

改进的蚁群算法在修磨轨迹优化中的应用

武利生¹, 权 龙¹, 杨付生²

(1. 太原理工大学机械电子工程研究所, 太原 030024; 2. 太原中海机械设备有限公司, 太原 030041)

摘 要: 提出一种适用于钢坯修磨轨迹优化问题的改进蚁群算法, 给出一种修磨轨迹优化问题的实用数学模型。针对蚁群算法对参数敏感的问题, 提出用启发信息归一化来解决的办法。仿真实验与初步试用结果表明, 经改进蚁群算法优化的修磨轨迹能大幅度减少修磨过程中的空行程。该算法具有一定的理论参考价值和实际意义。

关键词: 蚁群算法; 钢坯修磨; 轨迹优化

Application of Improved Ant Colony Algorithm in Grinding Path Optimization

WU Li-sheng¹, QUAN Long¹, YANG Fu-sheng²

(1. Research Institute of Mechanical and Electronic Engineering, Taiyuan University of Technology, Taiyuan 030024;

2. Taiyuan Shenhai Machinery Co. Ltd., Taiyuan 030041)

【Abstract】 An improved Ant Colony Algorithm(ACA) is presented that can be used to search shortest grinding path for the grinding machine and the grinding path optimization mathematical model is given. The attractiveness normalization is introduced and adopted to decrease the optimization result's dependence on the parameters selected in the basic model ant colony algorithm. Simulation experiment and application show that the location path is shortened greatly after the path is optimized by ant colony algorithm.

【Key words】 Ant Colony Algorithm(ACA); steel grinding; path optimization

1 概述

修磨是钢材轧制加工过程中常用的一道加工工序, 特别是对不锈钢或某些特殊钢, 几乎是不可或缺的工序。修磨的目的是去除钢坯表面的夹渣、裂纹、麻点、划伤等缺陷, 以保证下道工序的正常进行, 或满足出厂要求。为了去除表面缺陷可以采用全修磨, 就是完全去除钢坯表面的一层金属, 俗称“剥皮”; 也可采用局部修磨, 即只对缺陷局部区域修磨, 俗称“点修”。局部修磨与全修磨相比能大大减少金属去除量, 节约成本, 同时具有砂轮消耗小, 节能、环境污染小等一系列优点, 是钢坯修磨的发展方向。

ZXM 纵向修磨机是太原中海机械设备有限公司研制的用于棒材精整生产线的局部修磨设备。ZXM 修磨机在修磨前先对钢坯表面进行涡流探伤, 确定缺陷位置, 然后控制磨头从停靠位置出发, 按一定的修磨轨迹移动磨头, 逐点修磨缺陷。因轧制原因, 缺陷位置在表面随机离散分布, 确定一条通过各缺陷的最短或较短修磨轨迹, 是提高修磨效率的最有效途径。本文尝试用蚁群算法来确定一条最优或次优的修磨路径。

2 基本蚁群算法

蚁群算法(Ant Colony Algorithm, ACA)是 20 世纪 90 年代由 M.Dorigo 等人提出的一种新型的模拟进化算法^[1]。蚁群算法具有正反馈、分布式计算的特点, 研究显示蚁群算法在求解离散优化问题方面具有一定的优势。

真实蚂蚁在没有视觉的情况下, 能够找到从食物源到蚁穴的最短路径。这是因为在蚂蚁个体之间通过一种称为信息素(pheromone)的物质进行信息传递。蚂蚁在运动中, 不但能

够在它所经过的路径上留下该物质, 而且能够感知这种物质的存在及其强度, 并朝该物质强度高的方向移动, 以此指导自己的运动方向。因此, 由大量蚂蚁组成的蚁群的集体行为表现出一种信息正反馈现象: 某一路径上走过的蚂蚁越多, 则后来的蚂蚁选择该路径的概率越大。蚂蚁个体之间就是通过这种信息交流进行路径的最优选择, 从而达到搜索食物的目的。

