

基于自适应退火遗传算法的车间日作业计划调度方法

刘 敏 严隽薇

(同济大学电子与信息工程学院 CIMS 研究中心 上海 200092)

摘 要 遗传算法、模拟退火算法、最优个体保护法在全局收敛性、种群早熟化、收敛速度慢等方面存在应用缺陷。文中提出了自适应退火遗传算法解决车间日作业计划的调度问题。该算法针对遗传算法中组成编码串的变异概率在整个搜索过程中是固定不变的,而且取值较小,促使算法的求解过程很长,且易走向局部最小值,提出自适应变异概率的概念与理论改善遗传算法的收敛速度;针对选择算子对种群多样性的影响,提出整体退火选择的方式(Boltzmann 概率选择机制)选择杂交母体,以克服种群早熟化,避免过早收敛。最后结合车间日作业计划静态调度模型给出求解算法和求解实例。

关键词 自适应退火遗传算法;遗传算法;车间日作业计划;调度;生产计划

中图法分类号 TP391

An Adaptively Annealing Genetic Algorithm based Scheduling Method of Workshop Daily Operating Planning

LIU Min YAN Jun-Wei

(CIMS Research Center, School of Electronic and Information Engineering, Tongji University, Shanghai 200092)

Abstract Genetic Algorithm, Simulated Annealing Algorithm and Optimum Individual Protecting Algorithm origin from the order of nature, they exist some application limitations in the global astringency, population precocity and convergence rapidity. The Adaptively Annealing Genetic Algorithm (AAGA) is provided to deal with the scheduling question of workshop daily operating planning based on the above algorithms. In AAGA, the adaptive mutation probability is built to improve the convergence rapidity of genetic algorithm through adaptively changing mutation probability to shorten the entire optimizing process and to avoid the local optimization, the Boltzmann probability selection mechanism from simulated annealing algorithm is applied to select the crossover parents, which can solve the population precocity and the local convergence. At last, the AAGA based scheduling algorithm and domain model of workshop daily operating planning are discussed, the computing results are depicted and compared between AAGA and GA.

Keywords adaptively annealing genetic algorithm; genetic algorithm; workshop daily operating planning; scheduling; production planning

1 引 言

随着市场的日益国际化和用户需求的不断增

长,越来越多的企业以单件小批订货生产方式组织生产^[1-2]。然而客户的需求千差万别,对于每一个订单的投产计划(或出产计划)而言,都需要重新进行产品设计(全新设计、适应性设计或变型设计)、工艺

设计、生产作业计划重排等开发过程。

针对单件小批订货生产企业在技术准备、计划管理方面的约束,基于分布式设计与资源动态反馈的思想,基于分布式动态工艺设计的 CAPP,PPS 集成化系统模型(Integrated System Model of Computer Aided Process Planning and Production Planning & Scheduling Based on Distributed & Dynamic Process Planning,简称 ICAPPS)实现了 CAPP,PPS 功能与信息在时间轴上的集成与并行化^[3]。在 ICAPPS 车间级计划层的车间日作业计划生成过程中,需要结合零件工序的变动方案(更换工序、调整设备等),将零件工序的加工任务调度到适当的机床上,以制订有效的车间日作业计划,从而确保当日任务的顺利完成及关键资源不出现等待时间等问题。

遗传算法、模拟退火算法、最优个体保护法都是起因于自然界某些规律的算法,由于具有极强的稳健性和整体优化性^[4-5],已被应用于复杂组合优化问题求解^[6]、智能控制^[7]。然而,随着研究的深入与推广,这些算法在全局收敛性、种群早熟化、收敛速度等方面的缺陷也日益显现^[8-9]。因此,Sirag 和 Weisser 提出了面向一致化热动态算子(Towards unified thermodynamic operator)^[10],Bosesniuk 和 Ebeling 将玻尔兹曼、达尔文和黑格尔策略用于优化问题(Boltzmann-Darwin-and -Heackel-strategies in optimization problems)^[11],Golberg 和 Mahfoud 讨论了并行再生模拟退火算法(Parallel Simulated Annealing,PRSA)^[12],来改进这些算法的缺陷。

本文基于 GA 算法,结合最优个体保护法和 SA 算法的特点,提出了自适应退火遗传算法(Adaptively Annealing Genetic Algorithm,AAGA),通过在搜索过程中保持种群多样性,采用自适应变异概率、SA 算法中的 Boltzmann 概率选择机制等策略来克服遗传算法的过早收敛现象,以解决车间日作业计划的调度问题。第一部分针对标准遗传算法、最优个体保护法、模拟退火算法等进化算法存在种群早熟化、收敛速度慢的应用缺陷,提出自适应退火遗传算法的概念与求解理论;第二部分着重研究了自适应退火遗传算法,针对遗传算法中组成编码串的变异概率在整个搜索过程中是固定不变的,而且取值较小,促使算法的求解过程很长,且易走向局部最小值,提出自适应变异概率的概念与理论改善遗传算法的收敛速度;针对选择算子对种群多样性的影响,提出整体退火选择的方式选择杂交母体,以克服种群早熟化,避免过早收敛;第三部分建立了车间静

态调度问题的模型及基于自适应退火遗传算法的求解算法;最后结合实例进行讨论。

2 自适应退火遗传算法(AAGA)

