# Scheduling the Finnish 1st Division Ice Hockey League

**Jari Kyngäs, Kimmo Nurmi**

Satakunta University of Applied Sciences
Tiedepuisto 3, 28600 Pori, Finland
jari.kyngas@samk.fi, cimmo.nurmi@samk.fi

**Abstract**

Generating a schedule for a professional sports league is an extremely demanding task. Good schedules have many benefits for the league, for example higher incomes, lower costs and more interesting and fairer seasons. This paper presents a successful solution method to schedule the Finnish 1st division ice hockey league. The solution method is an improved version of the method used to schedule the Finnish major ice hockey league. The method is a combination of local search heuristics and evolutionary methods. An analyzer for the quality of the produced schedules will be introduced. Finally, we propose a set of test instances that we hope the researchers of the sports scheduling problems would adopt. The generated schedule for the Finnish 1st division ice hockey league is currently in use for the season 2008-2009.

## 1. Introduction

Many new timetabling problems have been introduced in recent years. Most of the timetabling research used to concentrate on university and school timetabling, but especially rostering and sports scheduling have been quite extensively studied recently. Excellent overviews on sports scheduling can be found in (Easton et al. 2004) and (Rasmussen and Trick 2008).

In the last decade, the sports scheduling focus has moved from theoretical results to practical applications. Some of the most important theoretical results can be found in (Schreuder 1980, 1992; de Werra 1981, 1988, 1990; Easton et al. 2001; Elf et al. 2003; Urrutia and Ribeiro 2004; Dinitz et al. 2007). An extensive summary of the theoretical results can be found in (Rasmussen and Trick, 2008).

Even if quite efficient algorithms have recently been designed for sports scheduling problems, to the best of our knowledge, there are only a few cases where academic researchers have been able to close a contract with a sports league owner: the major baseball league in USA (Nemhauser and Trick 1998), the major soccer league in Austria (Bartsch et al. 2006), the 1st division soccer in Chile (Durán et al. 2006), the major basketball league in New Zealand (Wright 2006), the major soccer league in Belgium (Goossens and Spieksma 2006), the major soccer league in Denmark (Rasmussen 2008), the major volleyball league in Argentina (Bonomo et al. 2008) and the major ice hockey league in Finland (Kyngäs and Nurmi 2009). This paper presents a new case: the Finnish 1st division ice hockey league.

The sports league scheduling problems solved in this paper are constrained minimum break problems (see e.g. Rasmussen 2008). The problem is to find a schedule with the minimum number of breaks and at the same time take additional requirements and requests into account. For the sports scheduling terminology used in this paper we refer to (Kyngäs and Nurmi 2009).

The focus of this paper is to solve a highly constrained sports scheduling problem. In Section 2 we give an overview of our earlier sports scheduling algorithm. Then we present an improved version of the algorithm. Section 3 presents the Finnish 1st division ice hockey league problem. The problem is extremely difficult both in terms of finding a feasible solution and of optimizing the requests from the league. Computational results are reported in this section. It will be seen that our algorithm produces excellent results compared to the manual schedule used in the previous season. An analyzer for the quality of the produced schedules will be introduced in Section 4. The use of the analyzer is vital in producing the final schedule for the league authorities. Finally in Section 5, we propose a set of test instances that we hope the researchers of the sports scheduling problems will adopt. It will be seen that our solutions for these instances are competitive

## 2. The Improved Algorithm

Our basic algorithm for solving sports scheduling problems is presented in (Kyngäs and Nurmi 2009), (Nurmi and Kyngäs 2007) and (Nurmi 1998). The algorithm is a genetic algorithm (Goldberg 1989) with one mutation operator and no recombination operators. The two most important features of the algorithm are the greedy hill-climbing mutation (GHCM) operator, which generates a new solution candidate from the current solution, and the adaptive genetic penalty method (ADAGEN), which is a multi-objective optimization method. The algorithm uses three mechanisms to help the search procedure to avoid local optima: genetic reproduction (Syswerda 1989), tabu search (Glover et al. 1985) and simulated annealing (Kirkpatrick et al. 1983). The use of these methods differs somewhat from their usual application (see Nurmi 1998).

