DOI：10.13196/j.cims.2007.08.178.zhangzq.022

文章编号: 1006 – 5911(2007) 08 – 1632 – 07

求解装配线平衡问题的一种改进蚁群算法

张则强, 程文明, 钟 斌, 王金诺
(西南交通大学 机械工程学院, 四川 成都 610031)

摘 要: 为求解给定节拍最小化工作站数的第 iv 类装配线平衡问题, 提出了一种改进的蚁群算法。在该算法中, 针对装配线平衡问题的具体特点, 给出了蚂蚁分配方案的生成策略。通过在任务和任务分配序列的位置之间释放信息素, 并采用信息素总合规则进行更有效的信息素累积。为提高搜索效率, 以综合考虑装配任务作业时间和后续任务数的分级位置权重为蚁群算法的启发式信息。最后, 通过对大量测试问题集的验证, 说明了算法的有效性。

关键词: 装配线平衡; 蚁群算法; 启发式方法
中图分类号: TH165; TP301. 6 文献标识码: A

Improved ant colony optimization for assembly line balancing problem

ZHANG Ze-qiang, CHENG Wen-ming, ZHONG Bin, WANG Jin-nuo
(School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China)

Abstract: To solve type 1 of the Simple Assembly Line Balancing Problem (SALBP-1) , which minimized the number of stations for a given cycle time, an improved ant colony optimization was proposed. According to the characteristics of the SALBP-1, the method of dynamic generation of the tour network of ants was developed. The proposed algorithm made use of the trail information which was deposited between the task and the task selected position, and pheromone summation rules was adopted. The heuristic information was set to the position weight for tasks of SALBP-1, which considered processing time of the task and the number of successors. Finally, the proposed algorithm was tested and compared with literature test problems, and the result indicated the effectiveness of the proposed algorithms.

Key words: assembly line balancing; ant colony optimization; heuristic method

0 引言

装配线平衡问题(Assembly Line Balancing Problem, ALBP) 是制造领域的一个重要问题, 直接关系到设施利用率和生产效率。ALBP 也是一个典型的 NP-hard 组合优化问题[1], 问题复杂度随任务数的增加呈几何级数增长, 难以在合理的时间完全解决问题。因此, 该问题在工业和学术上都引起了广泛的关注。

由于精确算法(如线性规划法、整数规划法、分枝界限法等[2]) 在计算时间和计算规模上的局限性, 当前主要采用启发式方法来解决 ALBP[3], 如位置权重法、凯尔布里奇和韦斯特方法、Moodie-Young 的启发式方法、霍夫曼的矩阵法、紧后更新首次适应

收稿日期: 2006 – 09 – 01; 修订日期: 2006 – 11 – 14。Received 01 Sep. 2006; accepted 14 Nov. 2006.
基金项目: 国家 863/CIMS 主题资助项目(2003AA4Z1100) ; 四川省科技攻关计划资助项目(2006Z08 – 037) 。**Foundation item:** Project supported by the National High-Tech. R&D Program for CIMS, China (No. 2003AA4Z1100), and the Science and Technique Foundation of Sichuan Province, China (No. 2006Z08 – 037).

作者简介: 张则强(1978 –) , 男, 浙江东阳人, 西南交通大学机械工程学院讲师, 博士, 主要从事物流系统、智能优化等的研究。
E-mail: zhangzeqiang@gmail.com。

(Immediate Update First-Fit, IUFF) 方法和分级指派法等。基于简单启发式规则的传统算法仅按某种规则求得一种可行方案, 由于 ALBP 的结构特征, 如任务间优先顺序关系、作业时间长短分布等的差异, 至今尚无某一优先规则确保对该类所有问题都能够求得最优解, 因而难以提供具有普适性的实用算法。随着智能优化算法的发展, 遗传算法、禁忌搜索和模拟退火法等亚启发式方法在 ALBP 中得以应用[4-8], 并表现出良好的性能。

由模仿蚂蚁觅食行为发展起来的蚁群算法, 在旅行商问题(Traveling Saleman Problem, TSP)、二次分配问题、单机调度问题、车辆路径问题等研究中得到了广泛应用[9-12], 在求解组合优化问题中也表现出卓越的性能, 特别适合大规模启发式搜索, ALBP 的结构适合采用蚁群算法求解。

