OPTIMA 106

Issue 106 September 2023

Chair's Column

August 31, 2023. As we are in the second half of 2023, the Mathematical Optimization Society (MOS) is looking forward to returning to our traditional triennial in-person International Mathematical Programming Symposium (ISMP) to be held in Montréal, Québec, Canada, July 21–26, 2024. While we were hopeful to have been able to assemble in person in Beijing in August 2022, travel restrictions at the time made such an in-person gathering in Beijing infeasible, leading to the online collection of superb plenary speakers who filled us in on the latest discoveries in our constantly evolving field last year. I thank all the speakers for providing such a rich and dynamic set of talks, Katya Scheinberg, chair of the Program Committee, and the other Program Committee members for putting together such an excellent lineup, and all of you for attending and helping to continue the ISMP tradition of celebrating our field.

At the conclusion of the ISMP in 2022, we welcomed new MOS Council Members, Fatma Kılınç Karzan, Andreas Wächter, Angelike Wiegele, and Wolfram Wiesemann, as well as Chair-elect, Miguel Anjos, and Treasurer-elect, Sam Burer. Next month, Miguel will assume the role of Chair (as I transition to Vice-Chair) and Sam will take on the role of Treasurer from Marina Epelman in ensuring the good financial health of the Society despite the challenging times that we have faced over the past years.

In addition to last year's on-line version of the ISMP, we have had multiple MOS-organized meetings in person, such as ICCOPT 2022 at Lehigh University in July 2022, IPCO 2023 at University of Wisconsin-Madison in July 2023, and ICSP 2023 at University of California, Davis, in July 2023. We also look forward to IPCO 2024 to be held in Wroclaw, Poland, in early July 2024, and ICCOPT 2025 to be held in July 2025. The next ISCP site will be announced shortly as well.

MOS's publications have continued to showcase the most important research on mathematical optimization under the direction of editors-in-chief, Daniel Kuhn, for *Mathematical Programming,* and Jonathan Eckstein for *Mathematical Programming Computation* (MPC). Of special note, is that, while MPC has only a short history of being included in the Journal Impact Factor reports from Clarivate Analytics, it received the highest such rating out of 267 journals classified as Applied Mathematics. That is great testimony to the excellence that you and MOS have fostered for our field.

Through these meetings and publications, MOS strives to serve you as you work to extend the reach and influence of our field. I thank you for this and your support of all that is mathematical optimization and look forward to seeing you at the ISMP in Montréal next August.

John R. Birge, The University of Chicago
john.birge@chicagobooth.edu

Note from the Editors

Algorithms shape almost all aspects of modern life – e.g., search, social media, e-commerce, supply chains, power system operations, and urban transportation, to name a few. They process data generated by complex socio-economic systems and find decisions largely driven by speed and efficiency. However, in this pursuit, there is an increasingly disparate impact and frequent unintended consequences of algorithmic decision-making on heterogenous populations. In this issue, we present you with two articles: in the first article, Mohit Singh and Santosh Vempala highlight how group-fairness may be achieved in clustering and PCA, by solving challenging optimization problems. The second article is contributed by our co-editor Swati Gupta, and collaborators Vijay Kamble and Jad Salem, on the computational challenges that arise in online optimization due to ensuring fairness at the time at which these decisions were taken. We sincerely hope that you will enjoy reading these articles, and welcome suggestions and ideas for topics for the future.

Sebastian Pokutta, Editor
Swati Gupta, Co-Editor
Omid Nohadani, Co-Editor

Group Fairness in Optimization and Clustering

Mohit Singh and Santosh S. Vempala

Fairness in decision making, including allocation of resources, planning and prediction, is of critical importance. In common situations where optimizing a single objective function is the goal of an algorithm, the outcome can be tangibly unfair for subsets of the data set. Taking this into consideration leads to a new generation of optimization problems, whose study has already offered rich, new perspectives and novel techniques. We survey results in this burgeoning area, focusing on their structural insights.

1 Introduction

Optimization is the basis of modern decision making. The choice of model, objective and constraints varies according to the domain and application. With steady progress over many decades, algorithms for optimization are able to handle very large real-world data sets efficiently in many cases. Optimization is also the backbone of machine learning algorithms, where a typical approach is to optimize a *loss function* over a given set of labeled data, in order to later make predictions based on the "learned" (i.e., optimized) model. The widespread use of these methods, increasingly in situations that have direct impact on society and humans, has led to important considerations beyond the optimization objective. The most important of these is *fairness*. When the choice of the optimization algorithm affects a population non-uniformly, could it be tangibly unfair to some individuals or subgroups? Unfortunately, this turns out to be the situation across many application areas, as demonstrated multiple times in recent years, from image classification/recognition to bank loans and mortgages to facility allocation. Thus, the following is an important and time-critical goal: *To understand the inherent (un)fairness of existing algorithms for optimization, and to design algorithms that are fair, and whose fairness is analyzed along with their efficiency and optimality.*

Real-world machine learning and optimization algorithms have been shown to produce unfair outcomes in many domains. For example, Google Photos labeled African Americans as gorillas [32,35], image queries for CEOs returned overwhelmingly male and white images [23], searches for African American names caused the display of arrest record advertisements with higher frequency than searches for white names [33], facial recognition has starkly different accuracy for white men compared to dark-skinned women [10], and recidivism prediction software has labeled low-risk African Americans as high-risk at significantly higher rates than low-risk white people [3]. These results can be unfair to individuals (e.g., in loan or mortgage decision) or to subgroups, e.g., racial or ethnic groups.

While a single global objective function is the typical choice in optimization and clustering algorithms, the outcome can be unfair to individuals or subgroups. There are many possible ways to model and address such unfairness. We mention the most general ones. The first in individual fairness, where in the maximum cost (in a minimization problem) to any single individual (i.e., data point) should be minimized; the second, which we focus on here is *group fairness*, where there exist subgroups (or more generally weightings of the data) and the goal is to minimize the maximum cost to any single subgroup; and the last is to enforce fairness as a constraint in a clustering problem, by requiring that each cluster satisfy prescribed proportionality constraints. In this survey we focus on group fairness. We note that while the maximum cost over all groups (for a minimization problem) could be a good measure of fairness, many other functions of the cost or value to each group are possible and the methods we discuss will in fact apply to more general fairness evaluations.

We note up front that if the problem being addressed is a convex optimization problem in the form min $f(x)$, $x \in K$ for some convex set K, then considering the maximum of m objectives $f_i(x)$ remains a convex problem. However, if the original (single-objective) problem happens to have structure that implies an integral solution, this might not be preserved in the multi-objective version. Thus, in general, the complexity of fair combinatorial optimization can vary from problem to problem, even if the single-objective version is polynomial time solvable. This will become apparent in forthcoming examples.

1.1 Fair Dimensionality Reduction and Fair PCA

Dimensionality reduction is a classical technique widely used for data analysis. It is a core primitive for modern machine learning and is being used in image processing, biomedical research, time series analysis, etc. Among the most widely used dimensionality reduction method is the Principal Component Analysis (PCA). The attractive nature of PCA relies from efficient computation as well as the guarantee that it minimizes the average reconstruction error over the whole data set. Unfortunately, when we consider data based on subpopulations partitioned by sensitive attributes, such

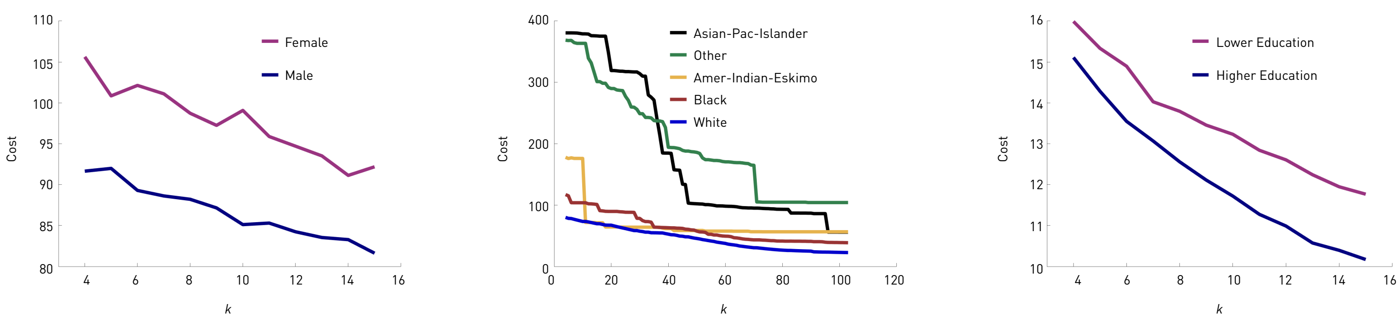
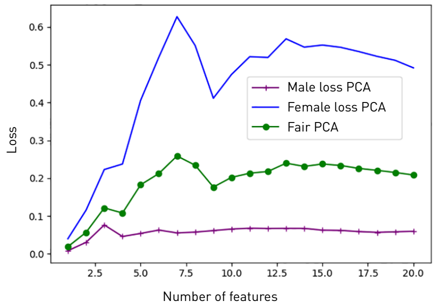


Figure 1. The standard Lloyd's algorithm results in a significant gap in the average clustering costs of different subgroups of the data. The data sets used are the Adult census data set (for gender and race) and the Credit data set (for education level).



