文章编号: 1000-6788(2012)09-2031-06 中图分类号: TP301 文献标志码: A

采用十进制免疫遗传算法求解高校排课问题

李红婵, 朱颢东

(郑州轻工业学院 计算机与通信工程学院, 郑州 450002)

摘 要 论文深入分析了高校排课问题, 建立了其数学优化模型, 构建了其基本求解框架. 针对高校排课问题的特点, 引入遗传算法来加以解决, 设计了多种改进方案, 包括: 十进制编码方案、初始种群生成方案、适应度函数设计方案、免疫策略、自适应交叉概率和自适应变异概率设计方案. 仿真结果表明该算法能够满足高校排课问题的多重约束条件, 能更有效地解决高校排课问题.

关键词 高校排课问题; 遗传算法; 十进制编码; 适应度函数; 免疫策略

Decimal immunization GA used to solve UTP

LI Hong-chan, ZHU Hao-dong

(School of Computer & Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China)

Abstract University timetabling problem (UTP) was analyzed detailedly, an optimization mathematical model of UTP was established, and the framework structure to solve UTP was found. According to characteristics of UTP, GA was introduced, a variety of improved schemes were designed, includes: decimal coding scheme, initial population design scheme, fitness function design scheme, immunization strategy, adaptive crossover probability and adaptive mutation probability design scheme. Simulation results show that the proposed GA can satisfy multiple constraint conditions and resolve UTP more effectively.

Keywords UTP; GA; decimal code; fitness function; immunization strategy

1 引言

排课是教务工作中一项繁重的任务, 它关系着高校的整体教学质量和教学资源的利用. 为适应科技时代的需求, 高校专业数量、学生数量日益增多, 这使得课表编排变得更加迫切 [1−5]. 但是, 排课问题是一个多约束、多目标的组合优化问题, 已被证明是一个 NP- 问题 [6−10]. 对于这个问题, 人们已提出了多种算法来解决, 例如分组优化决策算法、分支定界算法等等 [7], 这些算法虽然对后继研究有一定的启发, 但是却存在以下不足: 1) 专家系统技术虽然可以对排课的规则知识进行有效地组织, 但是排课过程中需要的各要素关联规则很难获取, 而且求解结果也不理想; 2) 课表的优劣判定标准较少, 算法只能在问题的某一方向进行求解, 不可能对问题的多方向同时进行优化; 3) 搜索过程的启发信息依赖于实际情况, 排课问题的求解只能针对个别的实际问题, 不能形成一种通用有效的排课方法.

由于遗传算法在进化过程中仅需要影响搜索方向的目标函数和相应的适应度函数, 其整体搜索策略和优化搜索方法在搜索过程中不依赖于梯度信息或其它辅助知识, 并且它还不依赖于问题的具体领域, 对问题的种类有很强的鲁棒性 [8]. 鉴于此, 本文采用遗传算法来解决排课问题, 并详细设计了其每一步骤.

2 排课问题

2.1 排课目标

排课问题可以看作是一个资源分配问题, 也即在满足一些约束条件的前提下, 把某些定量的资源分配给各个需求个体. 其主要目标就是依据教学计划将教室、教师、班级、课程安排在一周内某一个不发生冲突的

收稿日期: 2010-05-31
资助项目: 河南省基础与前沿技术研究计划项目 (102300410266, 122300410287); 郑州轻工业学院博士科研基金 (2010BSJJ038)
作者简介: 李红婵 (1983–), 女, 河北石家庄人, 硕士, 助教, CCF 会员, 证号: E200014642M, 研究方向: 智能信息处理; 朱颢东 (1980–), 男, 河南虞城人, 博士, 讲师, CCF 会员, 证号: E200012527M, 研究方向: 文本挖掘, 智能信息处理, 计算智能.

时间里 [9].

2.2 影响排课的因素

排课过程中存在众多冲突, 其主要影响因素如下 [10]:

时间因素: 在排课问题中, 通常按周计算上课时间. 每周上课时间不超过 7 天, 每天分为上午、下午和晚上三个时间段, 每个时间段都有各自的上课节数, 如上午为 P1, 下午为 P2, 晚上为 P3. 上课的最小单元是节, 一节就是一个课时, 一般一门课程的上课时间是两个课时.

课程因素: 各门课程都有自己的编号、名称、开课院系, 并且都有各自的授课计划, 例如哪周开始、哪周结束以及每周上几个学时等等.

教室因素: 每个教室都有相应的编号、门牌号和名称, 同一时间内只能接纳一门课程的授课, 并且教室容量应该大于等于上课的人数.