The GHCM operator moves a game, *g1,* from its old round, *r1,* to a new round, *r2*, and then moves another game, *g2,* from round *r2* to a new round, *r3,* and so on, ending up with a sequence of moves. The initial game selection is random. The new round for the game is selected considering all possible rounds and selecting the one which causes the least increase in the cost function value when considering the relocation cost only. Moreover, the new game from that round is again selected considering all the games in that round and picking the one for which the removal causes the most decrease in the cost function value when considering the removal cost only.

The ADAGEN method is an adaptive penalty method for multi-objective optimization. A traditional penalty method assigns positive weights (penalties) to the soft constraints and sums the violation scores to the hard constraint values to get a single value to be optimized. The ADAGEN method assigns dynamic weights to the hard constraints.

The reproduction operation of the algorithm is, to a certain extent, based on the steady-state reproduction (Syswerda 1989). We use marriage selection (Ross and Ballinger 1993) to select a schedule from the population of schedules for a single GHCM operation. The new schedule replaces the old one if it has a better or equal fitness. Furthermore, the least fit is replaced with the best one when $n$ better schedules have been found, where $n$ is the size of the population.

Next we present two changes to the original algorithm. These changes will help the search procedure to escape from local optima as well as better explore the fitness landscape.

The original algorithm uses a simulated annealing refinement. The initial temperature $T_0$ is calculated by

$$T_0 = C^+ / \log(1/X_0) ,$$

where $X_0$ is the degree to which we want to accept an increase in the cost function (we use a value of 0.75) and $C^+$ is an average increment in the cost function for 100 random moves. This method was proposed by (van Laarhoven and Aarts 1987). The exponential cooling scheme is used to decrement the temperature:

$$T_k = \alpha T_{k\text{-}1} ,$$

where $\alpha$ is usually chosen between 0.8 and 0.995. Our new test runs showed that a good strategy is to stop the cooling at some predefined temperature. Therefore, after a certain number of iterations $m$ we will continue to accept an increase in the cost function with some constant probability $p$. Using the initial temperature given above and the exponential cooling scheme, we can calculate the value:

$$\alpha = (-1/(T_0 \log p))^{-m} .$$

Our preliminary test runs showed that we can get surprisingly good results by choosing $m$ equal to the maximum number of iterations with no improvement to the cost function and $p$ equal to 0.0015. The new annealing schedule seems to produce superior solutions compared to the well-known annealing schedules. The reason might be that it enables the search procedure to continue to escape from local optima. We will study this method more closely in our next paper.

The other change to the original algorithm concerns shuffling the current solution. A hyperheuristic (Cowling et. al. 2000) is a mechanism that chooses a heuristic from a set of simple heuristics, applies it to the current solution, then chooses another heuristic and applies it, and continues this iterative cycle until the termination criterion is satisfied. We use the same idea, but the other way around. We introduce a number of simple heuristics that are normally used to improve the current solution but, instead, we use them to shuffle the current solution - that is, we allow worse solution candidates to replace better ones in the current population. We use five shuffling operations:

1. Select a random game and move it to a random round, and do this $k_1$ times
2. Swap two random games, and do this $k_2$ times
3. Select a random round and move $k_3$ random games from that round to random rounds
4. Swap all the games in two random rounds
5. Select a random game A-B and swap it with the game B-A, and do this $k_4$ times.

We select one random shuffling operation in every $m$/20th iteration of the algorithm, where $m$ equals the maximum number of iterations with no improvement to the cost function. The best results have been obtained using the values $k_1 = 3$, $k_2 = 2$, $k_3 = 3$ and $k_4 = 2$. The shuffling seems to produce better solutions than without shuffling. The reason might again be that it enables the search procedure to continue to escape from local optima. We will again study this method more closely in our next paper.

Table 1 shows the result of the comparison between the original and the improved algorithms. Three different problems were used to compare their performance: one that minimizes just number of breaks, one that includes further restrictions (intermediate test problem) and finally a complex real-world problem. The improved algorithm using simulated annealing and shuffling refinements performs clearly better than the original algorithm.

