



Mainz School of Management and Economics
Discussion Paper Series

A Note on Symmetry Reduction for Circular Traveling Tournament Problems

Timo Gschwind and Stefan Irnich

April 2010

Discussion paper number 1006

Contact details

Timo Gschwind
Chair of Logistics Management
Johannes Gutenberg-Universität Mainz
Jakob Welder-Weg 9
55128 Mainz
Germany

gschwind@uni-mainz.de

Stefan Irnich
Chair of Logistics Management
Johannes Gutenberg-Universität Mainz
Jakob Welder-Weg 9
55128 Mainz
Germany

irnich@uni-mainz.de

All discussion papers can be downloaded from <http://wiwi.uni-mainz.de/DP>

A Note on Symmetry Reduction for Circular Traveling Tournament Problems

Timo Gschwind^{*,a}, Stefan Irnich^a

^a*Chair of Logistics Management, Johannes Gutenberg University Mainz,
Jakob-Welder-Weg 9, D-55128 Mainz, Germany.*

Abstract

The traveling tournament problem (TTP) consists of finding a distance-minimal double round-robin tournament where the number of consecutive breaks is bounded. Easton *et al.* (2001) introduced the so-called circular TTP instances, where venues of teams are located on a circle. The distance between neighboring venues is one, so that the distance between any pair of teams is the distance on the circle. It is empirically proved that these instances are very hard to solve due to the inherent symmetry. This note presents new ideas to cut off essentially identical parts of the solution space. Enumerative solution approaches, e.g. relying on branch-and-bound, benefit from this reduction. We exemplify this benefit by modifying the DFS* algorithm of Uthus *et al.* (2009) and show that speedups can approximate factor $4n$.

Key words: Timetabling, sports league scheduling, traveling tournament problem, circular instances, symmetry reduction

1. Introduction

The traveling tournament problem (TTP) consists of finding a distance-minimal double round-robin tournament where the number of consecutive breaks is bounded (cf. Easton *et al.*, 2001). Teams and venues are given by $T = \{1, 2, \dots, n\}$ for an even number $n \in 2\mathbb{N}$. Throughout the paper, we use indices $t, t' \in T$ for teams and indices $i, j \in T$ for their venues. Let $D = (d_{ij})$ for $i, j \in T$ be the distance between venues i and j . Any feasible solution to the TTP can be represented by tours $p^t = (i_1^t, \dots, i_{2(n-1)}^t)$ for all teams $t \in T$. Herein, $i_s^t \in T$ describes for each time slot $s \in \{1, 2, \dots, 2(n-1)\}$ at which venue team t plays. For $i_s^t = t$, team t is playing at *home*, otherwise t is playing *away*. Obviously, a solution is only feasible if tours are compatible, i.e. an away game $i_s^t = t' \neq t$ implies a home game $i_s^{t'} = t$. With the definitions $i_0^t := i_{2n-1}^t := t$ for all teams $t \in T$, the cost of a solution p is the overall distance traveled $\sum_{t \in T} \sum_{s=0}^{2(n-1)} d_{i_s^t, i_{s+1}^t}$. Additionally, no-repeater constraints (NRCs) require that a home game t against t' is not followed by the corresponding away game, where t' plays at home against t .

Easton *et al.* (2001) introduced the so-called circular TTP instances, where venues of the teams are located on a circle. The distance between neighboring venues is one so that the distance between two teams $i > j$ is $d_{ij} = \min\{i - j, n + j - i\}$. It is empirically proved that these instances are very hard to solve due to the inherent symmetry. The purpose of this note is, therefore, to fully explicate the symmetry and exploit it algorithmically.

This paper is structured as follows: In Section 2 we describe different kinds of symmetry and, if symmetry is reduced, we compare the worst-case number of solutions to consider in enumerative solution approaches for the TTP. Section 3 empirically tests the symmetry reduction methods on the branch-and-bound algorithm that constitutes the kernel of the DFS* algorithm presented in (Uthus *et al.*, 2009). Final conclusions are drawn in Section 4.