当前已有蚁群算法用于 ALBP 的求解, 文献[13]研究了在给定节拍下最小化工作站数的 ALBP 的一种蚁群算法, 但其信息素累积方式过于简化, 以至对于许多测试问题不能找到最优解, 还有待进一步改善。文献[14]提出了一种基于蚂蚁技术的方法, 用于求解具有并行工作站、随机作业时间等复杂因素的混合装配线平衡问题, 试验表明算法具有良好的性能, 但文中算法仅采用了信息素的累积, 没有考虑问题内含的启发式信息, 故求解效率不高。

针对现有研究的不足, 本文结合 ALBP 的特征, 提出了一种改进的蚁群算法, 该方法采用在任务和任务分配序列的位置之间释放信息素的方式, 以及信息素总合规则, 以综合考虑任务作业时间和后续任务数的位置权重作为蚁群算法的启发值。试验结果表明, 该算法能快速有效地求解 ALBP, 并取得了令人满意的结果。

1　装配线平衡问题描述

装配线平衡是指在工艺条件等约束下, 将有限的任务集合分配到一定数量的工作站中, 以使每个工作站的作业时间满足一定的节拍要求, 减少工作站的闲置和过载时间, 最小化平滑指标。

单一型装配线平衡问题(Simple Assembly Line Balancing Problem, SALBP), 是只考虑工序间的优先关系约束的单一产品的直线型装配线, 可分为如下三类:

(1) 给定装配线节拍, 求最小工作站数, 被称为第 iv类 ALBP, 记为 SALBP-1。

(2) 给定装配线的工作站数, 使装配线节拍最小, 以获得最大的生产率, 被称为第 ㊦类 ALBP, 记为 SALBP-2。

(3) 给定工作站数的上下限, 最小化装配线总延误时间, 即最优化装配线效率, 这类更具一般性的问题, 记为 SALBP-E。

对于 SALBP-1, 数学模型表示为:

$$\max \eta = \frac{\sum_{i=1}^{n} t_i}{m \cdot c} \Leftrightarrow \min m \text{。} \tag{1}$$

$$\text{s.t. } S_i \cap S_j = \varnothing,\ i \neq j,\ i,\ j = 1,\ 2,\ \cdots,\ m; \tag{2}$$

$$\bigcup_{k=1}^{m} S_k = E; \tag{3}$$

$$\forall i \in S_x,\ j \in S_y,\ \text{若 } p_{ij} = 1,\ \text{则 } x \leqslant y; \tag{4}$$

$$t(S_k) \leqslant c,\ k = 1,\ 2,\ \cdots,\ m\text{。} \tag{5}$$

式中, E 为装配线上任务的集合, $E = \{1,\ 2,\ \cdots,\ n\}$; S_k 为分配到第 k 个工作站的任务集合, 即 $S_k = \{i \mid$ 任务 i 被分配到第 k 个工作站$\}$; c 为装配线的节拍; $t(S_k)$ 为第 k 个工作站的总作业时间, $t(S_k) = \sum_{j \in S_k} t_j$; $P = (p_{ij})_{n \times n}$ 为 ALBP 的优先矩阵, 任务 i 是任务 j 的直接前接元素时, $p_{ij} = 1$; 任务 i 不是任务 j 的直接前接元素时, $p_{ij} = 0$。

式(2) 为发生约束, 确保任一任务只能分配到某一工作站中, 即同一任务不能重复分配至多个工作站; 式(3) 确保所有任务都被分配; 式(4) 为优先关系约束, 确保任务分配满足装配顺序关系, 没有任务分配至一个比其先前任务所分配的工作站更早的工作站中, 即对于任一任务 j 分配至工作站 y 中, 若任务 i 是任务 j 的先前任务, 则任务 i 只能分配到编号为 1 ~ y 中的某一工作站上, 而不能分配到序号大于 y 的工作站; 式(5) 为节拍约束, 表示分配到任一工作站的所有任务的作业时间之和必须不大于节拍 c。

简单 ALBP 的一个实例(Jackson 问题) 如图 1 所示。作业优先顺序图是个有向无环图, 每个节点表示一项任务, 每项任务有唯一的任务序号(圈内数字表示) 和作业时间(圈上方数字表示)。作业顺序图中的箭头线表示任务间的优先顺序关系, 箭头线的头节点表示任务完成后, 才能开始实施箭头线的尾节点表示的任务。

