

基于遗传-蚁群混合算法的排课系统^①



孙 弋, 胡钜璋

(西安科技大学 通信与信息工程学院, 西安 710054)

通讯作者: 胡钜璋, E-mail: reushhh@163.com

摘 要: 在高校的教务管理中, 排课问题是复杂又关键的环节, 科目数量众多, 教学资源有限等等因素都制约着排课的复杂程度和结果. 排课本质就是将课程、班级在合适的时间段安排到合适的教学位置, 是一个 NP 问题的求解. 随着规模的不断扩大, 问题求解难度呈指数形式增加, 当规模达到一定程度时就很难在短的时间内求出最优解. 鉴于此, 本文提出了遗传-蚁群混合算法, 将两种算法混合使用, 依靠遗传算法生成信息素分布, 利用蚁群算法求最优解. 实验结果表明, 混合算法提高了排课的效率 and 课表的合理度.

关键词: 排课; NP 问题; 遗传算法; 蚁群算法; 混合算法

引用格式: 孙弋, 胡钜璋. 基于遗传-蚁群混合算法的排课系统. 计算机系统应用, 2019, 28(2): 81-86. <http://www.c-s-a.org.cn/1003-3254/6762.html>

Course Schedule System Based on Genetic-Ant Colony Hybrid Algorithm

SUN Yi, HU Ju-Hui

(College of Communication and Information Engineering, Xi'an University of Science and Technology, Xi'an 710054, China)

Abstract: In the administrative management of colleges and universities, the scheduling is a complex and critical task. The number of subjects and the limited teaching resources all restrict the complexity and results of class scheduling. The essence of class scheduling is to arrange the course and class to the appropriate teaching location at the appropriate time. It is a solution to the NP problem. As the scale continues expanding, the difficulty of solving problems increases exponentially. When the scale reaches a certain level, it is difficult to find the optimal solution in a short time. In view of this, this study proposes a genetic-ant colony hybrid algorithm, which uses a mixture of two algorithms, relies on genetic algorithm to generate pheromone distribution, and uses ant colony algorithm to find the optimal solution. The experimental results show that the hybrid algorithm improves the efficiency of class scheduling and the rationality of the class schedule.

Key words: course arranging; NP problem; genetic algorithm; colony algorithm; hybrid algorithm

引言

教务管理是教育活动的重要支撑, 随着高校教育改革发展的发展, 高校人数的增加, 但教学资源却相对有限的情况下, 排课工作的难度和复杂程度都有所提高. 因此, 如何设计一个能满足多约束条件并且高效的排课系统成为了目前待解决的问题^[1].

排课系统的核心是排课, 排课算法一个 NP 完全问

题, 即算法的计算时间呈指数增长^[2]. 常用的排课算法有: 蚁群算法、遗传算法、贪心算法、图论算法等. 遗传算法模拟生物进化过程, 具有自适应全局寻优和智能搜索等优点, 但是容易收敛得到局部近似最优解, 进行大量的无用迭代, 难以收敛于最优解^[3,4]. 蚁群算法启发于蚂蚁寻找最优路径的行为, 具有较强的鲁棒性, 是一种正反馈算法. 但当群体规模较大时, 搜索初期信息

① 收稿时间: 2018-08-05; 修改时间: 2018-08-30, 2018-09-07; 采用时间: 2018-09-11; csa 在线出版时间: 2019-01-28

素匮乏,容易产生停滞现象.单算法在解决问题时都会有局限性,花费的时间成本更多,求解的结果不理想^[5,6].

为了提高排课的效率和成功率,本文提出一种将遗传算法与蚁群算法结合使用的方法,首先使用遗传算法对初始种群进行优化选择,生成蚁群算法所需的信息素,再通过蚁群算法求得全局精确解.优劣互补,充分发挥两者的优势^[7].

1 排课问题的分析

排课问题主要涉及教师、教室、课程、班级、上课时间等因素,排课的实质就是要求课程表在安排的时候不能在上述五个因素中发生冲突.可使用五个集合来表示这些因素.

班级集合: $C = \{C_1, C_2, C_3, \dots, C_n\}$

课程集合: $S = \{S_1, S_2, S_3, \dots, S_m\}$

教师集合: $T = \{T_1, T_2, T_3, \dots, T_x\}$

教室集合: $R = \{R_1, R_2, R_3, \dots, R_l\}$,教室有多种类型,例如实验室、多媒体教室、机房等,每个教室容纳的人数也不相同.

