

Teaching Assistant Scheduler Application

Lawrence King, Ajay Mallya, Prabhakar Gubbala, and Gopal Gupta

This application attempts to solve the problem of scheduling a large number of Teaching Assistants (TAs) for a large number of classes. The problem is inherently NP-complete, and any attempt to find an optimal solution quickly leads to combinatorial explosion.

During summer 2002, a model was developed that enabled the problem to be programmed in Prolog using Constraint Logic Programming over Finite Domains (CLPFD) as the computational engine. Data was collected and the model pre-processed in order to reduce search space. This technique was used in previous work by King and Gupta. Constraints were then applied and a solution was found. Although pre-processing was cumbersome and restricted due to the limitations of Prolog list processing, the application was successfully used to schedule Teaching Assistants for the Computer Science Department of UTD in Fall 2002 and Spring 2003 semesters.

During summer 2003, the project was extended in two ways. First, a front-end package was developed by Mallya and Gubbala, with King acting as project manager and Gupta acting as overall project coordinator. This web-based package automated data collection from instructors and TAs, enabling much more sophisticated assignment criteria to be applied to the model. This package provided automated format and data checks, and automated the process of sending emails to both instructors and TAs. These emails contained URL links to web-based forms. Instructors could specify skills required by their courses and request specific TAs. TAs could indicate skills, previous TA experience, and courses taken with a grade of A. The package provided sophisticated response processing and output files necessary for the scheduling engine.

The second extension was to expand the scheduling engine from Prolog to an application in Java and Prolog. This work was performed by King with Gupta again acting as overall project coordinator. The data analysis and pre-processing was programmed in Java. This enabled a much more sophisticated level of TA pre-assignment and, through the use of a Graphical User Interface (GUI), enabled the scheduling administrator to make choices and set parameters for both pre-processing and the Prolog CLPFD engine. After pre-processing, the partial solution is passed to the CLPFD engine where the final solution is found and returned to the Java package. The solution is presented to the scheduling administrator in a GUI that allows efficient post-processing of the Prolog-generated assignments. The Java engine is capable of both rapid pre-assignment and of determining the optimum starting point for the Prolog CLPFD. Then, through a continual, automated constraint-relaxation heuristic, a solution can be found which is much closer to optimal than previously realized with the Prolog engine alone.

Testing during Fall 2003 have shown the need for a much more efficient data checking and processing as well as other limitations that will be addressed as work continues.