

热轧实施计划中最优倒垛问题的 整数规划模型及遗传算法¹⁾

唐立新 杨自厚

(东北大学系统工程研究所 沈阳 110006)

(E-mail: qhjytlx@mail. sy. ln. cn)

摘要 对钢铁企业板坯库中的最优倒垛问题建立了 0 和 1 整数规划模型. 这一模型是一个二次规划模型, 且目标函数的系数与变量的取值相关联, 属于 NP-难问题, 获得较大规模的最优解是不可能或非常困难. 为了求解此问题, 本文构造了改进遗传算法: (1) 提出了适合于最优倒垛问题的遗传编码, 运用此编码, 不但能够产生可行的初始染色体, 而且能够保证在交叉和变异操作后的染色体仍然可行; (2) 改进了遗传算法结构, 在新的结构中, 增加了一个培育操作, 改进了交叉操作. 通过精选随机产生的问题例子的实验显示出, 提出的算法的性能明显好于原系统的启发式算法, 最好的改进率达到 7.04%.

关键词 钢铁工业, 生产调度, 板坯倒垛, 整数规划, 改进遗传算法.

INTEGER PROGRAMMING MODEL AND MODIFIED GENETIC ALGORITHM FOR OPTIMAL TURNED-OUT SLAB PILE FOR HOT ROLLING SCHEDULE

TANG Lixin YANG Zihou

(Institute of Systems Engineering, Northeastern University, Shenyang 110006)

Abstract The optimal turned-out slab pile (TOSP) problem in the slab yard of iron & steel industry is formulated as a binary integer programming model in this paper. This is a quadratic programming model and the coefficients of the objective function are related to the values of variables. Because of NP-hardness of TOSP problem, it is difficult, or even impossible, to find the optimal solution to the large-scale actual problem. In order to solve TOSP, this paper develops the modified genetic algorithm (MGA) for this problem: 1) to construct the genetic coding suitable for the optimal TOSP problem. It can not only generate feasible initial chromosomes, but also ensure chromosome feasibility after crossover and mutation. 2) to form a MGA framework; a new cultivating operation is introduced; and crossover operation

1) 国家 863/CIMS 计划支持项目(863-511-708-009)和国家自然科学基金资助项目(79700006), 国家教育部优秀年轻教师研究基金和中国科学院机器人学开放实验室课题.

收稿日期 1999-01-13 收修改稿日期 1999-09-13

is improved. The computational experiments with the selected cases of the randomly produced problems show that the proposed new MGA is remarkably better than the original heuristics for Tosp problem, with the best improvement of 7.04%.

Key words Iron & steel industry, production scheduling, turned-out slab pile, integer programming, modified genetic algorithm.

1 引言

板坯库是连接炼钢和热轧生产工艺的中间缓冲环节. 它既是其紧前上游工序——炼钢工序生产出来的板坯的存放地, 也是其紧后下游工序——热轧工序需要进行加工的原料(板坯)的直接供给地. 热轧工艺是按照轧制单元计划进行组织生产的. 每一个轧制单元计划实质就是一个板坯的加工序列. 在制定轧制单元计划时, 主要是根据合同产品涉及到的板坯号、板坯钢级、板坯宽度、重量、厚度和交货期等信息, 不知道板坯库中的板坯的实际垛位位置情况. 因而在轧制单元计划实施前, 需要根据板坯的实际垛位情况, 对轧制单元计划中的每一个轧制位置的板坯从其对应的可交换板坯组中进行重新选择, 保证总的倒垛次数最少, 这个问题简称最优倒垛问题. 倒垛次数高, 吊机的负荷就大, 生产效率低. 因此, 最优倒垛问题直接关系到车间的生产效率和企业的生产效益, 是生产上迫切需要解决的问题.

对于板坯倒垛优化问题, 80 年代末, 德国的梯森在我国宝钢投产时开发了一个启发式算法, 这一程序一直运行到现在. 但是随着长时间的大量使用发现, 原有的算法倒垛效率低. 从国内外公开报道的文献来看, 已有的研究是围绕炼钢-连铸^[1~3]和热轧^[4~6]阶段的生产管理问题进行的, 对于倒垛问题还没有发现公开的报道. 本文的主要工作是对最优倒垛问题建立了数学规划模型, 用提出的改进遗传算法求解此问题.

