Tight approximation for the minimum bottleneck generalized matching problem

Julián Mestre[1] and Nicolás E. Stier Moses[2]

[1] School of Computer Science, University of Sydney.
[2] Facebook Inc.

Abstract. We study a problem arising in statistical analysis called the *minimum bottleneck generalized matching problem* that involves breaking up a population into blocks in order to carry out generalizable statistical analyses of randomized experiments. At a high level the problem is to find a clustering of the population such that each part is at least a given size and has at least a given number of elements from each treatment class (so that the experiments are statistically significant), and that all elements within a block are as similar as possible (to improve the accuracy of the analysis).

More formally, given a metric space (V, d), a treatment partition $\mathcal{T} = \{T_1, \ldots, T_k\}$ of V, and a target cardinality vector $(b_0, b_1, \ldots, b_k) \in Z_+^{k+1}$ such that $b_0 \geq \sum_{j=1}^{k} b_j$. The objective is to find a partition $M_1, \ldots, M_\ell$ of V minimizing the maximum diameter of any part such that for each part we have $|M_i| \geq b_0$ and $|M_i \cap T_j| \geq b_j$ for all $j = 1, \ldots, k$.

Our main contribution is to provide a tight 2-approximation for the problem. We also show how to modify the algorithm to get the same approximation ratio for the more general problem of finding a partition where each part spans a given matroid.

1 Introduction

In Social Science and related fields, designing experiments on a sample of the population so that the insights obtained from the experiments can be generalized to the whole population is a major challenge. Statistical techniques such as blocking, are used for designing sound experiments. Given a population the objective is to break it up into homogeneous *blocks* of at least a given minimum size and then randomly assign elements within blocks to treatment and control groups. It is important that these blocks are large enough (so that the results are statistically significant) and homogeneous (so that there are no hidden variables that could explain variabilities between treatment and control outcomes). While the concept of blocking and randomized experiments goes back to the seminal work of Fischer [5], the design of efficient algorithms for blocking has attracted the attention of the Statistics community [18, 7, 8] in more recent times. Indeed, the efficiency of the blocking algorithm used and the quality of the blockings found are crucial in the context of A/B testing in online advertising platforms where treatment effects on advertisers are typically small [13] yet very economically relevant due to the large scale of these platforms.

The work of Higgins *et al.* [8] is particularly relevant to our paper. The authors cast the problem of finding a good blocking as an optimization problem, which they call *minimum threshold blocking*: Given a metric space (V, d) and a cardinality lower bound b, the objective is to partition V so that each part has cardinality at least b and the maximum diameter of any one part is minimized. Here V is the population that we want to block and d a distance function capturing how similar any two elements in V are (low distance implying similarity). They showed that the problem admits a 4-approximation and that it is NP-hard to approximate within $2 - \epsilon$.

While designing experiments that use a good blocking structure is highly desirable, sometimes the treatment partition is already given to us, either because someone else performed the experiment, because the sample size is small, or because the dissimilarity function was not fully available at the time the experiment was run. In these situations, when analyzing the experimental results, we still want to partition our population so that each part is as homogeneous as possible, and each part gets enough representatives from each treatment class. Sävje *et al.* call this problem the *minimum bottleneck generalized matching*[3] and show how to generalize the 4-approximation of Higgins *et al.* [8] to get a 4-approximation for this more general problem.

In the Computer Science community, the problem of clustering points to minimize the maximum radius of the clusters such that each cluster has at least a given number of points has been studied by Aggarwal *et al.* [1] in the context of anonymity preserving clustering. This is identical to the minimum threshold blocking problem except that we need to minimize the maximum cluster radius rather than the maximum cluster diameter. The authors call this problem *r-gather* and they

[3] Matching here refers to the concept in Statistics. It should not be confused with the traditional concept from Graph Theory.

give an optimal 2-approximation algorithm that in term generalizes the classical 2-approximation for k-center of Hochbaum and Shmoys [9]. Although the radius and diameter objectives are not equivalent, it is known how to modify the algorithm for r-gather to minimize the diameters of the clusters rather than the radii [11]. Enforcing the treatment partition constraints on the clusters, however, cannot be reduced to the r-gather problem.