时间段集合: $P = \{P_1, P_2, P_3, \dots, P_i\}$, P_i 表示时间段,例如 P_1 为周日的1-2节课, P_2 为3-4节课.时间和教室的笛卡儿积为: $N = R * P = \{(r_1, p_1), (r_2, p_2), \dots, (r_n, p_n)\}$, N 中的元素是教室-时间对.排课问题由此转化为一门课寻找一个合适的教室时间对.

1.1 硬约束条件

硬约束是在排课过程中必须要遵守的约束条件,如果排课过程中无法遵守硬约束条件,则会导致日常的教学计划无法顺利的进行.例如:

(1) 同一个教师在同一时间段内只能上一门课,并且教室的性质要和课程要求相吻合;

(2) 同一个班级在同一时间段内只能接受一门课;

(3) 一门课程安排的教室座位数量要大于或等于本门课程上课的人数.

1.2 软约束条件

软约束条件需要将人体生物规律与排课进行结合.满足软约束条件越多,课表越能让教师与学生满意.例如:

(1) 尽量将专业课或者难度较大的课程安排在学生思维活跃的时间段;

(2) 体育课应安排在下午或者上午的第二大节,因

为体育课后人体疲惫不适宜安排专业性过强的课程.

2 算法概述

2.1 遗传算法

遗传算法是通过模拟生物界遗传选择和自然淘汰的生物进化过程的计算模型,借鉴了生物界适者生存、优胜劣汰的进化规律,从而演化的随机化搜索方法.遗传算法从一个种群开始,在这个种群中可能存在问题的潜在解,每个种群是由经过编码的 N 个个体组成,每个个体就是带有一定特征的染色体实体.在每一代中,选择适应度较大的个体进行培育,然后借助遗传学中的遗传算子进行多次组合交叉、变异等操作,产生出代表新的解集的种群.这样经过了多次演化之后得到的种群就可以看作问题的近似最优解.

遗传算法的流程如下:

(1) 随机产生一定数量的初始种群,数量由业务需求定义.

(2) 对全部个体进行适应度的评估,如果某个个体的适应度已经满足最优化的要求,则输出此个体,并结束计算.否则转向第(3)步.

(3) 从所有个体中取出适应度较强的个体成为再生个体.

(4) 依据交叉概率,变异概率生成新的个体.

(5) 由上述两步交叉和变异产生新一代种群,然后返回第(2)步.

2.2 蚁群算法

蚁群算法是一个寻找最优路径的方法.蚂蚁在寻找食物的过程中,通过释放信息素留下信息,后面的蚂蚁就会根据路上的信息素来选择路径,信息素的浓度与行走的路程成反比,走的路途越长,信息素浓度越低,则越不容易找到,路途越短,信息素浓度含量越高.随着时间的推移最优路上的信息素浓度越大,最终蚁群会找到一个最优路径.

2.3 遗传-蚁群混合算法

遗传,蚁群算法是两大仿生算法,都有各自的优势和不足.遗传算法的优势在于在大范围内具有快速全局搜索的能力,但在处理反馈信息时利用率过低,容易过早的收敛,或求解到一定范围时会做大量的无用迭代,从而会降低求解过程中的效率.蚁群算法采用正反馈机制,具有较强的鲁棒性,使搜索过程不断收敛,有更好的全局优化能力.但是在搜索初期信息素分布较

少时, 求解速率较慢.

而排课问题的实质就是一个 NP 完全问题, 影响排课因素的细微变动, 尤其是信息量一点点的增加, 都会给排课带来“组合爆炸”灾难, 当排课规模较大时, 单算法在排课问题中就表现出局限性, 会出现排课时间较长、排出的课表无法令教师学生满意等现象. 因此, 本文将两算法结合应用于排课问题中, 首先利用遗传算法在大范围内快速的生成信息素分布, 再利用蚁群算法求出全局最优解, 优势互补, 可以减少求解的时间, 提高精确率.

3 遗传-蚁群混合算法解决排课问题

3.1 遗传算法在排课问题中的应用

3.1.1 构造基因编码和染色体

实施遗传算法的第一步就是对需要解决问题的参数进行基因编码, 从而进行相关的遗传操作. 本文采用二进制对编码数据进行设计. 如表 1 所示.

