MLVis: Machine Learning Methods in Visualisation for Big Data (2024)
Daniel Archambault, Ian Nabney, and Jaakko Peltonen (Editors)

Introducing Fairness in Graph Visualization via Gradient Descent

Seok-Hee Hong[1], Giuseppe Liotta[2†], Fabrizio Montecchiani[2‡], Martin Nöllenburg[3] and Tommaso Piselli[2]

[1] seokhee.hong@sydney.edu.au - The University of Sydney, Australia
[2] {name.surname}@unipg.it - University of Perugia, Italy
[3] noellenburg@ac.tuwien.ac.at - TU Vienna, Austria

Abstract

Motivated by the need for decision-making systems that avoid bias and discrimination, the concept of fairness recently gained traction in the broad field of artificial intelligence, stimulating new research also within the information visualization community. In this paper, we introduce a notion of fairness in network visualization, specifically for straight-line drawings of graphs, a foundational paradigm in the field. We empirically investigate the following research questions: (i) What is the price of incorporating fairness constraints in straight-line drawings? (ii) How unfair is a straight-line drawing that does not optimize fairness as a primary objective? To tackle these questions, we implement an algorithm based on gradient-descent that can compute straight-line drawings of graphs by optimizing multi-objective functions. We experimentally show that one can significantly increase the fairness of a drawing by paying a relatively small amount in terms of reduced readability.

CCS Concepts

• ***Human-centered computing*** → ***Visualization;*** • ***Theory of computation*** → ***Design and analysis of algorithms;***

1. Introduction

In a recent survey concerning bias in machine learning, Mehrabi et al. [MMS*22] defined fairness as the absence of any prejudice or favoritism toward an individual or a group based on their inherent or acquired characteristics. As data-driven decision-making systems are becoming pervasive in our lives, it is indeed of great importance to avoid intentional or unintentional discrimination against certain groups or individuals. Within the broad field of artificial intelligence, the adoption of fairness constraints has been investigated for several algorithmic problems, such as clustering [FKN22, GSV21, KSAM19] and dimensionality reduction [STM*18, TSS*19].

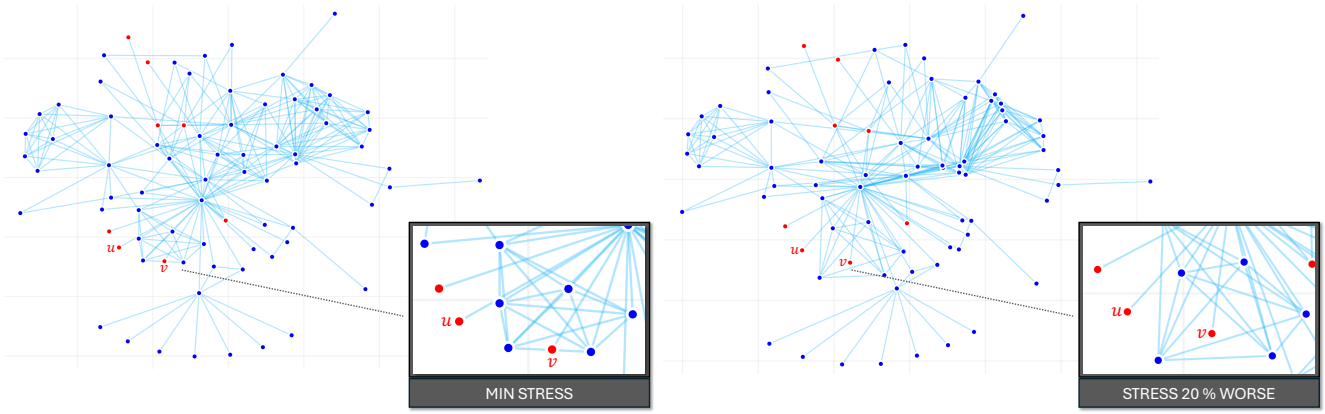
Information visualization tools are the tip of many data-driven decision making systems that require human feedback. Despite the rich body of literature studying fairness in artificial intelligence and related areas, fairness issues in the visualization of information have been surprisingly disregarded. The aim of this paper is to propose a novel research direction on this topic, focused on *fair visualizations of graphs*. Borrowing an example from [FKN22], imagine two competing parties, the reds and the blues. Also, suppose we are given a visualization of the graph modeling the relationships among the parties' members. Using recent graph drawing algorithms, we can effectively optimize a desired set of aesthetic criteria (see, e.g., [ALD*22] for a recent approach), hence producing a readable and effective layout of our graph. However, the global optimization process underlying our drawing algorithm will not give us any guarantee that the readability of the visualization "around" red vertices will be of the same quality as for blue vertices. In fact, while substantially every graph drawing algorithm optimizes global metrics of the computed layout and can easily incorporate local constraints (i.e., at vertex or edge level), only few algorithms can deal with more general constraints at subgroup level [Dwy09, DMW08, HBH18]. In contrast, a fair visualization should guarantee that no party is favored in terms of readability, that is, the possible visual complexity of the representation is equally charged to the two sets, which becomes particularly challenging in the case the cardinalities of the two sets are unbalanced. While a fair drawing might be suboptimal in terms of global readability, it provides greater insight to end users since it balances the readability for the two vertex groups. Figure 1 illustrates an example of the impact of fairness on the readability of a straight-line drawing of a graph with several blue vertices and few red vertices.

