MARIO HELD, IA

ANNOTATED PAIRING RULES

FOR THE

FIDE (DUTCH) SYSTEM

(2026 EDITION)

Rev. 2512151400

Contents

FOREWORD

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

(1) Introductory Remarks and Definitions: containing the basic concepts about the system and its control variables. The last paragraph (Round Pairing Outlook, Article 1.9) is an essential description of the pairing process.

(2) Pairing Criteria: defining limitations to the possible pairings of the players. Some of those limitations are common to all FIDE Swiss pairing systems, while others are specific to the FIDE (Dutch) system and give origin to some of its peculiarities. Those criteria fall into two distinct categories. The first consists of criteria that must be strictly adhered to (absolute criteria). The second relates to criteria that should be met as far as possible (quality criteria), but which can, to some extent, be relaxed. We will define a pairing as legal only if it fully complies with all the criteria of the first type. A pairing that is not legal can only be immediately discarded.

(3) Pairing process for a bracket: explaining a viable method 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). This section, contrary to others, does not give straightforward rules, but rather suggests a way to carry out the pairing, illustrating rather than defining the rules, which are in fact given by the pairing criteria in section 2 and the generation rules in section 4. In fact, the pairing can be made even with completely different methods, provided it yields identical results to those given by the algorithm described here.

(4) Rules for the sequential generation of the pairings: this section defines the transposition and exchange procedures, showing how to "shuffle" the players list when natural pairing is impossible (because two players have already played against each other, or because of colours incompatibility, and so on) or simply inadequate (e.g., because of clashing colour preferences).

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

With respect to the previous edition of the FIDE (Dutch) rules, in addition to some clarifications and window dressing, there are two major changes, which can cause pairings to be different than those made with the old rules. First, a pairing is now made in such a way that, by construction, the rest of the players can always be paired. Contrary to the past, now we never need to resort to a different pairing route. This allows us to get rid of those Special Collapsed Scoregroup (SCS), Collapsed Last Bracket (CLB) and Penultimate Pairing Bracket (PPB) that where a serious difficulty of the previous edition. This new rule can modify the resulting pairings but does not overturn them because, even when different opponents are chosen, they are still tantamount to the opponents chosen by the previous version of the rules. The second change is a different management of the pairing-allocated bye, which is now assigned in such a way that the assignee will always have the lowest possible score and, in most cases, as many played games as possible *(although in some instances this criterion cannot apply). This is a critical change, which can significantly modify the resulting pairing (hopefully, for the better!).*

The new version of the FIDE (Dutch) rules, as most FIDE approved Swiss pairing systems, replaced the traditional "pairing number" with the new "Tournament Pairing Number" ("TPN"). This is essentially equivalent to its predecessor in the workings of the pairing system but marks a discontinuity in the way we think of it. We used to think of the pairing number as an identifier intrinsically linked to the rankings. Thus, we called a "low pairing number" that of a participant sitting near the bottom of the rankings, even though that number was high rather than low. Now, we consider the TPN as a pure number, so a "small" (or low) TPN is simply <u>*a number which is low*</u> *and, therefore, is associated with a participant sitting in the high echelons of the ranking. On the contrary, a "large" (or high) TPN is a number that is high and, therefore, identifies a participant sitting in the bottom rankings.*

C.04.1
BASIC RULES FOR SWISS SYSTEMS

Approved by the Council on 28/10/2025
Applied from 1st February, 2026

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

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

> *As a rule, 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, for example when some external event disrupts the tournament, or when it is impossible to complete the pairing for a new round. See also Article 1.9.3 of the FIDE Swiss (Dutch) System (or Article 1.8.3 of Dubov system rules, or 1.8.3 of Burstein rules, or 1.9.3 of Team Pairing System, which say all the same thing) and the relevant comment.*

2. Two participants shall not play against each other more than once.

> *This is the only principle of Swiss Systems that a FIDE-approved system cannot dispense with - unless doing differently is really inevitable! (See, e.g., Article C.04.3:1.9.3).*

3. Should the number of participants to be paired be odd, one player is not paired. This participant 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.

   This number of points shall be the same for all pairing allocated byes.

> *This rule, besides stating that a Pairing-Allocated Bye ("PAB") has no opponent and no colour, allows event organisers to establish a different score value for the PAB instead of the usual whole point. For example, the tournament rules may specify that the PAB be worth half a point instead (or even no points at all). This helps to reduce the playing strength imbalance in subsequent round(s) for weak players. In fact, such players, after receiving a full-point PAB in the first round, will meet a (too) strong opponent in the next round, and possibly even in the following one, damaging both the player's morale and the opponent's Buchholz (or other tiebreak).*
>
> *Please note that, although the tournament rules can freely choose the value for the PAB, they cannot change it during the tournament.*

4. A participant who has already received a pairing allocated bye, or has already scored in one single round, without playing, as many points as rewarded for a win, shall not receive the pairing allocated bye.

> *Whatever the value of a pairing allocated bye, it cannot be assigned to any player who either has already received a previous one or has scored the equivalent of a win without playing. This includes forfeit wins, and, starting with these new rules, also the full-point bye (a "last option" sometimes used by the arbiter in order to correct some unforeseen circumstances). The allocation of a PAB, though, is not prevented by previous "byes on request" – when such provision is permitted by the tournament rules – even if the player received more than one Half-Point Bye.*

5. In general, participants 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.04.3:1.7.1).*

6. For each participant the difference between the number of rounds they play with Black and the number of rounds they play with White shall not be greater than 2 or less than –2. Each pairing system may have exceptions to this rule.

7. No participants shall receive the same colour three times in a row. Each pairing system may have exceptions to this rule.

> *Exceptions to Articles* ***6*** *and* ***7*** *above are* <u>*possible*</u>*, but* <u>*not compulsory*</u>*. The FIDE (Dutch) system makes such exceptions in the last round (but only when there are very good reasons to do so). Other systems do not do the same – e.g., the Dubov Swiss System does not allow such exceptions, which seem to be inconsistent with the basic principles of that system, while the Team Pairing System allows them in all rounds.*

8. In general, a participant is given the colour with which they played fewer rounds. If colours are already balanced, then, in general, the participant is given the colour that alternates from the last one with which they played.

> *This rule enforces the good colour balancing that all FIDE-approved Swiss Systems must give.*

9. 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 usually prepared with the help of a software program (which should be approved by FIDE if possible). Even when pairings are computer-made, the arbiter always takes full responsibility for them – and, therefore, must be able to explain and justify any given pairing. Hence, the pairing rules should not be so complex as to make an explanation impossible, even considering that a player could be completely unaware of how pairings are made – and, to call a spade a spade, the Dutch system barely qualifies… On a more important note, we should also emphasise that this rule leaves out all those systems that, because of their mathematical or logical complexity, can only be implemented by means of computer software, so the pairings they yield cannot be explained in plain words.*

C.04.2
GENERAL HANDLING RULES FOR SWISS TOURNAMENTS

Approved by the Council on 28/10/2025
Applied from 1st February, 2026

1. PAIRING SYSTEMS

The rules in this section try to prevent any tampering with the pairings in favour of some participants (e.g., to help a player to obtain a norm). To this effect, the rules must be well specified, transparent, and unambiguous in the first place.

1.1 The pairing system used for a FIDE-rated Swiss tournament should be one of the published FIDE Swiss Systems. Accelerated methods are acceptable if they were announced in advance by the organiser and are published under FIDE-Approved Accelerated Systems.

1.2 Pairing systems or accelerated methods not published by FIDE may be permitted, provided that a detailed written description of their rules:

1.2.1 be submitted in advance to the Qualification Commission (QC) and temporarily authorised by them; and

1.2.2 be included in the tournament regulations and explicitly presented to the participants before the start of the tournament.

These rules allow the use of an experimental (or custom) system also, on condition that it is authorised by the proper authority and that its rules are made duly known.

1.3 While reporting a tournament to FIDE, the Chief Arbiter shall declare which official FIDE Swiss system and acceleration method (if any) were used, or provide the temporary authorisation(s) given by the QC as per Rule 1.2.1.

This rule makes an effective check of the pairings both possible and practical.

1.4 The Swiss Pairing Systems defined by FIDE pair the participants in an objective, impartial and reproducible manner. In any tournament where such systems are used, different arbiters, or different tournament handler programs approved by FIDE, must be able to arrive at identical pairings. However, the use of such systems is deprecated, unless a tournament handler program approved by FIDE is available for them, provided with a free pairing-checker able to verify tournaments run with that system.

A pairing system officially recognised by FIDE cannot be itself deprecated. However, a system should preferably not be used in a tournament if there is no approved software to make and check the pairings.

1.5 It is not allowed to alter the correct pairings in favour of any participant. 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 QC to initiate disciplinary measures through the Ethics and Disciplinary Commission.

> *Terrible things happen to people who tamper with pairings... see, e.g., FIDE Arbiters Disciplinary Sub-Committee decision on Case DSC-002-2023-002 (spoiler: eighteen months ban for the arbiter) and FIDE Ethics and Disciplinary Commission decision EDC 10/23 (spoiler: four years ban plus IO title revoked for the father of a player).*

2. INITIAL ORDER AND LATE ENTRIES

2.1 Before the start of the tournament a measure of the participant's strength is assigned to each participant. The strength is usually represented by rating lists of the players. If one rating list is available for all participating players, then this rating list should be used. It is advisable to check all ratings supplied by players. If no reliable rating is known for a player, the Chief Arbiter should make an estimation of it as accurately as possible.

> *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 approximately halved at every round. Thus, in a tournament with T rounds, if the 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 ranking.*
>
> *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. 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. Chief Arbiters must use their sound judgment to obtain the best possible evaluation with what data is available.*

2.2 Before the first round the participants are ranked in order of, respectively:

2.2.1 Strength (for example ratings)

> *For individual competitions, the player's rating is usually adopted as playing strength measure. As mentioned above, this rating can be either official or presumptive. The rating, when used, is that applying at the day the tournament starts. Possible subsequent rating list updates do not affect the initial order. (However, the list can be sorted again – within limits, see Article 2.3 – when errors need to be corrected, or new participants are admitted to the tournament.)*
>
> *For team competitions, there is no team rating. Still, we have the individual ratings of the team members, from which an estimate of the team strength can be derived. However, there is no universal rule to do that – the responsibility to define a formula is left with the tournament rules.*

2.2.2 FIDE-title (GM-IM-WGM-FM-WIM-CM-WFM-WCM-no title), for individual tournaments

> *FIDE titles are ordered by descending nominal ratings. When ratings are equal, titles obtained through norms precede automatic ones. (This is the case for WGM and FM titles, and for WIM and CM.)*

2.2.3 Alphabetically (unless it has been previously stated that this criterion has been replaced by another one)

Alphabetical sorting is unessential, its only rationale being that of ensuring an unambiguous order. Thus, this criterion can be replaced by any other sorting method capable of giving an unambiguous order, provided this method has been previously declared. (For example, if many players are unrated, sorting such unrated players in random order might be a fairer solution than alphabetical sorting.)

