A Beam Search Approach to the Traveling Tournament Problem

Nikolaus Frohner, Bernhard Neumann, and Günther R. Raidl

Institute of Logic and Computation, TU Wien, Vienna, Austria
{nfrohner|raidl}@ac.tuwien.ac.at, e1634034@student.tuwien.ac.at

This extended abstract is derived from our full paper[1] to appear in [4][2].

Summary. The well-known traveling tournament problem as introduced by Easton, Nemhauser, and Trick [3] in 2001 is a hard optimization problem in which a double round robin sports league schedule has to be constructed while minimizing the total travel distance over all teams. The teams start and end their tours at their home venues, are only allowed to play a certain maximum number of games in a row at home or away, and must not play against each other in two consecutive rounds. The latter aspects introduce also a difficult feasibility aspect. We study a beam search approach based on a recursive state space formulation. We compare different state ordering heuristics for the beam search based on lower bounds derived by means of decision diagrams. Furthermore, we introduce a randomized beam search variant that adds Gaussian noise to the heuristic value of a node for diversifying the search in order to enable a simple yet effective parallelization. In our computational study, we use randomly generated instances to compare and tune algorithmic parameters and present final results on the classical National League and circular benchmark instances. Results show that this purely construction-based method provides mostly better solutions than existing ant-colony optimization and tabu search algorithms and it comes close to the leading simulated annealing based approaches without using any local search. For two circular benchmark instances we found new best solutions for which the last improvement was twelve years ago. The presented state space formulation and lower bound techniques could also be beneficial for exact methods like A* or DFS* and may be used to guide the randomized construction in ACO or GRASP approaches.

Results. We conducted all our experiments on Intel Xeon E5-2640 processors with 2.40 GHz and a memory limit of 32GB. We implemented our approach as a prototype in Python 3.7, being aware that an implementation in a compiled language would likely be substantially faster and would have a smaller memory footprint. Table 1 compares our randomized beam search variant with either lexicographic or random team ordering performed in parallel and independently on 30 cores with several state-of-the-art approaches on three difficult NL and CIRC instances[3]. Each beam search run was conducted with beam width $\beta =$

[1] https://www.ac.tuwien.ac.at/files/pub/frohner-19d.pdf
[2] http://www.evostar.org/2020/evocop/
[3] https://mat.tepper.cmu.edu/TOURN/

Table 1. Comparison of the final solution lengths of parallel randomized beam search using either lexicographic team ordering or random team ordering (RTO) with 30 independent runs each, parameters $\sigma_{\mathrm{rel}} = 0.001$, $\beta = 10^5$, and the CVRPH [4, Sec. 6] lower bound function (RBS-CVRPH) with the reported solution lengths of ant-colony optimization (AFC-TTP) [5], composite-neighborhood tabu search (CNTS) [2], simulated annealing (TTSA) [1], and population-based simulated annealing (PBSA) [6], where the latter is either used from scratch (PBSAFS) or starting from an already high quality solution (PBSAHQ) provided by a TTSA run. †New best feasible solutions.

inst	RBS-CVRPH		RBS-CVRPH-RTO		AFC-TTP		CNTS		TTSA		PBSAFS		PBSAHQ	
	min	mean	min	mean	min	mean	min	mean	min	mean	min	mean	min	mean
nl12	112680	113594.6	112791	113581.5	112521	114427.4	113729	114880.6	112800	113853.0	**110729**	**112064.0**	n/a	n/a
nl14	192625	198912.6	196507	199894.8	195627	197656.6	194807	197284.2	190368	192931.9	**188728**	190704.6	**188728**	**188728.0**
nl16	266736	271367.1	265800	270925.9	280211	283637.4	275296	279465.8	267194	275015.9	**261687**	265482.1	262343	**264516.4**
circ12	410	415.7	410	414.6	430	436.0	438	440.4	n/a	n/a	**404**	418.2	408	**414.8**
circ14	632	641.0	**630**†	**640.7**	674	692.8	686	694.4	n/a	n/a	640	654.8	632	645.2
circ16	918	933.8	**910**†	931.6	1034	1039.6	1016	1030.0	n/a	n/a	958	971.8	916	**917.8**
circ18	1300	1322.0	1296	1320.4	1486	1494.8	1426	1440.8	n/a	n/a	1350	1371.6	**1294**	**1307.0**

10^5 and randomization parameter $\sigma_{\mathrm{rel}} = 0.001$ resulting in equally gentle noise applied to the f-values of the states in every layer. Runtimes per run go up to 30 hours for the largest instances. The table shows minimum and mean values for solution lengths of finally best solutions. We observe that we can compete well with the other mainly constructive approach "ant colony optimization with forward checking and conflict-directed backjumping" (AFC-TTP) from [5] and the composite-neighborhood tabu search (CNTS) from [2] on the NL instances and obtain better results than these for the CIRC instances, without hybridizing with a final local search. For CIRC instances we can also obtain similar results to population-based simulated annealing from scratch (PBSAFS) from [6], which uses parallel simulated annealing. For the circular instances with 14 and 16 teams, we found new best feasible solutions, already incorporated in Michael Trick's TTP web page[4]. The strongest results overall for NL and CIRC are provided by simulated annealing (TTSA) from [1] and its parallel variant PBSA from [6].

Future Work. A reimplementation in a compiled language is desirable to tackle even higher beam widths. So far we did not consider any local search, but a natural extension would be to try to further improve a number of best solutions provided by our beam search by local search. To tackle instances with more than 18 teams with lower bound guidance, an interesting direction could be to use relaxed decision diagram for the bound pre-calulcations, in order to keep the memory and computational demand reasonably bounded.

[4] `https://mat.tepper.cmu.edu/TOURN/`

References

1. Anagnostopoulos, A., Michel, L., Van Hentenryck, P., Vergados, Y.: A simulated annealing approach to the traveling tournament problem. Journal of Scheduling **9**(2), 177–193 (2006)
2. Di Gaspero, L., Schaerf, A.: A composite-neighborhood tabu search approach to the traveling tournament problem. Journal of Heuristics **13**(2), 189–207 (2007)
3. Easton, K., Nemhauser, G., Trick, M.: The traveling tournament problem description and benchmarks. In: International Conference on Principles and Practice of Constraint Programming. LNCS, vol. 2239, pp. 580–584. Springer (2001)
4. Frohner, N., Neumann, B., Raidl, G.R.: A beam search approach to the traveling tournament problem. In: Paquete, L., Zarges, C. (eds.) Evolutionary Computation in Combinatorial Optimization – 20th European Conference, EvoCOP 2020, Held as Part of EvoStar 2020. LNCS, Springer, Sevilla, Spain (2020), `https://www.ac.tuwien.ac.at/files/pub/frohner-19d.pdf`, to appear
5. Uthus, D.C., Riddle, P.J., Guesgen, H.W.: An ant colony optimization approach to the traveling tournament problem. In: Proceedings of the 11th Annual conference on Genetic and evolutionary computation. pp. 81–88. ACM (2009)
6. Van Hentenryck, P., Vergados, Y.: Population-based simulated annealing for traveling tournaments. In: Proceedings of the 22nd National Conference on Artificial Intelligence. pp. 267–262. No. 1, MIT Press (2007)