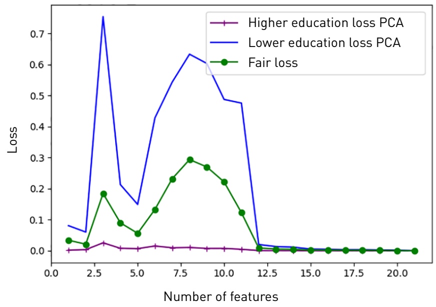


Figure 2. Loss of PCA and Fair PCA on the LFW (left) and Default Credit (right) data sets

as gender, race, and education level, similar guarantee do not hold (see Figure 2).

We begin by describing a framework for accounting for group fairness in dimensionality reduction problems generally. For simplicity of explanation, we first describe our framework for PCA. Consider the data points as rows of an $m \times n$ matrix A. For PCA, the objective is to find an $n \times d$ projection matrix P that maximizes the Frobenius norm $\|AP\|_F^2$ (this is equivalent to minimizing the reconstruction error $\|A - APP^T\|_F^2$). Suppose that the rows of A belong to different *groups* based on demographics or some other semantically meaningful clustering. One way to balance multiple objectives is to find a projection P that maximizes the minimum objective value over each of the groups (weightings) as defined in Samadi et al. [31]:

$$\max_{P \in \mathbb{R}^{n\times d}: P^\top P = I_d} \min_{1\le i\le k} \|A_i P\|_F^2 = \langle A_i^T A_i, PP^T \rangle. \qquad \text{(Fair-PCA)}$$

More generally, let $\mathcal{P}_d$ denote the set of all $n \times d$ projection matrices P, i.e., matrices with d orthonormal columns. For each group A_i, we associate a function $f_i : \mathcal{P}_d \to \mathbb{R}$ that denotes the group's objective value for a particular projection. We are also given an accumulation function $g : \mathbb{R}^k \to \mathbb{R}$. We define the (f, g)-multi-criteria dimensionality reduction problem as finding a d-dimensional projection P which optimizes

$$\max_{P \in \mathcal{P}_d} g(f_1(P), f_2(P), \ldots, f_k(P)). \qquad \text{(Multi-Criteria-Dimension-Reduction)}$$

While these definitions give a natural way to extend dimensionality reduction that optimizes for the whole data set to incorporating the utilities for each of the subgroups. Moreover the generality in picking functions f, g allows one to balance the utilities of various subgroups in rich ways. The main computational challenge of understanding the complexity of the problems remains that we now discuss.

The first result states the conditions when these problems can be solved exactly in polynomial time.

Theorem 1.1 ([34]). *There is a polynomial time algorithm that solves the* Fair-PCA*problem with 2 groups. More generally, there is a polynomial-time algorithm for 2-group* Multi-Criteria-Dimension-Reduction *problem when g is concave and monotone non-decreasing for at least one of its two arguments and each f_i is linear in PP^T, i.e., $f_i(P) = \langle B_i, PP^T \rangle$ for some matrix $B_i(A)$.*

These results rely on exactness of the SDP relaxation for the corresponding problem building on results of Barvinok [4] and Pataki [29] on existence of low rank SDP solutions. For more general groups, the following result shows a bi-criteria approximation algorithm.

Theorem 1.2 ([34]). *There is a polynomial time algorithm that given an instance of* Multi-Criteria-Dimension-Reduction *problem with a concave g that is monotone non-decreasing in at least one of its arguments and f_i that is linear in PP^T for each i, returns a $d + \left\lfloor \sqrt{2k + \frac{1}{4}} - \frac{3}{2} \right\rfloor$-dimensional embedding whose objective value is at least that of the optimal d-dimensional embedding.*

1.2 Clustering Problems

Clustering, or partitioning data into dissimilar groups of similar items, is a core technique for data analysis and used widely in various application including genetics, image segmentation, grouping search results and news aggregation, crime-hot-spot detection, crime pattern analysis, profiling road accident hot spots, and market segmentation. In many human-centric applications, the output of the clustering algorithms can easily lead to harmful results unless such affects are accounted for. For example, consider the results of widely used Lloyd's algorithm for k-means clustering as shown in Figure 1 which shows that different demographics pay different cost in the output clustering. Due to its wide-applicability and use, various notions fair clustering have been defined in literature[1, 2, 5, 6, 11, 12, 14, 20, 22, 24, 25, 27].

Group fair clustering introduced by [1, 14] offers a model to account for such discrepancies across demographic groups. Formally, consider any clustering problem such as k-means, k-median, k-center where the goal is to cluster the given set of points. In an instance, we are given a set of points $U \subseteq \mathbb{R}^n$ and goal is to find k centers $C = \{c_1, \ldots c_k\} \subset \mathbb{R}^n$ that minimizes the total ℓ_p distance of the points to the nearest center. For example, in the k-means problem where $p = 2$, the goal is to optimize $\sqrt{\sum_{u\in U} \min_{i=1}^k \|c_i - u\|^2}$.

In the group fair variant of the clustering problem, the points to be clustered belong to ℓ distinct groups $G_1, \ldots, G_r$, the goal is again to find k centers $c_1, \ldots, c_k \in \mathbb{R}^n$ where the cost paid by any group G_i for k-means clustering is exactly, $\Delta_i := \sqrt{\sum_{u\in G_i} \min_{i=1}^n \|c_i - u\|^2}$. More generally, for any $p \ge 1$, the cost of the group is defined by ℓ_p norm of the distances of the points in the group to their nearest center. The objective of the group fair k-means problem is then defined as $f(\Delta_1, \ldots, \Delta_r)$ where $f : \mathbb{R}^r \to \mathbb{R}$ is a function that tradeoffs the cost for each of the social group. Natural choices considered are $f(x) = \max x_i$ which leads to min-max fair k-means or $f(x) = \|x\|_q$

for some $q \geq 1$ where $q = \infty$ reduces to the min-max fair k-means. Let us denote (p, q)-fair clustering problem[13] defined with parameters p and q in the definitions above.

We state the following two results regarding the (p, q)-fair clustering problem.

Theorem 1.3 ([13, 28]). *For any $p, q \geq 1$, there exists an $O(k^{\frac{p-q}{2pq}})$-approximation algorithm and when $q = \infty$, there exists an $(\frac{\log n}{\log \log n})^{\frac{1}{p}}$-approximation algorithm for the (p, q)-fair clustering problem.*

More generally, these results are nearly tight based on hardness results. For the case when the number of groups r is constant one can achieve improved algorithms (see also [17] for results for algorithm that run in time exponential in k and r).

Theorem 1.4 ([15]). *For any $p, q \geq 1$, there exists $O(1)$-approximation algorithm (p, q)-fair clustering problem that runs in time $O(n^r)$ where n is the size of the instance and r is the number of groups.*

We close this section by discussing a fair version of one of the most popular algorithms for clustering, namely Lloyd's heuristic for k-means. This algorithm starts with some set of candidate centers, uses them to partition the data acccording to which center is closest to each point, and then recomputes the center of each cluster as simply the mean of the cluster. This iterative procedure is efficient to implement and widely used in practice, even though it does not have strong worst-case guarantees. [14] considers the fair version of this problem, where the data has two (or more) subgroups. Let the k-means cost of a set of points U with respect to a set of centers $C = \{c_1, \ldots, c_k\}$ and a partition $\mathcal{U} = \{U_1, \ldots, U_k\}$ of U be

$$\Delta(C, \mathcal{U}) := \sum_{i=1}^{k} \sum_{p \in U_i} \|p - c_i\|^2.$$

Then the fair k-means objective for two groups A, B such that $U = A \cup B$ is the larger average cost:

$$\Phi(C, \mathcal{U}) := \max\left\{\frac{\Delta(C, \mathcal{U} \cap A)}{|A|}, \frac{\Delta(C, \mathcal{U} \cap B)}{|B|}\right\},$$

where $\mathcal{U} \cap A = \{U_1 \cap A, \ldots, U_k \cap A\}$. The heuristic proposed, *Fair* Lloyd, is very similar to Lloyds, with the only change being that in place of the center of a cluster being its mean, the algorithm uses a *fair* center, that minimizes the above for a single cluster. As shown in [14], the fair center for each cluster lies in the convex hulls of the means of the subgroups, and is the solution of a quadratic convex program. In the case of two groups, the fair center can be computed very quickly via a line search.