自适应退火遗传算法针对遗传算法中组成编码串的变异概率在整个搜索过程中是固定不变的,而且取值较小,促使算法的求解过程很长,且易走向局部最小值,提出自适应变异概率的概念与理论改善遗传算法的收敛速度。算法针对选择算子对种群多样性的影响,提出整体退火选择的方式选择杂交母体,以克服种群早熟化,避免过早收敛。

2.1 自适应退火遗传算法中的几个概念

(1) 杂交、变异和选择^[13]

$|X^{n0}|, |X^{n1}|$ 分别表示空间 X^{n0} 和 X^{n1} 中的元素个数,令:

$$X^{n0} = \{S^{(1)}, \dots, S^{(i)}, S^{(i+1)}, \dots, S^{(X^{n0})}\},$$

$$X^{n1} = \{P^{(1)}, \dots, P^{(i)}, P^{(i+1)}, \dots, P^{(X^{n1})}\}.$$

记 $S^{(i)} (i=1, 2, \dots, |X^{n0}|)$ 为父代种群空间 X^{n0} 中的元素, $P^{(j)} (j=1, 2, \dots, |X^{n1}|)$ 为杂交种群空间 X^{n1} 中的元素, $P^{(m)} (m=1, 2, \dots, |X^{n1}|)$ 为中间种群空间 X^{n1} 中的元素, $C_{ij}(k)$ 表示算法在第 k 步由 $S^{(i)}$ 经杂交变为 $P^{(j)}$ 的概率, $M_{jm}(k)$ 表示算法在第 k 步由 $P^{(j)}$ 经变异变为 $P^{(m)}$ 的概率,而 $S_{mq}^{(i)}(k)$ 表示算法在第 k 步,当父代为 $S^{(i)}$ 时,由中间种群 $P^{(m)}$ 变为新的父代种群 $S^{(q)}$ 的转移概率。

根据 GA 算法中杂交、变异和选择的定义, $C_{ij}(k), M_{jm}(k)$ 和 $S_{mq}^{(i)}(k)$ 具有以下性质:

$$\sum_{j=1}^{|X^{n1}|} C_{ij}(k) = 1, i=1, 2, \dots, |X^{n0}|, k=0, 1, 2, \dots \quad (1)$$

$$M_{jm}(k) = \prod_{i=1}^{n1} (P_M(k))^{h_i} (1 - P_M(k))^{1-h_i},$$

$$m, j=1, 2, \dots, |X^{n1}|, k=0, 1, 2, \dots \quad (2)$$

$$S_{mq}^{(i)}(k) = \begin{cases} \prod_{x_i \in S^{(q)}} J_k(f_i) / \left(\sum_{x_j \in P^m} J_k(f_j) \right)^{n0}, & S^{(q)} \subset P^m \\ 0, & \text{其它} \end{cases} \quad (3)$$

其中 h_1, h_2, \dots, h_{n1} 为 $P^{(j)}$ 和 $P^{(m)}$ 对应个体之间的 Hamming 距离, $P_m(k)$ 是第 k 步变异概率。

(2) 自适应变异概率

常用的遗传算法中,被讨论问题组成编码串的变异概率在整个搜索过程中是固定不变的,且取值较小,促使算法的求解过程很长,且易走向局部最小

值. 在马晋韬提出的变异概率方法的基础上^[14], 我们给出了自适应变异概率的概念:

在变异初始, 变异概率 $P_m(k)$ 大, 则码串变异大, 并对适应值小于平均适应值的个体给予较大的 $P_m(k)$, 以促使适应值小的码串自身变异; 随迭代次数的增加, 逐渐逼近最优解时, 减少 $P_m(k)$ 抑制适应值大的码串变异, 以提高算法计算速度, 扩大搜索范围.

自适应变异概率按下式求取:

$$P_m^i(k) = \begin{cases} P_M, & f_i \geq f_a \\ P_M \left(1 + \exp \left(\eta \frac{f_a - f_i}{f_a} \right) \exp(-k) \right), & \text{其它} \end{cases} \quad (4)$$

其中, $P_m^i(k)$ 为第 k 代迭代时第 i 个个体 x_i 的变异概率;

f_a 表示平均适应值;

f_i 为第 i 个个体 x_i 的适应值 $f(x_i)$;

k 为迭代代数;

P_M 为给定的变异概率;

η 为根据具体问题确定的常数.

(3) 整体退火选择

遗传算法通常采用按比例从种群 P_k 中随机独立地选取 N 个个体作为新的父代种群 F_k , 若允许重复选取, 则 $x_i \in P_k$ 被选中的概率为

$$P(x_i) = J_k(f_i) / \sum_{x_j \in P_k} J_k(f_j),$$

$J(f(x))$ 为问题的一个适应度函数, $J: R \rightarrow R$ 为一严格单增函数, 且使 $J(f(x)) > 0$.

引入模拟退火中的 Boltzmann 概率选择机制 (又称为整体退火选择机制), 得到整体退火选择下的子代产生方法: 以概率的方式从一个多点集 P_k 中选取另一多点集 F_k , $x_i \in P_k$ 被选入 F_k 的概率:

$$P(x_i) = \exp(f_i/T_k) / \sum_{x_j \in P_k} \exp(f_j/T_k) \quad (5)$$

式中, T_k 为渐趋于 0 的退火温度, 式(5)就是整体退火选择下的选择概率, 相应适应度函数为

$$J_k(f(x)) = \exp(f(x)/T_k) \quad (6)$$

整体退火选择下的算法所产生种群的每一个体都以概率 1 收敛到整体最优解, 不仅保证收敛, 而且比标准遗传算法有快得多的收敛速度、有更强地逃脱局部极值和避免过早收敛的全局优化能力.

