FIDE SPP Committee

Meeting in Istanbul 2012

Annex 1 to the Agenda

Question: Dear Mr. Gijssen. I am an arbiter who has the pleasure to know you because you were my Chief Arbiter at the Olympic Games in Turin in 2006. I'm particularly interested in the Regulations for the Swiss System Tournaments, especially in the Regulations for THE DUTCH SYSTEM and THE LIM SYSTEM, and my attention has been struck by what is shown on the Handbook at the section C04.5 General Handling Rules.
The Rule B (General rules for Swiss System for individual tournaments), that enumerates the minimum requirements which must bet by each swiss system to be regarded as a fair system, reports in paragraphs (d) and (e):

> ***d.*** *No player's colour difference will become >+2 or <-2, except for a player having a score of 50%* **or more** *in the last round, if this helps to avoid additional floaters.*
>
> ***e.*** *No player will receive the same colour three times in a row, except for a player having a score of 50%* **or more** *in the last round, if this helps to avoid additional floaters.*

On the same subject, however, the section C04.1 Swiss System Based on Rating (The Dutch System) reports in the Note of the Rule B.2 some Absolute Criteria:

> ***Note****: If it is helpful to reduce the number of floaters when pairing "top scorers" B2 may be ignored*

When the A.10 tells:

> ***A.10 Definitions: Top scorers, Backtracking***
> *Top scorers are players who have a* **score of over 50%** *of the maximum possible score when pairing the last round.*
> *Backtracking means …*

Therefore, on the final round, the Regulations The Dutch System allows the exemption to only players with a score greater than 50% of that possible, while for The General Handling Rules the exception also extends to the players with a score of 50% of that possible.
On the section C04.2 (Regulations for Swiss System Tournaments – The Lim System) the Rule 13 reports:

> ***13 Exceptions applicable to the last round***
> *In the last round, Rule 3, requiring players with the same score to be paired if they had not met in an earlier round, shall have priority over alternation and equalisation of colours, even if it is necessary for one of the players to be given the same colour for the third round in succession, or to be given three more of one colour than the other.*

Otherwise, according to The Lim System, against what is required by The General Handling Rules "would seem" that "ALL" players at the last round can have *the same colour for the third round in succession, or to be given three more of one colour than the other*.
Thank you in advance. Kind regards. **I.A. Manlio Simonini, (Roma, Italy).**

FIDE SPP Committee Meeting in Istanbul 2012
Annex 2 to the Agenda

by Roberto Ricca

Last year we did an extensive work in rewording the Dutch System . We had to solve some ambiguities that existed with the previous wording and I think we made a big step forward.

However the work is not finished yet. I am the first buffer to programmers that look for the endorsement of their software and I still get questioned by them about some points that are not completely clear in the current Dutch rules.

The problem presented here was discovered by Mr. Cervesato during the endorsement procedure of his software, *Javapairing*. It is something that we left in the new wording that is so blatantly wrong that we can safely call it a distraction.

In C.12, our attention was all for exactly clarifying what kind of backtracking we were allowing *(and still didn't finish but will discuss that later)* and forgot that we had changed the name of the variable maintaining the number of pairs requested in a whole score bracket. That number is **P1**, as specified in C.2 and C.14, but in C.12 **P** is wrongly used as we didn't correct the old wording.

For homogeneous brackets **P** and **P1** are the same value, but for heterogenous brackets, when getting in C.12, **P** just represents the number of moved-down players and pairing all moved-down players is not enough to prevent backtracking.

Let's look at the following example:

```
F (5.5): can only meet A
A (5.0): can meet F, B and C
B (5.0): can only meet A
C (5.0): can only meet A
```

F has an opponent but the pairing is not yet finished for this bracket! As two pairs were requested and they cannot be found in the bracket, we try to backtrack. Maybe we could move down from the previous bracket a player **G** that can play with either B or C...

Bottom line: in C.12, in place of

> ... and this now allows P pairings to be made ...

there should be written

> ... and this now allows P1 pairings to be made ...

C.12: what is <u>same size</u>?

Let's look at the article C.12:

> If there are moved down players: Backtrack to the previous score bracket. If in this previous score bracket a pairing can be made whereby **another set of players of the same size** will be moved down to the current one, and this now allows P1 pairings to be made then this pairing in the previous score bracket will be accepted.
> Backtracking is disallowed when already backtracking from a lower score bracket.