1.3 Network Design Problems

We now consider network design problems via the lens of group fairness constraints. Consider the general survivable network design problem where we are given a graph $G = (V, E)$ and pair-wise requirements r_{ij} for each pair of vertices i and j. The goal is to find to a subgraph $H = (V, F)$ of G that has at least r_{ij}-edge disjoint paths between every pair of nodes i, j. The objective is to typically minimize the total cost of the edges chosen where we are given one cost function $c : E \to \mathbb{R}_+$. The above problem generalizes many classical network design problems such as minimum spanning tree, minimum Steiner tree, minimum Steiner forest etc. A seminal result by Jain [21] gives a 2-approximation for the general problem. In the group fair survivable network design problem, we aim to find a single network where multiple players have different or competing valuations and must agree on a common network (e.g., infrastructure design). Formally, we are given multiple cost functions $c_i : E \to \mathbb{R}_+$ for $1 \leq i \leq k$, one for each of the subgroups. Our goal is still to select a single subgraph (V, F) where $F \subseteq E$ that satisfies the connectivity requirements. The objective that we aim to optimize is appropriate function of the different costs paid by the groups such as minimizing the maximum cost over the k groups $\min \max_{i \in \{1,\ldots,k\}} c_i(F)$.

Classically, these problems have been studied for the special cases of spanning trees [18, 30] and other graph structures [18] as multi-criteria network design problems. We state the following two results: for the group fair spanning tree problem as well as the group fair survivable network design problem. The first results shows that for the spanning tree problem, there is an polynomial time approximation scheme (PTAS) when the number of groups is constant.

Theorem 1.5 ([18, 30]). *For any fixed $\epsilon, k > 0$, there is a polynomial time algorithm for the group fair spanning tree problem with k groups that returns a $(1 + \epsilon)$-approximation and runs in time $O(n^{\frac{k^2}{\epsilon}})$.*

For the fair survivable network design problem with the min-max objective.

Theorem 1.6 ([26]). *For any $k \geq 2$, there is a polynomial time k-approximation algorithm for the group fair survivable network design problem with k groups.*

We remark that for $k = 1$, the problem is the classical survivable network design problem where Jain [21] gives a 2-approximation. Interestingly, the above result shows that we obtain a 2-approximation when considering the fair version with two groups.

2 Open Problems and Future Directions

We conclude with a discussion of some open problems. For the group fair-PCA problem, the problem is polynomial time solvable for $k = 2$ groups and has an approximation algorithm for larger k. While the problem is NP-hard for general k, it is open whether the problem is polynomial-time solvable for fixed k. We do remark that the problem is polynomial time solvable when both d and k are fixed [34], via representation results that use solutions of quadratic maps [19].

For the fair survivable network design problem, there is a k-approximation for k groups. The integrality gap of the linear programming relaxation is k [26] which implies that new ideas are needed to obtain any improvement. A natural question is whether there is a $o(k)$-approximation as is shown for the case of spanning trees.

The classical Max-cut problem has also been studied in this vein: we have multiple graphs $G_1, G_2, \ldots, G_k$ over the same vertex set. The goal is find a single cut that maximizes the minimum value of the cut in any of the graph. This problem has been called the simultaneous max-cut problem [9]. The problem generalizes the max-cut problem and a natural question is how close is the approximability to the Goemans–Williamson bound $\alpha_{GW} = 0.87856\ldots$

for the standard setting of a single group [16]. The current best approximation is 0.8780 for any fixed k [8] while [7] show that best approximation algorithm cannot achieve a $\alpha_{GW} - 10^{-5}$-approximation assuming the Unique Games Conjecture. Identifying the precise approximation threshold is still an open problem.

For dimensionality reduction and the clustering problem, we considered a wide variety of objectives to balance the competing objectives of different groups while considering the fair variants. A similar detailed analysis for other problems is a potentially fruitful future research direction. More generally, it would be interesting to obtain a characterization of functions that can be efficiently used to combine the utility/cost of each of the groups.

Mohit Singh, Algorithms and Randomness Center
Georgia Institute of Technology, Atlanta, GA 30332, USA
mohit.singh@isye.gatech.edu

Santosh S. Vempala, College of Computing
Georgia Institute of Technology, Atlanta, GA 30332, USA
vempala@gatech.edu

Bibliography

[1] M. Abbasi, A. Bhaskara, and S. Venkatasubramanian. Fair clustering via equitable group representations. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 504–514, 2021.

[2] S. Ahmadian, A. Epasto, R. Kumar, and M. Mahdian. Clustering without over-representation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 267–275, 2019.

[3] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias – propublica. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing, 2018.

[4] A. I. Barvinok. Problems of distance geometry and convex properties of quadratic maps. *Discrete & Computational Geometry*, 13(2):189–202, 1995.

[5] S. Bera, D. Chakrabarty, N. Flores, and M. Negahbani. Fair algorithms for clustering. *Advances in Neural Information Processing Systems*, 32, 2019.

[6] I. O. Bercea, M. Groß, S. Khuller, A. Kumar, C. Rösner, D. R. Schmidt, and M. Schmidt. On the cost of essentially fair clusterings, 2018. arXiv:1811.10319.

[7] A. Bhangale and S. Khot. Simultaneous max-cut is harder to approximate than max-cut. In *35th Computational Complexity Conference (CCC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[8] A. Bhangale, S. Khot, S. Kopparty, S. Sachdeva, and D. Thiruvenkatachari. Near-optimal approximation algorithm for simultaneous max-cut. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1407–1425. SIAM, 2018.

[9] A. Bhangale, S. Kopparty, and S. Sachdeva. Simultaneous approximation of constraint satisfaction problems. In *International Colloquium on Automata, Languages, and Programming*, pages 193–205. Springer, 2015.

[10] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, pages 77–91, 2018.

[11] X. Chen, B. Fain, L. Lyu, and K. Munagala. Proportionally fair clustering. In *International Conference on Machine Learning*, pages 1032–1041. PMLR, 2019.

[12] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems*, pages 5029–5037, 2017.

[13] E. Chlamtác̆, Y. Makarychev, and A. Vakilian. Approximating fair clustering with cascaded norm objectives. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2664–2683. SIAM, 2022.

[14] M. Ghadiri, S. Samadi, and S. Vempala. Socially fair k-means clustering. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 438–448, 2021.

[15] M. Ghadiri, M. Singh, and S. Vempala. Constant-factor approximation algorithms for socially fair k-clustering, 2022. arXiv:2206.11210.

[16] M. X. Goemans, A. V. Goldberg, S. Plotkin, D. B. Shmoys, E. Tardos, and D. P. Williamson. Improved approximation algorithms for network design problems. Technical report, Cornell University Operations Research and Industrial Engineering, 1995.

[17] D. Goyal and R. Jaiswal. Tight FPT approximation for socially fair clustering, 2021. arXiv:2106.06755.

[18] F. Grandoni, R. Ravi, M. Singh, and R. Zenklusen. New approaches to multi-objective optimization. *Mathematical Programming*, 146(1-2):525–554, 2014.

[19] D. Grigoriev and D. V. Pasechnik. Polynomial-time computing over quadratic maps i: sampling in real algebraic sets. *Computational complexity*, 14(1):20–52, 2005.

[20] L. Huang, S. Jiang, and N. Vishnoi. Coresets for clustering with fairness constraints. *Advances in Neural Information Processing Systems*, 32, 2019.

[21] K. Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[22] C. Jung, S. Kannan, and N. Lutz. A center in your neighborhood: Fairness in facility location, 2019. arXiv:1908.09041.

[23] M. Kay, C. Matuszek, and S. A. Munson. Unequal representation and gender stereotypes in image search results for occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3819–3828. ACM, 2015.

[24] M. Kleindessner, P. Awasthi, and J. Morgenstern. Fair k-center clustering for data summarization. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[25] M. Kleindessner, P. Awasthi, and J. Morgenstern. A notion of individual fairness for clustering, 2020. arXiv:2006.04960.

[26] A. Laddha, M. Singh, and S. S. Vempala. Fair network design via iterative rounding. *Arxiv*, 2022.

[27] S. Mahabadi and A. Vakilian. Individual fairness for k-clustering. In *International Conference on Machine Learning*, pages 6586–6596. PMLR, 2020.

