Finding a feasible course schedule using Tabu search

基于遗传禁忌算法解决排课问题

期刊：Discrete Applied Mathematics

THE JOURNAL OF COMBINATORIAL ALGORITHMS, INFORMATICS AND COMPUTATIONAL SCIENCES组合算法，信息与计算科学期刊

Alain Hertz

作者信息:

I am the conductor of the CBTASNH choir

Here is a [recording](http://www.gerad.ca/~alainh/UvNucha.mp3) of our interpretation of the song Uv'Nucha Yomar from Rosenblatt

And here is a [recording](http://www.youtube.com/watch?v=Ej8WK9uqDQQ&feature=youtu.be) of our interpretation of the song Heye im Pifiot from Rivleen

And here is a [recording](http://www.youtube.com/watch?v=s_hb0hnGdug) of our interpretation of the song Veal Yedey Avadecha from Kwartin

And here is a [recording](http://www.youtube.com/watch?v=EoVnqZavGyk) of our interpretation of the song Kol Nidrey

<https://www.gerad.ca/~alainh/>

Departement dlnformatique et de Recherche Operationnelle, University de Montrial, Case Postale 6128, succursale A, MontrPai, Que., Canada H3C 3J7

Abstract

Hertz, A., Finding a feasible course schedule using Tabu search, Discrete Applied Mathematics 35 (1992) 255-270.

We consider a course scheduling problem in which the total number of time periods assigned to each topic has to be split into daily amounts of consecutive time periods called daily quantum. This daily quantum may be arbitrary but are bounded by given minimal and maximal values.

In addition to the classical constraints of course scheduling problems, precedence and time windows requirements are taken into account. We describe a new method based on Tabu search techniques for finding a feasible course schedule.

Keywords. Course scheduling, Tabu search, assignment problems.

我们考虑了排课问题，其中分配给每门课程的总的时间被划分成每日连续时间段被称为每日量。这种每日量可以是任意的，但给定了最小和最大界。

除了排课问题的传统限制，优先，时间窗口要求（time windows requirements）都考虑在内。我们描述寻找一个可行的排课技术——基于禁忌搜索技术找到新方法。

关键词：排课，禁忌搜索，分配问题。

1. Introduction

Finding a feasible course schedule is not an easy task. Conflicts due to courses taking place simultaneously but involving common students or teachers have to be avoided. In addition to these classical constraints, teachers’ availabilities, precedence, compactness, geographical and time windows requirements have to be taken into account. Moreover, a maximal allowed number of time periods is assigned to each working day. Furthermore, a total number of time periods is assigned to each topic; this total amount has to be split into daily amounts of consecutive time periods called daily quantum and bounded by given minimal and maximal values. The set of consecutive time periods of a daily quantum is defined to be a course. It follows that the number of courses is not known in advance and their length (number of consecutive time periods) is not fixed. Timetabling problems have already been studied by many authors.

Different approaches have been proposed [l-5,7,9,12,15] but most of them take into account only parts of the constraints and cannot be easily adapted to the case where the number of courses and their length are not known in advance. The purpose of this paper is to present a new global approach based on Tabu search techniques for tackling course scheduling problems in which the length of the courses are not necessarily known in advance. We shall first develop a model for the class-teacher timetabling problem. It will then be shown that this model can easily be extended for handling course scheduling problems in general. In the next section the course scheduling problem is formulated. The basic ideas of Tabu search techniques are sketched in Section 3. The adaptation of Tabu search to the class-teacher timetabling problem is described in Section 4, and experiments are reported in Section 5. Finally, extensions are discussed in Section 6 and concluding remarks are given in Section 7.

一个可行的课程安排不是一件容易的事。由于很多课程同时在开，但对于同一​​学生或教师的课程冲突必须避免。除了这些传统的限制，教师可以上课，优先级，紧凑性，地点和时间窗口的要求，必须加以考虑。此外，每个工作日都有时间段数量的限制的。此外，每个时间段都被分配了一门课，这个时间段每天分配给连续时间段被称为每日量。并有最小值和最大值的界限。一天中连续的时间量被定义为一门课程。它遵循的课程的数量事先是不知道的，它们的长度（数个连续的时间段）是不固定的。已经有很多人在研究排课问题。不同的方法已经被提出[1- 5,7,9,12,15]但其中大多数只考虑了部分的限制，并且在课程的数量和它们的长度不在已知的情况下提前不适用。本文的目的是提出了一种基于禁忌搜索技术的新全局方法，可以解决在课程的长度实现并不知道的排课问题。我们首先制定了班级与教师排课问题的模型。该模型能够很容易地扩展来处理一般排课问题。在下一节中，排课问题被形式化表示。第3节大概表述了禁忌搜索技术的基本思想，第4节 描述了修改禁忌搜索算法来解决排课问题。第五部分做了实验，最后在第六部分做了扩展，第七部分是结束语。

