A RANDOM HOP-AND-FIT COMPUTER ALGORITHM TO SCHEDULE ACADEMIC COURSES IN UPM

C.B.S. Teh

Department of Land Management, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia (chris@agri.upm.edu.my; Tel: 03-89466976; Fax:03-89434419)

INTRODUCTION

As our university supports the increase use of ICT to improve its working environment and daily operations, it is regrettable that there is a crucial area seemingly overlooked by everyone for improvement. This neglected area is the way academic courses are scheduled in UPM. Though UPM's current course scheduling system is widely accepted to be inefficient and problematic, there have been surprisingly little attempts to overhaul the system such as to introduce a fully computerised system.

Every semester our university offers more than 1400 courses and each of these courses has to be scheduled into its appropriate time slot and room location. There are two major challenges in finding an appropriate time slot and room location for a given course. The first challenge is to schedule a given course in such way that it does not *clash* with other related courses. The term *clash* denotes a situation whereby students are unable to take a group of courses together because one or more courses in that group are scheduled at the same time. The second challenge is to find an available room that has a large enough capacity to accommodate the anticipated number of students taking that given course. And because the availability of large capacity rooms is often limited, scheduling courses having large number of students are often very problematic.

Ultimately, these two challenges create a very intricate and time-consuming task for a single person to schedule all courses in the university especially when it involves an over 1400 courses every semester. Unfortunately, UPM handles these problems by a divide-and-conquer technique, whereby it is the onus of the respective faculties to schedule their own courses only. This technique is inefficient because there is a great deal of "student crossovers" among faculties, whereby students are not limited to just taking courses from their own faculties. A student, for example, may take courses offered by as many as three faculties in a semester. Consequently, this method requires a great deal of co-operation among faculties and patience among the involved parties because a late change in the schedule for one course offered by a faculty can disrupt the schedule for most, if not all, courses offered by another faculty. In other words, there is a great deal of inter-dependencies among the course schedules, supposedly developed in separation by the various faculties.

Consequently, the objective of this paper is to introduce a computerised system called *PKK* (*Penjadualan Kursus Berkomputer*) which uses a *random hop-and-fit* algorithm to schedule all courses in the university efficiently in terms of effort, time and cost. It is hoped that PKK would be a step to a more modern and efficient solution to the way courses are scheduled in our university.

DESCRIPTION OF PKK

PKK is a computerised course scheduler; that is, PKK attempts to fit a list of courses into a given time frame and available rooms. The resultant schedule is one that is free of time clashes. It is important to note that PKK does not attempt to find the best or optimum schedule for all courses; its primary objective is only to create a schedule that fits all the given courses. PKK may fail to fit all courses especially if there are too many courses to be fitted into a too

limited time frame or if there is an insufficient number of available rooms. Most often PKK will fail if there are too many potential clashes among courses (*e.g.*, trying to find a time slot for a popular course so that it will not clash with as many as 30 other courses).

That PKK may fail may seem to be a serious flaw, but it is actually this imperfection that enables a computerised course scheduler to be finally developed. A failure to fit all courses may not necessarily indicate a program fault, but more often indicate too few resources available for too many courses or too stringent requirements (*i.e.*, too many potential clashes for a popular course). In the past, there may have been attempts by UPM researchers to develop some sort of a computerised course scheduling system but they may have failed due to their developers' instinctive requirement that a 100% success rate or an optimum schedule to be established each time the program is run. A perfect success rate may simply not exist due to the limited available resources, and that the task of successfully fitting all the given courses in whichever way often outweighs the importance of finding the best or optimum schedule.

WHAT KIND OF DATA PKK REQUIRES

PKK requires five kinds of data. The first kind is simply a list of courses that will be offered in the coming semester. These courses are those that will be fitted by PKK into a schedule. Each course should have information about the anticipated (or, usually, the maximum allowable) number of students and the number of lecture and laboratory credits. Each course should also be specified the *time slot limit*, which refers to latest time the course can be scheduled at. A course with a time slot limit of 16:00 hours, for example, means the course can be slotted up to 16:00 hours inclusive, but not after 17:00 hours (which would also mean no class for this course after office hours).