蚁群算法最初是为解决旅行商问题(TSP)而提出的。算法的基本实现过程如下: 假设将 m 只蚂蚁放入到 n 个随机选择的城市中, 每只蚂蚁根据一定的概率选择下一个它还没有访问过的城市。蚂蚁选择下一个目标城市的主要依据有以下 2 点: (1) $\tau_{ij}(t)$ 为 t 时刻连接城市 i 和 j 的路径上的信息的浓度。初始时刻, 各条路径上信息量相等, 在实验中设 $\tau_{ij}(0) = C$ (C 为常数)。 (2) η_{ij} 为由城市 i 转移到城市 j 的可见度, 亦称启发信息, 该启发信息根据所要解决的问题确定。通常在 TSP 问题中, 取 $\eta_{ij} = 1/d_{ij}$, d_{ij} 表示城市 i 和 j 间的距离。 t 时刻位于城市 i 的蚂蚁选择城市 j 为目标城市的概率为

$$\rho_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t)\eta_{ij}^\beta(t)}{\sum_{s \in allowed_k} \tau_{is}^\alpha(t)\eta_{is}^\beta(t)} & j \in allowed_k \\ 0 & \text{否则} \end{cases} \quad (1)$$

蚂蚁 k 根据此概率选中某个城市, 可用轮盘赌算法确定。为了避免对同一个城市的重复访问, 每一只蚂蚁都保存一个

作者简介: 武利生(1972—), 男, 讲师、硕士, 主研方向: 机电一体化; 权 龙, 教授、博士、博士生导师; 杨付生, 高级工程师

收稿日期: 2008-08-26 **E-mail:** lisheng_wu@sina.com

列表 $Tabu(k)$, 用于记录到目前为止蚂蚁已经访问过的城市集合。 $Tabu(k)$ 随着蚂蚁寻优过程作动态调整。为了避免残留信息素过多引起残留信息淹没启发信息, 在每一只蚂蚁走完一步或者完成对所有 n 个城市的访问后(也即一个循环结束后), 对残留信息进行更新处理。同时随时间推移原有的信息素要挥发一部分, 用 $\rho \in [0,1]$ 表示信息素的保留原度, $1-\rho$ 表示信息的挥发度, $(t+1)$ 时刻路径上的信息素浓度为

$$\tau_{ij}(t+1) = \rho \tau_{ij}(t) + \Delta \tau_{ij} \quad (2)$$

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k \quad (3)$$

根据信息素更新策略的不同, 有 ant-cycle, ant-quantity 和 ant-density 3 种不同的蚁群算法模型^[2]。其中, ant-quantity 和 ant-density 模型信息素的更新机制利用的是局部信息; 而 ant-cycle 利用的是整体信息。

蚁群算法提出后针对其收敛速度慢、计算时间长、易于过早陷入局部最优解的问题, 陆续提出了各种改进算法^[3-4], 增强了蚁群算法寻找最优解的能力。

从蚁群搜索最短路径的机理看到, 算法中 α, β, ρ, Q 有关参数的不同选择对蚁群算法的性能有至关重要的影响^[5]。不同的参数选择对解的好坏以及收敛速度影响很大, 有时甚至会导致算法不收敛。通常根据实验和经验确定这些参数。

3 修磨轨迹优化的数学模型

理论上的 TSP 问题中的城市抽象为一点, 路径抽象为联结两点的直线。而实际应用中钢坯修磨缺陷有一定的长度及宽度, 为适用蚁群算法计算作如下简化处理。

3.1 缺陷的城市表示

设共有 n' 个缺陷, 用 $F_{i'}$, $i' = 1, 2, \dots, n'$ 表示第 i' 个缺陷, 缺陷廓线上 2 个最远点为 $A_{i'}(x_{i'a}, y_{i'a})$ 和 $B_{i'}(x_{i'b}, y_{i'b})$, 缺陷 $F_{i'}$ 用两城市 C_j 及 C_{j+1} 表示, 坐标分别为 $A_{i'}(x_{i'a}, y_{i'a})$ 和 $B_{i'}(x_{i'b}, y_{i'b})$, C_j 和 C_{j+1} 组成一城市对。也就是用两城市来代表缺陷, 城市坐标为缺陷中相距最远的 2 点的坐标。

