Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling

Solving the Traveling Tournament Problem with Iterative-Deepening A*

David C. Uthus
NRC/NRL Postdoctoral Fellow
Washington, DC 20375, USA
david.uthus.ctr@nrl.navy.mil

Patricia J. Riddle
University of Auckland
Private Bag 92019
Auckland, New Zealand
pat@cs.auckland.ac.nz

Hans W. Guesgen
Massey University
Private Bag 11222
Palmerston North, New Zealand
h.w.guesgen@massey.ac.nz

Abstract

We give an overview of our journal paper on applying iterative-deepening A* to the traveling tournament problem, a combinatorial optimization problem from the sports scheduling literature. This approach involved combining past ideas and creating new ideas to help reduce node expansion. This resulted in a state-of-the-art approach for optimally solving instances of the traveling tournament problem. It was the first approach to solve the classic NL10 and CIRC10 instances, which had not been solved since the problem's introduction.

Introduction

In this paper, we give an overview of our journal paper (Uthus, Riddle, and Guesgen 2012). This investigation involved solving problem instances of the traveling tournament problem (TTP) (Easton, Nemhauser, and Trick 2001) using a modified iterative-deepening A* (IDA*) approach called TIDA*. TIDA* incorporated both old ideas and newly-created ideas for reducing node expansion. The results of this work was a state-of-the-art approach for solving instances of the TTP. It was the first to solve the classic NL10 and CIRC10 problem instances along with the new instance of GALAXY10. Additionally, this paper introduced a new problem set of instances for the TTP: GALAXY.

Traveling Tournament Problem

The TTP is a combinatorial optimization problem from the sports scheduling literature. It abstracts some of the important features of scheduling for Major League Baseball's regular season. It is a difficult problem to solve optimally, with only the smaller instances having been solved to date. Even finding good, feasible solutions require lengthy amounts of computational time. A website is maintained by Michael Trick, keeping track of the best solutions so far (Trick 2010).

The TTP involves n teams, with n being even. The goal is to create a double round-robin tournament of minimal travel distance, with each team playing the other teams twice, once at home and once away. Distance is calculated only when a team is traveling – there is no distance cost when a team plays consecutive games at home. There are two constraints: teams cannot play one another in back-to-back games, and a team cannot play more than three consecutive games at home or away.

The TTP was first introduced with two problem sets: NL and CIRC. NL used the distances of the National League of Major League Baseball. CIRC assumed all teams were located on a circle, with an arc between neighboring pairs of teams. Distance between two teams was then calculated as the minimal number of arcs a team must travel through to get to the other team. Later problem sets included NFL (using distances of the teams in the National Football League), CON (all distances are constant between all teams), and SUPER (using distances of the Super 14 Rugby League).

A contribution of this paper was the creation of a new problem set, GALAXY. Unlike previous problem sets where distances were on a 2D-coordinate plane, GALAXY used a 3D-coordinate plane, in this case the distances (measured as light years) between exoplanets and Earth.

Our Approach

We solved the TTP using an IDA* approach. TIDA* incorporated concepts such as subtrees, parallelization, node ordering, disjoint pattern databases, pattern matching for constraint propagation, and symmetry breaking. Additionally, we developed new techniques to further improve performance. General, problem-independent ideas to reduce node expansion included forced deepening, elite paths, and subtree skipping. Ideas specific to the TTP include team reordering and additional symmetry breaking.

Forced deepening addressed the problem of IDA* going through too many iterations when applied to problem instances of real distances between teams (e.g., NL, SUPER, GALAXY). This is due to IDA* expanding only a small number of new nodes each iteration, unlike more classic applications where an exponential number of nodes are expanded for each iteration. Forced deepening overcame this by forcing IDA* to find f-values at a deeper depth for each iteration, creating an upper limit on the number of iterations.

Elite paths addressed some new problems created from applying forced deepening. Forced deepening would cause some extra nodes to be expanded that were not normally seen during a traditional application of IDA*. Elite paths solved this by storing the best partial solution for each iteration, and then traversing that partial solution first during

Instance	B&P	DFS*	TIDA*
NL6	509	0.98	0.8
NL8	43 321	262.42	195.0
CIRC6	10 789	2.05	0.07
CIRC8	300 087	337.09	22.25
SUPER6	–	0.27	0.49
SUPER8	–	361.20	687.24

Table 1: Comparison of TIDA* with DFS* and branch-and-price (B&P). All times are in seconds. Also, B&P did not solve CIRC8 optimally, it only found a lower bound.

the next iteration. Additionally, elite paths also reduced the number of nodes expanded during the final iteration, as the optimal solution will be found in one of the children of the elite path.

Subtree skipping built upon concepts of subtree forests (which was a combination of past ideas). Subtree skipping allowed some of the search space to be skipped each iteration in some specific cases. This could be done without invalidating IDA*'s guarantee of optimality.

The final two new ideas, both specific to the TTP were team reordering and additional symmetry breaking. Team reordering was a simple concept that reordered the teams so that teams were selected (when building a solution) in respect to the distance between them and the other teams. This was done to allow teams with a greater impact on the total distance of the schedule to be tried first. Additional symmetry breaking was also introduced specific to the CIRC instances (for when subtree forests are applied). This helped to further reduce node expansion for those set of instances.

Results

We applied TIDA* to the TTP. Our paper also gave an overview of the effectiveness of the various introduced ideas (e.g., force deepening, elite paths, and subtree skipping), showing how they help reduce the number of nodes expanded. Forced deepening had the greatest impact when applied to real-distance problem instances, reducing the number of nodes needing to be expanded between 56%-99%. Elite paths and subtree skipping helped reduce node expansion for all problem instance types, though not as a great of reduction as seen with forced deepening.

We then compared TIDA* with two recent approaches, a branch-and-price approach (Irnich 2010) and a DFS* approach (Uthus, Riddle, and Guesgen 2009). As seen in Table 1, TIDA* was able to find optimal solutions faster for most instances of 6 and 8 teams.

Also, as mentioned earlier, TIDA* was the first to solve classic instances NL10 and CIRC10. This involved a significant amount of computation, running TIDA* on 120 processors in parallel. TIDA* was also able to solve instances of GALAXY4-10.

Conclusions

Our journal paper presents a new IDA*-based approach for solving instances to optimality for the TTP. TIDA* incorporated new ideas of forced deepening, elite paths, and subtree skipping along with past ideas, resulting in an approach exhibiting state-of-the-art performance. Most important, it was the first to solve all 10-team instances, including classic instances of NL10 and CIRC10. Last, we also introduced a new problem set for the TTP.

References

Easton, K.; Nemhauser, G.; and Trick, M. 2001. The traveling tournament problem description and benchmarks. In Walsh, T., ed., *In Proceedings of Principles and Practice of Constraint Programming*, volume 2239 of *Lecture Notes in Computer Science*, 580–584. Springer Berlin Heidelberg.

Irnich, S. 2010. A new branch-and-price algorithm for the traveling tournament problem. *European Journal of Operational Research* 204(2):218 – 228.

Trick, M. 2010. Challenge traveling tournament problems. http://mat.gsia.cmu.edu/TOURN/.

Uthus, D. C.; Riddle, P. J.; and Guesgen, H. W. 2009. DFS* and the traveling tournament problem. In Hoeve, W.-J., and Hooker, J., eds., *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 5547 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 279–293.

Uthus, D. C.; Riddle, P. J.; and Guesgen, H. W. 2012. Solving the traveling tournament problem with iterative-deepening A*. *Journal of Scheduling* 15(5):601–614. doi: 10.1007/s10951-011-0237-x.