Designing a Scheduling Competition[*]

Toby Walsh
NICTA and UNSW
Sydney, Australia
tw@cse.unsw.edu.au

Abstract

I discuss some important issues in the design of a competition for scheduling systems. There are many critical decisions to be made in designing such a competition including: the number and format of the tracks, the source and format of benchmarks, and the metrics and ranking system used to evaluate solvers.

Introduction

Given the perceived success of competitions in planning, theorem proving, propositional satisfiability and elsewhere, it is timely to consider a similar competition for scheduling systems. This discussion is inspired in part by my experiences as a judge of the CASC theorem proving competition, as well as a judge of the SAT competition.

Motivation

Why run a competition?

First, competitions appear to help advance research. The CASC and SAT competitions are perceived to have delivered clear returns to their research fields. SAT solvers in particular have advanced rapidly once a competition was started. Competitions permit low level development, which is needed to provide practical systems, to be recognized and rewarded. It is often difficult for such development to be recognized within conference paper tracks.

Second, competitions can provide rewards to their participants. In my own research institute, success in a competition is one of several factors considered by promotion panels alongside more traditional factors like publication lists. Competitions also can make research more personnaly rewarding for participants. Research is often a solitary endeavour, whilst competitions tend to be much more social.

Third, competitions can help to bring academic research closer to industry. Competitions may expose academics to more realistic problems. Competitions may also push academics to solve problems of a scale and messiness closer to that required by industry. On the other side, competitions may give industry a better appreciation of the strengths of new methods. They may therefore feel more confident in investing in the more promising new methods.

Despite all these benefits, competitions are not without their problems. First, competitions can sometimes hinder research. The competition may encourage small incremental local improvements. It may be difficult for radical new approaches to be proposed if they need several years of development to reach and eventually exceed the performance of existing methods. For this reason, it may pay to rest the competition every few years. Second, competitions are hard to design well. We want to ensure scientific progress is made, and not just algorithm tweaking. This requires clear objectives and careful design to meet these objectives. Third, competitions require considerable investment by the participants. It is important therefore to ensure the competition is designed to reward both the organizers and the competitors.

Some design issues

There are, it seems, strong arguments to hold a scheduling competition. However, the design of such a competition is not without problem. Some of the issues are common to competitions in other areas (e.g. how do we rank systems?). However, a few issues are more specific to scheduling (e.g. what types of scheduling problems should we consider?). The following is an non-exhaustive list of some of the design issues that need to be addressed to ensure a successful and useful scheduling competition.

Multiple tracks: Like theorem proving, scheduling is a broad field requiring a variety of solution methods. At CASC, there are separate tracks for different types of problems (e.g. Horn problems, and problems with equality). Theorem proving methods good on Horn problems may not be able to deal with equality effectively and vice versa. Similarly, a scheduling competition will require different tracks to deal with the different types of scheduling problems (e.g. pre-emptive, open shop, job shop, flow shop). Different tracks may also be desirable to deal with the different types of optimization criteria. For instance, a solver good at reducing makespan may not be good at reducing tardiness. Separate tracks permit different types of

[*] NICTA is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Research Council DCITA and ARC through Backing Australia's Ability and the ICT Centre of Excellence program. Thanks to Patrik Haslum and John Slaney for useful comments.

Copyright © 2007, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

solver to compete. In particular, complete methods which prove optimality should not be compared directly with incomplete methods that do not. Complete methods can of course still compete in the incomplete tracks but not vice versa. One danger is having too many tracks. This makes the competition more difficult to run. In addition, there may be too few participants in each track to make it competitive. In CASC, there is one combined track where problems can be of any form. This is the most prestigious track and has encouraged the development of versatile and robust solvers.

Problem format: It is important for the community to agree upon a common problem format. Given the diversity of scheduling problems, this needs to be extensible. As in the planning competition, initial competitions can be with very simple and generic scheduling problems. However, over time, the competition can evolve towards more realistic scheduling problems. The choice of a problem format may have long reaching effects. For example, as part of the DIMACS special year on combinatorial optimization in 1993, a problem format was put forwards in order to run an one-off competition on SAT solving. This format has became the standard input format for the community and is used by all SAT solvers today.

Benchmark library: The problems should be drawn from an associated benchmark library which is freely available. This enables competitors to prepare their systems. With CASC, problems are drawn from the TPTP library. New problems from this library are held back from the library in the months leading up to the competition so that the competition is on instances not previously seen by the competitors. The number of problems in TPTP has grown consistently over time. This helps prevent ceiling effects where systems are all doing close to optimal, as well as over-fitting of system parameters to the benchmark library. Another issue with a benchmark library is that the problems can easily become the default for experimental analysis in the field. Great care has to be taken that the problems are truly challenging. In addition, a standard format discourages valuable research on modelling problems, reformulation and related issues.

Solution correctness: Before the competition, systems needs to be "checked" for correctness. For incomplete systems, a variety of problems need to be solved and the solutions returned checked for feasibility. For complete systems, we need to test on problems with known optimal solutions to ensure that the correct optimal is returned. Any system which returns an incorrect answer can be eliminated from the competition. In the SAT competition, any such solver is permitted to continue but "hors concours". It helps greatly to have publically available code to test any solution. In this way, participants can quickly debug their solvers.

Time-outs: With a large number of competing systems, it may be impossible to give each a long time-out on every benchmark. A strategy adopted within the SAT community that has merit is a two round competition. In the first round, solvers are given a relatively short time-out (a few minutes at most), then only the top few solvers advance to the second round where there is a longer time-out to select the top rankings. The first round can be run before the conference, but the second round should be run during the conference (perhaps with a large real-time scoreboard in coffee breaks) to encourage interest.

Metrics: There need to be simple metrics to compare the different systems. For incomplete methods, this might be the quality of the solution found within the time-out. For complete methods, this might be the number of solutions proved optimal, tie-breaking on the quality of the best solution for problems not solved to optimality. CASC experimented with a variety of complex weighting schemes. However, the winner is typically the systems proving the most theorems within the time-out. Another important metric is the baseline provided by last year's winner. In fact, CASC only awards a prize if last year's winner is beaten. Ultimately, we want to compare the time *v.* solution quality curves of each solver. One possibility is to identify any solver which is not Pareto dominated. However, this may give too many "winners".

Ranking: Ultimately a competition needs to compute a winner. Indeed, we usually want to declare at least the top three systems. This can be seen as a ranking problem. We might therefore borrow methods for social choice like the Borda rule to compute a ranking based on the ranking of systems on individual problems.

Judges: Even if the competition rules are written very carefully, there are likely to be situations arising within the competition that casue dispute. Participants can also be very competitive. For this reason, it helps to have a couple of independent judges, preferably senior members of the research community without a direct interest in the result.

Incentives: Few incentives are needed to get a strong field of competitors at the CASC and SAT competitions. However, incentives are useful for the organizers. Running a competition is a very time-consuming business. CASC has traditionally reported results in an annual journal paper. This provides some return to the organizers on their investment of time. In additional, the organizers present a summary of the results in a conference talk.

Conclusions

All in all, I believe a well designed scheduling competition would profit the ICAPS research community greatly. It requires considerable effort from a small number of individuals. However, it is likely to be help push the field forward, towards more versatile and robust systems, and ultimately towards real economic return in deployed systems.