

基于拟人策略的高校排课算法^{*}

陈卫东 李吉桂

(华南师范大学计算机系 广州510631)

Personification Algorithms for the University Timetabling Problem

CHEN Wei-Dong LI Ji-Gui

(Dept. of Computer Science, South China Normal University, Guangzhou 510631)

Abstract For the university timetabling problem that is NP-hard, some new strategies of tackling it are proposed, and two heuristic algorithms based on personification strategies are presented, which outperform the known straightforward heuristic algorithms in the quality of solution. The experimental results show that our algorithms are practical.

Keywords Timetabling, NP-hardness, Personification, Heuristic algorithms

1 引言

高校排课问题是典型的 NP-难问题^[2],即在 $P \neq NP$ 的假设下,找不到一个算法能保证在多项式时间内得到最优解。因此,为了实际应用的需要,对于这类问题,往往利用问题的一些启发式知识来探求能快速求其近似最优解的算法,即启发式算法。目前求解排课问题的启发式算法大致可分为两大类:一类是构造型算法,另一类是改进型算法。构造型算法一般是按照某种次序逐步安排每门课程进而得到一张合法课表,常使用的是一种称之为直接启发式算法^[1]。它们一般是模拟人工排课的经验,基本策略都是“优先安排受限制最多的课程”,而差别仅在于赋予“受限制最多”的含义不同。改进型算法则一般是在(合法)课表上进行改进使得课表更加合理,如禁忌搜索、模拟退火算法、遗传算法^[1,6,7]等现代优化算法,但这些算法有时需要在预先产生的若干个初始合法课表上进行工作。相比之下,直接启发式算法,简单、直接、快速,其启发式知识往往是针对具体问题的特点提出的,虽然通用性较差,但是如果启发式知识得当,则往往能快速得到较好的解。而现代优化算法尽管具有通用性和理论上的全局最优性,但是往往收敛慢。在实用的系统中大都是采用直接启发式算法。

我们给出的启发式算法吸收了构造型和改进型这两类算法的特点。包含有两个算法:Exploring 算法从效果上相当于是一个局部搜索算法,它在构造课表的过程中同时也对课表进行了局部优化,所得课表的质量明显高于直接启发式算法所得的课表;Learning 算法是在前者的基础上引入了拟人策略来跳出“局部陷阱”的搜索算法,进一步提高了课表的质量。

鉴于 UTTP 的约束多样性和目标模糊性,本文首先给出了 UTTP 的一个数学描述,将约束条件进行量化,确定了优化目标。接着描述了传统的直接启发式算法。然后制定了几个新的求解策略,由此得出两个基于拟人策略的排课算法,并将它们与直接启发式算法进行了比较。实验结果表明我们的算法求解速度快、效果好。

2 问题的数学描述

根据我国高校的典型情况,约定高校排课问题具有如下

特点:一门课程的教学可包含多个自然班级;每个班级上课的课室不固定;一门课程可根据周学时多次授课,每次授课的时间连堂1、2或3节;同一教师可以讲授多门不同的课程等等。

为了方便,将每次授课所包含的几个要素组成的4元组(课程,班级列表,教师,周学时)称作课元,将一周内可供安排的每节上课时间称作时间片,由时间片和课室构成的序偶(时间片,课室)称作时空片。将课室相同,时间在同一个上午、下午或晚上且连续的若干个时空片的集合称作时空片簇。因此排课的任务即是为给定的每个课元合理地分配时空片簇。“合理地”体现为排课方案满足一定的约束条件。一般包括以下三个层次:

1)硬约束条件,主要包括:(i)按课元周学时分配大小相当的时空片簇,且课室类型(如体育馆)和容量满足课元的要求;(ii)每个时空片最多只能分配给一个课元;(iii)每个自然班在同一个时间片内最多只能有一门必修课;(iv)每个教师在同一个时间片内最多只能有一门课;(v)属于同一门课程的不同课元不能安排在同一天;(vi)其它必须满足的条件(比如,由于某种原因要求某些特定时间不能排课等)。

2)优化约束条件。旨在使课程的安排尽可能合理、科学,例如:(i)同一门课程的多个课元的时间要间隔均匀,而课室则尽可能相同;(ii)尽可能给重点课程的课元分配好的时间片和好的课室;(iii)尽可能将文理课程的课元、主副课程的课元交错安排;(iv)尽可能满足每个课元教学的合理要求,比如对多媒体课室的要求;等等。

