This is an author produced version of a paper published in Proceedings of the 8th International Conference on Algorithms and Complexity. This paper has been peer-reviewed but does not include the final publisher proof-corrections or journal pagination.

Citation for the published paper:

Fabijan, Aleksander; Nilsson, Bengt J.; Persson, Mia. (2013). Competitive Online Clique Clustering. Proceedings of the 8th International Conference on Algorithms and Complexity, issue 8, p. null

URL: https://doi.org/10.1007/978-3-642-38233-8_19

Publisher: Springer

This document has been downloaded from MUEP (<https://muep.mah.se>) / DIVA (<https://mau.diva-portal.org>).

Competitive Online Clique Clustering

Aleksander Fabijan¹, Bengt J. Nilsson², and Mia Persson²

¹ Faculty of Computer and Information Science, University of Ljubljana, Slovenia.

`Aleksander.Fabijan@mf.uni-lj.si`

² Department of Computer Science, Malmö University, Sweden.

`Bengt.Nilsson.TS@mah.se`, `Mia.Persson@mah.se`

Abstract. Clique clustering is the problem of partitioning a graph into cliques so that some objective function is optimized. In online clustering, the input graph is given one vertex at a time, and any vertices that have previously been clustered together are not allowed to be separated. The objective here is to maintain a clustering the never deviates too far in the objective function compared to the optimal solution. We give a constant competitive upper bound for online clique clustering, where the objective function is to maximize the number of edges inside the clusters. We also give almost matching upper and lower bounds on the competitive ratio for online clique clustering, where we want to minimize the number of edges between clusters. In addition, we prove that the *greedy* method only gives linear competitive ratio for these problems.

1 Introduction

The correlation clustering problem and its different variants have been extensively studied over the past decades; see e.g. [2, 5, 6]. Several objective function are used in the literature, e.g., maximize the number of edges within the clusters plus the number of non-edges between clusters (MaxAgree), or minimize the number of non-edges inside the clusters plus the number of edges outside them (MinDisagree). In [2], Bansal et al. show that both the minimization (minimizing the number of disagreement edges) and the maximization (maximizing the number of agreement edges) versions are in fact NP-hard. However, from the point of view of approximation the maximization and minimization versions differ. In the case of maximizing agreements this problem actually admits a PTAS whereas in the case of minimizing disagreements it is APX-hard. Several efficient constant factor approximation algorithms are proposed when minimizing disagreements [2, 5, 6] and maximizing agreements [5].

Other measures require that the clusters are cliques, complete subgraphs of the original graph, in which case we can maximize the number of edges inside the cluster or minimize the number of edges outside the clusters. These measures give rise to the maximum and minimum edge clique partition problems (Max-ECP and Min-ECP for short) respectively; the computational complexity and approximability of the aforementioned problems have attracted significant attention recently [7, 9, 11], and they have numerous applications within the areas of gene expression profiling and DNA clone classification [1, 3, 8, 11].

In this paper, we consider the online variant of clique clustering where the vertices arrive one at a time, i.e. the input sequence is not known in advance. Specifically, upon an arrival of a new vertex, it is either clustered into an already existing cluster or forms a new singleton cluster. In addition, existing clusters can be merged together. The merge operation in an online setting is irreversible, once vertices are clustered together, they will remain so, and hence, a bad decision will have significant impact on the final solution. This online model was proposed by Charikar et al. [4] and has applications in information retrieval.

Our results. We investigate the online Max-ECP and Min-ECP clustering for unweighted graphs and we provide upper and lower bounds for both these versions of clique clustering. Specifically, we consider the natural greedy strategy and prove that it is not constant competitive for Max-ECP clustering but has a competitive ratio that is at best inversely proportional to the number of vertices in the input. We prove that no deterministic strategy can have competitive ratio larger than $1/2$. We further give a strategy for online Max-ECP clustering that yield constant competitive ratio. For Min-ECP clustering, we show a lower bound of $\Omega(n^{1-\epsilon})$, for any $\epsilon > 0$, for the competitive ratio of any deterministic strategy. The greedy strategy provides an almost matching upper bound since greedy belongs to the class of *maximal* strategies and these are shown to have competitive ratio $O(n)$. See Table 1 for a summary of our results.