Contribution. Our main results are as follows:

- We provide a conceptual contribution by formalizing the notion of fair straight-line graph drawings, based on the concept of stress, a well-known and widely adopted quality function (see,

† Work of Giuseppe Liotta supported in part by MUR of Italy, under PRIN Project n. 2022TS4Y3N - EXPAND.
‡ Work of Fabrizio Montecchiani supported in part by MUR of Italy, under PRIN Project n. 2022ME9Z78 - NextGRAAL, and in part by University of Perugia, Fondo di Ricerca di Ateneo 2022, under Project "MiRA: Mixed Reality and AI Methodologies for Immersive Robotics".

DOI: 10.2312/mlvis.20241124



(a) (b)

Figure 1: *Two straight-line drawings of the same graph. Figure (a) is obtained by optimizing the stress function (a well-known quality function), while Fig. (b) is obtained starting from (a) and by subsequently optimizing the fairness function without worsening the stress of the drawing by more than 20%. One can observe (see the zoomed windows) the increase on the readability around the two red vertices u and v when optimizing fairness (the red vertices are fewer than the blue ones). In particular, in Fig. (a), the edge incident to vertex u overlaps with a blue vertex, while vertex v overlaps with an edge between two blue vertices. Both ambiguities are resolved in Fig. (b).*

e.g., [GKN04, MWH21]). We recall that the stress of a graph drawing measures the difference between the geometric distance and the graph-theoretic distance over all pairs of vertices. Thus, based on the well-accepted idea that having low stress makes a straight-line drawing more readable, we define a notion of fairness for straight-line drawings. Clearly, the concept of fair straight-line drawings can be transferred to other aesthetic criteria (e.g., number of crossings), as well as to other graph drawing paradigms (e.g., orthogonal drawings). Therefore, several new problems of both theoretical and practical interest naturally arise from our research and are discussed at the end of the paper.

- We provide empirical results concerning the price of fairness to be paid in terms of additional stress with respect to stress-minimal (but potentially unfair) solutions. Namely, we describe the outcome of an experimental analysis aimed at understanding what is the actual price of fairness on a set of benchmark graphs. To this aim, we implement a gradient-descent based algorithm, following recent ideas in [DAL*19, ADD*20, ALD*22], that can optimize multiple drawing criteria. Our investigation reveals that multi-objective functions that optimize fairness and stress together can output straight-line drawings with good fairness at the expenses of a relatively small increment of stress.

Paper organization. In Section 2, we briefly summarize the main literature related to our research. In Section 3, we formalize our notion of fairness in straight-line drawings based on the stress function. In Section 4, we describe our algorithm and the experiments we performed. Section 5 discusses possible future directions.