2. Problem formulation

2.问题描述

This paper has been motivated by a real-life class-teacher timetabling problem which is described in this section. There are several classes of students and each class follows its own fixed curriculum. A set of available working days within which all class curriculums must be completed is given. Each curriculum consists of a given set of subjects. For each subject, the earliest working day (release date) at which it can start and the latest working day (due date) at which it must be completed are specified. Each subject consists of a given set of topics（专题）. The topics of the curriculum of a class are partially ordered: this ordering specifies a predecessor-successor relation between the topics. Topics of different subjects can be in relation while topics from different classes are not in a predecessor-successor relation.

A fixed teacher is assigned to each topic and a teacher can teach one or more topics to one or more classes. The teachers and the classes are not necessarily available at each time period (a time period is usually defined as a 45-minute lecture hour) of each working day. For each working day a total number of time periods is given which can vary from day to day. Moreover, lunch breaks are fixed in advance and a course should not be interrupted by these breaks. A total number of time periods is assigned to each topic. This total amount must be split during the scheduling process into daily amounts of consecutive time periods called daily quantum. The set of consecutive time periods of a daily quantum is defined to be a course. The length of a course is its number of consecutive time periods. We distinguish two kinds of topics:

- for static topics, the daily quantum is fixed and ordered in advance,

- for dynamic topics, the daily quantum is arbitrary but bounded by given minimal and maximal values. For example, a dynamic topic with twelve time periods and for which each course consists of two or three consecutive time periods can be divided into the following unordered sequences of daily quantum: (2,2,2,2,2,2), (2,2,2,3,3) or (3,3,3,3). Dynamic topics appear quite naturally in practical problems. When a school has to build a course schedule, each teacher initially indicates how the total amount of time periods of his topics should approximately be distributed. This kind of information is in general not precise and the scheduler decides which is the length of each course. These choices may dramatically reduce the number of feasible course schedules. A model with dynamic topics avoids these undesired restrictions.

Let US show by an example how the requests of a teacher can be taken into account without fixing the length of the courses in advance. Let us consider a teacher who has to distribute eight time periods in a week. He would like to teach one course of length two on Friday and the six remaining time periods should be scheduled before Thursday. The length of each course should be two or three. In such a situation, the scheduler can define two different topics:

- One static topic (with one daily quantum of value two) representing the course to be scheduled on Friday. The release and the due date of this topic are Friday.

- One dynamic topic with six time periods. The release date of this topic is Monday, its due date is Wednesday, the maximal value of a daily quantum is three and the minimal value is two.

The six time periods of the dynamic topic represent either three courses of length two or two courses of length three. A model without dynamic topics would forbid one of these two possibilities. The problem to solve is to find a feasible course schedule for the given class curriculums and the given set of working days such that: (a) each teacher is involved in at most one topic at a time, (b) each class is involved in at most one topic at a time, (c) the predecessor-successor relation between the topics is satisfied, (d) the daily quantum of a static topic are correctly ordered, (e) the release and the due dates of all subjects are respected, (f) each topic is scheduled in such a way that the corresponding teacher is available during each time period of each course of the topic, (g) each course is scheduled in an uninterrupted way during available time periods of a working day, (h) two courses of a same topic are scheduled on two different working days, (i) the minimal and maximal numbers of consecutive time periods of each daily quantum of a dynamic topic are not violated.

This problem is known to be difficult. If we relax constraints (c), (d), (e), (h)and (i), and if we assume that we have no dynamic topic, the relaxed problem of finding a course schedule satisfying constraints (a), (b), (f) and (g) is proved to be NP-complete unless all teachers and all classes are always available. In classical timetabling problems, a given set of courses has to be scheduled, taking into account several constraints. We solve here a quite different problem. We do not know the number of courses and their length in advance. This makes the problem more complicated. The numerous scheduling methods described in the literature can generally not be applied in this case. For solving this timetabling problem we shall adapt the Tabu search technique which will be described in the next section.

