

文章编号: 0253-987X(2002)09-0967-04

一维下料方案的遗传算法优化

贾志欣, 殷国富, 胡晓兵, 舒斌

(四川大学制造科学与工程学院, 610065, 成都)

摘要: 在对一维下料方案数学模型分析的基础上, 提出了一种基于遗传算法的求解方法. 主要思想是把零件的一个顺序作为一种下料方案, 并视作组合优化问题来求解. 在求解过程中, 给出了应用遗传算法求解关键问题的编码、解码方法、遗传算子及适应度函数的定义, 并根据这一算法开发出一维下料方案的优化系统. 实际应用表明, 采用该方法求解一维下料方案, 可提高材料的利用率, 而且还可以提供多个优化方案.

关键词: 一维下料; 遗传算法; 优化

中图分类号: TH122; TP391.72 **文献标识码:** A

Optimization for One-Dimensional Cutting-Stock Problem Based on Genetic

Jia Zhixin, Yin Guofu, Hu Xiaobing, Shu Bin

(School of Manufacturing Science and Engineering, Sichuan University, Chengdu 610065, China)

Abstract: The one-dimensional cutting problem occurs in many industry processes. Based on genetic algorithm, method for one-dimensional cutting-stock problem is presented after analyzing the mathematical model. The main idea is to change the solution to an permutation and solve it as a combinative problem. The key methods for applying genetic algorithm, coding, decoding, genetic algorithm operators and fitness definition are given. An optimal system for one-dimensional cutting stock problem is developed based on the method. Result shows that the algorithm is valid and efficient.

Key words: one-dimensional cutting-stock problem; genetic algorithm; optimization

一维下料方案的优化是讨论材料不同长度的组合搭配, 以使材料的利用率最高. 这个优化问题在型材、棒材和管材、金属结构材料、建筑材料、甚至布料下料中广泛存在, 因此人们提出了多种优化方法^[1~5]. 本文讨论了将遗传算法用于一维下料方案优化的求解问题.

1 一维下料方案优化问题的数学模型

1.1 问题的定义

通常在实际生产中, 原材料的长度 L 是给定的, 现使用这种原材料进行下料, 将其切割成 n 种规格的零件, 每种零件的长度为 l_i , 需要的数量为 d_i 个 ($i = 1, 2, \dots, n$, 且 $n \geq 2, l_i \leq L$). 如何下料才能使用料最省, 这就是一维下料方案的优化问题.

1.2 数学模型

建立该问题的数学模型可以分解为2步:首先找到这些零件的组合方式,即哪几个零件组合使得每一个原材料的余料(料头)最小,假设有 m 种下料方式;根据每种零件的需求量,求取每一种下料方式应用的次数。

对第 j 步,有 m 种下料方式。第1种方式是在第1种零件下为 a_{11} 个,第2种零件下为 a_{21} 个,依次类推,第 n 种零件下为 a_{n1} 个,余料长度为 b_1 ;第 m 种下料方式是在零件1下为 a_{1m} 个,零件2下为 a_{2m} 个,依次类推,零件 n 下为 a_{nm} 个,余料长度为 b_m 。 a_{ij} 表示用第 j 种下料方式所能得到的第 i 种零件的个数。

由上面的下料方案模型,可得到如下对应的系数矩阵

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \quad (1)$$

设 x_j 为使用第 j 种下料方案需要切割的原材料的数目,即第 j 种下料方式应用的次数,则使用 m 种不同的下料方式需要切割原材料的根数为

$$X = (x_1, x_2, \dots, x_m)^T \quad (2)$$

利用式(1)能够很方便地求出每种下料方式所产生的废料长度,即

$$b_j = L - (a_{1j}l_1 + a_{2j}l_2 + \dots + a_{nj}l_n) \\ j = 1, 2, 3, \dots, m$$

式中: a_{1j} 、 a_{2j} 、 a_{nj} 为第 j 种下料方案得到的第1种、第2种、第 n 种零件的数目。

利用式(2),使得 $b_1x_1 + b_2x_2 + \dots + b_jx_j + \dots + b_mx_m$ 尽可能地小,而优化的目标应使式(2)的解对于给定的下料模式所产生的废料为最少,即材料利用率最高。因此,将每种下料方案产生的 b_j 作为目标函数的系数,可使目标函数为最小值,即所产生的废料总长度最短。综上所述,优化下料的数学模型为

$$\text{目标函数} \quad \min_{j=1}^m b_j x_j \quad (3)$$

$$\left. \begin{aligned} \text{约束条件} \quad & \sum_{j=1}^m a_{ij} x_j = d_i \quad (i = 1, 2, \dots, n) \\ & x_j \geq 0 \quad (j = 1, 2, \dots, m) \end{aligned} \right\} \quad (4)$$

