Technical Report AC-TR-25-001

August 2025

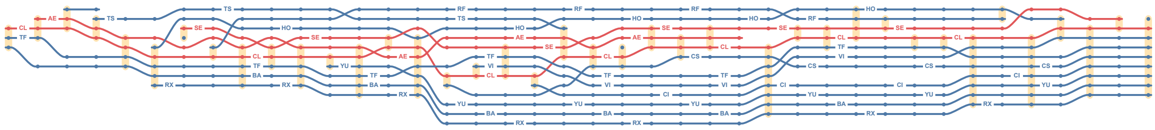
F²Stories: A Modular Framework for Multi-Objective Optimization of Storylines with a Focus on Fairness

Tommaso Piselli, Giuseppe Liotta, Fabrizio Montecchiani, Martin Nöllenburg, Sara Di Bartolomeo

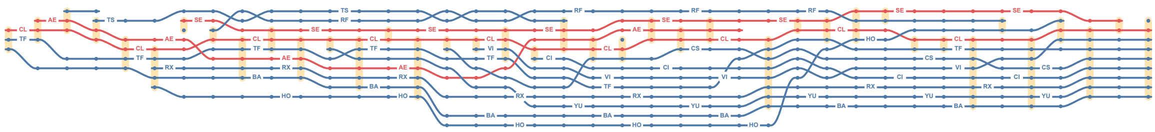
F²Stories: A Modular Framework for Multi-Objective Optimization of Storylines with a Focus on Fairness

Tommaso Piselli, Giuseppe Liotta, Fabrizio Montecchiani, Martin Nöllenburg, Sara Di Bartolomeo

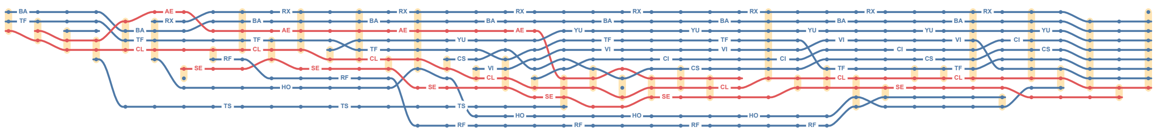
Technical Report AC-TR-25-001



`standardMode` is the default mode that minimizes global aesthetic criteria, in this case resulting in 26 crossings overall. This optimization creates an unfair distribution with 2.21 crossings on average per blue character, while red characters face a far higher value of 7 crossings each on average.



`fairnessMode` distributes the amount of crossings equitably between the two groups of characters, resulting in an average of 4 crossings for both blue and red characters. This fairness is achieved with an increase of the total number of crossings from 26 to 34.



`focusMode` prioritizes the readability of one group (red characters) while maintaining good readability for the other group. Here, we have an average of 5 crossings per red character and 3.21 crossings per blue character. As a result, even though the total number of crossings went from 26 to 30, no two red characters cross each other, like they do in both previous solutions.

Fig. 1: *Storyline layouts of the same instance (Final Fantasy VII) using three different optimization modes of F²Stories. There are 3 main characters in the red group and 14 secondary characters in the blue group. The three layouts share the same general optimization goal (minimize crossings), but with different emphasis on how crossings are distributed among both character groups.*

Abstract—Storyline visualizations represent character interactions over time. When these characters belong to different groups, a new research question emerges: how can we balance optimization of readability across the groups while preserving the overall narrative structure of the story? Traditional algorithms that optimize global readability metrics (like minimizing crossings) can introduce quality biases between the different groups based on their cardinality and other aspects of the data. Visual consequences of these biases are: making characters of minority groups disproportionately harder to follow, and visually deprioritizing important characters when their curves become entangled with numerous secondary characters. We present F²Stories, a modular framework that addresses these challenges in storylines by offering three complementary optimization modes: (1) `fairnessMode` ensures that no group bears a disproportionate burden of visualization complexity regardless of their representation in the story; (2) `focusMode` allows prioritizing a group of characters while maintaining good readability for secondary characters; and (3) `standardMode` globally optimizes classical aesthetic metrics. Our approach is based on Mixed Integer Linear Programming (MILP), offering optimality guarantees, precise balancing of competing metrics through weighted objectives, and the flexibility to incorporate complex fairness concepts as additional constraints without the need to redesign the entire algorithm. We conducted an extensive experimental analysis to demonstrate how F²Stories enables more fair or focus group-prioritized storyline visualizations while maintaining adherence to established layout constraints. Our evaluation includes comprehensive results from a detailed case study that shows the effectiveness of our approach in real-world narrative contexts. An open access copy of this paper and all supplemental materials are available at `osf.io/e2qvy`.

Index Terms—Storyline layouts, optimization, fairness

1 Introduction

The widespread adoption of automated decision-making systems has significantly changed the way in which everyday situations and problems are addressed and resolved. While these systems offer efficiency and scalability, they also pose a substantial risk in the potential to reflect and amplify existing biases in the society. These biases may arise from imbalanced training data that mirrors historical inequalities [42] or from the design of the algorithms themselves, unintentionally leading to unfair outcomes [1, 34]. As a result, ensuring *fairness* for affected individuals or groups has become a central focus in computer science research, where fairness is a multifaceted concept that can be expressed in several distinct forms. Efforts to address biases, both in data analysis and algorithm design, have driven the development of advanced computational methods aimed at mitigating disparities and promoting fairness in automated systems [28]. In particular, the focus on algorithmic fairness extends beyond decision-making systems and has been explored in

- *Tommaso Piselli, Giuseppe Liotta and Fabrizio Montecchiani are with the University of Perugia. E-mail: tommaso.piselli@dottorandi.unipg.it*
- *Martin Nöllenburg and Sara Di Bartolomeo are with TU Wien.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

various areas of computer science, including bias mitigation in machine learning models [5, 42], clustering algorithms [4, 16], dimensionality reduction [40], and computational social choice [23, 25]. However, the role of fairness in visualization has received comparatively less attention so far. Network visualization, in particular, presents unique challenges in ensuring fair representation of different entities within a dataset. When dealing with network data, elements such as nodes and edges may belong to different categories (e.g., species in a phylogenetic tree or reaction types in a metabolic network). If one category is significantly smaller than the others, applying a layout optimization algorithm to the entire network may disproportionately disadvantage the minority category. Consider visualizing character interactions in a movie where main characters (protagonists) interact frequently with both each other and supporting characters. Traditional optimization might create layouts where important character trajectories become obscured by the numerous interactions with secondary characters, making it difficult to follow the main narrative arc. However, particularly in settings where decision making is critical [2], it is important to provide a fair visual representation for all involved groups to facilitate a fair assessment and understanding of the network's properties across all categories and protected features.

Building on seminal research on fairness in data visualization [38], Eades et al. [13] have recently introduced the concept of *fairness in network visualization*, a research direction aimed at understanding how to integrate and evaluate these fairness concerns in different graph drawing paradigms. While Eades et al. investigated classical paradigms, namely straight-line and orthogonal graph drawings, our work addresses *storyline visualizations*, a popular visualization paradigm [44] to illustrate dynamic relationships among characters over time and to more easily visualize key interactions of characters or important events in the story (e.g., see Fig. 1 for the storyline visualization of the plot of a famous video-game, Final Fantasy VII). Traditional storyline layout algorithms optimize global metrics. For example, they minimize the total number of crossings or the total number of wiggles (vertical movements) across all characters. This behavior can inadvertently introduce biases against different groups of characters, based on their respective cardinality. Such biases manifest themselves in two key ways: first, characters from different groups may experience disproportionately more crossings and wiggles, making their trajectories harder to follow. Second, important characters in the story might be visually deprioritized when their curves get entangled with numerous secondary characters. While our methodology can be readily extended to handle multiple groups, in this work we focus on balancing fairness between two distinct groups of characters, which allows us to explore fundamental fairness trade-offs while keeping the mathematical formulation relatively simple to describe.

Our Contributions. We introduce `F`^{2}`Stories`, a modular framework with three complementary *optimization modes*:

`fairnessMode` implements our notion of fairness by ensuring no group experiences a disproportionate share of visual complexity, measured by relevant metrics, regardless of their proportional representation in the story. This mode is essential in analytical contexts where analysis requires equal attention to all character groups; bias in visual representation could lead to misinterpretation; or the goal is to reveal structural patterns across groups impartially.

`focusMode` addresses scenarios where certain characters (e.g., protagonists or characters of particular interest) warrant visual prominence. Rather than privileging a single always-present character as in prior work [19], our `focusMode` allows prioritizing an entire group of characters while maintaining acceptable readability for secondary characters. This is valuable when the analysis centers on a specific subset of characters; the narrative structure emphasizes certain character relationships; or users wish to highlight particular patterns within a complex storyline.

`standardMode` represents the conventional approach to storyline layout, where we optimize globally for aesthetic metrics without considering any character groups. This serves as our baseline for comparison and is suitable for cases where character grouping is either not present or not relevant to the analysis task.

`F`^{2}`Stories` is implemented as a modular framework, that enables users to switch between three optimization modes while maintaining compatibility with established storyline layout constraints. It offers quantifiable, adjustable trade-offs between global and group-specific metrics and provides the flexibility to enable, disable, or adjust the weight of different optimization components without redesigning the algorithm. This allows users to prioritize aspects of the visualization according to their specific needs.

To achieve precise control over these competing and generally NP-hard optimization objectives [24], we formulate our approach using Mixed Integer Linear Programming (MILP), a versatile exact approach in discrete optimization. While MILP is computationally more intensive compared to heuristic methods, it offers several critical advantages for our work. First, it provides *optimality guarantees*, ensuring provably optimal solutions given our constraints and objective function, which is crucial for obtaining the highest-quality layouts. Second, its *multi-objective flexibility* allows precise balancing of competing metrics through weighted objectives, enabling tunable trade-offs between fairness and traditional readability metrics. Finally, MILP's *constraint expressiveness* makes it possible to incorporate complex fairness concepts as additional constraints without requiring a complete redesign of the algorithm. Moreover, our work is in line with the best and most recent exact algorithms in the literature on storyline visualizations [11, 14, 17], since they are all based on integer programming. In summary, we propose:

- A new multi-objective MILP model for minimizing crossings, wiggles and skewness in storyline visualization that offers different optimization approaches: balancing fairness within two separate groups of characters, while also providing a flexible mechanism to emphasize narratively significant characters without sacrificing overall readability of the storyline.
- An open source implementation of the model, distributed on OSF and GitHub. Additionally, we provide an interactive tool for visualizing storylines, accessible at this link.
- Results of computational experiments on ten different storylines, one of which is reported as a case study in the main body of the paper, while the others can be found in the supplemental material.

2 Related Work

Storyline Visualization. Storyline visualizations depict narratives as temporal networks, representing characters as lines over time (x-axis). These visualizations, first popularized by an XKCD comic [29], place interacting characters close together to reflect shared moments. Following its debut, many works sought to automate XKCD's hand-drawn layouts, leading to a range of layout algorithms with various optimizations and specialized features. These draw on graph readability criteria [39], aiming to minimize crossings, line wiggles, and white space [44]. Because time maps to vertical layers, storyline layouts naturally align with layered graph drawing techniques. Many algorithms adapt the Sugiyama framework [41], often using barycentric heuristics [15, 41] while ensuring adjacency during interactions [17]. Greedy algorithms support applications beyond narratives, such as software evolution [33] and patient monitoring [2, 9]. Tanahashi et al. [43] introduced a streaming-compatible greedy method, while Van Dijk et al. [47] proposed minimizing block crossings. Other methods include HTN planning [36,37] and SAT-based formulations [10,48], or even a balance approach between hand-drawn storylines and automatic layouts [45].

We focus on approaches that formulate the problem as an Integer or Mixed-Integer Linear Program (ILP/MILP), where the goal is to optimize a linear objective under linear constraints, with some or all variables restricted to integers. The main challenge lies not only in solving the ILP itself, but in expressing layout rules—like spatial relationships, minimizing crossings, and preserving readability—as linear constraints. When well-formulated, (M)ILP yields globally optimal, clean visualizations, which is especially useful for small to medium datasets where precision matters, such as metro maps [31, 32]. The first ILP-based solution for storyline layouts was proposed by Gronemann et al. [17], and later extended by Dobler et al. [12], who introduced time intervals. Hegemann et al. [19] added the concept of a protagonist—an always straight, crossing-free line—shifting focus to a single character's clarity at the cost of fairness to others. Recent work also explores ILP models with improved efficiency [11, 50].

Fairness in Visualization. The rise of AI in sensitive domains has increased the demand for visualization tools that enhance transparency and fairness [3]. At the same time, these principles must apply to the visualizations themselves to avoid misleading human analysts. Recent work has introduced fairness constraints in low-dimensional projections [22, 26, 38, 46] and network layouts [13, 20]. Eades et al. [13] formalize fairness for orthogonal and straight-line graph drawings, using bend and stress distributions across groups. Their results show that fairness can be improved with minimal loss in readability by using multi-objective optimization.

3 Design of Fair Storyline Visualizations

Based on previous work on storyline optimization [10–12, 19, 43, 44] and on fairness in visualization [13, 38], we identified the following design requirements for F^2Stories. We use them as guidelines to create our modular multi-objective optimization function and its constraints.

Storyline Constraints:

- **[DR1]**: Characters are drawn as x-monotone curves and belong to one of two easily distinguishable groups.
- **[DR2]**: Multiple characters that interact in a given timestep must be drawn as a contiguous group.

Fairness Constraints:

- **[DR3]**: The visualization should maintain narrative coherence by keeping character lines visually stable and clear from crossings.
- **[DR4]**: The visual complexity, as measured by metrics such as crossings, should be fairly distributed across different groups.
- **[DR5]**: When narratively appropriate, the visualization should support emphasizing a specific group of characters to have visual predominance and reduced visual complexity.

Optimization Goals and Metrics. Design requirements [DR1] and [DR2] are general storyline constraints we guarantee by design. To satisfy the other DRs, we focus on three main *optimization goals* for storyline visualizations: minimizing crossings, wiggles, and skewness.

Crossings occur when character curves intersect between timesteps, causing clutter and reducing readability. Minimizing them is key to clarity, as noted in graph drawing literature [39], and helps preserve flow. For this reason, nearly all prior works include crossing minimization in their objectives [10–12, 17, 19], directly supporting [DR3].

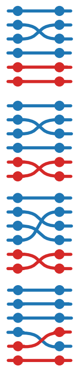
Wiggles are vertical shifts in character lines between timesteps. Too many create clutter and make trajectories harder to follow. While some are necessary, our model minimizes them to keep paths flat. Wiggles were mainly studied in early works [43, 44] and less so in MILP settings [14]. Reducing them improves visual stability and supports [DR3].

Skewness is the minimum number of characters to remove to make a layout crossing-free [21, 30]. Though studied in layered graphs [30], it hasn't been used in storyline layouts. Minimizing skewness reveals which characters add visual complexity and further supports [DR3].

These three optimization goals form the foundation of the `standardMode`, which focuses on global aesthetic metrics without considering character groups. This mode aims to produce visually coherent storylines with minimal visual complexity across the entire visualization. In our `fairnessMode`, we extend these metrics to incorporate fair distribution considerations, addressing design requirement [DR4]. As an introductory example, following an approach similar to [13], we define *crossing unfairness* for two groups: the majority (blue characters) and the minority (red characters). It is measured as the difference in crossings involving each group, weighted by their respective cardinalities (see Fig. 2 for an explanation). The goal is to minimize this unfairness, ensuring a more balanced distribution of crossings between the groups. We apply this same principle to define *wiggle unfairness* and *skewness unfairness*, extending each base metric to account for the group-based distribution. In the pursuit of fairness, however, a trade-off emerges: global readability may be compromised to achieve balanced group readability. This behavior is commonly referred to as the *price of fairness* [13, 40]. We are also interested in studying scenarios where we optimize specific metrics for one group while maintaining acceptable global readability. This is the goal of the `focusMode`, which addresses

$$\text{unfairness} = \left| \frac{\text{crossings among reds}}{\text{number of reds}} - \frac{\text{crossings among blues}}{\text{number of blues}} \right|$$

Unfairness is computed based on the number of crossings, within each character group. A crossing between same-colored elements adds 2 to that group's total, while a crossing between different-colored elements adds 1 to each [35].



