Improving Ranking Quality and Fairness in Swiss-System Chess Tournaments

Pascal Sauer\textsuperscript{a}, Ágnes Cseh\textsuperscript{b,c}, and Pascal Lenzner\textsuperscript{d}

\textsuperscript{a} Potsdam Institute for Climate Impact Research, Potsdam, Germany;
\textsuperscript{b} Institute of Economics, HUN-REN Centre for Economic and Regional Studies, Budapest, Hungary;
\textsuperscript{c} Department of Mathematics, University of Bayreuth, Bayreuth, Germany;
\textsuperscript{d} Hasso Plattner Institute, University of Potsdam, Potsdam, Germany

ARTICLE HISTORY
Compiled December 14, 2023

ABSTRACT
The International Chess Federation (FIDE) imposes a voluminous and complex set of player pairing criteria in Swiss-system chess tournaments and endorses computer programs that are able to calculate the prescribed pairings. The purpose of these formalities is to ensure that players are paired fairly during the tournament and that the final ranking corresponds to the players’ true strength order.

We contest the official FIDE player pairing routine by presenting alternative pairing rules. These can be enforced by computing maximum weight matchings in a carefully designed graph. We demonstrate by extensive experiments that a tournament format using our mechanism (1) yields fairer pairings in the rounds of the tournament and (2) produces a final ranking that reflects the players’ true strengths better than the state-of-the-art FIDE pairing system.

KEYWORDS
Swiss system, tournaments, fairness, ranking, matching, Kendall $\tau$, chess

A 2-page abstract of this work appeared at The 23rd ACM Conference on Economics and Computation (EC’22).

CONTACT Ágnes Cseh. Email: agnes.cseh@uni-bayreuth.de
1. Introduction

How can one determine the relative strength of players who engage in a one-on-one competitive game? This is easy to find out for a group of two players: just let them play a match. For more players, tournaments solve this problem by ranking the players after a limited number of pairwise matches among the participants. The tournament format defines a general structure of matches to be played and the method for deriving a ranking from the results of those matches.

Tournament Formats

Most tournaments follow an elimination, a round-robin, or a Swiss-system format. In each round of an elimination tournament, such as the second stage of the FIFA World Cup, only players who won their match in the previous round are paired again. The last player standing wins the tournament, and the remaining players’ strength can only be estimated very roughly from the round they were eliminated in. Round-robin tournaments are also called all-play-all tournaments, because each player plays against each other player once. The player with the highest score at the end of the tournament is declared the winner. The pool stage of the FIFA World Cup consists of round-robin tournaments.

The Swiss-system tournament format is widely used in competitive games like most e-sports, badminton, and chess, the last of which this paper focuses on. In such tournaments, the number of rounds is predefined, but the pairing of players in each of these rounds depends on the results of previous rounds. This format offers a convenient golden middle way between the earlier mentioned two tournament formats. However, the features of the Swiss system challenge organizers greatly. Firstly, unlike in elimination tournaments, the goal is to determine a whole ranking of the players and not only to declare the winner. Secondly, the final ranking of each player is greatly influenced by her assigned opponents, which is not an issue in round-robin tournaments.

Therefore, a mechanism that computes suitable player pairings for Swiss-system tournaments is crucially important. However, designing such a system is a challenging task as it boils down to solving a complex combinatorial optimization problem. Inter-
estingly, the state-of-the-art solution to this problem in chess tournaments relies on a complex set of declarative rules and not on a combinatorial optimization algorithm. In this paper we provide an algorithmic approach and we demonstrate that it outperforms the declarative state-of-the-art solution. For this, we do not try to mimic the FIDE solution but instead focus on the most important features of the Swiss system and derive a maximum weight matching formulation that enforces them.

*The Swiss-System in Chess*

In Swiss-system chess tournaments, there are two well-defined and rigid *absolute* and two milder *quality* pairing criteria. These criteria form the backbone of the much more specific and rigid declarative FIDE rules (FIDE, 2020, Chapter C.04).

(A1) No two players play against each other more than once.

(A2) In each round before the last one, the difference of matches played with white and matches played with black pieces is between −2 and 2 for every player.

(Q1) Opponents have equal or similar score.

(Q2) Each player has a balanced color distribution.

Criterion (A1) ensures variety, while criterion (A2) ensures fairness, since the player with white pieces starts the game, and thus has an advantage over her opponent (Heny, 1992; Milvang, 2016). These absolute criteria must be obeyed at any cost, which often enforces the relaxation of the two quality criteria.

In order to implement criterion (Q1), players with equal score are grouped into *score groups*. In each round, a chosen *pairing system* allocates each player an opponent from the same score group. If a complete pairing is not possible within a score group, then one or more players are moved to another score group. Criterion (Q2) requires that after each round of the tournament, the difference between matches played with black and white pieces is small for each player.

Adhering to these four criteria makes pairing design truly challenging. Besides incorporating the four criteria, the FIDE also requires that the same pairing is generated whenever the same situation arises at competitions. To ensure this uniqueness, additional rigid declarative rules have been added (FIDE, 2020, Chapter C.04). Pairings at
FIDE tournaments were traditionally calculated manually by so-called arbiters, often using trial-and-error. Today, pairings are computed by decision-making software, but the FIDE pairing criteria are still written for human instead of computer execution. Over the years, more and more criteria were added to resolve ambiguities, which increased the complexity to a level at which pairing decisions are very challenging to comprehend for most players and even arbiters.

1.1. Related Literature

Novel algorithms that assist tournament scheduling regularly evoke interest in the AI and Economics communities (Bimpikis, Ehsani, and Mostagir, 2019; Chatterjee, Iliesen-Jensen, and Tkadlec, 2016; Dagaev and Suzdaltsev, 2018; Gupta, Roy, Saurabh, and Zehavi, 2018; Harbring and Irlenbusch, 2003; Hoshino, 2018; Karpov, 2018; Kim and Williams, 2015; Larsson, Goossens, and Spieksma, 2023; Larson, Johansson, and Carlsson, 2014; Van Bulck and Goossens, 2019). Also, due to its relation to voting, analyzing tournament solutions, mostly variants of round-robin tournaments, is a prominent research direction in Economics and Social Choice Theory (Brandt, Brill, Seedig, and Suksompong, 2018; Brandt, Conitzer, Endriss, Lang, and Procaccia, 2016; Brandt and Fischer, 2007; Hudry, 2009; Kim, Suksompong, and Williams, 2017; Laslier, 1997; Moulin, 1986; Saile and Suksompong, 2020; Stanton and Williams, 2011). We first elaborate on existing work on tournament formats, and then turn to approaches that utilize matchings for scheduling tournaments. For the reader not familiar with graphs and combinatorial optimization, we suggest to consult the book of Korte and Vygen (2012).

Comparing Tournament Formats

Appleton (1995) gives an overview of tournament formats and compares them with respect to how often the best player wins. Scarf, Yusof, and Bilbao (2009) simulate different tournament formats using team data from the UEFA Champions League. Recently, a comparative study by Sziklai, Biró, and Csató (2022) found that the Swiss-system tournament is the most efficient format in terms of approximating the true
ranking of the players. Moreover, Elmenreich, Ibounig, and Fehérvári (2009) compare several sorting algorithms, including one based on a Swiss-system tournament, with respect to their robustness, which is defined as the degree of similarity between the resulting ranking and the true strength order of players. They find round-robin sort, merge sort, and Swiss-system sort to be the most robust overall.

Swiss-System Tournaments

Sports tournaments are by far not the only application area of the Swiss system. Self-organizing systems (Fehérvári and Elmenreich, 2009), person identification using AI methods (Wei, Tian, Wang, and Huang, 2015), and choosing the best-fitting head-related transfer functions for a natural auditory perception in virtual reality (Voong and Oehler, 2019) are all areas where the Swiss system appears as a solution concept. To the best of our knowledge, there are only a few papers that analyze Swiss-system tournaments. The works of Csató (2013; Csató, 2017, 2021) study the ranking quality of real-world Swiss-system tournaments, in particular, whether based on the match results a fairer ranking could have been obtained by different scoring rules. However, this research is orthogonal to our approach since pairing rules are not considered.

Automated Matching Approaches

A tournament schedule can be seen as a set of matchings—one for each round. Glickman and Jensen (2005) propose an algorithm based on maximum weight perfect matchings to find the schedule. This algorithm maximizes the information gain about players’ skill. The authors’ approach compares favorably against random and Swiss-system pairing if at least 16 rounds are played. However, almost all real-world Swiss-system chess tournaments have less than 10 rounds according to chess-results.com (Herzog, 2020a).

