Fair Bubble Sort: Provably Optimal Fair Ranking with Continuous Sensitive Attributes

Antonio Ferrara
Intesa Sanpaolo AI Research
Turin, Italy
antonio.ferrara@intesasanpaolo.com

Fabio Vitale
Intesa Sanpaolo Innovation Center
Turin, Italy
fabio.vitale@intesasanpaolo.com

Atsushi Miyauchi
Intesa Sanpaolo
Turin, Italy
atsushi.miyauchi@intesasanpaolo.com

Francesco Bonchi
Intesa Sanpaolo AI Research
Turin, Italy
francesco.bonchi@intesasanpaolo.com

Abstract

Most fair-ranking methods assume categorical protected attributes, requiring continuous attributes such as age or income to be discretized into groups. This introduces arbitrary thresholds, loses fine-grained information, and ignores the natural ordering of the attribute. In this paper, we study the problem of fair ranking with continuous protected attributes without relying on thresholding. We introduce Continuous Pairwise Parity (CPP), a fairness notion based on Kendall correlation between ranking positions and the protected attribute, and seek rankings that satisfy a prescribed CPP constraint while minimizing their Kemeny distance from an initial score-based ranking. To solve this problem, we propose Fair Bubble Sort (FBS), an efficient post-processing algorithm based on targeted adjacent swaps. We prove that FBS is exactly optimal for the unweighted Kemeny distance and provide approximation guarantees for the score-weighted setting. FBS is output-sensitive, with running time determined by the number of swaps required to meet the fairness constraint. Experiments on synthetic and real-world datasets show that FBS is scalable and consistently outperforms discretization-based baselines, achieving a superior fairness-accuracy Pareto front.

1 Introduction

Algorithmic ranking systems increasingly influence decisions in high-stakes domains ranging from hiring to criminal justice, from credit scoring to college admission. As a consequence, fairness aspects in algorithmic ranking have attracted growing attention [18, 28, 31, 41, 30]. Much of such literature deals with *group fairness* along the lines of *demographic parity* [9] or *equal opportunity* [16], i.e., requiring the same chances of a positive outcome for different demographic groups, as defined by protected attributes, such as gender or race. In the case of ranking, this is typically cast as the problem of maximizing utility under a *group-fairness constraint*, which requires a minimum fraction of individuals from each group to be included in the top-k positions [39, 5, 34]. The very notion of groups implies an underlying categorical attribute. However, in many real-world scenarios, sensitive attributes are inherently continuous, such as age, income, or health scores. When dealing with such variables, the standard practice of discretizing them into groups introduces arbitrary thresholds, discards fine-grain information about the population (individuals falling into the same bucket are treated equivalently), and destroys the natural ordinality of the data (e.g., the fact that the bucket of age [20,30) comes after the bucket [13,20) and before the [30,45) one). Despite recent progress on continuous sensitive attributes in classification and regression, fair ranking with continuous attributes

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

remains largely unexplored. The core challenge lies in the combinatorial nature of rankings: standard continuous fairness regularizers cannot be trivially applied to listwise utility metrics or discrete ranking permutations.

In this paper, we study the problem of producing a ranking that is as close as possible to a given score-based ranking while satisfying a fairness constraint defined directly on a continuous protected attribute, without any discretization. Our starting point is a pairwise view of fairness: in a fair ranking, an item with a larger protected-attribute value should be no more likely to rank above another item than below it. We capture this principle through **Continuous Pairwise Parity (CPP)**, defined using the Kendall correlation between ranking positions and the protected attribute. CPP requires no discretization, preserves the ordinal information in the attribute, and reduces to pairwise statistical parity for binary protected attributes.

Given a score-based ranking, our objective is to find the closest ranking satisfying a prescribed CPP constraint. We measure distortion using the Kemeny distance, as it mirrors the pairwise structure of the fairness notion, decomposing the utility loss as a sum over pairs of items whose order in the produced ranking disagrees with the score-based ranking. Weighting each pair by their difference in score further captures the intuition that swapping items of near-identical merit is essentially costless, while swapping items of very different scores is expensive. This shared pairwise decomposition of cost and fairness is precisely what allows a single algorithmic primitive – the adjacent swap – to navigate both quantities, and is the structural feature on which our theoretical guarantees rest (see Appendix B for an extended discussion motivating Continuous Pairwise Parity).

Paper Contributions. The main contributions of this paper are summarised as follows:

1. We introduce the fairness notion of Continuous Pairwise Parity (CPP), extending *pairwise statistical parity* [29] to continuous protected attributes.
2. We propose **Fair Bubble Sort (FBS)**, the first theoretically grounded post-processing algorithm that optimally navigates the utility-fairness trade-off for continuous attributes via targeted adjacent swaps.
3. We prove that FBS is *exactly optimal* for the unweighted Kemeny distance (Theorem 1). For the weighted case, we prove an asymptotic ratio of $(1+\sqrt{2})/2 \approx 1.21$ for integer scores in the all-ascending attribute case with n a power of two (Proposition 3). Extensive empirical evidence further suggests that FBS may satisfy a general 2-approximation bound.
4. We show that FBS is highly scalable, running in $O(n \log n + K^*)$ time for the unweighted case and in $O\big((n + K^*) \log n\big)$ time for the weighted case, where K^* is the minimum number of adjacent swaps required to satisfy the fairness constraint (Propositions 4).
5. We validate FBS on four real-world datasets, empirically demonstrating its scalability and its superior performance over non-trivial baseline models.

2 Related Work

Our work sits at the intersection of algorithmic fair ranking and the treatment of continuous sensitive attributes, bridging two traditionally separate lines of research.

Fair Ranking. The literature on fair ranking has grown significantly, driven by the need to mitigate historical biases in search, recommendation, selection, and allocation systems [18, 28, 31, 41, 30]. With a few exceptions that focus on individual fairness [9, 3, 33], or that tackle individual and group fairness jointly [12, 40], vast majority of the approaches formulate fairness as a group-level constraint, assuming that the protected attribute is categorical (e.g., binary gender or discrete race). Solutions typically enforce variants of demographic parity or proportional representation, ensuring that a minimum fraction of items from each group appears in the top-k positions or in every prefix of the ranking [38, 5, 39, 34, 37, 2, 13]. Conceptually closest to our work is the notion of *pairwise fairness* [25, 29], which evaluates the probability of a protected candidate being ranked above a non-protected one. A fundamental limitation of all these approaches is their reliance on predefined protected groups, requiring either categorical protected attributes or discrete bins. When dealing with inherently continuous protected attributes, enforcing these metrics requires arbitrary thresholding, which discards fine-grained intra-group variance and destroys the ordinal nature of the data.

Continuous Sensitive Attributes. In the broader machine learning research, the flaws of discretizing continuous sensitive attributes have been recently recognized, primarily within the contexts of regression and classification. Seminal works have proposed matching continuous distributions or penalizing statistical dependence to enforce fairness. For instance, [27] introduced fairness-aware learning using the Rényi maximum correlation coefficient, while [24] formulated non-convex optimization techniques for fair regression. Another prominent line of research leverages causal inference, where continuous sensitive attributes are naturally treated as nodes in structural equation models. In this context, [26] introduced *counterfactual fairness*, ensuring that predicted distributions remain invariant under counterfactual changes to the continuous protected attribute, a framework later extended by [6] to isolate path-specific fair effects. Alternatively, frameworks based on *individual fairness* [9] accommodate continuous spaces by requiring that similar individuals receive similar predictions. While mathematically robust, these continuous regularizers, segmentation strategies, and causal frameworks are designed for differentiable loss functions and point-wise predictions, making them inapplicable to the discrete, combinatorial nature of listwise ranking systems.

Bridging the Gap. The specific intersection of fair ranking and continuous protected attributes is still largely unexplored. While recent preliminary work by Giuliani et al. [14] has highlighted the necessity of moving beyond discrete groups by exploring continuous attributes within long-term, dynamical allocation systems, the fundamental algorithmic challenge of single-shot, static fair ranking remains unresolved. To the best of our knowledge, we propose the first theoretically grounded post-processing algorithm to enforce fairness over continuous attributes in static ranking.

3 Preliminaries

Let $U = [n]$ be a set of n items and S_U be the symmetric group of all permutations of U. A ranking r is an element of S_U, i.e., a bijection $r : U \to [n]$, where $r(u)$ denotes the position of item u in the ranking. Each item $u \in U$ has a continuous **sensitive attribute** $a(u) \in \mathbb{R}_+$ and a **score** $s(u) \in \mathbb{R}_+$. The score $s(u)$ indicates the overall fitness/utility of u for the task at hand: it might be, e.g., the results of a test in a college admission, or the output of a complex machine learning model in a hiring setting. We assume higher score s is better. Ranking simply by s is not considered adequate in the fairness literature, as s might encode historical biases, which fair ranking aims to fix by producing a ranking that is as close as possible to the one induced by s while meeting the given fairness criteria. We denote r_s the "unfair" **score-based ranking** obtained by sorting items in descending order of s.

Kemeny distance. We denote by $\mathrm{Inv}(r_s, r) := \{(i,j) \mid 1 \le i < j \le n,\, (r_s(j) - r_s(i))(r(i) - r(j)) > 0\}$ the inversion set of r w.r.t. r_s, the item pairs whose relative order in r differs from that in r_s. We denote by $d(r_s, r)$ the Kemeny distance[1] between r_s and an alternative ranking r, defined as

$$d(r_s, r) = \sum_{(i,j) \in \mathrm{Inv}(r_s, r)} w_{ij},$$

where $w_{ij} \ge 0$ is a non-negative weight assigned to each pair. While the proposed algorithm and all theoretical results hold for any weight function $w_{ij} \ge 0$, in this paper we consider two cases: the *unweighted* case with uniform weights $w_{ij} = 1/\binom{n}{2}$ and the *score-gap weighted* case with $w_{ij} = |s(i) - s(j)|/W$, where $W = \sum_{1 \le k < l \le n} |s(k) - s(l)|$ is a normalization constant. Both choices yield $d \in [0, 1]$, but in the score-gap weighted case swapping items with closer scores is cheaper than swapping items with further scores.

Continuous Pairwise Parity (CPP). Let $N_\nearrow(r) := \sum_{1 \le i < j \le n} \mathbb{1}\big[(r(j) - r(i))(a(j) - a(i)) > 0\big]$ and $N_\searrow(r) := \sum_{1 \le i < j \le n} \mathbb{1}\big[(r(j) - r(i))(a(j) - a(i)) < 0\big]$ denote the number of *concordant* and *discordant* pairs between the ranking and the attribute values, respectively. The *Kendall-τ correlation* between a ranking r and the attribute a is $\tau(r, a) = \frac{N_\nearrow(r) - N_\searrow(r)}{\binom{n}{2}}$. A ranking is fair when this correlation is close to zero. Since $\tau(r, a) \in [-1, 1]$, we define the CPP fairness measure as

[1]The number of pairs that two rankings order oppositely is also known as the Kendall-τ distance [10]; Kemeny and Snell [21] derived this distance from a set of axioms, and the related term "Kemeny ranking" denotes the ranking that minimizes the total distance to a given set of rankings [7]. Our $d(r_s, r)$ is a normalized version of this distance in which each inverted pair carries a weight. We call it the Kemeny distance to keep it distinct from the Kendall-τ *correlation* that defines the fairness measure below.

$\phi(r) := 1 - |\tau(r, a)|$, so that $\phi(r) = 1$ indicates perfect fairness and $\phi(r) = 0$ maximal unfairness. CPP needs no binning of the attribute, uses only its order, and decomposes over pairs in the same way as the Kemeny distance; when the attribute is binary, $\phi(r) = 1$ is exactly the pairwise statistical parity of [25] (Appendix B).

Problem 1. *Given the set of n items U, their sensitive attribute a and score s, given the weighted or unweighted Kemeny distance $d(\cdot, \cdot)$ and the CPP fairness measure $\phi(\cdot)$, as defined above, find the ranking r that minimizes the distance from the score-based ranking r_s, subject to the fairness constraint $\phi(r) \geq \psi$, where $\psi \in [0, 1]$ is a predefined fairness threshold:*

$$\operatorname*{arg\,min}_{r \in S_U \,:\, \phi(r) \geq \psi} \quad d(r_s, r). \tag{1}$$

Remark. Throughout the paper we assume that s and a are both injective, i.e. no two items share the same score or the same attribute value. This is without loss of generality: whenever $s(u) = s(v)$, the pair $\{u, v\}$ contributes $|s(u) - s(v)| = 0$ to the weighted Kemeny distance, so swapping them is free; whenever $a(u) = a(v)$, the pair contributes to neither $N_{\nearrow}$ nor $N_{\searrow}$, so their ordering does not affect ϕ. Ties can therefore be broken by an arbitrary total order, without losing the theoretical results presented in the paper (see detailed discussion at the end of Section 4).

4 Method

Both our notions of fairness and distance from the score-based ranking are defined in terms of pairwise comparisons. This shared structure suggests a natural algorithmic primitive: the *adjacent swap*. Swapping two items at consecutive positions $k, k{+}1$ flips the orientation of exactly one pair – it either creates or resolves a single inversion with respect to r_s, and it changes the difference $N_{\nearrow} - N_{\searrow}$ by exactly ± 2 (Observation 1). No other pair is affected. This locality makes adjacent swaps the atomic unit of both cost and fairness progress: each swap incurs the smallest possible perturbation to the ranking while making a discrete, predictable step toward the fairness target. In the following, for convenience, we refer to $\operatorname{sign}(N_{\nearrow} - N_{\searrow})$ as the *majority sign* and we work with the *imbalance* $\Delta(r) := |N_{\nearrow}(r) - N_{\searrow}(r)|$, so that the constraint $\phi(r) \geq \psi$ is equivalent to

$$\Delta(r) \leq \Delta^* := \left\lfloor \tfrac{n(n-1)}{2}(1 - \psi) \right\rfloor.$$

The key design insight is that, when $\Delta(r) > \Delta^*$, there always exists at least one adjacent pair of the majority sign whose swap reduces Δ by 2 (Lemma 3). A greedy strategy that repeatedly picks such a swap is therefore always well defined; moreover, since each swap reduces Δ by exactly 2, the total number of swaps K^* is determined by the initial imbalance $\Delta_0 := \Delta(r_s)$ and the threshold alone. Note that Δ retains the parity of Δ_0; the algorithm therefore sets $\Delta^* \leftarrow \max\big(\Delta^*,\, \Delta_0 \bmod 2\big)$ to guarantee feasibility. Every swap made by FBS improves fairness and increases the distance from r_s (see the proof of Theorem 1), so FBS never moves back towards r_s. One may ask whether this costs optimality, that is, whether some optimal ranking can only be reached by paths that at some point move back towards r_s. Lemma 2 in Section 5 shows that it does not: every optimal ranking is reachable from r_s by adjacent swaps along which the distance from r_s never decreases.

The resulting algorithm, dubbed **Fair Bubble Sort (FBS)**, is provided in Algorithm 1 for the unweighted Kemeny distance. The pseudocode for the weighted case is instead given in Algorithm 2 in Appendix C. Like classical bubble sort, our method operates exclusively via adjacent swaps, selecting at each step a swap that improves fairness. In the pseudocode we use uppercase arrays indexed by position: $R[i]$ is the item at position i and $A[i]$ its attribute (i.e., R stores the ranking in one-line notation, the inverse of the function $r : U \to [n]$ defined in Section 3). Lines 1–6 initialise the algorithm. The arrays are initialized according to the score-based ranking r_s (descending order of score, line 1). The sign array σ records the orientation of each adjacent pair (line 2), and the global counts $N_{\nearrow}$, $N_{\searrow}$ are computed in $O(n \log n)$ via a merge-sort–based count (line 3). The parity-safe threshold Δ^* is derived from ψ (line 4), the majority sign μ identifies which type of pair is in excess (line 5), and the linked list L collects all positions eligible for swapping (line 6).

The main loop (lines 7–24) iterates until the imbalance $\Delta(r)$ falls within the threshold Δ^*. At each iteration, an arbitrary eligible position k^* is retrieved from L in $O(1)$ (line 8) and the corresponding adjacent pair is swapped (line 9). The for-loop (lines 10–22) performs the local update: it examines the three sign entries that may have changed – $\sigma[k^*{-}1]$, $\sigma[k^*]$, $\sigma[k^*{+}1]$ – and updates L accordingly. The global counts are updated in $O(1)$ (line 23).

Algorithm 1 Fair Bubble Sort (FBS) – unweighted Kemeny distance

Require: Items $U = [n]$, attribute $a : U \to \mathbb{R}_+$, score $s : U \to \mathbb{R}_+$, fairness threshold $\psi \in [0, 1]$
Ensure: Ranking R solving Problem 1 with the unweighted Kemeny distance

$(R, A) \leftarrow$ arrays initialized according to r_s (descending order of score) ▷ initial r_s
$\sigma[k] \leftarrow \mathrm{sign}(A[k+1] - A[k])$ for $k = 1, \ldots, n-1$ ▷ adjacent-pair signs
$N_{\nearrow} \leftarrow |\{i < j : A[i] < A[j]\}|$; $N_{\searrow} \leftarrow |\{i < j : A[i] > A[j]\}|$
$\Delta^* \leftarrow \max\big(\lfloor (1-\psi)\, n(n-1)/2 \rfloor,\ |N_{\nearrow} - N_{\searrow}| \bmod 2\big)$ ▷ parity-safe threshold
$\mu \leftarrow \mathrm{sign}(N_{\nearrow} - N_{\searrow})$ ▷ majority sign
$L \leftarrow$ doubly-linked list of $\{k : \sigma[k] = \mu\}$ ▷ build in $O(n)$
while $|N_{\nearrow} - N_{\searrow}| > \Delta^*$ **do**
    $k^* \leftarrow$ any element of L ▷ $O(1)$
    swap $R[k^*] \leftrightarrow R[k^*+1]$, $A[k^*] \leftrightarrow A[k^*+1]$
    for $\ell \in \{k^*-1,\ k^*,\ k^*+1\} \cap \{1, \ldots, n-1\}$ **do**
        $\sigma_{\mathrm{old}} \leftarrow \sigma[\ell]$
        if $\ell = k^*$ **then**
            $\sigma[\ell] \leftarrow -\sigma[\ell]$ ▷ middle sign flips
        else
            $\sigma[\ell] \leftarrow \mathrm{sign}(A[\ell+1] - A[\ell])$
        end if
        if $\sigma_{\mathrm{old}} = \mu$ **and** $\sigma[\ell] \neq \mu$ **then**
            remove ℓ from L ▷ $O(1)$
        else if $\sigma_{\mathrm{old}} \neq \mu$ **and** $\sigma[\ell] = \mu$ **then**
            insert ℓ into L ▷ $O(1)$
        end if
    end for
    $N_{\nearrow} \leftarrow N_{\nearrow} - \mu$; $N_{\searrow} \leftarrow N_{\searrow} + \mu$
end while
return R

Unweighted vs. weighted Kemeny distance. Algorithm 1 presents FBS for the unweighted Kemeny distance, where every swap has unit cost. Since the total cost depends only on the *number* of swaps K^* and not on which majority-sign pairs are selected (Theorem 1), any majority-sign adjacent pair is equally good, and a doubly-linked list L suffices to maintain the pool of eligible positions in $O(1)$ per swap. When the Kemeny distance is weighted by score gaps $w_{ij} = |s(i) - s(j)|$, different swaps have different costs and FBS must select the cheapest one at each step. The algorithm additionally maintains a score array $S[i]$; the linked list is replaced by a min-heap keyed by the adjacent cost $c[k] = |S[k] - S[k+1]|$, yielding an $O((n + K^*) \log n)$ algorithm (the full pseudocode is given in Algorithm 2 in Appendix C).