2.2 AAGA 算法的自然收敛准则

AAGA 算法在 GA 算法的步骤中引入 Boltzmann 概率选择机制——整体退火选择, 退火选择与标准遗传算法中二者择一接受准则的不同之处在

于: 个体 x_i 按式(5)的概率从一个多点集 P_k 中选取一多点集 F_k , 从而在 AAGA 算法中适应度函数变为式(6). 此时, 式(3)变为

$$S_{mq}^{(i)}(k) = \begin{cases} \prod_{x_j \in S^{(q)}} \exp(f_j/T_k) / \left(\sum_{x_j \in P^m} \exp(f_j/T_k) \right)^{n_0}, & S^{(q)} \subset P^m \\ 0, & \text{其它} \end{cases} \quad (7)$$

为简化讨论, 假定变异概率 $P_M(k)$ 和杂交方式不随 k 变化, 并记 $G(T_k) = (G_{iq}(k))$, 其中,

$$G_{iq} = \sum_{m=1}^{|B^{n1}|} \left(\sum_{j=1}^{|B^{n1}|} C_{ij} M_{jm} \right) S_{mq}^{(i)}(k) \quad (8)$$

这样采用整体退火选择式(5)下的 AAGA 算法可表示为如下 Markov 链

$$V(k+1) = V(k)G(T_k) \quad (9)$$

为了分析 AAGA 算法的收敛性, 先引入一些记号和定义: 设 $S = (x_0, x_1, \dots, x_{n_0}) \in X^{n_0}$, 记

$$f(S) = (f(x_0), f(x_1), \dots, f(x_{n_0})) \quad (10)$$

定义 1. 设 n 维向量 f_1, f_2 满足或 $f_1 = f_2$, 或 $f_1 - f_2$ 的第一个非零分量为正, 则称 f_1 不劣于 f_2 , 并记 $f_1 \geq f_2$, 若 $S^{(1)}, S^{(2)} \in X^{n_0}$ 满足 $f(S^{(1)}) \geq f(S^{(2)})$, 则称 $S^{(1)}$ 不劣于 $S^{(2)}$ 并记 $S^{(1)} \geq S^{(2)}$.

记

$$\begin{cases} I_1 = \{i: S^{(i)} \geq S^{(j)}, \forall S^{(j)} \in X^{n_0}\} \\ I_2 = \{i: i \notin I_1 \text{ 且 } \exists P^m \in X^{n_1} \text{ 使 } S^{(i)} \geq S^{(j)}, \\ \quad \forall S^{(j)} \subset P^m\} \\ I_3 = \{1, 2, \dots, |X^{n_0}|\} \setminus (I_1 \cup I_2) \end{cases} \quad (11)$$

式中, 当允许父代参加竞争时, $n = n_0 + n_1$, 而不允许父代参加竞争时, $n = n_1$. 由“ \geq ”的定义知, 若 $i \in I_1$, 则 $S^{(i)} = (x_0, x_1, \dots, x_{n_0})$ 满足

$$f(x_1) = \dots = f(x_{n_0}) = \max_{x \in X} f(x) \quad (12)$$

因此有

$$S^{(i)} \cap S^* \neq \emptyset \quad (\text{事实上 } S^{(i)} \subset S^*) \quad (13)$$

同样, 若 $i \in I_2$, 则存在 $P^m \in X^n$ 使得 $S^{(i)} = (x_0, x_1, \dots, x_{n_0}) \subset P^m$ 满足

$$f(x_1) = \dots = f(x_{n_0}) = \max_{x \in P^m} f(x) < \max_{x \in X} f(x) \quad (14)$$

从而有

$$S^{(i)} \cap S^* \neq \emptyset, i \in I_2 \quad (15)$$

定理 1. 整体退火选择准则式(5)下 AAGA 算法收敛的充要条件是允许父代参加竞争^[3].

因此, 与比例选择下的遗传算法不同, 允许父代参加竞争的 AAGA 算法总是整体收敛的, 而且整体

退火下的 AAGA 算法能保证种群中的任一个体以概率 1 的方式收敛到全局最优解,即 AAGA 算法是整体收敛的。

2.3 AAGA 算法的流程

结合 GA、SA 算法的流程,给出 AAGA 算法的计算步骤:

- 1. 初始化 $T = T_0, i = 0$,用随机方法初始化群、计算群的适应度;
- 2. 若 $T > T_{\text{final}}$,做以下各步;否则结束,返回最优解;
- 3. 若 $i_{\text{loop}}(\text{内循环次数}) > \text{Constant}$,做以下各步;否则 $T_i = T_{i+1}, T_{i+1} < T_i, i = i + 1$,返步 2;
- 4. 从群中选择 n 对个体作为父本,做:
 - 4.1. 由父本 P_1, P_2 ,使用交叉、变异算子生成子代 C_1, C_2 ,计算 C_1, C_2 的适应性;
 - 4.2. 若 $f_{ci} > f_{pi}, i = 1, 2$,则 C_i 用代替 P_i ;否则,以概率 $\exp((f_{ci} - f_{pi})/T)$ 保持 P_i ;
- 5. 返回步 3.