*Corresponding author.

Email addresses: gschwind@uni-mainz.de (Timo Gschwind), irnich@uni-mainz.de (Stefan Irnich)

2. Symmetry in Circular TTP Instances

It is possible to rotate the teams and venues of a feasible solution, i.e. replace t by $t+1$ (for $t < n$) and n by 1. The rotated solution is also feasible and has identical costs. Furthermore, reflections through opposite vertices or edges, as depicted in Figure 1, produce symmetric solutions with identical costs.

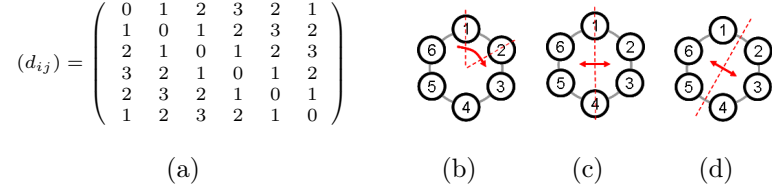


Figure 1: Circular Instance for $n = 6$; (a) Distances, (b) Rotation by $2\pi/6$, (c) and (d) Reflections through Vertices or Edges

Formally, the symmetry group of an n -sided regular polygon, which is known as the dihedral group D_n , operates on the teams/venues (for an introduction to group theory we refer to Aschbacher, 2000). D_n consists of n rotations by multiples of $2\pi/n$ (including the identity) and n reflections through opposite vertices and edges. We can exploit this operation in different ways, either by considering home-away assignments or oriented 1-factors. Both will be explained in the following.

Symmetry on Home-Away Assignments. A *home-away assignment (HAA)* partially characterizes a solution to the TTP. A HAA specifies, for each team and each time slot, whether the team plays at home or away. Since there are equally many teams playing at home and away in a time slot, the overall number of *slot HAAs* is $\binom{n}{n/2}$. D_n operates on the set of slot HAAs of a given slot s .

For $n = 6$, there are $\binom{6}{3} = 20$ possible slot HAAs. The slot HAAs that are symmetric to $(hhhaaa)$ are defined as the *orbit*. For instance, the orbit of $(hhhaaa)$ is $\{(hhhaaa), (ahhhaa), (aahhha), (aaahhh), (haaahh), (hhaaah)\}$. The three different orbits with lengths 2, 6, and 12 are depicted in Figure 2(b).

Due to symmetry in the circular instances, it suffices to consider one *representative* slot HAA from each orbit for a given time slot s . This is the key observation for symmetry reduction. Without loss of generality, we consider the first time slot $s = 1$ in the following. In branch-and-bound, symmetry reduction can be implemented as an a priori branching rule in the first level of the branch-and-bound tree. For $n = 4$, there are just two branches: The first branch can fix that teams 1 and 2 play at home, while the others play away; the second branch can fix that teams 2 and 4 play at home and 1 and 3 away, as depicted in Fig. 2(a). Also other enumerative solution approaches for circular TTP instances benefit from choosing representatives, e.g. the branch-and-price algorithm by Irnich (2010).

Moreover, Figure 2(c) compares standard branching with a priori branching on slot HAA representatives. In the worst case, all $\binom{n}{n/2}$ HAAs in slot 1 are checked in a standard branch-and-bound approach. Taking representatives creates just as many branches as orbits exist. Thus, the ratio of the two numbers is an (optimistic) estimate what can be gained from selecting representatives. This ratio is 3 for $n = 4$ and increases to approximately 18 for $n = 12$. Since the length of an orbit divides the order of the symmetry group, the ratio is here bounded (from above) by $|D_n| = 2n$. The numbers in the table presented in Figure 2(c) were computed with a straightforward enumeration procedure.

