Mario Held

Tournament development

with the

FIDE (Dutch) System

An example of a six-round tournament

(With Baku 2016 FIDE C.04 Swiss Rules)

Ver. 071120161200D

CONTENTS

PART ONE – THE RULES

ANNOTATED RULES FOR THE FIDE (DUTCH) SWISS SYSTEM

Hereafter, we present the general rules for Swiss Systems (FIDE Handbook C.04.1 and C.04.2) and the Rules for the FIDE (Dutch) System (FIDE Handbook C.04.3), together with some notes to explain them.

The first part contains rules that define the technical requirements any Swiss pairing system must obey, whilst the second part targets a set of various aspects relating to the handling of tournaments, from the fairness of the systems to the management of late entrants, and several rules that are common to all the FIDE approved systems.

The third part contains the Rules for the FIDE (Dutch) Swiss System, which in its turn is comprised of the following sections:

(A) Introductory Remarks and Definitions: containing the basic concepts about the system and its control variables; namely, the last paragraph (A.9) is an essential description of the pairing process that will be described and regulated in detail by section (B).

(B) Pairing process for a bracket: this section explains how to build a candidate pairing, determine its quality (checking it against the pairing criteria), and, when necessary, improve the quality of the pairing (by looking for better candidates).

(C) Pairing Criteria: defining limitations to the possible pairings of the players. Some of those limitations are common to all Swiss pairing systems, while others are specific to the FIDE (Dutch) system and give origin to some of its peculiarities.

(D) Rules for the sequential generation of the pairings: this section defines the transposition and exchange procedures, showing how to "stir" the players list when natural pairing is not possible (because two players have already played against each other, or because of colours incompatibility, and so on)

(E) Colour Allocation Rules: after the completion of the pairing, each player receives its colour according to these rules.

With reference to previous versions, the FIDE (Dutch) rules have been almost completely reworded, in order to make them simpler and more intuitive. The algorithm, which used to occupy the whole section C, has now been completely evicted from the rules, together with the whole old section B. Instead of the latter, a new section C contains a revised list of the pairing quality criteria, which is both more detailed and clearer than the previous one.

For all this rewording, the real changes in the pairings address only a few cases, while a vast majority of the pairings remain just the same as they were with the previous rules[1].

We would like to suggest you to carefully study the Rules until you feel you master their principles and meanings, before starting to study the tournament example.

---

[1] *Readers may find detailed information about those changes in the documents relating Abu Dhabi 2015 and Baku 2016 FIDE Congresses, in the FIDE Swiss Pairings Program Commission website* pairings.fide.com .

C.04 FIDE SWISS RULES

C.04.1 BASIC RULES FOR SWISS SYSTEMS

The following rules are valid for each Swiss system unless explicitly stated otherwise.

a. The number of rounds to be played is declared beforehand.

After the start of the tournament, we are not allowed to change the number of rounds (however, this may become inevitable by force of circumstances).

b. Two players shall not play each other more than once.

This is the only principle of Swiss Systems we cannot dispense with (unless doing differently is absolutely inevitable...)!

c. Should the number of players to be paired be odd, one player is unpaired. This player receives a pairing-allocated bye: no opponent, no colour and as many points as are rewarded for a win, unless the rules of the tournament state otherwise.

Please note that this rule allows event organizers to establish a different value for byes (e.g. half a point) instead of the usual whole point.

d. A player who has already received a pairing-allocated bye, or has already scored a (forfeit) win due to an opponent not appearing in time, shall not receive the pairing-allocated bye.

However, and whatever its value is, a pairing allocated bye ("PAB") cannot be assigned to any player who has already received a previous one, or a forfeit win. The allocation of a PAB, though, is not prevented by a previous bye "on request"[2] (when such a provision is permitted by the tournament rules).

e. In general, players are paired to others with the same score.

The location of this principle before colour balancing rules highlights its greater importance with respect to the latter. It is because of this rule that we cannot make players float to suit colour preferences that are not absolute (see C.04.3:A.6.a).

f. For each player the difference of the number of black and the number of white games shall not be greater than 2 or less than –2.

Each system may have exceptions to this rule in the last round of a tournament.

We should emphasize that the exceptions to rules f and g for the last round are possible, <u>but not compulsory</u>. The FIDE (Dutch) system adopts them, tough in practice only when there are very good reasons to do so. Other systems do not do the same - e.g., the Dubov Swiss System definitely refuses to make such exceptions, which seem not to be consistent with the basic principles of that system.

g. No player will receive the same colour three times in a row.

Each system may have exceptions to this rule in the last round of a tournament.

h. 1. In general, a player is given the colour with which he played less games.

This rule warrants the good colour balancing typical of all FIDE approved Swiss Systems.

---

[2] *In the previous versions of the Swiss Rules, any number of points got without playing, like e.g. a requested "Half Point Bye", <u>did</u> prevent the allocation of a PAB.*

2. If colours are already balanced, then, in general, the player is given the colour that alternates from the last one with which he played.

i. The pairing rules must be such transparent that the person who is in charge for the pairing can explain them.

Sometimes, players ask the Arbiter to justify, or explain, the pairings, which, nowadays, are most usually prepared with the help of a software program (which should be a FIDE endorsed one, if only possible). However, we want to remember that, even if the pairings are made by means of a computer, it is always the arbiter who takes responsibility for the pairing, not the software.

C.04.2 GENERAL HANDLING RULES FOR SWISS TOURNAMENTS

A Pairing Systems

1. The pairing system used for a FIDE rated tournament shall be either one of the published FIDE Swiss Systems or a detailed written description of the rules shall be explicitly presented to the participants.

All the rules in this section tend to the same aim: to prevent any possible tampering with the pairings in favour of one or more participants (such as helping a player to obtain a norm). To this effect, the pairing rules must be well specified, transparent, and unambiguous in the first place.

2. While reporting a tournament to FIDE, the Arbiter shall declare which of the official FIDE Swiss systems was used. If another system was used, the Arbiter shall submit the rules of this system for checking by the Systems of Pairings and Programs Commission (SPPC).
3. Accelerated methods are acceptable if they were announced in advance by the organizer and are not biased in favour of any player.
4. The FIDE Swiss Rules pair the players in an objective and impartial way, and different arbiters or software programs following the pairing rules should arrive at identical pairings.
5. It is not allowed to alter the correct pairings in favour of any player.

   Where it can be shown that modifications of the original pairings were made to help a player achieve a norm or a direct title, a report may be submitted to the Qualification Commission to initiate disciplinary measures through the Ethics Commission.

B Initial Order

1. Before the start of the tournament a measure of the player's strength is assigned to each player. The strength is usually represented by

The fundamental principle of all Swiss systems is to pair tied players (i.e. players with the same number of points) so that, in the top echelon, the number of ties is halved at every round. Thus, in a tournament with T rounds, if the rating lists of the players. If one rating list is available for all participating players, then this rating list should be used.

number N of players is less than 2^T [i.e. $T \geq \log_2 (N)$], we should (theoretically) have no ties for the first place.

However, practice shows that, to reach this goal in a real environment (which includes draws and unexpected results), a precise evaluation of the strength of players is essential.

2. It is advisable to check all ratings supplied by players. If no reliable rating is known for a player the arbiters should make an estimation of it as accurately as possible.

When no better information is available, the estimated rating of an unknown player can be determined based on a national rating (if available) using the appropriate conversion formulas; or other rating lists, tranches, tournament results and so on may be used, if reliable. In conclusion, the Arbiter shall have to use sound judgment and reasoning, to obtain the best possible evaluation with what data is available.

3. Before the first round the players are ranked in order of, respectively:

   [a] Strength (rating)

   [b] FIDE title (GM - IM - WGM - FM - WIM - CM - WFM - WCM - no title)

   [c] alphabetically (unless it has been previously stated that this criterion has been replaced by another one)

FIDE titles are ordered by descending nominal rating; when ratings are equal, titles obtained through norms take precedence with respect to automatic ones.

Alphabetical sorting is unessential, its only rationale being that of ensuring an unambiguous order. Thus, this criterion can be substituted for by <u>any</u> other sorting method capable of giving an unambiguous order, provided this method has been previously declared in the tournament regulations.

4. This ranking is used to determine the pairing numbers; the highest one gets #1 etc.

   If, for any reason, the data used to determine the rankings were not correct, they can be adjusted at any time. The pairing numbers may be reassigned accordingly to the corrections, but only for the first three rounds. No modification of a pairing number is allowed after the fourth round.

Please notice that a lower numeric value corresponds to a higher ranking; this choice may not seem "natural", but it is deeply rooted in common language by now.

Pairing numbers are used by all Swiss pairing systems except Dubov. Thus, a change in pairing numbers changes the pairings too. We would expect this to happen, if at all, in the first round of a tournament - in some (rare) instances even in the second or in the third round - and, when such changes happen, they make the checking of the pairings rather difficult. Hence, in order to make it easier to perform such checks on advanced stages of a tournament, the rule prohibits late changes of the pairing numbers.

As correct ratings, titles and so on are needed to correctly rate the tournament, such data may always be corrected, even in late rounds (and even after the tournament is finished!), but without changing the pairing numbers.

C Late Entries

1. According to FIDE Tournament Rules, any prospective participant who has not arrived at the venue of a FIDE competition before the time scheduled for the drawing of lots shall be excluded from the tournament unless he shows up at the venue in time before a pairing of another round.

   An exception may be made in the case of a

It seems appropriate to point out that the declaration of delay must be given in advance, in writing, and stating reasons for it. Verbal communications (telephone, etc.) do not suffice. Since exceptions <u>may</u> be made, it is the Arbiter's responsibility to grant or decline such requests.

registered participant who has given written notice in advance that he will be unavoidably late.

2. Where the Chief Arbiter decides to admit a latecomer,
   - if the player's notified time of arrival is in time for the start of the first round, the player is given a pairing number and paired in the usual way.
   - if the player's notified time of arrival is in time only for the start of the second (or third) round ("Late Entry"), then the player is not paired for the rounds which he cannot play. Instead, he receives no points for unplayed rounds (unless the rules of the tournament say otherwise), and is given an appropriate pairing number and paired only when he actually arrives.

We want to take notice that the admission of a latecomer is a choice of the Chief Arbiter, who takes the final decision – and must take the responsibility too, especially if during the round there are empty seats... Thus, before accepting a latecomer and making the actual pairing, we want to be very sure that the player will actually be there in time to play. If we are not that sure, it is probably better to let the player enter the tournament, and be paired, only for a subsequent (second, third) round.

3. If there are late entries, the Pairing Numbers that were given at the start of the tournament are considered provisional. The definitive Pairing Numbers are given only when the List of Participants is closed, and corrections made accordingly in the results charts.

Entering a late player in the tournament causes the pairing numbers to change according to the new ranking list; some of the players will thus play the following rounds with a different pairing number, and this may cause some perplexity among the players. For example, consider a player, correctly registered from the beginning, but entering a tournament (say, with 100 players) on the second round, as #31. In the first round that player had no pairing number – hence, the players who (now) have numbers 33, 35, 37 and so on, in the first round had even pairing numbers and thus the colour opposite to that of player #1.

By the way, we should also observe that the limit imposed in C.04.2.B.4 on the regeneration of pairing numbers does <u>not</u> extend to the case of a newly added late player.

D Pairing, colour and publishing rules

1. Adjourned games are considered draws for pairing purposes only.
2. A player who is absent without notifying the arbiter will be considered as withdrawn unless the absence is explained with acceptable arguments before the next pairing is published.
3. Players who withdraw from the tournament will no longer be paired.
4. Players known in advance not to play in a particular round are not paired in that round and score zero (unless the rules of the tournament say otherwise).
5. Only played games count in situations where

Basically, we look only at <u>actually played</u> games, skipping the colour sequence is meaningful. So, for instance, a player with a colour history of BWB=W (i.e. no valid game in round-4) will be treated as if his colour history was =BWBW. WB=WB will count as =WBWB, BWW=B=W as = =BWWBW and so on.

"holes", which float to the top of the list. Thus, for example, in the comparison between the colours histories of two players, the sequence == WB is equivalent to =W=B and WB== (and the latter two are equivalent to each other!).

6. Two paired players, who did not play their game, may be paired together in a future round.

7. The results of a round shall be published at the usual place of communication at announced time due to the schedule of the tournament.

The application of this rule and the next requires us to set (and post!) a timetable for the publication of pairings. Above all, these rules put a constraint on the possible revision of the pairings: if an error is not reported within the specified deadline, all subsequent pairings, as well as the final standings, shall be prepared making use of the wrong result as if it were correct.

8. If either
   - a result was written down incorrectly, or
   - a game was played with the wrong colours, or
   - a player's rating has to be corrected (and playing numbers possibly recomputed as in C.04.2.C.3),

   and a player communicates this to the arbiter within a given deadline after publication of results, the new information shall be used for the standings and the pairings of the next round. The deadline shall be fixed in advance according to the timetable of the tournament.

   If the error notification is made after the pairing but before the end of the next round, it will affect the next pairing to be done.

   If the error notification is made after the end of the next round, the correction will be made after the tournament for submission to rating evaluation only.

9. After a pairing is complete, sort the pairs before publishing them.

   The sorting criteria are (with descending priority):
   - the score of the higher ranked player of the involved pair;
   - the sum of the scores of both players of the involved pair;
   - the rank according to the Initial Order (C.04.2.B) of the higher ranked player of the involved pair.

Even when using a pairing software program, it is mostly advisable to check boards order before publishing the pairing, because many players interpret even an incorrect board order as a "pairing error".

10. Once published, the pairings shall not be changed unless they are found to violate C.04.1.b (*Two players shall not play against each other more than once*).

C.04.3 FIDE (DUTCH) SYSTEM

Version approved at the 87th FIDE Congress in Baku 2016.

A) INTRODUCTORY REMARKS AND DEFINITIONS

A.1 Initial ranking list

See C.04.2.B (General Handling Rules - Initial order)

A.2 Order

For pairings purposes only, the players are ranked in order of, respectively:

a. score

b. pairing numbers assigned to the players accordingly to the initial ranking list and subsequent modifications depending on possible late entries or rating adjustments

Players are ordered in such a way that their presumable strengths are likely to decrease from top to bottom of the list (see also C.04.2:B).

Please notice that when we include a late entry, the list should be sorted again, thus assigning new pairing numbers to the players (C.04.2:C.2,3). The same may be done when some wrongly entered rating had to be corrected. When this happens, some participants may play subsequent rounds with new, different numbers; and, of course, this change may, if not adequately advertised, muddle players who, in reading the pairings, still look for their old numbers.

A.3 Scoregroups and pairing brackets

A scoregroup is normally composed of (all) the players with the same score. The only exception is the special "collapsed" scoregroup defined in A.9.

A (pairing) bracket is a group of players to be paired. It is composed of players coming from one same scoregroup (called resident players) and of players who remained unpaired after the pairing of the previous bracket.

This definition solves any ambiguity between scoregroups and pairing brackets, stating that the scoregroup is the "backbone" of a pairing bracket, which is made of a scoregroup together with the players remaining from the pairing of the previous bracket. The players from the scoregroup are called "resident", and usually have all the same score, which is called resident score and is the "nominal score" of the bracket. Only when the scoregroup is the "Special collapsed" one, the resident players may have different scores.

A (pairing) bracket is homogeneous if all the players have the same score; otherwise it is heterogeneous.

The difference is that in a homogeneous bracket there are no score differences between players to be taken care of (to be homogeneous, a bracket must be made of just a (normal) scoregroup and nothing more).

A remainder (pairing bracket) is a sub-bracket of a heterogeneous bracket, containing some of its

Article B.3 illustrates how to build a candidate pairing for a bracket and explains how and when a remainder is built and used.

resident players *(see B.3 for further details).*

A.4 Floaters and floats

a A downfloater is a player who remains unpaired in a bracket, and is thus moved to the next bracket.

In the destination bracket, such players are called "moved-down players" (MDPs for short).

A player may become a downfloater because of several reasons; first, the bracket may contain an odd number of players, so that one shall unavoidably remain unpaired. Then, the player may have no possible opponent (and hence no legal pairing) in the bracket. Sometimes, two or more players share between them a number of possible opponents in such a way that no player is incompatible, but we cannot pair all of them (e.g., two players with only one possible opponent, three players with only two possible opponents, and so on)[3]. Last, but not least, in some instances the player may have to float down, in order to allow the pairing of the following bracket.

In analogy to "downfloater", we will use the term "upfloater" to indicate a player paired to another one having a higher score[4] (usually, the opponent of a downfloater).

b After two players with different scores have played each other in a round, the higher ranked player receives a downfloat, the lower one an upfloat.

A player who, for whatever reason, does not play in a round, also receives a downfloat.

Downfloats and upfloats are a sort of markers, used to record previous unequal pairings of the player. The reason to keep track of such pairings is that, in general, we want to minimise, and, as far as possible, avoid, their occurrence for the same players. Actually, a pairing between floaters constitutes a disturbance to the general principle of Swiss systems that the players in a pair should have the same score, and therefore the rule try to limit the repetition of such events[5].

We want to notice that <u>any</u> player who did not play a round <u>receives a downfloat</u>. This is important because it affects the following two pairings for that player. For example, it becomes unlikely that such a player may receive a downfloat or get the PAB [A.5] in the next round[6].

A.5 Byes

See C.04.1.c (*Should the number of players to be paired be odd, one player is unpaired. This player receives a pairing-allocated bye: no opponent, no colour and as many points as are rewarded for a win, unless the regulations of the tournament state otherwise*).

In other Swiss systems (e.g. Dubov) the player, whom the PAB will be assigned to, is selected <u>before</u> starting the pairing for the round.

In the FIDE (Dutch) system, on the contrary, the round-pairing (see A.9) <u>ends up</u> with an unpaired player, who will receive the pairing-allocated bye (PAB).

---

3 *This situation is sometimes (unofficially) called semi-incompatibility or island-(in)compatibility.*

4 *Please notice that in other Swiss pairing systems (e.g. Dubov), the same term "upfloater" may indicate a player transferred to a higher bracket.*

5 *We may also note that the FIDE (Dutch) system uses a "local" approach to this problem, which looks only to the last two rounds. On the contrary, the Dubov system adopts also a "global" approach, putting also a limit on the total number of floats in the whole tournament (three floats for tournaments up to nine rounds, four for longer tournaments).*

6 *On the contrary, the previous rules did not assign a downfloat to a player who forfeited a game, so such players had no protection against getting a PAB or a downfloat in the following round. Because of this, a weak player absent in the first round could get a PAB in the second round.*

A.6 Colour differences and colour preferences

The colour difference of a player is the number of games played with white minus the number of games played with black by this player.

During pairing, we will try to accommodate (as much as possible) the colour preferences of the players – and this is the reason for the good balance of colours of Swiss modern systems.

The colour preference is the colour that a player should ideally receive for the next game. It can be determined for each player who has played at least one game.

Participants, who have not played any games yet, just have no preference, and shall therefore accept any colour (see A.6.d).

a. An <u>*absolute colour preference*</u> occurs when a player's colour difference is greater than +1 or less than -1, or when a player had the same colour in the two latest rounds he played. The preference is white when the colour difference is less than -1 or when the last two games were played with black. The preference is black when the colour difference is greater than +1, or when the last two games were played with white.

In general, the colour difference should not become greater than 2 or less than -2 – with the possible exception of high ranked players in the last round, which can receive, if necessary, the third colour in a row or a colour three times more than the opposite (but this is still a relatively rare event).

To determine an absolute colour preference, we examine only the <u>actually played</u> rounds, skipping any unplayed games[7] (whatever the reason may be) in compliance with [C.04.2:D.5] (e.g., the sequence WBBW=W gives an absolute colour preference).

b. A <u>*strong colour preference*</u> occurs when a player's colour difference is +1 (preference for black) or -1 (preference for white).

Notice that any disregarded colour preference, be it strong or mild, will give origin to an absolute colour preference on the subsequent round.

c. A <u>*mild colour preference*</u> occurs when a player's colour difference is zero, the preference being to alternate the colour with respect to the previous game he played.

d. Players who did not play any games have no colour preference (the preference of their opponents is granted).

If neither player has a colour preference (as is normal when pairing the first round, but may sometimes happen also in subsequent rounds), we resort to the colour allocation rules in section E. There, by means of the initial-colour (decided by drawing of lots before the pairing of the first round) and of rule E.5, we will be able to assign the correct colour to both players.

A.7 Topscorers

Topscorers are players who have a score of over 50% of the maximum possible score when pairing the final round of the tournament.

Such high-scoring players are especially important in the determination of the winner and of the top ranking[8]. Hence, we may apply some special treatment criteria to their pairings - e.g., a player may receive a same colour three times more than the other one, or three times in a row, if this is needed to make it meet an opponent better

---

[7] *Please note the difference with floats, for which we look at the last two rounds of the tournament schedule (but remember that an unplayed game gives a downfloat).*

[8] *Not all the "topscorers" are really competing for top ranking places; nonetheless, they are more likely to be of importance in the formation of the top standings than low-ranked players, in several collateral ways – e.g. they may be opponents to prospective prize winners, or their score may give a determinant contribute in tiebreak calculations, and so on.*

suited to the strength the player demonstrated.

A.8 Pairing Score Difference (PSD)

The pairing of a bracket is composed of pairs and downfloaters.

