Weighted Matching in Chess Tournaments

Author(s): Snjólfur Ólafsson

Source: *The Journal of the Operational Research Society*, Jan., 1990, Vol. 41, No. 1 (Jan., 1990), pp. 17-24

Published by: Palgrave Macmillan Journals on behalf of the Operational Research Society

Stable URL: http://www.jstor.com/stable/2582935

J. Opl Res. Soc. Vol. 41, No. 1, pp. 17–24, 1990
Printed in Great Britain.
0160-5682/90 $3.50 + 0.00

Weighted Matching in Chess Tournaments

SNJÓLFUR ÓLAFSSON
Science Institute, University of Iceland, Reykjavík

In many chess tournaments, e.g. when the Swiss system is used, the number of players is much larger than the number of rounds to be played. In such tournaments the pairing for a round depends on the results in earlier rounds, and the pairing process can be very complicated. In these pairing systems the main goals are to let players with equal scores play together, and that each player should alternately play white and black, with the restriction that no player may face the same opponent more than once. The paper describes how a weighted matching algorithm is used to find 'the best pairing' by converting the pairing rules into penalty points.

Key words: chess, combinatorial optimization, scheduling, matching

INTRODUCTION

In some chess tournaments the pairing of the players for each round can be both time-consuming and difficult. In this paper we describe how an efficient computer program was made for this pairing. In order to do so, the problem was formulated as a series of maximum weight matching problems. After some general information on chess tournaments and more detailed description of the pairing system, the paper focuses on this problem formulation.

Chess tournaments are of two types: either each player plays with all the other players, or he plays with relatively few of the other players. Here we are entirely concerned with the latter type, often called a Swiss tournament or a Monrad tournament. In such a tournament there are typically 30–80 players and 7–11 rounds played. These numbers are, however, far from being lower and upper bounds.

The basic idea in these systems is that players with equal scores should be paired together but two players may only play against each other once. The winner of a game scores one point, and in case of a draw each player scores half a point. Every player plays in every round. See the Appendix for a closer description.

Since players with a full score are paired together, the upper bound for the number of players with a full score after R rounds, in a tournament with N players, is $0.5^{R} \cdot N$. Therefore, when at least $\log_2 N$ rounds have been played, there is at most one player with a full score. ($\log_2 32 = 5$, so when there are 32 players, five rounds are needed.) Usually there are some 'extra' rounds so that the winner(s) have to face many of the top players. When there are many players of varying strength, they tend to get opponents of similar strength in all except the first rounds.

The Swiss system was first used in 1903 in the Swiss national tournament. In April 1925 a pairing system constructed by K. D. Monrad was published in Denmark in *Skakbladet*. Both systems have been widely used in several variants since then.

The main difference between the Monrad system and the Swiss system is that in the latter an attempt is made to pair each player so that he has alternatively white and black, whereas in the former very little attention is paid to the colours when paired. Thus it can happen in a Monrad tournament that someone plays white in the two first rounds and black in the remaining five rounds. (The one who plays white is to move first, which is an advantage.)

Local chess clubs usually run their tournaments using some variant of the Monrad system, since it is much easier to use. One person can do the pairing in a few minutes. The procedure could be like this for each round:

0. Given is some 'initial ordering' of the players (e.g. the registration number or the ELO rating).
1. Order the players according to score in decreasing order (i.e. player number 1 has the highest score). Within a score group, order the players according to the initial ordering. This ordering gives each player a temporary number.
2. Is there an unpaired player? If no: quit, the pairing is done. If yes: go to step 3.
3. Try to pair the unpaired player who has the lowest temporary number:

(i) If there is at least one unpaired player with whom he has not played, then pair him with one of those with the lowest temporary number and go to step 2.
(ii) If he has played with all the unpaired players, then delete the last pairing and try to pair all the unpaired players in 'some good way'. If necessary, delete more pairings. Quit.

