WEIGHTS IN McMAHON PAIRING

Geoff Kaniuk *geoff@kaniuk.co.uk*

January 2012

Contents

1 BACKGROUND AND PURPOSE **3**

2 McMAHON RULES **3**

2.1 The scoring process . . . 3

2.2 The core rules . . . 3

2.3 Uneven games . . . 4

2.4 Even game seeding . . . 4

2.5 Area . . . 5

3 ADDITIONAL RULES **5**

3.1 Banding . . . 5

3.2 Special cases . . . 5

3.3 Practical pairing . . . 6

4 WEIGHTS IN *MacMahon* **7**

4.1 Cost components . . . 7

4.2 Cost model . . . 7

4.3 Transformation to weight model . . . 8

5 WEIGHTS IN *Gotha* **10**

5.1 Weight components . . . 10

5.2 Weight model . . . 11

6 GENERALISED WEIGHT MODEL **13**

6.1 function forms . . . 13

6.2 repeat . . . 14

6.3 mms . . . 14

6.4 uneven . . . 15

6.5 even seeding . . . 18

6.6 colour . . . 18

6.7 area . . . 19

7 WEIGHTS FOR ADDITIONAL RULES **20**
7.1 banding . . . 20
7.2 play-in . . . 21
7.3 play-out . . . 21

8 GRADE DEPENDENT PARAMETERS **22**
8.1 scoring components . . . 22
8.2 colour . . . 22
8.3 area . . . 23

9 MAXIMUM WEIGHTED PAIRING **23**
9.1 Brief description . . . 23
9.2 Edge removal . . . 23

10 PAIRING ALGORITHM OPTIMISATION **24**
10.1 Introduction . . . 24
10.2 Pairing quality . . . 24
10.3 Tournament simulation . . . 25

A INDEX OF NOTATION **28**

B MAXIMUM CARDINALITY MATCHING **29**
B.1 Example . . . 29
B.2 Weight of an augmenting path . . . 29
B.3 Condition for increase in augmented path weight . . . 30
B.4 Theorem . . . 31

1 BACKGROUND AND PURPOSE

A maximum weighted matching algorithm has been used for many years in the pairing programs managing McMahon tournaments in Europe. The earliest program *MacMahon* was produced by Christoph Gerlach [1] in 1994. His method for allocating weights to each possible pairing between players is relatively straightforward and the weight model is governed by just a handful of prescribed constants.

The next program known to use a matching algorithm is *Gotha*, produced by Luc Vannier [2] in 2004. His model for allocating weights is more sophisticated, and the set of constants specifying the model is correspondingly larger.

One of the earliest successful pairing programs, *GoDraw* produced by the author [3] in 1990, uses a local search heuristic method for achieving the pairing, rather than a maximum weighted matching algorithm. However, it is recognised that weighted matching has much to offer in terms of flexibility, and plans are now afoot to develop the weighted matching pairing technique for *GoDraw*.

The purpose of this document is to generalise the weight models used in *MacMahon* and *Gotha* so that the various pairing algorithms using a maximum weighted matching algorithm have a common way to optimise the parameters in the weight models. The material presented here also functions as a design document for the next version of *GoDraw*.

2 McMAHON RULES

The scoring system in a McMahon [4] tournament has these characteristics:

2.1 The scoring process

1. Each player is given an initial McMahon score related to the player's grade.
2. A player's McMahon score m_i increases by 1 for a win and does not change for a loss.
3. At the end of the tournament, the player with the highest McMahon score is the winner.
4. All players above a certain grade β (called the bar) are given the same McMahon score. A player i is *above* the bar if the player's grade g_i satisfies $g_i \geq \beta$.

2.2 The core rules

The core set of rules governing the choice of opponents is:

1. Players do not meet more than once.
2. Pair players on the same McMahon score, so that they play opponents of equal strength.
3. If it is necessary to pair players on different McMahon scores, then minimise the McMahon score difference.
4. If both players are below the bar then pair players from different *areas* (country, club).
5. For each player, half the games should be played as black, half as white.

These core rules leave two important issues unspecified, namely:

1. Who gets chosen to play an uneven game when required, for example when a McMahon group is odd.
2. How to pair players within a McMahon group i.e. at random or seeded in some way.

Naturally enough, different philosophies have evolved to cope with these issues.

2.3 Uneven games

If a player in a McMahon group with an odd number of players is paired to play an opponent in a lower group then the game is easier than it should be, and the player is expected to win. To restore the balance, the player will play up to a higher McMahon group if the opportunity arises in a future round. A variation developed by the British Go Association reduces the number of uneven games by leaving the balance uncorrected if the player achieves the unexpected.

2.4 Even game seeding

One procedure for pairing players within a McMahon group of size S is simply to pair them at random. Another is to order players by strength (often by SOS). We define a seed s_i as the relative position of the player in the ordering. Some common pairing methods based on seeding are:

split&match The top half of the group is paired in order with the bottom half of the group.

split&random The top half of the group is paired randomly with the bottom half of the group.

split&fold The top half of the group is paired with the bottom half of the group in reverse order.

Seeding is also used in *MacMahon* and *Gotha* to govern pairing for uneven games.

2.5 Area

Below the bar, players from the same area should not meet. So if there is one large area and a number of small ones then there will be a lot of unhappy players if small areas play against other small areas, leaving the large area to play itself. In *GoDraw* a McMahon group is sorted in order of area size. A packing algorithm along the lines of *split&match* is used to ensure pairing between players from different areas.

3 ADDITIONAL RULES

A number of additional rules have evolved to deal with very large tournaments, or special circumstances like not pairing family members playing in different clubs for example. These rules are common enough to warrant inclusion in the weight model.

3.1 Banding

There are a number of tournaments in Europe and in the USA where players are organised in groups defined by a range of grades. This can be achieved by allocating the same initial McMahon score to all players in the group. Extra rules will be needed when it is necessary to pair players from different bands, because now there could be a wide disparity in player strengths chosen from McMahon groups in neighbouring bands .

3.2 Special cases

It does happen that members of the same family attend a tournament and may have similar grades, so there is a chance that at some point they have equal McMahon scores and could be paired by the program. This should be discouraged even if they are from different clubs. The *GoDraw* program has a feature for allocating specified players to named 'groups' so that players in the group *must not* play each other. Naturally the opposite grouping, where players in a group *must play* each other is also sometimes useful.

The above groupings result in deterministic pairings, but for completion there are two other conceivable groupings which provide non-deterministic pairings and are occasionally useful. Players in a group denoted *play-in* prefer to play others in the group if possible. Those in a designated *play-out* group prefer to play others not in the group. For example, in large cities, players may support several local clubs so, on travelling to an 'away' tournament they really should not play each other if possible.

3.3 Practical pairing

Any pairing algorithm for McMahon tournaments has to be able to cope with a number of possible extreme cases. One further useful rule which may be needed to deal with oddities is Matthew Macfadyen's *minimise-the-whingeing* rule. One application of this is that younger double figure kyu players strongly prefer not to play others they meet regularly, and so would rather have a higher handicap game than a same-area even game.

