

# 求解双边装配线平衡问题的改进蚁群算法

郑巧仙<sup>1,2</sup>, 李明<sup>3,4</sup>, 李元香<sup>2</sup>, 唐秋华<sup>4</sup>

(1. 湖北大学数学与计算机科学学院, 湖北武汉 430062; 2. 武汉大学计算机学院, 湖北武汉 430072;  
3. 武汉科技大学理学院, 湖北武汉 430065; 4. 武汉科技大学机械自动化学院, 湖北武汉 430081)

**摘要:** 结合第2类双边装配线平衡问题(TALBP-2)的特点, 提出了一种改进蚁群算法. 针对 TALBP-2 中操作间的序关系, 对分配至同一个边工位中相邻的两项操作间定义了一种新的信息素; 提出了一种新的定界策略, 根据所得的最好解, 减小工位时间的上界, 利用已分配工位的平均实际作业时间界定工位时间的下界, 缩小蚂蚁的搜索空间; 针对 TALBP-2 提出了一种新的操作分配准则, 根据所定义的3种理想操作的优先分配权限分配操作; 提出了一种边工位确定原则, 平衡两边工位时间的增加速度. 算例的比较结果表明算法有较好的求解效果和稳定性.

**关键词:** 双边装配线; 装配线平衡问题; 蚁群算法; 组合优化

**中图分类号:** TH166; TP18

**文献标识码:** A

**文章编号:** 0372-2112 (2014)05-0841-05

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.3969/j.issn.0372-2112.2014.05.002

## An Improved Ant Colony Optimization for Two-Sided Assembly Line Balancing Problem

ZHENG Qiao-xian<sup>1,2</sup>, LI Ming<sup>3,4</sup>, LI Yuan-xiang<sup>2</sup>, TANG Qiu-hua<sup>4</sup>

(1. College of Mathematic and Computer Science, Hubei University, Wuhan, Hubei 430062, China; 2. College of Computer, Wuhan University, Wuhan, Hubei 430072, China; 3. College of Science, Wuhan University of Science and Technology, Wuhan, Hubei 430065, China;  
4. College of Mechanics and Automation, Wuhan University of Science and Technology, Wuhan, Hubei 430081, China)

**Abstract:** According to the characteristics of the type 2 two-sided assembly line balancing problem, an improved ant colony optimization is proposed. A novel pheromone between two adjacent tasks in the same side station is defined to describe the order relation between them. A new bound strategy is proposed to reduce the search space of ants, by decreasing the upper bound of station times according to the current best solution, and bounding their lower bounds with the mean processing time of assigned stations. An improved task assignment rule is applied to assign the suitable task to station, in which three kinds ideal task with different prior permissions are used. A side station determination rule is proposed to balance the increase speed of both side times. Computational results show the effectiveness and stability of proposed algorithm.

**Key words:** two-sided assembly line; assembly line balancing problem; ant colony optimization; combinatorial optimization

## 1 引言

装配是生产制造过程中的最后环节, 是电子、汽车等加工业的核心制造工艺. 在装配线中, 整个产品的装配被分解为若干项操作, 装配时, 未完成产品在传送带上被连续地从一个工位传送到下一个工位, 每个工位包含几项操作. 其中操作的作业时间称为操作时间, 每个工位上操作的作业时间之和称为该工位的工位时间.

装配线中, 如果部分操作要求必须在传送带的左边完成(称为左边操作), 部分要求只能在右边作业(称为右边操作), 其余的则可以在任意一边作业(称为双边操

作), 此装配线称为双边装配线. 由于双边装配线的装配在传送带的两边同时进行, 所以每个工位又有左边和右边两个子工位, 统称为边工位. 在双边装配线中, 部分操作间存在优先关系. 操作、操作时间以及操作间的优先关系可通过优先关系图进行描述, 如图1所示. 图1中有16项操作, 各操作上方给出了该操作的操作时间及其所在工位的位置( $L$ 表示左边,  $R$ 表示右边,  $E$ 表示双边), 箭头指向描述了操作间的优先关系.