In more 'serious' tournaments, e.g. international tournaments, usually some variant of the Swiss system is used (see the Appendix). One of the main reasons for using complicated rules is to give each player alternating colours as far as possible. Sometimes the pairing is very difficult, and it may take two or three men 2–4 hours to achieve an acceptable pairing in a tournament with between 50 and 100 players. Even so, they can expect to hear some complaints from the players. Thus it is worth a great deal to have an automatic way to do this pairing.

THE BASIC PRINCIPLES OF THE PAIRING SYSTEM

The regulations for Swiss system open tournaments, as approved by the 1985 General Assembly of FIDE (Fédération Internationale des Echecs), are used in this work. In this section we outline the pairing process, without going into details, but in the next section some details are given, and in the Appendix we give the rules of the Swiss system as published by FIDE. In addition to these rules, there are unofficial working rules.

Before the pairing of the first round, the players receive *rank numbers* according to decreasing ELO ratings; i.e. the player with highest ELO rating gets rank number 1. In cases of equal ELO ratings, rank is determined by international titles or other means. ELO ratings are computed and published by FIDE twice every year.

In the first round the half of the players with the lowest rank numbers play with the other half. For example, in a tournament with 30 players, numbers 1 and 16 would be paired together, 2 and 17, and so on.

In other rounds, players with equal scores should play together as far as possible. When there are many players in a *score group* (players with equal scores), the pairing that 'equalizes the number of whites and blacks best for each player' should be chosen. This means that, as far as possible, the players who 'seek white' (i.e. have played black more often than white) get opponents that 'seek black'. These are the most important goals, but there is a third one, namely that within a score group, the top half (lowest rank numbers) should be paired with the lower half in such a way that the topmost in each half play together, and so on. (This was done in the first round above.)

It is generally not possible to reach these goals. It would be possible to define a goal function and request a pairing that would minimize that function. Then the tournament manager would need to solve a minimization problem, and it would be hard to verify the solution. Therefore this possibility is practically infeasible. Instead, the pairing is done for one score group at a time, starting at both ends and moving towards the middle.

If, at some time, it is not possible to pair all the players in a score group together, then some of the players are transferred to the next group and then paired with the players in that group. We start by pairing the topmost players, and continue to just above the middle group. Then we go from the bottom and upwards to the middle group, where all loose ends are tied.

CONVERTING THE PAIRING RULES INTO AN ALGORITHM

The basic idea in our problem formulation is that, in principle, all possible pairs are assigned penalty points in such a way that the pairing (where all players are paired) with as low sum of penalty points as possible is the pairing that best fulfils the pairing rules.

The total problem could be solved as one minimum weight pairing problem (where each player is represented by a point on a graph), and in that way the pairing that comes closest to the goals of the system could be obtained. There are, however, several reasons for splitting the problem into, roughly, one minimum weight problem for each score group. First, the rules explicitly say that one score group should be paired at a time. Secondly, solving the pairing for the whole tournament as one problem could result in a pairing that is hard to understand, and finally, the pairing of the 5–10 top players is more important than the pairing of the other players.

In our notation we let a player score two points for a victory, one point for a draw, and zero for a lost match. Thus, when R rounds have been played, the players have between 0 and 2R score points, inclusive. We now assign score groups according to points; i.e. there are 2R + 1 score groups, the top players are in group 2R and the median group is group R. There may be some empty groups.

When pairing for round R + 1, pair the players in the groups, according to the rules, in the order 2R, 2R − 1, ..., R + 1, 0, 1, ..., R − 1, R. If a player gets no opponent, when his group is paired he is transferred to the next group, as shown in Figure 1.

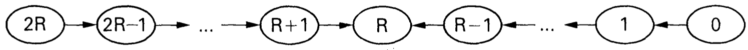

FIG. 1. *Direction of transfers between score groups.*

Now we describe the pairing of group K, where R < K < 2R. For 0 < K < R the pairing is similar; for K = 0 and K = 2R it is a bit simpler; and K = R is a special case treated later. This description is not complete as it omits some details, mainly representing the working rules. However, all the main ideas are here.