4 WEIGHTS IN *MacMahon*

The *MacMahon* model minimises the total *cost* of the matching, rather than maximising the total weight. The cost C_{ij} of a pairing between two players i and j is the sum of eight different components of cost [5].

The following briefly describes each of the eight components and introduces the conditions used to build the cost model.

4.1 Cost components

repeat Non-repeat pairings, expressed by $R_{ij} = 0$ have zero cost, but repeat pairings expressed by $R_{ij} = 1$, attract the maximum possible cost.

mms For any two players i and j, the cost increases quadratically with the difference $m_i - m_j$ in player's McMahon scores.

even Within a McMahon group, and when either of the players is above the bar, the cost of a pairing increases linearly with the deviation from the *split&fold* pairing rule. The seed s_i is taken to be the relative position of the player in the ordering, expressed as a percent of the number of players S in the group. So the weakest at seed 0 plays the strongest at seed 100.

uneven In games between players in different McMahon groups, the cost increases linearly with deviation from a *split&match* like rule. The weakest in the lower group plays the strongest in the upper group.

handicap Handicap games are allowed where *both* players are below the bar. However very large handicap games attract an additional cost if the McMahon score difference exceeds 9 stones, which condition we denote by H_{ij}.

colour For even games below the bar, the colour cost increases linearly with the *colour balance* of each player. The colour balance is defined as the quantity $Y_i = 1 - \min\{B_i, W_i\}/\max\{B_i, W_i\}$ where player i has played B_i games as Black, and W_i games as White.

area Below the bar, a pairing attracts an additional cost if the players belong to the same country or the same club.

4.2 Cost model

The total cost for the pairing between two players can be expressed as a weighted sum of functions

$$C_{ij} = \sum_{r=1\cdots 8} c_{ij}^r C_r$$

The cost coefficients C_r are constants for each of the components: *repeat, mms, even, uneven, handicap, colour, area-club,area-country.* In the following table we give the form of the weight function and the default value of the coefficients used in *MacMahon*. The definition of all symbols used in the formulae can be found in the Index Of Notation in Appendix A.

COMPONENT	FORMULA	WEIGHT
repeat:	$c_{ij}^{repeat} = R_{ij}$	$C_{repeat} = 268435455$
mms:	$c_{ij}^{mms} = \frac{1}{4}(m_i - m_j)^2$	$C_{mms} = 1000$
even:	$c_{ij}^{even} = \underline{B}_{ij} E_{ij}(\|s_i + s_j - 100\|)$	$C_{even} = 10$
uneven§ :	$c_{ij}^{uneven} = U_{ij}(100 - s_i + s_j)$	$C_{uneven} = 100$
handicap:	$c_{ij}^{handicap} = \overline{B}_{ij}\ H_{ij}$	$C_{handicap} = 80$
area-club:	$c_{ij}^{club} = \overline{B}_{ij}\ K_{ij}$	$C_{club} = 80$
area-country:	$c_{ij}^{country} = \overline{B}_{ij}\ L_{ij}$	$C_{country} = 20$
colour:	$c_{ij}^{colour} = \overline{B}_{ij}\ E_{ij}(Y_i + Y_j)$	$C_{colour} = 50$

§ It is assumed that $m_i \geq m_j$

4.3 Transformation to weight model

In order to compare the *MacMahon* model with the *Gotha* model, and to prepare it for use in the Gabow algorithm, we transform the cost model to a standard weight model form. Each of the component cost functions c_{ij}^r identified above has a maximum value for all pairs denoted by $\hat{c}^r$. Hence the component *weight* function defined by $\hat{w}_{ij}^r = \hat{c}^r - c_{ij}^r$ is maximum when the cost component function is minimum. Moreover, in order to more easily see how the various components rank in importance, we introduce a normalised version of the weight function defined for each component r by:

$$w_{ij}^r = \hat{w}_{ij}^r / \hat{c}^r = 1 - c_{ij}^r / \hat{c}^r$$

Then w_{ij}^r lies in the range $[0, 1]$, and the total cost of a pairing for players i and j is:

$$C_{ij} = \sum_{r=1\cdots 8} \hat{c}^r (1 - w_{ij}^r) C_r$$

Now define:

$$W_r = \hat{c}^r C_r$$
$$W = \sum_{r=1\cdots 8} W_r$$
$$W_{ij} = \sum_{r=1\cdots 8} w_{ij}^r W_r$$

Then the pairing cost is:

$$C_{ij} = W - W_{ij}.$$

Since w_{ij}^r has a maximum value of 1, the maximum value of W_{ij} is W, and it then follows that $C_{ij} = 0$ when W_{ij} is maximum. Hence the total pairing cost $\sum C_{ij}$ is minimised when the total weight $\sum W_{ij}$ is maximised.

5 WEIGHTS IN *Gotha*

The *Gotha* model maximises the total *weight* of the matching. The weight W_{ij} of a pairing between two players i and j is the sum of nine different components of weight [6]. These are briefly described in the next section along with a specification of conditions needed to build the weight model. Some of these conditions are the same as in the *MacMahon* model and we then use the notation developed there.

In many components, the weight is expressed as a continuous function of a *rule deviation* δ^r_{ij}.

5.1 Weight components

repeat Non-repeat pairings attract the largest possible weight. Repeat pairings are given the lowest possible weight of 1 (effectively zero if we were to eliminate such edges).

random This optional component adds random noise to the weight of each pairing. The noise value ρ can be drawn from a uniform distribution in the range $[0, 1]$, or calculated as a hash of the names of the players in the pairings.

colour A player's colour balance is defined as excess of games played white over games played black: $C_i = W_i - B_i$. The colour weight function depends on two mutually exclusive balance conditions C^{opp}_{ij} and C^{mixed}_{ij} and is applied only if the game is even.

bands Players grouped in a band b_i prefer to play each other. Banded weights are always applied if the number of bands B exceeds 1. In this case a banding rule deviation is defined by: $\delta^{bands}_{ij} = |b_i - b_j|/B$.

mms McMahon score difference is minimised for all pairs. The *mms* rule deviation is $\delta^{mms}_{ij} = |m_i \ - m_j \ |/M$, where M is the total number of McMahon groups.

uneven The weight function depends on the players' pairing history e.g. number of games played up (U_i) and number played down (D_i), and is applied only if the pairing is uneven. It minimises 'up' games and 'down' games separately. For a given player, it balances a draw in one direction with a subsequent draw in the opposite direction if the opportunity arises. The weight also depends on player seeds within each of the groups containing players i and j. The pairing rules include:

- Pair the strongest in the higher group with the weakest in the lower group as in *MacMahon*.

- Pair the weakest in the higher group with the strongest in the lower group, in keeping with the McMahon spirit of pairing players as equally as possible.