This is an important idea: the pairing of a bracket is not made only of pairs: the downfloaters are part of it too – and a very important part, at that! In fact, as we shall see, the choice of the downfloaters may determine if it will be possible to pair the remaining players – and therefore if the pairing is a valid one.

Its Pairing Score Difference is a list of score-differences (SD, see below), sorted from the highest to the lowest.

For each pair in a pairing, the SD is defined as the absolute value of the difference between the scores of the two players who constitute the pair.

For each downfloater, the SD is defined as the difference between the score of the downfloater, and an artificial value that is one point less than the score of the lowest ranked player of the current bracket (even when this yields a negative value).

Note*: The artificial value defined above was chosen in order to be strictly less than the lowest score of the bracket, and generic enough to work with different scoring-point systems and in presence of non-existent, empty or sparsely populated brackets that may follow the current one.*

The Pairing Score Difference allows the best management of the overall difference in scores between the paired players. In practice, it is a list of the score differences, built as follows: we calculate the score differences (SD) in each pair and for each downfloater, then sort them from higher to lower, thus obtaining a string of numbers. Each single difference is taken in absolute value (so that it is always positive), because it's irrelevant which one of the players have a higher score.

While the meaning of the SD is obvious for pairs, it is far less obvious for downfloaters, who have no opponent yet. Nonetheless, we need to account, somehow, for the perspective score difference relative to the player when it will finally be paired - in such a way that giving a float, or a PAB, to a higher scored player should be worse than giving it to a lower scored one. So we go for a "presumptive" score difference, establishing a hypothetical score for the residents of the (yet undefined!) next bracket.

In order to be sure that we can accommodate a wide variety of possible next brackets, we choose a value lower enough than that of the current bracket, namely one point less than the minimum score of its (resident) players. In the last two brackets, this may yield a negative value – e.g., in the 0.5 points bracket this value is -0.5 points. This is not a problem, as we will simply take the difference between a positive value and this one, so the result will always be positive.

Please note that in the last bracket the only possible downfloater is the player who is going to get the PAB. Thus, this calculation provides an easy and uniform way to minimise the score of the players who get the PAB.

PSD(s) are compared lexicographically *(i.e. their respective SD(s) are compared one by one from first to last - in the first corresponding SD(s) that are different, the smallest one defines the lower PSD).*

PSDs are compared following the lexicographical order (the "order of the dictionary"). We start by comparing the first number of the first PSD with the first number of the second PSD: if one of those two is smaller than the other one, the PSD it belongs to is the "smaller". If they are equal, we proceed to the second element of each PSD, and repeat the comparison. Then, if needed, we go on to the third, the fourth, and so on - until we reach the end of the strings[9].

An alternative (but fully equivalent) method of comparison is the following: substitute a letter for each number of each PSD, following the correspondence A=0, B=0.5,

[9] *Of course, this method only has significance if the two PSD have the same length; but this is always the case, because the PSD comparison is used only when pairings with the same number of pairs are involved. Were the number of pairs different, we would never get to a PSD comparison.*

C=1, D=1.5, E=2 and so on. Doing so, we transform the PSDs in alphabetical words, which can be compared using the simple alphabetical order. The word that comes first (alphabetically) corresponds to the "smaller" PSD.

A.9 Round-Pairing Outlook

This article is essentially a guideline giving a panoramic vision of the pairing process, both in the more common case in which the pairing can be completed by normal means, and in the special case in which this is not possible. This is a very important thing to do, as the new Rules do not any more contain an algorithm to dictate a step-by-step procedure.

The pairing of a round (called round-pairing) is complete if all the players (except at most one, who receives the pairing-allocated bye) have been paired and the absolute criteria C1-C3 have been complied with.

We want to notice that that this definition refers not to a bracket but to the complete round. Thus, we cannot accept unpaired players (apart from a possible PAB) - all players must be paired. On the other hand, the constraints for such a pairing are very loose, not to say minimal – we are only asking for it to comply with the absolute criteria. This does not mean that we may feel free to make a poor pairing: in general, several complete pairings will be possible for each round, and "the" pairing – the correct one - shall simply be the one among them that best satisfies all the pairing criteria.

If it is impossible to complete a round-pairing, the arbiter shall decide what to do.

This is something really brand-new: for the first time ever, the case in which no pairing at all can be done is referred to by the rules. From a practical point of view, this is not a very helpful rule - in fact, in these (luckily rare) cases, the arbiters must act according to their best judgment – but, at least, the possibility has been accounted for.

Otherwise, the pairing process starts with the top scoregroup, and continues bracket by bracket until all the scoregroups, in descending order, have been used and the round-pairing is complete.

The pairing process starts with the topmost scoregroup; with it, we build the first bracket and try to pair it. This pairing may possibly leave some downfloaters that, together with the next scoregroup, will form the next bracket, and so forth – until all players have been paired.

However, if, during this process, the downfloaters (possibly none) produced by the bracket just paired, together with all the remaining players, do not allow the completion of the round-pairing, a different processing route is followed.

Before starting the pairing of a bracket, we must verify that at least one legal pairing (i.e. a pairing that complies with the absolute criteria) exists for all the players as yet unpaired, together with the downfloaters (of course, possibly none) left from the bracket just paired[10]. This requirement is informally called the "Requirement Zero", and its check is called a "Completion test".

If this check fails before pairing the first bracket, there is no way at all to complete the round-pairing, so we have an impossible pairing - which is bad news.

When, on the contrary, this happens after the pairing of the first bracket, we already know that at least one legal pairing exists for the entire round (we checked this before pairing the first bracket!). Nevertheless, if the set formed by the downfloaters together with all of the remaining players cannot be paired, it means that, given those downfloaters, we cannot complete the pairing without infringing the absolute criteria.

In this situation, the pairing produced by the last (in fact, still current!) paired bracket is not adequate, and we need to modify it before proceeding. We must restart with this

[10] *Of course, this check is far simpler than the actual complete pairing, because (for the moment) we are not interested in finding the best (correct) pairing, but only in showing that at least a legal one exists.*

same bracket, while changing the pairing conditions, in order to be able to find the pairing (which, as we already know, must undoubtedly exist). This change of conditions may have two effects: the first, and less invasive, is a different choice of downfloaters[11], while the second is an increase in the number itself of downfloaters. (The latter is of course the only option available when the original pairing did not produce any floater.)

The last paired bracket is called Penultimate Pairing Bracket (PPB). The score of its resident players is called the "collapsing" score. All the players with a score lower than the collapsing score constitute the special "collapsed" scoregroup mentioned in A.3.

First, we pool together all the players, whose score is lower than the collapsing score. Then, with those players, we build the "special collapsed scoregroup" (SCS) - whose players are all resident, regardless of their score.

The bracket just tentatively paired, and which we are now going to pair again, is now called PPB[12].

The pairing process resumes with the re-pairing of the PPB. Its downfloaters, together with the players of the collapsed scoregroup, constitute the Collapsed Last Bracket (CLB), the pairing of which will complete the round-pairing.

The primary goal in pairing the PPB is to have it produce a set of downfloaters that allows a complete pairing of the SCS [C.4]. With those downfloaters, together with the SCS, we build the CLB, which is by definition the last bracket. The pairing of those two brackets requires some special attentions[13].

<u>Note</u>*: Independently from the route followed, the assignment of the pairing-allocated bye (see C.2) is part of the pairing of the last bracket.*

By stating that the assignment of the PAB is always part of the pairing of the last bracket, this note is telling us that criterion C.2, which regulates the assignment of the PAB, is only significant when the last bracket is in some way involved in the pairing – that is to say:

- *when pairing the last bracket (be it a normal bracket or the CLB)*
- *when evaluating the optimisation of the next bracket (see C.7), in pairing the last-but-one (normal) bracket*
- *when re-pairing the PPB (after a completion failure), during the evaluation of C.4 (see)*
- *when checking that the floaters give a legal pairing for the remaining players (completion test).*

Without this note, we might think the allocation of the PAB to be something to be done <u>after</u> having paired the last bracket – in fact, just as if that bracket had produced a floater - to be paired with a fictitious player in a virtual after-the-last bracket. Hence, if that player could not receive the PAB, we would have to consider the last bracket as the PPB, and subsequently restart the pairing process from this point of view... This note is specifically meant to avoid any possible ambiguity, explicitly excluding such an interpretation.

Moreover, the note also states that, even when it is readily apparent that from the current bracket a downfloater will result, who is bound to get the PAB (e.g., in the next

---

11 *Please note that we check (and, if necessary, change) the selected downfloaters in two completely different situations: the first is when we try to optimise the number of pairings and PSD in the next bracket (see C.7). The second is when the rest of the players cannot be paired and the PPB must give the correct floaters to allow a complete pairing. Here we are referring to the latter situation.*

12 *Actually, the name for this bracket comes from the previous version of the Swiss rules, which included a bracket with a similar function.*

13 *For further details, see [B.7].*

bracket(s) there is no player who can get it), the choice of the floater shall not keep in mind the allocation of the PAB.

Section B describes the pairing process of a single bracket.

Section C describes all the criteria that the pairing of a bracket has to satisfy.

Section E describes the colour allocation rules that determine which players will play with white.

We should also notice that pairs are made based also on expected colours, but actual colour assignment is only done at the end of the pairing.

For those who knew the old version of the Dutch rules, it may be useful to spend some words about the new structure. The new Sections B and C contain all the rules that were previously detailed in the algorithmic section with the support from (previous) Section B. Nonetheless, in the previous version of the Rules, the pairing route was different. When the pairing of a bracket was completed, it was accepted (for the moment), and the pairing went forward to the next bracket. If the next bracket was satisfactorily paired (and, sometimes, not even satisfactorily, since a downfloater could create a situation in which a resident player of the new bracket was made incompatible), the pairing for the previous one became (almost) final. If, on the contrary, a better pairing was possible for the next bracket (i.e., one that produced more pairs, or a smaller PSD), we went back to the previous bracket (backtracking) to pair it again, looking for better downfloaters. This is of course equivalent to verify that the floaters produced are the best possible choice before starting the pairing of the next bracket.

For the last bracket, where an unsatisfactory pairing means the impossibility to complete the pairing, the backtracking could be more complicate. First, when pairing the last bracket, a simple backtracking to the previous one was not always enough. Sometimes we had to join ("collapse") those two brackets, in order to be able to gain access to the preceding bracket and change its floaters - and sometimes this process had to be repeated until an acceptable pairing was found.

It is readily evident that this backwards course had to go up, starting from the last bracket, until the point was reached, in which the produced downfloaters did actually allow the pairing of the rest of the players. Hence, the backtracking did necessarily extend until it reached the bracket that, with the look-ahead methodology, is at once defined as the PPB - thus bringing us back to the same conditions. The new look-ahead method is then equivalent to the backtracking - with the advantage of a fairly simpler logic.

Anyway, the new wording of the Rules does not specify any particular method to enforce compliance with the pairing criteria. Hence, both the arbiter and the programmer enjoy complete freedom in choosing their preferred method to implement the system (look-ahead, backtracking, weighted matching or other), as long as the rules are fully complied with.

B) PAIRING PROCESS FOR A BRACKET

This section's goal, from the Rules standpoint, is to univocally define the sequence of generation for the candidate pairings - and, to this aim, it precisely defines the constraints inside which the pairing must be built. From the arbiter's point of view, however, this section may also be used as a roadmap to actually build the pairing and evaluate its quality. In fact, it can be readily adopted as a guideline to make - or, far more often, prove - a pairing.

B.1 Parameters definitions

a M0 is the number of MDP(s) coming from the previous bracket. It may be zero.

b MaxPairs is the maximum number of pairs that can be produced in the bracket under consideration (see C.5).

<u>Note</u>: *MaxPairs is usually equal to the number of players divided by two and rounded downwards. However, if, for instance, M0 is greater than the number of resident players, MaxPairs is at most equal to the number of resident players.*

c M1 is the maximum number of MDP(s) that can be paired in the bracket (see C.6).

<u>Note</u>: *M1 is usually equal to the number of MDPs coming from the previous bracket, which may be zero. However, if, for instance, M0 is greater than the number of resident players, M1 is at most equal to the number of resident players. Of course, M1 can never be greater than MaxPairs.*

In a given bracket we have a given number M0 of MDPs[14] *(possibly none), but we have no certainty that all those MDPs can be paired*[15].

Thus, we define a second parameter M1, representing the number of MDPs that can <u>actually</u> be paired - where, of course, M1 is less than or equal to M0. In summary, the bracket will contain MaxPairs pairs, at most M1 of which contain a downfloater.

We want also to observe that, while M0 is a well-known constant, we usually do not know precisely how many players, and especially MDPs, can be paired, until the actual pairing is made – actually, we need to "divine" M1 and MaxPairs out of sound reasoning, assuming a tentative value, <u>which might initially be wrong</u>. Nonetheless, those numbers, however identified, are considered constants - and that is why there is no rule to change them.

B.2 Subgroups (original composition)

To make the pairing, each bracket will be usually divided into two subgroups, called S1 and S2.

S1 initially contains the highest N1 players (sorted according to A.2), where N1 is either M1 (*in a heterogeneous bracket*) or MaxPairs (*otherwise*).

S2 initially contains all the remaining resident players.

The composition of the original subgroups is different when we have MDPs, because those players, having already floated, need now some "special protection".

In setting the number of pairs to be done to M1 for heterogeneous brackets, we focus <u>only on MDPs</u>, who (or, at least, the maximum possible number of them) actually are to be paired first[16]*. On the contrary, setting the number of pairs to MaxPairs says that we are trying to pair the <u>entire bracket</u> all at once (so it must be homogeneous).*

When M1 is less than M0, some MDPs are not

After M1 moved-down players have been selected for pairing, the remaining MDPs, in number M0-M1, cannot

[14] *We want to remember that the "Moved-down players" (MDPs) are the downfloaters of the previous bracket.*

[15] *For example, the number of MDPs may be greater than MaxPairs; or some among them may be incompatible; or we may have a semi-incompatibility, in which a group of players 'compete' for too few possible opponents, just like the situation described in the comment to A.4.*

[16] *To avoid any misunderstanding, please take notice that this is only a procedural indication that has nothing to do with the order of generation of candidates. In fact, independent of the method and algorithm used to generate them, each candidate is regarded <u>as a whole</u>; and, when we choose the 'earlier' candidate from a pool of equivalent ones, we only consider the order of generation of the <u>complete candidates</u>.*

included in S1. The excluded MDPs (*in number of M0 - M1*), who are neither in S1 nor in S2, are said to be in a Limbo.

Note: the players in the Limbo cannot be paired in the bracket, and are thus bound to double-float.

be paired in the bracket[17]. Those players form a subgroup called "Limbo". During the pairing proceedings, it may happen that some players need to be swapped between S1 and the Limbo - but, at the end of the pairing, the players still in the Limbo will be bound to float again.

B.3 Preparation of the candidate

S1 players are tentatively paired with S2 players, the first one from S1 with the first one from S2, the second one from S1 with the second one from S2 and so on.

In a homogeneous bracket: the pairs formed as explained above and all the players who remain unpaired (bound to be downfloaters) constitute a candidate (pairing).

In a heterogeneous bracket: the pairs formed as explained above match M1 MDPs from S1 with M1 resident players from S2. This is called a MDP-Pairing. The remaining resident players (if any) give rise to the remainder (see A.3), which is then paired with the same rules used for a homogeneous bracket.

> *Note: M1 may sometimes be zero. In this case, S1 will be empty and the MDP(s) will all be in the Limbo. Hence, the pairing of the heterogeneous bracket will proceed directly to the remainder.*

A candidate (pairing) for a heterogeneous bracket is composed by an MDP-Pairing and a candidate for the ensuing remainder. All players in the Limbo are bound to be downfloaters.

Here is where we build the candidate pairing. In the most general case, this is done in two steps:

- *first, we build M1 pairs, each of them containing an MDP,*
- *then, we pair the remaining resident players.*

Of course, if the bracket is homogeneous, or if none of the MDPs is pairable (i.e. if M1 is zero), the first step is omitted.

Thus, in general, the candidate comprises three parts:

- *an MDP-Pairing (heterogeneous brackets only), made of M1 pairs (maybe none) containing an MDP and a resident player each;*
- *a set of pairs of resident players, coming from the pairing of the homogeneous bracket; or from the pairing of the remainder of a heterogeneous bracket;*
- *a set of unpaired players, coming both from the Limbo and from the resident players that cannot be paired - and hence can't help but get a downfloat.*

B.4 Evaluation of the candidate

If the candidate built as shown in B.3 complies with all the absolute and completion criteria (*from C.1 to C.4*), and all the quality criteria from C.5 to C.19 are fulfilled, the candidate is called "perfect" and is (immediately) accepted. Otherwise, apply B.5 in order to find a perfect candidate; or, if no such candidate exists, apply B.8.

Having prepared a candidate, we must evaluate its quality; that is, we must check the compliance of the candidate with the pairing criteria given in Section C.

If we are very lucky, it may be "perfect": in this case, we accept it straight away.

Otherwise, we must apply some changes to try and make it perfect (B.5). If this proves impossible, the last resource is accepting a candidate that, although it is not perfect, is nonetheless the best we can have (B.8). Of course, a candidate that does not comply with the absolute criteria is not even acceptable.

[17] *Those players are not necessarily incompatible in the bracket – there may just be no place to pair them. E.g., if two MDPs share the same one possible opponent, neither of the two is incompatible - but nonetheless one of the two MDPs cannot be paired!*

After the pairing is made, and before accepting it and proceeding to the next bracket, we will have to perform a completion test, to check that all the remaining players, including the downfloaters from the bracket just paired, allow the round-pairing to be completed (see A.9). If this completion test fails, we define the Collapsed Last Bracket and proceed as explained in A.9.

B.5 Actions when the candidate is not perfect

The composition of S1, Limbo and S2 has to be altered in such a way that a different candidate can be produced.

The articles B.6 (for homogeneous brackets and remainders) and B.7 (for heterogeneous brackets) define the precise sequence in which the alterations must be applied.

After each alteration, a new candidate shall be built (*see B.3*) and evaluated (*see B.4*).

The process of pairing is an iterative one: if the pairing is not perfect, we try (one by one) a precise sequence of alterations in the subgroups S1, Limbo, and S2, and each time we repeat the preparation and evaluation of the candidate. There are, in fact, two different sequences:

- *one for homogeneous brackets (B.6), which contain no MDPs; this sequence also applies to remainders*
- *one for heterogeneous brackets (B.7); those contain MDPs, some of which (in number M0-M1, which may be zero) are in a Limbo, so the alterations must keep into account not only the usual possible alterations in S1 and S2, but also the possibility to change the composition of the Limbo.*

The first perfect candidate found in this process is the required pairing. If there is no perfect candidate, we shall have to use the best available one; since we are scrutinizing all candidates, we can find this best candidate as we proceed. To do that, when we find the first legal (but not perfect) candidate, we mark it as a "provisional-best". Each time we find another legal candidate, we shall compare[18] it with the current provisional-best candidate. If the former is better than the latter, we store it as the new provisional-best; otherwise, we keep the old one. In the end, all candidates have been examined; hence, the surviving provisional-best is actually the best possible (although imperfect) candidate, which will be accepted as pairing, because of rule B.8.

The main guideline to carry out this task is the "minimum disturbance": every alteration must be the minimum possible, so that the resulting pairing can be as similar as possible to a "perfect" one.

For more detail about the iterative pairing process, see B.6 and B.7.

B.6 Alterations in homogeneous brackets or remainders

Alter the order of the players in S2 with a transposition (*see D.1*). If no more transpositions of S2 are available for the current S1, alter the original S1 and S2 (*see B.2*) applying an

Since we are now managing only homogeneous brackets, we do not need to worry about pairing MDPs.

[18] *Two candidates are compared based on the compliance with the pairing criteria, which are defined in order of priority in section C. The first check is on the priority of the higher infringed criterion: the higher it is, the lower is the quality of the candidate. Then the second check is on a "failure value" which is peculiar to that criterion – this will often be the number of times the criterion is infringed (e.g., the numbers of disregarded colour preferences) but it may also be of a completely different nature (e.g., the PSDs of two candidates to be compared). Then we go to the second higher infringed criterion; then to the latter's failure value - and so on until we find a difference. When there is no difference at all, the first generated candidate takes precedence.*

exchange of resident players between S1 and S2 (*see D.2*) and reordering the newly formed S1 and S2 according to A.2.

The possible actions to be tried here are:

- *a transposition, consisting in applying a different order to the players in S2. In simple words, a transposition "shuffles" the players in S2 according to specific rules (see D.1), but keeping them separate from the players of S1. This leads to a change in the second player in some pairs. The basic idea is to alter the pairing by modifying players' order in as low as possible rankings.*
- *an exchange, consisting in swapping one or more players from subgroup S1 with the same number of players from subgroup S2. As above, the basic idea is to try to alter the pairing as little as possible. To this aim, we swap players in as low as possible rankings of S1 with players in as high as possible rankings of S2 - assuming that, being near in ranking, they have more or less equivalent playing strength. After any exchange, both the subgroups S1 and S2 must be put in order again with the usual rules. An exchange makes the pairing between players of the same original subgroup possible.*

After we made transpositions in a bracket, alterations in the order are desired; hence, players in the S2 subgroup should not be sorted again (while S1 does not need to be sorted, as it has not been changed).