What is the exact meaning of *the same size*? It means, of course, that we are talking of the same number of moved-down players, but what about the **score** of such players? In looking for another set of downfloaters, can we increase their scores?

Here is an example of what I mean: let a tournament be in the situation below:

```
A (7.0): has already met B
B (6.5): has already met A and D
C (6.0):
D (5.5): has already met B
   The field is at 5.0 or below. Expected
   colours don't matter
```

Later we will discuss similar examples using colours, but let's ignore them for the moment. **A** floats in the bracket with **B** and both **A and B** float in the bracket with C. A-C, with B floating is the correct pairing, but now B-D cannot play.

What do we do now?

Basically there are two possibilities.

[1] backtrack to the {A, B, C} bracket, pair with B with C, let A float *(for the third time!)* and form the pair A-D

[2] affirm that B cannot be exchanged with A because it has less points than A and therefore let both B and D float in the group at 5.0.

If A and B cannot be exchanged, it will be easy to add in C.12 to another set of players of the same size (as emphasized above): **and with the same scores**.

If they can be exchanged, though, I think I should prepare myself to deal with questions related to (re)floaters like in the following example:

```
A (6.5): has already met B
B (6.5): has already met A
C (6.0): has already met F
D (6.0): has already met F
E (6.0): has already met F
F (5.5): has already met C, D and E
   The field is at 5.0 or below. Expected
   colours don't matter.
```

Let's do the pairing: A and B float in {C, D, E}. First try is A-C B-D and E floats; E-F have already played so we try to change the downfloater. Second try is A-C B-E and D floats: doesn't work. Third try is A-D B-E and C floats: doesn't work either. And now?

Nobody, I think, is going to make B float again just to pair F, isn't it? But, if scores of floaters can be increased, why a pairing like A-C E-D B-F is not allowed?

A.10: Top scorers

The definition of *top scorer* is found in A.10

> Top scorers are players who have a score of over 50% of the maximum possible score when pairing the last round

Although it is a definition that can still be perfected, nevertheless it is a clear definition.

In the note after B.2 there is written:

If it is helpful to reduce the number of floaters when pairing top scorers B2 may be ignored

The ambiguity is in the wording **top scorers**. Does this mean that **both** players in the possible pair should be *top scorers* or is it enough that just one of them be?

If the answer is **both**, there is no more problem (besides find a less ambiguous wording, but that should be easy).

If, however, the answer is **one is enough**, please be aware that it may arise a situation like this:

```
A (4.5): colour sequence:    BWBWBWBB
B (4.0): colour sequence:    BBWBWBWB
```

Last round, of course. For simplicity, let us hypothesize that either A and B meet or both float: who gets *white*?

A is a *top scorer*, B is not. We are dropping the B2 rule **only** for *top scorers*, that is only for A. B is still protected by the B.2 rule, so it cannot get *black* if his preference is absolute white. So *white* goes to B and A gets the third *black* in a row.

A7.a (absolute preference) and F.3

This comes from Mr. Mario Held, an Italian lecturer on Swiss Systems.

I am writing to you because I noticed a minor problem in the determination of an absolute colour preference for a player in Dutch Swiss Rules (C04.1), which states that:

A7.a: An absolute colour preference occurs when a player's colour difference is greater than +1 or less than -1, or **when a player has played with the same colour in the two latest rounds**. The preference is white when the colour difference is less than -1 or when the last two games were played with black. The preference is black when the colour difference is greater than +1, or when the last two games were played with white.

F.3: A player who after five round has a colour history of BWW-B (i.e. no valid game in round 4) will be treated as -BWWB with respect to E3. So WB-WB will count as -WBWB and BWW-B-W as --BWWBW.

Now, A7a refers explicitly to the last two rounds, unrelatedly to the presence of a given player, but then mentions the last two played games, and it seems not obvious that unplayed games should not be considered in the evaluation. Moreover, F.3, which instructs to skip unplayed games, clearly states that this reasoning applies only with respect to E.3 - and thus does not apply in other situations, e.g. in the determination of a colour preference.