- Pair middle players from the higher group with middle players from the lower group.

seed *Gotha* applies one of 3 matching rules for players within the same McMahon group: *split&random*, *split&match*, or *split&fold*. The seed is the relative position in an ordering by player strength expressed as a fraction of the group size S.

> ***split&random*** The weight function is the discrete condition T_{ij}, that the players lie in different halves of the group.
>
> ***split&match*** The rule deviation is $\delta_{ij}^{match} = (|s_i - s_j| - S)/S$.
>
> ***split&fold*** The rule deviation is $\delta_{ij}^{fold} = ((s_i + s_j) - (S-1))/(S-1)$.

handicap The weights for handicap pairing are designed to minimise any required handicap. The weight function is determined by a scaled McMahon score (denoted by $\hat{m}_i$) and is applied only if selectable conditions depending on grade or number of wins are valid for the players. Where a handicap is applied, a rule deviation is defined in terms of the scaled McMahon scores by: $\delta_{ij}^{handicap} = |\hat{m}_i - \hat{m}_j|/\hat{m}_{range}$.

area Players below the bar are segregated by country and club if selectable conditions depending on grade or number of wins are valid for the pair. *Gotha* recognises (as does *GoDraw*) that below a certain level, players rather have a higher handicap game than a same club or country game.

5.2 Weight model

The total weight for the pairing between two players is expressed as a weighted sum of functions

$$W_{ij} = \sum_{r=1\cdots 9} w_{ij}^r W_r$$

The weight coefficients W_r are constants for each of the 9 components: *repeat, random, colour, bands, mms, uneven. seed, handicap, area.*

In order to avoid the problem of the degeneracy caused by weights dependent on a linear rule deviation, *Gotha* introduces a *concavity* function:

$$\Upsilon(x) = (1-x)(1+kx)$$

defined for x in the range $[0, 1]$ and some fixed parameter $k > 0$ having the default value $\frac{1}{2}$. Within the specified range, $\Upsilon(x)$ decreases from 1 at $x = 0$ to 0 at $x = 1$.

In the following table we give the form of the weight function, and the default value of the coefficients used in *Gotha*.

COMPONENT	FORMULA	WEIGHT
repeat:	$w_{ij}^{repeat} = 1 - R_{ij}$	$W_{repeat} = 5 \times 10^{14}$
random:	$w_{ij}^{random} = \rho$	$W_{random} = 10^9$
colour:	$w_{ij}^{colour} = E_{ij}\,(C_{ij}^{opp} + \frac{1}{2}C_{ij}^{mixed})$	$W_{colour} = 10^6$
bands:	$w_{ij}^{bands} = \Upsilon(\delta_{ij}^{bands})$	$W_{bands} = 2 \times 10^{13}$
mms:	$w_{ij}^{mms} = \Upsilon(\delta_{ij}^{mms})$	$W_{mms} = 10^{11}$
uneven:	$w_{ij}^{uneven} = U_{ij}(f^s(s_i, s_j) + f^h(i, j))$	
	$f^s(s_i, s_j) = f_{lo}^s(s_i) + f_{hi}^s(s_j)$	
	$lo\|hi = \{bot, mid, top\}$	
	$f_{bot}^s(s) = s/2S$ $f_{mid}^s(s) = (S - 1 - \|2s - (S - 1)\|)/2S$ $f_{top}^s(s) = (S - 1 - s)/2S$	$W_{uneven} = 10^8$
	$f^h(i, j) = f^h(U_i, D_i, U_j, D_j)$	
seed:	$w_{ij}^{split\&random} = T_{ij}\,\rho$	
	$w_{ij}^{split\&fold} = (1 - (\delta_{ij}^{fold})^2)$	$W_{seed} = 10^6$
	$w_{ij}^{split\&match} = (1 - (\delta_{ij}^{match})^2)$	
handicap:	$w_{ij}^{handicap} = \Upsilon(\delta_{ij}^{handicap})$	$W_{handicap} = 2 \times 10^{13}$
area:	$w_{ij}^{area} = \overline{B}_{ij}\ f^{area}(L_{ij}, K_{ij})$	$W_{area} = 10^{11}$

6 GENERALISED WEIGHT MODEL

6.1 function forms

The core rules for McMahon tournaments were specified in Section 2.2. The *MacMahon* and *Gotha* components implementing these core rules are *repeat*, *mms*, *colour* and *area*.

In *Gotha*, and in the transformed *MacMahon* model, all component weight functions w^r_{ij} take values in the range $[0, 1]$. In many components the weight function is a quadratic dependent on a continuous *rule deviation*, which in *Gotha* is usually scaled to take values also in the range $[0, 1]$. This scaling is necessary in order to prevent the concavity weight function Υ from going negative. It can be convenient to work with unscaled hyperbolic weight functions such as:

$$\begin{aligned} \operatorname{sech}(x) &= \frac{2}{e^x + e^{-x}} \\ h(x) &= \frac{1}{2}(1 - \tanh(\mu(x - a)) \operatorname{sech}(\nu x) \end{aligned}$$

The second form is a skew function, and for suitable choices of the parameters μ, ν, a, $h(x)$ approximates quadratic forms very well. It can easily be adjusted to model a step function or a bell shape if this is needed. The graph below compares *Gotha*'s concavity function with the other hyperbolic functions mentioned.

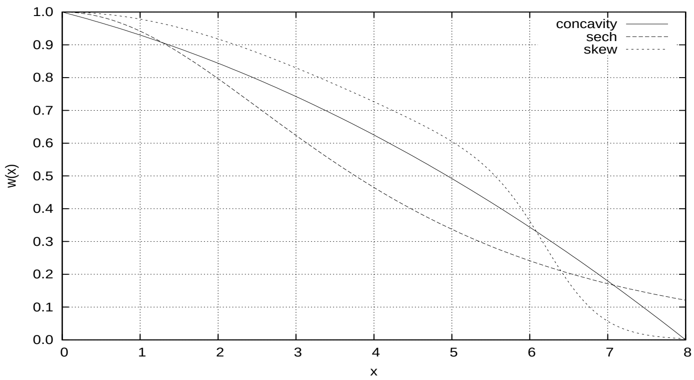

Figure 1: Comparison of weight functions.

Whatever weight function is used, the main requirement is that its value lies in the range $[0, 1]$, and it decreases steadily from a maximum value of 1 at zero rule deviation to a minimum value of zero at the maximum rule deviation.

In most cases we can model the core rules in *MacMahon* and *Gotha* in a similar way using hyperbolic weight functions.

6.2 repeat

In both *MacMahon* and *Gotha* repeat games are given a very nearly zero weight. This leaves open the possibility, that in order to complete a pairing, the program *may* need to include a repeat game. The motivation for this is the *guarantee* that the program pairs *all* players.

However, if the tournament entry is either so low, or so pathological that a pairing can be completed only by including repeat games, then it may be better to just report the situation and allow the draw-master to influence the pairing manually.