On the contrary, after exchanges, which swap one or more players between subgroups S1 and S2, we must sort both subgroups S1 and S2 according to A.2, to re-establish a correct order before beginning a new sequence of pairing attempts. If the first attempt of the new exchange fails to give a valid result, we will try transpositions too, thus changing the natural order in the modified S2.

Both transpositions and exchanges should not be applied at random: to comply with the general principle of minimal disturbance of the pairing, section D dictates a precise sequence of possible transpositions and exchanges. This sequence begins with alterations that give only mild disturbances to the pairing (with respect to the "natural" one), moving gradually towards those changes that cause definitely important effects.

The order of actions is as follows: first, we try, one by one, all the possible transpositions (see D.1). If we find one that allows a perfect pairing, the process is completed. Otherwise, we try the first exchange (see D.2): with this, we proceed again to try every possible transposition[19], until we succeed - or use them up. In the latter case, we try the second exchange, once again with all the possible transpositions, and so on.

If we get to the point in which we have used up all the

[19] *Suppose we exchanged player A from S1 with player B from S2. After the exchange, player B, now in S1, has a rank that is lower than that of player A, now in S2. As transpositions proceed, we will get to a point in which the candidate puts together players B and A – and then of course some other pairs of players. Now, before making the exchange, we tried all transpositions in S2, and thus also the one which contains the pair A-B and all the same other pairs as well – in summary, this candidate has already been evaluated! Reasoning along the same lines, we reach the conclusion that the same holds true also for exchanges involving more players. We can thus deduce that every time a pair contains a player from S1 with a lower rank (higher BSN) than its opponent from S2, this pair belongs to a candidate that has already been evaluated, and therefore we do not need to evaluate it again.*

possible transpositions and exchanges, then a perfect pairing simply does not exist. In that case, we apply B.8, thus accepting a less than perfect result.

B.7 Alterations in heterogeneous brackets

Operate on the remainder with the same rules used for homogeneous brackets (see B.6).

> *Note: The original subgroups of the remainder, which will be used throughout all the remainder pairing process, are the ones formed right after the MDP-Pairing. They are called S1R and S2R (to avoid any confusion with the subgroups S1 and S2 of the complete heterogeneous bracket).*

This article, a companion to the previous one, addresses the case of heterogeneous brackets. This kind of bracket is paired in two logical steps[20]*:*

- *in the first step, we build an MDP-Pairing (see B.3), which takes care of the pairable moved-down players (as many as possible of them), giving raise to a remainder (and possibly a Limbo).*
- *in the second step, after the MDPs have been paired, we proceed to pair the remainder, which is made only of resident players (but we need to take notice that, when we are processing a CLB, those players may well have different scores. In this case, the PSD is of importance and must be accounted for – we will go back to this presently).*

The rules to operate on the remainder are just the same that apply for a homogeneous bracket. The difference shows only when we reach the point in which all of the possible transpositions and exchanges in the remainder have been unsuccessfully tried.

If no more transpositions and exchanges are available for S1R and S2R, alter the order of the players in S2 with a transposition (*see D.1*), forming a new MDP-Pairing and possibly a new remainder (to be processed as written above).

In a homogeneous bracket, this is the moment when we lower our expectations, settling for a less than perfect pairing (see B.6). In a heterogeneous bracket, however, we are not yet ready to surrender: before laying down arms, we can try to change the composition of the remainder.

To do that, we try a new, different MDP-pairing by applying a transposition to the <u>original</u> subgroup S2 (viz. the subgroup S2 of the complete bracket, not that of the remainder!). This may leave us with a new, different remainder, which we process (just as described above) trying to find a complete pairing – and, if we have no success, we try transposition after transposition until we succeed, or exhaust them all[21]*.*

As we hinted above, the PPB and the CLB are subject to slightly different pairing rules: the downfloaters of the PPB are no longer required to optimise the pairing in the next bracket (as it would be for normal brackets, see C.7), but <u>just to allow it</u> (see C.4). With those downfloaters, together with the SCS, we build the CLB, which is (by definition) the last bracket.

---

[20] *Of course, a practical implementation need not necessarily compose the pairing in two steps, as long as the final effect is the same as specified by the rules.*

[21] *Actually, we do not need to try all of the transpositions, because not all of them are meaningful: in fact, we only have to try those transpositions that actually change the players, or their order, in the first part of the subgroup S2 – i.e. those players, who are going to be paired with the MDPs from S1. All the other players in S2 do not take part in this phase of the pairing and are thus irrelevant (at least for the moment).*

This is a rather unusual bracket: it is by definition heterogeneous[22]*, and its residents often have different scores (because they come from the SCS). Its pairing is different from that of the usual heterogeneous bracket in that we have a remainder that must be paired just like if it were homogeneous, but without disregarding the needs of players with different scores.*

Thus, we must enforce some criteria that usually are not important in remainders. The main goal in pairing the CLB is to get the lowest possible PSD (because, basically, the number of pairs is determined by the number of PPB floaters). To find this minimum PSD, we have to look not only at the MDP(s) and at their opponents (as usual), but also at the pairs that can be made inside the remainder (i.e. between SCS residents).

When several candidates have the lowest possible PSD, we must also enforce some criteria for the remainders, which are not usually required. If in a pair there are players with different scores, to such players we must apply all those criteria that limit the repetition of floats [C.12 to C.15] and the score difference of the protected players whose protection has already failed once or more [C.16 to C.19].

If no more transpositions are available for the current S1, alter, if possible (i.e. if there is a Limbo), the original S1 and Limbo (*see B.2*), applying an exchange of MDPs between S1 and the Limbo (*see D.3*), reordering the newly formed S1 according to A.2 and restoring S2 to its original composition.

If all the possible transpositions have been used up, we have a resource yet: trying to change the MDPs to be paired. Of course, this is only possible if there is a Limbo in the bracket. In this case, we can exchange one or more of the MDPs with a same number of players from the Limbo. This is called an MDP-exchange (see D.3).

After any MDP-exchange, we are actually pairing an altogether different bracket; hence, we need to reorder S1 and restore S2 to its original composition, in fact starting the pairing process anew. As it was for the homogeneous case, the MDP-exchanges must be tried in the correct sequence, one by one; and, for each one of them, we shall try all the possible transpositions in S2, thus generating a different remainder - that will of course have to undergo all the usual pairing attempts as described above.

B.8 Actions when no perfect candidate exists

Choose the best available candidate. In order to do so, consider that a candidate is better than another if it better satisfies a quality criterion (C5-C19) of higher priority; or, all quality criteria being equally satisfied, it is generated earlier than the other one in the sequence of the candidates (*see B.6 or B.7*).

This is where we must make ourselves content with what best we can: if we arrive here, we have already tried all possible transpositions and exchanges, only to reach a simple, if dismal, conclusion - there is no perfect candidate! Hence, we choose the best available candidate, which is the final provisional-best found during the evaluation of all candidates as illustrated in B.5.

The Sieve Pairing

A very interesting alternative to this method – not necessarily a practical one, but very important from the theoretical point of view – is the one we shall call "Sieve pairing" (because of its similarity with the famous

[22] *Remember that the CLB is born from a failure in a completion test. This means that the "rest of the players", with the current downfloaters (possibly none!) from the just (unsuccessfully!) paired bracket, cannot be paired - it therefore requires some adequate MDPs.*

Eratosthenes' Sieve).

The basic idea is very simple: we build all the possible acceptable pairings (i.e., all those that comply with the absolute criteria). Then we start applying all the pairing criteria, one by one - but this time we start with the most important one and proceed downwards.

Each criterion will eliminate part of the acceptable pairings, so that, as we proceed, the number of candidates becomes lower and lower. If, at some stage of the process, only one candidate remains, we choose that one – it may even be a rather bad one, but there is nothing better.

If, after applying all the pairing criteria, we are left with more than one candidate, then we choose the one that would be the first to be generated in accordance with the sequence defined by Section B.

C) PAIRING CRITERIA

Absolute Criteria

The absolute criteria correspond to the requirements of Section C.04.1, "Basic Rules for Swiss Systems" in the FIDE Handbook, which we may want to look at closely.

No pairing shall violate the following absolute criteria:

Those criteria must be complied with always: they cannot be renounced, whatever the situation[23]. To enforce them, players may even float down as needed.

C.1

See C.04.1.b (*Two players shall not play against each other more than once*)

If the game is won by forfeit, for the purposes of pairing those two players have never met. As a result, that pairing may be repeated later in the tournament (and sometimes this happens, too!).

C.2

See C.04.1.d (*A player who has already received a pairing-allocated bye, or has already scored a (forfeit) win due to an opponent not appearing in time, shall not receive the pairing-allocated bye*)

Please notice that, contrary to the previous rules, only PABs and forfeit wins prevent the allocation of a PAB (see A.5) – on the contrary, a player who received a requested bye (usually, half point) may receive the PAB in a subsequent round.

C.3

Non-topscorers (*see A.7*) with the same absolute colour preference (*see A6.a*) shall not meet (*see C.04.1.f and C.04.1.g*)

This criterion does not apply to topscorers (A.7) or topscorers' opponents, who are the only possible exception to C.04.1.f/g.

Two players, who cannot be paired to each other without infringing criteria C.1 or C.3, are said to be incompatible.

Completion Criterion

C.4

If the current bracket is the PPB (*see A.9*): choose the set of downfloaters in order to complete the round-pairing

This is an absolute criterion too, but it applies only to the processing of the PPB – hence, only after a completion test failure (see A.9). Contrary to ordinary brackets (whose downfloaters are chosen in order to optimise the pairing of the next bracket - see C.7), for the PPB we just

[23] *There are however situations in which no pairing at all exists, which complies with the absolute criteria – in such cases, the arbiter must apply his better judgment to find a way out of the impasse (see A.9).*

require a choice of downfloaters that allows a completion of the round-pairing - independent from the optimization of the next bracket, which is of course the CLB, and hence must be completely paired.

Please note that, since C.4 precedes both C.5 and C.6, the compliance with this criterion may cause a reduction in the number of pairs, or an increase in the final PSD, with respect to the previous pairing[24].

Quality Criteria

The above criteria set conditions that must be obeyed: a candidate that does not comply with them is discarded. The following criteria are of a different kind, in that they establish a frame of reference for a quantitative evaluation of the "goodness" of the pairings, by setting a sequence of "test points" in order of decreasing importance, according to the internal logic of the system. The level of compliance with each one of the following criteria is not a binary quantity (yes/no) but a numerical (integer or fractional) quantity. We will measure it by means of a "failure value", whose meaning is of course tightly connected to the criterion itself (e.g., the number of pairs less than MaxPairs for C.5, or the number of players not getting their colour preference for C.10, and so forth).

When we compare two candidates, we in fact compare the failure values of the candidates for each criterion, one by one, in the exact sequence given by the Rules. If the two failure values are identical, we proceed to the next criterion. If they are different, we keep the candidate with the better value and discard the other one.

It seems worth noting that a candidate having a better failure value on a higher criterion is selected, even if the failure values for the following criteria are far worse. In other words, the optimisation with respect to a higher criterion may have a dramatic impact on the remaining failure values – and, we may add, the optimisation with respect to a criterion is always only relative to the current status, because even a small difference in a higher criterion may change the situation completely.

To obtain the best possible pairing for a bracket, comply as much as possible with the following criteria, given in descending priority:

Relative criteria are not so important as absolute ones, and they can be disregarded, if this is needed to achieve a complete pairing. In general, they are not important enough to make a player float – in fact, the first one of them, and hence the most important, instructs us to do just the very opposite, minimising the number of downfloaters!

Apart from the remaining player in odd brackets, only incompatible (or semi-incompatible) players should float. This too is an evidence of the attention of the FIDE (Dutch) system towards the choice of the "right strength opponent".

C.5

Maximize the number of pairs (*equivalent to: minimize the number of downfloaters*).

The first "quality factor" is of course the number of pairs, a reduction of which increases the number of floaters (and, usually, also of the overall score difference between players).

Maximising the number of pairs actually means, build MaxPairs pairs (see B.1). At the beginning of the pairing process, though, MaxPairs, or the maximum number of pairs that can be built (which is a constant of the bracket), is actually unknown – hence, we need to "divine" it.

Actually, the only things we know for sure are the total number N of players in the bracket, and the number M0 of MDPs entering the bracket. We want to observe that the number of pairs can never be greater than N/2; thus, this value should make a good starting point, independent of the kind of bracket (homogeneous or heterogeneous).

[24] *Of course, since the bracket we are pairing is a PPB, it has already been paired once.*

The actual value of MaxPairs can be less than that, because some players might be impossible to pair in the bracket. Moreover, if this bracket is a PPB, it must also provide the downfloaters required to complete the round-pairing (see C.4), and that might detract to the number of pairs that can actually be built. Hence, the process to determine MaxPairs value is somewhat empirical and may require some "experimenting".

If the bracket is heterogeneous (M0≠0), then as many MDPs as possible (M1) must be paired. They will be paired first, before proceeding with the rest of the players (see B.3) - but, as it happened for the value of MaxPairs, we still do not know the true value of M1, and we must divine it too. A first educated guess for its value is M0 – minus, of course, any incompatible MDPs.

If there is no way to make all those pairs, our estimate of the value of M1 was apparently too optimistic – in this case, we will have to gradually decrease it, until we succeed. Any remaining MDPs join the Limbo (see B.2) and shall eventually float (after the completion of the pairing for the bracket).

The number of pairs made in the MDP-pairing will be subtracted into the total number of pairs to be made in the bracket, yielding the (plausible) number of pairs to be built in the remainder[25]*.*

Here too applies the same line of reasoning: if we cannot make all those pairs, our initial estimation of MaxPairs was apparently too optimistic – hence, we will have to gradually decrease their number. Any remaining players become downfloaters, and will eventually float down into the next bracket.

The same line of reasoning also holds for a homogeneous bracket - which, by definition, contains no Limbo or MDPs, but is otherwise essentially similar to a remainder.

C.6

Minimize the PSD (*This basically means: maximize the number of paired MDP(s); and, as far as possible, pair the ones with the highest scores*).

In heterogeneous brackets, even when the same number of pairs is made, different choices of floaters, or different pairings, can lead to different mismatching between players' scores (for an example, see the many possible ways to pair a heterogeneous bracket containing many players all having different scores). This important criterion, directly related to rule C.04.1:e, directs us to minimise the overall difference in scores. Its location before the colour related criteria (C.8-C.11) is suggestive of the attention the FIDE (Dutch) system gives to the choice of a "right strength" opponent rather than a "right colour" one.

The method to compute and compare the PSDs is explained in detail in the comment to article A.8.

[25] *We always want to remember that the pairing of the MDPs and of the remainder are two phases of a single operation, which is performed as a unit. Thus, we do not "go back" from the remainder pairing to the MDP-pairing, because we are already inside the same operation.*

C.7

If the current bracket is neither the PPB nor the CLB (*see A.9*): choose the set of downfloaters in order first to maximize the number of pairs and then to minimize the PSD (*see C.5 and C.6*) in the following bracket (*just in the following bracket*).

When we get here, we have already complied with the absolute criteria (hence the pairing is a legal one) and optimised the most important pairing quality parameters (number of pairs, PSD).

Before going ahead to optimise colours and MDPs treatment, we take a look ahead to the next bracket. We do not want to ever come back to the current bracket again. Thus, we must make sure that the choice of downfloaters we are going to send to the next bracket will be <u>the best possible one</u> to comply with C.5 and C.6.

First, we check that the downfloaters (which will be the MDPs of the next bracket) will allow us to compose the maximum possible number of pairs.

For example, let us suppose that the current bracket produces only one downfloater and that the next scoregroup contains an odd[26] *number of players, one of which has no possible opponent. If we can choose between two possible downfloaters, both compatible in the destination bracket, but only one of them can be paired to the "problematic" player, we must choose that one - because choosing the other one would leave an incompatible player (and hence an unavoidable downfloater!) in the destination target.*

Only when the number of pairs have been maximised, we proceed to look into the PSD in the destination target. This in practice means that, when we may choose between two or more possible downfloaters, if all other conditions are equivalent, we must choose the downfloater that may be paired <u>with the lowest score difference</u>[27]*.*

This optimisation is to be extended only to the next bracket. Actually, there are situations in which a small change in a previous pairing would bring in large benefits - but looking several brackets ahead would be too much difficult an operation to be carried on every time. So the rules settle for a practical optimisation, renouncing those that are out of reasonable reach. But the reason is not only this one: in the basic philosophy of the FIDE (Dutch) system, the pairings for the higher ranked players are considered far more important than those for the lower ones. Hence, altering the pairing of the current bracket for the benefit of some player, who is located two brackets below this one, would simply be opposite to that philosophy.

Having already made sure that both the number of floaters and their scores are at a minimum, we now start to optimise colour allocation. Actually, colour is less important than difference in score – and that's why, consistently with the basic logic of the system, the colour allocation criteria are located after those that address number of pairs and PSD.

---

[26] *Please note that if the next scoregroup contained an even number of players, the bracket built with it and the current downfloater would be odd. Hence, it would in any case produce (at least) one downfloater and the choice of the MDP would not be critical for the number of pairs.*

[27] *Since this criterion does not apply for the PPB, the next bracket's resident players will all have the same score. Thus, it is not possible for moved-down players to be paired with players having different scores - but, if they cannot be paired in the bracket, they will have to float again, and this makes the PSD change!*

C.8

Minimize the number of topscorers or topscorers' opponents who get a colour difference higher than +2 or lower than -2.

C.9

Minimize the number of topscorers or topscorers' opponents who get the same colour three times in a row.

Article C.3, in accordance with C.04.1:f-g, states that when two non-topscorers meet, their absolute preferences <u>must</u> be complied with. Here we have the special case of a topscorer who, for some reason, is bound to be paired with a player (who may or may not be also a topscorer) having the same absolute preference. The outcome of those players' games may be very important in determining the final ranking and podium positions; and this is an exception explicitly provided for by C.04.1:f-g, so we may compose such pairs. Thus, we choose the best possible matched opponent – but there must not be more such pairs than the bare minimum.

The subdivision into two individual rules establishes a definite hierarchy, giving more importance to colour differences than to repeating colours. Suppose that, for one same opponent, we can choose between two possible topscorers, and all those players have the same absolute colour preference. In this case, we must select the components of the pair in such a way that colour differences are minimised (as far as possible).

As hinted above, a player, who has an absolute colour preference without being a topscorer, may happen to be paired with a topscorer having an identical absolute colour preference. These two rules equate the players of the pair - thus, a player might be denied its absolute colour preference just as if it were a topscorer, even if it is not one!

C.10

Minimize the number of players who do not get their colour preference.

We can have an idea about the minimum number of players who cannot get their colour preference, by inspecting the bracket, prior to the pairing.

*Let us suppose that **m** players prefer a colour and **n** players prefer the other one, with **m ≥ n**. We can thus compose <u>no more</u> than **n** pairs in which the players are expecting different colours; and the colour preferences in these pairs can - and must - be satisfied.*

*The remaining **m-n** players all expect the same colour; and they will have to be paired among themselves. In each of the pairs thus composed, one of the two players cannot get its preferred colour. The number of such pairs, and henceforth of such players too, is **x=(m-n)/2**, rounded downward to the nearest integer if needed. Sometimes, in addition to those **m+n** players, the bracket contains also **a** more players who have no colour preference at all. Those players may get any colour, but, of course, they will usually get the minority colour, so that they will subtract to the number of disregarded preferences. Taking one more step further, we may reason that we can build a maximum of **MaxPairs** pairs. Among those, **n+a** pairs can satisfy both the colour preferences, whilst the remaining **x=MaxPairs-n-a** cannot help but disregard one colour preference. Of course, **x** cannot be less than zero (a negative number of pairs has no practical meaning); thus, we obtain the final and general definition for **x**:*

$\boldsymbol{x = \max(0, MaxPairs\text{-}n\text{-}a)}$

*Please take notice that a perfect pairing always has exactly **x** disregarded colour preferences – no more, no less.*

Actually, there might be even more pairs in which a player does not get its preference - because of incompatibilities due to absolute criteria, as well as "stronger" relative ones. Thus, at first we propose to make the minimum possible number of such pairs – but we may need to increase this number, to find our way around various pairing difficulties.

Since the general philosophy of the FIDE (Dutch) system gives more importance to the correct choice of opponents than to colours, the pairs containing a disregarded colour preference will typically be among the first to be made[28].

C.11

Minimize the number of players who do not get their strong colour preference.

Only now, having maximised the number of "good" pairs, we can set our attention to satisfying as many strong colour preferences as possible.

*The minimum number of players not getting their strong colour preference, which is usually represented by **z**, is of course a part of the total number **x** of disregarded colour preferences (see note to C.10) – therefore, **z** is at most equal to **x**.*

*For instance, let the number W_T of white seekers be greater than the number B_T of black seekers (we call White "the majority colour"). The **x** players will all be White seekers, and as many as possible among them should have mild colour preferences, while the rest will have strong colour preferences*[29]*. Hence we can estimate **z** simply as the difference between **x** and the number W_M of White seekers who have a mild colour preference, with the obvious condition that **z** cannot be less than zero; hence:*

$\boldsymbol{z = \max(0, x - W_M)}$ **if** $\boldsymbol{W_T \geq B_T}$ *(White majority)*

$\boldsymbol{z = \max(0, x - B_M)}$ **if** $\boldsymbol{W_T < B_T}$ *(Black majority)*

With a careful choice of transpositions and/or exchanges, we might be able to minimise the number of disregarded strong preferences[30].

