

TESS: an interactive support system for school timetabling

W.Y. Ng

Department of Information Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
e-mail: wyng@ie.cuhk.edu.hk

Abstract

A timetabling package called TESS, which stands for Timetabling Expert Support System, has been developed for use in Hong Kong schools. It is representative of the decision support system (DSS), being highly interactive and supportive of different practices of expert timetablers. Its distinctive feature is a suite of seemingly simple tools. Being simple, they behave in manners transparent to the user, who is then ready to command them at will to solve his/her particular timetabling problem.

Keywords

Educational management, interactive, tools, algorithms

1 INTRODUCTION

It is a humbling experience for researchers, especially those who are theoretically inclined, to observe how seemingly intractable problems are solved in practice by the naked mind. School timetabling, an instance of a scheduling problem, is a typical case. Consider Hong Kong where there are over one thousand schools. That translates to over one thousand NP-hard problems solved per year, each with an average of 1000+ lessons, a variety of resource constraints, as well as many conflicting goals for educational and administrative purposes. Typically, a team of one to four teachers spends a few days to produce an acceptable timetable. Apart from a minority of schools, many do so without direct computer assistance. However, the majority have developed ad hoc habits and tactics to facilitate their work.

We may draw a few observations from this. First, whatever the problem of school timetabling may be, it is NOT that feasible solutions cannot be found soon enough. Rather it is the toil of the timetabling process which demands much concentration and hard work from the team. Secondly, stable ad hoc habits and tactics are the mark of an experienced practitioner. Such habits and tactics vary widely, being strongly influenced by the particular school situation such as different degrees of resource scarcity and the relative importance of the many goals and constraints.

To computerize timetabling has been a professed goal of many researchers in operational research (OR) for some thirty years (de Werra, 1985). The classical OR methodology is often applied; namely, a problem model is first derived, working assumptions are then made to adapt it to a tractable form, finally a standard or tailored algorithm is found, and shown to solve the problem with reasonable performance. However, such results mostly remain academic, and few have been implemented for practical use. In fact, school timetabling to date remains largely a handicraft.

There are in fact many useful timetabling packages available (Chahal and de Werra, 1989; de Gans, 1981). The typical package allows the user to describe a school timetabling problem in the form of a particular model deemed amenable to the powerful scheduling algorithm implemented. However none have been in popular use due to many reasons. Some of the reasons are economical: the benefit is not worth the costs of computerization which remained high for schools until recent years. Others are technical, which we shall discuss in more detail later. At this point, it suffices to say, the core problem is that their being powerful in a sophisticated way has made them overly difficult to use.

We are going to describe yet another timetabling package, TESS (CUHK, 1995), which stands for Timetabling Expert Support System. We intend TESS to distinguish itself by properly acknowledging the user as a timetabling expert with particular habits and tactics. TESS assumes the role of a supportive assistant who works (1) to release the expert of the chore of managing the many data and constraints; (2) to compile statistics and comparisons to assist in making better informed decisions; and (3) to carry out iterative and repetitive tasks in a supervised manner. TESS is by no means intended for automation. Its success shall be as an intelligent secretary, who enables its master to concentrate on higher objectives. In school timetabling, that means the quality of class schedules, resource usage, resolution of conflicting goals, etc. Such higher objectives are often compromised when the expert is burdened under the toil of constructing the timetable manually.

We elaborate in the next section our system goals, and in section three the system design and functionality. By then it shall be apparent to the reader that the utility of TESS comes from the synergy of simple but specialized tools integrated in a highly interactive and controllable environment. Section four reports on developing TESS as the Timetabling Application of SAMS (CUHK, 1995; ED, 1995), the School Administration and Management System developed by The Education Department for the 1000+ schools in Hong Kong. TESS was used for the first time in summer 1995 and we shall recount some feedback from the first time users. The final section concludes the paper with lessons learnt and the outlook for future development.

2 SYSTEM GOALS AND APPROACH

2.1 Problem Definition

As previously remarked, the problem with school timetabling is not that feasible solutions cannot be found soon enough. Frankly speaking, the solution is often feasible not with respect to the initially specified set of requirements, goals and constraints, but only with respect to the set compromised to some extent. For instance, some teachers may not have their free periods as requested and some classes may have too many English lessons on one day. However, such compromise is the rule rather than the exception. There being no accurate feasibility test, the set of requirements, goals and constraints have to be continually manipulated at the same time as the timetable is being constructed.

The toil of the timetabling process is however a grave problem. The team suffer heavy cognitive load during those few days, and consequently they are often ready,

probably too ready, to compromise when facing time clashes. The quality of the constructed timetable suffers as a result. As the team is continually drawn to “low-level” tasks related to feasibility, many higher objectives are not given their due attention.