2　算法介绍

蚁群觅食等行为中蕴含着自组织、正反馈、随机性、分布式等重要机制, 蚂蚁能够自适应地找到从巢

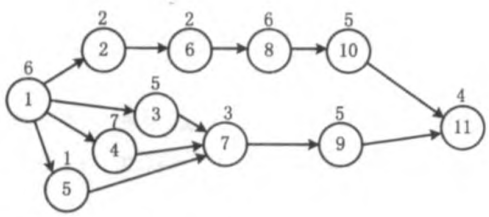

图1 作业优先顺序图实例

穴到食物源的最短路径。蚂蚁在行进的过程中不断分泌信息素,其他的蚂蚁会根据地面信息素的强弱选择行走路径。这样,距离较短的路径信息素水平高,从而又可以吸引更多的蚂蚁;而距离长的路径,信息素水平低,只有很少的蚂蚁经过。通过这种正反馈作用,最优的路径就会逐渐凸现出来。由此,意大利学者 Dorigo 等提出了一种分布式仿生的搜索算法——蚂蚁系统模型[15],在后续工作中针对蚂蚁系统中存在的问题进行了一系列改进,提出了一些新的算法模型[9],并进一步总结出了求解组合最优化问题的蚁群算法元模型——蚂蚁群体优化(Ant Colony Optimization, ACO)[16]。

蚁群算法先将所研究的问题抽象为节点模型,不同的节点对应问题中的不同实物。人工蚂蚁在节点图中根据转移概率的大小随机游走,每完成一次周游,就构成问题的一个完整解。通过信息素的正反馈作用,最后收敛到最优解。但对于本文研究的 ALBP,算法需要作适当转换才能得到较好的应用。

3 求解装配线平衡问题的蚁群算法

3.1 分配方案的生成策略

由于 ALBP 任务间优先顺序关系约束的存在,使问题复杂化而难以求解,通过采用候选任务集方法,直接在可选任务列表中进行任务的选择分配,通过减少问题搜索空间来提高搜索速度。为定义和生成候选任务集,对装配线任务作如下定义:

(1) 可选任务(available task) 从优先顺序约束角度,若一个尚未分配的任务 i 的所有先行任务 $h \in P_i$ 都已分配(P_i 表示任务 i 的紧前任务集),则任务 i 是可供选用的。

(2) 可供分配的任务(assignable task) 从优先顺序关系和节拍这两个约束角度,若一项尚未分配的任务 i 的所有先行任务都已分配至工作站 k 或更早的工作站 1, 2, …, $k-1$,以及当前工作站 k 的空闲时间对于该任务 i 是充分的,该任务被称为可供分配的任务。

单只蚂蚁 ALBP 解的构造,即分配方案生成策略,在部分分配方案中通过逐步添加可供分配的任务而生成完整的可行分配方案。将蚂蚁某时刻的所有可供分配的任务集合称为该时刻的候选任务集,记为 N_j。解的构造过程如下:① 在尚未分配的任务集合中,按优先顺序关系构建可选任务集合,进而根据节拍时间,构建当前可供分配的任务集,即候选任务集 N_j;②若 N_j 为非空集合,则按某规则从集合 N_j 中选出一项任务,分配至当前工作站;③重复 ① 和 ② 直到 N_j 变为空集,若还剩有尚未分配完的任务,则开启一个新的工作站;④ 重复 ① ~ ③ 直到全部任务分配完毕,即求得了一个解。其求解流程如图 2 所示。蚂蚁使用伪随机比率选择规则来分配任务,每构造一步解,候选任务集 N_j 作相应地更新。采用该分配方案生成策略产生可行分配方案,可行分配方案集里总是存在着最优分配方案。