3.2 城市距离定义

ZXM 纵向修磨机磨头的横向与纵向运动无关, 两方向可同时运动, 但不联动, 两点(城市)间的距离不能用通常公式计算。如用 v_x 表示纵向运动速度, v_y 表示横向运动速度。城市 $C_i(x_i, y_i)$ 与城市 $C_j(x_j, y_j)$ 间的距离 d_{ij} 定义为

$$d_{ij} = v_x \cdot t \quad (4)$$

其中, $t = \max\left\{\frac{|x_a - x_b|}{v_x}, \frac{|y_a - y_b|}{v_y}\right\}$ 。

经过以上简化处理并将磨头停靠点定义为城市 $C_0(x_0, y_0)$, 则寻找最佳修磨轨迹问题转化为寻找一条从 C_0 出发经过所有城市的最短路径的 TSP 问题。

修磨轨迹的确定是在线的, 必须在有限的时间中完成轨迹优化, 而且随棒材规格不同缺陷间平均距离变化范围较大, 缺陷的分布情况也因不同加工工艺区别很大, 算法的参数无法通过实验来确定, 只能预先给定。本文针对实用中参数无法改变这一情况, 在传统蚁群算法的基础上参考蚁群算法研究的最新成果, 以归一化的启发信息为基础提出一种改进的蚁群算法。

4 基本算法的改进

4.1 启发信息归一化

从式(1)可知, 决定蚂蚁转移的概率由信息素与启发信息共同确定。传统的算法中信息素给定一初始值后动态变化,

启发信息则取两城市距离的倒数。可见, 因信息素与启发信息的大小可能相差较大, 当参数选择不合适时, 信息素或启发信息中的一项过早地起决定性作用, 很快收敛到某一路径, 而这往往并非最短路径。只有将信息素与启发信息限制在适当的范围, 才能取得较好的搜索效果。为此目的可以将启发信息进行归一化处理, 步骤如下:

(1)按通常定义求出启发信息: $\eta_{ij} = 1/d_{ij}$ 。

(2)从所有启发信息中找出最大启发信息 η_{\max} 与最小启发信息 η_{\min} 。

(3)用下式计算归一化的启发信息:

$$\eta'_{ij} = \frac{\eta_{ij} - \eta_{\min}}{\eta_{\max} - \eta_{\min}} \quad (5)$$

显然, 启发信息 η'_{ij} 被限制在 $[0,1]$ 之间。

4.2 信息素限制

为了避免算法过早收敛, 已有学者提出将信息数限制在某一区间^[6]。在本算法中为与启发信息相当将信息素限制在 $[0,1]$ 之间, 并设定初始信息素为 0.5。

4.3 新转移规则

随搜索的进行, 经过一段时间的运行, 路径上信息素会逐步集中于较短的路径上, 而这很可能是局部最短路径, 也就是算法陷入局部最优解而停滞, 不能得到全局最优解。因此, 为了能减少停滞, 可以动态地增加搜索的多样性, 也就是随信息素残留增加, 蚂蚁局部选择可能性加大, 而通过增加搜索的多样性, 可以减小停滞^[4]。参考文献[4]并加以改进, 为每一只蚂蚁设一个转移判断阈值 $q_k \in [0.3, 0.8]$, 在每一次搜索完成后, 按下式动态调整并将其限制在 $[0.3, 0.8]$ 中:

$$q_k = \frac{\max L - L_k}{\max L - \min L} \quad (6)$$

其中, L_k 为第 k 只蚂蚁在本次搜索中所经过的路径长度。这样, 路径长度越短的蚂蚁的 q_k 越大, 也就是下一次循环中随机搜索的可能性加大, 而路径越长的越小, 下一次循环中信息素与启发信息的作用加强。