2 问题的描述及数学模型

2.1 垛位的结构

板坯库中每一个垛位的结构如图 1 所示. 图 1 中, 需要的板坯是指轧制单元计划所建议的板坯, 在图中用阴影表示. 在需要板坯的上面的板坯为倒垛板坯, 只有这些倒垛板坯被移到其它垛位上, 才能露出所需要的板坯. 需要板坯露出后, 通过吊车运到相应的加热炉中进行加热, 然后才能进行轧制. 垛高是指该垛位叠放的板坯数. 板坯的记数是从垛底开始的, 这样,

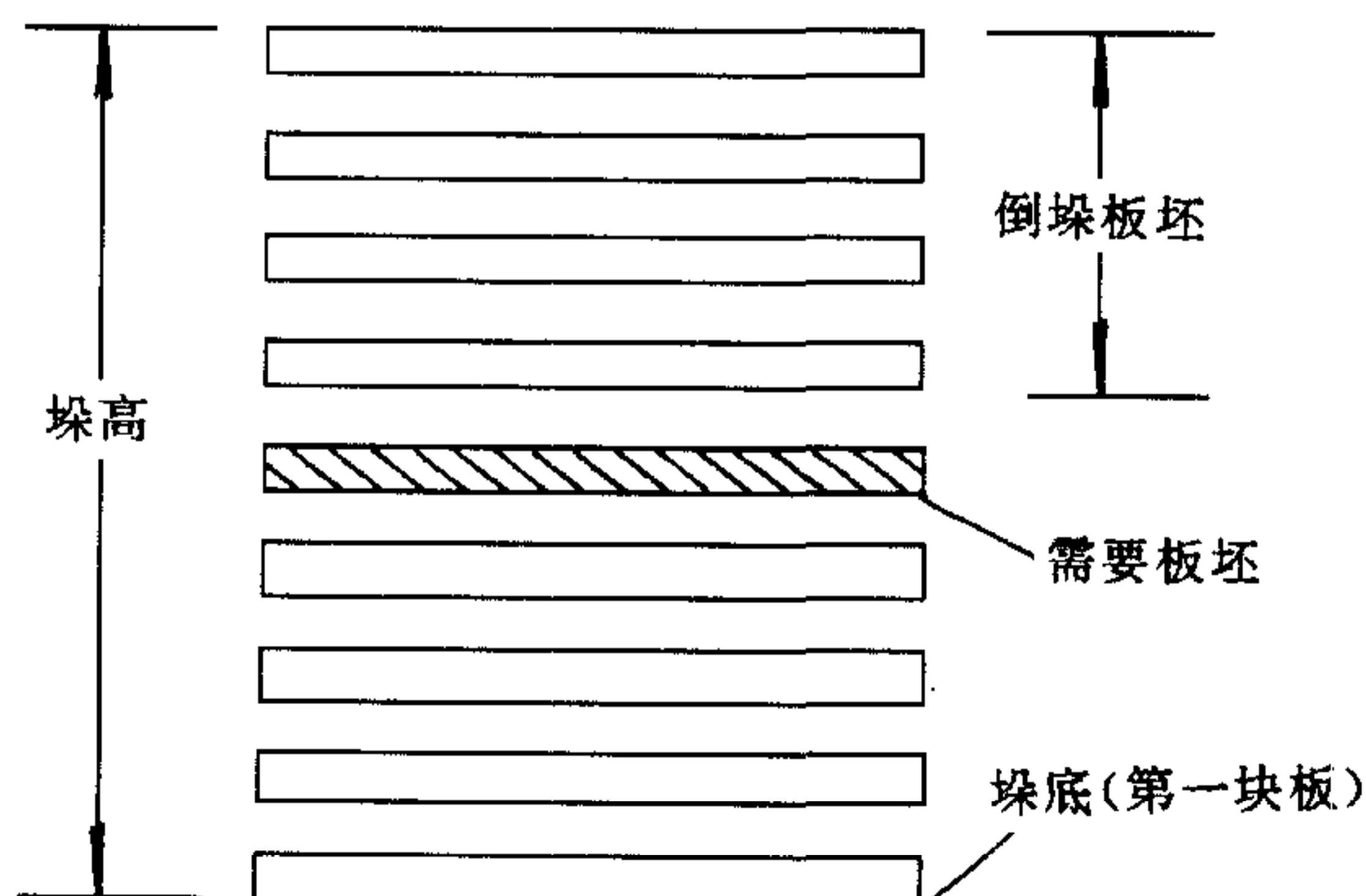


图 1 板坯库一个垛位的结构

垛底的板坯为垛位中的第一块板坯。

2.2 最优倒垛问题的定义及数学模型

一个轧制单元计划是由 M 个轧制项目(板坯)组成, 这些轧制项目的顺序和位置在热轧计划中被确定, 根据生产工艺的要求是固定不变的. 一般在一个轧制单元计划实施前, 需要释放 3~5 个轧制单元计划, 以便从其对应的板坯构成计划预选池, 轧制单元计划中的每个项目位置在板坯库的计划预选池中都有若干块适合条件的板坯可以作为可交换板坯, 从而构成对应这一位置的可交换板坯组. 在轧制单元计划中引入下面一些符号, 以便定义模型.

已知参数如下:

$\Omega = \{1, \dots, M\}$ 为整个轧制单元计划的组成项目的轧制位置集合, M 为一个轧制单元计划包括的项目数.

$S = \{1, \dots, N\}$ 为整个计划预选池中板坯集合, N 为一个板坯库计划预选池中板坯总数.

$S_i =$ 为第 i 个轧制位置对应的可交换板坯集合, $S_i \cap S_k = \emptyset$, \emptyset 为空集, 其中 $i, k \in \Omega$, 但 $i \neq k$. 在这里应满足 $S = \bigcup_{i \in \Omega} S_i$.

决策变量为:

$$X_{ij} = \begin{cases} 1, & \text{如果轧制单元计划中第 } i \text{ 个位置选择第 } j \text{ 块板坯, 其中 } i \in \Omega, j \in S_i. \\ 0, & \text{否则.} \end{cases}$$

数学规划模型:

(P)