其中,Constant 是内循环次数, T_0, T_{final} 分别是初始温度和终止温度;算法求 f 最大即 $(-f)$ 最小。

3 ICAPPS 中车间日作业计划的调度模型

车间静态调度的任务是根据车间前一天的计划完成情况、生产准备情况、设备利用状况安排本日及第二日的加工任务(车间日作业计划),给出各类机床上零件工序的加工顺序及其在不同机床间的流转,为了确保当日任务的顺利完成及关键资源不出现空闲等待,以加权平均流通时间最小(工序的完成时间与准备时间之差)为优化目标,结合零件的替代工序(或替代设备),将零件工序的加工任务调度到适当的机床上,求出机床最优的工序加工顺序。

3.1 车间日作业计划的调度问题描述

根据 ICAPPS 模型中调度层所担负的功能要求,将静态调度问题归结如下。

- 给定资源约束状况、车间作业计划,计划包括:
- 加工计划的零件集合,包括零件若干(允许零件重复);
 - 每种零件的工艺方案集合;
 - 每种零件各工序加工时间已知;
 - 工序开工、完工时间已知;
- 求详细的车间日作业计划(机床上的加工顺序)。

与上述模型相关的概念有:调度空间、各种零件的生产顺序,即多种可行的调度计划;工艺规划空间、加工每种工序可采用若干种不同替代工序;资源

需求空间,每种可加工工艺路线中的每个工序都可能在若干种不同的机床上实现。

由于每种替代工序和分配给它的资源紧密联系,所以往往将资源需求空间与工艺规划空间作为一个整体来考虑,而每种加工工序实现的可能方法数为每个工序可选择资源数的乘积,因此总的规划、资源空间大小可表示为

$$\prod_{i=1}^N \prod_{j=1}^n \prod_{k=1}^{n_{op}} n_{\text{resource}}[i, j, k],$$

其中, N 为一次调度中的零件种类数; n 表示该次调度中零件 i 的加工工序数; n_{op} 表示生产零件 i 的第 j 种加工工序的可能替代工序数; $n_{\text{resource}}[i, j, k]$ 为生产零件 i 的第 j 种加工工序中第 k 个替代工序的可用资源数。

3.2 若干相关假设

参考反映静态调度问题的一般特征模型^[15],给出求解 ICAPPS 模型中静态调度问题的若干相关假设:

- (1) 每件工作可生产多种零件,每种零件数量若干;
- (2) 一个加工车间可加工多种零件;
- (3) 在同一车间内,每种工序有若干种替代工序;
- (4) 每种规划由一系列串行加工工序构成(并行加工工序可通过制订多种工艺规划过程转变为串行加工工序);
- (5) 每种工序需要若干资源,如机床、刀具、夹具等(仅考虑机床、刀具);
- (6) 每台机床可用于加工不同类型的工序,每种工序也可选用不同机床;
- (7) 每种工序需占用某机床一段时间;
- (8) 每件工作由一个需求序列组成;
- (9) 每种需求序列对应一个零件,一件工作生产多个零件,工作中就有多少需求;
- (10) 一个工件不能同时在几台不同的机床上加工;
- (11) 工件在加工过程中采取平行移动方式,即当上一道工序完工后,送下一道工序加工;
- (12) 不允许中断. 当一个工件一旦开始加工,必须一直进行到完工,不得中途停止或插入其它工件;
- (13) 每道工序只在一台机床上完成;
- (14) 工件数、机床数和加工时间已知,加工时间与加工顺序无关;

(15) 每台机床同时只能加工一个工件.

求解这类静态调度问题就是设法将生产各种零件的工序有效地、合理地分配到可供使用的机床上,以求尽可能地保证顺利完成任务,避免拖期.

3.3 调度的表示方法

每台机床的调度用分配到该机床上各种零件工序的操作表示,操作按时间顺序排列,2 个 H_2 与 2 个 H_1 零件在三台机床上的加工可以表示如下:

$H_{2-1} \quad H_{1-1} \quad H_{2-2} \quad H_{1-2}$
 $o_1 o_2 o_7 \quad o_1 o_4 o_6 \quad o_1 o_5 o_6 \quad o_1 o_4 o_6$
 $M_1 M_1 M_2 \quad M_1 M_1 M_2 \quad M_1 M_3 M_3 \quad M_1 M_2 M_2$

其中, H_{i-j} 为第 i 种零件第 j 个; o_k 为实现相应零件的工序; M_1, M_2, M_3 为加工工序所采用的机床.

在上述调度描述中,通过设备替代与工序替代,在车间内零件 H_1 与 H_2 各有 2 种加工实现方式,对于这 2 种方式,可以称之为零件在本车间内的 2 种工艺实现. 为此,将零件 H_i 的工艺实现采用一个三维向量 HH_i^v 表示,每个向量包括工序号、加工时间和加工该工序所需的设备资源.

$HH_i^v = [(a, t_a^v, R_a^v), \dots, (k, t_k^v, R_k^v), \dots, (b, t_b^v, R_b^v)]$. 式中, a, \dots, i, \dots, b 表示工序号, $v = 1, 2, \dots$ 表示可替代的工艺实现, R_k^v 为工艺实现 v 中工序 k 的加工设备, t_k^v 为工艺实现 v 中工序 k 的加工时间.

下面首先给出零件与工序的框架表示,然后定义机床调度所需的数据结构.

