Strategic clustering

Ana-Andreea Stoica
Columbia University
astoica@cs.columbia.edu
Christos Papadimitriou
Columbia University
christos@columbia.edu

ABSTRACT

How much does the quality of the clustering — typically measured by the conductance, or by the number of edges cut, or the average distance to the centers — deteriorate if the nodes are strategic and can change clusters? And among reasonable utilities for the nodes, which one hurts quality the least? We investigate these questions both theoretically, by studying the equilibria of *hedonic games* (simplified clustering games with unconstrained number of clusters), and experimentally, by measuring the quality of pure Nash equilibria of more realistic clustering games. We introduce a new utility function for the nodes which we call *closeness*, and which we believe is an attractive alternative to previously studied node utilities. We study the properties of the closeness utility theoretically and demonstrate experimentally its advantages over other established utilities such as the modified fractional utility. Finally, we present a polynomial-time algorithm which, given a clustering with optimal quality, finds another clustering with better average utility, and in fact the one that maximizes the ratio of the gain in average utility over the loss in quality.

ACM Reference Format:
Ana-Andreea Stoica, Columbia University, astoica@cs.columbia.edu and Christos Papadimitriou, Columbia University, christos@columbia.edu . 2018. Strategic clustering. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Clustering is an important paradigm of unsupervised learning, in which partitions of interest are discovered for a given metric on points; it has many applications in image segmentation, document classification, genetic classification, marketing and advertising (when a population of consumers is segmented for targeting (see e.g. Kleinberg et al. [28], Kuo et al. [33]), community detection, subdivision of participants into areas of interest, or of workers into teams or floors, etc. When the clustered points are people, as in the last few examples, socioeconomic considerations enter the picture, in addition to the usual attention to the *quality* of the clustering (large conductance, small number of cut edges, etc.), one has to consider how good a cluster is for its nodes — that is to say, incentives and utilities. This is the subject of this paper. Another important consideration is, how fair is the clustering to minority groups? Even though here we focus on utilities, we shall see that dealing with fairness requires some of the same ideas.

Among all possible types of input data for clustering problems, here we focus on unweighted undirected graphs whose nodes are people, or agents, and among clustering methodologies we focus on *spectral clustering* (see Von Luxburg [42] and the related work subsection), where the agents are represented as points in truncated spectral space and then grouped by an algorithm applied on the spectral projections of the nodes (e.g. k-means [25]); however, our methodology can be applied to any clustering algorithm. There are known objective *quality* measures for evaluating any clustering — that is, any subdivision of the nodes of the graph into clusters — such as *conductance*, *average distance from the k-means centers*, *number of cut edges*, etc. (see e.g. Almeida et al. [3], Emmons et al. [15]); here we will use mostly conductance.

When the nodes are agents and have a utility, besides the quality of the clustering we are interested in the *average welfare:* How are agents as a whole faring in the clustering in terms of their utility? Furthermore, attention must be paid to *stability*, that is, the clustering should be at least a Nash equilibrium — or even better, a Pareto optimum. Finally, we want to know to what extend these considerations result in the deterioration of the quality of the clustering.

Utilities in clustering have been studied in the past in the context of a variant called *hedonic games* [12]: clustering games with utilities and no restriction on the number of clusters. One particular utility, which has been studied extensively in the recent past, is the *modified fractional utility* [14], asserting that each node in a coalition has a payoff equal to the fraction of the other nodes in the coalition that are neighbors in the original graph. In other words, the utility of a node i is the number of the nodes in the cluster that are i's neighbors in the graph, divided by $n - 1$, the number of other nodes in the cluster. Importantly, we define *a new family of utilities* generalizing the modified fractional one, of which the simplest is the *closeness utility:* the ratio of the degree of the node in the cluster by the sum of distances from the other agents in the cluster, where distances are measured in the original graph.

1.1 Our contributions:

Overarching themes. This paper is a theoretical and experimental investigation of the interplay and trade-offs between the main objectives in network clustering: quality and welfare. We believe that this is the first work in which these important objectives of

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

clustering are considered together and contrasted with one another. *Fairness* is another important objective, which, as we shall see in passing, can be treated in similar ways.

The closeness utility. We introduce the *closeness* utility of a node i, relative to a cluster C to which it belongs: $\mathrm{cls}(i, C) = \frac{\deg_C(i)}{\sum_{j \in C} \mathrm{dist}_G(i,j)}$: the ratio of the number of nodes in C who are adjacent to i in G by the sum of all distances from i, in G, of the other nodes in C. Compared with [14]'s recently defined *modified fractional utility* $\mathrm{mfu}(i, C) = \frac{\deg_C(i)}{|C|-1}$, our utility modifies the denominator by weighing each node in the cluster by its distance from i — intuitively incentivizing more compact clusters. Another way to see the closeness utility, is as $\mathrm{mfu}(i, C)$ divided by the *average distance* to the other nodes of the cluster — an important measure of communication efficiency in a group of people, who want to communicate information to their neighbors.

Hedonic games. are a mathematical abstraction of clustering with a utility function. Importantly, they lack the constraint that the number of clusters be a small integer k, and as a result their clusters tend to be tiny (of size 1–3 in many cases) or huge (the whole graph). But what the hedonic games lack in realism viz. clustering, they make up in analytical tractability. In Section 3 we theoretically extend to the new utility the important results of the recent papers [14] and [8]: The closeness utility has small price of Pareto optimality (see Section 3 for the definitions), has simple coalition structure of Pareto optimal solutions, and its welfare maximization problem can be solved in polynomial time.

In fact, our proofs of these results parallel, and further illustrate the power of, the sophisticated mathematical techniques in the two papers mentioned above. The results hold for a much richer class of utilities generalizing the mfu utility. The closeness utility can be seen as a variant of the mfu utility, in which the $n-1$ nodes other than i in the denominator are *weighted* each by its distance from i (in mfu, all weights are one). Consider now any weighting scheme in which nodes adjacent to i have weight one, and all other nodes have weight at least two. It turns out that *the main results on the structure, price, and complexity of Pareto optimality hold for any such weighting scheme.*

Comparing utilities in clustering. We propose in Section 4 a framework for comparing utilities in the context of clustering by evaluating the quality of the clustering resulting from the behaviors induced by each utility — in this case, the conductance of the clusterings emerging as Nash equilibria. In other words, we postulate that one measure of goodness of a utility is the degree to which it encourages, and thus aligns better with, clustering quality. We then use this framework to establish experimentally the superiority, in this sense, of the new utility over other utilities in the literature.