[28] Y. Makarychev and A. Vakilian. Approximation algorithms for socially fair clustering. In *Conference on Learning Theory*, pages 3246–3264. PMLR, 2021.

[29] G. Pataki. On the rank of extreme matrices in semi-definite programs and the multiplicity of optimal eigenvalues. *Mathematics of operations research*, 23(2):339–358, 1998.

[30] R. Ravi and M. X. Goemans. The constrained minimum spanning tree problem. In *Scandinavian Workshop on Algorithm Theory*, pages 66–75. Springer, 1996.

[31] S. Samadi, U. Tantipongpipat, J. H. Morgenstern, M. Singh, and S. Vempala. The price of fair pca: One extra dimension. In *Advances in Neural Information Processing Systems*, pages 10976–10987, 2018.

[32] T. Simonite. When it comes to gorillas, google photos remains blind. https://www.wired.com/story/when-it-comes-to-gorillas-google-photos-remains-blind/, Jan 2018.

[33] L. Sweeney. Discrimination in online ad delivery. *Communications of the ACM*, 56(5):44–54, 2013.

[34] U. Tantipongpipat, S. Samadi, M. Singh, J. H. Morgenstern, and S. S. Vempala. Multi-criteria dimensionality reduction with applications to fairness. *Advances in neural information processing systems*, 32, 2019.

[35] Twitter. Jacky lives: Google photos, y'all fucked up. My friend's not a gorilla. https://twitter.com/jackyalcine/status/615329515909156865, June 2015.

Comment by Sam Burer

Algorithmic fairness is a critical issue facing the closely aligned fields of optimization and machine learning. Indeed, society's acceptance of computer-generated decisions requires that those decisions treat different subgroups of people – whether organized, for example, by race, gender, or age – in a fair manner.

Of course, this naturally leads to many important research questions. What does "fair" mean exactly? How should existing models and algorithms be adjusted to ensure fairness? Or do we need wholly new ideas? And what is the computational complexity of ensuring fairness?

In this timely survey, Singh and Vempala outline an important stream of research, especially active in the past five years, which is helping our community address these questions. The authors introduce the exension of several important algorithms – for dimension reduction, clustering, and network design – to incorporate fairness with respect to subgroups, and they discuss known polynomial-time and approximation results. They also highlight for the reader some of the optimization techniques involved, including SDP, convex QP, and graph approximation algorithms.

Sam Burer, Tippie College of Business
The University of Iowa, Iowa City, IA 52242-1994, USA
samuel-burer@uiowa.edu

Comment by David B. Shmoys

The advent of machine learning as a central tool in modern data-driven decision-making has opened up new avenues for the application of optimization models and algorithms. The authors present a number of ways in which issues of fairness have been exposed by these recent developments, starting with questions that are directly linked to some of the most salient domains – dimension reduction (or feature identification) and classification procedures based on notions of clustering. Fairness issues are by no means new to the optimization community – one need only think of an example such as bandwidth allocation in routing to see that there is a long tradition of augmenting optimization models with fairness considerations. What is profoundly new is the extent to which algorithmic decision-making has a day-to-day impact on all aspects of society. The authors focus primarily on group fairness, which is indeed an extremely important issue. One significant implication of the article is to show the extent to which the examples addressed are just starting points – the mainstay of the optimization community is the efficient allocation of scarce resources, and hence, there is the accompanying question of fairness in the resulting allocation, in a host of application domains. Furthermore, the article points to the rich vein of research to be explored to investigate a breadth of notions of fairness as they might apply to a wide variety of decision-making settings.

David B. Shmoys, Operations Research and Information Engineering
Cornell University, Ithaca, NY 14853, USA
david.shmoys@cornell.edu

Algorithmic Challenges in Ensuring Temporal Fairness in Online Decision-Making

Swati Gupta, Vijay Kamble and Jad Salem

1 Introduction

In recent years, a wide range of organizations has deployed sophisticated data-driven algorithms that repeatedly output decisions having a serious impact on human lives. For example, a recommendation engine shows job advertisements to a stream of arriving customers to maximize their click-through-rates and platform engagement [3]; a retail pricing algorithm repeatedly offers a good at different prices over time to arriving customers to learn the most profitable price [25]; a resume screening algorithm screens out unqualified candidates to provide the most lucrative subset to hiring managers [30]. This widespread pursuit of efficiency and optimality has largely ignored a long-standing issue with decision-making with partial information: the perceptions of unfairness that may arise as a result of experimenting with decisions for the purpose of information-gathering in an online optimization and learning framework. The goal of this article is to highlight the rich interplay between fairness desiderata and temporal decision-making that results in new and exciting research challenges in online and iterative optimization, following a recent stream of works [8,17,20,22–24,28].

As a running example throughout this article, consider a demand learning algorithm that may experiment with different prices over time to determine the optimal price for a good. While such experimentation is necessary to learn the optimal price, arbitrarily changing prices may create a sense of unfairness amongst customers, e.g., a customer may receive a price much higher than a previous customer who arrived before her for no apparent reason. Indeed, there has been extensive research in behavioral sciences investigating consumer perception of price fairness, which has concluded that notions of price fairness essentially stem from comparison: without explicit explanations, customers think they are similar to other customers buying the same item, and thus should pay equal prices [5, 11, 18, 21, 32]. In the above scenario, there is likely no palatable explanation that the firm can provide to justify the temporal disparities that arise in the search for the profit-maximizing price.

Such issues with increased experimentation are further complicated by the growing litigation and policymaking surrounding algorithmic decisions; e.g., Amazon was recently sued for allegedly price-gouging during the pandemic, as they increased the price of essential goods by more than 450 % (e.g., see Figure 1) compared to previously seen prices (*McQueen and Ballinger v. Amazon.com*, Inc., 422 U.S. Case 4:20-cv-02782 (2020).). Such misalignment of business practices geared towards profitability, and evolving consumer protection laws (e.g., [6]) raises a natural question for various operational tasks: *can firms enforce appropriate fairness desiderata in their online decision-making systems while still achieving good operational performance?* These questions about attaining "good" performance while adhering to certain constraints enforced due to fairness generate a class of novel and interesting optimization problems.

We will illustrate these challenges by mainly focusing on a recently proposed temporal fairness objective of ensuring *fairness at the time of decision* (FTD) [17, 28], to mitigate perceptions of unfairness when an entity receives a decision. In particular, FTD requires that when a person A receives a decision that affects them at time t, comparing this decision to the available data on decisions received by other entities at or before time t must not reveal any problematic disparity in the treatment of entity A. This requirement amounts to a basic prerogative from the perspective of any decision-maker who is interested in guaranteeing immunity to legal claims of discrimination at the time of the decision [17]. The notion of "problematic disparity" can be modeled by any task-specific notion of fairness

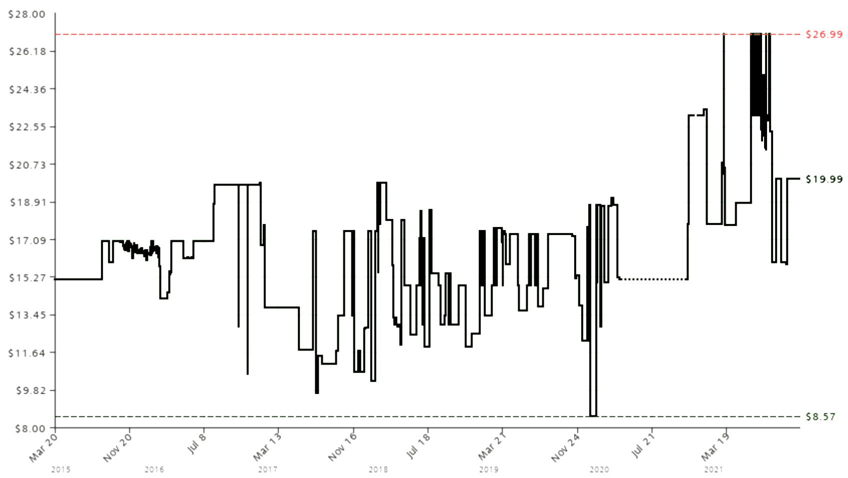


Figure 1. The price path of PURELL Advanced Hand Sanitizer Soothing Gel, Fresh cent, with Aloe and Vitamin E, 8 Fl Oz Pump Bottle (Pack of 4) on Amazon from March 2015 to March 2021 (from camelcamelcamel.com/product/B00U2KYUAY)

for static decision-making, generally requiring that each entity gets a decision that is (approximately) as conducive as that received by others. Thus, FTD adapts to the growing literature on task-specific variants of comparative fairness across a variety of operational applications such as demand learning [10], assortment optimization [8], kidney exchange [13], etc. Importantly, since fairness is not required to be guaranteed from the perspective of an entity *after* the entity receives the decision, FTD generally allows decisions to become more conducive over time, thus allowing room for (constrained) experimentation and learning.

