



<http://researchspace.auckland.ac.nz>

ResearchSpace@Auckland

Copyright Statement

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

This thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author's right to be identified as the author of this thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from their thesis.

To request permissions please use the Feedback form on our webpage.

<http://researchspace.auckland.ac.nz/feedback>

General copyright and disclaimer

In addition to the above conditions, authors give their consent for the digital copy of their work to be used subject to the conditions specified on the [Library Thesis Consent Form](#) and [Deposit Licence](#).

Note : Masters Theses

The digital copy of a masters thesis is as submitted for examination and contains no corrections. The print copy, usually available in the University Library, may contain corrections made by hand, which have been requested by the supervisor.

Sports Scheduling: An Artificial Intelligence Approach

David C. Uthus

A thesis submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science

The University of Auckland

July 2010

ABSTRACT

This thesis looks at the Traveling Tournament Problem (TTP) from the sports scheduling literature. It presents two approaches to this problem: a metaheuristic Ant Colony Optimization (ACO) approach to find good solutions in a reasonable time frame and a heuristic search Iterative-Deepening-A* (IDA*) approach to find optimal solutions.

The first approach combines ACO with constraint processing techniques in order to handle the hard constraints of the TTP. The key component is creating a framework which uses forward-checking and conflict-directed backjumping to handle the constraints while using ACO for choosing the values. This is further improved by introducing new ideas of unsafe backjumping and pattern matching for constraint propagation while incorporating an old concept of ant restarts. This approach has been found to improve on past ACO approaches to the TTP and showed results which are more competitive with state-of-the-art metaheuristic approaches.

The second approach presents a parallel version of IDA*, combining past concepts of tree decomposition and node ordering with a new idea of subtree skipping. This new idea allows for parts of the search tree to be skipped for some iterations while still guaranteeing optimality for the final solution that is found. Two additional ideas are presented. The first, called forced deepening, helps to reduce node expansion when applying IDA*-like algorithms on real-world distance problems. The second, called elite paths, helps to both improve the performance of forced deepening while also allowing for the optimal solution to be found faster during the final iteration of IDA*. The results of applying this new approach to the TTP shows that it is state-of-the-art, finding known optimal solutions in a fraction of the time of past approaches and finding new optimal solutions to some unsolved problem instances.

Acknowledgements

I thank my supervisors Patricia Riddle and Hans Guesgen for all the help and guidance they have given me throughout these years of research. Thank you both so much for your time, advice, great ideas, and your friendship. I will greatly miss working with you both.

I also thank ScienceIT, who have helped greatly providing the hardware and software needed to make possible the experiments for this work.

I give thanks to Michael Trick, who helped get me started down the road of sports scheduling, and more importantly, the traveling tournament problem.

In addition, I thank Brad Clement and Michael Sims of NASA, who hosted my talks at their respective facilities, which allowed me to further refine my ideas.

Finally, I thank my wife, Emi, who has helped me in so many ways to become a better thinker, questioner, and scientist in the end.

Contents

Acknowledgements	v
1 Introduction	1
1.1 Contributions	2
1.2 Organization	3
2 The Traveling Tournament Problem	5
2.1 Round Robin Tournament Nomenclature	5
2.2 Traveling Tournament Problem	7
2.2.1 Problem Description	7
2.2.2 Problem Sets	9
2.3 Literary Review of Past Work	11
2.3.1 Artificial Intelligence and Metaheuristics	11
2.3.2 Operations Research, Exact Algorithms, and Lower Bounds .	16
3 Ant Colony Optimization	21
3.1 Ant Colony Optimization Metaheuristic	22
3.1.1 Colony of Ants	23
3.1.2 Pheromone	23
3.1.3 Solution Construction	25
3.1.4 Local Search	26
3.1.5 Algorithms	27
3.2 Constraint Processing	29

3.2.1	Looking Ahead	29
3.2.2	Looking Back	30
3.2.3	Forward Checking and Conflicted-Directed Backjumping	31
3.3	Integrating ACO with Forward Checking and Conflicted-Directed Back- jumping	31
3.3.1	Relation To Past Work	32
3.3.2	AFC Description	32
3.3.3	Value Selection	34
3.3.4	Unsafe Backjumping	35
3.3.5	Ant Restarts	36
4	Ant Colony Optimization and the Traveling Tournament Problem	39
4.1	Applying AFC to the TTP	39
4.1.1	Variables and Values	40
4.1.2	Pheromone	41
4.1.3	Local Search	42
4.2	Pattern Matching	43
4.2.1	Creating Patterns	44
4.2.2	Using Patterns	48
5	Ant Colony Optimization Results	51
5.1	AFC Components and Configuration	52
5.1.1	AFC Components	52
5.1.2	ACS's and AS's Rules	55
5.1.3	Pheromone Update Schedule	58
5.2	Comparison With Past Approaches	59
5.2.1	Past ACO Approaches	59
5.2.2	Past TS and SA Approaches	61
5.3	Limits of the Results	61

