DFS* and the Traveling Tournament Problem

David C. Uthus[1], Patricia J. Riddle[1], and Hans W. Guesgen[2]

[1] University of Auckland, Auckland, New Zealand
[2] Massey University, Palmerston North, New Zealand
{dave,pat}@cs.auckland.ac.nz, h.w.guesgen@massey.ac.nz

Abstract. Our paper presents a new exact method to solve the traveling tournament problem. More precisely, we apply DFS* to this problem and improve its performance by keeping the expensive heuristic estimates in memory to help greatly cut down the computational time needed. We further improve the performance by exploiting a symmetry property found in the traveling tournament problem. Our results show that our approach is one of the top performing approaches for this problem. It is able to find known optimal solutions in a much smaller amount of computational time than past approaches, to find a new optimal solution, and to improve the lower bounds of larger problem instances which do not have known optimal solutions. As a final contribution, we also introduce a new set of problem instances to diversify the available instance sets for the traveling tournament problem.

1 Introduction

The traveling tournament problem (TTP)[3] emerged from the difficulties of scheduling real-world sports leagues. It simplified many of the constraints and requirements found in sports leagues to a problem manageable for theoretical research. Despite this simplification, it has been found to be a difficult problem: solutions have been proven optimal for only the smallest few problem instances. The difficulty lies in its unusual structure (round robin tournaments) and its feasibility constraints.

We present a new approach to the TTP by applying the search algorithm DFS*[9]. We are able to improve its performance for this application by using a new idea of storing expensive heuristic estimates in memory to greatly reduce the running time. In addition, we exploit a symmetry property found in the TTP, in which all schedules are symmetrically equivalent to one other schedule[5]. Our results show that this approach is one of the best performing approaches for the TTP, finding previously known optimal solutions in a fraction of the time of other approaches while being able to find both a new optimal solution and new lower bounds for unsolved problem instances. We also introduce a new set of problem instances for the TTP. This new set is derived from the rugby union league Super 14, which is composed of teams from Australia, New Zealand, and South Africa.

W.-J. van Hoeve and J.N. Hooker (Eds.): CPAIOR 2009, LNCS 5547, pp. 279–293, 2009.

2 Traveling Tournament Problem

The TTP is a sports scheduling combinatorial optimization problem. It involves n teams, n being even, and takes in an $n \times n$ matrix of distances between the teams. The problem consists of a double round robin tournament with $2 \cdot (n-1)$ rounds. The teams must play once during each round and are required to play every other team twice, once at home and once away. The objective is to create a tournament such that the total summed travel distance amongst all of the teams is minimized. The travel distance, which is calculated individually for each team, is the total distance a team must travel between locations during the tournament. All teams start at home prior to the first round and end at home after the final round. There is no travel cost for playing consecutive games at home.

The TTP has two feasibility constraints. The first is the *no_repeat* constraint. This forbids a team from playing the same team in consecutive rounds. The second constraint is the *at_most* constraint, which restricts a team to being able to play at most three games consecutively at home or away.

There are four problem sets for the TTP[7]. The NL set and the larger NFL set are derived from Major League Baseball and the National Football League respectively. Both of these sets use real distances between the cities where the teams are located. The other two problem sets use artificial distances. The CIRC set has all the teams placed on a circle, and the distance between two teams is the minimal number of teams a team has to go through to get to the other team. The CON set has all distances set to 1, which changes the problem to minimizing the number of trips[7].

In this paper, we focus our work on the NL and CIRC sets. We do not work with the NFL set since the size of the instances are too large for us to work with while the CON set has been solved for all instances except for 20 teams. So far, only the smallest few problem instances for the NL and CIRC sets have been solved to optimality. These are NL4, NL6, and NL8 for the NL set and CIRC4 and CIRC6 for the CIRC set[7]. For larger instances, only lower and upper bounds of the optimal solutions have been found. Some of the various techniques that have been used to find optimal solutions are integer and constraint programming[4], Lagrangian relaxation and constraint programming[1], independent lower bound estimations[8], and branch-and-price with column generation[5].

3 DFS*

We are taking a new approach to the TTP by applying DFS*. DFS* is an algorithm which combines IDA* with depth-first branch-and-bound (DFB&B) search. It goes through an iterative process like IDA*, starting with a small upper bound and increasing it after every iteration. However, it differs in that it uses DFB&B for the final iteration. This is due to the difference in how it creates the new upper bound thresholds. In IDA*, the new upper bound after every iteration is the minimal lower bound that exceeded the upper bound of the previous iteration. When the first solution is found, it knows that this solution is