

文章编号:1004-1478(2011)03-0116-06

采用十进制小生境 GA 求解高校排课问题

李红婵

(郑州轻工业学院 计算机与通信工程学院, 河南 郑州 450002)

摘要:在分析了高校排课的排课目标、影响因素、约束条件、求解目标等问题的基础上,建立了其数学优化模型,构建了其基本求解框架,设计了包括十进制编码、初始种群生成、适应度函数设计、小生境策略、自适应交叉概率和自适应变异概率设计等多种遗传算法改进方案. 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;小生境策略

中图分类号:TP301

文献标志码:A

Solution method of university timetabling problem based on decimal niche GA

LI Hong-chan

(College of Comp. and Com. Eng., Zhengzhou Univ. of Light Ind., Zhengzhou 450002, China)

Abstract: An optimization mathematical model of UTP (university timetabling problem) was established based on UTP analysis detailedly, which includes: timetabling target, main influencing factors, constraints, and solving target. The framework structure to solve UTP was founded. A variety of improved GA schemes were designed, which include: decimal encoding scheme, initial population design scheme, fitness function design scheme, niche strategy, adaptive crossover probability and adaptive mutation probability design scheme. Simulation results showed that the proposed GA can satisfy multiple constraint conditions and resolve UTP more effectively.

Key words: university timetabling problem; GA; decimal encoding; niche strategy

0 引言

排课是教务工作中一项繁重的任务,它关系着高校教学资源的利用. 随着高校学生规模的扩大和课程数量的剧增,排课问题更加突出,迫切需要一个有效的排课方法来解决这一问题^[1]. 对于这一问题,国外在1950年代末就进行了研究. 1963年 Got-

lieb^[2]提出了排课问题的数学模型,并利用匈牙利算法解决了三维线性运输问题,它标志着排课问题的研究正式进入神圣的科学殿堂,但实践中遇到的困难使人们对排课问题的题解是否存在产生了疑问. 1976年 S. Even等^[3]和 Tim B. Cooper等^[4]证明了排课问题是 NP 完全的,这既回答了排课在实践中遇到困难的原因,同时又宣布了利用计算机解决

收稿日期:2011-01-02

基金项目:河南省基础与前沿技术研究计划项目(102300410266)

作者简介:李红婵(1983—),女,河北省石家庄市人,郑州轻工业学院助教,硕士,主要研究方向为智能信息处理.

排课问题无法实现. 因为计算机难解性理论指出, 现代计算机尚未找到解决 NP 完全问题的多项式算法. S. Even 等人的论证正式确立了排课问题的学术地位, 把对课表编排复杂性的认识提高到了理论的高度.

1980 年代初, 我国开始研究排课问题, 所用方法从模拟手工排课到运用人工智能构建专家系统或决策支持系统. 南京工学院的 UTSS (university timetable scheduling system) 系统、清华大学的 TISER (timetable scheduler) 系统、大连理工大学的智能教学组织管理与课程调度系统等, 大都是模拟手工排课过程, 以“班”为单位, 运用启发式函数进行编排^[5].

经过几十年的发展, 目前国内外诸多专家对排课问题已提出了多种解决算法, 例如分组优化决策算法、分支定界算法、有限回溯等^[6]. 这些算法虽然对以后的研究有一定的启发, 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的组织, 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) 课表的优劣判定的标准较少, 算法只能在问题的某一方向进行求解, 不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, 不能形成通用的排课方法.

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他辅助知识. 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^[7]. 鉴于此, 本文拟采用遗传算法来解决排课问题, 同时提出多种改进方案, 包括十进制编码方案、多目标适应度函数设计方案、初始种群生成方案、小生境策略、自适应交叉概率和自适应变异概率设计方案.

1 排课问题

1.1 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时间里^[8].

1.2 影响排课的因素

排课过程中存在许多冲突, 其主要影响因素如下^[9].

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 d, 每天分为上午、下午和晚上 3 个时间段, 每个时间段都有各自的上课节数, 如上午为 P_1 , 下午为 P_2 , 晚上为 P_3 . 上课的最小单元是节, 1 节就是 1 个课时, 一般 1 门课程的上课时间是 2 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. 另外, 对某些课程来说, 由于开课班级较多, 从而出现班级之间难以协调、难以满足教师调课要求等情况, 对此应事先给定时间或教室.