2.3 This ranking is used to determine the participant's Tournament Pairing Number ("TPN"); the highest ranked participant 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 TPNs may be reassigned accordingly to the corrections. No modification of a TPN for this reason is allowed after the fourth round has been paired.

Tournament Pairing Numbers (TPN) are used as a guide by most Swiss pairing systems – hence, changing them will change the subsequent pairings. We would expect this to happen, if at all, in the first round of a tournament – or, in some (rare) instances, even in the second or third round. When such changes happen, checking the pairings becomes difficult. To make such checks a little easier in the advanced stages of a tournament, this rule prohibits late changes of TPNs.

However, to be able to rate the tournament correctly, we must correct ratings, titles, and so on. Such data may be amended any time an error is discovered, even in late rounds (even after the tournament is finished!) – only, in late rounds, we do that without changing TPNs.

Please note that this restriction to the renumbering of participants only applies to corrections – late entries (see Article 2.4 below), whatever the entry round, should get their correct place in the list, and TPNs should be reassigned consequently.

2.4 A Late Entry is a participant who is only taken into account for the pairing of rounds after the first. If admitted to the tournament, late entries receive no points for unplayed rounds (unless the rules of the tournament say otherwise), and are given an appropriate TPN and paired only when they actually arrive.

The admission of a Late Entry is a responsibility of the Chief Arbiter (in agreement with the Organiser). A Late Entry may or may not be admitted to the tournament, depending on the specific conditions and, in some instances, on the event rules. (For example, these rules may stipulate that a participant be admitted to a later round with a given number of points – e.g., the so called "Half-Point Bye", or "HPB").

The rule also clarifies that Late Entries should not be given a TPN (and paired) until they actually join the tournament (thus avoiding the risk of having empty boards).

2.5 Due to late entries, the TPNs given at the start of the tournament are provisional. The definitive TPNs are given only when the List of Participants is closed, and corrections made accordingly in the results charts.

See also Article 2.3.

3. PAIRING, COLOUR, AND PUBLISHING RULES

3.1 Adjourned games (in an individual competition) or adjourned matches (in a team competition) are considered draws for pairing purposes only.

Nowadays, adjourned games or matches are only used when some unforeseen event prevents the game or match from being completed normally. So, usually, by the time the next round is paired, we do not know the outcome yet. Thus, to make the pairings, we need some presumptive result (which might well be wrong). This rule assigns to both participants a "temporary draw", to limit the possible difference with the real result. Since pairings obtained by means of this presumptive result are potentially incorrect, we should minimise the number of rounds in which the real result is unknown – i.e., complete the adjourned game or match as soon as possible.

3.2 Participants who withdraw from the tournament will no longer be paired.

3.3 Participants 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).

The exception clause (the one in parentheses) gives to Organisers the freedom to accept "Half Point Byes" in their tournament rules.

3.4 Only played games or matches count in situations where the colour sequence is meaningful. So, for instance, a participant with a colour history of BWBuW ("u" for unplayed, i.e., no valid game or match in round-4) will be treated as if their colour history was uBWBW. WBuWB will count as uWBWB, BWWuBuW as uuBWWBW and so on.

We look only at games that were actually played, skipping the possible "holes", which float to the top of the list. Thus, for example, in the comparison between the colour histories of two players, the sequences WuuB, WuBu, WBuu, uWuB, uWBu are all equivalent to uuWB.

3.5 Two paired participants, who did not play their game or match, may be paired together in a future round.

3.6 After a pairing is complete, sort the pairs before publishing them. The recommended sorting criteria are (with descending priority):

3.6.1 the highest score of the higher ranked participant of the involved pairs;

3.6.2 the highest sum of the scores of both participants of the involved pairs;

3.6.3 the smallest TPN of the higher ranked participant of the involved pairs.

Even when using a pairing software, we want to check board order before publishing the pairing, because some players interpret an incorrect board order as a "pairing error". Please note that these sorting criteria are only recommended, in order to allow the management of special needs – e.g., the arbiter can change the board order, or assign a fixed board to some players, to manage specific problems as well as health issues. Some pairing software can do this automatically.

When using an accelerated pairing system (see C.04.7), some participants receive virtual points that are added to the real score for the purposes of pairing and sorting pairs (see Article C.04.7:1.5).

4. COMPETITION RULES

4.1 Any prospective participants who have not confirmed their presence to a FIDE competition before the time scheduled for the drawing of lots shall be excluded from the tournament unless the Chief Arbiter decides otherwise.

The admission of a latecomer is a choice of the Chief Arbiter, who takes the final decision – and must take responsibility too (e.g., if during the round there are empty seats). Thus, before accepting a latecomer and making the actual pairing, we want to be quite sure that the player will 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.

When we accept late entries in the tournament, whatever the round, the new players should occupy their correct place in the players' list. Hence, entering a late player causes the TPNs to change according to the new ranking list. Some players will thus play the following rounds with a different TPN, and this may cause bewilderment. For example, consider a player, who is registered from the beginning, but enters a tournament (say, with 100 players) in the second round, as #31. In the first round that player had no TPN – hence, the players who (now) have numbers 33, 35, 37 and so on, in the first round had even TPNs and thus the colour opposite to that of player #1.

In addition, we want to remember that the limit imposed by Article 2.3 on the regeneration of TPNs does not extend to the case of a newly added late player.

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

4.3 If either

4.3.1 a result was written down incorrectly, or

4.3.2 a game or a match was played with the wrong colours, or

4.3.3 a player's rating has to be corrected,

and this is notified to the Chief 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 is notified after the pairing but before the end of the next round, it will affect the next pairing to be done. If the error is notified after the end of the next round, the correction will be made after the tournament and only for submission to rating evaluation.

The application of rules 4.2 and 4.3 requires Chief Arbiter to establish, and post, a timetable for the publication of pairings. 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, <u>as well as the final standings</u>, shall be prepared making use of the wrong result just as if it were correct. However, the rating variation must always be computed based on the actual results, so the correction is made just before submitting the tournament for rating.

4.4 Once published, the pairings shall not be changed unless one of the following situations occurs and the Chief Arbiter considers that changing the pairings is in the best interest of the tournament:

4.4.1 the accidental pairing of two participants who have already played each other

4.4.2 the rules of the specific competition explicitly allow the Chief Arbiter to modify the pairings

4.4.3 two participants close in the standings are without an opponent and both agree to play each other

4.4.4 a late entry is admitted to the tournament, the changes to the published pairings are minimal and are agreed by all participants involved

4.4.5 unforeseeable circumstances (such as a sudden withdrawal or an incorrectly recorded result) affect the top boards of the final round (Note: "top boards" are those identified as such by the Chief Arbiter)

The underlying principle is that a player could devote several hours to prepare for the announced opponent, and this effort should not be wasted.

However, in some instances, a modification of the pairings might be in order, and this article was extended to cover them all, including those mentioned in C.05 "General Regulations for Competitions". In the past, the only case when pairing alterations were allowed was when a match was (accidentally) repeated (on this regard, see also Article C.04.3:1.9.3 and the relevant comment). Now, we can intervene also when some participants have no opponent (extending what is mentioned in Articles C.05:6.4 and C.05:6.7.3), or when something occurs that can disrupt the correctness of the pairings in the final rounds (Article 4.4.5). The latter situation provides for cases in which the system, which usually "repairs" errors during subsequent pairings, has no time to do that, so that the final standings would be adversely affected. Of course, different tournaments may call for different decisions. In a high-level tournament, players would probably be upset by a change in the pairings. On the contrary, in a children tournament for beginners, it is far more important to have all children playing than to preserve an improbable preparation. It is always the responsibility of Chief Arbiters to use their sound judgement in the best interest of the tournament.

C.04.3
FIDE (DUTCH) SYSTEM

Approved by the Council on 28/10/2025
Applied from 1st February, 2026

0. TERMS AND DEFINITIONS

Terms and Definitions added at the 88th FIDE Congress in Goynuk 2017.
See https://tec.fide.com/2025-fide-dutch-terms-and-definitions.

1. INTRODUCTORY REMARKS AND DEFINITIONS

1.1 Tournament Pairing Number (TPN)

For the definition and management of TPNs, see Article 2 of the General Handling Rules for Swiss Tournaments (Initial Order and Late Entries).

1.2 Order

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

1.2.1 Score

1.2.2 TPN (in ascending order)

> *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.04.2:2). When we accept late entries, whatever the round, the new players should occupy their correct place in the players list. Therefore, the list should be sorted again, to assign new TPNs to the players (see C.04.2, Articles 2.3, 2.4, 2.5). The same may be done when some wrong rating had to be corrected. When this happens, some participants may play subsequent rounds with new, different numbers; if not adequately advertised, this change can muddle players who, in reading the pairings, still look for their old TPNs.*

1.3 Scoregroups and Pairing Brackets

1.3.1 A scoregroup is composed of all the players with the same score.

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

> *These two definitions solve any ambiguity between scoregroups and pairing brackets, stating that the scoregroup is the "backbone" of a bracket, which is made of a scoregroup together with all the players remaining from the pairing of the previous bracket. The players from the scoregroup are called resident players and have all the same score – this is called the resident score and is the "nominal score" of the bracket.*

Please note that the previous edition of these rules provided for a "Special Collapsed Scoregroup" (SCS), where resident players could have different scores. The big news in this edition is that this special scoregroup is no more required. This simplifies the logic of the pairings, which are now made following always the very same strategy.

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

In a homogeneous bracket there are no score differences between players – such brackets are made of just a (normal) scoregroup and nothing more, whereas heterogeneous brackets contain players who were not paired in the preceding bracket(s) and therefore have higher score(s).

1.3.4 A remainder pairing bracket ("remainder") is a sub-bracket of a heterogeneous bracket, containing some of its resident players (see Article 3.3 for further details).

Article 3.3 illustrates a way to build a candidate pairing for a bracket and explains how and when a remainder is built and used.