Allowing a repeat pairing is liable to generate complaints, and this violates the principle of 'minimise the whingeing'. We can completely avoid repeat games by ensuring that such potential pairings are removed from the list of edges that needs to be supplied to the weighted matching algorithm. It would be possible to include an initial pass in the pairing process which solves for a maximal cardinality matching, and reports back if the pairing is not complete.

6.3 mms

We satisfy the requirement to pair players with the same McMahon score, as in *MacMahon* and *Gotha* via a rule deviation without scaling and a simple hyperbolic weight function:

$$\delta_{ij}^{mms} = m_i - m_j$$
$$w_{ij}^{mms} = \text{sech}(\alpha \delta_{ij}^{mms})$$

It is interesting to compare the performance of this in dealing with the problem of linear rule deviations discussed in [6]. The table below shows weight results for two matchings: $M_1 = \{(2k, 2k), (3k, 3k), (4k, 4k), (1k, 5k)\}$ and $M_2 = \{(1k, 2k), (2k, 3k), (3k, 4k), (4k, 5k)\}$. We examine three different weight functions:

$$w_{linear}(d) = 1 - d$$
$$w_{concavity}(d) = (1 - d)(1 + \frac{d}{2})$$
$$w_{hyperbolic}(\delta) = \text{sech}(0.4\delta)$$

The unscaled rule deviation is $\delta_{ij} = |g_i - g_j|$, where g_i is the grade of player i. The scaled rule deviation for *linear* and *concavity*, is $d = \delta_{ij}/5$, since there are 5 McMahon groups.

MATCHING	w_{linear}	$w_{concavity}$	$w_{hyperbolic}$
M_1	3.20	3.28	3.39
M_2	3.20	3.52	3.70

The table shows that *linear* fails to distinguish the two matchings, but *concavity* and *hyperbolic* will select the desired matching i.e. M_2.

6.4 uneven

The early versions of *GoDraw* had a similar uneven game rule as *Gotha*'s historical component f^h defined in Section 5.2. Players in an odd McMahon group are sorted in order of $m^{bal} = U - D$, where U is the number of games played up and D is the number of games played down. The program then reduces m^{bal} for both players in a pair.

Subsequently, the BGA recognised that unexpected results should be treated more fairly. If a player plays up and wins, then playing down later to a weaker player confers an unfair advantage to the stronger. Likewise if a player plays down and loses, then later on playing up to a stronger player is a disadvantage to the weaker. So in these cases we avoid subsequent corrections. In addition to reducing m^{bal}, *GoDraw* also attempts to reduce the number of times that a player plays *outside* his/her McMahon group i.e. we try to minimise $m^{out} = U + D$.

In the next sections we construct weight functions for both the historical component of the full *Gotha* rule and the reduced *GoDraw* rule.

full uneven rule

Consider a pairing ij with $m_i < m_j$ so i plays up and j plays down. In the ideal pairing each player has had just one uneven game and the pairing reduces the unbalance to zero for both players. Consequently the conditions required for an ideal pairing are:

$$m_i^{out} = 1 \qquad m_j^{out} = 1 \tag{1}$$

$$m_i^{bal} = -1 \qquad m_j^{bal} = 1 \tag{2}$$

It is convenient to treat the effect of m^{bal} separately from the requirement to minimise the number of uneven games. To this end we write

$$w_{ij}^{uneven} = w_{ij}^{out} w_{ij}^{bal}$$

The requirement for w^{out} is that it has a local maximum of 1 when the conditions in (1) are satisfied. One function that meets this requirement is simply:

$$w_{ij}^{out} = \operatorname{sech}(\alpha(m_i^{out} + m_j^{out} - 2))$$

The constant α controls the speed with which the weight falls off as the number of uneven games for each player rises.

We can also minimise the games played up or played down separately for each player as in *Gotha* using:

$$w_{ij}^{out} = \operatorname{sech} \alpha U_i \operatorname{sech} \alpha(D_i - 1) \operatorname{sech} \alpha(U_j - 1) \operatorname{sech} \alpha D_j$$

This has a local maximum when (1) and (2) are satisfied.

The requirements for w^{bal} are more complex, as can be seen in the *Gotha* model, which treats the following cases arranged in order of pairing quality:

good	The unbalance improves for both.
average	The weaker's unbalance improves, the stronger's worsens.
average	The weaker's unbalance worsens, the stronger's improves.
poor	The unbalance worsens for both.

Each of these cases is distinguished by the values of m_i^{bal} and m_j^{bal}, and the weight function w^{bal} should vary from 1 to very low as the pairing quality varies from good to poor. In view of the condition $m_i < m_j$, the unbalance improves for *both* players when m_i^{bal} and m_j^{bal} have opposite sign *and* in addition, $m_i^{bal} < 0$. This condition can be wrapped in a variable:

$$x = m_i^{bal} - m_j^{bal} + 2 \tag{3}$$

This expression for x has the properties:

- If $m_i^{bal} < -1$ and $m_j^{bal} > 1$ then $x < 0$
- If equation (2) is satisfied then $x = 0$
- if $m_i^{bal} > 0$ and $m_j^{bal} < 0$ then $x > 0$

The variable x can thus detect *good* and *poor* pairing qualities as well as the ideal case when $x = 0$.

For the *average* pairing qualities, the signs of the unbalances are the same, so when paired, one unbalance improves at the expense of the other. This behaviour can be represented by the variable:

$$y = m_i^{bal} + m_j^{bal} \tag{4}$$

For the ideal pairing as in equation (2), $y = 0$, and y can reach its extremal values only when both balances have the same sign.

Once *both* x and y are given, the two unbalances are found from the transform:

$$m_i^{bal} = \frac{1}{2}(y + x - 1)$$
$$m_j^{bal} = \frac{1}{2}(y - x + 1)$$

We can therefore formulate the weight function entirely in terms of the variables x and y, i.e. $w_{ij}^{bal} = f(x, y)$, where the function $f(x, y)$ is required to have the properties listed below. We use $f_y(x)$ to denote the function values for fixed y and $f_x(y)$ to denote values for fixed x.

range	$f(x, y)$ lies in the range $[0, 1]$ for all x and y.
peak	It has a single isolated local maximum at $x = 0, y = 0$.
poor quality	For $x > 0$ and increasing, $f_y(x)$ decreases rapidly.
good quality	For $x < 0$ and decreasing, $f_y(x)$ decreases slowly.
average quality	As $\lvert y\rvert$ increases, $f_x(y)$ decreases modestly.