The second kind of data required is the time frame for the schedule. Usually, this would be from 8:00 to 16:00 hours excluding the "lunch hour" at 13:00 hours. However, due to the large number of courses in UPM, the time frame can be extended to 21:00 hours (*i.e.*, allow classes at night or after office hours). But since most courses should not be held late in the evening or at night, these courses can have time slot limits set to 16:00 hours, whereas courses which may have late classes can have their time slot limits set to 21:00 hours.

The third kind of data required is the list of available rooms and their room capacities. Each room is also tagged its "owner" so that PKK would always fit courses preferably to rooms belonging to the faculty that offers those courses.

The fourth kind of data is the time slots for the *permanent courses*, if any. These courses are so-called because they are already scheduled (*i.e.*, already given or fixed) and they are beyond the users' (*i.e.*, faculty) control; that is, permanent courses are such as SKP*xxxx* or BBI*xxxx* for which their time slots and location cannot be changed from the users' point of view. It should be stressed that the list of permanent courses are optional and only required if, for some reason, there are courses that are scheduled *a priori* and cannot be changed.

The fifth kind of data required by PKK is called the *clash groups*. A clash group is a group of courses that cannot exist in the same time slot because two or more courses in this group tend to be registered simultaneously by students. The main challenge faced by PKK is actually to find time slots in such a way that it will not cause time clashes among the courses belonging in the same clash group.

Consequently, PKK users must ***anticipate*** the courses that tend to be registered simultaneously by students. A practical way to find clash groups is to group courses based on the given course scheme for a particular degree level (*e.g.*, DP, DPPM, DKHP, BSBI and BSPT). Note: the course scheme referred here is the given guideline that lists courses that should be registered by students for each semester of their study. For example, the list of

recommended courses for semester 3 and 4 in the course scheme for the BSBI degree is: BIP3201, BIP3501, SHW3001, KOH3333, SAK3002, MGM2111, BIP3402, BIP3301, PLP3004, SST3302, EPT3101 and FST3003 (**Table 1**). If these 12 courses are placed in a single clash group this means PKK will find time slots in such a way that students can register for any of these 12 courses without any risk of time clash. However, putting too many courses (such as up to 20) into a single clash group–though tempting–should be avoided because PKK may fail to find suitable, clash-free time slots for all of them.

HOW PKK WORKS: THE RANDOM HOP-AND-FIT ALGORITHM

How PKK works can be described analogously to fitting chess pieces in a certain way in a chessboard. *Figure 1*, for example, shows five boards where each board represents a day in a week from Monday to Friday, and each board is divided in the same way into several rows and columns. Every column and row in a board denotes a room and a one-hour period in a day, respectively. In the first board (Monday), for example, the columns represent every available room in the university (including the room capacity), and the rows represent one-hour periods in a day from 08:00 to 16:00 hours (excluding the "lunch hour" at 13:00 hours). In other words, these boards represent the available resources for course scheduling: that is, the available rooms and their capacities, and the hours that are available for both lecture and laboratory classes.

The main task of PKK is then to fit lecture and laboratory *units* in the five boards (*Figure 1*). In PKK terminology, a *unit* is a one-hour period of lecture or laboratory class. The course SST3005 (Basic Soil Science), for example, has 2+1 *credits*, or as PKK would interpret it, 2+3 *units*. This means to schedule SST3005, PKK will have to fit a total of two lecture units and three laboratory units (*i.e.*, one laboratory unit each for the three hours of laboratory).

PKK schedule courses using a random hop-and-fit algorithm. This algorithm is summarised as follows:

1. Select at random a course from the list of courses to be fitted. The COUNTER is reset to one. The COUNTER counts the number of attempts that is made to fit the selected course. If all courses have been fitted, proceed to **step 7**, else to **step 2**.