班级因素: 每个班级都有编号和名称并且同一时间只能上一门课程.

教师因素: 每个教师都有编号和姓名并且同一时间只能上一门课程.

2.3 排课过程的约束条件

排课过程中的约束条件分为两类 [11−12]: 硬约束和软约束. 其中, 硬约束指的是学生、教师和教室在时空概念上出现了不可能出现的情况, 这是排课过程中最基本的约束条件, 也是众多排课模型中都涉及的约束条件; 软约束是指排课过程中满足更佳但不满足又无妨的约束条件, 它们的违反与否往往是与排课实际情况相关. 在两类约束条件之中, 硬约束是衡量排课方案是否切实可行的标准, 软约束是衡量排课方案优劣的标准, 通常判别一个排课方案的优劣标准有多个.

2.4 排课的求解目标

排课问题实质上是一个多约束、多目标的组合规划问题, 对组合规划问题如果想找到最优解, 必须有相应的约束条件来实现 [13]. 可是, 对于部分属于人文范畴的排课问题, 不可能找到充足的约束条件, 而且由于课表方案优劣差异的融合, 能够找到的解将不可避免的是一个解集合, 这个集合中的所有解都是可行的.

因此, 本文放弃了寻求 “绝对最优” 的企图, 除了基本的 “教师、班级、教室在任意时间的安排只能出现一次” 的硬约束条件外, 课表能够满足人工排课中的 “合理、实用、有特色” 的要求, 就认为被安排的课表是可行的并且相对占优的.

3 本文排课问题的数学模型

3.1 数学模型描述

假设学校有 C 个班级, G 位教师, L 门课程, R 个教室, T 个时间段, 该模型具体描述如下:

班级集合 $C=\{c_1,c_2,\cdots,c_c,\cdots,c_C\}$, 每个成员为一个班级. 各班级分别有 $\{k_1,k_2,\cdots,k_c,\cdots,k_C\}$ 人.

教师集合 $G=\{g_1,g_2,\cdots,g_g\cdots,g_G\}$, 各教师对应课程数 $\{y_1,y_2,\cdots,y_g\cdots,y_G\}$.

课程集合 $L=\{l_1,l_2,\cdots,l_l,\cdots,l_L\}$, 其中元素为课程编号. 为了简化问题求解, 同时也为了符合目前高校的实际情况, 强制要求 “每一课程对应一位教师”, 具体做法如下: 对于同一门课程, 分配给不同的老师时, 也同时给这个课程分配不同的编号, 例如: 张三和李四同时讲授数据结构, 张三讲授的数据结构编号为 001, 李四讲授的数据结构编号为 002. 各课程对应的班级数为 $\{z_1,z_2,\cdots,z_z,\cdots,z_Z\}$.

教室集合 $R=\{r_1,r_2,\cdots,r_r,\cdots,r_R\}$, 各教室可容纳人数为 $\{x_1,x_2,\cdots,x_r,\cdots,x_R\}$.

时间集合: $T=\{t_1,t_2,\cdots,t_t,\cdots,t_T\}$.

时间与教室对的笛卡尔积为: $M=T\times R=(t_1,r_1),(t_2,r_2),\cdots,(t_t,r_r),\cdots,(t_T,r_R)$, 排课问题由此转化成为一门课寻找一个合适的时间教室对.

3.2 模型中的硬约束条件

同一时间, 一个班级不能同时有一门以上的课程, 即 $\sum_{g=1}^{G}\sum_{l=1}^{L}\sum_{r=1}^{R}c_cg_gl_lr_rt_t\le 1$, 其中 $c=1,2,\cdots,C$; $t=1,2,\cdots,T$. 班级 c_c 在时间 t_t、教室 r_r 中由教师 g_g 讲授课程 l_l, 表示为 $c_cg_gl_lr_rt_t=1$, 反之为 0.

同一时间, 一个教师不能同时有一门以上的课程, 即 $\sum_{c=1}^{C}\sum_{l=1}^{L}\sum_{r=1}^{R}c_cg_gl_lr_rt_t\le 1$, 其中 $g=1,2,\cdots,G$; $t=1,2,\cdots,T$. 教师 g_g 在时间 t_t、教室 r_r 中给班级 c_c 讲授课程 l_l, 表示为 $c_cg_gl_lr_rt_t=1$, 反之为 0.