本文的动机是被在本节中描述的一个真实的排课问题所激发的。有几个班的学生，每个班有自己固定的课程安排。一周工作日内，所有不同的课程都必须上完。每个课程安排由一组给定的科目组成。对于每一个科目，可以开始的最早的工作日（release date）和必须完成的最后的工作日（due date）都是确定的。每个科目（subject）由一组给定的课程（topics）组成。一个班级的课程安排的课程是部分有序的：这个顺序指定课程之间的前后继承关系。不同学科的课程可以是有关系的，而不同班级的课程没有前后继承关系。

每一门课有固定的老师，一个老师能给多个班级交多门课。教师和班级不一定在每个工作日每个时间周期都有课（时间周期通常是一个45分钟的时间段）。每个工作日时间周期的总数目可以不同。此外，午休时间被预先固定并且一门课不能被这些休息时间打断。每门课被分配了一定的学时。学时总数在排课过程中必须分配到每天的连续时间，称为每天量。每天量的连续时间段集合被定义为一门课程。一个课程的长度是它的连续时间周期的数目（比如：48个学时）。我们区分两种类型的课程（topics）：

- 静态的课程，每天量是固定的，并且提前安排好了

- 动态的课程，每天的量是任意的，而是有给定的最小值和最大值的限制。例如，有十二个时间段的动态课程，每节课包含了两到三个连续的时间段可以被划分为下面未排序的每天量序列（2,2,2,2,2,2），（ 2,2,2,3,3）或（3,3,3,3）。动态课程在实际问题中出现很正常。当学校必须建立一个课程表，每位老师一开始会说明他的课程安排的大致分布。这种信息通常都不是精确的并且调度者决定每节课的长度。这些选择可以大大降低可行的课程安排的数量。动态课程模型避免这些不需要的限制。让我们通过一个例子来说明即使没有事确定课程的长度如何考虑老师的要求。我们考虑，一个老师拥有在一周分配八个时段。他想在周五教一门两个课时的课，剩下的6个课时应该在周四之前的时间安排。每节课的长度应该是两个或三个课时。在这种情况下，调度程序可以定义两个不同的规则（topics）：

- 一个静态的规则（有两小时的每天量）代表课程在周五安排。这个规则中开始和结束时期都是周五

- 有6个时间段一个动态的规则。本规则的发布的日期是星期一，它的到期日是星期三，每日量子的最大值是三个，最小值为两个。

动态主题的6个时间段长度代要么是三次两个课时要么是两次三个课时。没有动态规则的模型禁用两个规则之一。要解决问题就要为得到可行的课表，在给定课程安排和给定的工作日集合：（a）每个教师在一次只能上一门课（b）一个班级一次只能上一门课（c）课程之间的前后继承关系得到满足，（d）静态课程的日常量正确排序，（e）开始和结束的日期要满足（f）每个课程用这样的方式安排：相​​应的教师在每门课程的每个时间段都要是可以上课的（没有其他课需要上）（g）每一课程是在工作日期间的可用时间段，（h）一门课程的两节课要安排在两个不同的工作日（i）每个动态课程的每天量的最小和最大数不能违反。

这个问题都知道是很困难的。如果我们放松约束（c），（d），（e），（h）和（i），并且如果我们假设我们有没有动态的课程，寻找满足约束（a），（b），（f）和（g）的课程表被证明是NP完全问题（NP完全问题(NP-C问题)，是世界七大数学难题之一。 NP的英文全称是Non-deterministic Polynomial的问题，即多项式复杂程度的非确定性问题。），除非所有的老师和所有的班级都是始终是可以上课的。古典时间安排问题中，考虑一些约束，来安排课程集合。在这里，我们解决了一个完全不同的问题。我们不知道的课程的数量和它们的预先长度。这使问题更为复杂。在文献中描述的许多调度方法通常不能在此情况下被应用。为了解决这个时间表问题，我们修改了的禁忌搜索技术, 将在下一节中描述。

3. Tabu search techniques

Tabu search is a metaheuristic designed for getting a global optimum to a combinatorial optimization problem. It has been first suggested by Glover [lo] and independently by Hansen et al. [l 11 for a specific application, and later developed in a more general framework. A description of the method can be found in [6, 10]. Tabu search has already been efficiently adapted to a large collection of applications [6,11-14,16] We shall sketch here the basic ideas of the technique.

An objective function f has to be minimized on a set X of feasible solutions. A neighborhood N(s) is defined for each solution s in X. The set X and the definition of the neighborhood induce a state-space graph G (which may by the way be infinite). Tabu search is basically an iterative procedure which starts from an initial feasible solution and tries to reach an optimal solution by moving step by step in the state-space graph G. Each step consists in first generating a collection V\* of solutions in the neighborhood N(s) of the current solution s, and then moving to the best solution s’ in V\*, even if f (s?> f (s). Let us write s’=s⊕m with the meaning that s’ is obtained by applying a modification m to s. The solutions consecutively visited in the iterative process induce an oriented path in G. Finding the best solution in V\* may sometimes be a nontrivial matter. It may be necessary to solve the optimization problem min(\_f(si) 1 Si E V\*} by a heuristic procedure.