表 1 编码方式

教师 ID	班级 ID	课程 ID	教室	上课时间
-------	-------	-------	----	------

3.1.2 适应度函数

在遗传算法的进化过程中, 其根据适应度函数来确定进化方向. 适应度是对课表优劣的一种体现. 适应度越大则证明课表的效果越优秀. 因此适应度函数直接决定着排课方案的寻优速率和能否找到最优解.

所以这就需要制定一个合适的适应度函数(期望). 在本文中, 课表的期望值可以分为以下三种: 课程离散程度期望, 节次优化度期望和特殊课程期望.

(1) 课程离散程度期望

同一门课程安排的时间应该分布均匀, 每周安排的时间不能分散也不能太集中. 本文将每天的教学时间段分为 4 个, 每周一共 20 个时间段. 用编号相同课程的时差来表示离散程度的大小如表 2 所示.

表 2 课程离散程度期望

时差	1, 2, 3, 18, 19	4, 5, 15, 16, 17	6, 7, 12, 13, 14	8, 9, 10, 11
期望	0.2	0.4	0.6	0.8

每一个课程时差对应一个期望, 期望函数公式为:

$$F_{cs} = \sum F_{cs}(i) \quad (1)$$

(2) 节次优化度期望值

人体生物钟规律应与教学内容相匹配, 逻辑思维活跃的时间段一般在早晨, 而下午更容易疲倦所以学习效率较低. 根据人体生物钟规律节次的不同的课程期望值也不同, 如表 3 所示.

表 3 节次优化度期望值

节次	1, 5, 9, 13, 17	2, 6, 10, 14, 18	3, 7, 11, 15, 19	4, 8, 12, 16, 20
期望	0.8	0.7	0.5	0.1

对于难度比较大的课程就可以安排在期望值高的节次上, 对于节次期望值函数的公式为:

$$F_{so} = \sum F_{so}(i) \quad (2)$$

(3) 特殊课程的期望值

特殊课程包含体育课与自习课, 此类课程最好安排在下午的第二节. 当体育课结束后, 人体处于疲倦状态, 不适于将专业性过强的课程安排在体育课后. 如表 4 对特殊课程给出了不同的期望.

表 4 特殊课程的期望值

节次	1, 5, 9, 13, 17	2, 6, 10, 14, 18	3, 7, 11, 15, 19	4, 8, 12, 16, 20
期望	0	0	2	4

对体育课排课就依据上表的期望值为:

$$F_t(i) = \sum F_t(i) \quad (3)$$

应将课表中没有课程的时间段安排给自习课. 则计算特殊课程期望值的公式为:

$$F_{ts} = \sum (F_t(i) + F_z(i)) \quad (4)$$

根据以上分析可得出适应度函数 F , 如下公式为:

$$F = \frac{K_1 \times F_{cs} + K_2 \times F_{so} + K_3 \times F_{ts}}{3} \quad (5)$$

式中, K_1, K_2, K_3 为调控期望值对总期望影响程度的参数.

3.1.3 种群操作

当形成初始种群, 并设计好适应度函数后, 下一步就是要进行遗传操作. 遗传算法包括三个基本操作: 选择、交叉和变异.

(1) 选择操作

选择操作就是依照适应度的值保留优良的个体, 适应度越高证明越近似于最优解. 本文中保留优良个体的依据是期望值方法, 选择期望值高的个体进行交叉变异操作.

(2) 交叉操作

交叉操作就是把两个个体的部分结构进行替换重组从而产生新的个体. 根据选择操作的结果, 选择两条染色体作为父代, 进行交叉操作. 为了防止染色体的结构发生更改, 本文的交叉操作只针对教室和上课时间, 这样教师和班级都没有发生改变, 维持了染色体原本的结构. 同时设定交叉概率为 P_J , 若 P_J 大于随机值 r ($0 < r < 1$), 就进行交叉操作.

(3) 变异操作

变异操作能够随机的对个体的局部进行搜索, 提高种群的多样化. 本文中的变异操作

就是对染色体编码中教室和时间段的局部编码进行修改, 可以提高染色体的适应度. 同时设定变异概率为 P_G , 若 P_G 大于随机值 r ($0 < r < 1$), 就进行变异操作.

3.2 蚁群算法在排课问题中的应用

蚁群算法最初用于解决 TSP (Traveling Salesman Problem), TSP 具有广泛的代表意义, 许多问题都可类似的抽象为 TSP 问题的求解.