(零件号(工序数)
(工艺实现数)
(工艺实现 1(工序号,加工时间,机床号))
...
(工艺实现 v (工序号,加工时间,机床号))
);
(工序号(属于(零件号))
(优先工序(工序 1,工序 2, ..., 工序 a))
(后继工序(工序 1,工序 2, ..., 工序 b))
);

```
struct op_maching{
    unsigned char o; //工序号
    float mach_time; //加工时间
    unsigned char mach; } //机床号
struct order{
    unsigned char order_num; //定义零件号
    unsigned char plan_num;
    //定义零件 order_num 的工艺实现号
    struct op_machine op_mach[PLAN_LENGTH];
    //定义一种工艺实现,PLAN_LENGTH 为
    //该工艺实/现的工序数}
```

```
struct individual{
    struct order or[CHROM_LENGTH];
    //定义一个染色体(调度计划),CHROM_
    //LENGTH 为染色体的需求数
    float fitness; //定义染色体的适应度
} newpop[POP_SIZE], oldpop[POP_SIZE], * p_old-
pop, * p_newpop; //POP_SIZE 为群体规模
```

3.4 调度的优化评价函数

车间静态调度的任务是:考虑前一天的拖期任务,详细地安排车间日作业计划,给出各类机床上的加工顺序、不同机床间的流转,确保关键资源不出现空闲等待以及宽裕度小的零件先加工等,为此以加权平均流通时间最小为优化目标建立静态调度的评价函数. 通过约束工序的流通时间保证机床以最少的加工时间完成工序的加工,不仅为零件在机床间的流转提供时间保障,而且保证关键资源上的任务尽快完成(由于计划的滚动性,可以将第二天在该资源上加工的计划提前,以保证资源不出现等待时间),另外为了使宽裕度小的零件先加工,加权平均流通时间增加优先级(权重) w_i 的概念以反应工序 i 的宽裕程度.

为此,给出加权平均流通时间 F_w 评价指标,对调度计划进行评价:

$$F_w = \sum_{i=1}^n w_i F_i / \sum_{i=1}^n w_i,$$

其中, n 为该调度计划中工序的总数, w_i 为工序 i 的权重, F_i 为工序 i 的流通时间(工序 i 的完成时间 C_i 与准备时间 r_i 之差,即 $F_i = C_i - r_i$),若所有 w_i 均为 1,则不考虑各加工工序的优先级.

4 车间日作业计划调度问题的 AAGA 算法设计

在给出车间静态调度的表示方法之后,建立如

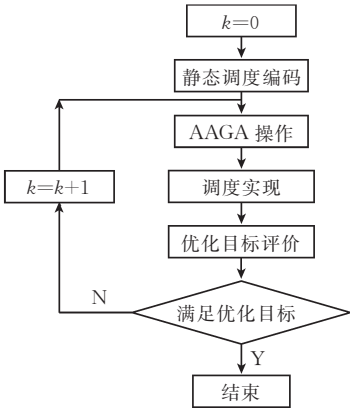


图 1 基于 AAGA 的车间静态调度流程

图 1 所示基于 AAGA 算法的静态调度求解流程图，该算法主要由静态调度编码、AAGA 操作、静态调度生成、优化目标评价等模块组成，每个模块又对应于一种算法。

4.1 调度编码

静态调度编码是将所有待加工零件的工序依次编号为 1,2,3,⋯,随机生成一条满足工序约束关系的编码链 L 。其中，零件工序字符的表示形式与工艺优化中编码的表示形式相同，编码链中工序的先后顺序表示该工序进行调度推算的优先次序，每一条编码链均对应一个确定的调度方案，为了生成初始编码链，首先给出 2 个定义。

定义 2. 若一个工序的所有直接前继工序都已放入编码链中，则称该工序为可编码工序。

定义 3. 所有可编码工序的集合称为可编码工序集。

利用上述概念，产生静态调度编码链 L 的算法如下：

- 1. 初始化。给各零件的工序编号，形成调度任务集 S ；
- 2. 处理工艺约束，形成可编码工序集 S_0 ；
- 3. 随机从 S_0 集中选择一个工序，放入编码链 L ；
- 4. 重新构造可编码工序集 S_0 ；
- 5. 若 S_0 为空，结束。否则，转向步 3。

例如图 2 所示 4 个零件的 12 道工序，起初，可编码工序集 S_0 为 (1, 2, 4, 5, 6, 9, 10, 11)，随机挑选 1 放入 L ，重新生成 S_0 为 (2, 4, 5, 6, 9, 10, 11)， S_0 不为空，返回步 3，随机选 6 放入 L ，重新生成 S_0 为 (2, 4, 5, 7, 9, 10, 11)， S_0 不为空，⋯。上述过程，生成的 L 可能为 (1, 6, 2, 4, 9, 5, 7, 10, 3, 11, 8, 12)， L 也可能为 (4, 6, 2, 1, 7, 5, 9, 10, 3, 11, 8, 12) 或 (1, 6, 9, 4, 10, 5, 7, 2, 3, 11, 12, 8) 等。