Our algorithm uses random initial solutions. It has been claimed in many different contexts that better initial solutions lead to better final solutions. It has also been argued that it is a good idea to use canonical schedules (Schreuder 1980) as initial solutions for sports scheduling methods since canonical schedules minimize the number of breaks. We tested our algorithm using canonical starting schedules thus producing good initial solutions. We ran the algorithm several times using both artificial problems and real-world problems. The results were clear. Canonical schedules were unable to produce better final solutions than random schedules as initial solutions to our algorithm.

Table 1: The percentage of the best solutions found for the original algorithm and for the improved algorithm. The improved algorithm uses simulated annealing and shuffling refinements.

| Problem type | Original algorithm | Improved Algorithm |
|---|---|---|
| Break optimization only | 1% | 8% |
| Intermediate test problem | 13% | 21% |
| Complex real-world problem | 7% | 14% |

## 3. The Finnish 1st Division Ice Hockey League

Ice hockey is the biggest sport in Finland, both in revenue and number of spectators. The Finnish 1st division ice hockey league is managed by the Finnish Ice Hockey Association. The Competition Manager of the league is responsible for producing the schedule. Prior to the 2008−2009 season, the schedule was produced manually. When the manager heard that we had scheduled the major league (Kyngäs and Nurmi 2009), he contacted us. He told us that the 1st division is an even more difficult problem than the major league. We agreed to generate a sample schedule for them.

The league has twelve teams. Two of the teams are located in big cities (over 100 000 citizens) and the rest in smaller cities. One team is quite far up north, one on the west coast, three teams are located in the east and the rest in Central Finland (see Figure 2).

The schedule format for the league has been stable for many years. The basis of the schedule is a quadruple round robin tournament concluding in 44 games for each team. In addition, each team plays at home against the Finnish U20 team (national team for players under 20 years of age). Therefore, there are 45 games for each team and a total of 276 games to be scheduled. The games should be scheduled on Wednesdays and Saturdays.

The league first fixes the dates on which the rounds will be played. They only fix 44 dates - that is. the basic schedule should be a compact schedule. The U20 games are preassigned to given dates.

Often, there are also parties other than the league and the teams involved in the scheduling process. Examples of such parties include TV networks and other leagues. In the case of the Finnish 1st division ice hockey league the TV network chooses the games to show before the scheduling process. These games are preassigned to given rounds. The Finnish major ice hockey league introduces further requirements. Five teams in the 1st division are located in (or very close to) the same cities as the teams in the major league. The major league is scheduled first and these five teams should not play at home on the same days as their counterparts in the major league. Furthermore, three other teams are competing with the Finnish major basketball league for the same spectators. This league is again scheduled first, so these three teams should not play at home on certain days.

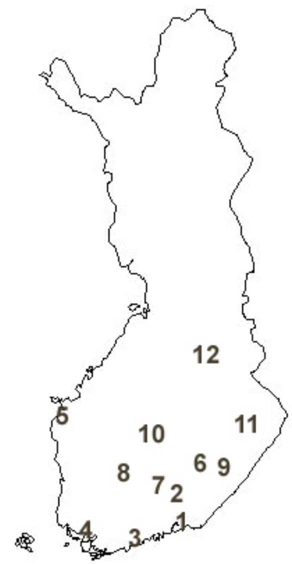


Figure 2: The map of Finland and the twelve teams in the Finnish 1st division ice hockey league.

For the following terminology and notation we refer to (Kyngäs and Nurmi 2009). The league and the teams gave the following requirements for the 2008−2009 season:

*H1*. Every team plays in every round exactly once (a compact schedule).

*H2*. 36 home games are forbidden on certain days.

*H6*. A team cannot play at home on two consecutive calendar days.

*H7*. 61 preassigned games.

*H8*. There cannot be a break in the second round.

In addition, the league and the teams gave the following requests:

*S2*. A team cannot have more than two consecutive home games.

*S3*. A team cannot have more than two consecutive away games.

*S4*. The LeKi, Hokki, Jokipojat, Kiekko-Vantaa and TuTo teams wish to play a few away tours.

*S5*. There must be at least six rounds before two teams meet again.
*S7*. All teams wish to play their home games on Saturdays.
*S9*. The D-Team, HeKi, Kiekko-Vantaa, LeKi and TuTo teams do not want to play at home on the same day as their major league counterparts.
*S12*. The difference between the number of home games and the number of away games for each team should be as small as possible after each round.