Symmetric 1-Factors. For each time slot s , a solution imposes a tournament, i.e. an oriented 1-factor of the complete graph with node set T (cf. de Werra, 1980). For instance, $f := \{(3, 2), (1, 6), (5, 4)\}$ means that team 3 plays at home against 2, 1 at home against 6, and 5 at home against 4. The dihedral group D_n naturally operates on oriented 1-factors. On f , D_n produces an orbit consisting of four elements: $\{f, \{(2, 1), (4, 3), (6, 5)\}, \{(1, 2), (3, 4), (5, 6)\}, \{(2, 3), (4, 5), (6, 1)\}\}$. As before, in an enumerative solution approach such as branch-and-bound, all elements of an orbit are equivalent. Therefore, it suffices to consider one *representative* oriented 1-factor for the slot $s = 1$.

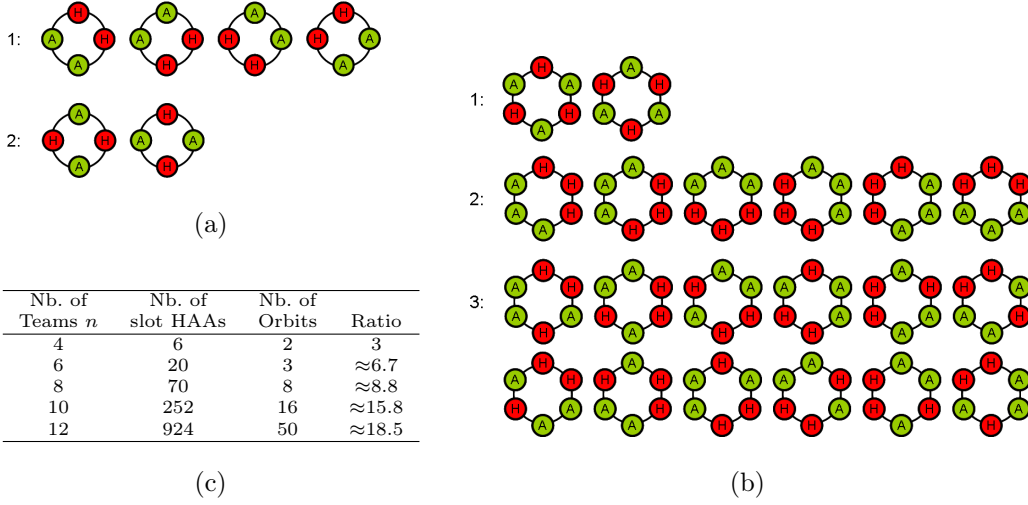


Figure 2: (a) Orbits of HAAs for $n = 4$, (b) for $n = 6$, and (c) Number of HAAs and Orbits

Figure 3(a) shows for $n = 4$ the three orbits with four elements each. Reducing symmetry in branch-and-bound for $n = 4$ can a priori branch on the three representatives, e.g. the first branch assigns the games $(1, 2), (4, 3)$ to slot 1, the second $(1, 2), (3, 4)$, and the third $(1, 3), (2, 4)$. The table in Figure 3(b) has the same meaning as the one presented for the slot HAAs. In particular, the ratio shows the reduction in the number of cases to be considered when reducing symmetry with 1-factors. Interestingly, the ratio here is larger compared to the ratio for slot HAAs. For $n \geq 10$, it is approximately equal to the order $2n$ of the symmetry group D_n .