We will first introduce a general setup to motivate and define the notion of fairness at the time of decision. We will then illustrate the new challenges it introduces in a specific context of dynamic pricing. We finally discuss related problems and extensions that contribute to this growing interface between online optimization and fairness.

2 Fairness in Static Decision-Making

Consider first static decision-making scenario, where there is a finite set $\mathcal{J} = \{1, \cdots, n\}$ of entities representing the stakeholders receiving scalar decisions. For example, an entity could represent a customer segment (such as "youth") in a pricing scenario or a job applicant in an applicant-screening scenario. The decisions over the set of entities lie in the set $\mathcal{X} \subseteq \mathbb{R}^n$ (e.g., the range of prices that can be offered to a customer, or accept/reject decisions for screening). The decision-maker needs to select $x \in \mathcal{X}$, which minimizes some cost function $f(x)$ representing some operational performance measure such as negative social welfare [9] or net cost of operations [2]. If one wants to enforce some notion of *comparative* fairness, then the space of feasible decisions $\mathcal{X}$ can be intersected with fairness-induced constraints $\mathcal{F}$. Let's consider two examples:

1. *Individual Fairness.* Dwork et al. [12] proposed in 2012 a general notion of comparative fairness, called *individual fairness*, where they asked that each entity $i \in [n]$ represented by a context $c_i \in \mathbb{R}^d$ receive decisions that are "similar" to the decisions received by other "similar" entities. This can be modeled using a Lipchitz constraint using given contexts for the entities:
$$|x(i) - x(j)| \leq L|c(i) - c(j)|, \forall i, j \in [n]. \tag{1}$$
These constraints can be easily separated over and also linearized, since the number of contexts is fixed to be n. There has been a stream of papers in the fair ML community since then to understand the construction of the contexts, how such constrained might be separated over using auditors [4, 15].
2. *Fair pricing.* Consider a product pricing scenario with n customer segments. Let $x(i)$ be the price offered to segment i. To ensure comparative fairness from the perspective of segment i, we may require certain upper bounds on the price offered to i depending on the prices offered to the other segments. For instance, we may require that $x(i) \leq x(i') + s(i, i')$, where $s(i, i') \in \mathbb{R}$ represents a permissible additive slack that depends on the two segments in comparison; e.g., if the two segments are youth (1) and adults (2), we may require that $x(1) \leq x(2)$, i.e., the price offered to the youth segment must be at most that offered to the adults. Similarly, we may require that $x(2) \leq x(1) + 5$, i.e., while the adults can be priced higher than the youth, the difference cannot be larger than \$5. Such a notion of price fairness has been recently proposed by Cohen et al. 2022 [10] under the assumption that the slacks are non-negative and symmetric (so that the constraints amount to requiring that $|x(i) - x(i')| \leq s(i, i') = s(i', i)$ for all segments i, i').
3. *Fair assortment planning.* Chen et al. (2022) [8] recently considered the scenario where a platform needs to make assortment planning decisions over n products/sellers to show to a customer. Each product i has some weight $w_i \geq 0$ that measures how popular the product is to customers; in particular, upon offering the customers an assortment $\mathcal{S}$ of products, the customers purchase product $i \in \mathcal{S}$ according to the Multinomial Logit Choice (MNL) model, with probability $w_i/(1 + w(\mathcal{S}))$ where $w(\mathcal{S}) = \sum_{j \in \mathcal{S}} w_j$. Given a probabilistic assortment choice of the platform, suppose that $x(i)$ is the probability that i is included in the assortment, referred to as the *visibility* offered to product i. [8] propose a notion of fairness that requires that for every product i and for some $\delta > 0$,
$$x(i) \geq \frac{w_{i'}}{w_i} x(i') - \delta, \quad \text{for all} \quad i, i' \in [n].$$

That is, product i must get at least as much visibility (with a δ slack) as product i' after adjusting for their relative differences in popularities.

For each of these scenarios, the feasible decision space $\mathcal{X}$ can be intersected with certain fairness constraints $\mathcal{F}$. The static optimization problem of minimizing the cost of decisions subject to fairness constraints can then be expressed as follows:

$$\text{minimize} \quad f(x) \tag{2}$$
$$\text{subject to} \quad x \in \mathcal{X} \cap \mathcal{F}, \tag{3}$$

which is easy to optimize as long as the cost function has nice properties like convexity or unimodality, and $\mathcal{X} \cap \mathcal{F}$ can be separated over in polynomial time [16].

3 Ensuring Fairness in Dynamic Settings

The introduction of time in the above setup makes it more interesting. Suppose that one needs to take a decision x_t for each time period $t \in \{1, \ldots, T\} = [T]$. First, note that if there is no reason to choose different decisions across time periods, e.g., in settings where the cost function $f(\cdot)$ is unchanging and known to the decision-maker, then the introduction of time is redundant since the decision-maker can simply choose the same static-optimal fair decision at each time. However, there may be reasons that may make the decision-maker want to vary decision rules over time. For example, the cost function $f(\cdot)$ itself could be time-dependent in certain settings. Or $f(\cdot)$ could be unknown to the decision maker, and the optimal fair decision must be learned over time. In our illustrative dynamic pricing example that we will discuss in Section 4, we will focus on this latter case. Thus, a temporal fairness constraint should ideally allow room for some changes in decisions over time to allow for learnability. At the same time, it should satisfy a meaningful desideratum of equity and non-discrimination from the perspective of the entities. Let's first try to understand the concerns in defining a temporal fairness constraint that satisfies these requirements.

Fairness within Each Period. The first and most basic possibility in this regard is to simply ensure that the static fairness constraints are satisfied for each group independently in each time period, i.e., there is no interaction between x_t and $x_{t'}$, except that each time period the decisions x_t must satisfy fairness constraints, i.e., $x_t \in \mathcal{X}\ \forall t \in [T]$. Learning unknown cost functions $f(\cdot)$ are easy in this case, using the theory of online convex optimization, as long as the cost functions are well-behaved. For instance, it may be acceptable that salaries of women and men are constrained to be within a factor of each other for equitability, but it does not matter if they rise or fall together.

Fairness across All Time. The second possibility lies on the opposite end of the spectrum, requiring that the fairness of a decision must be satisfied with respect to decisions across all times. That is, each decision $x_t(i)$ for an entity i is "fair" with respect to all decisions across time for every entity, i.e., $x_{t'}(j)$ for all $t' \in [T], j \in [n]$. While ideal from a fairness perspective, this requirement is too stringent since it often disallows any change in decisions over time [17]. For example, individual fairness constraint (1) when imposed for the same entity across time results in being unable to change decisions for that entity across time, i.e.,

$$|x_t(i) - x_{t'}(j)| \le L|c(i) - c(j)| \quad \text{for all} \quad i, j \in [n], t, t' \in [T], \tag{4}$$
$$\Rightarrow x_t(i) = x_{t'}(i) \quad \text{for all} \quad t, t' \in [T]. \tag{5}$$

As noted in (5), this means that learning is not possible, since decisions for entity i cannot change over time. One could consider a slack to the Lipchitz constraint, but this would still restrict the set of feasible solutions significantly, unless the slack incorporates the notion of time.

Fairness at the Time of Decision. The above discussion brings us to *Fairness at the Time of Decision* (FTD) [17, 28] that relaxes the fairness constraint to incorporate the notion of time: decision received by entity i at time t is guaranteed to be "fair" with respect to all decisions received by any entity *before* time t. In other words, we care about the fairness of a decision only with respect to the data available at the time of the decision. The decision of each entity must respect the tightest lower bound obtained by considering all decisions in the past (assuming higher decisions are more conducive). Mathematically, to ensure individual fairness at *the time of decision*, as an example, we require that for each $i \in [n]$,

$$x_t(i) \ge x_{t'}(j) - L|c(i) - c(j)|, \quad \text{for all} \quad j \in [n], t' \le t. \tag{6}$$

In particular, such a constraint then asks for iterates $x_1, \ldots, x_t$ to be monotone in each coordinate, i.e., $x_t(i) \ge x_{t'}(i)$ for all $t' < t$. Such a one-sided notion of fairness has philosophical connections with stare decisis in law (every case is held to the standard of past precedents) [27], markdown pricing in retail (prices reduces over time) [22], and whataboutisms in political science (every opinion is questioned with respect to opinions in similar past cases) (see [17]).