3)特殊约束条件。主要是指班级、教师个人提出的一些特殊的合理要求。比如,住在校外的教师、老教师需要在上课的时间和地点方面酌情关照等等。

设 C 是所有课元的集合, TR 是所有时空片的集合,由 TR 可导出所有时空片簇的集合。将问题的解定义为给每个课元按照其周学时数分配互不相交的时空片簇的方案,满足了全部硬约束条件的解就是可行解(即一张课表)。最优解为最大程度满足优化约束条件和特殊约束条件的可行解(即一张最满意的课表)。

将所有的优化约束条件和特殊约束条件依次编号为1,2,...,s,并对它们赋予相应的权重 $W_i \geq 0 (1 \leq i \leq s)$ 。为了在数量

^{*}国家自然科学基金资助项目(10201009);广东省自然科学基金资助项目(021072)。陈卫东 讲师,主要研究方向:计算复杂性理论、算法设计与分析。李吉桂 教授,主要研究方向:人工智能。

上体现这些约束条件的满足情况,我们引入两个概念。

定义2.1 设 t 是可行解, $i \in C$ 是课元, 定义函数

$$f(i, t) = W_1 X_1 + W_2 X_2 + \dots + W_s X_s \quad (1)$$

并称 $f(i, t)$ 为课元 i 在可行解 t 中的不满意度。其中, $X_j \in \{0, 1\} (1 \leq j \leq s)$ 表示在 t 中课元 i 的安排是否违反了第 j 个约束条件, 若违反, $X_j = 1$, 否则 $X_j = 0$ 。

定义2.2 设 t 是可行解, 定义评价函数(罚函数):

$$E(t) = \sum_{i \in C} f(i, t) \quad (2)$$

并称 $E(t)$ 为可行解 t 的不满意度。

由于排课问题的模型没有定式, 考虑到我国高校的实情, 我们参照许多文献的做法, 取(2)式作为我们的目标函数^[1, 6, 7]。于是我们可以对高校排课问题作如下形式描述:

$$[\text{UTTP}] \quad \min_{t \text{ 是可行解}} E(t) \quad (3)$$

满足(3)式的可行解 t 即为相对于权重 $W_1, W_2, \dots, W_s$ 的最优解, 简称为最优解。

对于权重 $W_i \geq 0 (1 \leq i \leq s)$ 的确定, 一般来说, 优化约束条件的权重较大, 而特殊约束条件的权重较小。当然, 人们在实施排课时, 若需要优化标准侧重于某个条件, 则可以人为地赋予该条件较大的权重, 使得产生的课表可以满足实际要求。

使用启发式算法来求解(3)式, 就是求取尽可能使 $E(t)$ 达到较小值的可行解 t 。

3 传统的直接启发式算法

直接启发式算法, 一般是首先根据各课元的约束性赋予以优先权, 然后按照课元的优先权次序来依次给每个课元安排其“理想”的时空片簇, 直到得到整张课表。这里结合 UTTP (3)式来描述直接启发式算法。它们通常要采用如下两个策略: 优先权策略和最佳分配策略。

3.1 优先权策略

课元一般具有属性: 类型、周学时、班级数、教师、所属课程包含的课元个数等。根据这些属性可以给课元赋以优先权, 比如可采用属性的线性组合来确定每个课元的优先权。

一般地, 课元的优先权值越大, 课元的约束就越多, 从而可供其选择的满意的时空片簇就少, 所以应该优先安排。优先权策略就是, 按照课元的优先权由大到小依次来给课元安排时空片簇。

3.2 最佳分配策略

定义3.1 在任一时刻, 称由所有已被“合法”安排了时空片簇(即满足硬约束条件)的课元子集 C_{sub} 在 TR 中形成的部分课表 $TR(C_{sub})$ (它同时也表示了剩余时空片簇)为系统在此时刻的格局。所有课元均已被安排了时空片簇的格局称为目标格局(即可行解)。剩余课元中尚有课元可合法安排的非目标格局为活格局, 否则称为死格局。

定义3.2 在某活格局 $TR(C_{sub})$ 下, 对于课元 $c \in C - C_{sub}$, 称 $TR(C_{sub})$ 中每一个能合法安排下 c 的空闲时空片簇为课元 c 的可行时空片簇。