装配线平衡问题(ALBP)是指将存在优先关系约束的操作分配至预先设定的工位完成, 使得某些目标达到最优的一系列优化问题. 常见的优化目标有两类: (1) 给

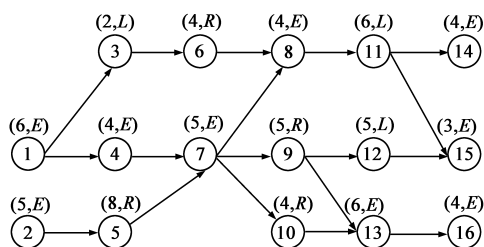


图1 双边装配线平衡问题的优先关系图

定装配线的节拍,即最大工位时间,最小化其工位个数;(2)给定装配线的工位个数,最小化其节拍。相应的双边装配线平衡问题(TALBP)分别称为第1类双边装配线平衡问题(TALBP-1)和第2类双边装配线平衡问题(TALBP-2)。

两类问题都需要满足下面约束:(1)每项操作被分配且仅分配至一个工位的某一个边工位上;(2)如果操作 $i_1$ 优先于操作 $i_2$ ,则 $i_1$ 的结束时间应不晚于 $i_2$ 的开始时间;(3)每个工位的持续时间不超过装配线的节拍;(4)左(右)边操作只能被分配至对应的边工位。TALBP-1和TALBP-2都是组合优化中的NP难问题。

TALBP由Barthodi(1993)首次提出<sup>[1]</sup>,目前已有较多方法求解TALBP-1<sup>[2~4]</sup>,但对TALBP-2的研究则相对较少。吴尔飞等(2005)提出了一种基于归组策略的启发式平衡算法<sup>[5]</sup>;Kim等(2000)对TALBP-2提出了一种基于遗传算法的平衡算法<sup>[6]</sup>,其后对其建立了混合整数规划模型,并将邻域搜索的思想融入遗传算法,提出了一种邻域遗传算法(n-GA)<sup>[7]</sup>。本文提出一种改进蚁群算法求解TALBP-2,即在给定工位数的条件下,最小化装配线的节拍。蚁群算法是模仿蚂蚁觅食行为的一种群智能进化算法,由意大利学者Dorigo(1991)首次提出<sup>[8]</sup>。该算法具有自组织性、并行性、正反馈性等优良特性,在求解组合优化问题时表现卓越<sup>[9]</sup>。与多数其它元启发式算法相比,蚁群算法的求解遍历过程与装配线的设计过程相似性较高,更能充分地利用问题自身的特性。所提算法利用三种启发式因素和所定义的两信息素帮助蚂蚁选择操作,根据所提的操作分配准则、定界策略<sup>[10]</sup>和边工位的确定原则,消减工位中的闲置时间,减小问题的可行解空间,提高解的质量。

## 2 改进蚁群算法

### 2.1 算法流程

在改进蚁群算法中,蚂蚁依次为各个工位:(1)根据启发式因素和信息素选择操作;(2)利用边工位确定原则确定待分配的边工位;(3)依据操作分配准则分配合适的操作至工位,从而得到各个工位的操作分配方案;然后根据所得的可行解,释放信息素,根据定界策

略减少工位时间的变化范围。重复上述过程直至达到最大迭代次数,输出最优解及最好节拍,算法结束,其主要流程如图2所示。

### 2.2 启发式因素

在给定的TALBP-2中,假设共有 $n$ 项操作,其集合记为 $I = \{i | i = 1, 2, \dots, n\}$ ,操作 $i$ 的操作时间记为 $t_i$ 。引入两个0-1矩阵 $P$ 和 $P^*$ 分别表示操作间的直接优先关系和优先关系,其中 $P = (P_{i_1, i_2})_{n \times n}$ ,如果操作 $i_2$ 是操作 $i_1$ 的直接优先操作,则 $P_{i_1, i_2} = 1$ ,否则为0; $P^* = (P^*_{i_1, i_2})_{n \times n}$ ,如果操作 $i_2$ 是操作 $i_1$ 的优先操作,则 $P^*_{i_1, i_2} = 1$ ,否则为0。定义当前边为 $A$ 边( $A$ 可取 $L$ 或 $R$ ,分别表示左边工位或右边工位)的候选操作集合为 $CA$ ,集合中的操作 $i$ 满足:

