

文章编号:1000-6788(2000)05-0104-05

# 进化算法 PBIL 在时间表问题中的应用

金炳尧<sup>1</sup>, 蔚承建<sup>2</sup>, 何振亚<sup>2</sup>

(1. 浙江师范大学计算机系, 浙江 金华 321004; 2. 东南大学无线电系, 江苏 南京 210096)

**摘要:** 用任意整数编码的 PBIL(基于人口的增量学习)算法求解两类时间表问题, 在计算中采用了系统信息熵来估计进化计算的进行程度并作为计算的结束条件。计算结果表明, 用该方法解决时间表安排问题适应性强、结果稳定, 是一个非常理想的算法。

**关键词:** 进化计算; 时间表问题; 熵

**中图分类号:** TP301.6

## Application of PBIL Algorithm on Timetable Problems

JIN Bing-yao<sup>1</sup>, WEI Cheng-jian<sup>2</sup>, HE Zheng-ya<sup>2</sup>

(1. Zhejiang Normal University, Jinhua 321004; 2. Southeast University, Nanjing 210096)

**Abstract:** In this paper, two applications of PBIL (Population-Based Incremental Learning) algorithm on timetable problems are researched. Here PBIL algorithm is modified. The values of the alleles can take an integer number not only 0 or 1. Also, entropy is used here as the termination condition and to estimate the degree of evolution process in PBIL algorithm. The results are satisfactory. It has proved that PBIL algorithm is a good method in solving timetable problems because of its good-adaptability and stability.

**Keywords:** evolutionary algorithms; timetable problems(TTP); entropy

### 1 引言

进化计算是借鉴生物遗传与生物进化的思想而发展起来的一类搜索方法, 被广泛应用于各类优化问题。进化计算模型的主要代表有遗传算法(Genetic Algorithms)、进化规划(Evolutionary Programming)、进化策略(Evolution Strategies)<sup>[1]</sup>。进化计算的特点是使用了一种群体搜索机制取代传统的个体局部搜索, 使算法不易陷入局部极值点。进化计算的算法简单、适应性广, 是一种非常鲁棒(Robust)的算法, 尤其适用于传统方法难以处理的各种组合优化问题(其中多数为 NP 完全问题或 NP 难题)。一般地, 进化算法由三部分组成: 解的表示方法; 进化选择机制; 后代的产生方法。遗传算法用二进制串表示解, 而进化规划与进化策略都直接用十进制数表示。选择一般都采用优胜劣汰的竞争机制; 选择目标函数值(在进化计算中又称为适应值 Fitness)较优的解作为下一代的父代用以产生后代。后代的产生方法是进化计算的关键。GA 采用父代双亲的基因交叉重组, 而 EP 与 ES 采用单个父代的高斯变异。美国卡内基梅隆大学的 Baluja, S. 提出了 PBIL(Population-Based Incremental Learning)的进化算法<sup>[2]</sup>, 将进化过程视为学习过程。用学习所获取的知识——学习概率来指导产生后代。这种概率是整个进化过程的信息积累, 用它指导产生的后代将会更优生(比起 GA 的双亲基因重组及 EP、ES 的单个父代 Gaussian 变异), 因而能在许多应用问题中获得更快的收敛速度及更优的结果。

时间表问题(Time Table Problem—TTP)是一个应用广泛的组合优化问题。如列车时刻表、体育竞赛日程表、工程计划表、学校课程表等都是这类时间表问题的实例。时间表问题已经证明是 NP 难题<sup>[3]</sup>。因此,

时间表问题的求解只能借助于各种搜索算法,使用较多的是启发式算法<sup>[4,5]</sup>. A. Colomi 等人首次用遗传算法解决时间表问题,开创了进化计算在时间表问题中的应用,并取得令人满意的结果<sup>[6]</sup>.