A single crossing among blue elements creates unfairness for the blue set, as it is the only one that has a crossing. In this case, unfairness is computed as: $|0/2 - 2/4| = 1/2$.

Although this case shows an equal number of red and blue crossings, it's still unfair: with more elements in the blue group, each crossing weighs less than in the red group. Here, unfairness is $|2/2 - 2/4| = 1/2$.

With 4 blue crossings and 2 red, unfairness is $|4/4 - 2/2| = 0$. This is the fairest configuration shown.

In cases where there is a crossing between a blue line and a red line, the crossing will be calculated as 1 for each group. Thus, unfairness in this case is $|1/4 - 1/2| = 1/4$.

Fig. 2: *An illustration of unfairness computed on different examples.*

design requirement [DR5] by deliberately targeting the optimization of aesthetic criteria for a specific focus group of characters without excessively degrading the same metrics outside the focus group. This mode is particularly useful when certain characters hold greater narrative significance and demand for an enhanced visual clarity.

4 Notation

We follow the notation to Dobler et al. [11] and refer to Fig. 3 for an illustration. A *storyline instance* is defined as a 4-tuple (T, C, $\mathscr{I}$, A) where: $T = \{1, 2, \dots, \ell\}$ is a set of ordered *time steps* (also called *layers*); $C = \{c_1, c_2, \dots, c_n\}$ is the set of *characters* of the story, with $n = |C|$; $\mathscr{I} = \{I_1, I_2, \dots, I_m\}$ is the set of *interactions* between characters, with $m = |\mathscr{I}|$; A is the *activity function* that defines when characters appear in the story. The characters in our storyline are divided into two distinct groups, according to [DR1], which we identify by colors red and blue. Formally, we define $C_R \subseteq C$ as the set of red characters and $C_B \subseteq C$ as the set of blue characters, where $C = C_R \cup C_B$ and $C_R \cap C_B = \emptyset$. In the following, when `fairnessMode` is active, C_R is used as the minority group; meanwhile, when `focusMode` is active, C_R serves as the focus group. We also define $n_R = |C_R|$ and $n_B = |C_B|$.

The *activity function* A maps each character $c \in C$ to a set $A(c) = \{i, i+1, \dots, j\}$ of consecutive time steps ($1 \leq i \leq j \leq \ell$) during which the character appears in the story. We say that c enters the story at time step i and exits after time step j and that c is *active* at the time steps in $A(c)$. From this, we define the set of active characters at a specific time step t as $AC(t) = \{c \in C \mid t \in A(c)\}$. We denote as $AC_R(t) = \{c \in C_R \mid t \in A(c)\}$ and $AC_B(t) = \{c \in C_B \mid t \in A(c)\}$ the set of red active characters and blue active characters, respectively. We also define the set of characters that remain active throughout an entire time interval $[i, j]$, with $1 \leq i \leq j \leq \ell$, as the intersection $AC(i, j) = AC(i) \cap AC(i+1) \cap \dots \cap AC(j)$. We define $AC_R(i, j)$ (and similarly $AC_B(i, j)$) as the set of red (blue) characters that are continuously present in the story from time i to time j. An *interaction* $I \in \mathscr{I}$ occurs at a specific time step $time(I) \in T$ and involves a set of characters, denoted as $char(I) \subseteq C$. For each interaction I, it holds that $char(I) \subseteq AC(time(I))$, meaning that only active characters can participate in interactions. For any time step $t \in T$, we define the set of all interactions occurring at t as $\mathscr{I}(t) = \{I \in \mathscr{I} \mid time(I) = t\}$ and the set of characters that are part of those interactions as $charSet(t) = \cup_{I \in \mathscr{I}(t)} char(I)$. We also define two sets that partition possible character crossings based on group membership: $\Gamma_R = \{(t, i, j) \mid t \in 1, \dots, \ell, i \in AC_R(t, t+1), j \in AC(t, t+1)\}$ and $\Gamma_B = \{(t, i, j) \mid t \in 1, \dots, \ell, i \in AC_B(t, t+1), j \in AC(t, t+1)\}$. These sets capture all potential crossings involving red and blue characters, respectively, across consecutive time steps.

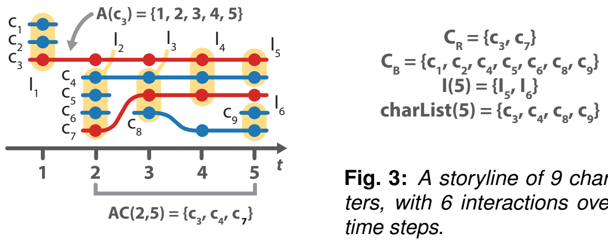


Fig. 3: *A storyline of 9 characters, with 6 interactions over 5 time steps.*

Given a set $X = \{x_1, \dots, x_n\}$, a *permutation* π is a linear order of the element of the set. It can also be seen as a bijective function from $\{1, 2, \dots, |X|\}$ to X. Given two elements $x, y \in X$, if x comes before y in the permutation π we write $x \prec_\pi y$.

Given a storyline instance $(T, C, \mathscr{I}, A)$, we call *solution* or *layout* of the instance a sequence $S = (\pi_1, \dots, \pi_\ell)$ where each π_t is a permutation of $AC(t)$ satisfying [DR2]: for every interaction $I \in \mathscr{I}(t)$, the characters in $char(I)$ must appear consecutively in π_t. Formally, for any $c_a, c_b \in char(I)$ and any $c \in AC(t) \setminus char(I)$, if $c_a \prec_{\pi_i} c_b$, then we either have $c \prec_{\pi_i} c_a$ or $c_b \prec_{\pi_i} c$. That is, characters participating in the same interaction at a given time step must form an uninterrupted sequence within the permutation for that time step.

5 The F^{2}Stories MILP Optimization Model

Modules. Our optimization model is structured around distinct modules, each consisting of a weighted term in the objective function and an associated set of constraints. Users can enable, disable, or adjust the weight of each module by modifying its coefficient. In Sec. 3 we introduced three optimization metrics: crossings, wiggles, and skewness, each with an associated (un)fairness metric. Each has two coefficients: one for the standard version and one for its fairness variant. These are denoted as λ_c and λ_c^f for crossings; λ_w and λ_w^f for wiggles; and λ_s and λ_s^f for skewness. We refer to the former as *standard coefficients*, the latter as *fairness coefficients*. To enhance readability, we use a consistent color-coding scheme that visually distinguishes modules and their fairness variants, with each pair sharing the same base hue and a different shade.

Modes. This paragraph presents three complementary modes for storyline visualization; each emphasizes different aspects of the layout and strives for distinct optimization goals. By adjusting module coefficients, users can create configurations that prioritize different objectives, ranging from single-metric optimizations to complex multi-objective scenarios balancing competing goals. The configurations used in our experiments are listed in Tab. 2 and discussed further in Sec. 6.

`standardMode` represents the conventional approach to storyline layout, focusing on global aesthetic metrics without considering character groups, and serves as our baseline. It is activated by setting non-zero values for standard coefficients while keeping fairness coefficients at zero. This mode implements optimization approaches comparable to state-of-the-art methods [17], extended to support our modular framework. It is implemented through three primary optimization terms:

- The cr term represents *crossings* between character curves. Its related coefficient is λ_c.
- The wg term represents vertical movements in character curves, called *wiggles*. Its related coefficient is λ_w.
- The sk term represents the *skewness* of the storyline. Its related coefficient is λ_s.

`fairnessMode` addresses inherent biases in traditional storyline optimization algorithms by balancing visual complexity across character groups. This mode is activated when fairness coefficients are set to high values, while maintaining standard coefficients at lower values. These are the fair-variant optimization terms of the previously presented ones:

- The cr^f term represents *crossing unfairness* through balancing crossing distributions between groups using a weighted approach where red-red crossings, blue-blue crossings, and mixed crossings are considered proportionally to group sizes. Its related coefficient is λ_c^f.
- The wg^f term represents the *wiggles unfairness*, i.e., the number of wiggles of the red characters compared to the number of wiggles of the blue characters, weighted by their group cardinality. Its related coefficient is λ_w^f.
- The sk^f term represents *skewness unfairness*, i.e., the number of characters to be removed to achieve planarity is balanced between the character groups, based on their respective cardinality. Its related coefficient is λ_s^f.

`focusMode` provides a targeted approach when specific character groups, such as protagonists or key figures in data analysis, require visual emphasis. This mode prioritizes the visual clarity of a designated focus group while preserving overall readability. Unlike `fairnessMode`, which aims for a balanced representation across all groups, `focusMode` creates a visual hierarchy that draws attention to significant characters. This emphasis is pursued through three mechanisms. Crossings: prioritizes clarity of focus character trajectories, with highest emphasis on crossings within the group. Wiggles: creates flatter curves for focus characters by emphasizing minimized vertical movement. Skewness: gives higher optimization priority to focus characters when resolving planarity via skewness.

5.1 Objective Function

We now formalize the multi-objective optimization model of F^{2}Stories. We extend classical storyline techniques [11, 17] with group-aware components and modular flexibility. The model includes a term for each of the three core metrics, plus a fairness counterpart. Each term is weighted by a coefficient, allowing users to prioritize or disable components. We now define the optimization terms, starting from:

$$cr = \sum_{t=1}^{\ell-1} \sum_{\substack{i,j \in AC(t,t+1),\\ i<j}} \alpha_{i,j} \cdot y_{t,i,j}$$

> **Explanation:** cr counts the total number of crossings across all time steps, where $y_{t,i,j}$ is 1 if characters i and j cross between consecutive time steps t and $t+1$, and 0 otherwise. The coefficient $\alpha_{i,j}$ is a weight to prioritize the minimization of crossings between characters i and j, based on their group memberships.

$$wg = \sum_{t=1}^{\ell-1} \sum_{i \in AC(t,t+1)} \alpha_i \cdot w_{t,i}$$

> **Explanation:** wg counts the sum of combinatorial vertical movements (wiggles) across all time steps, where $w_{t,i}$ represents the number of positions character i shifts vertically between time t and $t+1$. The coefficient α_i is a weight to prioritize the optimization of the objective towards character i, based on its group membership.

$$sk = \sum_{i \in C} \alpha_i \cdot S_i$$

> **Explanation:** sk counts the total number of characters to be removed in order to obtain a crossing-free storyline. Variable S_i equals 1 if character i has to be removed, 0 otherwise.

The α parameters in the above equations provide a flexible mechanism for weighting characters based on their group membership during the optimization process. In `standardMode` and `fairnessMode`, all α parameters are set to 1. In `focusMode`, instead, these parameters take different values based on the specific optimization metric and the character group membership. In particular:

- The pairwise parameter $\alpha_{i,j} = 1$ if i, j are non-focus characters, $\alpha_{i,j} = \gamma$ for some $\gamma > 1$ if exactly one of i, j is a focus characters, and $\alpha_{i,j} = \gamma^2$ if both i, j are in the focus group.
- The individual parameter $\alpha_i = 1$ if i is a non-focus character and $\alpha_i = \gamma > 1$ if i belongs to the focus group.

The three fairness terms, cr^f, wg^f, and sk^f, measure the differences in visual complexity between the two character groups, normalized by

Table 1: *Summary of definitions for the* F^{2}Stories *MILP model*

Variable defintions

Name	Type	Description
$x_{t,i,j}$	Bin	1 if character i is above character j at time t, 0 otherwise.
$y_{t,i,j}$	Bin	1 if characters i and j cross between time t and $t+1$, 0 otherwise.
cr^f	Float	Fairness variable for crossings in the MILP.
$w_{t,i}$	Int	Wiggle for character i between time t and $t+1$.
wg^f	Float	Fairness variable for wiggles in the MILP.
S_i	Bin	1 to select character i for the skewness set, 0 otherwise.
sk^f	Float	Fairness variable for skewness in the MILP.
b_t	Int	Reference position at time t used for wiggle computation.
$\beta_{t,i,j}$	Bin	Auxiliary variable to avoid overcounting of crossings.
$\delta_{t,i}$	Bin	Auxiliary variable to avoid overcounting of wiggles.

Set defintions

Name	Description
$C = C_R \cup C_B$	C: all characters, C_R: red characters, C_B: blue characters.
$AC(t)$	Active characters at time t.
$AC(t,t+1)$	Active characters between time steps t and $t+1$.
$\Gamma_R(t,i,j)$	Potential crossings involving red characters between time t and $t+1$.
$\Gamma_B(t,i,j)$	Potential crossings involving blue characters between time t and $t+1$.

their respective group sizes. Minimizing these values in the objective function achieves a fair distribution of metrics across character groups.

$$cr^f = \left| \frac{1}{n_R} \sum_{(t,i,j)\in\Gamma_R} y_{t,i,j} - \frac{1}{n_B} \sum_{(t,i,j)\in\Gamma_B} y_{t,i,j} \right|$$

Explanation: cr^f is the difference between the average number of crossings involving red characters and the average number of crossings involving blue characters. By normalizing by group sizes (n_R and n_B), we ensure fair comparison between groups of different cardinalities. When minimized in the objective function (weighted by λ_c^f), the optimization distributes crossings fairly between groups.

$$wg^f = \left| \frac{1}{n_R} \sum_{t=1}^{\ell-1} \sum_{i\in AC_R(t,t+1)} w_{t,i} - \frac{1}{n_B} \sum_{t=1}^{\ell-1} \sum_{i\in AC_B(t,t+1)} w_{t,i} \right|$$

Explanation: wg^f represents the difference between the average wiggle score for the red characters and that for the blue characters. By normalizing by group sizes, this metric ensures a fair comparison regardless of group imbalance. When minimized in the objective function (weighted by λ_w^f), it creates layouts where both groups experience similar wiggle trends in their trajectories.

$$sk^f = \left| \frac{1}{n_R} \sum_{i\in C_R} S_i - \frac{1}{n_B} \sum_{i\in C_B} S_i \right|$$

Explanation: sk^f measures the difference between the proportion of red characters and the proportion of blue characters that must be removed to achieve planarity. This fairness metric ensures that the complexity of resolving crossings is distributed fairly between groups relative to their size. When minimized in the objective function (weighted by λ_s^f), it prevents disproportionate removal of characters from any single group.

When we put together the optimization coefficients with the previous definitions of each term, we obtain the following *objective function*:

$$\min \quad (\lambda_c \cdot cr + \lambda_c^f \cdot cr^f + \lambda_w \cdot wg + \lambda_w^f \cdot wg^f + \lambda_s \cdot sk + \lambda_s^f \cdot sk^f) \quad (1)$$

5.2 Constraints

In our multi-objective optimization model, we define constraints that enforce structural properties of storyline visualizations. These address our optimization metrics while supporting design requirements. Table 1 provides a summary of the used terms and variables. Constraints (2)–(5) follow the ILP model by Dobler et al. [11].