A risk of cycling exists as soon as a solution s’ worse than s is accepted. In order to prevent cycling to some extent, modifications which would bring us back to a previously visited solution should be forbidden. But it may sometimes be useful to come back to an already visited solution and continue the search in another direction from there. This is realized in Tabu search by keeping a list T containing the last k modifications (k may be fixed or variable). Whenever a modification m is made for moving from s to s’, m is introduced in T and its reverse is considered as tabu.

Deciding that some moves are tabu moves may be too absolute: it is shown in [6] that moves to solutions which have not been visited may be tabu. For this reason, it should be possible to cancel the tabu status of a move if it seems desirable to do so. This is realized as follows. Let s be the current solution and m a modification which we want to apply to s. A penalization a(s, m) and a threshold value A(s, m) are computed: if a(s, m) <A(s,m),then the tabu status of m(at s) is cancelled. We can for example define a(s,m)=f(s+m) and A(s,m)=f(s\*) where s\* is the best solution encountered so far: the tabu status of m is cancelled if the solution s’=s+ m is better than the previous best solution s\*.The function A is called the aspiration function.

Stopping rules have to be defined. if a lower bound f \* of the minimum value of f is known then the process may be interrupted when the value of the current solution is close enough to f \*, Moreover, the procedure is terminated if no improvement of the best solution s\* found so far has been made during a given number nmax of iterations.

A general description of Tabu search is given in Fig. 1. In the next section we shall describe the adaptation of Tabu search to the course scheduling problem.

3.禁忌搜索技术

禁忌搜索是一个亚启发式设计的专为解决组合优化问题得到一个全局最优解。它最先由Glover提出并因一个特定应用被Hansen独立出来，后来在一个更通用的框架中开发。该方法的描述可以在[6,10]中找到。禁忌搜索已经被有效地适应于很多应用[6,11-14,16]。我们将在这里描述该技术的基本思路。

目标函数f必须要在可行解集合X最小化。邻域N（S）是在集合X中的每一个解。集合X和邻域的定义能够得到解空间图G（可以是无限的）。禁忌搜索基本上是从一个初始可行解开始并试图通过在状态空间图G中一步步移动来达到最优解。每一步的关键在于首先在当前解s的邻域N(s)中产生一个解集V\*，然后即使f (s‘)> f (s)也在V\*中移动到最优解。记s’=s⊕m表示s’用过应用一个从m到s的变换。结果连续地在迭代程序中产生一个有向图。找到在V\*中的最优解可能有时候是一个很不容易的事情。通过启发式过程来解决min(f(si) | Si ∈V\*}的最优问题是很有必要的。

只要比s 差的结果s’被接受，那么就会存在循环风险。为了在一定程度上防止循环，可能会把我们带回之前访问过的解的变化是被禁止的。但是有时候回溯到一个已经访问过的解可能会有用，并从那里继续从另一个方向上搜索。这在禁忌搜索是通过保持一个包含了最后k（k是固定或可变的）个操作的列表T实现。每当操作m被用于从s移动到s’而取得，m 在T中引入，它的反向被认为是禁忌搜索。

确定一些动作是禁忌搜索动作可能是太绝对了：文献[6]所示，移动到还没有被访问过的解可能是禁忌的解决方案。出于这个原因，如果值得这样做做应该有可能取消一个移动的禁忌状态。实现方式如下。令s为当前解决方案，m是我们希望应用到s的变形。

一个惩罚a(s, m)和阈值A(s,m)计算：如果a(s, m) <A(s,m)，则m的禁忌状态（在S）被取消。例如，我们可以定义a(s,m)=f(s+m)和A(s,m)=f(s\*)，其中S \*是目前的最好的解决方案：m的禁忌状态将被取消，如果解s’=s+ m比以前的最优解决方案S \* 好.那么函数A被称为启发函数。

停止循环的规则必须定义。如果f的最小值的一个下界f\*是已知的，那么该过程可以在当前解的值是接近f \*时停止，此外，如果最优解s\*在迭代了给定的最大次数之后目前仍没有改进，那么该过程被终止。

禁忌搜索的一般性描述图1中给出。在下一节中，我们将介绍禁忌的适应搜索到排课问题。

4. Adaptation of Tabu search

Many authors have formulated the course scheduling problem as an assignment

problem [3,4,9,12]. Guidelines for adapting Tabu search to assignment problems

have been described in [12]; the same paper contains a description of the algorithm

TAT1 which is an example of such an adaptation to a course scheduling problem

where starting times have to be assigned to courses. For our timetabling problem

we cannot use the algorithm TAT1 since the number of courses and their length are

not known in advance.