本文将基因位是二进制编码的 PBIL 算法作适当的改进,使每个基因位能直接使用任意整数编码,解决两个时间表问题的实际应用.并且,提出了用系统信息熵值(Entropy)作为进化进程的度量与进化结束的判据.计算结果表明:该算法计算速度快、稳定性好,是解决时间表问题的较为理想的算法.

## 2 PBIL 算法描述

### 2.1 PBIL 算法原理

设  $S$  为表示解的长度为  $n$  的二进制位串,  $S_i (1 \leq i \leq n)$  是  $S$  的第  $i$  位,在进化计算中称为第  $i$  个基因位.其取值为 0 或 1. 优化问题表示为:

$$\min f(S) \quad S \in \text{可行的解空间}, f \text{ 为优化目标函数}$$

$P$  为 PBIL 的学习概率,  $P_i (1 \leq i \leq n)$  是与  $S_i$  对应位置的学习概率.  $P_i$  表示  $S_i$  位取 1 的概率 ( $1 - P_i$  表示  $S_i$  位取 0 的概率). PBIL 进化过程如下:

- 1) 初始化学习概率  $P; P_i = 0.5 (1 \leq i \leq n)$
- 2) 由学习概率  $P$  指导产生  $k$  个个体(解)
- 3) 计算由 2) 产生的  $k$  个个体的目标函数值,并根据目标函数值找出一个最优解:  $B$
- 4) 用此 3) 找到的最优解  $B$  修正学习概率  $P$

$$P_i = P_i + (B_i - P_i) \times \epsilon \quad \epsilon \text{ 为修正常数}, 1 \leq i \leq n$$

- 5) 返回 2) 循环,直至满足一定的结束条件为止

PBIL 算法与 GA 算法相似,采用二进制编码.但实际应用中每个基因位  $S_i$  的取值范围可能有多值,这时要用多位二进制数据来表示.这可能会给一些应用带来不便.本文提出基因位是任意整数编码的 PBIL 算法:

设  $S$  为表示  $n$  维空间的解,  $S_i (1 \leq i \leq n)$  是  $S$  的第  $i$  维或第  $i$  个基因位,  $S_i$  有  $m$  个取值 ( $m$  为正整数).  $P$  为 PBIL 的学习概率,  $P_{ij} (1 \leq i \leq n, 1 \leq j \leq m)$  是  $S_i$  位的第  $j$  个取值的概率,对于某一代的  $k$  个样本中的最佳解  $B$ ,相应的修正学习概率  $P$  及概率归一化处理的公式分别为:

$$P_{ij} = P_{ij} + \epsilon \quad \epsilon \text{ 为修正常数}, 1 \leq i \leq n \text{ 且 } j = B_i$$

$$P_{ij} = \frac{P_{ij}}{\epsilon + 1} \quad 1 \leq i \leq n, 1 \leq j \leq m$$

在进化开始时,  $P_{ij} = 1/m$ ,这是保证开始时每个基因位的  $m$  个取值具有相同的概率.随着进化进程不断地进行,各个位置的概率开始偏离(但其总和永远为 1)初始的平均概率.到收敛时各个位置的概率一般都集中在到某一个取值.下面为任意整数编码的 PBIL 算法描述:

- 1) 初始化学习概率  $P; P_{ij} = 1/m (1 \leq i \leq n, 1 \leq j \leq m)$
- 2) 由学习概率  $P$  指导随机产生  $k$  个个体  
方法如下(某一样本在某一位置的采样):
  - a. 将同一基因位置的  $m$  个不同取值的概率依次累加,得到  $m$  个累加概率
  - b. 产生一随机数  $r$ ,累加概率中大于  $r$  的最小的累加概率所对应的取值即为采样值
- 3) 计算  $k$  个个体的目标函数值,根据目标函数值找出最优解:  $B$
- 4) 用此最优解  $B$  修正学习概率:

$$P_{ij} = P_{ij} + \epsilon \quad \epsilon \text{ 为修正常数}, 1 \leq i \leq n \text{ 且 } j = B_i$$

- 5) 概率归一化处理:  $P_{ij} = \frac{P_{ij}}{\epsilon + 1} (1 \leq i \leq n, 1 \leq j \leq m)$
- 6) 返回 2) 循环,直至满足规定的结束条件.