1.4 Floaters and Floats

1.4.1 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; for example, the bracket may contain an odd number of players, so that one of them remains unpaired; or the player has no possible opponent (and hence no legal pairing) in the bracket. Sometimes, two or more players share between them some possible opponents, but not enough of them, in such a way that no player is incompatible, but still 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 – this is sometimes called semi-incompatibility or island-(in)compatibility). In this case some player cannot be paired, although they are not incompatible, and will have to float.

Last, but not least, sometimes some players may have to float down to comply with the Completion criterion (see Articles 1.9 and 2.2) and thus allow the completion of the round-pairing.

In analogy to "downfloater", we often (but unofficially!) use the term "upfloater" to indicate a player paired with another one having a higher score (i.e., the opponent of a downfloater). Please notice that in other Swiss pairing systems (e.g., Dubov, TPS), the same term "upfloater" indicates a player transferred to a higher bracket, which is something completely different.

1.4.2 After two players with different scores have played each other in a round, the higher ranked player (see Article 1.2) receives a downfloat, the lower one an upfloat.

1.4.3 A downfloat is also given to any player who receives a pairing-allocated bye (see Article 1.5) or who, without playing in a round, scores more points than those rewarded for a loss.

1.4.4 No players other than those listed in the previous two articles can receive floats.

Downfloats and upfloats are a sort of markers used to record previous unequal pairings of the player. We keep track of such pairings because we want to minimise, and, as far as possible, avoid, their occurrence for the same player. In fact, 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 this rule tries to limit the repetition of such events in the same direction. The FIDE (Dutch) system uses a "local" approach to this problem, looking only to the last two rounds. On the contrary, the Dubov system adopts also a "global" approach, putting a limit on the number of floats in the whole tournament. The Team Pairing System uses yet a different approach, limiting the repetition of any float, independent of their direction.

We want to note that any player who scored points without playing in a round receives a downfloat also. This reduces the probability of a pairing with a lower-ranked opponent in the following two rounds.

1.5 Pairing-Allocated Bye ("PAB")

See Article 3 of the Basic Rules for Swiss Systems (*Should the number of participants to be paired be odd, one participant is not paired. This participant 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. This number of points shall be the same for all pairing-allocated byes*).

In other Swiss systems (e.g., Dubov, Burstein) the player, whom the pairing-allocated bye (PAB) is assigned to, is selected before starting the pairing for the round. In the FIDE (Dutch) system, on the contrary, the round-pairing (see Article 1.9) ends up with an unpaired player, who will receive the PAB. However, this version of the pairing system introduces important innovations in the choice of the PAB-assignee, whose score must now be as low as possible, and whose number of played games should now be as high as possible. We will discuss these innovations in detail in due time (see Articles 2.3.1, 2.4.4).

1.6 Colour Difference

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.

1.7 Colour Preference

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.

During the pairing process, we will try to accommodate (as much as possible) the colour preferences of the players – this is the reason for the good balance of colours of all modern Swiss systems. Participants, who have not played any games yet, just have no preference and shall therefore accept any colour (see Article 1.7.4).

1.7.1 An absolute colour preference 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 they played. The preference is for White when the colour difference is less than -1 or when the last two games were played with Black.

The preference is for 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. The only possible exception are high-ranked players in the last round. Those players can receive, if necessary, a 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 rounds that were played, skipping any unplayed games (whatever the reason may be), in compliance with Article C.04.2:3.4 (e.g., the sequence WBBWuW gives an absolute colour preference). Please note the difference with floats, for which we look only at the last two rounds of the tournament schedule.*

1.7.2 A strong colour preference occurs when a player's colour difference is +1 (preference for Black) or -1 (preference for White).

1.7.3 A mild colour preference occurs when a player's colour difference is zero, the preference being to alternate the colour with respect to the previous game they played.

> *It is worth noting that any disregarded colour preference, be it strong or mild, will give origin to an absolute colour preference in the next round.*
>
> *Of course, a strong or mild colour preference has no meaning when the same player has an absolute preference (see Article 1.7.1).*

1.7.4 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, we assign colours through rule 5.2.5 and the initial-colour decided by drawing of lots before the first round (see Section 5, Colour allocation rules). This is normal when pairing the first round but may sometimes happen also in subsequent rounds.*

1.8 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 rankings. Even if not all of them are really competing for top-ranking places, they are nonetheless likely to be of more 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). Hence, we apply special treatment to their pairings – namely, a player may receive the same colour three times more than the other one, or three times in a row, if this is needed to make them meet an opponent better suited to the strength the player demonstrated.*
>
> *Please note that the player's score must be over 50% – i.e., a player who has half the possible points is not a topscorer. For example, in a nine-rounds tournament, the maximum score a player can have before the last round is eight points. Thus, the topscorers are all the players with 4.5 points or more (please note that topscorers are not top-scorers – top-ranking players will probably have more than six points).*

1.9 Round-Pairing Outlook

This article essentially gives a panoramic vision of the pairing process. Additionally, it also states that the PAB-assignee is a downfloater. This is very important because, in pairing the last bracket (which is when we allocate the PAB), we must consider possible previous downfloats, and select the assignee accordingly.

1.9.1 The pairing of a round (called round-pairing) is complete if all the players (except at most one, who downfloats from the last bracket and receives the pairing-allocated bye) have been paired and the absolute criteria [C1]-[C3] *(see Article 2.1)* have been complied with.

This definition does not refer to a single bracket but to the complete round. Thus, we cannot accept unpaired players (apart from a possible PAB) – all players must be paired. The requirements for a complete pairing, which are set forth here, are therefore the fundament for the Completion test (see [C4]).

On the other hand, the constraints for a complete pairing are definitely loose, not to say minimal – we only ask for it to be legal, i.e., 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 are possible for each round, but "The Pairing" – the right one – is the one that best satisfies all our pairing criteria.)

1.9.2 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. The pairing of this bracket can possibly leave some downfloaters that, together with the next scoregroup, will form the next bracket, and so forth – until all players have been paired.

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

There are some (luckily, rare) instances when no pairing at all is possible without violating some fundamental rules – i.e., absolute criteria. In those exceptional circumstances, this rule gives the right, and the duty as well, to the Chief Arbiter to break such rules, and act according to their best judgment to avoid the disruption of the tournament, if possible.

Before starting the pairing of the first bracket, we must verify that at least one legal pairing exists for all the players. This condition is informally called the "Requirement Zero". Its check is called a "Completion test" and it is required by the Completion criterion (see [C4]. This test is always reasonably simple because we are not looking for the "right" pairing, we only want to make sure that a legal pairing exists.

We must perform this test before pairing the first bracket. If it fails, then there is no way at all to complete the round-pairing – we have an impossible pairing – which is unwelcome news. The Chief Arbiter can then usually choose among prematurely terminate the tournament and send (angry) players home; give a "strategic" bye to a well-chosen player; make a player meet a same opponent for the second time; or do anything else that can make *the pairing possible. For example, if the number of players is odd, we could treat a forfeit win for one or more players (as low as possible in the ranking) as if they were played games, so that that player (or one of those players) can get a PAB, possibly resolving the impasse.*

If this early test succeeds, we start to pair the bracket already confident that (at least) one legal pairing exists – so we cannot be stuck!

Then, we perform this check again – and, in fact, repeatedly – when we choose the downfloaters set, to verify that all the players still unpaired, together with the downfloaters left from the current bracket (of course, possibly none), allow a legal pairing. If the Completion test fails, the candidate pairing for the bracket is discarded and we look for a valid one.

Note: Article 2 defines all the criteria that the pairing of a bracket has to satisfy (in order of priority).

Article 3 describes the procedures for pairing a bracket.

Article 4 defines the rules for the sequential generation of the pairings.

Article 5 defines the colour allocation rules that determine which players will play with White.

Pairs are made based on expected colours (also), but colours should be assigned to players at the end of the pairing, at least in principle. (However, in practice, rather than waiting for the pairing process to be complete, we usually assign colours while pairing is still in progress.)

HISTORICAL NOTE

In previous Rules, the Completion test was performed only at the end of the bracket pairing. When it failed, the pairing was repeated, but this time using different rules for the choice of downfloaters. Now, on the contrary, the check is performed during the pairing process, becoming an essential part in the choice of downfloaters. Thus, we need no special pairing route or rules, and the pairing process is unified.

The Rules do not dictate a method to enforce compliance with the pairing criteria – 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), provided the rules are fully complied with.

2. PAIRING CRITERIA

2.1 Absolute Criteria

No pairing shall violate the following absolute criteria:

Absolute criteria correspond to some of the requirements of Section C.04.1, "Basic Rules for Swiss Systems", which we may want to look at closely. Those criteria must be complied with, <u>always.</u> They cannot be renounced, whatever the situation (but there are situations in which no pairing at all exists, which complies with the absolute criteria – in such cases, Chief Arbiters must apply their better judgment to *find a way out of the impasse, see 1.9.3). To enforce these criteria, players may even float as needed.*

2.1.1 [C1] See the Basic Rules for Swiss, Article 2 (*Two participants shall not play against each other more than once*).

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

2.1.2 [C2] See the Basic Rules for Swiss, Article 4 (*A participant who has already received a pairing-allocated bye, or has already scored in one single round, without playing, as many points as rewarded for a win, shall not receive the pairing-allocated bye*).

Please notice that only PABs, forfeit wins and full-point byes prevent the allocation of a PAB (see Article 1.5). A player who received a requested bye (a half-point bye or an announced absence) may receive the PAB in a subsequent round, although the new criterion [C9] reduces the probability of such events. (For PAB assignment, see also [C5] and [C9].)

2.1.3 [C3] Non-topscorers (see Article 1.8) with the same absolute colour preference (see Article 1.7.1) shall not meet (see the Basic Rules for Swiss, Articles 6 and 7).

In general, a pairing between two players who have the same absolute preference is forbidden by C.04.1, Articles 6 and 7. In the last round of a tournament, this can obstruct a game that would be decisive in determining correct rankings (for example, assume that #1 and #2 of the ranking did not meet, but have the same absolute preference – in FIDE (Dutch) philosophy, they should play against each other for the first prize). Because of this, Article 1.8 defines the "topscorers" – players who, in the pairing of last round, have a high score (i.e., above half the possible score). Such players (who are of course well-known during this last pairing, but their opponents are still unknown) should therefore play with the "best possible" opponent, i.e., we want to pair them in the best conceivable way – and that's why this criterion frees them and their opponents from the duty of compliance with absolute colour preferences.

Two players, who cannot be paired to each other without infringing criteria [C1] or [C3], are said to be incompatible.

2.2 Completion Criterion

2.2.1 [C4] A pairing complying with all the absolute criteria (see Article 2.1) shall always exist for all players not yet paired.