2. For the selected course, choose at random if a lecture or laboratory unit is to be fitted to the boards. If the course has no laboratory units, then no random selection is performed. Proceed to **step 1** if all units of the selected course have already been fitted successfully.

3. The COUNTER is incremented by one. If COUNTER exceeds LIMIT (usually set at 1000), proceed to **step 6** else continue. Select at random one the boards (*i.e.*, day of week), and for the selected board, a *cell* is in turn randomly selected. As shown by *Figure 1*, a cell location is denoted by its row and column number on the board, and a cell represents a room location and a one-hour time period of class. If the course unit to be fitted is a lecture unit, proceed to **step 4**, else to **step 5** for a laboratory unit.

4. An attempt is done to fit the lecture unit to the selected cell, but whether the unit could occupy the current cell is decided based on these criteria:

    a) Cell occupation fails if the capacity of the selected room is smaller than the course size (*i.e.*, number of students). Proceed to **step 3**.

    b) Continue if the capacity of the selected room is the same or larger than the course size. Because other course units may occupy the current cell row, a check is performed to determine if the occupying course unit would clash with any of the resident courses.

I. Cell occupation fails if the occupying course unit clashes with *two or more* other courses in the same cell row. Cell occupation also fails if it clashes with a laboratory unit. Proceed to **step 3**.

II. If the occupying course unit clashes with only *one* resident course in the same cell row, then calculations are performed to compare the two courses' *popularity* and *efficient use of room*. The number of courses it can potentially clash with determines a course's popularity. Thus, a more popular class is one with the higher number of potential clashes. The efficient use of a room, on the other hand, is determined by subtracting the room size with the course size, which actually gives the size of the room not occupied by students. Thus, a lower value in size difference denotes greater efficient use of the room.

i. Cell occupation fails if the occupying course unit has a lower popularity and lower room use efficiency than that by the resident course. Proceed to **step 3**.

ii. Cell occupation succeeds if the occupying course unit has either higher popularity or greater room use efficiency than that by the resident course. The resident course unit is then displaced (removed) from its current cell location so that it could be re-fitted elsewhere. Proceed to **step 3** to fit the displaced course unit.

III. Cell occupation succeeds if the occupying course unit does not clash with any other courses in the same cell row. Proceed to **step 2**.

5. A laboratory units is fitted differently from a lecture unit, whereby from the selected cell, only the hour period is used to determine the suitability of cell occupation by the laboratory unit. This is unlike the fitting of a lecture unit whereby both the hour period and room is used to determine the suitability of cell occupation (*see* step 4). This is because laboratory classes are often held in specialised laboratories or at the field; thus, the venues for laboratory classes are not as interchangeable as that for lecture classes. To determine the suitability of cell occupation by the laboratory unit, a check is first performed to determine if the occupying laboratory unit would clash with any of the resident courses in the current cell row.

a) The laboratory unit cannot be scheduled for the selected time period if the occupying laboratory unit clashes with *two or more* other courses in the same cell row. Proceed to **step 3**.

b) If the occupying laboratory unit clashes with only *one* resident course in the same cell row, then calculations are performed to compare the two courses' popularity (*see* step 4).

I. The laboratory unit cannot be scheduled for the selected time period if the occupying laboratory unit is less popular than the resident course. Proceed to **step 3**.

II. The laboratory unit can be scheduled for the selected time period if the occupying laboratory unit is more popular than the resident course. The resident course unit is then displaced (removed) from its current cell location so that it could be re-fitted elsewhere. Proceed to **step 3** to fit the displaced course unit.

c) The laboratory unit can be scheduled for the selected time period if the occupying laboratory unit does not clash with any of the resident courses. Proceed to **step 2**.

6. All boards are cleared (*i.e.*, cells emptied), and the task to fit all courses is attempted again but with a new random seed. Nevertheless, the number of times the boards are cleared are limited to the number of times determined by MAX_RESET (usually set at 50). If this limit is reached, the program aborts and PKK has failed to schedule all courses. In contrast, if MAX_RESET has not been reached, proceed to **step 1** after the boards have been cleared and seeded with a new random number.

