

# 一维下料问题的改进自适应遗传算法

魏凉良 叶家玮

(华南理工大学交通学院, 广东 广州 510640)

**摘 要:** 提出采用改进的自适应遗传算法与求解此类问题的一般算法——BFD 算法相混合, 构成一种改进的自适应混合遗传算法(MAHGA). 并将其与另一种新的求解此类问题的 EPFF 算法进行了精度、速度等方面的比较. 结果表明, 本文方法可以取得较为满意的效果. 文章最后给出了一个实例.

**关键词:** 一维下料; 改进自适应遗传算法; EPFF 算法

**中图分类号:** U 671.2

**文献标识码:** A

在某些以条状型材为原材料的生产部门, 经常遇到如下形式的下料问题: 现有原材料  $k$  根, 原料长度为  $L$ , 要求截成  $m$  种长度不同并满足需求量的零件; 零件长度为  $l_1, l_2, \dots, l_m$ ; 需求量为  $n_1, n_2, \dots, n_m$ . 问应采取什么样的下料方案, 使得材料利用率最大, 从而达到减少材料损失, 降低成本, 提高经济效益的目的, 这就是所谓一维最优下料问题.

以往有不少人研究过这个问题. 当下料的成品品种数较少时, 问题不难解决; 当原料品种数和成品品种数均较大时, 问题则相当复杂, 人工建模也很困难. 本文先列出求解这类问题的一般算法和 EPFF 算法, 然后提出一种改进的混合遗传算法, 并进行计算比较. 结果表明, 本文提出的算法是解决该类问题的又一较好的方法.

## 1 求解下料问题的一般算法

一维下料问题可建模为装箱问题. 装箱问题的一般提法如下: 设有  $n$  个物品  $u_1, u_2, \dots, u_n$  要装箱发送, 已知每个物品的体积为  $V(i)$ ,  $V(i) \in (0, 1)$ ,  $i = 1, 2, \dots, n$ , 现规定每个箱子的容量为 1, 问如何确定装箱方案可使得装完这  $n$  个物品所需要用的箱子数目为最小? 在本问题中, 原材料即可看做箱子, 所

需零件即物品. 从计算复杂性来讲, 这是一个 NP(Nondeterministic Polynomial) 难题, 很难精确求解. 通常的算法有 3 种<sup>[1]</sup>: 1) FF(First Fit) 近似算法, 将  $n$  个物品  $u_1, u_2, \dots, u_n$  顺次装箱, 设箱的次序为  $B_1, B_2, \dots, B_n$ , 对某一物品  $u_i$ , 它总是被装到第一个能装下它的箱子中; 2) BF(Best Fit) 近似算法, 将物品顺次装箱, 要求每个物品装进箱子后留下的空隙最小; 3) FFD(First Fit Decreasing) 或 BFD(Best First Decreasing) 近似算法, 如果在装箱之前先把他们按体积大小降次排列, 再利用 FF(或 BF) 近似算法就得到 FFD(或 BFD) 近似算法. 近似算法的求解结果与给定数据有很大关系, 效果往往不太理想.

## 2 求解下料问题的 EPFF 算法

EPFF 算法<sup>[2]</sup> 是求解装箱问题的一种较新的算法. 它将所有的箱子分成 8 组, 将实数列中的元素分成 8 类, 称  $(2/3, 1]$ ,  $(7/12, 2/3]$ ,  $(1/2, 7/12]$ ,  $(5/12, 1/2]$ ,  $(1/3, 5/12]$ ,  $(1/4, 1/3]$ ,  $(1/5, 1/4]$ ,  $(0, 1/5]$  上的元素分别为  $\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3, \beta_4, \gamma$  类元素,  $\alpha_1, \alpha_2, \alpha_3$  统称为  $\alpha$  类元素, 同时用  $a, a_1, a_2, a_3, b_1, b_2, b_3, b_4, r$  分别表示  $\alpha, \alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3, \beta_4, \gamma$  类元素的个数, 因此有  $a = a_1 + a_2 + a_3$ . EPFF 算法步骤如下: (1) 从  $\beta_2, \beta_3, \beta_4$  类元素中取出  $a_3$  个元素与  $\alpha_3$  类元素放在一起 (或  $a_3 > b_2 + b_3 + b_4$ , 则取完所有的  $\beta_2, \beta_3, \beta_4$  类元素为止), 再从  $\beta_3, \beta_4$  类元素中取出  $a_2$  个元素与  $\alpha_2$  类元素放在一