The most important requests from the league were assigned a larger weight in the ADAGEN method. The following weights were used for hard constraints:

- 3-25 for *H1*.
- 3-10 for *H2*.
- *H6* to *H8* were preassigned.

The construction of the schedule was quite a difficult task. First of all, there were a considerable number of restrictions − over 10% more than in the Finnish major ice hockey league (Kyngäs and Nurmi 2009). Secondly, the teams had many more wishes than the teams in the major league.

Table 2: The manual solution for the 2007−2008 season and our best solution for the 2008−2009 season. Unfortunately it is not possible to calculate values for other constraints from the 2007−2008 manual schedule.

| | 2007-2008 (manual) | 2008-2009 (algorithm) |
|---|---|---|
| *H1*: number of rounds (does not include those needed for the away tours) | 64 | 44 |
| *H2,H7*: number of forbidden and preassigned games | < 60 | 97 |
| *S2,S3*: number of 3-breaks (at home + away) | 68 | 2 + 4 |
| *S4*: number of away tours | 13 | 17 |
| *S7*: minimum number of home games on weekends | 9 | 12 |

We generated two schedules as we did for the major league. After the generation of the first schedule the Competition Manager discovered that he had forgotten a few restrictions. We added them to the program and produced a second schedule. The second schedule was accepted and only two games were relocated both due the fact that the teams would have had to play three consecutive away games because of home venue unavailability. So the schedule is in use almost as generated. Table 2 shows our solution for the 2008−2009 season and a comparison with the solution produced manually for the 2007−2008 season. The algorithm was run on an Intel Core 2 Duo PC with a 3.8GHz processor and 2GB of random access memory running Windows XP. Our solution (best of the 50 runs) was found in six hours of computer time, whereas the manual solution took several weeks to construct.

The Competition Manager was very satisfied with the schedules we generated and we closed a contract with the league.

## 4. The Analyzer

It is very difficult to examine the quality of the generated schedule. Even if someone were to have the patience to do it, it would take quite a long time. To ease this process we have made an analyzer for analyzing sports schedules.

Actually we have made two analyzers: one for analyzing the output file of our program (algorithm) and one for analyzing any general sports schedule. The analyzer takes the schedule, requirements and requests as input, given as text files and produces a simple text file as output, where it is very easy to examine the conflicts in the schedule. The output file details each restriction and possible conflicts for each team in a readable form. A few examples:

- All breaks
- Weekday preferences are listed day by day (given lower bound, given upper bound, actual value).
- Every game that has a violation in the k-value.

The output file is excellent for presenting the results to the customer. When we introduced our program to the Competition Manager we generated three somewhat different schedules for the first half of the season. The manager could very quickly see the benefits of our program just by inspecting the output file. We claim that the use of the analyzer is vital in producing the final schedule for the league authorities.

## 5. A Set of Test Instances

The generation of standard test problems does not receive much attention. Some benchmark instances for round robin tournaments have been introduced in Henz (2000). For the Traveling Tournament Problem (Easton et al. 2001), test instances can be found in (Trick 2008). No set of standard test instances exists for the constrained minimum break problem.

Researchers quite often only solve some special artificial cases or one real-world case. The strength of random test instances is the ability to produce many problems with many different properties. The strength of practical cases is self-explanatory. However, an algorithm performing well on one practical problem may not perform satisfactorily on another practical problem. Our future work will present a set of both artificial and practical test instances for the constrained minimum break problem. In this section we present a collection of test instances found in the literature as well as some new test instances.

Table 3: Twelve 2RR test instances: R50 (1), R100 (2), B10 (3), B12 (4), B14 (5), B10K3 (6), B12K3 (7), B12K10 (8), R14K7P208 (9), B8K0P30 (10), B8K2P30 (11), B10K2C4 (12).

