Final Version 1: 2009-04-30

AGA Swiss McMahon Pairing Protocol Standards

This document describes the Swiss McMahon pairing system used by the American Go Association (AGA). For questions related to user interfaces and reporting standards, refer to the AGA document "AGA Swiss McMahon System: Program Interface and Reporting Requirements."

Initialization

Ordering the players

Players are ordered by rating. If no AGA rating is available, the best available substitute shall be used. The Tournament Director (TD) may adjust the ordering according to the known or estimated differences between different rating systems.

Setting the Bar

In a Swiss McMahon tournament, all players above a certain strength play in the top group. This strength threshold is called "the Bar," which should be set so that all players who have a realistic chance of winning the tournament are included in the top group.

The Tournament Director is responsible for setting the Bar, and in theory, it need not be announced in advance. In practice, it is usually determined and announced ahead of time, and is based on a rating equivalent to a rank, e.g., 7 dan and higher. At their discretion, Tournament Directors may allow lower rated players into the top band ("playing above the Bar").

Band construction

Players in a Swiss McMahon tournament are grouped by strength into bands. Depending on the size of the tournament, a band may contain players of only one rank or it may contain players having several different ranks (e.g., the 1-3 kyu band).

The usual separation between bands is one McMahon point, but this is increased when a band contains a large number of players. The separation between a given band and the **next higher** one is given by integer part of

$$S = D * \left(\frac{P}{R} \right),$$

where S is the band separation, P is the number of players in the band, R is the number of rounds in a tournament and D is a parameter. The value of S ranges between a minimum of 1 and a maximum of R/3, rounded up (e.g.., a maximum separation of 2 for a 6 round tournament). The value of D defaults to 0.75 but may be adjusted by the TD provided it does not conflict with the limitations on the value of S.

Band construction should be performed automatically and then presented to the TD for editing prior to the start of the first round of the tournament. The TD should be able to merge bands and manually alter individual band separations, as well as move individual players to other bands.

Assigning initial McMahon scores

Once bands are created and the band separation is calculated, players are assigned initial McMahon scores based on their band placement. This can be done in two ways.

By tradition, the top group is assigned an initial score of 0, with values decreasing to negative initial scores in lower bands, according to the value of the band separation. While this method has been used successfully for many years, it has the disadvantage that the vast majority of players end up with final tournament scores that are negative. This can be discouraging to new players, whose scores will be the most negative in the field.

Accordingly, it is also possible to set the bottom score group at an initial value of 0 or slightly higher (recommended to allow for late arrivals ranked below the initial starting band), with values increasing from there.

Sample band calculation:

Consider a tournament that has six rounds (R=6) and uses a value of D=0.75, run by a TD who does not choose to enforce a single point separation between the top and second groups. For registration as shown below, the band structure would be:

Group	Size (P)	Separation = D·(P/R)	Initial Score (0 at top)	Initial Score (0 at bottom)
Open section	19 players	-	0	6
Second	17 players	0.75·(17/6) = 2.125 → 2	-2	4
Third	10 players	0.75·(10/6) = 1.25 → 1	-3	3
Fourth	11 players	0.75·(11/6) = 1.375 → 1	-4	2
Fifth	24 players	0.75·(24/6) = 3 but capped at (R/3)=2	-6	0

Pairing

The basic pairing algorithm is repeated for each round of the tournament.

1. Remove players receiving byes
2. Sort the field
3. Create score groups
4. Pair each score group

Remove Players Receiving Byes

For a variety of reasons, a player may ask not to play in a particular round. In such cases the player is not paired and is awarded a voluntary bye.

If there are an odd number of players remaining in the tournament after awarding all of the voluntary byes, it will not be possible to pair everyone in the field. In this case, one player must receive a forced bye. If the Tournament Director previously identified a “bye volunteer”, this player is awarded a forced bye and is not paired for the round. If no bye volunteer is available, then one will be selected during the pairing process.

Sort the Field

Players are sorted first by McMahon score and then by entry rating. Most player's McMahon scores will change during the tournament, see Game Results, below, for details.

Create Score Groups

All players with the same McMahon score constitute a score group. Programmers should not assume that a player’s McMahon score will be an integer; the Tournament Director may choose to award fractional points for draws. While draws in individual games of Go are unusual, draws are more common in a team competition.

Pair Each Score Group

A pairing should satisfy the following criteria. The first two criteria are mandatory, the second two are desirable.

1. Players do not play each other twice

2. A player is not assigned a second forced bye unless he volunteers for it. A player may ask for any number of voluntary byes, however.