Table 1. Summary of our results.

<i>Problem</i>	<i>Lower Bound</i>	<i>Upper Bound</i>
Greedy Max-ECP	$2/(n-2)$	$1/(n-2)$
Max-ECP	$1/2$	0.032262
Greedy Min-ECP	$(n-2)/2$	$2n-3$
Min-ECP	$n^{1-\epsilon}/2$	

2 Preliminaries

We begin with some notation and basic definitions of the Max-ECP and Min-ECP clustering problems. They are defined on an input graph G , where $G = (V, E)$ with vertices V and edges E . Hence, we wish to find a partitioning of the vertices in V into clusters so that each cluster induces a clique in the corresponding subgraph of G . In addition, we want to optimize some objective function associated to the clustering. In the Max-ECP case, this is to maximize the total number of edges inside the clusters (agreements), whereas in the Min-ECP case, we want to minimize the number of edges outside the clusters (disagreements).

We will use the online models, motivated by information retrieval applications, proposed by [4], and by [10] for the online correlation clustering problem. We define *online* versions of Max-ECP and Min-ECP clustering in a similar way as [10]. Vertices (with their edges to previous vertices) arrive one at a time and must be clustered as they arrive. The only two operations allowed are:

$\text{singleton}(v)$, that creates a singleton cluster containing the single vertex v and $\text{merge}(C_1, C_2)$, which merges two existing clusters into one, given that the resulting cluster induces a clique in the underlying graph. This means that once two vertices are clustered together, they can never be separated again.

Let C be a clustering consisting of clusters $c_1, c_2, \dots, c_m$, where each cluster c_i forms a clique. The *profit* of c_i is $p(c_i) \stackrel{\text{def}}{=} \binom{|c_i|}{2}$ and the profit of C is

$$p(C) \stackrel{\text{def}}{=} \sum_{i=1}^m p(c_i) = \sum_{i=1}^m \binom{|c_i|}{2}.$$

We define the *cost* of C to be $|E| - p(C)$, where E is the set of edges in the underlying graph. Hence, in Max-ECP, we want to maximize the profit of the generated clustering and in Min-ECP we want to minimize the cost of the clustering.

It is common to measure the quality of an online strategy by its *competitive ratio*. This ratio is defined as the worst case ratio between the profit/cost of the online strategy and the profit/cost of an offline optimal strategy, one that knows the complete input sequence in advance. We will use the competitive ratio to measure the quality of our online strategies.

Note that in the online model, the strategy does not know when the last vertex arrives and as a consequence any competitive ratio needs to be kept for every vertex that arrives, from the first to the last.

We henceforth let OPT denote an offline optimal solution to the clustering problem we are currently considering. The context will normally suffice to specify which optimum solution is meant. We use OPT_{kn} to denote the offline optimum solution on vertices $v_{k+1}, \dots, v_n$, where $k < n$ and these vertices are indexed in their order in the input sequence. We also use OPT_n to denote OPT_{0n} . Similarly, we use S_n to denote the solution of an online strategy on the n first vertices.

Note that we make no claims on the computational complexity of our strategies. In certain cases, our strategies use solutions to computationally intractable problems (such as clique partitioning problems in graphs) which may be considered impractical. However, our interest focuses on the relationship between the results of online strategies and those of offline optimal solutions, we believe that allowing the online strategy to solve computationally difficult tasks gives a fairer comparison between the two solutions.

3 Online Max-ECP Clustering

3.1 The Greedy Strategy for Online Max-ECP Clustering

The *greedy* strategy for Max-ECP clustering merges each input vertex with the largest current cluster that maintains the clique property. If no such merging is possible the vertex is placed in a singleton cluster. Greedy strategies are natural first attempts used to solve online problems and can be shown to behave well for certain of them. We show that the greedy strategy can be far from optimal for Max-ECP clustering.

