

基于改进粒子群优化算法的排课问题

王念桥*, 姚四改

(武汉职业技术学院 电子信息工程学院, 武汉 430074)

(*通信作者电子邮箱 red_lotus@139.com)

摘要:深入分析了排课问题,提出一种基于离散粒子群的排课算法,构建了相应的解题框架。针对粒子群算法有后期收敛速度慢、易收敛于局部最优的缺点,结合排课问题的特点,对粒子群算法作了改进。在三维空间中建立模型,采用避免冲突的种群初始化加快收敛,并且引入变异操作避免陷入局部最优等。实践表明改进后的粒子群算法能有效地解决排课问题。

关键词:排课问题;粒子群优化算法;组合优化;变异

中图分类号: TP301.6 **文献标志码:** A

Curriculum scheduling based on improved particle swarm optimization algorithm

WANG Nianqiao*, YAO Sigai

(Department of Electronic and Information, Wuhan Institute of Technology, Wuhan Hubei 430074, China)

Abstract: After analyzing the problems of curriculum scheduling, an algorithm based on discrete particle swarm algorithm for the curriculum scheduling was proposed, with a framework for solving the problem. Because the particle swarm algorithm has a slow convergence speed during late period of its iterations and can easily be trapped in local optimal solution, an improved algorithm was applied with fully considering the features of curriculum scheduling. The algorithm was modeled in three-dimensional space, its particles were initialized with avoiding conflicts, and mutation was introduced to avoid being trapped in optimal solution, etc. The application makes it clear that the proposed algorithm can solve the curriculum scheduling problem effectively.

Key words: curriculum scheduling problem; Particle Swarm Optimization (PSO) algorithm; combinatorial optimization; mutation

0 引言

排课是教务工作中一项基本工作,其内容是根据授课计划制定出一个合理的、可行的课表。排课涉及到教师、课程、教室、学生、时间等诸多因素,因而处理起来相当复杂^[1]。排课问题已被证明是一个 NP 完全问题^[2],解决此问题的重要思路就是寻求近似最优解,以降低计算复杂度。自 20 世纪 70 年代以来,人们对于这个问题做了大量的工作,将许多算法应用于排课问题,例如动态规划^[3]、回溯法^[4]、图形着色^[5]、贪心法^[6]等,并在实践中取得了一些成绩。近年来智能优化算法如神经网络、遗传算法、蚁群算法、模拟退火算法等^[7]也逐步用于解决排课问题。

粒子群优化(Particle Swarm Optimization, PSO)算法^[8]是一种基于群智能的随机优化算法,由 Kennedy 等于 1995 年共同提出,适用于求解大量非线性、多目标的复杂优化问题。其原理和机制简单,调整参数少,算法容易实现。传统的 PSO 算法只能应用于连续问题的优化,针对离散问题, Kennedy 等提出了一种离散二进制 PSO(Binary Particle Swarm Optimization, BPSO)算法^[9]。

本文以 BPSO 为基础,结合排课系统的特点,提出一种新型的解决排课问题的离散粒子群算法,在三维空间中建立模型,并引入了冲突检测及变异等操作。

1 排课问题分析

1.1 问题描述

排课涉及多方面的因素,由教师、教室、课程、班级和时间五元素组成,要解决的目标是上述五元素的最优化配置,也即教学资源的合理安排及使用。这是一个非线性的多目标组合优化问题,此类问题一般不存在唯一的最优解,而是寻求一种合理的近似最优解。

一个合理的可行的排课表必须满足硬约束条件,并且尽量满足软约束条件。硬约束条件是排课表必须满足的条件,如果不满足,则上课会出现冲突;而软约束条件是应当尽量满足的条件,满足软约束条件可使课表更合理。通常,硬约束条件包含以下方面:

- 1) 一位教师在同一基本教学时间段只能上一个班级的一门课程;
- 2) 一个班级在同一基本教学时间段只能上一位教师的一门课程;
- 3) 一个教室在同一基本教学时间段只能有一位教师上课;

软约束条件根据各院系情况不同而有所差别,但一般包含以下方面。

- 1) 重要的课程尽量安排在较好的时间段(上午 4 课时,

收稿日期:2012-07-04;修回日期:2012-08-12。

作者简介:王念桥(1971-),男,湖北武汉人,讲师,硕士,主要研究方向:软件工程、智能控制机器人;姚四改(1965-),女,湖北襄樊人,副教授,主要研究方向:计算机辅助设计。