### 万方数据

### 2.2 PBIL 进化计算的熵估计

在进化计算中,结束条件的确定一直是比较困难的.一般是直接用最大进化代数作为结束进化计算的条件.但是,不同的问题及不同的规模其收敛速度的差异是很大的.对某一问题来说是进化很充分的代数对另一问题可能是进化还没有结束.尤其是对进化难度大的问题,如果根据目标函数值的变化来判断往往会出现偏差.在这里,我们提出一个用熵来判断 PBIL 进化计算的结束判据.熵是用来表示系统信息量的.显然,整个系统的熵值随着进化的不断进行而不断减小.到最后收敛时熵等于零(此时的概率非 0 即 1).因此在这里熵作为进化结束的判据是最方便,也是最准确的.系统信息熵的计算公式如下:

$$S_{\text{entropy}} = - \sum_{i=1}^n \left( \sum_{j=1}^m P_{ij} \ln P_{ij} \right)$$

利用系统信息熵的计算公式可以非常容易地估计进化计算的进行程度.设某问题有  $N$  个基因位置,每个基因位有  $M$  个取值.其初始熵(初始概率  $P_{ij}$  为  $1/m$ )的计算公式如下:

$$S_{\text{initial}} = n \ln m$$

PBIL 算法的进程是一个系统熵值从  $S_{\text{initial}}$  下降到 0 的过程.为了估计进化到一定程度时系统的熵值,特作如下的假设:1)每个基因位置的进化程度是均等的,即概率的分布是一致的.2)同一个基因位置中除了某一个主要取值的概率外,其它取值的概率值相等.依据上述的假设我们可以得到主要取值的概率进化到一定百分比  $C\%$  时的系统熵值的计算公式(主要取值概率为  $C\%$ ,其它非主要取值概率为  $(1-C\%)/(m-1)$ ):

$$S_{C\%} = n((1-C\%) \ln(m-1) - (1-C\%) \ln(1-C\%) - C\% \ln C\%)$$

由上述熵值计算公式,很容易估计出进化到某一程度的熵值.表 1 是  $N=20,30,50,100(M=N)$  时在进化到不同程度的系统熵值.

表 1 PBIL 算法的系统熵值与进化进行程度的关系

$N$	初始熵值	50%熵值	80%熵值	90%熵值	95%熵值	99%熵值	100%熵值
20	59.914	43.307	21.785	12.390	6.914	1.709	0.000
30	102.035	71.303	35.215	19.854	11.006	2.690	0.000
50	195.601	131.952	63.938	35.713	19.655	4.746	0.000
100	460.517	299.070	141.942	78.459	42.827	10.195	0.000

3 时间表问题计算实例

3.1 时间表模型一

模型描述 有  $N$  个工作安排在  $N$  个时间片区,并满足一定的约束条件.

学校的课表安排是这类典型的时间表安排问题.传统的计算机课程表安排方法大都采用启发式的搜索算法.在这里我们构造一个用 PBIL 算法安排课程表的方法: $N$  个时间片区作为解的  $N$  个基因位,每个基因位的取值为  $N$  个课程.这里的每一个课程只能安排在一个时间片区.时间表安排问题的解的优化目标函数值一般是根据课程表安排的约束条件而制订.一般把约束条件分成几类:必须满足的条件为硬约束,而另一些条件则为软约束也即在可能的情况下能满足最好.给这些约束条件赋以不同的罚值.解的目标函数值就是该解违反约束条件所得的罚值之和.在这样目标函数指导下进化,就能得到目标函数值越来越小的解.并且用多次重复计算可以得到多个不同的解,便于使用者从中挑选最适合的方案.

