蚁群算法求解装配线平衡第一类问题①

毛凌翔　郑永前　(同济大学 机械工程学院 上海 201804)

摘　要： 装配线平衡问题是生产管理中重要且较难解决的问题，其中第一类问题是装配线平衡问题的关键问题。本文通过对装配线平衡问题的分析与建模，提出了利用蚁群算法这种人工智能优化算法求解一般装配线平衡第一类问题的步骤和算法。采用启发式的方法构造分配方案的生成策略，并对信息素的更新采用局部更新与全局更新相结合的规则，从而使得该算法具有较好的目的性，大大提高了获得最优解的效率。通过该蚁群算法能得到装配线平衡第一类问题质量较优的解，且有速度快、鲁棒性、通用性等优势。

关键词： 蚁群算法 装配线平衡 人工智能

Ant Colony Algorithm for Solving the First Category of Assembly Line Balancing Problem

MAO Ling-Xiang, ZHENG Yong-Qian

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

Abstract: Assembly line balancing problem is an important and difficult problem in production management. The first category is the key to the assembly line balancing problem. Based on the assembly line balancing problem analysis and modeling, this paper proposes steps and algorithms of the first category of general assembly line balancing problem using artificial intelligence optimization algorithm of ant colony algorithm, which uses heuristic methods to generate structure program distribution strategy. It introduces pheromone local update policy with a combination of global rules, so that the algorithm has a good purpose, which greatly improved the efficiency of access to the optimal solution. This algorithm can better solve the first type of assembly line balancing problem, and have advantages like high speed, robustness and universal property etc.

Keywords: ant colony algorithm; assembly line balance; artificial intelligence

装配线是一种广泛地运用于现代化大生产的生产组织方式，装配线的平衡问题(ALB, Assembly Line Balancing)是装配线设计和运行中的重要问题，决定了生产效率和资源的利用率。第一类问题是 ALB 问题的重要和常用的基本问题。

1 引言

装配线平衡问题是一种典型的 NP-hard 问题[1]，在解决 ALB 问题中，主要经历过枚举解法、运筹学方法(如动态规划算法)、启发式算法和人工智能算法。这些方法都能解决一些装配线平衡问题，但是这些方法都具有一定的局限性。蚁群算法是一种近年来所提出的智能优化算法，它从蚂蚁觅食中得到启示，通过人工智能模拟蚂蚁的觅食路径选择来得到目标函数的最优解，具有通用性、鲁棒性、并行搜索的优点[2]。

本文主要针对装配线平衡的第一类问题进行了研究，提出了解决一般装配线平衡的第一类问题的蚁群优化算法。该算法获得 SALBP-I 问题解的质量较高，且在计算机平台上求解速度较快，同时对于大规模、多约束、复杂条件下的装配线平衡问题也有较好的求解启发性。

① 收稿时间:2009-04-14

2 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[3]。

针对平衡目标的不同，装配线平衡可以分为四类问题[4]:

第 I 类目标装配线的平衡：已知装配线的节拍，最优化装配线上的工作站数；

第 II 类目标装配线的平衡：已知装配线上的工作站数，最优化装配线上的节拍；

第 III 类目标装配线的平衡：已知装配线上的工作站数、最小化装配线的均衡指数；

第 IV 类目标装配线的平衡：已知装配线上的工作站数，最优化装配线的相关指数。

其中第 I 类问题主要在装配线的新建和扩建中要考虑的，第 II 类问题在业已运行的装配线实行生产效率的优化，第 III 和 IV 类问题多是对装配线平衡效果的评价上。其中第 I 类问题是装配线平衡问题中的主要研究的问题，也是本文所要解决的主要问题。

对于 SALBP-I 类问题可以建立以下的数学模型：

目标函数：

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

约束条件：

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

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

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

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

式中，E 为装配线上任务的集合，$E = \{1,2,\cdots,m\}$；S_k 为分配到第 k 个工作站的任务集合，即 S_k={i|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，$t(S_k) = \sum_{j \in S_k} t_j$，$p_{ij} = (p_{ij})_{n \times n}$ 为 ALBP 的优先矩阵，任务 i 是任务 j 的直接前接元素时，$p_{ij} = 1$，否则 $p_{ij} = 0$。

3 算法设计

3.1 蚁群算法

蚁群算法通过对自然界中蚂蚁觅食过程仿生而提出的一种新型的进化算法在自然界中，蚂蚁具有在爬行过程中释放信息素(pheromone trail)而找到从蚁穴到食物采集地之间最短路的能力。蚂蚁通过信息素相互传递关于路径的信息：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。信息素在这种正反馈作用下，逐渐会聚集到较短的路径上，最优路径会逐渐凸显出来。蚁群算法就是运用这种蚂蚁觅食的生物学原理来设计仿真算法的。蚁群算法的流程如图 1 所示。

