一种基于遗传算法的自动排课系统设计

徐克圣， 张素芳

（大连交通大学 软件学院，辽宁 大连 116028）

摘 要：该文提出并实现了一种高校自动排课算法，利用遗传算法建立数据模型，定义了时间片、授课单元、切片算子、不完全两点交叉和适应度函数。通过使用遗传算法，对课程进行编排和对课表进行优化；并用 VC++ 进行编程，Matlab 进行仿真，用文件输出结果；实验结果表明，遗传算法对课表的编排和优化有着比较显著的作用。

关键词：遗传算法；适应度函数；排课

Designing of Automatic Curriculum Scheduling Based on Genetic Algorithm

XU Ke-sheng , ZHANG Su-fang

(Software Academy of DaLian Jiaotong University, DaLian 116028, P.R.China)

Abstract: A time table problem(TTP) algorithm was proposed to conduct the arrangement of curriculum schedule in universities. First, a data model was set up using genetic algorithms(GA) and then we defined a timesheet chromosome representation, teaching cell representation, slice algorithm operator, two points of incomplete cross and fitness function. Using genetic algorithm, we arranged the curriculums and optimized the curriculum schedules,then programmed by VC++ and simulated by Matlab, and the results were output by files. The results indicate that genetic algorithm is very useful to arranging and optimization of curriculum schedule.

Key words:genetic algorithm; fitness function; curriculum schedule

1 引言

自动排课算法（TTP: Time Table Problem），是一个 NP 完全问题，集成了时间、空间的双重约束。从数学上讲，排课问题是一个在时间、教师、学生和教室四维空间，以教学计划和各种特殊要求为制约条件的组合规划问题。其实质就是解决各因素之间的冲突。无冲突是指不存在一个教师被同时安排给一个以上的班级上课，或一个以上教师被同时安排给同一个班级上课等。

在众多软件公司研究的排课系统中，较多地采用传统回溯算法，但真正投入应用的排课软件却很少。原因在于，如果仅仅采用简单的回溯算法，通常情况下得到课表的适应度非常低，并且算法的复杂度比较高，通常为 $m\times O(n^3)$。如某一门课程连续两天在同一时间上课，其他时间却没有安排此课程；或者不能满足教师连堂课的要求。然而，如果想得到一张高质量的课表，就要设定一周多学时课程的上课时间间隔，也不合适，因为初始排课时总的搜索空间非常大（班级数 × 周学时数 ^ 总课程数），对一个有 12 个班级，周学时为 20(4×5)，共有 20 门课的学校，这样的搜索空间是 1.25829×10^{27}；即使大部分课程已安排完毕，其余课程在有限的搜索空间中搜索合适的上课时间，由于定义了不适当的搜索间隔，可能永远也搜索不到合适的上课时间。显然，这样的效果是不能令人满意的。为了解决这一问题，考虑把人工智能的知识引入这一领域，结果发现使用遗传算法，可以大大减少搜索空间，使用适应度函数评估个体能够找到最优解，使用自适应的交叉和变异能使最优解尽快收敛。

2 排课中的基本问题

2.1 名词解释

（1）时间片：任一上课时间段，2 个小时为一个时间片，一天内共分配 4 个时间片。

（2）学时：40-60 分钟的上课时间，即为一个学时。

（3）授课单元：根据教学计划得到的课程、教师和班级的对应关系。如《C 语言程序设计》由 1066 号教师为 200501002 班讲授，这样一个对应关系就是一个授课单元。

2.2 约束定义

设矩阵 X 表示一个可能的课表，其行值对应全部教师编号，列值为一周内可用的时间片编号，它记录了在不同的时间片每个教师要上课的次数。设矩阵 Y 的行值为课程编号，课程编号是对学校全部班级的所有课程进行统一编号，列值为一天内的时间片号，若课程 i 的某一次课安排在时间片 j，则 y_{ij} 为 1，否则为 0。设 m 为全校可用

基金项目：符合国际标准的网络化工业产品零件库技术应用研究（05L040 ）

的教室总数，n 为全校的班级数，p 为学校全部班级的课程种类数，q 为一周时间内各个教室的时间片数总和，t 为教师总数，w 为一周内可用的时间片数，对于一个给定的班级 u，其课程表位于 Y 矩阵的 lu–lu' 行。