The welfare-quality trade-off. Finally, no matter what definition of utility one adopts, it is very likely to be at loggerheads with quality. We believe this is an important trade-off, and in Section 5 we offer a useful algorithmic tool for exploring it: Given a point in the trade-off — in particular, the clustering that optimizes quality with no attention to utility — we show how to construct in polynomial time the next point in the trade-off. That is, to discover the perturbation of the clustering which is the most efficient, in that it maximizes the ratio of improvement in utility over decrease in quality. Our method employs an ancient algorithm (see [26, 34], and [20]) for finding in a doubly-weighted graph the cycle with smallest ratio of weights.

2 RELATED WORK

Our work aims to bridge theoretical foundations of strategic considerations in clustering, as studied through the idealized hedonic games, with the practical realities of algorithms such as Spectral Clustering, where one has to worry about stability (is the cluster a Nash equilibrium?) and quality (how much does the Nash constraint result in deterioration of the quality?).

A related theoretical line of work is that of hedonic games, as described in the introduction. First introduced by [12], hedonic coalitions have been studied as an organic mechanism for group formation based on individual preferences over people in the same group (e.g. [6, 7]). Recent works have studied utility functions based on the cost of clustering for algorithms like k-means and correlation clustering (e.g. [16, 17]), while [14] and [8] study Pareto optimality for utility functions based on individual preference and cluster size. Our work adds to this literature by theoretically proving a small price of Pareto optimality for the closeness utility as well as providing a polynomial time algorithm for finding optimal coalitions. Similarly, [40] study stability conditions for the modified fractional hedonic games. Our experimental work differs in two ways: (1) the number of clusters is fixed (as opposed to variable in the hedonic games setting), which opens an avenue for studying the utility of clustering algorithms on networks; and (2) our proposed utility function (closeness utility) takes into account the local neighborhood of an individual in a network, which leads to creating clusters with better conductance than other utility functions.

A large body of literature considers the issue of fairness in Machine Learning problems (e.g. [13, 21, 29]), while recent literature brings into question the problem of *utility* in applications where people are strategic, [36] showing the long-term effects of machine learning, and [23, 24, 39], and [27] analyzing social welfare in the context of classification. While [1] and [19] analyze clustering through the lens of k-means cost, and [4] through the ability to access information, the question of utility with respect to clustering algorithms remains an open one. Since clustering often results in the minimization of the number of connections between clusters (e.g. [41, 42]), the closeness utility is particularly useful in applications like facility location, market segmentation, or online grouping, which imply that connections *within* clusters are heavily used, while connections occurring between members of *different* clusters carry an extra cost. A related line of work defines the problem of fairness in the context clustering, with the works of Chierichetti et al. [10] defining fairness as balanced clusters in terms of demographics (clusters in which each demographic is proportionally represented), followed by contributions for various algorithms [2, 9], including for Spectral Clustering [11, 31].

3 HEDONIC GAMES AND THE CLOSENESS UTILITY

A *hedonic game* has n players who are nodes of a *connected* graph $G = (N, E)$. All players have the same set S of strategies over N.

Once each player i has chosen a strategy $s_i \in S$, we let $C(i)$ be the set of all players who chose the same strategy as i; we call this set the *coalition* of player i. Thus the choices of all players result in a partition of N into nonempty subsets called coalitions. Payoffs in hedonic games are calculated in a particular way, which makes them good models of clustering: After all players have chosen a strategy, and a partition $\Pi = \{C_1, \ldots, C_m\}$ has resulted, the payoff of each player i depends only on i and the coalition $C(i)$ of Π to which i belongs. Naturally, this utility function $u(i, C(i))$, mapping a node and a set to a real, will implicitly depend on G, and on i's position in it. For example, one perfectly reasonable utility is the function $D(i, C) = \deg_C(i)$, the number of nodes in $C(i) - \{i\}$ that are adjacent to i in G. Given such a partition $\Pi = \{C_1, \ldots, C_m\}$, the *social welfare* of Π, $SW(\Pi)$, is the sum over ß $\in N$ of the utility of i in Π, that is, $u(i, C(i))$.

In analyzing these games, it is a good idea to adopt a notion of stability that is stronger than the Nash equilibrium, as proposed in [14]. A partition is *welfare optimum* if there is no other partition with larger welfare. We say that a partition Π *Pareto dominates* partition Π' if no player has smaller utility in Π than it has in Π', and at least one player has larger. We call a partition *Pareto optimum* if it is not Pareto dominated by another partition. Pareto optimum partitions always exist, as the welfare optimum partition is obviously one. The *price of Pareto optimality (PPO)* of a game — proposed in [14] as a more robust analog of the price of anarchy, defined in [32] — is the largest ratio of the optimum welfare by that of a Pareto optimum partition. In the example utility D defined above, it is not hard to see that the only Pareto optimum partition is the one that has only one coalition, N (recall that G is connected).

One important utility introduced and studied in the pioneering paper [14] is the *modified fractional utility* defined as $\mathrm{mfu}(i, C) = \frac{\deg_C(i)}{|C|-1}$. That is, the degree of i in C is divided by the number of the other nodes in C. It turns out that this simple definition gives rise to a surprisingly rich theory. The following fundamental, and mathematically difficult, results were shown in [14] and subsequently in [8]. (By "clique" below we mean a completely connected graph with two or more nodes.)

Theorem 3.1. *For any graph G and for the modified fractional utility:*
(a) Each coalition in a Pareto optimum partition is either an isolated node, or a star, or a clique.
(b) A welfare optimum partition can be found in polynomial time, and there is a welfare optimum partition that has no stars.
(c) The PPO is one (that is, all Pareto optimal partitions are welfare optimal).

Our main contribution in this section is the introduction and analysis of a new utility, which we feel has a role to play in the study of clustering and hedonic games: The *closeness utility* is defined as $\mathrm{cls}(i, C) = \frac{\deg_C(i)}{\sum_{j \in C} \mathrm{dist}_G(i,j)}$, weighing each node in the denominator by its distance, in G, from i. We can prove these properties of the closeness utility:[1]

Theorem 3.2. *For any graph G and for the closeness utility:*
(a) Any coalition in a Pareto optimal partition is either an isolated node, or a star, or a clique.
(b) A welfare optimal partition can be found in polynomial time, and all welfare optimal partitions contain no stars.
(c) The PPO is at most $\frac{4}{3}$, and there are graphs for which the PPO is arbitrarily close to $\frac{4}{3}$.