同一时间, 一个教室不能同时有一门以上的课, 即 $\sum_{c=1}^{C}\sum_{g=1}^{G}\sum_{l=1}^{L}c_cg_gl_lr_rt_t\le 1$, 其中 $r=1,2,\cdots,R$; $t=1,2,\cdots,T$. 教室 r_r 在时间 t_t 由教师 g_g 给班级 c_c 讲授课程 l_l, 表示为 $c_cg_gl_lr_rt_t=1$, 反之为 0.

分配的教室 r_r 可容纳人数 x_r 应该大于等于上课的班级 c_c 的学生人数 k_c, 即 $x_r \geq k_c$.

3.3 模型中的软约束条件

本文模型中的软约束条件如下:

1) 由于课程授课效果与授课节次紧密相关, 在课程编排过程中较重要的课程最大程度地安排在授课效果较好的节次中.

2) 满足教师所提出的上课时间和地点的要求.

3) 多学时课程的周次安排要错开. 在实际的排课过程中, 一般对于每周多学时 $(n \geq 4)$ 的课程, 应该能够尽量将其隔 1 天以上安排, 才能保证有较好的教学效果.

4) 资源的利用率问题. 一个良好的课表安排结果可以节省大量资源.

3.4 本文排课问题基本求解框架

根据确立的排课目标和建立的数学模型, 把排课问题的求解过程分为两个部分来进行: 第一部分根据教学任务书将无序的原始数据进行随机可行排课操作, 生成有序的最终数据表, 这部分使用一个初始种群生成算法来实现; 第二部分应用遗传算法对产生的随机可行排课方案进行 "全局优化".

4 本文遗传算法设计方案

4.1 十进制编码方案

经典遗传算法所采用的二进制编码不能很好地反映排课问题的实际特点和本文排课问题的数学模型, 因此, 本文采用十进制编码. 在排课问题研究中, 每条染色体代表每位教师的课表, 其十进制结构如表 1 所示:

表 1　十进制编码

班级 ID	课程 ID	教师 ID	教室 ID	时间 ID

在染色体中, 班级 ID、课程 ID、教师 ID、教室 ID、时间 ID 均采用 4 位十进制编码, 共计 20 位. 例如: 某一个教师编号为 0050, 要教授 "编译原理" 这门课 ("编译原理" 课程编号为 7003, 周学时为 6, 班级为 2004, 随机产生上课时间, 随机选择容纳人数大于班级总人数的教室, 则可生成染色体如:"20047003005064232241" 表示编号为 2004 的班级上编号为 0050 的老师讲授的编号为 7003 的软件工程课程在 6423 教室, 时间为每周二第二个教学单元 (即上午 3、4 节) 和星期四第一个教学单元 (即上午 1、2 节).

按如上编码, 两条染色体对后 8 位作交叉操作, 不会影响到每位教师所教授的课程, 也不会造成教师课表内含其他教师的教授课程或每代演化后染色体结构不合理等问题.

4.2 适应度函数设计方案

排课问题既要满足硬约束条件, 即资源分配不冲突外, 又要满足软约束条件, 即资源分配效果最佳. 遗传算法在进化过程中以个体适应度大小为依据来获取下一代种群. 适应度函数设定的好坏直接影响到遗传算法的收敛速度和能否找到最优解. 本文适应度函数的设计思想是由于排课问题的软约束有多个, 即优化目标有多个, 因此采用多目标化化和适应度函数相结合的个体适应度函数.

1) 重要的课程尽量安排在教学效果较好的节次. 用 $\alpha_i (i = 1, 2, 3, 4, 5)$, 每天 5 个教学单元, 其中, 每天的第 1、3、5 教学单元效果较好, 记为 $\alpha_i = 1 (i = 1, 3, 5)$, 第 2、4 教学单元效果较差, 故记 $\alpha_i = 0 (i = 2, 4)$. 用 $\beta_j (j = 1, 2, 3, 4)$ 表示课程的重要程度, 也称权重, 我们把课程分为选修课、基础课、专业课、学位课, 其权重分别为 1、2、3、4. 由此, 优化目标:

$$\max (f_1) = \sum (\alpha_i \times \beta_j) \tag{1}$$

2) 尽量满足教师所提出的上课时间和地点的要求, 根据教师的职称设定系数 $\chi_i (i = 1, 2, 3, 4)$, 分别为助教、讲师、副教授、教授, 其值分别为 1, 2, 3, 4. 教师在给定时间上课的意愿为 δ_i, δ_i=0, 1, 2, 分别表示不愿意、无所谓和愿意. 优化目标:

$$\max (f_2) = \sum (\chi_i \times \delta_j) \tag{2}$$

3) 在课程编排时, 对那些每周多学时 $(n \geq 4)$ 的课程, 应该能够尽量将其隔 1 天以上安排, 才能保证有较好的教学效果. β_i 含义同式 (1) 相同; 用 $\varepsilon_i (i = 1, 2, 3, 4)$ 表示一门课程安排隔 i 天的教学效果系数, 其值