将约束条件分为两级约束，一级约束为必须满足的基本约束，二级约束是在一级约束满足的基础上进行调整，以便更好地满足要求。

一级约束条件集合 $Req=\{req_1, req_2, req_3, req_4, req_5, req_6, req_7\}$ 的描述如下：

req1：某一时间片内一个教师只能上一门课程；

$x_{rs} \le 1 \quad r \in [0, t-1],\ s \in [0, w-1]$

req2：某一教室的某一时间片只能被一门课程占用；

$$\sum_{i}^{q-1} y_{ij} \le 1 \quad j \in [0, q-1]$$

req3：某课程 m 必须安排在预定的时间片 n 上；

$$y_{ij} = 1 \quad i = m,\ j = n$$

req4：某教师 m 在时间片 n 时不能上课；

$$x_{ij} = 0 \quad i = m,\ j = n$$

req5：某班学生在某一时间片只能被安排在一个教室上课；

$$\sum_{a=lu}^{lu'} \sum_{b=0}^{m-1} y_{a(b*w+v)} \le 1 \quad v \in [0, w-1]$$

req6：同一课程不要在连续的时间片内连续的开课；

$$y_{ij} * y_{i(j-1)} = 0 \quad i \in [0, p-1],\ j \in [0, q-1]$$

req7：某些课程对教学设备有特殊的要求。

实验中，此约束条件在表示课程的文件中对课程要求的教学设备进行说明。

3 求解算法

3.1 编码

如图 1 所示，定义一个三维矩阵 A={Time,Day,ClassRoom}，作为时间片集合，Time 代表每天的教学时间片，Day 表示每周的上课天数，ClassRoom 表示教室。根据教学计划的要求，再定义一个三维矩阵 B={Curricula,Teacher,Class} 来表示授课单元集合，Curricula 代表课程编号，Teacher 代表教师编号，Class 代表自然班级编号。最终的排课方案就是把集合 B 无冲突地对应到集合 A 中。

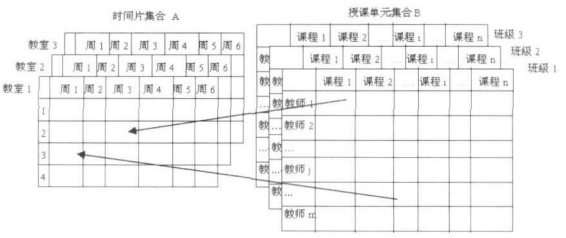

图 1　时间片集合和授课单元集合

3.2 适应度函数

整个遗传算法是在适应度函数的引导下进行的，在实验中主要考虑 5 个约束条件：

p1：较高的教室利用率（上课人数 / 教室的座位数）；

p2：有特殊要求的课程指定特殊的教室；

p3：教师上课时间没有冲突；

p4：班级上课时间没有冲突；

p5：一周内课程的均匀分布；

$F=\{\alpha, \beta, \gamma, \varsigma, \eta, \theta\}$，式中

$$\alpha = \begin{cases} x & 0 \le x \le 1 \\ 0 & x \ge 1 \end{cases}$$

x 为排课单元中选课学生人数除以教室座位数；

$$\beta = \begin{cases} 1 & \text{特殊课程安排在指定教室} \\ 0 & \text{特殊课程未安排在指定教室} \end{cases}$$

$$\gamma = \begin{cases} 1 & \text{教师上课时间没有冲突} \\ 0 & \text{教师上课时间有冲突} \end{cases}$$

$$\varsigma = \begin{cases} 1 & \text{班级上课时间没有冲突} \\ 0 & \text{班级上课时间有冲突} \end{cases}$$

$$\eta = \begin{cases} 1 & \text{1周内}j\text{门课间隔大于3个授课单元} \\ 0 & \text{1周内}j\text{门课间隔小于3个授课单元} \end{cases}$$

θ 为全部其他约束的适应度之和，如教师要求课程安排在晚上，学生选修课程时间冲突等。

适应度函数为 $F=\sum_{i=1}^{0} W_{ijmn} F[i] + Count * C$，其中 W_{ijmn} 为第 i 门课程对应于第 j 个教室的星期 m 的第 n 个时间片的权重。Count 为集合 B 中某个单元的冲突次数，C 为一负数，其绝对值足够大，以至于只要出现一次冲突，该适应值便为负，这样便于终止准则的选定。因为所求解要求无任何冲突，容易造成各个体间适应值相差过大的情况，所以采用线性排名的策略。

3.3 算子和自适应概率

3.3.1 切片算子

根据编码问题可以将切片算子分为切片算子 π_1 和切片算子 π_2。π_1 用于对时间片进行操作，如立方体 A 代表一

个时间片，实行切片操作$\Pi_{1(A,\pi)}$，其中π代表要切片的维数，如$\pi = ClassRoom$，运算结果是教室课程表，简称教室切片。立方体B代表一个授课单元集合，实行切片操作$\prod_2(B,\omega)$，其中$\omega = Class$，运算结果是班级课程表，简称班级切片。通过类似的方法可以得到教师课程表。在本实验中，把教室切片和班级切片作为基因个体，教师切片仅在输出教师课程表时使用。