Handling ties in the attribute. Both algorithms (Algorithms 1 and 2) are presented, for simplicity sake, under the general-position assumption that attribute values are pairwise distinct. In practice, real-valued attributes may contain ties (e.g. integer-valued age or discretised scores). Two modifications suffice to handle ties correctly; the algorithm structure, the greedy optimality proof, and the complexity bounds are otherwise unchanged.

1. **Adjacent signs.** The sign array becomes $\sigma[k] \in \{+1, 0, -1\}$, where $\sigma[k] = 0$ when $A[k] = A[k+1]$. Swapping a tied pair does not change the orientation of that pair—it was tied before and remains tied after—so $\Delta(r)$ is unaffected. Positions with $\sigma[k] = 0$ are therefore never inserted into L (or H) and are never selected for swapping.
2. **Neighbour updates.** After a swap at position k^*, the recomputed sign $\sigma[\ell]$ for a neighbour $\ell \in \{k^*-1,\ k^*+1\}$ may transition to or from 0. If $\sigma[\ell]$ was μ and becomes 0, the entry is removed from L (or H); if it was 0 and becomes μ, it is inserted. The middle entry $\sigma[k^*]$ flips sign as before ($+1 \leftrightarrow -1$) and cannot become 0, since the swap reverses a strict inequality.

All remaining logic – the imbalance $\Delta(r) = |N_{\nearrow} - N_{\searrow}|$, the threshold Δ^*, the majority sign μ, and the $O(1)$ update of $N_{\nearrow}, N_{\searrow}$ per swap – carries over without change.

5 Optimality of FBS for the Unweighted Problem

The proof of optimality builds on four results: (i) an adjacent swap changes the imbalance by exactly ± 2 and the number of inversions by exactly ± 1 (Observation 1); (ii) any ranking can be realized by adjacent swaps only, with the shortest sequence of swaps having length equal to $|\mathrm{Inv}(r_s, r)|$ (Lemma 1); (iii) a valid majority-sign swap always exists when the imbalance exceeds the threshold (Lemma 3); (iv) FBS uses the minimum number of swaps among all feasible solutions (Proposition 1). A further structural result motivates the design of FBS: every optimal ranking is reachable from r_s by adjacent swaps along which the distance from r_s never decreases (Lemma 2), so the search space of algorithms that never move back towards r_s, to which FBS belongs, still contains every optimal ranking.

Throughout this subsection, an adjacent swap action τ_k exchanges positions k and $k{+}1$ in a ranking r; an *algorithm* π is an ordered sequence of adjacent swaps $(\tau_{t_1}, \ldots, \tau_{t_m})$ such that their composition $\pi = \tau_{t_m} \circ \cdots \circ \tau_{t_1}$ determines the output ranking $\pi \circ r_s$. The intermediate rankings produced by the sequence of swaps are denoted $r^{(0)} = r_s,\ r^{(1)} = \tau_{t_1} \circ r^{(0)},\ \ldots, r^{(m)} = \tau_{t_m} \circ r^{(m-1)}$. We let A_{swap} denote the set of all such algorithms and $A_{swap,\nearrow} \subseteq A_{swap}$ the subset of algorithms such that $d(r_s, r^{(t)})$ is non-decreasing in t. A key elementary fact used repeatedly below is the following.

Observation 1 (Single-swap increment)**.** *Let r' be obtained from r by an adjacent swap of the items u, v at positions $k, k+1$, $r' = \tau_k \circ r$. Then $d(r_s, r') - d(r_s, r) = \pm\, w_{uv}$, where the sign is $+$ iff the swap creates an inversion with respect to r_s and $-$ iff it resolves one. Moreover, the same swap flips the Kendall orientation of the pair $\{u, v\}$ between $\nearrow$ and $\searrow$ and leaves all other pair orientations unchanged, so*

$$\Delta(r') - \Delta(r) \in \{-2, +2\}.$$

Lemma 1 (Reduction to adjacent swaps [23])**.** *For every ranking $r \in S_U$ there exists an algorithm $\pi \in A_{swap}$ such that $\pi \circ r_s = r$, and the shortest such algorithm has length $|\mathrm{Inv}(r_s, r)|$.*

Lemma 2 (Reduction to non-decreasing swaps)**.** *Let r^* be an optimal solution to Problem 1, then there exists an algorithm $\pi \in A_{swap,\nearrow}$ such that $\pi \circ r_s = r^*$.*

Lemma 3 (Existence of a valid greedy swap)**.** *For any $r \in S_U$, consider $b_k := a(r^{-1}(k))$ for $k = 1, \ldots, n$, where $r^{-1}(k) \in U$ is the item at rank k. Under the injectivity assumption of $a(\cdot)$, the sequence $b_1, \ldots, b_n$ consists of pairwise distinct reals. Hence, a pair of ranks (i, j) with $i < j$ is $\nearrow$ iff $b_i < b_j$ and $\searrow$ iff $b_i > b_j$. If $N_{\nearrow}(r) \neq N_{\searrow}(r)$, then r contains at least one adjacent pair of the majority sign, i.e. an index $k \in \{1, \ldots, n-1\}$ whose flip strictly reduces $\Delta(r)$ by 2.*

Proposition 1 (Minimality of the number of greedy swaps)**.** *Assume without loss of generality $N_{\nearrow}(r_s) \geq N_{\searrow}(r_s)$ and set $\Delta_0 := \Delta(r_s)$. If $\Delta_0 > \Delta^*$, then FBS is well defined and terminates after exactly $K^* := \lceil (\Delta_0 - \Delta^*)/2 \rceil$ swaps. Moreover, any algorithm in A_{swap} that outputs a feasible ranking uses at least K^* swaps.*

Theorem 1 (Optimality of FBS for the unweighted Kemeny distance)**.** *Let r_{FBS} denote the ranking returned by FBS. Then r_{FBS} is an optimal solution of Problem 1 for the unweighted Kemeny distance. Moreover, $d(r_{\mathrm{FBS}}, r_s) = K^* / \binom{n}{2}$ where $K^* = \lceil (\Delta_0 - \Delta^*)/2 \rceil$ as in Proposition 1.*

All proofs for the results above are given in Appendix D.

6 Approximation Bound for the Weighted Problem

When the Kemeny distance is weighted, different swaps have different costs and FBS must select the cheapest majority-sign adjacent pair at each step. Algorithm 2 (see pseudocode in Appendix C) replaces the doubly-linked list of Algorithm 1 with a min-heap keyed by the adjacent cost $c[k] = |S[k] - S[k{+}1]|$. Because FBS does not necessarily select the K^* feasible pairs of smallest cumulative weight, it loses the optimality property of the unweighted case. Nonetheless, for any strictly positive weight function, we can bound its sub-optimality in terms of the weight spread.

All the proofs of this section are given in Appendix E.

Proposition 2 (Weight-spread approximation bound for FBS)**.** *Let $w_{ij} > 0$ be an arbitrary strictly positive weight function, and let r_{FBS} be the ranking produced by FBS and r^* an optimal solution to Problem 1 for the corresponding weighted Kemeny distance. Let $w_{\min} := \min_{i<j} w_{ij}$ and $w_{\max} := \max_{i<j} w_{ij}$. Then: $d(r_s, r_{\mathrm{FBS}}) \leq w_{\max}/w_{\min}\, d(r_s, r^*)$.*

In the remainder of this section, we study a natural and practically relevant setting in which the scores s of the n items are mapped to $[n]$ – only the ordinal information of the score s is maintained – and the items are identified according to their score: the highest-score item will be identified with n, the second highest by $n-1$, and so on till the smallest-score item 1. Having $s(i) = i$ for $i \in [n]$, score-gap weights reduce to positional gaps $w_{ij} = |i-j|/W$, penalizing large displacements more than local rearrangements. In this setting, Proposition 2 yields $w_{\max}/w_{\min} = n-1$, too loose to be informative. To obtain tighter bounds, we exploit the following identity, whose proof builds on a left/right inversion decomposition in the spirit of the displacement-inversion analysis of Diaconis and Graham [8].

Lemma 4 (Spearman–Kendall identity)**.** *For integer scores $s(i) = i$, $i = 1, \dots, n$, with unnormalised pair weights $w_{ij} = |i-j|$ and any ranking r,*

$$d(r_s, r) \;=\; \tfrac{1}{2W}\, S_2(r), \qquad \textit{where} \quad S_2(r) := \sum_{i=1}^{n} \big(r(i) - i\big)^2 \textit{ is the Spearman squared displacement.}$$

This identity transforms the weighted distance from a sum over $O(n^2)$ *pairs* into a sum over n *items*, where each term depends only on how far item i has moved from its original position. The approximation ratio becomes a ratio of squared ℓ_2 norms: $d(r_s, r_{\mathrm{FBS}})/d(r_s, r^*) = \|\boldsymbol{\delta}_{\mathrm{FBS}}\|_2^2 / \|\boldsymbol{\delta}^*\|_2^2$, where $\boldsymbol{\delta}_i = r(i) - i$. Exploiting this identity, we can derive the exact FBS cost in closed form for the worst-case attribute configuration (i.e., the *all-ascending* case $a = s$ where all pairs are concordant and thus the initial imbalance is maximal), maximal fairness requirement $\psi = 1$, and n a power of two.

Proposition 3 (Tight bound for the all-ascending case)**.** *For integer scores $s(i) = i$ with $a = s$, $\psi = 1$, and $n = 2^t \geq 4$, FBS produces a ranking r_{FBS} such that:*

$$\frac{d(r_s, r_{\mathrm{FBS}})}{\mathrm{LP}} \;\longrightarrow\; \frac{1+\sqrt{2}}{2} \;\approx\; 1.207 \qquad \textit{as } n \to \infty,$$

where LP *is the LP relaxation lower bound,* $\mathrm{LP} = \sum_{k=1}^{K^*} w_{(k)}$*, where $w_{(1)} \leq w_{(2)} \leq \cdots \leq w_{(K^*)}$ are the sorted pair weights. In particular, $d(r_s, r_{\mathrm{FBS}}) < 2\, d(r_s, r^*)$ for all $n = 2^t$ with $t \geq 2$.*

The above asymptotic ratio $d_{\mathrm{FBS}}/\mathrm{LP} \to \frac{1+\sqrt{2}}{2}$ is *independent of* n and has been further verified computationally for all $n = 2^t$ with $t = 2, \dots, 21$ (Table 1 in Appendix E.3).

Our analysis and experiments suggest that the 2-approximation holds in general, dropping the *all-ascending* assumption $a = s$. That is, we conjecture that $d(r_s, r_{\mathrm{FBS}}) < 2\, d(r_s, r^*)$ for integer scores $s(i) = i$, for any attribute array a, and any feasible Δ^*. Indeed, combining the Spearman–Kendall identity (Lemma 4) with the classical Diaconis–Graham inequality $K \leq F \leq 2K$ [8] yields a natural proof target of factor 2, since the right-hand inequality $F \leq 2K$ is tight in general. Tightening below 2 would require arguments beyond classical norm comparisons that exploit the specific structure of FBS's output (see Appendix E.4 for further discussion). We next report empirical evidence supporting the 2-approximation.

Brute-force exact computation for small instances. For $n = 3, \dots, 9$ with integer scores $s(i) = i$, we enumerate every attribute permutation a and every feasible fairness threshold Δ^* (all values with the correct parity). For each instance we compute the FBS cost d_{FBS} and the exact optimum d^* by brute-force search over all $n!$ permutations, recording the ratio d_{FBS}/d^*. Figure 1(a) shows a scatter plot of all 699,282 non-trivial instances across these values of n. Every ratio lies strictly below 2, with the worst observed value ≈ 1.46 at $n = 9$, supporting the 2-approximation conjecture.

Exact MILP optimum for medium instances. For $n \in \{20, 50, 100\}$ we generate random and correlated attribute arrays and solve the integer programme exactly via MILP (binary inversion variables with FAS-style triangle constraints, see Appendix I). We use HiGHS via `scipy.optimize.milp` with a 180 s solver time-limit per instance, a 0.1% relative gap tolerance, and a 300 s wall-clock subprocess cap to handle pathologically hard correlated configurations. We also report the LP-relaxation lower bound $d^*_{\mathrm{LP}} \leq d^*$ on 612 instances, so that $d_{\mathrm{FBS}}/d^*_{\mathrm{LP}}$ provides a complementary upper-bound on the true approximation ratio. Figure 1(b) compares the two: the LP-bound and MILP-exact distributions coincide for the hardest correlated instances at $n = 20$ and $n = 50$ (the LP relaxation is tight there), but the MILP-exact ratio is strictly smaller for the random configurations at $n = 100$,

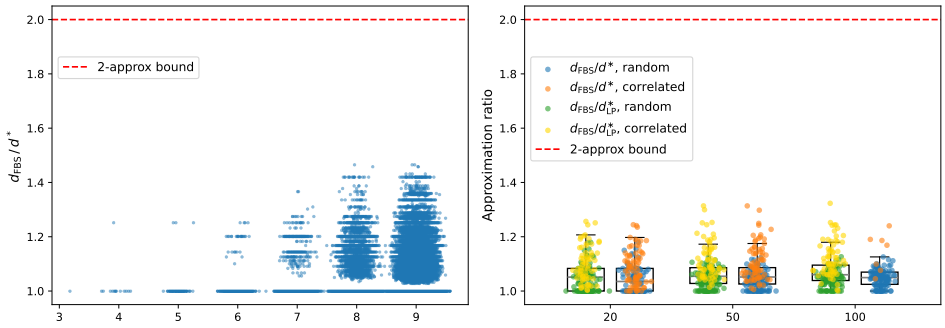


Figure 1: Empirical approximation ratio of FBS for the weighted Kemeny distance with integer scores $s(i) = i$ and fairness threshold Δ^* set to the tightest feasible value. **(a)** Exact ratio d_{FBS}/d^* for all attribute permutations and all feasible Δ^*, $n = 3, \ldots, 9$ (699,282 instances). **(b)** For $n \in \{20, 50, 100\}$, MILP-exact ratio d_{FBS}/d^* (blue: random instances, orange: correlated instances) and LP-bound upper bound $d_{\mathrm{FBS}}/d^*_{\mathrm{LP}}$ (green: random, yellow: correlated); at each n, the left box summarizes the LP-bound ratios and the right box the MILP-exact ratios. The dashed red line marks the conjectured 2-approximation bound.

where the LP bound has slack. Across all 542 MILP-solved instances the worst observed exact ratio is 1.31, well below the conjectured 2. The ratio $d_{\mathrm{FBS}}/\mathrm{LP}$ for the all-ascending case $a = s$ (up to $n \approx 2 \times 10^6$, including non-power-of-two values) is reported in Appendix J.

7 Comparison with Baselines

We next compare FBS against baselines on the quality of the ranking produced: as FBS is optimal in the unweighted case, we focus these experiments on the weighted case. Given that existing fair-ranking algorithms assume discrete protected groups, we discretize the continuous attribute into G equal-frequency bins and apply two representative methods using their original implementations: **DetConstSort** [13] (FairRankTune library [4], tested at $G \in \{2, 3, 5, 10\}$) and **FA*IR** [39] (fairsearchcore library [22], with $G = 2$ since FA*IR is inherently binary). We also include two baselines of our own: a simple **median-split interleaving** that splits items at the median attribute value into two groups, sorts each by descending score, and alternately picks the next highest-scored item from each group, and **Blend**, a continuous heuristic that sorts items by a convex combination of their score-rank and a fair-target rank. We defer detailed descriptions of all four baselines, including their parameters and implementation choices, to Appendix H.

All methods are evaluated on the same axes: achieved continuous fairness ϕ (computed on the original, undiscretized attribute) and weighted Kemeny distance from the score-based ranking. On synthetic instances with $n \in \{100, 1000, 5000\}$ and score–attribute correlation $\mathrm{corr} \in \{0, 0.5, 1\}$, FBS consistently traces a Pareto front that dominates all baselines (full results in Appendix J.2).

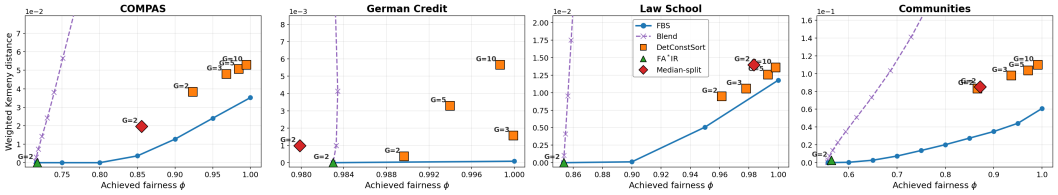


Figure 2: FBS vs. discretization baselines on real datasets. FBS (blue line) traces a Pareto front by sweeping ψ; baselines, except the continuous heuristic Blend, are shown as individual points for each G. FBS dominates all baselines across all four datasets.