Kujansuu, Lindberg, and Mäkinen (1999) use the stable roommates problem, see Irving (1985), to model a Swiss-system tournament pairing decision. Each player \( p \) has a preference list, which ranks the other players by how desirable a match between player \( p \) and each other player would be. The desirability depends on score difference and color balance. In comparison to the official FIDE pairing, this approach produces
pairings with slightly better color balance but higher score differences between paired players, or, in other words, clearly favors criterion (Q2) over (Q1).

*Weighted Matching Models for Chess Tournaments*

The two papers closest to ours focus on modeling the exact FIDE pairing criteria and computing the prescribed pairings. Olafsson (1990) pairs players using a maximum weight matching algorithm on a graph, where players and possible matches are represented by vertices and edges. Edge weights are set so that they model the 1985 FIDE pairing criteria. At that time, pairing criteria were more ambiguous than today, and pairing was done by hand, which sometimes took several hours. In contrast, using Olafsson’s method, pairings could be calculated fast. Pairings calculated with the commercial software built by Olafsson are claimed to be preferred by experts to manually calculated pairings. However, Olafsson only provides examples and does not present any comparison based on formal criteria.

A more recent attempt to convert the FIDE pairing criteria into a weighted matching instance was undertaken by Biró, Fleiner, and Palincza (2017). Due to the extensive criterion system, only a subset of the criteria were modeled. The authors show that a pairing respecting these selected criteria can be calculated in polynomial time, and leave it as a challenging open question whether the other FIDE criteria can also be integrated into a single weighted matching model. The contribution appears to be purely theoretical, since neither a comparison with other pairing programs, nor implementation details are provided.

Our work breaks the line of research that attempts to implement the declarative FIDE pairing criteria via weighted matchings. Instead, we design new pairing rules along with a different mechanism to compute the pairings, and demonstrate their superiority compared to the FIDE pairing criteria and engine. This clearly differentiates our approach from the one in Biró et al. (2017); Olafsson (1990).
1.2. Preliminaries and FIDE Criteria

Players are entities participating in a Swiss-system tournament. Each player has an Elo rating, which is a measure designed to capture her current playing strength from the outcome of her earlier matches (Elo, 1978). In a match two players, \(a\) and \(b\), play against each other. The three possible match results are: \(a\) wins and \(b\) loses, \(a\) and \(b\) draw, \(a\) loses and \(b\) wins. The winner receives 1 point, the loser 0 points, while a draw is worth 0.5 points. A Swiss-system tournament consists of multiple rounds, each of which is defined by a pairing: a set of disjoint pairs of players, where each pair plays a match. At the end of the tournament, a strict ranking of the players is derived from the match results.

Bye Allocation

In general, each player plays exactly one match per round. For an odd number of players, one of them receives a so-called ‘bye’, which is a point rewarded without a match. This is always the player currently ranked last among those who have not yet received a bye.

Color Balance

The FIDE Handbook (FIDE, 2020, Chapter C.04.1) states that ‘For each player the difference between the number of black and the number of white games shall not be greater than 2 or less than -2.’ This criterion may only be relaxed in the last round. This corresponds to our criterion (A2). Also, a ban on a color that is assigned to a player three times consecutively, and further milder criteria are phrased to ensure a color assignment as close to an alternating white-black sequence as possible (FIDE, 2020, Chapters C.04.3.A.6 and C.04.3.C). The color assignment in the first round is drawn randomly.

Pairing Systems

Players are always ranked by their current tournament score. Furthermore, within each score group the players are ranked by their Elo rank. The score groups and this ranking
are the input of the *pairing system*, which assigns an opponent to each player. Three main pairing systems are defined for chess tournaments. Table 1 shows an example pairing for each of them.

- **Dutch**: Each score group is cut into an upper and a lower half. The upper half is then paired against the lower half so that the $i$th ranked player in the upper half plays against the $i$th ranked player in the lower half. Dutch is the de facto standard for major chess tournaments.
- **Burstein**: For each score group, the highest ranked unpaired player is paired against the lowest ranked unpaired player repeatedly until all players are paired.
- **Monrad**: In ascending rank order each unpaired player in a score group is paired against the next highest ranked player in that score group.

The vast majority of chess tournaments use the Dutch system, however, Burstein is the main pairing principle behind the team pairings at the prestigious Chess Olympiads ([FIDE], 2023), while Monrad is commonly used in Denmark and Norway ([Wikipedia], 2023).

| Dutch | Burstein | Monrad |
|-------|----------|--------|
| 1–5   | 1–8      | 1–2    |
| 2–6   | 2–7      | 3–4    |
| 3–7   | 3–6      | 5–6    |
| 4–8   | 4–5      | 7–8    |

Table 1.: Example pairing for each pairing system in a score group of 8 players. Players are referenced by rank within the score group, i.e., player 1 has the highest Elo rank.

For comparison, we propose two additional pairing systems based on randomness.

- **Random**: Every player within a score group is paired against a random player from her score group.
- **Random2**: Every player from the top half of her score group is paired against a random player from the bottom half of her score group.

*Floating Players*

Players who are paired outside of their own score group are called *floaters*. To ensure that opponents are of similar strength—our criterion (Q1)—, the FIDE criteria require to
minimize the number of such floaters and aim to float them to a score group of similar score. However, floating is unavoidable, e.g., in score groups with an odd number of players, and also in score groups where the first or second criterion eliminates too many possible matches.

**The BBP Pairing Engine**

A *pairing engine* is used to calculate the pairing for each round, based on the results of previous rounds. The BBP pairing engine was developed by [Bierema](#) (2017). It implements the FIDE criteria strictly ([FIDE](#) 2020, C.04.3 and C.04.4.2) for the Dutch and Burstein pairing systems and outputs the unique pairing adhering to each of them. BBP uses a weighted matching algorithm, similarly as the approaches in [Bíró et al.](#) (2017); [Ólafsson](#) (1990). The main difference to our algorithm is that while the weighted model of BBP was designed to follow the declarative criteria of FIDE and output the prescribed pairings, our pairing engine relies on a different weighted model, computes completely different pairs, and while doing so, it is able to reach a better ranking quality and a higher degree of fairness. The output of Dutch BBP will serve as a base for our comparisons throughout the paper, because Dutch is the sole pairing system implemented by programs currently endorsed by the FIDE ([FIDE](#) 2020, C.04.A.10.Annex-3) and because the BBP pairing engine is open-source. We further remark that even though a Burstein BBP code exists and has been made public, its author draws attention to the fact that it is “a flawed implementation of a version of the Burstein system, not endorsed by the FIDE Systems of Pairings and Programs Commission.”

**Final Ranking**

The major organizing principle for the final ranking of players is obviously the final score. Players with the same final score are sorted by tiebreakers. The FIDE ([FIDE](#) 2020, Chapter C.02.13) defines 14 types of tiebreakers, and the tournament organizer lists some of them to be used at the specific tournament. If all tiebreaks fail, the tie is required to be broken by drawing of lots.
1.3. Our Contribution

In this paper, we present a novel mechanism for calculating pairings in Swiss-system chess tournaments. With this, we contest the state-of-the-art FIDE pairing criteria, which are implemented by the BBP pairing engine. We compare the two systems by three measures: ranking quality, number of floaters, and color balance quality, in accordance with the FIDE tournament schedule goals. Our main findings are summarized in the following list and in Table 2.

(1) We implemented the pairing systems Dutch, Burstein, Monrad, Random, and Random2 with an extensible and easy-to-understand approach that uses maximum weight matchings.

(2) The pairing systems in descending order by expected ranking quality are: Burstein > Random2 > Dutch = Dutch BBP > Random > Monrad. In particular, our implementations of Burstein and Random2 both yield higher ranking quality, while our implementation of Dutch yields similar ranking quality as the one reached by the Dutch BBP pairing engine.

(3) We utilize our weighted matching model to define a novel measure called ‘normalized strength difference’, which we identify as the main reason for a good ranking quality. This also explains why our approach outperforms the Dutch BBP engine.

(4) The pairing systems in ascending order by expected number of floaters are: Burstein < Random2 = Dutch = Monrad < Dutch BBP < Random. Compared to Dutch BBP, almost all variants of our mechanism are fairer in terms of matching more players within their own score group.

(5) All our pairing systems ensure the same color balance quality as Dutch BBP, with Random even reaching a better color balance. Moreover, we show that our approach can easily be modified to enforce an even stronger color balance. This does not significantly affect the ranking quality—only the number of floaters increases slightly.