下面矩阵是用切片算子形成的班级切片，对于一个班级而言，由于这学期开的课程和教师是相对固定的，可以把课程和教师当作同一变量来对待。

班级切片 $B(\ldots,\ldots,200509001)=\begin{bmatrix}1029 & 0 & 1053 & 0 & 1024 & 1019\\ 0 & 1011 & 1065 & 1008 & 0 & 1039\\ 1018 & 1005 & 0 & 1054 & 1067 & 0\\ 0 & -1 & 1018 & 1034 & -1 & 0\end{bmatrix}$ 表示了200509001班一周的排课情况，其中0表示没有排课，−1表示连续上课，其他数字表示课程编号。矩阵代表的具体含义如下：

$$B(\ldots,\ldots,200509001)=\begin{bmatrix}\text{数值分析} & [\,] & \text{数值分析} & [\,] & \text{第一外语} & \text{第二外语(日)}\\ [\,] & \text{英语口语} & \text{遗传算法} & \text{面向对象} & [\,] & \text{第二外语(法)}\\ \text{JAVA程序设计} & \text{辩证法} & [\,] & \text{矩阵论} & \text{JAVA程序设计} & [\,]\\ [\,] & \sim\sim & \text{矩阵论} & \text{软件工程} & \sim\sim & [\,]\end{bmatrix}$$

3.3.2 交叉操作

选择操作虽然能够从旧群体中选出优异个体，但不能产生新个体，因此遗传算法提出了交叉操作，进而模拟生物进化过程中的繁殖现象，通过父类染色体的交叉操作，来产生新的优良品种。

一般来讲，多点交叉采用较少，单点交叉对优异个体影响较大，所以本试验采用“不完全两点交叉”方法[3]，即在父类中有相同遗传因子的位置进行互换，剩余遗传因子顺序移入的交叉方法。

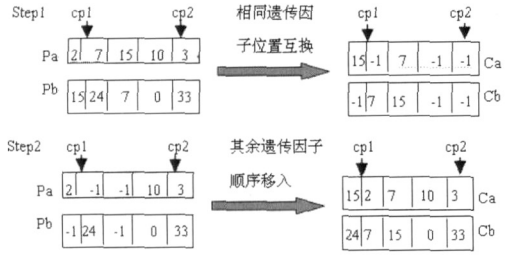

图2 不完全两点交叉处理

如图2所示，表示了本实验所用的交叉过程，cp1、cp2代表两个交叉点，pa、pb代表排课过程中形成的教师课表的部分父类染色体，数字代表教师编号，Ca、Cb代表子类交叉点间的部分染色体。

(1) 将交叉点间的两个父类染色体Pa、Pb中相同的遗传因子按位置分别拷贝到子类染色体Cb、Ca中，子类的交叉点和父类的交叉点位置相同；

(2) 在Ca、Cb中其他的基因位置处插入−1；

(3) 在Pa、Pb中有相同遗传因子的位置分别插入−1，参照图2−step1；

(4) 将Pa、Pb中其余遗传因子顺序移入Ca、Cb中，参照图2−step2。

3.3.3 交叉和变异概率

交叉概率Pc和变异概率Pm的选取在很大程度上影响算法的收敛速度和近优解的质量，当前代的最优个体不参于交叉和变异，较优个体多参于交叉和变异，以便加快算法的搜索效率和有效防止陷于局部最优解，从而采用自适应的调整概率[4]：

$$P_c=\begin{cases}k_1\sin\left(\frac{\pi}{2}\times\frac{f_{\max}-f_c}{f_{\max}-f_{avg}}\right), & \text{当} f_c \geq f_{avg};\\ k_2, & \text{当} f_c < f_{avg}.\end{cases}$$

$$0<k_1,k_2,k_3,k_4\leq 1.$$

$$P_m=\begin{cases}k_3\sin\left(\frac{\pi}{2}\times\frac{f_{\max}-f_m}{f_{\max}-f_{avg}}\right), & \text{当} f_m \geq f_{avg};\\ k_4, & \text{当} f_m < f_{avg}.\end{cases}$$

其中，f_{max}是群体中的最大适应度函数值，f_{avg}是平均适应度函数值，f_c是交叉的两染色体串中适应度较大者，f_m是变异串的适应度函数值。当$f_c=f_{max}$，$f_m=f_{max}$时概率为0，当前代可以直接转入下一代。