Real-world datasets. We apply FBS to four datasets commonly used in the fair-ranking literature, reporting for each the initial fairness $\phi_0 := \phi(r_s)$ of the score-based ranking: **COMPAS** ($n = 7{,}214$, $\phi_0 = 0.72$), **German Credit** ($n = 1{,}000$, $\phi_0 = 0.98$), **Law School** ($n = 18{,}692$, $\phi_0 = 0.85$), and **Communities and Crime** ($n = 1{,}994$, $\phi_0 = 0.56$). Detailed descriptions of each dataset (score and attribute choice, source, preprocessing) are deferred to Appendix G. For each dataset we sweep the

fairness threshold ψ from 0 to 1 in steps of 0.05 and record the achieved fairness ϕ, the weighted Kemeny distance from the original score-based ranking, and the number of swaps K^*. Results are reported in Figure 2 . On all four datasets, FBS traces a smooth Pareto front that dominates the baselines: DetConstSort with $G \in \{2, 3, 5, 10\}$ and FA*IR ($G = 2$) achieve comparable fairness levels but at substantially higher cost, while median-split interleaving reaches moderate fairness at moderate cost but cannot match FBS's efficiency. The advantage of FBS is most pronounced on COMPAS and Communities, where the initial ranking is far from fair ($\phi_0 = 0.72$ and $\phi_0 = 0.56$, respectively) and the continuous attribute carries fine-grained information that is lost by discretization. German Credit is nearly fair to begin with ($\phi_0 = 0.98$), so FBS achieves $\phi \approx 1$ at negligible cost.

Appendix J extends this comparison in three directions. Two repair-then-sort baselines, which first remove the dependence of the score on the attribute and then sort, reach the same fairness as FBS only at a higher cost on all four datasets (Appendix J.3). NDCG with the original score as relevance, a utility measure that weighs the top of the list, stays at or above 0.98 for FBS at the fairness of the best baseline, and a variant of FBS that uses the NDCG loss of a swap as its cost keeps NDCG at 1.00 on the top 10 and top 100 positions at the same fairness (Appendix J.4). Finally, on the three datasets whose initial ranking is far from fair, FBS also reduces the Spearman correlation, the distance correlation, the Hilbert–Schmidt independence criterion and a positional discrepancy between position and attribute as ψ increases (Appendix J.5).

8 Runtime and Scalability

We first report theoretical runtime and space complexity of FBS, then the empirical scalability.

Proposition 4 (Runtime and space complexity of FBS)**.** *Let $K^* = \lceil (\Delta_0 - \Delta^*)/2 \rceil$ be the number of swaps performed by FBS. Both variants use $O(n)$ auxiliary space beyond the input. The running times are $O(n \log n + K^*)$ (for the unweighted case) and $O\big((n + K^*) \log n\big)$ (for the weighted case). In both cases the worst case is $K^* = \Theta(n^2)$.*

Proof. See Appendix F.

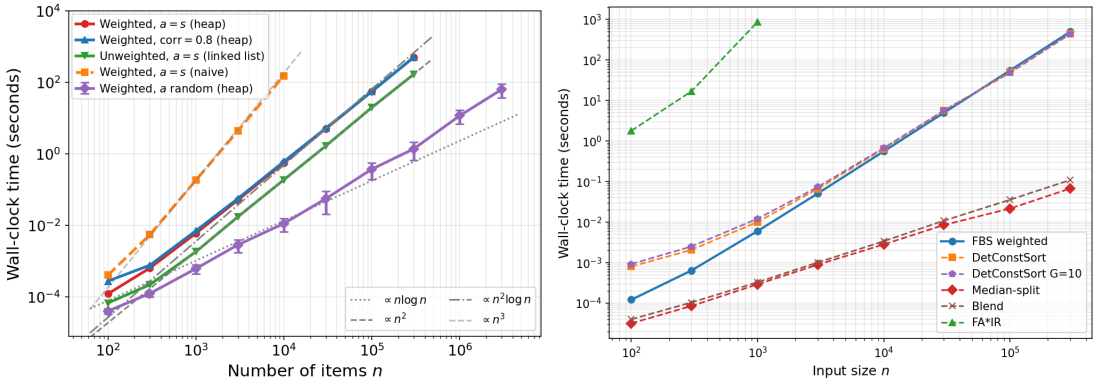
Empirical scalability analysis. We measure wall-clock time of FBS as n grows from 100 to 3×10^6. The following configurations are compared:

- **Weighted heap** (Algorithm 2) with $a = s$ (perfectly correlated) and with $\mathrm{corr}(s, a) = 0.8$, both at $\psi = 1.0$: $K^* = \Theta(n^2)$, exercising the worst-case $O(n^2 \log n)$ branch.
- **Unweighted linked list** (Algorithm 1) with $a = s$, $\psi = 1.0$: each swap is $O(1)$, giving $O(K^*) = O(n^2)$ total.
- **Naïve linear scan** with $a = s$, $\psi = 1.0$: $O(n)$ per swap, giving $O(nK^*) = O(n^3)$ total.
- **Random regime** (a uniform random, $\psi = 1.0$): K^* is much smaller than in the correlated case, yielding sub-quadratic runtime (10 independent instances per n).

Figure 3(a) shows the results on a log-log scale with reference lines for $n \log n$, n^2, $n^2 \log n$, and n^3. The weighted heap closely tracks $n^2 \log n$ in both correlated settings ($a = s$ and $\mathrm{corr} = 0.8$), the unweighted linked-list variant tracks n^2, and the naïve variant follows n^3. In the random regime, K^* is much smaller and the runtime is sub-quadratic; error bars (one standard deviation over 10 random instances) show moderate variance that increases with n, reflecting the instance-dependent nature of K^*. A runtime comparison of FBS against the baselines is reported in Figure 3(b).

9 Conclusions

We studied the problem of producing a ranking that is as close as possible to a given score-based ranking while satisfying a fairness constraint defined on a *continuous* protected attribute, without resorting to discretization. To this end, we introduced the fairness notion of *Continuous Pairwise Parity* and proposed Fair Bubble Sort, a greedy adjacent-swap algorithm that minimises the distance from the score-based ranking under given fairness constraint. On the theoretical side, we proved that FBS is *optimal* for the unweighted Kemeny distance (Theorem 1) and provided a data-dependent approximation bound for the weighted case. For the case of scores congruent to $[n]$ we sharpened this



(a) (b)

Figure 3: (a) Wall-clock time vs. input size n (log-log scale). The weighted heap-based FBS scales as $O(n^2 \log n)$ in the correlated regime ($a = s$ and $\mathrm{corr} = 0.8$) and sub-quadratically when K^* is small (random attributes). The unweighted linked-list variant scales as $O(n^2)$, while the naïve linear-scan implementation scales as $O(n^3)$. (b) Time comparison of FBS vs. baselines ($\mathrm{corr} = 1$, $\psi = 1.0$).

bound to a constant $(1+\sqrt{2})/2 \approx 1.207$ in the worst-case attribute configuration and for n a power of two (Proposition 3), and extensive computation suggests the ratio stays well below 2 for arbitrary attributes. On four real datasets (COMPAS, German Credit, Law School, and Communities and Crime) FBS dominates the Pareto front of fairness vs. accuracy across all tested fairness thresholds, outperforming both group-based baselines and a continuous heuristic.

Limitations. Our results have several limitations that suggest natural directions for future work. (i) The 2-approximation is conjectured but unproved: while $d_{\mathrm{FBS}} < 2\, d^*$ is empirically validated in every tested configuration, a formal proof remains open. (ii) FBS handles a *single* continuous protected attribute; extending to multiple attributes simultaneously adds further combinatorial constraints and is left for future work. (iii) Our setting is *static and one-shot*: long-term dynamics, repeated exposure, and feedback loops [14] are beyond the scope of this paper.

Acknowledgments and Disclosure of Funding

The authors received no third-party funding for this work and declare no competing interests.

References

[1] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. *ProPublica*, May 2016.

[2] Abolfazl Asudeh, H. V. Jagadish, Julia Stoyanovich, and Gautam Das. Designing fair ranking schemes. In *SIGMOD*, pages 1259–1276, 2019.

[3] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. Equity of attention: Amortizing individual fairness in rankings. In *The 41st international acm sigir conference on research & development in information retrieval*, pages 405–414, 2018.

[4] Kathleen Cachel and Elke Rundensteiner. Fairranktune: A python toolkit for fair ranking tasks. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 5195–5199, 2024.

[5] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. Ranking with fairness constraints. In *ICALP*, 2018.

[6] Silvia Chiappa. Path-specific counterfactual fairness. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial*

Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019.

[7] Vincent Conitzer, Andrew Davenport, and Jayant Kalagnanam. Improved bounds for computing kemeny rankings. In *AAAI*, volume 6, pages 620–626, 2006.

[8] Persi Diaconis and Ronald L Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 39(2):262–268, 1977.

[9] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 214–226, 2012.

[10] Ronald Fagin, Ravi Kumar, Mohammad Mahdian, D Sivakumar, and Erik Vee. Comparing and aggregating rankings with ties. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 47–58, 2004.

[11] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 259–268, 2015.

[12] David García-Soriano and Francesco Bonchi. Maxmin-fair ranking: Individual fairness under group-fairness constraints. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '21, page 436–446, 2021.

[13] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining*, pages 2221–2231, 2019.

[14] Luca Giuliani, Eleonora Misino, Roberta Calegari, and Michele Lombardi. Long-term fairness strategies in ranking with continuous sensitive attributes. In Roberta Calegari, Virginia Dignum, and Barry O'Sullivan, editors, *Proceedings of the 2nd Workshop on Fairness and Bias in AI co-located with 27th European Conference on Artificial Intelligence (ECAI 2024), Santiago de Compostela, Spain, October 20th, 2024*, CEUR Workshop Proceedings. CEUR-WS.org, 2024.

[15] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005.

[16] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *NeurIPS*, pages 3315–3323, 2016.

[17] Hans Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

[18] H. V. Jagadish, Francesco Bonchi, Tina Eliassi-Rad, Lise Getoor, Krishna Gummadi, and Julia Stoyanovich. The responsibility challenge for data. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, page 412–414, 2019.

[19] James E Johndrow and Kristian Lum. An algorithm for removing sensitive information. *The Annals of Applied Statistics*, 13(1):189–220, 2019.

[20] Morgan Kelly. Inequality and crime. *Review of economics and Statistics*, 82(4):530–539, 2000.

[21] John G. Kemeny and J. Laurie Snell. Preference ranking: An axiomatic approach. In *Mathematical Models in the Social Sciences*, pages 9–23. Ginn, Boston, MA, USA, 1962.

[22] Ivan Kitanovski. fairsearchcore: A Python library with the core algorithms used to do fair search. Python package, version 1.0.4, 2019. `https://pypi.org/project/fairsearchcore/`.

[23] Donald E Knuth. *The Art of Computer Programming: Sorting and Searching, volume 3*. Addison-Wesley Professional, 1998.

[24] Junpei Komiyama, Akiko Takeda, Junya Honda, and Hajime Shimao. Nonconvex optimization for regression with fairness constraints. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2737–2746. PMLR, 10–15 Jul 2018.

[25] Caitlin Kuhlman and Elke Rundensteiner. Rank aggregation algorithms for fair consensus. *Proceedings of the VLDB Endowment*, 13(12), 2020.

[26] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[27] Jeremie Mary, Clément Calauzènes, and Noureddine El Karoui. Fairness-aware learning for continuous attributes and treatments. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4382–4391. PMLR, 09–15 Jun 2019.

[28] Shira Mitchell, Eric Potash, Solon Barocas, Alexander D'Amour, and Kristian Lum. Algorithmic fairness: Choices, assumptions, and definitions. *Annual Review of Statistics and Its Application*, 8(1):141–163, 2021.

[29] Harikrishna Narasimhan, Andrew Cotter, Maya Gupta, and Serena Wang. Pairwise fairness for ranking and regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5248–5255, 2020.

[30] Gourab K. Patro, Lorenzo Porcaro, Laura Mitchell, Qiuyue Zhang, Meike Zehlike, and Nikhil Garg. Fair ranking: A critical review, challenges, and future directions. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 1929–1942, 2022.

[31] Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. Fairness in rankings and recommendations: an overview. *The VLDB Journal*, 31(3):431–458, 2022.

[32] Michael Redmond. Communities and Crime. UCI Machine Learning Repository, 2002. DOI: https://doi.org/10.24432/C53W3X.

[33] Yuta Saito and Thorsten Joachims. Fair ranking as fair division: Impact-based individual fairness in ranking. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1514–1524, 2022.

[34] Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *KDD*, pages 2219–2228, 2018.

[35] Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. Measuring and testing dependence by correlation of distances. *Annals of Statistics*, 35(6):2769–2794, 2007.

[36] Linda F. Wightman. LSAC national longitudinal bar passage study. LSAC research report series, Law School Admission Council, 1998.

[37] Ke Yang, Vasilis Gkatzelis, and Julia Stoyanovich. Balanced ranking with diversity constraints. In *IJCAI*, pages 6035–6042, 2019.

[38] Ke Yang and Julia Stoyanovich. Measuring fairness in ranked outputs. In *SSDBM*, pages 22:1–22:6, 2017.

[39] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. Fa*ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM Conference on Information and Knowledge Management (CIKM)*, pages 1569–1578, 2017.

[40] Meike Zehlike, Philipp Hacker, and Emil Wiedemann. Matching code and law: achieving algorithmic fairness with optimal transport. *Data Mining and Knowledge Discovery*, 34(1):163–200, 2020.

[41] Meike Zehlike, Ke Yang, and Julia Stoyanovich. Fairness in ranking, part i: Score-based ranking. *ACM Computing Surveys*, 55(6):1–36, 2022.

Technical Appendix and Supplementary Material

Appendix Contents

A Broader Impact **14**

B Motivations for the Continuous Pairwise Parity (CPP) measure ϕ **14**

C Pseudocode of FBS for the weighted Kemeny distance **16**

D Proofs of Section 5 **16**

D.1 Proofs of structural lemmas 16

D.2 Proof of Theorem 1 (Optimality for the unweighted case) 19

E Proofs of Section 6 **19**

E.1 Proof of Proposition 2 (Weight-spread approximation bound) 19

E.2 Proof of Lemma 4 (Spearman–Kendall identity) 20

E.3 Proof of Proposition 3 (Tight bound for the all-ascending case) 20

E.4 Towards a proof of the 2-approximation 25

F Proof of Section 8 **26**

G Datasets **27**

H Baselines **28**

I MILP formulation for the exact optimum **29**

J Additional experimental results **29**

J.1 Approximation ratio for the all-ascending case 29

J.2 Synthetic baseline comparisons 29

J.3 Continuous-fairness repair baselines 32

J.4 Ranking quality beyond the Kemeny distance: NDCG and FBS-NDCG 32

J.5 Other measures of dependence between position and attribute 34

K Hardware, Software and Code **35**

A Broader Impact

Fair ranking algorithms are deployed in high-stakes settings including hiring, lending, college admissions, content moderation, and criminal justice. Almost all existing algorithms require the protected attribute to be discrete, which forces practitioners to either categorise inherently continuous variables (age, income, health scores) into ad-hoc bins or to abandon fairness control entirely. Both choices are problematic: the former discards information and introduces arbitrary thresholds that can themselves be unfair (e.g., who decides where the age bins begin and end?), while the latter offers no protection at all. By providing a theoretically grounded algorithm that operates directly on continuous attributes, our work lowers the barrier to using fair-ranking methods in these settings and removes a class of arbitrary modelling choices that have historically been difficult to justify.

At the same time, we caution against treating FBS—or any post-processing fairness intervention—as a complete solution. The fairness measure ϕ targets pairwise statistical parity with respect to a continuous attribute, but it does not by itself guarantee proportional top-k representation, equal exposure, calibration across sub-populations, or freedom from indirect discrimination through correlated features. Rankings that look fair under ϕ can still encode biases inherited from the underlying score function, and a ϕ-fair re-ranking may give a misleading impression of fairness when applied on top of a flawed score. Responsible use therefore requires (a) auditing the score function itself, (b) verifying that ϕ is the appropriate fairness criterion for the application context, and (c) combining FBS with complementary checks (e.g., exposure or representation audits) where multiple fairness notions are simultaneously relevant. The choice of which protected attribute to enforce fairness on is itself a normative decision that should involve the affected stakeholders rather than be made unilaterally by the system designer.

B Motivations for the Continuous Pairwise Parity (CPP) measure ϕ

This section motivates the choice of $\phi(r) = 1 - |\tau(r, a)|$ as the fairness measure for ranking with continuous sensitive attributes. We argue that (i) ϕ is the natural pairwise extension of statistical parity to the continuous-attribute setting; (ii) it strictly generalises the Pairwise Statistical Parity of [25] from discrete groups to continuous values; and (iii) it satisfies a list of desirable properties—boundedness, symmetry, discretisation-freeness, and tractability—that the standard group-based measures cannot meet without arbitrary binning.

From group statistical parity to a pairwise principle. In classification, *statistical parity* [9] requires that the probability of a positive outcome be equal across protected groups, $\Pr(\hat{y} = 1 \mid g = A) = \Pr(\hat{y} = 1 \mid g = B)$. Adapting this principle to ranking is non-trivial: a "positive outcome" has no immediate analogue in a permutation, so the literature replaces it with proxies such as "appearing in the top-k" [39, 5, 34], which require predefined groups A, B and a discrete cut-off k. When the protected attribute is continuous, neither the groups nor the cut-off are defined: discretising a into bins forces an arbitrary choice of thresholds, discards fine-grained variation, and destroys the ordinality of the attribute (e.g. the bin $[20, 30)$ is treated as categorically distinct from $[30, 40)$, even though their boundary is conventional).