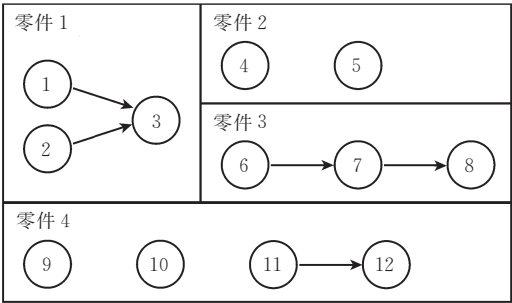


图 2 零件 1~零件 4 的工艺约束

4.2 AAGA 操作

(1) 适应度函数

为了反映工序的加权平均流通时间 F_w ，定义适

应度函数：

$$f(i)=F_w(i),$$

i 为遗传进化的代数。

(2) 选择算子

在 AAGA 整体退火选择下，选择算子按照式 (5) 计算个体的生存概率。

(3) 可替代工序链表

可替代工序的集合称为一个工序的可替代工序链表。

根据车间的工序资源状况以及工艺加工规则，形成可替代工序链表，具体算法如下：

- 1. 选择该车间资源能够加工的所有工序组成链表 L ，并令 $i=0$ ；
- 2. 对于链表 L 中所有元素，从表头开始，执行如下操作；
- 3. 选择一元素 D ，并形成子链表 $L_i=(D)$ ；
- 4. 搜索链表 L ，按照工艺加工规则，如果某一元素 U 可作为 D 的替代工序，将该元素加入到由 D 形成的子链表 L_i 中，此时 $L_i=(D,U)$ ；
- 5. 判断链表 L 空，若是，结束；否则， $i=i+1$ ，转步 3。

(4) 交叉算子

由于车间静态调度问题的特殊性，其编码的交叉操作受到工艺加工规则的约束，其交叉操作不能套用任何一种固定模式，结合可替代工序链、PMC 交叉的方法，提出部分工序匹配替换法提高交叉操作的可执行性，交叉操作的对象为静态调度编码链 L 。

部分工序匹配替换法的交叉操作中，先依据均匀随机分布产生两个位串交叉点，定义这两点之间的区域为一匹配区域，并使用部分工序匹配替换交换两个父串的匹配区域。例如两父编码链： $x_1=984|567|132$ ， $x_2=871|239|546$ ，部分工序匹配替换法按以下的步骤操作：

- 1. 将 x_1, x_2 的匹配区域赋予字符串 S_1, S_2 ；
- 2. 求 S_1, S_2 字符串长度 $LEN=LENGTH(S_1)$ ；
- 3. 对于每一个 $i, i=1, 2, \dots, LEN$ ，执行
FOR $i=1$ to LEN
IF 工序 $S_2[i]$ 是 $S_1[i]$ 的可替换工序 THEN
 $ch=S_1[i]$
 $S_1[i]=S_2[i]$
 $S_2[i]=ch$
END IF
NEXT
- 4. 将 S_1, S_2 分别插入到 x_1, x_2 的原位置。

其中， $S_1[i]$ 代表字符串 S_1 中的第 i 个字符， ch 为临时字符变量。

从而，部分工序匹配替换法交换了静态调度编码链 x_1, x_2 的部分工序，子代工序加工顺序部分地

由父编码链确定.

(5) 变异算子

选用位点随机变异算子.

由于上述交叉算子、变异算子可能产生违反工艺约束的非法解,采用‘仅当工序 a 不在工序 b 的前继工序集中时,才允许 a 与 b 位置交换’的方法予以解决.

(6) AAGA 操作算法

1. 初始化群体大小 N ,总工序数 Z ,交叉概率 P_c ,变异概率 P_m ;
2. 读取数据库中的染色体编码链,形成初始群体 $P(0)$;
3. 读取数据库中的优化目标评价值 F_w ;
4. 调用选择算子进行染色体复制,并把复制的串拷入群体 $P(g)$;
5. 置计数器 $i=1$;
6. 按交叉概率 P_c 对染色体串 C_i 进行交叉操作,形成 C_{1i} ;
7. 按变异概率 P_m 对染色体串 C_{1i} 进行变异操作,形成 C_{2i} ;
8. 判断 $i < N$,若是 $i=i+1$,转向步 6. 否则转向步 9;
9. 将形成的新群体 $P(g)$ 存入数据库.

4.3 调度问题的 AAGA 实现算法

为了生成一个调度,需记录每台机床的调度情况,由于调度到每台机床上的工序经常变动,且数目无法预先确定,因此采用如下数据结构记录机床调度情况.

```
struct schedule{
    int op;
    float s_time; //工序的起始时间
    float e_time; //工序的终止时间
    struct schedule *next;}
```

每台机床需一个链表,表头指针组成一个数组:

```
struct schedule *machine_schedule[TOTAL_MACHINE].
```

为便于染色体适应度值的计算,还需记录染色体中每个规划的开始时间和终止时间:

```
float plan_start_t[CHROM_LEN],
```

```
f_plan_finish_t[CHROM_LEN].
```

由此,可方便地实现由染色体 i 生成相应调度的方法,其过程如下:

1. 初始化 i 的机床调度链表 $*machine_schedule$,并从染色体的第一个规划 $or[l].plan_num$ 开始处理;
2. 检查 i 中所有规划是否处理完;若是,调度生成过程结束,并返回最后规划的结束时间;
3. 检查 $or[k].op_mach[m]$ 中的 m 是否为 1,若是,则以机床 $or[k].op_mach[m].mach$ 上的上一操作的结束时间为 $or[k].op_mach[m].op$ 的开始时间;否则,取该机床上前一操作结束时间和 $or[k].op_mach[m-1].op$ 的结束时间的较大值为操作 $or[k].op_mach[m].op$ 的开始时间,该操作的结束时间为其开始时间加所需占用的时间;
4. 检查 $or[k].op_mach[m]$ 中的 m 是否为 $PLAN_LENGTH$,即第 k 个规划的最后一个操作是否已处理完,若是,则记录规划结束时间,转步 7;
5. 将该操作调度加入到它所分配的机床 $or[k].op_mach[m].mach$ 的链表 $*machine_schedule$ 上;
6. $m = m + 1$; 转步 3,处理下一操作 $or[k].op_mach[m]$;
7. $k = k + 1$; 转步 2,处理 i 的下一规划.