There are a variety of functions which have this behaviour, one of which is $g(x)\,\mathrm{sech}(\gamma y)$ where $g(x)$ is a skew function having the form:

$$g(x) = A(1 - \tanh(\kappa(x - a)))\,\mathrm{sech}(\alpha x)$$

A graph of this skew function was sketched in Section 6.1.

reduced uneven rule

For the less strict uneven pairing rule where corrections are not made for unexpected results, we split the up and down counts as follows:

U^-	number of games played up and lost
U^+	number of games played up and won
D^-	number of games played down and lost
D^+	number of games played down and won

We can then define the uneven balances for the players

$$\hat{m}^{bal} = U^- - D^+$$

Defining the variables $\hat{x} = \hat{m}_i^{bal} - \hat{m}_j^{bal} + 2$, and $\hat{y} = \hat{m}_i^{bal} + \hat{m}_j^{bal}$ in analogy with equations (3) and (4), we then have the same form for the weight function as in the full uneven pairing, i.e.

$$\begin{aligned} w_{ij}^{uneven} &= w_{ij}^{out}\hat{w}_{ij}^{bal} \\ \hat{w}_{ij}^{bal} &= g(\hat{x})\,\mathrm{sech}(\alpha\hat{y}) \end{aligned}$$

This has the required properties for the historical component of uneven pairing.

seeded uneven pairing

In *MacMahon* the weakest player in the lower McMahon group plays up to the strongest player in the higher McMahon group. *Gotha* generalises this via the seeded components of the weight function $f^s_{bot}(s_i) + f^s_{top}(s_j)$. There is also a case to be made for pairing players as equally as possible and this is provided by the combination $f^s_{top}(s_i) + f^s_{bot}(s_j)$, in which the strongest player from the lower group plays the weakest player from the upper group. A suitable form for the seeding component of the uneven weight function is

$$w_{ij}^{uneven-seed} = sech(\alpha f^s(s_i, s_j))$$

6.5 even seeding

In both *MacMahon* and *Gotha*, pairing within a McMahon group can follow a variety of seeding forms as listed in Section 5.1. We start with an ordering of players in a McMahon group by strength. A seed s_k is then defined in terms of the index k of the player in the ordering. However, it is possible for two players to have exactly the same strength but nevertheless occupy different positions in the ordering. So the seed could equally well be taken to be the index of the set of players with the same strength in the ordering.

For each pairing method, a rule deviation δ_{ij} is defined as a function of the player's seeds s_i and s_j. In general, the weight function is chosen to minimise the rule deviation for the matching, and so for *split&fold* or *split&match* can have the form:

$$w_{ij}^{split-method} = \text{sech}(\alpha\delta_{ij})$$

There can be considerable degeneracy in the rule deviation δ_{ij} when a sub-optimal matching needs to be chosen - perhaps because one of the pairs needed would be a repeat game. For example, consider a group of 8 players paired by *split&match* with simple seeds $s_i = i$. The rule deviation is $\delta_{ij} = |j - i| - 4$, and the following table shows the magnitudes of these for two matchings M_1 and M_2:

M_1	1-5	2-6	3-4	7-8
$\|\delta_{ij}\|$	0	0	3	3
M_2	3-7	4-8	1-2	5-6
$\|\delta_{ij}\|$	0	0	3	3

The distribution of rule deviations $|\delta_{ij}|$ for the two matchings is the same, so the sum of weights for these matchings are identical. We might consider the second matching preferable because the two strongest players are correctly matched and the errors are at the weaker ends of the two half groups.

Gotha's *split&random* can be implemented via a random permutation of the players in each half of the group, with a discrete weight function T_{ij} as defined in 5.1. *GoDraw*'s pure random pairing within a group can be implemented via a random permutation of the whole group with a weight function $w_{ij}^{random} = 1$ for any pair ij.

6.6 colour

The weight functions for colour in *MacMahon* and *Gotha* meet the aim of ensuring that players have an equal number of black and white games over the course of the tournament. It is also desirable that the colour sequence alternates, and a small modification of the weight function allows us to include this if desired.

As before, C_i is the colour balance (excess of games with white over games with black). Let the player's colour in the *previous* round be represented by c_i, where $c_i = -1$ denotes black, and $c_i = 1$ denotes white. There are two cases to consider:

case 1 If the players each have a zero colour balance and $c_i + c_j = 0$, then we have a pairing which perfectly satisfies the alternating requirement.

case 2 If the players each have a non-zero colour balance and $C_i + C_j = 0$, then we have a matching satisfying the requirement to minimise the colour balances for both players.

In order to distinguish these cases, let Q_{ij} be zero only when *both* players have zero colour balance, and 1 otherwise. Then we can define the rule deviation and weight function as follows:

$$\delta_{ij}^{colour} = Q_{ij}(C_i + C_j) + (1 - Q_{ij})(c_i + c_j)$$
$$w_{ij}^{colour} = \text{sech}(\alpha\delta_{ij}^{colour})$$

6.7 area

Consider an even McMahon group of size N, and suppose that there is one dominant area A of size a players, which contains more than half the number of players in the group, i.e. $a > N/2$. There is only one solution which ensures the minimum number of same-area games:

alien-games: $N - a$ gamesin which all the other areas play against A.

same-area: $a - N/2$ games between players in A.

This solution is always achieved by the maximum weighted matching algorithm with a discrete weight function $w_{ij} = 0$ for a same-area game and $w_{ij} = 1$ for an alien-game.

If there is no single dominant area, the same weight function will guarantee no same-area games. However, it is desirable that there is as much mixing of areas as possible. With the weight function above, the mixing that does occur is purely at the whim of the matching algorithm, and will not in any sense be optimal. Even if the payers are randomly shuffled before the matching, there can still be a huge number of solutions where the mixing is clearly lower than it could be.

One algorithm which mixes areas very effectively is this *mixed-area* matching:

MA1. Random shuffle, then order the McMahon group into subsets with decreasing area size. We obtain L sets of players from the same area with sizes $n_1, n_2 \cdots n_L$, and $n_i \geq n_{i+1}$.

MA2. Pair a player from the largest set (n_1) with a player from the smallest set (n_L). The values of n_1 and n_L reduce by 1.

MA3. Repeat MA2 until the first set is no longer the largest i.e. $n_1 = n_2 - 1$.

MA4. Repeat MA1 until $n_2 = 0$.

We can illustrate the effectiveness of this algorithm in achieving good area mixing by examining the matrix of game counts between different areas. Consider a group of size 24 players with 5 different areas with player counts $(2, 4, 5, 6, 7)$: In the matrices following, **A** arises from a random matching and **B** arises from the matching obtained by the mixed-area algorithm. The matrix entries are the counts for games between different areas and the zeros along the diagonals (showing that there are no same-area games) have been suppressed for clarity.

$$\mathbf{A} = \begin{pmatrix} . & 0 & 0 & 0 & 2 \\ 0 & . & 0 & 2 & 2 \\ 0 & 0 & . & 3 & 2 \\ 0 & 2 & 3 & . & 1 \\ 2 & 2 & 2 & 1 & . \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} . & 0 & 0 & 1 & 1 \\ 0 & . & 1 & 1 & 2 \\ 0 & 1 & . & 2 & 2 \\ 1 & 1 & 2 & . & 2 \\ 1 & 2 & 2 & 2 & . \end{pmatrix}$$

It is clear that **B** has more non-zero entries than **A** and the non-zero ones have a lower spread of values, so we would say that **B** has the better area mixing. One measure we could use to compare matchings is the variance of the game counts in the mixing-matrices above. **A** has a variance of 1.70 and **B** has a variance of 0.50.

