The KTS High School Timetabling System

Jeffrey H. Kingston

School of Information Technologies,
The University of Sydney, NSW 2006, Australia
jeff@it.usyd.edu.au
http://www.it.usyd.edu.au/∼jeff

Abstract. KTS is a web-based high school timetabling system, freely accessible on the Internet. This paper is a general survey of KTS, including its data model, user interface, and solver. The solver uses operations research models in a polynomial-time heuristic framework to produce high quality solutions in a few seconds. Results are presented for six instances taken from Australian high schools.

1 Introduction

The problem of automatically constructing high school timetables has drawn the attention of researchers for many years. In the early days, *class-teacher timetabling* was the main focus [14], but it was never a realistic model of real-world problems. Progress on these has been slow. About ten years ago, a standard survey could find only a handful of papers reporting solvers for similar problems (including university course timetabling) in use in institutions [5]; and although firm data are hard to come by, the situation does not seem to have changed much since then.

Indeed, since that time the field seems to have been in decline; for example, the PATAT conferences of this period [1,2,3] contain only a few high school timetabling papers, and there are still no standard data sets on which researchers can compare results. Several recent papers, however, point to a renaissance; they attempt to solve real-world instances with such diverse methods as constraint programming [12], tabu search [8], and a hybrid approach [7]. These papers all relate to European high schools, which seem quite similar to the Australian high schools of this paper, the main differences being that European teachers are more likely to be preassigned to their classes, and are utilized less (around 50% of the time) than Australian teachers (around 75%), giving rise to compactness requirements not present in Australian problems.

This paper and the work it reports are motivated by a belief that, even if the technical problems can be overcome, software solvers will be widely used in high schools only if they are inexpensive and able to respond quickly to changing requirements. Such solvers would need to be available on-site under the control of the school's timetable planner, rather than off-site under the control of an expert (and hence expensive and busy) consultant.

E.K. Burke and H. Rudová (Eds.): PATAT 2006, LNCS 3867, pp. 308–323, 2007.

KTS is a web server for high school timetabling created by the author. Its web interface puts the system on the desk of the school's timetable planner, and its polynomial time heuristic solver delivers a very good timetable in a few seconds. Together these features support non-traditional requirements such as rapid evaluation of alternative scenarios and incorporation of late changes, as well as the traditional one of solving a fixed instance to near-optimality. The system is fully operational and available continuously on the Internet [10], but at the time of writing the process of gathering a user community has only just begun (60 accounts have been created, but only a few are active).

This paper is a general overview of the KTS system. Section 2 presents a detailed specification of the high school timetabling problem as defined by KTS. Section 3 describes the user interface. Section 4 describes the solver, and Section 5 presents results for six instances taken from Australian high schools.

2 Data Model

The KTS data model is object-oriented. It is described in this section, with a few minor omissions.

An *account object*, or just *account*, represents one user's account with the KTS system. Each account contains any number of *institutions*, representing educational institutions for which the user wishes to construct timetables. Each institution contains any number of *instances*, each representing that institution's timetabling problem for a particular year, or semester, etc.

Each instance contains a *time group* object, holding all information about time. KTS has a simple time model in which time is divided into individual *times* of equal duration, ordered chronologically, with each time optionally separated from the next by a break, which could be a meal break or the end of a day, etc. The full sequence of times is called the *cycle*.

A sequence of one or more times that follow each other chronologically and do not span a break is called a *time block*. Any set of times may be viewed as a set of time blocks, by grouping the times into blocks of maximal size. The sizes of these blocks, written as a sequence of integers, form the *block structure* of the set of times. For example, the set of times $\{Mon1, Mon2, Tue5, Tue6, Thu3\}$ presumably has block structure 2 2 1. The order in which the elements of a block structure are written does not matter; non-increasing order is used by convention. Meetings may specify that their times should have a particular block structure.

In addition to the instance's set of available times, the time group contains any number of *time subgroups*, which are subsets of the times, used when defining workload limits and *time conditions*. These latter place requirements on the sets of times assigned to meetings, and are either *limit conditions*, which limit the number of times from a given subgroup that a meeting may contain, for example limiting to 1 the number of undesirable times, or *spread conditions*, which require the time blocks assigned to a meeting to be spread evenly over a sequence of time subgroups, such as the days of the week.