We are to pair M players, of whom m are floaters; i.e. they are transferred from a higher group. (Usually m is 0 or 1.) We give the floaters the numbers $1, \ldots, m$. The rest of the players, $m + 1, \ldots, M$, are in score group K and have the same order as initially (i.e. according to decreasing rank number). After R rounds we define the following parameters, where i and j take the values $1, \ldots, M$:

$C(i)$ = the number of times player i has played white minus the number of times player i has played black [the colour number];
$S(i)$ = the score of player i ($0 \leqslant S(i) \leqslant 2\mathrm{R}$) [the score number];
$S_{ij} = |S(i) - S(j)|$ [the score difference].

The number of penalty points given to a pair is a sum of several terms:

$$T_{ij} = O_{ij} + N_{ij} + C_{ij} + H_{ij}$$

[total = old floater + new floater + colour difference + halves situation].

Following is a description of the terms in this sum. There are several parameters used in the definitions that control the weights of these terms. The values of the parameters were determined to a large extent by studying the Swiss rules, but finally they were adjusted by means of trial and error. In fact, some of the working rules were 'defined' in this way.

The values actually used in this program are given in parentheses.

$$O_{ij} = \begin{cases} 0 & \text{if } S(i) = S(j) = \mathrm{K}, \\ c_1 \cdot S_{ij} \cdot |i - j| - c_2 & \text{otherwise} \quad (c_1 = 25;\ c_2 = 500). \end{cases}$$

The c_2 term in O_{ij} guarantees that as many as possible of the (old) floaters get an opponent. The term containing c_1 has the effect of giving a floater (say i) an opponent with high rank number (i.e. low j).

$$N_{ij} = c_3 \cdot (i + j) \qquad (c_3 = 10).$$

Which players are to be transferred, i.e. to become the new floaters? If we cannot pair all the players, we leave out players with as low a ranking as possible (i.e. as high values for i and j as possible).

$$C_{ij} = \begin{cases} c_4 \cdot c_5^{\mathrm{P}} & \text{if } C(i) \cdot C(j) > 0 \quad (c_4 = 25;\ c_5 = 8), \\ 0 & \text{otherwise,} \end{cases}$$

where $P = \min\{|C(i)|, |C(j)|\}$.

C_{ij} contains the colour penalty points. If $C(i)$ and $C(j)$ have the same sign, then players i and j should get the same colour if possible, and thus there is a penalty for pairing them. The reason for using $\min\{|C(i)|, |C(j)|\}$ is that the player with the lower $|C(.)|$ value will get the wrong colour.

$$H_{ij} = \begin{cases} (B - |i - j|)^2 & \text{if } S(i) = S(j) = \mathrm{K}, \\ 0 & \text{otherwise,} \end{cases}$$

where $B = (M - 2m)/2$.

Players in 'similar places' in the two halves should be paired as far as possible. The ideal is to pair the transferred players from above with the players $i = m + 1, \ldots, 2m$ and the remaining $M - 2m$ players such that $|i - j| = B = (M - 2m)/2$. For example, if there are nine players in a group and $m = 1$, then the pairing should be 3–6, 4–7, 5–8, and number 9 would become a floater. See Figure 2 for an explanation.

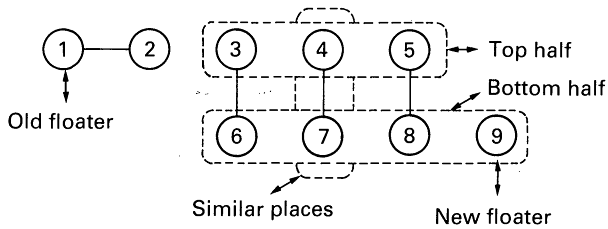

FIG. 2. *In a score group the top half should preferably be paired with the bottom half.*

This formulation has been shown to be very flexible for implenting different variants of pairing rules. As said before, in practice we construct T_{ij} in a more complex way in order to include several working rules used by tournament managers. The best pairing in this group is found by solving a maximum weight matching problem on a graph, where a point represents a player. There is an arc between two points if the players have not played before, and the weight for this arc is defined as $w_{ij} = \text{constant} - T_{ij}$. The constant is large, and guarantees that as many pairs as possible will be put in the matching.

