

蚂蚁算法在矩形件优化排样中的应用

黄红兵

(福建农林大学 计算机学院 福建 福州 350002)

摘 要 蚂蚁算法是解决优化问题的一种相对较新的启发式算法,大规模的矩形件优化排样问题是个 NP 难题。文中尝试用蚂蚁算法求解矩形件优化排样问题,根据提出的求解算法,开发出了基于蚂蚁算法的计算机辅助优化排样系统,并将蚂蚁算法的求解结果和遗传算法进行了对比,试验结果证明了用蚂蚁算法求解矩形件优化排样问题的可行性和有效性。

关键词 蚂蚁算法;矩形件排样;启发式算法;遗传算法

中图分类号:TP301

文献标识码:A

文章编号:1002-2336(2006)02-0098-03

1 引言

蚂蚁算法(Ant Algorithm)^[1]是一种源于生物世界的仿生类进化算法,该算法模仿蚂蚁觅食时的行为,按照启发式思想,借助信息素 Pheromone 的作用,通过个体之间的信息交流与协作,逐步收敛到问题的全局最优解。

蚂蚁算法自问世以来表现出了强大的生命力。蚂蚁算法出现后,用来求解的典型问题主要有旅行商问题、二次规划、度限制树、背包问题^[3]、工件排序优化^[4]、装箱问题^[5]等。

矩形件优化排样是指在给定的矩形板材上排放出所需要的矩形件,使材料的浪费尽量少。如何合理地原材料上切割出给定规格的零件,尽可能提高原材料的利用率,是一个有重大经济意义的课题。大规模的矩形件优化排样问题是个 NP 难题,目前的求解算法多采用启发式算法,如遗传算法、模拟退火算法等。启发式算法能够在较短的时间内求出问题的较优解,但不能保证求得问题的最优解。本文尝试用蚂蚁算法来求解矩形件优化排样问题。

2 矩形件排样问题的蚂蚁算法求解

2.1 问题求解的基本思路

对于给定的 n 个矩形,排样时除了要确定每个矩形的排放顺序和排放方式外,还要确定矩形在板材上具体的排放位置。对于排放顺序已定的矩形序列,用何种排放算法进行排放,对问题的求解效果影响很大。本文中采用的矩形排放算法为基于最低水平线的搜索算法^[6]。

矩形件优化排样问题的蚂蚁算法求解,其基本思想是将问题分为两个相互融合的部分:先通过蚂蚁算法确定矩形的排放顺序,然后按照特定的矩形排放算法生成排样图,即:借助信息素,通过蚂蚁对矩形的访问,用蚂蚁找出矩形的排放先后顺序并确定每个矩形的排放方式。按矩形排放算法生成排样图后,通过比较不同的矩形访问序列对应的板材利用率,最终得到较优的排样方案。

在蚂蚁算法中,信息素对蚂蚁的行动具有指导作用,而蚂蚁访问矩形的先后顺序对应矩形被排放的先后次序,所以在用蚂蚁算法求解矩形排样问题时,矩形上信息素的计算及信息素的更新规则如何确定才比较合理,是

问题求解的关键。

2.2 信息素的计算和更新

在实际排放过程中,矩形往往按面积从大到小进行排放。一般情况下,零件面积越小,板材的面积越大,材料的利用率就会越高。当大零件排放完后再排放小零件,板材空间变得越来越小且会产生很多碎块,如果比较小的矩形最后排放,填充这些碎块的可能性将会增加,有利于提高板材的利用率;反之如果先排放小零件,后面留给大零件的空间选择余地就会越来越少,板材的利用率就会越来越低。另一方面,对于长宽比相差较大的矩形件,应该优先排放^[7]。

考虑到以上两点,本文对矩形 $next$ 上的信息素 $pheromone$ ($next$),定义为: $pheromone(next) = WeightArea * Area(next) + WeightRatio * (L/W)_{next}$,其中 $WeightArea$ 表示面积因素在信息素中计算中所占的权重, $Area(next)$ 表示矩形 $next$ 的面积, $WeightRatio$ 表示矩形长宽比值因素在信息素中所占的权重, $(L/W)_{next}$ 表示矩形 $next$ 的长度与宽度的比值。