---

[28] *Actually, transpositions swap players beginning with the last positions of S2 and going upwards, causing the bottom pairs of the bracket to be modified early in the transposition process, while the top pairs are modified later. Hence, a "colour-defective" pair located at the bottom of the candidate has a higher probability to be changed soon than a similar pair located at the top – therefore, perfect pairings with top "colour-defective" pairs have a definitely higher probability. Incidentally, we might also mention that players often seem to worry about "colour doublets" (like, for example, WWBB) and think that such colour histories are more frequent with the FIDE (Dutch) system than with other Swiss pairing systems. This is not so. In fact, such histories are usual enough (and unavoidable) in all manners of Swiss pairings – in the FIDE (Dutch) system they may seem more frequent just because they appear more often in the top pairs of the bracket, therefore involving higher ranked players, which makes them more noticeable.*

[29] *We want to notice that, during the last round, some absolute colour preferences might be disregarded for topscorers or their opponents (see C.8, C.9), so that part of **x** may represent such players. In those instances, our line of reasoning should be suitably adapted.*

[30] *Of course, since the total number of disregarded preferences must remain the same (we cannot have it smaller, and do not want it to grow larger!), this may only happen at the expense of a same number of mild preferences. A brief*

For several reasons, however, the number of players who cannot get their strong preference may be greater than that.

The following group of criteria optimises the management of floaters, which is the last step towards the perfect pairing.

C.12

Minimize the number of players who receive the same downfloat as the previous round.

C.13

Minimize the number of players who receive the same upfloat as the previous round.

C.14

Minimize the number of players who receive the same downfloat as two rounds before.

C.15

Minimize the number of players who receive the same upfloat as two rounds before.

Rule C.04.1:e states that, in general, players should meet opponents with the same score. This is (of course) best achieved by pairing each player inside its own bracket. However, there are some situations, in which a player cannot be paired in its bracket - and then, by necessity, must float. These criteria limit the frequency with which such an event can happen to a same player - but they are "very weak criteria", in the sense that they are almost the last to be enforced - and almost the first to be ignored in case of need.

Here, each criterion establishing a certain protection for downfloaters is immediately followed by a similar one establishing the very same protection for upfloaters.

Because of this, there is a certain residual asymmetry in the treatment; viz. downfloaters are (just a little bit) more protected than upfloaters. Please note that, in some other Swiss systems, floaters' opponents are <u>not</u> considered floaters themselves, and therefore enjoy no protection at all.

C.16

Minimize the score differences of players who receive the same downfloat as the previous round.

C.17

Minimize the score differences of players who receive the same upfloat as the previous round.

C.18

Minimize the score differences of players who receive the same downfloat as two rounds before.

C.19

Minimize the score differences of players who receive the same upfloat as two rounds before.

The four previous rules minimised the number of players who, having floated in the last two rounds, may get a float again in this round. However, those rules do not give any special protection either to a player who, being already a MDP in a bracket (in <u>this</u> round), cannot be paired and must float down again, or to its opponent. Such players, and their opponents, will have larger score differences than their fellow "single" floaters and are usually called <u>double-floaters</u>.

The criteria C16-C.19 are for protected players whose protection has already failed once or more, and try to prevent such players from further floating. When we must make some players float down, we try, as long as possible, to choose those players who are not MDPs. Sometimes, however, this is not possible, and we must make some MDP float down. In this case, we should, as far as possible, choose those MDPs that are not (or are least) protected because of previous floats. Of course, the same holds (almost) symmetrically for the MDPs' opponents.

*example may shed some light on the matter. Consider the bracket {1Bb, 2b, 3Bb, 4b}, where we have **x**=2, but **z**=0. The latter means that we can build the pairs in such a way that any one of them contains no more than one strong colour preference – and, in fact, a simple transposition allows us to obtain just this result.*

For example, in a CLB (see A.9) that contains players with many different scores, the effect of these rules is that, if we have two possible prospective floaters and only one of them is protected, we try to pair the latter with a SD as little as possible[31].

D) RULES FOR THE SEQUENTIAL GENERATION OF THE PAIRINGS

This section states the rules to determine the sequence in which transpositions, exchanges, and MDP-exchanges must be tried, in order to generate the candidates in the correct order. The general basic principle is, as always, that of "minimal disturbance" of the pairing. This means that we have always to move that player (or those players) whose displacement will cause the least possible difference of the pairing from the "natural" one[32] - while at the same time allowing the best possible quality of the pairing itself.

Before any transposition or exchange take place, all players in the bracket shall be tagged with consecutive in-bracket sequence-numbers (BSN for short) representing their respective ranking order (according to A.2) in the bracket (*i.e. 1, 2, 3, 4, ...*).

The use of pairing-ids, in this phase, may sometimes be confusing. Therefore, we give temporary sequence numbers to the players, as a very handy remedy to simplify the application of the rules below.

D.1 Transpositions in S2

A transposition is a change in the order of the BSNs (*all representing resident players*) in S2.

All the possible transpositions are sorted depending on the lexicographic value of their first N1 BSN(s), where N1 is the number of BSN(s) in S1 (*the remaining BSN(s) of S2 are ignored in this context, because they represent players bound to constitute the remainder in case of a heterogeneous bracket; or bound to downfloat in case of a homogeneous bracket - e.g. in a 11-player homogeneous bracket, it is 6-7-8-9-10, 6-7-8-9-11, 6-7-8-10-11, ..., 6-11-10-9-8, 7-6-8-9-10, ..., 11-10-9-8-7 (720 transpositions); if the bracket is heterogeneous with two MDPs, it is: 3-4, 3-5, 3-6, ..., 3-11, 4-3, 4-5, ..., 11-10 (72 transpositions)*).

All transpositions are sorted or compared based on the dictionary ("lexicographical") order, so that one given transposition precedes or follows another one if the string formed by the players BSNs of the first one precedes or follows that of the second one. The method to compare the strings is the very same already illustrated for the comparison of PSDs[33].

The subgroup S1 may or may not have the same number of players as S2. For the comparison to have a meaning, we must define the number of elements of each of the two strings of BSNs that we are comparing.

We are looking for mates for each element in S1 (which of course represent a player each). Thus, we consider the number N1 of elements in S1– while the remaining players are (for the moment) irrelevant.

A simple example will help us clarify the matter: consider a heterogeneous bracket {S1=[1]; S2=[2, 3, 4]}. All the possible transpositions of S2 (properly sorted, and including the original S2) are:

[2,3,4]; [2,4,3]; [3,2,4]; [3,4,2]; [4,2,3]; [4,3,2][34].

---

[31] *Another example is the case of two MDPs with different scores, and a protected resident who must be paired with one of those two MDPs: the resident should be paired to the MDP who has the lower score of the two.*

[32] *But, to avoid misunderstandings, we should keep in mind that any change in the order in S2 (transposition) is by definition preferable to even a single exchange between S1 and S2.*

[33] *See the comment to C.6 [page 24] for details. Please note that the use of alphabet letters would be completely equivalent to that of numbers, at least for brackets with less than 26 players. The use of numbers, however, allows an identical treatment for all brackets, whatever the number of players they contain.*

[34] *Please note that, in the very simple case where every BSN is a single digit, the string may be interpreted as a number, which becomes larger and larger as we proceed with each new transposition: 234, 243, 324, 342, 423, 432.*

[35] *Of course, this equivalence is in no way general – it depends only on the fact that we are looking for just one element!*

As we want to pair #1 with the first element of S2, it is at once apparent that [2,3,4] and [2,4,3] have the very same effect[35]; and the same holds for [3,2,4] and [3,4,2]; and for [4,2,3] and [4,3,2]. Hence, the actual sequence of transpositions is as follows (elements between braces "{…}" are 'irrelevant' and are ignored in this phase):

[2]{3, 4}; [3]{2, 4}; [4]{2, 3}

D.2 Exchanges in homogeneous brackets or remainders (original S1 ↔ original S2)

An exchange in a homogeneous bracket (also called a resident-exchange) is a swap of two equally sized groups of BSN(s) (*all representing resident players*) between the original S1 and the original S2.

The exchanged sets must of course have the same size - because, were it not so, we would be changing the sizes of S1 and S2.

In order to sort all the possible resident-exchanges, apply the following <u>comparison rules between two resident-exchanges</u> in the specified order (*i.e. if a rule does not discriminate between two exchanges, move to the next one*).

The priority goes to the exchange having:

However, to evaluate the "weight" of the change, we must take into consideration not only the size of the exchanged sets but also the choice of players. To do that, we need a set of criteria addressing the various aspects of this choice. The aim is, as always, the "minimal disturbance" – viz. to try and have a pairing as similar as possible to the natural one.

a the smallest number of exchanged BSN(s) (*e.g. exchanging just one BSN is better than exchanging two of them*).

The first criterion is, of course, the number of involved players: the less, the better!

b the smallest difference between <u>the sum of the BSN(s) moved from the original S2 to S1</u> and <u>the sum of the BSN(s) moved from the original S1 to S2</u> (*e.g. in a bracket containing eleven players, exchanging 6 with 4 is better than exchanging 8 with 5; similarly exchanging 8+6 with 4+3 is better than exchanging 9+8 with 5+4; and so on*).

From a theoretical point of view, all players in S1 should be stronger than any player in S2 is. Therefore, when we have to swap two players across subgroups, we try to choose the weakest possible player in S1 and swap it with the strongest possible one from S2.

To do so, we can use the BSNs to choose a player as low-ranked as possible from S1, and a player as high-ranked as possible from S2, and then swap them, assuming that a higher rank should indicate a stronger player.

Thus, the difference between exchanged numbers is (or, at least, should be) a direct measure of the difference in (estimated) strength and should therefore be as little as possible.

c the highest different BSN among those moved from the original S1 to S2 (*e.g. moving 5 from S1 to S2 is better than moving 4; similarly, 5-2 is better than 4-3; 5-4-1 is better than 5-3-2; and so on*).

When two possible choices of players to be exchanged show an identical difference in the sum of their respective BSNs, we choose the set which disturbs S1 as little as possible, i.e. the one in which the (highest BSN) player from S1 has a lower rank.

In the example, 5-2 is better than 4-3 because exchanging #5 is better than exchanging #4. Similarly, (5,4,1) is a better choice than (5,3,2), because exchanging #4 is better than exchanging #3[36].

[36] *Sometimes, just as it happens in the above example, we might end up exchanging a higher-ranked player, as a side effect of enforcing the exchange of the lowest possible player. To understand this, we want to remember that, in the exchange, we do not operate on "several single players" but <u>on a whole set</u> of them, and we just have to decide if a set is better or worse than another one. In this case, (5, 4, 1) is better than (5, 3, 2) – therefore, we exchange #1, who is the top-player, <u>because this is the way to exchange #4 rather than #3</u>.*

d the lowest different BSN among those moved from the original S2 to S1 (*e.g. moving 6 from S2 to S1 is better than moving 7; similarly, 6-9 is better than 7-8; 6-7-10 is better than 6-8-9; and so on*).

Finally, having optimised the difference in ranking and the disturbance in S1, we can optimise the disturbance in S2 too.

Contrary to S1, now we try to exchange the lower possible BSNs. Hence, 6-9 is better than 7-8, because exchanging #6 is better than exchanging #7 – and so forth.

D.3 Exchanges in heterogeneous brackets (original S1 ↔ original Limbo)

An exchange in a heterogeneous bracket (also called a MDP-exchange) is a swap of two equally sized groups of BSN(s) (*all representing MDP(s)*) between the original S1 and the original Limbo.

Here we are changing the composition of the set of pairable MDPs. Of course, this alteration may only occur when M1 < M0[37]*, because only in this situation does a Limbo exist. This means that we must choose which MDPs to exclude from the pairing. Sometimes the decision is easy – e.g. there may be some incompatible MDP, and we may have no choice at all*[38]*.*

In order to sort all the possible MDP-exchanges, apply the following comparison rules between two MDP-exchanges in the specified order (*i.e. if a rule does not discriminate between two exchanges, move to the next one*) to the players that are in the new S1 after the exchange.

When we have a choice, we start by trying to pair as many MDPs as possible, and as high ranked as possible [B.2]. If we must change this original composition, we need to apply an MDP-exchange. The following criteria allow us to determine the priority among all the possible exchanges. Please note that this result is achieved by inspecting the composition of the new S1, not that of the Limbo.

The priority goes to the exchange that yields a S1 having:

a the highest different score among the players represented by their BSN (*this comes automatically in complying with the C.6 criterion, which says to minimize the PSD of a bracket*).

To a hasty reader, it might seem that, pairing a player with lower score would yield a lower score difference, and thus a lower PSD. Of course, this is definitely wrong! When we put a higher scored player in the Limbo, that player will float – hence, the corresponding SD, which is calculated with the artificial value defined in A.8, will be very high. To minimise the PSD, the Limbo must contain a minimum of players, and those must have as low a score as possible. Hence, complying with C.6, which instructs us to minimise the PSD, automatically satisfies this criterion too.

We also want to take notice that the number of exchanged players is not all-important. For example, consider an S1 with three players and a Limbo with two: in some circumstances, exchanging the two lower ranked players may give better results than exchanging just the top one.

b the lowest lexicographic value of the BSN(s) (sorted in ascending order).

This is the criterion we must strive to comply with. When the involved players have the same scores, we have to choose the lower ranked players. This is easily accomplished by comparing the BSNs of the players comprised in S1 after the exchange - in the very same way as we did in the previous cases.

Any time a sorting has been established, any application of the corresponding D.1, D.2 or D.3

If we are lucky enough, the first attempt to a transposition, exchange, or MDP-exchange will yield the desired result. Often, though, we must persevere in the attempts until we

[37] *See B.1, p. 15.*

[38] *We want to remember that, because of C.7, the downfloaters from the previous bracket (i.e. the MDPs of the current bracket) have already been optimised. Thus, if we have an incompatible here, it means that there was no alternative at all. Hence, there is no going back to the previous bracket ("backtracking").*

rule, will pick the next element in the sorting order.

get a successful one. In this case, we must follow the order (sequence) established by the three rules above illustrated.

Ideally, we should start by establishing a full list of all the possible transformations - be them transpositions or exchanges of any kind - sorting that list by D.1, D.2 or D.3 (as the case may be), and then trying one after another until we find the first useful one[39].

E) COLOUR ALLOCATION RULES

Initial-colour

It is the colour determined by drawing of lots before the pairing of the first round.

The Initial-colour is not referred to any particular player. It is actually a parameter of the tournament – and the only one left to fate! – that allows the allocation of the correct colour to each player who has not a preference yet.

For each pair apply (with descending priority):

E.1

Grant both colour preferences.

E.2

Grant the stronger colour preference. If both are absolute (topscorers, see A.7) grant the wider colour difference (*see A.6*).

When two absolute preferences are involved, rule E.2 takes into consideration also the colour differences (see A.6) of the players. This, of course, may happen only for topscorers, and hence only in the last round (in previous rounds, a pairing with colliding absolute colour preferences is not allowed!). Let's consider the example of two topscorers with the same absolute colour preferences and the following colour histories:

1: WWBWBW
2: BBWBWW

Here, player #1 has a colour difference C_D=+2, while player #2 has C_D=0. Thus, we try to equalize the colour differences by assigning to player #1 his preferred colour.

Please note that this rule applies only to pairs in which both players have an absolute preference, while in all other cases the rule does not apply – e.g., in the pair:

1: BWWBWBW (strong preference, C_D=+1)
2: =BBWBWW (absolute preference, C_D=0)

the absolute preference shall be satisfied, no matter how large the colour difference is.

[39] *In common practice, exchanges and transpositions will be tried together (for each exchange, we will likely try one or more transpositions). To avoid mistakes, it is most advisable to annotate the last transformation (of each kind) used so that, on the following attempt, we can be sure about which element of the sequence is the next one.*

E.3

Taking into account C.04.2.D.5, alternate the colours to the most recent time in which one player had white and the other black.

To correctly manage colour assignments when one or both players have missed one or more games, we often need comparing colours histories by means of rule C.04.2:D.3.

For example, in the comparison between the colours histories of two players, the sequence == WB is equivalent to BWWB and WBWB (but the latter two are not equivalent to each other!).

E.4

Grant the colour preference of the higher ranked player.

We may want to pay particular attention to this point: in all other conditions being equal, the higher ranked player gets not white but its own preferred colour!

E.5

If the higher ranked player has an odd pairing number, give him the initial-colour; otherwise give him the opposite colour.

> *Note: Always consider sections C.04.2.B/C (Initial Order/Late Entries) for the proper management of the pairing numbers.*

When we get here, both players of the pair have no colour preference. Therefore, we use the Initial-colour decided by lot before the start of the tournament, to allocate colours to the players.

Of course, this rule will be used always in the first round (obtaining the usual results[40]), but it will be useful also in subsequent rounds, when we have a pairing between two players who did not play in the previous rounds (e.g. late entries or forfeits).

We ought to remember that players, who are actually entering the tournament only at a given round after the first – and who therefore were not paired in the previous rounds – in fact, do not exist, even if (seemingly) listed in the players' list. An obvious side effect of this is that we cannot expect all "odd-numbered" and "even-numbered" players to have the same colour as would be usual (viz., as they would have in a "perfect" tournament).

Actually, such late entries may have different effects on the pairing numbers, depending on how they are managed.

If we insert all the players in the list straight from the beginning, the pairing numbers will not change on the subsequent rounds, but the pairing of the first round will have to "skip" the absent players. For example, if player #12 is not going to play on the first round, players #13, #15, and so forth, who should seemingly get the initial-colour, will actually have the opposite colour; while players #14, #16, and so on will get the initial-colour.

If, on the contrary, we insert a new player only when it actually enters the tournament, we must find the correct place to put it. All the subsequent players will therefore have their pairing numbers changed, in order to accommodate the new entry. For example, if the newly inserted player gets #12, the previous #12 (who had colour opposite to the initial-colour) will now be #13; and so on for all subsequent players.

[40] *Please note that, if we are using an accelerated pairing system, the usual colour alternation is disrupted unless the first score group contains a number of players multiple of four.*

PART TWO – THE TOURNAMENT

1 FOREWORD

This chapter illustrates a step-by-step example of pairing procedure for a six rounds Swiss tournament by means of the FIDE (Dutch[41]) Swiss pairing system, in the hope to help those who wish to improve their knowledge of the system or get more familiar with it.

During the FIDE Congress in Abu Dhabi 2015, the Swiss Rules for the FIDE (Dutch) system were partially modified and reworded in order both to avoid misunderstanding in some points and to correct some peculiar behaviours in particular situations - and thus get better pairings in some instances[42]. In the following Congress in Baku 2016, the Swiss Rules were completely reworded, with the aim to make them clearer and easier – but this time without introducing behavioural changes.

Only a general knowledge of the FIDE (Dutch) system is required to follow the exercise, but keeping a handy copy of the Rules is advisable.

Before ending this short introduction, two side notes about language are in order: first, this work has not been intended for, nor written by, native speakers - hence, the language is far from perfect, but we hope that it will be easy enough to understand, and that any possible native speakers will forgive its many flaws. Second, and possibly more important, is that we definitely do not want to address a player as either man or woman. Luckily, English language offers a very good device to this end in the use of neutral pronouns - therefore, our readers are advised that our player will always be "it".

Warm and heartfelt thanks go to IA Roberto Ricca for his valuable and patient work of technical review and the many useful suggestions.

Happy reading!

***Notice**: to help the reader, the text contains many references to relevant regulations. These references are printed in italics in square brackets "[ ]" - e.g., [C.04.2:B.1] refers to the FIDE Handbook, Book C: "General Rules and Recommendation for Tournaments", Regulations 04: "FIDE Swiss Rules", Section 2: "General Handling Rules", item (B), paragraph (1). Since a great deal of our references will be made to section C.04.3: "FIDE (Dutch) System", these will simply point to the concerned article or subsection - e.g., [A.7.b] indicates item (b) of Article (7) of section (A) of those Rules. All regulations can be downloaded from the website of FIDE (www.fide.com).*

[41] *The FIDE (Dutch) Swiss pairing system, so named with reference to its promoter and developer, Dutch IA Geurt Gijssen, was adopted by FIDE in 1992. Its rules are codified in the FIDE Handbook, available on www.fide.com.*

[42] *For details about the changes, see the minutes of Abu Dhabi 2015 Congress in the Swiss Systems of Pairings and Programs Commission webpage, http://pairings.fide.com.*

2 INITIAL PREPARATIONS

The preliminary stage of a tournament consists essentially in the preparation of the list of participants. To this end, we sort all players in descending order of score[43], FIDE rating and FIDE title[44] *[C.04.2:B]*. Homologous players (i.e. those players who have identical scores, ratings and titles) will normally be sorted alphabetically, unless the regulations of the tournament or event explicitly provide a different sort rule.