Our main contribution is a 2-approximation algorithm for the minimum bottleneck generalized matching problem, which matches the hardness of approximation of Higgins *et al.* [8] for the special case of minimum threshold blocking. We also extend our 2-approximation algorithm to handle more complex constraints that go beyond the treatment partition constraints and involve finding a partition whose parts span a given matroid [17].

1.1 Related work

The problem of clustering points in a metric space has been studied extensively in Algorithm Theory. Many objectives have been proposed such as k-center [9, 6] where we want to minimize the maximum radius of the clusters, k-median [15] where we want to minimize the sum of the cluster radii, and k-means [16] where we want to minimize the total intra-cluster variance.

In addition to different objectives, researchers have proposed side constraints to the clustering problem such as allowing the algorithm to leave a small set of outliers unclustered [3], imposing capacity constraints [12] or anonymity constraints [1] on the clusters, or a matroid constraint on the set of centers we can pick [20].

To the best of our knowledge, none of these works deals with the bottleneck generalized matching problem of Sävje *et al.* [19]. The most closely related work that we are aware is the work of Li *et al.* [14] on ℓ-diversity clustering: Here they want a clustering such that each cluster has at least ℓ points, and all of its points come from different treatment classes and the goal is to minimize the maximum radius of any cluster. Our cluster constraints are in a sense complementary; namely, instead of upper bounding how many points we need from a treatment class, we want to get at least a prescribed number.

2 Formal problem definition and notation

The input of the *minimum bottleneck generalized matching* problem is a metric space (V, d), a treatment partition $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ of V, and a target cardinality vector $(b_0, b_1, b_2, \ldots, b_k) \in Z_+^{k+1}$ such that $b_i \leq |T_i|$ for all i and $b_0 \geq \sum_{j=1}^{k} b_j$. The distance function $d : V \times V \to R$ is non-negative, symmetric, and obeys the triangle inequality.

Our ultimate goal is to compute a partition $\mathcal{M} = \{M_1, M_2, \ldots\}$ of V. A partition is said to be feasible if for all $M \in \mathcal{M}$ we have $|M| \geq b_0$ and $|M \cap T_j| \geq b_j$ for all $j \in [k]$. Here $[k]$ is a short hand notation for the set $\{1, \ldots, k\}$. Later we will use $[0, \ldots, k]$ to denote the set $\{0, 1, \ldots, k\}$.

We define the cost of a partition $\mathcal{M}$ to be the maximum diameter[4] among parts $M \in \mathcal{M}$:

$$cost(\mathcal{M}) = \max_{M \in \mathcal{M}} \max_{u,v \in M} d(u,v)\,.$$

The goal of the minimum bottleneck generalized matching problem is to find a feasible partition $\mathcal{M}$ with minimum cost. Our main result is a 2-approximation algorithm for this problem. The approach is based on ideas from an algorithm of Aggarwal *et al.* [1] for the r-gather problem, where they only have a lower bound on the size of the cluster, but no treatment partition constraints.

3 Minimum bottleneck generalized matching

In this section we prove that there is a 2-approximation for our problem.

Theorem 1. *There is a polynomial time 2-approximation algorithm for the minimum bottleneck generalized matching problem.*

Let opt be the cost of the optimal solution. Suppose that we had a polynomial time routine parametrized by a scalar g such that:

- if $g \geq$ opt the routine returns a solution with cost $\leq 2g$, and
- if $g <$ opt the routine either reports "failure" or returns a solution with cost $\leq 2g$.

We can use this parametrized routine to design a polynomial time 2-approximation algorithm as follows. For each pair $u, v \in V$, run the routine with $g = d(u,v)$ and return the best solution found.
Note that one of these choices of g must equal opt, so for that choice we are guaranteed a solution with cost 2opt. Returning the best solution found can only yield a better result. Therefore, the correctness of the 2-approximation hinges on the existence of the parametrized routine. The rest of this section is devoted to developing this routine.

3.1 Description of the parametrized routine

Our routine attempts to build a feasible solution of cost at most $2g$ in three steps. First we pick a set of centers. Second, we build, if possible, a partial cluster around each of the centers that fulfills the treatment partition cardinality constraints. Third, we augment this partial solution by assigning the remaining points to a nearby cluster. Only the second step may not be possible to be carried out, in which case we declare "failure".

[4] The diameter is defined as the maximum distance between nodes in a set.