收稿日期: 2002-10-23

作者简介: 魏凉良(1979-), 女, 硕士研究生, 主要从事计算机辅助船舶设计、制造及人工智能技术应用研究.

起;(2)将 $\alpha_1$ 类元素装入第1组箱子中,每个箱子中装入1个元素;(3)将 $\alpha_2$ 类元素及取出的 $\beta_3, \beta_4$ 类元素装入第2组箱子中,每个箱子最多装一个 $\alpha_2$ 类元素和一个 $\beta_3$ 或 $\beta_4$ 类元素;(4)将 $\alpha_3$ 类元素及取出的 $\beta_2, \beta_3, \beta_4$ 类元素装入第3组箱子中,每个箱子最多装一个 $\alpha_3$ 类元素和一个 $\beta_2$ 或 $\beta_3$ 或 $\beta_4$ 类元素;(5)将 $\beta_1$ 类元素及剩余的 $\beta_2, \beta_3, \beta_4$ 类元素分别装入第4~7组箱子中,各组箱子分别装2,2,3,4个 $\beta_1, \beta_2, \beta_3, \beta_4$ 类元素;(6)将 $\gamma$ 类元素按FF算法装入第8组箱子中。

### 3 求解下料问题的改进自适应遗传算法

#### 3.1 算法思想

遗传算法是一种基于自然选择和群体遗传机理的搜索算法,它模拟了自然选择和自然遗传过程中发生的繁殖、杂交和突变现象。在利用遗传算法求解问题时,问题的每个可能的解都被编码成一个“染色体”,即个体,若干个个体构成了群体(所有可能解)。在遗传算法开始时,总是随机地产生一些个体(即初始解),根据预定的目标函数对每个个体进行评价,给出一个适应度值。基于此适应度值,选择个体用来复制下一代。选择操作体现了“适者生存”原理,“好”的个体被选择用来复制,“坏”的个体则被淘汰。然后选择出来的个体经过交叉和变异算子进行再组合生成新一代。这一群新个体由于继承了上一代的一些优良性状,因而在性能上要优于上一代,这样逐步朝着更优解的方向进化。因此,遗传算法可以看做是一个由可行解组成的群体逐代进化的过程。

遗传算法由于其运算简单和解决问题的有效能力而被广泛应用到众多的领域。理论上已经证明,遗传算法能从概率的意义上以随机的方式寻求到问题的最优解。但另一方面,实践表明遗传算法的应用也会出现一些不尽人意的的问题,这些问题中最主要的是它容易产生早熟现象及局部搜索能力差等。要解决这些问题,就要对基本的遗传算法本身进行调整改进。此外,将遗传算法与问题的特有知识集成到一起所构成的混合遗传算法,有可能产生求解性能较佳的方法。在求解一维下料问题时,本文采用一种基于个体适应度值交叉率和变异率自适应调整的遗传算法,同时,引入BFD算法的思想,修正无效的染色体,提高了算法的运行效率和解的质量。

#### 3.2 遗传算法的基本构成

遗传算法的形式化定义为<sup>[3]</sup>:

$$GA = (C, E, P_0, M, \Phi, \Gamma, \Psi, T)$$

式中: $C$ 为个体的编码方法; $E$ 为个体适应度评价函数; $P_0$ 为初始群体; $M$ 为群体大小; $\Phi$ 为选择算子; $\Gamma$ 为交叉算子; $\Psi$ 为变异算子; $T$ 为遗传运算终止条件。