Thus, for a player with a history BW-W, a strictly literal interpretation of the Rules leads to assign only a strong colour preference, whereas an absolute colour preference seems far more logical; hence, I'd like to suggest a minor modification in the rules, rephrasing the mentioned paragraphs as follows (proposed modifications are in bold):

A7.a: 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 he played**. The preference is white when the colour difference is less than -1 or when the last two games were played with black. The preference is black when the colour difference is greater than +1, or when the last two games were played with white.

F.3: A player who after five round has a colour history of BWW-B (i.e. no valid game in round 4) will be treated as -BWWB with respect to E3 **and A7**. So WB-WB will count as -WBWB and BWW-B-W as --BWWBW.

Moved-down players exchange

When all the moved-down players (which are M0) cannot find an opponent among the resident players of a bracket, C.14 is applied and M1, who at the start is equal to M0, is reduced. However it is nowhere specified in the rules how to choose these M1 moved-down players among the existing M0.
To be more precise: the first try is obvious (the first M1 moved-down players) but how and when to change them is unspecified. Any legal pairing will be accepted and this does not sound as the proper solution.

Let's look at this example:

```
 A (5.5): pref: w  can meet only C
 B (5.5): pref: b  can meet only C
 C (5.0): pref: w  can meet A and B
 D (5.0): pref: w  can meet only E
 E (5.0): pref: b  can meet only D
    The field is at 4.5 or below.
```

When we get to C.14 and reduce M1, we restart with {A} in S1 and {B, C, D, E} in S2. After a few transpositions, we get to S2={C, D, E, B} which does not work because B.4 is not fulfilled (X1 is 0). Next move, however, is to change the requirements (in our example, X1, applying 10.e, is incremented) not to exchange A with B, which would bring us to S1={B} S2={C, D, E, A}, which creates a pairing that better serves B.4.

Last year we made a change in the C.4 rule that helped to clarify a few things and perfect others. However that change creates problems in this specific situation, as the old C.4 rule could have solved this problem.

I think that we should just complete last year change adding what I call **The exchange rule for moved-down players**. It is a perfect solution, if we can define it properly.

Usually, in a heterogeneous group, when we are pairing the moved-down players, we shuffle between C.6 and C.7 and then go directly to C.10 (in order to change requirements).

If, between C.7 and C.10, we could add a step where we apply a sort of transposition to the M0 moved-down players in order to chose a different group of M1 of them, we would solve our problem.

There is an ideal place where to put such a rule: C.8. The first part will not change:

> In case of a homogeneous (remainder) group: apply a new exchange between S1 and S2 according to D2 and restart at C5

We could add a second part, which would deal with exchanges among moved-down players. Something like:

> In case of a heterogenous group: if M1 is less than M0, choose another set of M1 players to put in S1 according to D3 and restart at C.5

D.3 is not existing yet, but it should be easy to write it. Same composition rule as D.1 (order M0 players in lexicographic order), then put the first M1 of them in S1. With a caveat (if we wish): that the M1 players form a **set** and therefore "2 5 1 3" is equivalent to "1 2 3 5". In other words: use only the transpositions where the first M1 players are in increasing order.

D.3 is not the main point though, A wording will be found. It is the principle that we have to accept.

If we accept this, there might be also a corollary consequence: that the latter part of A.3

> A heterogeneous score bracket of which at least half of the players have come from a higher score bracket is also treated as though it was homogeneous.

could be dropped. Why would we still need such a clause? In other words: which situation may be covered that the new C.8 addition would leave open? Or vice versa? More on this later.

By the way, this is just food for thought, but the rules could also state something obvious, that when the moved-down players (M0) are more than the resident players (R), at least M0-R among the M0 moved down players will end up as downfloaters. This could help in a better definition of P0 (which at most is R) and M1 (same).
As a further subnote: P0 could also be better defined if we take into account that sometimes (when backtracking) we already know how many downfloaters the bracket is going to produce.

B.3 and derivates

Among our rules there is one that is very problematic to interpret: B.3. The current wording is:

> The difference of the scores of two players paired against each other should be as small as possible and ideally zero.

Sometimes it is very clear what that means, sometimes it is not.