$$\min. \sum_{i=1}^M \sum_{j=1}^N C_{i,j} * X_{i,j}; \quad (1)$$

$$\sum_{j=1}^N x_{i,j} = 1, \quad i \in \Omega; \quad (2)$$

$$\sum_{i=1}^M x_{i,j} \leq 1, \quad j \in S; \quad (3)$$

$$\sum_{j \in S_i} x_{i,j} = 1, \quad i \in \Omega; \quad (4)$$

$$x_{i,j} = 0, \quad i \in \Omega, j \in S \text{ 但 } j \notin S_i. \quad (5)$$

在这里系数 $C_{i,j}$ 表示当位置 i 需要板坯 j 时, 需要的净倒垛次数. 它是动态变化的, 不是一个常数, 它的取值依赖于前 $i-1$ 个项目选择的板坯垛位情况. $C_{i,j}$ 应是一个净数量, 设置 D_j 在计划执行前第 j 块板坯上面的原始板坯数量, 则 $C_{i,j}$ 具体表达如下:

$$C_{i,j} = D_j - \sum_{n=1}^{i-1} \sum_{k \in \{m | \varphi(m) = \varphi(j)\}} \min\{\max\{D_j - D_k, 0\}1\} \times X_{n,k}.$$

目标函数 (1) 是使得总的导垛次数最小. 约束 (2) 使得每一个轧制位置必须有一块且只能有一块板坯. 约束 (3) 是使得任一板坯最多只能分配到一个轧制单元计划中的相应位置当中. 约束 (4) 是使得每一个轧制位置一定是由其对应的可交换板坯集合的任一板坯来填充. 约束 (5) 是使得每一个轧制位置不能由其对应的可交换板坯集合以外的板坯来填充. 这实质是一个二次规划模型, 求解其最优解是相当困难的.

3 最优倒垛问题(P)的改进遗传算法

遗传算法(GA)是一类基于自然进化和选择机制自适应的搜索算法,它成功地应用到多种优化问题的求解^[7].一般遗传算法求解最优倒垛问题有两个困难:1)可行化的遗传编码构造;2)最优性能难以保证.针对这种情况,本文构造了下面的改进遗传算法.