#### 3.3 求解一维下料问题的 MAHGA

##### 3.3.1 染色体编码方法

假设原材料编号依次是 $A_1, A_2, \dots, A_k$ ,各个零件编号依次是 $u_1, u_2, \dots, u_N$ ,其中零件 $u_1$ 的尺寸为 $l_1$ , $u_2$ 的尺寸为 $l_2$ ,其余依次类推可得其它尺寸的零件编号,将各个零件所在原材料的编号依次排列就构成该问题的染色体编码。

本文的编码使用的是等长度的字符代码编码方法,符号集为数字序号表,如 $\{1, 2, \dots, k\}$ 。初始群体由 $1, 2, \dots, k$ 的随机排列产生。由个体的染色体码串即可统计出下料方案。例如 $[6, 3, 2, \dots, 3, 8]$ 就代表一种下料方案(群体中的一个个体),表示从6号原料中截取第1个零件,从3号原料中截取第2个零件等等。

##### 3.3.2 目标函数和适应度函数

按照本问题的目标要求应使材料利用率最大,工程上通常希望的结果是:对于一批给定的原料和待下料对象,下料后的材料中除一根材料具备最长余料外(便于随后的进一步利用),其它被下料材料的余料总长度最小,且完全未使用的原料尽可能多。换言之,即将一已用材料剩余量的最大值,连同完全未使用材料的长度总和相加作为优化目标函数。由此可表达为下述目标函数<sup>[4]</sup>:

$$f(x) = \max(L - \sum_{j=1}^m l_j a_{ij} + k' L) \quad (1 \leq i \leq k).$$

式中: $k'$ 为完全未使用的原料数量; $a_{ij}$ 为第 $i$ 根原材料上第 $j$ 种零件的数量; $L$ 为原材料长。

为满足适应度函数的非负性,采用如下方法:

$$F(X) = \begin{cases} f(X), & f(X) > 0 \\ 0, & f(X) \leq 0 \end{cases}$$

对给定原料数量 $k$ 的取值,若 $k$ 过大将直接导致搜索空间的扩大,不利于搜索效率的提高;但 $k$ 过小又易导致运算错误。应先估计一个有富余的原料数量,可根据EPFF算法的思想进行估计。公式如下:

$$\begin{cases} k = a_1 + a_2 + a_3 + b_1/2 + b_2/2 + b_3/3 + b_4/4 + r/x \\ x = L/L_{\max} \end{cases}$$

式中: $L_{\max}$ 为 $\gamma$ 类元素中的最大长度。将公式中所有

小数部分取整。

### 3.3.3 遗传算子的选择

选择算子采用比例选择方法,同时使用最优保存策略。经验证,使用保存最佳个体策略的遗传算法总能够以概率1搜索到最优解。交叉算子采用单点交叉算子。变异算子采用编码字符集 $\{1, 2, \dots, k\}$ 范围内的均匀随机变异,并且随着所需材料数量的变化修改 $k$ 值。

### 3.3.4 运行参数的选择

(1) 长度 $l$ ; (2) 群体大小 $M$ : 群体中所含个体数量; (3) 交叉概率 $P_c$ 及变异概率 $P_m$ :  $P_c$ 和 $P_m$ 的选择是影响遗传算法行为和性能的关键所在,直接影响算法的收敛性。将 $P_c$ 和 $P_m$ 定为确定值,效率不高,并且存在“早熟”的可能性。本文根据自适应遗传算法的思想,采用改进的自适应的 $P_c$ 和 $P_m$ ,其目的是进一步抑止早熟现象发生和提高优化能力。其公式如下<sup>[5]</sup>:

$$P_c = \begin{cases} P_{c1} - \frac{P_{c1} - P_{c2}}{f_{\max} - f_{\text{avg}}} \cdot (f' - f_{\text{avg}}), & f' \geq f_{\text{avg}} \\ P_{c1}, & f' < f_{\text{avg}} \end{cases}$$

$$P_m = \begin{cases} P_{m1} - \frac{P_{m1} - P_{m2}}{f_{\max} - f_{\text{avg}}} \cdot (f_{\max} - f), & f \geq f_{\text{avg}} \\ P_{m1}, & f < f_{\text{avg}} \end{cases}$$