3. A player should not be floated (see the section on Odd-Numbered Groups) twice in a row in the same direction. This rule does not apply in the final round of a tournament having fewer than six rounds nor to the last two rounds of a longer tournament.

4. A player should not be assigned the same color more than three times in a row, or for more than 50% of the rounds in a tournament in any case. Some Tournament Directors do not apply this principle, while others allow players to select color by *nigiri.*

Odd-Numbered Groups ("Floaters")

A score group must contain an even number of players to be paired. If the group has an odd number, a player from the next lower score group is "floated" up to make the group even. Subject to criterion 3 (see "Pairing Each Score Group"), the floater is selected as the strongest player in the next lower score group that will permit a valid pairing to be created.

When players with different McMahon scores are paired, the player with the greater score is termed a "down floater" and his opponent is termed an "up floater".

If the score group in question is the last remaining score group (i.e., the lowest score group) then a floater will not be available and a player must be chosen to receive a forced bye. The selected player is the one with lowest rating who has not already received a forced bye and who will allow a valid pairing to be created with the remaining players. If no such player exists in the lowest score group, the pairings in successively higher score groups are undone until a bye recipient can be found.

Basic Pairing

Score groups are paired from the top of the tournament to the bottom (i.e., in order of decreasing score). This is in contrast to the ordinary Swiss system, which pairs from the top down to the middle and then from the bottom up to the middle.

In the simplest case, a score group has an even number of players, none of whom have played each other. For this type of score group, a pairing is created with the split-and-shift method: the group is paired so that the top half of the score group plays the bottom half of the group. For example, the pairing for a score group having eight players would be:

Player 1	vs.	Player 5
Player 2	vs.	Player 6
Player 3	vs.	Player 7
Player 4	vs.	Player 8

For score groups that do not permit the above optimal pairing, alternate pairings are tried in the following order:

1. The positions of two players within their half of the score group are exchanged ("a transposition"). In the case above, exchanging the positions of players 7 and 8 would be an example of a transposition

2. The positions of two players within opposite halves of the score are exchanged ("a switch"). In the case above, exchanging the positions of players 2 and 6 would be an example of a switch

3. If the score group includes a floater, replace that floater with an alternate floater from the next lower score group

4. Relax criterion 4 above (color assignment)

5. Relax criterion 3 above (floating more than once in the same direction)

Alternate pairings are generated in the order specified, exhausting all possible pairings at one level before moving to the next. For example, all possible transpositions are attempted before considering a switch. In cases where an alternate pairing procedure has multiple choices available to create a valid pairing (e.g., any of several possible transpositions), the program chooses the option that affects the lowest-ranked player(s) in the score group.

If a valid pairing cannot be created according to these procedures, the score group is merged with the next lower score group and the pairing procedure is applied recursively. In such a case, the larger group is paired in such a way as to minimize the total McMahon score difference between the paired players while still producing a valid pairing.

Example

Consider the problem of pairing a score group that has a McMahon score of 1 and is composed of players A, B, C, D, E and F, where players A-D have all played each other. No combinations of transpositions or switches will allow a valid pairing to within this group. The group is merged with the next lower score group (i.e., McMahon score 0) and paired as a whole, with the most desirable pairing being one that still pairs as many players as possible against opponents having the same McMahon score. If this new score group now includes players G, H, I and J, the first pairing that would be considered would be:

Player A (score = 1)	Player E (score = 1)
Player B (score = 1)	Player F (score = 1)
Player C (score = 1)	Player G (score = 0)
Player D (score = 1)	Player H (score = 0)
Player I (score = 0)	Player J (score = 0)

Note that this provisional match pairs opponents with the same McMahon score as much as possible.

Because this procedure is applied recursively, it will eventually find a successful pairing or it will include all remaining players within the tournament. If all remaining players are included and a valid pairing still cannot be found, the recursive merging of score groups is undone and the pairing from the next higher score group is re-paired. Re-pairing may require using an alternate arrangement of transpositions and switches, using a different floater, or may require applying the recursive merging method to the higher score group. In all cases, the underlying principle for generating a pairing is one that least disturbs the normal ideal.

Manual pairing

For a variety of reasons it may be desirable to force or prohibit specific pairings. Although Tournament Directors should only resort to manual intervention as a last resort, pairing programs must provide the option for the TD to do this.

Additional considerations

Especially for tournaments at the local level, Tournament Directors may wish to avoid pairing players from the same family, club, city, state or country if possible. Avoiding handicap games may also be desirable. These restrictions should be applied only by switching or transposing players within a score group. These considerations must be dropped in the last half of the tournament (e.g., rounds 4-6 of a six round tournament).