Finding centers. The first step of the parametrized routine is to select a set of centers $c_1, \ldots, c_\ell$ such that every element in V has a center at distance at most g and the distance between two centers is greater than g. More formally, the centers have the following two properties

1. $\min_{i \in [\ell]} d(u, c_i) \leq g$ for all $u \in V$, and
2. $d(c_i, c_{i'}) > g$ for all $i, i' \in [\ell]$ where $i \neq i'$.

We can compute such a set of centers using the iterative approach of Hochbaum and Shmoys [9]: Iteratively pick an arbitrary element v of V, declare v to be a center and remove from V all elements at distance at most g from v. The process ends when all elements of V have been removed.

Finding a partial solution. The second step is to construct a partial partition[5] $\{M_1, \ldots, M_\ell\}$ of V such that for each $i \in [\ell]$ we have the following three properties:

1. $|M_i| = b_0$,
2. $|M_i \cap T_j| \geq b_j$ for each $j \in [k]$, and
3. $d(v, c_i) \leq g$ for all $v \in M_i$.

We can find such a partial partition, if one exists, by solving a maximum flow problem in a layered directed graph depicted in Figure 1 and described below.

The vertex set of the network flow instance is as follows:

- In the first layer, we have the source s by itself.
- In the second layer, we have $k + 1$ dummy vertices for each center; namely, we have a vertex a_i^j for each $i \in [\ell]$ and $j \in [0, \ldots, k]$.
- The third layer contains the ground set V.
- Finally, in the fourth layer, we only have the sink t.

The layers are connected as follows:

- For all $i \in [\ell]$, the source s is connected to a_i^0 with an edge with capacity $b_0 - \sum_{j=1}^{k} b_j$.
- For all $i \in [\ell]$ and $j \in [k]$, the source s is connected a_i^j with an edge with capacity b_j.
- Each a_i^0 is connected to each $v \in V$ with an edge without capacity if $d(c_i, v) \leq g$.
- Each a_i^j is connected to each $v \in T_j$ with an edge without capacity if $d(c_i, v) \leq g$.
- Finally, each $v \in V$ is connected to t with an edge with capacity 1.

We solve this problem using any of the traditional combinatorial algorithms [2] for maximum s-t flow. These algorithms return an integral flow that obeys the capacity constraints and sends the maximum amount of flow from s to t. If the value of the maximum flow is less that $b_0 \cdot \ell$ then the parametrized routine declares "failure". Otherwise, we create a partial partition by setting M_i to be the set of nodes $v \in V$ such that there exists a unit of flow going from some a_i^j to v.

[5] A partial partition of V is a partition of a subset of V.

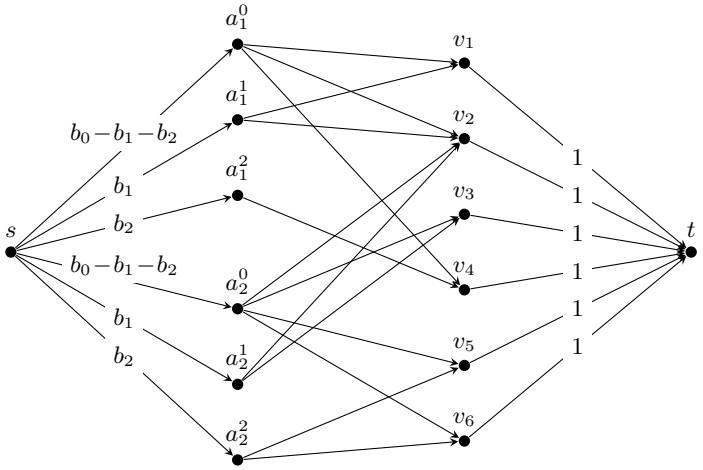

Fig. 1. Maximum flow instance for step two of the parametrized routine. In this example $\ell = 2$, $k = 2$, $T_1 = \{v_1, v_2, v_3\}$, and $T_2 = \{v_4, v_5, v_6\}$.

Augmenting the partial solution. The third and final step is to augment our partial solution by adding every vertex v not assigned so far to one of the parts M_i such that $d(v, c_i) \leq g$. Notice that because of the way the centers were constructed in the first step, we are always able to identify such a center.

If the algorithm does not declare failure in the second step, it returns the augmented solution from the third step that forms a full partition of V.