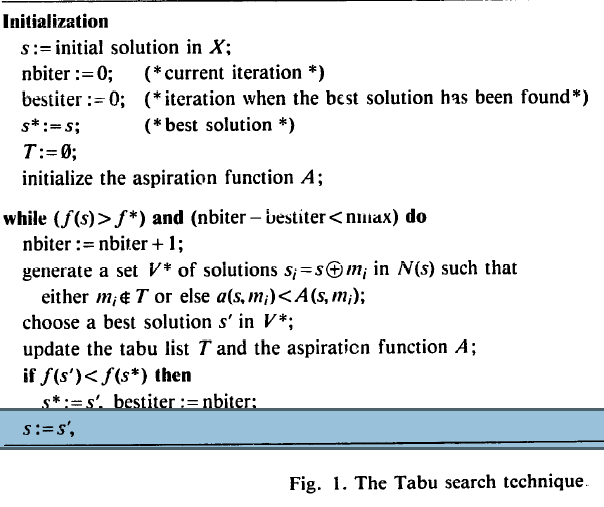
Let us however formulate our course scheduling problem as an assignment problem

where each element from a set S of conflicting objects is assigned to exactly one

element of a set P.

The daily quantums of a static topic are fixed in advance. They induce courses

of given length. These courses are defined to be s-objects.



We shall consider each time period of a dynamic topic as a second kind of objects

which are called d-objects.

The set S consists of all s-objects and d-objects, and the set P consists of all

available time periods. Now, an available time period of a working day has to be

assigned to each object of S; for an s-object the time period represents the starting

time of the course. Any assignment satisfying all the constraints (a)-(i) induces a

feasible course schedule.

4.1. The set X of feasible solutions

Eiselt and Laporte [7] have divided the requirements into hard, medium and soft

ones. The hard requirements have to be respected at all costs while soft ones are only

desirable; medium requirements should be satisfied but are nevertheless relaxable.

In most approaches, a timetable is defined to be feasible if it satisfies all hard requirements:

soft requirements are modelled as objectives while hard ones appear as

constraints. The solution methods designed for dealing with timetabling problems

usually consist of two phases: in phase I a heuristic procedure tries to find a feasible

timetable, and in phase II the current solution is improved by satisfying more

medium or soft requirements and without violating any hard one.

Let us define the following state-space graph G:

- each feasible timetable is a node in G,

- an arc links node x to node y if y is considered as an improvement of x in phase

Phase I can be viewed as a generation of a node in G, and phase II is a descent

method which moves step by step in the state-space graph G until a local optimum

is reached. For large scale timetabling problems, finding a feasible course schedule

is not an easy task. Moreover, if phase I succeeds in finding a feasible solution, this

solution and the best one are not necessarily in the same connected component of G.

We shall define a connected state-space graph G: and phase II will be replaced

by an adaptation of Tabu search techniques. We have now to define the set X of

feasible solutions, that is the nodes of the state-space graph G.

There are two ways for handling with a constraint: it can either be satisfied by

each solution of the set X of feasible solutions, or else it can be penalized in the objective

function. It is to be decided which from among all the requirements will

always be satisfied and which one will be penalized.

An object should never be assigned to a time period if such an assignment can

only induce course schedules which are not feasible. For example, each object is a

portion of a subject x and thus should never be scheduled after the due date or

before the release date of x. For this reason, constraint (e) should always be

satisfied.

On the other hand, a conflict between two objects should not be a sufficient condition

for hindering an assignment.

Constraint (a) for example should only be penalized and not always satisfied.

Guided by these ideas, it was decided to penalize constraints (a)-(d) and always

satisfy constraints (e)-(h).

Constraint (i) has been split into two distinct requirements: (i1) (respectively (i2))

requires that the minimal (respectively maximal) number of consecutive time periods

of each daily quantum of a dynamic topic is not violated.

It was decided to penalize (i1). This choice is justified by the fact that the assignment

of exactly one object is changed at each iteration of the Tabu search procedure:

for dynamic topics this means that one time period is moved at a time, and

removing from a working day a whole daily quantum with minimum value > 1 can

only be achieved by violating constraint (il). Constraint (i2) is included in the set

of always satisfied requirements since any feasible schedule can always be obtained

without violating this requirement.