Theorem 1. *The greedy strategy for Max-ECP clustering is no better than $2/(n-2)$ -competitive.*

Proof. Consider an adversary that provides input to the strategy to make it behave as badly as possible. Our adversary gives greedy $n = 2k$ vertices in order from 1 to $2k$. Each odd numbered vertex is connected to its even successor, each odd numbered vertex is also connected to every other odd numbered vertex before it, and similarly, each even numbered vertex is also connected to every even numbered vertex before it; see Figure 1.

Fig. 1. Illustrating the proof of Theorem 1.

The greedy strategy clusters the vertices as odd/even pairs, giving the clustering GDY_n having profit $p(GDY_n) = k$. An optimal strategy clusters the odd vertices in one clique of size k and the even vertices in another clique also of size k . The profit for the optimal solution is $p(OPT_n) = k(k-1)$. Hence, the worst case ratio between greedy and an optimum solution is at most $1/(k-1) = 1/(n/2-1)$ so the competitive ratio is at most $2/(n-2)$. $\square$

Next, we look at the upper bound for the greedy strategy.

Theorem 2. *The greedy strategy for Max-ECP clustering is $1/(n-2)$ -competitive.*

Proof. Consider an edge e inside a cluster produced by the greedy strategy on n vertices. We introduce a weight function $w(e)$ as follows: if e also belongs to a cluster in OPT_n , then we set $w(e) \stackrel{\text{def}}{=} 1$. If e does not belong to any cluster in OPT_n , then the two endpoints v and v' belong to different clusters of OPT_n . Denote these two clusters by c_e and c'_e and we set $w(e) \stackrel{\text{def}}{=} |c_e| + |c'_e| - 2$, i.e., the number of edges in c_e and c'_e connected to v and v' . Note that not both c_e and c'_e can be singleton clusters, so $w(e) \geq 1$ in all cases.

Consider now the sum, $\sum_{e \in GDY_n} w(e)$, where we abuse notation slightly and let $e \in GDY_n$ denote that two end points of edge e lies in the same cluster of GDY_n , the greedy clustering on n vertices. The sum counts every edge in OPT_n at least once, since an edge lying in both GDY_n and OPT_n is counted once and an edge lying in GDY_n but not in OPT_n counts all the edges in OPT_n connected to the two endpoints. The fact that the sum counts all the edges in OPT_n follows

since no two clusters in greedy can be merged to a single cluster. Hence,

$$\begin{aligned} p(OPT_n) &\leq \sum_{e \in G DY_n} w(e) \leq \sum_{e \in G DY_n} |c_e| + |c'_e| - 2 \leq (|c_1| + |c_2| - 2) \cdot \sum_{e \in G DY_n} 1 \\ &\leq (|c_1| + |c_2| - 2) \cdot p(G DY_n) \leq (n - 2) \cdot p(G DY_n), \end{aligned}$$

where c_1 and c_2 denote the two largest clusters in OPT_n . $\square$

3.2 A Lower Bound for Online Max-ECP Clustering

We present a lower bound for deterministic Max-ECP clustering.

Theorem 3. *Any deterministic strategy for Max-ECP clustering is at most $1/2$ -competitive.*

Proof. Again we use an adversarial argument and let the adversary provide $2k$ vertices, where every odd numbered vertex is connected to its subsequent even numbered vertex, v_1 to v_2 , v_3 to v_4 , etc. The game now continues in stages with the strategy constructing clusters followed by the adversary adding edges. In each stage the adversary looks at the clusters constructed; these are either singletons or pairs $\{v_{2i-1}, v_{2i}\}$. For each newly constructed pair cluster, the adversary adds two new vertices, v'_{2i-1} connected to v_{2i-1} , and, v'_{2i} connected to v_{2i} ; see Figure 2. When the strategy fails to produce any new pair clusters in a stage, the adversary stops.

Fig. 2. Illustrating the proof of Theorem 3.