A more natural primitive in the ranking setting is the *pair*: for any two items u, v, one is ranked above the other, and one has a larger attribute value than the other. Pairwise statistical parity [25] formalises fairness as the requirement that the conditional probability of one group being ranked above the other equals the converse:

$$\Pr\big(r(u) < r(v) \mid g(u) = A,\, g(v) = B\big) = \Pr\big(r(u) < r(v) \mid g(u) = B,\, g(v) = A\big).$$

The translation to a continuous attribute is direct: replace "$g(u) = A,\, g(v) = B$" by "$a(u) > a(v)$" (and the converse by "$a(u) < a(v)$"), so that fairness becomes the requirement that, for a uniformly random pair (u, v), the higher-attribute item is equally likely to be ranked above or below the lower-attribute one. Empirically over the n items, this reads $N_{\nearrow}(r) \approx N_{\searrow}(r)$, which is precisely the condition $\tau(r, a) \approx 0$, equivalently $\phi(r) \approx 1$.

Reduction to Pairwise Statistical Parity in the binary case. The CPP measure recovers, as a special case, the Pairwise Statistical Parity of [25], a notion also defined by [29]. Suppose a takes only two values, $a(u) \in \{0, 1\}$, encoding membership in a low group $A = \{u : a(u) = 0\}$ and a

high group $B = \{u : a(u) = 1\}$ of sizes n_A and n_B; this is the situation obtained by discretising a continuous attribute at a threshold, for example its median. Let

$$P(r) := \Pr\big(r(v) < r(u) \mid u \in A,\ v \in B\big)$$

be the probability that a uniformly random B-item is ranked above a uniformly random A-item. The rank-parity score of [25] is $P(r)$ for group B and $1 - P(r)$ for group A, and their Pairwise Statistical Parity requires the two scores to be equal, i.e. $P(r) = \frac{1}{2}$.

Proposition 5 (Binary CPP equals Pairwise Statistical Parity)**.** *For a binary attribute as above,* $\Delta(r) = n_A n_B\,|2P(r) - 1|$*, and hence*

$$\phi(r) = 1 - \frac{n_A n_B}{\binom{n}{2}}\,\big|2P(r) - 1\big|.$$

In particular, $\phi(r) = 1$ *if and only if* $P(r) = \frac{1}{2}$*, i.e. if and only if* r *satisfies the Pairwise Statistical Parity of [25].*

Proof. Two items of the same group have equal attribute values, so their pair contributes to neither $N_{\nearrow}$ nor $N_{\searrow}$ (tie convention of Section 3); only the $n_A n_B$ cross-group pairs count. Among them, $n_A n_B P(r)$ pairs have the B-item ranked above the A-item and $n_A n_B(1 - P(r))$ have it ranked below. By the definition of Section 3, a cross-group pair is concordant when the item with the larger attribute value, the B-item, has the larger position index, i.e. is ranked below the A-item; hence $N_{\nearrow} = n_A n_B(1 - P(r))$, $N_{\searrow} = n_A n_B P(r)$, and $\Delta(r) = |N_{\nearrow} - N_{\searrow}| = n_A n_B |2P(r) - 1|$. Dividing by $\binom{n}{2}$ gives $|\tau(r, a)|$ and the expression for $\phi(r)$. □

The quantity $P(r)$ is the cross-group AUC, i.e. the Mann–Whitney U statistic of the two groups' positions divided by $n_A n_B$, and $2P(r) - 1$ is Somers' D. CPP therefore subsumes the discrete-group definition without loss of expressivity, and discretising a continuous attribute recovers that metric exactly. For $G > 2$ ordered groups the same computation gives $N_{\nearrow} - N_{\searrow} = \sum_{g<h} n_g n_h\,(1 - 2P_{gh}(r))$, where $P_{gh}(r)$ is the probability that a random item of the higher group h is ranked above a random item of the lower group g: the signed imbalance is the size-weighted sum of the pairwise parity terms over all pairs of groups.

Statistical interpretation. At the strict-fairness threshold $\psi = 1$ (equivalently $\Delta^* = 0$ when $n(n-1)/2$ is even), $\phi(r) = 1$ admits a probabilistic reading: for a uniformly random ordered pair of distinct items (u, v),

$$\Pr_{(u,v)}\big(r(u) < r(v) \mid a(u) > a(v)\big) = \tfrac{1}{2},$$

i.e. the relative ranking position carries no monotonic signal about the attribute. Intermediate thresholds $\psi \in (0, 1)$ correspond to a controlled relaxation of this equality, with the maximum imbalance $\Delta^* = \lfloor n(n-1)(1-\psi)/2 \rfloor$ measuring how far from the strict statistical-parity condition the practitioner is willing to deviate.

Comparison with related pairwise notions. [29] introduced a pairwise fairness criterion for learning-to-rank and regression, requiring equality of the probability that a relevant item from group A is ranked above a non-relevant item from group B, and vice versa. Their formulation targets the supervised-learning setting (training a scoring function under a fairness constraint) and relies on discrete groups. CPP, in contrast, is defined directly on the output ranking and on a continuous attribute, making it suitable for post-processing without retraining. The two notions agree in spirit—both anchor fairness at the pairwise level—but operate in different regimes.

Properties of ϕ. The CPP measure satisfies the following desiderata, none of which are jointly achieved by the standard group-based ranking-fairness measures.

1. **Boundedness and calibration.** $\phi(r) \in [0, 1]$ for every ranking, with $\phi(r) = 1$ iff $\tau(r, a) = 0$ (perfect fairness) and $\phi(r) = 0$ iff $|\tau(r, a)| = 1$ (deterministic monotonic dependence between ranking and attribute).
2. **Direction symmetry.** ϕ is invariant under reflection of the attribute, $\phi(r; a) = \phi(r; -a)$: the measure does not privilege "high-attribute on top" over "low-attribute on top".

3. **Discretisation-freeness.** ϕ is defined directly on the continuous values $a(u)$ and requires neither thresholds nor groups; in particular, it is invariant under any strictly monotone transformation of a.
4. **Robustness to attribute perturbations.** A small perturbation of the attribute values that preserves their relative order leaves ϕ exactly unchanged; an attribute swap that flips a single pair changes ϕ by $O(1/n^2)$. CPP is therefore stable under measurement noise that preserves rank order.
5. **Pairwise locality.** ϕ decomposes additively over pairs, mirroring the pairwise structure of the (unweighted) Kemeny distance. This shared structure is what enables FBS to optimise both quantities through a single algorithmic primitive—the adjacent swap—and underlies the optimality result of Theorem 1.
6. **Tractability.** $\phi(r)$ is computable in $O(n \log n)$ via merge-sort–based inversion counting, matching the cost of evaluating the Kemeny distance.

Comparison with alternative dependence measures. Other measures of dependence between position and attribute could replace the Kendall-τ correlation in the definition of ϕ. Spearman's correlation is also monotone, but it measures squared rank differences rather than pairwise concordance, so it does not share the pairwise structure of the Kemeny distance that lets a single adjacent-swap primitive optimise fairness and utility together. Distance correlation [35] and the Hilbert–Schmidt independence criterion (HSIC) [15] also capture non-monotone dependence, but they are not additive over pairs, and the effect of an adjacent swap on them depends on the whole ranking, so they admit neither the exact analysis of FBS nor a tractable hard constraint. Binning the attribute and applying a group-parity measure discards its ordinal structure and introduces binning artifacts. The Kendall-based CPP is the notion that is at the same time binning-free, ordinal and pairwise-decomposable, which is what lets one adjacent-swap primitive optimise both fairness and utility; the price is that CPP captures monotone dependence only, as discussed next.

Limitations. CPP captures *monotonic* dependence between ranking and attribute. A ranking in which the highest- and lowest-attribute items occupy the middle positions and mid-attribute items are placed at the extremes can have $\tau(r, a) = 0$ yet exhibit a clearly non-uniform attribute distribution along positions. Detecting such non-monotonic biases requires complementary measures (e.g., position-conditional moments of a), which we view as orthogonal extensions rather than as substitutes for a pairwise primitive.

C Pseudocode of FBS for the weighted Kemeny distance

When the Kemeny distance is weighted by score gaps $w_{ij} = |s(i) - s(j)|$, different swaps have different costs and FBS must select the cheapest majority-sign adjacent pair at each step. We replace the doubly-linked list L of Algorithm 1 with a min-heap H keyed by the adjacent cost $c[k] = |S[k] - S[k+1]|$ (the normalisation $1/W$ where $W = \sum_{k<l} |s(k) - s(l)|$ is a monotone rescaling and plays no role in the arg-min), supporting extraction, insertion, deletion, and key updates in $O(\log n)$. Ties in cost are broken by picking the leftmost pair (via the secondary key k).

D Proofs of Section 5

D.1 Proofs of structural lemmas

Proof of Observation 1 (Single-swap increment). Let $u, v \in U$ be the items with $r(u) = k$ and $r(v) = k{+}1$, so that after the adjacent swap $r'(u) = k{+}1$, $r'(v) = k$, and $r'(w) = r(w)$ for every $w \notin \{u, v\}$.

Pairwise locality. Let w be any item distinct from u and v, with rank $r(w) \notin \{k, k{+}1\}$. Since $r(w)$ is either strictly less than both k and $k{+}1$ or strictly greater than both, its order relative to u (and to v) is unchanged by the swap, and the rank of w itself is fixed. Hence for every pair of items other than $\{u, v\}$ the relative order in the ranking is the same in r and r'; consequently both the r_s-inversion status (which depends only on the relative orders in r and r_s) and the Kendall orientation (which depends only on the relative order in r and the attribute values) are preserved. The only pair whose status can change is $\{u, v\}$, whose relative order is reversed by the swap.

Algorithm 2 FBS (Fair Bubble Sort) – weighted, with priority queue

Require: Items $U = [n]$, attribute $a : U \to \mathbb{R}_+$, score $s : U \to \mathbb{R}_+$, fairness threshold $\psi \in [0, 1]$
Ensure: Ranking R solving Problem (1) with the weighted Kemeny distance

$(R, S, A) \leftarrow$ arrays initialized according to r_s (descending order of score) ▷ initial r_s
$c[k] \leftarrow |S[k] - S[k+1]|$ for $k = 1, \dots, n-1$
$\sigma[k] \leftarrow \operatorname{sign}(A[k+1] - A[k])$ for $k = 1, \dots, n-1$ ▷ adjacent-pair signs
$N_{\nearrow} \leftarrow |\{i < j : A[i] < A[j]\}|$; $N_{\searrow} \leftarrow |\{i < j : A[i] > A[j]\}|$
$\Delta^* \leftarrow \max\big(\lfloor (1-\psi)\, n(n-1)/2 \rfloor,\ |N_{\nearrow} - N_{\searrow}| \bmod 2\big)$ ▷ parity-safe threshold
$\mu \leftarrow \operatorname{sign}(N_{\nearrow} - N_{\searrow})$ ▷ majority sign ($+1$ or -1)
$H \leftarrow$ min-heap keyed by $(c[k],\, k)$ over $\{k : \sigma[k] = \mu\}$ ▷ build in $O(n)$
while $|N_{\nearrow} - N_{\searrow}| > \Delta^*$ **do**
  $k^* \leftarrow$ EXTRACT-MIN(H) ▷ $O(\log n)$
  swap $R[k^*] \leftrightarrow R[k^*+1]$, $S[k^*] \leftrightarrow S[k^*+1]$, $A[k^*] \leftrightarrow A[k^*+1]$
  for $\ell \in \{k^* - 1,\ k^*,\ k^* + 1\} \cap \{1, \dots, n-1\}$ **do** ▷ update neighbours
    $c[\ell] \leftarrow |S[\ell] - S[\ell+1]|$
    $\sigma_{\text{old}} \leftarrow \sigma[\ell]$
    if $\ell = k^*$ **then**
      $\sigma[\ell] \leftarrow -\sigma[\ell]$ ▷ middle sign flips (Observation 1)
    else
      $\sigma[\ell] \leftarrow \operatorname{sign}(A[\ell+1] - A[\ell])$ ▷ recompute from new attributes
    end if
    if $\sigma_{\text{old}} = \mu$ **and** $\sigma[\ell] \neq \mu$ **then**
      remove ℓ from H ▷ $O(\log n)$
    else if $\sigma_{\text{old}} \neq \mu$ **and** $\sigma[\ell] = \mu$ **then**
      insert ℓ into H with key $(c[\ell],\, \ell)$ ▷ $O(\log n)$
    else if $\sigma[\ell] = \mu$ **and** $c[\ell]$ changed **then**
      update key of ℓ in H to $(c[\ell],\, \ell)$ ▷ $O(\log n)$
    end if
  end for
  $N_{\nearrow} \leftarrow N_{\nearrow} - \mu$; $N_{\searrow} \leftarrow N_{\searrow} + \mu$
end while
return R

Effect on d. The pair $\{u, v\}$ contributes w_{uv} to $d(r_s, \cdot)$ iff it lies in $\mathrm{Inv}(r_s, \cdot)$, and the swap toggles this membership:

- if $\{u, v\} \notin \mathrm{Inv}(r_s, r)$ (i.e. u, v are in the same relative order as in r_s), then $\{u, v\} \in \mathrm{Inv}(r_s, r')$ and the swap *creates* an inversion, so $d(r_s, r') - d(r_s, r) = +w_{uv}$;
- if $\{u, v\} \in \mathrm{Inv}(r_s, r)$, then $\{u, v\} \notin \mathrm{Inv}(r_s, r')$ and the swap *resolves* an inversion, so $d(r_s, r') - d(r_s, r) = -w_{uv}$.

By pairwise locality no other pair contributes to the change, establishing the first claim.

Effect on $N_{\nearrow} - N_{\searrow}$. Under the general-position assumption (no ties in a), every pair is either $\nearrow$ or $\searrow$. By pairwise locality the orientation of every pair other than $\{u, v\}$ is preserved, while the orientation of $\{u, v\}$ flips between $\nearrow$ and $\searrow$. Hence $N_{\nearrow}$ changes by exactly ± 1 and $N_{\searrow}$ by exactly ∓ 1, giving

$$\big(N_{\nearrow} - N_{\searrow}\big)(r') - \big(N_{\nearrow} - N_{\searrow}\big)(r) \in \{-2, +2\},$$

with sign $+2$ if $\{u, v\}$ was $\searrow$ in r (and becomes $\nearrow$ in r') and -2 otherwise. □

Proof of Lemma 1 (Reduction to adjacent swaps). Since r_s and r are bijections $U \to [n]$, the map $\pi := r \circ r_s^{-1}$ is a permutation of $[n]$, and $r = \pi \circ r_s$. It is classical that S_n is generated by the adjacent transpositions $\tau_k := (k, k+1)$, $k = 1, \dots, n-1$: every $\pi \in S_n$ admits a finite decomposition $\pi = \tau_{k_m} \circ \tau_{k_{m-1}} \circ \cdots \circ \tau_{k_1}$, and the minimum number of factors equals $|\mathrm{Inv}(\pi)| = |\mathrm{Inv}(r_s, r)|$ (e.g. the bound is attained by bubble sort) [23]. An adjacent swap at rank k sends a ranking r' to $\tau_k \circ r'$, so applying $\tau_{k_1}, \tau_{k_2}, \dots, \tau_{k_m}$ in order to r_s produces the trajectory

$$r_s,\ \tau_{k_1} \circ r_s,\ \tau_{k_2} \circ \tau_{k_1} \circ r_s,\ \dots,\ \tau_{k_m} \circ \cdots \circ \tau_{k_1} \circ r_s = \pi \circ r_s = r,$$

i.e., an algorithm in A_{swap} whose output is r. □

Proof of Lemma 2 (Reduction to non-decreasing swaps). We construct a sequence of adjacent swaps from r_s to r^* along which $d(r_s, \cdot)$ is non-decreasing. Run bubble sort on r^*, sorting items toward the order prescribed by r_s: at each step, with current ranking q, find adjacent ranks $k, k{+}1$ such that the items u, v at those ranks in q (i.e., $q(u) = k$, $q(v) = k{+}1$) are in opposite r_s-order — equivalently, $\{u, v\}$ contributes an inversion to $\mathrm{Inv}(r_s, q)$ — and swap them. Such a pair exists whenever $q \neq r_s$, and each swap reduces $|\mathrm{Inv}(r_s, q)|$ by exactly 1, so the algorithm terminates at r_s after $m = |\mathrm{Inv}(r_s, r^*)|$ swaps. By Observation 1 each swap resolves the r_s-inversion of the corresponding item pair $\{u, v\}$ and decreases $d(r_s, \cdot)$ by $w_{uv} \geq 0$; hence the trajectory

$$r^* = q^{(0)} \xrightarrow{\tau_{k_1}} q^{(1)} \xrightarrow{\tau_{k_2}} \cdots \xrightarrow{\tau_{k_m}} q^{(m)} = r_s$$

satisfies $d(r_s, q^{(0)}) \geq d(r_s, q^{(1)}) \geq \cdots \geq d(r_s, q^{(m)}) = 0$.

Adjacent transpositions are involutions ($\tau_k^2 = \mathrm{id}$), so applying $\tau_{k_m}, \tau_{k_{m-1}}, \ldots, \tau_{k_1}$ in this reversed order to r_s visits the same rankings in reverse,

$$r_s = q^{(m)} \to q^{(m-1)} \to \cdots \to q^{(0)} = r^*,$$

along which $d(r_s, \cdot)$ is non-decreasing. The corresponding algorithm therefore lies in $A_{swap,\nearrow}$ and produces r^*. □

Proof of Lemma 3 (Existence of a valid greedy swap). We argue by contrapositive. The claim "$N_{\nearrow}(r) > N_{\searrow}(r) \Rightarrow$ some adjacent pair is $\nearrow$" is logically equivalent to

$$\text{no adjacent pair of } r \text{ is } \nearrow \implies N_{\nearrow}(r) \leq N_{\searrow}(r).$$

Because the values $b_1, \ldots, b_n$ are pairwise distinct, every pair is either $\nearrow$ or $\searrow$, so the hypothesis "no adjacent $\nearrow$" is equivalent to

$$b_k > b_{k+1} \qquad \text{for every } k = 1, \ldots, n-1, \qquad (\star)$$

i.e. every adjacent pair is $\searrow$. We shall actually establish the *stronger* conclusion $N_{\nearrow}(r) = 0$ from $(\star)$; since $N_{\nearrow}(r) = 0 \leq N_{\searrow}(r)$ this suffices.