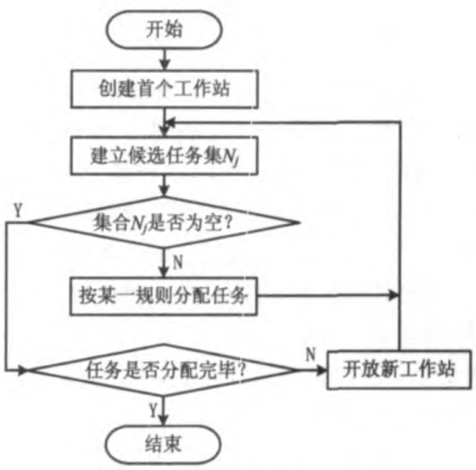

图2 装配线平衡问题的解的构造策略

3.2 蚁群算法模型

3.2.1 构造图

ALBP 可用图 $G=(C, L)$ 来表示,其中 C 是 ALBP 中任务和分配序列位置的集合,L 为任务与分配序列位置之间的连接所构成的集合。将任务分配到各序列位置的过程,可以看成是一群蚂蚁在信息素轨迹和局部启发式信息的指引下沿着图 G 移动,同时蚂蚁的移动会受到一些条件的约束,从而确保蚂蚁在移动过程中找到合适的解。

3.2.2 信息素

用蚁群算法求解 TSP 时,信息素轨迹 τ_{ij} 通常表示为紧随城市 i 访问城市 j 的期望,即提供城市 i 和城市 j 相对位置期望的一些信息。由于在 TSP 中,数列是循环的,即仅有组成解的元素的相对次序是

重要的, 数列 $\pi = (1, 2, \cdots, n)$ 和数列 $\pi' = (n, 1, 2, \cdots, n-1)$ 有相同的周游长度, 代表相同的周游。因此, 对于 TSP, 基于相对位置的信息素轨迹是合适的选择。而在 ALBP 中, π 和 π' 代表两个不同的解, 因此针对 ALBP 采用基于位置的信息素轨迹, 即信息素轨迹 τ_{ij} 代表分配任务 i 到特定序列位置 j 的期望。

蚂蚁在走过的路径上释放信息素轨迹, 该轨迹可以进一步影响后来的蚂蚁, 蚁群的集体行为表现为一种信息正反馈现象。如果在蚂蚁搜索中较多蚂蚁选择了某一分配弧段, 即一项任务 i 相对地多次分配到一个特定的序列位置 j 中, 其上留下的信息素也就越多, 后来蚂蚁选择该分配(任务 i 到序列位置 j)方式的概率也越高, 从而更增加了该路径上信息素轨迹。

3.2.3　启发式信息

启发式信息值代表了一个关于特定问题的优先信息, 选择的优劣与否直接影响算法的求解效率。针对 ALBP 的特征, 选取相应任务 i 的位置权重 pw_i 作为启发式信息 η_i, 即任务 i 及其后续任务的工时之和: $pw_i = t_i + \sum_{j \in F_i} t_j$, $i = 1, 2, \cdots, n$。其中, F_i 为任务 i 的后续任务集合。

该启发式规则同时考虑了作业时间最长的任务和后续作业数最多的任务, 即作业时间最长和后续作业数最多两规则的结合。在该方式下, 后续任务较多且需较长作业时间的任务更倾向于优先分配。在整个算法中, 位置权重不变, 即位置权重大的任务有较高的优先分配概率。

3.3　可行解的构造

首先, 每项任务和序列位置间的连接以非常少量的信息素轨迹浓度 τ_0 作为系统的初始输入。基于给定的约束, 考虑任务的优先顺序关系和节拍限制, 构建可行解并创建一个搜索列表, 由此构成了解系统的初始解阶段; 然后, 大量蚂蚁爬过解邻域, 它们从一个节点(任务)移动到另一可行节点(将任务分配到分配序列的某一位置), 并相应地进行信息素更新。重复这一过程, 直至求得最优解或达到给定的循环次数。

经典的蚁群算法用于节点模式时的 ALBP, 其转移概率的定义尚有不足的地方。考虑如下一种情况: 任务 i 的 τ_{ij} 的值很大, 表示任务 i 被分配至第 j 个位置装配的概率很大。假设此时碰巧将 τ_{ij} 很小的任务 k 分配至第 j 个作业位置, 为了能得到一个比较优的解, 任务 i 需要尽快被添加到任务序列中。如果任务 i 的 $\tau_{im}(m > j)$ 又很小, 即任务 i 被放在第 j 个位置后面的概率很小, 最终的结果可能是任务 i 被分配在非常靠后的位置, 花费了很多计算时间得到一个无用的解。

为解决上述问题, 将转移概率的定义改为如下的总合规则: 在计算转移概率时, 把任务 i 在位置 j 及 j 之前的所有信息素求和, 可以有效地避免上述情况的出现, 从而提高求解的效率。