算例 1 表 2 是 10 个班级的周课程、任课教师信息(每个班级各有 15 个时间片,共有 8 个教室可供使用).其中硬约束为:1)同一教师在同一个时间片内只能安排一个班级上课(不同班级相同编号的课程为同一任课教师);2)10 个班级中最多能有 8 个班级同时在教室上课;3)同一个班级的相同的课程(编号相同)

表 2 班级课程及教师安排信息表

班级	课 程 信 息														
1	0	0	0	0	1	5	6	6	7	7	8	8	9	9	9
2	0	0	0	0	1	17	17	18	13	13	9	16	16	19	19
3	0	0	0	0	3	18	18	20	20	20	21	21	19	22	22
4	0	0	0	0	2	6	6	7	7	10	11	11	11	12	12
5	0	0	0	0	2	22	22	23	24	24	24	25	25	26	26
6	0	0	0	0	3	25	25	27	27	28	28	26	29	29	29
7	0	0	0	0	0	11	11	12	12	14	14	15	15	16	16
8	0	0	0	0	0	30	30	31	31	31	32	32	28	28	27
9	0	0	0	0	0	0	0	5	5	8	8	14	14	16	16
10	0	0	0	0	0	0	0	33	33	34	34	32	32	30	30

表 3 进化计算得到的各班级课程表之一

班级	星期一			星期二			星期三			星期四			星期五		
1	7	9	8	5	0	0	9	6	7	8	0	0	9	6	1
2	0	13	0	17	16	18	13	19	1	17	9	16	0	19	0
3	20	19	0	21	18	22	0	20	3	21	0	0	18	20	22
4	11	10	0	12	7	6	11	0	0	0	6	2	11	12	7
5	23	26	24	25	22	0	0	24	0	26	25	0	24	22	2
6	29	27	0	0	26	3	28	25	29	0	27	0	29	25	28
7	12	0	0	11	14	0	16	15	0	11	14	0	12	15	16
8	28	0	31	32	0	0	30	31	0	28	32	27	30	0	31
9	0	8	0	0	5	16	14	8	0	16	5	0	14	0	0
10	33	34	0	30	32	0	34	0	0	30	33	0	0	32	0

不能安排在同一天。4)同一班级相同编号的课程不能安排在连续的两天的时间片中。5)体育课必须安排在下午。而这里的软约束为:1)自习课最好安排在下午。给硬约束的罚值为 100;而软约束的罚值为-1(这些数据的定义是人为的)。在没有任何概率预置的条件下用 PBIL 算法进行计算(概率修正常数为  $\epsilon=0.05, N=15, 10$  个班级分别采样,每代采样个数为 50),在 10 次重复计算中 100%地都取得极为满意的结果(9 次为-28,1 次为-25),而进化到最佳解所须的平均评价人口数仅为 75000 个。表 3 是上述结果之一。用该算法对我校多个院系的课程表进行计算全部获得成功。表 2 中课程信息表的含义:

0 表示自习课,不占用教室;1-4 表示体育课,不占用教室;5 以上为普通课程;

课程编号是任课教师信息的反映,每个课程号由一个任课教师担任;同一班级的相同课程号是该课程周学时的信息;不同班级课程编号相同是表示同一教师任课,但课程可以是不同的。

3.2 时间表模型二

模型描述 在一个已安排好的时间表中安排一项新的内容(利用原时间表中的自由时间),并满足一定的约束条件。

在实际应用中,经常会碰到这类问题。如列车时刻表中添加新的车次使原来时刻表的改动最小;对全校学生在一周时间安排一次活动使安排的时间表对教学影响最小等。这种时间表安排问题与传统的时间表问题不同。用 PBIL 算法可以比较容易地实现。设有  $N$  个班级,  $M$  个时间片,要求每个班级各安排一项新课程,并且满足一定的约束条件(如每个时间片所安排的新课程不能超过  $K$  个。我校由于历史原因,英语听力课与计算机上机操作课都是在各系课程安排好以后再根据各系所安排的课程表来安排这两门课程。每个学期都花费大量的时间来排课,结果还不理想。)