Assume $(\star)$. For any pair $i < j$, chaining the $j - i$ adjacent strict inequalities in $(\star)$ and using transitivity of $>$ on the distinct reals $b_1, \ldots, b_n$ yields

$$b_i > b_{i+1} > \cdots > b_j \implies b_i > b_j,$$

so (i, j) is $\searrow$. This holds for every $i < j$, hence $N_{\nearrow}(r) = 0$ as announced. Returning to the original direction: if $N_{\nearrow}(r) > N_{\searrow}(r) \geq 0$ then $N_{\nearrow}(r) \geq 1$, so $(\star)$ must fail, i.e. at least one adjacent pair is $\nearrow$. The $\searrow$-majority case is obtained by exchanging $>$ and $<$ throughout.

The argument only invokes transitivity on the pairwise distinct values $b_1, \ldots, b_n$ of the current ranking, so it applies unchanged at every greedy iteration, not only at the initial $r = r_s$. □

Proof of Proposition 1 (Minimality of the number of greedy swaps). Lower bound. By Observation 1, any adjacent swap changes $N_{\nearrow} - N_{\searrow}$ by exactly ± 2. Since $\Delta = |N_{\nearrow} - N_{\searrow}|$, the reverse triangle inequality gives $\Delta' \geq \Delta - 2$ after any single swap (Δ' can equal $\Delta - 2$, Δ, or $\Delta + 2$; the middle case occurs only when $\Delta = 1$ and the swap flips the sign of $N_{\nearrow} - N_{\searrow}$). Going from Δ_0 to any value $\leq \Delta^*$ therefore requires at least $\lceil (\Delta_0 - \Delta^*)/2 \rceil = K^*$ swaps.

Parity remark. Each swap preserves the parity of $N_{\nearrow} - N_{\searrow}$, so $\Delta \bmod 2 = \Delta_0 \bmod 2$ throughout. If $\Delta^* < \Delta_0 \bmod 2$, the target $\Delta \leq \Delta^*$ is unreachable. This is precisely why the algorithm sets $\Delta^* \leftarrow \max(\Delta^*, \Delta_0 \bmod 2)$: the adjusted threshold ensures K^* is a non-negative integer and $\Delta_0 - 2K^* \geq 0$.

FBS attains the bound. We assume without loss of generality $N_{\nearrow}(r_s) \geq N_{\searrow}(r_s)$, so the majority sign is $\mu = +1$ and $N_{\nearrow} - N_{\searrow} = \Delta_0 > 0$ (the $\mu = -1$ case is symmetric). We prove by induction on k the loop invariant

$$N_{\nearrow} - N_{\searrow} = \Delta_0 - 2k \qquad (0 \leq k \leq K^*).$$

Base case ($k = 0$): immediate from the definition of Δ_0.

Inductive step ($k \to k{+}1$, for $k < K^*$): by the inductive hypothesis $N_{\nearrow} - N_{\searrow} = \Delta_0 - 2k$. Since $k < K^* = \lceil (\Delta_0 - \Delta^*)/2 \rceil$, we have $\Delta_0 - 2k > \Delta^* \geq 0$, so in particular $N_{\nearrow} > N_{\searrow}$. Lemma 3 then guarantees the existence of an adjacent $\nearrow$ pair, so FBS can select one. By Observation 1, swapping it decreases $N_{\nearrow}$ by 1 and increases $N_{\searrow}$ by 1, giving $N_{\nearrow} - N_{\searrow} = \Delta_0 - 2(k{+}1)$.

After K^* swaps, $\Delta = \Delta_0 - 2K^* \in \{0, 1\} \subseteq [0, \Delta^*]$ (using $\Delta^* \geq \Delta_0 \bmod 2$), and the algorithm terminates. The greedy therefore uses exactly K^* swaps, matching the lower bound. □

D.2 Proof of Theorem 1 (Optimality for the unweighted case)

Proof of Theorem 1. Unlike the weighted case, the unweighted Kemeny distance depends only on the *number* of inversions, not on which specific pairs are inverted:

$$d(r_s, r) = \frac{|\mathrm{Inv}(r_s, r)|}{\binom{n}{2}}.$$

If $\Delta_0 \leq \Delta^*$, then r_s is already feasible and $d(r_s, r_s) = 0$ is trivially optimal. We therefore assume $\Delta_0 > \Delta^*$, so that $K^* = \lceil (\Delta_0 - \Delta^*)/2 \rceil \geq 1$. It suffices to show that $|\mathrm{Inv}(r_s, r_{\mathrm{FBS}})| = K^*$ and that $|\mathrm{Inv}(r_s, r)| \geq K^*$ for every feasible r.

Lower bound. Fix any feasible r, i.e. any r with $\phi(r) \geq \psi$. By Lemma 1, the minimum number of adjacent swaps taking r_s to r is $|\mathrm{Inv}(r_s, r)|$. This minimum-length sequence is an algorithm in A_{swap} whose output is the feasible ranking r, so by Proposition 1 it uses at least K^* swaps. Therefore

$$|\mathrm{Inv}(r_s, r)| \;\geq\; K^* \qquad \text{for every feasible } r.$$

FBS matches the bound. By Proposition 1, FBS terminates after exactly K^* adjacent swaps at a ranking r_{FBS} with $\Delta(r_{\mathrm{FBS}}) \leq \Delta^*$, so r_{FBS} is feasible. By Observation 1, each adjacent swap creates or resolves exactly one r_s-inversion, so $|\mathrm{Inv}(r_s, \cdot)|$ changes by exactly ± 1 per swap. Since $|\mathrm{Inv}(r_s, r_s)| = 0$, we have

$$|\mathrm{Inv}(r_s, r_{\mathrm{FBS}})| \;\leq\; K^*.$$

Combined with the lower bound (applied to the feasible r_{FBS}) we get $|\mathrm{Inv}(r_s, r_{\mathrm{FBS}})| = K^*$, and hence

$$d(r_s, r_{\mathrm{FBS}}) = \frac{K^*}{\binom{n}{2}} = \min_{r:\phi(r) \geq \psi} d(r_s, r).$$ □

E Proofs of Section 6

E.1 Proof of Proposition 2 (Weight-spread approximation bound)

Proof of Proposition 2. If $\Delta_0 \leq \Delta^*$, then r_s is already feasible, $K^* = 0$, and $d(r_s, r_{\mathrm{FBS}}) = d(r_s, r^*) = 0$; the bound holds trivially. We therefore assume $\Delta_0 > \Delta^*$, so that $K^* \geq 1$.

Upper bound on $d(r_s, r_{\mathrm{FBS}})$. By Proposition 1, FBS performs exactly K^* adjacent swaps starting from r_s. By Observation 1, each swap changes $|\mathrm{Inv}(r_s, \cdot)|$ by exactly ± 1. Since $|\mathrm{Inv}(r_s, r_s)| = 0$ and r_{FBS} is feasible (hence $|\mathrm{Inv}(r_s, r_{\mathrm{FBS}})| \geq K^*$ by Lemma 1 and Proposition 1), we obtain $|\mathrm{Inv}(r_s, r_{\mathrm{FBS}})| = K^*$ (in particular, every swap creates a new r_s-inversion). Therefore

$$d(r_s, r_{\mathrm{FBS}}) = \sum_{(i,j) \in \mathrm{Inv}(r_s, r_{\mathrm{FBS}})} w_{ij} \;\leq\; K^* \cdot w_{\max}.$$

Lower bound on $d(r_s, r^*)$. By Lemma 1 and Proposition 1, every feasible ranking r satisfies $|\mathrm{Inv}(r_s, r)| \geq K^*$. Hence

$$d(r_s, r^*) = \sum_{(i,j) \in \mathrm{Inv}(r_s, r^*)} w_{ij} \;\geq\; K^* \cdot w_{\min}.$$

Dividing the upper by the lower bound (both positive since $K^* \geq 1$ and $w_{\min} > 0$) gives the stated ratio. □

Score-rank convention. Throughout this appendix we write "$s(i) = i$" as shorthand for equally-spaced scores: only the score gaps enter the weights, yielding pair weights $w_{uv} = |s(u) - s(v)| = |u - v|$ up to a global normalisation that cancels in all ratios, so the labeling $r_s = \mathrm{id}$ (item i at position i) is without loss of generality. Similarly, "$a = s$" refers to the configuration $a(i) = i$, which under $r_s = \mathrm{id}$ gives $\tau(r_s, a) = +1$ — the maximum-imbalance case in which all $\binom{n}{2}$ pairs are concordant.

E.2 Proof of Lemma 4 (Spearman–Kendall identity)

Proof of Lemma 4. Items are labelled $1, \ldots, n$ with scores $s(i) = i$ and unnormalised pair weights $w_{uv} = v - u$ for $u < v$ (the global factor $1/W$ cancels in all ratios and is omitted throughout this appendix). For a ranking r, we must show

$$\sum_{\substack{u<v \\ r(u)>r(v)}} (v - u) \;=\; \sum_i i^2 - \sum_i i\, r(i) \;=\; \tfrac{1}{2} \sum_i (r(i) - i)^2. \tag{2}$$

The second equality follows from $\sum r(i)^2 = \sum i^2$ (both range over $\{1, \ldots, n\}$). For the first equality, define for each item i:

$$\begin{aligned} \ell(i) &:= \#\{j < i : r(j) > r(i)\} \quad \text{(left inversions)}, \\ \rho(i) &:= \#\{j > i : r(j) < r(i)\} \quad \text{(right inversions)}. \end{aligned}$$

Step 1: $\ell(i) - \rho(i) = i - r(i)$. Among items $\{1, \ldots, i-1\}$, let p denote how many occupy positions $< r(i)$. Then $\ell(i) = (i-1) - p$. Since exactly $r(i)-1$ items in total sit before position $r(i)$, $p + \rho(i) = r(i)-1$, giving $p = r(i)-1-\rho(i)$. Substituting: $\ell(i) = (i-1) - (r(i)-1-\rho(i)) = i - r(i) + \rho(i)$, hence $\ell(i) - \rho(i) = i - r(i)$.

Step 2: The right-hand side equals $\sum_i i\,(\ell(i) - \rho(i))$. Rewriting the inversion sum:

$$\sum_{\substack{u<v \\ r(u)>r(v)}} (v - u) = \sum_v v\,\ell(v) - \sum_u u\,\rho(u) = \sum_i i\,\big(\ell(i) - \rho(i)\big).$$

Conclusion. By Step 1, $\sum_i i(\ell(i) - \rho(i)) = \sum_i i(i - r(i)) = \sum_i i^2 - \sum_i i\, r(i)$. □

Remark. The identity connects two classical rank-correlation measures: the weighted Kendall distance (with weights $v-u$) equals half the Spearman squared displacement. This is specific to equally-spaced (integer) scores and does not hold for general score arrays.

E.3 Proof of Proposition 3 (Tight bound for the all-ascending case)

The proof of Proposition 3 rests on two ingredients: (i) an exact characterisation of the FBS output and its closed-form cost for the all-ascending case $a = s$ (Proposition 6), and (ii) an LP relaxation lower bound on d^* (Proposition 7). We develop each in turn, then combine them to prove, for this special case, the asymptotic convergence of the ratio $d_{\mathrm{FBS}}/\mathrm{LP}$ to $(1+\sqrt{2})/2$ and the strict 2-approximation $d_{\mathrm{FBS}} < 2\,d^*$.

Setup. When $a = s = (1, 2, \ldots, n)$, all $\binom{n}{2}$ pairs are concordant. With $\psi = 1$, FBS performs $K^* = \lceil \binom{n}{2}/2 \rceil$ swaps. We restrict to $n = 2^t$ ($t \geq 2$) where $\binom{n}{2}$ is even and $K^* = n(n-1)/4$. Write $m := n/2$.

Proposition 6 (FBS output and closed-form cost for $a{=}s$, $n{=}2^t$)**.** *When $a = s = (1, \ldots, n)$ and $n = 2^t \geq 4$, FBS produces the ranking r_{FBS} defined by the following five blocks:*

(B1) *Positions $1, \ldots, m/2$: items $m,\ m-1,\ \ldots,\ m/2+1$* (reversed upper-left quarter)*;*

(B2) *Position $m/2+1$: item n;*

(B3) *Positions $m/2+2, \ldots, m+1$: items $m/2,\ m/2-1,\ \ldots,\ 1$* (reversed lower-left quarter)*;*

(B4) *Position* $m{+}2$*: item* $n{-}1$*;*

(B5) *Positions* $m{+}3, \ldots, 2m$*: items* $n{-}2,\ n{-}3,\ \ldots,\ m{+}1$ (reversed right half).

The total FBS cost is

$$d_{\mathrm{FBS}}(n) \;=\; \frac{m(8m^2+21m-14)}{24}, \qquad m = n/2. \tag{3}$$

We prove the two parts of the proposition separately. We first derive the closed-form cost (3) assuming the five-block structure, and then prove the five-block structure itself by induction on t.

Proof of Eq. (3). Assuming the five-block structure, we compute $d_{\mathrm{FBS}} = \frac{1}{2}S_2(r_{\mathrm{FBS}})$ using Lemma 4. The displacement of item i is $\delta_i := r_{\mathrm{FBS}}(i) - i$. We evaluate $S_2 = \sum_i \delta_i^2$ block by block.

Block B1 (positions $j = 1, \ldots, m/2$, item $m{+}1{-}j$). Displacement $\delta_j = m{+}1{-}2j$, taking all odd values from $m{-}1$ down to 1. Hence $S_2^{(1)} = m(m^2{-}1)/6$.

Block B2 (position $m/2{+}1$, item $2m$). $S_2^{(2)} = (3m/2{-}1)^2$.

Block B3 (positions $m/2{+}2, \ldots, m{+}1$). $\delta = 2k$ for $k = 1, \ldots, m/2$, giving $S_2^{(3)} = m(m{+}1)(m{+}2)/6$.

Block B4 (position $m{+}2$, item $2m{-}1$). $S_2^{(4)} = (m{-}3)^2$.

Block B5 (positions $m{+}3, \ldots, 2m$, items $2m{-}2, \ldots, m{+}1$). Item $i = 3m{+}1{-}j$ sits at position j, giving $\delta = 2j{-}3m{-}1$. As j ranges from $m{+}3$ to $2m$, these are the $m{-}2$ consecutive odd values $5{-}m,\ 7{-}m,\ \ldots,\ m{-}3,\ m{-}1$. The full set of odd values from $-(m{-}1)$ to $m{-}1$ has m elements with sum of squares $m(m^2{-}1)/3$; Block B5 omits the two largest absolute values $m{-}1$ and $m{-}3$ (which belong to the items in blocks B1 and B4 respectively). Hence $S_2^{(5)} = m(m^2{-}1)/3 - (m{-}1)^2 - (m{-}3)^2$.

Total. The $(m{-}3)^2$ terms from B4 and B5 cancel. Combining the three fractional terms:

$$\begin{aligned}\frac{m(m^2{-}1)}{6} + \frac{m(m{+}1)(m{+}2)}{6} + \frac{m(m^2{-}1)}{3} &= \frac{m(m{+}1)\big[(m{-}1) + (m{+}2) + 2(m{-}1)\big]}{6} \\ &= \frac{m(m{+}1)(4m{-}1)}{6}.\end{aligned}$$

The remaining squared terms give $(3m/2{-}1)^2 - (m{-}1)^2 = m(5m{-}4)/4$. Therefore

$$\begin{aligned}S_2 &= \frac{m(m{+}1)(4m{-}1)}{6} + \frac{m(5m{-}4)}{4} = \frac{2m(m{+}1)(4m{-}1) + 3m(5m{-}4)}{12} \\ &= \frac{m(8m^2+21m{-}14)}{12},\end{aligned}$$

where the last step uses $2(m{+}1)(4m{-}1) = 8m^2{+}6m{-}2$ and $3(5m{-}4) = 15m{-}12$, giving $8m^2{+}21m{-}14$. Hence $d_{\mathrm{FBS}} = S_2/2 = m(8m^2+21m{-}14)/24$. □

Proof of the five-block structure. The proof proceeds by induction on t, exploiting a hierarchical merge structure forced by FBS's min-cost rule. We establish the required auxiliary definitions and lemmas, then give the main induction.

Definition 1 (Reversed block)**.** *A* reversed block of size b *at positions* $p, p{+}1, \ldots, p{+}b{-}1$ *containing items* $\{q, q{+}1, \ldots, q{+}b{-}1\}$ *means that position* $p{+}j$ *holds item* $q{+}b{-}1{-}j$ *for* $j = 0, \ldots, b{-}1$.

Lemma 5 (Phase 1: disjoint transpositions)**.** *The first* m *FBS swaps transpose the pairs at positions* $(1,2), (3,4), \ldots, (n{-}1, n)$*, each at cost* 1*. After phase* 1*, the state consists of* m *reversed blocks of size* 2*: block* j *at positions* $2j{-}1, 2j$ *contains items* $\{2j{-}1, 2j\}$ *in reversed order, for* $j = 1, \ldots, m$.