Assume that the strategy at the end of the stages has constructed k' pair clusters, $k' \leq k$, thus giving a profit of k' . Note that the strategy can never produce the pairs $\{v_{2i-1}, v'_{2i-1}\}$ or $\{v_{2i}, v'_{2i}\}$ since these are revealed only if the pair $\{v_{2i-1}, v_{2i}\}$ is produced. The optimal solution in this case has profit $k + k'$ since this solution produces $2k'$ pair clusters $\{v_{2i-1}, v'_{2i-1}\}$ or $\{v_{2i}, v'_{2i}\}$, where the strategy produces $\{v_{2i-1}, v_{2i}\}$, in addition to $k - k'$ pairs $\{v_{2i-1}, v_{2i}\}$, where the strategy produces singleton clusters. Hence, the competitive ratio is

$$\frac{k'}{k + k'} \leq \frac{1}{2}, \quad \text{for } 0 \leq k' \leq k,$$

giving the result. $\square$

3.3 A Constant Competitive Strategy for Online Max-ECP Clustering

We present a new strategy for Max-ECP clustering and prove that it has constant competitive ratio. If the optimum solution OPT_n does not have any clusters of size larger than three, the strategy follows the greedy strategy. Otherwise, the strategy places arriving vertices in singleton clusters until the profit ratio between the current solution S'_n and the offline optimum solution OPT_n (of the n currently known vertices) goes below a threshold value c . When this happens, the strategy computes the *relative optimum* solution given the current clustering. The strategy is given in pseudocode below.

Strategy <i>Lazy</i> /* Maintain clustering S_n with profit $p(S_n)$ and let c be a constant */ 1 $n = 1$ 2 while new vertex v_n arrives do 2.1 $S'_n := S_{n-1} + \text{singleton}(v_n)$ 2.2 Compute OPT_n 2.3 if the largest cluster in OPT_n has size ≥ 4 then 2.3.1 if $p(OPT_n) > c \cdot p(S'_n)$ then 2.3.1.1 Compute the relative optimum of S'_n , $\widehat{OPT}(S'_n)$ 2.3.1.2 Construct S_n from $\widehat{OPT}(S'_n)$ using only merge operations else 2.3.1.3 $S_n := S'_n$ endif else 2.3.2 Construct S_n using the Greedy strategy endif 2.4 Report S_n 2.5 $n := n + 1$ endwhile End <i>Lazy</i>

Given a clustering S , the *relative optimum*, $\widehat{OPT}(S)$, is defined as follows: construct a graph G_S such that, for every cluster in S there is a vertex in G_S and two vertices in G_S are connected by an edge, if every pair of vertices in the two underlying clusters are connected. $\widehat{OPT}(S)$ is now the offline optimal clustering in G_S .

Given the current clustering, S'_n , the new clustering, S_n , is easily generated by constructing a cluster in S_n for each cluster in $\widehat{OPT}(S'_n)$ by merging the corresponding clusters in S'_n .

The following theorem follows directly from the construction of the strategy since the ratio between the profit of the optimal solution (given the current n vertices) and the profit of the online solution S_n never falls below the threshold c .

Theorem 4. *The Lazy strategy is $1/c$ -competitive for online Max-ECP clustering.*

We will establish the value of c to be $c = (154 + 16\sqrt{61})/9$ in Lemma 3.

We give a relationship between the profits of the two clusterings OPT_{n-1} and OPT_n .

Lemma 1. *Let $c_{\max}$ be the largest cluster in OPT_{n-1} , having size k , then, for all $n > 2$ the profit $p(OPT_n) \leq p(OPT_{n-1}) \cdot (k+1)/(k-1)$.*

Proof. The maximum increase occurs if v_n , the arriving vertex, can be clustered together with the vertices in $c_{\max}$. The increase in profit in this cluster goes from $\binom{k}{2}$ to $\binom{k+1}{2}$. The maximum increase for the whole clustering occurs if $c_{\max}$ is the only non-singleton cluster in OPT_{n-1} , giving us a ratio of $\binom{k+1}{2}/\binom{k}{2} = (k+1)/(k-1)$. $\square$

Let G be an undirected graph and let G_A and G_B be the two subgraphs induced by some partitioning of the vertices in G . Let C be a clustering on G and let A and B be the clusterings induced by C on G_A and G_B respectively.

