. European J of Operational Research, 2008, 184(2): 480-489.
- [5] Battini D, Boysen N, Emde S. Just-in-time supermarkets for part supply in the automobile industry[J]. J of Management Control, 2013, 24(2): 209-217.
- [6] Emde S, Fliedner M, Boysen N. Optimally loading tow trains for just-in-time supply of mixed-model assembly lines[J]. Iie Transactions, 2012, 44(2): 121-135.
- [7] Fathi M, Rodríguez V, Fontes D B M M, et al. A modified particle swarm optimisation algorithm to solve the part feeding problem at assembly lines[J]. Int J of Production Research, 2015, 54(3): 878-893.
- [8] Boysen N, Bock S. Scheduling just-in-time part supply for mixed-model assembly lines[J]. European J of Operational Research, 2010, 211(1): 15-25.
- [9] Boysen N, Bock S, Fliedner M. Scheduling of inventory releasing jobs to satisfy time-varying demand: An analysis of complexity[J]. J of Scheduling, 2012, 16(2): 185-198.
- [10] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm[J]. J of Global Optimization, 2007, 39(3): 459-471.
- [11] Pan Q K, Tasgetiren M F, Suganthan P N, et al. A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem[J]. Information Sciences an Int J, 2011, 181(12): 2455-2468.
- [12] 龙志强,蔡楹,徐昕. 基于分布估计算法的磁浮列车故障综合评判[J]. 控制与决策, 2009, 24(4): 551-556.
(Long Z Q, Cai Y, Xu X. Comprehensive fault evaluation on maglev train based on estimation of distribution algorithms[J]. Control and Decision, 2009, 24(4): 551-556.)
- [13] Wang Z, Guo J, Zheng M, et al. Uncertain multiobjective traveling salesman problem[J]. European J of Operational Research, 2014, 241(2): 478-489.
- [14] Alvarado-Iniesta A, Garcia-Alcaraz J L, Rodriguez-Borbon M I, et al. Optimization of the material flow in a manufacturing plant by use of artificial bee colony algorithm[J]. Expert Systems with Applications an Int J, 2013, 40(12): 4785-4790.

(责任编辑: 孙艺红)