3.1 改进遗传算法(MGA)结构

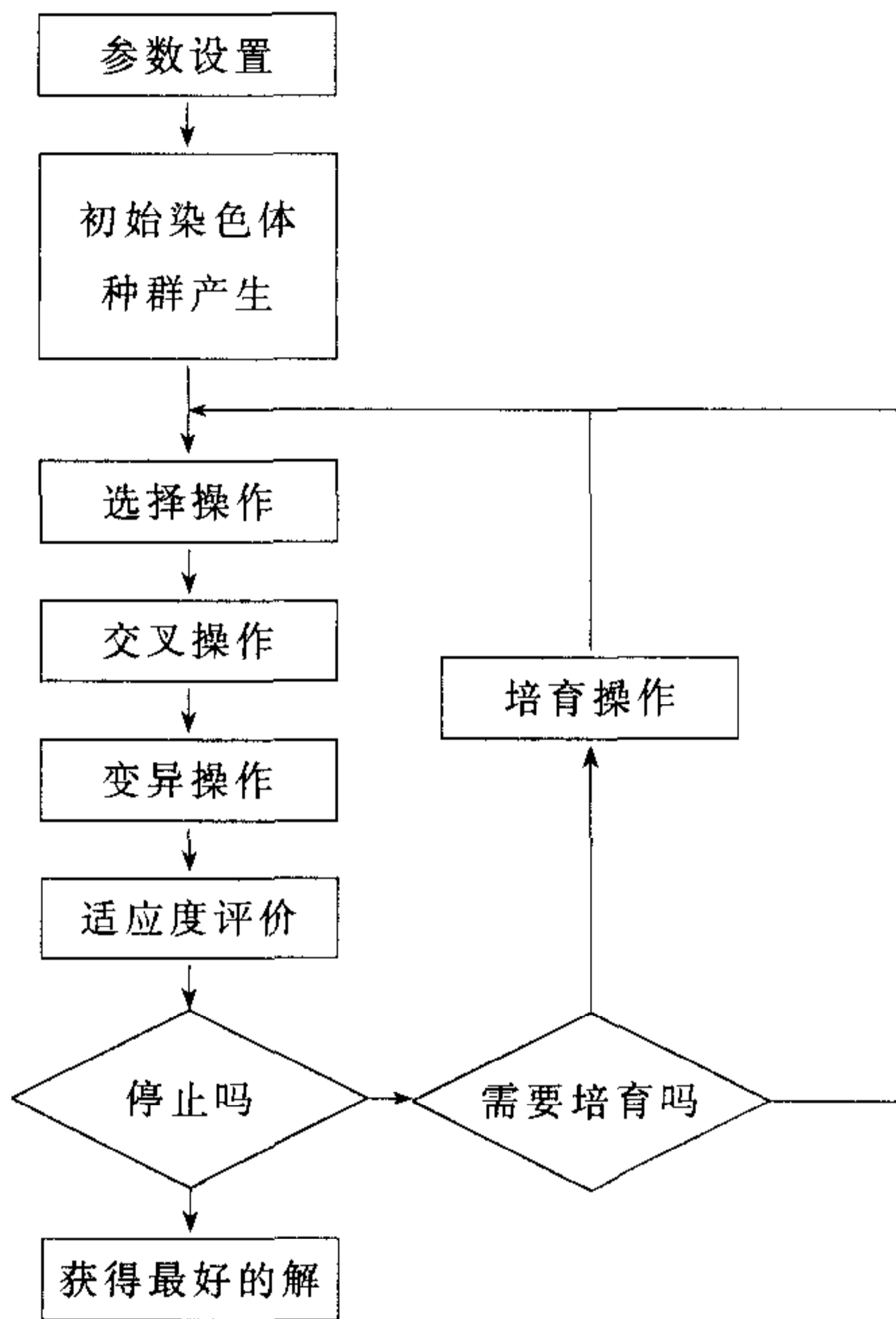


图 2 改进遗传算法的总体结构

提出的改进遗传算法结构如图 2 所示,与一般遗传算法相比有两点不同:增加了一个培育操作,改进了交叉操作.

培育操作:通过实验显示出,当遗传算法在一定代数不能改进时,大约为 30 代,如果继续迭代下去,则解的质量很难改变或改变较少.在这里修改了遗传算法操作,如果当连续迭代的代数超过一个给定数量时,解的质量仍然没有改进,则调用培育操作,用于培育最有希望的个体,既把最好的染色体通过邻域搜索算法进行改进,然后返回选择操作当中,继续进行下一代的迭代.

种子模式交叉操作:传统的交叉模式是随机产生两个染色体配对进行交叉,产生两个新的子代,这种交叉效果不是很好.而种子模式交叉操作是选择直到当前代最好的染色体和随机产生的另一个染色体相配对,配

对的两个染色体采用位交叉方法进行交叉.