7. PKK has successfully scheduled all the given courses. Print the timetable and exit.

The above algorithm governs the way PKK fits lecture and laboratory units to the boards. In addition, the algorithm also ensures that:

1. For a given course, its lecture hour is only once a day. This means there are no two- or three-hour lecture in a single day (*e.g.*, there can never be a stretch of 08:00 to 10:00 hours lecture on Monday). Likewise, there are also no two or more laboratory classes in a single day. Nonetheless, laboratory hours may happen on the same day as the lecture hour (*e.g.*, lecture is at 08:00 hours on Monday and laboratory hours are from 09:00 to 12:00 hours on the same day).

2. Laboratory units are always fitted in a sequence of three in three continuous hours (*see Figure 1*). This means laboratory classes can be scheduled from 09:00 to 12:00 hours (a three-hour continuous stretch), but not from 11:00 to 13:00 hours, breaking for lunch, then the laboratory class resumes for an hour at 14:00 hours.

3. Priority is to fit a course into a room belonging to the faculty that "owns" or offers the course. In other words, a course cannot be placed in a room belonging to another faculty (*e.g.*, courses belonging to Faculty of Agriculture cannot be placed in rooms at the Faculty of Engineering). If rooms at the faculty are unavailable, the next option is to find a room at the university level such as room DK1 and KAP-C1.

Two notable points also follow from the above steps:

1. Laboratory hours for a given course may start at 08:00 hours, not necessarily only at 09:00 hours onwards, as currently–and oddly–practised by some faculties.

2. A given course may be placed in different rooms on different days, not necessarily only in the same room (*e.g.*, the course SST1001 may be placed in room BSST1 and BS002 for the 08:00 hours lecture on Monday and Wednesday, respectively).

SYSTEM REQUIREMENTS

PKK is written in the C++ language and was compiled and tested using Visual C++ 6.0 (SP5) (Microsoft. Corp, Washington). PKK is written to comply with ISO/ANSI C++ standards so that PKK is independent of any hardware or software platforms, and can be compiled with any modern C++ compiler. PKK also uses the C++ Standard Template Library (STL) heavily to create a more efficient and bug-free program code.

HOW FAST IS PKK?

The speed of PKK depends not so much on the number of courses to be fitted or the number of available rooms. Its speed instead depends strongly on the number of clash groups, the number of courses in each clash group, and the degree of inter-relationships among the different clash groups. The latter means a course appearing in two or more clash groups

increases the degree of dependency or relationship between clash groups. If the complexities of the clash groups are small, PKK is able to fit 1500 simulated courses in just 10 seconds.

A trial run using actual data nonetheless took several attempts by PKK to fit 110 courses offered by the Faculty of Agriculture for the November semester 2002/03. In this trial run, PKK took about 5 minutes for a successful fitting. Trial runs of PKK have so far been very promising, showing huge savings in terms of effort and time.

THE FUTURE OF PKK

PKK is still in its developmental stage, where its random hop-and-fit algorithm can be easily changed to fit courses in a different way than stated here in this paper. PKK can also be adapted easily to schedule final examination venues and dates.

Nonetheless, the future of PKK actually depends more on the relevant university authorities breaking *status quo* and desiring a huge improvement in the way academic courses are scheduled. Acceptance of PKK also depends strongly on how far human idiosyncrasies can be removed from the course scheduling system. Examples of human idiosyncrasies are such as scheduling lecture classes to be only held in the mornings and not in the evenings, and personal preference of a lecturer for a certain day and time for his or her classes. Such human idiosyncrasies are often irrelevant and disrupt the way courses are scheduled effectively.

PKK code is available upon request from the author.