This problem can be solved in polynomial time, and an $O(n^4)$ algorithm, given by Lawler,[1] is used in this work. The first polynomial-time algorithm for this problem is due to Edmonds.[2] The current best algorithm for this problem has the complexity $O(n(m \log \log_{[m/n+1]} n + n \log n))$, according to Gali,[3] where n is the number of nodes and m is the number of arcs.

For group R the situation is special. Then there can be players transferred from both above and below. All unpaired players must be paired now. If it is not possible to pair them all within this group, then the pairing of the adjacent groups must be altered. That is done by pairing groups R − 1, R and R + 1 simultaneously and treating all players in R − 1 and R + 1 as floaters.

We have so far concentrated on the pairing within the score groups, and will now consider one important detail concerning the overall algorithm.

Assume that we are pairing group K with one old floater from group K + 1, and two new floaters are made since it is impossible to find $M/2$ pairs. Then it may be possible to replace the old floater from group K + 1 with another floater, in such a way that $M/2$ pairs can be found in group K. Then this other floater should be chosen since the goal to pair players with equal scores is more important than other goals.

A simple example is shown in Figure 3, where an arc means that the players have not played against each other before. We assume there that R < K < 2R.

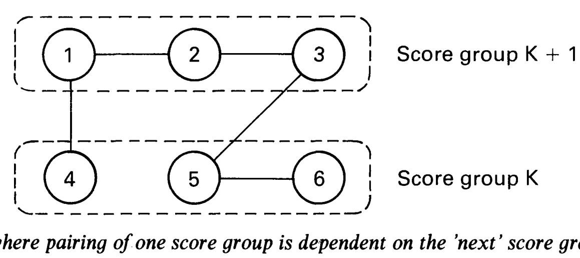

FIG. 3. *A situation where pairing of one score group is dependent on the 'next' score group.*

If 1 and 2 are paired and 3 is transferred down, then both 4 and 6 must be transferred from group K. If 2 and 3 are paired, then no player has to be transferred from group K. Whatever the colour situation is, the latter pairing is the better one. In order to avoid such mistakes as choosing the first pairing, the algorithm does the following: if more than one player is to be transferred from a score group which already includes a transferred player, then combine the score group with the preceding group and try to find a pairing with fewer transfers from the latter group. This is done simply by treating all the players in group K + 1 as floaters in group K.

It could happen that a change of a floater from group K + 2 would affect the number of floaters from group K. This possibility is considered neither in the rules nor in the working rules and hence not in the algorithm.

THE MATCHMAKER SYSTEM

The pairing algorithm described in this paper was programmed in TURBO-PASCAL by the author and then integrated into a complete chess-tournament administration system programmed by a software house (VKS, Bildshöfda 13, 112 Reykjavík, Iceland). This system, MATCHMAKER, is now commercially available. In this section we give a short description of the system and some experience of the use of it.

The system can handle not only Swiss tournaments but also Monrad tournaments and tournaments where all players play together (round robin). It can handle very large tournaments, or up to 1000 participants in up to 10 sections playing up to 30 rounds.

During registration, all necessary information about the players, such as ELO rating, titles and nationality, is entered into the system.

The system can deliver various information, e.g.:

—which players play together in the next round,
—the status of every player, such as score and colour status,
—which players have won or can possibly gain an international title,
—the final position of the players and an updated ELO rating,
—reports to FIDE.

When the pairing is performed, it is optional to fix some pairs and let the program pair the remaining players. The user does not have to, and cannot, adjust the parameters used by the matching algorithm. On the other hand, the system can quite easily be adapted to other variants of the Swiss system by the programmers.

The software has been used in several tournaments without any complaints about the pairing. When it has been compared with manual pairing, it has always found the same pairing or a better one. The following is an example of this.

The international tournament Reykjavik Grand Open 1986 took place in February 1986. The number of participants was 80 (of whom 26 were grand masters), and 11 rounds were played.