For a dynamic topic, the set of d-objects which are scheduled on the same day

is defined to be a course. Let us consider a course of a dynamic topic satisfying constraint

(g). The following modifications can be obtained with several steps of the

Tabu search procedure while always satisfying constraint (g):

- the amount of consecutive time periods is not changed but the course is moved

to another portion of the working day,

- the daily quantum is increased or decreased.

These justify our decision to never violate requirement (g).

Constraint (h) has to be considered more carefully. If a static topic can only be

scheduled in a number n of different working days and if its number of daily quantums

is equal to n then constraint (h) should only be penalized since the daily quantums

are not necessarily correctly ordered, and satisfying constraint (d) can then

only be achieved by violating requirement (h). But this case is not frequent in real life

timetabling problems and it was supposed that such a situation does not occur.

Moreover, the definition of a course of a dynamic topic implies that constraint (h)

is satisfied for any dynamic topic. For these reasons, it was decided to forbid any

solution violating requirement (h).

In summary, the set X of feasible solutions contains any assignment of time

periods to the objects of S such that constraints (e), (f), (g), (h) and (i2) are satisfied.

4.2. The objective function

For each topic t, we denote:

n(t): its number of daily quantums,

qi(t): (i= l,..., n(t)) one of its courses,

r(t): the minimum number of time periods in a daily qantum,

b(t): the first time period of course qi(t)

ei(t): the last time period of course qi(t)

B(t): the first time period of t (B(t) = min{ b,(t) |i = 1, . . . , n(t)}),

E(t): the last time period of t (E(t)=max(t) |i = 1, …,n(t))),

P(t): the set of immediate predecessors of t (if t1and t2 precede t and t1

precede t2, then only t2 belongs to P(t)).

If t is a static topic, then n(t) is fixed and qi(t) must precede qj(t) for any j>i.For a dynamic topic i, n(t) and the values ei(t) - bi(t) are variable.For two objects i and j of S, the value d(i, j) denotes the number of time periods during which i and j are held simultaneously. Since the length of an s-object may be greater than one, it follows that two objects assigned to different time periods can be in conflict. The class involved in an object i of S is denoted c(i), and t(i), is the teacher which gives the course. Let ntopic denote the number of topics.

For any solution s of X, the fallowing objective function f is defined:

The parameters p1, . . . ,p5 give relative weights to constraints (a), (b), (c), (d) and

(i1) respectiveiy. A solution s induces a feasible course schedule if and only if

f(s)=O= f \*.

4.3. The neighborhood N(s)

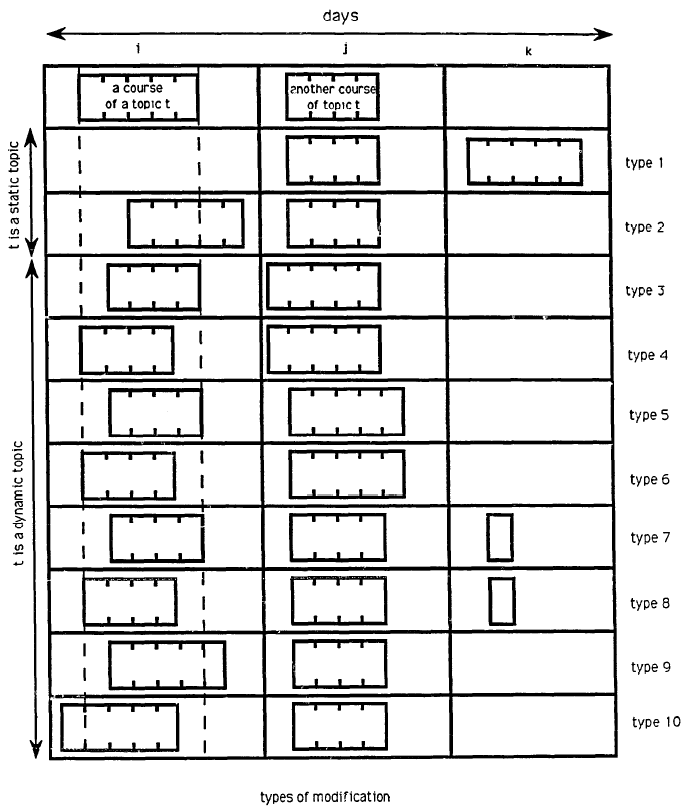
A solution S’ ∈ X is considered as a neighbor solution of s in X if it is obtained

by changing the assignment of exactly one object o of a topic t. Let d be the current

working day in which o is assigned. In order to satisfy constraints (g) and (h), the

set of possible modifications must be restricted to ten types which are represented

in Fig. 2:



- If o is an s-object, then it can be assigned to a new working day d’≠d which

does not contain any course of I (type 1), or it can be moved to another portion of

d (type 2).