Handicaps

Traditionally, all games within a band are played without a handicap. However, if player's ranks are not distributed uniformly, and there are large gaps in the player's ranks, the Tournament Director may choose to run a handicap tournament.

It is common for Tournament Directors to reduce handicaps by one or even two stones. Pairing programs must allow a tournament to run with full handicap, handicap minus 1, handicap minus 2 or no handicap (all games even). Programs must also have the option to ensure that all games involving players in the top band are even, regardless of the handicap settings for the remainder of the field. Programs must also provide the TD with the option to adjust the handicap for a particular game after pairing is complete.

Game Results

At the end of each round, the McMahon score of players who have won their games, including a wins by forfeit, is increased by 1. Scores of players who have lost their games remain the same. Players involved in games ending without result (e.g., jigo) each have their scores increased by 0.5. The scores of players who have received a double forfeit are not changed.

A player who receives a forced bye as a result of a tournament field having an odd number of players is credited with a victory and that player's McMahon score is increased by one. The McMahon scores of players receiving voluntary byes do not change.

Final Standings--Tiebreaks

The final order of the field is by McMahon score, and inevitably there will ties. While cash prizes can be split without difficulty, other prizes may have to be awarded by tiebreaks. A variety of tie breaking methods have been proposed, but the AGA uses four standard tie break procedures in the following order:

1. SOMS: The Sum of McMahon Scores for all of a player's opponents
2. SODOMS: The Sum of Defeated Opponents McMahon Scores
3. Face to face result (if applicable).
4. Random draw

The Swiss McMahon pairing approach is designed with the expectation that the SOMS, SODOMS tiebreak system will be used and the rules are specifically formulated to make those distinctions as significant as possible. Alternative methods may be desirable for many reasons and are permitted. Manual pairing may be guided by the TD's understanding of the tie breaking system being used.

If a player involved in a tie break situation has received a forced bye, the bye is treated as a victory against a phantom player who started with the same initial McMahon score as the tied player and who won half his games in the tournament. Voluntary byes are treated in the same manner, except that the voluntary bye is treated as a loss against the same phantom player.

In the case of a player involved in a tiebreak situation who had one or more opponents who received a voluntary bye or a double forfeit, the SOMS and (when the relevant opponent was defeated) SODOMS scores are to be credited with 0.5 McMahon points per instance.

Example

Bob the Beginner plays in a 5-round weekend tournament. He plays in the lowest division of the tournament, which happens to have an initial McMahon score of -8. His tournament results are:

Round	Opponent	Result
1	Barry	Win
2	Forced bye	-
3	Carmilla	Win
4	Sally	Loss
5	Marvin	Win

Bob's final McMahon score at the end of the tournament is

$$Initial\ McMahon\ Score + 3\ Wins + Forced\ Bye = -8 + 3 + 1 = -4$$

At the end of the tournament, Bob ends up being tied for first place in his division. For the purposes of the tiebreak calculation, the forced bye in round two is treated as if Bob played against a phantom player. The phantom player is assumed to have started with the same initial McMahon score as Bob (-8) and scored wins in exactly half of the tournament rounds for a final McMahon score of

$$Initial\ McMahon\ Score + 0.5 * Number\ of\ Rounds = -8 + 5/2 = -5.5$$

This score is credited to Bob's SOMS. Because this is a forced bye, the game is treated as a win and Bob receives a credit of -5.5 towards his SODOMS score as well.

If Bob's second round was a voluntary bye rather than a forced bye, the second round game would be treated as a loss rather than a win and Bob's McMahon score a the end of the tournament would be

$$Initial\ McMahon\ Score + 3\ Wins + Involuntary\ Bye = -8 + 3 + 0 = -5$$

Now consider the case where one of Bob's opponent (say Sally) is involved in a tie. In this case Bob's tournament results will be used in the tiebreak calculations. For the case where Bob received a forced bye, Bob's final McMahon score is -4 and this value is used in calculating his opponent's tiebreak scores.

If Bob took a voluntary bye in the second round, however, he would only contribute his final score of -5 to Sally's tiebreak values. To avoid penalizing Sally for the game Bob missed, an extra 0.5 McMahon point bonus is added to her tiebreak score. In effect, Bob is assumed to have an effective final McMahon score of

$$Final\ McMahon\ Score + Number\ of\ Voluntary\ Byes/2 = -5 + 1/2 = -4.5$$

for the purpose of calculating Sally's tiebreak score.