Last but not least, timetabling is more than fitting lessons into time slots. Data preparation takes up a significant portion of the overall time and effort spent. Collected data have to be maintained; class curriculum, staff deployment and lesson accommodation have to be planned; and timetables and various summary reports have to be compiled afterwards.

Many timetable packages have their focus on the design of powerful sophisticated algorithms that minimize the number of kick-outs. With little useful results from the OR community, many developers resort to designing heuristics, some of which emulate what the experienced practitioner does. We contend that such an approach is not effective for the following reasons. First, it is not addressing the true problem of timetabling. Second, it often structures the timetabling so much that the experienced practitioner finds his worthy habits and tactics difficult to apply. Third, it tends to create new problems which sometimes more than offset any time saved from automation. The new problems come from (1) the difficulty of using sophisticated algorithms and (2) the kick-out dilemma - the well-known fact that the fewer the kick-outs, the more difficult they are to handle (Johnson 1980). (The reasoning goes like this: The fewer the kick-outs, the more “powerful” the algorithm must be. Therefore the kick-outs must be more difficult to handle as even the more powerful algorithm does not manage them.)

2.2 System Goals

Naturally, we formulate the system goals in relation to the problem definition above.

First, the system shall enable the user to construct the timetable manually if he so wishes, with comparable and hopefully greater ease than he barehanded. The timetable content may be directly manipulated while moves feasible and infeasible are made visible and logically controlled.

Second, the user should be in control all the time - initiating, configuring, interrupting and backtracking command executions at will; and well informed to be alert to and act upon problems, choices and opportunities early.

Third, the system shall assist the user in handling time clashes - preempting them as much as possible, and finding resolving moves should any arise.

The kick-out dilemma mentioned is a result of time clashes not resolved early enough. Some time clashes are due to incompatible requirements whose resolution necessitates the user’s deliberation, which may not be taken care of by any algorithm. Our system instead aims to empower the user to detect and resolve clashes “on the spot”.

Fourth, the system shall provide comprehensive “spreadsheet” like support for data maintenance as well as timetable planning.

On the whole, the system shall be a supportive assistant who complements the human user throughout the timetabling process. The user shall be released of the chore of low-level tasks, but remains in good control, free to exercise his/her own habits and tactics and attend to higher objectives more effectively.

3 SYSTEM DESIGN AND FUNCTIONALITY

3.1 System Design

An event-driven graphical user interface (GUI) which supports direct manipulation of visually rendered timetable entities is almost mandatory for our stated goals. TESS was implemented as a Microsoft Windows application with a WIMP interface.

Timetable data comprise a rich collection of related entities, including classes, teachers, accommodation, subjects, time slots, requirements, constraints, groups, etc. which warrant careful data modelling. We decided to build TESS as a database application. A database schema was designed to capture the rich semantics of the timetable data. System functions were then built on top of the database management system. Care was taken to ensure that the TESS schema is sufficiently generic for the wide variety of Hong Kong schools.

TESS transactions involve potentially compute-intensive search procedures and the quality of the GUI is critical to its success. Performance and programming flexibility are crucial. Therefore, unlike most database applications implemented on top of 4GL systems like FoxPro or DBase, TESS uses an embedded database engine so as to remain a proper Windows application and avoid the compromise on performance and programming flexibility.

On system functions, we follow the “toolbox” approach (Kernighan and Plauger, 1976) which is appropriate for systems intended for users who are expert in their domain. The toolbox approach calls for the design of a collection of software tools, each of which is highly specialized for a simple task that the user understands well. (That implies we shall not implement any single “blackbox-does-it-all” algorithm.) The system is more transparent as a result and its behaviour more predictable. We also make an effort to make tools more controllable by allowing the user to configure and interrupt tool actions.

3.2 System Functionality

Functionality design is guided by a task model of the timetabling process (Sprague and Watson, 1993) which comprises three stages distinguished by where attention is focused: (1) a relatively conflict-free stage characterized by sparsely filled timetables and high lesson insertion rates; (2) a conflict avoidance stage when conscious effort is spent to preempt time clashes; and (3) a conflict resolution stage when clashes arise that have to be resolved.

Furthermore the tools are designed with the following principles in mind: (1) usability - useful and easy to use; (2) control - with behaviour easy to understand and control and with minimal invisible side effects; (3) flexibility - with minimal constraints on the way they may be used; and (4) power - incorporating powerful algorithms; and in that order. That means, for instance, complex heuristic rules which could have improved the insertion rate, and may be considered powerful, are NOT used due to poor control (being difficult to understand) and potentially adverse usability (being prone to the kick-out dilemma).

Data Model and Management

We focus the description on data management during the core stage when unscheduled lessons are being inserted into time slot.