2. Related work

The design of visualizations that promote transparency and fairness has stimulated new research within the Visualization community [AL20, CEH*19, DS22, WXC*21], see also the survey by Chatzimparmpas et al. [CMJ*20] about the adoption of visualization to enhance trust in machine learning. Different from our work, such a research stream does not deal with the problem of computing graph visualizations that are fair in terms of readability with respect to different subgroups. Indeed, as it will be formalized in the next section, we aim to balance the readability of a graph visualization when considering two distinct subgroups of vertices (or edges). This might be translated into a layout constraint at the subgroup level. In this direction, the graph drawing literature offers a wide range of theoretical and practical research on constrained layouts. For instance, Bläsius et al. [BLR16] consider orthogonal drawings with constraints on the number of bends per edge, while Kieffer et al. [KDMW13] (among others) incorporate alignment and grid-like constraints in force-directed methods. Notably, the SetCoLa system proposed by Hoffswell et al. [HBH18] incorporates a domain-specific language for describing high-level constraints, such as alignment, position, and order constraints. While these methods can be used to improve the visual quality of the representation of subgroups of vertices or edges in a graph visualization, they do not directly capture the notion of fair drawings proposed in this paper. From a different perspective, the problem of computing low-dimensional representations of high-dimensional data with fairness constraints has been recently studied [PXNN23, TSS*19].

3. Fairness of Straight-line Drawings

A *straight-line drawing* Γ of a graph $G = (V, E)$ maps each vertex of G onto a point of the Euclidean plane, and each edge of G onto a straight-line segment connecting the corresponding endpoints. For a pair of vertices $u, v \in V$, let $\delta(u, v)$ be the length of any shortest path in G between u and v. Also, let $||\Gamma(u) - \Gamma(v)||_2$ be the Eu-

clidean distance of u and v in Γ. Moreover, let $\omega : V \times V \to \mathbb{Q}$ be a weighting function. The *stress* of Γ is defined as follows:

$$\text{stress}(\Gamma) = \sum_{u,v \in V} \omega(u,v)(||\Gamma(u) - \Gamma(v)||_2 - \delta(u,v))^2.$$

For a vertex v in Γ, the *stress of* v is defined as

$$\text{stress}(\Gamma, v) = \sum_{u \in V} \omega(u,v)(||\Gamma(u) - \Gamma(v)||_2 - \delta(u,v))^2.$$

Assume now that the vertex set V of G is the union of two non-empty disjoint subgroups of vertices, that is, $V = V_R \cup V_B$ (with $V_R \neq \emptyset$ and $V_B \neq \emptyset$); vertices in V_R (V_B) are called *red* (*blue*). Thus, let $G = (V_R \cup V_B, E)$ be a graph and let Γ be a straight-line drawing of G. To convey the notion of fairness in Γ, we can refine the concept of stress by either focusing exclusively on the red vertices or on the blue vertices.

$$\text{stress}_R(\Gamma) = \sum_{u \in V_R, v \in V} \omega(u,v)(||\Gamma(u) - \Gamma(v)||_2 - \delta(u,v))^2$$

$$\text{stress}_B(\Gamma) = \sum_{u \in V_B, v \in V} \omega(u,v)(||\Gamma(u) - \Gamma(v)||_2 - \delta(u,v))^2$$

Ideally, Γ should not be unfair to any of the two sets of vertices, that is, the difference between $\text{stress}_R(\Gamma)$ and $\text{stress}_B(\Gamma)$ normalized by their cardinalities should be as close to zero as possible. More formally, below we conveniently define the *unfairness* $\lambda(\Gamma)$ of Γ, whose minimization leads to a fair drawing

$$\lambda(\Gamma) = \left(\frac{\text{stress}_R(\Gamma)}{|V_R|} - \frac{\text{stress}_B(\Gamma)}{|V_B|} \right)^2.$$

Observe that we take the square of the difference (rather than, e.g., the absolute value) to obtain a differentiable function.

4. Algorithm and Experiments

We begin by describing the algorithm we developed to compute drawings by minimizing stress and unfairness. We then describe the experimental setup and the results of our evaluation.

4.1. An algorithmic strategy via Gradient Descent

We developed an algorithm following the ideas in [DAL*19, ADD*20, ALD*22], in which a gradient-descent approach is proposed to compute drawings by optimizing multi-objective quality functions. We recall that the idea of stress minimization via (stochastic) gradient descent have been introduced in [ZPG19], in which however the loss function only takes stress into account. Differently, our implementation contains the novel definition of fairness discussed in this paper and it contains a scheduling strategy specifically tailored for our experiments.