General Storyline Constraints

The core of the MILP formulation for the visualization of storylines is based on the binary variables $x_{t,i,j}$ that encode the relative ordering of characters in the permutation of each time step. Formally, for characters $i, j \in AC(t)$, we define $x_{t,i,j} = 1$ if $i \prec_{\pi_t} j$ (character i appears above character j in the permutation π_t) and $x_{t,i,j} = 0$ otherwise (hence $j \prec_{\pi_t} i$). To maintain consistency in our permutation representation, a first constraint is to enforce that exactly one of these ordering relationships must hold between any pair of characters at each time step:

$$x_{t,i,j} + x_{t,j,i} = 1 \quad \forall t \in T, \forall i, j \in AC(t), i \neq j \qquad (2)$$

Explanation: for any pair of active characters i, j at the same time step t, either i is above j or j is above i, but never both.
Number of constraints generated: $O(\ell \cdot n^2)$

To ensure that character orderings form a valid permutation at each time step, we must enforce *transitivity* in relative positions. At any time t, if character i is above character j and character j is above character k, then character i must be above character k.

$$0 \leq x_{t,i,j} + x_{t,j,k} - x_{t,i,k} \leq 1 \quad \forall t \in T, \ \forall i, j, k \in AC(t) \qquad (3)$$

Explanation: if character i is above j and j is above k, then i must be above k. Also, if either $x_{t,i,j} = 0$ or $x_{t,j,k} = 0$ (but not both), no constraint is added for $x_{t,i,k}$.
Number of constraints generated: $O(\ell \cdot n^3)$

According to [DR2], characters that interact must be drawn as a contiguous group at the time step where the interaction occurs. This ensures that interacting characters appear consecutively in the permutation, with no non-interacting characters between them.

$$x_{t,i,k} - x_{t,j,k} = 0 \qquad (4)$$

$$\forall I \in \mathcal{I} \text{ and } t = time(I), i, j \in char(I), i \neq j, k \in AC(t) \setminus char(I)$$

Explanation: for any interaction I, all characters that participate in I have the same relative position with respect to the same non-interacting characters.
Number of constraints generated: $O(m \cdot n^3)$, though typically much lower in practice as most interactions involve few characters.

This was the last of the three fundamental constraints for storyline visualizations. We will now describe the constraints related to each one of the three optimization metrics and their related fairness definition.

Crossing Detection and Fair Crossing Constraints

Crossings. The classical way to detect crossings in a layered network is to check when two characters swap their relative positions between consecutive layers (see, e.g., [8, 17, 51]). Building on our previously defined ordering variables $x_{t,i,j}$, we introduce the binary variable $y_{t,i,j}$ to capture whether characters i and j cross between time steps t and $t+1$. By minimizing cr in the objective function, $y_{t,i,j} = 1$ if a crossing occurs and $y_{t,i,j} = 0$ otherwise.

$$\begin{aligned} y_{t,i,j} &\geq x_{t,i,j} - x_{t+1,i,j} \\ y_{t,i,j} &\geq x_{t+1,i,j} - x_{t,i,j} \end{aligned} \quad \forall t \in [1, \ell-1], \forall i, j \in AC(t,t+1), i < j \quad (5)$$

Explanation: When characters i and j swap positions between time steps t and $t+1$, either $x_{t,i,j} - x_{t+1,i,j} = 1$ or $x_{t+1,i,j} - x_{t,i,j} = 1$, forcing $y_{t,i,j} \geq 1$. If no swap occurs, $x_{t,i,j} = x_{t+1,i,j}$ and $y_{t,i,j} \geq 0$.
Number of constraints generated: $O(\ell \cdot n^2)$

Fair Crossings. To address [DR4], we introduce constraints that measure and minimize the unfairness in crossing distribution between character groups. These constraints define the fairness variable

cr^f used in our objective function:

$$cr^f \geq \frac{1}{n_R} \sum_{(t,i,j)\in\Gamma_R} y_{t,i,j} - \frac{1}{n_B} \sum_{(t,i,j)\in\Gamma_B} y_{t,i,j}$$
$$cr^f \geq \frac{1}{n_B} \sum_{(t,i,j)\in\Gamma_B} y_{t,i,j} - \frac{1}{n_R} \sum_{(t,i,j)\in\Gamma_R} y_{t,i,j} \quad (6)$$

Explanation: cr^f is the absolute difference between the average number of crossings involving red characters and the average number of crossings involving blue characters.
Number of constraints generated: $O(1)$

We introduce an additional constraint to address a limitation of the standard crossing detection in Eq. (5). When `fairnessMode` is active, it only sets upper bounds on $y_{t,i,j}$, which may not be tight—allowing the solver to set $y_{t,i,j} = 1$ even without an actual crossing, to boost fairness. The new constraint ensures $y_{t,i,j} = 1$ if and only if a true crossing occurs. We introduce an auxiliary binary variable $\beta_{t,i,j}$ to ensure accurate crossing detection.

$$y_{t,i,j} + x_{t,i,j} + x_{t+1,i,j} + 2\beta_{t,i,j} = 2 \quad (7)$$
$$\forall t \in [1, \ell-1], \forall i, j \in AC(t,t+1), i < j$$

Explanation: when characters maintain their relative positions and do not cross, i.e., $x_{t,i,j} = x_{t+1,i,j}$ and their sum is either 0 or 2, the auxiliary binary variable $\beta_{t,i,j}$ must take the value 1 or 0, respectively, to satisfy the constraint. In either case, $y_{t,i,j}$ is forced to be 0.
Number of constraints generated: $O(\ell \cdot n^2)$

Skewness and Fair Skewness Constraints

Skewness. We use binary variables S_i to indicate whether character i must be removed ($S_i = 1$) or not ($S_i = 0$) to achieve planarity.

$$S_i + S_j \geq y_{t,i,j} \quad \forall t \in [1, \ell-1], \forall i, j \in AC(t,t+1), i < j \quad (8)$$

Explanation: This constraint enforces that if characters i and j cross (i.e., $y_{t,i,j} = 1$), then at least one of these characters must be removed to achieve planarity (either $S_i = 1$ or $S_j = 1$ or both). When minimizing the sum of S_i, the optimization process identifies the smallest subset of characters whose removal eliminates all crossings.
Number of constraints generated: $O(\ell \cdot n^2)$

Fair Skewness. To fairly distribute skewness across character groups, we impose the following fairness constraints that measure the normalized absolute difference in character removal proportions.

$$sk^f \geq \frac{1}{n_R} \sum_{i\in C_R} S_i - \frac{1}{n_B} \sum_{i\in C_B} S_i$$
$$sk^f \geq \frac{1}{n_B} \sum_{i\in C_B} S_i - \frac{1}{n_R} \sum_{i\in C_R} S_i \quad (9)$$

Explanation: sk^f is the difference between the average skewness of the red character set and that of the blue character set.
Number of constraints generated: $O(1)$

$$\sum_{i\in C} S_i \leq tr \cdot optSK \quad (10)$$

Explanation: This constraint establishes an upper bound on the total number of characters that can be removed, preventing excessive character removal when optimizing for fairness. The term $optSK$ represents the optimal skewness value obtained from a preprocessing step that minimizes only the global skewness. It is computed using Eq. (8). The *threshold coefficient tr* allows for a reasonable increase in the total skewness to accommodate fairness goals, representing the *price of fairness* we are willing to pay.
Number of constraints generated: $O(1)$

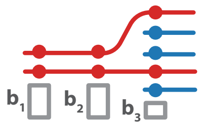


Fig. 4: *Visualization of wiggles and baseline. Each character's vertical position is determined by its relative position to other characters plus a baseline value for that time step. Unavoidable wiggles occur when these positions change between consecutive time steps.*

Wiggles and Fair Wiggles Constraints

Wiggles. We define $w_{t,i}$ as the magnitude of vertical displacement for character i between consecutive time steps t and $t+1$.

$$w_{t,i} \geq \sum_{j\in AC(t)} x_{t,i,j} + b_t - \sum_{j\in AC(t+1)} x_{t+1,i,j} - b_{t+1}$$
$$w_{t,i} \geq \sum_{j\in AC(t+1)} x_{t+1,i,j} + b_{t+1} - \sum_{j\in AC(t)} x_{t,i,j} - b_t \quad (11)$$
$$\forall t \in [1, \ell-1], \forall i \in AC(t,t+1), j \neq i$$

Explanation: These constraints measure the absolute difference in a character's vertical displacement between two consecutive time steps. The term $\sum_{j\neq i} x_{t,i,j} + b_t$ represents the vertical position of character i at time t, counting how many characters are below it plus a baseline value b_t. When this position changes, $w_{t,i}$ captures the magnitude of that change. By minimizing $w_{t,i}$ in the objective function, we create more flat character trajectories.
Number of constraints generated: $O(\ell \cdot n)$

Baseline. Baseline variables b_t define reference points for vertical positioning at each time step, enabling global shifts without adding wiggles and preserving relative positions and order.

$$0 \leq b_t \leq |C| \qquad \forall t \in T \quad (12)$$

Explanation: These constraints bound the baseline values between 0 and the total number of characters. The baseline can be interpreted as the number of *phantom characters* positioned below all actual characters at a given time step, allowing for vertical shifts of the entire character arrangement without affecting relative positions.
Number of constraints generated: $O(\ell)$

Fair Wiggles. To ensure wiggles are fairly distributed between character groups, we introduce fairness constraints that compare the average wiggle magnitude experienced by characters in each group.

$$wg^f \geq \frac{1}{n_R} \sum_{t=1}^{\ell-1} \sum_{i\in AC_R(t,t+1)} w_{t,i} - \frac{1}{n_B} \sum_{t=1}^{\ell-1} \sum_{i\in AC_B(t,t+1)} w_{t,i}$$
$$wg^f \geq \frac{1}{n_B} \sum_{t=1}^{\ell-1} \sum_{i\in AC_B(t,t+1)} w_{t,i} - \frac{1}{n_R} \sum_{t=1}^{\ell-1} \sum_{i\in AC_R(t,t+1)} w_{t,i} \quad (13)$$

Explanation: wg^f is the absolute difference between the average magnitude of wiggles of the red character set and the average magnitude of wiggles of the blue characters set.
Number of constraints generated: $O(1)$

Similar to the crossing constraints, the following upper bound constraints on $w_{t,i}$ (together with Constraints (11)) ensure that $w_{t,i}$ represents the exact vertical displacement rather than just an upper bound. In particular, $\delta_{t,i}$ serve as auxiliary binary variables, and the constant M (set to $M = 2n$) deactivates the constraint that does not apply, ensuring $w_{t,i}$ equals the true magnitude of vertical movement.

$$w_{t,i} \leq \sum_{j\in AC(t)} x_{t,i,j} + b_t - \sum_{j\in AC(t+1)} x_{t+1,i,j} - b_{t+1} + M\delta_{t,i}$$
$$w_{t,i} \leq \sum_{j\in AC(t+1)} x_{t+1,i,j} + b_{t+1} - \sum_{j\in AC(t)} x_{t,i,j} - b_t + M(1-\delta_{t,i}) \quad (14)$$
$$\forall t \in [1, \ell-1], \forall i \in AC(t,t+1), j \neq i$$

Technical Report AC-TR-25-001

Explanation: In Eq. (11), one of the two lower bound constraints is non-negative and the other non-positive, depending on whether character i moves up or down at time t. The auxiliary binary variables $\delta_{t,i}$ ensure that exactly one constraint becomes both a lower and upper bound, while the other—relaxed by a large constant M—is trivially satisfied. This mechanism prevents the optimization from introducing *phantom wiggles*—changes made solely to improve fairness without altering the actual layout.
Number of constraints generated: $O(\ell \cdot n)$

6 Experiments

To test the capabilities of our solution, we perform a computational evaluation, as is usual with many other (M)ILP-based methods [7, 11].

Dataset. Our dataset of 10 storylines combines data from prior work [11], dblp API queries [6], and LLM-generated stories refined by us. It is available in the supplemental material. The number of characters ranges from 13 to 46, and the number of time steps from 39 to 95.

Group Structure. Similar to [13], we assign the characters of each storyline to the minority (red) and majority (blue) groups using three different levels of group balance to demonstrate how `fairnessMode` and `focusMode` behave depending on the relative size of the minority. In `coloring1` we pick a single protagonist[1]; in `coloring2` we select a minority group of significant characters, in the range of $15-25\%$ of the total cast; in `coloring3` we identify a minority group of important characters, accounting for $35-45\%$ of the total cast. These colorings reflect characters' narrative importance; see, for example the three colorings of Fig. 5 for the `JurassicPark` storyline.

Procedure. For each storyline instance, we test twelve different configurations: the cross-product of three different modes (`standardMode`, `fairnessMode` and `focusMode`) with four different combinations of optimization objectives (cross, skew + cross, wiggles + cross and skew + wiggles + cross). Each configuration is also tested on three different group assignments, resulting in 36 experiments per storyline instance. As this is an extensive amount of experiments, we report the bulk of our results in the supplemental material. In Tab. 2, we present the parameter values used in our experimental setup for the different modes and optimization objectives. High values for certain λ-parameters are intentionally chosen to prioritize specific objectives. For example, when simultaneously minimizing fair crossings and crossings, we assign significantly higher weights to the fairness coefficient, ensuring that, for the model, fairness is worth significantly more than the introduction of additional crossings, yet, between two solutions of the same fairness, the one with fewer crossings will always be preferred. Specifically, the fairness coefficient value $(1{,}000)$ was chosen empirically as a good default value for all fairness configurations. This approach, which ensures clear prioritization among competing objectives, is a common practice in multi-objective optimization problems [50]. To address Eq. (10), we set the threshold coefficient $tr = 1.5$ and $optSk$ to the skewness value when crossings are minimized. In this sense, we are willing to pay an increase in global skewness of at most 50% to minimize unfairness.

Setup. All experiments were conducted on systems equipped with Intel Xeon E5-2640 v4 processors (10 cores, 2.40GHz) and 160GB RAM running Ubuntu 18.04.6 LTS. Experiments were executed in single-threaded mode. The code to generate the MILP file is implemented in Python 3.8.16. To solve the MILPs we used Gurobi 11.0.1 [18]. We also set the following parameters: the memory limit is 64GB for all experiments and we set the time limit to 5 hours (18000s). To account for performance variability, all experiments were conducted using a fixed seed. The source code is available on OSF and GitHub and a showcase interactive tool at the following link[2].

[1] In contrast to [19], where the protagonist is an always active character participating in all interactions, we simply mean the main character of the story.
[2] https://tinyurl.com/mwsm5shz

Table 2: *Values of the* F^2`Stories` *parameters for the experimental setup*

	Mode	λ_s^f	λ_s	λ_c^f	λ_c	λ_w^f	λ_w	γ
	`standardMode`	0	0	0	1	0	0	0
cross	`fairnessMode`	0	0	1000	1	0	0	0
	`focusMode`	0	0	0	1	0	0	10
	`standardMode`	0	1	0	1	0	0	0
skew + cross	`fairnessMode`	1000	1	0	1	0	0	0
	`focusMode`	0	1	0	1	0	0	10
	`standardMode`	0	0	0	1	0	1	0
wiggles + cross	`fairnessMode`	0	0	0	1	1000	1	0
	`focusMode`	0	0	0	1	0	1	10
	`standardMode`	0	1	0	1	0	1	0
skew + wiggles + cross	`fairnessMode`	1000	1	1000	1	1000	1	0
	`focusMode`	0	1	0	1	0	1	10

Results. Our evaluation of F^2`Stories` focuses on assessing how effectively the framework balances global readability with group-specific fairness across different storyline datasets. We now report the aggregated results for the whole dataset and, for brevity, detailed results only for a case study in Tab. 3 that we discuss in Sec. 7.

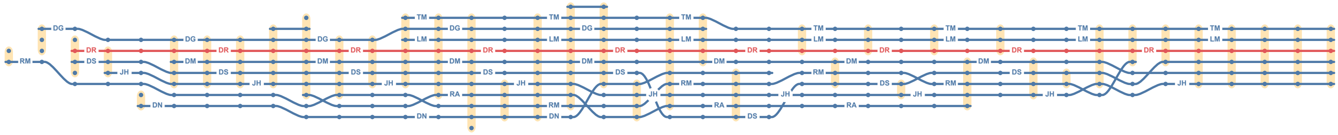
Single-metric `standardMode` optimizations run fastest (avg. 0.71 s), while the most complex configuration (skew + wiggles + cross) averages 817.32 s. `fairnessMode` incurs a substantial computational weight, with some complex multi-objective optimizations hitting our timeout threshold of 5 hours. This performance issue is primarily due to the additional fairness constraints that significantly increase the MILP problem size. `focusMode` falls between these extremes and, sometimes, it is even faster than the `standardMode` approach: for example, for the crossing minimization configuration, it is slower than its counterpart (1.35 s on average) but it is significantly faster for the most complex configuration (126.48 s on average).). This suggests `focusMode` offers a balanced compromise when character prioritization is needed but full fairness optimization is computationally too costly.