Simulations show that random matching sometimes comes close to *mixed-area* matching, but seldom has a lower variance. It would be possible to seed players according to the result of the *mixed-area* algorithm. Then set weights to minimise the seeding error as is done in the other seeding examples in section 6.5.

7 WEIGHTS FOR ADDITIONAL RULES

Here we examine the weight models for the extra rules of banding, play-in, and play-out as discussed in section 3

7.1 banding

Players are grouped into K bands $B_1, \cdots B_K$, and the bands are considered ordered in the sense that pairing weights decrease as the separation between bands increases. with b_i the index of the band for player i, a suitable weight model for banding takes the form:

$$w_{ij}^{band} = sech(\alpha(b_i - b_j))$$

7.2 play-in

Each player in a play-in group introduced in 3.2, is given a count P^{in} of how many times the player has managed to play another player *in* the group. Players with the lowest counts get the highest pairing weights:

$$w_{ij}^{play-in} = sech(\alpha(P_i^{in} + P_j^{in}))$$

Two players with the highest play-in counts would get the lowest weights.

7.3 play-out

Let M_i denote the McMahon group containing player i. We assume the extended McMahon group $M = M_i \bigcup M_j$ contains at least two players from a set H that prefers not to play each other. If we were to choose 2 players at random, the chance that they both come from H is:

$$P_H = \frac{h(h-1)}{m(m-1)}$$

where h is the size of H and m is the size of M.

We need to assign a weight which reduces as the number of games played *in* H increases, and a weight function like $w_{ij} = \text{sech}(\alpha(P_i^{in} + P_j^{in}))$ does have the required property. It would however encourage pairings between players who have *never* played in H. We modify this weight function so that the weight decreases as the probability of playing in decreases. Define $P_{ij}^{out} = 1$ if at least one of the players i, j is not in H, and 0 otherwise.

$$\begin{aligned} w_{ij}^{play-out} &= P_{ij}^{out} + (1 - P_{ij}^{out})\,\text{sech}(\alpha(P_{offset} + P_i^{in} + P_j^{in})) \\ P_{offset} &= \text{sech}^{-1}(P_H) \end{aligned}$$

For example in a group of 10 players suppose there are two players from a play-in set H . The probability that a random choice pairs these two is 0.022. This gives $P_{offset} = 4.5$ providing a good reduction in weight for players who have not yet played others in the set H, i.e. have $P_i^{in} = P_j^{in} = 0$.

8 GRADE DEPENDENT PARAMETERS

As before, the total weight of a pairing is expressed in the form:

$$W_{ij} = \sum_{r=1\cdots 8} w^r_{ij} W_r$$

Each of the weight functions w^r_{ij} is fully determined by a small number of parameters, some of which are independent of any particular constraints imposed by the tournament population. These parameters determine the shape of the functions w^r and here we explore the possibility that the shape may usefully be varied with player grade.

8.1 scoring components

All the components of weight determined by score i.e. *mms*, *uneven*, and *seed* have similar levels of importance at the various grades grades. For example in $w^{mms}_{ij} = \text{sech}(\alpha_{mms}(m_i - m_j))$, the parameter α_{mms} controls the speed with which the weight falls off as the McMahon score difference increases. At the strong end of the draw we need the weight to fall off quickly (α_{mms} is large) so that players are evenly matched. However at the bottom end of the draw there are usually large gaps in McMahon scores and these players will be playing handicap games anyway. It is therefore possible to reduce the size of α_{mms} at the bottom end of the draw so players with sizeable score differences get paired with handicap.

The rules for uneven games and seeding also apply more strictly to stronger players above the bar, so a similar variation for their parameters would be considered. A suitable form for the behaviour of α with grade is possibly linear, or for more control, we could consider:

$$\alpha_{score}(g) = \frac{1}{2}(1 - \tanh(\gamma(g - g_0)))$$

The constant g_0 determines the point of inflection in the decreasing behaviour of α_r. It would probably be placed in the mid-kyu range.

8.2 colour

It is likely that the very strongest players will eventually run out of equal opponents and end up playing more games as white than black. It is also inevitable that the lowest graded will play many more games as black and probably with handicap. Strong kyu players sometimes complain if their colour sequence is off balance.

These remarks suggest that the colour rules should be more strictly applied in upper kyu range, much less strictly applied at the lower grade end and not so strictly applied at the dan player end. One way to achieve this is to let the α parameter in the rule for w_{ij}^{colour} in Section 6.6 have the form:

$$\alpha_{colour}(g) = sech(\gamma(g - g_0))$$

Here the constant g_0 determines the position of the peak value of α_{colour}, and the high kyu end might be an appropriate value for it.

8.3 area

As already mentioned, avoiding same-area pairing is much more important at the double figure kyu end than in the single figure kyu or dan groups. In this case the α parameter in the rule for area pairing could take the form:

$$\alpha_{area}(g) = \frac{1}{2}(1 + \tanh(\gamma(g - g_0)))$$

Here the constant g_0 is the point of inflection in α_{area}, which increases with g, and could be set at the start of the double figure kyu position.

9 MAXIMUM WEIGHTED PAIRING

9.1 Brief description

The method used in all programs relying on maximum weighted matching is briefly:

- Allocate weights to each possible pairing between players.
- A maximum weighted matching algorithm then finds the optimal pairing giving the largest sum of weights for the matched pairs.

The method represents the players as vertices in a graph. The edges of the graph are the legitimate pairings between players, and non-zero weights are assigned to those edges.

9.2 Edge removal

If weights are badly allocated it would be possible for the algorithm to pair a 1 dan against a 20 kyu. This of course would not be tolerated, and can be prevented by restricting the edges to correspond to sensible pairing ranges. It is also possible to avoid repeat pairings by removing edges used in earlier rounds.

In either of the above cases, edges have been removed from the graph and this increases the danger that for a particular choice of weights, the matching will be imperfect: i.e. some players will not be paired at all. This is not really too much of a surprise since it is not always possible to run a McMahon tournament when the ratio of players to number of rounds is too low. Experience suggests three times as many players as rounds are needed for a plausible McMahon tournament.

It is possible that the algorithm will find a maximum weight matching leaving some players unpaired even though they could be paired, all be it with a lower weight. We can however add a constant to all weights to ensure that the matching *is* maximum cardinality. A proof of this is provided in Appendix B.