Lemma 2. *If $p(A) > 0$ and $p(C)/p(A) = z > 1$, then $p(B)/p(C) \geq r(z)$ where $r(z)$ is*

$$r(z) = 1 - \frac{\sqrt{1+8z} - 2}{z}.$$

Proof. Our proof is by induction on the number of clusters in C . We assume the clusters $c_1, \dots, c_m$ in C are sorted on increasing number of vertices in a_i , where a_i is the cluster in A induced by the cluster c_i in C .

Similarly we denote by b_i the cluster in B induced by the cluster c_i in C .

We say that a cluster c_i is a *null cluster*, if the induced cluster a_i in A has $p(a_i) = 0$. This happens if a_i is either empty or a singleton set.

We first prove the base case, where we assume that C contains exactly one non-null cluster, i.e., $c_1, \dots, c_{j-1}$ are clusters such that $p(a_i) = 0$, for $1 \leq i < j$ and c_j is the first cluster where $p(a_j) > 0$. Assume that $p(c_j)/p(a_j) = z''$ and that $|a_j| = l$, $|b_j| = l'$ and $|c_j| = l + l'$.

We prove the base case of the induction also using induction and assume for this base case that $j = 1$. In this case, $z = p(C)/p(A) = p(c_1)/p(a_1) = z''$ and we get by straightforward calculations that

$$\frac{p(B)}{p(C)} = \frac{p(b_1)}{p(c_1)} = 1 - \frac{\sqrt{1+4zl(l-1)} - l}{z(l-1)} \geq r(z),$$

since the expression before the inequality is increasing in l , and therefore minimized when $l = 2$.

For the inductive case of the base case, we assume the result holds for $j-2 \geq 0$ null clusters and one non-null cluster and prove it for $j-1$ null clusters and one non-null cluster. Let $\{c_2, \dots, c_j\}$ be denoted by C' and let A' and B' be the induced clusterings of C' in G_A and G_B . We set $p(C')/p(A') = z'$ and have when we add null cluster c_1 to the clustering that

$$z = \frac{p(C)}{p(A)} = \frac{p(C') + p(c_1)}{p(a_j)} = z' + \frac{p(c_1)}{p(a_j)},$$

giving us that $z' \leq z$ and

$$\frac{p(B)}{p(C)} = \frac{p(b_1) + p(B')}{p(C)} \geq \frac{p(c_1) - |c_1| + 1 + p(B')}{p(c_1) + p(C')}.$$

The inequality stems from the fact that a_1 can either be a singleton or an empty cluster, so b_1 either contains the same number of vertices as c_1 or one less.

By the induction hypothesis we have that $p(B') \geq r(z') \cdot p(C')$, and since $p(A) = p(A') = p(a_j) = p(c_1)/(z - z')$, $p(C') = z'p(a_j) = zp(a_j) - p(c_1)$, and $|c_1| = (1 + \sqrt{1 + 8(z - z')p(a_j)})/2$, this gives us that

$$\begin{aligned} \frac{p(B)}{p(C)} &\geq \frac{(z - z')p(a_j) + (1 - \sqrt{1 + 8(z - z')p(a_j)})/2 + z'r(z')p(a_j)}{zp(a_j)} \\ &= 1 + \frac{z'(r(z') - 1)}{z} + \frac{1 - \sqrt{1 + 8(z - z')p(a_j)}}{2zp(a_j)} \\ &\geq 1 + \frac{z'(r(z') - 1)}{z} + \frac{1 - \sqrt{1 + 16(z - z')}}{4z}. \end{aligned}$$

The last expression is a decreasing function of z' between 0 and z , so increasing z' to z yields $p(B)/p(C) \geq r(z)$. Hence the base case when C has zero or more null clusters and exactly one non-null cluster has been completed.

For the general induction step, assume the formula holds for $m - 1$ clusters, prove it for m clusters. We let $p(C)/p(A) = z$, $C' = \{c_1, \dots, c_{m-1}\}$, $C = C' \cup \{c_m\}$, $p(C')/p(A') = z'$ and $p(c_m)/p(a_m) = z''$.