Main Take-Away: Our new implementations of Dutch, Burstein, and Random2 either outperform or are on a par with Dutch BBP in all measured aspects. Thus, we propose
to use our implementations as a pairing engine in future FIDE chess tournaments.

| Ranking quality | Burstein | Random2 | Dutch | Dutch BBP | Random | Monrad |
|-----------------|----------|---------|-------|-----------|--------|--------|
| Number of floaters | Burstein | Random2 | Dutch | Monrad | Dutch BBP | Random |
| Color balance | Random | Random2 | Dutch | Dutch BBP | Burstein | Monrad |

Table 2.: The hierarchy of the discussed six pairing systems, with the benchmark Dutch BBP being marked blue. Each row represents a metric, and the earlier a system appears in the row, the better it performs measured with the corresponding metric. Blocks in rows denote ties.

Rankings derived from pairwise comparisons constitute an active research area, mainly due to their versatile applicability, extending from web search and recommender systems (Beutel, Chen, Doshi, Qian, Wei, Wu, Heldt, Zhao, Hong, Chi, and Goodrow, 2019; Chen, Bennett, Collins-Thompson, and Horvitz, 2013) to sports tournaments (Csató, 2013; Simunay-Stern, 1988). For simplicity and reproducibility, we focus on the latter application area, but our approach has the potential to be applied to other areas as well, where pairwise comparison schedules are designed.

2. Pairings via Maximum Weight Matching

Our novel mechanism is based on computing a maximum weight matching (MWM) in an auxiliary, suitably weighted graph. The MWM engine is optimized for simplicity: score groups, color balances, and the employed pairing system are modeled by weights, so only a single computation of a MWM is needed in each round. We now describe the MWM engine.

2.1. Input

Each tournament has $n$ players $P = \{p_1, \ldots, p_n\}$, a chosen pairing system (Dutch, Burstein, Monrad, Random, or Random2), and a maximum allowed color difference $\beta$. As criterion (A2) states, FIDE aims for $\beta = 2$. If $n$ is odd, the weakest performing player who has not received a bye yet is given one, in accordance with the FIDE rules. In the MWM engine we will exclude the same player while constructing the auxiliary graph. Hence, from this point on we can assume that $n$ is even.
Before each tournament round, the following input parameters are defined for each player $p_i \in P$:

- $Elo(p_i)$: the Elo rating of $p_i$ prior to the tournament. This remains unchanged for all rounds.
- $s(p_i)$: the current score of $p_i$, defined as the sum of points player $p_i$ collected so far.
- $r(p_i)$: the current rank of $p_i$, calculated from ordering all players in decreasing order according to their scores and their Elo ratings. Higher score and higher Elo rating yield better rank. Players with equal Elo rating are ordered randomly at the beginning, and their order is kept for all rounds.
- $cd(p_i)$: the current color difference of $p_i$, defined as the number of matches played with white minus the number of matches played with black pieces.

### 2.2. Graph Construction

With these parameters as input, we construct the corresponding auxiliary weighted graph $G_r = (V, E, w)$ for round $r$ as follows. Let $V := P$ and for all pairs of players $p_i \neq p_j$, let the edge set $E$ contain the edge $\{p_i, p_j\}$ if

1. $p_i$ and $p_j$ have not yet played against each other, and
2. $|cd(p_i) + cd(p_j)| < 2\beta$.

These rules ensure criteria (A1) and (A2). The second condition in our model will enforce $-2 \leq cd(p_i) \leq 2$ together with our color assignment rule in Section 2.3. In Section 4.4 we additionally consider a variant where $-1 \leq cd(p_i) \leq 1$ is enforced. This implements FIDE’s criterion that the color assignment should be as close to an alternating white-black sequence as possible and that no player can be assigned the same color three times in a row.

The weight of an edge $\{p_i, p_j\} \in E$ is defined as the tuple

$$w(p_i, p_j) := (-|s(p_i) - s(p_j)|, -|cd(p_i) + cd(p_j)|, \pi(p_i, p_j)),$$

where the value of $\pi(p_i, p_j)$ depends on the pairing system as follows.
• Monrad: $\pi(p_i, p_j) := -|r(p_i) - r(p_j)|$.
• Burstein: $\pi(p_i, p_j) := |r(p_i) - r(p_j)|^{1.01}$.
• Dutch: $\pi(p_i, p_j) := -\left|\frac{\text{sg size}}{2} - |r(p_i) - r(p_j)|\right|^{1.01}$, where sg size is set to 0 if $p_i$ and $p_j$ belong to different score groups, and it is the size of the score group of $p_i$ and $p_j$ otherwise.
• Random: $\pi(p_i, p_j) := \text{random number in the interval (0, 1)}$.
• Random2: $\pi(p_i, p_j)$ is set to a random number in the interval (0, 1) if $p_i$ and $p_j$ belong to different halves of the same score group, otherwise it is set to a random number in the interval ($-1, 0$).

The exponent 1.01 in the function for Burstein rewards a larger rank difference, i.e., the Burstein pairing in Table [I] indeed carries a larger weight than the Dutch pairing, which has the same sum of rank differences. Similarly, the exponent for Dutch penalizes a larger distance from $\frac{\text{sg size}}{2}$. Notice that this exponent could be an arbitrary number as long as it is larger than 1.

See Figure [I] for an illustration that shows the corresponding edge weights using the Dutch pairing system for a sample instance.

2.3. Algorithm

The edge weights of $G_r$ are compared lexicographically and a maximum weight matching is sought for. This implies that pairing players within their score groups has the highest priority, optimizing color balance is second, and adhering to the pairing system is last. The comprehensive rules of our framework consist of our two absolute rules for including an edge in the graph $G_r$, and this priority ordering serving as our quality rule.

Before round $r$, we compute a maximum weight matching $M$ in graph $G_r$ and derive the player pairing from the edges in $M$. If $\{p_i, p_j\} \in M$ then the players $p_i$ and $p_j$ will play against each other in round $r$. Between them, the respective player with the lower color difference will play white. If they have the same color difference—e.g., in the first round—, then colors are assigned randomly.

Example 2.1. We consider pairings of an example 4-round tournament with 8 players
Figure 1.: A sample instance of the graph construction for a tournament with eight players. The current ranking at the tournament is shown on the top left, the constructed graph is shown top right and its edge weights that arise from using the Dutch pairing system are depicted on the bottom. Edge weight values are rounded. Bold edges are possible matches within the same score group whereas dashed edges are other possible matches. Missing edges are matches that were already played or that are forbidden due to the color balance criterion. For example, the edge weight \( w(p_1, p_8) = (0, 0, 0) \) is shown.

Initially players are sorted decreasingly according to their Elo rating. In the following figures, bold edges are possible matches within the same score group, whereas dashed edges are other possible matches. The maximum weight matchings are shown in red. Arrows within the tables indicate the match outcomes (winner points to loser, no draws), and the color column shows the corresponding color distribution. The table for round \( i + 1 \) is based on the table of round \( i \).

- As score and color difference are equal, the pairing in round 1 is enforced by the Dutch pairing system. See Figure 2.
- The pairing in round 2, depicted in Figure 3, is the outcome of optimizing first for criterion (Q1) and then for criterion (Q2), e.g., in \( G_2 \) we have \( w(p_1, p_3) = w(p_4, p_6) = (0, 0, -1) \) and \( w(p_1, p_4) = w(p_3, p_6) = (0, -2, 0) \) so the MWM chooses

\[
\begin{align*}
\text{w}(p_1, p_2) &= (0, 0, -1) & \text{w}(p_2, p_8) &= (0, 0, 0) & \text{w}(p_5, p_6) &= (1, 0, -2.01) \\
\text{w}(p_1, p_4) &= (-1, 0, -1) & \text{w}(p_3, p_4) &= (0, 0, 0) & \text{w}(p_5, p_7) &= (0, 0, 0) \\
\text{w}(p_1, p_6) &= (0, 0, -1) & \text{w}(p_3, p_5) &= (-2, 0, -6.11) & \text{w}(p_5, p_8) &= (-1, 0, -1) \\
\text{w}(p_2, p_3) &= (-1, 0, -3.03) & \text{w}(p_3, p_6) &= (-1, 0, -4.06) & \text{w}(p_6, p_7) &= (-1, 0, -3.03) \\
\text{w}(p_2, p_4) &= (-1, 0, -2.01) & \text{w}(p_3, p_4) &= (-2, 0, -5.08) & \text{w}(p_6, p_8) &= (0, 0, -1) \\
\text{w}(p_2, p_7) &= (-1, 0, -4.06) & \text{w}(p_4, p_7) &= (-2, 0, -5.08) & \text{w}(p_7, p_8) &= (-2, 0, -6.11)
\end{align*}
\]
the edges \{p_1, p_3\} and \{p_4, p_6\}.