$$\sum_{i_1 \in I} P_{i_1, i} = \sum_{i_2 \in VI} P_{i_2, i}, i \notin VI \subset I, i \in (S_A \cup S_E) \text{ 其中 } VI \text{ 为当前的已分配操作集合; } S_A \text{ 为只能分配至 } A \text{ 边的操作集合, } S_E \text{ 为可分配至两边的操作集合。}$$

1. 初始化
2. for 遍历搜索次数  $nc = 1$  to  $NC$  do
3.   for 蚂蚁  $k = 1$  to  $s$  do
4.     for 工位  $j = 1$  to  $m$  do
5.       根据启发式因素和信息素随机选择一项操作,利用边工位确定原则确定此操作所在的边工位,并将其作为首项操作分配至此边工位。
6.       根据边工位的确定原则确定待分配的边工位,根据启发式因素和信息素随机选择一项操作,基于操作分配准则分配操作至此边工位。
7.     end for
8.     蚂蚁在操作与两个边工位之间以及满足条件的操作间分别释放信息素。
9.   end for
10. 保留最好解及其节拍,更新工位时间的上界和下界,更新信息素。
11. end for
12. 输出最好解及其最小节拍。

图2 改进蚁群算法的主要流程

算法在已有启发式因素:操作的位置权重<sup>[11]</sup>的基础上,提出了两种新的启发式因素:操作所在边的位置和闲置时间,其权值 $F_i$ 和 $Idle_i$ 分别为: $F_i = 1, i \in S_A \cap CA$ ;  $F_i = 0.5, i \in S_E \cap CA$ ,其中 $CA$ 为当前的候选操作集合。 $Idle_i = t_i^f - ST_{a,j,k}$ ,如果 $t_i^f > ST_{a,j,k}$ 且 $P^*_{i', i} = 1, i' \in \bar{S}_{a,j,k}$ ;否则为0。式中 $t_i^f$ 为操作 $i$ 的优先操作 $i'$ 的作业结束时间; $ST_{a,j,k}$ 为蚂蚁 $k$ 在工位 $j$ 的 $A$ 边工位的当前结束时间, $a$ 可取1和2,分别对应工位的左边( $A = L$ )和右边工位( $A = R$ ); $\bar{S}_{a,j,k}$ 表示操作 $i'$ 作为第 $r$ 项操作被蚂蚁 $k$ 分配至工位 $j$ 的 $\bar{A}$ 边( $\bar{A}$ 的定义与 $A$ 相反)。综合三种启发式因素得候选操作 $i$ 的选择权值为:

$$\xi_i = \frac{W_i \cdot F_i}{Idle_i} \quad (1)$$

### 2.3 信息素及其更新策略

在 TALBP-2 中,各个边工位和操作间的组合,同一边工位的操作间的前后排序都会影响问题的解,针对这两种要求,算法引入了两类信息素:(1)边工位和操作间的信息素;(2)操作和操作间的信息素.蚂蚁  $k$  对这两类信息素的贡献值分别定义如下:

$$\Delta\tau_{A_{j,i}^{(k)}} = \begin{cases} \frac{Q}{(\max_a \{\max_j \{ST_{a,j,k}\}\})^2}, i \in SA_{j,r,k} \\ 0, \text{Otherwise} \end{cases}$$

式中  $\Delta\tau_{A_{j,i}^{(k)}}$  为蚂蚁  $k$  在操作  $i$  和工位  $j$  的  $A$  边工位间所释放的信息素贡献值,  $Q$  为参数.

$$\Delta\tau_{i_1,i_2}^{(k)} = \begin{cases} \frac{Q}{(\max_a \{\max_j \{ST_{a,j,k}\}\})^2}, i_1 = SA_{j,r,k}, i_2 = SA_{j,r+1,k} \\ 0, \text{Otherwise} \end{cases}$$

其中  $\Delta\tau_{i_1,i_2}^{(k)}$  为蚂蚁  $k$  在操作  $i_1$  和操作  $i_2$  间释放的信息素贡献值.