- If t is a dynamic topic and o is the first or the last time period of a course, then

it can be assigned to any working day d’. If d’≠ d and d’ already contains a course

q of t, then o can be placed exactly before the first time period of q (types 3, 4)

or exactly after its last time period (types 5,6). If d’ ≠ d and d’ does not contain any

course of I, then o can be assigned to any time period of d’ (types 7, 8). If d’= d,

then the first time period of a course can be moved exactly after the last one (type

9) and the last time period can be moved before the first one (type 10).

Of course, all these modifications m are acceptable at s only if s⊕m = S’ belongs

to x.

4.4. The other ingredients

When an object o currently scheduled on the working day d is moved to a new

time period, the pair (0, d) is introduced in the tabu list T. This means that for |T|

steps of the Tabu search procedure, it is not allowed to assign any time period of

d to 0.

The set V\* is generated randomly but contains only new assignments obtained by

changing the starting time of an object which gives a strictly positive contribution

to f(s). In other words, objects which respect all the constraints are not moved.

In order to allow the tabu status of a move to be cancelled, we consider the

penalization function a(s, m) = f (s⊕m) and the aspiration function A(s,m)= f (s\*).

4.禁忌搜索修改

许多作者都将排课问题形式化为分配问题[3,4,9,12]。为适应禁忌搜索到分配问题的指导方针在[12]已经描述; 包含算法的描述TAT1的相同的论文，这是开始时间必须被分配课程的一个课程调度问题的例子。对于我们的时间表问题我们不能使用算法TAT1，因为课程的数量和它们的长度是事先不知道的。

然而，让我们制定我们当然调度问题作为一个分配问题，冲突对象的集合S的每个元素被分配到一个单独的集合P的元素。

静态的日常量是固定的提前。这导致课程是固定长度的。这些课程被定义为S-对象。

\*\*Fig.01

我们应考虑动态主题的每个时间段作为被称为d对象的第二类对象。

集合S包含所有s-对象和d-对象，并且集合P由所有的可用的时间段组成。现在，一个工作日的可用时间周期要分配给S的每个对象;对于一个s-对象，时间段表示出开始课程的时间。任何满足所有约束分配(a)-(i)可行的课程安排导致了一个可行的课程安排。

4.1可行解结合X

Eiselt和Laporte [7] 将需求划分为硬性、中性、软性。硬要求必须找到所有的解，而软要求只要找到满足的即可;中性要求应该得到满足，但仍然比较宽松。

在大多数的方法，如果满足所有硬性要求可行，则时间表可行，当硬性要求表现得强制的时候，软性要求被模拟为目标。被用来设计解决时间表问题的解决方法通常包括两个阶段：在第一阶段启发式方法试图找到一种可行的时间表，并且在第二阶段 当前解被通过满足更短的中性或者软性要求得到改进而不违反任何硬性要求。

让我们定义以下状态空间图G：

- 每一个可行的时间表是在G中的一个节点，

- 第II.步，如果y被认为是x的改进，连接节点x到节点Y的圆弧

第一阶段可以被看作是图G中的一个节点的扩展，而第二阶段是一个下降方法：通过在状态空间图G一步一步移动，直到达到局部最优。对于大型时间表的问题，找到一个可行的课程安排是不容易的事。此外，如果第一阶段成功地找到一个可行的解决方案，这个解和最好的不必在一个图G中的相同连接中。

我们定义一个连接的状态空间图G：第二部分将会被Tabu搜索技术的改编替代。现在，我们要定义一个可行的解集合X，这是状态空间图G中的节点。

有两种方法用于处理约束：它可以通过满足可行解集合X的每一个解，或者它会被在目标函数中禁忌。这是用来决定哪个将会满足，哪个会被禁忌。

如果分配只能产生不可行的解，那么目标永远都不会被分配一个时间段。例如，每一个对象是一个主题x的一部分，因此不应该在截止期日之后安排或在开始时间x之前安排。出于这个原因，约束（E）应始终满意。