1.3 切缝问题

在实际生产中,有的切割方法会产生切缝,如气

割、火焰切割、锯等,而不同的切割方法所产生的切缝宽度不同。有些切割方法(如气割等)会使切割后的零件切割处发生诸如退火等现象,影响零件的使用性能,因此需将退火部分进行处理。为了对切缝和退火部分的长度进行处理,把要切割的每一种零件的 l_i 加上切缝宽度 g 和退火部分长度 e ,即作为切割的一个零件长度 $l_i = l_i + g + e$,并用于取代原来长度为 l_i 的零件(对于精密下料,可将这2个变量赋零值)。由于最后一个零件可不计切口密度,所以式(3)中的 b_j 变为

$$b_j = L + g - (a_{1j}l_1 + a_{2j}l_2 + \dots + a_{nj}l_n) \\ \min_{j=1}^m b_j x_j \quad (5)$$

由一维下料方案的数学模型,可以看出下料方法是一个整数规划的问题。传统的下料优化方法是将整数变量进行松弛,并视为线性规划,采用单纯形方法求解,然后对结果进行调整,但这样做并不能保证得到最优解^[2,3],而且调整之后产生一种或几种零件的过量或不足量生产,因此需要进一步进行处理^[4]。采用针对整数规划求解的分支定界法求解,也只能在变量数目较少时适用(组合方式 m 小),而当原材料的 L 和所割零件的 l_i 之比大于一定数值后,就会出现可能的切割下料组合方式很多,即方案数 m 增加,因此用分支定界法也难以求解。

1.4 视作组合优化的问题

从组合优化的角度,一维下料方案的优化问题还可以视作一维装箱问题,其表述是当给定容积为 B 的箱子和一组长度为 b_i ($i = 1, 2, \dots, n$, 且 $n \geq 2$, $b_i \leq B$)的物品,并确定将这些物品全部装入所需的最小箱数,即确定一个最优的集合划分 S_1, S_2, \dots, S_k ,满足

$$b_i \leq B, \quad 1 \leq i \leq k \quad (6)$$

式中: S_j 为一个箱子内存放物品的集合。这2种模型的差别在于给定的零件是否按长度分组,以及每个切割方案(类似箱子)是否被多次应用。装箱问题是NP(Nondeterministic Polynomial)完全问题^[3~6],在多项式时间内没有最优解。本文将一维下料方案的优化问题视作组合优化问题,采用遗传算法求解。

2 一维下料方案优化问题的求解方法

2.1 解的编码方法

用遗传算法求解问题,必须先确定编码方法。不同的编码方法,遗传算子实现的方式也会不同。本文

采用十进制的编码方法,由于在实际操作中减少了调整次数,同样可以提高效率,有可能要求相同零件邻接下料.因此,编码分为2种情况:不要求同种零件相邻,将每一种零件进行编号, $i = 1, 2, \dots$,

$$n_{\text{total}}, n_{\text{total}} = \sum_{i=1}^n d_i, \text{零件编号构成一个整数串个体}$$

$P(P = \{p_1, p_2, \dots, p_3, \dots, p_i, \dots, p_{n_{\text{total}}}\}, 1 \leq p_i \leq n_{\text{total}})$, $p_1 \sim p_i$ 为一个具体的数,表示一种下料方案(一个下料的先后顺序);要求相同零件连续下料,须对零件种类号进行编码, $P = \{p_1, p_2, \dots, p_i\}, 1 \leq p_i \leq n$. 例如,零件种数 $n = 3, d_1 = 4, d_2 = 5, d_3 = 7$, 编码为 $\{2, 1, 3\}$, 表示先下 5 个 2 号零件,然后再下 4 个 1 号零件,最后下 7 个 3 号零件.

2.2 解码方法

由编码方法产生一个个体的编码后,需通过解码算法得到其所需原材料的数目,计算其适应度值后,才能对该个体进行评价.针对是否要求同种零件相邻,相应的解码算法如下.