两类信息素都按下面更新策略进行更新:

$$\tau \leftarrow (1 - \rho) \cdot \tau + \rho \cdot \sum_{k=1}^s \Delta\tau^{(k)} \quad (2)$$

式中  $\rho$  为信息素的挥发系数,满足  $0 \leq \rho \leq 1$ ;  $s$  为蚂蚁的最大个数.

算法中,边工位和操作间的信息素只用于选择工位  $j$  的首项操作,而操作间的信息素则用于为已分配操作  $i_1$  选择其同一边工位中的后继相邻操作.

基于启发式因素和信息素,蚂蚁按式(3)为当前工位以概率选择操作.

$$P_i = \frac{(\xi_i)^\alpha \cdot (\tau_i)^\beta}{\sum_{i \in CA} (\xi_i)^\alpha \cdot (\tau_i)^\beta}, i \in CA \quad (3)$$

式中  $\alpha, \beta$  为参数,分别表示启发式权值  $\xi_i$  和信息素值  $\tau_i$  在操作选择概率中的重要程度,其初始值都取一个较小的常数.

### 2.4 边工位的确定原则

在 TALBP-2 中,装配需在工位的双边同步进行,两边的操作分配互相影响,所以在对工位进行操作分配前应先确定待分配的边工位.由于首项操作和其它操作在分配前的工位时间状况不同,所以针对这两类操作,算法采取了不同的原则确定其边工位.

(1) 首项操作 由于当前工位双边的作业时间都为 0,且无法确定从哪边开始分配所得的方案更优.所以算法首先根据启发式因素和信息素值计算  $CA' = (CA\_L, CA\_R)$  中各候选操作的选择权值,然后利用轮盘赌的方法随机选择一项操作,根据此操作在数组  $CA'$  中的位置确定其所分配的边工位.

(2) 其它操作 结束首项操作分配后,算法首先按照下面准则确定待分配的边工位,然后为此边工位选择操作,最后利用 2.6 节的操作分配准则分配合适的操作至此边工位.具体准则如下:

记蚂蚁  $k$  所分配操作的工位为  $j$ ,双边的工位结束时间分别为  $ST_{1,j,k}$  和  $ST_{2,j,k}$ ,左右两边的候选操作集合分别为  $CA\_L$  和  $CA\_R$ ,当前的工位时间上界为  $ST_{\max}$ ,其取值见(2.5)节.则

(a) 如果  $((CA\_R = \phi) \vee (ST_{1,j,k} < ST_{2,j,k})) \& (ST_{1,j,k} \leq ST_{\max}) \& (CA\_L \neq \phi)$ ,对左边工位进行分配,其中  $\phi$  表示空集.

(b) 如果  $((CA\_L = \phi) \vee (ST_{1,j,k} > ST_{2,j,k})) \& (ST_{2,j,k} \leq ST_{\max}) \& (CA\_R \neq \phi)$ ,对右边工位进行分配.

(c) 如果  $(ST_{1,j,k} = ST_{2,j,k}) \& (ST_{1,j,k} \leq ST_{\max}) \& (CA\_R \neq \phi) \& (CA\_L \neq \phi)$ ,则随机选择一个边工位进行分配.

(d) 否则,终止对该工位的操作分配.