分别为 1, 3, 4, 2. 优化目标:

$$\max(f_3) = \sum(\beta_i \times \varepsilon_j) \tag{3}$$

4) 资源的利用率问题. 一个良好的课表安排结果可以节省大量资源. 一次授课中, 在一个教室上课的班级 c_c 的学生人数 k_c 与该教室的容量 r_r 的比值越大, 资源利用率越高, 最大值为 1, 即刚好容纳, 优化目标为:

$$\max(f_4) = \sum \frac{k_c}{r_r} \tag{4}$$

综上所述, 排课问题的适应度函数根据各个目标值加权所得, 即:

$$F = \sum_{i=1}^{4} \theta_i \times f_i \tag{5}$$

其中, $\theta_i (i = 1, 2, 3, 4)$ 的值可以由管理人员自行定义, 代表着排课各个目标的重要程度, 本文分别取值为 3, 1, 2, 4.

4.3 初始种群生成方案

初始过程主要为后面的各种操作提供初始种群, 一般是通过随机搜索的方式产生 [14]. 但是随机搜索方式生成的初始种群适应度一般都非常低, 为了提高初始种群的适应度, 本文提供了一个启发式搜索算法来生成初始种群, 当存在冲突时, 辅以空闲空间随机搜索和调动算法来消除冲突.

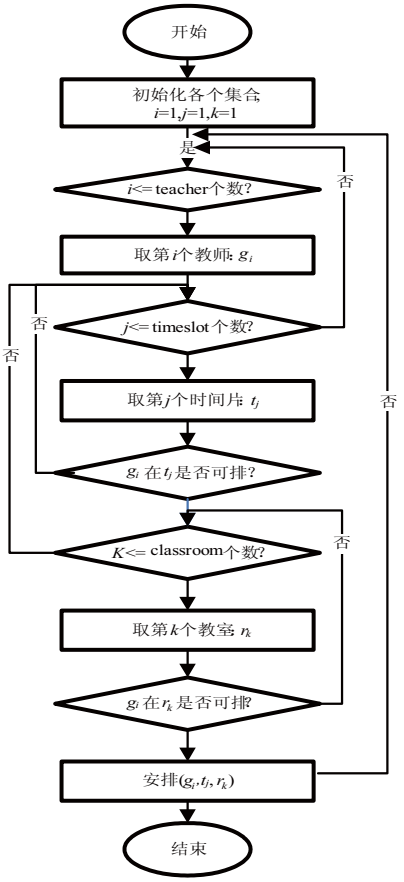

图 1 初始种群生成算法

4.4 免疫策略

经典遗传算法存在 “早熟” 现象, 从而使算法陷入局部最优. 本文利用免疫原理来防止 GA 的早熟收敛, 主要是利用免疫抑制原理通过抗体浓度来影响 GA 的选择压力, 从而保持 GA 的群体多样性. 根据信息熵定义抗体间的亲和度及抗体的浓度, 在选择算子中引入抗体的促进和抑制机制来控制选择压力.

①抗体信息熵

设有 N 个抗体, 每个抗体的长度为 M, 采用的符号集大小为 S, 则抗体基因座 j 的信息熵 $H_j(N)$ 可定义为:

$$H_j(N) = -\sum_{i=1}^{s} p_{ij} \ln p_{ij} \tag{6}$$

式中 N 是抗体的数量, p_{ij} 是第 i 个符号出现在基因座 j 上的概率, 可定义为:

p_{ij} = 基因座 j 上出现第 i 个符号的总个数/N.

N 个抗体间的平均信息熵定义为:

$$H(N) = \frac{1}{M} \sum_{j=1}^{M} H_j(N) \tag{7}$$

②抗体亲和度

两个抗体 u 和 v 之间的亲和度定义为:

$$A_{u,v} = \frac{1}{1 + H(2)} \tag{8}$$

它实际上描述了两个抗体 u 和 v 之间的相似程度, 取值在 0 到 1 之间, 如果其值等于 1, 则表示两者的基因完全相同.

③抗体的浓度

抗体的浓度 C_j 是指群体中与其相似的抗体所占的比重, 定义为: C_j = 和抗体 j 的亲和度大于某个常数的抗体数量/抗体的总数量.