In general, most stress is put on the pairing of the topmost players, especially in the last few rounds. Much time is therefore spent in finding the very best pairing for the top, but the set of acceptable solutions for the less important players is usually quite large. This was also the case in this tournament.

The pairing for the last round turned out to be very difficult. The situation of the 14 topmost players is described in Table 1. The pairing of the other players is not important for this example.

TABLE 1. *The status of the topmost players before the last round*

No.	Name	Score	Opponents	Last colour
1	Mikhail Tal	7	5 7 8 11	B
2	Bent Larsen	7	3 4 7 8 9	W
3	Predrag Nikolic	7	2 4 5 7 8 9 13	W
4	Florin Georghiu	7	2 3 7 8 13	B
5	Valery Salov	7	1 3 7 13	W
6	Nick De Firmian	7	9 13	W
7	Curt Hansen	7	1 2 3 4 5 8 9	W
8	Jóhann Hjartarson	7	1 2 3 4 7 14	B
9	Anthony J. Miles	6, 5	2 3 6 7 11 14	B
10	Larry Christiansen	6, 5	13	W
11	Efim Geller	6, 5	1 9 12 13	W
12	Lev Alburt	6, 5	11 14	B
13	Robert Byrne	6, 5	3 4 5 6 10 11	B
14	Helgi Ólafsson	6	8 9 12	B

In Table 1 the numbers in the first column are temporary numbers. Eight of the players have scored seven points, and since some of them are going to play each other, at least one will be the winner of the tournament, or one of the winners in the case of many winners. For each player, all of his earlier opponents in this group are listed. It is easy to see that it is impossible to get four pairs of players with seven score points. Hence two of them will become floaters and be paired with the next score group. When that score group has been paired, there will be one new floater, who will be paired with Helgi Ólafsson, who is therefore in Table 1. For every player except number 12, the colour difference is 0; i.e. they have played black and white the same number of times. In this case, as many players as possible should get a different colour from last time, which explains the last column in the table. This is a generally accepted working rule in the spirit of rule 3.3 in the Appendix.

This pairing turned out to be very difficult. Three experienced tournament managers spent several hours trying to find the best pairing, but MATCHMAKER needs approximately one minute to pair all the players. When compared, MATCHMAKER's pairing is seen to be the better one. The pairings are given in Table 2. Both pairings have three floaters. In the manual pairing four players have to play the same colour twice, but none in MATCHMAKER's pairing.

TABLE 2. *Comparison of manual and MATCHMAKER's pairings. FL indicates a floater; BB and WW mean that a player will play the same colour twice*

Manual pairing		MATCHMAKER's pairing	
Pair	Penalty	Pair	Penalty
1–4	BB	1–2	
5–2	WW	4–6	
6–3	WW	8–5	
12–7	FL	12–3	FL
8–10	FL	13–7	FL
9–13	BB	9–10	
14–11	FL	14–11	FL

FINAL REMARKS

The computer software MATCHMAKER optimizes pairing of chess tournament players under Swiss system rules as approved by FIDE. It is easy for the programmers to adjust the system to meet other variants of the Swiss system.

In optimization, the goal function usually has some connection with money or time. In our case, 'best pairing' means that the pairing satisfies the rules better than all other pairings. The program user gets a 'good pairing' which he understands, and since he is unable to obtain a better one himself, he is satisfied. In general, 'good, understandable solutions' is what is wanted rather than the 'best of all possible solutions'.

The mathematical programming technique presented here is integrated into a complete system and is thus invisible to the user. This work resulted from co-operation between an operational researcher, a software firm and people with experience in managing chess tournaments.

APPENDIX

The following is quoted from the *International Rating List, January 1986*, published by FIDE (Fédération Internationale des Echecs, P.O. Box FIDE, 6002 Lucerne, Switzerland):

Regulations for Swiss System Open Tournaments.

Short title 'FIDE Swiss Rules'.

Approved by 1985 General Assembly, Graz.