By the induction hypothesis we have that

$$\frac{p(B)}{p(C)} \geq \frac{r(z') \cdot p(C') + r(z'') \cdot p(c_m)}{p(C') + p(c_m)} = r(z') \frac{z'(z - z'')}{z(z' - z'')} + r(z'') \frac{z''(z' - z)}{z(z' - z'')}$$

The last expression decreases as z'' tends towards z , again giving us $p(B)/p(C) \geq r(z)$, thus proving our result. $\square$

Lemma 3. *If, for a certain value of n , the selection in Step 2.3.1 yields true in the lazy strategy, then the profit $p(OPT_n) \leq a \cdot p(S_n)$ where $a < c$ is some constant.*

Proof. When the largest clusters in OPT_n has size at most three, we have from the proof of Theorem 2 that greedy has competitive ratio $1/4$, and lazy will do at least as well in this case, since it follows the greedy strategy. So, we can assume that the largest cluster in OPT_n has size at least four. This also means that the size of the largest cluster in OPT_{n-1} is at least three.

We make a proof by induction on n , the number of steps in the algorithm. The base cases when $n = 1, 2$ and 3 follow immediately, since lazy (and greedy) computes optimal solutions in these cases, so $a \leq 4$ can be chosen as the constant, since the competitive ratio is $1/a \geq 1/4$.

Assume for the inductive case that Step 2.3.1 yields true at the n -th iteration and assume further that the previous time it happened it was in iteration k

(or that the strategy followed greedily in this step). By the induction hypothesis we know that $p(OPT_k) \leq a \cdot p(S_k)$ for some constant $a < c$. Let OPT'_k be the clustering obtained from OPT_n induced by the vertices $v_1 \dots v_k$. It is obvious that $p(OPT'_k) \leq p(OPT_k)$. Let E_{kn} be the set of edges between vertices inside clusters of OPT_n that have both endpoints among the vertices $v_{k+1} \dots v_n$. Similarly, we define E'_{kn} to be the set of edges inside clusters that have one end point among the vertices $v_1 \dots v_k$ and the other among $v_{k+1} \dots v_n$. We now have that

$$p(OPT'_k) + |E'_{kn}| + |E_{kn}| = p(OPT_n).$$

Let S'_n be the clustering solution in iteration n just before the strategy reaches Step 2.3.1, i.e., when vertex v_n is put in a singleton cluster. This gives us, since $p(S'_n) = p(S_k)$,

$$p(OPT_n) > c \cdot p(S'_n) = c \cdot p(S_k) \geq \frac{c}{a} \cdot p(OPT'_k)$$

Since $p(OPT_n)/p(OPT'_k) \geq c/a$, by Lemma 2 the ratio $|E_{kn}|/p(OPT_n) \geq r(c/a)$.

Note that E_{kn} forms a clustering of vertices $v_{k+1}, \dots, v_n$ that is independent of how vertices $v_1, \dots, v_k$ are clustered. Therefore, when a new cluster S_n is recomputed in Step 2.3.1.1, it includes at least as many edges as both S_k and E_{kn} together. Furthermore, $p(S_{n-1}) = p(S_k)$ and $p(OPT_{n-1}) \leq c \cdot p(S_{n-1})$, since otherwise Step 2.3.1.1 would have been done already in the previous iteration. We have that

$$\begin{aligned} p(S_n) &\geq p(S_k) + p(OPT_{kn}) \geq p(S_k) + |E_{kn}| = p(S_{n-1}) + |E_{kn}| \\ &\geq \frac{p(OPT_{n-1})}{c} + |E_{kn}| \geq \frac{p(OPT_n)}{2c} + |E_{kn}| \\ &\geq \frac{p(OPT_n)}{2c} + r(c/a) \cdot p(OPT_n). \end{aligned}$$

The second to last inequality follows from Lemma 1, since the largest cluster in OPT_{n-1} must have size 3, and the last inequality was given above.