下午前2课时),课程可分为专业核心课、专业基础课、公共课、选修课等,重要程度依次降低。每天4个教学时间段,第一个时间段教学效果最好,其后教学效果逐渐下降。

2)一位教师一天授课尽量不超过4课时,且尽量满足教师对授课时间的要求。

3)课程对教室的要求,比如有些课程是理论课,全部在普通教室上课;有些课是实践课,全部在实验室上课;有些是理论加实践课,需要交错使用普通教室和实验室;对不同类型实验室(多媒体教室可作为实验室来处理)的要求等。

4)学生的上课时间应均匀分布在一周,比如不应出现某天8节课,而另外的某天无课上的情况。

5)同一班级的同一课程上课时间应均匀分布在一周,比如一周4课时的课程每周上2次课,则隔2天上课是均匀分布。

1.2 模型设计

排课问题中的5个基本元素可用集合描述如下:

教师集合 $P = \{p_i\}, i = 1, 2, \dots, n_p$

课程集合 $L = \{l_i\}, i = 1, 2, \dots, n_l$

班级集合 $C = \{c_i\}, i = 1, 2, \dots, n_c$

教室集合 $R = \{r_i\}, i = 1, 2, \dots, n_r$

时间集合 $T = \{t_i\}, i = 1, 2, \dots, n_t$

其中 n_p, n_l, n_c, n_r, n_t 分别表示教师总数、课程总数、班级总数、教室总数以及时间总数。以上集合皆为有序集合,可按编码对集合中的元素排序。教室包含普通教室、多媒体教室、机房以及实验室等,按不同类别依次排序,形成一维线性结构。教学时间为每周一至周五,每次授课为一基本教学时间段,通常为2课时,每天授课最多5个基本教学时间段,即10课时。时间集合中包含每天5次课(10课时),从周一至周五,每次课一个元素,共计25个元素,即 $n_t = 25$ 。

排课问题的输入可用集合 $I = \{i_k\}, k = 1, 2, \dots, n$ 来描述,集合中的每一个元素 $i_k = (p_k, l_k, c_k, w_k, h_k, r_k, o_k)$ 。其中: $p_k \in P; l_k \in L; c_k \in C; w_k$ 是课程 c_k 的周学时; h_k 是课程 c_k 的学期总学时; R_k 是课程 c_k 对教室要求,即软约束3)中提到的要求; o_k 是其他要求,包括教师对上课时间的要求,课程的起始和结束教学周等。每一名教师所带的不同课程、不同班级在集合 I 中有且仅有一项,即对于集合 I 中的每一个元素,教师、课程及班级三者不能全部相同。

通常 PSO 算法采用数字串形式的编码。针对排课问题的特点,本文采用三维形式的编码方案,可采用逻辑上的三维形式数据结构实现此编码方案。以班级 C 、教室 R 、时间 T 为轴,建立三维坐标系 CRT ,如图1所示。此三维空间的点取值为(教师,课程)。即:

$$f_{cr}(c, r, t) \in \{(p, l) | p \in P, l \in L\} \quad (1)$$

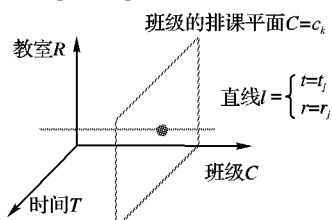


图1 排课的三维空间坐标系

依据排课问题的输入,填充此三维空间,得到一个可能的解,即一个粒子。

2 算法设计

2.1 种群初始化

初始化生成初始的种群,其中每一个粒子为一个可能的解。初始种群的大小 M 定义为问题规模 N 的1.5倍:

$$M = 1.5 \times N \quad (2)$$

$$N = HS/2 \quad (3)$$

其中 HS 为所有课程一周的总课时量,因为一个基本教学时间段通常为2课时,所以此处将 $HS/2$ 作为问题规模。粒子群用集合 $X = \{X_i\}, i = 1, 2, \dots, M$ 表示,第 i 个粒子表示为 $X_i = (X_{i1}, X_{i2}, \dots, X_{in})$, $n = N$ 。逻辑上,每个粒子 X_i 都有自己的三维空间,记为 CRT_i 。 X_i 经历过的最好位置记为 P_i ,即局部最优粒子。群体中所有粒子经历过的最好位置记为 P_g ,即全局最优粒子。