This is sort of an absolute criterion too, but in a class of its own. If, at <u>any</u> stage of the pairing process, a legal pairing must always exist for all players not yet paired, then every pair must be built in such a way that the completion of the round-pairing remains possible (e.g., we could not pair the only player that can receive the PAB) – but it is in the choice of downfloaters that we *actually verify that the completion of the round-pairing is possible – independent of the optimisation of the next bracket (which is sought for by criterion [C8] below).*

In the previous edition of the Swiss rules, this criterion applied only to the processing of the now abolished "Penultimate Pairing Bracket" (see the Foreword) – i.e., only after a Completion test failure. Now, on the contrary, it applies to all brackets, and this brings a dramatic change in the pairing strategy, because a failure of the Completion test after the pairing of a bracket is now impossible.

Please note that [C4] precedes both [C6] and [C7] Compliance with this criterion may therefore cause a reduction in the number of pairs, or an increase in the (local) overall difference in score between paired players.

Criterion [C4] even precedes [C5] (or "PAB Criterion"), so it could even prevent the allocation of the PAB in the lowest possible scoregroup.

2.3 PAB Criterion

2.3.1 [C5] Minimise the score of the assignee of the pairing-allocated-bye.

In previous editions of the rules, it was sometimes possible that the PAB ended up to a leading player, even the topmost one, while most people would expect it to go to the lowest possible position of the ranking. This caused (understandable) bafflement among players, who considered it not a "proper" pairing. This new criterion was then introduced, which forces the assignment of the PAB to a player as low-ranked as possible.

The scoregroup in which the potential assignees of the PAB reside (the so-called "PAB-candidates") is usually called the "PAB-candidates scoregroup" (PCSG for short). We want to determine it right at the beginning of the pairing process, as this tells us the score of the PAB-candidates – which is most useful information.

Please note that, when (only) one player downfloats from the PCSG, that player will then continue to downfloat, bracket after bracket, until the last one, to be finally assigned the PAB (no player in the subsequent scoregroups can get the PAB, or else the current scoregroup would not be the PCSG).

(For PAB assignment, see also [C2] and [C9].)

2.4 Quality Criteria

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

The criteria in the previous paragraphs (Articles 2.1–2.3) set conditions that <u>must</u> be obeyed. A candidate pairing that does not comply with them is not legal and is therefore immediately discarded. The following criteria are of a quite different kind, in that they establish a frame of reference for a <u>quantitative</u> evaluation of the quality of the pairings. This is done by setting a sequence of "test points", in order of decreasing importance, checking the pairing in accordance with 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 rather a numerical quantity. We measure it by means of a "failure value" (FV), whose meaning is tightly connected to the criterion itself. For example, the failure value for [C6] could be the number of pairs less than the maximum possible (see MaxPairs, Article 3.1.2); or the list of downfloater scores that are higher than the minimum, for [C7]; or the number of players not getting their colour preference, for [C12] and [C13]; and so forth.

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

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

Relative criteria are not so strict as absolute ones, and they can be disregarded, if this is needed to achieve a legal pairing. 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! The spare player in odd brackets and incompatible or semi-incompatible players (see Article 1.4.1 and Article 3.1.3) are "inevitable" floaters. However, players could have to float also to ensure compliance with the Requirement Zero (see Article 1.9.3) or to allow the best choice of the PAB-assignee (see Article 2.3.1). This too is evidence of the attention of the FIDE (Dutch) system towards the choice of the "right strength opponent".

2.4.1 [C6] Minimise the number of downfloaters (*equivalent to: maximise the number of pairs*).

The first quality factor is the number of pairs, because its reduction increases the number of floaters. Maximising it means building MaxPairs pairs, which is the maximum number of pairs that can be built (see Article 3.1.2).

2.4.2 [C7] Minimise the scores (taken in descending order) of the downfloaters.

With the abolition of the Special Collapsed Scoregroup and Collapsed Last bracket (see comment to Article 1.3.2), the new rules could renounce the old concept of PSD in favour of a simpler way to obtain the same results.

Criterion [C6] determines the number of players that must downfloat from the bracket, but nothing is said about <u>who</u> those downfloaters should be.

Criterion [C7] tells us to choose as downfloaters those players who have the lowest possible scores. Of course, this criterion is useless in homogeneous brackets, where all possible downfloaters share the same score. On the contrary, it is essential in those brackets that contain MDPs. In general, if only possible, no MDP must float again – however, if some of them must necessarily float again, they must be those who have the lowest scores.

In addition, this criterion specifies that their scores must be minimised in descending order – this means that we must first minimise the highest between them, independent of what happens to the remaining ones; then we minimise the second highest; and so on. Sometimes, minimising the highest downfloater score might worsen the scores of subsequent downfloaters; nevertheless, we do it this way. (We will see an example of this presently.)

In such brackets, the choice of the downfloaters set is somewhat more complex than usual, as we need to do several things.

- First, we determine all the possible downfloaters sets, each of which must contain the number of downfloaters fixed by [C6]. Those downfloaters must be chosen in such a way that they allow the successful pairing both of the players that remain in the bracket from which they are downfloating, and of the whole Rest. Moreover, when required, they must also allow the allocation of the PAB to a player belonging to the PCSG.

- Then, we associate to each set of downfloaters the list of the corresponding scores, sorted in the required descending order. Since different players may well have the same score, several sets may share the same list.

- Now, we must sort those lists in lexicographic order. The first list corresponds to the set(s) of downfloaters with the lowest scores, among which we must choose the actual downfloaters set (all sets corresponding to subsequent lists are discarded).

The chosen list represents the scores of downfloaters (some of which will be residents, others MDPs). Its initial part contains the list (in descending order, by construction) of the scores of the MDPs that downfloat again – we call this part the Refloaters Score List (RSL for short). This list can be used as a tool that will be useful later in the pairing (see Article 3.1.3).

From all the above, it follows that the higher scored MDPs are going to be paired (thus keeping the differences in score as little as possible). We already mentioned that, in some instances, the choice to downfloat the lowest scored MDPs – and, thus, to pair the highest ranked ones – may even cause a reduction in the total number of paired MDPs (nevertheless, the "top" MDPs must be paired first, whatever the price). This may seem rather obscure, so an example may be in order.

Example

Suppose that we have a bracket that must supply two downfloaters (as directed by [C6]), and let 1, 2, 3, and 4 be four MDPs with decreasing scores, while 5, 6, 7, and 8 are the residents of the bracket: {$1^{(5)}$, $2^{(4)}$, $3^{(3)}$, $4^{(2)}$, 5, 6, 7, 8}. Also, suppose that #6, #7 and #8 cannot float because they are incompatible in the next bracket, which is the last one; so, at least one MDP is going to downfloat. The possible opponents for the players in the bracket be respectively Opps(1)={5,7}, Opps(2)={5,7}, Opps(3)={6,8}, Opps(4)={6}, Opps(5)={1,2}, Opps(6)={3,4,8}, Opps(7)={1,2}, Opps(8)={3, 6} (similar situations are not unusual in later rounds of a tournament).

Now, let us examine the effects of different choices of the downfloaters. If we make #4 float, the only possible pairing for #6 is with #8 (who cannot float), so #3 shall float also. If we choose #3 as downfloater, again we must have the *pair 6-8, so #4 must float. In both cases, two MDPs (#3 and #4) are going to downfloat. Choosing #2, we would have the pairing 1-7, 3-8, 4-6, with #5 as a floater, i.e., only one MDP should float again. However, and this is the point, [C7] prevents this pairing, so two MDPs are going to float again. Finally, we can also compile the Refloaters Score List {3, 2}, which indicates the scores allowed for prospective downfloaters, and can be used to help the subsequent choice of the actual downfloaters set.*

*

The location of [C7] before the colour-related criteria ([C10]-[C13]) is suggestive of the attention that the FIDE (Dutch) system gives to the choice of an opponent who has the "right strength" rather than one with the "right colour".

2.4.3 [C8] Choose the set of downfloaters so that in the following bracket every criterion from [C1] to [C7] (see Articles 2.1 to 2.4.2) is complied with.

When we get here, we have already done many things. We have complied with all absolute criteria (hence, the pairing is a legal one); made sure that the round-pairing can be completed; minimised the PAB-assignee score; and optimised the most important pairing quality parameters (number of pairs, downfloaters' scores).

Before proceeding to optimise colours and the treatment of MDPs, we look ahead into the next bracket ([C8]). The key point is that we do not want to come back to the current bracket ever again. Thus, we must make sure that the choice of the downfloaters we are going to send to the next bracket is the one that, in the same next bracket, best complies with [C6] and [C7], beyond of course criteria [C1]-[C5], which must be complied with always – and, in fact, ensuring compliance with [C4] can be somewhat tricky. From the current bracket, we must choose a set of downfloaters "well suited" to the next bracket – which, in turn, can produce downfloaters, and those downfloaters must allow the round-pairing to come to fruition ([C4], Requirement Zero). The minimisation of the PAB-assignee score ([C5]) may raise additional difficulties. It is worth noting that, although this criterion only addresses the next bracket, it might well influence subsequent brackets too.

- *First, we check that the downfloaters (which will become the MDPs of the next bracket) 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 number of players. Also, let us suppose that (1) one player in the next scoregroup has no possible opponent and that (2) we can choose between two possible downfloaters with the same score, both compatible in the destination bracket, but only one of them can be paired to the "problematic" player. Then, we must choose the downfloater that can be paired with that player, because the other one would leave an incompatible player in the destination bracket (and therefore a downfloater that could be avoided). (Note that, if the next scoregroup contained an even number of players, then the bracket built with it and the current downfloater would have an odd number of players – it would then necessarily yield at least one* *downfloater, so that, probably, the choice of the downfloater would not be critical for the number of pairs.)*

- *Only when the number of pairs have been maximised, we proceed to optimise the choice of downfloaters <u>in the destination target</u>. In the previous step, we focused on the pairing of all resident players – now, we are focusing on the pairing of MDPs. This means that, if we can choose between two or more possible downfloaters and all other conditions are equivalent, we must choose a downfloater that can be paired, thus giving the lowest score for the downfloater that will result from the next bracket. (All resident players of the next bracket have the same score – however, players that cannot be paired in the next bracket will have to float again.)*

This optimisation is extended <u>only to the next bracket</u>. In fact, there are situations in which a minor change in a previous pairing would bring in large benefits – however, looking several brackets ahead would be an operation too difficult to be carried on every time. So, the rules settle for a practical optimisation, renouncing what is out of reasonable reach. But this is not the only reason – in the core philosophy of the FIDE (Dutch) system, the pairings for higher ranked players are considered far more important than those for lower ranked ones. Changing the pairing of the current bracket for the benefit of some player located two brackets below this one, would be opposite to that philosophy (even if, sometimes, we must do just that, to be able to assign the pairing-allocated bye to the lowest-scored player as per [C5]).