3.3.4 变异算子

当运算结果接近最优解领域时，利用变异算子的局部的随即搜索能力可以加速向最优解收敛，变异算子的主要作用是调整授课单元和时间片的冲突，可以选择教室中的任意时间片进行变异。

3.4 终止条件

同时满足以下两个条件：

(1) 该种群中最大适应值为一正数；

(2) 当前种群中最大适应值与以前各代中最大适应值相差不大，这时说明进化效果已不太显著，再进化下去没有必要，仍有冲突可以手工调整。

3.5 TTP算法

输入：时间片集合A、授课单元集合B（包括专业课程数据，公共课程数据，教师不能上课的时间数据等）。

输出：教室课表、班级课表、教师课表

step1：根据时间片集合生成空白教室课表 $A(...,...,ClassRoom_1), A(...,...,ClassRoom_2),..., A(...,...,ClassRoom_m)$；

step2：随机地从B集合中取出一个 b_{ijk} $i\in[0,p-1], j\in[0,t-1], k\in[0,n-1]$，根据集合 R_{eq} 把 b_{ijk} 安排到 $A(...,...,ClassRoom_k)$ $k\in[0,m-1]$ 对应的时间片内，直到 b_{ijk} $i=p-1, j=t-1, k=n-1$ 为止。

step3：计算每个个体的适应度函数F；

step4：如果适应度函数值达到一定的要求（如连续代适应度不再改变），或者进化代数达到一个指定数值，则跳转到9；

step5：重新计算Pc、Pm，再按概率选择优秀的排课方案，淘汰适应度差的排课方案；

step6：进行教室切片和班级切片操作，对教室切片进行交叉运算；

step7：进行变异运算；

step8：跳转到4；

step9：从较优群体中选择一个最优切片方案，分别得到教室课表、班级课表和教师课表。

4 实验结果分析

选择大连交通大学研究生部为测试范例，该研究生部2005-2006学年上学期共216个排课单元，课程数83个，9间教室，152个座位，约295名学生，12个教学班级。

4.1 实验结果

如图3所示，此算法的排课输出结果——教室课表，

图3 输出结果（教室课表）

图4显示了适应度函数值与遗传代数的关系，可以看出适应度函数随着遗传代数的增加呈上升趋势，当进化到40代时，适应度函数值为2980，当进化到50代时最大适应度函数值为3000，在PIIIPC机上运行，进化到50代停止，耗时1232.438s。pCrossover和pMutation是最终的交叉和变异概率，显示结果无一例教室、教师、班级冲突。

4.2 分析

遗传算法对适应度的提升是比较明显的，从40代开始收敛于2980-3000。这也是遗传算法的特点，它能够很快收敛于近优解，但却在最优解附近徘徊甚至停止。这是遗传算法现在还没有很好解决的问题，即很难在收敛速度和收敛于最优解之间取得平衡。

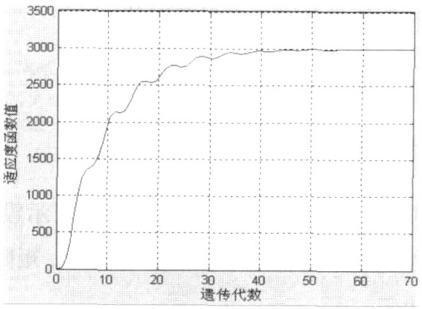

图4 适应度函数与遗传代数的关系

5 结论

本文论述了利用遗传算法求解高校排课问题，实验证明遗传算法能够大大减少搜索空间，文中提出的染色体编码方案、切片算子、不完全两点交叉方法和适应度函数是可行的，适应度函数值能够随着遗传代数的增加而呈不断上升趋势，自适应的交叉和变异能使解空间尽快收敛到最优解，实验结果令人满意。

参考文献：

[1]CALDEIRA J.P., Rosa A.C. School Timetabling Using Genetic Search[A]. PATAT97[C]. Toronto, 1997.

[2] 唐勇，唐雪飞．基于遗传算法的排课系统[J]．计算机应用．2002,22(10):93-94,97.

[3] 张春梅，行飞．用自适应的遗传算法求解大学课表安排问题[J]．内蒙古大学学报（自然科学版）．2002,33(4):459-464.

[4] 平野广美．遗传的アルゴリズムと遗传的プログラミング[M]． 3. 东京： パーソナルメディア株式会社， 2003. 207-224.

作者简介：徐克圣(1965年－),男,副教授,硕士,主要研究方向：制造业信息化、算法分析与设计；张素芳(1981年－),女,在读研究生,硕士,主要研究方向：算法分析与设计、制造业信息化.

收稿日期：2007-07-11