定义3.3 在某活格局 $TR(C_{sub})$ 下, 考虑课元 $c \in C - C_{sub}$ 的安排。设 $FS \neq \emptyset$ 是该格局下课元 c 的可行时空片簇集。对于任意的 $ts \in FS$, 定义课元 c 对 ts 的不满意度为:

$$g(c, ts, TR(C_{sub})) = W_1 X_1 + W_2 X_2 + \dots + W_s X_s \quad (4)$$

其中 $W_j (1 \leq j \leq s)$ 为评价函数 $E(t)$ 中的权值, $X_j \in \{0, 1\}$ 表示在部分课表 $TR(C_{sub})$ 中若将 ts 分配给 c 后 c 的安排是否满足第 j 个约束条件, 若不满足, 则 $X_j = 1$, 否则 $X_j = 0$ 。

显然, 若从格局 $TR(C_{sub})$ 最后能到达一目标格局 $TR(C)$, 则有 $g(c, ts, TR(C_{sub})) \leq f(c, TR(C))$ 。

最佳分配策略就是将满足 $\min_{ts \in FS} \{g(c, ts, TR(C_{sub}))\}$

的 ts 分配给课元 c 。

3.3 算法描述

根据上面的两个策略, 对直接启发式算法描述如下。

```

procedure Heuristic
input: 课元集  $C$ , 时空片集  $TR$ 
output: 课表  $TR(C_{sub})$ 
begin
    确定每个课元的优先权值, 并将课元按照其优先权值由大到小
    排序
    设初始格局为  $TR(C_{sub})$ , 其中  $C_{sub} = \emptyset$ 
    FailSet =  $\emptyset$  // 算法中不能安排的课元集
    for  $c = 1$  to  $|C|$  do
        1. 在课表  $TR(C_{sub})$  中获取课元  $c$  的可行时空片簇集  $FS$ 
        2. 若  $FS = \emptyset$ , 将  $c$  加入 FailSet 中
        3. 若  $FS \neq \emptyset$ , 将满足  $\min_{ts \in FS} \{g(c, ts, TR(C_{sub}))\}$  的  $ts$  分
        配给课元  $c$ , 并更新格局  $TR(C_{sub})$ 
    end for
end.

```

当然, 算法中可能会遇到某课元无法安排的情况。一些文献^[1]中采用的办法就是, 在遇到这种情况时算法就进行一定限度的回溯, 即调整一个或者多个课元的安排, 给它们换一个较满意的可行时空片簇, 直到当前课元能安排下为止。

通过实验我们发现, Heuristic 算法的结果对于课元优先次序很敏感。即, 排课结果往往会因课元的优先次序不同而差别较大。下面给出的 Exploring 算法能在较大程度上消除这种敏感性, 而 Learning 算法具有学习的能力, 它能够学习到课元的某种优先次序。

4 基于拟人策略的启发式算法

为了避免盲目搜索, 提高所得解的质量, 我们在 Heuristic 算法的基础上针对 UTTP 提出了如下三个新的排课策略。

4.1 靠边策略

在某活格局 $TR(C_{sub})$ 下, 考虑课元 $c \in C - C_{sub}$ 的安排, 其周学时为 p 。在部分课表 $TR(C_{sub})$ 中任意大小为 p 的空闲时空片簇都有可能是其可行时空片簇。例如, 某个最大空闲时空片簇如图1所示。

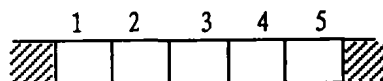


图1

其大小为5。假定 $p=2$, 对于课元 c 而言, 实际上它对应4个大小为2的空闲时空片簇: 1-2、2-3、3-4、4-5。我们仅考虑靠边的两个空闲时空片簇1-2和4-5。这就是时空片簇的靠边策略。即, 在某格局下, 从部分课表中的所有最大空闲时空片簇的靠边子时空片簇中获取当前课元的可行时空片簇。

直观上看, 采用靠边策略一方面可以缩小搜索范围, 另一方面可以使得时空片的剩余空间更好利用, 为后面课元的安排留下更多的机会。

4.2 前景探测策略

在某活格局 $\alpha = TR(C_{sub})$ 下, 考虑课元 $c \in C - C_{sub}$ 的安排。设 $FS \neq \emptyset$ 是该格局下课元 c 的可行时空片簇集。对于任意的 $ts \in FS$, 新格局 $TR(C_{sub}) \cup \{c \leftarrow ts\}$ (在 $TR(C_{sub})$ 中将课元 c 安排在 ts 处后所得的新课表) 均可能是格局 α 的后继格局。用 $h(c, ts, \alpha)$ 来度量每个新格局的前景:

从新格局 $TR(C_{sub}) \cup \{c \leftarrow ts\}$ 出发, 使用最佳分配策略依次对 $C - (C_{sub} \cup \{c\})$ 中的课元进行排课, 若最后能达到一目标格局(可行解), 该可行解的不满意度即为 $h(c, ts, \alpha)$; 若最终到达的是一死格局, 则 $h(c, ts, \alpha) = W_0 * |FailSet|$, 其中

FailSet 是该死格局中未能安排的课元集, W_0 是比任何可行解的不满意度都大的一个数。其搜索过程如图2所示。

前景探测策略就是在当前格局 α 下, 选择前景最佳的那个新格局作为下一个格局, 即将满足 $\min_{ts \in FS} \{h(c, ts, \alpha)\}$ 的 ts 分配给课元 c 。

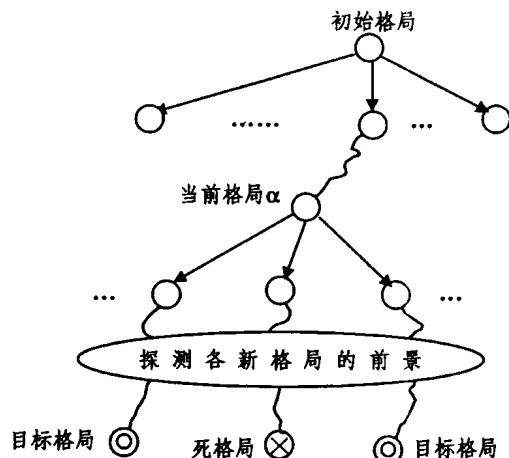


图2

该策略是从博弈中受到启发得来的一种拟人策略, 它模拟对弈者的经验, 每次从当前的格局走向下一个格局之前, 总是要向前看若干步。从效果上看, 该策略又可视为是一种局部搜索策略, 因为在其搜索过程中实际上可能探测到许多可行解, 它选择往更好的可行解方向继续搜索。其效果显然比最佳分配策略要好, 下面的定理2表明了这一点。

4.3 学习策略

学习策略就是, 根据前景探测策略, 若得到一个可行解 $TR(C)$, 则按照(1)式来计算每个课元在这张课表中的不满意度 $f(c, TR(C))$, 然后挑选出最不满意的课元; 若不能得到一个可行解, 而是到达一个死格局, 则认为其中未能安排的课元是最不满意的课元。对最不满意的课元增加其优先权值, 然后再应用前景探测策略进行下一轮排课, 如图3所示。

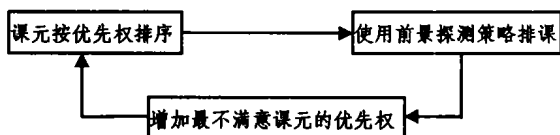


图3 学习策略模型

前景探测策略可看作是一个局部搜索策略, 使用它到达一目标格局或死格局时相当于到达了局部极小值点。学习策略就是如何从局部极小跳到前途可能更好的地方。因此, 该策略能比前景探测策略具有更大的搜索范围。这种策略实质上也是一种拟人的策略。通过实验, 我们发现直接启发式算法及前景探测策略的排课结果都对课元的优先权次序敏感。使用学习策略, 通过若干回合的学习就可以很自然地得到能使算法有较好结果的课元的某种优先次序。

基于上面的策略, 我们得到两个基于拟人策略的排课算法, 其描述如下。

4.4 Exploring 算法

procedure Exploring //基于前景探测策略的排课算法
input: 课元集 C , 时空片集 TR //课元已按其优先权值由大到小排序
output: 课表 $TR(C_{sub})$
begin

设初始格局为 $\alpha = TR(C_{sub})$, 其中 $C_{sub} = \emptyset$

FailSet = $\emptyset$ //算法中不能安排的课元集

for $c = 1$ to $|C|$ do

1. 在课表 $TR(C_{sub})$ 中用靠边策略来获得课元 c 的可行时空片簇集 FS

2. 若 $FS = \emptyset$, 将 c 加入 FailSet 中

3. 若 $FS \neq \emptyset$, 将满足 $\min_{ts \in FS} \{h(c, ts, \alpha)\}$ 的 ts 分配给课元 c , 并更新格局 $\alpha = TR(C_{sub})$

end for

end.