式中: $f_{\max}$ 代表群体中最大的适应度值; $f_{\text{avg}}$ 代表每代群体的平均适应度值; $f'$ 代表要交叉的两个个体中较大的适应度值; $f$ 代表变异个体的适应度值; $P_{c1} = 0.9$ ;  $P_{c2} = 0.6$ ;  $P_{m1} = 0.1$ ;  $P_{m2} = 0.001$ 。

### 3.3.5 引入BFD思想

由于初始群体和进化过程中会产生一些无效染色体,这些无效染色体所表示的下料方案中,会有部分原材料上的零件长度之和超过原料的长度,从而使得运算效率降低,也会导致不太好的运算结果。本文引入BFD近似算法的思想,修正无效染色体为合理的染色体,大大提高遗传算法的局部搜索能力,从而获得一个可行且性能较为优良的新群体。这个具有BFD思想的解码过程可描述如下:

(1) 整理所用原材料号数。将未用原材料号删除,已用原材料号按 $1, 2, \dots, m$ 顺序排列, $m$ 为实际使用的原材料数;

(2) 查询原材料 $A_i$ 上的零件号;

(3) 若 $A_i$ 上的零件长度之和超出原材料长度 $L$ ,则将其长度最小的零件取出放入临时存放区;

(4) 重复步骤(3)直至 $A_i$ 上的零件长度之和不大于 $L$ ;

(5) 重复步骤(2)~(4),直至 $i = m$ ;

(6) 将临时存放区中的零件按布置后总长不超出 $L$ 且空隙最小的原则布置在 $A_i$ 上( $i = 1, 2, \dots, k$ ),布置不下的零件取出且使 $m = m + 1$ ;

(7) 计算目标函数值 $f(x)$ 。

当所需原料数量变化时应对 $k$ 值进行修改。

### 3.3.6 对进化完成后方案的改进

一般最后一代的适应度最高的个体即为最好的解决方案。工程上往往需要其中一根原料有更大的余料以便继续使用,这样可以提高综合效益,对该解决方案进行以下操作:找出其中余料最大的那根原料 $A_x$ ,将其中的零件与其它原料 $A_i$ ( $i = 1, 2, \dots, k$ ;  $i \neq x$ )上的零件交换,使得满足交换后 $A_x$ 的余料更大且 $A_i$ 上的零件长度之和不超过 $L$ 。经此操作即可得到问题的最终解。

### 3.3.7 算法流程

本算法的流程图如图1所示。

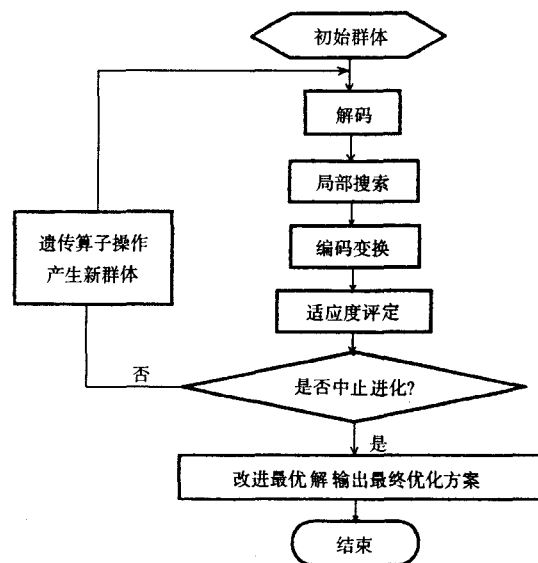


图1 MAHGA流程图

Fig. 1 The flow chart of MAHGA

## 4 算例

有一工程项目,原材料规格为12 m,所需角钢的长度和数量如表1所示。

EPFF算法求解得到的下料方案如表2所示。

从表2中可知,共需型材25根,平均利用率为93.51%。由于零件数量和长度的组合与型材规格和数量的组合通常不会“匹配”成充分利用材料的排样结果,且一般有一根表现为含有较多的余料,表2

中编号为25的型材即为此例.若不计编号为25的型材不“匹配”的讨论(事实上该型材可在下一轮的排样中获得进一步利用),则余下24根材料平均利用率为96.57%.表3给出了使用MAHGA的一次终止代数数为300、群体数为50的计算结果.