- In round 3, depicted in Figure 4 in \(G_3\) players \(p_3\) and \(p_4\) are paired since \(w(p_3, p_4) = (0, 0, 0)\) whereas the weight of any other incident edge of both \(p_3\) and \(p_4\) has lexicographically lower weight.

- Finally, the round 4 matching in \(G_4\), depicted in Figure 5 is enforced by maximizing the number of matches within score groups. If \(p_1\) and \(p_2\) would be paired, then, since \(p_3\) and \(p_4\) already played, player \(p_4\) would float to a match with a player with score 1, which implies that no match within the group with score 1 is possible.

Example 2.2. For comparison, we now apply Dutch BBP to the same instance from Example 2.1 and display the calculated player pairings in Figure 6. Match results in
the final round are not displayed, as they do not influence the pairings. Even though we copied as many match results as possible from Example 2.1, the two engines calculate largely different player pairings.

![Figure 5: Round 4 pairing of the example tournament.](image)

### Table: Player pairings for the tournament from Example 2.1 calculated via Dutch BBP

| round 1 | round 2 | round 3 | round 4 |
|---------|---------|---------|---------|
| player | rank | score | cd | color | player | rank | score | cd | color | player | rank | score | cd | color | player | rank | score | cd | color |
| p1 | 2 | 0 | 0 | b | p1 | 2 | 1 | +1 | w | p1 | 1 | 1 | +1 | b | p1 | 1 | 3 | +1 | b |
| p2 | 3 | 0 | 0 | w | p3 | 3 | 1 | −1 | w | p2 | 3 | 0 | 0 | w | p2 | 2 | 2 | +2 | b |
| p3 | 4 | 0 | 0 | b | p4 | 4 | 1 | +1 | b | p3 | 4 | 1 | 0 | b | p4 | 4 | 2 | −1 | w |
| p4 | 5 | 0 | 0 | b | p5 | 5 | 0 | −1 | b | p4 | 5 | 1 | 0 | b | p5 | 5 | 1 | −1 | w |
| p5 | 6 | 0 | 0 | w | p6 | 6 | 0 | −1 | w | p5 | 6 | 1 | 0 | w | p6 | 6 | 1 | +1 | w |
| p6 | 7 | 0 | 0 | b | p7 | 7 | 0 | −1 | w | p6 | 7 | 0 | 0 | b | p7 | 7 | 1 | +1 | b |
| p7 | 8 | 0 | 0 | w | p8 | 8 | 0 | +1 | b | p7 | 8 | 0 | 0 | w | p8 | 8 | 0 | −1 | b |

Figure 5. Round 4 pairing of the example tournament.

Figure 6. Player pairings for the tournament from Example 2.1 calculated via Dutch BBP. Match results are consistent with the corresponding results from Example 2.1.

The pairing in the first round is solely determined by the main Dutch pairing principle, and thus is identical to the pairing produced by our Dutch MWM engine. However, the color assignment is different, as in the match between $p_1$ and $p_5$, Dutch BBP assigns white to $p_1$, whereas Dutch MWM assigns white to $p_5$. This is to be expected, as both engines are supposed to draw the first-round color assignment randomly—however, as our tests with Dutch BBP revealed, the initial color assignment there is deterministic. In the first round, we gave the exact same match results to Dutch BBP as to Dutch MWM. Despite of this, from the second round on, player pairings calculated by the two engines are completely different. For those pairs that do not appear in Example 2.1 we calculated the outcome from Milvang’s probability distribution (Milvang, 2016), otherwise we kept the same match result as in Example 2.1.

A notable difference of both approaches for the same sample tournament is that Dutch MWM overall achieves a better color balance than Dutch BBP. In the latter one, we have color differences of 2 in round 3 whereas the color differences in Dutch
MWM never exceed 1. In Dutch MWM the players perfectly alternate between playing with white and black pieces.

3. Assumptions and Experimental Setup

In our simulations we assume that each player $p_i \in P$ has true playing strength $str(p_i)$ that is approximated by her Elo rating $Elo(p_i)$ and we treat both values as constant throughout the tournament. It is crucial that $str(p_i)$ and $Elo(p_i)$ might differ, as they are used in different components of our model, which we describe in the next two paragraphs.

The probabilities of match results and optimal rankings are defined by the playing strength. More precisely, each player’s playing strength is a random number drawn from an uniform distribution of values between 1400 and 2200. We also justified our claims on ranking quality using other realistic player strength distributions. We elaborate on these in the appendix. The results are in line with the results for the uniform distribution.

Elo ratings are used for computing $r(p_i)$ and for breaking ties in the final order. The Elo rating of player $p_i$ is randomly drawn from a normal distribution with mean $str(p_i)$ and standard deviation $\frac{3000-str(p_i)}{20}$. This function mirrors the assumption that a higher Elo rating estimates the strength more accurately.

To avoid the noise introduced by byes, we assume that the number of players $n$ is even. The number of rounds is chosen to lie between $\lceil \log_2 n \rceil$ and $\frac{n}{2}$, as at least $\lceil \log_2 n \rceil$ rounds ensure that a player who wins all matches is the sole winner and at most $\frac{n}{2}$ rounds ensures that, according to Dirac’s theorem (Dirac, 1952), a perfect matching always exists. The tiebreakers used for obtaining the final tournament ranking are based on the FIDE recommendation (FIDE, 2020, C.02.13.16.5).

**Computing the Maximum Weight Matching**

First we transform each edge weight given as a tuple to a rational number. In particular, $w(p_i, p_j)$ is transformed to $10000 \cdot (-|s(p_i) - s(p_j)|) + 100 \cdot (-|cd(p_i) + cd(p_j)|) + \pi(p_i, p_j)$. The factors 10000 and 100 ensure that each lexicographically maximum so-
olution corresponds to a maximum weight solution with the new weights and vice versa. We compute pairings using the LEMON Graph Library (Dezső, Jüttner, and Kovács 2011) implementation of the maximum weight perfect matching algorithm, which is based on the blossom algorithm of Edmonds (1965) and has the same time and space complexity (Kolmogorov 2009). The implementation we use has $O(nm \log n)$ time complexity, where $n$ is the number of players and $m$ is the number of edges in the constructed graph $G_r$.

Realistic Probabilistic Model for Match Results

The results of the individual matches are computed via a probabilistic model that is designed to be as realistic as possible. Match results are drawn at random from a suitably chosen probability distribution based on the players’ strength and on the assigned colors for that match. For this, we use the probability distribution proposed by Milvang (2016), which was featured in a recent news article of the FIDE commission System of Pairings and Programs (FIDE SPP Commission 2020). Milvang’s probability distribution was engineered via a Data Science approach that used real-world data from almost 4 million real chess matches from 50,000 tournaments. It is based on Elo ratings and color information, whereas we use true strength values instead of Elo ratings to get unbiased match result probabilities.

Using Milvang’s approach, the probability for a certain outcome of a match depends on the actual strengths of the involved players, not only on their strength difference. Draw probability increases with mean strength of the players. The probabilities also depend on colors, as the player playing with white pieces has an advantage. See Table 3 for some example values drawn from Milvang’s distribution.

| Player Strengths | Win White | Win Black | Draw |
|------------------|-----------|-----------|------|
| 1200 (w) vs 1400 (b) | 26 % | 57 % | 17 % |
| 2200 (w) vs 2400 (b) | 14 % | 55 % | 31 % |
| 2400 (w) vs 2200 (b) | 63 % | 11 % | 26 % |

Table 3.: Example match outcome probabilities drawn from Milvang’s probability distribution (Milvang 2016).
Measuring Ranking Quality

Ranking quality measures how similar the tournament’s final ranking is to the ranking that sorts the players by their strength. One popular measure for the difference between two rankings is the Kendall $\tau$ distance (Kendall, 1945). It counts the number of discordant pairs: pairs of elements $x$ and $y$, where $x < y$ in one ranking, but $y < x$ in the other. We use its normalized variant, where $\tau \in [-1, 1]$, and $\tau = 1$ means the rankings are identical, while $\tau = -1$ means one ranking is the inverse of the other. A higher Kendall $\tau$ is better, because it indicates a larger degree of similarity between the true and the output ranking.

We also justify our claims on ranking quality using two other well-known and possibly more sophisticated similarity measures, the Spearman $\rho$ distance (Spearman, 1904) and normalized discounted cumulative gain (NDCG). We elaborate on these measures and their behavior for our problem in the appendix. The results are in line with the ones derived for the Kendall $\tau$ distance.