根据以上定义, 对抗体进行抑制和促进, 这通过控制选择概率来实现, 使得当个体适应度越大时, 则选择概率越大, 个体浓度越大时, 则选择概率越小, 这样既可保留适应度高的个体又可确保个体多样性, 改善未成熟收敛.

4.5 自适应交叉和变异操作设计方案

交叉概率 p_c 和变异概率 p_m 的选取在很大程度上影响算法的收敛速度和解的质量. 本文设计思想如下: 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, 具体公式表示如下:

$$p_c = \begin{cases} a_1 \sin\left(\dfrac{\pi}{2} \times \dfrac{f_{\max} - f_c}{f_{\max} - f_{avg}}\right), & \text{若 } f_c \geq f_{avg} \\ a_2, & \text{若 } f_c < f_{avg} \end{cases} \tag{9}$$

$$p_m = \begin{cases} a_3 \sin\left(\dfrac{\pi}{2} \times \dfrac{f_{\max} - f_m}{f_{\max} - f_{avg}}\right), & \text{若 } f_m \geq f_{avg} \\ a_4, & \text{若 } f_m < f_{avg} \end{cases} \tag{10}$$

其中 a_1, a_2, a_3, a_4 为 0 到 1 的随机数, $f_{\max}$ 是当前群体中最优个体的适应度值, f_{avg} 是当前群体的平均适应度值, f_c 是参加交叉操作的个体中较大的适应度值, f_m 是变异个体的适应度值.

5 仿真实验

5.1 实验数据

本实验所用数据各要素分布如表 2 所示.

表 2　实验数据

要素	学生	教师	班级	课程	教室	开课任务书
数量	6200	387	125	669	168	669

5.2 实验参数设置

由于算法中交叉概率和变异概率是自适应生成, 这两个参数无须设置, 其它相关参数以及参数含义为:

1) POPSize 表示种群规模. POPSize 过小时, 目标值波动较大, 不能反映优化的各个目标; 过大时, 各目标值虽然收敛, 但收敛时间长, 较耗内存. 由表 2 可知实验中有 125 班级, 则本文取种群规模为 150.

2) MAXGen 表示最大迭代代数. MAXGen 过小时, 各目标不收敛且部分目标有下降趋势; 过大时, 虽收敛, 但不是全局最优, 本文取 1000.

5.3 实验结果

实验中, 使用文献 [15] 的方法 (简称 GA) 与本文方法 (简称 DIGA) 做对比.

实验做了 10 次, 每进化 100 代就记录一次该种群的最佳适应度值, 取这 10 次记录的最佳适应度值的平均值作为适应度值方面的实验结果, 如图 2 所示; 每进化 100 代就记录一个时间, 取这 10 次记录的时间平均值作为时间方面的实验结果, 如图 3 所示.

从图 2 和图 3 可以看出, 在每隔 100 代计算的 10 次最佳适应度的平均值和平均时间上, 本文方法都优于文献 [15] 提供的方法, 这说明使用本文方法解决排课问题总体效果较好.

从表 3 可以看出采用本文方法后, 无论是学生主干课程每周上课天数, 还是同一课程上课间隔天数以及学生每天平均上课节数都得到了一定程度的改善. 从表 3 还可以看出采用本方法后, 在教室资源总体利用率、遗漏排课程总数、教师总体满意度、排课总体冲突率方面都有不同程度的改善.

6 结束语

论文在深入分析排课问题的基础上, 建立了排课问题的数学模型, 给出了求解框架. 针对排课问题的特点引入遗传算法并设计了该算法的各步改进方案. 仿真结果表明本文方法在一定程度上能够解决排课问题, 取得了较好的效果.