Here we face our first problem: the FIDE (Dutch) system belongs to the group of <u>rating controlled Swiss systems</u>[45], which means that the resulting pairings depend *very closely* on the rating of the players - therefore, to get a proper pairing for the round, the players' ratings *need* to be the correct ones – i.e. they must correctly represent each player's strength. Because of this, the Rules require us to *carefully verify all of the ratings* and, when a player does not have one, to make an estimation as accurate as possible *[C.04.2:B.2]*. When a player has a national rating, but no FIDE rating, we can convert the first to an equivalent value - in some cases directly, in others by using appropriate formulas. For instance, when a player has no rating at all, we shall usually need to estimate its strength according to current practices or national regulations.

After we prepared the list as indicated above, we can assign to each player its <u>pairing number</u>, which is, at this stage, only *provisional*. Additional players may be allowed to join the tournament in later rounds and, in this case, we will need to reorder the list and, consequently, assign new and different pairing numbers[46] *[C.04.2:C.3]*.

Pairing Number	Player	Title	Rating
1	Alice	GM	2500
2	Bruno	IM	2500
3	Carla	WGM	2400
4	David	FM	2400
5	Eloise	WIM	2350
6	Finn	FM	2300
7	Giorgia	FM	2250
8	Kevin	FM	2250
9	Louise	WIM	2150
10	Marco	CM	2150
11	Nancy	WFM	2100
12	Oskar	--	2100
13	Patricia	--	2050
14	Robert	--	2000

Our tournament is comprised of 14 players. The players' list, already properly sorted according to *[C.04.2:B]*, is on the right.

Because of a perhaps a bit controversial (but none the less almost universal) language convention, players who are first on this list ("*higher*

[43] *Of course, at the beginning of the tournament all players have a null score, unless an accelerated pairing is used.*

[44] *The descending order for FIDE titles is GM, IM, WGM, FM, WIM, CM, WFM and WCM - followed by all untitled players [C.04.2:B.3.b].*

[45] *The "Rating Controlled Swiss Systems" belong to a more general class of "Controlled (or Seeded) Swiss Systems", in which the initial ranking list is not random or assigned by lots, but sorted according to given rules.*

[46] *Sometimes a player may be registered with a wrong rating, which needs to be corrected (this is necessary for rating purposes too). In such cases, the pairing numbers <u>may</u> be reassigned, but <u>only for the first three rounds</u>; from the fourth round on, pairing numbers shall not be changed, even if players' data have to be adjusted [C.04.2:B.4].*

ranked" players) are said to have the *highest* pairing numbers - in short, *number 1 is higher than 14*... This is somewhat odd – but, in time, it will become a habit.

The number of rounds is established by the tournament regulations, and *cannot be changed after the tournament has started*. We may want to notice that this number is, or should be, in close relation with the number of players, because a Swiss tournament can reasonably identify the winner only if the number N of players is less than or at most equal to 2 raised to the number T of rounds: $N \leq 2^T$. As a rule of thumb, each additional round enables us to correctly determine one more position in the final standings. For example, with 7 rounds we can determine the strongest player (and, therefore, the player who deserves to win) among at most 128 players while we will be able to correctly select the second best among only 64 players, and the third best only if the players are at most 32[47]. Thus, it is generally advisable to carry out one or two rounds more than the theoretical minimum: e.g., for a tournament with 50 players, 8 rounds are adequate, 7 are acceptable - while, strictly speaking, a 6 rounds tournament (which are the "bare minimum" with respect to the number of players) would not be advisable[48].

The preliminary stage ends with the possible preparation of "*pairing cards*", a very useful aid for the management of a manual pairing. They are sort of a personal card, the heading of which contains player's personal data (name, date and place of birth, ID, title, rating and possibly additional useful data) and of course the pairing number of the player. The body of the card is comprised of a set of rows, one for each round to be played, in which all pairing data are recorded (opponent, colour, float status[49], game result or scored points, progressive points). The card may be made in any of several ways, as long as it is

(GM) MANZONI Alessandro 2650 **1**
ITA 251260 ID 123456/ FIDE 890123

T	Opp	Col	Flt	Res	Pnts
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

[47] *This is true only if in every game the highest rated player ends up as winner. In practice, the occurrence of different results, such ad draws, forfeits and so on, may change the situation, making the individuation of a definite winner (the so called "convergence") either slower or faster, according to specific circumstances.*

[48] *Of course, this is just a theoretical point of view. In practice, many tournaments are comprised of 5 rounds, because sometimes this is the best we can put together in a weekend. Thus, the determination of the players who end up in the winning positions of the final standings must be entrusted to tiebreaks, which should therefore be chosen with the utmost care.*

[49] *See "Scoregroups and brackets", page 40.*

easy to read and to use. Here, we see a typical example.

The basic advantage of pairing cards is that we can arrange them on the desk, sorting them by rank, rearranging and pairing them in an easy and fast fashion. Nowadays, anyway, actual use of pairing cards has become pretty rare because an arbiter is very seldom required to manually make a pairing from scratch - but it's not unusual that an unhappy player asks for detailed explanations, so that the arbiter has to justify an already made pairing (usually produced by computer software). With a little practice, we can work out such an explanation right from the tournament board - which, in this case, needs to contain *all* of the necessary data, just like a pairing card. In this paper, we too will follow this latter method.

Now we will draw by lot the Initial-colour[50] *[see section E]*. The colours to assign for the first round to all players will be determined by this *[A.6.d, E.5]*. After that, we will be ready at last to begin the pairing of the first round. Let us say that *a little child, not involved in the tournament*, drew White as Initial-colour.

3 THE MAKING OF THE FIRST ROUND

The rules to make the first round are described in slightly different ways in Dubov and FIDE (Dutch) Swiss systems, but *the resulting pairings are always the same*. The players list, ordered as described above, is then divided into two subgroups, called S1 and S2; the former contains the first half, rounded down, of the players, while the latter contains the second half, rounded up[51] *[B.2]*:

$$\begin{cases} \mathbf{S1 = [\ 1, 2, 3, 4, 5, 6, 7]} \\ \mathbf{S2 = [\ 8, 9, 10, 11, 12, 13, 14]} \end{cases}$$ [52]

Now, we pair the first player from S1 with the first one from S2, the second one from S1 with the second one from S2 and so on, thus getting the (unordered) pairs {1-8, 2-9, 3-10, 4-11, 5-12, 6-13, 7-14}. Since this is the first round, unless there is some very special reason to do differently[53], there is nothing to stop these pairings – so, to complete the pairing process, now we just need to assign to each player its appropriate colour.

---

[50] *Some arbiters, misinterpreting the drawing of lots, assign colour at own discretion. It should be emphasized that the Rules explicitly require the drawing of lots (which, by the way, may be at the centre of a nice opening ceremony).*

[51] *When the number of players is odd, S2 will contain one player more than S1.*

[52] *Since names are inessential, from now on we will indicate players only by their own pairing numbers.*

[53] *For example, in certain events, we might have specific rules, or reasons, to avoid players or teams from the same federation or club meet in the first round(s), or at all. Such cases usually occur only in major international tournaments, championships, Olympiads and so on, while in "normal" tournaments, in practice, nothing of the kind happens.*

Since no player has a colour preference yet, all colour allocation shall be regulated per [E.5]. Hence, in each pair, the higher ranked player (who comes from S1) gets the initial-colour if its pairing number is odd, while it gets the opposite colour if the pairing number is even. Thus, players 1, 3, 5 and 7 shall receive the initial-colour, for which we drawn white, while players 2, 4, 6 shall receive the opposite, which is black.

The opponents to each player from S1 shall receive, out of necessity, the opposite colour with respect to their opponents; therefore, the complete pairing will be:

1 : 1 - 8
2 : 9 - 2
3 : 3 - 10
4 : 11 - 4
5 : 5 - 12
6 : 13 - 6
7 : 7 - 14

Before publishing the pairing, we have to put it in order *[C.04.2:D.9]* with the following criteria: 1) the score of the higher ranked player in the pair, 2) the sum of scores of both players, 3) the rank according to the initial order *[C.04.2:B]* of the higher ranked player. In the vast majority of cases, the FIDE (Dutch) system already generates pairings in the right order (but we always want to check).

At last, we are almost ready to publish the pairing. Before that, we want to check it once again and with extreme care, because *a published pairing should not be modified* *[C.04.2:D.10]*[54], except when two players should play with each other again.

In the event of an error (wrong result, game played with wrong colours, wrong ratings…), the correction will affect only the pairings yet to be done[55] and only if the error is reported by the end of the next round, *after which it will be taken into account only for the purposes of rating calculation* *[C.04.2:D.8]*. This means that, in such cases, the final standings will include the wrong result, just as if it were correct!

The last thing to do (and the Pairings Controller may do it while everyone is playing) is the compilation of the tournament board, on which we will post pairings and results for each player. When we renounce the use of pairing cards, as we do here, the board should also contain any other relevant information needed to compose the pairings for following rounds.

For each game, we should indicate *at least* opponent, assigned colour, and result – the choice of symbols is free, as long as it is *clear, unambiguous, and uniform*. Here we will

[54] *But, in this regard, see also FIDE Handbook C.05: "FIDE Competition Rules", item 7.4.*

[55] *Please note that this rule explicitly forbids the making of new pairings – which is a somewhat frequent request of players - in the event of an error.*

show each pairing by means of a group of symbols comprised of the opponent's pairing number, followed by a letter indicating the assigned colour (B for "Black", "W" for "White"); next, we can have some optional "utility" symbols, and finally the result ("+", "=" or "-", with obvious meaning).

Unplayed games are indicated in different ways, depending on their nature: a "PAB" indicates a Pairing-Allocated Bye, while "HPB" (Half Point Bye) or "ZPB" (Zero Point Bye) indicate an announced leave. In case of a forfeit (viz. a game that was scheduled but not played), we will use "+000" for the player who is present and "-000" for the absent one. Since we do not make use of pairing cards, our board will also show the players' progressive scores, which help us in the preparation of pairings (and of intermediate standings too).

After collecting the results of all the games, we can proceed to the pairing of the next round.

1	**1 (0.0) - 8 (0.0)**	**1-0**
2	**9 (0.0) - 2 (0.0)**	**0-1**
3	**3 (0.0) - 10 (0.0)**	**1-0**
4	**11 (0.0) - 4 (0.0)**	**½-½**
5	**5 (0.0) - 12 (0.0)**	**1-0**
6	**13 (0.0) - 6 (0.0)**	**0-1**
7	**7 (0.0) - 14 (0.0)**	**1-0**

4 SECOND ROUND (BYES, TRANSPOSITIONS AND FLOATERS)

Here is the tournament board after the first round:

Player	**PN**	**1**		**2**		**3**		**4**		**5**		**6**	
		Pair	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**
Alice	1	8W+	1										
Bruno	2	9B+	1										
Carla	3	10W+	1										
David	4	11B=	0.5										
Eloise	5	12W+	1										
Finn	6	13B+	1										
Giorgia	7	14W+	1										
Kevin	8	1B-	0										
Louise	9	2W-	0										
Mark	10	3B-	0										
Nancy	11	4W=	0.5										
Oskar	12	5B-	0	ZPB									
Patricia	13	6W-	0										
Robert	14	7B-	0										

Player #12 (Oskar) informed us in advance that he will not be able to play the second round, thus *he shall not be paired [C.04.2:D.4]* and shall score zero points[56]: hence, we already posted a "ZPB" in the tournament board. In this round we will then have an odd number of

[56] *Please note that the Tournament regulations may provide for a different score [C.04.2:D.4].*

players - hence, a player will end up unpaired and receive a "pairing allocated bye" (PAB): one point[57], no opponent, no colour *[A.5, C.04.1:c]*.

Scoregroups and brackets

Now players have different scores, and a basic principle of all Swiss pairing systems is that *paired players shall have scores as similar as possible [C.04.1:e]*. To achieve this result, we shall sort the players according to their scores. To this end, we define the *scoregroup*, which normally is a set of players who, in a given round, have identical scores[58] *[A.3]*.

A score group is the main component of a (pairing) *bracket*, which is a set of players to be paired among them. A bracket that contains just a scoregroup (i.e., all players in it have the same score) is called *homogeneous*. The pairing usually proceeds towards decreasing scores, one score group at a time, from the upper one (i.e. the one corresponding to the maximum score) to the lower one (corresponding to the minimum score) – hence, the first task in pairing a round is to divide the players into scoregroups.

In practice, it happens rather frequently that one or more players in a bracket cannot be paired within their own same bracket. Those players shall be moved down, to join the next scoregroup; therefore, in this bracket, they are called *downfloaters*.

The next scoregroup and those moved players, together form the next bracket, which contains players with different scores – and is thus called a *heterogeneous bracket*.

Such brackets will have to be treated somewhat differently from homogeneous brackets, because some players will meet opponents with different scores, usually called "floaters". A player moved down from the previous bracket (thus having a higher score) is called a *moved-down player* (MDP for short) *[A.4]*, while its opponent is usually said to be an "upfloater".

First, we divide and group players according to their score, thus forming the various scoregroups *[A.3]*. Then, as already mentioned, those scoregroups will be processed ("paired") one by one. We always begin with the topmost scoregroup, containing the highest ranked players; in this round, they have scored one point: they are *{1, 2, 3, 5, 6, 7}*. The next scoregroups, formed by the players who scored half point, contains *{4, 11}*. Last, we have

[57] *The usual score for a player who gets the PAB is that of a win (most commonly, one point). However, the Tournament regulations may provide for a different score.*

[58] *There is a particular situation in which we define the "Special Collapsed Scoregroup" (SCS for short), which is a scoregroup containing players with different scores [A.9]. We will not discuss SCSs now, but we will encounter such scoregroups later in the tournament.*

the scoregroup of the players who scored zero[59]: *{8, 9, 10, 13, 14}*.

Pairing parameters

Now, it is time to begin the real pairing. Since this is our first time, we will perform a detailed, systematic process. Then, as we proceed in the tournament, we will cut a little short on the more mundane tasks, to dwell only on the more interesting ones.

Let us start with the first bracket, which is constituted by the first scoregroup. The first step is determining some important parameters of the bracket *[B.1]*. The first one is the number **M0** of moved-down players, and it is very easy: since this is the very first bracket, there can be no MDPs at all – therefore, M0=0. Since there are no MDPs to be paired, also M1=0.

The second parameter is **MaxPairs**, or the number of pairs to be built in the bracket. There is no straightforward method to determine this number – actually, we must "divine" it – but an "educated guess" usually allows us to estimate it in a reasonably precise and reliable manner.

The first thing to consider is that the number of pairs cannot be larger than half the number of players, which is therefore an "absolute theoretical maximum" to the pairs we can build. The actual number of pairs may however be less than that, because of several reasons:

- there may be some players who, for whatever reason, cannot play with *any* other player in the bracket[60]. Such players are called *incompatibles*[61] (here, there is none).
- Sometimes it happens that in a bracket there are certain players who "compete" for the same opponent(s), in such a way that any, but not all, of them can be paired[62]. This situation is usually called "semi–(in)compatibility", or "island-compatibility".
- in some circumstances, the next bracket may require some floaters to be moved down to it, to make the pairing possible at all. We will come back to this problem later on.

In this bracket, the task is fairly easy: there are six players, and no incompatibles of any sort (it is too early in the tournament). We can thus safely guess that we will be able to build three pairs.

---

[59] *Please note that this scoregroup does not contain #12, who shall not be paired (because of its announced absence).*

[60] *There is no way to pair such a player in the bracket - therefore, the player can't help but go away, which means float to the next bracket.*

[61] *In a second round, anyone may still play with almost everyone else... hence, we usually can't have incompatible players – except when special circumstances arise, such as those already mentioned (see note 53, page 37).*

[62] *For example, consider the bracket {1, 2, 3, 4} in which players #1, #2, and #3 can all play only with player #4. No player here is incompatible: we can pair any one among them, but we cannot pair them all.*

Colour preferences

Each player who played at least one game has a *colour preference* (or *expected colour*). To determine it, we need first to define the *colour difference* C_D. This is simply the difference between the number W of *actually played* rounds in which the player had the white, and the number B of those in which it had the black: $C_D = W - B$ *[A.6]*. This difference is positive for a player who had more often white, negative if it had more often black – while it is zero if the colours are balanced. The latter is, of course, the ideal situation, and the pairing shall try to comply with it as much as possible.

The colour preference is determined as follows:

- A player has an *absolute colour preference* *[A.6.a]* when $C_D > 1$ or $C_D < -1$ – that is, when it had a colour (at least) twice more than the other one – or when it had the same colour for two games in a row[63]. The preference is towards the colour that it received fewer times, or respectively the colour that it did not receive in the last two games. In any case, the player *must* receive its due colour (and we shall write it right away on the pairing card or on the tournament board). *The only exception may happen in the last round, for a player with more than half of the maximum possible score* (this is called a "*topscorer*", see *[A.7]*) *or its opponent [C.3]*: in this case, indeed, top ranking positions may be at stake, and pairing players of equal scores is therefore particularly important. In all other cases, the colour preference *shall* be honoured, period. It is an *absolute* criterion and, in order to obey it, players may float as necessary.
- A player has a *strong colour preference* *[A.6.b]* when $C_D = \pm 1$ (i.e. when it had a colour once more than the opposite), the preference being of course for the colour it received fewer times *[C.04.1.h.1]*.
- If $C_D = 0$, the player has a *mild colour preference* *[A.6.c]* for the colour opposite to what it had in the previous game, so as to alternate colour in its history[64] *[C.04.1.h.2]*.
- Finally, a participant, who did not play any games yet (a "*late entry*", or a player who received a PAB in the first round, or was involved in a forfeited game), has no colour preference at all *[A.6.d]* and receives the colour opposite to that awarded to its opponent.

---

[63] *Please note that we always refer only to actually played games: the colour allocated for a game that was forfeited, is irrelevant, and shall be ignored.*

[64] *The "colour history" of a player is the sequence of colours it received in the previous rounds.*

Strong and mild colour preferences may be disregarded, when necessary, so that the player might also get the colour opposite to its preferred one. In all such cases, however, this player gains an *absolute* colour preference for the next round.

During the pairing process, we need to keep colour preferences for each player handy. To avoid the use of yet another table, we will temporarily record all colour preferences in the tournament board, in the column bound to the pairing for the round (when it is time to post the pairings, we will not need the preferences any more).

Now, we want to establish a code to indicate the various kinds of colour preferences[65]:

– A lower case "w" or "b" indicates a ***mild*** colour preference

– A couple "Ww" or "Bb" indicates a ***strong*** colour preference

– A capital "W" or "B" indicates an ***absolute*** colour preference

– A capital "A" indicates a player who has no colour preference (as it happens, we have no such players in this tournament).

We should now determine the colour preference for each player, and we do that by examining the colour history in all previous actually played games of the player.

The round we are going to pair is an even numbered one. Hence, any participant who did not miss any games played an odd number of them, thus obtaining a strong colour preference (this early in the tournament, we cannot have absolute colour preferences yet). In the bracket, we will indicate such preferences by means of the above defined symbols, putting them right after the player's pairing number: *{1Bb, 2Ww, 3Bb, 5Bb, 6Ww, 7Bb}*.

We already estimated that MaxPairs (or the maximum number of pairs) for this bracket is three. Now we want to check how many of these pairs cannot fully satisfy the colour preferences: here, two players expect white and four expect black – out of the three pairs, at least one will necessarily include two players who both expect black – and therefore one player who shall receive a colour different from its preference. The minimum number of pairs that shall contain a disregarded preference is usually called $\boldsymbol{x}$. In a perfect pairing, the number of disregarded colour preferences will be exactly $\boldsymbol{x}$. We can compute this number easily enough by taking the integer part of half the difference between the number of players expecting white and the number of players who expect black – any number of players without any preference would be counted as having the same preference of the minority (but

[65] *Please note that this code is far from universal and other papers may use completely different codes.*

we have no such players here). Out of necessity, we will consider any pairing that contains ***x*** pairs with disregarded preferences as perfect (having less than that is simply impossible). However, we will not accept any pairing that contains more than that – unless, of course, when this is unavoidable *[C.10]*.

In the present instance, we have W=2, B=4, A=0 (all players have a preference) – hence, ***x*** = (4-2-0)/2 = 1, which means that, as anticipated above, at least one pair must contain a disregarded colour preference. This disregarded preference will of course be one among the most numerous – that is, for black, which is here called the "majority colour".

Another parameter we want to know is the *minimum number z of pairs in which it will be necessary to disregard a strong colour preference*. This number will help us in the process of optimization, which requires that, if we have to disregard a colour preference, it should be a mild one rather than a strong one[66]. This number is obtained by subtracting from ***x*** the number of players with mild preferences for the colour of the majority.

However, if, apart from the absolute preferences, in a bracket we have only strong ones, or only mild ones, this parameter is useless – just as criterion C.11 is – and we may omit them both.

In the current bracket, we shall have to disregard at least one (black) strong preference.

Preparation of the candidate

Now we can divide the players of the bracket between subsets S1 and S2 *[B.2]*. We put into S1 the first MaxPairs players of the bracket (in this case the first half of the players), while the rest (namely the second half) ends up in S2:

S1 = [1Bb, 2Ww, 3Bb]
S2 = [5Bb, 6Ww, 7Bb]

Then we sort each of the subgroups according to the usual rules *[A.2]*. This order normally coincides with the original one, and so there is no need to do anything unless we got to this point after an exchange[67] of players between S1 and S2.