因此以 TSP 来描述蚁群算法的模型: 一只蚂蚁要去 n 个有食物的地区, 它先随机选择一个地方作为起点, 然后在逐个到达所有地区, 留下信息素, 最后再返回起点, 每个地区只能到达一次, 蚂蚁要怎么样选择顺序才能使行程最短. 下面建立蚁群算法的数学模型:

m 为蚁群中的蚂蚁数量; n 为问题的规模大小 (有食物地方的个数); $b_i(t)$ 为 t 时刻位于 i 地区的蚂蚁数量, 故蚂蚁数量为:

$$m = \sum_{i=1}^n b_i(t) \tag{6}$$

其中, $\tau_{ij}(t)$ 为 t 时刻路径 (i, j) 上的信息素; $\mu_{ij}(t)$ 为蚂蚁从 i 到 j 地区的期望程度; $p_{ij}^k(t)$ 为 t 时刻蚂蚁 k 从 i 转移到 j 的概率.

蚂蚁 k 是根据 $\tau_{ij}(t)$ 的结果来决定该往哪个方向前进, 一般会选择 $\tau_{ij}(t)$ 值高的路径, 则蚂蚁 k 在 t 时刻由 i 地转移到 j 的概率为:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\mu_{ij}(t)]^\beta}{\sum_{s \in allowed_k} [\tau_{is}(t)]^\alpha [\mu_{is}(t)]^\beta}, & \text{若 } j \in allowed_k \\ 0, & \text{否则} \end{cases} \tag{7}$$

其中, α 为信息启发因子, 表示在蚂蚁选择行走路径时信息量的影响; μ 为期望启发式因子, 表示启发信息对

蚂蚁选择路径时产生的影响程度.

在每只蚂蚁访问完所有的地区以后, 每条路上的信息素浓度会发生变化, 因此需要对信息素进行更新.

$$\tau_{ij}(t+n) = (1-\rho) \times \tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{8}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \tag{9}$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L_k \\ 0 \end{cases} \tag{10}$$

其中, ρ 是信息素挥发系数 ($0 < \rho < 1$), $1-\rho$ 表示路径残留的信息素量; $\Delta\tau_{ij}(t)$ 为从 i 到 j 地区路上增加的信息素量; $\Delta\tau_{ij}^k(t)$ 为蚂蚁 k 在 i 到 j 地区路上留下的信息素量; Q 为表示信息素浓度; L_k 为蚂蚁 k 需要行走的路径长度.

3.2.1 蚁群算法的二分图模型

当遇到需要用图结构解决的问题时可以使用蚁群算法. 在本文中, 用蚁群算法解决排课问题, 首先需将排课问题用图结构 G 来描述:

$$G = (N, S, C)$$

上式中, N 是图的顶点; S 是边的集合; C 是边集有关的权值集合.

排课问题涉及五个要素, 在蚁群算法中这五个要素的关系可转化为“课程, 教师, 班级”关系与“时间, 教师”关系, 排课问题就是解决这两个关系所构成的二分图的最大匹配问题.

(1) 二分图的顶点集定义

在本文将“课程, 教师, 班级”的关系定义为 R_{LSC} , $R_{LSC} = L \times S \times C$, R_{LSC} 的元素组成的集合为 G_{LSC} , 称其为二分图左侧的顶点集. “时间, 教室”的关系为 R_{TR} , $R_{TR} = T \times R$, R_{TR} 的元素组成的集合为 G_{TR} , 为二分图右侧的顶点集.

(2) 二分图的边集定义

在排课中对教室的安排要满足两个要求: 一是安排人数不能大于教室座位数量, 二是教室类型满足课程要求. 因此, G_{LSC} 与 G_{TR} 之间只有满足上述两个基本要求的节点才能进行连接. 二分图的边集就是 G_{LSC} 中的节点与 G_{TR} 中相匹配节点的连线.

(3) 二分图中边的权值

每条边上的权值根据具体课程时间的期望度来构造. 例如, 较难的专业课嵌入式教师对课程的期望度是上午的第二节课, 那么 G_{LSC} 中所有嵌入式课程与 G_{TR} 中时间段为满足上午第二节课的所有节点连接的边

权值要尽可能的小. 通过设置权值, 才能更加合理的排出质量高的课表.