3.2 对于最优倒垛问题的遗传编码构造

对于最优倒垛问题,用传统的二进制编码和无重叠的自然数编码^[7],初始染色体无法保证可行,更谈不上交叉和变异后保持可行.在下面构造的一种新的遗传编码方法,不但能够产生初始可行的染色体,而且交叉和变异后的染色体仍然可行.

编码的基本思想和方法的是,用轧制单元计划对应的板坯序列号构造染色体.一个染色体对应一个轧制单元计划可行的板坯序列.如果一个轧制单元计划是一个由 M 块板坯组成的板坯序列,每一个轧制位置对应的可交换板坯集合中的元素用自然数从小到大进行单独编号,则染色体的长度为 M ,基因位置与轧制单元计划的轧制位置相一致,并一一对应,染色体的基因值是由对应轧制位置的可交换板坯集合中的板坯序号构成的. $\forall i$ 每一个轧制位置的可交换板坯集合中的板坯序号的不同取值,构成了遗传算法的搜索空间.设染色体的组成为序列 $P_1 P_2 \cdots P_L$,则其数学描述及具体规定如下:

$$1) L = M; \quad (6)$$

2) $P_i \in \{1, \dots, |S_i|\}, i \in \{1, \dots, M\}$, $|S_i|$ 为第 i 个可交换板坯集合的元素数量,范围为

$$1 \leq |S_i| \leq 9. \quad (7)$$

例 1. 已知 $O_1 = \{1, 2, 3\}$, $O_2 = \{1, 2\}$, $O_3 = \{1, 2, 3, 4, 5\}$, $O_4 = \{1, 2, 3, 4, 5, 6\}$ 分别为一个轧制单元计划对应的 4 个轧制位置的可交换板坯集合. 一个为 $A = 2\ 1\ 4\ 5$ 的染色体的含义是轧制单元计划有 4 块板坯组成, 第一个基因值表示第一个轧制位置是由 O_1 中第 2 块板坯组成, 第二个基因值表示第二个轧制位置是由 O_2 中第 1 块板坯组成, 第三个基因值表示第三个轧制位置是由 O_3 中第 4 块板坯组成, 第四个基因值表示第四个轧制位置是由 O_4 中第 5 块板坯组成.

性质 1. 满足条件(6)和(7)形式的染色体为原问题的可行解.

证明. 满足原问题可行解的充分条件是要满足模型(P)的约束条件(2), (3), (4), (5). 设一染色体 $P_1\ P_2 \cdots P_L$ 满足上面的条件(6)和(7), 则 $P_i (i \in \Omega)$ 对应可交换板坯集合 S_i 中的第 P_i 个元素, 因为在模型(P)中整个板坯序号属于大排序的, S_i 中的第 P_i 个元素对应一个 $J(i)$ 值 (也就是对应一个确定性的板坯序号), 其中 $J(i) = \Psi(P_i)$, 且 $J(i) \in S_i$. 这样对于 $X_{i, J(i)} = 1$; 而对于 $j \in S \setminus \{J(i)\}$, 有 $X_{i, j} = 0$; 对于 $k \in \Omega \setminus \{i\}$, 有 $X_{k, J(i)} = 0$, 所以存在如下关系:

1) 对于 $i \in \Omega$, $\sum_{j \in S} X_{i, j} = X_{i, J(i)} + \sum_{j \in S \setminus \{J(i)\}} X_{i, j} = 1$, 所以模型(P)的约束(2)满足.

2) 对于 $j \in S$, 分为 $j \in R = \{J(1), J(2), \dots, J(M)\}$ 和 $j \in S \setminus R$ 两种情况.

对于 $j \in R$, 设任一 $j = J(m)$, 则 $X_{m, J(m)} = 1$, 其他 $k \in \Omega \setminus \{m\}$, 有 $X_{k, J(i)} = 0$, 所以 $\sum_{i \in \Omega} X_{i, j} = 1$, 对于 $j \in S \setminus R$, 所有 $i \in \Omega$, 存在 $X_{i, j} = 0$, 所以 $\sum_{i \in \Omega} X_{i, j} = 0$.

综合以上两种情况, 对于 $j \in S$, 存在 $\sum_{i \in \Omega} X_{i, j} \leq 1$. 这样, 模型(P)的约束(3)满足.

3) 对于 $i \in \Omega$, $\sum_{j \in S_i} X_{i, j} = X_{i, J(i)} + \sum_{j \in S_i \setminus \{J(i)\}} X_{i, j} = 1$, 这样, 模型(P)的约束(4)满足.