Fairness at the time of the decision thus occupies a convenient middle ground that straddles the two extremes of temporal notions of fairness discussed above. Other ways of incorporating fairness across time using sliding windows [20] or by including fairness as one of multiple objectives (not as a constraint) [4] have also been proposed. Each of these notions lead to technically new challenges in online optimization. FTD, in particular, offers enough flexibility in decisions that could potentially allow for some experimentation and learning, and at the same time these intertemporal constraints are novel in the context of online optimization and require new design tools and techniques to assimilate them while ensuring good cost guarantees. In the next section, we illustrate these design challenges that arise in the context of multi-group stochastic convex optimization.

4 Trajectory-Constrained Stochastic Convex Optimization

To illustrate the challenges in designing a FTD algorithm in an online learning setting, we consider the framework of stochastic convex optimization. Consider a simple setting where a decision-maker chooses decisions in $\mathcal{X} = [x_{\min}, x_{\max}]^2 \subseteq \mathbb{R}^2_{\ge 0}$ for two groups, 1 and 2, repeatedly over T periods. A decision $x(i)$ for group i leads to a cost $f_i(x(i))$, where f_i is assumed to be a β-smooth and α-strongly concave function for each $i \in \{1, 2\}$. The function f_i is not known a priori to the decision-maker. Upon choosing a decision $x_t(i)$ at time t for group i, the decision-maker observes the noisy function value $f_i(x_t(i)) + \epsilon_{i,t}$, where $(\epsilon_{i,t})_t$ is a sequence of independent zero-mean sub-Gaussian random variables for each $i \in [n]$. There is no gradient information readily available, i.e., we consider the bandit feedback setting.

The notion of fairness we consider is a form of approximate *envy freeness*. In particular, we assume without loss of generality that higher decisions are more conducive for the groups, and we require that each group shouldn't envy the other's decision too much, where the notion of "too much" is captured by certain slacks. In particular, the static fairness constraint we require that $x(i) \geq x(j) - s(i,j)$ for $i, j \in \{1,2\}$, where $s(i,j) \geq 0$ is the permissible slack representing how much the decision of i is allowed to be lower than that of j. We assume that $s(i,i) = 0$ for each i. For simplicity, we refer to this notion of fairness as Envy Freeness (EF). Therefore, the set of feasible decisions $\mathcal{X} \cap \mathcal{F}$ is:

$$\mathcal{X} \cap \mathcal{F} = \begin{Bmatrix} x(i) \geq x(j) - s(i,j) & \text{for} \quad i,j \in \{1,2\}, \\ x_{\min} \leq x(i) \leq x_{\max}, & \text{for} \quad i \in \{1,2\} \end{Bmatrix}. \tag{7}$$

We now add the temporal constraints to ensure fairness at the time of decision and constrain the iterates over time as follows:

$$x_t(i) \geq x_{t'}(j) - s(i,j), \quad \forall t \geq t', \quad \forall i,j \in \{1,2\}. \tag{8}$$

We refer to these as EFTD constraints (i.e., envy-freeness at the time of decision). Assuming that $s(i,i) = 0$ for all $i \in \{1,2\}$, it is easy to show that (8) is equivalent to:

$$x_t \in \mathcal{X} \cap \mathcal{F}, \quad \text{for all} \quad t \in [T], \tag{9}$$

$$x_t(i) \geq x_{t'}(i) \quad \text{for each} \quad i \in [n] \quad \text{and} \quad t \geq t', \tag{10}$$

which gives us a trajectory-constrained stochastic convex optimization over $\mathcal{X} \cap \mathcal{F}$. We refer to constraints (9) as cross-coordinate constraints, and (10) as monotonicity constraints. Our goal is to minimize the regret for the trajectory-constrained optimization problem, where regret is simply given by:

$$\sum_{t=1}^{T} \sum_{i \in \{1,2\}} \mathbb{E}\Big(f_i(x_t(i))\Big) - T \min_{x \in \mathcal{X} \cap \mathcal{F}} \sum_{i \in \{1,2\}} f_i(x(i)), \tag{11}$$

where the expectation is over the randomness in $x_1, \ldots, x_T$, and x_t satisfy the cross-coordinate and monotonicity constraints.

For minimizing unknown and noisy smooth and strongly convex (SSC) functions $f(\cdot)$ over convex sets $\mathcal{X}$, without enforcing monotonity of iterates, it is well-known that $\Omega(\sqrt{T})$ regret is inevitable, due to a result of Shamir (2013) [29]. A near-optimal algorithm for this case was given by Agarwal et al. (2013) [1]. In the one-dimensional case, their approach is the most related to the golden-section search procedure of Kiefer (1953) [26]: it iteratively uses three-point function evaluations to "zoom in" to the optimum, by eliminating a point and sampling a new point in each round. Its mechanics however render it infeasible to implement it in a fashion that respects the monotonicity of decisions. Another algorithm that gets the $O(\sqrt{T})$-regret in bandit feedback setting is due to Hazan and Levy (2014) [19]. They use gradient-descent with a one-point gradient estimate constructed by sampling uniformly in a ball around the current point. This key idea repeatedly appears in several works on convex optimization with bandit feedback (e.g., [31], [14], [19]). However, due to the randomness in the direction chosen to estimate the gradient, such an approach does not satisfy monotonicity of iterates. It was thus largely unclear if the optimal regret rate of $\widetilde{O}(\sqrt{T})$ can be obtained for SSC stochastic convex optimization with bandit feedback, while satisfying the EFTD constraint. The result below from Salem et al. (2022) [28] answers this question in the affirmative for our setting.

Theorem 4.1 ([28]). *Suppose that the minimizer $x^* = \arg\min_{x \in \mathbb{R}^2} \sum_{i \in \{1,2\}} f_i(x(i))$ is such that $x^*(i) > x_{\min}$ for each i. Then there exists an algorithm for choosing decisions for two groups over time that satisfies EFTD and that attains a regret of $\widetilde{O}(\sqrt{T})$ under noisy bandit feedback assuming that $f(x) = f_1(x(1)) + f_2(x(2))$ is β-smooth and α-strongly convex.*

We explain the algorithmic ideas to obtain this result next, and highlight open questions thereafter.

4.1 Key Algorithmic Ideas

There are several algorithmic ideas in achieving the result of Theorem 4.1. Discussing all of them is beyond the scope of this article and so we focus on the key ones, and defer the details to [28].

Let's first focus on the problem of minimizing a single-dimensional function $f(\cdot)$ over $\mathcal{X} = [x_{\min}, x_{\max}]$, using bandit feedback *while ensuring that the decisions x_t are monotonically non-decreasing*. In this case, the high-level trade-off between minimizing regret and increasing decisions is the following. Starting from $x_{\min}$, the decisions should increase to the unknown optimum at a sufficient rate to ensure low regret. However, hastiness is associated with an increased risk of overshooting the optimum due to lack of confidence in the gradient estimate, which would lead to high regret since backtracking is not allowed.

The main idea for addressing this trade-off is to tailor the degree of caution (i.e., the speed of approach) to the local gradient. Indeed, if we had access to gradient feedback, then it is easy to show that the standard gradient descent dynamics

$$x_{t+1} = x_t - \eta \nabla f(x_t),$$

beginning with $x_0 = x_{\min}$ *monotonically* converge to the optimum at an exponential rate *while never overshooting the optimum*, assuming that the step-size η is chosen appropriately as a function of β (the smoothness parameter). However, to execute such a procedure with noisy bandit feedback, an estimate of the gradient at x_t must be constructed. We can do so by sampling the function repeatedly at two points x_t and $x_t - \delta$ separated by some lag $\delta > 0$ (we first sample at $x_t - \delta$ and then at x_t to ensure monotonicity). Overshooting can then be avoided by moving from the lagged point, i.e.,

$$x_{t+1} = x_t - \delta - \eta g_t,$$

assuming that g_t is a gradient estimate that satisfies $\nabla f(x_t - \delta) \leq g_t \leq \nabla f(x_t)$. If x_{t+1} thus calculated is smaller than x_t, then we stop the procedure to ensure monotonicity. The following lemma characterizes the sample complexity of calculating g_t.

Lemma 4.2 (sandwich lemma). *[28] Let $f : \mathbf{R} \to \mathbf{R}$ be an α-strongly convex function. Let $x < y$ and let $\delta = y - x$. Fix $p \in (0,1)$. Then one can define $\bar{f}(x)$, $\bar{f}(y)$ to be the averages of*

$$\Theta\left(\frac{\log \frac{1}{p}}{\alpha^2 \delta^4}\right)$$

samples at x and y, respectively, so that the estimated secant

$$g = \frac{\bar{f}(y) - \bar{f}(x) + \frac{\alpha \delta^2}{4}}{\delta}$$

satisfies $\nabla f(x) \leq g \leq \nabla f(y)$, with probability at least $(1-p)^2$.