这里  $N$  个班级的进化计算的基因位置编号,  $M$  个时间片为每个位置的取值。由于原来已经安排了时间表,这时每个基因位置的取值只能在原来时间表的自由时间中选取。利用概率初始化的方法可以非常容

易地做到这一点:把原来已安排课程的时间片设为 0,未安排课程的时间片设为 1,然后把每个基因位置上的概率归一化。目标函数的计算也采用罚值法:将所得到的解中违反约束条件的次数累加所得到的值即可作为解的目标函数值。

算例 2 表 4 为 40 个班级在 20 个时间片的算例(表中,为已安排的位置,\*为自由时间,行为时间片序号;列为班级号序号)。其中的自由时间数据用随机方法产生。用 PBIL 算法平均花费 4000 个人口即可得到非常满意的结果(在 486 计算机上,Pascal 语言每次用时 20 秒,20 次重复 95%获得最佳结果 0,另一次结果为 1)。表 5 为其中的结果之一,而用随机搜索方法花费 50 万个评价人口还未搜索到最佳安排(其中最佳结果为 3)。

	1234567890123456789012345678901234567890
01	.....*...*.*.*.*.*.....**.....
02	.*.....*.....*.*.....*.....
03	*.....*.....*.....*.....
04	.....*.....*.....*.....*.....
05	.....*.....*.....*.....*.....
06	.....*.....*.....*.....*.....
07	.....*.....*.....*.....*.....
08	.....*.....*.....*.....*.....
09	.....*.....*.....*.....*.....
10	.....*.....*.....*.....*.....
11	.....*.....*.....*.....*.....
12	.....*.....*.....*.....*.....
13	.....*.....*.....*.....*.....
14	.....*.....*.....*.....*.....
15	.....*.....*.....*.....*.....
16	.....*.....*.....*.....*.....
17	.....*.....*.....*.....*.....
18	.....*.....*.....*.....*.....
19	.....*.....*.....*.....*.....
20	.....*.....*.....*.....*.....

表 4 40 个班级在 20 个时间片的自由时间表

表 5 40 个班级的时间安排表的 PBIL 的计算结果之一

班级	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
时间片	3	19	15	4	6	7	11	13	10	19	14	20	10	5	12	5	18	8	9	2
班级	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
时间片	1	13	15	11	17	16	8	3	12	4	17	14	1	7	18	2	16	9	6	20

用此方法来安排计算机公共课上机时间表及外语听力课时间安排表既快速又方便,而且安排的时间表质量好。

4 结论

PBIL 作为一种概率指导的进化计算方法,用学习概率代替进化的人口(积累了一定知识的人口)在两个方面显示出其优越性。第一,它的后代产生更具代表性,因而往往能获得更快收敛收速度。第二,它产生后代的方法简单,免去了普通进化计算为选择好的生成后代的方法而进行的探索(不同的重组与变异方式产生的结果有时差异极大)。在时间表问题中的应用经验表明该算法快速、稳定。因此,该进化方法在时间表问题中的应用是非常合适的。

参考文献

1 姚新,陈国良,徐惠敏等. 进化计算研究进展. 计算机学报,1995,18(9):694~705.

2 Baluja S. Genetic Algorithms and Explicit Search Statistics. Advances in Neural Information Processing System. MA:MIT press, 1996.

3 Even S. Itai A, Shamir A. On the Complexity of Timetable and Multicommodity Flow Problems. SIAM Journal on Computing, 1976,5(4):691~703.

4 李盘林,李立健,刘晓红等. 基于启发性知识研究生院课表编排系统,计算机学报,1992,15(11):876~880.

5 黄干平,陈洛资. 解“时间表”(TTP)的启发式算法. 计算机应用与软件,1997,(2):60~64.

6 Colorni A, Dorigo M, Maniezzo V. Genetic Algorithms and Highly Constrained Problems; The Time-Table Case. Proceedings of the First International Conference on Parallel Problem Solving from Nature (PPSN). Springer-Verlag, Lecture Notes in Computer Science,1991, 496:55~59