### 2.5 定界策略

为了得到平衡率较高的解,算法在操作分配准则中引入工位时间区间  $[ST_{\min}, ST_{\max}]$ ,界定各工位的工位时间,即要求各边工位的工位时间满足:

$$ST_{\min} \leq ST_{a,j,k} \leq ST_{\max}, a = 1, 2; \\ j = 1, 2, \dots, m; k = 1, 2, \dots, s \quad (4)$$

式中  $m$  为给定的工位个数.

初始化工位时间的上界和下界分别为:  $ST_{\max} = Cm + \Delta t$  和  $ST_{\min} = Cm - \frac{\Delta t}{2}$ ,其中  $Cm = \frac{1}{2m} \sum_{i \in I} t_i$ ,为平均工位时间,  $\Delta t$  为参数.

得到更好的解后,算法根据其节拍按照下面两式分别对上界和下界进行更新.

$$ST_{\max} = \begin{cases} ST_{\max}^{mc} - \delta, ST_{\max}^{mc} < ST_{\max} \\ ST_{\max}^{mc}, \text{Otherwise} \end{cases}$$

$$ST_{\min} = \begin{cases} Cm - (ST_{\max}^{mc} - Cm), ST_{\max}^{mc} < ST_{\max} \\ ST_{\min}^{mc}, \text{Otherwise} \end{cases}$$

式中,  $ST_{\max}^{mc}$  为第  $mc$  次遍历所得的最好节拍;  $\delta$  为工位时间上界的减小步长,随着搜索次数增加,其值根据式(5)进行调整.

$$\delta \leftarrow \left\lfloor \frac{NC - nc + 1}{NC} \delta \right\rfloor \quad (5)$$

式中  $NC, nc$  分别为算法最大搜索次数和当前搜索次数.

### 2.6 操作分配准则

假设蚂蚁  $k$  为工位  $j$  的  $A$  边工位所选择的候选操作为  $i$ ,此边工位的当前结束时间为  $ST_{a,j,k}$ ,记已分配至  $A$  边的所有工位  $1, \dots, j$  的操作的集合为  $VI\_A$ ,当前候

选操作集合记为  $CA\_A$ ,集合中各候选操作所对应的闲置时间组成集合  $Idle$ .

如果候选操作同时满足:

$$\sum_{i^* \in V_{-A}} t_{i^*} + Idle_{i'} + t_{i'} \geq j \cdot \frac{ST_{\min} + ST_{\max}}{2}, i' \in CA\_A \quad (6)$$
$$ST_{a,j,k} + Idle_{i'} + t_{i'} \geq ST_{\min}, i' \in CA\_A \quad (7)$$
$$ST_{a,j,k} + Idle_{i'} + t_{i'} \leq ST_{\max}, i' \in CA\_A \quad (8)$$
$$Idle_{i'} = 0, i' \in CA\_A \quad (9)$$

则称其为理想操作 1,其集合记为  $CA\_A1$ .如果同时满足式(6),式(8)和式(9),则称该候选操作为理想操作 2,其集合记为  $CA\_A2$ .如果同时满足式(7),式(8)和式(9),则称该候选操作为理想操作 3,其集合记为  $CA\_A3$ .

规定三种理想操作的分配优先级依次为理想操作 1>理想操作 2>理想操作 3,则设置操作分配准则如下:

如果蚂蚁  $k$  所选择的操作  $i$  满足式(8),则根据分配优先级确定一类理想操作集合,并基于启发式权值随机从中分配一项至工位  $j$ ;如果三种理想操作都不存在,则将所选择的操作  $i$  直接分配至工位.

如果所选择的操作  $i$  不满足式(8),则首先判断上述三种理想操作是否存在,如果存在按上述方法分配一项操作至工位;否则判断是否存在候选操作满足式(8),如果存在,根据选择权值随机分配一项至工位;否则终止对此工位的操作分配.