So far, we only performed the necessary preliminary steps – now we are ready to prepare a candidate, which is a tentative pairing built as explained in [B.3]. To build it, we associate the first player of S1 with the first player of S2, the second player of S1 with the second player of S2, and so on, just as we did for the first round, thus obtaining the (tentative) pairs:

[66] *Of course, absolute colour preferences can never be disregarded – the only possible exception being for topscorers or their opponents, during last round.*

[67] *We will first meet exchanges during the pairing of the third round (see page 54).*

S1	S2
1Bb	5Bb
2Ww	6Ww
3Bb	7Bb

Evaluation of the candidate

Now, having built a candidate, we proceed to evaluate it *[B.4]*. First, we must check it for compliance with the absolute criteria C.1 (players who already met) and C.3 (clashing absolute colour preferences). Criterion C.2 does not apply to this bracket, because it is not the last one, so we do not have to allocate a PAB here. Criterion C.4 does not apply because this is an ordinary bracket and not a PPB. A candidate, which, as this one, complies with all the (relevant) absolute criteria, is said to be *legal* and may be evaluated for quality. A candidate that is not legal is instead immediately discarded.

Now that we are sure that the absolute criteria are obeyed, we must evaluate the compliance of the candidate with quality criteria C.5 – C.19. This compliance is measured by means of a series of *failure values*. Those are numerical values that describe "how good" the pairing is – the lower the failure value is, the better is the candidate.

Of course, as we will presently realize, not all criteria need to be considered in any situation – in many instances, some of them are simply irrelevant[68]. Therefore, we usually limit our attention only to the significant ones. However, since this is the very first time we pair a non-trivial bracket, let us briefly examine them all, one by one.

To synthesise the global quality of the candidate, we build a simple table in which we will accommodate the failure values of the bracket for each one of the criteria, obtaining a sort of "report card" for the candidate.

C.5	C.6	C.7	C.8	C.9	C.10	C.11	C.12	C.13	C.14	C.15	C.16	C.17	C.18	C.19

We must now fill our report card with the failure values; let us then examine the candidate with reference to each criterion, and determine the respective failure values.

[C.5] maximises the number of paired players – or, (almost) equivalently, minimises the number of downfloaters. The simplest (and most obvious) choice for the failure value here is the *number of players that we cannot pair*[69]. In our

[68] *For example, criteria C.8 and C.9 concern only topscorers and their opponents – hence, they apply only in the last round of the tournament, and only to some brackets, thus being completely irrelevant in most situations.*

[69] *Please note that the choice of the failure value for a criterion is largely arbitrary, in that it may be any of a (infinite) number of functions (in the mathematical sense of the word) – but the simplest and more natural choice is of course the number of failures (unpaired players, disregarded colour preferences and so on).*

example, we need to build three pairs and we are going to make all of them – since we are making all the possible pairs, the failure value is zero.

[C.6] minimises the overall difference of scores between paired players (PSD, see A.8). This very important parameter is always zero in any candidate pairing of a homogeneous bracket, because all players have by definition identical scores (actually, trying to apply that criterion in a homogeneous bracket makes no sense at all). In fact, this criterion only applies for heterogeneous brackets, where minimising the PSD is practically equivalent to pair as many MDPs as possible, and the most natural failure value is the PSD itself. However, we will let the matter be for the time being, to be back there later on.

[C.7] chooses the best downfloaters, viz. those that best pair the next bracket, maximising the number of pairs and minimising the pairing score difference. Of course, this criterion does not apply in any bracket that, as the present one, does not produce downfloaters at all. The failure value here is actually not a single number as usual, but a couple of numbers: the first one indicates the number of pairs that cannot be built in the next bracket, while the second one indicates the PSD in the next bracket.

[C.8], [C.9] minimise the numbers of topscorers, or topscorers' opponents, who get a colour difference with an absolute value higher than 2 *[C.8]*; or who get the same colour three times in a row *[C.9]*. These criteria only apply when pairing the last round, and only in processing those brackets that actually contain topscorers. The failure values would be the numbers of topscorers, or topscorers' opponents, mentioned above. In the present bracket, both those numbers are zero, as there are no topscorers at all (this is not the last round!).

[C.10] minimises the number of players whose colour preference is disregarded. The failure value for this criterion is the number of players who do not get their colour preference. The minimum number of colour preferences that shall unavoidably be disregarded in a bracket is $\boldsymbol{x}$, and we already know that this number is not necessarily zero. Therefore, to determine if the present candidate is a perfect one *[B.4]*, we must compare the number of colour preferences actually disregard in the candidate against the number $\boldsymbol{x}$, for which we found a value $\boldsymbol{x}$=1[70]. A direct inspection of the candidate shows that three out of three

[70] *See the computation of* ***x*** *for this bracket in "Colour preferences" (page 43).*

pairs contain one disregarded colour preference – hence, the failure value for this criterion is 3 and, since the reference value is 1, *this candidate is not perfect*.

[C.11] minimises the number of players whose strong colour preference is disregarded. The failure value for this criterion is the number of players who do not get their strong colour preference, and the minimum for this number is ***z***. All the considerations just made about C.10 also apply in respect to this criterion[71].

[C.12] … [C.15] minimise the number of players who, having floated in the last two rounds, float again in this round. The failure value for each one of these four criteria is the number of floaters of the two kinds that float again. Since this bracket contains no floaters, all the failure values are zero. For the time being, we will not discuss those criteria any further – but we will go back to the matter later on.

[C.16] … [C.19] Minimise the score difference of players who receive a same float as they already got in the last two rounds. For these criteria, we will use as failure values the SDs of the involved players. As mentioned above, in the current bracket there are no floaters – hence, the failure values are all zero.

Now we can synthesise the global quality of the candidate, filling in the failure values of the bracket for each one of the criteria into the report card.

C.5	C.6	C.7	C.8	C.9	C.10	C.11	C.12	C.13	C.14	C.15	C.16	C.17	C.18	C.19
0	0	0/0	0	0	3	3	0	0	0	0	0	0	0	0

Looking for a better candidate: transpositions

As we mentioned above, *this candidate is not a perfect one*, because the number of disregarded colour preferences is larger than the bare minimum (***x***). We must then take some action to obtain a different candidate, looking for a better one. The process is simple enough: we alter the subgroups S2 and (if needed) S1 looking for a perfect, or at least better, candidate *[B.5]*. After each alteration, we must build and evaluate a new candidate, just in the same way as we did it above.

Since this is a homogeneous bracket, we are directed to rule B.6, which instructs us to first apply a *transposition of S2*. We will try all the possible transpositions, one by one, until we find the first one among them that gives a perfect pairing. Only if that procedure fails to

[71] *Please note that, in the current bracket, the colour preferences are all strong, and therefore criterion C.11 is useless and may be ignored. However, since this is our first example, we will consider it all the same.*

yield an acceptable result, we will also apply one or more *exchanges between S1 and S2* – and, for each exchange, we will look for the first useful transposition as described above.

The first step is thus to try to alter the subgroup S2 applying a transposition, to see if we can reach the goal. A transposition changes the order of the players in S2, starting with the lowest ranked players and then gradually moving towards higher ranks - until an acceptable solution is found. The method to do that is explained in section D, which gives the rules for the sequential generation of candidates.

Before applying any transposition or exchange, each player is temporarily labelled with a number, representing the player's ranking order in the bracket. This number is called "In-Bracket Sequence Number", BSN for short. The BSNs will help us to keep track of the transformations we are going to apply to the subgroups, as they never change when we transpose or exchange players. In our bracket, there are six players, who shall be labelled with numbers from 1 to 6[72]:

player	1Bb	2Ww	3Bb	5Bb	6Ww	7Bb
BSN	1	2	3	4	5	6

We are going to apply a transposition, which involves only players in S2, viz. {4, 5, 6}. The transpositions of the bracket are represented by all the possible dispositions of those three numbers, sorted in lexicographic order – which in practice means to arrange in ascending order all numbers that can be constructed with these figures (in our case: 456, 465, 546, 564, 645, 654) *[D.1]*. The first of those transpositions always corresponds with the basic order of the subgroup, which was used in the first pairing attempt. Therefore, we now start trying from the second transposition on, which is 465, or [5Bb, 7Bb, 6Ww]:

S1	**S2**
1Bb	5Bb
2Ww	7Bb
3Bb	6Ww

In this candidate pairing, the pair 1-5 does not meet all of the colour preferences, while the subsequent 2-7 and 3-6 do. Hence, the failure values for C.10 and C.11 are now both 1. Since we already know that (at least) one pair shall disregard a colour preference, this

[72] *Of course we could also choose any other set of numbers (or, why not, hexadecimal figures, letters of the alphabet, random words…), as long as they form an arithmetical progression (a sequence of equally spaced numbers in strict ascending order) – in this case, the rules just indicate the easiest possible choice.*

candidate is perfect, and we accept it[73]. The pairs formed are hence [(1,5), (2,7), (3,6)]. Colours to be assigned to each player remain yet to be defined – we will do that only after the pairing of all players is complete.

Now, this latter candidate is the first perfect one that we found, and there can be no better and earlier candidate than this one, which is therefore immediately accepted. Hence, there is no need to compare it with the previous one. However, we will do that all the same, just as a useful exercise in evaluating the candidate. First, we prepare the report card, and put it side-to-side with that of the previous candidate:

	C.5	C.6	C.7	C.8	C.9	C.10	C.11	C.12	C.13	C.14	C.15	C.16	C.17	C.18	C.19
new	0	0	0/0	0	0	1	1	0	0	0	0	0	0	0	0
old	0	0	0/0	0	0	3	3	0	0	0	0	0	0	0	0

Now, to compare the candidates, we start from the first failure value (C.5) and compare them: if one of the two is smaller than the other one, that one of course relates to the better candidate, and the comparison process stops here. As long as the failure values are identical, the candidates are still "equivalent" and we proceed downwards to the next failure value (C.6) – then to C.7 and so forth, always operating in the same way, possibly until the end (C.19). If, in the end, there were no differences at all, the candidates are actually equivalent (from the point of view of the quality of pairing) and therefore we must choose the one which was produced first of the two in the sequence of generation.

This kind of complete comparison is of course of great theoretical value. However, when we operate by hand, in practice we will almost never need to make recourse to this explicit formal procedure – in general, it will suffice to compare the relevant failure values.

Towards the next bracket – The "Requirement Zero"

Now the bracket has almost been paired. The last step, before proceeding to the next bracket, is to verify that this round-pairing can actually be completed. To do this, we must prove that at least one pairing exists for all the yet unpaired players. This is informally called the "Requirement zero". We do not need to find the correct pairing now; we are only looking for *any* legal pairing, and quality is no concern of ours, so that this is, of course, a far easier task – actually, it is usually easy enough[74].

[73] *It is worth noting that, since we choose the first useful transposition, it is possible (and even statistically likely) that the pairs in which we find the disregarded colour preferences are formed at the top of the bracket. Note that this may be different from what happens with other Swiss systems.*

[74] *We need no special method to find this acceptable pairing – it is just the first legal mesh-up we can think of!*

First, we need to define the set of players yet to be paired – we usually call it the "rest"[75]. Looking into the crosstable, we readily find that the rest is {4, 8, 9, 10, 11, 13, 14}. Then we can immediately find a legal pairing, e.g. [4-8, 9-10, 11-13, 14-PAB]. Of course, this pairing, which is "almost random", will almost certainly not be the correct one – but this is irrelevant: as we mentioned, *we just want to be sure that at least one legal pairing exists*. Since we just found one, we are now certain that this pairing can be completed.

By the way, we ought to note that this check, which is called a "Completion test", is essentially useless in the early rounds of the tournament, because only very few players have already met and it is therefore virtually impossible that the pairing does not come to fruition. It will however become more and more important as we proceed in the tournament, especially if there are not many players.

The next bracket

Now, let us move to the next bracket, which contains the players who have scored 0.5 points, namely *[4Ww, 11Bb]*. We know that player #4 already played with #11 in the first round. Thus, it has no compatible opponent in the bracket. Since no possible pairing exists, we have no other option but to move both players #4 and #11 down to the next bracket.

From a formal point of view, we can say that our initial estimate of MaxPairs was wrong, and we must thus mend it – actually, the correct value is zero, and therefore the perfect candidate will have no pairs and two downfloaters.

Those players, who are downfloaters in this bracket, will be *moved-down players* (MDPs) in the next bracket, where they are going to play against opponents with lower scores. Likewise, their opponents, who are usually called "upfloaters", will play against higher scoring opponents.

The pairing of two players with different scores, although sometimes unavoidable, is a violation of the basic principles of Swiss systems *[C.04.1.e]*. Therefore, in order to avoid making players float too often, every player who is going to play with a lower-scored opponent, receives a special flag, which is called a *downfloat*. In the same way, every player who is going to play with a higher-scored opponent receives a special flag called *upfloat*. We mark those events on the players' cards, or on the tournament board, respectively with a downward arrow "↓" (often replaced for convenience by a lowercase "v") for downfloats; or with an upward arrow "↑" (often replaced by a "^") for upfloats. The pairing system protects

75 *Please note that this rest has nothing to do with the* remainder*, which is the residual part of a heterogeneous bracket, after the MDP-Pairing (see B.3).*

players from repetitions of a same kind of floating, limiting such repetitions for the next round *[C.12, C.13]* and for the following one *[C.14, C.15]*.

Before proceeding to the next, and last, bracket, we should verify that the selected downfloaters maximise the pairing in the next bracket, and that the Requirement Zero is satisfied. Actually, there is no need to do this now, because the new bracket will contain all the rest from the previous bracket, and we already know that that rest can be paired (because it passed the completion test at the end of the pairing for the previous bracket).

The last bracket

Having exhausted (so to speak...) the half point bracket, we finally go to the last and lowest bracket, namely the one with zero points. This is a heterogeneous bracket, since it contains not only players with zero points, but also the two half point downfloaters from the previous bracket. For clarity, we keep downfloaters separated from other players: [4Ww 11Bb] [9Bb 8Ww 10Ww 14Ww 13Bb] (player #12 is absent, and therefore is not in the list and receives a zero points forfeit, with no opponent and no colour, which is a downfloat too).

Before proceeding with the pairing, we try to compute the usual parameters. First, we have 7 players; since this is the last bracket, we are certain (thanks to the completion test!) that we will be able to pair all them, except the "odd one", who shall get the PAB. Hence, for once we *know* that **MaxPairs=3**. Moreover, we have two MDPs, **M0 = 2**; and we have no reason to think that they cannot be paired – therefore, at least for the moment[76], we can safely assume **M1 = 2**. Four players "prefer" white, while three prefer black. Hence, we should be able to satisfy all colour preferences and ***x* = 0**, while *z* is useless.

Since the bracket is a heterogeneous one, the procedure is a little different from that that we used for the previous brackets *[B.3]*. The formal procedure is as follows: first, putting in S1 only the MDPs, we build an MDP-Pairing, obtaining some pairs and a remainder. The latter is then paired in the same way as a homogeneous bracket, giving some more pairs. All the pairs, put together, form the complete pairing of the bracket, which shall be evaluated.

We shall therefore put in S1 only the two MDPs, to form two pairs. The initial pairing scheme is:

[76] *The parameter M1 must be "divined" in much the same way as MaxPairs. As we mentioned, an educated guess usually allows us to estimate its value, but sometimes we may find our estimate wrong, and therefore we have to correct it. Nonetheless, we should remember that this parameter is not a variable but a constant of the bracket.*

S1	S2
4Ww	8Ww
11Bb	9Bb
	10Ww
	13Bb
	14Ww

The resulting MDP-Pairing *[B.3]* is 4-8, 11-9. We should then proceed to build the remainder and pair it. However, of course, we do not need to be as dumb as a computer: in this case, it is apparent that the colour matching is unsatisfactory in both the pairs of the MDP-Pairing – while, since $\boldsymbol{x} = 0$, we know we should satisfy all colour preferences. Hence, we go straight on to B.5, where we learn that we should apply B.7 to straighten the situation.

The first attempt should be to apply transpositions and exchanges in the remainder, to try to make the pairing better; but, of course, *no alteration in the order of the remainder can possibly change the MDP-Pairing*. Therefore, we boldly jump straight to the next step, applying a transposition to the S2 subgroup *of the complete bracket*, in order to modify the MDP-Pairing (perhaps changing the remainder too, but for the moment we do not care).

Once again, we must thus make use of rule D.1, which give instructions as to how to generate all the transpositions in the correct order. For heterogeneous brackets, this rule gives us a most practical hint, specifying that we are interested in the lexicographical order of only the first N1 elements of S2, where N1 is the number of elements in S1 – in our case, this is the number of MDPs to be paired, which is M1=2. Hence, having assigned the BSNs to the players of the bracket in the usual way [1, 2][3, 4, 5, 6, 7], we only need to focus on the first two elements of the list for S2. There, we need to change both the pairings, and the first transposition that meets this need is [4, 3, x, x, x] – while, of course, any other transpositions brings higher BSNs in the first two positions and is hence higher in the lexicographical order[77]. The next candidate to try is therefore:

S1	S2
4Ww	9Bb
11Bb	8Ww
	10Ww
	13Bb
	14Ww

We thus obtain the pairs 4-9 and 8-11 for the MDP-Pairing, which seem to be satisfactory, and can now proceed to the pairing of the remainder, which is {10Ww, 13Bb, 14Ww}. We begin by building the subgroups S1R and S2R.

[77] *A not rigorous but simple way to see the procedure is as follows: take the first player of S1, then scroll S2 until a match is found, keeping in mind that we have to make* $\boldsymbol{x}$ *pairs containing a disregarded colour preference. Then repeat this procedure with the second element of S1, the third, and so on, until all of S1 is used up.*

The pairing scheme is now:

S1R	**S2R**
10Ww	13Bb
	14Ww

The pairing of the remainder is 10-13; player #14 ends up unpaired. We can now put together the MDP-Pairing and the remainder pairing, to build the complete candidate, which is 4-9, 8-11, 10-13, (14). We must now evaluate this candidate: since it complies with all the pairing criteria, it is perfect – and is therefore immediately approved. Player #14, as directed by the Rules, receives a PAB: 1 point, no opponent, no colour *[A.5]*. The player who receives the PAB also receives a downfloat *[A.4.b]*, which is annotated on the player's card.

To complete the preparation of the round, we now assign colours and rearrange chessboards. The unordered pairs we built are: 1Bb-5Bb, 2Ww-7Bb, 3Bb-6Ww, 4Ww-9Bb, 11Bb-8Ww, 10Ww-13Bb; #12 is absent, while bye goes to #14. We need to examine those pairs one by one, accordingly to colour allocation criteria (see part *E* of the Rules), which are very logical and reasonable:

– If possible, we satisfy *both players* *[E.1]*;

– If we can't satisfy both players, we satisfy the *strongest colour preference*: first are absolute preferences, then strong ones, mild ones come last *[E.2]*;

– All above being equal, we *alternate colours* with respect to the last time they played with different colours *[E.3]*. It may happen that in the sequence of colours (or "*colour history*") there are "holes", of course in correspondence with unplayed games (due to a bye or forfeit). In this case, we simply skip those "holes", moving them to the beginning of the sequence – basically, this means that we look at the colour of the previous played game.

– All above being still equal, we satisfy the colour preference of the higher ranked player – thus, the player with higher score or, if scores are tied, the one who comes first in the initial ranking list *[E.4]*.

The last item is just the one that applies in assigning colours to the pair 1-5: the players in this pair have the same colour preference and identical colours histories. We shall therefore assign black to player #1. In all other pairs, we can satisfy both players - and so we shall do.

Having thus finished the preparations for the second round, we check the order of chessboards and publish the pairing (indeed, to cut it short we post the results too):

1	**5 (1.0) - 1 (1.0)**	**1-0**
2	**2 (1.0) - 7 (1.0)**	**1-0**
3	**6 (1.0) - 3 (1.0)**	**½-½**
4	**4 (0.5) - 9 (0.0)**	**1-0**
5	**8 (0.0) - 11 (0.5)**	**0-1**
6	**10 (0.0) - 13 (0.0)**	**1-0**
7	**14 (0.0): PAB**	**1F**

5 THIRD ROUND (EXCHANGES)

We must now pair the third round. We already had a little practice, so we can go a bit faster – but without neglecting any of the necessary checks and cautions!

As usual, our first task is the determination of all the colour preferences. We want to notice that player #5 has an absolute colour preference – hence, we know right from the beginning that the player *shall* be assigned its due colour. We may also want to observe that players #12 and #14 played one game less than the others did, so their colour preference is strong, while all remaining players have only mild preferences.

There are no absent players in this round, so all players shall be paired. The tournament board (see below) has been duly updated with all the relevant data (opponents, colours, results, scores, floats, colour preferences).