Proof. Our proof of (a) follows the beautiful, and formidable, proof in [14], so we shall sketch that proof and the modifications needed. We consider a coalition (A, E) in a Pareto optimum partition. We assume that it is not a singleton, star, or clique, and then show that it can be further broken down in a way that demonstrates the partition was not Pareto optimum. We start by decomposing A into a minimum vertex cover C and the remaining independent set I. Then we define a particular flow f from I to C, that is, a function f from $I \times C$ to the nonnegative reals such that the sum of $f(i, j)$ over all $i \in I$ is one. This flow subdivides I and C into $h + 1$ disjoint subsets each $I_0, \ldots, I_h$ and $C_0, \ldots, C_h$, in increasing order of the amount of flow received by the nodes in C, such that (a) no flow goes from I_k to $C_{k'}$ with $k' < k$; (b) the ratio $|I_k|/|C_k$ is increasing with k. We shall henceforth consider each I_k and C_k. It is shown that, for each $k > 0$, I_k can be decomposed into $|C_k|$ subsets whose cardinality is within one of each other — and is therefore at most $\lceil \frac{|I_k|}{|C_k|} \rceil$ — and which induce $|C_k|$ stars in the graph, covering all of I_k and C_k and with the nodes of C_k as centers, and such that all nodes in $I_k \cup C_k$ fare no worse in the new partition, while one of them fares strictly better. To carry out this step for the closeness metric, we first need to show a basic fact about the closeness utility:

Lemma 3.1. *If a node in A can only be connected to nodes in a subset B of A, its closeness utility is at most $\frac{|B|}{2|A|-|B|-2}$.*

To prove the lemma, let d be the degree of i in the coalition (A, E); obviously $d \leq |B|$. Then the closeness of i in the partition is at most $\frac{d}{d+2(|A|-d-1)}$. The reason is that the distance of the nodes in A not connected to i must be at least 2. This is maximized when d attains its maximum value, $|B|$.

To continue the proof of the theorem, let $C^{\geq k}$ be the union of all C_l's with $l \geq k$; we know that node $i \in I_k$ can only be connected to this set. Thus, by the Lemma, $\mathrm{cls}(i, A)$ is at most

$$\frac{|C^{\geq k}|}{2|A| - |C^{\geq k}| - 2} \leq \frac{|C^{\geq k}|}{|C^{\geq k}| + 2|I^{\geq k}| - 2} \leq \frac{1}{1 + 2\frac{|I^k|-1}{|C^k|}} \leq \frac{1}{2\lceil \frac{|I^k|}{|C^k|} \rceil - 1},$$

where the next to last inequality takes into account the fact that the ratios are increasing with k. This last quantity is precisely the closeness utility of a leaf of the new star, which is what all elements of I_k will become. (The center of the star will have utility one, the largest possible.) For the requirement that at least one node does strictly better, notice that the first inequality is not strict only when $k = 1$ and $A = C_1 \cup I_1$, in which case it is easy to find an element of I_1 that does better.

It remains to take care of C_0 and I_0 (which takes up most of the 7-page proof). It can be shown that this part of A has a special structure (illustrated in Figure 3 of [14]): It contains a matching M within C_0 and a matching M' between the unmatched nodes of C_0 and nodes of I_0, plus another matching between the remaining

[1] In fact, it is not hard to check that our proof establishes that these results hold for a larger class of utilities, where the nodes of the cluster are weighted by weighs $w(d)$ depending on their distance d from i, as long as $1 = w(1) \leq w(2) = 2 \leq w(3), w(4), \ldots$

("free") nodes of I_0, if any, and nodes of C_0 — and in addition, no two of these latter nodes of C_0 are matched in M. If there are no free nodes, this means that $C_0 \cup I_0$ has a complete matching, a coalition giving all nodes utility one, the maximum. But what if there are free nodes? They can certainly be accommodated in a star with two leaves, which gives them a utility of $\frac{1}{3}$ (in the proof of [14] this quantity is $\frac{1}{2}$). But what if one of them already has a higher utility in A? A complicated case analysis shows that, in that case, this node can become a part of a triangle, again attaining a utility of one and completing the proof of (a).

To prove part (b), we know the welfare optimum coalition consists of isolated points, cliques, and stars. It is easy to see that there can be no stars with two or more leaves, because any such star can improve the total welfare by shedding all leaves but one. Hence it is all cliques and isolated points. Now, any clique with an even number of points can be decomposed into 2-cliques (edges, that is) with the same welfare. And any clique with an odd number of points can be similarly decomposed into 2-cliques and 3-cliques. We conclude that to compute the welfare maximum is tantamount to seeking the set of disjoint edges and triangles that cover the most nodes of the graph — and this can be done in polynomial time through the ingeniously simple algorithm of [22].

Finally, for (c) any Pareto optimal matching can be improved in welfare by shedding all but one leaf from every star, and by a result implicit in [8] (Theorem 6.2) and based on Edmonds's 1965 characterization of optimality in matching, the resulting cover by edges and triangles will be optimum. The increase in welfare is by at most a factor of $\frac{4}{3}$, since stars with n leaves have welfare $\frac{3n-1}{2n-1}$ while the optimum 2-clique has welfare 2. If the graph is such a star, the lower bound follows for large n. □

We present, for completeness, a couple of additional results in the context of Nash equilibria. Firstly, we note that without randomness, a finite improvement path does not always exist:

Proposition 3.1. *There exists a graph G which does not have the finite improvement path property under best response dynamics.*

Proof. Similar to the example given for the mfu [40], a best response path may cycle in the presence of a graph formed by a star with vertices $\{1, 2, 3, 4, 5\}$ and center $\{1\}$ and a leaf node $\{6\}$ attached to $\{1\}$. Then, the sequences $1, 2, 5, 6, 1, 2, 5, 6$ cycles back to the initial partition given the closeness utility (which penalizes non-neighbors in a cluster even more than the mfu). We note that a simultaneous best response for all nodes leads to infinite cycling as well, since starting with the partition $\{1, 6\}$ and $\{2, 3, 4, 5\}$ will lead to all nodes continuously switching their group. □

However, noting that the grand coalition is always a Nash equilibrium for the closeness utility, adding randomness in an improvement path dynamics always leads to a Nash equilibrium empirically.