Proof. Initially all $n{-}1$ adjacent pairs are concordant with cost 1. FBS picks position 1 (leftmost minimum-cost eligible position). After swapping positions 1 and 2: position 1 becomes discordant ($\sigma[1] = -1$); position 2 has cost $|s(1){-}s(3)| = 2$; all positions $k \geq 3$ remain at cost 1. Hence the

next FBS choice is position 3. By induction on j: after j transpositions, the first $2j$ positions form reversed pairs with discordant internal signs, while positions $2j{+}1, \ldots, n{-}1$ remain concordant at cost 1. FBS next selects position $2j{+}1$. After all m transpositions, every position $2j{-}1$ holds the pair $(2j, 2j{-}1)$ and no cost-1 eligible positions remain. □

Lemma 6 (Post-phase-1 eligible positions)**.** *After phase 1, the eligible (concordant) adjacent positions are precisely the even indices $k = 2, 4, \ldots, n{-}2$, each with cost 3. There are $m{-}1$ such positions.*

Proof. At even position $k = 2j$, the items are $2j{-}1$ (right element of block j) and $2j{+}2$ (left element of block $j{+}1$). Since $2j{-}1 < 2j{+}2$, the pair is concordant with cost $|s(2j{-}1){-}s(2j{+}2)| = 3$. At odd positions, items form a reversed pair within a block and are discordant. □

Lemma 7 (Merge Lemma)**.** *Suppose the current state contains two adjacent reversed blocks of size b: a* left block *at positions $p, \ldots, p{+}b{-}1$ containing items $\{q, \ldots, q{+}b{-}1\}$ reversed, and a* right block *at positions $p{+}b, \ldots, p{+}2b{-}1$ containing items $\{q{+}b, \ldots, q{+}2b{-}1\}$ reversed. Suppose furthermore that:*

(i) *the boundary position $p{+}b{-}1$ (between items q and $q{+}2b{-}1$) is the unique eligible position with minimum cost within these $2b$ positions;*

(ii) *no eligible position outside these $2b$ positions has cost less than the boundary cost $2b{-}1$.*

Then FBS performs exactly b^2 swaps within these $2b$ positions, merging them into a single reversed block of size $2b$ containing items $\{q, \ldots, q{+}2b{-}1\}$ in decreasing order. The total cost of these b^2 swaps is b^3. The merge proceeds in b rounds, where round r ($r = 0, \ldots, b{-}1$) consists of b swaps with costs $2b{-}1{-}r,\ 2b{-}2{-}r,\ \ldots,\ b{-}r$.

Proof. We prove by induction on r ($r = -1, 0, \ldots, b{-}1$, where $r = -1$ denotes the initial state) the following invariant.

Invariant after round r: relative to local positions $0, \ldots, 2b{-}1$ (offset by p), the state consists of three segments:

- **Segment 1** (positions $0, \ldots, r$): items $q{+}2b{-}1,\ q{+}2b{-}2,\ \ldots,\ q{+}2b{-}1{-}r$ in decreasing order.
- **Segment 2** (positions $r{+}1, \ldots, b{+}r$): items $q{+}b{-}1,\ q{+}b{-}2,\ \ldots,\ q$ in decreasing order (the original left-block items, shifted right by $r{+}1$).
- **Segment 3** (positions $b{+}r{+}1, \ldots, 2b{-}1$): items $q{+}2b{-}2{-}r,\ \ldots,\ q{+}b$ in decreasing order (the remaining right-block items).

All adjacent pairs within each segment are discordant. The unique eligible position with minimum cost (within these $2b$ positions) is the boundary between segments 2 and 3 at position $b{+}r$, with cost $2b{-}2{-}r$.

Base case ($r = -1$): Segment 1 is empty, segment 2 is the left block, segment 3 is the right block. The only eligible position is the boundary at position $b{-}1$, with cost $2b{-}1$.

Round $r{+}1$ execution: The boundary between segments 2 and 3 is at position $b{+}r$, where item q sits at position $b{+}r$ and item $q{+}2b{-}2{-}r$ sits at position $b{+}r{+}1$. FBS swaps these two items at cost $2b{-}2{-}r$.

After this initial swap, item $q{+}2b{-}2{-}r$ at position $b{+}r$ propagates leftward through segment 2. At each step the leftward swap is the cheapest eligible move: the propagation front has strictly decreasing cost ($2b{-}3{-}r,\ 2b{-}4{-}r,\ \ldots$), while the only other eligible position—item q adjacent to segment 3's leftmost item at position $b{+}r{+}1$—has fixed cost $2b{-}3{-}r$. At the first step both costs are equal and leftmost tie-breaking selects the propagation; at every subsequent step the leftward cost is strictly smaller.

After $b{-}1$ leftward swaps, item $q{+}2b{-}2{-}r$ reaches position $r{+}1$, adjacent to item $q{+}2b{-}1{-}r$ in segment 1; since $q{+}2b{-}2{-}r < q{+}2b{-}1{-}r$, this pair is discordant. The state matches the

invariant with r replaced by $r{+}1$. Total swaps in round $r{+}1$: $1 + (b{-}1) = b$, with costs $2b{-}1{-}(r{+}1),\ 2b{-}2{-}(r{+}1),\ \ldots,\ b{-}(r{+}1)$.

After round $b{-}1$: Segment 1 spans all $2b$ positions with items $q{+}2b{-}1, \ldots, q$ in decreasing order; segments 2 and 3 are empty. The merge is complete.

Cost calculation: Each round r sums an arithmetic series of b terms from $b{-}r$ to $2b{-}1{-}r$, giving per-round total $b(3b{-}1{-}2r)/2$. Summing over $r = 0, \ldots, b{-}1$ yields another arithmetic series from $b(3b{-}1)/2$ to $b(b{+}1)/2$, so the total cost is $b \cdot \frac{b(3b-1)/2+b(b+1)/2}{2} = b^3$. The swap count is $b \cdot b = b^2$. □

Lemma 8 (Level separation)**.** *At the start of level ℓ ($\ell = 1, \ldots, t{-}2$), the state consists of $n/b = 2^{t-\ell}$ reversed blocks of size $b = 2^\ell$, where block j contains items $\{(j{-}1)b{+}1, \ldots, jb\}$. All boundary costs between adjacent blocks equal $2b{-}1$. Consequently, FBS completes all merges at level ℓ before touching any boundary at a higher level, and merges proceed left to right within each level.*

Proof. At the start of level ℓ, the boundary between any two adjacent blocks has cost $2b{-}1$: the right end of each block holds its smallest item and the left end holds its largest, separated by $2b{-}1$.

After the leftmost merge (blocks $1, 2$) completes, the merged block of size $2b$ is fully reversed: all internal pairs are discordant and ineligible. The boundary between the merged block and the next unmerged block 3 has cost $3b{-}1$: the merged block's smallest item is 1 (at its right end) and block 3's largest item is $3b$ (at its left end), giving $3b - 1$. Since $3b{-}1 > 2b{-}1$ (for $b \geq 1$), FBS proceeds to the next intra-pair boundary (blocks $3, 4$, still at cost $2b{-}1$). The argument generalises: after any merge pair completes, the inter-pair boundary has cost $3b{-}1 > 2b{-}1$, so merges proceed left to right.

After all merges at level ℓ complete, the state consists of $n/(2b)$ reversed blocks of size $2b$, each with boundary cost $4b{-}1 = 2(2b){-}1$—precisely the starting configuration for level $\ell{+}1$. □

Proof of Proposition 6 (five-block structure). We proceed by induction on t.

Base case ($t = 2$, $n = 4$). $K^* = 3$. The FBS trace is: swap positions $(1, 2)$ at cost 1, then $(3, 4)$ at cost 1, then $(2, 3)$ at cost 3, producing $(2, 4, 1, 3)$. This matches: B1$= \{2\}$, B2$= \{4\}$, B3$= \{1\}$, B4$= \{3\}$, B5 empty.

Inductive step ($t \geq 3$).

Phase 1. By Lemma 5, the first m swaps produce m reversed blocks of size 2.

Levels 1 through $t{-}2$ (full merges). By Lemma 6, after phase 1 the eligible positions are the $m{-}1$ even indices, all at cost $3 = 2 \cdot 2{-}1$. These are the intra-pair boundaries for level-1 merges. By Lemma 8, FBS processes merges left to right at each level. By the Merge Lemma 7, each level-ℓ merge of two blocks of size $b = 2^\ell$ uses b^2 swaps at cost b^3. At level ℓ there are $n/(2b)$ merges, giving $2^{t-\ell-1} \cdot b^2 = 2^{t+\ell-1}$ swaps per level.

After level $t{-}2$, the state consists of two reversed blocks of size m: the left block at positions $1, \ldots, m$ with items $m, \ldots, 1$, and the right block at positions $m{+}1, \ldots, 2m$ with items $2m, \ldots, m{+}1$.

Swap budget. Total swaps through level $t{-}2$: $\sum_{k=t-1}^{2t-3} 2^k = 2^{t-1}(2^{t-1}{-}1) = m(m{-}1)$. Since $K^* = m(2m{-}1)/2$, exactly $K^* - m(m{-}1) = m/2$ swaps remain.

Top level (partial merge). The boundary at position m has cost $2m{-}1$. By the Merge Lemma's round-0 structure, item $n = 2m$ (leftmost element of the right block) propagates leftward from position $m{+}1$ through positions $m,\ m{-}1,\ \ldots,\ m/2{+}1$, displacing items $1, 2, \ldots, m/2$ one position to the right. This uses exactly $m/2$ swaps (the remaining budget) at costs $2m{-}1,\ 2m{-}2,\ \ldots,\ 3m/2$. The algorithm terminates because after K^* total swaps, $\Delta = \binom{n}{2} - 2K^* = 0 \leq \Delta^*$.

Final state. After the partial merge:

- Positions $1, \ldots, m/2$: items $m, m{-}1, \ldots, m/2{+}1$ (undisturbed upper portion of the left block). This is **B1**.
- Position $m/2{+}1$: item $2m$ (propagated here). This is **B2**.

- Positions $m/2{+}2, \ldots, m{+}1$: items $m/2, m/2{-}1, \ldots, 1$ (lower portion of the left block, shifted right by one). This is **B3**.
- Position $m{+}2$: item $2m{-}1$ (second element of the right block, unmoved since only item $2m$ propagated left). This is **B4**.
- Positions $m{+}3, \ldots, 2m$: items $2m{-}2, \ldots, m{+}1$ (remainder of the right block). This is **B5**.

This matches the five-block structure of Proposition 6. □

LP relaxation lower bound. To compare the FBS cost against a lower bound, we relax the constraint that the inverted pairs form a valid inversion set (i.e., arise from a permutation). Let $P^{\nearrow} := \{(i,j) : i < j,\ \alpha_{ij} = +1\}$ denote the set of concordant pairs in r_s. Introducing indicators $x_{ij} \in [0,1]$ for each concordant pair and dropping the transitivity constraint gives the *fractional knapsack* relaxation:

$$\mathrm{LP} := \min \sum_{(i,j)\in P^{\nearrow}} w_{ij}\, x_{ij} \qquad \text{s.t.} \qquad \sum_{(i,j)\in P^{\nearrow}} x_{ij} = K^*, \quad 0 \le x_{ij} \le 1.$$

The solution is immediate: set $x_{ij} = 1$ for the K^* pairs with smallest weight. Hence $\mathrm{LP} = \sum_{k=1}^{K^*} w_{(k)}$, where $w_{(1)} \le w_{(2)} \le \cdots$ are the sorted pair weights. Since LP relaxes constraints present in the true optimisation, $\mathrm{LP} \le d^*$.

Proposition 7 (LP closed form for $a = s$)**.** *For integer scores $s(i) = i$ with all pairs concordant, there are exactly $(n{-}d)$ pairs of weight d for each $d = 1, \ldots, n{-}1$. The LP selects pairs greedily from cheapest to most expensive:*

$$\mathrm{LP}(n) = \sum_{d=1}^{D-1} d\,(n{-}d) + D \cdot R,$$

where D is the smallest integer with $\sum_{d=1}^{D}(n{-}d) \ge K^*$ and $R := K^* - \sum_{d=1}^{D-1}(n{-}d)$.

Proof of Proposition 7. For $s(i) = i$, each pair (i,j) with $i < j$ has weight $w_{ij} = j{-}i =: d$. For each $d \in \{1, \ldots, n{-}1\}$ there are exactly $n{-}d$ pairs at distance d (namely $(i, i{+}d)$ for $i = 1, \ldots, n{-}d$). The LP selects the K^* cheapest pairs, so it fills all $n{-}1$ pairs of weight 1, then all $n{-}2$ pairs of weight 2, and so on through weight $D{-}1$, consuming $\sum_{d=1}^{D-1}(n{-}d)$ pairs at total cost $\sum_{d=1}^{D-1} d(n{-}d)$. The remaining $R = K^* - \sum_{d=1}^{D-1}(n{-}d)$ pairs are taken from the $n{-}D$ available pairs of weight D (note $R \le n{-}D$ by minimality of D), contributing $D \cdot R$. □

The critical weight satisfies $D = (2{-}\sqrt{2})\, m + O(1)$ as $m \to \infty$: solving $\sum_{d=1}^{D}(n{-}d) = K^*$, i.e. $Dn - D(D{+}1)/2 = m(2m{-}1)/2$, gives $D^2 - 2(2m{-}1)D + m(2m{-}1) = 0$, whose positive root is $(2{-}\sqrt{2})m + O(1)$. Despite being called "LP", the value requires no solver and is computed in $O(n)$ time.

Proof of Proposition 3 (Tight bound for the all-ascending case). Write $m = n/2$ and $c = 2 - \sqrt{2}$.

FBS cost. By Proposition 6, $d_{\mathrm{FBS}} = m(8m^2 + 21m - 14)/24 = m^3/3 + O(m^2)$.

LP lower bound. By Proposition 7, $\mathrm{LP} = \sum_{d=1}^{D-1} d(n{-}d) + D \cdot R$ with $D = cm + O(1)$. Expanding: $\sum_{d=1}^{D-1} d(2m{-}d) = c^2(1{-}c/3)\, m^3 + O(m^2)$. The remainder term $D \cdot R$ contributes $O(m^2)$, so $\mathrm{LP} = c^2(1{-}c/3)\, m^3 + O(m^2)$.

Convergence. The ratio satisfies

$$\frac{d_{\mathrm{FBS}}}{\mathrm{LP}} = \frac{m^3/3 + O(m^2)}{c^2(1{-}c/3)\, m^3 + O(m^2)} \longrightarrow \frac{1}{3c^2(1{-}c/3)}.$$

Substituting $c = 2{-}\sqrt{2}$: $c^2 = 6{-}4\sqrt{2}$, $1{-}c/3 = (1{+}\sqrt{2})/3$, so $3c^2(1{-}c/3) = (6{-}4\sqrt{2})(1{+}\sqrt{2}) = 6{+}6\sqrt{2}{-}4\sqrt{2}{-}8 = 2(\sqrt{2}{-}1)$. Hence the limit is $1/[2(\sqrt{2}{-}1)] = (1{+}\sqrt{2})/2 \approx 1.207$.

Strict 2-approximation. Since $\mathrm{LP} \le d^*$, it suffices to show $d_{\mathrm{FBS}} < 2\,\mathrm{LP}$ for all $t \ge 2$. For even $m \ge 2$, the $m/2$ cheapest weight classes contain $\sum_{d=1}^{m/2}(2m-d) = m(7m-2)/8 \le m(2m-1)/2 = K^*$ pairs (the inequality reduces to $m \ge 2$), so

$$\mathrm{LP} \;\ge\; \sum_{d=1}^{m/2} d\,(2m-d) \;=\; \frac{m(m+2)(5m-1)}{24}.$$

The condition $d_{\mathrm{FBS}} < 2\,\mathrm{LP}$ then becomes

$$\frac{m(8m^2+21m-14)}{24} \;<\; \frac{m(m+2)(5m-1)}{12},$$

which simplifies to $2m^2-3m+10 > 0$. Since the discriminant $9-80 = -71 < 0$ and the leading coefficient is positive, this holds for all real m. □

t	$n = 2^t$	d_{FBS}	LP	$d_{\mathrm{FBS}}/\mathrm{LP}$
2	4	5	3	1.667
3	8	33	22	1.500
4	16	222	160	1.387
5	32	1 580	1 205	1.311
6	64	11 800	9 348	1.262
7	128	90 928	73 587	1.236
8	256	713 312	583 900	1.222
9	512	5 649 600	4 652 075	1.214
10	1 024	44 968 320	37 139 950	1.211
11	2 048	358 830 848	296 812 700	1.209
12	4 096	2 866 980 352	2 373 273 400	1.208
13	8 192	22 921 169 920	18 981 273 200	1.208
15	32 768	1.47×10^{12}	1.21×10^{12}	1.207
17	131 072	9.38×10^{13}	7.77×10^{13}	1.207
19	524 288	6.00×10^{15}	4.97×10^{15}	1.207
21	2 097 152	3.84×10^{17}	3.18×10^{17}	1.207

Table 1: FBS cost vs. LP relaxation for $a = s$, $n = 2^t$, $t = 2, \ldots, 21$. The ratio converges to $(1+\sqrt{2})/2 \approx 1.207$.

E.4 Towards a proof of the 2-approximation

This appendix explains why the constant 2 in the 2-approximation conjecture is structurally motivated, outlines the proof path it suggests, and identifies the remaining open step.

Step 1: FBS and optimum live in the same Kendall ball. By Theorem 1, every feasible ranking has at least K^* inversions, and FBS achieves exactly K^*. Hence

$$K(r_{\mathrm{FBS}}) \;=\; K(r^*) \;=\; K^*,$$

i.e. both rankings sit on the sphere $\{r : |\mathrm{Inv}(r_s, r)| = K^*\}$. The two solutions therefore differ only in *which* K^* pairs are inverted, not in their cardinality.

Step 2: Diaconis–Graham yields a factor-2 ratio on the Spearman footrule. The classical Diaconis–Graham inequality [8] states that for every ranking r,

$$K(r) \;\le\; F(r) \;\le\; 2K(r), \qquad F(r) := \sum_i |r(i) - i| = \|\boldsymbol{\delta}\|_1.$$

Applying this to both rankings (which have $K = K^*$ by Step 1):

$$\|\boldsymbol{\delta}_{\mathrm{FBS}}\|_1 \;=\; F(r_{\mathrm{FBS}}) \;\le\; 2K^*, \qquad \|\boldsymbol{\delta}^*\|_1 \;=\; F(r^*) \;\ge\; K^*.$$

Dividing gives the *footrule-ratio bound*

$$\frac{\|\boldsymbol{\delta}_{\mathrm{FBS}}\|_1}{\|\boldsymbol{\delta}^*\|_1} \;\le\; 2.$$

Step 3: The Spearman–Kendall identity propagates the ratio to d. By Lemma 4, $d(r_s, r) = \frac{1}{2}\|\boldsymbol{\delta}\|_2^2$, so

$$\frac{d(r_s, r_{\mathrm{FBS}})}{d(r_s, r^*)} \;=\; \frac{\|\boldsymbol{\delta}_{\mathrm{FBS}}\|_2^2}{\|\boldsymbol{\delta}^*\|_2^2}.$$

Converting the footrule ratio of Step 2 into a squared-displacement ratio is straightforward when the two displacement profiles are comparably spread. A standard tool is Hölder's inequality

$$\|\boldsymbol{\delta}\|_2^2 \;\le\; \|\boldsymbol{\delta}\|_\infty \cdot \|\boldsymbol{\delta}\|_1,$$

which combined with Step 2 gives

$$d(r_s, r_{\mathrm{FBS}}) \;\le\; \tfrac{1}{2}\|\boldsymbol{\delta}_{\mathrm{FBS}}\|_\infty \cdot \|\boldsymbol{\delta}_{\mathrm{FBS}}\|_1 \;\le\; \|\boldsymbol{\delta}_{\mathrm{FBS}}\|_\infty \cdot K^*.$$

A factor-2 bound on d then follows whenever $\|\boldsymbol{\delta}_{\mathrm{FBS}}\|_\infty / \|\boldsymbol{\delta}^*\|_\infty$ is bounded, i.e. whenever FBS does not concentrate its displacement on a single item disproportionately more than the optimum does.