Measuring Fairness

We measure fairness in terms of the two relaxable criteria of Swiss-system chess tournaments: (Q1) on the equal score of opponents and (Q2) on the color distribution balance. Adhering to (Q1) is measured by the number of float pairs, which equals the number of matches with opponents from different score groups throughout the tournament. We measure the absolute color difference of a round as the sum of color differences for all players: $acd = \sum_{p_i \in P} |cd(p_i)|$. Note that as $|cd(p_i)| \geq 1$ for all players after each odd round, $acd \geq n$ in those rounds.

Presentation of the Data

Data is presented in the form of violin plots (Hintze and Nelson, 1998), letter value plots (Hofmann, Wickham, and Kafadar, 2017), and scatter plots (Friendly and Denis, 2005). For violin plots, kernel density estimation is used to show a smoothed probability density function of the underlying distribution. Additionally, similar to box plots, quartiles are shown by dashed lines. Letter value plots are enhanced box plots that show more quantiles. Unlike violin plots, they are suitable for discrete values, as all
shown values are actual observations without smoothing.

Our plots compare the MWM implementation of the five pairing systems with the BBP implementation of Dutch.

4. Simulation Results

All simulations use the following parameters, unless noted otherwise:

- number of players $n$: 32
- number of rounds: 7
- strength range: between 1400 and 2200
- maximum allowed color difference $\beta$: 2
- sample size: 100 000

These values were chosen to be as realistic as possible, based on parameters of more than 320 000 real-world tournaments uploaded to the website [chess-results.com]^{2}

The experiments were run on a computer server using version 20.04.1 of the Ubuntu operating system. It is powered by 48 Intel Xeon Gold 5118 CPUs running at 2.3 GHz and 62.4 GiB of RAM. We emphasize that the standard real-life challenge at a tournament, that is, computing a single pairing via a maximum weight matching for a tournament round can be solved in a fraction of a second on a standard laptop.

4.1. Ranking Quality

The pairing system of a Swiss-system tournament has a major impact on the obtained ranking quality, as Figure 7 shows. Burstein and Random2 achieve the best ranking quality, followed by Dutch and Dutch BBP. Random has a worse ranking quality and Monrad performs by far the worst. For other strength ranges, Figure 8 shows consistent results. See also Tables 4 and 5 for the corresponding mean and median Kendall $\tau$ values.

Comparing Dutch to Dutch BBP shows that they behave very similarly, with slight advantage for Dutch. This is remarkable, since Dutch BBP is based on complex and

^{2}The data was kindly provided by Heinz Herzog, author of the FIDE-endorsed tournament manager Swiss-Manager [Herzog, 2020b] and chess-results.com [Herzog, 2020a].
Figure 7.: Ranking quality measured by normalized Kendall $\tau$. A higher value means a better ranking quality.

Figure 8.: Ranking quality measured by normalized Kendall $\tau$ for different strength ranges.

rigid declarative criteria that are time-tested, while Dutch is the output of our easy-to-understand, purely matching-based approach. Together with the performance of Burstein and Random2 this shows that more transparent pairing systems can outperform the state-of-the-art Dutch BBP in terms of ranking quality.

We provide additional experimental results on the ranking quality in the appendix. There we present consistent results also for fewer or more players, for other strength range sizes, and for different player strength distributions.

4.2. Reasons for High Ranking Quality

Here we elaborate on how our flexible maximum weight matching model enables us to detect the exact reason why certain pairing systems produce better rankings, which might help designing better pairing systems in the future. In particular, we provide experiments that shed light on why Burstein, Random2, Dutch, and Dutch BBP reach
| Strength Range | Burstein | Dutch BBP | Dutch | Random2 |
|----------------|----------|-----------|-------|---------|
| 1000 – 1800    | 0.624    | 0.586     | 0.588 | 0.607   |
| 1400 – 2200    | 0.671    | 0.633     | 0.634 | 0.654   |
| 1800 – 2600    | 0.721    | 0.685     | 0.686 | 0.706   |

Table 4.: Mean normalized Kendall $\tau$ values averaged over 100,000 simulated tournaments for each configuration as in Figure 8.

| Strength Range | Burstein | Dutch BBP | Dutch | Random2 |
|----------------|----------|-----------|-------|---------|
| 1000 – 1800    | 0.629    | 0.590     | 0.591 | 0.610   |
| 1400 – 2200    | 0.673    | 0.634     | 0.637 | 0.657   |
| 1800 – 2600    | 0.723    | 0.688     | 0.690 | 0.710   |

Table 5.: Median normalized Kendall $\tau$ values averaged over 100,000 simulated tournaments for each configuration as in Figure 8.

...a better ranking quality than Random and Monrad and why Burstein and Random2 outperform Dutch BBP.

In order to rank players correctly, their relative playing strength must be approximated from match results. We call a match result unforesen if a weaker player wins against a stronger opponent. Unforesen match results hinder the approximation of both players’ strengths, so pairing systems should aim to minimize the number of unforeseen match results. Figure 9a confirms this, by showing a strong negative correlation between the proportion of unforeseen results and ranking quality for Dutch.

Figure 9.: Observed correlations after seven rounds, paired with Dutch BBP.
BBP. A similar correlation can be observed for all pairing systems.

The probability of an unforeseen match result increases as the strength difference of paired players decreases because the outcome of those matches is less predictable. In general, a higher mean strength difference in a tournament lowers the number of unforeseen match results, which then leads to better ranking quality. Our results in Section 4.1 justify the observation that mean strength difference seems to be positively correlated with ranking quality, as mean strength difference is low when using Monrad, medium with Random, and high for Burstein, Random2 and Dutch/Dutch BBP.

However, when looking at results from Dutch BBP only, there is a small negative correlation instead, as Figure 9b shows. This is also true for Dutch, Burstein, and Random2. This seemingly unforeseen correlation can be explained as follows. A better ranking leads to a smaller mean strength difference for these pairing systems. In an optimal ranking, each player is in her correct score group, together with players of similar strength, so the mean strength difference will be low. However, in a suboptimal ranking, some players are in a score group that does not reflect their strength. Therefore, these players are either stronger or weaker compared to the other players in their score group, which results in higher mean strength difference.

Figure 10 shows empirical evidence for this effect: the pairing in round one is always the same, but unforeseen match results due to randomness lead to different rankings, which then determine the mean strength difference in round two. In our experiment, the same single randomly paired first round was played 10 000 times. Each time the ranking quality after round one and the mean strength difference of the Dutch BBP pairing for round two was recorded. For the analysis, we use the Pearson correlation coefficient that is a standard measure for the linear dependence between two variables. In our case, a negative Pearson correlation coefficient indicates that on average, a higher Kendall τ is observed together with a lower mean strength difference.

The problem with mean strength difference is that it does not take into account whether a low mean strength difference was the result of a pairing system’s choice or due to unfavorable score groups. This can be avoided by taking the maximum possible strength difference into account. For this, we define the normalized strength difference as the total strength difference divided by the maximum possible total
Figure 10.: The scatter plot (top) shows the correlation between normalized Kendall τ and mean strength difference. The violin plot (bottom) shows the distribution of Pearson correlation coefficients if that experiment is repeated for 1 000 different tournaments, whose first round was simulated 1 000 times.

For computing the normalized strength difference it is essential to calculate the maximum possible strength difference. For this, we again use our maximum weight matching approach, but this time with a pairing system that maximizes strength difference. In particular, we use a modification of our Burstein edge weights $w(p_i, p_j)$ where we set $\pi(p_i, p_j) := |str(p_i) − str(p_j)|$. Remember that $str(p_i)$ and $str(p_j)$ are the true strength values of players $i$ and $j$, respectively. Of course, this new pairing system requires knowledge of all true player strengths, so it cannot be used in realistic settings. We only use it as an analytical tool.
Figure 11 compares the normalized strength difference for Dutch BBP and for our maximum weight matching implementation of Burstein and Dutch, where our version of Burstein clearly beats Dutch BBP in ranking quality. Firstly, this figure shows a positive correlation between normalized strength difference and normalized Kendall $\tau$ for each of Burstein, Dutch, and Dutch BBP after seven rounds. Simulations with each of Random2, Random, and Monrad also indicate a similar positive correlation. Secondly, Figure 11 also demonstrates the positive correlation between the normalized strength difference and the ranking quality across pairing systems. In particular, Burstein clearly beats Dutch BBP in normalized strength difference and also in ranking quality. This correlation is true in general: considering all pairing systems, exactly the ones with a high normalized strength difference (Burstein, Random2, Dutch, Dutch BBP) lead to a good ranking quality, while the ones with medium and low normalized strength difference (Random and Monrad) lead to medium and low ranking quality.