初始化通常采用随机搜索的方式产生。好的初始群体能提高算法的收敛速度,但是简单的随机搜索方式生成的粒子适应度一般都比较低。为此,本文在随机搜索方式的基础上增加了约束条件,可极大地提高初始粒子的适应度。生成一个粒子 X_i 的具体过程如下。

初始时,三维坐标系 CRT_i 中的点均为未初始化,即对于 CRT_i 中所有的点 (c, r, t) , 有 $f_{cr}(c, r, t) = \text{null}$ 。对于集合 I 中的每一个元素 i_k , 依据其班级 c_k 在 CRT_i 中取平面 $c = c_k$, 称为班级 c_k 的排课平面,在此平面上随机搜索可用点 (c_k, r_j, t_j) , 即 $f_{cr}(c_k, r_j, t_j) = \text{null}$ 的点。为了避免冲突,定义冲突检查函数:

$$\text{conflict}(t_j, r_j, p_k) = \begin{cases} \text{false}, & p_k \notin \{f_{cr}(c, r, t) | \forall (c, r, t) \in l\} \\ \text{true}, & \text{其他} \end{cases} \quad (4)$$

其中直线 l 的方程如下:

$$\begin{cases} t = t_j \\ r = r_j \end{cases} \quad (5)$$

若 $\text{conflict}(t_j, r_j, p_k) = \text{false}$, 表明不存在冲突,则令 $f_{cr}(c_k, r_j, t_j) = (p_k, l_k)$, 即安排了一节课 $(c_k, r_j, t_j, p_k, l_k)$; 否则,在平面 $c = c_k$ 上随机搜索下一个可用点。位于式(5)中的直线 l 上的点说明在时间 t_j 上课或使用了教室 r_j , 如果某位教师在直线 l 上出现了1次以上,即出现了冲突。安排 i_k 需要 $w_k/2$ 个这样的点。依次处理完集合 I 中的每一个元素,即可产生一个粒子,这将在 CRT_i 中安排 N 个点。

重复 M 次以产生初始粒子群。依据此方式初始化,即可满足硬约束条件。

2.2 适应度函数

排课问题中,除了要满足硬约束条件外,还应满足软约束条件,才能生成合理的、切实可行的课表。粒子群算法中,适应度函数就是用来评价粒子与最优解之间的符合程度,其群体的更新是以个体适应度为依据的,因此适应度函数直接影响算法的收敛速度以及能否找到最优解。排课问题的约束条件有多个,即有多个优化目标,因此本文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)同一班级的同一门课程其上课时间应均匀分布在一周,评价函数定义为:

$$f_1 = \sum_{i=1}^{n_l} \sum_{j=1}^{n_c} f_{lc}(l_i, c_j) \quad (6)$$

函数 f_{lc} 评价某一个班级的某一个课程的一周分布情况,其参数 $l_i \in L, c_j \in C$ 。 f_{lc} 的结果值分别为0,1,2,3,分别表示

班级 c_j 所上课程 l_i 的一周分布好坏,值越小,则分布越好。通常,课程的周学时为 0,2,4,6,8。函数 f_{lc} 首先查询班级 c_j 所上课程 l_i 的周学时 w ;然后在班级 c_j 的排课平面上查询课程 l_i 两次上课的最小间隔天数 d ;最后根据表 1 计算函数返回值 r 。同一课程同一天两次上课则间隔天数为 0,相邻两天上课则间隔天数为 1,依此类推。周 8 学时的课程一般要求 4 课时连上,因此其处理与周 4 学时的课程相同。

表 1 课程周分布评价对应关系

课程周学时 w	最小间隔天数 d	函数值 r
0,2		0
4,8	3	0
4,8	2,4	1
4,8	1	2
4,8	0	3
6	1	1
6	2	0
6	0,3,4	3

2) 尽量满足课程对教室的使用需求,评价函数定义为:

$$f_2 = \sum_{i=1}^{n_l} f_{lr}(l_i) \quad (7)$$

对于开设了课程 $l_i (l_i \in L)$ 的所有班级,函数 f_{lr} 查看其排课平面,检查课程 l_i 的教室使用情况。根据教室是否全部满足要求,部分满足要求或是全部未满足要求,函数 f_{lr} 结果值分别为 0,1,2,值越小,则教室对课程的满足程度越好。