每当一个矩形被蚂蚁访问时,就表示对应的矩形被选中参与排样,矩形上的信息素就要及时进行更新,由于每一个矩形迟早都会被不同的蚂蚁访问到,只不过是蚂蚁访问到的先后顺序有差别,本文中约定:蚂蚁访问到矩形的不同顺序对应不同的优先级,如果一个蚂蚁访问到一个矩形的次序越早,那么该次访问的优先级越高,蚂蚁在相应的矩形上留下的信息素浓度增量就越大;反之,则相应的蚂蚁在对应的矩形上留下的信息素浓度增量就越小。即:多个蚂蚁即使都访问了同一个矩形,由于访问到的先后次序不同,每个蚂蚁留给同一个矩形的信息素浓度增量可以不一样。同一个蚂蚁访问了面积大小不等的矩形后,给各个矩形带来的信息素浓度增量是不一样的,增量与矩形的面积成反比。对于信息素的更新,本文采用如下的处理方法: $pheromone_k(next) = (1 - 0.9 * \alpha * k/n) * Unit(1/Area(next))$,其中 $pheromone_k(next)$ 表示第 k 个蚂蚁给矩形 $next$ 留下的信息素浓度增量, $\alpha * k/n$ 表示蚂蚁 k 从出发矩形开始,当访问到矩形

next 时已经访问过的矩形总数 (包括矩形 next) n 表示要排放的矩形总数, Unit 表示一个标准的信息素单位, Area (next) 表示矩形 next 的面积。

2.3 蚂蚁算法的参数设置

(1) 状态转移规则

在确定状态转移概率规则时, 要综合考虑矩形面积和矩形长宽比两个方面的因素。在下面的式子 (1) 中, current、next 表示矩形, Importance 是常数参数, 表示矩形面积大小对蚂蚁选择影响程度的系数, remain 表示尚未排放的矩形。如果蚂蚁 k 选择矩形 next 作为下一个要访问的矩形, $\text{JumpProbability}_k(\text{current}, \text{next})$ 表示第 k 只蚂蚁从当前所在矩形 current 转移到下一个矩形 next 的概率; 当 next 属于蚂蚁 k 已经访问过的矩形时, $\text{JumpProbability}_k(\text{current}, \text{next})=0$, 否则 $\text{JumpProbability}_k(\text{current}, \text{next}) = \frac{[\text{pheromone}(\text{next})]^{\text{Importance}} \cdot \text{Area}(\text{next})^{\text{Importance}}}{\text{pheromone}(\text{remain}) \cdot \text{Area}(\text{remain})^{\text{Importance}}}$ 。

(2) 全局信息素修正规则

当每一只蚂蚁都访问完所有的矩形时, 对每个矩形上的信息素作一次全面的修正, 公式如下 $\text{pheromone}_i(t + \text{span}) = \text{Last} \cdot \text{pheromone}_i(t) + \text{pheromone}_i(1 \leq i \leq m)^{[8]}$, $0 < \text{Last} < 1$, Last 表示矩形上信息素的持久性系数, m 对应蚂蚁的个数, 蚂蚁的编号从 1 到 m , pheromone_i 表示 m 个蚂蚁留在矩形 R_i 上的信息素总和, span 表示一个时间段。

(3) 局部信息素修正规则

$\text{pheromone}_k(R) = (1 - 0.9 \cdot \text{order}[k/n]) \cdot \text{Unit} \cdot (1/\text{Area}(R)) + [1 - 0.5(1/m)] \cdot \text{pheromone}_k(R)$

其中, $[1 - 0.5(1/m)] \cdot \text{pheromone}_k(R)$ 是为了使蚂蚁算法跳出局部最优解陷阱而采取的措施。

(4) 蚂蚁数量的选择: 文献^[9]经过复杂计算和部分经验, 得出最佳的蚂蚁数量是 10。

(5) 目标函数: 本文中目标函数的定义为 $f(x) = \text{Area1}/\text{Area2}$, 其中 Area1 为要排放的矩形总面积, Area2 为排样图最大高度以下的板材面积。