TESS displays the timetable of a class, a teacher or a location (classroom, special rooms, playground etc.) in a window which shows time slots as being empty, blocked, or occupied by scheduled lessons. A timetable window also responds to mouse actions for inserting or removing lessons.

A lesson in TESS is either simple or a composite lesson block (CLB). A simple lesson is a record with five attributes, namely, class, teacher, location, duration and scheduled-time attributes. A lesson with its scheduled-time attribute being null-valued is unscheduled, and is scheduled otherwise. A CLB is a collection of simple lessons of the same duration that are constrained to have identical scheduled-time attributes. CLB models either explicit concurrence constraints, or implicit ones like split classes and combined classes.

A browser window provides a flexible interface for retrieving and filtering unscheduled lessons by attributes. While scheduled lessons appear as icons in time slots

of their respective timetables, unscheduled lessons are list items with all attributes displayed within the browser window when retrieved. Scheduled lessons may also be click-dragged from one time slot to another.

Prioritizing Unscheduled Lessons

A most powerful tool of TESS is the priority function. TESS computes for each unscheduled lesson a numerical measure that estimates the chance of not being able to find a feasible time slot for it. Retrieved unscheduled lessons may then be sorted to put “difficult” lessons on top. The user is encouraged to exercise the “difficult-lessons-first” heuristic which is most effective for conflict preemption.

The priority function is unique in that we believe it probably outperforms any experienced user in early detection of imminent conflicts.

This tool is placed in the browser window, and the unscheduled lessons may also be sorted by their attributes. Using the retrieval functions and the sorting functions together effects very flexible manipulation of even a large number of unscheduled lessons.

Scheduling Unscheduled Lessons

By means of the retrieval and sorting functions, the user may select and order unscheduled lessons at will in the browser window for subsequent scheduling. TESS provides two FIND functions for finding time slots for a given unscheduled lesson, namely, FIND-FEASIBLE and FIND-BEST. FIND-FEASIBLE finds the time slots for which all entities involved, namely class(es), teacher(s) and location(s) for the unscheduled lesson, are available. FIND-BEST selects the “best” among the feasible, whose usage results in a most even distribution of scheduled lessons in the timetables of the entities involved. It is another known rule that the more evenly distributed the available time slots, or equivalently the scheduled lessons, are for a timetable, the less likely shall clashes arise. FIND-BEST is intended for preempting clashes.

An AUTO-PUT function is available which performs FIND-BEST for the current list of selected unscheduled lessons in the browser window. However, as lessons are inserted by AUTO-PUT, the priority function values may change. The user may set a count of insertions for updating the values of remaining unscheduled lessons. He may choose a small count for more effective execution of the “difficult-lessons-first” heuristic, and a large count for more speedy execution of AUTO-PUT.

For user control, AUTO-PUT may also be interrupted any time by a mouse click. It also halts at any unscheduled lesson for which no “best” slots are found, whence the user may analyze the clash, resolve it himself, or call up the conflict resolver to help. (We could have let AUTO-PUT attempt to move some scheduled lessons to make way for it but we do not. Movement of lessons already scheduled will render the system less predictable, and may even undo deliberate choices the user previously made. This is another case of the concern for usability and control being over that of power.)

Resolving Clashes

TESS incorporates a conflict resolver to assist the user in finding feasible resolving moves (FRMs) when no “best” slots are found for an unscheduled lesson (Lam, 1993). An FRM is a sequence of displacement of some scheduled lessons that creates a feasible slot for the unscheduled lesson.

There are many ways to construct a conflict resolver in terms of its behaviour and the algorithm used. We again aim for one that is simple and with good utility.

First, the TESS resolver is designed to be called from a timetable window of a particular class, teacher or location, and finds FRMs that involves only scheduled lessons therein. A useful consequence is that FRMs found may be visualized in the same window for easy analysis and comparison.

Second, the user may specify whether the FRMs may or may not pass over some soft constraints, namely, maximum teaching load per day for teachers and balanced subject distribution for classes.

Third the user may also specify the maximum “depth” of search, or equivalent, the maximum length of FRMs returned.

Fourth, scheduled lessons may be toggled so that only FRMs that do not involve them will be returned.

The TESS resolver uses a brute-force breadth-first search to look for FRMs one by one. The user may also specify the number of FRMs returned. The resolver terminates when one of the following happens: (1) the specified number of FRMs is found; (2) the maximum depth of search is exhausted; (3) the user interrupts. FRMs found are then displayed for the user to choose to use.