3.2 Correctness of the parametrized routine

If the routine does not fail, it returns a feasible partition $\{M_1, \ldots, M_\ell\}$ where $c_i \in M_i$ for each $i \in [\ell]$. This is because the centers are more than g apart from one another and we only assign to M_i that are at distance at most g to c_i. Furthermore, for any two vertices $u, v \in M_i$ we have $d(u, v) \leq d(u, c_i) + d(c_i, v) \leq 2g$, so the solution has cost at most $2g$ as desired.

If the routine fails, we need to argue that $g < \mathrm{opt}$. We prove the contrapositive: If $g \geq \mathrm{opt}$ then the routine does not fail. This boils down to arguing that the network flow problem defined in the second step of the routine is feasible. To that end, consider an optimal solution $\mathcal{O} = \{O_1, O_2, \ldots\}$. Recall that any two centers in $c_1, \ldots, c_\ell$ are at distance strictly greater than $g \geq \mathrm{opt}$ from one another. It follows that they must lie in different sets in the optimal solution. Assume, without loss of generality, that $c_i \in O_i$. We build a flow as follows. For each $j = 1, \ldots, k$, pick b_j elements $v \in O_i \cap T_j$ and push one unit of flow along the path $\langle s, a_i^j, v, t \rangle$; finally pick any $b_0 - \sum_{j=1}^{k} b_j$ elements $v \in O_i$ that were not

chosen so far and push one unit of flow along the path $\langle s, a_i^0, v, t\rangle$. (The existence of the elements is guaranteed by the feasibility of $\mathcal{O}$.) The resulting flow is feasible and has value $b_0 \cdot \ell$ as needed.

3.3 Time complexity and implementation details

The most expensive step of the parametrized routine is the computation of the maximum flow. An alternative to computing a flow would be build a bipartite graph where s and t are removed and the a_i^j vertex is replaced with b_j copies for $j \in [k]$ and $b_0 - \sum_{j=1}^{k} b_j$ copies for $j = 0$. The objective in this new graph is to find a maximum cardinality matching (matching in the standard Graph-theoretic sense). Using the Hopcroft-Karp algorithm[10], this can be done in $O(n^{2.5})$ time where $n = |V|$.
In principle this would have to be repeated for each possible choice of g of which there are $O(n^2)$ many. However, one can perform binary search on the candidate values of g until we find the smallest value of g for which the parametrized routine does not fail, which only adds a $O(\log n)$ factor to the $O(n^{2.5})$ running time.

4 Generalization to matroid constraints

In this section we explore a generalization of the basic setting that involves a richer set of constraints on each part that involves matroids.
Before we proceed any further, it is worth recalling some basic terminology from Matroid Theory [17]. A subset system is a pair (V, E) where E is a collection subsets of V such that for all $A \in E$ and $A' \subset A$ we have $A' \in E$. A subset system is a matroid if for all $A, B \in E$ such that $|A| < |B|$, there exists $x \in B \setminus A$ such that $A + x \in E$. The rank function associated with an independence system E is $\operatorname{rank} A = \max_{B \subseteq A : B \in E} |B|$, that is, the rank of A is the cardinality of the largest independent subset of A. Finally, a set A is said to span the matroid if $\operatorname{rank}(A) = \operatorname{rank}(E)$.
For our generalization, instead of a target partition and a cardinality vector like we had before, we are given a matroid (V, E) defined by the ground set V that we are to partition and an independence system E. The objective is to compute a partition $\mathcal{M} = \{M_1, M_2, \ldots\}$ of V such that M_i spans (V, E) for all $i \in [\ell]$ and the maximum diameter of any one part is minimized:

$$cost(\mathcal{M}) = \max_{M \in \mathcal{M}} \max_{u,v \in M} d(u, v).$$

As we shall see in Lemma 1, the constraints of the standard bottleneck generalized matching problem can be achieved with a carefully designed matroid system, so this new problem is a strict generalization of the former. For example, if each element in the ground set is associated with an edge in some auxiliary graph, then we could ask that each cluster forms a connect subgraph using a graphic matroid. We call this new problem the *minimum bottleneck generalized matching problem with a matroid constraint*.