### 3 算例

算法首先以 P205 中  $m = 10$  为例,参考相关研究<sup>[12]</sup>,对三个主要参数  $\alpha, \beta, \rho$  进行参数分析,所有实验都在一台配置为:AMD Sempron LE-1250 processor,512 MB of Ram 的个人电脑上基于 MATLAB 软件进行.对参数  $\alpha, \beta$  分别取值为 1,2,3,4,5,其它参数的取值依次为: $s = 500, NC = 200, Q = \left( Cm + \frac{(Cm + \Delta t)}{2} \right)^2, \Delta t = 100, \rho = 0.5, \delta = 10$ .计算 20 次求解结果的平均节拍、最好节拍、最差节拍及均方差,比较得两参数的最佳取值为  $\alpha = 2, \beta = 1$ .对参数  $\rho$ ,依次取 0.1,0.2,⋯,0.9,分别计算 20 次,比较其结果得其最佳取值为  $\rho = 0.4$ .

为了检验改进蚁群算法求解 TALBP-2 的有效性和稳定性,对三个较大型的经典案例 P65, P148 和 P205 的共 23 组算例分别进行求解.为了便于和已有算法的求解结果进行比较,算法对案例 148,采用了与文献[6]相同的做法,即将案例中较大操作时间的 2.81 和 3.83 分别改为 1.11 和 0.43,然后基于新数据对算例进行求解.

算法在同样的环境下针对 23 组算例,分别运行 20 次,运行至 3600s 时结束运算.算法中各参数的取值见表 1(其它参数见前面数值实验的最佳取值).

| 表 1 改进蚁群算法中的参数取值 |     |      |      |      |      |      |            |      |          |    |
|------------------|-----|------|------|------|------|------|------------|------|----------|----|
| 参数               | $s$ |      |      | $NC$ |      |      | $\Delta t$ |      | $\delta$ |    |
|                  | P65 | P148 | P205 | P65  | P148 | P205 | P65        | P148 | P205     |    |
| 取值               | 100 | 200  | 500  | 50   | 50   | 200  | 20         | 50   | 100      | 10 |

20 次计算所得的最好节拍,节拍平均值以及节拍均方差见表 2(第 7~9 列).

| 表 2 蚁群算法与 n-GA 的结果比较 |     |                |          |         |         |          |         |         |                |
|----------------------|-----|----------------|----------|---------|---------|----------|---------|---------|----------------|
| 案例                   | $m$ | 平均<br>工位<br>时间 | n-GA     |         |         | ACO      |         |         | 提高<br>率<br>(%) |
|                      |     |                | 最好<br>节拍 | 平均<br>值 | 均方<br>差 | 最好<br>节拍 | 平均<br>值 | 均方<br>差 |                |
| P65                  | 4   | 637.4          | 641      | 643.7   | 0.39    | 640      | 640.7   | 0.48    | 0.47           |
|                      | 5   | 509.9          | 515      | 518.4   | 0.40    | 512      | 512.7   | 0.48    | 1.10           |
|                      | 6   | 424.9          | 432      | 434.8   | 0.52    | 429      | 429.3   | 0.67    | 1.26           |
|                      | 7   | 364.2          | 372      | 377.7   | 0.83    | 368      | 370.1   | 0.99    | 2.01           |
| P148                 | 8   | 318.7          | 327      | 334.2   | 0.90    | 322      | 323.6   | 0.84    | 13.17          |
|                      | 4   | 640.5          | 641      | 642.0   | 0.17    | 641      | 641.0   | 0.00    | 0.16           |
|                      | 5   | 512.4          | 514      | 514.9   | 0.18    | 513      | 513.0   | 0.00    | 0.70           |
|                      | 6   | 427.0          | 428      | 430.6   | 0.27    | 427      | 427.6   | 0.52    | 0.70           |
| P205                 | 7   | 366.0          | 368      | 369.2   | 0.21    | 367      | 367.0   | 0.00    | 0.60           |
|                      | 8   | 320.3          | 323      | 323.7   | 0.15    | 321      | 321.0   | 0.00    | 0.83           |
|                      | 9   | 284.7          | 287      | 289.6   | 0.53    | 286      | 286.0   | 0.00    | 1.24           |
|                      | 10  | 256.2          | 259      | 262.4   | 0.51    | 257      | 257.0   | 0.00    | 2.06           |
| P205                 | 11  | 232.9          | 237      | 238.5   | 0.42    | 234      | 234.0   | 0.00    | 1.89           |
|                      | 12  | 213.5          | 218      | 221.3   | 0.60    | 215      | 215.0   | 0.00    | 2.85           |
|                      | 6   | 1945.4         | 1984     | 2004.3  | 5.36    | 1978     | 1980.8  | 1.52    | 1.17           |
|                      | 7   | 1667.5         | 1709     | 1723.7  | 2.99    | 1711     | 1717.5  | 3.58    | 0.36           |
| P205                 | 8   | 1459.1         | 1507     | 1546.5  | 9.23    | 1493     | 1505.1  | 9.05    | 2.68           |
|                      | 9   | 1296.9         | 1337     | 1362.7  | 8.84    | 1337     | 1345.4  | 2.39    | 1.27           |
|                      | 10  | 1167.3         | 1189     | 1221.0  | 5.57    | 1204     | 1207.7  | 1.87    | 1.09           |
|                      | 11  | 1061.1         | 1095     | 1122.5  | 7.95    | 1091     | 1098.1  | 2.63    | 2.18           |
| P205                 | 12  | 972.7          | 1039     | 1061.1  | 3.45    | 1005     | 1014.1  | 2.59    | 4.42           |
|                      | 13  | 897.9          | 944      | 975.8   | 4.73    | 944      | 944.0   | 0.00    | 3.26           |
|                      | 14  | 833.8          | 944      | 953.3   | 1.33    | 944      | 944.0   | 0.00    | 0.98           |