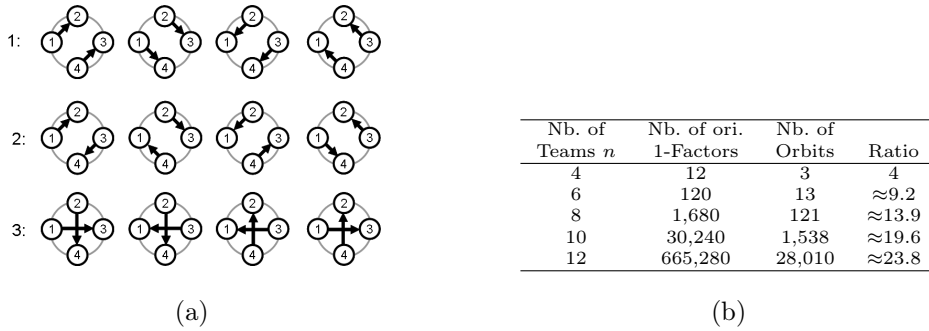
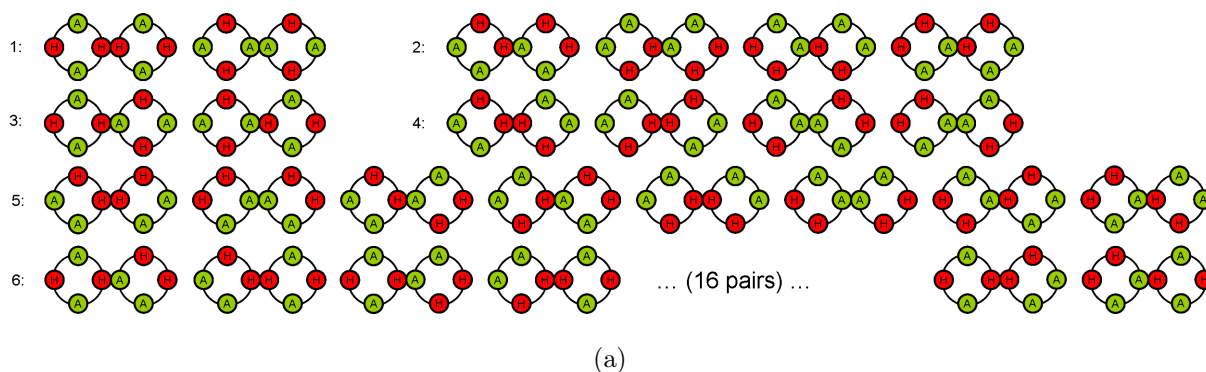


Figure 3: (a) Orbits of oriented 1-Factors for $n = 4$, (b) Number of oriented 1-Factors and Orbits

Reversing Tours. Another type of symmetry can be utilized if distances d_{ij} are symmetric. In this case, any solution with tours $p^t = (i_1^t, i_2^t, \dots, i_{2(n-1)}^t)$ of the teams $t \in T$ has a corresponding *reverse solution* with reverse tours $(i_{2(n-1)}^t, \dots, i_2^t, i_1^t)$ for all teams t . The symmetry group is the group $\mathbb{Z}_2 = \{id, rev\} \cong \{+1, -1\}$ consisting of the identity $id = +1$ and the reversal $rev = -1$. Several authors have already exploited this symmetry: In the branch-and-price algorithm of Irnich (2010), arcs are removed from the pricing network of one team (say team 1) in such a way that only one of the two symmetric tours can be generated. In their branch-and-bound approach, Uthus *et al.* (2009) require for team 1 that the first half of the schedule (i.e. games in slots 1 to $n - 1$) has more home than away games (or vice versa). This is checked when the schedule of team 1 is constructed round by round. In any case, the speedup that can be gained from exploiting reverse tours is obviously bounded by factor 2.

Symmetry on Pairs of HAAs. Both symmetry within time slot 1 (slot HAAs or oriented 1-factors) and symmetry induced by tour reversal can be combined. The symmetry group is $G_n = D_n \times \{+1, -1\}$ with $4n$ elements. Any group element $g = (d, \varepsilon) \in G$ operates on a solution with tours $(i_1^t, i_2^t, \dots, i_{2(n-1)}^t)$, $t \in T$ as follows: the resulting solution is given by the tours $(i_1^{d(t)}, i_2^{d(t)}, \dots, i_{2(n-1)}^{d(t)})$ for $\varepsilon = +1$ and $(i_{2(n-1)}^{d(t)}, \dots, i_2^{d(t)}, i_1^{d(t)})$ for $\varepsilon = -1$.