Lemma 1. *Let (V, E) be a subset system where $A \in E$ if and only if $\sum_i \max(|A \cap T_i| - b_i, 0) \leq b_0 - \sum_i b_i$. Then (V, E) is a matroid and $A \in E$ is maximal if and only if $|A| = b_0$ and $|A \cap T_i| \geq b_i$ for all i.*

Proof. For any $A \in E$, we have

$$\begin{aligned}
|A| &= \sum_i |A \cap T_i| \\
&= \sum_i (|A \cap T_i| - b_i) + \sum_i b_i \\
&\leq \sum_i \max(|A \cap T_i| - b_i, 0) + \sum_i b_i \\
&\leq b_0 - \sum_i b_i + \sum_i b_i \\
&= b_0
\end{aligned}$$

Thus, $|A| \leq b_0$ for all $A \in E$. Furthermore, for any $A \in E$ if $|A| = b_0$ all inequalities are strict, so $b_i \geq |A \cap T_i|$ for all i. On the other hand, for any $A \in E$ if $|A| < b_0$ then the subset is clearly not maximal.
To see why the system is a matroid, let $A, B \in E$ such that $|A| < |B|$. If $\sum_i \max(|A \cap T_i| - b_i, 0) < b_0 - \sum_i b_i$ then for any $x \in B \setminus A$ we have $A + x \in E$. Otherwise, since $|A| < |B|$, there must exist i such that $|A \cap T_i| < b_i$ and $|A \cap T_i| < |B \cap T_i|$ in which case for any $x \in B \cap T_i \setminus A$ we have $A + x \in E$. □

To solve the problem we proceed as before, by designing a routine that is parametrized by a scalar g. If $g \geq$ opt, the routine returns a feasible solution with cost $2g$, or if $g <$ opt either returns a "failure" message or a solution with cost $2g$.
We can use this routine in the same way as we did in the previous problem to get a 2-approximation by guessing the value of opt and running the routine on each choice.

4.1 Parametrized routine

The first and third steps remain the same as before: In the first step we compute a set of centers $c_1, \ldots, c_\ell$ such that every element in V has a center at distance at most g and the distance between centers is strictly greater than g; while in the third step we augment the partial partition found in the modified second step. The key difference is how we find the partial solution that satisfies the matroid constraints in the second step.

Modified second step. The new second step involves solving a matroid intersection problem defined by two matroids (V', E_1') and (V', E_2').
The ground set of the matroids V' contains ℓ copies of each element in V, more formally,

$$V' = \left\{ v^i : \text{ for all } v \in V, i \in [\ell] \right\}.$$

The independence system of the first matroid enforces that the i-th copy of the element chosen is independent in the input matroid

$$E_1' = \left\{ X \subseteq V' : \left\{ v : v^i \in X \right\} \in E \text{ for all } i \in [\ell] \right\}.$$

The independence system of the second matroid enforces that we select at most one copy of each element

$$E_2' = \left\{ X \subseteq V' : \left| \left\{ v^i : i \in [\ell] \right\} \cap X \right| \leq 1 \text{ for all } v \in V \right\}$$

We use a matroid intersection algorithm to find a maximum cardinality set X in $E_1' \cap E_2'$. If $|X| < \text{rank}(V, E) \cdot \ell$, then we declare "failure". Otherwise, we create a partial partition with parts $M_i = \left\{ v \in V : v^i \in X \right\}$ for each $i \in [\ell]$. At this point each part spans the input matroid (V, E), however, there are elements that may not have been assigned. We assign each of these remaining elements $v \in V$ to a part M_j such that $d(v, c_j) \leq g$. Such a center is guaranteed to exist due to the way the centers are selected.

4.2 Correctness

The correctness of the new parametrized routine is similar to that of the old routine. If the routine returns a partition $\{M_1, \ldots, M_\ell\}$ then it satisfies the matroid spanning requirements; indeed, the partial partition $\{M_1', \ldots, M_\ell'\}$ already has the spanning property, namely $M_i' \in E$ and $|M_i'| = \text{rank}(V, E)$, so M_i' spans (V, E), and therefore so does M_i. Furthermore, the diameter of any part is at most $2g$ since for any two $u, v \in M_i$ we have $d(u, v) \leq d(u, c_i) + d(c_i, v) \leq 2g$.
Finally, we argue that if $g \geq \text{opt}$ then the parametrized routine never fails. Let $\mathcal{O} = \{O_1, O_2, \ldots\}$ be an optimal solution. Recall that any two centers indeed $c_1, \ldots, c_\ell$ are at distance strictly greater than $g \geq \text{opt}$ from one another. It follows that they must lie in different parts in the optimal solution. Assume, without loss of generality, that $c_i \in O_i$. Let $M_i' \subseteq O_i$ be a maximum cardinality independent set. Because O_i spans (V, E), it must be the case that $|M_i'| = \text{rank}(V, E)$. Let $X = \cup_{i \in [\ell]} \left\{ v^i : v \in M_i' \right\}$ be a subset of the ground set of the matroid intersection instance $(V', E_1' \cap E_2')$ defined in step two of the parametrized routine. Notice that $X \in E_1'$ because each $M_i' \in E$ and $X \in E_2'$ because the sets $\{M_1', \ldots, M_\ell'\}$ are disjoint.