Our computational experiments confirm that the three optimization modes produce distinctly different visual results, each with their own advantages (see Fig. 1). The `standardMode` consistently achieves the lowest absolute crossing counts across all datasets, as expected for a global optimization approach. However, our `fairnessMode` successfully balances crossing distributions between character groups, with crossing unfairness (cr^f) values approaching zero in most cases. This comes at a price: the total number of crossings increases on average by 53% compared to `standardMode`, while the fairness improves, on average, by 83%. Examining the multi-objective optimizations of `fairnessMode`, it reveals important trade-offs between metrics. When optimizing for skewness fairness (sk^f), we observe that achieving perfectly fair skewness is structurally more constrained than other metrics. We observe an increase in skewness of 1.8% on average, while the related fairness improves by 44%. Wiggle fairness (wg^f) exhibits the greatest variability across optimization metrics, suggesting that character trajectory smoothness is most sensitive to fairness: even though the average global number of wiggles increases by 28%, the fairness metric exhibits an average improvement of 93.3% compared to `standardMode`. The same conclusions can be drawn from the complete model in which: crossings go up by 46% on average while cr^f improves by 91.5%; wiggles increase by 50% on average, but their fairness improves by 95%; skewness remains more or less stationary, (8% increase) with an improvement of the skewness fairness of 44%, on average. `focusMode` exhibits particularly interesting behavior in relation to group size. When the focus group is very small (as in `coloring1` with a single red character), we observe near-perfect optimization for focus characters, reducing the number of crossings involving red characters to 0 and the number of red wiggles to 0. As the focus group size increases (in `coloring2` and `coloring3`), the advantage diminishes proportionally, as expected, suggesting a practical limit to the approach.

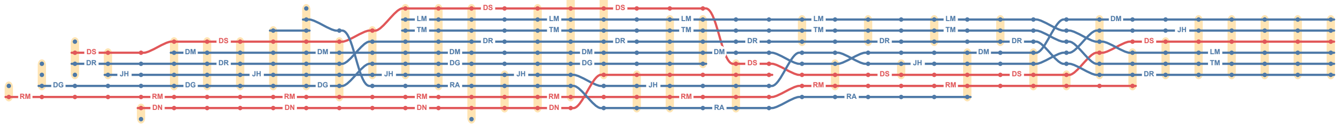
7 Case Study

Here we present and analyze results from our case study using the movie *Jurassic Park* (1993) as our narrative storyline. Being one of the

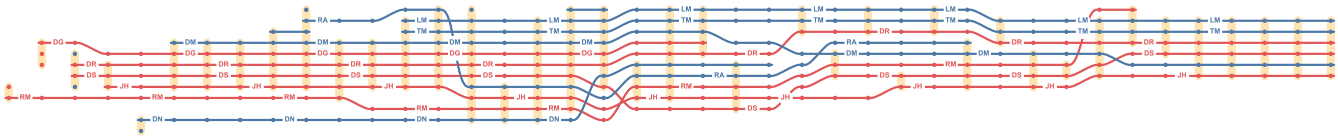
Technical Report AC-TR-25-001



Jurassic Park, coloring 1, focusMode + skew + wiggles + cross.



Jurassic Park, coloring 2, focusMode + skew + wiggles + cross.



Jurassic Park, coloring 3, focusMode + skew + wiggles + cross.

Fig. 5: *Examples of the three coloring styles on the Jurassic Park dataset. Optimization metrics: skew, wiggles and cross in* focusMode .

pioneering examples of handwritten storyline visualizations presented in XKCD's narrative charts [29], the plot of this movie serves as an ideal example to demonstrate the concepts presented throughout this paper. The specific data we analyzed are derived from more recent work [11].

The JurassicPark storyline comprises 20 characters across 41 time steps. We tested three different character groupings: coloring1 designates Dr. Alan Grant (DR) as the sole protagonist; coloring2 selects three co-protagonists (DS, RM, and DN); and coloring3 places 40% of the characters in the red group, representing the team of scientists central to the movie. Some characters appear briefly, such as a worker attacked by a velociraptor in the opening scene, and so they are represented as individual dots in a single time step. The detailed results are presented in Tab. 3, with rows corresponding to optimization metrics and columns showing four optimization configurations, each with results for the three optimization modes. Computation time varies predictably with optimization complexity: single-metric optimization (like crossing minimization in standardMode) executes faster, while multi-objective optimization (as, e.g., skew + wiggles + cross in fairnessMode) requires more processing time. This pattern is consistent across all colorings.

For the JurassicPark storyline, our standard optimization achieved minimum crossings $cr = 18$, minimum skewness $sk = 3$ when optimizing for crossings, and minimum wiggles $wg = 36$ when simultaneously minimizing crossings. When optimizing all three metrics together, crossings and wiggles remain unchanged ($cr = 18, wg = 36$), while skewness slightly increases ($sk = 4$), since it is layout dependent. These results in standardMode remain consistent across all colorings. The fairnessMode columns show results when minimizing unfairness for each metric. Both cr^f and wg^f reach 0 in their respective configurations. Meanwhile, sk^f does not fall below 0.08 given its bound (Eq. (10)). The focusMode columns present results from optimization focused on specific characters. The key objective of focusMode is not to achieve overall fairness, but to minimize the target metric for the characters in the focus group. Results vary significantly by coloring: in coloring1, the total crossings increase from 18 to 19 compared to standardMode , but the crossings of the single red character drop from 2 to 0; in coloring2, the total crossings rise from 18 to 26, with only 1 instead of 2 red-red crossing and 12 instead of 16 blue-red crossings; and in coloring3, results closely resemble standardMode results because the nearly equal distribution of characters (40% red, 60% blue) effectively eliminates the focus group emphasis during optimization. The impact of focusMode , accross all metrics, is most pronounced if the focus group represents, as intended, a minority of the characters.

8 Discussion

The primary contribution of this paper is the introduction of a novel storyline model that integrates fairness into layout optimization, providing significant value by ensuring balanced distributions of crossings, skewness, and wiggles between minority and majority groups in the storyline. By minimizing the unfairness, fairnessMode produces a storyline visualization in which both groups of characters have an equitable distribution of crossings, skewness and/or wiggles. For example, in the second layout of Fig. 1, we can see that by a 30% increase in the number of crossings with respect to the minimum, a perfectly fair distribution of crossings can be obtained. focusMode enhances visual clarity for key characters, ensuring less cluttered trajectories while maintaining readability for the other characters. As shown in Fig. 1, the third layout removes crossings for the focus group, unlike standardMode (first layout). With emphasis on a single optimized protagonist (Fig. 5 top), this character appears as a perfectly horizontal and uncrossed line, showcasing the effectiveness of targeted optimization.

One of the key aspects to consider is the **price of fairness**: the trade-off between achieving optimal overall readability and producing a fair visualization, where fairness may lead to a higher number of crossings, wiggles, or skewness. In several cases, we observed that when a crossing was unavoidable in the disadvantaged group, the solver would deliberately introduce a crossing in the advantaged group to improve fairness, even if the result may appear counterintuitive from a global readability perspective. To investigate this phenomenon and address the question, "What is a good balance between global optimality and fair distribution of crossings?", we conducted a series of experiments (Fig. 6). In each experiment, we upper bounded the number of allowed crossings, starting from the global minimum (obtained from standardMode) and increasing the upper bound by 1 in each successive run. This behavior was achieved by introducing a new simple constraint that ensures $cr \leq CC$, with CC being the number of allowed crossings. Regarding the values for the λ-parameters, only λ_c^f was enabled, with its value set to 1. For example, in our case study, given an optimal crossing count of 18, we ran multiple fairness-optimizing computations for each of the three colorings, with $CC = 18, 19, 20, \ldots, 50$. This allowed us to study how fairness evolves as the crossing allowance increases. From the results in Fig. 6, we observe that most unfairness plots exhibit an inflection point—a noticeable shift in the rate of change. These points represent meaningful trade-offs between fairness and overall readability. We recommend using such inflection points as thresholds for balancing the two competing objectives, although identifying them does require running the experiment multiple times. As shown in Tab. 3, some instances (marked in light gray) exceed time or memory limits. The MILP solver Gurobi outputs the best-found solution even if it hits the allocated computation time, though optimality is not guaranteed.

Table 3: *Result values for Jurassic Park, across 3 different ways to assign characters to group (i.e. coloring). Each column represents one of 12 experiments, split in 4 different optimization objectives and 3 different modes (Standard, Fair and Focus), with results reported for 13 metrics. Values in* ***bold*** *represent the best value for a metric across all experiments, and colored cell backgrounds represent the best results for unfairness, with the most vivid color representing especially good results. In the timing column, "oot" and "oom" indicate respectively "out of time" and "out of memory". We report those columns as grayed out. Gurobi returns partial results for oot and oom solutions even if the computation can not completely finish, and, although optimality cannot be guaranteed on those, they still report valid solutions with the best result obtained while the computation was working.*

Jurassic Park

coloring 1

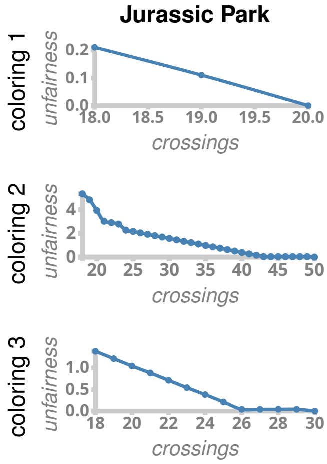
		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.4	5.5	0.3	0.8	0.8	0.4	23.1	138.4	16.7	30.4	oot	17.0
crossings	Blue-Blue	**16**	18	19	17	17	19	**16**	**16**	19	**16**	18	19
	Red-Red	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	0	**0**
	Blue-Red	2	2	**0**	1	1	**0**	2	2	**0**	2	2	**0**
	Total	**18**	20	19	**18**	**18**	19	**18**	**18**	19	**18**	20	19
	Unfairness	0.21	**0.00**	2.00	0.84	0.84	2.00	0.21	0.21	2.00	0.21	0.00	2.00
skewness	Blue	-	-	-	**3**	**3**	**3**	-	-	-	4	3	**3**
	Red	-	-	-	**0**	**0**	**0**	-	-	-	**0**	0	**0**
	Total	-	-	-	**3**	**3**	**3**	-	-	-	4	3	**3**
	Unfairness	-	-	-	**0.16**	**0.16**	**0.16**	-	-	-	0.21	0.16	**0.16**
wiggles	Blue	-	-	-	-	-	-	**36.0**	38.0	45.0	**36.0**	57.0	45.0
	Red	-	-	-	-	-	-	**0.0**	2.0	**0.0**	**0.0**	3.0	**0.0**
	Total	-	-	-	-	-	-	**36**	40	45	**36**	60	45
	Unfairness	-	-	-	-	-	-	1.89	**0.00**	2.37	1.89	0.00	2.37

coloring 2

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.3	oom	0.3	0.7	5.9	0.3	23.3	oot	10.1	30.7	oot	13.8
crossings	Blue-Blue	**0**	40	13	**0**	**0**	13	1	6	13	1	40	14
	Red-Red	2	5	**1**	3	3	**1**	2	3	**1**	2	5	**1**
	Blue-Red	16	**5**	12	15	15	12	15	14	12	15	**5**	12
	Total	**18**	50	26	**18**	**18**	26	**18**	23	26	**18**	50	27
	Unfairness	5.73	0.00	2.43	6.12	6.12	2.43	5.33	5.14	2.43	5.33	0.00	2.31
skewness	Blue	-	-	-	**0**	2	6	-	-	-	1	2	6
	Red	-	-	-	3	2	**1**	-	-	-	3	2	**1**
	Total	-	-	-	**3**	4	7	-	-	-	4	4	7
	Unfairness	-	-	-	1.00	0.55	**0.02**	-	-	-	0.94	0.55	**0.02**
wiggles	Blue	-	-	-	-	-	-	**10.0**	68.0	36.0	**10.0**	85.0	35.0
	Red	-	-	-	-	-	-	26.0	**12.0**	17.0	26.0	15.0	17.0
	Total	-	-	-	-	-	-	**36**	80	53	**36**	100	52
	Unfairness	-	-	-	-	-	-	8.08	0.00	3.55	8.08	0.00	3.61

coloring 3

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.4	oot	0.5	0.7	1.1	1.5	23.2	oot	12.2	30.6	6162	11.1
crossings	Blue-Blue	**1**	11	2	**1**	2	2	**1**	**1**	5	**1**	11	5
	Red-Red	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**
	Blue-Red	12	14	**11**	12	**11**	**11**	12	15	12	12	14	12
	Total	**18**	30	**18**	**18**	**18**	**18**	**18**	21	22	**18**	30	22
	Unfairness	1.58	0.00	1.38	1.58	1.38	1.38	1.58	1.71	0.92	1.58	**0.00**	0.92
skewness	Blue	-	-	-	**1**	2	2	-	-	-	**1**	2	3
	Red	-	-	-	**2**	**2**	**2**	-	-	-	3	**2**	**2**
	Total	-	-	-	**3**	4	4	-	-	-	4	4	5
	Unfairness	-	-	-	0.17	0.08	0.08	-	-	-	0.29	0.08	**0.00**
wiggles	Blue	-	-	-	-	-	-	**10.0**	30.0	36.0	**10.0**	33.0	36.0
	Red	-	-	-	-	-	-	26.0	**20.0**	21.0	26.0	22.0	21.0
	Total	-	-	-	-	-	-	**36**	50	57	**36**	55	57
	Unfairness	-	-	-	-	-	-	2.42	0.00	0.38	2.42	**0.00**	0.38



There is a fine balance that must be considered when optimizing crossing fairness: in order to make the crossings fair between the two groups, the resulting number of crossings will be a little higher than the total possible minimum crossings in the visualization. This is the **price of fairness**. The charts on the side represent the changing value of fairness when we allow for a certain maximum number of total crossings in the network, computed on the case study. The maximum number of crossings allowed corresponds to the number of crossings computed when cr^f is minimized, for each coloring. Each dot represents an individual experiment, and we keep track, on the y-axis, of the trend for the unfairness.

Fig. 6: *The price of fairness, illustrated.*

Limitations Our approach to fairness in storyline visualizations, while effective in balancing readability across character groups, has some limitations. The wiggle optimization in F^2`Stories`, for instance, cannot add spaces between lines—only at the top or bottom. Allowing this could reduce wiggles further. The computational complexity of `fairnessMode` is another key limitation. As seen in several "out of time" and "out of memory" cases, achieving optimal fairness can be computationally prohibitive for large instances or complex multi-objective problems. While our approach still yields valid solutions when resource limits are hit, they come without formal optimality guarantees.

Future Work. Several research directions could further advance fairness in storyline visualizations. Extending our approach beyond two character groups may enhance narrative analysis, requiring generalized fairness metrics and efficient algorithms to handle greater group complexity. Supporting ubiquitous characters [10] would improve applicability to stories with omnipresent entities, requiring specialized handling to prevent unfair influence. Conducting comprehensive user studies would help validate whether mathematically fair visualizations translate to improved readability and comprehension, potentially revealing disparities between mathematical definitions of fairness and user perceptions. Exploring additional dimensions of fairness beyond traditional readability metrics could yield valuable insights, such as the perceptual influence of characters based on narrative importance or the semantic significance of specific interactions. These directions would strengthen both theoretical foundations and practical applications of fair storyline visualizations, expanding the impact of approaches like F^2`Stories` in supporting balanced visual representations of complex narratives. In particular, we find it promising to apply F^2`Stories` to gameplay analysis (e.g., [27,49]) by focusing on a subset of players and their interactions.