In theory, one common measure of equilibrium quality is the Price of Anarchy (PoA), which defines the worst ratio of total social welfare of an optimum outcome and am equilibrium outcome (in this case, Nash equilibrium). We can show, similarly to the proofs in [40], that the price of anarchy grows quadratically in the number of agents (compared to linearly for mfu):

Lemma 3.2. *For any weighted graph with non-negative edge-weights G and the closeness utility:*

$$PoA(G, closeness) \leq \frac{n(n-1)}{2}. \tag{1}$$

Proof. Similar to the modified fractional utility [40], the lower bound of a node i's utility in an equilibrium partition is $\frac{w^i_{max}}{1+2+\cdots+n-1}$, where w^i_{max} is the maximum weight between i and other nodes (since we are at equilibrium), while the utility of i in an optimal partition is upper bounded by w^i_{max} (when the distance to the other nodes in its cluster is 1). The lemma follows. We note that the worst case is attained by the path graph with an even number of nodes, for which $PoA(G, closeness) = \Omega(n^2)$, as the optimal partition is in pairs of nodes (thus everyone has utility 1), while the worst partition is the grand coalition (in which everyone has utility $\sim 1/n^2$). □

4 EVALUATING UTILITIES THROUGH EQUILIBRIA

Hedonic games are useful arenas for studying the theoretical properties of utilities, but are poor models of clustering as the organic formation of groups through hedonic games often leads to partitions of 2-3 individuals or a partition consisting of the entire network.

Here we propose a principled empirical methodology for comparing utilities in the context of clustering: A utility defines a *clustering game,* in which players choose one of k clusters and receive the resulting utility. Suppose we can sample the pure Nash equilibria of this game; then a utility is *good* — that is to say, well aligned with the clustering context — if these equilibria are of high quality, that is, conductance. The main difference with the previous section is that we consider Nash equilibria instead of Pareto optimal partitions, and, most importantly, the number k of clusters is fixed; note that, since both in mfu and closeness the utility of an isolated node is 0, no node has an incentive to create its own new coalition, so the initial partition defines the number of clusters present at equilibrium.[2]

In terms of methodology, we find Nash equilibria through a randomized, parallel variant of best response dynamics (which we feel is an algorithmic maneuver of more general applicability and interest). We start from a random clustering with k clusters, and repeatedly identify the players who have an incentive to defect to a different coalition. We implement a (uniformly) random subset of these defections at each time. This speeds up convergence considerably (for all utilities, data sets and starting points, between 10 and 20 iterations are needed) and, while cycling is in principle possible, we find empirically that this process always, in thousands of experiments, leads to a Nash equilibrium.

In our experiments we analyze the following data:

(1) The APS citation network from [35] with 1,853 nodes which represent papers written in two main topics: Classical Statistical Mechanics (CSM), representing 37.5% of the papers, and Quantum Statistical Mechanics (QSM), representing the rest of 62.5% of the papers. As [35] analyzes, the dataset has high homophily;

[2] Unless of course all nodes of a cluster want to migrate, and our randomization (see the next paragraph) chooses all of them to do so; in thousands of runs, this has never come close to happening.

(2) A Facebook dataset from [38], of 4, 038 nodes of true Facebook friends; this data has several (anonymized) features, from which we use gender to denote a sensitive attribute. While gender itself is anonymized, we get that a majority community of one gender represents approximately 60% of the network, and the other represents 40% of the network.[3] The network is homophilic.

(3) A highschool friendship network from [37] (also used in [31]), with 127 nodes in its main connected component represented by students who self-identify as male of female. The majority are female (60%), and the network is homophilic.

At equilibrium, the closeness utility (in purple) leads to better clusters in terms of conductance than mfu (in blue), as Figure 1 shows, for a varied number of clusters (ranging from 2 to 8, with similar results for higher number of clusters). In particular, both the closeness utility and mfu do not stray far from the conductance achieved through spectral clustering (in black) and they reach equilibrium within 10-20 randomized best-response steps. In other words, we establish empirically that there is an alternative mechanism to spectral clustering, which allows people to make choices based on their utility, and can achieve quickly a partition of good conductance. In all experiments, results are averaged over 50 iterations, illustrated with error bars, and we note that both utilities partition the graph in approximately equal clusters.

Beyond evaluating in terms of quaality (as conductance), one may ask how the equilibrium solutions fare in comparison to Spectral Clustering in terms of utility. Figure 2 shows that Spectral Clustering (in black solid lines) achieves slightly better average utility than the equilibrium partitions (in black dashed lines), while mfu more closely aligns with Spectral Clustering (sometimes even better). Note that the initial partition defines the equilibrium reached (e.g., if starting from the partition given by Spectral Clustering, one would obtain better conductance at equilibrium than starting from random partitions).

Note: In this section we have been arguing that the closeness utility is superior to mfu in many respects. Does the fact that mfu has better PPO and price of anarchy (1 vs. $\frac{4}{3}$, and $O(n)$ vs. $O(n^2)$, calculation in the Appendix) suggest otherwise? We believe that the "price" criteria, while informative in general, are misleading when used as measures of quality of a utility. The reason a utility such as closeness is good is, intuitively, because it penalizes poor features, such as long paths in the clusters. Yet, if such features manage, very rarely, to survive in stable solutions, the corresponding price will necessarily be larger.

5 THE TRADE-OFF ALGORITHM

Finally, having contemplated the interplay between quality and utility in clustering, we provide an algorithmic tool for navigating such trade-offs: Suppose that we have a clustering $C = \{C_1, \ldots, C_k\}$ and two ways of evaluating it, say its quality $q(C)$ and its utility $f(C)$.