(1) 当对同种零件分散下料时,零件编号构成的 $P = \{p_1, p_2, \dots, p_i, \dots, p_{n_{\text{total}}}\}, 1 \leq p_i \leq n_{\text{total}}$, 已知其对应的零件长度为 $\{l_1, l_2, \dots, l_i, \dots, l_{n_{\text{total}}}\}$, 按零件的序号 $i = 1, 2, \dots, n_{\text{total}}$, 依次取零件长度累加,求得所需原材料的数目.

(2) 在要求同种零件相邻的情况下,零件种类序号构成的 $P = \{p_1, p_2, \dots, p_i, \dots, p_n\}, 1 \leq p_i \leq n$ 对于 n 种规格的零件,每种零件的长度为 l_i ,需要的数量为 d_i 个 ($i = 1, 2, \dots, n$, 且 $l_i \leq L$),可按零件的种类序号和其对应的数量取零件长度累加,以求解所需原材料的根数.

2.3 适应度函数

用适应度函数来评价遗传算法时,适应度越大,解的质量越好.对一维下料问题,自然的想法是取所需原材料总根数的倒数,但若 2 个个体具有相同的根数,适应度值相同时,仍有好坏之分.例如,对于长度分别为 0.8、1.0、0.3、1.6 m 的 4 种零件,所对应的个数分别为 3、4、5、1. 在 $L = 3$ m 的原材料上进行下料时,图 1 所示的 2 种方案都需要 4 根原材料,而方案图 1a 优于方案图 1b,这是因为前者比后者的最后一根剩余材料多.因此,本文采用了如下的方法来计算适应度函数

$$f(P) = \sum_{i=1}^n l_i \cdot d_i / (L \cdot (Q_m - 1) + L)$$

式中: Q_m 为该个体 P 所需原材料的总数; L 为最后一根原材料用去的长度,其适应度值最大为 1.

共需要 4 根棒材		单位: mm			
第 1 根		0.3	1.6	0.3	0.8
第 2 根		1.0	1.0	1.0	
第 3 根		0.3	0.8	0.8	1.0
第 4 根		0.3	0.3		

(a) 下料方案 1

共需要 4 根棒材		单位: mm			
第 1 根		0.3	1.6	0.3	0.8
第 2 根		1.0	1.0	1.0	
第 3 根		0.3	0.8	0.8	0.3
第 4 根		1.0			

(b) 下料方案 2

图 1 2 种下料的方案

3 一维下料方案遗传算法的求解过程

3.1 初始种群

给定 n 种规格的零件,每种零件的长度为 l_i ,需要的数量为 d_i 个 ($i = 1, 2, \dots, n$, 且 $n > 2$). 根据是否要求同种零件相邻进行编码,随机产生由 n_{Scale} 个子种群,且长度为 n 或 n_{total} 的编码构成的初始种群,并计算每一个个体的适应度.

3.2 遗传算子

(1) 交叉算子

将父辈群体中的个体随机两两配对,进行交叉操作,产生 n_{Scale} 个子种群,并分别采用单点交叉和双点交叉算子,具体操作方法是设 2 个要交叉的个体分别为

$P = \{p_1, p_2, p_3, \dots, p_n\}, Q = \{q_1, q_2, q_3, \dots, q_n\}$
单点交叉是在 $1 \sim n$ 个范围内随机生成 1 个交叉位 b_1 ,从 P 中将位于 b_1 之前的元素拷贝至子辈个体 P_{new} 中,作为前面的元素,剩余元素按 Q 中出现的先后顺序拷贝至 P_{new} ,以同样的方法可以产生另一新个体 Q_{new} .

双点交叉是在 $1 \sim n$ 个范围内随机生成 2 个交叉位 b_1 和 b_2 ,从 P 中将位于 b_1, b_2 之间的元素拷贝至 P_{new} 中,作为前面的元素,剩余元素按 Q 中出现的先后顺序拷贝至 P_{new} ,从而产生了新个体 P_{new} ,以同样的方法产生另一新个体 Q_{new} .