3.2.2 二分图模型

根据上节构造好的二分图模型, 蚂蚁都是从左侧 G_{LSC} 的顶点集出发, 根据转移概率的值和边集在右侧 G_{TR} 寻找匹配的节点. 然后再返回至 G_{LSC} 进行下一次的搜索工作, 如此反复直到蚂蚁为 G_{LSC} 的顶点集全部安排好上课地点和时间后, 一次循环结束.

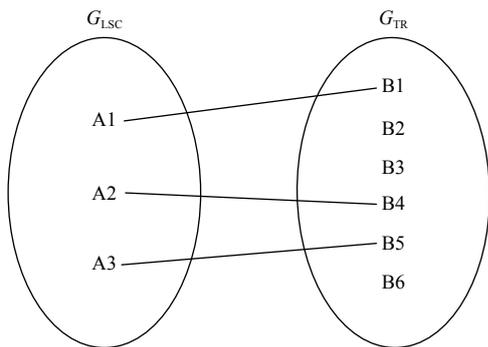


图1 二分图模型

图1展示了蚂蚁一次循环的过程, 蚂蚁从 A1 出发, 则蚂蚁此次行走的路径为:

$$A1 \rightarrow (B1 \rightarrow A2) \rightarrow (B4 \rightarrow A3) \rightarrow B5$$

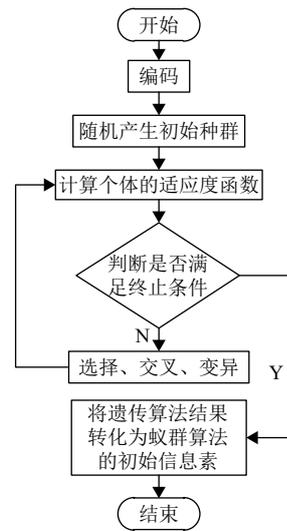
其中, B1 是第一次匹配的终点, A2 是第二次匹配的起点, B1 和 A2 可以看成是一个逻辑节点. 同理, B4 和 A3 在此次周游中也可以看成是相同的逻辑节点.

3.3 遗传-蚁群混合算法在排课问题中的应用

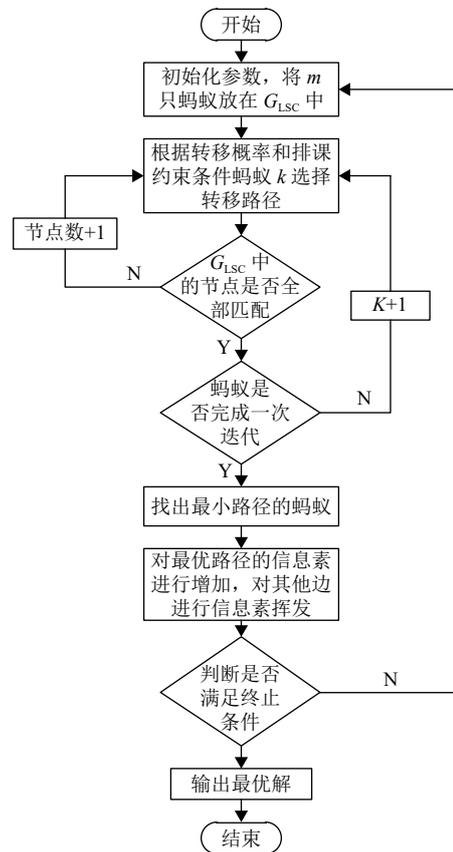
本文将遗传与蚁群算法进行混合, 首先利用遗传算法生成信息素分布, 在利用蚁群算法求出问题的精确解. 在遗传算法生成信息素时, 为了防止其在收敛后进行无用的迭代, 要设置相应的终止条件. 在运算工程中, 当连续两次运算结果出现的差值小于 0.01, 或迭代 100 次, 满足其中一个条件时可停止运算并将取得的最优解作为蚁群算法的初始解. 蚁群算法在进行 100 次迭代之后, 选择路径最短的解即可作为最优解. 遗传-蚁群混合算法对排课问题解决步骤如图2所示.

3.4 实验结果及分析

为了验证遗传-蚁群混合算法在排课系统中的应用, 选取一个学院对排课系统进行测试, 该学院的专业数量为 6, 班级数量为 21, 约有 800 名学生, 55 名教师, 22 间教室, 其中有 16 名教师对排课有特殊要求, 排课单元为 202.