蚂蚁使用伪随机比率选择规则, 从当前候选任务集 N_j 中选择一项任务 i, 分配到序列位置 j。即蚂蚁 k 以概率 q_0 从候选任务集 N_j 中选择任务分配到序列位置 j, 其中 i 为使 $\left[\sum_{k=1}^{j}\tau_{ik}\right]^{\alpha}[\eta_i]^{\beta}$ 达到最大的任务。该选择方式意味着蚂蚁将以概率 q_0 将最大可能的任务选入蚂蚁所构造的解; 除此之外, 蚂蚁以 $(1-q_0)$ 的概率按式(7)选择一项任务 i, 分配到序列位置 j, 即蚂蚁的状态转移公式为

$$i = \begin{cases} \arg\max\limits_{l \in N_j}\left[\sum\limits_{k=1}^{j}\tau_{lk}\right]^{\alpha} \cdot [\eta_l]^{\beta}, & q \leqslant q_0; \\ I, & \text{否则。} \end{cases} \tag{6}$$

式中, $q_0 \in (0, 1)$ 为常数, $q \in (0, 1)$ 为随机数, α 和 β 分别是控制信息素浓度和启发式信息在蚂蚁分配任务中相对重要程度的参数。

$$p_{ij} = \begin{cases} \dfrac{\left[\sum\limits_{k=1}^{j}\tau_{ik}\right]^{\alpha}[\eta_i]^{\beta}}{\sum\limits_{s \in N_j}\left[\sum\limits_{k=1}^{j}\tau_{sk}\right]^{\alpha}[\eta_s]^{\beta}}, & i \in N_i; \\ 0, & \text{否则。} \end{cases} \tag{7}$$

式中, N_j 表示蚂蚁 k 当前可供分配的任务集合, 列表 $tabu_k$ 记录了蚂蚁 k 当前已分配的任务。当所有 n 项任务都加入到 $tabu_k$ 中时, 蚂蚁 k 便完成了一次周游, 此时蚂蚁 k 所选择的任务序列便是任务的一个可行分配序列, 根据节拍的限制即可求解任务的分配方案, 即求得了问题的一个可行解。

对各选择状况下所列举的分配可能任务分别赋予一定的选择概率, 然后利用轮盘赌法, 通过取样从中随机地选择一项任务。

3.4　信息素更新策略

3.4.1　局部信息素更新

局部信息素更新的作用是使已选的分配方案对

后来的蚂蚁具有较小的影响力, 从而使蚂蚁对没有被访问的边有更强的探索能力。当蚂蚁将任务 i 分配到序列位置 j 后, 边 ij 上的信息素量按式(8) 进行更新:

$$\tau_{ij} \leftarrow (1-\rho_1)\tau_{ij} + \rho_1\tau_0。 \quad (8)$$

式中, ρ_1 为局部信息素蒸发系数, $0 \leqslant \rho_1 \leqslant 1$; τ_0 为初始化信息素值 (一个很小的初始输入, $\tau_0 = 1/(n \cdot m^*)$, $m^* = [\sum_{i=1}^{n} t_i/c]$)。

局部更新规则的作用是, 蚂蚁经过边(i, j) 后, 其信息素 τ_{ij} 降低, 从而使该边对后续的蚂蚁只有较小的吸引力, 即增加了对没有访问的边的开发力度。在实际中, 算法具有不出现停滞行为(stagnation behavior) 的效果。

3.4.2 全局信息素更新

当蚂蚁完成一次周游后, 所有任务到序列位置间的信息素都要按如下规则进行信息素全局更新:

$$\tau_{ij} \leftarrow (1-\rho_2)\tau_{ij} + \rho_2 \Delta\tau_{ij}^{bs}; \quad (9)$$

$$\Delta\tau_{ij}^{gb} = \begin{cases} 1/L^{gb}, & \text{边}(i,j)\text{ 包含在当前最优分配方案中;} \\ 0, & \text{否则。} \end{cases} \quad (10)$$

其中, L^{gb} 为当前最优分配序列(方案) 所需的工作站数, ρ_2 为全局信息素蒸发系数, $0 \leqslant \rho_2 \leqslant 1$。