In other words, to estimate the gradient of a strongly convex function between two points separated by δ, it is sufficient to sample these points $\widetilde{\mathcal{O}}(1/\delta^4)$ times (we can also argue that this is necessary). This means that δ cannot be too small, otherwise excessive regret would be incurred in the gradient estimation process. At the same time, if δ is too large in relation to g_t then the procedure may stop prematurely and far from the optimum, resulting in high regret. We can show that one can choose an appropriate value of δ that balances the estimation regret and the stopping regret to yield a $\widetilde{\mathcal{O}}(T^{2/3})$ regret monotone algorithm.

To attain the near-optimal $\widetilde{\mathcal{O}}(\sqrt{T})$ regret, we adaptively tailor the lag size to the local gradient estimate. In particular, if we find that the algorithm has stopped moving for a particular lag size δ, then we halve the value to $\delta/2$ and attempt to keep moving, halving the value further as necessary. The benefit of this approach is that smaller values of the lag size δ, which result in high sampling rate for gradient estimation, are utilized only when the decisions are close to the optimum where they result in low regret. There is one remaining issue, which is that the decision of lowering the lag size δ must be made before sampling at x_t to ensure monotonicity. We tackle this issue by constructing interim gradient estimates to search for the right lag size before sampling x_t. The details of this design are beyond the scope of this article. Overall, we can show that this procedure, which is called *Adaptive Lagged Gradient Descent* (ADA-LGD) attains the required $\widetilde{\mathcal{O}}(\sqrt{T})$ regret [28].

Now to address the two-group case, we need to additionally address the cross-coordinate constraints, i.e., $x_t(1) \geq x_{t'}(2) - s(1,2)$ and $x_t(1) \geq x_{t'}(2) - s(2,1)$ for all $1 \leq t' \leq t \leq T$. Our overall approach is once again more simply described as a continuous-time procedure in the case where we have access to perfect gradient feedback, i.e., $\nabla f_i(x_t(i))$:

1. *Coordinate-descent phase (continuous-time).* Starting with $x_0(1) = x_0(2) = x_{\min}$, pick an arbitrary coordinate, say i, and increase it while keeping the other coordinate $j_{\neq i}$ fixed until either (a) $\nabla f_i(x_t(i)) = 0$, or (b) $x_t(i) = x_t(j_{\neq i}) + s(j_{\neq i}, i)$. Switch the coordinate $i \leftarrow j_{\neq i}$ and repeat until, for both $i = 1, 2$, either $x_{i,t} = x_t(j_{\neq i}) + s(j_{\neq i}, i)$ or $\nabla f_i(x_t(i)) = 0$. Once that happens, go to step 2.
2. *Combined-descent phase (continuous-time).* If $\nabla f_i(x_t(i)) = 0$ for both $i = 1, 2$, then we are done – the EFTD constraint doesn't bind and the unconstrained optimum is the same as the constrained optimum. Else, there is some $i^* \in \{1, 2\}$ such that $\nabla f_i(x_t(i^*)) = 0$ and $x_t(j^*_{\neq i}) = x_{i^*,t} + s(i^*, j^*_{\neq i})$ (since we assumed at least one slack is non-zero). At that point, we can deduce that the corresponding EF constraint will be tight at the optimum (which means we can again reduce to the single group case). Then define $h(x) = f_{i^*}(x) + f_{j^*_{\neq i}}(x + s(i^*, j^*_{\neq i}))$, and continue reducing $x_{i^*,t}$ and $x_{j^*_{\neq i},t} = x_{i^*,t} + s(i^*, j^*_{\neq i})$ jointly until $\nabla h(x_{i^*,t}) = 0$.

The challenge then is to convert this process into a practical discrete-time procedure with only noisy bandit feedback on each dimension, *while ensuring optimal overall regret*. One major piece of the algorithmic approach is to utilize the ADA-LGD procedure alternatingly on each dimension in the coordinate descent phase and then jointly in the combined descent phase. However, there are several design details, such as when to decide to switch between coordinates, when to enter the combined phase, how to transition the adaptive lag size schedule into the combined phase, etc., which are outside the scope of this article. We refer the reader to [28] for details.

5 Open Directions

These ideas naturally lead to the following open question: for which settings can one achieve order-optimal regret or convergence rates when constraining the trajectory of iterates within iterative decision-making scenarios? This question is largely open, in terms of types of decision sets, types of fairness constraints and resultant trajectory constraints, and also the assumptions on the functions to be optimized. In the current literature, we know very little about lower bounds for various settings. For example, the assumption of smoothness (which allowed elimination of overshooting) and strong convexity (which is crucial for the sandwich lemma) appears to be crucial for the result of Theorem 4.1 for even single-dimensional problems. For single-dimensional problems, Jia et al. [22] and Chen et al. [7] have recently considered bandit stochastic optimization for general Lipschitz functions. In particular, for unimodal functions, they show that enforcing monotonicity of decisions results in the optimal achievable regret of $\tilde{\Theta}(T^{3/4})$, which is larger than a regret of $\tilde{\Theta}(T^{3/4})$ under no monotonicity constraints. So it appears that some form of regularity of the functions is necessary to ensure no impact of the EFTD constraint on the optimal regret (at least up to logarithmic terms). In fact, it appears that smoothness is necessary upon considering the worst-case examples of [22] and [7]. For more open problems with respect to interesting settings for trajectory-constrained optimization, we refer the reader to [28].

Swati Gupta, Massachusetts Institute of Technology, Sloan School of Management, 100 Main Street Cambridge, MA 02142, USA
swatig@mit.edu

Vijay Kamble, Department of Information and Decision Sciences
University of Illinois Chicago, Chicago, IL 60607, USA
kamble@uic.edu

Jad Salem, School of Mathematics
Georgia Institute of Technology, Atlanta, GA 30332, USA
jsalem7@gatech.edu

Bibliography

[1] A. Agarwal, D. P. Foster, D. Hsu, S. M. Kakade, and A. Rakhlin. Stochastic convex optimization with bandit feedback. *SIAM Journal on Optimization*, 23(1):213–240, 2013.
[2] G.-Y. Ban and N. B. Keskin. Personalized dynamic pricing with machine learning: High-dimensional features and heterogeneous elasticity. *Management Science*, 67(9):5549–5568, 2021.
[3] S. Bansal, A. Srivastava, and A. Arora. Topic modeling driven content based jobs recommendation engine for recruitment industry. *Procedia computer science*, 122:865–872, 2017.
[4] Y. Bechavod, C. Jung, and S. Z. Wu. Metric-free individual fairness in online learning. *Advances in Neural Information Processing Systems*, 33:11214–11225, 2020.
[5] L. E. Bolton, L. Warlop, and J. W. Alba. Consumer perceptions of price (un) fairness. *Journal of consumer research*, 29(4):474–491, 2003.
[6] CCPA. TITLE 1.81.5. California Consumer Privacy Act of 2018 (Title 1.81.5 added by Stats. 2018, Ch. 55, Sec. 3.).
[7] N. Chen. Multi-armed bandit requiring monotone arm sequences. *Advances in Neural Information Processing Systems*, 34:16093–16103, 2021.
[8] Q. Chen, N. Golrezaei, F. Susan, and E. Baskoro. Fair assortment planning. *Available at SSRN 4072912*, 2022.

[9] V. X. Chen and J. Hooker. Fairness through social welfare optimization, 2021. arXiv:2102.00311.

[10] M. C. Cohen, A. N. Elmachtoub, and X. Lei. Price discrimination with fairness constraints. *Management Science*, 2022.

[11] P. R. Darke and D. W. Dahl. Fairness and discounts: The subjective value of a bargain. *Journal of Consumer psychology*, 13(3):328–338, 2003.

[12] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science (ITCS)*, pages 214–226. ACM, 2012.

[13] G. Farnadi, W. St-Arnaud, B. Babaki, and M. Carvalho. Individual fairness in kidney exchange programs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11496–11505, 2021.

[14] A. D. Flaxman, A. T. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394, 2005.

[15] S. Gillen, C. Jung, M. Kearns, and A. Roth. Online learning with an unknown fairness metric. *Advances in neural information processing systems*, 31, 2018.

[16] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[17] S. Gupta and V. Kamble. Individual fairness in hindsight. *Journal of Machine Learning Research*, 22(144):1–35, 2021.

[18] K. L. Haws and W. O. Bearden. Dynamic pricing and consumer fairness perceptions. *Journal of Consumer Research*, 33(3):304–311, 2006.