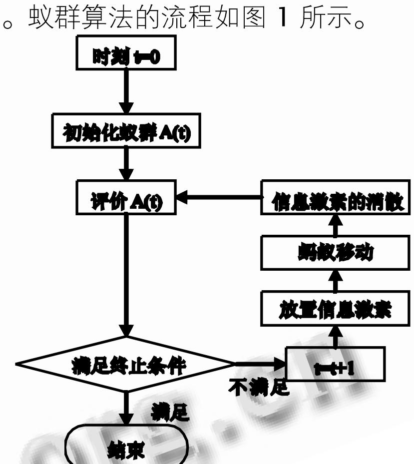

图 1　蚁群算法流程图

3.2 蚁群算法求解 SALBP-I 类问题

本文采用由 Dorigo M, Maniezzo V 在文献[2]提出的 AS 启发式方法作为求解 SALBP-I 问题的起点：在每次迭代中设定 m 只蚂蚁，每只蚂蚁在一定的概率的基础上，由优化规则和已经过该路径的蚂蚁的信息素来决定该蚂蚁的路径选择(详见第 3.3 节)，在此有表示第 i 个任务的启发式信息，用表示当前路径上的信息素浓度。

基于以上的假设，候选任务集 D 中任务 j 的选到的可能性(p_{ij})用下式计算[5]

$$p_{ij} = \frac{\left(\sum_{k=1}^{i}[\tau_{kj}]\right)^{\alpha} \cdot [\eta_j]^{\beta}}{\sum_{h \in D}\left(\sum_{k=1}^{i}[\tau_{kh}]\right)^{\alpha} \cdot [\eta_h]^{\beta}} \tag{6}$$

其中，τ_{kj} 和 τ_{kh} 取决于信息素的更新策略；η_j 和 η_h 决定于任务的优先选择策略，η_j 与任务 i 的候选任务数之间存在线性关系。k 取决于信息素的更新策略，代表已选择的任务、位置或工作站。α 和 β 是启发式的常数，会影响到蚂蚁行进中的启发式信息和信息素的更新策略。

本文中蚂蚁采用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，分配到序列位置 j。蚂蚁选择一项任务 i，分配到序列位置 j，其选择概率可以用下式计算[3]：

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

式中，N_j 表示蚂蚁 k 当前可供分配的任务集合。蚂蚁的状态转移公式为[3]：

$$i=\begin{cases}\arg\max\limits_{i\in N_j}\left[\sum_{k=1}^{j}\tau_{lk}\right]^{\alpha}\bullet[\eta_l]^{\beta}, q\le q_0; \\ I, \quad 否则.\end{cases} \tag{8}$$

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。

列表 tabuk 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 tabuk 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。

在每个蚂蚁子群都完成对当前位置的路径选择后，信息素就要进行一次更新，且只有路径最优的蚂蚁才会对所行走的直接或反复路径释放信息素。

当蚁将任务 i 分配到序列位置 j 后，让蚂蚁对已选择的方案的有较小的选择概率，就要对信息素进行一次局部信息素的更新，使蚂蚁对没有被访问的边有更强的探索能力，局部信息素的更新策略可以用下式表示：

$$\tau_{ij}=(1-\rho_1)\tau_{ij}+\rho_1\tau_0 \tag{9}$$

式中，ρ_1 为局部信息素蒸发系数，$0\le\rho_1\le 1$；τ_0 为初始化信息素值。

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，使最优解能够在多次的搜索后能找到最优解：

$$\tau_{ij}=(1-\rho_2)\tau_{ij}+\rho_2\Delta\tau_{ij} \tag{10}$$

其中，为全局信息素蒸发系数，，为前后最优工作站的信息素差值。

这种信息素的更新机制具有很好的目的性，通过多次迭代，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。

由于在利用蚁群算法求解 ALBP 中局部搜索策略起到了致关重要的作用，使每个蚂蚁子群都有 1%的机率获得最优解[6]，本文中采用的局部搜索策略如第 3.3 节所述。

3.3 求解 SALBP-I 问题的贪婪算法

求解 SALBP-I 问题的贪婪算法具体步骤可以用图 2 来表示，通过这种贪婪算法就可以在满足约束条件下，把所有的任务分配到一定的工作站中。该算法可以作为蚁群算法求解 ALBP 的一部分，可以生成令人满意的初始可行解。

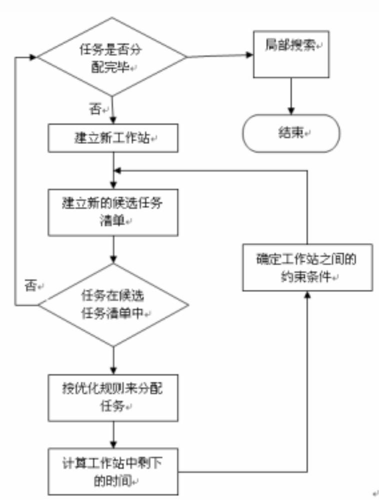

图 2　获得 ALBP 可行解的贪婪算法流程图

由于这种算法会把造成后序工作站的空闲时间比较大，故要对可行解进行在邻域内的局部爬山搜索策略：①交换相邻工作站中的两个任务，且交换后的工作时间比先前缩短；②把后序的工作站中的任务移动到前一个工作中。通过这种爬山搜索策略，可以得到比初始可行解更优的可行解。在该算法中不断重复爬山搜索策略直到无法获得更优的解。