Why the constant cannot be lowered through this route. The right-hand side $F \le 2K$ in the Diaconis–Graham inequality is *tight*: it is saturated, for example, by the long shift $r = (2, 3, \ldots, n, 1)$, which has $K = n - 1$ inversions and $F = 2(n-1)$. Consequently, no proof going through Diaconis–Graham can yield a constant below 2. Tightening the conjecture would require an inequality that exploits the specific structure of r_{FBS}—in particular, the fact that FBS's min-cost selection rule produces *spread-out* displacement profiles, while the optimum may concentrate displacement on a few items.

The remaining open step: anti-concentration of FBS. The structural property still to be formalised is that FBS's min-cost rule prevents displacement concentration: an item that has already been displaced far creates expensive adjacent pairs, which FBS avoids in favour of cheaper swaps elsewhere. Concretely, we conjecture an *injective charging argument*: every swap that contributes a large term to $\|\boldsymbol{\delta}_{\mathrm{FBS}}\|_2^2$ can be paired with at least one earlier swap of comparable cost in r^*. Combined with Steps 1–3 this would close the proof.

Empirical support. For the all-ascending case $a = s$ with $\psi = 1$, the strict 2-approximation is proved analytically (Proposition 3), with the ratio converging to $(1+\sqrt{2})/2 \approx 1.207$. For arbitrary attribute arrays, the exhaustive verification at $n \le 9$ (worst ratio 1.46) together with MILP-exact computation up to $n = 100$ ($d_{\mathrm{FBS}}/d^* \le 1.31$ across 542 instances; LP-bound ≤ 1.32 across 612 instances, Appendix I) strongly supports the conjecture, and indeed suggests that the sharp constant may be substantially below 2.

F Proof of Section 8

Proof of Proposition 4 (unweighted case). **Initialisation (lines 1–6).** Sorting by score costs $O(n \log n)$. Computing the sign array σ costs $O(n)$. The Kendall counts $N_\nearrow$ and $N_\searrow$ are each computed by merge-sort–based counting in $O(n \log n)$. Building the linked list L is a single pass over σ: $O(n)$. Total initialisation: $O(n \log n)$.

Main loop (lines 7–24). The loop executes exactly K^* iterations (Proposition 1). In each iteration: retrieving an element from L is $O(1)$; the swap itself is $O(1)$; updating σ for three neighbours and maintaining L (at most 3 insertions/deletions in a doubly-linked list) is $O(1)$; updating $N_\nearrow, N_\searrow$ is $O(1)$. Total per iteration: $O(1)$. Total for the loop: $O(K^*)$.

Overall time (part (i)). $O(n \log n) + O(K^*) = O(n \log n + K^*)$. In the worst case $K^* = \Theta(n^2)$ and the $n \log n$ term is absorbed, giving $O(n^2)$.

Space (part (ii)). The algorithm stores: the arrays $R[\cdot]$, $A[\cdot]$ ($O(n)$); the sign array $\sigma[1..n-1]$ ($O(n)$); the doubly-linked list L with at most $n-1$ entries ($O(n)$); and $O(1)$ scalars. Total: $O(n)$. □

Proof of Proposition 4 (weighted case). **Initialisation (lines 1–7).**

- *Line 1 (sort).* Sorting the n columns by score costs $O(n \log n)$.

- *Lines 2–3 (adjacent costs and signs).* Each array has $n-1$ entries computed in $O(1)$ each, giving $O(n)$ total.
- *Line 4 (Kendall counts).* Computing $N_{\nearrow} = |\{i < j : A[i] < A[j]\}|$ and $N_{\searrow} = |\{i < j : A[i] > A[j]\}|$ independently, each by a merge-sort–based count, costs $O(n \log n)$.
- *Lines 5–6 (threshold and majority sign).* $O(1)$ each.
- *Line 7 (heap construction).* Build a min-heap H over the $\leq n-1$ entries k with $\sigma[k] = \mu$, keyed by $(c[k], k)$. Bottom-up heap construction costs $O(n)$.

Total initialisation: $O(n \log n)$.

Main loop (lines 8–28). By Proposition 1 the loop executes exactly K^* iterations. In each iteration:

- *Line 9 (extract minimum).* EXTRACT-MIN(H) returns k^* in $O(\log n)$.
- *Line 10 (swap).* The swap itself is $O(1)$.
- *Lines 11–26 (neighbour updates and heap maintenance).* Exactly 3 entries $\ell \in \{k^*-1,\ k^*,\ k^*+1\}$ are inspected (fewer at the boundaries). For each:
  - Recompute $c[\ell]$ and $\sigma[\ell]$ in $O(1)$.
  - Perform at most one heap operation (insert, delete, or decrease-key), each costing $O(\log n)$.

  Total per iteration: at most 3 heap operations, costing $O(\log n)$.
- *Line 27 (Kendall counts).* The global counters $N_{\nearrow}, N_{\searrow}$ are updated in $O(1)$.

Total for the loop: $O(\log n) \times K^* = O(K^* \log n)$.

Overall time (part (i)). $O(n \log n) + O(K^* \log n) = O\big((n + K^*) \log n\big)$.

Worst case. The maximum value of K^* occurs when $\Delta_0 = \binom{n}{2}$ (all pairs concordant, e.g. $a \equiv r_s$) and $\Delta^* = 0$ (or 1), giving $K^* = \lfloor \binom{n}{2}/2 \rfloor = \Theta(n^2)$. The overall time is then $O(n^2 \log n)$.

Space (part (ii)). The algorithm stores: the arrays $R[\cdot]$, $S[\cdot]$, $A[\cdot]$ (input, $O(n)$); the auxiliary arrays $c[1..n-1]$ and $\sigma[1..n-1]$ ($O(n)$ each); the min-heap H with at most $n-1$ entries ($O(n)$); the scalars $N_{\nearrow}$, $N_{\searrow}$, Δ^*, k^*, μ ($O(1)$). Total auxiliary space: $O(n)$. □

G Datasets

For the real-world experiments we use four datasets commonly adopted in the fair-ranking literature. For each, we report the chosen score, the chosen continuous protected attribute, the size n after removing missing entries, and the initial fairness ϕ_0 (the value of ϕ on the score-sorted ranking). All datasets are publicly available; preprocessing is intentionally minimal to keep results reproducible.

COMPAS [1]. The ProPublica COMPAS dataset of recidivism risk assessments. *Score:* `decile_score` (integer 1–10, higher = higher predicted risk). *Continuous attribute:* `age` (years). After dropping rows with missing values, $n = 7{,}214$ and $\phi_0 = 0.72$.

German Credit [17]. The Statlog (German Credit) dataset from the UCI repository. *Score:* `credit_amount` (Deutschmark, continuous). *Continuous attribute:* `age` (years). $n = 1{,}000$ (no missing values), $\phi_0 = 0.98$.

Law School [36]. The LSAC National Longitudinal Bar Passage Study. *Score:* `lsat` (LSAT score, continuous). *Continuous attribute:* `ugpa` (undergraduate GPA, continuous). After dropping rows with missing values, $n = 18{,}692$ and $\phi_0 = 0.85$.

Communities and Crime [32]. A community-level dataset from the UCI repository combining socio-economic data from the 1990 US Census, 1995 FBI Uniform Crime Report, and 1990 Law Enforcement Management and Administrative Statistics survey. All variables are normalized to $[0, 1]$. *Score:* `ViolentCrimesPerPop` (continuous, normalized violent-crime rate per population). *Continuous attribute:* `PctPopUnderPov` (continuous, fraction of the population below the poverty line). $n = 1{,}994$, $\phi_0 = 0.56$.

Score and attribute choice. For COMPAS and German Credit we follow the standard fair-ranking convention of using the dataset's primary risk/utility score and treating age as the continuous protected attribute. For Law School we follow the same convention used for predicting bar passage, with LSAT as the score and undergraduate GPA as the continuous attribute on which the ranking should remain balanced. For Communities and Crime, all numerical variables are continuous and pre-normalized; we choose the violent-crime rate as the score (the natural ranking target in this dataset) and the poverty rate as the protected attribute, motivated by the well-documented correlation between poverty and reported crime rates [20]. This choice yields the lowest initial fairness among the four datasets ($\phi_0 = 0.56$) and therefore exhibits the richest fairness–cost trade-off curve.

H Baselines

We compare FBS against four baselines. Three are existing group-fair re-ranking algorithms that require a discrete protected group; we adapt them to the continuous setting by quantile-binning the attribute a into G equal-frequency bins. The fourth, *Blend*, is a continuous heuristic of our own that interpolates between the score order and a perfectly-fair target order. All baselines are evaluated on the same axes as FBS: achieved continuous fairness ϕ (computed on the original undiscretized attribute) and weighted Kemeny distance $d(r_s, r)$ from the score-based ranking.

DetConstSort [13]. A deterministic constrained-sorting algorithm that enforces proportional representation of each protected group within every prefix of the output ranking. Given group proportions $p_g = |\{i : g(i) = g\}|/n$, it iteratively appends the highest-scoring remaining item subject to the constraint that, for every group g, the count in the current prefix is at least $\lfloor p_g \cdot k \rfloor$ for prefix length k. We use the implementation provided by FairRankTune [4]. We report results for $G \in \{2, 3, 5, 10\}$.

FA*IR [39]. A re-ranking algorithm that guarantees the protected group meets a minimum representation in every prefix, certified by a statistical test with significance $\alpha = 0.1$. FA*IR is intrinsically binary: a single "protected" group versus the rest. We restrict it to $G = 2$ (median split on a), assigning as protected the group most under-represented in the top half of the score-sorted ranking. Since the mtable computation is exponential in k, in the Pareto experiments, we use $k = 400$, as suggested by the authors. Furthermore, we use the implementation provided by fairsearchcore [22].

Median-split interleaving (ours). A simple binary baseline that splits items at the median attribute value into two groups, sorts each group by descending score, and builds the output ranking by alternately picking the next highest-scored item from each group. When one group is exhausted, the remainder of the other is appended. This serves as a sanity-check baseline: it always achieves close-to-perfect attribute balance on the binary partition but ignores both score gaps and the continuous nature of a.

Blend (continuous heuristic, ours). Unlike the above, Blend does not discretize a. It computes, for each item i: (a) its *score-position* $\rho_s(i)$, the rank of i in the score-sorted order; (b) a *fair-position* $\rho_f(i)$, obtained by sorting items by a and interleaving the high-attribute and low-attribute halves to produce a permutation with $\phi \approx 1$. The output ranking is then obtained by sorting items by the convex combination

$$\rho_\beta(i) \;=\; \beta\,\rho_s(i) \;+\; (1-\beta)\,\rho_f(i), \qquad \beta \in [0,1].$$

At $\beta = 1$ this reduces to the original score order ($\phi = \phi_0$, $d = 0$); at $\beta = 0$ it returns the perfectly-fair target order ($\phi \approx 1$, d potentially very large). Sweeping β traces a curve in the fairness–cost plane. Because the blend is between integer ranks rather than scores, Blend is *cost-blind*: a swap between two items with very different scores costs the same in the blend space as a swap between two near-equal-score items. This is the main reason Blend underperforms FBS on real data with

skewed score distributions, where greedy minimum-cost swapping (FBS) keeps the cumulative cost much smaller for the same fairness gain.

Discretization information loss. The first three baselines depend on a discretization parameter G. Increasing G allows finer fairness control but also increases the number of constraints and hence the cost of satisfying them. Crucially, even at $G = 10$ the algorithm is blind to within-bin variation of a, while FBS optimises the continuous fairness measure ϕ at the pairwise level.

I MILP formulation for the exact optimum

Section 6 (*Exact MILP optimum for medium* n) reports the empirical approximation ratio d_{FBS}/d^* obtained by solving the weighted Kemeny problem with fairness constraint exactly. Here we describe the MILP formulation used.

For each unordered pair (i, j) with $i < j$ we introduce a binary variable $x_{ij} \in \{0, 1\}$ that equals 1 if and only if the pair is inverted in the output ranking with respect to the score-sorted ranking r_s. Writing $\alpha_{ij} = \mathrm{sgn}(a(j) - a(i)) \in \{-1, +1\}$ and $D_0 = \sum_{i<j} \alpha_{ij}$, the problem reads:

$$\min_{x \in \{0,1\}^{\binom{n}{2}}} \sum_{i<j} |s(i) - s(j)|\, x_{ij} \qquad \text{s.t.} \quad \Big|D_0 - 2 \sum_{i<j} \alpha_{ij}\, x_{ij}\Big| \;\le\; \Delta^*,$$

together with the standard FAS-style triangle (transitivity) constraints

$$x_{ij} + x_{jk} - x_{ik} \le 1, \qquad -x_{ij} - x_{jk} + x_{ik} \le 0 \qquad \forall\, i < j < k,$$

which together rule out the only two binary inversion patterns inconsistent with a permutation. The fairness constraint is encoded as the pair of linear inequalities $\alpha^\top x \le (D_0 + \Delta^*)/2$ and $\alpha^\top x \ge (D_0 - \Delta^*)/2$.

For $n \in \{20, 50, 100\}$ this yields a MILP with $\binom{n}{2}$ binary variables and $2 + 2\binom{n}{3}$ linear constraints (e.g., 190 binaries and 2,282 constraints at $n = 20$; 1,225 binaries and 39,202 constraints at $n = 100$). We solve each instance with HiGHS via `scipy.optimize.milp`, capping the solver at 180 s with a 0.1% relative-gap tolerance, and additionally wrapping every solve in a 300 s subprocess wall-clock cap. We generate 155 instances per n (100 random, 50 correlated, 5 reversed). Across the 542 instances solved within the limits, the worst observed exact ratio is 1.31, attained on a correlated configuration at $n = 50$, and the LP relaxation upper-bounds the ratio by 1.32 across 612 instances. Together with the exhaustive enumeration up to $n = 9$ (worst ratio 1.46 over 699,282 instances), this provides strong empirical evidence for the 2-approximation conjecture.

J Additional experimental results

J.1 Approximation ratio for the all-ascending case

Proposition 3 predicts that, for the all-ascending configuration $a = s$ and $n = 2^t$, the ratio $d_{\mathrm{FBS}}/\mathrm{LP}$ converges to $(1+\sqrt{2})/2 \approx 1.207$. We compute the closed-form FBS cost (Proposition 6) and the LP relaxation bound for each $n = 2^t$ with $t = 2, \ldots, 21$ (up to $n \approx 2 \times 10^6$). Figure 4 shows the convergence: the ratio decreases monotonically from 1.67 (n=4) and stabilises at 1.207 for $t \ge 10$.

For general (non-power-of-two) n, the five-block structure of Proposition 6 does not apply, and the ratio exhibits a persistent oscillation. Figure 5 shows $d_{\mathrm{FBS}}/\mathrm{LP}$ for $n = 4, \ldots, 200$ (every integer) and selected values up to $n = 75{,}000$, all with $a = s$ and Δ^* set to the tightest feasible value. The local minima, occurring near powers of two, approach the $(1+\sqrt{2})/2$ limit, while the local maxima reach ≈ 1.42 but remain well below 2.

J.2 Synthetic baseline comparisons

We generate synthetic instances with $n \in \{100, 1000, 5000\}$ and score–attribute correlation $\mathrm{corr} \in \{0, 0.5, 1\}$, averaging over 5 repetitions. FBS consistently traces a Pareto front that dominates all baselines: it achieves the same fairness at lower cost, or higher fairness at the same cost. The gap is most pronounced at high correlation ($\mathrm{corr} = 1$), where the discretization baselines pay a substantial

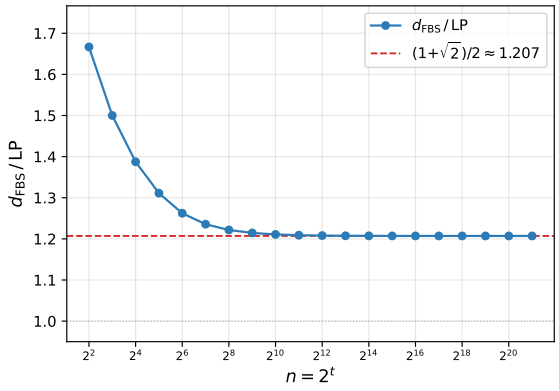


Figure 4: Ratio $d_{\mathrm{FBS}}/\mathrm{LP}$ for the all-ascending case $a = s$ with $n = 2^t$, $t = 2, \ldots, 21$. The dashed red line marks the theoretical limit $(1+\sqrt{2})/2 \approx 1.207$ from Proposition 3.