| ID | n | Break min. | k | #Constr. | Optimal #breaks | Our solution |
|---|---|---|---|---|---|---|
| **1** | 50 | No | 0 | 0 | – | found |
| **2** | 100 | No | 0 | 0 | – | found |
| **3** | 10 | Yes | 0 | 0 | 8 | 8 |
| **4** | 12 | Yes | 0 | 0 | 10 | 10 |
| **5** | 14 | Yes | 0 | 0 | 12 | 14 |
| **6** | 10 | Yes | 3 | 0 | ? | 16 |
| **7** | 12 | Yes | 3 | 0 | 16 | 22 |
| **8** | 12 | Yes | 10 | 0 | ? | 26 |
| **9** | 14 | No | 7 | 208 P | – | found |
| **10** | 8 | Yes | 0 | 30 P | ? | 10 |
| **11** | 8 | Yes | 2 | 30 P | ? | 12 |
| **12** | 10 | Yes | 2 | 4 C | ? | 10 |

To the best of our knowledge, the best test instances presented in the literature so far are those by (Rasmussen and Trick 2007). We use four of their problems, two of which are slight modifications. All twelve test instances are double round robins. Table 3 shows the instances. The first two (abbreviated as Rn) are simple round robin problems where the only challenge is to find a round robin tournament. In the next three instances (Bn) the challenge is to find the minimum number of breaks. The next three instances (BnK) are also break minimization problems, but in these instances two games with the same opponents must be separated by at least $k = 3$ or $k = 10$ rounds. In the instance R14K7P208 the challenge is again just to find a round robin tournament, but now with $k = 7$. Furthermore, there are four home game restrictions and four away game restrictions in each round totaling a number of 208 restrictions (place constraints). The next two instances (B8) are break minimization problems with place constraints and the other one with $k = 2$. Here we had to modify the original problems by (Rasmussen and Trick 2007) because their place constraints would have caused extra breaks to occur. The final instance B10K2C4 introduces complementary constraints – that is, two teams cannot play at home at the same day. The instance includes four complementary constraints.

It should be noted that our algorithm was not designed to merely minimize the number of breaks but to solve complex real-world problems. However, we claim that our algorithm also works very well for the artificial test instances. We were able to find the best possible solution for five of the test instances. For three of the instances the optimum is not yet known. The up-to-date collection of test instances can be found on the web (Nurmi and Kyngäs 2009).

## 6. Conclusions and Future Work

We scheduled the Finnish 1st division ice hockey league. Our algorithm found a feasible and an acceptable schedule for the the 2008−2009 season. The generated schedule is currently in use. We also proposed a set of test instances that we hope the researchers of the sports scheduling problems will adopt. Our solutions to the test instances were competitive.

Our direction for future research will be to further study the improved algorithm and its various parameters. We will also publish an extensive set of both real-world instances and test instances for the constrained minimum break problem. We have already set up a group of collaborators for this goal.

## References

Bartsch, T., Drexl, A., and Kröger, S. 2006. “Scheduling the professional soccer leagues of Austria and Germany”, *Computers and Operations Research* 33(7): 1907–1937.

Bonomo, F., Burzyn, A., Cardemil, A., Durán, G., and Marenc, J. 2008. “An application of the traveling tournament problem: the Argentine volleyball league”. In *Proc. of the 7th International Conference on the Practice and Theory of Automated Timetabling (PATAT),* Montreal (Canada).

Cowling, P., Kendall, G., and Soubeiga, E. 2000. A hyperheuristic Approach to Scheduling a Sales Summit. In *Proc. of the 3rd International Conference on the Practice and Theory of Automated Timetabling (PATAT)*, 176-190.

Dinitz, J. H. , Froncek, D. , Lamken, E. R. , and Wallis, W. D. 2007. “Scheduling a tournament” in *The CRC Handbook of Combinatorial Designs*, C.J. Colbourn and J. H. Dinitz, Eds. CRC Press Inc, Florida, USA, 591–606 (previous edition of this book was published in 1996).

Durán, G., Guajardo, M., Miranda, J., Sauré, D., Souyris, S., and Weintraub, A. 2007. “Scheduling the Chilean

Soccer League by Integer Programming", *INTERFACES* 37: 539–552.

Easton, K., Nemhauser, G., and Trick, M. 2001. "The traveling tournament problem: description and benchmarks". In *Proc. of the 7th. International Conference on Principles and Practice of Constraint Programming*, Paphos, 580–584.

Easton, K., Nemhauser G., and M. Trick. 2004. "Sports scheduling" in *Handbook of Scheduling: Algorithms, Models and Performance Analysis*, J. T. Leung, Ed. CRC Press Inc, Florida, USA, pp 1–19.