表1 角钢的尺寸和数量

Table 1 Size and amount of angle iron

长度/mm	数量/个	长度/mm	数量
3 010	6	1 832	9
2 980	8	1 786	9
2 850	5	1 762	3
2 505	10	1 758	5
2 450	4	1 640	8
2 245	4	1 508	6
2 108	10	1 442	9
1 950	8	1 386	18
1 880	3	1 350	7
1 866	6	1 206	9

表2 由EPFF算法获得的下料方案

Table 2 Cutting solution got by EPFF algorithm

编号	零件长度/mm,数量/个 (括号内为数量)	余料/mm	利用率/%
1	3 010(3),2245(1)	725	93.96
2	3 010(3),2 245(1)	725	93.96
3	2 980(4)	80	99.33
4	2 980(4)	80	99.33
5	2 850(4)	600	95.00
6	2 850(1),2 505(3),1 508(1)	127	98.94
7	2 505(4),1 950(1)	30	99.75
8	2 505(3),2 450(1),1950(1)	85	99.29
9	2 450(3),2 245(2)	161	98.67
10	2 108(5),1 442(1)	18	99.85
11	2 108(5),1 442(1)	18	99.85
12	1 950(6)	300	97.50
13	1 880(3),1 866(3)	763	93.65
14	1 866(3),1 832(3)	906	92.45
15	1 832(6)	1 008	91.60
16	1 762(3),1 758(3),1 386(1)	55	99.55
17	1 758(2),1 786(4),1 206(1)	135	98.88
18	1 786(5),1 640(1),1 386(1)	44	99.63
19	1 640(7)	521	95.67
20	1 508(5),1 442(3)	134	98.88
21	1 442(4),1 386(4)	689	94.27
22	1 386(8)	913	92.40
23	1 386(4),1 350(4)	1 057	91.20
24	1 350(3),1 206(6)	715	94.05
25	1 206(2)	9 588	20.01

表3 由MAHGA获得的下料方案

Table 3 Cutting solution got by MAHGA

编号	零件长度/mm,数量/个 (括号内为数量)	余料/mm	利用率/%
1	3 010(1),1 832(1),1 508(1),1 442(2), 1 386(1),1 350(1)	30	99.75
2	2 505(2),1 950(1),1 866(1),1 786(1), 1 386(1)	2	99.98
3	3 010(1),2 505(1),2 245(1),2 108(1), 1 950(1)	182	98.48
4	2 245(1),2 108(1),1 880(2),1 832(2)	223	98.14
5	1 866(1),1 640(2),1 442(2),1 386(1), 1 350(1),1 206(1)	28	99.77
6	3 010(1),1 950(2),1 832(1),1 786(1), 1 206(1)	266	97.78
7	2 980(1),2 245(1),1 832(1),1 786(1), 1 640(1),1 508(1)	9	99.93
8	2 450(1),1 832(1),1 758(1),1 786(1), 1 508(1),1 442(1),1 206(1)	18	99.85
9	2 450(1),1 866(1),1 786(2),1 386(1), 1 350(2)	26	99.78
10	3 010(2),2 505(1),2 108(1),1 206(1)	161	98.66
11	2 850(1),2 505(1),2 450(1),1 758(1), 1 206(2)	25	99.79
12	2 980(2),2 505(1),2 108(1),1 386(1)	41	99.66
13	2 108(2),1 762(1),1 640(1),1 508(2), 1 350(1)	16	99.87
14	2 980(1),2 245(1),2 108(1),1 950(1), 1 386(1),1 206(1)	125	98.96
15	2 980(2),2 850(2)	340	97.17
16	1 386(2),1 640(1),1 350(1),1 206(1)	5 032	58.07
17	2 850(1),2 108(1),1 950(1),1 832(1), 1 786(1),1 386(1)	88	99.27
18	1866(2),1 832(1),1 762(1),1 758(1), 1 442(2)	32	99.73
19	2 505(1),2 108(1),1 786(1),1 640(1), 1 386(1),1 350(1),1 206(1)	19	99.84
20	3 010(1),2 980(2),2 505(1)	525	95.63
21	2 850(1),2 505(1),1 950(1),1 880(1), 1 386(2)	43	99.64
22	2 450(1),1 762(1),1 786(1),1 640(1), 1 508(1),1 386(2)	82	99.32
23	2 505(1),1 758(2),1 640(1),1 442(1), 1 386(2)	125	98.96
24	2 108(1),1 950(1),1 866(1),1 832(1), 1 442(1),1 386(2)	30	99.75