4) 因为对于 $i \in \Omega$, $j \in S \setminus J(i)$ 中包括 $j \in S$ 但 $j \notin S_i$ 的情况, 所以对于 $i \in \Omega$, $j \in S$ 但 $j \notin S_i$, $X_{i, j} = 0$. 模型(P)的约束(5)满足. 证毕.

3.3 适应度函数的译码计算和选择策略

要计算适应度函数, 首先要计算目标函数值. 根据一个染色体基因值确定目标函数的过程称为译码计算. 从性质 1 的证明过程可看出, 对于任一染色体 $P_1\ P_2 \cdots P_M$ 对应问题(P)一个可行解, 所以对于 $X_{i, j}$, $i \in \Omega$, $j \in S$ 的值全部都能确定. 根据(P)中(1)式能够计算目标函数. 这样就把染色体的值转换为目标函数的值. 选择策略采用传统的滚动轮策略^[7].

3.4 交叉方法

采用位交叉方法进行交叉. 其过程为: 根据随机产生的交叉点, 两个随机配对的染色体从交叉点后进行互换, 形成两个新的染色体.

性质 2. 两个可行染色体通过位交叉操作后产生的两个新的染色体仍为原问题的可行解.

证明. 假设有两个配对的染色体

$$O_1 = A_1\ A_2\ A_3 \cdots A_M,$$

$$O_2 = B_1\ B_2\ B_3 \cdots B_M;$$

假设交叉位置在 k , 交叉以后的染色体为

$$D_1 = A_1 A_2 A_3 \dots B_k B_{k+1} \dots B_M,$$

$$D_2 = B_1 B_2 B_3 \dots A_k A_{k+1} \dots A_M.$$

因为 O_1 和 O_2 为两个可行染色体, 有 $A_i \in \{1, \dots, |S_i|\}, B_i \in \{1, \dots, |S_i|\}, i \in \Omega$. 这就说明, D_1 和 D_2 满足约束(6)和(7)式的要求, 根据性质 1, 交叉后的染色体 D_1 和 D_2 仍然可行. 证毕.

3.5 变异方法

对于变异, 经常使用的两种方法是: 1) 变值变异方法, 是将需要变异的基因由 0(或 1) 到 1(或 0) 进行变值. 这种方法只适用于二进制编码的遗传算法当中. (2) 交换变异方法, 是将两个基因位置进行交换. 这种交换策略适合于用无重叠的自然数编码的遗传算法变异当中. 上面两种变异策略对于倒垛问题(P)都不能保证变异后可行.

例 2. 对于(P)采用前面 3.2 节的遗传编码方法产生的染色体为 $Q=52461$, 假设 $|S_1|=5, |S_2|=2$, 采用位交换变异方法如下:

$$Q=52461 \xrightarrow[\text{位交换变异}]{\text{第 1 位和第 2 位进行交换}} Q_1=25461.$$

因为 $|S_2|=2$, 而 Q_1 中第二位是 5, 大于 $|S_2|$, 根据性质 1, Q_1 是不可行染色体.

本文提出了适合倒垛问题的变异方法, 对于任一基因 P_i , 当变异时候, 其值变为 $H_i = |S_i| + 1 - P_i$, 采用这种值变异方法, 能够保证变异后的染色体仍然可行.

性质 3. 一个可行的染色体通过上面的值变异方法变异后仍为原问题的可行解.

证明. 设任一可行染色体为 $P_1 P_2 \dots P_M$, 假设其中的 P_g 经过变异后变为 H_g , 染色体其它基因不变, 因为 $1 \leq P_i \leq |S_i|, i \in \Omega$, 所以存在 $1 \leq P_g \leq |S_g|$, 两边乘 -1 后得到 $-|S_g| \leq -P_g \leq -1$, 两边加上 $|S_g| + 1$ 得, $1 \leq |S_g| + 1 - P_g \leq |S_g|$, 所以有 $1 \leq H_g \leq |S_g|$, 这样对于变异后染色体的新基因仍然满足性质 1 的条件. 则变异后的染色体仍然可行. 证毕.

3.6 邻域搜索算法

本节叙述 3.1 节提到的培育操作中使用的邻域搜索算法的邻域结构.

定义 1. 替换邻域结构: 对于整个轧制单元计划的每一个位置的轧制项目, 都可以用其对应的组选板坯集合中的板坯来替换, 构成一个邻域结构. 设这个邻域结构为 δ .