只有全局最优的蚂蚁才允许释放信息素, 即在一次循环中只有构成全局最优解的任务分配方案才会增加信息素, 其他任务分配方案的信息素保持不变。通过上述方式的信息素更新, 经过有限几次搜索, 属于最优解的任务分配方案的信息素量就会明显高于其他分配方案, 搜索的目的性大大增强。

信息素更新机制满足更好的解(即分配更少的工作站数), 并将释放更多的信息素。被更多的蚂蚁分配至一些特定分配序列的任务, 和更少工作站数的解决方案的那部分任务, 其信息素轨迹将得到进一步强化。因此, 在算法的后续运行过程中, 这些任务将会更有可能被分配至同样的分配序列中。该选择模式有助于直接搜索到更好的解。上述过程体现了正反馈的机制, 使得蚂蚁能够找到问题的最优解。

3.5 算法步骤

在上述定义的基础上, ALBP 的蚁群算法实现的框架描述如下:

```
Procedure ALBP 的蚁群算法
    设置参数;
    初始化信息素;
    While (不满足结束条件)
      For 蚁群中的每只蚂蚁
       For 每个解构造步(直到构造出完整的可行解)
         蚂蚁按信息素及启发式信息的指引构造一步问题的解;
         进行信息素局部更新。
       End for
      End for
      进行全局信息素更新;
      保存当前最优解;
    End while
End procedure
```

4 计算实例

本文对实例库中的 Lutz1 问题进行求解验证[1], 该问题包含 32 项任务, 给定节拍 $c = 1\,572$, 其作业顺序如图 3 所示, 任务作业时间在标号上方给出。

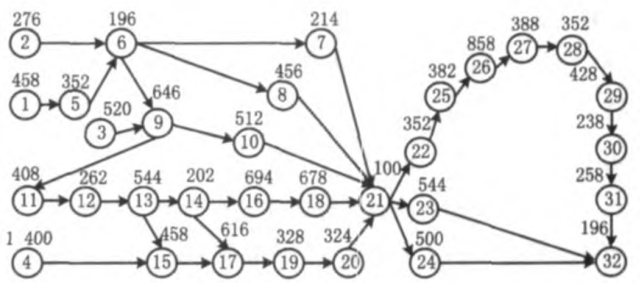

图3 Lutz1问题的作业顺序图

根据上述算法, 在 Windows XP 下用 delphi 7 开发了一个试验程序, 在 P Ⅳ2.6 G, 256 M 内存的 PC 机上, 使用该试验程序进行了多次实验计算。根据 Dorigo 等[9] 提出的蚁群算法以及多次组合实验, 设置算法各参数如下: 取蚂蚁数 $n_{ant} = 10$, 最大迭代次数 $N c_{max} = 20$, $\alpha = 1.0$, $\beta = 2.0$, $\rho_1 = 0.1$, $\rho_2 = 0.1$, $q_0 = 0.9$。本文采用的该算法在不到 1 s 时间内即可完成 Lutz1 问题的计算, 算法在第八次就求得了 Lutz1 问题的最优解(如图 4), 在第一次循环就得到了工作站数 11 的较优解, 说明了算法的高效性。位置权重启发式规则的使用, 使算法具有良好的搜索能力, 具有快速找到较优解的能力, 而蚁群算法使用伪随机比率选择规则分配任务, 能有效地脱离局部最优解, 更有利于搜索到最优解。对该问题所求得的分配方案和相应的工位时间如表 1 所示, 测试问题平衡后装配线工位负荷如图 5 所示。

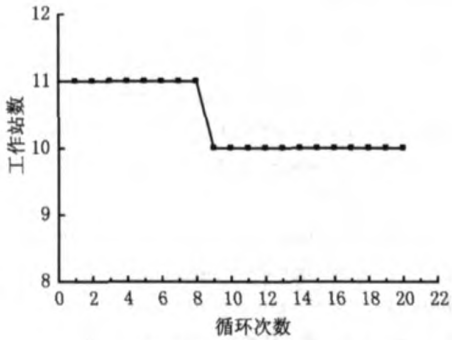

图4　测试问题目标函数的变化

表 **1**　实例计算结果

工作站序号	站作业元素	工位时间/s	空闲时间/s
1	1, 3, 5	1 330	242
2	2, 6, 9, 11	1 526	46
3	7, 12, 13, 14	1 222	350
4	4	1 400	172
5	8, 15, 17	1 530	42
6	10, 16, 19	1 534	38
7	18, 20, 21, 22	1 454	118
8	23, 24, 25	1 426	146
9	26, 27	1 246	326
10	28, 29, 30, 31, 32	1 472	100