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 1 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 1 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 1 门课程.

1.3 排课过程的约束条件

排课过程中的约束条件分为 2 类^[10]: 硬约束和软约束. 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能满足更佳、不能满足也无妨的约束条件, 它们的违反与否往往是与排课实际情况相关的. 在 2 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, 其标准是综合软、硬约束的结果.

1.4 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, 此类问题如果想找到最优解, 必须有相应的约束条件来实现^[11]. 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 1 次”这些硬约束条件外,若课表能够满足人工排课中的“合理、实用、有特色”的要求,就被认为是可行的并且是相对占优的。

2 排课问题的数学模型构建

2.1 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 $C = \{c_1, c_2, \dots, c_c, \dots, c_C\}$, 需要进行排课的班级的集合, 每个成员为 1 个班级. 各班级分别有 $\{k_1, k_2, \dots, k_c, \dots, k_C\}$ 人.

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

课程集合 $L = \{l_1, l_2, \dots, l_l, \dots, l_L\}$, 每 1 门课程对应 1 位教师, 各课程对应的班级数为 $\{z_1, z_2, \dots, z_l, \dots, z_L\}$.

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

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

时间与教室对的笛卡尔积为

$$M = T \otimes R =$$

$$(t_1, r_1), (t_2, r_2), \dots, (t_t, r_t), \dots, (t_T, r_R)$$

排课问题由此转化成为 1 门课寻找 1 个合适的时间教室对。

2.2 模型中的硬约束条件

同一时间, 1 个班级不能同时有 1 门以上的课程, 即 $\sum_{g=1}^G \sum_{l=1}^L \sum_{r=1}^R c_g g_l r_t \leq 1$, 其中 $c = 1, 2, \dots, C$; $t = 1, 2, \dots, T$. 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 $c_g g_l r_t = 1$, 反之为 0.

同一时间, 1 个教师不能同时有 1 门以上的课程, 即 $\sum_{c=1}^C \sum_{l=1}^L \sum_{r=1}^R c_g g_l r_t \leq 1$, 其中 $g = 1, 2, \dots, G$; $t = 1, 2, \dots, T$. 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 $c_g g_l r_t = 1$, 反之为 0.

同一时间, 1 个教室不能同时有 1 门以上的课, 即 $\sum_{c=1}^C \sum_{g=1}^G \sum_{l=1}^L c_g g_l r_t \leq 1$, 其中 $r = 1, 2, \dots, R$; $t = 1, 2, \dots, T$. 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 $c_g g_l r_t = 1$, 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 $x_r \geq k_c$.

2.3 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 d 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. 一个好的课表安排结果可以节省大量资源.

2.4 排课问题基本求解框架

根据本文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 2 个部分来进行: 1) 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, 这部分使用初始种群生成算法来实现; 2) 应用遗传算法对产生的随机可行排课方案进行全局优化.

3 本文遗传算法中各步骤相关设计方案

3.1 十进制编码方案

经典遗传算法使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 染色体的十进制编码方式

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 的班级在 6423 教室上编号为 0050 的老师讲授的编号为 7003 的课程, 时间为每周二第 2 个教学单元 (即上午 3, 4 节) 和每周四第 1 个教学单元 (即上午 1, 2 节).

按如上编码, 2 条染色体对后 8 位作交叉操作,

不会影响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。

3.2 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;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,因此采用多目标优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 $\alpha_i (i = 1, 2, 3, 4, 5)$ 表示每天 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,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,优化目标为

$$\max(f_1) = \sum (\alpha_i \times \beta_j) \quad ①$$

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 $\chi_i (i = 1, 2, 3, 4)$,分别代表助教、讲师、副教授。教师在给定时间上课的意愿为 $\delta_j, \delta_j = 0, 1, 2$,分别表示不愿意、无所谓和愿意。优化目标为

$$\max(f_2) = \sum (\chi_i \times \delta_j)$$

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 d 以上安排,才能保证有较好的教学效果。 β_j 含义与式 ① 相同;用 $\varepsilon_i (i = 1, 2, 3, 4)$ 表示一门课程安排隔 i d 的教学效果系数,其值分别为 1, 3, 4, 2。优化目标为

$$\max(f_3) = \sum (\beta_j \times \varepsilon_i)$$

4)资源的利用率问题。一个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_i 的学生人数 k_i 与该教室的容量 r_i 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

$$\max(f_4) = \sum \frac{k_i}{r_i}$$

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

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

$\theta_i (i = 1, 2, 3, 4)$ 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。