(2) 变异算子

采用 2 种变异算子,一是颠倒位序变异,在 $1 \sim n$ 或 $1 \sim n_{\text{total}}$ 范围内随机产生一个整数变异位的 b_1 ,以较小的概率 p_{m1} 对子辈个体中位于 b_1 之后的

零件序列进行颠倒. 二是位置变异, 以较小的概率 p_{m2} , 在 $1 \sim n$ 个或 $1 \sim n_{\text{total}}$ 个范围内随机产生 2 个整数 b_1 、 b_2 , 并对子辈个体中的位于 b_1 、 b_2 的 2 个零件对调.

(3) 选择算子

求 n_{Scale} 个子辈个体的适应度值, 将父辈与子辈个体的适应度函数值由大到小排序, 取排在前面的 n_{Scale} 个个体作为下一代进化的父辈个体.

3.3 终止准则

重复执行上面的(1)、(2)、(3)步骤, 直到最好解的适应度值达到要求或满足了预定的进化代数, 才能停止计算, 并输出最优解.

4 应用实例与结果分析

以 AutoCAD2000 为平台, 以 ObjectARX 为开发工具, 采用上述遗传算法, 开发了一维下料优化系统, 在 AutoCAD2000 上作为一个功能模块集成, 系统的用户界面见图 2. 零件的信息输入既可以在该界面下完成, 也可在 Windows 的写字板中进行, 并以数据文件的形式导入或保存, 把优化下料的结果以图形和数据的形式显示出来(见图 3).

对于图 3 给出的数据, 当取种群规模 $n_{\text{Scale}} = 20$ 、

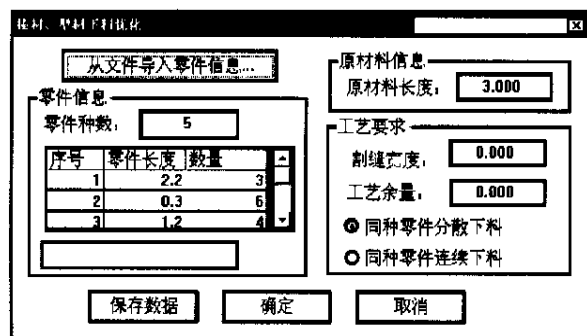


图2 系统的用户界面

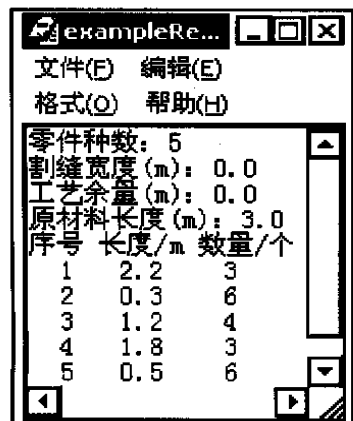


图3 问题数据

进化代数为 100、交叉概率为 1、变异概率 $p_{m1} = p_{m2} = 0.5$ 时, 在同种零件分散下料条件下运行的结果如图 4 所示. 遗传算法的优势在于对于同一个问题, 取相同的参数多次运行, 就可以得到切割顺序不同而材料利用率相近的结果(图 1 所示的 2 个方案就是 2 次运行的结果). 通过查询下料方案材料利用率值的大小, 也为用户选择最优的方案提供了方便.

图4 相同零件分散下料的优化结果

5 结论

针对一维下料问题, 提出了一种基于遗传算法的求解方法. 采用该方法求解, 材料的利用率高, 对同一问题多次运行可以给出多种利用率相近的方案, 便于从多个优化结果中择优. 因此, 简化了工程技术人员的工作, 提高了工作效率, 也节约了企业的生产成本. 该系统与 AutoCAD2000 融为一体, 具有操作方便、易于学习和掌握的优点.

参考文献:

- [1] 文祥斌. 板材下料的最优化[J]. 重型汽车, 1998, 46(6): 15~17.
- [2] 苗延义. 应用线性规划优化 C62A 敞车制动管下料[J]. 机车车辆工艺, 1999(2): 21~24.
- [3] 卢开澄. 组合数学算法与分析[M]. 北京: 清华大学出版社, 1983. 9.
- [4] Stadler H. A one-dimensional cutting stock problem in the aluminium industry and its solution[J]. European Journal of Operational Research, 1990(44): 209~223.
- [5] Vance P H. Branch-and-price algorithm for the one-dimensional cutting stock problem[J]. Computational Optimization and Applications, 1998(9): 211~228.
- [6] 邢文训, 谢金星. 现代优化计算方法[M]. 北京: 清华大学出版社, 1999. 8.

(编辑 管咏梅)