Table 1: The recommended courses (and their total credits) for the third and fourth semester for the BSBI (Bac. Sc. Bioindustry) students

Semester 3		**Semester 4**	
BIP3201	3	BIP3402	3
KOH3333	4	BIP3301	3
BIP3501	3	PLP3004	3
SHW3001	4	SST3302	3
SAK3002	3	EPT3101	3
MGM2111	3	FST3003	3

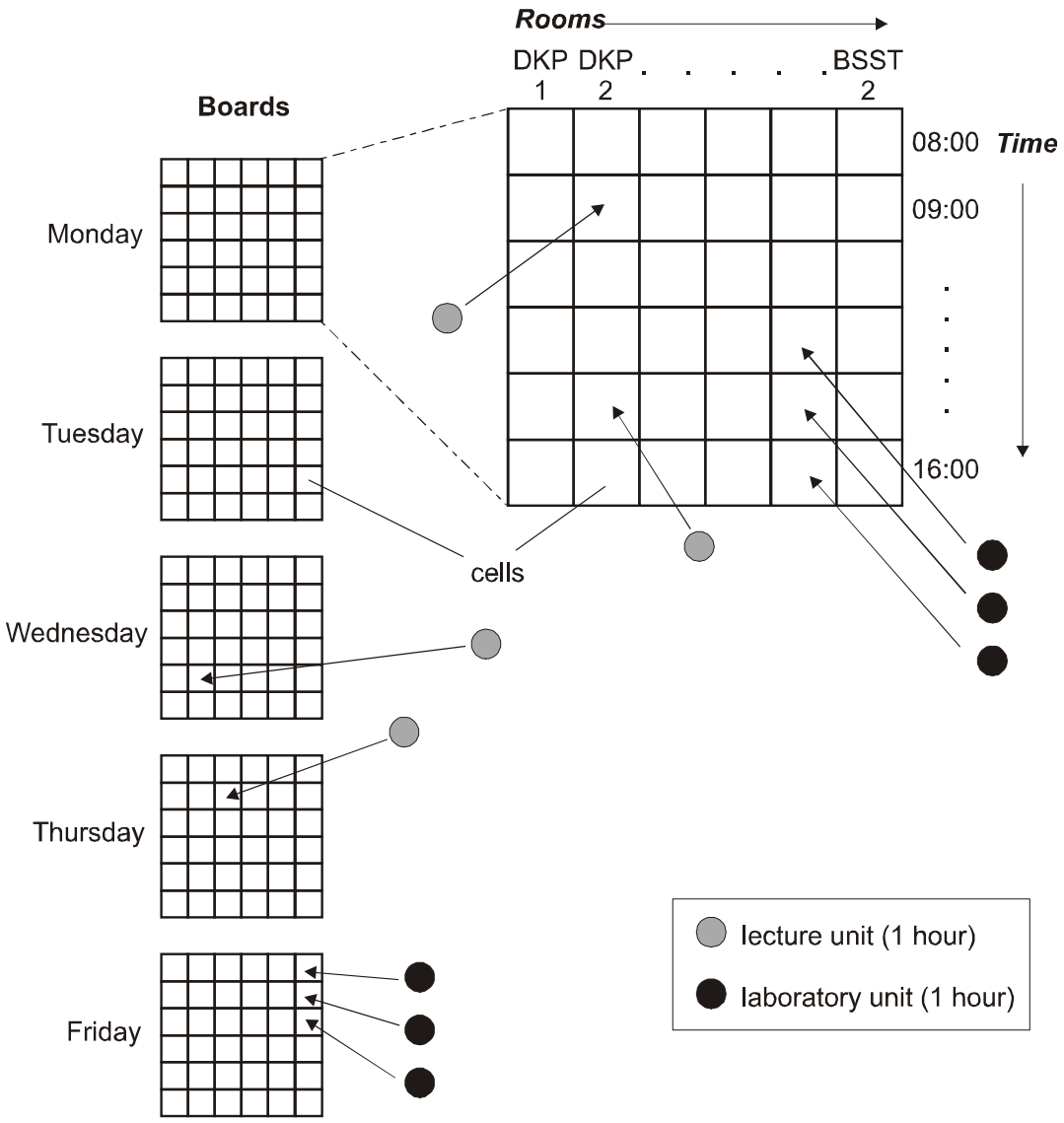

Figure 1 The random hop-and-fit algorithm