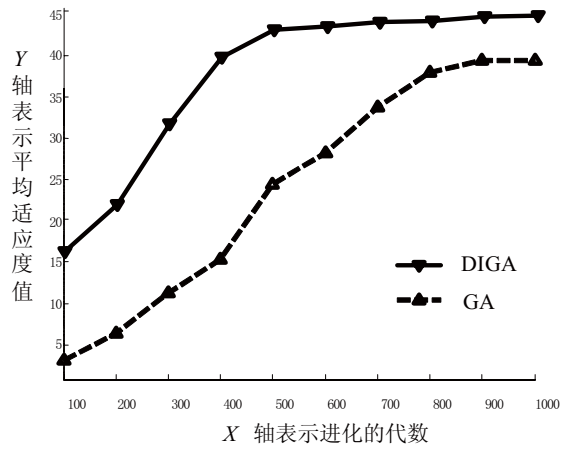

图 2 适应度比较结果图

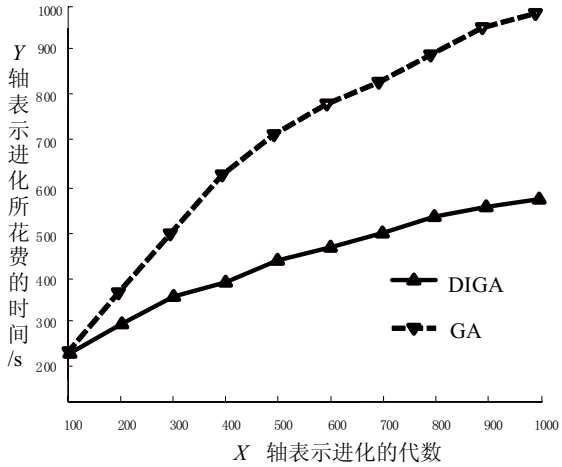

图 3 消耗时间比较结果

表 3 排课结果比较

方法	学生主干课程每周上课天数	同一课程上课间隔期	学生每天平均上课节数	教室资源总体利用率	遗漏排课程总数	教师总体满意度	排课总体冲突率
GA	2–2.5 天	1 天 –1.4 天	4 节 –8 节	86%	17	84%	15%
DIGA	2.5–2.7 天	1.3 天 –1.5 天	4 节 –5 节	98%	2	96%	1%

参考文献

[1] Burke E K, McCollum B, Meisels A, et al. A graph-based hyper-heuristic for educational timetabling problem[J]. European Journal of Operational Research, 2007, 176(1): 177–192.

[2] González-del-Campo R, Sáenz-Pérez F. Programmed search in a timetabling problem over finite domains[J]. Electronic Notes in Theoretical Computer Science, 2007, 177(1): 253–267.

[3] Al-Yakoob S M, Sherali H D. A mixed-integer programming approach to a class timetabling problem: A case study with gender policies and traffic considerations[J]. European Journal of Operational Research, 2007, 180(3): 1028–1044.

[4] De Causmaecker P, Demeester P, Berghe G V. A decomposed metaheuristic approach for a real-world university timetabling problem[J]. European Journal of Operational Research, 2009, 195(1): 307–318.

[5] Adewumi A O, Sawyerr B A, Montaz A M. A heuristic solution to the university timetabling problem[J]. Engineering Computations, 2009, 26(8): 972–984.

[6] Shi J. Research on application of IGA (immune genetic algorithm) to the solution of course-timetabling problem[C]// The 2009 4th International Conference on Computer Science and Education, USA: IEEE Computer Society, 2009: 1105–1109.

[7] Aladag C H, Hocaoglu G, Basaran M A. The effect of neighborhood structures on tabu search algorithm in solving course timetabling problem[J]. Expert Systems with Applications, 2009, 36(10): 12349–12356.

[8] Detienne B, Péridy L, Pinson E, et al. Cut generation for an employee timetabling problem[J]. European Journal of Operational Research, 2009, 197(3): 1178–1184.

[9] Pillay N, Banzhaf W. A study of heuristic combinations for hyper-heuristic systems for the uncapacitated examination timetabling problem[J]. European Journal of Operational Research, 2009, 197(2): 482–491.

[10] Lee Y S, Chen C Y. A heuristic for the train pathing and timetabling problem[J]. Transportation Research Part B: Methodological, 2009, 43(9): 837–851.

[11] Zhang D F, Liu Y K, Hallah R M. A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems[J]. European Journal of Operational Research, 2010, 203(3): 550–558.

[12] Berghammer R, Kehden B. Relation-algebraic specification and solution of special university timetabling problem[J]. Journal of Logic and Algebraic Programming, 2010, 79(8): 722–739.

[13] Anmar A, Masr A. Multi-neighbourhood particle collision algorithm for solving course timetabling problems[C]// The 2009 2nd Conference on Data Mining and Optimization, USA: IEEE Computer Society, 2009: 21–27.

[14] Guyon O, Lemaire P, Pinson E. Cut generation for an integrated employee timetabling and production scheduling problem[J]. European Journal of Operational Research, 2010, 201(2): 557–567.

[15] Pillay N, Banzhaf W. An informed genetic algorithm for the examination timetabling problem[J]. Applied Soft Computing, 2010, 10(2): 457–467.