We do realize that experienced practitioners sometimes use FRMs that involve scheduled lessons not all contained in one class, teacher or location timetable. Our hope is that a proper use of TESS preempts serious clashes and FRMs found by the TESS resolver should suffice for those that do arise. In fact, some experienced TESS users manage to develop effective tactics in using the resolver in conjunction with many FIND functions that show the “degrees of freedom” of scheduled lessons and empty slots across multiple timetables. Such is possible with the toolbox approach and the event-driven user interface of the TESS environment. We also hope that useful tactics that experienced practitioners have developed in the past may also be adapted effectively to be carried out in TESS.

4 IMPLEMENTATION EXPERIENCE IN HONG KONG

TESS has been selected and subsequently developed to be the Timetabling Application of the School Administration and Management System (SAMS) for all government and aided schools funded through the Education Department in Hong Kong.

TESS was essentially a byproduct of an applied research project (Lam, 1993; Wong, 1992) developed for field use. We have essentially revamped the input and output interfaces for user-friendliness and coherence with established SAMS standards. Much design effort was also spent in developing a Planner Module for planning class curriculum, staff deployment and lesson accommodation. As this tends to be an iterative process, especially when planning for staff deployment, a spreadsheet like WYSIWYG interface is incorporated.

However, the core functions that actually construct the timetables are relatively untouched for some good reasons. First, we believe and the test users seem to agree that the core functions as they stand already provide very good utility. Second, proposed modifications at such an early stage come from sophisticated users that tend to trade usability for power. For instance, some have suggested to enhance the TESS resolver to find FRMs across multiple timetables. This however would be undesirable to less sophisticated users who are the majority of our target users in the 1000+ Hong Kong schools.

5 CONCLUSION

We have received very positive responses from our client and the first batch of school users. There are criticisms of course but most concern the I/O interface rather than the overall system approach.

As the Timetabling Application of SAMS, TESS will be further refined and progressively distributed to all schools in Hong Kong in a few years time. Being deployed at such a scale, we anticipate TESS to become a standardization vehicle de facto and encourage better sharing of expert skills in timetabling. Further, we hope TESS users shall become the new generation of school timetabling experts who develop

habits and tactics no less effective than current ones while being more portable and easier to acquire.

TESS is representative of the decision support system (DSS) approach to designing systems for domain experts to handle ill-structured problems (Sprague and Watson, 1993). The DSS approach relies on a high level of interactivity and usability that is potentially costly. It is only the cost-effective PC technology available in recent years that puts a system like TESS within reach of schools. Beyond TESS, we anticipate DSSs of other kinds to assist other educational management and planning functions. The classical view sees DSS as the next step after MIS (Sprague and Watson, 1993) and it points to a natural way forward for SAMS to deliver computerized support to Hong Kong schools.

6 REFERENCES

- Chahal, N. and de Werra, D. (1989) An interactive system for constructing timetables on a PC. *European Journal of Operational Research*, **40**, 32-37.
- Chinese University of Hong Kong (1995) *Timetabling Expert Support System - system manual*. Unpublished manual, Hong Kong.
- Education Department (1995) *School Administration and Management System*. Hong Kong Government.
- de Gans, O.B. (1981) A computer timetabling system for secondary schools in the Netherlands. *European Journal of Operational Research*, **7**, 175-182.
- Johnson, K. (1980) *Timetabling*, Hutchinson, London.
- Kernighan, B.W. and Plauger, P.J. (1976) *Software Tools*. Addison-Wesley, Reading, Mass., USA.
- Lam, H.M. (1993) *A conflict resolver for an interactive timetabling support system*. Unpublished B.Eng. Thesis, Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong.
- Sprague, R.H. and Watson, H.J. (1993) *Decision support systems: putting theory into practice*. Prentice Hall, Englewood Cliffs, N.J., USA.
- Wong, K.H. and Ng, W.Y. (1990) An interactive system for school timetabling, in *Proceedings of the International Conference on Systems Management'90*, Hong Kong, 307-313.
- Wong, K.H.(1992) *A decision support system for school timetabling*. Unpublished M.Phil. Thesis, The Chinese University of Hong Kong, Hong Kong.
- de Werra, D. (1985) An introduction to timetabling. *European Journal of Operational Research*, **19**, 151-162.
- Yoshikawa, M., Kaneko, K., Yamanouchi, T. and Watanabe, M. (1996) A constraint-based high school scheduling system. *IEEE Expert*, **(11)**, 63-72.

7 BIOGRAPHY

Wai-Yin Ng received his B.A. in engineering in 1985 and Ph.D. in Control engineering in 1989, both from the University of Cambridge, U.K. He has been a Lecturer of Information Engineering in the Chinese University of Hong Kong since 1988. Being a professed interdisciplinary researcher, he works on and supervises R&D projects of a broad range of subjects. His current research interests include decision making, interactive systems, distributed applications, 3D image analysis, image coding, computer-aided design, optimization, control engineering, and any interesting engineering problems he appreciates enough to work on.