4.4 ICAPPS 中优化目标评价

针对每一种调度方案,需建立优化目标的评价算法求其优化评价函数,算法描述如下:

1. 读取调度方案;
2. 计算 F_w ;
3. 判断是否达到优化目标要求. 如果达到要求,将评价价值存入数据库,结束. 否则,将评价价值存入数据库,转向执行 AAGA 操作.

5 ICAPPS 中车间静态调度的实例

将基于 AAGA 算法的车间静态调度模型应用于某厂某一车间某一天的计划进行调度,生成该车间的车间日作业计划.

在 3 台机床上调度图 2 所示的 4 个零件,各零件的加工实现方式及相应的工艺路线约束见表 1.

表 1 零件工艺路线约束

零件序号	零件代号	工艺实现 3
1	HB206. 1. 1-3	工艺实现 1:[(1,6,1) (2,3,2) (3,3,3)]
		工艺实现 2:[(1,6,2) (2,4,3) (3,5,1)]
		工艺实现 3:[(1,8,3) (2,8,1) (3,5,2)]
2	HB206. 1-5	工艺实现 1:[(4,4,3) (5,3,2)]
		工艺实现 2:[(4,4,1) (5,5,1)]
		工艺实现 3:[(4,4,3) (5,6,3)]
3	HB206. 2. 8	工艺实现 1:[(6,4,2) (7,5,1) (8,2,3)]
		工艺实现 2:[(6,6,3) (7,6,2) (8,4,1)]
		工艺实现 3:[(6,7,1) (7,7,1) (8,5,1)]

(续 表)

零件序号	零件代号	工艺实现 3
4	HB206. 2-3	工艺实现 1: [(9, 3, 3) (10, 3, 1) (11, 6, 3) (12, 2, 1)] 工艺实现 2: [(9, 4, 3) (10, 5, 3) (11, 7, 2) (12, 2, 3)] 工艺实现 3: [(9, 4, 2) (10, 5, 2) (11, 9, 1) (12, 4, 1)]

AAGA 所用评价指标为加权平均流通时间 F_w , 种群大小 $N=30$, 选取 $P_c=0.4, P_m=0.015$, 采用常规的 GA 算法, 则进化过程中的平均适应度值 f 见图 3 中的曲线 A 所示。

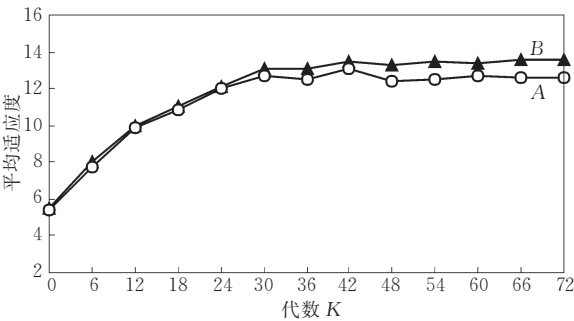


图 3 GA 与 AAGA 进化过程中的平均适应度对比曲线

采用 AAGA 算法, 初始时种群大小 N, P_c, P_m 的选择同上, 经过 24 代的进化后, 平均适应度值 f 曲线逐渐偏离曲线 A, 进化 72 代后的曲线如图 3 中的曲线 B 所示, 平均适应度值有了明显提高, 此时进化过程中的工序平均加权通过时间的变化曲线如图 4 所示。

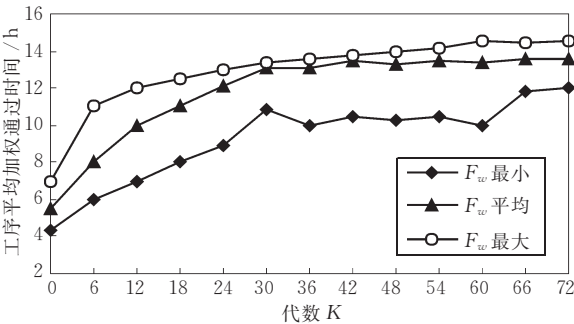


图 4 进化过程中的 F_w 的变化曲线

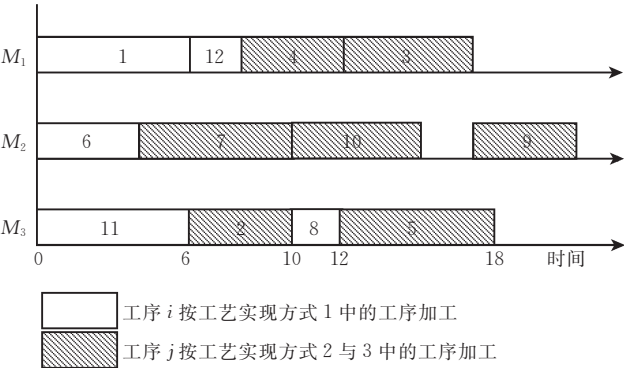


图 5 最终最优调度结果

图 5 则给出了进化 72 代后的调度结果(车间日作业计划)。

6 结 语

结合 GA 算法、最优个体保护法和 SA 算法, 提出了自适应退火遗传算法(AAGA), 建立了车间静态调度问题的模型及其基于 AAGA 的求解算法, 讨论了在 3 台机床上调度 4 个加工零件的实例. AAGA 算法针对遗传算法中组成编码串的变异概率在整个搜索过程中是固定不变的, 而且取值较小, 促使算法的求解过程很长, 且易走向局部最小值的特性, 提出自适应变异概率的概念与理论改善遗传算法的收敛速度; 算法针对选择算子对种群多样性的影响, 提出整体退火选择的方式选择杂交母体, 以克服种群早熟化, 避免过早收敛. 进一步的研究工作将包括基于 AAGA 的车间动态调度模型、基于 PSO 等其他进化算法的车间调度模型等内容。