每次蚂蚁转移前产生一个随机数 q , 转移到城市 j , 有

$$j = \begin{cases} \arg \max\{\tau_{iu} \times \eta'_{iu} \times \eta'_{iu}\}_{u \in J_k(s-1)} & q \geq q_k \\ \text{用赌轮法确定} & \text{否则} \end{cases} \quad (7)$$

其中, $J_k(s) = \{1, 2, \dots, n\} - tabu_k(s)$, 表示第 k 只蚂蚁下一步可以选择的节点集。

4.4 联合使用全局及局部信息素的更新机制

(1)全局更新规则

当 m 只蚂蚁成功地完成一次寻径过程后, 将选出总路径最小的路由, 按下式进行全局信息素的更新, 使得较优解保留下来, 对后继蚂蚁产生影响, 加快收敛到最优解的速度。

$$\Delta \tau_{ij} = \begin{cases} d\tau & \text{在本次最短路径上} \\ 0 & \text{否则} \end{cases} \quad (8)$$

其中,

$$d\tau = \begin{cases} 0.12 \max L / \min L & \text{本次最短路径比记录最短路径小} \\ 0.1 \max L / \min L & \text{否则} \end{cases} \quad (9)$$

其中, $\min L$ 是本次循环中蚂蚁经过的最短路径; $\max L$ 是本次循环中蚂蚁经过的最长路径。

(2)局部更新规则

按下式进行信息素局部更新:

$$\Delta \tau_{ij}^k = \begin{cases} Q & \text{蚂蚁 } k \text{ 在本循环中经过路径 } ij \\ 0 & \text{否则} \end{cases} \quad (10)$$

其中, $Q = 0.01/n$, n 是城市数量。

4.5 增加干扰

经过一段时间的运行,仍可能收敛到局部最短路径。为避免过早收敛而停顿,可在完成一次搜索后,分析各蚂蚁的路径。当大部分蚂蚁的路径是当前最短路径时,可在信息素中增加随机干扰,以增加搜索的多样性。本算法按如下公式增加干扰:

$$\tau_{ij} = 0.5 \times \tau_{ij} + 0.5 \times r \quad (11)$$

其中, r 是[0,1]的随机数。

5 改进蚁群算法求解修磨最佳路径

用改进群算法优化修磨路径步骤如下:

Step1 读入缺陷数据,按上述城市化简方法确定城市数目 n 及城市坐标。

Step2 按式(4)计算城市间距离。

Step3 按式(5)计算启发信息 η'_{ij} 。

Step4 $\tau_{ij} = 0.5, NC = 100$; 将 $m = n$ 只蚂蚁放到 n 个城市中,每个城市一只蚂蚁。 NC 是总循环次数。

Step5 确定蚂蚁 k 要由当前城市 C_i 转移到城市 C_j 。如果 C_i 的配对城市在尚未访问记录中,则 C_j 为配对城市;否则,按新转移规则确定 C_j ,记录信息素增量 $\Delta\tau_{ij} = \Delta\tau_{ij} + Q$, 本步重复 m 次,所有蚂蚁转移一次。

Step6 重复 Step5 n 次,所有蚂蚁完成一次搜索。

Step7 计算本次搜索最长、最短路径长度,如果本次最短路径比记录最短路径更短,更新最短路径记录。判断是不是结束搜索,是则到 Step12, 否则继续。

Step8 按式(8)增加最短路径上的信息数增量。

Step9 所有路径信息数按式(2)更新,其中 $\rho = 0.9$ 。

Step10 按式(6)动态更新 q_k 。

Step11 判断是否需增加信息素干扰,是则按式(11)增加干扰,转移到 Step5。

Step12 结束,输出最短路径。

6 仿真实验结果及讨论

6.1 对Eil51的计算结果

为验证算法的有效性,对Eil51的TSP问题进行计算,在 $\alpha = 1, \beta = 2, \rho = 0.9$ 固定参数下300次循环计算的最佳结果为428.87,所找到的最佳路径如图1所示。