Player	**PN**	**1**		**2**		**3**		**4**		**5**		**6**	
		Pair	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**
Alice	1	8W+	1.0	5B-	1.0	w							
Bruno	2	9B+	1.0	7W+	2.0	b							
Carla	3	10W+	1.0	6B=	1.5	w							
David	4	11B=	0.5	9W↓+	1.5	b							
Eloise	5	12W+	1.0	1W+	2.0	**B**							
Finn	6	13B+	1.0	3W=	1.5	b							
Giorgia	7	14W+	1.0	2B-	1.0	w							
Kevin	8	1B-	0.0	11W↑-	0.0	b							
Louise	9	2W-	0.0	4B↑-	0.0	w							
Mark	10	3B-	0.0	13W+	1.0	b							
Nancy	11	4W=	0.5	8B↓+	1.5	w							
Oskar	12	5B-	0.0	ZPB↓	0.0	Ww							
Patricia	13	6W-	0.0	10B-	0.0	w							
Robert	14	7B-	0.0	PAB↓	1.0	Ww							

From the board, we can build the five scoregroups. For our convenience, we indicate colour preference and possible float markers for the last and previous rounds[78] for each player. Let

[78] *The convention we adopt, which is widespread enough, uses a downward arrow for a downfloat and an upward arrow for an upfloat. A single arrow indicates a float had in the last round. When there are two arrows, the first (from left to right) indicates the float marker for the last round, while the second indicates the float marker for the previous (last-but-one) round. When we have to indicate a float in the last-but-one round but no float in the last round, we put a "-" sign in the first place – e.g., "-↓" indicates a downfloat in the last-but-one round, and no float in the last round.*

us notice that the number of upfloats can be less than or equal to the number of downfloats. In this round, in fact, we have more downfloats than upfloats – this is normal and happens every time there are byes or unplayed games.

Before even starting the pairing, we want to check that the players can actually be paired (the by now usual "Requirement Zero"). This is easy. For example, looking at the crosstable we see that (2-5, 3-4, 6-11, 1-7, 10-14, 8-9, 12-13) is a legal pairing[79], so we can proceed.

Here are the scoregroups:

2.0	{2b, 5B}
1.5	{3w, 4b↓, 6b, 11w↓}
1.0	{1w, 7w, 10b, 14Ww↓}
0.0	{8b↑, 9w↑, 12Ww↓, 13w}

As usual, we process the above scoregroups one by one from top to bottom, building and pairing the brackets as we go. By the way, there is no half point scoregroup. This happens now and then and is completely normal. After having paired the one point scoregroup, we simply proceed to the next one, which is the one with zero points.

The first bracket, whose players scored 2 points, is [2b, 5B] (***MaxPairs=1, M1=M0=0, x=1***, while ***z*** is ignored because there is just one non-absolute preference)[80]. We are requested to form just one pair. Since the two players have not played each other, they can be paired. This is the only one possible pairing, and it is legal – therefore, there is no need to evaluate it, no better pairing can exist! However, we must perform (as always) the completion test, to see that the rest of the players can actually be paired. This check is straightforward (it may be descended right from the previous one), and is successful, so we may proceed.

To complete the pairing for the bracket, we only need to allocate colours[81]. Here, we must satisfy the stronger colour preference, which is the absolute one of player #5, so the pairing is 2-5. As already mentioned, we may as well observe that player #2 will have an absolute colour preference for black on the next round.

The next bracket, with 1.5 points, is [3w, 4b↓, 6b, 11w↓] (***MaxPairs=2, M0=M1=0, x=0***). Before starting the pairing, we may want to look at the bracket and take notice of its possible

---

[79] *Once again, we want to emphasise that we do not care in the least how bad this pairing may be (and, by the way, actually is) – it is a legal one, and this is all we need to be certain that at least one legal pairing exists!*

[80] *From now on, we will make explicit reference to the computation of the bracket parameters only when necessary; however, their values are always found as previously discussed.*

[81] *In the previous rounds, we (correctly) left the allocation of colours as a final phase of the pairing, to better focus the problems of pairing itself. However, after the completion test succeeds, we will never have to change the pairing of the bracket, so we have no strong reason to delay the colour allocation.*

peculiarities. For example, here we observe that we already had the games 6-3 and 11-4, and that players 4 and 11 just had a downfloat. Such information will help us not to waste time and efforts in the proceedings. And now, to business! The initial candidate is:

S1	**S2**
3w	6b
4b↓	11w↓

Here, both pairs are forbidden (the players already played each other *[C.1]*). Therefore, <u>this candidate is not a legal one</u>, and we cannot help but reject it.

Therefore, to build the next candidate, we move on to the first transposition (which, in this case, is also the only one) *[B.5, D.1]*:

S1	**S2**
3w	11w↓
4b↓	6b

Still we are not lucky: this candidate contains two pairs that disregard colour preferences – therefore, since $x = 0$, <u>*it is not perfect*</u>. However, we store it somewhere as a "provisional-best". If we cannot find a better candidate, this one will still be usable. As we build more candidates, we will evaluate them against the current provisional-best, always keeping the better of the two as provisional-best. If at any time we find a perfect candidate, we choose it at once, discarding any possible provisional-best. However, after all the possible candidates are used up, if we found no perfect candidate at all, the surviving provisional-best is actually the best candidate we can have, and therefore the one to be used.

Since this was the last possible transposition, and the bracket is homogeneous, we must try a <u>*resident exchange*</u>, which is a swap of resident players between S1 and S2 *[B.6, D.2]*. We take a player from S2 and swap it for a player from S1, in an attempt to obtain an acceptable pairing. If the exchange of one player is not enough, we can swap two, three, and so on[82] - until we find a solution (or use up all the possible exchanges).

[82] *Let us consider the example of a six players bracket {[1,2,3][4,5,6]}. Let us exchange players 3 and 4 between S1 and S2, thus obtaining the new subgroups composition {[1,2,4][3,5,6]}. This exchange is "useful", because it gives us candidates that were not previously found (e.g., 1-3, …). Now, let us exchange players (3,5), then (2,4), then again (2,5), and so on. Each new exchange gives some new candidates, so all these exchanges are useful (at least until we try to exchange player #1: actually, this player was already paired with all possible opponents during the previous exchanges, so this exchange is useless). However, candidates containing pairs formed by the exchanged players are <u>always</u> useless, because they were already checked – e.g., exchanging (3,4) we obtain among the others, also the candidates (1-5, 2-6, 4-3) and (1-6, 2-5, 4-3) that were already examined as (1-5, 2-6, 3-4) and (1-6, 2-5, 3-4) – and this is easily extended to brackets containing <u>any</u> number of players. Now, let us consider the exchange of two players per subgroup, e.g. {[1,4,5][2,3,6]}: now, every candidate contains at least one pair formed by exchanged players and is therefore useless. We may see that this situation happens in any bracket, each time the number of exchanged players is greater than half the number of players in S2 (which is the largest subgroup). We conclude that <u>there is a theoretical maximum to the number of useful exchanges in the bracket, which is half the number of players in S2</u> (rounded down if needed).*

A single exchange (one player for one player) is usually rather easy, but the matter may become definitely tricky when more players are involved. To avoid errors in the exchange, we want to follow the general rules always:

– first, we shall exchange as few players as possible – this item needs no explanation...

– then, we exchange the smallest possible difference between the sum of BSNs of the moved players. To clarify this rule, let us consider the example of an exchange between players P1a and P1b from S1 and players P2a and P2b from S2 – all represented by their BSNs. From the principle that the exchanged players must be as near in ranking as possible, we obtain that, on the whole, the differences between the BSNs of exchanged couples must be as little as possible. To comply with this principle, the rules choose to minimise the difference of the sums, which is equal to the sum of those differences[83]

– then again, when we have two possible exchanges with the same difference, we must decide which one is to be tried first of the two. The first criterion it to choose the lowest possible exchanged player(s) in S1 (and therefore the greatest different BSN[84]). For example, exchanging players (3, 4) from S1 is better than exchanging players (2, 4) – but it is worse than exchanging players (1, 5), because, *in S1, exchanging #5 is always better than exchanging #4*, whoever the higher ranked players in the exchange may be

– finally, when two exchanges have not only the same differences, but also the same exchanged players from S1, we choose the exchange with the higher players(s) of S2 (and therefore the one with the lowest different BSN). Just as in the previous case, this may sometimes lead to seemingly peculiar choices. For example, exchanging players (7, 10) is better than exchanging players (8, 9) , but is worse than exchanging (6, 11) – because, *in S2, exchanging #7 is always better than exchanging #8, and exchanging #6 is even better*, whoever the lower ranked players in the exchange may be.

After the exchange, the subgroups S1 and S2 must be put in order in the usual way *[A.2]* (which we only seldom need to do, because they usually are already in the right order).

---

[83] *In general, this sum is S = (P2a-P1a) + (P2b-P1b) + (P2c-P1c) + (P2d-P1d) + … or, readjusting the terms, S = (P2a+P2b+P2c+P2d+…) – (P1a+P1b+P1c+P1d+…). Actually, the difference of the sums would be simpler to understand as a rule. However, the Rules choose to use the difference of the sums because, from a practical point of view, computing it is a bit simpler than computing the sum of differences.*

[84] *To compare the two exchanges and find the first of the two, we begin with the lowest players (highest BSNs), comparing the corresponding BSNs; if they are different, we choose the exchange with the higher BSN. If, on the contrary, they are equal, we proceed to the next lowest players, and repeat the comparison until we find the first difference or we use up all the players to exchange.*

Well, now we may proceed to the first possible exchange, which is between players #4 and #6 and yields the following candidate:

S1	S2
3w	4b↓
6b	11w↓

This candidate is perfect, so we can now discard the previously stored "provisional-best" and form the pairs 3-4, 11-6. Now, we perform the completion test – which will of course be successful. The allocation of colour is straightforward: 3-4, 11-6.

Now we can move on to the one point bracket: [1w, 7w, 10b, 14Ww↓] (***MaxPairs=2, x=1***). Here, players 7 and 14 already played with each other. One of the players has a strong colour preference and a downfloat.

The first pairing candidate is:

S1	S2
1w	10b
7w	14Ww↓

and of course it is not acceptable *[C.1]*. Let's then proceed to the first (and, once again, only one) transposition:

S1	S2
1w	14Ww↓
7w	10b

Since ***x*** = *1*, this is a perfect pairing. The completion test is passed and the pairs are 14-1 (*[E.2]*: the colour preference of player 14 is stronger than that of player 1) and 7-10 (*[E.1]*).

No players have a half point score; the next bracket to be paired is the lowest one, with zero points. It is comprised of [8b↑, 9w↑, 12Ww↓, 13w] (***MaxPairs=2, x=1***). Player #12, who was absent in the previous round and therefore lost by forfeit, has now a strong colour preference and a downfloat[85] *[A.4.b]*. Then we have the following candidate pairing:

S1	S2
8b↑	12Ww↓
9w↑	13w

Strangely enough, we were lucky at the first shot... Since the candidate is perfect, let us thank our good fate and accept the proposed pairs. As to the colours, the first pair is 12-8, in agreement with both preferences *[E.1]*, while for the second one, in which players have not

[85] *We register the presence of float markers as a matter of completeness – but please note that previous upfloats are irrelevant in any ordinary homogeneous bracket, while downfloats are irrelevant if the bracket does not produce downfloaters.*

only identical preferences but also the same colours histories, we satisfy the colour preference of the higher ranked player *[E.4]*, thus obtaining 9-13.

We are done! After checking everything as usual, and particularly the order of chessboards, we may publish the pairing and let the round begin.

1	**2 (2.0) - 5 (2.0)**	**½-½**
2	**3 (1.5) - 4 (1.5)**	**½-½**
3	**11 (1.5) - 6 (1.5)**	**0F-1F**
4	**14 (1.0) - 1 (1.0)**	**0-1**
5	**7 (1.0) - 10 (1.0)**	**1-0**
6	**12 (0.0) - 8 (0.0)**	**½-½**
7	**9 (0.0) - 13 (0.0)**	**1-0**

Twist! Player #11 does not show in time to play, thus forfeiting the game: *we need to fix the pairing cards (if used) and/or the tournament board* to reflect this mishap, especially in the light of the fact that the pairing between #6 and #11, not having actually been put in practice, may be repeated in a future round. Moreover, both players get a downfloat.

6 FOURTH ROUND (CHOOSING THE BEST POSSIBLE CANDIDATE)

After the third round, our tournament board is as follows. For our convenience, from now on, we will report the colour preferences and the possible last two floats for each player. The hyphen ("-") indicates that the player did not float in the last round, but it did in the previous round. By the way, at this point, a piece of advice is in order: as we proceed in the tournament, we collect more and more data, and overlooking something becomes easier and easier... We should <u>*always pay extreme attention*</u> while posting data on the board, and inspect everything two, three or even more times: as strange as it may seem, making a mistake is really easy!

Player	**PN**	**1**		**2**		**3**		**4**		**5**		**6**	
		Pair	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**
Alice	1	8W+	1.0	5B-	1.0	14B+	2.0	**W**					
Bruno	2	9B+	1.0	7W+	2.0	5W=	2.5	**B**					
Carla	3	10W+	1.0	6B=	1.5	4W=	2.0	Bb					
David	4	11B=	0.5	9W↓+	1.5	3B=	2.0	Ww-↓					
Eloise	5	12W+	1.0	1W+	2.0	2B=	2.5	Bb					
Finn	6	13B+	1.0	3W=	1.5	+000↓	2.5	b↓					
Giorgia	7	14W+	1.0	2B-	1.0	10W+	2.0	Bb					
Kevin	8	1B-	0.0	11W↑-	0.0	12B=	0.5	Ww-↑					
Louise	9	2W-	0.0	4B↑-	0.0	13W+	1.0	Bb-↑					
Mark	10	3B-	0.0	13W+	1.0	7B-	1.0	Ww					
Nancy	11	4W=	0.5	8B↓+	1.5	-000↓	1.5	w↓↓					
Oskar	12	5B-	0.0	ZPB↓	0.0	8W=	0.5	b-↓					
Patricia	13	6W-	0.0	10B-	0.0	9B-	0.0	**W**					
Robert	14	7B-	0.0	PAB↓	1.0	1W-	1.0	b-↓					

The scoregroups now are:

2.5	{2B, 5Bb, 6b↓}
2.0	{1W, 3Bb, 4Ww-↓, 7Bb}
1.5	{11w↓↓}
1.0	{9Bb-↑, 10Ww, 14b-↓}
0.5	{8Ww-↑, 12b-↓}
0.0	{13W}

As always, before starting the pairing we must check the Requirement Zero, viz. we must verify that at least one pairing exists for all the players. For example, we may have the pairing (2-6, 5-3, 1-4, 7-11, 9-10, 14-8, 12-13). As usual, this pairing is almost certainly <u>not</u> the one we are looking for – but the pairing is possible, and therefore we may proceed.

The first bracket is: [2B, 5Bb, 6b↓] (***MaxPairs=1, x=1***). Here, players #2 and #5 already played with each other *[C.1]* and the first candidate pairing is therefore not legal. We should go to the first transposition *[C.7]*, which yields the pair 6-2 – while player #5 shall float to the next bracket (with 2 points):

[5Bb][1W, 3Bb, 4Ww-↓, 7Bb] (***MaxPairs=2, M0=M1=1, x = 0***), which gives:

S1	S2
5Bb	1W
	3Bb
	4Ww-↓
	7Bb

We already had the pair 5-1 in the second round. Hence, the first MDP-Pairing *cannot give origin to any legal candidate* – and, therefore, we may discard it at once.

The first useful transposition yields for the MDP-Pairing the pair 5-3, which is legal but does not comply with C.10, because of colliding colour preferences (since the failure value for C.10, relative to the <u>complete</u> pairing of the bracket, cannot be less than 1, no pairing originating from this MDP-Pairing will be perfect).

However, the evaluation should be made on the complete bracket pairing, so we proceed to pair the remainder, which is now [1W, 4Ww-↓, 7Bb]. The first candidate is thus (5-3, 1-4, 7 to float), which presents us with a failure value for criterion C.10 equal to 2 – we will write it as C.10(2) for short. Since the candidate is legal, we store it as provisional-best and proceed with a transposition in the subgroup S2R of the remainder *[B.7]*, obtaining the new candidate (5-3, 1-7, 4 to float), which has failure values C.10(1), and C.14(1). This candidate is not perfect, but has a lower failure value for C.10 than the provisional-best. Hence, it becomes the new provisional-best (the previous one is discarded).

However, we shall proceed to examine more candidates, because we did not find a perfect one yet. Since we used up all transpositions in S2R, we now try an exchange between S1R and S2R: we swap the last (and only) element of S1R with the first element of S2R. With this exchange, the remainder now becomes [4Ww-↓, 1W, 7Bb]. The first candidate is again (5-3, 1-4, 7 to float), which we already examined[86] (and discarded). We therefore apply the only possible transposition in S2R, obtaining the new candidate (5-3, 4-7, 1 to float). This candidate had failure value C.10(1) and is therefore the new provisional-best.

The next, and last, remainder exchange gives [7Bb, 1W, 4Ww-↓], but all the candidates originating from this remainder have already been evaluated.

Now we have completed the evaluation of all the possible candidates given by the current MDP-Pairing. Since none among them is perfect, we must proceed with another transposition in the original S2, to look for a perfect candidate. The next useful transposition yields for the MDP-Pairing the pair 4-5, which is legal, and the remainder [1W, 3Bb, 7Bb]. This in turn provides us with (4-5, 1-3, 7 to float) that is, at last, a perfect candidate.

Before choosing this candidate, however, we must check that its floater maximises the number of pairs and minimises the PSD in the next bracket –and just in that one: be they as they may, we do not proceed to inspect the following brackets! The next bracket would be [7Bb] [11 w↓↓]: the two players are not incompatibles, so that we have the maximum number of pairs, and no player from the bracket just paired would lend us a better PSD.

The last check to perform is the completion test. The rest is now {7Bb, 11w, 9Bb, 10Ww, 14b, 8Ww, 12b, 13W}, and (7-11, 9-10, 14-8, 13-12) is a legal (although awful) pairing, so we may proceed.

Before going to the next bracket, however, we may want to do some thinking about the procedure we just used. Actually, we may reason that, although we cannot be sure, we may well suspect that a MDP-Pairing with unsatisfactory failure values may (likely) bring us nowhere. Sometimes, a brief inspection of the whole bracket might show that a perfect pairing exists (just as it was here) and that we do not really need to waste precious time in analysing imperfect candidates. This is of course sound reasoning, but skipping steps and jumping ahead is always a potential risk. For example, we may easily verify that in the bracket [5Bb][1B, 3Bb, 4Ww-↓, 7Bb] ($x=1$), which is rather similar to the previous one, the correct pairing would have been (3-5, 4-1, 7↓): *a different colour preference in a player who*

[86] *See note 19, page 19.We also want to note that every time a player from the original S1 goes to S2, it can only be paired to another player of the original S1 or float, because any other pairing would give origin to a candidate that has already been evaluated (see also note 82, page 56).*

is incompatible with the MDP, changed the latter's pairing! The lesson we must learn from this, is that we must always be very careful, and beware of all shortcuts.

The next bracket, which is heterogeneous, is:

[7Bb] [11 w↓↓] (***MaxPairs=1, M1=M0=1, x=0***).

Since we have only one possible candidate, and players #7 and #11 did not play with each other, we can make the pairing at once: 11-7. The rest is now {9Bb, 10Ww, 14b, 8Ww, 12b, 13W}, and (9-10, 14-8, 13-12) is a legal pairing – therefore, the completion test is passed.

The next bracket is: [9Bb-↑, 10Ww, 14b-↓] (***MaxPairs=1, x=0***), which gives us:

S1	S2
9Bb-↑	10Ww
	14b-↓

Here, all players are compatible and therefore can play with each other, but we have a small problem: the "natural" pairing would leave #14 unpaired - but this player had a downfloat in the second round and therefore should not get one more now *[C.14]*.

The pairing is legal, but not perfect: we thus store it as provisional-best and then proceed to look for a possibly perfect, or simply better, candidate. First, we try a transposition:

S1	S2
9Bb-↑	14b-↓
	10Ww

Here the problem is that the players' colour preferences are not matched well enough *[C.10]*. Let us compare this candidate with the provisional-best: the latter fails (once) for C.14, while this candidate fails (once) for C.10, which is worse – hence, we keep the current provisional-best.

Thus, even with a transposition, we cannot come to a valid conclusion, and we have to try one homogeneous exchange:

S1	S2
10Ww	9Bb-↑
	14b-↓

We already tried the pair 10-9. Thus, once again we go on to a transposition, which yields:

S1	S2
10Ww	14b-↓
	9Bb-↑

At last, we get the perfect pairing with 10-14, while player #9 floats to the next bracket.

To get this pairing we followed a formal procedure, which is correct but is not always the simplest possible one. Actually, there is an alternative method of pairing which, especially in the case of a bracket with few players, may be more convenient. We call it *Sieve pairing*[87], and it is very simple: first, we generate, in due order, all the possible candidates. Then, starting from criterion C1 and proceeding one by one through all the pairing criteria, we check the failure values of the candidates, and keep only those candidates *whose failure value is not worse than the best failure value for that criterion*, discarding all the others. In this way, we gradually reduce the number of possibly acceptable candidates, until we are left with only one (which is immediately selected), or with a small group of them, in which case we select the first one (as they were generated in due order).

For the last bracket, which was very simple, the possible candidates were just four:

Candidate	Failure values	
float 9Bb-↑, 10Ww, 14b-↓	C.5(1)	dropped at C.5[88]
(9Bb-↑, 14b-↓), float 10Ww	C.10(1)	dropped at C.10
(9Bb-↑, 10Ww) , float 14b-↓	C.14(1)	dropped at C.14
(10Ww, 14b-↓), float 9Bb-↑	none	perfect – selected after C.14

In this very easy example, there is a perfect candidate, which is of course the one we must select. However, even when there is no perfect candidate, this method is very convenient because it always allows us to choose the best one easily and safely[89].