9 Conclusion

In this paper, we introduced F^2`Stories`, a novel framework for creating storyline visualizations that balance readability across different character groups. Through a modular Mixed Integer Linear Programming approach, F^2`Stories` provides three complementary optimization modes: `standardMode` for traditional global optimization, `fairnessMode` for balanced distribution of the visual complexity across character groups, and `focusMode` for prioritizing specific important characters. Our experimental results across multiple storyline instances demonstrate that fair layouts can be achieved with a manageable increase in global metrics such as crossings, wiggles, and skewness, quantifying what we call the *price of fairness* in storyline visualization. Similarly, `focusMode` creates effective visual priorities that highlight significant character curves without excessively compromising overall readability, offering an important tool when storytelling clarity demands emphasis on certain characters.

Supplementary Materials

We include as supplementary materials:

- The appendix to this paper, which includes additional results and explanations.
- The code that we used to create and test the MILP formulations, which also includes code used to generate the figures in the paper and render the storyline visualizations.
- Results and statistics (including models, runtimes and solutions) for 10 stories, using all possible combinations of `fairnessMode`, `focusMode`, `standardMode`, each in 3 different colorings and using 4 different optimization objectives (cross, skew + cross, wiggles + cross, skew + wiggles + cross), for a total of 36 experiments per individual story.
- We created a GitHub repository available at the following link `https://github.com/tommaso-piselli/f2stories`. A webpage offering quick access to statistics and visualizations of all of these results, available at `https://tommaso-piselli.github.io/f2stories/src/showcase.html`.

The above is offered through OSF at `osf.io/e2qvy`, released under CC-BY 4.0 license.

Acknowledgments

The work of Sara Di Bartolomeo was funded by the Austrian Science Fund (FWF) [10.55776/ESP513]. For open access purposes, the authors have applied a CC BY public copyright license to any author accepted manuscript version arising from this submission. Work of Giuseppe Liotta supported in part by (i) MUR PRIN Proj. 2022TS4Y3N - "EXPAND: scalable algorithms for EXPloratory Analyses of heterogeneous and dynamic Networked Data"; (ii) MUR PON Proj. ARS01_00540 - "RASTA: Augmented Reality and Automated Story-Telling for Cultural Heritage and Turistic Itineraries". Work of Fabrizio Montecchiani supported in part by: (i) MUR PRIN Proj. 2022ME9Z78 - "NextGRAAL: Next-generation algorithms for constrained GRAph visuALization"; (ii) University of Perugia, Fondo di Ricerca di Ateneo 2022, under Project MiRA "Mixed Reality and AI Methodologies for Immersive Robotics". Work of Martin Nöllenburg supported in part by the Vienna Science and Technology Fund (WWTF) [10.47379/ICT19035].

References

[1] S. Barocas and A. D. Selbst. Big data's disparate impact. *Calif. L. Rev.*, 104:671, 2016. doi: 10.2139/ssrn.2477899 1

[2] T. Baumgartl, M. Petzold, M. Wunderlich, M. Höhn, D. Archambault, M. Lieser, A. Dalpke, S. Scheithauer, M. Marschollek, V. Eichel, N. T. Mutters, H. Consortium, and T. von Landesberger. In search of patient zero: Visual analytics of pathogen transmission pathways in hospitals. *IEEE Trans. Vis. Comput. Graph.*, 27(2):711–721, 2021. doi: 10.1109/TVCG.2020.3030437 2

[3] A. Chatzimparmpas, R. M. Martins, I. Jusufi, K. Kucher, F. Rossi, and A. Kerren. The state of the art in enhancing trust in machine learning models with the use of visualizations. *Comput. Graph. Forum*, 39(3):713–756, 2020. doi: 10.1111/cgf.14034 3

[4] X. Chen, B. Fain, L. Lyu, and K. Munagala. Proportionally fair clustering. In K. Chaudhuri and R. Salakhutdinov, eds., *Proc. 36th International Conference on Machine Learning, (ICML'19)*, vol. 97 of *Proceedings of Machine Learning Research*, pp. 1032–1041. PMLR, 2019. `http://proceedings.mlr.press/v97/chen19d.html`. 2

[5] D. Danks and A. J. London. Algorithmic bias in autonomous systems. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI'17)*, pp. 4691–4697, 2017. doi: 10.24963/ijcai.2017/654 2

[6] dblp sparql. `https://sparql.dblp.org/`, Accessed Mar 24, 2025. 7

[7] S. Di Bartolomeo, T. Crnovrsanin, D. Saffo, E. Puerta, C. Wilson, and C. Dunne. Evaluating graph layout algorithms: A systematic review of methods and best practices. *Comput. Graph. Forum*, 43(6):e15073, 2024. doi: 10.1111/cgf.15073 7

[8] S. Di Bartolomeo, M. Riedewald, W. Gatterbauer, and C. Dunne. Stratisfimal layout: A modular optimization model for laying out layered node-link network visualizations. *IEEE Trans. Vis. Comput. Graph.*, 28(1):324–334, 2022. doi: 10.1109/TVCG.2021.3114756 5

[9] S. Di Bartolomeo, Y. Zhang, F. Sheng, and C. Dunne. Sequence braiding: Visual overviews of temporal event sequences and attributes. *IEEE Trans. Vis. Comput. Graph.*, 27(2):1353–1363, 2021. doi: 10.1109/TVCG.2020.3030442 2

[10] E. Di Giacomo, W. Didimo, G. Liotta, F. Montecchiani, and A. Tappini. Storyline visualizations with ubiquitous actors. In *Proc. Graph Drawing and Network Visualization (GD'20)*, vol. 12590 of *LNCS*, p. 324–332. Springer, 2020. doi: 10.1007/978-3-030-68766-3_25 2, 3, 9

[11] A. Dobler, M. Jünger, P. J. Jünger, J. Meffert, P. Mutzel, and M. Nöllenburg. Revisiting ILP Models for Exact Crossing Minimization in Storyline Drawings. In S. Felsner and K. Klein, eds., *Proc. Graph Drawing and Network Visualization (GD'24)*, vol. 320 of *LIPIcs*, pp. 31:1–31:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi: 10.4230/LIPIcs.GD.2024.31 2, 3, 4, 5, 7, 8

[12] A. Dobler, M. Nöllenburg, D. Stojanovic, A. Villedieu, and J. Wulms. Crossing minimization in time interval storylines. *39th European Workshop on Computational Geometry: EuroCG2023: Book of Abstracts*, 2023. doi: 10.48550/arXiv.2302.14213 2, 3

[13] P. Eades, S. Hong, G. Liotta, F. Montecchiani, M. Nöllenburg, T. Piselli, and S. Wismath. Introducing fairness in network visualization. *Information Sciences*, 691:121642, 2025. doi: 10.1016/j.ins.2024.121642 2, 3, 7

[14] T. Fröschl and M. Nöllenburg. Minimizing wiggles in storyline visualizations. In F. Frati and K.-L. Ma, eds., *Proc. Graph Drawing and Network Visualization (GD'17)*, vol. 10692 of *LNCS*, pp. 585–587. Springer, 2017. doi: 10.1007/978-3-319-73915-1 2, 3

[15] E. Gansner, E. Koutsofios, S. North, and K.-P. Vo. A technique for drawing directed graphs. *IEEE Trans. Software Eng.*, 19(3):214–230, 1993. doi: 10.1109/32.221135 2

[16] M. Ghadiri, S. Samadi, and S. Vempala. Socially fair k-means clustering. In M. C. Elish, W. Isaac, and R. S. Zemel, eds., *Proc. ACM Conference on Fairness, Accountability, and Transparency (FAccT'21)*, pp. 438–448. ACM, 2021. doi: 10.1145/3442188.3445906 2

[17] M. Gronemann, M. Jünger, F. Liers, and F. Mambelli. Crossing minimization in storyline visualization. In Y. Hu and M. Nöllenburg, eds., *Proc. Graph Drawing and Network Visualization (GD'16)*, vol. 9801 of *LNCS*, pp. 367–381. Springer, 2016. doi: 10.1007/978-3-319-50106-2_29 2, 3, 4, 5

[18] Gurobi solver. `https://www.gurobi.com/`, Accessed Feb 20, 2025. 7

[19] T. Hegemann and A. Wolff. Storylines with a Protagonist. In S. Felsner and K. Klein, eds., *Proc. Graph Drawing and Network Visualization (GD'24)*, vol. 320 of *LIPIcs*, pp. 26:1–26:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi: 10.4230/LIPIcs.GD.2024.26 2, 3, 7

[20] S.-H. Hong, G. Liotta, F. Montecchiani, M. Nöllenburg, and T. Piselli. Introducing fairness in graph visualization via gradient descent. In D. Archambault, I. Nabney, and J. Peltonen, eds., *Machine Learning Methods in Visualisation for Big Data (MLVis@EuroVis'24)*. The Eurographics Association, 2024. doi: 10.2312/mlvis.20241124 3

[21] P. C. Kainen. A lower bound for crossing numbers of graphs with applications to K_n, $K_{p,q}$, and $Q(d)$. *J. Comb. Theory, Ser. B*, 12(3):287–298, 1972. doi: 10.1016/0095-8956(72)90042-1 3

[22] M. M. Kamani, F. Haddadpour, R. Forsati, and M. Mahdavi. Efficient fair principal component analysis. *Mach. Learn.*, 111(10):3671–3702, 2022. doi: 10.1007/S10994-021-06100-9 3

[23] L. Kellerhals and J. Peters. Proportional fairness in clustering: A social choice perspective. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, eds., *Proc. Neural Information Processing Systems (NeurIPS'24)*, 2024. `https://nips.cc/virtual/2024/poster/95639`. 2

[24] I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, and D. Strash. On minimizing crossings in storyline visualizations. In E. Di Giacomo and A. Lubiw, eds., *Proc. Graph Drawing and Network Visualization (GD'15)*, vol. 9411 of *LNCS*, pp. 192–198. Springer, 2015. doi: 10.1007/978-3-319-27261-0_16 2

[25] M. Lackner and P. Skowron. *Multi-winner voting with approval preferences*. Springer Nature, 2023. doi: 10.1007/978-3-031-09016-5 2

[26] J. Lee, H. Cho, S. Yun, and C. Yun. Fair streaming principal component analysis: Statistical and algorithmic viewpoint. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, eds., *Proc. Neural Information Processing Systems (NeurIPS'23)*, 2023. `https://neurips.cc/virtual/2023/poster/71553`. 3

[27] J. Lin, C. Yang, and C. Kao. Visualizing NBA information via storylines. *Comput. Graph.*, 127:104169, 2025. doi: 10.1016/J.CAG.2025.104169 9

[28] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A

survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6):115:1–115:35, 2022. doi: 10.1145/3457607 1

[29] R. Munroe. Movie narrative charts, 2009. https://xkcd.com/657/, Accessed: 2025-02-20. 2, 8

[30] P. Mutzel. An alternative method to crossing minimization on hierarchical graphs. *SIAM J. Optim.*, 11(4):1065–1080, 2001. doi: 10.1137/S1052623498334013 3

[31] M. Nöllenburg and A. Wolff. A mixed-integer program for drawing high-quality metro maps. In P. Healy and N. S. Nikolov, eds., *Proc. Graph Drawing (GD'05)*, vol. 3843 of *LNCS*, pp. 321–333. Springer, 2006. doi: 10.1007/11618058_29 2

[32] M. Nöllenburg and A. Wolff. Drawing and labeling high-quality metro maps by mixed-integer programming. *IEEE Trans. Vis. Comput. Graph.*, 17(5):626–641, 2011. doi: 10.1109/TVCG.2010.81 2

[33] M. Ogawa and K.-L. Ma. Software evolution storylines. In A. C. Telea, C. Görg, and S. P. Reiss, eds., *Proc. Software Visualization (SOFTVIS'10)*, p. 35–42. ACM, 2010. doi: 10.1145/1879211.1879219 2

[34] C. O'Neil. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown, 2017. 1

[35] J. Pach and G. Tóth. Which crossing number is it anyway? *J. Comb. Theory, Ser. B*, 80(2):225–246, 2000. doi: 10.1006/jctb.2000.1978 3

[36] K. Padia, K. H. Bandara, and C. G. Healey. Yarn: Generating storyline visualizations using HTN planning. In *Proc. Graphics Interface Conference (GI'18)*, p. 26–33. ACM, 2018. doi: 10.20380/GI2018.05 2

[37] K. Padia, K. H. Bandara, and C. G. Healey. A system for generating storyline visualizations using hierarchical task network planning. *Comput. Graph.*, 78:64–75, 2019. doi: 10.1016/j.cag.2018.11.004 2

[38] J. Peltonen, W. Xu, T. Nummenmaa, and J. Nummenmaa. Fair neighbor embedding. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, eds., *Proc. International Conference on Machine Learning (ICML'23)*, vol. 202 of *Proceedings of Machine Learning Research*, pp. 27564–27584. PMLR, 2023. https://proceedings.mlr.press/v202/peltonen23a.html. 2, 3

[39] H. C. Purchase. Metrics for graph drawing aesthetics. *J. Vis. Lang. Comput.*, 13(5):501–516, 2002. doi: 10.1006/jvlc.2002.0232 2, 3

[40] S. Samadi, U. Tantipongpipat, J. H. Morgenstern, M. Singh, and S. Vempala. The price of fair PCA: One extra dimension. *Proc. Neural Information Processing Systems (NeurIPS'18)*, 31, 2018. https://proceedings.neurips.cc/paper/2018/hash/cc4af25fa9d2d5c953496579b75f6f6c-Abstract.html. 2, 3

[41] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Trans. Syst. Man Cybern.*, 11(2):109–125, 1981. doi: 10.1109/TSMC.1981.4308636 2

[42] H. Suresh and J. Guttag. A framework for understanding sources of harm throughout the machine learning life cycle. In *Proc. Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO'21)*, pp. 17:1–17:9. ACM, 2021. doi: 10.1145/3465416.3483305 1, 2

[43] Y. Tanahashi, C.-H. Hsueh, and K.-L. Ma. An efficient framework for generating storyline visualizations from streaming data. *IEEE Trans. Vis. Comput. Graph.*, 21(6):730–742, 2015. doi: 10.1109/TVCG.2015.2392771 2, 3

[44] Y. Tanahashi and K.-L. Ma. Design considerations for optimizing storyline visualizations. *IEEE Trans. Vis. Comput. Graph.*, 18(12):2679–2688, 2012. doi: 10.1109/TVCG.2012.212 2, 3

[45] T. Tang, S. Rubab, J. Lai, W. Cui, L. Yu, and Y. Wu. iStoryline: Effective convergence to hand-drawn storylines. *IEEE Trans. Vis. Comput. Graph.*, 25(1):769–778, 2019. doi: 10.1109/TVCG.2018.2864899 2

[46] U. Tantipongpipat, S. Samadi, M. Singh, J. Morgenstern, and S. S. Vempala. Multi-criteria dimensionality reduction with applications to fairness. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, eds., *Proc. Neural Information Processing Systems (NeurIPS'19)*, pp. 15135–15145, 2019. https://proceedings.neurips.cc/paper/2019/hash/2201611d7a08ffda97e3e8c6b667a1bc-Abstract.html. 3

[47] T. van Dijk, M. Fink, N. Fischer, F. Lipp, P. Markfelder, A. Ravsky, S. Suri, and A. Wolff. Block crossings in storyline visualizations. *J. Graph Algorithms Appl.*, 21(5):873–913, 2017. doi: 10.7155/jgaa.00443 2

[48] T. C. van Dijk, F. Lipp, P. Markfelder, and A. Wolff. Computing storyline visualizations with few block crossings. In F. Frati and K. Ma, eds., *Proc. Graph Drawing and Network Visualization (GD'17)*, vol. 10692 of *LNCS*, pp. 365–378. Springer, 2017. doi: 10.1007/978-3-319-73915-1_29 2

[49] G. Wallner, L. Wang, and C. Dormann. Visualizing the spatio-temporal evolution of gameplay using storyline visualization: A study with league of legends. *Proc. ACM Hum. Comput. Interact.*, 7(CHI PLAY):1002–1024, 2023. doi: 10.1145/3611058 9

[50] C. Wilson, E. Puerta, T. Crnovrsanin, S. Di Bartolomeo, and C. Dunne. Evaluating and extending speedup techniques for optimal crossing minimization in layered graph drawings. *IEEE Trans. Vis. Comput. Graph.*, 31(1):1061–1071, 2025. doi: 10.1109/TVCG.2024.3456349 3, 7

[51] D. C. Zarate, P. L. Bodic, T. Dwyer, G. Gange, and P. Stuckey. Optimal Sankey diagrams via integer programming. In *IEEE Pacific Visualization Symposium (PacificVis'18)*, pp. 135–139, 2018. doi: 10.1109/PacificVis.2018.00025 5

10 Rendering Procedure

After the results of the MILP movels have been computed by Gurobi, we render the result in the browser using d3.js to display the results visually. In cases where the optimization objective does not include wiggles, we run a brief postprocessing step to smoothen the lines, described below:

Postprocessing for smoothness We implemented a simple, heuristic-based postprocessing step to improve the appearance of the final renders wherever wiggliness was not part of the optimization objectives. The effects of this are visible, for instance, in Figure 1, where the optimization objective is exclusively crossing, and, in postprocessing, the lines are spread out vertically to make them more straight, and easier to read.