性质 4. 对于一个包括 M 个轧制项目的轧制项目, 采用上面的 δ 作为邻域结构, 则邻域的计算量为 $\sum_{i=1}^M (|S_i| - 1)$.

证明. 因为对于一个轧制单元计划中的板坯 P_i , 其对应的可交换板坯集合的元素个数为 $|S_i|$, 但是因为可交换板坯集合中包括现有板坯 P_i , 这样 P_i 可替换的板坯数为 $|S_i| - 1$, 因为轧制项目的邻域的计算量为这种替换的全部单回路(单程)过程枚举之和, 该轧制单元计划包括 M 个轧制项目, 所以其邻域计算量为 $\sum_{i=1}^M (|S_i| - 1)$. 证毕.

4 仿真实验和分析

4.1 仿真实验设计

仿真数据的产生涉及到以下三种因素:

1)问题的结构.问题的结构涉及到两个因素:一个是轧制单元计划包括的项目数,另一个是垛高分布情况.在此实验中,轧制项目数分为5种情况;而垛高分布分为4种情况,共有20种不同情况.每一种情况,产生10次不同的随机数,板坯虚拟编码的变化范围是不变的,这样总共产生200种情况的例子.

2)问题的规模.项目数意指一个轧制单元计划中包括的板坯数.为了测试不同项目数情况下的算法性能,选择了5种不同的项目数的情况,如表1所示.因为实际一个轧制单元计划最多包括180个板坯,所以最大的项目数设置为180.

垛高是指一个垛位中的板坯数量.为了测试不同垛高变化情况下的算法性能,选择了4种不同的垛高数的变化情况.每一种垛高的变化范围如表2所示,数字的选择覆盖了实际的垛高范围.每一种情况的垛高是在其规定范围内按照均匀分布产生的整数.

表1 项目数的变化情况

项目数	20	60	100	140	180
-----	----	----	-----	-----	-----

表2 垛高的变化范围

垛高范围: $H(XY)$ (整数)	2-10	4-12	4-16	4-20
垛高分布: $H(XY)$ 平均值	6	8	10	12

4.2 算法的实现及计算结果

用MS.C.6.0在Pentium主频为133的计算机上实现了最优倒垛问题的改进遗传算法,通过大量的仿真,改进遗传算法的最优参数配置如表3所示.

表3 改进遗传算法的最优参数配置

最大迭代代数 MAX_GENE	种群大小 POP_SIZE	交叉概率 P_c	变异概率 P_m
200	100	0.99	0.01

为了比较提出的算法和原系统的算法性能,本文也实现了老系统的启发式算法,并且通过了实际数据的案例,检验了算法实现的正确性.对于20种不同的情况,算法的计算结果如表4所示.在表4中,第2和3列表示算法的相对最优性能,用下面的相对测量方法来表示相对最优性.对于每一种情况的结果,分别表示为 $(\sum_{i=1}^{10} (C1_i/C_i^*)) / 10$ 和 $(\sum_{i=1}^{10} (C2_i/C_i^*)) / 10$ 的计算结果,其中 $C_i^* = \min\{C1_i, C2_i\}$, $C1_i$ 和 $C2_i$ 分别为同一种情况,算法1和算法2对应第*i*次随机数的目标函数值.在表5和表6中是以表4为基础进行计算的,分别表示倒垛问题随项目数变化和垛位分布的新老系统性能比较结果.

从表4~表6的计算结果得出如下结论:

1)在最优性方面,对于二十种情况,从表4的第二列和第三列可以看出,提出的改进遗传算法全部都优于原启发式算法.最好的情况可以达到改进率7.04%.

2)从表5的第二列和第三列可以看出,随着项目数的增加,改进遗传算法与原系统的启发式算法比较,相对性能明显提高.

3)从表6的第二列和第三列可以看出,随着垛位分布数的下降,改进遗传算法与原系

统的启发式算法比较,相对性能明显提高.

4)计算时间方面,遗传算法比启发式算法消耗的时间多.从实际运行来看,由遗传算法产生的方案比原系统的启发式算法虽然时间消耗长一点,但是它换来了产生最终方案人工调整时间的缩短.从而产生最终方案的时间比原系统还是缩短了.