4.5 Learning 算法

procedure Learning //基于学习策略的排课算法

input: 课元集 C , 时空片集 TR , 迭代次数 MAXTRIES

output: 具有最小 $E(TR(C_{sub}))$ 的课表 $TR(C_{sub})$

begin

for $i = 1$ to MAXTRIES do

1. 将课元按照其优先权值由大到小排序

2. 调用算法 Exploring 得完整课表 (FailSet = $\emptyset$ 时) 或部分课表 (FailSet $\neq \emptyset$ 时) $TR(C_{sub})$

3. 计算评价函数 $E(TR(C_{sub}))$: 若 $TR(C_{sub})$ 是完整课表, 按照式(2)来计算; 若 $TR(C_{sub})$ 是部分课表, 则 $E(TR(C_{sub})) = W_0 * |FailSet|$

4. 如果 $E(TR(C_{sub}))$ 是目前最小值, 则保存 $TR(C_{sub})$

5. 找出最不满意的课元 c

6. 若 c 的优先权值已足够大, 则随机选择一个课元增加其优先权值; 否则, 增加课元 c 的优先权值 //使用学习策略

end for

end.

4.6 算法分析

关于算法的时间和空间复杂度, 有如下结果:

定理4.1 Heuristic 算法、Exploring 算法、Learning 算法的时间复杂度分别为 $O(nm^2)$ 、 $O(n^2m^3)$ 、 $O(n^2m^3 * MAXTRIES)$ 。空间复杂度均为 $O(n+m)$ 。其中, m 是课室个数, n 是课元个数, MAXTRIES 是 Learning 算法中的迭代次数。

证明: 在某格局下确定某课元 c 的可行时空片簇时, 时空片簇个数为 $O(m)$, 而判断一个时空片簇是否可行需要时间 $O(m)$, 因此获得 c 的所有可行时空片簇需要时间 $O(m^2)$ 。计算 g 的时间为 $O(m)$, 从而对 c 应用最佳分配策略需要时间 $O(m^2)$ 。因此 Heuristic 算法的计算时间为 $O(nm^2)$ 。对于 c 的每个可行时空片簇 ts , 计算 $h(c, ts, \alpha)$ 时, 要在余下的所有课元上应用最佳分配策略并最后计算所得结果 $E(TR(C_{sub}))$, 需要时间 $O(nm^2)$ 。因此 Exploring 算法中一个格局的变迁需要的时间为 $O(nm^3)$, 而算法运行过程中格局的变迁次数不超过 $O(n)$, 因此 Exploring 算法的时间复杂度为 $O(n^2m^3)$ 。由此容易得到 Learning 算法的时间复杂度为 $O(n^2m^3 * MAXTRIES)$ 。在存储空间方面, 三个算法的都需要空间 $O(n)$ 来存储课元, $O(m)$ 来存储时空片集, 而其它数据的存储空间不会超过 $O(n+m)$, 故共需存储空间 $O(n+m)$ 。□

我们的算法仅在问题的解空间的一个极小部分进行有针对性的搜索, 故对于算法的解质量, 不易从理论上作出定量分析。但由于 Learning 算法的搜索范围包含了 Exploring 算法的搜索范围, 而 Exploring 算法的搜索范围又包含了 Heuristic 算法的搜索范围, 故有如下结果:

定理2 Exploring 算法所得解的质量不低于 Heuristic 算法所得解的质量; Learning 算法所得解的质量不低于 Exploring 算法所得解的质量。

5. 算例及评论

目前已经对一个系的最近两个学期的排课任务的实例使用 C 语言编程在 IBM PC-586 机上进行了自动排课。其中一个实例的描述见图4(a)。

我们对于三个算法的性能通过实验进行了比较, 结果见图4(b)。算法中 $W_i (1 \leq i \leq s)$ 是按照一定比例确定的, 且满足

对于任何可行解 t 有 $E(t) < W_0$, 具体值见表5。可以看出, 对于解的质量, Learning 算法要优于 Exploring 算法, 而 Exploring 算法则明显要优于 Heuristic 算法。从实验中我们还发现, 即使课元优先次序不是按预先给定的课元优先权值

得到, 而是随机的或者是某种自然次序(比如年级次序), 通过若干次学习, 仍然能够得到一种使得算法有较好结果的课元优先次序, 这说明我们的 Learning 算法具有这种学习能力。