To summarize, our flexible maximum weight matching model enabled us to detect the exact reason why certain pairing systems produce better rankings. Our surprising finding is that even though at first sight, a high mean strength difference seems to be the pivotal factor, it is actually a high normalized strength difference that results in a better ranking quality. This discovery might help designing better pairing systems in
the future.

4.3. Fairness

The highly complex pairing criteria of the FIDE were designed with a focus on two fairness goals phrased as quality criteria, (Q1): minimizing the number of float pairs and (Q2): minimizing the absolute color difference.

Criterion (Q1) is at the heart of Swiss-system tournaments as pairing players of equal score ensures well-balanced matches. As Figure 12 shows, Burstein, Dutch, and Random2 beat Dutch BBP in terms of the number of float pairs. In the appendix we show consistent results for other simulation parameters.

![Figure 12: Number of float pairs out of the $7 \cdot 16 = 112$ matches of the tournament. Recall that floating is often unavoidable due to the size of the score group. A lower number indicates a better implementation of criterion (Q1).](image)

Figure 12 focuses on criterion (Q2) and shows that an absolute color difference very similar to the one guaranteed by Dutch BBP can be achieved via our MWM engine. The pairing system Random even outperforms Dutch BBP in this regard. In the appendix, we provide additional experiments with different numbers of rounds and numbers of players that lead to consistent results.

Hence, our maximum weight matching approach with edge weights that prioritize matches within score groups and secondly optimize for color balance is on a par with the sophisticated official FIDE criteria for criterion (Q2) and it even outperforms them for criterion (Q1). Thus, our more transparent approach ensures the same color balance quality but achieves even fewer float pairs. Moreover, our approach also allows for a different trade-off between criteria (Q1) and (Q2) that does not affect the obtained
Figure 13.: Absolute color difference after 6 rounds. A lower \(acd\) means a better color distribution. Recall that a \(acd \geq n\) for each odd round, while \(acd = 0\) is possible after each even round.

4.4. Lower Maximum Allowed Color Difference

So far, for all our experiments we assumed that the maximum allowed color difference \(\beta\) equals 2, i.e., the difference of the number of matches played with white pieces and the number of matches played with black pieces is at most 2. This is in line with the official FIDE rules. However, due to the flexibility of our maximum weight matching approach, we can easily enforce an even stronger color difference constraint and observe the impact on the obtained ranking quality and the number of float pairs.

Interestingly, as Figure 14 shows, the obtained ranking quality is almost the same even if we look at the extreme case of setting \(\beta = 0.1\), which is equivalent to enforcing an alternating black-white sequence for all players. Notice that setting \(\beta\) to anything in the interval \((0, 0.5]\) implies that the absolute color difference is 0 for all even rounds and \(n\) for all odd rounds.

Naturally, the high ranking quality for a much more restricted \(\beta\) comes at a cost, which can be measured in the number of float pairs. The obtained number of float pairs is influenced by the maximum allowed color difference \(\beta\), because for higher \(\beta\) it is easier to fulfill criterion (Q1), i.e., to find suitable matches within the corresponding score group. In our experiments we investigate the increase in the number of float pairs when we assume the extreme case of \(\beta = 0.1\). Figure 15 shows that the number of float pairs increases for all pairing systems, compared to the case with \(\beta = 2\). However, the
Figure 14.: Ranking quality measured by normalized Kendall $\tau$. Results for $\beta = 0.1$ are shown in blue, results for $\beta = 2$ in orange.

increase is only moderate. This result offers a novel trade-off for tournament organizers: when using the MWM engine, they have the choice between keeping the number of floaters down at the cost of a standard color difference, as advised by FIDE, or they opt for slightly more float pairs, but can guarantee an alternating white-black color assignment for each player. The ranking quality is equally high in both variants.

Figure 15.: Number of float pairs for 7 rounds. Results for $\beta = 0.1$ are shown in blue, results for $\beta = 2$ in orange.

5. Conclusion

The experimental results of our MWM engine with Burstein or Random2 demonstrate that it is possible to outperform the state-of-the-art FIDE pairing criteria in terms of both ranking quality and fairness, i.e., criteria (Q1) and (Q2), with a novel efficient mechanism that is more transparent and intelligible to all participants. The direct comparison of our MWM Dutch engine versus Dutch BBP shows that even if the
same pairing system is used, MWM achieves the same ranking quality but is more powerful since it yields an improved fairness. We believe that the key to this is the direct formulation of the most important criteria as a maximum weight matching problem.

The only scenario for which we might advise against using our mechanism is when the arbiter has no access to a computing device. In order to manually produce pairings in our framework, the arbiter would need to calculate the edge weights and then execute Edmonds’ blossom algorithm. Instead, the FIDE (FIDE [2020] Chapter C.04.3.D) provides manually executable rules. However, these rules include exhaustive search routines that can make the execution very slow, i.e., highly exponential in the number of players (Biró et al., 2017). Therefore, the ill-fated arbiter has to choose between learning to execute Edmonds’ blossom algorithm and following a cumbersome exponential-time pairing routine. We remark that this latter routine is complex to the point that even pairing engines already endorsed by the FIDE make mistakes occasionally (GitHub, 2022).

A clear advantage of our mechanism is that it is easily extendable: as Random and Random2 already demonstrate, a new pairing system can be implemented simply by specifying how edge weights are calculated. Similarly, as we have also demonstrated, the color balance can be adjusted by simply changing the parameter $\beta$. By thinning out the edge set in our graph, we can also reach an alternating black-white sequence for each player instead of just minimizing the color difference in each round.

The flexibility of the maximum weight matching approach proved to be essential for uncovering the driving force behind the achieved high ranking quality: the normalized strength difference. Hence, our approach was not only valuable for computing better pairings but also in the analysis of the obtained ranking quality. Furthermore, the flexibility of the MWM engine likely allows to incorporate additional quality criteria like measuring fairness via the average opponent ratings. Also quality criteria of other games and sports tournaments organized in the Swiss system can be integrated into the model.

Last but not least, other fields using rankings derived from pairwise comparisons might also benefit from our work. A possible application area for our MWM approach
is the computation of pairings in a speed-dating type event. Also there a sequence of matchings must be computed, no pair should be matched repeatedly, and the goal is to match like-minded participants. Such events are organized at conferences and other networking occasions (Paraschakis and Nilsson, 2020), and also resemble mentor assignment preparation at universities (Guse, Schweigert, Kulms, Heinen, Martens, and Guse, 2016; Muurlink and Poyatos Matas, 2011) and trainee rotation schedules in medicine (Castaño and Velasco, 2020).

References

David Ross Appleton. 1995. May the best man win? *Journal of the Royal Statistical Society: Series D (The Statistician)* 44, 4 (1995), 529–538.

Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H. Chi, and Cristos Goodrow. 2019. Fairness in Recommendation Ranking through Pairwise Comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’19)*. Association for Computing Machinery, New York, NY, USA, 2212–2220. https://doi.org/10.1145/3292500.3330745

Jeremy Bierema. 2017. BBP Pairings, a Swiss-system chess tournament engine. https://github.com/BieremaBoyzProgramming/bbpPairings. Accessed: 2022-05-17.

Kostas Bimpikis, Shayan Ehsani, and Mohamed Mostagir. 2019. Designing dynamic contests. *Operations Research* 67, 2 (2019), 339–356.

Péter Biró, Tamás Fleiner, and Richárd Palincza. 2017. Designing chess pairing mechanisms. In *10th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications*. Department of Computer Science and Information Theory, Budapest University of Technology and Economics, Budapest, Hungary, 77–86.

Felix Brandt, Markus Brill, Hans Georg Seedig, and Warut Suksompong. 2018. On the structure of stable tournament solutions. *Economic Theory* 65, 2 (2018), 483–507.

Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. 2016. Introduction to computational social choice.

Felix Brandt and Felix A. Fischer. 2007. PageRank as a Weak Tournament Solution. In *Internet and Network Economics, Third International Workshop, WINE (Lecture Notes in Computer Science)*, Xiaotie Deng and Fan Chung Graham (Eds.), Vol. 4858. Springer, San
Fabián Castaño and Nubia Velasco. 2020. Exact and heuristic approaches for the automated design of medical trainees rotation schedules. Omega 97 (2020), 102107.

Krishnendu Chatterjee, Rasmus Ibsen-Jensen, and Josef Tkadlec. 2016. Robust draws in balanced knockout tournaments. In Proceedings of the 25th International Joint Conference on Artificial Intelligence. AAAI Press, New York City, New York, USA, 172–179.