表 2 中,第 1,2 列给出了算例的来源和设定的工位数;第 3 列给出了各算例的平均工位时间,此值可用于粗略检验所求节拍的优化程度;第 4~6 列给出了求解这三个案例所得到的目前最小节拍,其 20 次运行所得节拍的平均值及其均方差<sup>[7]</sup>;最后 1 列给出了蚁群算法所得节拍平均值相对于已有最好节拍平均值的提高率,提高率 =  $100 \times (\text{已有最好节拍平均值} - \text{蚁群算法节拍平均值最好节拍}) / \text{已有最好节拍平均值}$ .由表 2 的统计结果可知,在 23 个算例中,所提蚁群算法对其中的

17 个算例求得的最好节拍优于 n-GA 算法;4 个算例的结果相同,只有 2 个算例差于 n-GA 算法;而所得节拍的平均值,本文所提算法全部优于 n-GA 算法,其提高率最高为 4.42%,最低为 0.16%,平均为 1.57%。由上述统计结果可知所提算法的求解效率优于目前最好求解算法 n-GA 算法。就算法的稳定性而言,所提算法对 23 个算例 20 次运算所得节拍均方差的平均值为 1.20,最大值为 9.05,最小值为 0.00;而 n-GA 算法所得节拍均方差的平均值高达 2.41,最大值为 9.23,最小值为 0.15,显然所提蚁群算法在稳定性上明显优于 n-GA 算法。

## 4 小结

TALBP-2 为集分配和排序为一体的组合优化问题。求解 TALBP-2,需按操作方位的要求在满足操作优先关系的情况下,将操作分配至各个工位的某边工位上。由于优先约束会使得同一操作分配方案在不同的排序下所得的工位持续时间不尽相同,所以在向各个边工位分配操作的同时也应一并考虑两个边工位中操作间的排序问题。本文针对 TALBP-2 的上述特点和要求,提出了一种改进的蚁群算法。算法引入了启发式选择、操作分配准则、两种信息素、边工位确定原则、定界策略等多种技术手段提高解的质量。算例结果验证了所提算法的有效性和稳定性。虽然算法的求解精度较高,但其时间复杂度为  $O(NC \cdot s \cdot m \cdot n^2)$ ,求解时间较长。在后续研究中,结合 TALBP 的特性,将已有的求解单边装配线平衡问题的工位蚁群算法<sup>[10]</sup>加以改进应用于求解 TALBP-2,进一步提高算法的求解效果,同时对本算法的各个环节进一步改进,以降低算法的运行时间。