Note: The effective McMahon score is *only* for the purposes of calculating Sally's tiebreak score. Under no circumstances does the tiebreak credit apply to Bob when determining his final placing in the tournament.

Program Implementation Notes and References

There are at least three approaches to implementing a tournament computer pairing algorithm. The first involves a rules-based approach. Programmers interested in this method may wish to examine the wide variety of chess pairing programs that has been developed.

The second approach involves describing a penalty value for every possible pairing between two players in a tournament field. The pairing problem then becomes one of minimizing the total penalty across the entire field, which in the language of mathematical graph theory is the Minimum (or Maximum) Weight Perfect Matching problem. The technique is described in S. Olafsson, *Matching in Chess Tournaments*, J. Oper. Res. Soc., Vol. 41, No. 1, pp 17-24 (1990). This is a novel enough approach that all programmers are strongly encouraged to consult it before starting to write a pairing program, regardless of whether they choose to use this method.

This technique has been used successfully for almost two decades and has several advantages. First, the weighted perfect matching algorithm can be solved using a polynomial-time algorithm. This algorithm is guaranteed to produce a pairing if one exists, and furthermore, the result pairing is optimal. The major down side of using the weighted perfect matching algorithm is its complexity. Although the algorithm itself dates back to the 1960s, it has number of subtleties and it is not recommend trying to code it yourself.

Fortunately, there are several implementations of the weighted perfect matching algorithm:

1. A piece of academic demonstration code:
http://elib.zib.de/pub/Packages/mathprog/matching/weighted/
   This code does not have a distribution license and is suitable only for illustrative purposes.
2. The LEMON Graph Library: http://lemon.cs.elte.hu/trac/lemon
   This library has a weighted matching algorithm that is fast enough to pair a 2000 player tournament in under a minute.
3. The Goblin Graph Library: http://goblin2.sourceforge.net/
   This solver is based on a network-flow solution, and is noticeably slower than the other choices.

The asymptotic complexity of the basic weighted matching algorithm is $O(N^4)$, and minor changes have improved this to $O(N^3)$. More advanced algorithms exist, but their value is questionable in this application. The $O(N^3)$ algorithm is sufficient to pair a tournament of any practical size on a modern computer.

The Go players of Europe have used the weighted perfect matching approach with success for many years. The first person to implement this method was Christophe Gerlach, who studied the technique as part of his diploma and coded a program called MacMahon. A copy of the diploma thesis (written in German) can be found at http://www.cgerlach.de/go/MacMahon.html.

An open-source (GPL) pairing program called OpenGotha has been written by French player Luc Vannier. The program is written in Java and contains weighted-matching code. It can be found at http://vannier.info/jeux/gotournaments/opengotha.htm. The program follows the European

implementation of the McMahon system, which is slightly different than the system described in this document.

Correctly implementing the penalty function in a weighted perfect matching system contains some subtleties. It is important to set various priorities at different numerical scales. For example, the most important consideration is to minimize the McMahon score difference between opponents, with the split-and-shift pairing system coming as a later concern. This could be accomplished by defining a penalty function by

$$P_{ij} = 10000(\Delta M_{ij})^2 + 1(\Delta P_{ij})^2$$

where ΔM_{ij} is the difference in McMahon scores between players i and j. ΔP_{ij} reflects the distance of player j from player i's optimal opponent, based on the sorted order of players in the tournament. For example, in the case of the score group of eight players show below, $\Delta P_{15} = 0$, $\Delta P_{16} = 1$, $\Delta P_{17} = 2$, etc.

Player 1	vs.	Player 5
Player 2	vs.	Player 6
Player 3	vs.	Player 7
Player 4	vs.	Player 8

Other constraints, such avoiding pairing of players from the same state, would be assigned a penalty on a scale midway between that of ΔM_{ij} and ΔP_{ij}. The matching algorithm would then find it advantageous to remove the same-state penalty by increasing some of the ΔP_{ij} penalties associated with transposing or switching players within the same score group, but not so advantageous as to move players to other score groups. We refer readers to examine the OpenGotha program, the discussion in Christophe Gerlach's thesis and the Olafsson paper.

A third approach that is referenced for completeness uses the “stable roommates” algorithm. The method is described in E. Kujansuu, T. Lindberg and E. Mäkinen, *The Stable Roommates Problem and Chess Tournament Pairings*, Divulgaciones Matemáticas, Vol. 7, No. 1 (1999), pp. 19-28.