参 考 文 献

[1] Sohlenius G. Concurrent Engineering. Annals of the CIRP, 1992, 41(2): 645-655

[2] Qi Guo-Ning, Han Yong-Sheng, Chen Jun. Computer Integrated Product Engineering. Beijing: Chinese Economy Press, 1999(in Chinese)
(祁国宁, 韩永生, 陈俊. 计算机集成产品工程. 北京: 中国经济出版社, 1999)

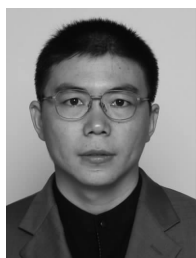
[3] Liu Min. Study on modeling and key questions of integrated system of CAPP/PPS based on distributed & dynamic process planning[Ph. D. dissertation]. Hangzhou: Zhejiang University, 1999(in Chinese)
(刘敏. 基于分布式动态工艺 CAPP/PPS 集成化建模及其关键问题研究[博士学位论文]. 杭州: 浙江大学, 1999)

[4] Xu Zong-Ben, Gao Yong. Analysis and precaution of genetic algorithm premature convergence. Science in China (Series E), 1996, 26(4): 364-375(in Chinese)
(徐宗本, 高勇. 遗传算法过早收敛现象的特征分析及其预防. 中国科学(E 辑), 1996, 26(4): 364-375)

[5] Holland J H. Adaptation in Natural and Artificial Systems. Ann Arbor: Michigan University Press, 1975

[6] Fogel D B. An introduction to simulated evolutionary optimization. IEEE Transactions on Neural Networks, 1994, 5 (1): 3-14

- [7] Goldberg D E. Genetic Algorithms in Search, Optimization and Machine Learning. Reading, MA: Addison-Wesley, 1989
- [8] Eiben A E, Aarts E H, Van Hee K M. Global convergence of genetic algorithms: An infinite markov chain analysis// Schwefel H P et al eds. Parallel Problem Solving from Nature. Heidelberg, Berlin: Springer-Verlag, 1991: 4-12
- [9] Qi X F, Palmieri F. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space, I: Basic properties of selection and mutation. IEEE Transactions on Neural networks, 1994, 5(1): 102-119
- [10] Sirag D J, Weissner D J. Towards unified thermodynamic operator//Proceedings of the International Conference on Genetic Algorithms and their Applications. Erbum Association, Hillsdale, NJ, 1987
- [11] Bosesniuk T, Ebeling W. Boltzmann-Darwin- and Heackel-strategies in optimization problems//Proceedings of the International Conference on Parallel Problem Solving from Nature. New York, 1990
- [12] Golberg D E, Mahfoud S W. A genetic algorithm for parallel simulated annealing//Proceedings of the International Conference on Parallel Problem Solving from Nature. Netherland, 1992
- [13] Zhang Jiang-She, Xu Zong-Ben, Liang Yi. Annealing genetic algorithm and its global convergence condition. Science in China (Series E), 1997, 27(2): 154-164(in Chinese)
(张讲社,徐宗本,梁怡. 整体退火遗传算法及其收敛充要条件. 中国科学(E辑), 1997, 27(2): 154-164)
- [14] Ma Jin-Tao. Genetic algorithm in electric power system optimization. Journal of Chinese Electrical Engineering, 1995, 15(5): 99-103(in Chinese)
(马晋韬. 遗传算法在电力系统无功优化中的应用. 中国电机工程学报, 1995, 15(5): 99-103)
- [15] Chen En-Hong, Liu Quan-Gui, Cai Qing-Sheng. A genetic algorithm based Job-Shop scheduling problem solving method. Journal of Software, 1998, 9(2): 139-143(in Chinese)
(陈恩红,刘全贵,蔡庆生. 基于遗传算法的 Job-Shop 调度问题求解方法. 软件学报, 1998, 9(2): 139-143)



LIU Min, born in 1970, Ph. D., associate professor. His main research interests include enterprise application integration and business process cooperation, service computing and network security.

YAN Jun-Wei, born in 1946, M. S., professor. Her main research interests include enterprise modeling and information integration.

Background

The aim of this work is to support enterprise business process integration and production plan optimization so that a proper job-shop planning can be made. This work is part of a National Science and Technology Key Program project "Model Driven Heterogenous System Integration and SOA based Platform". This project intends to research on theory and technologies for a heterogenous system integrated environment in which information portal, business process composite model, ontology and metadata warehouse are applied to build an entire integrated framework respectively in user interface layer, business process integration layer and data integration layer. As an application, this project provides one integrated

portal and a service composition platform for Shanghai EXPO 2010. In service compositing process, at first, BPEL templates about a business process, e. g. workshop operation planning process, and service ontology implementing a task, e. g. scheduling algorithm proposed in this paper, should be built, then a multi-agent model is presented to negotiate the process and service. In the integrated portal, firstly, Web based information resources existing in enterprise are abstracted into some Web resources, such as Web contexts, iframe resources, RSS resources and Web services, and then they are sorted, managed, registered and published, eventually they are reconfigured in portal.