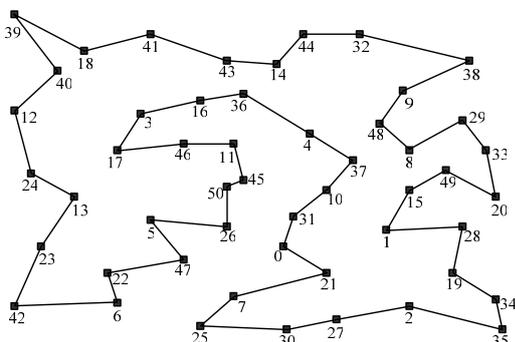


图1 Eil51 计算结果

6.2 对缺陷的仿真

在计算机中模拟生成500支钢坯的缺陷,其中棒材长度在4000mm~6000mm间随机分布,缺陷数在2~40间随机

分布。缺陷长度在0~1/5棒材长度之间随机分布。对生成的每一幅缺陷分布图进行修磨轨迹优化。将优化结果与顺序路径(按缺陷起始位置由小到大排列)进行比较。仿真参数 $\alpha = 1, \beta = 2, \rho = 0.9$ 。循环优化次数 $N = 100$ 次。仿真实验结果如表1所示。

表1 仿真实验结果

仿真参数	500次总计	平均
缺陷数	9 865	19.73
缺陷长度	4 839 739.38	9 679.478 76
顺序修磨轨迹长度	11 140 314.96	22 280.629 92
优化后轨迹长度	7 332 928.14	14 665.856 3
顺序修磨空行程长度	6 300 575.58	12 601.151 16
优化修磨空行程长度	2 493 188.76	4 986.377 52
空行程减少长度	3 807 386.82	7 614.773 64
相对空行程减少(100%)	60.4%	

由仿真结果可见,空行程数减小60.4%,优化效果明显。通过对仿真过程中间结果分析还发现,缺陷数越多,空行程减少的相对值也越大,这是因为缺陷较少时优化效果不明显。

需要指出仿真实验中假定 $v_x = v_y$, 城市间距离的定义也采用平面2点距离公式,这与实际使用中不同。

本算法已在申海公司生产的新的修磨机中试用。根据某用户初步统计显示修磨效率约提高5%~10%,比仿真结果小,这是因为砂轮修磨时运动速度低、时间长,而空行程运动相对来说速度高、时间短。因此,虽然空行程路程缩短较多,但所用时间缩短不明显。

7 结束语

本文提出了一种适用于钢坯修磨的修磨轨迹优化问题的蚁群算法。讨论了将修磨轨迹优化转化为TSP问题时如何确定城市数量、位置以及城市间距离的问题,并给出一种实用数学模型。为减少运算时间,加快算法收敛,算法中联合采用了信息素限制、局部与全局信息素更新、增加干扰等方法。针对蚁群算法对参数敏感的问题,提出了用启发信息归一化来解决的办法。仿真实验与初步试用结果表明,经过优化的修磨轨迹能大幅度减少修磨中的空行程,在提高效率的同时,减少了机器磨损,延长了使用寿命。

参考文献

- [1] Colomni A, Dorigo M, Maniezzo V. Ant System for Job-shop Scheduling[J]. Belgian Journal of Operations Research, Statistics and Computer Science, 1994, 34(1): 39-54.
- [2] Dorigo M, Bonabeau E, Theraulaz G. Ant Algorithms and Stigmergy[J]. Future Generation Computer Systems, 2000, 16(8): 851-871.
- [3] 陈宏建,徐晓华,屠莉.改进的增强型蚁群算法[J].计算机工程,2005,31(2):176-178.
- [4] 胡勇.动态跃迁转移蚁群算法[J].计算机工程,2005,31(1):167-168.
- [5] 叶志伟,郑肇葆.蚁群算法中 α, β, ρ 设置的研究——以TSP问题为例[J].武汉大学学报:信息科学版,2004,9(7):597-601.
- [6] Stutzle T, Hoos H H. Improvements on the Ant System: Introducing Max Min Ant System[C]//Proc. of Int'l Conf. on Artificial Neural Networks and Genetic Algorithms. [S. l.]: Springer-Verlag, 1997: 245-249.

编辑 顾逸斐