2.4 求解算法及步骤

由于蚂蚁算法是一种正反馈算法, 在算法收敛速度较快的同时, 也容易陷入局部最优解。本文在用蚂蚁算法求解矩形排样问题时, 为了减少蚂蚁算法在求解过程中陷入局部最优陷阱的可能性, 采取如下措施: 当一个矩形被一只蚂蚁访问时, 一方面蚂蚁留下了一定量的信息素; 另一方面, 除了挥发会使矩形上的信息素减少外, 被蚂蚁访问过的矩形的信息素还按照一定比例自动减少。即使某个矩形上最初信息素水平较高, 几只蚂蚁经过后, 它的信息素浓度就会有一定程度的下降, 使该矩形在后面的搜索过程中对其它蚂蚁的吸引力相对减弱, 其它蚂蚁就有更大的概率去搜索其它路径, 从而避免搜索过度集中而导致搜索停滞, 使蚂蚁算法有更大的几率搜索到问题的全局最优解。

为了算法说明的方便, 先对有关参数进行说明。 n 表

示要排放的矩形总数, NC 为迭代次数计数器, tabu 列表中保存的是已经被蚂蚁访问过的矩形, count 是 tabu 列表的计数器。

用蚂蚁算法求解矩形件排样问题的具体步骤:

(1) 初始化相关参数, 计算出每个矩形上的信息素浓度初始值。

(2) 按照一定策略确定每个蚂蚁访问的第一个矩形, 将每个蚂蚁的出发矩形加入到 $\text{tabu}_k(\text{count})$ 列表中, $\text{tabu}_k(\text{count})$ 表示第 k 只蚂蚁已经访问过的 count 个矩形, 对每个蚂蚁访问过的出发矩形进行信息素的局部修正。

(3) 循环执行 Operation1 操作 m 次, 为每个蚂蚁找出各自的矩形访问序列:

{ Operation1: 循环执行下列操作 $n-1$ 次, 为蚂蚁 k ($1 \leq k \leq m$) 找出剩余 $n-1$ 个矩形的访问序列, 按转移概率 $\text{JumpProbability}_k(\text{current}, \text{next})$ 从还没有访问过的矩形中选择出下一个要访问的矩形 next (如果转移概率相同的矩形有若干个, 那么就选取编号最小的那个); 将第 k 只蚂蚁移动到矩形 next, 然后将矩形 next 加入到 $\text{tabu}_k(\text{count})$ 列表中, count 的值增加 1; 对矩形 next 上的信息素 $\text{pheromone}(\text{next})$ 进行局部修正。 }

(4) 记录下 m 个蚂蚁所访问过的矩形序列, 根据产生的矩形序列按基于最低水平线的搜索算法排放矩形, 计算出对应排样图的材料利用率, 根据板材利用率最高的矩形访问序列, 得出本次迭代找到的矩形最佳访问序列。

保存当前找到的矩形最佳排放顺序和每个矩形排放的位置; 判断当前 m 只蚂蚁搜索到的解中是否存在比目前最优解还要好的解, 如果存在的话则保存这个最优解, 否则执行 5)。

(5) 根据等式 $\text{pheromone}_i(t + \text{span}) = \text{Last} \cdot \text{pheromone}_i(t) + \text{pheromone}_i(k)$, 为每个矩形进行全局信息素修正, 其中 span 表示一个时间段, $\text{pheromone}_i(k)$ 表示 m 个蚂蚁访问矩形 R_i 后给矩形 R_i 带来的信息素增量。

(6) $\text{NC} = \text{NC} + 1$, 如果 NC 小于最大迭代次数, 那么清空所有的 tabu 列表, 重新初始化相关参数, 转到 2); 如果迭代次数已经达到规定值, 则输出当前找到的最佳排样方案, 结束。