2.4.4 [C9] Minimise the number of unplayed games of the assignee of the pairing-allocated-bye.

Note: Apply to brackets that downfloat exactly one player, who will end up receiving the pairing-allocated bye.

This is a brand-new article. The underlying idea is that a player who already had fewer games than the other players in the bracket should not get another unplayed round if possible. Now, although the validity of the principle is general, there are some instances in which a full application of this criterion would be so complex as to involve too many brackets (sometimes, even those that precede the PCSG defined in the comment to 2.3.1). Hence, the note limits somewhat the scope of this criterion, which applies in many cases but not always.

For example, let B_A be a bracket with a higher score than the PCSG, and suppose that, in this bracket, we can choose among two or more possible downfloaters. It may then happen that only one of those possible downfloaters will allow to choose, in the PCSG, the downfloater with the highest number of played games – while choosing another downfloater in B_A will force us to choose a different downfloater in the PCSG, with fewer played games. In conclusion, to apply this criterion to a general extent, we should analyse the choice of downfloaters, and its consequences, in <u>every</u> bracket, which is impractical.

2.4.5 [C10] Minimise the number of topscorers or topscorers' opponents who get a colour difference higher than +2 or lower than -2.

2.4.6 [C11] Minimise the number of topscorers or topscorers' opponents who get the same colour three times in a row.

Having made sure that both the number of floaters and their scores are at a minimum, we now try to optimise colour matching. In fact, colour is less important than score differences – that's why, consistently with the inner logic of the system, the colour allocation criteria are located after those that address the number of pairs and choice of downfloaters, both in the current and in the following bracket.

Criterion [C3], in accordance with Articles C.04.1:6 and C.04.1:7, states that when two non-topscorers meet, their absolute preferences shall 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 has the same absolute preference – and that prospective opponent may or may not be also a topscorer. The outcome of those players' games can be important in determining the final ranking and podium positions, and such exceptions are explicitly provided for by Articles 6 and 7 just mentioned. We are therefore allowed to compose such pairs, choosing the best matched opponent – but there must not be more of them than the bare minimum!

The subdivision into two individual rules, [C10] and [C11], 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 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 absolute preference for the same colour. These two rules equate the players of the pair – thus, players may sometimes be denied their absolute colour preference, just as if they were a topscorer, even if they are not!

2.4.7 [C12] Minimise 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.

In the previous phases of the bracket pairing, we determined the number of floaters ([C6]) – thus, we know by now the number MaxPairs (see Article 3.1.2) of pairs that the bracket produces (it is the total number of players in the bracket reduced by the number of floaters and divided by two).

Let us suppose that ***m*** *players prefer a colour and* ***n*** *players prefer the other one, with* ***m ≥ n –*** *let us call those* ***m*** *players the "majority", and the* ***n*** *players the "minority". Sometimes, in addition to those m+n players, the bracket contains* ***a*** *more players who have no colour preference ("a" for "any") and, therefore, can get any colour. In principle, such players should get the minority colour, so that they subtract to the number of disregarded* *colour preferences. Thus, we have a total of* ***n+a*** *players who can "accommodate" the* ***m*** *players who belong to the majority.*

As mentioned, we must build ***MaxPairs*** *pairs. Among those,* ***n+a*** *pairs can satisfy both colour preferences, while the remaining* ***x = MaxPairs-n-a*** *cannot help but disregard one colour preference – but, of course,* ***x*** *cannot be less than zero (a negative number of pairs has no meaning). Thus, we obtain a general definition for* ***x****:*

x = max (0, MaxPairs-n-a)

A perfect pairing always has exactly ***x*** *disregarded colour preferences – no more (we don't want them!), no less (we can't!).*

In practice, there might be even more pairs in which a player does not get the expected colour. This follows from incompatibilities due either to absolute criteria or to 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. (Please note that transpositions (see Article 3.6.1) swap players beginning with the last positions of S2 (see Article 3.2.1) 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 high probability to be changed sooner than a similar pair located at the top. Perfect pairings with top "colour-defective" pairs have therefore a higher probability.)

Incidentally, we might also mention that players often seem to worry about "colour doublets" (like, for example, WWBB) and some 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 somewhat more often in the top pairs of the bracket, therefore involving higher ranked players, and this makes them more noticeable.

2.4.8 [C13] Minimise 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.

Of course, if x=0 all preferences must be complied with, be they strong or mild. Moreover, in principle, it should always be possible to comply with a colour preference belonging to the minority. Therefore, we only consider this issue when there are both strong and mild preferences of the majority colour, so that some mild preferences can be "sacrificed" to satisfy just as many strong ones. In fact, a strong preference can be complied also with by pairing the player with an opponent who has a mild preference for the same colour, because a strong colour preference (or absolute, at that) always pre-empts a *mild one in a clash.*

*As we did for preferences in general, we would like to know how many strong preferences will be unavoidably disregarded. Now, the minimum number of players not getting their strong colour preference, which is usually represented by **z**, is part of the total number **x** of disregarded colour preferences (see note to [C12]) – hence, **z** is at most equal to **x**.*

*For instance, let the number W_S of white seekers (players whose colour preference is for White) be greater than the number B_S of black seekers (we call White "the majority colour"). The **x** players will typically all be White seekers, and as many as possible among them should have mild colour preferences, while the rest will have strong colour preferences. (During the last round, some absolute colour preferences might be disregarded for topscorers or their opponents (see [C10], [C11]), so that part of **x** may represent such players. In those instances, our line of reasoning should be suitably adapted.) We can estimate **z** simply as the difference between **x** and the number W_{SM} (resp. B_{SM}) of White (resp. Black) seekers who have a mild colour preference, with the obvious condition that **z** cannot be less than zero. Hence:*

$z = \max(0, x - W_{SM})$ if $W_S \geq B_S$ *(White majority)*
$z = \max(0, x - B_{SM})$ if $W_S < B_S$ *(Black majority)*

With a careful choice of transpositions and/or exchanges, we might be able to minimise the number of disregarded strong preferences. Since the total number of disregarded preferences cannot change (again, we cannot have it smaller, and we do not want it to grow larger!), this may only happen at the expense of the same number of mild preferences.

*An example can shed some light on the matter. Consider the bracket {1B, 2b, 3B, 4b}, where B is a colour strong preference, b is a mild one. Here, we have **x**=2, but **z**=0. This 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 several reasons (e.g., incompatibilities), the number of players who cannot get their strong preference may be greater than that.

2.4.9 [C14] Minimise the number of resident downfloaters who received a downfloat the previous round.

2.4.10 [C15] Minimise the number of MDP opponents who received an upfloat the previous round.

2.4.11 [C16] Minimise the number of resident downfloaters who received a downfloat two rounds before.

2.4.12 [C17] Minimise the number of MDP opponents who received an upfloat two rounds before.

This group of criteria ([C14]-[C17]), which apply to residents only, aims to optimise the management of floaters, which is the last step towards the perfect pairing.

Article C.04.1:5 states that, in general, players should meet opponents with the same score. This is best achieved by pairing players inside their own scoregroup. However, there are situations in which a player cannot be paired in their scoregroup and then, by necessity, must float. These criteria limit the frequency with which such an event can happen to the same player – but they are 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 slight residual asymmetry in the treatment – viz. downfloaters are (just a little bit) more protected than upfloaters. In some other Swiss systems, floaters' opponents are not considered floaters themselves, and, therefore, enjoy no protection at all.

We want to note that criteria [C14] and [C16] do not protect MDPs – their scope is explicitly limited to resident players. This is because MDPs are players who floated into the current bracket and, therefore, there is no point in protecting them against floating – they are already downfloaters! However, this does by no means imply that they can downfloat again easily – in fact, they are protected by all the criteria that minimise the score of downfloaters in the current bracket ([C7]) and in the next one (through [C8], which enforces [C1]-[C7] in the next bracket); and by criteria ([C18] and [C20]).

2.4.13 [C18] Minimise the score differences (taken in descending order) of MDPs who received a downfloat the previous round.

2.4.14 [C19] Minimise the score differences (taken in descending order) of MDP opponents who received an upfloat the previous round.

2.4.15 [C20] Minimise the score differences (taken in descending order) of MDPs who received a downfloat two rounds before.

2.4.16 [C21] Minimise the score differences (taken in descending order) of MDP opponents who received an upfloat two rounds before.

Criteria [C14]-[C17] minimise the number of players who, having recently floated, may have to float again in this round. However, those rules give no special protection either to a player who, being an MDP in a bracket in this round, cannot be paired and must therefore float down again, or to those residents who have to face an MDP. Such players and their opponents will have larger score differences than their fellow "single" floaters – they are usually called "double floaters" or "Refloaters".

Criteria [C18]-[C21] take care of protected players whose protection has already failed once or more and try to prevent such players from further floating. Criteria [C18] and [C20] apply when some MDPs have to float down again. In this case, we should, as far as possible, choose those MDPs that are not (or are least) protected because of previous downfloats. Criteria [C19] and [C21] apply when some residents have to face a MDP from at least two scoregroups up. In this case, we should choose, as far as possible, *those residents that are not (or are least) protected because of previous upfloats.*

Although the score difference of a downfloater in not explicitly defined by the rules, we should assume that it is greater than any score difference of a resident player (even when this resident has zero score).

We must consider this when dealing with a downfloater who will become the PAB assignee. If such downfloater must be an MDP, and higher priority criteria do not discriminate (scores, number of games, colour preferences, and so on, are all equal), the application of criteria [C18] and [C20] ensures that the PAB goes to an MDP who is not protected against downfloating rather than to one who recently floated – and is therefore protected.

Examples

• *Let us suppose that a bracket contains two MDPs with different scores, and two resident players (hence, with identical score), only one of which is protected against upfloat. Thus, each resident must be paired with one of those two MDPs, but the protected resident should be paired to the MDP who has the lower score. Please note that, if we had only one resident, this resident would be paired to the higher-scored MDP, independent of float protection, and even of colour matching, because criterion [C7] has higher priority than either of these other criteria.*

• *Another interesting example is that of two MDPs with identical score and colour preference, only one of which is protected against downfloat, sitting in a bracket containing only one resident player. In this case, the resident will be paired to the protected MDP, while the other one is bound to double-float.*

3. 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 build the pairing and evaluate its quality. In fact, it can be readily adopted as a guideline to make – or, far more often, explain – a pairing.