表 4 倒垛问题随问题结构的各种情况变化的新老系统性能比较

项目数×垛高分布	最优性能比较		计算时间比较(s)	
	原系统启发式算法	改进遗传算法	原系统启发式算法	改进遗传算法
20×6	1.0000	1.0000	0.2000	3.1000
20×8	1.0405	1.0083	0.1000	3.2000
20×10	1.0000	1.0029	0.1000	3.1000
20×12	1.0000	1.0000	0.2000	3.1000
60×6	1.0244	1.0056	0.2000	19.0000
60×8	1.0045	1.0035	0.2000	19.2000
60×10	1.0046	1.0011	0.3000	19.0000
60×12	1.0073	1.0008	0.2000	19.2000
100×6	1.0291	1.0034	0.5000	52.8000
100×8	1.0347	1.0000	0.6000	52.4000
100×10	1.0066	1.0009	0.6000	52.5000
100×12	1.0128	1.0000	0.5000	52.3000
140×6	1.0510	1.0000	1.1000	105.4000
140×8	1.0689	1.0000	1.2000	105.7000
140×10	1.0246	1.0000	1.2000	106.0000
140×12	1.0135	1.0004	1.0000	105.5000
180×6	1.0704	1.0008	1.9000	181.9000
180×8	1.0414	1.0000	1.9000	180.4000
180×10	1.0269	1.0004	1.8000	181.2000
180×12	1.0155	1.0000	2.0000	182.2000

表 5 倒垛问题随项目数变化的新老系统性能比较

项目数	最优性能比较		计算时间比较(S)	
	原系统启发式算法	改进遗传算法	原系统启发式算法	改进遗传算法
20	1.0101	1.0028	0.1500	3.1250
60	1.0102	1.0028	0.2250	19.1000
100	1.0208	1.0011	0.5500	52.5000
140	1.0395	1.0001	1.1250	105.6500
180	1.0386	1.0003	1.9000	181.4250

表 6 倒垛问题随垛位分布变化的新老系统性能比较

垛位分布(平均垛高)	最优性能比较		计算时间比较(S)	
	原系统启发式算法	改进遗传算法	原系统启发式算法	改进遗传算法
6	1.0431	1.0036	0.7600	72.4600
8	1.0299	1.0007	0.8200	72.1600
10	1.0125	1.0011	0.8000	72.3600
12	1.0098	1.0002	0.7800	72.4600

5 结论

本文的主要贡献在于对钢铁企业生产管理每天都遇到的热轧计划实施中的板坯最优倒垛问题给出了一种新的定量方法,在建立了数学模型基础上,提出了改进的遗传算法.并以此模型和算法为核心,开发了整个最优倒垛调度仿真系统.该系统于1997年12月在微机环境下开发完毕,在宝钢热轧厂仿真系统进行了运行.根据实际数据的仿真计算,证明了提出的算法的有效性.从国内外公开报道的文献来看,还没有发现此类模型和及相应的算法,本文对这一问题的定量方法,除了具体参数不同外,对其他钢铁企业的板坯库的倒垛问题具有较好的通用性.

参 考 文 献

- 1 Lee H S, Murthy S S, Haider S W *et al.* Primary production scheduling at steelmaking industries. *IBM Journal of Research & Development*, 1996, **40**(2): 231~252
- 2 Box R E, and Herbe D G. A scheduling model for LTV steel's Cleveland works twinstrand continuous slab caster. *Interface*, 1988, **18**(1): 42~56
- 3 唐立新,杨自厚,王梦光. 基于准时制的炼钢-连铸计划调度问题的研究. *自动化学报*, 1998, **24**(1): 9~14
- 4 Jacobs T L, Wright J R. Optimal inter-process steel production scheduling. *Computers and Operational Research*, 1988, **15**: 497~507
- 5 Kosiba E D, Wright J R. Discrete event sequencing as a traveling salesman problem. *Computers in Industry*, 1992, **19**: 317~327
- 6 Lopez L, Carter M W, Gendreau M. The hot strip mill production scheduling problem: A tabu search approach. *European Journal of Operational Research*, 1998, **106**: 317~335
- 7 Goldberg D E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. MA: Addison-Wesley Reading, 1989, 5~10

唐立新 1966年生,1995年东北大学自动控制系博士毕业.1996年在香港科技大学工业工程系从事博士后研究.现为东北大学系统工程研究所副所长,教授,研究兴趣为:生产与运输中的调度理论及其应用,智能优化方法和运筹学理论与应用研究.

杨自厚 1926年生.1950年毕业于武汉大学电机系,现为东北大学系统工程研究所教授.研究领域为生产排序问题、组合最优化智能算法、决策支持系统构造及其应用.