B.3 versus A7.d

We have already seen an example where there are three players A, B, C, all with different scores, where A plays C despite both expecting the same colour because it is more important to keep differences at a minimum than to satisfy the colour preference.
In other words, we have already stated that B.3 is more important than B.4, which is fine because B.3 comes before B.4.

Now let's complicate matters a little bit:

```
A (7.0): pref: W    has already met B
B (6.5): pref: b    has already met A
C (6.0): pref:(W)
```

The field is at 5.5 or below.
W is an absolute preference for white. Regarding (W), see below

The round to pair is an odd round (the 9th, for instance). C didn't play one game and he got 4 blacks and 3 whites. So, accordingly to A7.d, players like him

> shall be treated like players having an absolute colour preference as long as this does not result in additional floaters.

Remember that the bracket [A, B, C] is formally homogeneous (as for the latter part of A.3). Can A play with C or that pair is prohibited by A7.d?

If we simply followed the rules, we should prohibit A-C because the pair B-C doesn't generate **additional** downfloaters. However it generates a downfloater that is ***bigger*** than the one that it could generate if we didn't apply A7.d.

The question here is whether the wording of A7.d is ok or we need to amend that rule in order to exclude ***bigger*** downfloaters?

In other words: what is more important? A7.d or B3?

If you answer A7.d, you can safely skip next paragraph.

B.3 versus B.2 (dropped)

If your answer to the question ending the previous paragraph was B.3, let me further complicate the picture. Now it is the last round and the colour preference of C for white is absolute:

```
A (7.0): pref: W    has already met B
B (6.5): pref: b    has already met A
C (6.0): pref: W
```

The field is at 5.5 or below.
W is an absolute preference for white.

As in the A7.d situation, according to the note after B.2:

> if it is helpful to reduce the number of floaters when pairing top scorers B2 may be ignored,

if we follow strictly the rules, the pair A-C is prohibited because ignoring B.2 will not reduce the **number** of floaters. However, if we ignore B.2, a ***smaller*** downfloater is generated (B will float instead of A) and it is a matter of opinions if this is better or not.

As in the previous paragraph, the question is whether the wording of the note after B.2 is ok or we should amend it in order to allow ***smaller*** downfloaters.

There also are other words to describe the above question: what is more important? B.3 or a dropped B.2?

Conclusion (of B3 matches)

Whatever decision we take is fine with me. I just want us to take a decision and put it in words.

Although B.3, as we have just seen, may be used in top brackets, it is more heavily involved when we are dealing with the lowest score bracket (LSB), particularly when, after applying C.13 more times, we can come up with brackets containing players with many different scores.

In 2009, C.13 was extended saying that after the two lowest brackets are merged:

> ... such a merged score bracket shall be treated as a heterogeneous score bracket with the latest added score bracket as S1.

As far as I know, this norm was introduced to avoid B.3 related problems. This goal is often reached, but it is not enough as (the latter part of) A.3 still exists and when the players coming from the penultimate score brackets aren't strictly less than the LSB players, the new LSB is still a homogeneous bracket where exchanges are possible between S1 and S2.

For instance, let's suppose that we get a LSB where we find the following players (note: it is an example, so please don't question now how we got to this LSB - we can discuss it later):

```
A (3.0):
B (2.5): has already met F
C (2.5): has already met F
D (2.0): has already met F
E (2.0): has already met F
F (1.5): has already met B, C, D and E
    Expected colours don't matter.
```

To pair it, we start with S1={A, B, C} and S2={D, E, F}, transpose until S2={F, D, E} and we find a pairing.
Have we finished? Well, let's compute the scores differences in A-F, B-D, C-E. They are 1.5, 0.5 and 0.5 respectively.
As said above, the bracket is homogeneous, therefore we can exchange C and D, restart with S1={A, B, D} and S2={C, E, F}, transpose until S2={F, C, E} and produce A-F B-C D-E where the scores differences are 1.5, 0 and 0, which are clearly better than the previous ones.

This is an example both simple and extreme. Simple because **just** six players were involved, And extreme, because it is a very evident that one pairing is better than the other one. During the various endorsement procedures, though, I had to deal with much more complicated case studies. LSB can become huge sometimes and then be paired in several different ways. The problem is: the rules don't clearly state which is the correct pairing.