We are assuming that each of the functions q and f is the sum, over all clusters C_i, of simpler functions of the set of nodes in each cluster. Note that all quality, fairness, and utility functions we have considered are of this form. Define an *elementary change* in a cluster C_i to be a node leaving the cluster, or a node joining the cluster, or the combination of the two, a node replacing another in the cluster. Next, a *change* in the clustering C is the clustering C' which results from a set of elementary changes defined by a sequence of nodes $v_1, \ldots, v_m$ with $m \geq 1$, all in different clusters — except that possibly the first and the last nodes are allowed to be in the same cluster. In the resulting clustering C', each v_i, where $i < m$, leaves its cluster and joins the cluster of v_{i+1}. Notice that a change is a sequence of elementary changes. Further, the difference in the functions f and q before and after a change is the sum of the differences of all the elementary changes which constitute the change. Also, these differences in f and q before and after each elementary change is easy to compute, for all evaluation functions considered in this paper. Denote the set of all changes of clustering C by changes(C).

Finally, suppose that the sum $q(C)+\alpha \cdot f(C)$ is optimum for some $\alpha \geq 0$. We say that C is *a point in the trade-off* between f and q. For concreteness, we shall be assuming that C is the optimum clustering in quality, and $\alpha = 0$; the case of general $\alpha > 0$ is handled similarly. In this section we present an efficient algorithm for finding the *optimum change of* C, that is the change C' which results in the largest ratio of increment in utility over the decrement in quality:

$$C' = \arg \max_{C' \in \text{changes}(C)} \frac{f(C') - f(C)}{q(C) - q(C')} \quad (2)$$

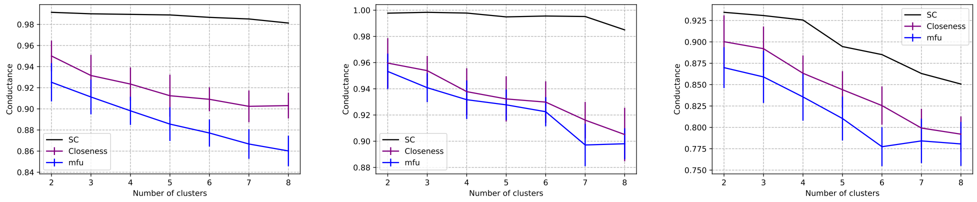
Theorem 5.1. *Given a point in the trade-off C, its optimum change C' can be computed in polynomial time.*

Proof. We sketch the proof for the case in which C is the clustering of optimum quality, that is, $\alpha = 0$; the general case is addressed in the Appendix. We employ an algorithm discovered in the 1960's by Eugene Lawler (Karp [26], Lawler [34], see Golitschek [20] for an exposition) for finding, in a doubly weighted graph, a cycle that maximizes the ratio of the weights (say, revenue over distance traversed). Given C, we construct a doubly weighted directed graph D whose weights capture the loss in quality and the gain in utility in every change from C (for a detailed description, see the Appendix):

- create a directed edge (u, v) for all u and v in different clusters; this edge will have a double weight (f_{uv}, q_{uv}), where f_{uv} is the utility gain by moving node u to the cluster that v belongs to, and another weight q_{uv} defining the cluster quality loss by replacing node v by node u in its cluster.
- create two nodes C_i and C_i' for each cluster C_i, and an edge (C_i, u) for any $u \in C_i$, with a double weight $(f_{C_i u}, q_{C_i u})$, which are the differences in utility (or fairness) and clustering quality, respectively, if node u moves out of cluster C_i. Also, an edge (u, C_i') for any $u \notin C_i$, of a double weight $(f_{uC_i'}, q_{uC_i'})$, which are the differences in utility and clustering quality, respectively, if node u moves to cluster C_i.
- finally, define another node called s, for "start", and create edges sC_i and $C_i's$ for every cluster C_i, in both cases with both weights zero; these edges turn any non-cyclic change of the clustering into a cycle (recall that Lawler's algorithm finds the best cycle, not path).

To complete the proof of the theorem, it suffices to check that every directed cycle in the new graph D corresponds to a change of the clustering C, and the two total weights of the cycle correspond

[3] This feature is embedded in the dataset as an anonymized binary feature and does not include, unfortunately, non-binary gender representations.

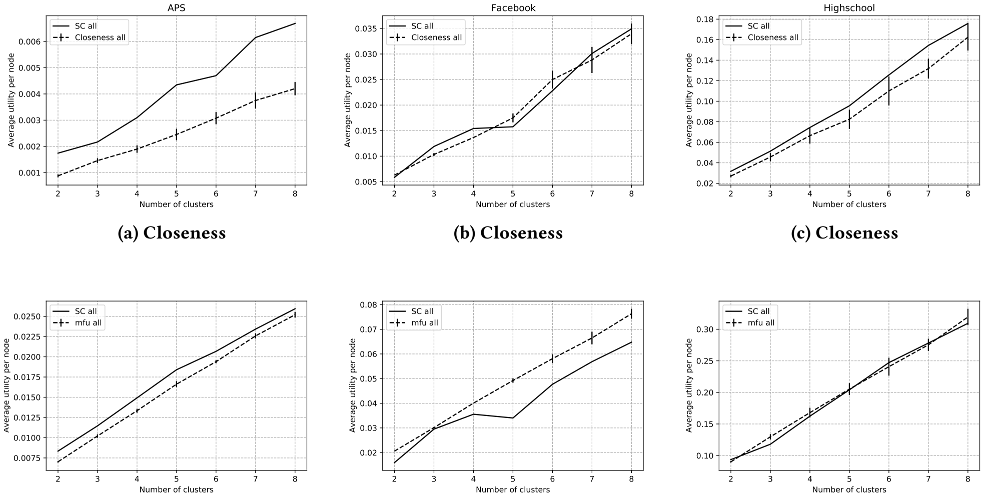


(a) APS dataset

(b) Facebook dataset

(c) Highschool dataset

Figure 1: Conductance comparison for APS, Facebook, and Highschool datasets between spectral clustering and the equilibrium partition, for a varied number of clusters.



(a) Closeness

(b) Closeness

(c) Closeness

(d) mfu

(e) mfu

(f) mfu

Figure 2: Average utility for the each dataset (by column), for closeness utility and mfu (by row), computed for Spectral Clustering (SC) and equilibrium solutions.

to the total differences in utility and quality of the change; and vice-versa. Thus, finding the cycle of graph D with the largest ratio of the two weights computes the optimum change of C. There is one important thing to check: Lawler's algorithm requires that the metric in the denominator (in our case, quality decrement) *does not have a non-positive cycle.* This holds, because we started from the quality-optimum clustering, and any negative cycle would improve the quality. (There may be zero cycles because of ties in quality between clusterings, but this can be taken care of by small perturbations in the weights.) □

While this algorithm correctly computes the optimum change from any point that optimizes the trade-off for some α, because of nonlinearities in the way the functions f and q are computed, the resulting change is *not* guaranteed to be an optimum point for some other α — such failure is manifested by negative cycles in the denominator. Still, empirically, this algorithm, together with heuristics for eliminating negative cycles, helps trace the trade-off curve in real problems. However, more research is needed for automating this process. The algorithm takes polynomial time, as noted further below.