表3显示共需型材24根,平均利用率为97.41%.若不计编号为16的型材可能因零件数量与长度的

组合与型材规格不“匹配”的讨论,则余下23根材料平均利用率高达99.12%。计算结果较理想。

就计算速度而言,在PⅢ500微机上运行上述算例,EPFF算法计算时间仅需1s,而MAHGA的终止代数数为300、群体数为50的计算时间大于20s。对于工程而言,上述算例计算时间均在可接受的范围。

## 5 结 论

对于求解一维下料问题的近似算法,实际计算显示EPFF算法执行时间短,计算速度快,求解精度较高,但求解结果与给定数据有很大关系。

计算表明,本文给出的遗传算法受数据结构影响较小,求解精度进一步提高。该计算模型也比一般遗传算法收敛速度快。

## 参考文献:

- [1] 谢政, 李建平. 网络算法与复杂性理论 [M]. 长沙: 国防科技大学出版社, 1995.
- [2] 冯晓慧, 李菊娥, 任春丽. 装箱问题的一种新算法及其性能比的证明 [J]. 西安电子科技大学学报, 1998, 25(2): 231 - 233.
- [3] 周明, 孙树栋. 遗传算法原理及应用 [M]. 北京: 国防工业出版社, 1999.
- [4] 李培勇, 王全华, 裘泳铭. 型材优化下料的混合遗传算法 [J]. 上海交通大学学报, 2001, 35(10): 1 557 - 1 560.
- [5] 段玉倩, 贺家李. 遗传算法及其改进 [J]. 电力系统及其自动化学报, 1998, 10(1): 39 - 52.

# Modified Adaptive Genetic Algorithm for One-dimensional Cutting Problem

Wei Liang-liang Ye Jia-wei

(College of Traffic and Communications, South China Univ. of Tech., Guangzhou 510640, China)

**Abstract:** This paper presents a modified adaptive genetic algorithm hybridized with BFD algorithm (MAHGA). A comparison between the MAHGA and a new good method such as EPFF algorithm is made in terms of calculation speed and precision. The result shows that the MAHGA can obtain satisfying effect. One sample is also presented to show the result.

**Key words:** one-dimensional cutting; modified adaptive genetic algorithm; EPFF algorithm

## “仿生型生物活性人造齿根”通过成果鉴定

2003年4月29日,华南理工大学材料科学与工程学院王迎军教授主持完成的“仿生型生物活性人造齿根”项目通过广东省科技厅组织的成果鉴定会。该项目设计并制备了仿生型生物活性人造齿根,实施了合理的涂层晶化工艺,成功地使涂层表面羟基磷灰石(HA)含量达98%,结晶度达82.6%,使涂层残余热应力得到缓解和消除,涂层结合强度达到50MPa以上,明显高于国内外同类材料的相应指标。所制备的涂层具有从生物稳定至生物活性、由致密到多孔的双梯度结构,并与信号因子结合进行生物组装,能作用于靶细胞,从而使人造齿根具有诱导成骨功能,能得到牢固的初始固位并有良好的远期效果。动物实验和临床试用结果表明,仿生型生物活性人造齿根具有良好的生物活性和成骨诱导作用,其涂层与金属基体及骨结合界面具有高的结合强度。鉴定委员会一致认为该项成果达到了国际领先水平。