In what follows, a *layout* of a graph G is an assignment of coordinates to all vertices of G, i.e., a matrix of size $n \times 2$, where n is the number of vertices of G and 2 is the number of dimensions of the layout. A layout X of G readily defines a straight-line drawing of G. We define a first loss function equal to the stress function (see Section 3). Next, we iterate the gradient descent steps, in which the gradient of the loss function is computed and the layout of the graph is updated accordingly. More precisely, for a layout X, the loss function L maps X to a real value that quantifies the quality (i.e., in this case, the stress) of the corresponding straight-line drawing. An improved layout X' can be found by following the negative gradient direction, $X' = X - \varepsilon \nabla_X L$, where ε is a small positive constant. We halt the computation after a predefined number of iterations and output a first straight-line drawing Γ_0 with low stress. Subsequently, we adopt as loss function the one for unfairness (see Section 3) and repeat the above procedure. However, we might stop the descent earlier if the stress of the layout reaches a predefined threshold (whose value will be discussed later). By this procedure, we compute a sequence of straight-line drawings $\Gamma_0, \Gamma_1, \ldots, \Gamma_h$ with decreasing values of unfairness, such that the ratio $\frac{\text{stress}(\Gamma_h)}{\text{stress}(\Gamma_0)} \leq T$, for a fixed value T. We remark that, as reported in [ALD*22], the adoption of such a scheduling strategy is often more effective than using a weighted combination of different loss functions that might be in conflict with each other.

We conclude by noting that we do not compute the gradient analytically, as it is computed automatically via automatic differentiation. Our implementation is written in Python and it is based on the PyTorch library; the source code is publicly available at the following URL: `https://t.ly/Hzw7g`

4.2. Experimental setup

We aim to investigate the following questions:

Q1 (PRICE OF FAIRNESS): With respect to a stress-minimum straight-line drawing of a graph, how large is the increase of stress required to compute a straight-line drawing with minimum unfairness for the same graph?

Q2 (UNFAIRNESS OF GLOBALLY OPTIMAL DRAWINGS): How unfair are stress-minimum straight-line drawings of graphs?

Our experimental analysis, described below, is intrinsically limited by the fact that we do not use exact methods to optimize stress and fairness; nevertheless, our experiments shed light on both questions.

We tested our code on an MSI Vector GP76 with a 2.7 GHz Intel Core i7, an Nvidia 3080Ti (Laptop) GPU and 16 GB of RAM. Our benchmark is constructed by selecting 30 graphs from the Sparse Matrix Collection [DH11]. Their sizes in terms of vertices range from about 100 to 6,000, and in terms of edges range from about 500 to 14,000. The dataset is available at the public link given above. The same graph collection has been used in previous papers evaluating stress-based layout algorithms [ZPG19, ALD*22].

For each graph in our dataset, we pre-compute the shortest paths between all pairs of vertices (which is needed to compute the loss function). Then, we initialize the positions of the vertices with 10 different random assignments. For each assignment, we compute the first drawing Γ_0 with minimum stress (see also the previous section). Next, for each graph, we randomly sample vertices with probability in the set $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, and color those vertices red, while the unpicked vertices are blue. To straighten our findings, we added an additional coloring configuration in which we took the 10% of vertices with larger stress according to Γ_0 and

	Γ_0		Γ_1		Γ_2	
p	AVG STRESS	AVG UNFAIR.	AVG STRESS VAR.	AVG UNFAIR. VAR.	AVG STRESS VAR.	AVG UNFAIR. VAR.
0.1	$6.31 \cdot 10^{-2}$	$1.46 \cdot 10^{-9}$	+0.57%	-96.2%	+0.77%	-97.9%
0.2	$6.31 \cdot 10^{-2}$	$6.08 \cdot 10^{-10}$	+0.43%	-94.1%	+0.46%	-94.5%
0.3	$6.31 \cdot 10^{-2}$	$4.77 \cdot 10^{-10}$	+0.34%	-92.9%	+0.38%	-93.9%
0.4	$6.31 \cdot 10^{-2}$	$4.51 \cdot 10^{-10}$	+0.34%	-92.5%	+0.38%	-93.4%
0.5	$6.31 \cdot 10^{-2}$	$3.40 \cdot 10^{-10}$	+0.28%	-92.2%	+0.29%	-92.2%
$\diamond$	$6.31 \cdot 10^{-2}$	$5.82 \cdot 10^{-8}$	+2.47%	-57.9%	+8.11%	-80.4%

Table 1: *Results of the experiments. For each configuration, the table reports the stress and the unfairness of Γ_0, as well as the stress and unfairness variation of Γ_1 and Γ_2. The first five rows refer to the random vertex coloring scenario, the drawings are grouped by the percentage of red vertices (p), and the values are averaged over all drawings in the same group. The last row ($\diamond$) refers to the scenario in which the red vertices are 10% of those with larger stress in Γ_0, and the values are averaged over all drawings in this group.*

colored them red, while the remaining vertices are blue. This choice is a very challenging scenario for our model, as optimizing fairness in this case is likely to require a substantial redrawing of the graph.