It is important to note that:

- We do not consider the postprocessing step as part of out contributions.
- We do not intend to encourage comparison between heuristic results and MILP results.
- The postprocessing step is only applied in instances where reducing wiggliness is not part of the optimization objectives.
- Any numerical result reported in the paper does not include results obtained after the postprocessing step — everything is computed before.

The following is a simple snippet of pseudocode explaining the process, for transparency purposes:

Algorithm 1 Wiggle Minimization Sweep

```
for all timesteps t in total set of timesteps do
    n ← uppermost node at timestep t
    while wiggliness improves do
        Push n (and all nodes below it) downward
    end while
end for
Repeat the sweep until no further improvement
```

The main idea behind this procedure is to iteratively reduce the visual "wiggliness" of a storyline layout by adjusting node positions in a structured way. At each timestep, the algorithm identifies the topmost node and attempts to push it—and consequently, all nodes below it—downward. This shift is applied only if it results in a smoother vertical trajectory for the characters across time, thereby improving the overall layout quality. The process is repeated in sweeping iterations over all timesteps until no further improvements in wiggliness are observed, allowing the layout to settle into a more visually stable configuration.

After this step, we continue with the rendering procedure, described below.

Reading results A simple graph structure is instantiated in javascript. The results of the MILP are read timestep by timestep: for every timestep t, starting from the earliest ones in the whole story, we collect all characters present at timestep t. Then, we sort them using the $x_{i,j,t}$ variables — which indicated relative positions of characters i and j at timestep t, and draw each one of them following this given order.

In case wiggliness was an optimization objective, the result will also contain an additional variable, b_t, which indicates a shift that needs to be done from the bottom of the visualization in order for the lines to appear smooth. In this case, a buffer, corresponding to the value of b_t at each timestep t, is added to the top of the visualization.

In case the b_t values are not present, we use the preprocessing step described above to determine the buffer.

Final rendering The steps before this allowed us to define coordinates for every character at every timestep. Once this is obtained, the last thing to do is to render the image in svg: a circle is added to highlight the position of the character at every timestep, and then a line is interpolated through all of these circles. Scenes are also drawn as yellow rectangles below the circles and lines: coordinates for each scene are determined based on the characters they contain at the timestep where they appear. It is sufficient for us to determine the topmost and bottommost character in each scene to determine the coordinates of such rectangles.

The final visualizations obtained through this method can be seen for instance in Table 4 or Figure 1, or, in more detail, on `https://anonymous.4open.science/w/fairstories-F2A0/src/showcase.html`.

Table 4: *Jurassic Park coloring 3, all modes and experiments*

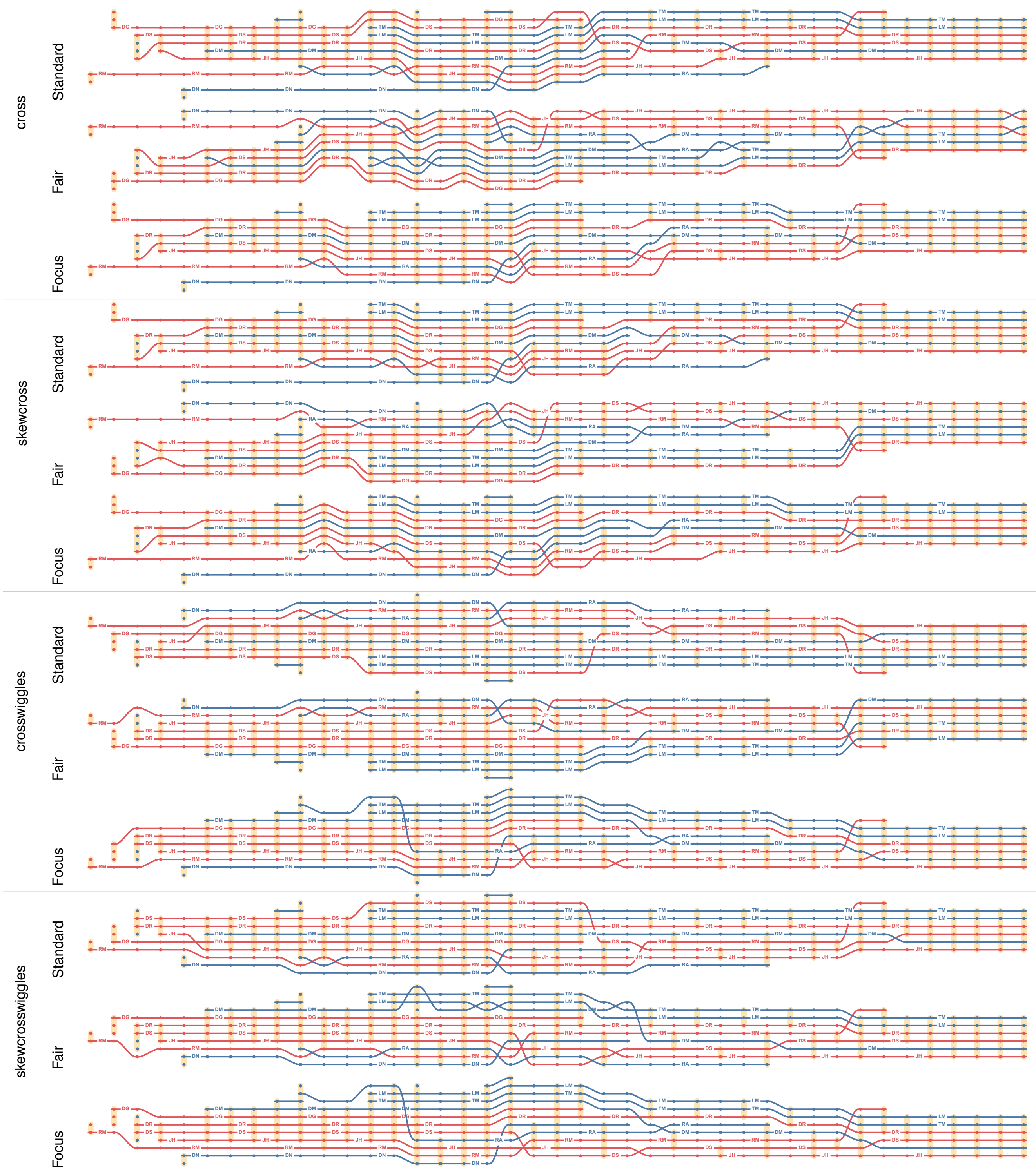


Table 5: *Result values for dblp.*

dblp — coloring 1

		cross: Standard	cross: Fair	cross: Focus	cross + skew: Standard	cross + skew: Fair	cross + skew: Focus	cross + wiggles: Standard	cross + wiggles: Fair	cross + wiggles: Focus	cross + skew + wiggles: Standard	cross + skew + wiggles: Fair	cross + skew + wiggles: Focus
	Time (s):	0.3	4.7	1.8	0.3	10.4	2.9	212	165	90.6	200	1237	113
crossings	Blue-Blue	10	24	19	7	16	19	9	9	20	7	24	21
crossings	Red-Red	0	0	0	0	0	0	0	0	0	0	0	0
crossings	Blue-Red	6	3	3	9	7	3	7	7	3	9	3	3
crossings	Total	16	27	22	16	23	22	16	16	23	16	27	24
crossings	Unfairness	4.47	0.00	0.59	7.65	4.71	0.59	5.53	5.53	0.47	7.65	0.00	0.35
skewness	Blue	-	-	-	3	4	5	-	-	-	3	4	5
skewness	Red	-	-	-	1	0	0	-	-	-	1	0	0
skewness	Total	-	-	-	4	4	5	-	-	-	4	4	5
skewness	Unfairness	-	-	-	0.82	0.24	0.29	-	-	-	0.82	0.24	0.29
wiggles	Blue	-	-	-	-	-	-	41.0	51.0	72.0	37.0	68.0	71.0
wiggles	Red	-	-	-	-	-	-	6.0	3.0	0.0	10.0	4.0	0.0
wiggles	Total	-	-	-	-	-	-	47	54	72	47	72	71
wiggles	Unfairness	-	-	-	-	-	-	3.59	0.00	4.24	7.82	0.00	4.18

dblp — coloring 2

		cross: Standard	cross: Fair	cross: Focus	cross + skew: Standard	cross + skew: Fair	cross + skew: Focus	cross + wiggles: Standard	cross + wiggles: Fair	cross + wiggles: Focus	cross + skew + wiggles: Standard	cross + skew + wiggles: Fair	cross + skew + wiggles: Focus
	Time (s):	0.3	oom	1.1	0.4	0.4	1.2	210	oot	11.6	200	oom	18.7
crossings	Blue-Blue	4	16	12	3	3	12	3	3	14	3	17	12
crossings	Red-Red	1	1	1	1	1	1	1	1	1	1	2	1
crossings	Blue-Red	11	10	8	12	12	8	12	12	9	12	8	8
crossings	Total	16	27	21	16	16	21	16	16	24	16	27	21
crossings	Unfairness	1.89	0.00	0.21	2.21	2.21	0.21	2.21	2.21	0.11	2.21	0.00	0.21
skewness	Blue	-	-	-	3	3	3	-	-	-	3	3	3
skewness	Red	-	-	-	1	1	1	-	-	-	1	1	1
skewness	Total	-	-	-	4	4	4	-	-	-	4	4	4
skewness	Unfairness	-	-	-	0.04	0.04	0.04	-	-	-	0.04	0.04	0.04
wiggles	Blue	-	-	-	-	-	-	33.0	42.0	59.0	33.0	49.0	52.0
wiggles	Red	-	-	-	-	-	-	14.0	12.0	6.0	14.0	14.0	8.0
wiggles	Total	-	-	-	-	-	-	47	54	65	47	63	60
wiggles	Unfairness	-	-	-	-	-	-	1.14	0.00	2.71	1.14	0.00	1.71

dblp — coloring 3

		cross: Standard	cross: Fair	cross: Focus	cross + skew: Standard	cross + skew: Fair	cross + skew: Focus	cross + wiggles: Standard	cross + wiggles: Fair	cross + wiggles: Focus	cross + skew + wiggles: Standard	cross + skew + wiggles: Fair	cross + skew + wiggles: Focus
	Time (s):	0.4	21.8	1.7	0.5	11.9	1.6	303	oot	42.4	309	oom	18.2
crossings	Blue-Blue	6	5	16	7	7	16	5	5	14	7	6	16
crossings	Red-Red	3	3	1	3	3	1	3	3	1	3	4	1
crossings	Blue-Red	7	10	11	6	6	11	8	8	12	6	8	11
crossings	Total	16	18	28	16	16	28	16	16	27	16	18	28
crossings	Unfairness	0.27	0.00	2.67	0.50	0.50	2.67	0.05	0.05	2.25	0.50	0.00	2.67
skewness	Blue	-	-	-	2	2	4	-	-	-	2	2	4
skewness	Red	-	-	-	2	2	1	-	-	-	2	2	1
skewness	Total	-	-	-	4	4	5	-	-	-	4	4	5
skewness	Unfairness	-	-	-	0.05	0.05	0.28	-	-	-	0.05	0.05	0.28
wiggles	Blue	-	-	-	-	-	-	30.0	30.0	57.0	34.0	30.0	66.0
wiggles	Red	-	-	-	-	-	-	17.0	24.0	10.0	13.0	24.0	10.0
wiggles	Total	-	-	-	-	-	-	47	54	67	47	54	76
wiggles	Unfairness	-	-	-	-	-	-	0.88	0.00	4.45	1.77	0.00	5.35

Table 6: *Result values for star wars.*

Star Wars — coloring 1

		cross: Standard	cross: Fair	cross: Focus	cross + skew: Standard	cross + skew: Fair	cross + skew: Focus	cross + wiggles: Standard	cross + wiggles: Fair	cross + wiggles: Focus	cross + skew + wiggles: Standard	cross + skew + wiggles: Fair	cross + skew + wiggles: Focus
	Time (s):	0.7	4.4	0.4	51.2	3.6	1.7	165	233	22.0	229	oot	147
crossings	Blue-Blue	28	36	46	28	36	46	28	30	46	28	36	46
crossings	Red-Red	0	0	0	0	0	0	0	0	0	0	0	0
crossings	Blue-Red	11	6	1	11	5	1	11	9	1	11	6	1
crossings	Total	39	42	47	39	41	47	39	39	47	39	42	47
crossings	Unfairness	5.85	0.00	6.15	5.85	0.92	6.15	5.85	3.69	6.15	5.85	0.00	6.15
skewness	Blue	-	-	-	5	5	5	-	-	-	5	5	6
skewness	Red	-	-	-	1	0	0	-	-	-	1	0	0
skewness	Total	-	-	-	6	5	5	-	-	-	6	5	6
skewness	Unfairness	-	-	-	0.62	0.38	0.38	-	-	-	0.62	0.38	0.46
wiggles	Blue	-	-	-	-	-	-	57.0	65.0	100.0	60.0	78.0	100.0
wiggles	Red	-	-	-	-	-	-	11.0	5.0	0.0	8.0	6.0	0.0
wiggles	Total	-	-	-	-	-	-	68	70	100	68	84	100
wiggles	Unfairness	-	-	-	-	-	-	6.62	0.00	7.69	3.38	0.00	7.69

Star Wars — coloring 2

		cross: Standard	cross: Fair	cross: Focus	cross + skew: Standard	cross + skew: Fair	cross + skew: Focus	cross + wiggles: Standard	cross + wiggles: Fair	cross + wiggles: Focus	cross + skew + wiggles: Standard	cross + skew + wiggles: Fair	cross + skew + wiggles: Focus
	Time (s):	0.7	11.5	8.5	69.3	13.0	20.7	175	oot	42.2	234	oot	84.8
crossings	Blue-Blue	23	35	27	23	23	27	23	6	27	23	35	27
crossings	Red-Red	1	0	0	1	1	0	1	13	0	1	0	0
crossings	Blue-Red	15	14	14	15	15	14	15	23	15	15	13	15
crossings	Total	39	49	41	39	39	41	39	42	42	39	48	42
crossings	Unfairness	3.42	0.00	1.33	3.42	3.42	1.33	3.42	5.91	1.75	3.42	0.42	1.75
skewness	Blue	-	-	-	5	6	6	-	-	-	5	6	7
skewness	Red	-	-	-	1	1	0	-	-	-	1	1	0
skewness	Total	-	-	-	6	7	6	-	-	-	6	7	7
skewness	Unfairness	-	-	-	0.08	0.00	0.50	-	-	-	0.08	0.00	0.58
wiggles	Blue	-	-	-	-	-	-	53.0	54.0	74.0	55.0	72.0	74.0
wiggles	Red	-	-	-	-	-	-	15.0	30.0	8.0	13.0	12.0	8.0
wiggles	Total	-	-	-	-	-	-	68	84	82	68	84	82
wiggles	Unfairness	-	-	-	-	-	-	3.08	0.00	2.17	1.92	0.00	2.17