We detail the algorithms described in Algorithms 1, 2, and 3. We reduced the problem of finding the next point in the trade-off to finding a cycle in a doubly-weighted graph that represent elementary changes. As an example for how this graph creates elementary changes in clustering, a cycle $u - v - w - z - u$ means "u moves in the cluster of v, v in the cluster of w, …, z in the cluster of u". A cycle involving s, such as $s - u - v - w - C$ means "u moves to the cluster of v, v to the cluster of w, w moves to C" (a move that changes the cardinalities of two clusters, namely of C and of the cluster of u, by 1). The weights in the edges reflect this: f_{uv} is the decrease in utility in v's cluster if u moves to the cluster of v and v leaves its cluster. Similarly, q_{uv} is the decrease in quality/increase in cost in the cluster of v if u replaces v in v's cluster; f_{uC} is the decrease in utility if u is added to C, and q_{uC} the increase in cost.

In solving Problem A, Golitschek [20] shows that for a doubly weighted graph G with weights $(f_{uv})_{(u,v)\in E(G)}$ and $(q_{uv})_{(u,v)\in E(G)}$, the problem of finding an optimal cycle with minimal ratio f_{uv}/q_{uv} for $(u, v) \in E(G)$ is equivalent to finding a negative cycle in another related graph. This other related graph by choosing a variable M

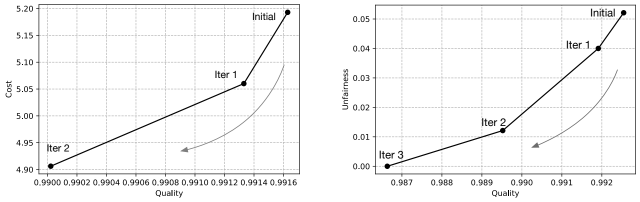
and creating a graph G' with weights $f_{uv} - M \cdot q_{uv}$. Finding a negative cycle in G' then means that M is larger than the optimal cycle value found, and smaller otherwise. The goal is to find M such that the negative cycle that we find becomes equal to 0. Because the clustering is optimal, there are no negative cycles in the $(q_{uv})_{u,v}$ weight (and zero cycles only by coincidence, which we can take care by perturbing a little one or two q_{uv} values).

Now we can apply the algorithmic idea from above, implementing a negative cycle finder through the Shortest Path Fastest Algorithm implementation, which is a faster implementation of the Bellman-Ford Algorithm ([5, 18]), obtaining a running time of $O(m + n)$, where m is the number of edges in the graph. The main steps are itemized below:

(1) Input Graph G(V,E) (e.g. generated from a Stochastic Block Model with 50 nodes, a probability vector, and a number of clusters $k = 5$, as described below).
(2) Run Spectral Clustering on G.
(3) Generate graph G_{new} with doubly-weighted edges f_{uv} and q_{uv}.
(4) Assert that G_{new} has no negative $q-$cycles.
   - if G_{new} has some cycles in the $t-$weight of weight 0, adjust all q_{uv}'s by $+\frac{f_{uv}}{M+\epsilon}$ for some $\epsilon << 1$ (only for $\alpha > 0$).
(5) Initialize M.
(6) Generate graph G_M from G_{new} with weights $f_{uv} - M \cdot q_{uv}$
(7) Run SPFA(G_M):
   - If there is a negative cycle, set $M/= 2$ and repeat SPFA
   - If there is no negative cycle, set $M* = 2$ and repeat SPFA
   - If cycle weight is 0, we found the optimal cycle and can compute the utility-conductance trade-off (and perhaps repeat with the new assignment of nodes to clusters by following this cycle)
   - After finding two values of M (a high and a low) for which there exists and not exists, respectively, a negative cycle, we run binary search using SPFA to find the best M that gets us a cycle of weight close to 0. The termination condition can be chosen by the user (say, the δ update of M in binary search is smaller than ϵ, for some $\epsilon << 1$.
(8) Perform changes according the best cycle we find from previous step, c, which gets us a new clustering, and compute new utility and clustering quality.

The pseudocode is presented as follows: step 1 is described in detail below; steps 2 and 3 are computed in Algorithm 1; steps 5 – 7 are computed in Algorithm 3; SPFA is defined in Algorithm 2. For taking care of negative cycles at subsequent iterations (for $\alpha > 0$), we may perturb the q_{uv} weights by a small quantity in the direction of the current slope (see step (4), which works in practice for cases with low non-linearity. Further work is needed for a complete generalization. Note that when the clustering algorithm used is not optimal, negative cycles in the q-weight may appear initially as well; sometimes, this can be an issue with Spectral Clustering, for example, as it does not find a global optimum in all cases. However, if an initial negative cycle in the q-weight is found initially, it can be 'fixed' by following the changes in the cycle for a better clustering in terms of quality.

In terms of complexity, Lawler's algorithm takes L iterations, where L is the number of bits in all the weights of the graph D, and each iteration seeks a negative-weight cycle in a graph with the same nodes and edges as D. Also, notice that D has $O(n)$ nodes. Hence, the total complexity is $O(n^3 B \log n)$, where B, the number of precision bits we require, can be taken to be a constant. By using a fast implementation of a negative cycle finder algorithm,[4] we can expect the time required to be closer to n^2.



(a) Closeness utility **(b) Statistical parity**

Figure 3: Trade-off curve between clustering quality and closeness utility (left) or statistical parity (right), for a graph generated from the Stochastic Block Model with 50 nodes and two homophilic communities. The initial point is in the most up right corner and subsequent iterations (Iter 1,2) trace the boundary of the trade-off, in the direction of the arrow. Quality decreases in subsequent iterations.