3.1 Parameter Definitions

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

3.1.2 MaxPairs is the maximum number of pairs that can be produced in the bracket under consideration (see [C6], Article 2.4.1).

Note: 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.

At the beginning of the pairing process, MaxPairs, which is a constant of the bracket, is unknown – so we need to make an "educated guess" for it. In fact, *we only know the total number N of players in the bracket and the number M0 of MDPs entering the bracket.*

Of course, 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).

The actual value of MaxPairs can be less than that because some players might prove impossible to pair in the bracket. Moreover, a bracket could have to provide the downfloaters required to complete the round-pairing (see [C4]), and that, too, would detract from the number of pairs that can be built. (This includes the downfloaters who can possibly be required to be able to assign the PAB in the scoregroup selected by means of [C5].)

Hence, the process to find the exact MaxPairs value is somewhat empirical and may require some experimenting. In practice, we start with a plausible guess – then, if we cannot make all those pairs, i.e., our initial estimation of MaxPairs was too optimistic, we gradually decrease the expected number of pairs. Any remaining players will eventually float down into the next bracket.

3.1.3 M1 is the number of MDP(s) that are paired in the bracket.

Note: 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. M1 can never be greater than MaxPairs.

In each bracket, we have a given number M0 of MDPs (who are the downfloaters of the previous bracket, possibly none), but we have no certainty that all those MDPs can be paired in this bracket. 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 as described in the comment to Article 1.4.1.

In the comment to [C7], we defined the Refloaters Score List, which contains the scores of the MDPs that cannot be paired and, therefore, are bound to float again. Now, we define a second parameter, M1, representing the number of MDPs that are paired. This number is the difference between the total number of MDPs and the number of Refloaters, which is also the length L_{RSL} of the Refloaters Score List, i.e.,

$M1 = M0 - L_{RSL}$.

Of course, M1 is less than or equal to M0. Ultimately, the bracket can yield at most MaxPairs pairs, among which M1 pairs contain an MDP.

From a different point of view, we should mention that, while M0 is a well-known constant, we usually do not know how many players, and especially MDPs, can be paired, until the actual pairing is made. Therefore, in fact, we do not know the true value of M1 yet, and we must 'predict' it somehow – the RSL is a powerful new tool, which can be used to do just that. Alternatively, a first educated guess is M0 minus the number of incompatible MDPs (if any). When there are semi-incompatible players (see comment to Article 1.4.1), we must determine how many of them cannot be paired and subtract them too.

> *Actually, we need to "divine" the Refloaters Score List, and thus M1, as well as MaxPairs, out of sound reasoning (based on [C6] and [C7]), assuming a tentative value, which might initially be wrong. Nonetheless, those numbers, however identified, are constants – that is why there is no rule to change them.*

3.2 Subgroups (Original Composition)

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

3.2.2 S1 initially contains a group of players (sorted according to Article 1.2). If the bracket is homogeneous, the group consists of the first MaxPairs players in ascending order of TPN, otherwise it consists of the first set of M1 pairable MDPs as defined by Article 4.4.2.

> *In a heterogeneous bracket, we have players who are already floating down (i.e., MDPs) and therefore need some special protection against further floating in the same round. This is why the composition of the original subgroups is different for homogeneous and heterogeneous brackets.*
>
> *If the bracket is heterogeneous (M0 ≠ 0), then we must populate S1 with a set of M1 MDPs, chosen in such a way that the scores of the MDPs that we are not going to pair coincide with the RSL we "divined" in the previous step. We will then pair them first, before proceeding with the rest of the players.*
>
> *Since we are focusing on pairing MDPs first, in the subgroup S1 we put M1 MDPs. If there is no way to make all those pairs, our guess for the value of M1 was apparently too optimistic and, therefore, we try a different value until we guess the right one. Any remaining MDPs can't help but join the Limbo (see Article 3.2.4) and shall eventually float again (after the completion of the pairing for the bracket). The number of pairs made in the MDP-pairing will be subtracted from the total number of pairs to be made in the bracket, yielding (a plausible guess of) the number of pairs we must build in the remainder (which contains all the resident players remaining after the MDP-pairing). Then, finally, to pair the remainder, we put in S1 the top half of its players (which means that we are trying to pair all of them).*
>
> *To avoid any misunderstanding, please note that the pairing of the MDPs and that of the remainder are two phases of a single operation, which is performed as a unit. Thus, "pairing MDPs first" is just a procedural indication that has nothing to do with the order of generation of candidates. Independent of the method and algorithm used to generate them, each candidate must be regarded as a whole unit. When we choose the 'earlier' candidate from a pool of equivalent ones, we only consider the order of generation of the complete candidates.*
>
> *A homogeneous bracket, by definition, does not contain MDPs and is thus essentially identical to a remainder. To pair it, we simply populate the subgroup S1 with MaxPairs players, which means that we are trying to pair the entire bracket all at once.*

3.2.3 S2 initially contains all the remaining resident players.

3.2.4 When M1 is less than M0, some MDPs are not 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.

After M1 moved-down players were selected for pairing, the remaining MDPs, in number M0-M1, cannot be paired in the bracket. (The number and scores of such players are defined in the Refloaters Score List, if we use it – see Article 2.4.2.) Those players are not necessarily incompatible in the bracket – there may just be no place for them (for example, if two MDPs share the same only possible opponent, neither of the two is incompatible – nevertheless, either one of the two cannot be paired). Those players form a special subgroup, called the "Limbo" (of course, the number and scores of the players in the Limbo are the same of those of the players whose scores are included in the Refloaters Score List. In other words, there is a one-to-one correspondence between the scores in the RSL and those of the players who belong to the Limbo). During the pairing proceedings, some players might need to be swapped between S1 and the Limbo (to be exact, we change the distribution of MDPs between S1 and Limbo, but the final effect is tantamount to swapping players – we will discuss this in Article 3.7.3 and Article 4.4). However, at the end of the pairing, the players still in the Limbo will be bound to float again.

3.3 Preparation of the Candidate

3.3.1 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.

3.3.2 In a homogeneous bracket: the pairs formed as explained in Article 3.3.1 and all the players who remain unpaired (bound to be downfloaters) constitute a candidate (pairing).

A candidate can contain no downfloater (this is normal when the number of players in the bracket is even). Somewhat less usual, but not at all impossible, is the case in which the candidate contains no pairs (only downfloaters).

3.3.3 In a heterogeneous bracket: the pairs formed as explained in Article 3.3.1 match M1 MDPs from S1 with M1 resident players from S2. This is called an MDP-Pairing. The remaining resident players (*if any*) give rise to the remainder (see Article 1.3), 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.

Sometimes, all the MDPs are incompatible in the bracket. This is not unusual in late rounds and high brackets, where players already met and there is no choice but make them float (maybe several times) until we can give them a viable opponent.

3.3.4 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 can be paired (i.e., if M1 is zero), the first step is omitted.

Thus, in general, the candidate is comprised of three parts:

- *an MDP-Pairing (heterogeneous brackets only), made of M1 pairs (possibly none) each containing an MDP and a resident player,*
- *a set of pairs of resident players, coming from the pairing either of a homogeneous bracket or of a remainder from a heterogeneous bracket,*
- *a set of unpaired players, containing players from the Limbo (if any) together with resident players that could not be paired. All those players cannot help but get a downfloat.*

3.4 Evaluation of the Candidate

3.4.1 If the candidate built as shown in Article 3.3 complies with all criteria from [C1] to [C5] (see Articles 2.1 to 2.3), and all the quality criteria from [C6] to [C21] (see Article 2.4) are fulfilled, the candidate is called “perfect” and is (immediately) accepted. Otherwise, apply Article 3.5 in order to find a perfect candidate; or, if no such candidate exists, apply Article 3.8.

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 2. If we are 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 (see Article 3.5). If this proves impossible, the last resource is accepting a candidate that, although it is not perfect, is nonetheless the best we can have (see Article 3.8).

Of course, a candidate that does not comply with the absolute criteria is illegal and therefore is not even acceptable. This also includes candidates that do not allow all the remaining players to be paired (see the “Requirement Zero”, in the comment to Article 1.9.3) and candidates in which the pairing allocated bye is not assigned to a player with the lowest possible score.

As we mentioned above, the evaluation is performed only when we have a complete candidate. However, in several instances we can save lots of time by applying the pairing criteria smartly. For example, if we are pairing a heterogeneous bracket and a pair in the MDP-pairing is illegal, no possible candidate containing that MDP-pairing can be legal – so we can safely discard it at once.

3.5 Actions when the Candidate is not perfect

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

3.5.2 Articles 3.6 (for homogeneous brackets and remainders) and 3.7 (for heterogeneous brackets) define the precise sequence in which the alterations must be applied.

3.5.3 After each alteration, a new candidate shall be built (see Article 3.3) and evaluated (see Article 3.4).

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 subgroup S2 only, then in S1 and S2 in homogeneous brackets and remainders, and finally in S1 and the Limbo in heterogeneous brackets. Each time we apply an alteration, we repeat the preparation and evaluation of candidates. There are, in fact, two different sequences:

- *one for homogeneous brackets (Article 3.6), which contain no MDPs (this sequence also applies to remainders),*
- *one for heterogeneous brackets (Article 3.7). Those brackets contain MDPs, some of which (in number M0-M1, which can be zero) are in the Limbo, so the alterations must also consider the possibility to change the composition of the Limbo.*

The first perfect candidate found in this process is the required pairing. As soon as we find it, the process is finished. If there is no perfect candidate, we use the best available one. Since we are scrutinising all candidates, we can find the best one as we proceed. To do that, when we find the first legal (but not perfect) candidate, we mark it as "provisional-best". Each time we find another legal (but still not perfect) candidate, we compare it with the current provisional-best and keep the best of the two as the new provisional-best.

Two candidates are compared based on their compliance with all the pairing criteria, which are defined in order of priority in section 2. The first check is on the priority of the highest infringed criterion – the higher it is, the lower the quality of the candidate.

The second check is on a "failure value" (see Article 2.4) which is peculiar to that criterion. Often, this will be the number of times the criterion is infringed (e.g., the numbers of disregarded colour preferences) but it may also be of a different nature (e.g., the number of games played by the only downfloater from the bracket where the PAB assignee is determined).

Then we go to the second highest 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.