In order to exploit this symmetry, we consider two slot HAAs simultaneously. The HAAs must refer to corresponding time slots (w.r.t reversal), say, the first time slot 1 and the last time slot $2(n-1)$. The overall number of HAA pairs is $\binom{n}{n/2}^2$. For $n = 4$, the operation of G_4 on the $\binom{4}{2}^2 = 36$ pairs of HAAs is summarized in Figure 4(a): G_4 produces six orbits with lengths 2, 2, 4, 4, 8, and 16. In general, the table in Figure 4(b) shows that the ratio approximates the order $4n$ of the symmetry group G_n . It is, from $n = 4$ on, larger than the ratios for single slot HAAs and oriented 1-factors, and it is for $n \geq 6$ larger than twice the ratio for single slot HAAs. These observations suggest that using pairs of HAAs may lead to larger reductions in computing time than the symmetry reduction techniques that only take the dihedral group D_n into account. We will empirically check this assumption in Section 3.



Nb. of Teams n	Nb. of HAA pairs	Nb. of Orbits	Ratio
4	36	6	6
6	400	26	≈ 15.4
8	4,900	186	≈ 26.3
10	63,504	1,682	≈ 37.8
12	853,776	18,168	≈ 47.0

Nb. of Teams n	Nb. of 1-Fact. Pairs	Nb. of Orbits	Ratio
4	144	17	≈ 8.5
6	14,400	652	≈ 22.1
8	2,822,400	89,093	≈ 31.7
10	914,457,600	?	?
12	4,425,974,784	?	?

(b)

(c)

Figure 4: Operation of $G_n = D_n \times \mathbb{Z}_2$ (a) Orbits of HAA pairs for $n = 4$, (b) Number of HAA Pairs and Orbits, (c) Number of 1-Factor Pairs and Orbits

Now it is straightforward to also combine oriented 1-factors with reversals. Since the number of oriented 1-factors quickly increases with n , the number of pairs grows even faster, as can be seen in the table in Figure 4(c). Due to the huge number of pairs we were not able to compute the orbits for $n = 10$ and $n = 12$. Thus, symmetry reduction based on pairs of oriented 1-factors is not a promising approach to accelerate enumerative algorithms. We will not consider pairs of oriented 1-factors any further.

3. Computational Results

Algorithm. To quantify the benefits of the different symmetry reductions, we implemented a modified version of the DFS* approach of Uthus *et al.* (2009). Simplifying, we apply only the kernel of their approach which is a standard depth-first branch-and-bound (DFB&B) search, where we initialize upper bounds with the best known solutions from the literature. The DFB&B uses recursive backtracking to construct the overall

Instance	Computation Time					Number of Recursions					Opt. Sol.
	None	Tour Reversal	HAA	1-Factor	HAA pairs (100%)	None	Tour Reversal	HAA	1-Factor	HAA pairs (100%)	
circ 4	?	?	?	?	0s	229%	145%	84%	66%	38	20
circ 6	320%	200%	60%	60%	0.5s	713%	457%	127%	85%	58 713	64
circ 8	2041%	1411%	275%	189%	44s	2259%	1553%	303%	205%	8 568 856	132
circ 10	-	-	-	-	$6.0 \cdot 10^7$ s	-	-	-	-	$1.2 \cdot 10^{13}$	242

Table 1: Constrained Instances: Computation Times and Number of Recursions (“?”=unmeasurable)

schedule time slot by time slot. In each step, it picks (in numerical order) the first team that has not been assigned in the respective time slot and pairs it with a feasible opponent, at home or away. For the first time slot, the integration of our symmetry reduction methods is straightforward: only games that match the given representative in the first time slot are allowed.