## 参考文献

- [1] J J Barthodi. Balancing two-sided assembly lines: A case study [J]. International Journal of Production Research, 1993, 31 (10): 2447 - 2461.
- [2] W Erfei, J Ye, B Jinsong et al. A branch-and-bound algorithm for two-sided assembly line balancing [J]. Int J Adv Manuf Technol, 2008, 39(9 - 10): 1009 - 1015.
- [3] H Xiaofeng, et al. A branch-and-bound algorithm to minimize the line length of a two-sided assembly line [J]. European Journal of Operational Research, 2010, 206(3): 703 - 707.
- [4] U Özcan, B. Toklu. A tabu search algorithm for two-sided assembly line balancing [J]. Int J Adv Manuf Technol, 2009, 43(7 - 8): 822 - 829.
- [5] 吴尔飞, 金烨, 汪峥. 双边装配线第二类平衡问题研究 [J]. 计算机集成制造系统, 2005, 11(11): 1604 - 1608.  
WU Er-fei, et al. Research on balancing problem of type 2 of two-sided assembly line [J]. Computer Integrated Manufactur-

ing Systems, 2005, 11(11): 1604 - 1608. (in Chinese)

- [6] Y K Kim, Y Kim, Y J Kim. Two-sided assembly line balancing: a genetic algorithm approach [J]. Production Planning & Control, 2000, 11(1): 44 - 53.
- [7] Y K Kim, W S Song, J H Kim. A mathematical model and a genetic algorithm for two-sided assembly line balancing [J]. Computers & Operations Research, 2009, 36(3): 853 - 865.
- [8] A. Colomi, M. Dorigo and V. Maniezzo. Distributed optimization by ant colonies [A]. Proceedings of ECAL91-European Conference on Artificial Life [C]. Paris Publishing Elsevier, 1991. 134 - 142.
- [9] 孙伟峰, 等. QIACO: 一种多 Qos 约束网络任务调度算法 [J]. 电子学报, 2011, 39(5): 1115 - 1120.  
SUN Wei-feng, et al. QIACO: An algorithm for grid task scheduling of multiple Qos dimensions [J]. Acta Electronica Sinica, 2011, 39(5): 1115 - 1120. (in Chinese)
- [10] Z Qiaoxian, L Ming, L Yuanxiang, T Qiuhua. Station ant colony optimization for the type 2 assembly line balancing problem [J]. Int J Adv Manuf Technol, 2013, 66: 1859 - 1870.
- [11] W C Chiang. The application of a tabu search metaheuristic to the assembly line balancing problem [J]. Annals of Operations Research, 1998, 77(0): 209 - 227.
- [12] 吴春明, 陈治, 姜明. 蚁群算法中系统初始化及系统参数的研究 [J]. 电子学报, 2006, 34(8): 1530 - 1533.  
WU Chun-ming, CHEN Zhi, JIANG Ming. The research on initialization of ants system and configuration of parameters for different TSP problems in ant algorithm [J]. Acta Electronica Sinica, 2006, 34(8): 1530 - 1533. (in Chinese)

## 作者简介



郑巧仙 女, 1978 年生于浙江省衢州市, 2013 年于武汉大学计算机学院获得工学博士学位, 现为湖北大学数学与计算机科学学院讲师, 主要研究方向为智能计算。  
E-mail: zqxlm1978@163.com



李明 男, 1976 年生于山东省潍坊市, 武汉科技大学机械与自动化学院在读博士研究生, 现为武汉科技大学理学院副教授, 主要研究方向为生产调度优化。  
E-mail: lmzqx@163.com