As an example of how the trade-off curve would look like for a case with low non-linearity, Figure 3 shows the trade-off curve that our proposed algorithms trace from the original clustering given by Spectral Clustering (Initial) through subsequent iterations. We can trace the boundary for different definitions of utility or fairness, for example for the closeness utility (left) or statistical parity (right), opening an avenue for connecting utility, quality, and fairness, as described by statistical parity constraints in recent works regarding clustering [10, 30].

6 DISCUSSION

Our paper is a theoretical and experimental exploration of the ways in which strategic behavior affects the quality of clustering. The closeness utility has several positive theoretical properties such as efficient maximization and a low price of Pareto optimality; in practice, it results in relatively equal clusters that compete in quality with Spectral Clustering and fare better than competing utilities.

Our polynomial time algorithm for finding an improvement of optimal ratio of utility improvement over quality decrease can be particularly useful for exploring intermediate levels of utility as well as generalizing for metrics beyond the closeness utility. This is especially relevant for connecting with fairness notions, such as statistical parity or balance, and generalizes for any metric of fairness or utility. All the code and data are available on an anonymized Github repository, at this link.

Several open questions emerge from this work: Generalize the trade-off algorithm to trace the entirety of the trade-off curve; establish a theoretical connection between clustering through equilibrium dynamics and utility functions and traditional clustering

[4] https://konaeakira.github.io/posts/using-the-shortest-path-faster-algorithm-to-find-negative-cycles.html which is a faster implementation of the Bellman-Ford Algorithm [5, 18]

optimization; and use the framework proposed here for understanding the root cause of bias in clustering and for developing ways to mitigate it when multiple metrics of fairness or utility are at play.

Naturally, welfare depends not just on the utility, but also on the way the utility is aggregated; here we mostly consider average utility, but of course the variance, or other measures of inequality such as the Gini coefficient, would also be of great interest.

ACKNOWLEDGEMENTS

CHP's research was partially supported by NSF awards CCF1763970 and CCF1910700, by a research contract with Softbank, and a grant from CAIT (Columbia Center for AI Research).

REFERENCES

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

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

[3] Hélio Almeida, Dorgival Guedes, Wagner Meira, and Mohammed J Zaki. Is there a best quality metric for graph clusters? In *Joint European conference on machine learning and knowledge discovery in databases*, pages 44–59. Springer, 2011.

[4] Hannah C Beilinson, Nasanbayar Ulzii-Orshikh, Ashkan Bashardoust, Sorelle A Friedler, Carlos E Scheidegger, and Suresh Venkatasubramanian. Clustering via information access in a network. *arXiv preprint arXiv:2010.12611*, 2020.

[5] Richard Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1): 87–90, 1958.

[6] Anna Bogomolnaia and Matthew O Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201–230, 2002.

[7] Robert Bredereck, Edith Elkind, and Ayumi Igarashi. Hedonic diversity games. *arXiv preprint arXiv:1903.00303*, 2019.

[8] Martin Bullinger. Pareto-optimality in cardinal hedonic games. In *AAMAS*, pages 213–221, 2020.

[9] Xingyu Chen, Brandon Fain, Charles Lyu, and Kamesh Munagala. Proportionally fair clustering. *arXiv preprint arXiv:1905.03674*, 2019.

[10] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems*, pages 5029–5037, 2017.

[11] Ian Davidson and SS Ravi. Making existing clusterings fairer: Algorithms, complexity results and insights. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3733–3740, 2020.

[12] Jacques H Dreze and Joseph Greenberg. Hedonic coalitions: Optimality and stability. *Econometrica: Journal of the Econometric Society*, pages 987–1003, 1980.

[13] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[14] Edith Elkind, Angelo Fanelli, and Michele Flammini. Price of pareto optimality in hedonic games. *Artificial Intelligence*, 288:103357, 2020.

[15] Scott Emmons, Stephen Kobourov, Mike Gallant, and Katy Börner. Analysis of network clustering algorithms and cluster quality metrics at scale. *PloS one*, 11 (7):e0159161, 2016.

[16] Michal Feldman and Ophir Friedler. A unified framework for strong price of anarchy in clustering games. In *International Colloquium on Automata, Languages, and Programming*, pages 601–613. Springer, 2015.

[17] Moran Feldman, Liane Lewin-Eytan, and Joseph Naor. Hedonic clustering games. *ACM Transactions on Parallel Computing (TOPC)*, 2(1):1–48, 2015.

[18] Lester Randolph Ford and Delbert Ray Fulkerson. *Flows in networks*. Princeton university press, 2015.

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

[20] M v Golitschek. Optimal cycles in doubly weighted graphs and approximation of bivariate functions by univariate ones. *Numerische Mathematik*, 39(1):65–84, 1982.

[21] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. *arXiv preprint arXiv:1610.02413*, 2016.

[22] Pavol Hell and David G Kirkpatrick. Packings by cliques and by finite families of graphs. *Discrete Mathematics*, 49(1):45–59, 1984.

[23] Lily Hu and Yiling Chen. Fair classification and social welfare. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 535–545, 2020.

[24] Lily Hu, Nicole Immorlica, and Jennifer Wortman Vaughan. The disparate effects of strategic manipulation. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 259–268, 2019.

[25] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[26] Richard M Karp. A characterization of the minimum cycle mean in a digraph. *Discrete mathematics*, 23(3):309–311, 1978.

[27] Jon Kleinberg and Manish Raghavan. How do classifiers induce agents to invest effort strategically? *ACM Transactions on Economics and Computation (TEAC)*, 8 (4):1–23, 2020.

[28] Jon Kleinberg, Christos Papadimitriou, and Prabhakar Raghavan. Segmentation problems. *Journal of the ACM (JACM)*, 51(2):263–280, 2004.

[29] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

[30] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-center clustering for data summarization. *arXiv preprint arXiv:1901.08628*, 2019.

[31] Matthäus Kleindessner, Samira Samadi, Pranjal Awasthi, and Jamie Morgenstern. Guarantees for spectral clustering with fairness constraints. *arXiv preprint arXiv:1901.08668*, 2019.

[32] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. *Computer science review*, 3(2):65–69, 2009.

[33] RJ Kuo, LM Ho, and Clark M Hu. Integration of self-organizing feature map and k-means algorithm for market segmentation. *Computers & Operations Research*, 29(11):1475–1493, 2002.