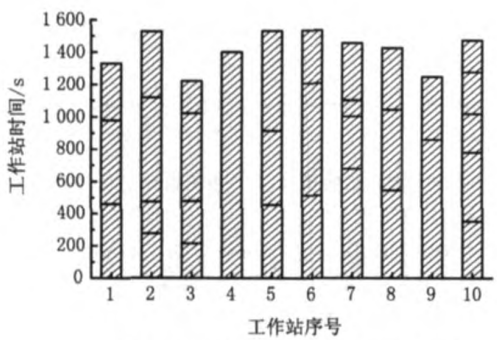

图5　测试问题平衡后装配线工位负荷图

对于较大规模问题, 适当增加算法的运算次数和蚂蚁数等控制参数, 仍然可以得到令人满意的结果, 只是在计算时间上比先前算例所需时间长。表 2 给出了分别用位置权重法和改进蚁群算法求解其他不同节拍的 20 个测试问题的解, 位置权重法对其中的 10 个问题求得了最优解, 其余 10 个求得了较最优解多一个工作站的解, 蚁群算法则对于每个问题求得了最优解。从表中可以看出, 蚁群算法计算结果明显优于位置权重法。传统的启发式方法(如位置权重法) 是从候选任务集中, 把具有某种特定性质(如位置权重最大) 的作业唯一选出的准则。而蚁群算法的选择方法不是唯一选出具有特定性质的作业, 各可供分配的任务都具有某种程度的选择可能性, 具有一定的选择概率, 且是一个动态的学习过程。因此, 能有效脱离局部最优解, 更有可能找到最优解。

表 **2**　测试问题的计算结果

问题	n	c	m_{min}	蚁群算法	位置权重法
Bowman	9	20	5	5	5
Jackson	11	10	5	5	6
Jackson	11	13	4	4	4
Buxey	29	27	13	13	14
Buxey	29	30	12	12	12
Buxey	29	33	11	11	11
Buxey	29	36	10	10	10
Buxey	29	47	7	7	8
Buxey	29	54	7	7	7
Lutz1	32	1 414	11	11	12
Lutz1	32	1 572	10	10	11
Lutz1	32	1 768	9	9	9
Lutz1	32	2 020	8	8	8
Lutz1	32	2 357	7	7	8
Lutz1	32	2 828	6	6	6
Kilbridge	45	57	10	10	11
Kilbridge	45	92	6	6	7
Kilbridge	45	110	6	6	6
Kilbridge	45	138	4	4	5
Kilbridge	45	184	3	3	4

5　结束语

本文针对 ALBP 提出了一种改进的蚁群算法, 其特点在于, 结合具体问题, 建立蚂蚁分配方案的生成策略, 在任务和任务分配序列的位置之间释放信息素, 并采用信息素总合规则进行更有效的信息素累积。以分级位置权重作为蚁群算法的启发式信息, 提高了搜索效率。实例测试结果表明, 该算法取得了很好的求解效果。

今后可将算法拓展应用至更为贴近现实的 ALBP, 如将直线型装配线改成 U 型线、人工装配和自动装配混合的装配线、动态环境中存在各种随机因素情况下的 ALBP 等。算法还可进一步推广应用至车间调度、资源受限工程调度等相关问题。

参考文献:

[1] SCHOLL A. Balancing and sequencing of assembly lines[M]. 2nd ed. Heidelberg, Germany: Physica-Verlag, 1999: 34-35.

[2] SCHOLL A, BECKER C. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing[J]. European Journal of Operational Research, 2006, 168(3): 666-693.

[3] PONNAMBALAM S G, ARAVINDAN P, MOGILEESWAR NAIDU G. A comparative evaluation of assembly line balancing heuristics[J]. International Journal of Advanced Manufacturing Technology, 1999, 15(8): 577-586.

[4] SABUNCUOGLU I, EREL E, TANYER M. Assembly line balancing using genetic algorithms[J]. Journal of Intelligent Manufacturing, 2000, 11(3): 295-310.

[5] SONG Huaming, HAN Yuqi. Genetic-algorithms-based assembly line balancing[J]. Systems Engineering, 2002, 20(1): 87-91(in Chinese). [宋华明, 韩玉启. 基于遗传算法的装配线平衡[J]. 系统工程, 2002, 20(1): 87-91.]