At this point, for each graph, for each initial drawing Γ_0, and for each vertex coloring, we apply our algorithm to compute a sequence of two additional straight-line drawings, Γ_1 and Γ_2, such that $\frac{\text{stress}(\Gamma_1)}{\text{stress}(\Gamma_0)} \leq 0.05$ and $\frac{\text{stress}(\Gamma_2)}{\text{stress}(\Gamma_0)} \leq 0.2$. In other words, starting from Γ_0, we optimize fairness ensuring that the stress of the resulting drawing has not increased by more than 5% in Γ_1 and by more than 20% in Γ_2. Drawing Γ_1 is a visualization in which the fairness is best possible constrained to the fact that the stress should remain almost the same as in Γ_0 – the 5% factor only allows for a little bit of freedom to explore solutions around Γ_0 with better fairness. On the other hand, Γ_2 is a drawing in which the fairness is best possible constrained to the fact that the stress should remain within a reasonable factor with respect to Γ_0. Thus, Γ_2 and Γ_1 can be used to investigate questions **Q1** and **Q2**, respectively.

We conclude the discussion of the experimental setup by mentioning that we set the maximum number of iterations of the algorithm to $1.5 \cdot 10^3$, the learning rate ε to 0.01, and we exploited the Adam optimizer available in PyTorch. This tuning of the parameters have been obtained by a preliminary experimental validation, also taking into account the diversity of the graphs.

4.3. Results

We first report the results for the scenario in which the red vertices are randomly sampled with probability in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$; see Table 1. The first notable fact is that the unfairness value of Γ_0 is already rather small, and it goes down to almost zero already in Γ_1. The second fact that strikes from the table is that the fraction of additional stress with respect to Γ_0 in both Γ_1 and Γ_2 is less than 0.8%, i.e., the algorithm often halts after a maximum number of iterations and hence $\Gamma_1 = \Gamma_2$ for many instances.

We next report the results for the scenario in which the vertices are ranked by descending stress in Γ_0 and the red vertices are the first 10% in this ranking; refer to the last row ($\diamond$) of Table 1, and see for instance Figure 1 which shows one pair Γ_0 and Γ_2 computed for this experiment. As expected, this scenario is more challenging and both Γ_1 and Γ_2 often meet their thresholds in terms of additional stress fraction with respect to Γ_0. Also, the unfairness of Γ_0 is relatively larger than in the previous scenario, but it drops down on average by more than 57% in Γ_1 and by more than 80% in Γ_2.

Finally, while computational efficiency is not a goal of our experiment, we report that each computation took on average 75 seconds, ranging from few seconds to, for a couple of outliers, 12 minutes.

4.4. Discussion and limitations

Our experiments suggest that stress-minimum drawings may originally be suboptimal in terms of fairness (especially in our second scenario), but incorporating fairness in the optimization process can effectively lead to notable improvements (**Q2**). Also, if one is willing to pay a small fraction of additional stress in the drawing, unfairness can be further minimized and brought down to almost zero (**Q1**). Therefore, for sensitive applications, our experiments support the introduction of fairness constraints. On the other hand, we only considered medium-size graphs with up to few thousands of elements; we plan to extend our analysis to larger graphs. To this aim, it would be interesting to design more efficient algorithms that make use of stochastic gradient descent. For this purpose, we should sample the batches in a way that respects the ratio between red and blue vertices. Also, it should be noted that even a small fraction of additional stress may cause visible distortions in the drawings of very regular graphs, for example, grid graphs.