3) 尽量满足教师对上课时间的要求,评价函数定义为:

$$f_3 = \sum_{i=1}^{n_p} f_{pt}(p_i) \quad (8)$$

函数 f_{pt} 首先查询到教师 $p_i (p_i \in P)$ 代课的所有班级,然后根据它们的排课平面合并教师 p_i 的所有教学时间,将结果与教师 p_i 提出的时间要求做比较,根据比较结果是否全部满足要求,部分满足要求或是全部未满足要求,分别返回 0,1,2,值越小,表示教师对教学时间安排越满意。

4) 学生的上课时间应该均匀分布在一周,评价函数定义为:

$$f_4 = \sum_{i=1}^{n_p} f_{ct}(c_i) \quad (9)$$

函数 f_{ct} 首先根据班级 $c_i (c_i \in C)$ 的周课时计算出一周最佳分布,比如假设周课时为 20,则每天 4 课时是最佳分布;若周课时为 16,则一周中 4 天课时均为 4,而另外一天为 0 或者一周中 3 天课时均为 4,另外 2 天均为 2 是最佳分布(最佳分布可能不唯一)。然后根据 c_i 的排课平面统计其周教学时间安排,将统计结果与最佳分布作比较,取不符合最佳分布的天数为函数结果值(最大值取 3),分别为 0,1,2,3,值越小,表示学生的教学时间安排越均匀。

5) 按照课程的重要程度,尽量将较好的教学时间段安排给较重要的课程,评价函数定义为:

$$f_5 = \sum_{i=1}^{n_l} f_{li}(l_i) \quad (10)$$

对于同一个班级的同一天课程,如果重要程度较高的课程的教学时间排在重要程度较低的之后,称为逆序。函数 f_{li} 查看课程 $l_i (l_i \in L)$ 的所有教学时间安排,统计其出现逆序的次数 c ,计算 c 占该课程总教学次数的百分比,其结果体现了该课程教学时间安排的合理程度,根据此结果,函数返回值 0,1,2,3,值越小,表示该课程的教学时间安排越合适。

综上所述,排课问题的适应度函数根据各评价函数加权所得:

$$f = \sum_{i=1}^5 f_i \times \varphi_i \quad (11)$$

其中 φ_i 是各目标的加权值,反映了目标的重要程度,本文依次取值 0.32, 0.26, 0.2, 0.14, 0.08。适应度函数 f 取值越小,则适应度越高。当 f 取值为 0 时,即达到了排课问题的理论最优解。

2.3 粒子更新

本文通过“交换子”的概念来解决粒子更新的问题^[10]。在传统的 PSO 算法中,更新粒子的速度时需要计算不同粒子之间或是同一粒子不同时刻之间的距离。这里通过适应度值来度量粒子间的距离。定义距离函数^[11]为:

$$dis(X_i, X_j) = |f(X_i) - f(X_j)| \quad (12)$$

其中 X_i 和 X_j 是 2 个不同的粒子; $f(X_i)$ 和 $f(X_j)$ 是它们的适应度值。另外定义一个“交换子”,并记为 $swap(X_i, X_j, c_k, u)$,其中 $X_i, X_j \in X$,表示将要执行交换操作的 2 个粒子; $c_k \in C$,其含义如下。

由以上初始化过程可知,班级 c_k 的所有排课都位于平面 $c = c_k$,平面包含的点的个数为 $n_r \times n_t$,其中包含的已设定值的个数等于班级 c_k 的周课时量 $w_k/2$,记为 c_{kwl} ,将此平面上的点映射到一维线性结构上。

1) 记 X_i 中关于班级 c_k 的排课平面 $c = c_k$ 映射到一维线性结构 S_{ik} 。首先在 S_{ik} 值不为 null 的点中寻找第 u 个点,记其值为 S_{iku} ,然后在 S_{jk} 中寻找值等于 S_{iku} 的点,比如,假定 S_{jk} 值不为 null 的点中第 v 个点的值等于 S_{iku} ,记此点为 S_{jkv} 。

2) 在 S_{ik} 和 S_{jk} 值不为 null 的点中,分别寻找 S_{iku} 和 S_{jkv} 后面出现的第一对值不相等的值,分别记为 S_{ikp} 和 S_{jkq} 。