4.3 Time complexity

Using the matroid intersection algorithm of Cunningham [4] we can find the needed maximum cardinality in $(V', E_1' \cap E_2')$ in $O(r^{1.5} n')$ calls to an independence oracle for the underlying matroids, where r is the maximum size of the common independent set and $n' = |V'|$. In our case, $r = O(n)$, $n' = O(\ell n) = O(n^2)$, and we can test independence in the matroids by using an oracle for the input matroid (V, E). Therefore, the running time is $O(n^{3.5} Q)$, where Q is the time it takes to test independence in (V, E).

As described in the previous section, we can implement the 2-approximation algorithm so as to perform $O(\log n)$ calls to the parametrized routine. Therefore, the overall running time is $O(n^{3.5}Q \log n)$.

5 Conclusion

In this paper we developed a tight 2-approximation algorithm for the minimum threshold generalized matching problem and showed that our approach can be generalized to tackle a more general version of the problem involving finding a partition whose parts span a given matroid. Our hope is that better approximations can lead to better statistical analyses.

Acknowledgement

We would like to thank Jasjeet Sekhon for early discussions on minimum bottleneck generalized matching.

References

1. G. Aggarwal, R. Panigrahy, T. Feder, D. Thomas, K. Kenthapadi, S. Khuller, and A. Zhu. Achieving anonymity via clustering. *ACM Transactions on Algorithms*, 6(3):49:1–49:19, 2010.
2. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, NJ, USA, 1993.
3. M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *Proc. of the 12th Annual Symposium on Discrete Algorithms*, pages 642–651, 2001.
4. W. H. Cunningham. Improved bounds for matroid partition and intersection algorithms. *SIAM Journal on Computing*, 15(4):948–957, 1986.
5. R. A. Fisher. The arrangement of field experiments. *Journal of the Ministry of Agriculture of Great Britain*, 33:503–513, 1926.
6. T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.
7. R. Greevy, B. Lu, J. H. Silber, and P. Rosenbaum. Optimal multivariate matching before randomization. *Biostatistics*, 5(2):263–275, 2004.
8. M. J. Higgins, F. Sävje, and J. S. Sekhon. Improving massive experiments with threshold blocking. *Proceedings of the National Academy of Sciences*, 113(27):7369–7376, 2016.
9. D. S. Hochbaum and D. B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985.
10. J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.

11. S. Khuller, 2019. Personal communication.
12. S. Khuller and Y. J. Sussmann. The capacitated K-center problem. *SIAM Journal of Discrete Mathematics*, 13(3):403–418, 2000.
13. R. A. Lewis and J. M. Rao. The Unfavorable Economics of Measuring the Returns to Advertising. *The Quarterly Journal of Economics*, 130(4):1941–1973, 2015.
14. J. Li, K. Yi, and Q. Zhang. Clustering with diversity. In *Proc. of the 37th International Colloquium on Automata, Languages and Programming*, pages 188–200, 2010.
15. S. Li and O. Svensson. Approximating k-median via pseudo-approximation. *SIAM Journal on Computing*, 45(2):530–547, 2016.
16. S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
17. J. G. Oxley. *Matroid Theory*. Oxford University Press, 1992.
18. P. R. Rosenbaum. Optimal matching for observational studies. *Journal of the American Statistical Association*, 84(408):1024–1032, 1989.
19. F. Sävje, M. J. Higgins, and J. S. Sekhon. Generalized Full Matching. *CoRR*, abs/1703.03882, 2019.
20. C. Swamy. Improved approximation algorithms for matroid and knapsack median problems and applications. *ACM Transactions on Algorithms*, 12(4):49:1–49:22, 2016.