4　算法实验

本文的算法采用 VB 6.0 编写的自制软件在 Intel Core 2 Duo E6550 2.33GHz 2G 内存的计算机平台上测试通过，下面就通过一个实际算例来验正该算法的有效性。下面就以生产某种型号的自行车装配线为例。在该装配线上生产一种型号的自行车的需完成 13 个独立的装配任务，其作业时间和优化顺序如图 3 所示(右上角任务的持续时间，单位为 s)。

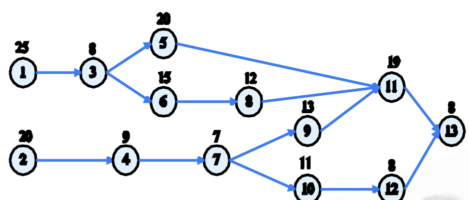

图 3　某自行车装配线的作业优先顺序图

由所示的任务顺序图可以算出装配一辆自行车的时间 T 为：

$$T=\sum_{i=1}^{N} t_i = 175$$

该种自行车每日需求量为 D=800 件，以一天工作时间 W=8 小时计算，则该自行车装配线的节拍 C 是：

$$C=\frac{W}{D}=\frac{8\times 60\times 60s}{800}=36s$$

则理论上的最小工作站数为：

$$N_t=\left[\frac{T}{C}\right]+1=[4.86]+1=5$$

利用自行开发的 VB 软件来求解该问题，取蚂蚁数 $n_{ant}=20$，最大迭代次数 $Nc_{\max}=20$，$\alpha=1.5$，$\beta=2.0$，$\rho_1=1.0$，$\rho_2=1.0$，$q_0=1.5$。程序运行过程中，工作站数随循环次数的增多而不断趋于最优，如图 4 所示，在进行 30 次的迭代后可行工作站数趋于稳定，达到最优值，运算结果如表 1 所示。

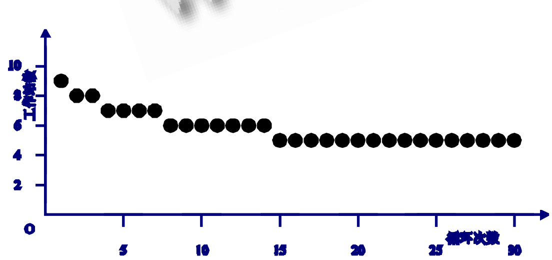

图 4　工作站数随循环次数的变化

该装配线的生产效率和平衡率分别为：

表 1　蚁群算法的计算结果

工作站序号	工作站作业任务	工作站作业时间	工作站空闲时间
1	1, 3	33	3
2	2, 4, 7	36	0
3	5, 6	35	1
4	8, 9, 10	36	0
5	11, 12, 13	35	1

$$\eta=\frac{\sum_{i=1}^{k} w_i}{k\cdot C}=\frac{175}{5\times 36}=97\%$$

$$\sigma=\sqrt{\sum_{i=1}^{k}(w_i-C)^2/k}=0.66$$

基于以上分析，本文所提出的蚁群算法的求解结果可以达到理论最优值，且该装配线的生产效率和平衡率均能达到令人满意的结果。

5　结论

本文通过对装配线平衡的第一类问题进行了数学建模，利用蚁群算法对该问题进行求解。在求解过程中，通过嵌入贪婪爬山搜索策略构建每次迭代的可行解，并建立可以较大可能获得最优解的信息素的更新机制。通过测试实例证明本文所提出的蚁群算法在求解装配线平衡第一类问题的有效性并具有很强的实用价值。利用蚁群算法求解装配线平衡问题具有求解速度快、迭代次数少、最优解的质量高等优势。

参考文献

1 Gutjahr AL, Nemhauser GL. An algorithm or the line balancing problem. Management Science, 1964,11(2): 308－315.

2 Dorigo M, Maniezzo V, Colorni A. The ant system: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B (S1094-6977), 1996,26(1):29－41.

3 张则强,程文明,钟斌,王金诺.求解装配线平衡问题的一种改进蚁群算法.计算机集成制造系统，2007, 13(8):1632－1638.

4 周亮.装配线平衡的最优化模型与算法研究[博士学位论文].南京:南京理工大学,2005.

5 Merkle D, Middendorf M. An ant algorithm with a new pheromone evaluation rule for total tardiness problems, Proc. of the EvoWorkshops. 2000.

6 Bauer AB. Bullnheimer R, Hartl F, Strauss C. An ant colony optimization approach for the single machine total tardiness problem. Proc. of the 1999 Congress on Evolutionary Computation (CEC'99). IEEE Press, Piscataway, NJ, 1999.1445－1450.