For example, a candidate that infringes criteria [C12] and [C13] is better than one that only infringes [C9]. Similarly, a candidate failing on [C9] and [C12] is worse than one failing on [C9] and [C13].

If the new candidate is better than the old provisional-best, we mark it as the new provisional-best, discarding the previous one – otherwise, we keep the old one. In the end, all candidates will have been examined – hence, the surviving provisional-best, although it may seem imperfect, is, in fact, the best possible candidate, so it is accepted as "the pairing", because of Article 3.8.

All detail about the iterative pairing process is given in Articles 3.6 and 3.7.

3.6 Alterations in Homogeneous Brackets or Remainders

3.6.1 Alter the order of the players in S2 with a transposition (see Article 4.2). If no more transpositions of S2 are available for the current S1, alter the original S1 and S2 (see Article 3.2) applying an exchange of resident players between S1 and S2 (see Article 4.3) and reordering the newly formed S1 and S2 according to Article 1.2.

Since we are now managing only homogeneous brackets, we do not need to worry about pairing MDPs. The possible actions 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 Article 4.2) but keeping them separate from the players of S1. This leads to a change of the second player in some pairs. The basic idea is to alter the pairing by modifying players' order in rankings as low as possible, thus minimising the disturbance in the upper pairs of the bracket (which are more important than the lower ones in the philosophy of the Dutch system).*
- *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 pairings as little as possible. We therefore 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, the swapped players have an approximately 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 it possible to pair players of the same original subgroup between them.*

After we made transpositions in a bracket, the alteration in order is desired – hence, players in the S2 subgroup <u>should not be sorted again</u> (S1 does not need to be sorted, as it has not been changed).

On the contrary, after each exchange, which swaps one or more players between subgroups S1 and S2, we must sort both subgroups S1 and S2 according to Article 1.2, to restore the correct order before beginning a new sequence of pairing attempts.

If the first pairing attempt after an exchange fails to give a valid result, we will try transpositions too, thus changing the natural order in the <u>modified</u> S2.

Both transpositions and exchanges should not be applied at random: section 4 dictates the precise sequence of possible transpositions and exchanges. This sequence begins with alterations that give mild disturbances to the pairing (with respect to the "natural" one), moving gradually towards changes that cause important effects.

The order of actions is as follows: first, we try, one by one, all the possible transpositions (see Article 4.2). If we find one that allows a perfect pairing, then the process is completed. Otherwise, we try the first exchange (see Article 4.3). With this exchange, we proceed trying every possible transposition, until we succeed or use them up. In this last 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 possible transpositions and exchanges, then a perfect pairing simply does not exist. In that case, we apply Article 3.8, thus accepting a less-than-perfect result.

HOW MANY PLAYERS CAN WE EXCHANGE?

Let us consider the example of a six players bracket {[1,2,3][4,5,6]}. In this bracket, let us exchange players 3 and 4 between S1 and S2, thus obtaining the new subgroups composition {[1,2,4][3,5,6]}. Please note that player 4, now in S1, has a higher BSN (i.e., a lower rank) than that of player 3, now in S2. This exchange is "useful" because it yields candidates that were not previously found (i.e., those that contain the pairs 1-3 or 2-3, and 4-5 or 4-6, which we did not find with the previous attempts). However, all candidates that pair together the two exchanged players are useless, because they were already checked (before the exchange, we tried all the possible transpositions in S2 – hence, we already tried all the candidates that contain the pair 3-4!) – e.g., exchanging (3,4) we obtain, among others, 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).

This is easily extended to brackets containing any number of players. Now, let us consider the exchange of two players per subgroup, e.g. {[1,4,5][2,3,6]}. Here, 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 there is a theoretical maximum to the number of players that can be usefully exchanged in the bracket, which is half the number of players in S2 (rounded down if needed).

In summary, we have the two following rules-of-thumb, which allow us to immediately discard many (in fact, most) candidates.

- ***Every time a pair contains a player from S1 with a higher BSN than their opponent from S2, this pair belongs to a candidate that has already been evaluated** (and can therefore be discarder right away).*
- ***Exchanging more than half of the original S2 players is useless**.*

3.7 Alterations in Heterogeneous Brackets

3.7.1 Operate on the remainder with the same rules used for homogeneous brackets (see Article 3.6).

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:

- *in the first step, we build an MDP-Pairing (see Article 3.3), which takes care of the pairable moved-down players (MDPs), giving rise 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.*

Of course, a practical implementation need not necessarily compose the pairing in two steps, as long as the final result is identical to the one yielded by the procedure described here (on this, see also the comment to Article 3.2.2).

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

3.7.2 If no more transpositions and exchanges are available for S1R and S2R, alter the order of the players in S2 with a transposition (see Article 4.2), forming a new MDP-Pairing and possibly a new remainder (to be processed as written in Article 3.4.1).

In a homogeneous bracket, this would be the moment when we lower our expectations, settling for a less-than-perfect pairing (see Article 3.6). In a heterogeneous bracket, however, we are not ready to surrender yet – 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 original subgroup S2 – i.e., 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 perfect pairing – and, if we have no success, we continue trying transposition after transposition, until we succeed or exhaust them all. (Actually, we do not need to try all the transpositions, because not all of them are meaningful. In fact, we only try those transpositions that 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 are just "spectators" – they do not take part in this phase of the pairing and are thus irrelevant, at least for the moment.)

3.7.3 If no more transpositions in S2 are available for the current S1, alter, if possible (i.e. if there is a Limbo), the original S1 and Limbo (see Article 3.2), selecting the next set of pairable MDPs according to Article 4.4.2 and restoring S2 to its original composition.

We got here because neither transpositions nor exchanges could bring us to a perfect pairing. After all the possible transpositions in the original S2 have been used up, if there is a Limbo, we still have a resource left – trying to change our choice of the MDP set (see Article 4.4). This means changing the MDPs to be paired and, thus, the original composition of S1.

To do that, we go to the next MDP set in the order established by Article 4.4.2, which minimises the scores of the downfloaters (and this in fact also maximises the scores of the paired MDPs).

After choosing a new MDP set, we are 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. We must try all the possible MDP sets one by one and in the correct sequence, until we find a perfect pairing or use up all sets. For each set, we shall try all the possible *transpositions in S2, thus generating a different remainder that will undergo all the usual pairing attempts as described above.*

3.8 Actions when no Perfect Candidate Exists

3.8.1 Choose the best available candidate. In order to do so, consider that a candidate is better than another if it better satisfies the PAB Criterion ([C5], see Article 2.3) or a quality criterion ([C6]-[C21], see Article 2.4) 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 Articles 3.6 or 3.7).

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 Article 3.5).

THE SIEVE PAIRING

An interesting alternative to the procedure described above – not always a practical one, but important from a theoretical point of view – is the one we shall call "Sieve pairing" (because of its similarity with the famous Eratosthenes' Sieve).

The basic idea is remarkably simple. We build all the possible legal pairings (i.e., all those that comply with the absolute criteria). Then we start applying all the pairing criteria, one by one, starting with the most important one and proceeding downwards. By applying each criterion, we can discard some of the pairings, so that, as we proceed, the surviving candidates becomes fewer and fewer. If, at some stage in the process, only one candidate remains, we choose that one – it may even be a lousy 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 3.

This method is sometimes a powerful help in pairing those small but ugly brackets we occasionally find in the late rounds of a tournament.

4. RULES FOR THE SEQUENTIAL GENERATION OF THE PAIRINGS

This section establishes rules to determine the sequence in which transpositions, exchanges, and MDP sets must be tried, to generate the candidates in the correct sequence. First, we should note that, in the FIDE (Dutch) system, any change of the order in S2 (transposition) is, by definition, preferable to even a single exchange of players between S1 and S2 (see Article 4.3.3). Then, we always move the player(s) whose displacement will cause the least possible difference of the candidate from the previous one (in the sense explained in the comments to Articles 3.5 and 3.6). The underlying idea is to operate by tiny steps, in order to obtain a pairing as similar as possible to the "natural" one, while at the same time allowing the best possible quality of the pairing itself.

4.1 In-Bracket Sequence-Number (BSN)

4.1.1 Before any transposition or exchange take place, all players in the bracket or in the remainder shall be tagged with consecutive BSNs representing their respective ranking order (according to Article 1.2) in the bracket (i.e. 1, 2, 3, 4, ...).

In this phase, the use of TPNs can be sometimes confusing. We will see presently that some rules are based on the numbers that represent the position of each player in the bracket. Those rules are meant for strictly sequential and contiguous numbers starting from 1 because we use the sum of such numbers to sort exchanges (see Article 4.3) – thus, in addition to their order, their actual numeric value is also important. The original TPNs are not necessarily in strict sequential order, so we give temporary sequence numbers ("BSN") to the players of the bracket or remainder, as a handy remedy to simplify the application of the following rules. Please note that the BSNs to be used for the pairing of the remainder in a heterogeneous bracket cannot be the same used for the MDP-pairing (they are not contiguous anymore). Thus, after the MDP-pairing, and before proceeding to the remainder, we must assign brand new BSNs to its players.

4.2 Transpositions in S2

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

4.2.2 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.

Note: *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. For example, 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 and compared based on the dictionary ("lexicographical") order, so that a 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. (To decide which string precedes the other one, we compare the first number of the first string with the first number of the second string. If either of the two is lower than the other one, the string it belongs to is the "smaller" one. When they are equal, we proceed to the second element of each string and repeat the comparison. Then, if needed, we go on to the third, the fourth, and so on – until we find the first difference. Please note that there must necessarily be a difference because the two lists contain the same elements taken in different order. For the same reason, the two strings being compared are always of the same length, because neither transpositions nor exchanges can change the number of players in the subgroup.

The subgroup S1 may or may not have the same number of players as S2. For a meaningful comparison, we must define the number of elements in each of the two strings of BSNs that we are comparing. We are looking for mates for *each element in S1 (each one of them represents a player). Thus, we only consider a number N1 of elements from S2, while the remaining ones are irrelevant, at least for the moment.*

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].

(By the way, in the simple case where every BSN is a single digit, the string can be interpreted as a number, which becomes larger and larger as we proceed with each new transposition: 234, 243, 324, 342, 423, 432.)

Since we want to pair #1 with the first element of S2, it is apparent at once that [2,3,4] and [2,4,3] have the very same effect (of course, this equivalence is in no way general – it depends only on the fact that we are looking for just one element!). 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 slashes "/…/" are irrelevant in this phase and we therefore ignore them for the time being): [2] /3, 4/; [3] /2, 4/; [4] /2, 3/.