另一方面，两个对象之间的冲突应是一个充分条件 用于阻止分配。例如约束（a）应该阻碍，而且不总是满足。

通过这些思想的指导下，就决定惩罚的限制(a)-(d)并总满足约束条件(e)-(h)。

约束(i)已被分成两个不同的要求：（i1）（i2）要求连续时间周期动态主题的每天量的最小（最大）值不能违反。

这由约束i1决定的。这种选择是由分配合理的恰好一个目的是在禁忌搜索过程的每一次迭代而改变：为动态的主题，这意味着一个时间段只能有一个移动，并通过约束i1从每天量与最小值> 1的工作日中去除。约束（I2）被包括在总是满足需求的集合中，因为总是可以在不违反这一要求的情况下得到任何可行的调度。

对于动态主题，同一天中目标d的集合被定义为一门课。考虑一个动态主题满足约束的过程（G）。以下改变可以用的几个步骤而获得禁忌搜索过程，同时始终满足约束条件（g）：

- 连续时间周期的量没有改变，但课程被移动到其他工作日，

- 每日量增加或减少。

这证明我们的决定从来没有违反约束（g）。

约束（H）必须更仔细地考虑。如果静态主题只能在不同工作日数n被安排，如果其每日量数等于n，那么约束（H）应该被禁止，因为日常的量程不必正确排序，并且满足约束d只能通过违反要求h。但是这种情况在现实生活的时间表问题并不常见。因此这样的情况被认为是不会发生的。此外，动态主题的课程的定义意味着约束（H）满足任何动态的主题。由于这些原因，决定禁止任何违反约束（h）的解。

综上所述，切实可行的解决方案的集合X包含的任何时间分配时期S的对象，满足约束（e）、（F）、（G）、（h）和（I2）。

4.2目标函数

每个主题T，我们表示：

N（t）为日常量，

Qi(T）：（i =1，...，N（t））课程之一，

R（T）：时间周期中每日量最小数，

B（T）：当然补气的第一时间段qi(t)

EI（T）：课程qi（t）的最后一个时间段。

B（t）:T的第一时间段（B（T）=min{B(t)|i=1,...，N（t）}），

E（T）：T的最后时间段（E（T）= max(t）|i = 1，...，N（T）））

P（t）:集合T的直接前辈（如果t1和 T2优于T、T1优于t2时，则T2只属于P(t)）。

如果T是一个静态的主题，那么n（t）是固定的，对于任意的j>I,qi(T)都优于qj(t)。对于动态主题I，N（t）,值ei（t）到bi（t）都是可变的。

S的两个对象i和j，值d（I，J）表示ij之间的时间段的数目。因为一个s-对象的长度可大于一，分配到不同的时间段的两个对象可能会出现冲突。

涉及S对象i的班级被表示为c（i），t（i）是老师这给出课程。用ntopic表示主题的数目。

如果X中的任何解S，以下目标函数f定义为：

\*\*Form01

参数p1…..P5给出了约束(a), (b), (c), (d), (i1)相对权重。当且仅当f(s)=O= f \*，解S可以归纳出可行的课程表。

4.3邻域N（S）

如果X集合中的解S’只是s只是通过改变主题t的一个对象就被认为是X集合中s的一个邻域解，设d是o被分配的当前工作日。为了满足约束（g）（h）中，在可能的移动的集合必须被限制为图2中展示的10个类型：

- 如果o为s-对象，那么他可以被安排到一个新的工作日d’≠d，且不包含任何类型1课程。

- 如果t是一个动态的主题且o为所述第一或课程的最后一段时间，然后它可以被分配给任何一个工作日D'。如果D'≠D和D'已经包含t中的课程Q，那么O可以被确定的放到第一个时间之前（类型3，4）或确定的放到时间段之后（5,6型）。如果D'≠ D并且D'不包含任何课程I，可那么o可以被分配至d的任意时间段（类型7，8）。如果D'= D，那么课程的第一个时间段可以被移动到最后一个时间段之后（类型9），并且最后一个时间段可以移动到第一个时间段之前（类型10）。当然，当且仅当s⊕m = S’，那么这些修改m都是可接受的。

\*\*Fig.2

4.4其它成分

当最近被分配到工作日d的目标o被移动到一个新的时间段，那么（o,d）被加入禁忌列表T，这意味着禁忌搜索的步长是|T|，不允许分配从d到o的任何的时间段。

集合V \*是随机产生的，但仅包含只改变起始时间的为f(s)做出正贡献对象的分配。换句话说，满足所有约束的对象是不移动的。

为了禁忌状态取消，我们定义惩罚函数a(s, m) = f (s⊕m)和启发函数A(s,m)= f (s\*).