[34] Eugene L Lawler. Optimal cycles in doubly weighted directed linear graphs. In *Proc. Int'l Symp. Theory of Graphs*, pages 209–232, 1966.

[35] Eun Lee, Fariba Karimi, Claudia Wagner, Hang-Hyun Jo, Markus Strohmaier, and Mirta Galesic. Homophily and minority-group size explain perception biases in social networks. *Nature human behaviour*, 3(10):1078–1087, 2019.

[36] Lydia T Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed impact of fair machine learning. In *International Conference on Machine Learning*, pages 3150–3158. PMLR, 2018.

[37] Rossana Mastrandrea, Julie Fournet, and Alain Barrat. Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PloS one*, 10(9):e0136497, 2015.

[38] Julian J McAuley and Jure Leskovec. Learning to discover social circles in ego networks. In *NIPS*, volume 2012, pages 548–56. Citeseer, 2012.

[39] Smitha Milli, John Miller, Anca D Dragan, and Moritz Hardt. The social cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 230–239, 2019.

[40] Gianpiero Monaco, Luca Moscardelli, and Yllka Velaj. Stable outcomes in modified fractional hedonic games. *Autonomous Agents and Multi-Agent Systems*, 34(1): 1–29, 2020.

[41] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14:849–856, 2001.

[42] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Algorithm 1: Reduction to negative cycle algorithm

```
Result: Compute a doubly-weighted graph $G_{new}$
        representing elementary changes in clusters, in
        utility or fairness and cost as weights.
Input: graph $G$, number of clusters $k$, fairness function $f(\cdot)$,
  quality function $c(\cdot)$, algorithm to optimize $q(\cdot)$ (e.g.
  spectral clustering as SC), initial $M$;
Compute clusters $C = (C_1, C_2, \cdots C_k) \leftarrow SC(G, k)$ ;
Compute utility or fairness of $C$ as $f(C)$ ;
Compute quality of $C$ as $c(C)$;
Initialize digraph $G_{new}$ with nodes from $G.nodes()$ ;
for $u, v \in G.nodes()$ do
    if $C(u) \neq C(v)$ then
        Denote $C'$ a clustering where $C'(u) = C(v)$ ($u$ has
          replaced $v$ in $v$'s cluster) ;
        Compute $f(C')$ and $q(C')$ ;
        Compute $f_{uv} = f(C') - f(C)$ and
          $q_{uv} = q(C') - q(C)$ ;
        $G_{new}$.add_edge(u,v,weight$_1$ =f$_{uv}$, weight$_2$ =q$_{uv}$) ;
    end
end
Add nodes $u_{|V(G)+i|}$, for $i = \overline{1,k}$, associated to each cluster
  for when a node joins another cluster;
for $u \in G.nodes()$ and $i = \overline{1,k}$ do
    if $C(u) \neq i$ then
        Denote $C'$ a clustering where $C'(u) = i$ ($u$ has
          moved to cluster $i$) ;
        Compute $f(C')$ and $q(C')$ ;
        Compute $f_{uv} = f(C') - f(C)$ and
          $q_{uv} = q(C') - q(C)$ ;
        $G_{new}$.add_edge(u,u$_{|V(G)+i|}$, $weight_1$ =f$_{uv}$,
          weight$_2$ =q$_{uv}$) ;
    end
end
Add nodes $u_{|V(G)+k+i|}$, for $i = \overline{1,k}$, associated to each
  cluster for when a node leaves its own cluster;
for $u \in G.nodes()$ do
    Denote $C'$ a clustering where $C'(u) \neq i$ ($u$ has been
      removed from its cluster) ;
    Compute $f(C')$ and $q(C')$ ;
    Compute $f_{uv} = f(C') - f(C)$ and $q_{uv} = q(C') - q(C)$ ;
    $G_{new}$.add_edge(u$_{|V(G)+k+i|}$, $u$, $weight_1$ =f$_{uv}$,
      weight$_2$ =q$_{uv}$) ;
end
Add a start node 'start'. ;
for $u \in G.nodes()$ do
    Denote $C'$ a clustering where $C'(u) \neq i$ ($u$ has been
      removed from its cluster) ;
    Compute $f(C')$ and $q(C')$ ;
    Compute $f_{uv} = f(C') - f(C)$ and $q_{uv} = q(C') - q(C)$ ;
    $G_{new}$.add_edge('start',u,weight$_1$ =f$_{uv}$, weight$_2$ =q$_{uv}$) ;
end
Return $G_{new}$
```

Algorithm 2: Shortest Path Fastest Algorithm

```
Result: Define the shortest path fastest algorithm; return a
        tracing array to find the negative cycle, if it exists.
Function SPFA($G$):
    Initialize $length$ = {}, $dis$ = {}, $pre$ = {}, and $queue$ = []
     for $v$ in $G.nodes()$ do
        $length[v] = 0$
        $dis[v] = 0$
        queue.append(v)
    end
    while $len(queue) > 0$ do
        u = queue.pop(0)
        for $(u, v) \in G.edges()$ do
            if $dis[u] + G[u][v]['weight'] < dis[v]$ then
                $pre[v] = u$
                $length[v] = length[u] + 1$
                if $length[v] == len(G.nodes())$ then
                    Return v,pre,"negative cycle detected"
                end
                $dis[v] = dis[u] + G[u][v]['weight']$
                if $v \notin queue$ then
                    queue.append(v)
                end
            end
        end
    end
    Return "no negative cycle detected"
```

Algorithm 3: Perform binary search to find the best M that obtains a cycle of weight close to 0, using SPFA

```
Result: $M$ as a product of binary search
Input: graph $G_{new}$, $M$, $M_{high}$, $M_{low}$
while $\delta_M$ > 10e-10 do
    if $M_{high} > M_{low}$ then
        $M = (M_{high} + M_{low})/2$
    end
    Initialize digraph $G_M$ with nodes from $G_{new}.nodes()$
    for $e \in G_{new}.edges()$ do
        G$_M$.add_edge($e$, weight= f$_{uv}$ $- M \cdot q_{uv}$)
    end
    if $SPFA(G_{fq}) \rightarrow$ negative cycle then
        $delta_M = M - M_{low}$
        $M_{high} = M$
    else if $SPFA(G_{new}) \rightarrow$ no negative cycle then
        $delta_M = M_{high} - M$
        $M_{low} = M$
end
Return $M$
```