Competitive Strategies for Online Clique Clustering

Marek Chrobak[1]*, Christoph Dürr[2][3], and Bengt J. Nilsson[4]

[1] University of California at Riverside, USA.
[2] Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, Paris, France.
[3] CNRS, UMR 7606, LIP6, Paris, France.
[4] Department of Computer Science, Malmö University, Malmö, Sweden.

Abstract. A *clique clustering* of a graph is a partitioning of its vertices into disjoint cliques. The quality of a clique clustering is measured by the total number of edges in its cliques. We consider the online variant of the clique clustering problem, where the vertices of the input graph arrive one at a time. At each step, the newly arrived vertex forms a singleton clique, and the algorithm can merge any existing cliques in its partitioning into larger cliques, but splitting cliques is not allowed. We give an online strategy with competitive ratio 15.645 and we prove a lower bound of 6 on the competitive ratio, improving the previous respective bounds of 31 and 2.

1 Introduction

A *clique clustering* of a graph $G = (V, E)$ is a partitioning of the vertex set V into disjoint cliques $C_1, C_2, ..., C_k$. The *profit* of this clustering is defined to be the total number of edges in these cliques, that is $\sum_{i=1}^{k} \binom{|C_i|}{2} = \frac{1}{2} \sum_{i=1}^{k} |C_i|(|C_i|-1)$. In the *clique clustering problem* the objective is to compute a clique clustering of the given graph that maximizes this profit value. For a graph G, by $\mathsf{O}(G)$ we denote the optimal profit for G.

We consider the online variant of the clique clustering problem, where the input graph G is not known in advance. (See [3], for more background on online problems). The vertices of G arrive one at a time. Let v_t denote the vertex that arrives at time t, for $t = 1, 2, ...$. When v_t arrives, its edges to all preceding vertices $v_1, ..., v_{t-1}$ are revealed as well. In other words, after step t, the subgraph of G induced by $v_1, v_2, ..., v_t$ is known, but no other information about G is available.

Our objective is to construct a procedure that incrementally constructs and outputs a clustering based on the information acquired so far. Specifically, when v_t arrives at step t, the procedure first creates a singleton clique $\{v_t\}$. Then it is allowed to merge any number of cliques (possibly none) in its current partitioning into larger cliques. No other modifications of the clustering are allowed.

* Research supported by NSF grants CCF-0729071 and CCF-1217314.

We avoid using the word algorithm for our procedure, since it evokes connotations with computational limits in terms of complexity and computability. In fact, we place no limits on the computational power of our procedure and to emphasize this, we use the word *strategy* rather than algorithm. This approach allows us to focus specifically on the limits posed by the lack of complete information about the input. Similar considerations played a role in some earlier work on online computation, for example for online medians [6,7,12], minimum-latency tours [5], and several other online optimization problems (see [8]).

Throughout the paper we will implicitly assume that any graph G has its vertices ordered $v_1, v_2, ..., v_n$, according to the ordering in which they arrive on input. For an online strategy $\mathcal{S}$ let $\mathsf{profit}_{\mathcal{S}}(G)$ be the profit of $\mathcal{S}$ when the input graph is G. We say that an online strategy $\mathcal{S}$ is *R-competitive* if for any input graph G we have

$$R \cdot \mathsf{profit}_{\mathcal{S}}(G) + \beta \geq \mathsf{O}(G), \tag{1}$$

for some constant β independent of G. The competitive ratio of $\mathcal{S}$ is the smallest R for which $\mathcal{S}$ is R-competitive[1]. This concept is sometimes referred to as the *asymptotic competitive ratio* in the literature, but we will omit the term "asymptotic" in the paper. If $\beta = 0$, then R is called the *absolute competitive ratio*.

The online model for clique clustering was studied by Fabijan *et al.* [10], who designed an online strategy with competitive ratio 31 and proved that no online strategy can have competitive ratio better than 2. They also showed that the greedy strategy's competitive ratio is linear with respect to the graph size, and they studied an alternative model where the objective is to minimize the number of edges that are not in the clusters.

The clique clustering problem arises in applications to gene expression profiling and DNA clone classification [14,2,11]. The offline variant is known to be NP-hard, and in fact not even approximable within factor $n^{1-o(1)}$ under some reasonable complexity-theoretic assumptions [9].

Our results. We provide two new bounds on the competitive ratio of online clique clustering, considerably improving the results in [10]. First, we present an online strategy with competitive ratio 15.645. The idea of the strategy is based on the "doubling" technique. Roughly (but not exactly), we divide the computation into phases, where the optimal profit of the set of vertices from phase j grows exponentially with j. After each phase j the cliques computed from this optimal clustering are added to the strategy's clustering of the current graph. We give an example showing that the competitive ratio of our strategy is no better than 10.92. We then also show that there is no deterministic online strategy for clique clustering with competitive ratio smaller than 6.