10 PAIRING ALGORITHM OPTIMISATION

10.1 Introduction

The main purpose of a tournament is to choose a unique winner. The main purpose of the pairing algorithm in McMahon tournaments is to choose pairs so that the players are as evenly matched as possible. The assumption is made that players enter tournaments with a grade that realistically reflects their true playing strength. The McMahon tournament pairing rules then ensure that players will meet other players of the same strength. Given that there are different interpretations of the McMahon rules, one might well enquire as to whether there is any difference in the quality of the resulting pairings obtained from the different implementations. If there is a measurable difference, then this can be used to improve the pairing quality by suitable adjustment of the weights in the pairing algorithm.

10.2 Pairing quality

In order to discuss pairing quality and whether it is affected by different pairing algorithms, we need to state how the quality is to be measured. At the end of the tournament players are ranked usually by McMahon score, followed by one or more tie-breakers to order players on the same score. Results are also processed by the European rating [7] system (which came into use around 1997) to update each player's rating - a measure of a player's current playing strength. These processes suggest there are two ways of measuring quality, one rank-based, the other rating-based.

Quality measured by ranking

Let us assume that for each player we know their true playing strength, so that we can order players by strength to get a true rank T_i for each player i. Each player then has a true rank T_i and a tournament rank R_i at the end of the tournament. A standard measure of the overall ranking error for all the players in the tournament is the root-mean-square value:

$$\varepsilon_{ranking} = \sqrt{\frac{1}{N}\sum_{i=1\cdots N}(R_i - T_i)^2}$$

True playing strength is difficult to estimate accurately, and there can be huge variations in a given player's performance from one tournament to the next. The best estimate we have at the moment is rating, and using this as true strength does provide plausible values for the ranking error. For example, in the 2008 European [8] we get a value of 8.5 for the top 50 players.

Quality measured by rating

Let us assume that the playing strength S_i of each player is stable. After many tournaments a player's rating r_i assigned by the rating system may well be different from the player's true strength simply due to the randomness in the outcome of a game between two players. Again a standard measure of the overall rating error for the whole population of players in the system is:

$$\varepsilon_{rating} = \sqrt{\frac{1}{N}\sum_{i=1\cdots N}(r_i - S_i)^2}$$

To compare with real-life data, assume that player's grades g_i are stable over a year and reflect a true estimate of their playing strength. The EGD statistical page [9] provides data which leads to the value $\varepsilon_{rating} = 0.33$ for the whole of Europe in 2011. This is a per-country estimate, which assumes that all players in the country have the same rating - grade difference. It is of course an overall figure resulting from many different tournaments run with different pairing programs.

10.3 Tournament simulation

In a given tournament, a player's rank and the change in a player's rating depend on two major factors:

- The strength of the player's opponents.
- The outcome of the games.

As noted in the previous sections, the strengths of real life players is very variable, but in a tournament *simulation* we can control players' strengths and thereby ensure that all the assumptions made for running a McMahon tournament and measuring quality can be met. For the purposes of simulation we start with a pool of players, each with a given fixed grade regarded as their true playing strength. The pairing algorithm then determines the strengths of players' opponents at each round.

The outcome of a game between two players is determined by the probability of win $p_{win}(g_i, g_j)$ for players with grades g_i, and g_j. The European rating system provides us with data[10] on measured probabilities of win in any game between players with grade differences in the range 1 to 4. This data has enabled us to construct a model [11] for any values of grade, and hence we can realistically simulate the outcome of any game in the tournament.

At the end of the last round we obtain a ranking for the one simulated tournament, and an update to the players' ratings dependent only on the players' ratings at the start of the tournament. By simulating a sequence of tournaments we obtain a sequence of measured values for $\varepsilon_{ranking}$ and ε_{rating}. These sequences should show the effect of varying any of the parameters used in the weight model and so allow settings to be chosen to give the best tournament quality.

Further adjustments to settings can of course be made following feedback from actual live tournaments, but the values obtained from simulation ought to provide a good starting point.

References

[1] Christoph Gerlach,MacMahon,
www.cgerlach.de/go/macmahon.html

[2] Luc Vannier, Open Gotha,
vannier.info/jeux/gotournaments/opengotha.htm

[3] Geoff Kaniuk,GoDraw,
www.britgo.org/downloads/godraw/

[4] BGA, McMahon pairing rules,
www.britgo.org/organisers/mcmahonpairing.html

[5] Christoph Gerlach,Thesis,Section 2.3.1,
www.cgerlach.de/go/diplom.pdf

[6] Luc Vannier,Open Gotha user's guide,
vannier.info/jeux/download/GothaHelp_en.pdf

[7] Ales Cieply,EGF Rating System,
www.europeangodatabase.eu/EGD/EGF_rating_system.php

[8] Geoff Kaniuk,egc_simulation_2008.pdf,
www.kaniuk.co.uk/articles/egc-champs/

[9] Aldo Podavini, EGD Country/Club Statistics,
www.europeangodatabase.eu/EGD/Stats_Country.php

[10] Aldo Podavini,Winning statistics,
www.europeangodatabase.eu/EGD/winning_stats.php

[11] Geoff Kaniuk,Probability of Win,
www.kaniuk.co.uk/articles/egd-analysis/pwin/prob-win.pdf

[12] Harold.N Gabow, Implementation of Algorithms for maximum matching on nonbipartite graphs(chapters 1 and 4),PhD Thesis,Stanford University,1974

A INDEX OF NOTATION

In all symbols, i identifies a single player, ij identifies a pair of players. Some of the symbols are enclosed in { }. These represent propositions shown in the description. In this case the symbol has the value 1 when the proposition is true and the value 0 when the proposition is false.

SYMBOL	DESCRIPTION	SECTION
β	The McMahon bar	4
b_i	Identifies the band for player i	5.1
B	Number of bands	5.1
B_i	Number of games played with black	4.1
$\{\underline{B}_{ij}\}$	At least one of the players is above the bar	4.2
$\{\overline{B}_{ij}\}$	Both players are below the bar (below means $<$)	4.2
B_k	Set of players in the k^{th} band	7.1
C_i	Algebraic colour balance $W_i - B_i$	5.1
C_{ij}	Cost of a pairing	4.2
$\{C_{ij}^{mixed}\}$	$\|C_i + C_j\| > 2$ and $C_iC_j = 0$	5.1
$\{C_{ij}^{opp}\}$	Players have opposite colour balance	5.1
δ_{ij}^r	Rule deviation for the two players	5
D_i	Number of games played down	5.1
$\{E_{ij}\}$	The game is even	4.2
g_i	Player's grade	4
$\{H_{ij}\}$	$m_i - m_j > 9$	4.2
$\{K_{ij}\}$	Players belong to the same club	4.2
$\{L_{ij}\}$	Players belong to the same country	4.2
m_i	McMahon score for player i	2
$\hat{m}_i$	Scaled McMahon score for handicap pairings	5.1
m^{bal}	Games played up - games played down	6.4
m^{out}	Total number of uneven games	6.4
M	Number of McMahon groups	5.1
M_i	McMahon group containing player i	7.3
P_i^{in}	Number of times played in group	7.2
$\{P_{ij}^{out}\}$	At least one player not in play-out group	7.3
P_H	Probability two players are in the set H	7.3
$\{Q_{ij}\}$	At least one of C_i or C_j is zero	6.6
ρ	Random noise in range $[0, 1]$ or a hash names	5.1
$\{R_{ij}\}$	The players have met in a previous round	4.1
s_i	Seed for player i	2.4
S	Size of seeded McMahon group	2.4
$\{T_{ij}\}$	Players i and j lie in different halves of a group	5.1
U_i	Number of games played up by player i	5.1
$\{U_{ij}\}$	The game is uneven: $U_{ij} = 1 - E_{ij}$	4.2
W_i	Number of games played with white	4.1
W_{ij}	Weight of a pair	5
Y_i	Non-negative colour balance for player i	4.1

B MAXIMUM CARDINALITY MATCHING

We examine the condition required for a maximum weight matching to have maximum cardinality.