What the programmers tell me (and I agree, because I am a programmer too) is that they need a clear direction of how to pair the LSB. Is B.3 involved? If so, how do we decide that a pairing is better than another one? Is there a rule (or a suggestion) telling us when to stop looking for a better pairing or do we need to always generate all the possible pairings in order to understand which is the better one?

I am ready to accept any definition of B3 (i.e. how to compute it), I just desire that one exists.

I can come up with a few proposals. The first one would be the perfect solution... ***if we were computers***. I will write it down, even though I very well understand that it could be hard for a human being to adopt it. The following two are less perfect, but more comprehensible.

[1] Compute B3 for pairing a single bracket in a mathematical (or geometric) way.

A pairing is a point P in the hyperspace. Its coordinates are the differences in scores in each pair. The perfection is the point O {0, 0, ..., 0}, i.e. the origin of the hyperspace. The closer a point P (a pairing) is to the origin, the better. The distance between P and O is given by the square root of the sum of the squares of P coordinates.

This will serve B3 beautifully. I agree that it can took a while to grasp it and some arbiter never will although it is a lot simpler than it looks. As a matter of fact, the lesser distance is often given by the lowest maximum coordinate (i.e. difference).
For instance: {1, 0} (max coord => 1) is worse than {0.5, 0.5} (max coord is 0.5); {2, 0, 0} (max coord => 2) is worse than {1, 1, 1} (max coord => 1).
Based on the last observation, here is a second proposal:

[2] For each player compute the difference in scores (in absolute value, may be zero) with the closest player he can meet. Then define D0 (for instance in A.6.d) as the maximum of these differences. Then modify B.3 saying that all differences in scores must be equal or less than D0 (if possible).
In the ***Pairing Procedures*** phase (Section C), put D1=D0 between C3.a and C3.b (or between C3.b and C3.c; or between C3.c and C3.d - *it depends on what we decide regarding B3 matches*). Then add a rule in C.10 (after C10.g; or between C3.f and C3.g; or between C3.e and C3.f; respectively) where D1 is incremented by half a point.
Is it the best solution? No, but it is a practical solution that is also easy to adopt for a human arbiter, who already checks all possible pairs in a bracket to see if a player is incompatible with the others.
It may also be a dangerous solution, because if the maximum of the minimums is high, all pairs in the bracket may be involved (although I don't think it would be a problem).
The biggest advantage of this solution is that it can be put in the ***Pairing Procedures*** and finally clarify when and how B3 is involved. With such a rule, no programmer will complain anymore.

However, if there still is the feeling that also this proposal is too complicated, there is a third solution (or maybe a **no-solution**):

[3] The *I-give-up* proposal: come on, it is the LSB! Do we really have to care so much? Just drop B3 in backtracking phases. We could end C.13 with:

> After two score brackets are merged, the B.3 criterion is not applied anymore in the lowest score bracket.

No more B3(s), no more problems!

==========

A final note regarding B.3. This criterion is the first one among the relative criteria. We tend, though, to forget it. In the article A.11, we give a quick overview of the goals of the Dutch System. In the middle of that article we find:

> The quality of the pairings is defined in descending priority as
> - the number of pairs
> - the number of pairs fulfilling the colour preference of both players (according to A7)
> - fulfilling the current criteria for downfloaters
> - fulfilling the current criteria for upfloaters

Do we notice that something is missing? Between *the number of pairs* and *the number of pairs fulfilling the colour preference* ... shouldn't we put something like:

> - the closeness of the scores of the players playing each other

Everything said until now shows that the above sentence will not be misplaced.

CLOCK SYSTEM

Presentation

The so-called "Clock" system is a pairing system that was the most used in Italy until the '90 (also called Italian-Swiss). It is still used in youth tournaments or when the ratings of all the participants are very similar.

The original system was basically a random pairing system. There is a logic behind the pairings, of course, but the luck of the draw was an integral part of the system. One of the modification of the original system tries to reduce the random factor.

The main advantage of this system is the simplicity of the pairings. In Italy, in the Seventies or Eighties, the majority of the players, put in front of the main grid of the tournament, could easily guess the correct pairings for the following round. It was that easy. Pairings could almost never be wrong because nearly everybody knew how to do them and any minimal mistake could be easily spotted.