We must guarantee that

$$\frac{p(OPT_n)}{2c} + r(c/a) \cdot p(OPT_n) \geq \frac{p(OPT_n)}{a}$$

to prove the lemma, which is equivalent to finding constants $a \leq 4$ and c as small as possible so that $1/(2c) + r(c/a) \geq 1/a$. The inequality holds for $a = 4$ and in equality for $c = (154 + 16\sqrt{61})/9 \approx 30.9960$. $\square$

From Theorem 4 it follows that the competitive ratio for the lazy strategy is $9/(154 + 16\sqrt{61}) \approx 0.032262$.

4 Online Min-ECP Clustering

4.1 A Lower Bound for Online Min-ECP Clustering

We present a lower bound for deterministic Min-ECP clustering.

Theorem 5. *Any deterministic strategy for Min-ECP clustering is no better than $n^{1-\epsilon}/2$ -competitive, for every $\epsilon > 0$.*

Proof. An adversary provides the following vertices of an input graph in sequence. First, one $(k-1)$ -clique, followed by one additional vertex connected to one of the previously given vertices, i.e., a lollipop graph; see Figure 3. We now consider different possibilities for clustering the k vertices.

Fig. 3. A lollipop graph with a $(k-1)$ -clique and a vertex connected by a single edge.

First, let us assume that the strategy has clustered the input in such a way that the $(k-1)$ -clique is not clustered as one cluster. Then this clustering has at least $k-2$ disagreements. An optimal clustering contains only one disagreement, between the $(k-1)$ -clique and a singleton cluster containing the vertex outside the clique. Hence, the competitive ratio in this case is at least $(k-2)/1 = k-2$ since the number of vertices is $n = k$.

Assume next that the strategy has clustered the input as one $(k-1)$ -clique and one singleton cluster. In this case, the adversary provides $k-1$ independent cliques of size m , where each of the vertices in an m -clique is connected to one particular vertex of the original $(k-1)$ -clique; see Figure 4. No other edges exist in the input.

Fig. 4. Each of the $(k-1)$ vertices in the central clique is connected with m edges to an m -clique.

The strategy can at best cluster the $k-1$ m -cliques as clusters, thus generating $m(k-1)$ disagreements. An optimal solution will, for m sufficiently large, cluster the vertices in the original $(k-1)$ -clique in each of the new cliques, generating a solution of $k-1$ $(m+1)$ -cliques. This solution has $\binom{k-1}{2}$ disagreements. If we set $m = (k-2)^t/4$, where k is chosen so that m is an integer and t is some sufficiently large integer, then the competitive ratio becomes

$$\frac{(k-1)m}{\binom{k-1}{2}} = \frac{(k-2)^{t-1}}{2} \geq \frac{1}{2} \left(n^{1/(t+1)} \right)^{t-1} \geq \frac{n^{1-\frac{2}{t+1}}}{2} = \frac{n^{1-\epsilon}}{2},$$

for all $\epsilon > 0$, since the number of vertices in the input is $n = (k-1)m + k = (k-1)(k-2)^t/4 + k \leq (k-2)^{t+1}$, proving the lower bound. $\square$

4.2 The Greedy Strategy for Online Min-ECP Clustering

In this section, we prove that the greedy strategy yields a competitive ratio of $n - 2$, almost matching the lower bound provided in section 4.1. The greedy strategy was presented in Section 3.1.

Theorem 6. *The greedy strategy for Min-ECP clustering is no better than $(n - 2)/2$ -competitive.*

Proof. We let an adversary generate the same input sequence of $n = 2k$ vertex pairs as in the proof of Theorem 1. Greedy generates $2\binom{k}{2}$ disagreement edges whereas the optimum solution has k disagreement edges. The competitive ratio becomes

$$\frac{2\binom{k}{2}}{k} = k - 1 = \frac{n - 2}{2}.$$

□

We say that a solution to Min-ECP clustering is *maximal*, if the solution cannot be improved by the merging of any clusters. A strategy for Min-ECP clustering is called *maximal*, if it always produces maximal solutions. Note that greedy belongs to this class of maximal strategies.

Theorem 7. *Any maximal online strategy for Min-ECP clustering problem is $2n - 3$ -competitive.*

Proof. Consider a disagreement edge e connecting vertices v and v' outside any cluster produced by the maximal strategy MAX_n on n vertices. We start by showing that either v or v' must have an adjacent edge that is a disagreement edge in OPT_n . We have two cases: if e is also a disagreement edge in OPT_n , there is a disagreement edge in OPT_n adjacent to v or v' .