3.3 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^[12]。但是随机搜索方式生成的初始种群一般适应度都非常低,为了提高初始种群的适应度,本文应用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 为算法流程图。

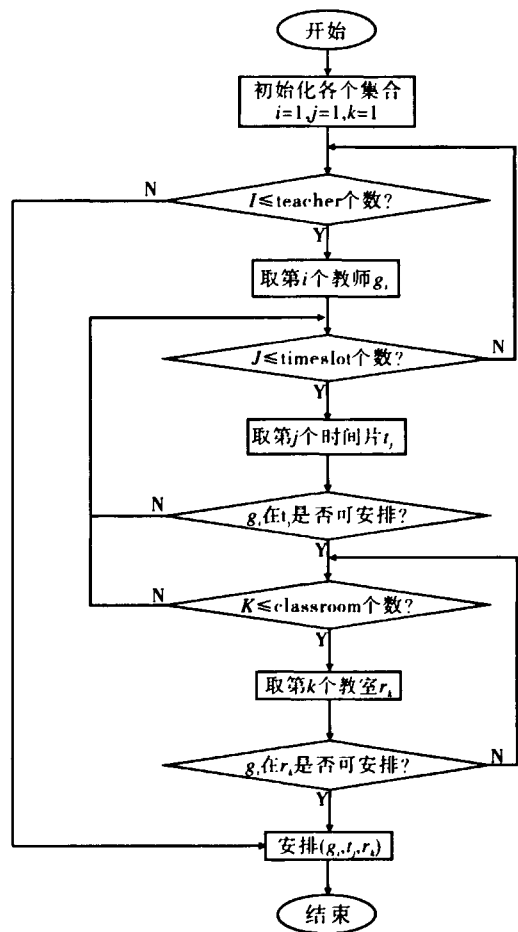


图 1 初始种群生成算法

3.4 小生境策略

简单遗传算法通常采用比例选择算子,该选择策略有明显的缺陷:当种群中有个别个体的适应度值远远大于种群的平均适应度值时,这些个体将在比例选择算子作用下迅速扩张,从而充满整个种群,这使得整个种群中个体差异度急剧减小,以至于种群多样性被严重破坏。种群多样性的不足是遗传算法全局搜索能力不强的主要原因,小生境技术

的引入正是为了解决这一问题. 具体方法如下: 如果 2 个染色体的海明距离

$$\|x_i - x_j\| = \sqrt{\sum_{k=1}^M (x_{ik} - x_{jk})^2}$$

小于给定值, 则比较其适应度大小, 取其中较小的一个染色体, 强行令其适应度为 10^{-3} , 这样在下一代中该染色体会以极大的概率被淘汰, 以保持种群的多样性, 防止出现早熟收敛.

3.5 自适应交叉概率和自适应变异概率设计方案

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, 以便提高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:

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

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

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.

4 仿真实验

4.1 实验数据

实验所用数据各要素如下: 学生为 6 200 人; 教师为 387 人; 班级为 125 个; 课程为 669 门; 教室为 168 间; 开课任务书为 669 份.

4.2 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 2 个参数无须设置, 其他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. 实验中有 125 个班级, 本文取种群规模为 150. 2) MAX-Gen 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; MAXGen 过大时, 虽收敛, 但不是全局最优, 本文取 1 000.

4.3 实验结果

实验中, 使用文献[13]的方法(简称 GA)与本文方法(简称 DNGA)做对比.

实验做了 10 次, 每进化 100 代就记录 1 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.