Star Wars — coloring 3

		cross: Standard	cross: Fair	cross: Focus	cross + skew: Standard	cross + skew: Fair	cross + skew: Focus	cross + wiggles: Standard	cross + wiggles: Fair	cross + wiggles: Focus	cross + skew + wiggles: Standard	cross + skew + wiggles: Fair	cross + skew + wiggles: Focus
	Time (s):	0.7	19.6	8.9	53.6	40.3	37.1	164	oot	20.3	236	oot	54.0
crossings	Blue-Blue	5	29	9	5	7	9	5	6	10	5	29	10
crossings	Red-Red	15	11	11	15	11	11	15	13	11	15	11	11
crossings	Blue-Red	19	23	22	19	23	22	19	23	25	19	23	25
crossings	Total	39	63	42	39	41	42	39	42	46	39	63	46
crossings	Unfairness	6.58	0.00	4.36	6.58	4.89	4.36	6.58	5.91	4.40	6.58	0.00	4.40
skewness	Blue	-	-	-	2	4	3	-	-	-	2	4	3
skewness	Red	-	-	-	4	2	2	-	-	-	4	2	3
skewness	Total	-	-	-	6	6	5	-	-	-	6	6	6
skewness	Unfairness	-	-	-	0.58	0.04	0.07	-	-	-	0.58	0.04	0.27
wiggles	Blue	-	-	-	-	-	-	29.0	54.0	46.0	30.0	63.0	46.0
wiggles	Red	-	-	-	-	-	-	39.0	30.0	28.0	38.0	35.0	28.0
wiggles	Total	-	-	-	-	-	-	68	84	74	68	98	74
wiggles	Unfairness	-	-	-	-	-	-	4.58	0.00	0.49	4.27	0.00	0.49

Table 7: *Result values for the lord of the rings.*

Lord of the Rings

coloring 1

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	3.2	oom	2.9	139	43.7	8.9	4972	oot	468	2465	oot	934
crossings	Blue-Blue	14	20	22	14	21	22	15	14	24	14	24	23
crossings	Red-Red	0	0	0	0	0	0	0	0	0	0	0	0
crossings	Blue-Red	6	5	0	6	2	0	6	6	0	6	6	0
crossings	Total	20	25	22	20	23	22	21	20	24	20	30	23
crossings	Unfairness	1.11	0.00	2.44	1.11	1.44	2.44	1.00	1.11	2.67	1.11	0.00	2.56
skewness	Blue	-	-	-	5	2	4	-	-	-	5	2	4
skewness	Red	-	-	-	0	0	0	-	-	-	0	0	0
skewness	Total	-	-	-	5	2	4	-	-	-	5	2	4
skewness	Unfairness	-	-	-	0.28	0.11	0.22	-	-	-	0.28	0.11	0.22
wiggles	Blue	-	-	-	-	-	-	43.0	54.0	75.0	44.0	81.0	76.0
wiggles	Red	-	-	-	-	-	-	11.0	6.0	0.0	11.0	9.0	0.0
wiggles	Total	-	-	-	-	-	-	54	60	75	55	90	76
wiggles	Unfairness	-	-	-	-	-	-	3.11	0.00	4.17	3.06	0.00	4.22

coloring 2

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	2.5	oom	2.7	108	21.4	14.2	4345	6149	133	2169	10080	425
crossings	Blue-Blue	6	19	19	6	6	19	7	14	22	6	17	20
crossings	Red-Red	2	4	1	2	2	1	2	2	1	2	2	1
crossings	Blue-Red	12	2	4	12	12	4	12	10	6	12	6	6
crossings	Total	20	25	24	20	20	24	21	26	29	20	25	27
crossings	Unfairness	2.50	0.00	1.12	2.50	2.50	1.12	2.38	1.12	1.12	2.50	0.00	0.88
skewness	Blue	-	-	-	4	4	3	-	-	-	4	4	3
skewness	Red	-	-	-	1	1	1	-	-	-	1	1	1
skewness	Total	-	-	-	5	5	4	-	-	-	5	5	4
skewness	Unfairness	-	-	-	0.00	0.00	0.06	-	-	-	0.00	0.00	0.06
wiggles	Blue	-	-	-	-	-	-	36.0	44.0	68.0	37.0	52.0	70.0
wiggles	Red	-	-	-	-	-	-	18.0	11.0	5.0	18.0	13.0	5.0
wiggles	Total	-	-	-	-	-	-	54	55	73	55	65	75
wiggles	Unfairness	-	-	-	-	-	-	2.25	0.00	3.00	2.19	0.00	3.12

coloring 3

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	3.0	oot	2.7	135	70.4	5.4	5513	oot	137	3309	oot	104
crossings	Blue-Blue	2	5	2	2	3	2	3	3	5	2	6	4
crossings	Red-Red	12	2	2	12	2	2	12	12	2	12	3	2
crossings	Blue-Red	6	23	18	6	18	18	6	6	20	6	21	20
crossings	Total	20	30	22	20	23	22	21	21	27	20	30	26
crossings	Unfairness	2.42	0.00	0.44	2.42	0.26	0.44	2.24	2.24	0.06	2.42	0.00	0.12
skewness	Blue	-	-	-	2	1	2	-	-	-	2	4	4
skewness	Red	-	-	-	3	1	1	-	-	-	3	3	1
skewness	Total	-	-	-	5	2	3	-	-	-	5	7	5
skewness	Unfairness	-	-	-	0.15	0.02	0.07	-	-	-	0.15	0.03	0.25
wiggles	Blue	-	-	-	-	-	-	14.0	44.0	37.0	15.0	38.0	38.0
wiggles	Red	-	-	-	-	-	-	40.0	36.0	31.0	40.0	31.0	31.0
wiggles	Total	-	-	-	-	-	-	54	80	68	55	69	69
wiggles	Unfairness	-	-	-	-	-	-	3.17	0.00	0.08	3.08	0.01	0.01

Table 8: *Result values for animal farm.*

Animal Farm

coloring 1

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.1	oom	0.9	0.3	0.5	0.4	96.0	80.0	12.9	166	oot	23.1
crossings	Blue-Blue	12	30	28	12	15	28	12	14	20	12	30	20
crossings	Red-Red	0	0	0	0	0	0	0	0	0	0	0	0
crossings	Blue-Red	5	4	1	5	4	1	5	4	2	5	4	2
crossings	Total	17	34	29	17	19	29	17	18	22	17	34	22
crossings	Unfairness	3.19	0.00	2.56	3.19	1.88	2.56	3.19	2.00	0.62	3.19	0.00	0.62
skewness	Blue	-	-	-	3	3	4	-	-	-	4	3	4
skewness	Red	-	-	-	1	0	0	-	-	-	0	0	0
skewness	Total	-	-	-	4	3	4	-	-	-	4	3	4
skewness	Unfairness	-	-	-	0.81	0.19	0.25	-	-	-	0.25	0.19	0.25
wiggles	Blue	-	-	-	-	-	-	41.0	48.0	63.0	41.0	64.0	63.0
wiggles	Red	-	-	-	-	-	-	8.0	3.0	0.0	8.0	4.0	0.0
wiggles	Total	-	-	-	-	-	-	49	51	63	49	68	63
wiggles	Unfairness	-	-	-	-	-	-	5.44	0.00	3.94	5.44	0.00	3.94

coloring 2

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.1	1.1	0.1	0.3	2.1	0.5	71.5	oot	3.8	150	129	3.1
crossings	Blue-Blue	3	26	22	3	3	22	3	7	17	3	26	17
crossings	Red-Red	5	4	0	5	5	0	5	6	0	5	4	0
crossings	Blue-Red	9	4	12	9	9	12	9	8	13	9	4	13
crossings	Total	17	34	34	17	17	34	17	21	30	17	34	30
crossings	Unfairness	5.26	0.00	0.00	5.26	5.26	0.00	5.26	5.10	0.98	5.26	0.00	0.98
skewness	Blue	-	-	-	2	5	4	-	-	-	3	5	4
skewness	Red	-	-	-	2	1	0	-	-	-	1	1	0
skewness	Total	-	-	-	4	6	4	-	-	-	4	6	4
skewness	Unfairness	-	-	-	0.52	0.02	0.29	-	-	-	0.12	0.02	0.29
wiggles	Blue	-	-	-	-	-	-	26.0	56.0	71.0	26.0	56.0	71.0
wiggles	Red	-	-	-	-	-	-	23.0	12.0	6.0	23.0	12.0	6.0
wiggles	Total	-	-	-	-	-	-	49	68	77	49	68	77
wiggles	Unfairness	-	-	-	-	-	-	5.81	0.00	3.07	5.81	0.00	3.07

coloring 3

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.1	oot	0.2	0.2	0.5	0.5	73.7	oot	5.2	152	oot	8.0
crossings	Blue-Blue	1	15	4	1	1	4	1	4	4	1	15	4
crossings	Red-Red	10	9	9	10	9	9	10	9	9	10	9	9
crossings	Blue-Red	6	10	5	6	7	5	6	5	5	6	10	5
crossings	Total	17	34	18	17	17	18	17	18	18	17	34	18
crossings	Unfairness	2.91	0.00	1.99	2.91	2.67	1.99	2.91	1.99	1.99	2.91	0.00	1.99
skewness	Blue	-	-	-	2	2	3	-	-	-	3	2	3
skewness	Red	-	-	-	2	2	1	-	-	-	1	2	1
skewness	Total	-	-	-	4	4	4	-	-	-	4	4	4
skewness	Unfairness	-	-	-	0.09	0.09	0.16	-	-	-	0.16	0.09	0.16
wiggles	Blue	-	-	-	-	-	-	17.0	40.0	24.0	17.0	40.0	24.0
wiggles	Red	-	-	-	-	-	-	32.0	28.0	26.0	32.0	28.0	26.0
wiggles	Total	-	-	-	-	-	-	49	68	50	49	68	50
wiggles	Unfairness	-	-	-	-	-	-	2.87	0.00	1.31	2.87	0.00	1.31

Table 9: *Result values for anna3.*

Anna 3

coloring 1

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.1	0.0	0.1	0.1	0.1	0.1	7.6	31.5	10.9	13.7	18.2	15.8
crossings	Blue-Blue	**0**	**0**	**0**	**0**	**0**	**0**	1	**0**	**0**	1	**0**	**0**
	Red-Red	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	Blue-Red	**0**	**0**	**0**	**0**	**0**	**0**	1	**0**	**0**	1	**0**	**0**
	Total	**0**	**0**	**0**	**0**	**0**	**0**	2	**0**	**0**	2	**0**	**0**
	Unfairness	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.93	**0.00**	**0.00**	0.93	**0.00**	**0.00**
skewness	Blue	-	-	-	**0**	**0**	**0**	-	-	-	1	**0**	**0**
	Red	-	-	-	**0**	**0**	**0**	-	-	-	**0**	**0**	**0**
	Total	-	-	-	**0**	**0**	**0**	-	-	-	1	**0**	**0**
	Unfairness	-	-	-	**0.00**	**0.00**	**0.00**	-	-	-	0.02	**0.00**	**0.00**
wiggles	Blue	-	-	-	-	-	-	**22.0**	45.0	48.0	**22.0**	45.0	48.0
	Red	-	-	-	-	-	-	6.0	1.0	**0.0**	6.0	1.0	**0.0**
	Total	-	-	-	-	-	-	**28**	46	48	**28**	46	48
	Unfairness	-	-	-	-	-	-	5.51	**0.00**	1.07	5.51	**0.00**	1.07

coloring 2

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.1	0.0	0.1	0.1	0.1	0.1	10.1	oot	2.5	11.5	oom	3.0
crossings	Blue-Blue	**0**	**0**	**0**	**0**	**0**	**0**	**0**	0	3	**0**	0	3
	Red-Red	**0**	**0**	**0**	**0**	**0**	**0**	1	1	**0**	1	0	**0**
	Blue-Red	**0**	**0**	**0**	**0**	**0**	**0**	1	2	1	1	0	1
	Total	**0**	**0**	**0**	**0**	**0**	**0**	2	3	4	2	0	4
	Unfairness	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.40	0.52	0.04	0.40	0.00	0.04
skewness	Blue	-	-	-	**0**	**0**	**0**	-	-	-	**0**	0	2
	Red	-	-	-	**0**	**0**	**0**	-	-	-	1	0	**0**
	Total	-	-	-	**0**	**0**	**0**	-	-	-	1	0	2
	Unfairness	-	-	-	**0.00**	**0.00**	**0.00**	-	-	-	0.14	0.00	0.05
wiggles	Blue	-	-	-	-	-	-	12.0	39.0	37.0	**11.0**	128.0	37.0
	Red	-	-	-	-	-	-	16.0	7.0	11.0	17.0	23.0	11.0
	Total	-	-	-	-	-	-	**28**	46	48	**28**	151	48
	Unfairness	-	-	-	-	-	-	1.98	0.00	0.62	2.15	0.00	0.62

coloring 3

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.1	0.1	0.1	0.1	0.0	0.1	10.0	oot	10.1	11.4	oom	13.9
crossings	Blue-Blue	**0**	**0**	**0**	**0**	**0**	**0**	**0**	0	**0**	**0**	0	**0**
	Red-Red	**0**	**0**	**0**	**0**	**0**	**0**	2	2	**0**	2	0	**0**
	Blue-Red	**0**	**0**	**0**	**0**	**0**	**0**	**0**	0	**0**	**0**	0	**0**
	Total	**0**	**0**	**0**	**0**	**0**	**0**	2	2	**0**	2	0	**0**
	Unfairness	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.22	0.22	**0.00**	0.22	0.00	**0.00**
skewness	Blue	-	-	-	**0**	**0**	**0**	-	-	-	**0**	0	**0**
	Red	-	-	-	**0**	**0**	**0**	-	-	-	1	0	**0**
	Total	-	-	-	**0**	**0**	**0**	-	-	-	1	0	**0**
	Unfairness	-	-	-	**0.00**	**0.00**	**0.00**	-	-	-	0.06	0.00	**0.00**
wiggles	Blue	-	-	-	-	-	-	**17.0**	17.0	26.0	**17.0**	31.0	26.0
	Red	-	-	-	-	-	-	**11.0**	11.0	19.0	**11.0**	20.0	19.0
	Total	-	-	-	-	-	-	**28**	28	45	**28**	51	45
	Unfairness	-	-	-	-	-	-	**0.00**	0.00	0.13	**0.00**	0.00	0.13