[19] E. Hazan and K. Levy. Bandit convex optimization: Towards tight bounds. *Advances in Neural Information Processing Systems*, 27, 2014.

[20] H. Heidari and A. Krause. Preventing disparate treatment in sequential decision making. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2248–2254, 2018.

[21] J. E. Heyman and B. A. Mellers. Perceptions of fair pricing. *Handbook of Consumer Psychology*, pages 683–697, 2008.

[22] S. Jia, A. Li, and R. Ravi. Markdown pricing under unknown demand. *Available at SSRN 3861379*, 2021.

[23] S. Jia, A. Li, and R. Ravi. Dynamic pricing with monotonicity constraint under unknown parametric demand model. *Advances in Neural Information Processing Systems*, 35:19179–19188, 2022.

[24] M. Joseph, M. Kearns, J. Morgenstern, and A. Roth. Fairness in learning: classic and contextual bandits. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 325–333, 2016.

[25] N. B. Keskin and A. Zeevi. Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. *Operations research*, 62(5):1142–1167, 2014.

[26] J. Kiefer. Sequential minimax search for a maximum. *Proceedings of the American mathematical society*, 4(3):502–506, 1953.

[27] J. Knight and L. Epstein. The norm of stare decisis. *American Journal of Political Science*, pages 1018–1035, 1996.

[28] J. Salem, S. Gupta, and V. Kamble. Algorithmic challenges in ensuring fairness at the time of decision. *(major revision), Operations Research*, 2023.

[29] O. Shamir. On the complexity of bandit and derivative-free stochastic convex optimization. In *Conference on Learning Theory*, pages 3–24. PMLR, 2013.

[30] A. K. Sinha, A. K. Akhtar, A. Kumar, et al. Resume screening using natural language processing and machine learning: A systematic review. *Machine Learning and Information Processing*, pages 207–214, 2021.

[31] J. C. Spall et al. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.

[32] L. Xia, K. Monroe, J. Cox, B. Kent, K. Monroe, and J. Jones. The price is unfair! a conceptual framework of price fairness perceptions. *Journal of Marketing*, 68:1–15, 11 2004. doi:10.1509/jmkg.68.4.1.42733.

The Evolution of *Mathematical Programming Computation*, Currently with the Highest Two-Year Impact Factor among Applied Mathematics Journals

Jonathan Eckstein

Impact

Mathematical Programming Computation (MPC) has been a Mathematical Optimization Society (MOS) journal for 14 years. Until recently, the Web of Science organization considered it an "emerging" journal and did not assign it an official impact factor. That changed this past June, when Web of Science assigned MPC a two-year impact factor of 8.059, which ranks *first* among 267 applied mathematics journals and sixth among 110 computer science and software engineering journals. Clearly, MPC made a dramatic entry into the ranks of established journals!

Without an impact factor, some authors had a strong incentive to avoid submitting manuscripts to MPC, because their institutions would award them little or no "credit" for publishing there. That situation is now over, so if MPC seems like the right fit for a manuscript, there should now be every reason to submit there.

MPC is a relatively low-volume journal in comparison to other optimization journals like *Mathematical Programming* Series A and B (MPA and MPB). MPC currently publishes about 20 papers per year. Since a journal's impact factor is an average computed over all the items it has published in the preceeding two or five years, MPC's impact factor can be expected to exhibit relatively high variance: an average computed over a sample of 40 will be more variable than an average computed over a sample 226 (the number of articles published by MPA in 2020–2021). Due to its high anticipated variability, it is uncertain whether MPC's impact factors will rank as high in future years, but it is clear that MPC is a journal that is well respected and here to stay.

History

MOS (then called the Mathematical Programming Society) launched MPC in July 2008, based on some discussions initiated by Martin Grötschel in January 2007. Bill Cook served as initial editor-in-chief.

From its inception, MPC has had a feature essentially unique not only among optimization journals, but among applied mathematics journals in general: authors submit not just a manuscript but also the computer code and data used to produce their computational results. Special reviewers called "technical editors" check whether the results seem reproducible, and evaluate both the quality of the code and its potential usefulness to others. Technical editors are listed as editorial board members in recognition of their invaluable service to the journal, although they function largely like referees. The only other journal I am aware of that has anything like MPC's software review process is the ACM Transactions on Mathematical Software (ACM TOMS), which has five categories of submissions, two of which require accompanying software. However, the journal's published policies do not specify how such software is reviewed.

The way technical editors have performed their task has evolved significantly since the formation of MPC. Initially, they downloaded authors' software and compiled and tested it on their own computer equipment. This arrangement put considerable burdens on authors to develop highly portable code and anticipate what environments their code might be tested in, and on technical editors to successfully compile and use code developed on potentially different systems. Any communication about this process generally had to be through an intermediary, since technical editors are generally anonymous, like referees.

Dan Bienstock, who served as editor in chief from 2014 to 2018, changed this system, maintaining a set of server computers on which gave accounts to both authors and technical editors. This way, authors had the responsibility for making sure their code would build and run correctly on a system to which the technical editor had anonymous access. Dan also overhauled the internal processes and workflow of the journal, better coordinating technical editors and associate editors, as well as changing the manuscript management system.

Dan's approach, while a great improvement, had some recurring difficulties involving software versioning conflicts between different manuscripts hosted on the same server. One manuscript's code might require a compiler, system library, or math library version incompatible with another manuscript's. While in principle one of the manuscripts involved in such a conflict could use different support software localized in its own account, such multi-configurations were time consuming for both Dan and for authors.

To address these issues, the technical review system evolved again under my leadership as editor-in-chief, which started in January 2019. MOS purchased a 32-core, 64-thread server which was installed at Rutgers Business School. It runs a virtual-machine (VM) operating system, within which each submission gets its own Linux virtual machine, with a unique temporary IP address. Authors have root-user system privileges on their submission's VM, and are free to configure it as they wish, updating or downdating any support software incompatible with their implementation. Rutgers Business School's Office of Technology and Instruction services (OTIS) provides basic support for the VM system; I would like to thank Rutgers Business School's Dean Lei Lei for sponsoring this valuable support.

Aims and Scope

MPC's aims and scope are now somewhat broader than the original conception, with topics including (but not limited to) the following aspects of optimization and closely related mathematical problems:

- New algorithmic techniques, with substantial computational testing
- New applications, with substantial computational testing
- Innovative software
- Comparative tests of algorithms
- Modeling environments
- Libraries of problem instances
- Software applications, frameworks, or libraries.

The journal considers manuscripts related to all kinds of optimization, discrete or continuous, linear or nonlinear, and deterministic or stochastic, as well as on closely related topics.

There are some key differences between MPC and other top-tied optimization journals: most importantly, articles accepted to MPC are not necessarily expected to contain major theoretical advances (although they can). However, all accepted articles are expected to describe new contributions of practical computational value in optimization or closely related areas. With occasional exceptions (for some problem instance libraries, for example), articles are expected to contain rigorous computional testing, which the technical editors endeavor to (approximately) reproduce using the authors' submitted code.

Links for Further Information

- ▹ Springer's MPC Page: www.springer.com/math/journal/12532
- ▹ Mathematical Optimization Society: www.mathopt.org
- ▹ Journal mirror site at Zuse Institute Berlin: mpc.zib.de

Jonathan Eckstein
Department of Management Science and Information Systems
Rutgers University, Piscataway, NJ 08854, USA
jeckstei@business.rutgers.edu

Mixed Integer Programming Society (MIPS): A New Section of the Mathematical Optimization Society

The Mixed Integer Programming Society (MIPS) is a newly created section of the Mathematical Optimization Society (MOS). Its goal is to serve as a catalyst for the community of researchers working in Mixed Integer Programming and its applications, both inside and outside academia, and promote continuity of events of interest for the community. In particular, it supports the organization of the annual MIP workshops and of the online Discrete Optimization Talks, and promotes the dissemination of results in the area.

For more information or to become a member, please visit mixedinteger.org or write to mail@mixedinteger.org.

Yuri Faenza
Chair, Mixed Integer Programming Society

IMPRINT ■ *Editor:* Sebastian Pokutta, Zuse Institute Berlin (ZIB), Takustraße 7, 14195 Berlin, Germany. pokutta@zib.de ■ *Co-Editors:* Swati Gupta, Massachusetts Institute of Technology, Sloan School of Management, 100 Main Street Cambridge, MA 02142, USA, swatig@mit.edu Omid Nohadani, Artificial Intelligence and Data Science, Benefits Science Technologies, Boston, MA 02110, USA, onohadani@gmail.com ■ *Founding Editor:* Donald W. Hearn ■ Published by the Mathematical Optimization Society, www.mathopt.org ■ Design and typesetting by Christoph Eyrich, Berlin, Germany.