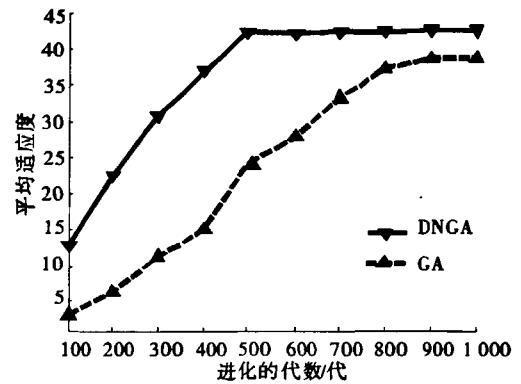


图 2 适应度对比结果

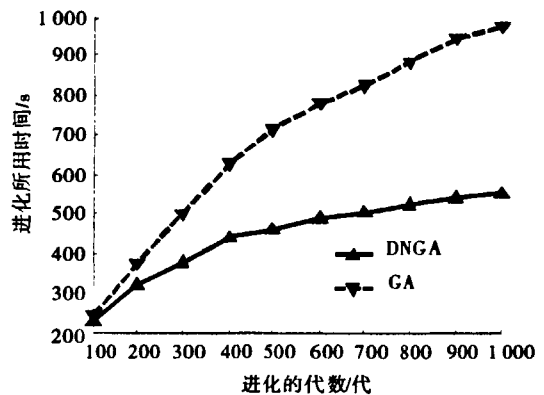


图 3 消耗时间对比结果

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[13]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.

排课效果比较如表 2, 表 3 所示.

表 2 排课效果比较(1)

方法	学生主干课程 每周上课天数/d	同一课程 上课间隔期/d	学生每天平均 上课数/节
GA	2~2.5	1~1.4	4~8
DNGA	2.5~2.7	1.4~1.6	4~5

注: 课程安排得越集中, 那么在有课的天数里, 学生每天平均的上课节数越多, 这会导致学生极其疲惫从而使得接收知识的效果较差, 故而把学生每天平均上课节数也作为一个评价指标.

从表 2 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。

表3 排课效果比较(2)

方法	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	86	17	84	15
DNGA	96	2	97	1

5 结语

本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。

参考文献:

- [1] Adewumi A O, Sawyerr B A, Montaz A M. A heuristic solution to the university timetabling problem [J]. Eng Comp, 2009, 26(8): 972.
- [2] Gotlieb. The construction of class-teacher timetables [C]// Proc IFIP Congress, Heidelberg: Springer, 1963: 73 - 74.
- [3] Even S, Itai A, Shamir A. On the complexity of timetable and multicommodity flow problems [J]. SIAM J on Comp, 1976(5): 691.
- [4] Cooper Tim B, Jeffrey H Kingston. The complexity of timetable construction problems [C]// Proc ICPTAT'95, Heidelberg: Springer, 1995: 183 - 295.
- [5] Shi Juan. Research on application of IGA (immune genetic algorithm) to the solution of course-timetabling problem [C]// Proc of 2009 4th Int Conf on Comp Sci and Edu, New York: IEEE, 2009: 1109 - 1105.
- [6] Aladağ C H, Hocaoglu G, Basaran M A. The effect of neighborhood structures on tabu search algorithm in solving course timetabling problem [J]. Expert Syst with Appli, 2009, 36(10): 12349.
- [7] Detienne B, Péridy L, Pinson É. Cut generation for an employee timetabling problem [J]. European J of Operational Research, 2009, 197(3): 1178.
- [8] Pillay N, Banzhaf W. A study of heuristic combinations for hyper-heuristic systems for the uncapacitated examination timetabling problem [J]. European J of Operational Research, 2009, 197(2): 482.
- [9] Lee Yusin, Chen Chuen-yih. A heuristic for the train pathing and timetabling problem [J]. Trans Research Part B: Methodological, 2009, 43(9): 837.
- [10] Zhang Defu, Liu Yongkai, Hallah R M. A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems [J]. European J of Operational Research, 2010, 203(3): 550.
- [11] Anmar A, Masr A. Multi-neighbourhood particle collision algorithm for solving course timetabling problems [C]// Proc of 2009 2nd Conf on Data Mining and Optimi, Selangor: Malaysia, 2009: 21 - 27.
- [12] Guyon O, Lemaire P, Pinson É. Cut generation for an integrated employee timetabling and production scheduling problem [J]. European J of Operational Research, 2010, 201(2): 557.
- [13] Pillay N, Banzhaf W. An informed genetic algorithm for the examination timetabling problem [J]. Applied Soft Comp, 2010, 10(2): 457.