3) 在 S_{ik} 中寻找值等于 S_{jkq} 的点,记为 S_{ikr} 。将 S_{ikr} 映射回 X_i 中关于班级 c_k 的排课平面 $c = c_k$,其在三维坐标系 CRT_i 中的坐标为 (c_k, r_j, t_i) ,执行冲突检测函数 $conflict(t_j, r_j, p_k)$,若返回 false,则交换 S_{iku} 和 S_{ikr} ;否则,转 2),继续寻找下一对值不相等的值。

4) 将改变的位置信息映射回 X_i 中关于班级 c_k 的排课平面 $c = c_k$ 。

在 2) 中,如果搜索达到一维线性结构的结尾,则定义下一个位置为一维线性结构的起始位置。为了加快收敛,交换子中也引入了冲突检测过程。

定义算法中的交换操作^[7]如下:当更新一个粒子时,每次随机选择一个班级 $c_k (c_k \in C)$,一个随机整数 $u (0 < u \leq c_{kwl})$,然后执行交换子 $swap(X_i, X_j, c_k, u)$,其中 X_i 是要更新的粒子, X_j 是全局最优粒子 P_g 或者局部最优粒子 P_i 。执行交换的次数取决于 $dis(X_i, X_j)$,取交换的次数为:

$$num_swap = \begin{cases} n_dis, & n_dis \leq max_swap \\ max_swap, & n_dis > max_swap \end{cases} \quad (13)$$

$$n_dis = \text{int}\left(\frac{k \cdot dis(X_i, X_j)}{f(X_j)} + 1.5\right) \quad (14)$$

其中: $\text{int}(\cdot)$ 是取整函数; k 是范围参数,取值为 (1,5) 内的整数; max_swap 是给定的最大交换次数,取值为 $n_c/3$, n_c 为班级个数。两个参数可根据经验调整。更新一个粒子时,依次执行交换子 $swap(X_i, P_i, c_k, u)$ 及 $swap(X_i, P_g, c_k, u)$ 。

2.4 变异

随着迭代的进行,越来越多的粒子将接近群体中的最优

粒子,从而失去活力,算法容易陷入局部最优。本文在算法中引入变异操作^[12]。定义:

$$\begin{cases} \bar{f} = \frac{1}{M} \sum_{i=1}^M f_i \\ \sigma_f^2 = \frac{1}{M} \sum_{i=1}^M (f_i - \bar{f})^2 \end{cases} \quad (15)$$

其中: f_i 是第 i 个粒子的适应度; M 是粒子群规模; \bar{f} 是所有粒子的平均适应度; σ_f^2 是适应度的方差,反映了群体的收敛程度,其值越小,群体越趋于收敛。定义:

$$\gamma^2 = \frac{\sigma_f^2}{\max\{(f_i - \bar{f})^2, i = 1, 2, \dots, M\}} \quad (16)$$

若 γ^2 小于一个给定的阈值,且排课问题的理论最优解或期望最优解尚未达到时,即粒子的适应度值偏离 0 较远,则认为系统趋于早熟,阈值为取值于 (0.001, 0.1) 区间的任意实数。此时对全局极值 P_g 执行变异操作,则可改变粒子前进的方向,以期跳出局部最优,进入搜索空间中的其他区域搜索。只对全局极值 P_g 执行变异操作可节省大量的计算时间。过程如下:

- 1) 随机选择 $c_k (c_k \in C)$;
- 2) 在班级 c_k 的排课平面 $c = c_k$ 随机选择点 (t_{i1}, c_k, r_{i1}) , 对点 (t_{i1}, c_k, r_{i1}) 执行冲突检测函数 conflict, 如果存在冲突则重复 2);
- 3) 在班级 c_k 的排课平面 $c = c_k$ 随机选择点 (t_{i2}, c_k, r_{i2}) , 对点 (t_{i2}, c_k, r_{i2}) 执行冲突检测函数 conflict, 如果存在冲突则重复 3);
- 4) 交换 (t_{i1}, c_k, r_{i1}) 和 (t_{i2}, c_k, r_{i2}) 的值。

表3 排课结果比较

方法	班级每天课时	班级同一课程间隔天数	主干课程每周上课天数	教学时段利用率/%	教室利用率/%	教师满意程度/%
自动	4~6	1~2	2~4	90	92	86
手工	4~8	1~3	2~4	85	87	80