B.1 Example

This table defines an incomplete weighted graph with 4 vertices. There is no edge connecting vertices 1 and 4.

edge	weight
1-2	3
1-3	2
2-3	9
2-4	1
3-4	4

As can be seen in the next table, there are maximum cardinality matchings with 2 edges. But there is only one maximum weighted matching. It has 1 edge, and leaves vertices 1 and 4 unmatched.

matching	edges	cardinality	total weight
M_1	1-2 3-4	2	7
M_2	1-3 2-4	2	3
M_3	2-3	1	9

The maximum weighted matching algorithm will find matching M_3 and reject the others of lower weight. However, if we add the same weight to all the edges, we will increase the weight of M_1 and M_2 more than we increase the weight of M_3 because they have more edges. In particular, if we add 2 to the weight of each edge, M_1 and M_3 will each have a total weight of 11, and the algorithm will choose M_1. We prove below that this procedure can be applied to any weighted graph to obtain a maximum weight and cardinality.

B.2 Weight of an augmenting path

Following the notation developed in [12], let $G(V, E, w)$ be a weighted graph. Let M be a matching on G and suppose that p is an augmenting path in the matched graph (G, M). Then p has the properties:

1. p is a sequence of $n = 2k$ distinct vertices $p = (v_1, v_2, \cdots, v_n)$.
2. Edge $e_{2j} = v_{2j}v_{2j+1} \in M$ for $j = 1 \cdots k - 1$.
3. Edge $e_{2j-1} = v_{2j-1}v_{2j} \notin M$ for $j = 1 \cdots k$.
4. No edge in M contains v_1 or v_n.

Let P be the set of edges generated by the path p i.e.

$$P = \{v_i v_{i+1} : i = 1 \cdots n-1\}. \tag{5}$$

Given any subset of edges $F \subseteq E$, define the *weight map* $W(F) = \sum_{e \in F} w(e)$. It is shown in [12] that the symmetric difference $P \bigoplus M$ is a new matching M^+ which has cardinality $|M^+| = |M| + 1$, and the weight of the augmented matching M^+ is given by:

$$\begin{aligned} W(M^+) &= W(M) + W(P, M) \\ W(P, M) &= W(P - M) - W(P \bigcap M) \end{aligned} \tag{6}$$

From the definition (5), we see that $P - M$ includes all the odd edges generated by p (Property 3. above), and $P \bigcap M$ includes all the even edges generated by p (Property 2. above). Let $w_i = w(e_i)$, then the component sums in (6) are

$$W(P - M) = \sum_{j=1}^{k} w_{2j-1} \tag{7}$$

$$W(P \bigcap M) = \sum_{j=1}^{k-1} w_{2j} \tag{8}$$

B.3 Condition for increase in augmented path weight

The augmented matching M^+ always has increased cardinality. However if $W(P, M) < 0$, then $W(M^+) < W(M)$ and the augmenting path will be rejected. The algorithm will accept an augmenting path only if $W(P, M) \geq 0$. Consider now a copy $\hat{G}(V, E, \hat{w})$ of the weighted graph G in which the weight function is defined as

$$\hat{w}(e) = w_0 + w(e) \quad \forall \ e \in F$$

For this graph the weight map is:

$$\hat{W}(F) = \sum_{e \in F} \hat{w}(e) = w_0|F| + \sum_{e \in F} w(e) = w_0|F| + W(F)$$

The component sums in (7) and (8) transform to:

$$\hat{W}(P - M) = kw_0 + \sum_{j=1}^{k} w_{2j-1}$$

$$\hat{W}(P \bigcap M) = (k-1)w_0 + \sum_{j=1}^{k-1} w_{2j}$$

Consequently, the increase in weight of a matching in $\hat{G}$ is:

$$\begin{aligned}\hat{W}(P,M) &= \hat{W}(P-M) - \hat{W}(P\bigcap M) \\ &= w_0 + \sum_{j=1}^{k} w_{2j-1} - \sum_{j=1}^{k-1} w_{2j} \qquad (9)\end{aligned}$$

We require $\hat{W}(P,M) \geq 0$, which from (9), can only be satisfied if:

$$w_0 \geq \sum_{j=1}^{k-1} w_{2j} - \sum_{j=1}^{k} w_{2j-1} \qquad (10)$$

The *RHS* of (10) has its maximum value of $X - Y$ when:

$$X = \max\left\{\sum_{j=1}^{k-1} w_{2j}\right\} \qquad (11)$$

$$Y = \min\left\{\sum_{j=1}^{k} \quad w_{2j-1}\right\} \qquad (12)$$

Let the weight range for the original graph G be:

$$w_{min} \leq w(e) \leq w_{max} \,\forall\, e \in E$$

Then from (11) and (12)

$$X = (k-1)w_{max} \qquad (13)$$

$$Y = k\, w_{min} \qquad (14)$$

B.4 Theorem

Let the maximum cardinality matching for G have n_c edges. If

$$w_0 = (n_c - 1)w_{max} - n_c w_{min} \qquad (15)$$

then $\hat{G}$ has a maximum weight matching with cardinality n_c.

Proof. For any matching M of $\hat{G}$, and any augmenting path with $n = 2k$ vertices for that matching, equation (9) applies and so with (15) we get:

$$\begin{aligned}\hat{W}(P,M) &= (n_c - 1)w_{max} - n_c w_{min} + \sum_{j=1}^{k} w_{2j-1} - \sum_{j=1}^{k-1} w_{2j} \\ &\geq (n_c - 1)w_{max} - n_c w_{min} + k w_{min} - (k-1)w_{max} \\ &= (n_c - k)w_{max} - (n_c - k)w_{min} \\ &= (n_c - k)(w_{max} - w_{min})\end{aligned}$$

Since $n_c >= k$ and $w_{max} \geq w_{min}$, the result $\hat{W}(P,M) \geq 0$ follows. □

The above value for w_0 is larger than actually needed in specific examples.