Xi Chen, Paul N. Bennett, Kevyn Collins-Thompson, and Eric Horvitz. 2013. Pairwise Ranking Aggregation in a Crowdsourced Setting. In Proceedings of the Sixth ACM International Conference on Web Search and Data Mining (WSDM ’13). Association for Computing Machinery, New York, NY, USA, 193–202. https://doi.org/10.1145/2433396.2433420

László Csató. 2013. Ranking by pairwise comparisons for Swiss-system tournaments. Central European Journal of Operations Research 21, 4 (2013), 783–803.

László Csató. 2017. On the ranking of a Swiss system chess team tournament. Ann. Oper. Res. 254, 1-2 (2017), 17–36. https://doi.org/10.1007/s10479-017-2440-4

László Csató. 2021. Tournament design: How operations research can improve sports rules. Springer Nature, Switzerland.

Dmitry Dagaev and Alex Suzdaltsev. 2018. Competitive intensity and quality maximizing seedings in knock-out tournaments. Journal of Combinatorial Optimization 35 (2018), 170–188.

Balázs Dezső, Alpár Jüttner, and Péter Kovács. 2011. LEMON—an open source C++ graph template library. Electronic Notes in Theoretical Computer Science 264, 5 (2011), 23–45.

Gabriel Andrew Dirac. 1952. Some theorems on abstract graphs. Proceedings of the London Mathematical Society 3, 1 (1952), 69–81.

Jack Edmonds. 1965. Paths, Trees, and Flowers. Canadian Journal of Mathematics 17 (1965), 449–467. https://doi.org/10.4153/CJM-1965-045-4

Wilfried Elmenreich, Tobias Ibounig, and István Fehérvári. 2009. Robustness versus performance in sorting and tournament algorithms. Acta Polytechnica Hungarica 6, 5 (2009), 7–18.

Arpad E Elo. 1978. The rating of chessplayers, past and present. Arco Pub., London.

István Fehérvári and Wilfried Elmenreich. 2009. Evolutionary Methods in Self-organizing System Design. In Proceedings of the 2009 International Conference on Genetic and Evolutionary Methods, GEM 2009, July 13-16, 2009, Las Vegas Nevada, USA, Hamid R. Arabnia and Ashu M. G. Solo (Eds.). CSREA Press, 10–15.
FIDE. 2020. FIDE Handbook. [https://handbook.fide.com/] Accessed: 2022-05-17.

FIDE. 2023. FIDE Handbook, D. Regulations for Specific Competitions / 02. Chess Olympiad. [https://handbook.fide.com/chapter/OlympiadPairingRules2022] Accessed: 2023-07-26.

FIDE SPP Commission. 2020. Probability for the outcome of a chess game based on rating. https://spp.fide.com/2020/10/23/probability-for-the-outcome-of-a-chess-game-based-on-rating/.

Michael Friendly and Daniel Denis. 2005. The early origins and development of the scatterplot. Journal of the History of the Behavioral Sciences 41, 2 (2005), 103–130.

GitHub. 2022. Suboptimal exchange in remainder #7. https://github.com/BieremaBoyzProgramming/bbpPairings/issues/7.

Mark E Glickman and Shane T Jensen. 2005. Adaptive paired comparison design. Journal of Statistical Planning and Inference 127, 1-2 (2005), 279–293.

Sushmita Gupta, Sanjukta Roy, Saket Saurabh, and Meirav Zehavi. 2018. When rigging a tournament, let greediness blind you. In Proceedings of the 27th International Joint Conference on Artificial Intelligence. IJCAI, Stockholm, Sweden, 275–281.

Jennifer Guse, Eva Schweigert, Gerhild Kulms, Ines Heinen, Claudia Martens, and Andreas H Guse. 2016. Effects of mentoring speed dating as an innovative matching tool in undergraduate medical education: a mixed methods study. PLoS One 11, 2 (2016), e0147444.

Christine Harbring and Bernd Irlenbusch. 2003. An experimental study on tournament design. Labour Economics 10, 4 (2003), 443–464.

Robert J Henery. 1992. An extension to the Thurstone-Mosteller model for chess. Journal of the Royal Statistical Society Series D: The Statistician 41, 5 (1992), 559–567.

Heinz Herzog. 2020a. Chess-Results.com, the international Chess-Tournaments-Results-Server. [https://chess-results.com/] Accessed: 2021-12-07.

Heinz Herzog. 2020b. Swiss-Manager. [http://www.swiss-manager.at/] Accessed: 2021-12-07.

Jerry L Hintze and Ray D Nelson. 1998. Violin plots: a box plot-density trace synergism. The American Statistician 52, 2 (1998), 181–184.

Heike Hofmann, Hadley Wickham, and Karen Kafadar. 2017. Value plots: Boxplots for large data. Journal of Computational and Graphical Statistics 26, 3 (2017), 469–477.

Richard Hoshino. 2018. A Recursive Algorithm to Generate Balanced Weekend Tournaments. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32. AAAI Press, New
Olivier Hudry. 2009. A survey on the complexity of tournament solutions. *Mathematical Social Sciences* 57, 3 (2009), 292–303.

Robert Irving. 1985. An efficient algorithm for the “stable roommates” problem. *Journal of Algorithms* 6, 4 (1985), 577–595.

Alexander Karpov. 2018. Generalized knockout tournament seedings. *International Journal of Computer Science in Sport* 17, 2 (2018), 113–127.

Maurice G Kendall. 1945. The treatment of ties in ranking problems. *Biometrika* 33, 3 (1945), 239–251.

Michael P Kim, Warut Suksompong, and Virginia Vassilevska Williams. 2017. Who can win a single-elimination tournament? *SIAM Journal on Discrete Mathematics* 31, 3 (2017), 1751–1764.

Michael P Kim and Virginia Vassilevska Williams. 2015. Fixing tournaments for kings, chokers, and more. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. AAAI Press, New Orleans, USA, 561–567.

Vladimir Kolmogorov. 2009. Blossom V: a new implementation of a minimum cost perfect matching algorithm. *Mathematical Programming Computation* 1, 1 (2009), 43–67.

Bernhard Korte and Jens Vygen. 2012. Combinatorial Optimization: Theory and Algorithms.

Eija Kujansuu, Tuukka Lindberg, and Erkki Mäkinen. 1999. The stable roommates problem and chess tournament pairings. *Divulgaciones Matemáticas* 7, 1 (1999), 19–28.

Roel Lambers, Dries Goossens, and Frits CR Spieksma. 2023. The flexibility of home away pattern sets. *Journal of Scheduling* 26, 5 (2023), 413–423.

Jeffrey Larson, Mikael Johansson, and Mats Carlsson. 2014. An Integrated Constraint Programming Approach to Scheduling Sports Leagues with Divisional and Round-Robin Tournaments. In *Integration of AI and OR Techniques in Constraint Programming*, Helmut Simonis (Ed.). Springer International Publishing, Cham, 144–158.

Jean-François Laslier. 1997. *Tournament solutions and majority voting*. Springer, Berlin.

Otto Milvang. 2016. Probability for the outcome of a chess game based on rating. https://pairings.fide.com/images/stories/downloads/2016-probability-of-the-outcome.pdf.

Hervé Moulin. 1986. Choosing from a tournament. *Social Choice and Welfare* 3, 4 (1986), 271–291.

Olav Muurlink and Cristina Poyatos Matas. 2011. From romance to rocket science: Speed dating in higher education. *Higher Education Research & Development* 30, 6 (2011), 751–
Snjólfur Ólafsson. 1990. Weighted matching in chess tournaments. *Journal of the Operational Research Society* 41, 1 (1990), 17–24.

Dimitris Paraschakis and Bengt J. Nilsson. 2020. Matchmaking Under Fairness Constraints: A Speed Dating Case Study. In *Bias and Social Aspects in Search and Recommendation*, Ludovico Boratto, Stefano Faralli, Mirko Marras, and Giovanni Stilo (Eds.). Springer International Publishing, Cham, 43–57.

Christian Saile and Warut Suksompong. 2020. Robust bounds on choosing from large tournaments. *Social Choice and Welfare* 54 (2020), 87–110.

Philip Scarf, Muhammad Mat Yusof, and Mark Bilbao. 2009. A numerical study of designs for sporting contests. *European Journal of Operational Research* 198, 1 (2009), 190–198.

Zilla Sinuany-Stern. 1988. Ranking of sports teams via the AHP. *Journal of the Operational Research Society* 39, 7 (1988), 661–667.