从表3可以看出,采用本文算法生成的课表其实际效果非常接近于手工排课的结果。学生每天课时、同一课程间隔天数等相比手工课表有一定程度的改善。另外,教学时段利用率(较好教学时段的使用)、教室(包括实验室和多媒体教室等)利用率及教师满意程度都有不同程度的提高。

由于实际课表中的可变因素较多,比如合班上课、毕业班上课问题、多个教师合带一门课程等,自动排课表有时还需要细微调整。

4 结语

排课问题是一个非线性多目标优化问题,本文在深入分析排课问题的基础上,建立了数学模型,并应用离散粒子群算法给出了解题的算法步骤。该算法在实际应用中取得了较满意的效果。

参考文献:

- [1] ADEWUMI A O, SAWYERR B A, MONTAZ A M. A heuristic solution to the university timetabling problem [J]. Engineering Computations, 2009, 26(8): 972-984.
- [2] EVEN S, ITAI A, SHAMIR A. On the complexity of time table and multi-commodity flow problems [J]. SIAM Journal on Computing, 1976, 5(4): 691-703.
- [3] 程学先, 祝苏薇. 一种基于动态规划的课表调度算法的研究与实现 [J]. 武汉理工大学学报: 交通科学与工程版, 2006, 30(3): 485-488.
- [4] 车明, 秦存秀, 刘凯. 基于改进回溯算法的计算机排课系统 [J].

5) 重复 1) ~ 4) max_swap 次。

2.5 算法步骤

总结起来,算法的基本流程如下。

- 1) 初始化粒子群。
- 2) 设置粒子的局部最优解为粒子的当前位置,全局最优为初始种群的最优粒子。
- 3) 判断算法是否满足终止条件: 如果满足则转 7); 否则执行 4)。
- 4) 对每个粒子执行更新操作。
- 5) 计算每个粒子的局部最优以及种群的全局最优。
- 6) 计算适应度的方差 σ_f^2 及 γ^2 , 判断算法是否需要执行变异操作: 如果不需要变异,则直接转 3); 否则对 P_g 执行变异操作,然后转 3);
- 7) 输出全局最优解 P_g 。

其中的终止条件定义为迭代次数是否达到预设定的最大次数或者全局极值 P_g 的适应度值接近 0。

3 实验数据及结果分析

将上述算法应用于实际的排课系统中,取武汉职业技术学院电信学院 2011—2012 学年第二学期的数据为样本数据来测试算法的有效性。数据如表 2 所示。

表2 样本数据

教师数	课程数	班级数	教室数	班级周学时
106	112	56	61	20~24

结果数据如表 3 所示。

沈阳工业大学学报, 2006, 28(6): 667-670.

- [5] SELIM S M. Split vertices in vertex colouring and their application in developing a solution to the faculty timetable problem [J]. The Computer Journal, 1988, 31(1): 76-82.
- [6] 邓曦辉. 浅谈贪心算法在排课系统中的应用 [J]. 电脑与电信, 2011(7): 29-30.
- [7] 梁艳春, 吴春国, 时小虎, 等. 群智能优化算法理论与应用 [M]. 北京: 科学出版社, 2009.
- [8] KENNEDY J, EBERHART R C. Particle swarm optimization [C]// Proceedings of IEEE International Conference on Neural Networks. Piscataway: IEEE Press, 1995: 1942-1948.
- [9] KENNEDY J, EBERHART R C. A discrete binary version of the particle swarm algorithm [C]// Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. Piscataway: IEEE Press, 1997: 4104-4108.
- [10] WANG K P, HUANG L, ZHOU C G, et al. Particle swarm optimization for traveling salesman problem [C]// Proceedings of the IEEE International Conference on Machine Learning and Cybernetics. Piscataway: IEEE Press, 2003: 1583-1585.
- [11] SHI X H, XING X L, WANG Q X, et al. A discrete PSO method for generalized TSP problem [C]// Proceedings of the Third International Conference on Machine Learning and Cybernetics. Piscataway: IEEE Press, 2004: 2378-2383.
- [12] 吕振肃, 侯志荣. 自适应变异的粒子群优化算法 [J]. 电子学报, 2004, 32(3): 416-420.