This pairing system satisfies the rule C.04.5.B.h in a manner that probably no other system be able to do. Actually, there will be no need to explain the pairings, because everybody is going to understand them.

The original system is explained here with a few modifications needed to make the system more appealable to the Swiss Rules: players will be not randomly sorted as they were before (integral draw) and some color-dependent restrictions will be introduced in order to set the pairings.

Pairing numbers

Before the start of the tournament, each player is assigned with a pairing number. How such number is assigned is described in Appendix A.

Odd number of players

If in any given round the number of participant players is odd, a fictitious player (bye) will be added with no points and 0 (zero) as its pairing number. The bye will always lose. Playing against the bye will net the same points given to a win, unless stated differently by the tournament rules.

Absolute color preference

A player who has played twice in a row with the same color or who has played with a color two more times than with the other color, has an absolute color preference for the other color.

Compatibility

Two players are said to be compatible if they have not played yet or if they don't have the same absolute color preference (unless it is the last round and both of them have more than 50% of the maximum possible points).
The fictitious player (bye) is also not compatible with each player that got any kind of points (i.e. more than zero) without actually playing

Player disposition

All the players are ideally placed around a circular table (clock), following their pairing numbers in such a way that the last player is followed by the bye, if it exists, and then by the first one.

Pivot

Pivot is a concept defined by its use, as shown below.
For each round, the initial pivot value is zero.

Picker, picked

Each pair is composed by a picker and a picked.

The picker is the player with the highest score among the yet unpaired players and with the first pairing number that follows (i.e. in the clock) the pivot.
After establishing the picker, the pivot position is set equal to the picker.

Candidate

The candidate (for being picked) is computed similarly as the picker, i.e. is the player with the highest score among the unpaired players and with the pairing number that follows as close as possible the pivot.
The pivot is set equal to the candidate.

If the candidate is compatible with the picker, the pair <picker, candidate> is formed and stored (see Appendix B for color assignment). Then a search for a new pair will begin.
A candidate who is not compatible with the picker is discarded and a new search for a candidate will begin.

Backtracking

If there is no compatible candidate for the picker, the last stored pair is undone, its picker becomes the new picker, the pivot is set to the picked player of the pair, and such player is declared temporarily incompatible with the new picker (this temporary incompatibility will end when a superior pair is undone); the search for a new candidate will restart.
The process can be repeated until a full pairing is finally found or all the compatible opponents for the picker (including the bye if available) are declared temporary incompatible. In the latter case, the process will continue backtracking further to undo a previous pair.

APPENDIX A - PAIRING NUMBERS

Initial setup

[a] The players are ranked in order of rating, FIDE title, alphabetically. Each player gets a ranking number from #1 to #N.

[b] If there is an odd number of players, the last one will get the last pairing number (PN-N) and is excluded from further considerations

[c] the slot is a container of players (also: it contains one macroplayer); the size of the slot is the number of players who are in it; the slot is named after the highest ranked player it contains; the initial size of the slot is 1

[d] there is a draw to define the master-color (black or white), which is also the color #1 will get in the first round

Recursive items

[1] there are 2S slots to fill

[2] S1 is the set that include the first S (macro)players; S2 is the set that include the last S (macro)players

[3] the slots are combined two-by-two in such a way that the I-th slot contains the I-th (macro)players from both S1 and S2; if the S1 part of the slot will get the white color in the first round, the other one will get black; if the S1 part gets black, the other one will get white. The new slot is named after the I-th (macro)player of S1

[4] There are S new slots, the size of which is twice the size of the old ones

[5] If S is an odd number, the last (S-th) slot (odd-slot) is filled with the median (macro)player, which is then excluded from further considerations; odd-slots will get alternatively the master-color and the other color in order of creation/placement

[6] If the number of new slots (which is an even number) is bigger than two, move back to the recursive item [1]

[7] otherwise, there are two slots; the first one will get the master-color, the last one the other color. The placement process terminates and each player can be put in the proper slot, getting white or black following the rules shown in [3]

Example (with 27 players)

[a] 27 players participate in the tournament; they are ranked from #1 to #27

[b] The last ranked player (#27) is excluded from the grid (the PN-27 is assigned to him)

[d] a draw defines the master color: black

[1] there are 26 slots

[2] S1 contains the macroplayers from 1 to 13, S2 the macroplayers from 14 to 26

[3/4] 13 macroplayers are created of size 2; each macroplayer is formed by the player X and the player X+13, i.e. these macroplayers are **1**-14, **2**-15, **3**-16, **4**-17, **5**-18, **6**-19, **7**-20, **8**-21, **9**-22, **10**-23, **11**-24, **12**-25, **13**-26; in these macroplayers if the first element gets white, the second one will get black and vice versa.