Charles Spearman. 1904. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology* 15, 1 (1904), 72–101.

Isabelle Stanton and Virginia Vassilevskia Williams. 2011. Manipulating Stochastically Generated Single-Elimination Tournaments for Nearly All Players. In *Internet and Network Economics - 7th International Workshop, WINE (Lecture Notes in Computer Science)*, Ning Chen, Edith Elkind, and Elias Koutsoupias (Eds.), Vol. 7090. Springer, Singapore, 326–337. [https://doi.org/10.1007/978-3-642-25510-6_28](https://doi.org/10.1007/978-3-642-25510-6_28)

Balázs R Sziklai, Péter Biró, and László Csató. 2022. The efficacy of tournament designs. *Computers & Operations Research* 144 (2022), 105821.

David Van Bulck and Dries Goossens. 2019. Handling fairness issues in time-relaxed tournaments with availability constraints. *Computers & Operations Research* 115 (11 2019), 104856. [https://doi.org/10.1016/j.cor.2019.104856](https://doi.org/10.1016/j.cor.2019.104856)

Tray Minh Voong and Michael Oehler. 2019. Auditory spatial perception using bone conduction headphones along with fitted head related transfer functions. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 1211–1212.

Lan Wei, Yonghong Tian, Yaowei Wang, and Tiejun Huang. 2015. Swiss-System Based Cascade Ranking for Gait-Based Person Re-Identification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI’15)*. AAAI Press, 1882–1888.

Wikipedia. 2023. Swiss-system tournament – Pairing procedure. [https://en.wikipedia.org/wiki/Swiss-system_tournament](https://en.wikipedia.org/wiki/Swiss-system_tournament) Accessed: 2023-08-06.
Appendix A. Ranking Quality

In the following we discuss additional simulation experiments that measure the obtained ranking quality for various parameter settings.

A.1. Different Tournament Sizes

We start with experimental results demonstrating that our findings on the ranking quality remain valid for tournaments of different sizes in terms of number of players and number of rounds.

Usually it is expected that a player who wins all matches also wins the tournament, without being tied for the first place. This can be ensured by playing at least $\lceil \log_2 n \rceil$ rounds: four rounds for 16 players, five rounds for 32 players and six rounds for 64 players. Most tournaments are five or seven rounds long, according to data from chess-results.com (Herzog, 2020a).

In general, more rounds lead to higher ranking quality, although with diminishing effect, as Figure A1 shows. In terms of the achieved ranking quality, the MWM engine with Burstein outperforms Dutch BBP in all cases, except for the unrealistic case of a tournament with only two rounds.

A.2. Different Strength Range Sizes

Here we vary the used strength range size, i.e., we sample the player strengths from different intervals. A smaller strength range size corresponds to a tournament among players with similar strength and larger strength range sizes model tournaments with more heterogeneous players. The results depicted in Figure A2 show that also for different strength range sizes the MWM engine with Burstein or Random2 outperforms Dutch BBP in terms of ranking quality and that Dutch is on a par with Dutch BBP.

A higher strength range size results in higher ranking quality and less variance. The increasing ranking quality can be explained by a higher mean strength difference, which results from a larger strength range size. Variance decreases, because match results become more predictable.

The difference in ranking quality between Burstein and Dutch BBP is much
Figure A1.: Ranking quality after 1-9 rounds, 32 or 64 players with strength range 1400-2200. Results for Burstein are shown in blue, Dutch BBP results are shown in orange.

higher for a strength range size of 400 compared to 800 and 1200. For small strength range sizes in all Dutch BBP paired matches it is more likely that a weaker player wins against a stronger opponent, while for Burstein at least some matches are still predictable.

A.3. Different Player Strength Distributions

We provide additional experimental results that indicate that our findings hold independently of the employed player strength distributions, i.e., we get the same behavior also for non-uniform distributions. Since no data is available that let’s us estimate how realistic player strength distributions look like, we focus on several natural candidates that deviate strongly from uniform distributions.

First, we considered player strength distributions that are derived from exponential distributions. For this, we consider in Figure A3 a case with many strong players
Figure A2.: Ranking quality measured by normalized Kendall τ for different strength range sizes.

and only a few weak players and in Figure A4 a case with many weak players and only a few strong players within the given strength range size. We also considered player

Figure A3.: Ranking quality measured by normalized Kendall τ for 32 players with an exponential player strength distribution in the range \([1400, 2200]\) with mean at 2000.

strength distributions derived from a normal distribution with a mean exactly in the middle of the strength range size and a standard deviation of a fourth of the strength range size. See Figure A5 for the corresponding results.

Finally, we investigated a player strength distribution that is derived from uniformly sampling player strengths from the real-world distribution of Elo scores of all 363,275 players listed by FIDE\(^3\) restricted to the desired strength range. Figure A6 shows also very similar results for this case.

A.4. Ranking Quality via Spearman ρ and NDCG

For comparison reasons, we provide an evaluation of the achieved ranking quality via the Spearman ρ and the normalized discounted cumulative gain (NDCG) measures.

\(^3\)See [https://ratings.fide.com/download_lists.phtml](https://ratings.fide.com/download_lists.phtml) for details.
Figure A4.: Ranking quality measured by normalized Kendall $\tau$ for 32 players with an exponential player strength distribution in the range $[1400, 2200]$ with mean at 1600.

Figure A5.: Ranking quality measured by normalized Kendall $\tau$ for 32 players with a normally distributed player strength distribution in the range $[1400, 2200]$ with mean at 1800 and standard deviation of 200.

Figure A6.: Ranking quality measured by normalized Kendall $\tau$ for 32 players uniformly sampled from the real-world distribution of Elo scores restricted to the range $[1400, 2200]$.

Besides Kendall $\tau$, Spearman $\rho$ is commonly used for comparing rankings. Here, we use a normalized variant of Spearman $\rho$, similar to the normalized Kendall $\tau$.

The NDCG measure is not commonly used for comparing rankings. It is used to evaluate search engines, by assigning a relevance rating to documents and awarding a higher score if highly relevant documents are listed early. Applied to our case, NDCG puts an emphasis on ranking the top players correctly, while ranking the lowest ranked
players correctly is basically irrelevant.

As shown in Figure A7 and Figure A8, the results with normalized Spearman $\rho$ and NDCG look almost identical to the results for normalized Kendall $\tau$ in Figure 7. Also, for different strength ranges or range sizes we get consistent results, see

![Figure A7: Ranking quality measured by normalized Spearman $\rho$.](image1)

Figure A8: Ranking quality measured by the normalized discounted cumulative gain (NDCG).

![Figure A9: Ranking quality measured by normalized Spearman $\rho$.](image2)

Figures A9, A10, A11 and A12.
Figure A10.: Ranking quality measured by the normalized discounted cumulative gain (NDCG).

Figure A11.: Ranking quality measured by normalized Spearman $\rho$.

Figure A12.: Ranking quality measured by the normalized discounted cumulative gain (NDCG).

Appendix B. Fairness

Here we present additional simulation results that measure the achieved fairness, i.e., results regarding the compliance with the quality criteria (Q1) and (Q2).
B.1. Number of Float Pairs

We consider the obtained number of float pairs for different strength ranges and different strength range sizes. Figures B1 and B2 show that we get consistent results for different strength ranges and different strength range sizes. Burstein has by far the lowest number of float pairs, but also Random2 and Dutch perform slightly better than Dutch BBP.

![Figure B1: Number of float pairs for different strength ranges.](image1)

![Figure B2: Number of float pairs for different strength range sizes.](image2)

Figure B3 shows a direct comparison of the obtained number of float pairs for Burstein and Dutch BBP for different numbers of players and different tournament lengths.

Also here we consistently get that Burstein achieves much fewer float pairs than Dutch BBP.

B.2. Absolute Color Difference

The measured absolute color difference increases slightly with the number of rounds and also with the number of players, as Figure B4 shows.

Note that in every odd round, the absolute color difference must be at least $n$, which can also be seen. All investigated pairing systems almost always meet this lower
Figure B3.: Number of float pairs for different tournament sizes and lengths. The results for Burstein are shown in blue, results for Dutch BBP in orange.

bound for odd rounds. Interestingly, Dutch BBP seems to perform slightly better in tournaments with at most 4 rounds compared to Burstein, but this tiny advantage vanishes for at least six rounds. We get similar results when comparing with Random2, Dutch, Random, and Monrad.
Figure B4.: Absolute color difference in rounds 1-9, 16-64 players with strength range 1400-2200. Results for Burstein are shown in blue, Dutch BBP results are shown in orange.