5. Conclusions

We have introduced a notion of fairness for straight-line graph drawings, and proposed a gradient-descent based algorithmic strategy to compute drawings by optimizing both stress and fairness. While our definition of fairness captures the need for drawings in which the visual complexity is balanced around red and blue vertices, it is arguably not the only possible one. Indeed, since fairness is certainly an elusive and multifaceted concept that may depend on multiple features of a visualization, it would be of great interest to design human experiments aimed at investigating cognitive aspects of how humans perceive different subgroups (potentially more than two) in a network visualization. Finally, the concept of fairness can be studied for other graph drawing paradigms, such as orthogonal drawings, where one can consider the number of bends along the edges as a natural optimization criterion. In this regard, also challenging theoretical questions concerning the computational complexity of optimizing both fairness and bends can be addressed.

References

[ADD*20] AHMED A. R., DE LUCA F., DEVKOTA S., KOBOUROV S. G., LI M.: Graph drawing via gradient descent, $(gd)^2$. In *GD 2020* (2020), Auber D., Valtr P., (Eds.), vol. 12590 of *Lecture Notes in Computer Science*, Springer, pp. 3–17. doi:10.1007/978-3-030-68766-3_1. 2, 3

[AL20] AHN Y., LIN Y.: Fairsight: Visual analytics for fairness in decision making. *IEEE Trans. Vis. Comput. Graph. 26*, 1 (2020), 1086–1095. doi:10.1109/TVCG.2019.2934262. 2

[ALD*22] AHMED A. R., LUCA F. D., DEVKOTA S., KOBOUROV S. G., LI M.: Multicriteria scalable graph drawing via stochastic gradient descent, $(sgd)^{2}$(sgd)2. *IEEE Trans. Vis. Comput. Graph. 28*, 6 (2022), 2388–2399. URL: https://doi.org/10.1109/TVCG.2022.3155564, doi:10.1109/TVCG.2022.3155564. 1, 2, 3

[BLR16] BLÄSIUS T., LEHMANN S., RUTTER I.: Orthogonal graph drawing with inflexible edges. *Comput. Geom. 55* (2016), 26–40. URL: https://doi.org/10.1016/j.comgeo.2016.03.001, doi:10.1016/j.comgeo.2016.03.001. 2

[CEH*19] CABRERA Á. A., EPPERSON W., HOHMAN F., KAHNG M., MORGENSTERN J., CHAU D. H.: FAIRVIS: visual analytics for discovering intersectional bias in machine learning. In *IEEE VAST 2019* (2019), Chang R., Keim D. A., Maciejewski R., (Eds.), IEEE, pp. 46–56. doi:10.1109/VAST47406.2019.8986948. 2

[CMJ*20] CHATZIMPARMPAS A., MARTINS R. M., JUSUFI I., KUCHER K., ROSSI F., KERREN A.: The state of the art in enhancing trust in machine learning models with the use of visualizations. *Comput. Graph. Forum 39*, 3 (2020), 713–756. doi:10.1111/cgf.14034. 2

[DAL*19] DEVKOTA S., AHMED A. R., LUCA F. D., ISAACS K. E., KOBOUROV S. G.: Stress-plus-x (SPX) graph layout. In *GD 2019* (2019), Archambault D., Tóth C. D., (Eds.), vol. 11904 of *LNCS*, Springer, pp. 291–304. doi:10.1007/978-3-030-35802-0_23. 2, 3

[DH11] DAVIS T. A., HU Y.: The university of florida sparse matrix collection. *ACM Trans. Math. Softw. 38*, 1 (2011), 1:1–1:25. URL: https://doi.org/10.1145/2049662.2049663, doi:10.1145/2049662.2049663. 3

[DMW08] DWYER T., MARRIOTT K., WYBROW M.: Interactive, constraint-based layout of engineering diagrams. *Electron. Commun. Eur. Assoc. Softw. Sci. Technol. 13* (2008). doi:10.14279/tuj.eceasst.13.168. 1

[DS22] DIMARA E., STASKO J. T.: A critical reflection on visualization research: Where do decision making tasks hide? *IEEE Trans. Vis. Comput. Graph. 28*, 1 (2022), 1128–1138. doi:10.1109/TVCG.2021.3114813. 2

[Dwy09] DWYER T.: Scalable, versatile and simple constrained graph layout. *Comput. Graph. Forum 28*, 3 (2009), 991–998. doi:10.1111/j.1467-8659.2009.01449.x. 1