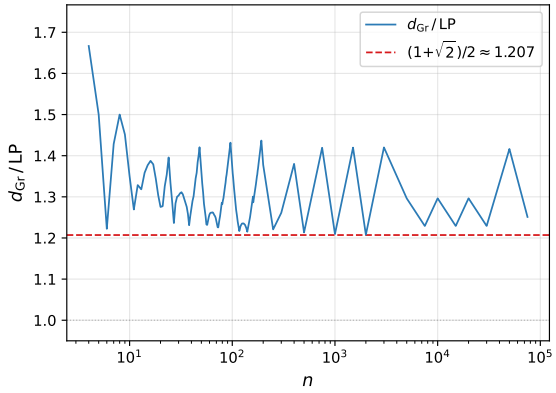


Figure 5: Ratio $d_{\mathrm{FBS}}/\mathrm{LP}$ for the all-ascending case $a = s$ with arbitrary n (dense for $n \leq 200$, sparse up to $n = 75{,}000$). The ratio oscillates but stays bounded; local minima near powers of two approach the $(1+\sqrt{2})/2$ limit (dashed red line).

cost to reach even moderate ϕ values, while FBS reaches $\phi = 1$ along a smooth, efficient trade-off curve. DetConstSort with more groups ($G = 5, 10$) reaches higher ϕ than with fewer groups, but at a cost that exceeds FBS for the same fairness level. FA*IR's one-sided minimum-representation constraint provides only a modest fairness improvement, as it addresses group underrepresentation but not the symmetric balance that ϕ measures. These results confirm that discretizing a continuous attribute incurs an inherent information loss: group-fair methods can only balance across bin boundaries, while FBS directly optimises the continuous fairness measure at the pairwise level.

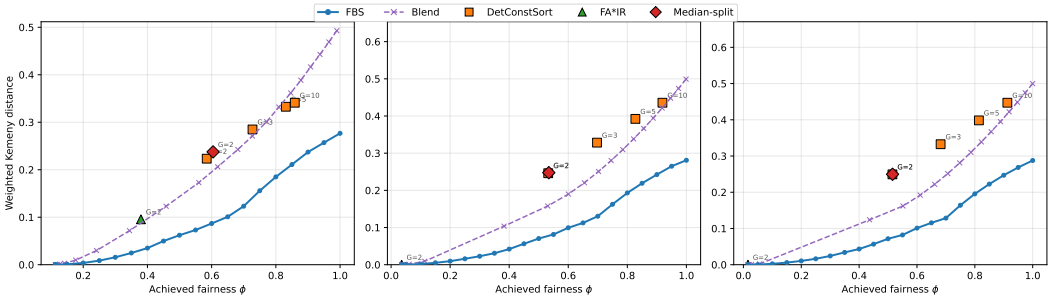


Figure 6: FBS vs. discretization baselines on synthetic data with $\mathrm{corr} = 0.5$, $n \in \{100, 1000, 5000\}$. Each panel shows achieved fairness ϕ (x-axis) vs. weighted Kemeny distance (y-axis). FBS (blue line) traces a Pareto front by sweeping ψ; baselines are shown as individual points for each number of groups G. FBS dominates all baselines.

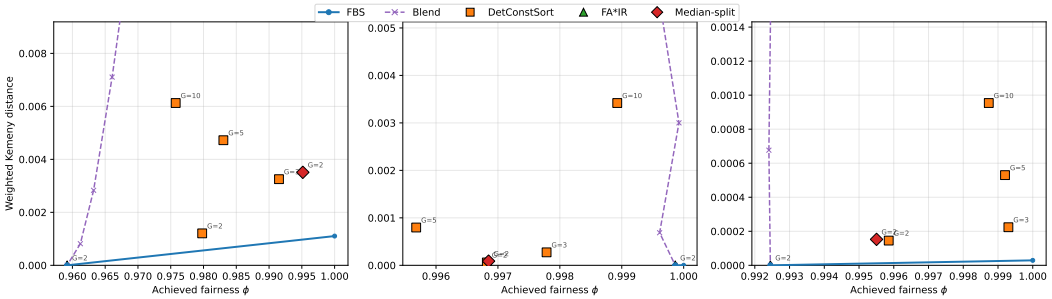


Figure 7: FBS vs. discretization baselines on synthetic data with $\mathrm{corr} = 0$ (uncorrelated scores and attributes), $n \in \{100, 1000, 5000\}$.

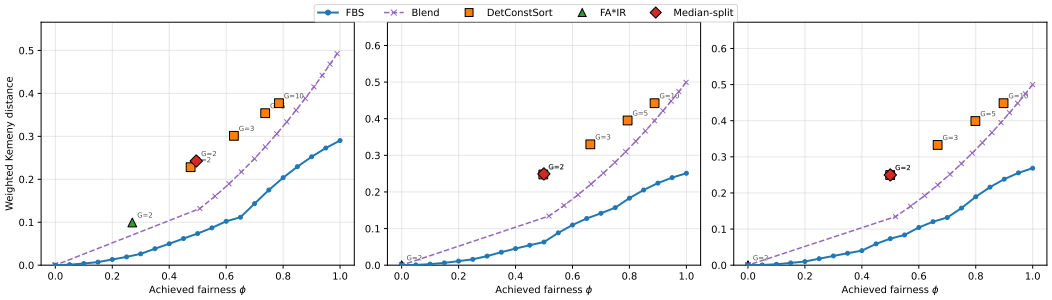


Figure 8: FBS vs. discretization baselines on synthetic data with $\mathrm{corr} = 1$ (perfectly correlated scores and attributes), $n \in \{100, 1000, 5000\}$.

J.3 Continuous-fairness repair baselines

The discretization baselines of Figure 2 enforce group constraints on a binned attribute. A different route to a fair ranking is to repair the score itself so that it no longer depends on the attribute, and then sort by the repaired score. We evaluate two such repair-then-sort baselines on the same data and axes as Figure 2: achieved fairness ϕ on the original continuous attribute, and weighted Kemeny distance from r_s.

Wasserstein repair. Following the geometric repair of [11], we bin the attribute into $G \in \{2, 3, 5, 10\}$ equal-frequency groups and move the score distribution of each group towards a common target, the pointwise median of the groups' quantile functions: an item at quantile q of its group receives the score $(1-\lambda)\, s + \lambda\, F^{-1}(q)$, where F^{-1} is the target quantile function and $\lambda \in \{0.25, 0.5, 0.75, 1\}$ is the repair strength ($\lambda = 0$ leaves r_s unchanged, $\lambda = 1$ gives every group the same score distribution). Items are then sorted by the repaired score.

Residual repair. In the spirit of [19], this variant works on the continuous attribute directly. We estimate the conditional mean of the score given the attribute, $\hat{f}(a)$, by a centred moving average of the scores over the items sorted by attribute (window of $\max(11, \lfloor n/50 \rfloor)$ items), and compute the residual $s - \hat{f}(a)$. Items are sorted by $(1-\lambda)\, \tilde{s} + \lambda\, \tilde{e}$, where $\tilde{s}$ and $\tilde{e}$ are the standardized score and residual and λ takes the same values as above; $\lambda = 1$ ranks by the residual alone, i.e. removes the local influence of the attribute on the score.

Figure 9 overlays the two families on the curves of Figure 2. For each repair ranking whose fairness lies within the range spanned by FBS, we compute the ratio between its cost and the cost of FBS at the same fairness, obtained by linear interpolation of the FBS curve; a ratio above one means that FBS reaches the same fairness at lower cost. Table 2 reports, for each family and dataset, the fairest ranking it produces and the median and maximum ratio. FBS is cheaper at every matched point on all four datasets. The Wasserstein repair comes closest (smallest ratio 1.05, on Law School; median ratios between 1.3 and 2.1 outside German Credit); the residual repair pays 1.4 to 2.6 times the FBS cost on COMPAS and Communities and far more on Law School, where removing the influence of the attribute reorders a large part of the list. The large ratios on German Credit reflect its near-fair starting point ($\phi_0 = 0.98$), where FBS reaches $\phi = 1$ at a cost of about 10^{-4}.

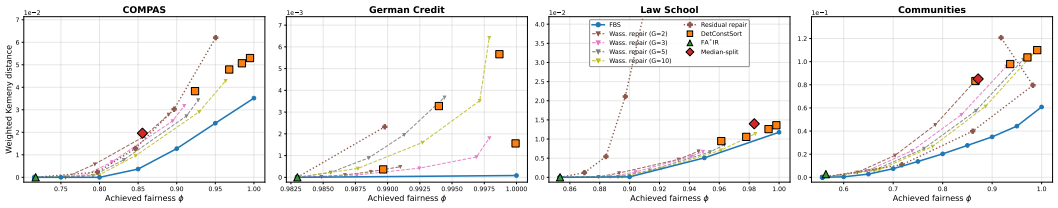


Figure 9: FBS vs. repair-then-sort baselines on the real datasets; axes as in Figure 2. Dashed lines: Wasserstein repair for $G \in \{2, 3, 5, 10\}$, sweeping the strength λ; dotted line with crosses: residual repair, sweeping λ; the discretization baselines of Figure 2 are shown as points for reference. Each panel shows the rankings at least as fair as r_s; on German Credit, three of the four residual-repair rankings are less fair than r_s, since the repair removes the influence of the attribute on the score rather than targeting ϕ, and are omitted. On Law School the y-axis is cut at three times the largest cost of the other methods; the residual repair continues beyond it, up to a cost of 0.20 at $\phi = 1$ (Table 2).

J.4 Ranking quality beyond the Kemeny distance: NDCG and FBS-NDCG

The weighted Kemeny distance charges every inverted pair by its score gap, wherever the pair sits in the list. NDCG with the original score as relevance is a second utility measure that weighs the top of the list more heavily: r_s is the ideal ranking and has NDCG equal to one, the relevance of an item is its score (linear gain), and position p is discounted by $\delta_p = 1/\log_2(p+1)$. We report NDCG over the full list and truncated at the top 10 and top 100 positions.

FBS as a cost-agnostic framework. At each step, FBS performs the cheapest fairness-improving adjacent swap under a per-swap cost, and any utility whose change under one adjacent swap is cheap

Baseline	Dataset	ϕ_0	ϕ reached	cost vs. FBS (median)	cost vs. FBS (max)
Wasserstein repair	COMPAS	0.72	0.96	2.1×	5.8×
	German Credit	0.98	1.00	18×	88×
	Law School	0.85	0.98	1.3×	11×
	Communities	0.56	0.99	2.0×	2.8×
Residual repair	COMPAS	0.72	0.95	2.6×	3.8×
	German Credit	0.98	0.99	70×	70×
	Law School	0.85	1.00	49×	189×
	Communities	0.56	0.98	1.4×	3.2×

Table 2: Repair-then-sort baselines vs. FBS at matched fairness. "ϕ reached" is the fairest ranking the family produces; the cost ratio is the repair cost divided by the FBS cost at the same ϕ, with median and maximum taken over the family's rankings within the fairness range of FBS.

to evaluate can serve as this cost. Algorithm 2 uses the score gap $|S[k] - S[k+1]|$ of the two items as the cost and minimizes the weighted Kemeny distance, with the guarantees of Theorem 1 and Proposition 2. Replacing the key of the heap by the decrease of DCG caused by the swap, $(S[k] - S[k+1])(\delta_k - \delta_{k+1})$, makes swaps near the top of the list expensive, so the greedy protects the head of the list; we call this variant FBS-NDCG. Both variants perform exactly K^* majority-sign swaps and reach the same fairness; they differ only in which swaps they choose, that is, in where the displacement is spent. FBS-NDCG is a greedy heuristic for NDCG.

Table 3 reports NDCG at matched fairness. For each dataset we take the discretization baseline of Figure 2 with the highest achieved fairness (DetConstSort with $G = 10$; $G = 3$ on German Credit) and evaluate FBS and FBS-NDCG at the same ϕ by linear interpolation along their ψ-sweeps (ψ from 0 to 1 in steps of 0.1). As a reference we also report a uniformly random permutation, averaged over 10 draws: it has $\phi \approx 1$ but is otherwise arbitrary, so its NDCG calibrates the scale. Two observations follow. First, fair rankings remain practical: the random fair ranking loses most of its top-heavy utility (NDCG@10 between 0.18 and 0.78), while FBS keeps NDCG@10 between 0.82 and 1.00 and full-list NDCG at or above 0.98 on every dataset. Second, FBS keeps the full-list displacement small but gives up some top-k utility, most visibly on the datasets with coarse scores (COMPAS, Law School), where it falls below the best baseline on NDCG@10 and NDCG@100. This is exactly the regime where FBS-NDCG, by pricing swaps near the top higher, recovers NDCG@10 and NDCG@100 of 1.00 at the same fairness, matching or exceeding the best baseline on every dataset and metric. Targeting top-k utility therefore requires only the NDCG cost inside the same routine.

Dataset (matched ϕ)	Metric	Random fair	Best baseline	FBS	FBS-NDCG
COMPAS (0.995)	NDCG	0.920	0.995	0.991	0.998
	NDCG@10	0.400	1.000	0.889	1.000
	NDCG@100	0.440	0.997	0.872	1.000
German Credit (1.000)	NDCG	0.790	0.999	1.000	1.000
	NDCG@10	0.210	0.992	1.000	1.000
	NDCG@100	0.290	0.996	1.000	1.000
Law School (0.998)	NDCG	0.980	1.000	0.999	1.000
	NDCG@10	0.780	1.000	0.932	1.000
	NDCG@100	0.780	1.000	0.951	1.000
Communities (0.991)	NDCG	0.810	0.951	0.980	0.995
	NDCG@10	0.180	0.944	0.826	1.000
	NDCG@100	0.240	0.753	0.943	1.000

Table 3: NDCG at matched fairness, with the original score as relevance (NDCG $= 1$ for r_s). For each dataset, the matched ϕ is the highest fairness reached by a discretization baseline (DetConstSort with $G = 10$, or $G = 3$ on German Credit); "Best baseline" is that baseline, FBS and FBS-NDCG are evaluated at the same ϕ, and "Random fair" is a uniformly random permutation ($\phi \approx 1$), averaged over 10 draws.

J.5 Other measures of dependence between position and attribute

CPP measures monotone dependence between position and attribute (Appendix B). To check whether FBS also reduces dependence as measured by other statistics, we evaluate four measures between the position of an item in the ranking and its attribute value, all of which are zero under independence: the absolute Spearman correlation; distance correlation [35]; the Hilbert–Schmidt independence criterion (HSIC) [15] with Gaussian kernels whose bandwidth is the median pairwise distance, normalised by the geometric mean of the two self-HSIC terms so that it lies in $[0, 1]$; and a positional discrepancy, the largest one-dimensional Wasserstein distance between the attribute distributions of the top, middle and bottom thirds of the ranking, divided by the standard deviation of the attribute. The last three also detect non-monotone dependence. Distance correlation and HSIC are quadratic in n, so on Law School ($n = 18{,}692$) they are computed on one fixed random subsample of $8{,}000$ items, the same for every ranking of that dataset; the other two measures use all items.

Table 4 reports the four measures for the score ranking r_s and for the FBS ranking at full fairness ($\psi = 1$). On the three datasets whose initial ranking is far from fair (Communities, COMPAS, Law School), FBS reduces all four measures substantially, by factors between 1.6 and 11; on German Credit the four measures are already near zero at r_s and change only negligibly. FBS therefore improves fairness as measured by these statistics as well. Some non-monotone dependence remains at $\psi = 1$ (distance correlation and HSIC stay above zero), consistent with CPP measuring monotone dependence.

Dataset	ϕ_0	|Spearman ρ|	distance correlation	HSIC	positional discrepancy
Communities	0.56	0.62 → 0.19	0.57 → 0.36	0.300 → 0.143	1.26 → 0.40
COMPAS	0.72	0.42 → 0.13	0.41 → 0.23	0.131 → 0.042	0.94 → 0.48
Law School	0.85	0.22 → 0.06	0.20 → 0.10	0.032 → 0.003	0.47 → 0.15
German Credit	0.98	0.03 → 0.01	0.06 → 0.07	0.005 → 0.005	0.16 → 0.17

Table 4: Dependence between rank position and attribute for the score ranking r_s (left of each arrow) and for the FBS ranking at $\psi = 1$ (right); lower means less dependence. Distance correlation and HSIC on Law School are computed on a fixed subsample of $8{,}000$ items.

K Hardware, Software and Code

All experiments were run on a single machine with the following specifications:

- **Hardware:** Apple MacBook Pro with Apple M1 Pro chip, 8-core CPU, 16 GB unified memory.
- **OS:** macOS 15.6.1 (Sequoia).
- **Code:** The source code to run FBS and to reproduce the experiments is publicly available at `https://github.com/Ambress92/FairBubbleSort`.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction list five contributions (CPP measure, FBS algorithm, optimality/approximation theorems, complexity analysis, empirical evaluation) that are substantiated in Section 3, the Method section, Section 5–8, and the Experiments section. Limitations and open conjectures are explicitly stated.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 9 contains a dedicated Limitations paragraph discussing the specific limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All theorems, lemmas, propositions, and conjectures are numbered and cross-referenced. Assumptions (e.g., general position, integer scores) are stated in Section 3 and in each result's statement. Complete proofs are provided in the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Full algorithmic pseudocode is given in Algorithms 1 and 2. Experimental settings (datasets, baselines, parameter sweeps, solvers, time limits) are detailed in the Experiments section and Appendices G, H, and I.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The source code to reproduce the experiments is publicly available at `https://github.com/Ambress92/FairBubbleSort`. All datasets used are publicly available (see Appendix G).

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The Experiments section specifies all parameters: fairness threshold sweeps, number of repetitions, MILP solver settings (time limits, gap tolerance), correlation regimes, and input sizes. Dataset-specific details (score and attribute choice, preprocessing) are in Appendix G; baseline configurations are in Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: For the scalability experiment with random attribute vectors (Figure 3), we report 1σ error bars (standard deviation) over 10 independent runs with different random instances. The remaining experiments are either deterministic (fixed correlation inputs) or exhaustive (all permutations for $n \leq 9$), so error bars are not applicable.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments were run on a single Apple MacBook Pro (M1 Pro, 8-core CPU, 16 GB unified memory). Hardware and software details are reported in Appendix K. Wall-clock times for the scalability experiments are shown in Figure 3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research proposes a fairness-enhancing algorithm and uses only publicly available datasets. No human subjects, deception, or privacy-sensitive data collection is involved.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section A discusses both positive and negative societal impacts of the proposed approach.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper proposes a ranking algorithm and does not release pre-trained models, generative systems, or scraped datasets that pose misuse risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets and code libraries used are properly cited. Dataset sources and preprocessing are described in Appendix G; baseline implementations and their libraries are credited in Appendix H.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not release new datasets or pre-trained models. The proposed algorithm is fully specified via pseudocode in Algorithms 1 and 2.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as a component of the core methodology in this research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.