(a) 遗传算法流程图



(b) 蚁群算法流程图

图2 遗传-蚁群混合算法流程图

为了验证遗传-排课算法在排课系如中的实际使用效果, 在排课单元为 100, 400, 800 的情况下, 对遗传-蚁群混合算法、遗传算法、蚁群算法三种算法的排课效果、运算时间及适应度进行了比较. 三种算法

各测试 10 组数据后取得平均值。

(1) 实验参数设置

遗传算法的参数设置如下: $M=40$, $P_J=0.4$, $P_G=0.02$; 蚁群算法设置的参数如下: $m=6$, $\alpha=1$, $\beta=4$, $\rho=0.25$, $Q=8$, $\tau_{\max}=800$, $\tau_{\min}=0.01$, $\mu_{ij}=1/c_{ij}$, $gen=100$ 。

(2) 排课效果的比较

通过以下四个指标来衡量排课课表的质量,如表 5 所示。1 为违反硬约束的次数; 2 为一周有两次课程且间隔少于一天的次数; 3 为难度较大的课程安排在时间段不好的次数; 4 为教师的特殊要求未满足的次数。

表 5 排课效果比较

排课方案	遗传算法	蚁群算法	遗传-蚁群
1	0	0	0
2	27.6	16.8	10.4
3	24.0	11.3	7.2
4	7.2	4.3	2.1

如表 5 可以看出, 遗传算法在软约束的冲突方面表现的最不理想, 基于遗传-蚁群混合算法排出的课程表, 在软约束方面的冲突次数最令人满意, 课程表的可行性及质量都要优于单算法排出的课程表。

(3) 实验结果分析

三种算法的平均运算时间如表 5 所示, 平均适应度结果如表 6 所示。

表 6 运算时间比较

排课单元	遗传算法	蚁群算法	遗传-蚁群
100	37	51	44
400	133	185	146
800	529	562	538

表 7 适应度比较

排课单元	遗传算法	蚁群算法	遗传-蚁群
100	182	204	238
400	276	290	320
800	301	365	413

由表 6 可得, 每种算法随着排课单元的增加, 所需要的排课时间随之增加。在排课单元相同的情况下, 遗

传算法需要的时间最短, 而蚁群算法需要的时间最长。

由表 7 可得, 三种算法的适应度与排课的单元成正比。当排课单元相同时, 遗传-蚁群混合算法的适应度最高。也就证明混合算法排出的课表最优。

由此可见, 在相同的条件下, 遗传-蚁群混合算法排课消耗的时间比遗传算法高一些, 但适应度却远远高于另外两个算法。说明混合算法解决排课问题具有良好的时间性能和优化性能。

在选择最优课表时, 可以将适应度最大的三个课表取出, 根据定义排课质量的四个指标, 满足软约束条件越多的课表则可作为最终使用的课表

4 结论

目前解决排课问题可使用的单算法很多, 但混合算法却相对较少。本文提出了一种遗传-蚁群混合的排课算法, 将两种算法的优势进行了结合。在经过测试后混合算法在运算时间上介于遗传与蚁群算法中间, 但适应度高于单算法。混合算法不仅解决了单算法的局限性问题, 还在排课效率上有所提高。

参考文献

- 薛冬梅. 充分利用资源 科学合理排课. 中原工学院学报, 2002, 13(S1): 97-98.
- 杜立智, 陈和平, 符海东. NP 完全问题研究及前景剖析. 武汉工程大学学报, 2015, 37(10): 73-78.
- 朱剑冰, 李战怀, 赵娜. 基于混合遗传算法的自动组卷问题的研究. 计算机仿真, 2009, 26(5): 328-331, 352. [doi: 10.3969/j.issn.1006-9348.2009.05.084]
- 许秀林, 胡克瑾. 基于约束满足和遗传算法的排课算法. 计算机工程, 2010, 36(14): 281-284. [doi: 10.3969/j.issn.1000-3428.2010.14.102]
- 李俊, 周虎, 李波. 基于虚拟蚂蚁的局部优化蚁群算法. 控制与决策, 1-10. [doi: 10.13195/j.kzyjc.2018.0298]
- 史文. 基于遗传算法的自动排课系统设计与实现[硕士学位论文]. 成都: 电子科技大学, 2013.
- 宗薇. 高校智能排课系统算法的研究与实现. 计算机仿真, 2011, 28(12): 389-392. [doi: 10.3969/j.issn.1006-9348.2011.12.095]