Elf, M., Jünger, M., and Rinaldi, G. 2003. "Minimizing breaks by maximizing cuts", *Operations Research Letters* 31: 343–349.

Glover, F., McMillan, C., and Novick, B. 1985. Intercative Decision Software and Computer Graphics for Architectural and Space Planning. *Annals of Operations Research 5*: 557-573.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*, Kluwer Academic Publishers, Boston, MA.

Goossens, D., and Spieksma, F. C. R. 2006. "Scheduling the Belgian soccer league". In *Proc. of the 6th International Conference on the Practice and Theory of Automated Timetabling (PATAT)*, Brno, Czech Republic, 420–422.

Henz, M., Müller, T., Thiel, S., and van Brandenburg, M. (Created 2000). Benchmarks and results for round robin tournaments. http://www.comp.nus.edu.sg/˜henz/roundrobin_benchmarks/.

Kirkpatrick, S., Gelatt, C.D. Jr, and Vecchi, M.P. 1983. Optimization by Simulated Annealing. *Science 220*: 671-680.

Kyngäs, J., and Nurmi, K. 2009. Scheduling the Finnish Major Ice Hockey League. IEEE Symposium on Computational Intelligence in Scheduling, Nashville, USA, accepted for publication.

van Laarhoven, P.J.M., and Aarts, E.H.L. 1987. *Simulated annealing: Theory and applications*. Kluwer Academic Publishers.

Nemhauser, G., and Trick, M. 1998. Scheduling a major college basketball conference, *Operations Research* 46(1): 1–8.

Nurmi, K., and Kyngäs, J. (Last update 28.1.2009). Sports Scheduling Problem [Online]. Available: http://www.samk.fi/ssp.

Nurmi, K. 1998. "Genetic Algorithms for Timetabling and Traveling Salesman Problems". Ph.D. diss., Dept. of Applied Math., University of Turku, Finland.

Nurmi, K., and Kyngäs, J. 2007. "A Framework for School Timetabling Problem". In *Proc. of the 3rd Multidisciplinary Int. Scheduling Conf.: Theory and Applications*, Paris, France, 386–393.

Rasmussen, R. 2008. "Scheduling a triple round robin tournament for the best Danish soccer league", *European Journal of Operational Research* 185(2): 795–810.

Rasmussen, R. and Trick, M. 2007. "A Benders approach for the constrained minimum break problem", *European Journal of Operational Research* 177: 198–213.

Rasmussen, P., and Trick, M. 2008. "Round robin scheduling - A survey", *European Journal of Operational Research* 188, pp. 617–636.

Ross, P. and Ballinger, G.H. 1993. "PGA - Parallel Genetic Algorithm Testbed", Department of Articial Intelligence, University of Edinburgh, England.

Schreuder, J.A..M. 1980. "Constructing timetables for sport competitions", *Mathematical Programming Study* 13, 58–67.

Schreuder, J. A. M. 1992. "Combinatorial aspects of construction of competition Dutch Professional Football Leagues", *Discrete Applied Mathematics* 35: 301–312.

Syswerda, G. 1989. "Uniform Crossover in Genetic Algorithms". In *Proc of the 3rd International Conference on Genetic Algorithms*, 2–9.

Trick, M. (Last update 31.7.2008). Challenge Traveling Tournament Instances, http://mat.gsia.cmu.edu/TOURN.

Urrutia, S., and Ribeiro, C.C. 2004. "Minimizing travels by maximizing breaks in round robin tournament schedules", *Electron Notes Disc Math* 18-C: 227–233.

de Werra, D. 1981. "Scheduling in sports" in *Studies on graphs and discrete programming. Amsterdam*, P. Hansen, Ed. North-Holland, pp. 381–95.

de Werra, D. 1988. "Some models of graphs for scheduling sports competitions", *Discrete Applied Mathematics* 21: 47–65.

de Werra, D., Jacot-Descombes, L., and Masson, P. 1990. "A constrained sports scheduling problem", *Discrete Applied Mathematics* 26: 41–49.

Wright, M.B. 2006. "Scheduling fixtures for basketball New Zealand", *Computers & Operations Research* 33, 1875–1893.