5.3.1	Pheromone Representation	61
5.3.2	Constraint Propagation and Pattern Matching	63
5.3.3	Parameter Optimization	63
6	Concurrent Iterative-Deepening-A*, Forced Deepening, and Elite Paths	65
6.1	Heuristic Search, A*, and IDA*	66
6.1.1	A*	66
6.1.2	Iterative-Deepening-A*	67
6.1.3	Parallel Approaches to Iterative-Deepening-A*	68
6.2	Concurrent IDA*	71
6.2.1	Subtrees and Subtree Forest	71
6.2.2	Parallelization	74
6.3	Forced Deepening and Elite Paths	77
6.3.1	Forced Deepening	77
6.3.2	Elite Paths	81
6.3.3	Combining CIDA* with FD and EP	81
6.4	Initial Analysis with the Traveling Salesman Problem	83
6.4.1	CIDA* Applied to the TSP	83
6.4.2	Subtree Forest	83
6.4.3	Forced Deepening and λ	87
6.4.4	Summary of Initial Results	88
7	Concurrent Iterative-Deepening-A* and the Traveling Tournament Problem	89
7.1	Depth-First Search	89
7.1.1	Choosing the Second Team	92
7.2	Forced Deepening and Subtree Forest	94
7.3	Team Reordering	95

7.4	Heuristic Estimate Calculations, Disjoint Pattern Databases, and Team Cache	96
7.4.1	Calculating Heuristic Estimates	96
7.4.2	Disjoint Pattern Databases	98
7.4.3	Team Cache	102
7.5	Symmetry Breaking	103
7.5.1	Reflective Symmetry	103
7.5.2	Rotational Symmetry	105
8	Concurrent Iterative-Deepening-A* Results	107
8.1	CIDA* and the TTP	108
8.1.1	IDA*, CIDA*, FD, EP, and the TTP	108
8.1.2	Length of Forced Deepening's λ	110
8.1.3	Subtrees and Load Balancing	111
8.1.4	Disjoint Pattern Databases and Team Cache	114
8.1.5	Team Ordering and Symmetry Breaking	115
8.2	CIDA* and the Final Results	117
8.2.1	Comparison with Branch-And-Price	117
8.2.2	Overall Results	117
8.3	Summary of Results	120
9	Conclusions	121
9.1	Summary	121
9.2	Future Work	122
	Bibliography	125

List of Tables

5.1	Comparison of solution quality using safe backjumping against unsafe backjumping, both using ant restarts and pattern matching.	54
5.2	Comparison of time needed to construct solutions for different values of ant restart's parameter b	55
5.3	Comparison of solution quality using AS's and ACS's rules for choosing values, with and with heuristic information.	57
5.4	Comparison of pheromone update schedules.	59
5.5	Comparison of AFC-TTP with previous ACO approaches.	60
5.6	Comparison of AFC-TTP with CNTS and TTSA.	62
8.1	The time needed, in seconds, to solve 8-team instances when looking at the depth of subtrees and its impact on problems in parallel. . . .	113
8.2	The time needed, in seconds, to solve 8-team instances when comparing number of processors used.	114
8.3	Comparison of CIDA* with Irnich's branch-and-price algorithm. . . .	118
8.4	Timings, in seconds, to solve instances to optimality with a single processor and with four processors in a shared-memory configuration. . .	118
8.5	Wall time in seconds for solving instances with 120 CPUs in parallel. . .	119

List of Figures

2.1	Example double round robin tournament schedule with even number of teams.	6
5.1	Comparison of safe and unsafe backjumping	53
5.2	Comparison of using AS's and ACS's rules.	58
6.1	A partial view of a tree with only the first two levels of the tree shown. A subtree s of depth 2, (A,E), is highlighted.	72
6.2	Comparing CIDA*+FD+EP, with and without subtrees skipping, against IDA*+FD+EP in terms of number of nodes expanded.	84
6.3	Comparing the effects of s^d with values of 2, 3, 4; compared to $s^d = 1$ in terms of number of nodes expanded.	85
6.4	Comparisons of parallelism against 1 processor for s^d values of 1, 2 and 3, with comparisons looking at the amount of time needed to find the optimal solution.	86
6.5	Comparisons of different values of increments for λ , compared to $\lambda = 1$ in terms of number of nodes expanded.	87
7.1	Symmetric schedules with equivalent total distances. Both are optimal for NL4.	103
7.2	Example of rotational schedules found in CIRC.	105

8.1	Comparison of nodes expanded using IDA* and CIDA*, both by themselves and with FD and EP. All comparisons are done against IDA* by itself.	109
8.2	Comparison of nodes expanded with and without subtree skipping for CIDA* with FD and EP. This is compared in relation with number of nodes expanded by IDA* with FD and EP.	110
8.3	Comparing different incremental values for λ for \mathcal{L} when using FD in terms of number of nodes expanded. All are compared against increment of 1.	111
8.4	Comparing different lengths of s^d , with comparisons against $s^d = 1$ in terms of number of nodes expanded.	112
8.5	Comparison of Symmetry Breaking and Team Ordering.	116