Lower bounds in each step are given by the sum of the distances traveled so far and the independent lower bounds (ILBs) for the remaining tours of each team individually. The ILBs for each team are determined prior to the DFB&B execution. Herein, we use all information available at the respective step except for the particular teams left to play at home. Thus, each ILB is indexed by the team t it belongs to, the time slot s of the last assigned game of team t , the venue i_s^t , the current number of breaks, and the teams against whom team t still has to play away. This contrasts to the approach of Uthus *et al.* (2009) who do not use the exact number of remaining home games resulting in a slightly lower memory usage by the cost of possibly weaker bounds in some cases. The ILBs are the second point where the symmetry on HAA pairs is applied: we allow only for those tours that match the pattern of a given representative in the last time slot, which leads to tighter ILBs in many cases. To sum up, our branch-and-bound algorithm works as follows: First all orbits and all ILBs are computed, followed by the DFB&B for one representative of each orbit. This offers a natural way to parallelize the overall optimization as the orbits can be processed independently of each other.

Computational Setup. We tested our symmetry reduction methods on the different variants of the circular TTP instances, i.e. the constrained instances, where the number of consecutive home/away games is bounded by 3, as well as the unconstrained instances with and without NRCs. The TTP benchmark instances can be found on Michael Trick’s website <http://mat.tepper.cmu.edu/TOURN>.

All algorithms were coded in C++, compiled in release mode with MS-Visual C++ 2008 version 9.0, .NET version 3.5 SP 1. Most runs were performed on a standard PC with Intel Core2 Duo E8400 CPU with 3.00 GHz, 4GB main memory, on Windows 7 using a single thread. For the harder instances (circ 8 unconstrained and circ 10) we distributed the computations on several computers with different performance, but normalized computation times.

Runtime Analysis. The first observation is that the costs for calculating the orbits are negligible as their computation always takes less than 0,1% of the overall runtime for instances with $n \geq 8$ teams. For the smaller instances the computation times are not measurable.

Our most important finding is that all symmetry reductions cut the computation times of the branch-and-bound algorithm significantly. Table 1 summarizes our results regarding the constrained instances. For each instance, it shows the overall computation time and the total number of recursions. Since both times and recursions are mostly smallest for symmetry reduction based on HAA pairs, we present absolute values (=100%) for HAA pairs but relative values in all other cases. In general, the larger the symmetry group, the larger the average length of an orbit can be. Note that the ratios presented in Section 2 are in fact the average lengths of orbits. Now we empirically observe that the larger the orbits, the larger the speedup. Interestingly, the gains in computation times are slightly smaller than the gains in the number of recursions. This is due to the fact that in the branch-and-bound algorithm recursions are less expensive the deeper they are in the search tree, and reduction methods with more orbits generally have to consider more recursions with lower depth.

With increasing complexity of the instances we observe increasing speedups that, for circ 8, almost reach the maximal possible savings for all symmetry groups as computed in Section 2. For instance, we calculated

Instance	Computation Time					Number of Recursions					Opt. Sol.
	None	Tour Reversal	HAA pairs	HAA	1-Factor (100%)	None	Tour Reversal	HAA pairs	HAA	1-Factor (100%)	
w. NRCs											
circ 4	?	?	?	?	0s	348%	220%	152%	128%	25	20
circ 6	825%	494%	306%	119%	1.6s	887%	525%	291%	128%	428 587	56
circ 8	-	-	829%	129%	179 094s	-	-	633%	146%	41 103 371 129	102
w/o. NRCs											
circ 4	?	?	?	?	0s	332%	229%	123%	113%	31	20
circ 6	775%	483%	258%	108%	1.2s	883%	546%	263%	130%	283 361	54
circ 8	-	-	795%	134%	149 611s	-	-	562%	147%	34 037 268 057	100

Table 2: Unconstrained Instances: Computation Times and Number of Recursions (“?”=unmeasurable)

a ratio of 26.3 for HAA pairs and attained a speedup of factor 20.4; for 1-factors we have a ratio of 13.9 and a speedup of factor 10.7, and for slot HAAs 8.8 and 7.4, respectively.