Now, if e is not a disagreement edge in OPT_n , then one of v and v' connects to a vertex u , assume it is v , such that the edge $e' = (v, u)$ is a disagreement edge in OPT_n , i.e., there is a cluster in MAX_n containing v and u but not v' and there is a cluster in OPT_n containing v and v' but not u . The vertex u must exist, otherwise MAX_n would have clustered v and v' together, a contradiction.

In this way, we have proved that to each disagreement edge in MAX_n , there must be an adjacent disagreement edge in OPT_n .

Consider now a disagreement edge e in OPT_n . Potentially, all its adjacent edges can be disagreement edges for MAX_n , giving us in the worst case $2n - 4$ adjacent disagreement edges different from e and one where they coincide. Hence the worst case competitive ratio is $2n - 3$. □

From our observation that greedy belongs to the class of maximal strategies we have the following corollary.

Corollary *Greedy is $2n - 3$ -competitive.*

5 Conclusion

We have proved almost matching upper and lower bounds for clique clustering. Our main result is a constant competitive strategy for clique clustering when the cost measure is to maximize the total number of edges in the cliques. Our strategy does not consider the computational feasibility of clique clustering, which can be considered a drawback. However, we feel that since the lower bound adversarial arguments also allow this computation, our measure is fairer to the strategy. In addition, the computational problems required to be solved efficiently for our strategy are indeed efficiently solvable for large classes of graphs, such as chordal graphs, line graphs and circular-arc graphs [7].

References

1. Alvey, S., Borneman, J., Chrobak, M., Crowley, D., Figueroa, A., Hartin, R., Jiang, G., Scupham, A., T., Valinsky, L., Della Vedova, and Yin, B. Analysis of bacterial community composition by oligonucleotide fingerprinting of rRNA genes. *Applied and Environmental Microbiology* 68, pages 3243-3250, 2002.
2. Bansal, N., Blum, A., and Chawla, S. Correlation Clustering. *Machine Learning* 56(1-3), pages 89-113, 2004.
3. Ben-Dor, A., Shamir, R., and Yakhini, Z. Clustering Gene Expression Patterns. *Journal of Computational Biology* 6(3/4), pages 281-297, 1999.
4. Charikar, M., Chekuri, C., Feder, T., and Motwani, R. Incremental Clustering and Dynamic Information Retrieval. *SIAM J. Comput.* 33(6), pages 1417-1440, 2004.
5. Charikar, M., Guruswami, V., and Wirth, A. Clustering with Qualitative Information. In *Proc. 44th Annual Symposium on Foundations of Computer Science FOCS 2003*, pp. 524-533, 2003.
6. Demaine, E. and Immorlica, N. Correlation Clustering with Partial Information. In *Proc. 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX 2003)*, pp. 1-13, 2003.
7. Dessmark, A., Jansson, J., Lingas, A., Lundell, E., and Persson, M. On the Approximability of Maximum and Minimum Edge Clique Partition Problems. *Int. J. Found. Comput. Sci.* 18(2), pages 217-226, 2007.
8. Figueroa, A., Borneman, J., and Jiang, T. Clustering binary fingerprint vectors with missing values for DNA array data analysis. *Journal of Computational Biology* 11(5), pages 887-901, 2004.
9. Figueroa, A., Goldstein, A., Jiang, T., Kurowski, M., Lingas, A., and Persson, M. Approximate clustering of incomplete fingerprints. *J. Discrete Algorithms* 6(1), pages 103-108, 2008.
10. Mathieu, C., Sankur, O., and Schudy, W. Online Correlation Clustering. In *Proc. 7th International Symposium on Theoretical Aspects of Computer Science (STACS 2010)*, pp. 573-584, 2010.
11. Shamir, R., Sharan, R., and Tsur, D. Cluster Graph Modification Problems. In *Proc. 28th International Workshop on Graph Theoretic Concepts in Computer Science (WG 2002)*, pp. 379-390, 2002.