[FKN22] FROESE V., KELLERHALS L., NIEDERMEIER R.: Modification-fair cluster editing. In *AAAI 2022* (2022), AAAI Press, pp. 6631–6638. 1

[GKN04] GANSNER E. R., KOREN Y., NORTH S. C.: Graph drawing by stress majorization. In *GD 2004* (2004), Pach J., (Ed.), vol. 3383 of *Lecture Notes in Computer Science*, Springer, pp. 239–250. doi:10.1007/978-3-540-31843-9_25. 2

[GSV21] GHADIRI M., SAMADI S., VEMPALA S. S.: Socially fair *k*-means clustering. In *FAccT 2021* (2021), Elish M. C., Isaac W., Zemel R. S., (Eds.), ACM, pp. 438–448. doi:10.1145/3442188.3445906. 1

[HBH18] HOFFSWELL J., BORNING A., HEER J.: Setcola: High-level constraints for graph layout. *Comput. Graph. Forum 37*, 3 (2018), 537–548. doi:10.1111/cgf.13440. 1, 2

[KDMW13] KIEFFER S., DWYER T., MARRIOTT K., WYBROW M.: Incremental grid-like layout using soft and hard constraints. In *GD 2013* (2013), Wismath S. K., Wolff A., (Eds.), vol. 8242 of *LNCS*, Springer, pp. 448–459. doi:10.1007/978-3-319-03841-4_39. 2

[KSAM19] KLEINDESSNER M., SAMADI S., AWASTHI P., MORGENSTERN J.: Guarantees for spectral clustering with fairness constraints. In *ICML 2019* (2019), Chaudhuri K., Salakhutdinov R., (Eds.), vol. 97 of *Proceedings of Machine Learning Research*, PMLR, pp. 3458–3467. 1

[MMS*22] MEHRABI N., MORSTATTER F., SAXENA N., LERMAN K., GALSTYAN A.: A survey on bias and fairness in machine learning. *ACM Comput. Surv. 54*, 6 (2022), 115:1–115:35. doi:10.1145/3457607. 1

[MWH21] MEIDIANA A., WOOD J., HONG S.: Sublinear-time algorithms for stress minimization in graph drawing. In *14th IEEE Pacific Visualization Symposium, PacificVis 2021, Tianjin, China, April 19-21, 2021* (2021), IEEE, pp. 166–175. doi:10.1109/PACIFICVIS52677.2021.00030. 2

[PXNN23] PELTONEN J., XU W., NUMMENMAA T., NUMMENMAA J.: Fair neighbor embedding. In *ICML 2023* (2023), Krause A., Brunskill E., Cho K., Engelhardt B., Sabato S., Scarlett J., (Eds.), vol. 202 of *Proceedings of Machine Learning Research*, PMLR, pp. 27564–27584. URL: https://proceedings.mlr.press/v202/peltonen23a.html. 2

[STM*18] SAMADI S., TANTIPONGPIPAT U. T., MORGENSTERN J., SINGH M., VEMPALA S. S.: The price of fair PCA: one extra dimension. In *NeurIPS 2018* (2018), Bengio S., Wallach H. M., Larochelle H., Grauman K., Cesa-Bianchi N., Garnett R., (Eds.), pp. 10999–11010. 1

[TSS*19] TANTIPONGPIPAT U., SAMADI S., SINGH M., MORGENSTERN J., VEMPALA S. S.: Multi-criteria dimensionality reduction with applications to fairness. In *NeurIPS 2019* (2019), Wallach H. M., Larochelle H., Beygelzimer A., d'Alché-Buc F., Fox E. B., Garnett R., (Eds.), pp. 15135–15145. URL: https://proceedings.neurips.cc/paper/2019/hash/2201611d7a08ffda97e3e8c6b667a1bc-Abstract.html. 1, 2

[WXC*21] WANG Q., XU Z., CHEN Z., WANG Y., LIU S., QU H.: Visual analysis of discrimination in machine learning. *IEEE Trans. Vis. Comput. Graph. 27*, 2 (2021), 1470–1480. doi:10.1109/TVCG.2020.3030471. 2

[ZPG19] ZHENG J. X., PAWAR S., GOODMAN D. F. M.: Graph drawing by stochastic gradient descent. *IEEE Trans. Vis. Comput. Graph. 25*, 9 (2019), 2738–2748. doi:10.1109/TVCG.2018.2859997. 3