[5] 13 is an odd number, so the last slot (which is PN25-PN26, as the size is two) is assigned to the median macroplayer, i.e. 7-20. 7 will get black (the master color), 20 white

[6/1] There are now 12 macroplayers (of size 2)

[2] S1 contains the macroplayers from 1 to 6; S2 from 8 to 13

[3/4] 6 macroplayers of size 4 are created; in each one the macroplayer Y is *paired* with the macroplayer Y+7 (as #7 was excluded). As always, in these macroplayers, if the first element gets white, the second one will get black and vice versa. The new macroplayers are: **1**-14-8-21, **2**-15-9-22, **3**-16-10-23, **4**-17-11-24, **5**-18-12-25, **6**-19-13-26

[5/1] The number of macroplayers (of size 4) is even,

[2] S1 contains the macroplayers from 1 to 3; S2 from 4 to 6

[3/4] 3 macroplayers of size 8 are created. In them, the macroplayer Z is paired with Z+3. Colors will be assigned in the usual way. The new macroplayers are: **1**-14-8-21-4-17-11-24, **2**-15-9-22-5-18-12-25, **3**-16-10-23-6-19-13-26

[5] 3 is an odd number, so the last slot (PN17-PN24) is assigned to the median macroplayer (which is the second one, i.e. 2-15-9-22-5-18-12-25). This macroplayer gets white (as the previous odd player got black), which means that #2 gets white, and #15, #9, #5 (the opponents of #2 in the various steps) will get black; the other colors are inferred from the above ones: #22, #18 and #12 will get white, #25 will get black

[6/7] There are just 2 macroplayers. The first one (#1) is put in the first slot (PN1-PN8) and gets black (the master color), the second one (#3) is put in the second slot (PN9-PN16) and gets white. Now the colors for all the players can be inferred

The complete grid is the following one:

Pairing#	Rank#
PN-1	#1
PN-2	#14
PN-3	#8
PN-4	#21
PN-5	#4
PN-6	#17
PN-7	#11
PN-8	#24
PN-9	#3
PN-10	#16

Pairing#	Rank#
PN-11	#10
PN-12	#23
PN-13	#6
PN-14	#19
PN-15	#13
PN-16	#26
PN-17	#2
PN-18	#15
PN-19	#9
PN-20	#22

Pairing#	Rank#
PN-21	#5
PN-22	#18
PN-23	#12
PN-24	#25
PN-25	#7
PN-26	#20
PN-27	#27
PN-0	BYE

The first round is:

(2-1)	#14-#1	**(12-11)**	#23-#10	**(22-21)**	#18-#5
(3-4)	#8-#21	**(14-13)**	#19-#6	**(23-24)**	#12-#25
(5-6)	#4-#17	**(15-16)**	#13-#26	**(26-25)**	#20-#7
(8-7)	#24-#11	**(17-18)**	#2-#15	**(27)**	#27 bye
(9-10)	#3-#16	**(20-19)**	#22-#9		

APPENDIX B - COLOR ASSIGNMENT

No color is assigned to whoever plays with the fictitious player (bye). Also no color is assigned when a scheduled game is not actually played because one or both players did not show up.

For the first round, the colors are defined by the grid builder (see Appendix A).

In following rounds, when just a player has an absolute preference for a color, he will get that color. Otherwise white is assigned to the player who has the biggest difference between games played with black and games played with white. If the players have the same difference, their color history is compared (unplayed games are discarded and put at the beginning of the list) and white is assigned to who played with black in the first round (going backwards) where their colors were different.

If the players have the same color history, white is assigned to the player with less points. If they have the same score, the player with the lower pairing number gets the same color he got in the last played round.