3 算例分析

根据以上算法, 笔者已经开发出了基于蚂蚁算法的矩形件优化排样系统, 其中蚂蚁算法的运行参数设置为: 蚂蚁个数 $m=10$, 矩形面积大小对蚂蚁选择影响程度的系数 $\text{Importance}=2$, 最大迭代次数 $\text{MaxNC}=10$, 权重系数 $\text{WeightArea}=0.1$, $\text{WeightRatio}=0.9$, 信息素的持久性系数 $\text{Last}=0.5$, 标准的信息素单位 $\text{Unit}=100$ 。

为了便于和其它启发式算法进行比较, 笔者根据基于最低水平线的搜索算法, 也开发出了基于基本遗传算法的矩形件优化排样系统。在遗传算法的运行参数中, 群体规模为 10, 进化代数数为 10 代, 变异概率为 0.6, 交叉概率为 0.5。

蚂蚁算法和遗传算法由于其概率性, 对同一数据, 每

次运行结果都不一定相同。为了对算法的性能进行评价,对同一问题采用多次运行,取多次算结果平均值来对算法的性能进行比较,本文中取 10 次适应度的平均值来进行对比。板材的规格为 500×400 矩形数据如下:

表 1 矩形数据

编号	1	2	3	4	5	6	7	8	9	10	11	12
长度	25	18	79	121	29	64	36	48	11	46	55	87
宽度	36	24	84	30	48	98	21	59	17	121	22	41
个数	4	5	3	4	11	2	2	3	2	2	1	2

编号	13	14	15	16	17	18	19	20	21	22	23	24
长度	39	31	78	41	19	63	10	50	100	50	150	40
宽度	72	25	24	65	11	36	30	61	20	30	50	20
个数	2	2	3	2	2	2	3	2	2	2	1	2

表 2 蚂蚁算法求出的板材利用率

编号	1	2	3	4	5	6	7	8	9	10
板材利用率	0.8679	0.8475	0.8525	0.8627	0.8812	0.8839	0.8627	0.8705	0.8785	0.8117

表 3 遗传算法求出的板材利用率

编号	1	2	3	4	5	6	7	8	9	10
板材利用率	0.7603	0.8049	0.8475	0.8071	0.8280	0.8094	0.8163	0.7746	0.8304	0.8186

表 4 遗传算法和蚂蚁算法的结果对比

	板材利用率最高值	最低值	平均值	所用时间
遗传算法	0.8475	0.7603	0.8097	1 s
蚂蚁算法	0.8839	0.8117	0.8620	3 s

图 1 和图 2 分别是基于蚂蚁算法的排样系统和基于基本遗传算法的排样系统在 10 次运行中找到的最优排样图。注:矩形左上角标注的数字表示矩形排放的先后顺序。以上两个排样系统的运行环境:操作系统 Windows2000, Intel CPU 2.00G, 内存 256M。

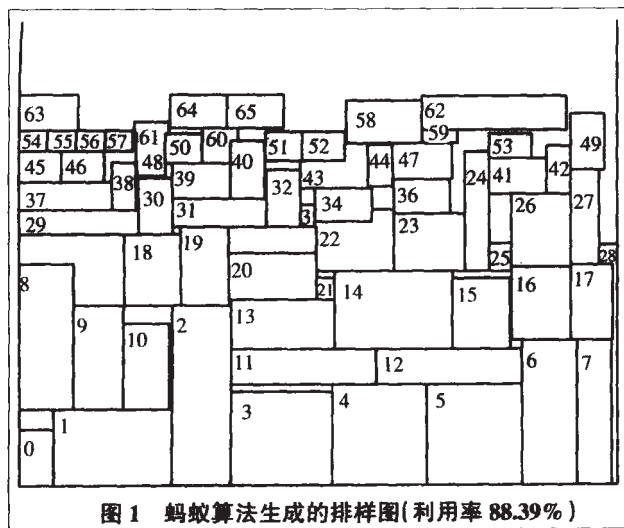


图 1 蚂蚁算法生成的排样图(利用率 88.39%)