We must now check that the selected floater, #9, optimises the pairing in the next bracket, which would be [9Bb-↑][8Ww-↑ 12b-↓]. Since 9-12 is a legal pairing, and a different selection of floater(s) would not give a better PSD, the floater is acceptable.

Before proceeding, we must perform the usual completion test, to see that the rest of the players may be paired. The rest is {9Bb, 8Ww, 12b, 13W}, and (9-8, 13-12) is a legal pairing, so we may proceed.

The next bracket is the half point one: [9Bb-↑][8Ww-↑ 12b-↓] (***MaxPairs=1, M1=M0=1, x=0***), where #8 and #12 are incompatible because of *[B.1.a]*.

S1	S2
9Bb-↑	8w-↑
	12b-↓

87 See "The Sieve Pairing" in [B.8], page 21.

88 Although surely not a very good one, a candidate with no pairs and all players floating down is however legal and must be considered among the possible pairings (and, sometimes, it may be the only possible one).

89 Of course, the Sieve method may be used always, whatever is the number of players to pair. However, when pairing manually, this method is especially useful for "small but complicated" brackets, while for large brackets it may become rather tiring.

Once again, the candidate is not perfect because #8 got an upfloat and #12 got a downfloat during the second round, so we have a failure for [C.14] and one for [C.15]. The only possible transposition cannot help us because ***x*** is zero and both 9, 12 have a colour preference for black, so we have a failure for [C.10].

On the other hand, an exchange would make #9 float down, giving a failure for [C.6]. Since this is the first time we find this case, we may want to expand a little about it, possibly giving an example on how to compute and compare PSDs. To do this, we need to remember the score of each player: #9 (1.0), #8 (0.5), #12 (0.5). Since a player shall float down from this bracket, we need to compute the artificial value used in [A.8] to calculate the score difference of that player. This value is one point less than the minimum score in the heterogeneous bracket, hence AV=-0.5. The fact that this value is negative is no concern of ours, because this value will be subtracted into a positive number, giving a positive score difference. The PSD of a candidate is simply the list of all score differences in the candidate, sorted in descending order. We have three possible candidates and the respective PSDs are:

- 9-8 [SD= score(9)-score(8)=0.5], 12-float [SD= score(12)-AV=1.0]: PSD={1.0, 0.5}
- 9-12 [SD= score(9)-score(12)=0.5], 8-float [SD= score(8)-AV=1.0]: PSD={1.0, 0.5}
- 8-12 [SD= score(8)-score(12)=0.0], 9-float [SD= score(9)-AV=1.5]: PSD={1.5, 0.0}

It is at once apparent that the PSD of the last candidate, which is the one in which the MDP is floating down again, is worse than the others are, because its first element is greater *[A.8]*.

In summary, we have three possible candidates, and neither of them is perfect. We must therefore choose the candidate whose quality is the best possible among them, and that is (9-8, downfloat to 12), because it only infringes C.14 and C.15, which is weaker than the other involved pairing criteria.

Now we must verify that the given downfloater maximises the pairing in the next bracket, which is the last one and contains only [12b-↓][13W]. The two players are compatible and there is no PAB to be assigned – therefore the floater is immediately acceptable. The final check is the completion test: now the rest is comprised only of the last bracket, which can be paired – hence, the test is passed.

In the last bracket, the only possible candidate is perfect and gives the pair 13-12. As usual, we check everything, rearrange (if necessary) the chessboards order, start the round - and reach the fifth round.

1	**6 (2.5) - 2 (2.5)**	**0-1**
2	**4 (2.0) - 5 (2.5)**	**½-½**
3	**1 (2.0) - 3 (2.0)**	**1-0**
4	**11 (1.5) - 7 (2.0)**	**1-0**
5	**10 (1.0) - 14 (1.0)**	**½-½**
6	**8 (0.5) - 9 (1.0)**	**½-½**
7	**13 (0.0) - 12 (0.5)**	**1-0**

7 FIFTH ROUND (PPB AND CLB)

After the fourth round is played out, the tournament board is as follows:

Player	**PN**	**1**		**2**		**3**		**4**		**5**		**6**	
		Pair	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**
Alice	1	8W+	1.0	5B-	1.0	14B+	2.0	3W+	3.0	b			
Bruno	2	9B+	1.0	7W+	2.0	5W=	2.5	6B+	3.5	w			
Carla	3	10W+	1.0	6B=	1.5	4W=	2.0	1B-	2.0	w			
David	4	11B=	0.5	9W↓+	1.5	3B=	2.0	5W↑=	2.5	b↑			
Eloise	5	12W+	1.0	1W+	2.0	2B=	2.5	4B↓=	3.0	W↓			
Finn	6	13B+	1.0	3W=	1.5	+000↓	2.5	2W-	2.5	B-↓			
Giorgia	7	14W+	1.0	2B-	1.0	10W+	2.0	11B↓-	2.0	w↓			
Kevin	8	1B-	0.0	11W↑-	0.0	12B=	0.5	9W↑=	1.0	b↑			
Louise	9	2W-	0.0	4B↑-	0.0	13W+	1.0	8B↓=	1.5	w↓			
Mark	10	3B-	0.0	13W+	1.0	7B-	1.0	14W=	1.5	b			
Nancy	11	4W=	0.5	8B↓+	1.5	-000↓	1.5	7W↑+	2.5	Bb↓↑			
Oskar	12	5B-	0.0	ZPB	0.0	8W=	0.5	13B↓-	0.5	Ww↓			
Patricia	13	6W-	0.0	10B-	0.0	9B-	0.0	12W↑+	1.0	b↑			
Robert	14	7B-	0.0	PAB↓	1.0	1W-	1.0	10B=	1.5	Ww			

After having checked that at least one pairing is legal (e.g. 1-2, 3-5, 4-6, 7-8, 9-10, 11-12, 13-14), we separate the scoregroups as usual:

3.5 {2w}
2.0 {1b, 5W↓}
2.5 {4 b↑, 6B-↓, 11Bb↓↑}
2.0 {3w, 7w↓}
1.5 {9w↓, 10b, 14Ww}
1.0 {8b↑, 13b↑}
0.5 {12Ww↓}

The first bracket, with 3.5 points, is [2w] (***MaxPairs=0***) - but, with a lonely player, there is not very much to do... it can't help but float down to the next bracket, which is the one with 3 points: [2w][1b, 5W↓] (***MaxPairs=1, M0=M1=0, x=0***). Here, 1-5 and 2-5 already met each other. Thus, the only remaining candidate is 2-1, which is legal and is hence accepted. The completion test is successful and we may proceed to the next bracket.

The next bracket is heterogeneous: [5W↓] [4b↑, 6B-↓, 11Bb↑] (***MaxPairs=2, M0=M1=1, x=1***, ***z=0***). The games 4-5 and 11-4 have already been played. Therefore, out of necessity, whichever the float status of players is, we will have to use the only legal pair 4-6, and therefore 5-11, even if this pairing is not perfect (it fails because of C.13). In this case, the Sieve pairing method helps us not to waste any time – just excluding any other candidate simply because it would be an illegal one.

Before going ahead to the next bracket, we want to perform the completion test. The rest is now {3w, 7w↓, 9w↓, 10b, 14Ww, 8b↑, 13b↑, 12Ww↓}, which allows the legal pairing (3-7, 9-10, 8-13, 12-14).

In the next bracket, with 2 points, we have players [3w, 7w↓] (***MaxPairs=1***), who did not play with each other in previous rounds - therefore they can be paired[90]. The players have identical preferences and colours histories – therefore, we satisfy the colour preference of the highest ranked player, thus obtaining the pair 3-7 – which satisfies the completion test.

With 1.5 points, we have [9w↓, 10b, 14Ww]. The first candidate yields 9-10 and player #14 floats to the next bracket, which is [14Ww][8b↑, 13b↑]. Here, all players are compatible, and therefore the selected downfloater complies with C.7. The rest is {14Ww, 8b↑, 13b↑, 12Ww↓}, which allows the pairing (8-13, 12-14), hence the completion test is passed too.

Let us now go to the next bracket: [14Ww][8b↑, 13b↑] (***MaxPairs=1, M0=M1=1, x=0***). The first candidate here is 14-8, which is legal but violates C.7 (because player #13, as a downfloater, does not maximise the pairing in the following bracket, where it is incompatible with the only resident, player #12) and C.13 (because player #8 already got an upfloat in the previous round). We store it as provisional-best, and proceed to examine the next candidate, which is 14-13. Once again, the candidate is legal but shows C.7 and C.13 failures: since it is not better than the current provisional-best, this candidate is discarded. With an exchange, we have 8-13, which violates both C.6 and C.10 and therefore is worse than the current provisional-best – so we discard it too. Since there are no more possible candidates, we must elect the current provisional-best. Let us then consider the first candidate, 14-8, and check if the player #13, as a downfloater, satisfies the Requirement Zero. The next scoregroup, as mentioned, contains only the player #12, who has already met #13 in the tournament. Hence, the latter, as a downfloater, not only violates C.7, but also yields a failure on the completion test: *we are stuck with a failure in Requirement zero*.

[90] *When, as in this bracket, all the colour preferences are for the same colour,* ***x*** *is useless and we may simply omit its calculation. The same applies when only one player in a bracket expects a different colour than all the other players, and the bracket does not produce floaters (because, in that case, the player's colour preference will be unavoidably complied with). In such cases, we may ignore the corresponding criterion C.10. Please note that, even if* ***x*** *is useless, the same is not necessarily true for* ***z.*** *If, for example, all the preferences are for black, and half of them are strong and the other half are mild, there is no way to change the number of total disregarded preferences (hence* ***x*** *is useless). However, there is a way to minimise the number of disregarded strong preferences and, therefore, we want to use* ***z*** *as a guideline for this optimisation.*

The bracket we are pairing, which is [14Ww][8b↑, 13b↑], now becomes the PPB, while the rest of the players – which, in this case, is simply {12Ww↓} – becomes the SCS *[A.3, A.9]*.

For the pairing of the CLB, we no longer have to comply with the optimisation of downfloaters *[C.7]*, but with a different criterion: the downfloater(s) generated by this bracket must pair the rest of the players. Since this criterion is of a higher level than the optimisation of the PSD in the bracket *[C.6]*, this allows us to let *any* player float. It is very easy to verify that the one and only downfloater, which complies with requirement C.4, is #14 (which, in this bracket, is simply the only player compatible with #12).

The pairing of the PPB is thus 13-8, leaving #14 to downfloat as required. Of course, there is no need to perform a completion test, because we already selected the floater in such a way as to make the complete pairing possible. The CLB is now built putting together the MDP from the PPB and the rest of the players: [14Ww][12Ww↓], and is immediately paired.

The last thing to do is colours allocation. Both players in the CLB have identical (strong) colour preferences. Let's look at the colours histories of the players: 14:B-WB; 12:B-WB, which are yet again identical. We cannot help but satisfy the colour preference of the higher ranked player *[E.4]*, which is of course #14 who has a higher score - thus, we obtain 14-12. Let's see what shall be of players #8 and #13: both have mild colour preferences, but now the colours histories are different: 8:BWBW; 13:WBBW - thus, we should alternate colours with respect to the last round in which they played with different colours *[E.3]*, obtaining 13-8. As usual, we double-check everything, then we start the round.

1	**2 (3.5) - 1 (3.0)**	**½-½**
2	**5 (3.0) - 11 (2.5)**	**1-0**
3	**4 (2.5) - 6 (2.5)**	**½-½**
4	**3 (2.0) - 7 (2.0)**	**0-1**
5	**9 (1.5) - 10 (1.5)**	**0-1**
6	**14 (1.5) - 12 (0.5)**	**1-0**
7	**13 (1.0) - 8 (1.0)**	**0-1**

8 SIXTH ROUND (PPB AND CLB AGAIN)

After the fifth round is played out, the tournament board is as follows:

Player	PN	1		2		3		4		5		6	
		Pair	Pnts	Pair	Pnts	Pair	Pnts	Pair	Pnts	Pair	Pnts	Pair	Pnts
Alice	1	8W+	1.0	5B-	1.0	14B+	2.0	3W+	3.0	2B↓=	3.5	Ww↓	
Bruno	2	9B+	1.0	7W+	2.0	5W=	2.5	6B+	3.5	1W↑=	4.0	Bb↑	
Carla	3	10W+	1.0	6B=	1.5	4W=	2.0	1B-	2.0	7W-	2.0	Bb	
David	4	11B=	0.5	9W↓+	1.5	3B=	2.0	5W↑=	2.5	6W=	3.0	B-↑	
Eloise	5	12W+	1.0	1W+	2.0	2B=	2.5	4B↓=	3.0	11W↓+	4.0	Bb↓↓	
Finn	6	13B+	1.0	3W=	1.5	+000↓	2.5	2W-	2.5	4B=	3.0	w	
Giorgia	7	14W+	1.0	2B-	1.0	10W+	2.0	11B↓-	2.0	3B+	3.0	W-↓	
Kevin	8	1B-	0.0	11W↑-	0.0	12B=	0.5	9W↑=	1.0	13B+	2.0	Ww-↑	
Louise	9	2W-	0.0	4B↑-	0.0	13W+	1.0	8B↓=	1.5	10W-	1.5	Bb-↓	
Mark	10	3B-	0.0	13W+	1.0	7B-	1.0	14W=	1.5	9B+	2.5	Ww	
Nancy	11	4W=	0.5	8B↓+	1.5	-000↓	1.5	7W↑+	2.5	5B-	2.5	w-↑	
Oskar	12	5B-	0.0	ZPB	0.0	8W=	0.5	13B↓-	0.5	14B↑-	0.5	W↑↓	
Patricia	13	6W-	0.0	10B-	0.0	9B-	0.0	12W↑+	1.0	8W-	1.0	B-↑	
Robert	14	7B-	0.0	PAB↓	1.0	1W-	1.0	10B=	1.5	12W↓+	2.5	b↓	

As usual, we check that at least one pairing is legal (e.g. 1-4, 2-3, 5-6, 7-8, 9-11, 10-12, 13-14), then we separate the scoregroups:

4.0	{2Bb↑, 5Bb↓↓}
3.5	{1Ww↓}
3.0	{4B-↑, 6w, 7W-↓}
2.5	{10Ww, 11w-↑, 14 b↓}
2.0	{3Bb, 8Ww-↑}
1.5	{9Bb-↓}
1.0	{13B-↑}
0.5	{12W↑↓}

First, we want to remember that this is the final round – hence, it is possible that some topscorers *[A.7]*, as well as their opponents, may have their absolute colour preference disregarded *[A.6]*.

Now, we must observe that players #1, #2 and #5 have all played with each other, so there is no meaning in trying to pair them – they must all float into the third scoregroup, forming the heterogeneous bracket:

[2Bb↑, 5Bb↓↓, 1Ww↓][4B-↑, 6w, 7W-↓] (***MaxPairs****=3,* ***M0****=*__*M1*__*=3,* ***x****=0, z=0*)

Here, the only possible opponent for #2 is #4; since, as we know, player #5 did already play with #1, *there are only two possible candidates*. The first generated one is:

S1	S2
2Bb↑	4B-↑
5Bb↓↓	6w
1Ww↓	7W-↓

This candidate is apparently far from perfect. Actually, it fails for colour matching (both for all preferences *[C.10]* and for strong preferences *[C.11]*) on two pairs.

It also fails on upfloaters protection *[C.19]* on pair 2-4 – the latter because the score difference between players #2 and #4 is not minimum (player #2 is a double-floater in this bracket, so #4 is too). Finally, we might also register the fact that #4 is now getting an upfloat for the second time running, thus failing on [C.15]. However, since the pairing 2-4 is the only possible one for player #2, all possible candidates will fail on these four particular criteria ([C.10], [C.11], [C.15], [C.19]), which therefore cannot help us find the best candidate.

Anyway, since the candidate is legal, we store it as provisional-best, and proceed to build and evaluate the next (and only other) possible one, which is:

S1	S2
2Bb↑	4B-↑
5Bb↓↓	7W-↓
1Ww↓	6w

The evaluation of this candidate shows that it fails for [C.10] on the first and third pair, just as the previous one above, but now we have only one failure for [C.11]. Therefore, we discard the previous candidate and keep the current one, which, although not perfect, is better. Since this is also the last possible legal candidate, we choose it and thus obtain the pairing (2-4, 7-5, 1-6).

We must now perform the completion test; the rest is now {10Ww, 11w, 14b, 3Bb, 8Ww, 9Bb, 13B, 12W}, which allows at least one possible pairing (e.g., 10-11, 13-3, 9-12, 8-14), so the Requirement Zero is satisfied.

The next bracket is {10Ww, 11w-↑, 14b↓}, and #10 already played with #14.

The first candidate (10-11, 14 to float) is legal but fails for C.10 and for C.12 – as usual, we store it as provisional-best. The second candidate is not even legal (because of 10-14) and we discard it straight away. An exchange lends us the next possible candidate, which is (11-14, 10 to float), which is perfect and is therefore immediately accepted.

We must now check that player #10, as a downfloater, maximises the pairing in the next bracket *[C.7]*, which is readily made, because in the bracket [10][3, 8] we can actually build one pair. The check for the Requirement Zero is successful (e.g. 3-13, 8-10, 9-12) and we may proceed to the next bracket, which is [10Ww][3Bb, 8Ww-↑]. The game 10-3 already took place – therefore 10-8 is the only possible MDP-pairing, with an empty pairing of the remainder and with player #3 getting a downfloat. We check that this floater maximises the next bracket (it does) and that the rest is pairable (it is).

We thus get to the next bracket, which is [3Bb][9Bb-↓]. Once again, we have only one possible pairing, which is 9-3. We must check that the rest is pairable – and, big surprise, we find that it is not (in the rest, there are now only two players, #12 and #13, and they already played each other), so we have a Requirement Zero failure. The current bracket (that is, the one we just paired) is now the PPB, and must give the floaters needed – we have no choice but to make players #3 and #9 float down. Hence, the pairing in the PPB produces no pairs at all, and we obtain the CLB: [3Bb, 9Bb-↓][13B-↑, 12W↑↓]

Now, we can be sure that at least one pairing exists for this bracket, because of the success of the completion test on the previous bracket. Actually, player #13 already played with #9 and #12. Hence, the pairing 3-13 is unavoidable, and this only leaves room for 12-9 as the other pair. We now make the usual checks on the pairings and their order, then... Ladies and gentlemen, please start clocks for the final round!

1	**2 (4.0) - 4 (3.0)**	**½-½**
2	**7 (3.00) - 5 (4.0)**	**½-½**
3	**1 (3.5) - 6 (3.0)**	**1-0**
4	**11 (2.5) - 14 (2.5)**	**0-1**
5	**10 (2.5) - 8 (2.0)**	**0-1**
6	**3 (2.0) - 13 (1.0)**	**1-0**
7	**12 (0.5) - 9 (1.5)**	**½-½**

9 FINAL STEPS

Now the tournament is over. The final operations, with regard to pairing, consist of the harvesting of results and final compilation of the tournament board.

Player	**PN**	**1**		**2**		**3**		**4**		**5**		**6**	
		Pair	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**	**Pair**	**Pnts**
Alice	1	8W+	1.0	5B-	1.0	14B+	2.0	3W+	3.0	2B↓=	3.5	6W+	4.5
Bruno	2	9B+	1.0	7W+	2.0	5W=	2.5	6B+	3.5	1W↑=	4.0	4W=	4.5
Carla	3	10W+	1.0	6B=	1.5	4W=	2.0	1B-	2.0	7W-	2.0	13W+	3.0
David	4	11B=	0.5	9W↓+	1.5	3B=	2.0	5W↑=	2.5	6W=	3.0	2B=	3.5
Eloise	5	12W+	1.0	1W+	2.0	2B=	2.5	4B↓=	3.0	11W↓+	4.0	7B=	4.5
Finn	6	13B+	1.0	3W=	1.5	+000↓	2.5	2W-	2.5	4B=	3.0	1B-	3.0
Giorgia	7	14W+	1.0	2B-	1.0	10W+	2.0	11B↓-	2.0	3B+	3.0	5W=	3.5
Kevin	8	1B-	0.0	11W↑-	0.0	12B=	0.5	9W↑=	1.0	13B+	2.0	10B+	3.0
Louise	9	2W-	0.0	4B↑-	0.0	13W+	1.0	8B↓=	1.5	10W-	1.5	12B=	2.0
Mark	10	3B-	0.0	13W+	1.0	7B-	1.0	14W=	1.5	9B+	2.5	8W-	2.5
Nancy	11	4W=	0.5	8B↓+	1.5	-000↓	1.5	7W↑+	2.5	5B-	2.5	14W-	2.5
Oskar	12	5B-	0.0	ZPB	0.0	8W=	0.5	13B↓-	0.5	14B↑-	0.5	9W=	1.0
Patricia	13	6W-	0.0	10B-	0.0	9B-	0.0	12W↑+	1.0	8W-	1.0	3B-	1.0
Robert	14	7B-	0.0	PAB↓	1.0	1W-	1.0	10B=	1.5	12W↓+	2.5	11B+	3.5

That's all!