Table 10: *Result values for ffvii.*

ffvii

coloring 1

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	1.2	oot	1.0	6.9	148	2.4	1938	2655	376	3364	oot	975
crossings	Blue-Blue	**21**	30	27	25	23	27	25	25	28	25	30	28
	Red-Red	**0**	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	0	**0**
	Blue-Red	5	4	**0**	1	3	**0**	4	4	**0**	4	4	**0**
	Total	**26**	34	27	**26**	**26**	27	29	29	28	29	34	28
	Unfairness	2.06	0.00	3.38	2.19	0.06	3.38	0.62	0.62	3.50	0.62	0.00	3.50
skewness	Blue	-	-	-	**6**	**6**	**6**	-	-	-	**6**	6	**6**
	Red	-	-	-	**0**	**0**	**0**	-	-	-	**0**	0	**0**
	Total	-	-	-	**6**	**6**	**6**	-	-	-	**6**	6	**6**
	Unfairness	-	-	-	**0.38**	**0.38**	**0.38**	-	-	-	**0.38**	0.38	**0.38**
wiggles	Blue	-	-	-	-	-	-	**64.0**	**64.0**	80.0	**64.0**	80.0	80.0
	Red	-	-	-	-	-	-	3.0	4.0	**0.0**	3.0	5.0	**0.0**
	Total	-	-	-	-	-	-	**67**	68	80	**67**	85	80
	Unfairness	-	-	-	-	-	-	1.00	**0.00**	5.00	1.00	0.00	5.00

coloring 2

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	1.4	12.4	1.6	8.8	24.5	1.8	2462	oot	99.6	3177	oot	164
crossings	Blue-Blue	11	24	15	**10**	11	15	15	11	16	15	24	16
	Red-Red	6	2	**0**	6	6	**0**	2	2	**0**	2	2	**0**
	Blue-Red	9	**8**	15	10	9	15	12	13	17	12	8	17
	Total	**26**	34	30	**26**	**26**	30	29	26	33	29	34	33
	Unfairness	4.79	**0.00**	1.79	5.19	4.79	1.79	2.33	3.17	2.17	2.33	0.00	2.17
skewness	Blue	-	-	-	**5**	**5**	9	-	-	-	**5**	5	10
	Red	-	-	-	1	1	**0**	-	-	-	1	1	**0**
	Total	-	-	-	**6**	**6**	9	-	-	-	**6**	6	10
	Unfairness	-	-	-	**0.02**	**0.02**	0.64	-	-	-	**0.02**	0.02	0.71
wiggles	Blue	-	-	-	-	-	-	**42.0**	70.0	69.0	**42.0**	70.0	69.0
	Red	-	-	-	-	-	-	25.0	15.0	**11.0**	25.0	15.0	**11.0**
	Total	-	-	-	-	-	-	**67**	85	80	**67**	85	80
	Unfairness	-	-	-	-	-	-	5.33	0.00	1.26	5.33	0.00	1.26

coloring 3

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	1.5	oot	1.1	9.6	25.3	1.6	3082	oot	32.6	3324	oot	45.9
crossings	Blue-Blue	7	12	7	5	**4**	7	6	6	7	6	12	7
	Red-Red	9	6	**5**	7	10	**5**	6	6	**5**	6	6	**5**
	Blue-Red	**10**	16	15	14	14	15	17	14	17	17	16	17
	Total	**26**	34	27	**26**	28	27	29	26	29	29	34	29
	Unfairness	1.60	0.00	0.67	1.60	2.66	0.67	1.24	1.11	0.76	1.24	0.00	0.76
skewness	Blue	-	-	-	4	**3**	4	-	-	-	4	4	4
	Red	-	-	-	**2**	**2**	**2**	-	-	-	**2**	3	**2**
	Total	-	-	-	6	**5**	6	-	-	-	6	7	6
	Unfairness	-	-	-	0.11	**0.01**	0.11	-	-	-	0.11	0.03	0.11
wiggles	Blue	-	-	-	-	-	-	**33.0**	50.0	52.0	**33.0**	50.0	52.0
	Red	-	-	-	-	-	-	34.0	35.0	**27.0**	34.0	35.0	**27.0**
	Total	-	-	-	-	-	-	**67**	85	79	**67**	85	79
	Unfairness	-	-	-	-	-	-	1.56	0.00	1.34	1.56	0.00	1.34

Table 11: *Result values for jean1.*

jean1

coloring 1

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.4	0.3	0.3	0.8	2.2	0.6	185	oot	29.4	399	oom	26.4
crossings	Blue-Blue	7	19	17	4	5	17	7	6	19	5	19	18
crossings	Red-Red	0	0	0	0	0	0	0	0	0	0	0	0
crossings	Blue-Red	3	1	0	6	5	0	5	4	0	7	1	0
crossings	Total	10	20	17	10	10	17	12	10	19	12	20	18
crossings	Unfairness	2.56	0.00	0.87	5.64	4.62	0.87	4.51	3.59	0.97	6.56	0.00	0.92
skewness	Blue	-	-	-	2	4	4	-	-	-	2	4	4
skewness	Red	-	-	-	1	0	0	-	-	-	1	0	0
skewness	Total	-	-	-	3	4	4	-	-	-	3	4	4
skewness	Unfairness	-	-	-	0.95	0.10	0.10	-	-	-	0.95	0.10	0.10
wiggles	Blue	-	-	-	-	-	-	44.0	78.0	73.0	44.0	78.0	74.0
wiggles	Red	-	-	-	-	-	-	11.0	2.0	0.0	12.0	2.0	0.0
wiggles	Total	-	-	-	-	-	-	55	80	73	56	80	74
wiggles	Unfairness	-	-	-	-	-	-	9.87	0.00	1.87	10.87	0.00	1.90

coloring 2

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.4	oot	0.3	0.9	4.2	0.6	199	oot	16.6	427	oot	35.0
crossings	Blue-Blue	2	18	5	0	2	5	0	1	5	0	20	5
crossings	Red-Red	3	1	1	6	5	1	5	4	1	7	3	1
crossings	Blue-Red	5	7	7	4	3	7	7	5	8	5	3	8
crossings	Total	10	26	13	10	10	13	12	10	14	12	26	14
crossings	Unfairness	1.30	0.02	0.77	2.16	1.65	0.77	2.22	1.65	0.88	2.56	0.02	0.88
skewness	Blue	-	-	-	0	3	3	-	-	-	0	3	4
skewness	Red	-	-	-	3	1	1	-	-	-	3	1	1
skewness	Total	-	-	-	3	4	4	-	-	-	3	4	5
skewness	Unfairness	-	-	-	0.43	0.05	0.05	-	-	-	0.43	0.05	0.02
wiggles	Blue	-	-	-	-	-	-	25.0	66.0	64.0	27.0	80.0	64.0
wiggles	Red	-	-	-	-	-	-	30.0	14.0	17.0	29.0	17.0	17.0
wiggles	Total	-	-	-	-	-	-	55	80	81	56	97	81
wiggles	Unfairness	-	-	-	-	-	-	3.53	0.00	0.49	3.32	0.00	0.49

coloring 3

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.4	oom	0.4	1.7	0.8	0.8	238	oot	15.7	496	oot	20.7
crossings	Blue-Blue	0	8	1	0	1	1	0	0	2	0	8	2
crossings	Red-Red	6	5	4	9	6	4	10	9	4	11	6	4
crossings	Blue-Red	4	7	6	1	3	6	2	4	9	1	6	9
crossings	Total	10	20	11	10	10	11	12	13	15	12	20	15
crossings	Unfairness	0.77	0.00	0.48	1.07	0.66	0.48	1.21	1.12	0.43	1.31	0.10	0.43
skewness	Blue	-	-	-	0	2	1	-	-	-	0	2	3
skewness	Red	-	-	-	3	2	3	-	-	-	3	2	3
skewness	Total	-	-	-	3	4	4	-	-	-	3	4	6
skewness	Unfairness	-	-	-	0.18	0.03	0.13	-	-	-	0.18	0.03	0.05
wiggles	Blue	-	-	-	-	-	-	15.0	34.0	53.0	17.0	46.0	53.0
wiggles	Red	-	-	-	-	-	-	40.0	25.0	34.0	39.0	34.0	34.0
wiggles	Total	-	-	-	-	-	-	55	59	87	56	80	87
wiggles	Unfairness	-	-	-	-	-	-	1.70	0.01	0.30	1.55	0.00	0.30

Table 12: *Result values for jean2.*

jean2

coloring 1

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.1	0.1	0.1	0.1	0.1	0.1	6.7	129	6.1	19.1	13.0	18.1
crossings	Blue-Blue	6	6	6	6	6	6	6	6	6	6	6	6
crossings	Red-Red	0	0	0	0	0	0	0	0	0	0	0	0
crossings	Blue-Red	0	0	0	0	0	0	0	0	0	0	0	0
crossings	Total	6	6	6	6	6	6	6	6	6	6	6	6
crossings	Unfairness	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
skewness	Blue	-	-	-	1	1	1	-	-	-	1	1	1
skewness	Red	-	-	-	0	0	0	-	-	-	0	0	0
skewness	Total	-	-	-	1	1	1	-	-	-	1	1	1
skewness	Unfairness	-	-	-	0.08	0.08	0.08	-	-	-	0.08	0.08	0.08
wiggles	Blue	-	-	-	-	-	-	23.0	23.0	23.0	25.0	25.0	25.0
wiggles	Red	-	-	-	-	-	-	0	0	0	0	0	0
wiggles	Total	-	-	-	-	-	-	23	23	23	25	25	25
wiggles	Unfairness	-	-	-	-	-	-	1.77	1.77	1.77	1.92	1.92	1.92

coloring 2

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.1	0.3	0.2	0.1	0.1	0.2	3.9	401	1.9	11.4	9.1	2.0
crossings	Blue-Blue	2	4	15	4	4	15	3	2	6	4	4	6
crossings	Red-Red	0	0	0	0	0	0	0	0	0	0	0	0
crossings	Blue-Red	4	3	0	2	2	0	3	4	1	2	3	1
crossings	Total	6	7	15	6	6	15	6	6	7	6	7	7
crossings	Unfairness	0.61	0.00	2.73	0.24	0.24	2.73	0.18	0.61	0.85	0.24	0.00	0.85
skewness	Blue	-	-	-	1	1	2	-	-	-	1	1	2
skewness	Red	-	-	-	0	0	0	-	-	-	0	0	0
skewness	Total	-	-	-	1	1	2	-	-	-	1	1	2
skewness	Unfairness	-	-	-	0.09	0.09	0.18	-	-	-	0.09	0.09	0.18
wiggles	Blue	-	-	-	-	-	-	21.0	22.0	27.0	23.0	22.0	27.0
wiggles	Red	-	-	-	-	-	-	2.0	6.0	0.0	2.0	6.0	0.0
wiggles	Total	-	-	-	-	-	-	23	28	27	25	28	27
wiggles	Unfairness	-	-	-	-	-	-	1.24	0.00	2.45	1.42	0.00	2.45

coloring 3

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.1	0.1	0.2	0.1	0.1	0.1	4.0	21.5	1.0	11.0	8.3	1.9
crossings	Blue-Blue	2	1	15	4	4	15	3	1	6	4	1	6
crossings	Red-Red	0	0	0	0	0	0	0	0	0	0	0	0
crossings	Blue-Red	4	6	0	2	2	0	3	7	1	2	6	1
crossings	Total	6	7	15	6	6	15	6	8	7	6	7	7
crossings	Unfairness	0.33	0.00	3.75	0.92	0.92	3.75	0.62	0.04	1.46	0.92	0.00	1.46
skewness	Blue	-	-	-	1	1	2	-	-	-	1	1	2
skewness	Red	-	-	-	0	0	0	-	-	-	0	0	0
skewness	Total	-	-	-	1	1	2	-	-	-	1	1	2
skewness	Unfairness	-	-	-	0.12	0.12	0.25	-	-	-	0.12	0.12	0.25
wiggles	Blue	-	-	-	-	-	-	21.0	16.0	27.0	23.0	20.0	27.0
wiggles	Red	-	-	-	-	-	-	2.0	12.0	0.0	2.0	15.0	0.0
wiggles	Total	-	-	-	-	-	-	23	28	27	25	35	27
wiggles	Unfairness	-	-	-	-	-	-	2.29	0.00	3.38	2.54	0.00	3.38

Table 13: *Result values for jean5.*

jean5

coloring 1

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.8	oot	0.7	2.1	7.8	4.6	870	oot	304	1275	oom	222
crossings	Blue-Blue	**8**	27	29	**8**	22	29	11	12	20	11	27	19
crossings	Red-Red	**0**	0	**0**	**0**	**0**	**0**	**0**	0	**0**	**0**	0	**0**
crossings	Blue-Red	9	3	**0**	9	4	**0**	8	5	2	7	3	2
crossings	Total	**17**	30	29	**17**	26	29	19	17	22	18	30	21
crossings	Unfairness	7.68	0.00	3.05	7.68	1.47	3.05	6.42	3.47	0.21	5.47	0.00	0.11
skewness	Blue	-	-	-	**3**	**3**	5	-	-	-	4	3	5
skewness	Red	-	-	-	1	**0**	**0**	-	-	-	1	0	**0**
skewness	Total	-	-	-	4	**3**	5	-	-	-	5	3	5
skewness	Unfairness	-	-	-	0.84	**0.16**	0.26	-	-	-	0.79	0.16	0.26
wiggles	Blue	-	-	-	-	-	-	34.0	57.0	61.0	**33.0**	76.0	62.0
wiggles	Red	-	-	-	-	-	-	8.0	3.0	**0.0**	10.0	4.0	**0.0**
wiggles	Total	-	-	-	-	-	-	**42**	60	61	43	80	62
wiggles	Unfairness	-	-	-	-	-	-	6.21	0.00	3.21	8.26	0.00	3.26

coloring 2

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.6	38.0	0.4	1.6	2.1	1.2	884	298	59.2	972	2819	146
crossings	Blue-Blue	**6**	20	11	**6**	9	11	**6**	9	9	**6**	25	9
crossings	Red-Red	6	2	**1**	5	5	**1**	6	4	**1**	5	4	**1**
crossings	Blue-Red	5	8	9	6	**3**	9	7	8	13	7	6	13
crossings	Total	**17**	30	21	**17**	**17**	21	19	21	23	18	35	23
crossings	Unfairness	3.19	**0.00**	0.81	2.88	1.94	0.81	3.56	2.38	1.81	3.06	**0.00**	1.81
skewness	Blue	-	-	-	**2**	4	3	-	-	-	3	4	4
skewness	Red	-	-	-	2	**1**	**1**	-	-	-	2	**1**	**1**
skewness	Total	-	-	-	**4**	5	**4**	-	-	-	5	5	5
skewness	Unfairness	-	-	-	0.38	**0.00**	0.06	-	-	-	0.31	**0.00**	**0.00**
wiggles	Blue	-	-	-	-	-	-	21.0	48.0	50.0	**20.0**	60.0	50.0
wiggles	Red	-	-	-	-	-	-	21.0	**12.0**	18.0	23.0	15.0	18.0
wiggles	Total	-	-	-	-	-	-	**42**	60	68	43	75	68
wiggles	Unfairness	-	-	-	-	-	-	3.94	**0.00**	1.38	4.50	**0.00**	1.38

coloring 3

		optimize for: cross			optimize for: cross + skew			optimize for: cross + wiggles			optimize for: cross + skew + wiggles		
		Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus	Standard	Fair	Focus
	Time (s):	0.7	858	0.8	1.7	6.5	1.4	668	655	80.2	1097	oot	129
crossings	Blue-Blue	2	13	**1**	2	2	**1**	2	2	2	2	17	**1**
crossings	Red-Red	13	10	**9**	13	15	**9**	15	12	10	14	13	**9**
crossings	Blue-Red	**2**	7	10	**2**	**2**	10	**2**	9	5	**2**	10	10
crossings	Total	**17**	30	20	**17**	19	20	19	23	**17**	18	40	20
crossings	Unfairness	2.57	**0.00**	2.02	2.57	3.01	2.02	3.01	2.48	1.96	2.79	0.00	2.02
skewness	Blue	-	-	-	**1**	2	**1**	-	-	-	**1**	2	3
skewness	Red	-	-	-	3	**2**	3	-	-	-	4	2	3
skewness	Total	-	-	-	**4**	**4**	**4**	-	-	-	5	4	6
skewness	Unfairness	-	-	-	0.24	**0.04**	0.24	-	-	-	0.35	0.04	0.06
wiggles	Blue	-	-	-	-	-	-	**8.0**	33.0	18.0	**8.0**	55.0	17.0
wiggles	Red	-	-	-	-	-	-	34.0	**27.0**	39.0	35.0	45.0	45.0
wiggles	Total	-	-	-	-	-	-	**42**	60	57	43	100	62
wiggles	Unfairness	-	-	-	-	-	-	3.05	**0.00**	2.70	3.16	0.00	3.45