The results suggest that similar speedups can be gained also for larger instances (if one were able to solve these). Thus, we think that the use of HAA pairs is the most promising approach to reduce symmetry when solving constrained instances. Using HAA pairs, we also confirm the result of Uthus et al. regarding the optimal solution of the constrained circ 10 instance (cf. <http://mat.tepper.cmu.edu/TOURN>).

The results for the unconstrained instances are given in Table 2 which is structured analogously to Table 1. However, our benchmark is now the symmetry reduction using 1-factors as computation times and number of recursions are smallest here. Still, the results are less uniform compared to the constrained case. While symmetry reduction based on slot HAAs and 1-factors again perform well and produce speedups approximating the ratios presented in Section 2, symmetry reduction with HAA pairs is slower than expected. We suspect that this is related to the gap between upper bound (or optimal solution) and ILBs, which is small for the constrained but huge for the unconstrained instances ($\frac{UB-ILB}{ILB}$ between 50% and 60% for $n = 6, 8$). When applying reduction with HAA pairs, the fixations in the last slot lead to slightly tighter bounds being particularly effective if the gap is small. For the unconstrained case, however, the small increase of the lower bounds typically does not suffice to terminate the tree search at an early stage. As a result, fixations with slot HAAs and HAA pairs need approximately the same number of recursions for one orbit, while there are many more orbits for HAA pairs leading to longer overall computation times. This explains why it is most beneficial to use 1-factors for the unconstrained instances.

Comparing constrained and unconstrained instances, the latter are considerably harder to solve (with approaches using ILBs). However, with the symmetry reduction methods we were able to prove optimality of the solutions of the unconstrained circ 8 instances provided by Langford (cf. <http://mat.tepper.cmu.edu/TOURN>). Using symmetry reduction based on 1-factors, circ 8 unconstrained with NRCs is solved in less than 50 hours, and circ 8 unconstrained without NRCs in less than 42 hours.

4. Conclusions

In this note we provided insights into the nature of the symmetry inherent in the circular TTP instances. The largest symmetry group is $G_n = D_n \times \mathbb{Z}_2$, which is the cartesian product of the dihedral group of order $2n$ (n is the number of teams) and the group of order 2. We have shown that G_n operates on pairs of home-away assignments of two corresponding time slots and can reduce the search space by a factor approximating $4n$. By selecting representatives from orbits it is easy to design algorithms that exactly cut off the symmetric parts of the solution space and thus are much faster.

Using a branch-and-bound approach we demonstrated that the speedups increase with the number n of teams. For the constrained circular instance with eight teams we observed a speedup of factor 20. A full analysis for more teams is impossible due to the large computation times, but the empirical results suggest that symmetry reduction can lead to speedups that approximate $4n$ for the larger constrained instances.

For the unconstrained circular instances, only the operation of D_n on 1-factors (implied by the games of a time slot) can be exploited effectively in the branch-and-bound approach. Here, speedups seem to approximate $2n$. Finally, with the accelerated branch-and-bound we were able to solve the two unconstrained instances circ 8 to proven optimality for the first time.

References

- Aschbacher, M. (2000). *Finite Group Theory*. Cambridge University Press.
- de Werra, D. (1980). Geography, games and graphs. *Discrete Applied Mathematics*, **2**, 327–337.
- Easton, K., Nemhauser, G., and Trick, M. (2001). The traveling tournament problem description and benchmarks. In T. Walsh, editor, *Principles and Practice of Constraint Programming - CP 2001*, volume 2239 of *Lecture Notes in Computer Science*, pages 580–585. Springer Verlag 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.
- Uthus, D. C., Riddle, P. J., and Guesgen, H. W. (2009). DFS* and the traveling tournament problem. Integration of AI and OR techniques in constraint programming for combinatorial optimization problems. In *6th International Conference, CPAIOR 2009*, Pittsburgh, Pennsylvania.