4.3 Exchanges in Homogeneous Brackets or Remainders (original S1 ↔ original S2)

4.3.1 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 we must not change the size of S1 and S2.

To evaluate the "weight" of the change, we want to consider the choice of players as well as the size of the exchanged sets. To do that, we need a set of criteria addressing the different aspects of this choice. The aim is, as always, the "minimum disturbance" – i.e., to try and have a pairing as similar as possible to the natural one (again in the sense explained in the comments to Articles 3.5 and 3.6).

4.3.2 In order to sort all the possible resident-exchanges, apply the following "comparison rules between two resident-exchanges" 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:

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

In sorting exchanges, the first criterion is the number of involved players – the less, the better. In fact, what the rules say, here, is that exchanging only one player, however strong, is better than exchanging two players, even if they are very weak. (Some players would not accept this so easily.)

We also want to note that, for each player exchanged from S1 to S2, two routes are possible –the player must either be paired with a higher-ranked opponent, or float (see "How many players can we exchange?", page 38).

2. the smallest difference between "the sum of the BSN(s) moved from the original S2 to S1 and the sum of the BSN(s) moved from the original S1 to S2" (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).

In principle, all players in S1 should be stronger than any player in S2 (even though, in fact, this is not always true). Hence, when we swap two players across subgroups, we try to choose the weakest possible player in S1 and swap that player with the strongest possible one from S2. Thus, we 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, with the idea that a higher rank should indicate a stronger player. 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. (This constraint is equivalent to minimising the sum of the differences between exchanged BSNs. For a given set of players exchanged from S2, it is tantamount to minimising the sum of BSNs in S1. In the same way, for a given set of players exchanged from S1, it is the same as maximising the sum of BSNs from S2.)

3. the largest differing 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 in which the lowest-ranked player from ***S1*** *(i.e., the one with the highest BSN) has a lower rank.*

In the above example, {5-2} is better than {4-3} (in the logic of the FIDE Swiss (Dutch) system, of course) 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.

Note: Sometimes, just as in the example above, we may end up exchanging a higher-ranked player (i.e., a player with a lower BSN), as a side effect of enforcing the exchange of the lowest-ranked possible player. This choice may seem peculiar, but it is correct if we consider that an exchange does not operate on "several single players" but on a <u>complete set</u> of them. Hence, we must determine if a given <u>set</u> 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, because this is the way to exchange #4 rather than #3. (We should note, in passing, that player #1, when exchanged, will unavoidably float, because there is no possible higher-ranked opponent for them – see "How many players can we exchange?", page 38).

4. the smallest differing 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 minimised the disturbance in S1, we can optimise the disturbance in ***S2*** *too.*

Contrary to S1, here we try to exchange the lower possible BSNs (i.e., a player as high-ranked as possible). Hence, 6-9 is better than 7-8, because exchanging #6 is better than exchanging #7 – and so forth.

4.4 Set of pairable MDPs

4.4.1 A set of pairable MDP(s) is valid if it leaves a Limbo compliant with [C7] (see Article 2.4.2).

If a bracket is heterogeneous, before starting the pairing process we must prepare and sort the sets of pairable MDPs. Those sets will be used to populate the subgroup S1 (see Article 3.2.2), to pair them with resident players. Those MDPs that are not pairable in the current bracket go into the Limbo (whence they will eventually downfloat), so there is no point in putting any of them in a set of MDPs to be paired. This article defines which sets are acceptable, while the next one sets a rule to sort them, thus establishing the correct sequence in which they should be used.

An MDP set is valid if what players remain in the Limbo (and, thus, are going to downfloat) minimise the scores of downfloaters [C7] (and, of course, it must also comply with the Completion criterion [C4], i.e., it must allow the completion of the round-pairing). This means that, in some instances, the set could contain fewer MDPs than we could pair in the current bracket (e.g., to leave some of them available to accommodate players that otherwise could find no possible opponent in subsequent brackets – or, possibly, the player that must receive the PAB).

We obtain the absolute minimum scores when all downfloaters are residents. When there are MDPs among the downfloaters, those MDPs have, by definition, higher scores than any resident. To comply with [C7], we must minimise the score of the top-scored downfloater; then of the second-top; and so on.

Now, the number of downfloaters from the current bracket is determined by [C6] and, therefore, it is a fixed quantity. Part of those downfloaters can be MDPs (as we saw in the comment to [C7]). In order to comply with [C7], we must find (in fact, by gazing into a crystal orb…) not only the number of MDPs going to downfloat, but also their score.

It is worth mentioning that the use of the RSL (see the comment to [C7]) can facilitate this step. The RSL gives us the number and scores of the players in the Limbo, so no subsequent "resizing" of the MDP set is in order and no crystal orb is required. Besides, it also ensures compliance with [C7] by construction, i.e., score minimisation was already made.

4.4.2 Valid sets of pairable MDP(s) are sorted according to their smallest differing BSN.

Article 4.4.1 told us how to build the set of MDPs, choosing them so that it allows compliance with criterion [C7] – besides, of course, the Requirement Zero ([C4]). Now, it is time to choose, from all these valid sets, which one should be used for the candidate. This article provides the rule that let us decide which set comes first. An example is the best explanation.

Let us suppose that the current bracket contains players {1, 2, 3, 4} as MDPs and that the following MDP sets are valid (see Article 4.4.1).

{1, 3}; {1, 4}; {3, 4}; {1}; {3}; {4}.

It is understood that the larger is the number of MDPs in the set, the better is the set (provided we choose the highest available scores – any MDP not included in the set is bound to downfloat, thus causing the score of downfloaters to grow). Therefore, we ought to select among the sets D_1={1, 3}, D_2={1, 4}, and D_3={3, 4}. The rule tells us to compare the smallest differing BSNs – thus, we compare the first elements of each set, and this immediately excludes D_3, whose first element is different (in fact, larger) than those of D_1 and D_2 (so the first elements of D_1 and D_2 are the "smallest differing" BSNs). The latter two, however, have identical first elements, so the first differing element is the second one. Choosing the smaller one, we finally give precedence to D_1. In the end, we try D_1 first, then D_2 and then again D_3.

4.5 Next Element (follow-up of all Articles 4.2 to 4.4)

4.5.1 Any time an order has been established in accordance with Articles 4.2 to 4.4, any application of the corresponding article will pick the next element in the sorting order.

So, the basic configuration of S1 and S2 (sometimes, we informally call it the "transposition zero") did not work, and we had to change it. If we are lucky, the first attempt at a different transposition, exchange, or choice of an MDP set will yield the desired result. More often, we must persevere in our attempts until we find a successful one. To do that, we must follow the order (or sequence) established by the three rules illustrated above. 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 according to (respectively) Articles 4.2, 4.3, or 4.4 and then trying them one by one until we find the first useful one. In common practice, exchanges and transpositions will be tried together (usually, we try one or more transpositions for each exchange). To avoid mistakes, it is most advisable to annotate the last transformation we used (of each kind) so that, on the following attempt, we can be sure about which element of the sequence should be the next one.

5. COLOUR ALLOCATION RULES

5.1 The initial-colour is the colour determined by drawing of lots before the pairing of the first round.

The initial-colour does not refer to any particular player. In fact, it is a parameter of the tournament, and the only one left to fate! It allows the allocation of the correct colour to each player who still has no colour preference.

5.2 For each pair apply (with descending priority):

5.2.1 Grant both colour preferences.

5.2.2 Grant the stronger colour preference (see Article 1.7). If both are absolute (topscorers, see Article 1.8) grant the wider colour difference (see Article 1.6).

When two or more absolute colour preferences are involved, Article 5.2.2 also considers the colour differences of the players (see Article 1.6.1). Of course, such differences can happen only for topscorers, and therefore only in the last round (in previous rounds, a pairing with colliding absolute colour preferences is forbidden). Let us consider the example of two topscorers with absolute colour preferences and the following colour histories:

1: BBWBWB
2: WWBWBB

Here, player #1 has a colour difference C_D=-2, while player #2 has C_D=0.

Thus, we try to equalise the colour differences by assigning the preferred colour to player #1.

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: WBBWBWB (strong preference, C_D=-1)
2: uWWBWBB (absolute preference, C_D=0)

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

5.2.3 Alternate the colours to the most recent time in which one player had White and the other Black.

Note: Always consider Article 3.4 of the General Handling Rules for Swiss Tournaments.

To correctly manage colour assignments when one or both players have missed one or more games, we often need to compare colour histories by means of Article C.04.2:3.4. For example, in comparing the colour histories of two players, the sequence uuWB is equivalent to BWWB and WBWB – but take notice that the latter two are not equivalent to each other!

5.2.4 Grant the colour preference of the higher ranked player (see Article 1.2).

We want to pay particular attention to this point – all other conditions being equal, the higher-ranked player does not get White but their expected colour. (This is a moderately common misunderstanding among players.)

The higher-ranked player is selected as per Article 1.2, i.e., based primarily on higher score, and then on higher initial ranking (e.g., if #1 has 3 points, and #7 has 3.5 points, the latter has higher rank than the former).

5.2.5 If the higher ranked player has an odd TPN (see Article 1.1), give them the initial-colour; otherwise give them the opposite colour.

We get here when both players of the pair have no colour preference at all. Therefore, we allocate colours to the players based on the initial-colour, which was decided by drawing of lots before the start of the tournament.

Of course, this rule will always be used in the first round, obtaining the well-known results, but it will also be useful in subsequent rounds, when we have a pairing between two players who played no games yet (e.g., late entries or forfeits). Please note that, when using an accelerated pairing system, the usual colour alternation is disrupted, unless the number of players in the first scoregroup is a multiple of four.

We ought to remember that players who enter the tournament only at a later round (and, therefore, were not paired in the previous rounds), in fact, do not exist yet. An obvious side effect of this is that we cannot expect all "odd-numbered" and "even-numbered" players in subsequent rounds to have the same colours as usual (i.e., as they would have in a "perfect" tournament). In fact, late entries may have different effects on TPNs, depending on how they are managed.

If we insert all the players in the list straight from the beginning, TPNs will not change in 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 in the first round, players #13, #15, and so forth, who should seemingly get the initial-colour, will in fact have the opposite colour; while players #14, #16, and so on will get the initial-colour.

If, on the contrary, we insert late players only when they actually enter the tournament (as required by Article C.04.2:2.4), we must put them in their correct place in the players' list. All the subsequent players shift forward in the list to accommodate the new entries and, thus, have their TPNs changed. (For example, if a newly inserted player gets #12, then the previous #12, who had colour opposite to the initial-colour, will now be #13, and so on for all subsequent players.)