Scope: These regulations are intended for use in FIDE competitions and in FIDE registered competitions which are declared to be conducted by 'FIDE Swiss Rules'. In this case, only minor departures from these regulations are permitted, and such departures must be declared before the competition begins and the attention of participants specially drawn to the departures.

1. Basic Principles of Swiss System Pairing

The following principles (article 1) are listed in descending order of priority, and take precedence over anything stated in the subsequent articles.

1.1. All participants play in one tournament.

1.2. Players with the same score are paired together so far as possible, but two players may only play against each other once. In the case of a game which is paired but not actually played, for example a game decided by forfeit when one of the players fails to arrive, the same players may be paired again in a later round. If a game is not actually played, each player is assumed to have no color.

1.3. If, for any round, there is an odd number of players in the tournament, one player must receive a bye. This is a player who has not hitherto had the bye, of lowest possible score, who best equalizes the colors of the remaining contestants with the same score. He scores 1 point for that round, and is assumed to have had no color.

1.4. As far as possible, at the end of each round, all players must have had an equal number of whites and blacks.

1.5. When there is an odd number of players in a score group, or one or more players in a score group have met every other player in that score group, then that player must be 'floated' to the next score group.

1.6. The final ranking order is determined by the aggregate of points scored, 1 point for a win, 0.5 points for a draw, and 0 points for a loss. A player whose opponent fails to appear for a scheduled game receives one point.

2. General Pairing Procedure

2.1. Before the pairings are made for the first round the list of participants is prepared, with the players ranked according to their ratings: No. 1 is the highest ranked with the highest rating. Players with the same rating or without FIDE rating are placed in order of FIDE title, perhaps local rating, and then lot.

2.2. The pairings for round one, for example in a tournament of 64 players, would be either: 1 v 33, 34 v 2, 3 v 35, 36 v 4, ..., or 33 v 1, 2 v 34, 35 v 3, 4 v 36, ... The color of the first pairing is decided by lot. This is the only occasion when colors need to be determined by drawing lots.

2.3. For all subsequent rounds players are divided into groups with the same score.

2.4. Pairings are started from the highest group, and proceed down to just above the median group. Then the pairings should be completed, working from the bottom group upwards.

2.5. The 'floater' (see 1.5) is a player who is paired outside his score group.

2.5.1. If there is a choice as to which player floats to a lower group, the player chosen is the one who best balances the remaining group into players seeking white and players seeking black. If there is still a choice, the player is chosen who has the lowest ranking.

2.5.2. If there is a choice as to which player floats to a higher group, the player chosen is the one who best balances the remaining group into players seeking white and players seeking black. If there is still a choice, the player is chosen who has the highest ranking.

2.6. After a player or players have been floated out of a score group, an even number of players will remain in that group. The group is then paired in accordance with the basic principles (articles 1.2. and 1.4.) in ranking order, top half against bottom half as far as possible.

2.7. If an intended floater cannot be paired with a compatible player in the next score group, he will if possible be switched for another player, otherwise he will be floated further.

2.8.1. If there are two or more downward floaters within any group, the floater with the highest score is paired first, or the highest ranked floater if scores are equal.

2.8.2. If there are two or more upward floaters within any group, the floater with the lowest score is paired first, or the lowest ranked floater if scores are equal.

3. Allocation of Colors

3.1. Colors are allocated as part of the pairing process.

3.2. If both players in a pairing had the same color in the previous round, then the colors they had in the round previous to that shall decide, and so on.

3.3. If both players in a pairing had identical histories of color allocation then the higher ranked player is given the color that will help to equalize his color allocation, however, if the players had equal allocation of colors (for example 3W and 3B) then the higher ranked player will have the alternating color.

REFERENCES

1. E. L. Lawler (1976) *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart & Winston, New York.
2. J. Edmonds (1965) Path, trees and flowers. *Can. J. Math.* **17,** 449–467.
3. Z. V. I. Galil (1986) Efficient algorithms for finding maximum matching in graphs. *Computing Surveys* **18**, 23–38.