[6] PI Xingzhong, FAN Xiumin, YAN Junqi. Application of genetic algorithm based on task sequences to assembly line balancing[J]. Mechanical Science and Technology, 2003, 22(1): 35-38(in Chinese). [皮兴忠, 范秀敏, 严隽琪. 用基于作业序列的遗传算法求解装配线平衡问题[J]. 机械科学与技术, 2003, 22(1): 35-38.]

[7] LAPIERRE S D, RUIZ A, SORIANO P. Balancing assembly lines with tabu search[J]. European Journal of Operational Research, 2006, 168(3): 826-837.

[8] EREL E, SABUNCUOGLU I, AKSU B A. Balancing of U-type assembly systems using simulated annealing[J]. International Journal of Production Research, 2001, 39(13): 3003-3015.

[9] DORIGO M, GAMBARDELLA L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66.

[10] GAMBARDELLA L M, TAILLARD E D, DORIGO M. Ant colonies for the quadratic assignment problem[J]. Journal of the Operational Research Society, 1999, 50(2): 167-176.

[11] MERKLE D, MIDDENDORF M. Ant colony optimization with global pheromone evaluation for scheduling a single machine[J]. Applied Intelligence, 2003, 18(1): 105-111.

[12] WAN Xu, LIN Jianliang, YANG Xiaowei. Improved MMAS for vehicle routing problem with time window[J]. Computer Integrated Manufacturing Systems, 2005, 11(4): 572-576(in Chinese). [万 旭, 林健良, 杨晓伟. 改进的最大-最小蚂蚁算法在有时间窗车辆路径问题中的应用[J]. 计算机集成制造系统, 2005, 11(4): 572-576.]

[13] BAUTISTA J, PEREIRA J. Ant algorithms for assembly line balancing[C]//Proceedings of the 3rd International Workshop, ANTS 2002. Berlin, Germany: Springer-Verlag, 2002: 65-75.

[14] MCMULLEN P R, TARASEWICH P. Using ant techniques to solve the assembly line balancing problem[J]. IIE Transactions, 2003, 35(7): 605-607.

[15] DORIGO M, MANIEZZO V, COLORNI A. Ant system: optimization by a colony of cooperating Agents[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 1996, 26(1): 29-41.

[16] DORIGO M, BLUM C. Ant colony optimization theory: a survey[J]. Theoretical Computer Science, 2005, 344(2-3): 243-278.

(上接第 1578 页)

参考文献:

[1] WILLIAMSON O E. Transaction-cost economics: the governance of contractual relations[J]. Journal of Law and Economics, 1979, 22(2): 233-261.

[2] SAATY T L. How to make a decision: the analytic hierarchy process[J]. European Journal of Operational Research. 1990, 48(1): 9-26.

[3] YANG Weiwen, DENG Xianghua. Research on the corporate governance of virtual enterprises[J]. Economic Management, 2002(4): 27-30(in Chinese). [杨伟文, 邓向华. 虚拟企业的公司治理探究[J]. 经济管理, 2002(4): 27-30.]

[4] LI Jianyong, ZHA Jianzhong, E Mingcheng. Supercirculation theory and the cellular structural model of dynamic alliance [J]. China Mechanical Engineering, 2000, 11(12): 1414-1416(in Chinese). [李建勇, 查建中, 鄂明成. 超循环理论与动态联盟的细胞化结构模型[J]. 中国机械工程, 2000, 11(12): 1414-1416.]

[5] MIN Guiqin. General theories of life science and technology [M]. Beijing: Science Press, 2003(in Chinese). [闵桂琴. 生命科学技术总论[M]. 北京: 科学出版社, 2003.]

[6] SAMBAMURTHY V, ZMUD R W. Arrangements for information technology governance: a theory of multiple contingencies[J]. MIS Quarterly, 1999, 23(2): 261-290.

[7] GRANOVETTER M. Economic action and social structure: the problem of embeddedness[J]. American Journal of Sociology, 1985, 91(3): 481-510.

[8] JONES C, HESTERLY W S, BORGATTI S P. A general theory of network governance: exchange conditions and social mechanisms[J]. The Academy of Management Review, 1997, 22(4): 911-945.