班级数目	22
教师数目	50
课程数目	48
课元数目	74
课程周总学时数	239
每个班级的周平均学时数	23
每位教师周平均学时数	4.8
课室数量	8
可用时空片总数	360
可用时空片数/课程周总学时数	1.5

(a)

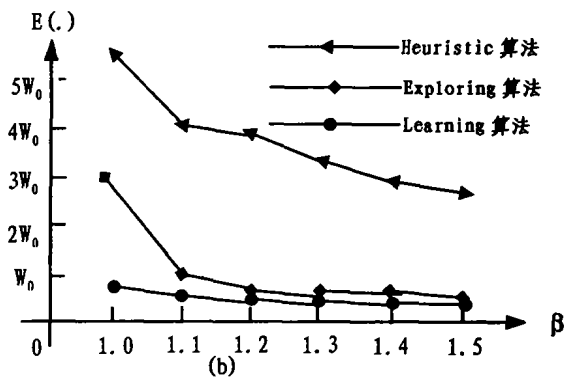


图4 (a)一个实例的描述。(b)对三种算法解的比较。其中 β = 可用时空片数/课程周总学时数, 其值是为了比较算法的性能按照一种自然的时间先后顺序减少可用时空片数而得。图中每个点代表两个算例的结果取平均。其中 Learning 算法的参数 MAXTRIES = 50。

将我们的算法的解与人工排课的结果相比较, 结果见表5。可见, Exploring 算法、Learning 算法的效果不逊色于人工,

而且速度快(执行 Exploring 算法约2秒钟), 而人工排课则需要一周多的时间。

表5 对于图4(a)描述的实例, 手工排课结果与 Exploring 算法、Learning 算法结果的比较

约束条件 (优化约束条件和特殊约束条件)	权值百分比 (%)	手工排课结果中违反某 约束的次数 ($E=2634$)	Exploring 算法结果中违反 某约束的次数 ($E=1957$)	Learning 算法结果中违反 某约束的次数 ($E=1841$)
同课程的多个课元时间间隔应均匀	2.8	34	32	25
同课程的多个课元的课室应该相同	7	0	0	2
多媒体要求应该满足	28	6	0	0
课程类型搭配要均匀	3.5	6	6	6
重点课程要有好时间片(指上午)	3.5	66	37	24
某些时段最好不要排课	7	40	37	40
班级日负荷均衡	7	34	8	0
教师日负荷均衡	5	4	4	3
好时间片要充分利用	1.4	77	54	51
好课室要充分利用	1.4	63	52	52
剩余时空片集中“碎片”要少	1.4	15	9	7
教师的特殊要求应该满足	15	14	6	3
其它若干约束条件	17	/	/	/

三个结果均是可行解, E 是其评价函数值。其中 Learning 算法的参数 MAXTRIES = 50。

我们实现的系统具有交互性能, 可全自动排课, 也可在排课过程中手工排课和自动排课交互进行。通过调整模型中的参数, 可得到符合不同要求的课表。由于系统可提供人工干预的机会, 能灵活地处理一些特殊的排课要求, 因此具有良好的适应性和实用性。当然, 算法结束时可能会出现有课元未能安排的情况, 此时可通过加大迭代次数或增加时空片来解决。

参考文献

- Schaerf A. A Survey of Automated Timetabling. Artificial Intelligence Review, 1999, 13: 87~127
- Garey M R, Johnson D S. Computers and Intractability- A guid to

NP-completeness. San Francisco: W. H. Freeman and Company, 1979

- 黄文奇, 金人超. 求解 SAT 问题的拟物拟人算法—Solar. 中国科学(E 辑), 1997, 27(2): 179~186
- 黄文奇, 等. 解 Packing 及 CNF-SAT 问题的拟物拟人方法. 华中理工大学学报, 1998, 26(4): 5~7
- 黄文奇, 许如初. 支持求解圆形 Packing 问题的两个拟人策略. 中国科学(E 辑), 1998, 29(4): 347~352
- Fernandes C, et al. High school weekly timetabling by evolutionary algorithms. In: Proc. of the 1999 ACM symposium on Applied computing, Feb. 1999. 344~350
- 黄干平, 姚自珍, 张铁静. 使用模拟退火算法解解表问题. 武汉大学学报, 2000(10): 559~563