4 结束语

蚂蚁算法这种来自大自然的随机搜索寻优方法,目前已在许多方面表现出相当好的性能,蚂蚁算法的出现为许多困难问题的求解提供了新的途径。在过去的短短几年时间里蚂蚁算法的应用领域不断扩大。

用启发式算法求解 NP 难题时,算法的效率是决定算法能否推广应用的一个重要因素。蚂蚁算法隐含的并行

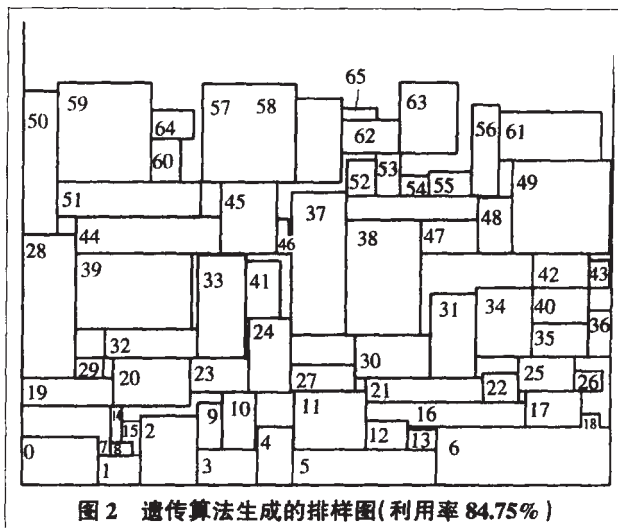


图 2 遗传算法生成的排样图(利用率 84.75%)

性使之具有极强的发展潜力。蚂蚁算法作为一种新型的模拟进化算法,其研究刚刚开始,理论还不是很成熟,有很多不完善的地方有待改进。相信随着研究的深入,蚂蚁算法也将同其它模拟进化算法一样,获得越来越广的应用。

本文尝试用蚂蚁算法来求解矩形件优化排样问题,试验结果证明了用蚂蚁算法求解矩形件优化排样问题的可行性和有效性。在用蚂蚁算法来求解矩形件优化排样问题时,蚂蚁算法的运行参数对求解结果和求解效率有一定的影响,但目前尚无合理选择它们的理论依据,实际应用中需要针对具体情况,根据多次试验结果来确定它们的合理取值。如何才能使蚂蚁算法求解矩形排样问题的效果更好,有待于作进一步的深入研究。

[参考文献]

- [1] Colomi A, Dorigo Mand Maniezzo V. Distributed optimization by ant colonies[A]. In: Proc. of 1st European Conf. Artificial Life C]. Paris, France: Elsevier, 1991. 134- 142
- [2] 马良, 王龙德. 背包问题的蚂蚁优化算法[J]. 计算机应用 2001 21 (8): 4- 5.
- [3] 孙新宇. 基于蚂蚁算法的工件排序优化[J]. 系统工程理论与实践 2003 (11) 20- 24.
- [4] 赵中凯, 梅国建. 基于混合蚂蚁算法的二维装箱问题求解[J]. 计算机应用 2004 24(6) 297- 298.
- [5] 龚志辉. 基于遗传算法的矩形件优化排样系统研究[D]. 长沙: 湖南大学 2003 24(6) :14- 15.
- [6] 计华. 一种基于四叉树结构的排料算法[J]. 计算机工程, 2003 29(9) 80- 82.
- [7] Marco Dorigo, Eric Bonabeau, guy Theraulaz. Ant algorithms and stigmergy[J]. Future Generation Computer System 2000, (1): 851- 871.
- [8] Marco Dorigo and Luca Maria Gambardella. Ant Colony System: A Cooperative Learning Approach to the Salesman Traveling Problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 11 (1): 53- 66.

(编辑 昊天)

作者简介: 黄红兵(1974-)男,硕士,研究方向为 CAD/CAM。
收稿日期 2005-09-19