Related work. Clustering is a dynamic and important field of research with multiple applications in almost all areas of sciences, humanities and engineering. There are many clustering models in the literature, with varying criteria for data

[1] Earlier papers on online clustering define the competitive ratio as the maximum value of $\mathsf{profit}_{\mathcal{S}}(G)/\mathsf{O}(G)$, which is the inverse of the value we use.

similarity (which determines whether two data items can be clustered together), quality measures for clustering, and requirements for the number of clusters.

Approximation algorithms for incremental clustering, where the only operations allowed are to create singleton clusters and merge existing clusters, were first studied by Charikar *et al.* [4], although for a different clustering model than ours. Mathieu *et al.* [13] applied this incremental approach in the model of online correlation clustering, initially introduced in [1,2]. In correlation clustering, as in our model, the similarity relation is represented by an undirected graph, but the objective function is equal to the sum of the number of edges in the clusters plus the number of non-edges outside clusters. The results in [13] include a lower bound of 1.245 and an upper bound slightly below 2 on the competitive ratio (the ratio 2 can be achieved with a greedy strategy).

2 A Competitive Strategy

In this section we give our competitive online strategy OCC. Roughly, the strategy works in phases. In each phase we consider the "batch" of nodes that have not yet been clustered with other nodes, compute an optimal clustering for this batch, and add these new clusters to the strategy's clustering. The phases are defined so that the profit for consecutive phases increases exponentially.

The overall idea can be thought of as an application of the "doubling" strategy (see [8], for example), but in our case a subtle modification is required. Unlike in other doubling approaches, in our strategy the phases are not completely independent: the clustering computed in each phase, in addition to the new nodes, needs to include the singleton nodes from earlier phases as well. This is needed, because in our objective function singleton clusters do not bring any profit.

We remark that one could alternatively consider using profit value $\frac{1}{2}p^2$ for a clique of size p, which is a very close approximation to our function if p is large. This would lead to a simpler strategy and much simpler analysis. However, this function is a bad approximation when the clustering involves many small cliques, which is also in fact the most challenging scenario in the analysis of our algorithm, and instances with this property are also used in the lower bound proof.

The Strategy OCC. Formally, our method works as follows. Fix some constant parameter $\gamma > 1$ of the strategy which we will later optimize. The strategy works in phases, starting with phase $j = 0$. At any moment the clustering maintained by the strategy contains a set U of *singleton* cliques. Each arriving vertex is added into U. As soon as there is a clustering of U of profit at least γ^j, the strategy creates these clusters, adds them to its current clustering, and moves to phase $j + 1$.

Note that phase 0 ends as soon as one edge is released, since then it is possible for OCC to create a clustering with $\gamma^0 = 1$ edge. The last phase may not be complete; as a result all nodes released in this phase will be clustered as singletons. Note also that the strategy never merges non-singleton cliques.

Asymptotic Analysis of OCC. It is convenient to think of the computation as lasting forever. We then want to show that at each step of the computation, the optimal profit is at most R times the profit of OCC, plus some absolute additive constant, where $R \approx 15.645$ is the claimed competitive ratio.

For every phase $j = 0, 1, \ldots$, denote by Δ_j the optimal profit of the vertices that arrived in phase j. Let $\mathsf{S}_j = \Delta_0 + \ldots + \Delta_j$ be the total profit of the strategy and O_j the total profit of the adversary at the end of phase j. By the definition of OCC, for all phases j we have $\Delta_j \geq \gamma^j$ and $\mathsf{S}_j \geq (\gamma^{j+1} - 1)/(\gamma - 1)$.

We fix some instance and start with some observations. First, at the end of phase 0 the strategy is optimal. Also, in each step, except for the last step of a phase, the strategy's profit does not change while the optimum profit can only increase. Therefore it suffices to compare the optimal profit O_j at the end of a phase $j \geq 1$, with the strategy's profit right before the end of the phase, which is equal to S_{j-1}.

After any phase j, the optimal clustering of U may include some singletons. If this is so, the adversary can release those vertices during the next phase instead, and the behavior of OCC will remain unchanged. We can thus assume without loss of generality that the optimal clustering of U does not contain any singletons. As a result, after each phase j, all clusters of OCC have at least two vertices.

With the above assumption, we can divide the vertices into disjoint *batches*, where batch B_j contains the vertices released in phase j. During phase j, the clustering of OCC is then the union of clusterings of all its batches $B_0, B_1, \ldots, B_{j-1}$, plus the singletons released in phase j.

Let $\bar{B}_j = B_0 \cup B_1 \cup \ldots \cup B_j$ be the set of vertices released in phases $0, 1, \ldots, j$. Consider the optimal clustering of $\bar{B}_j$. In this clustering, every cluster C has some number a of nodes in $\bar{B}_{j-1}$ and some number b of nodes in B_j. Let $k_{a,b}$ be the number of clusters of this form in the optimal clustering. Then we have the following bounds, where the sums range over all integers $a, b \geq 0$.

$$\mathsf{O}_j = \sum \binom{a+b}{2} k_{a,b} \qquad (2) \qquad\qquad \Delta_j \geq \sum \binom{b}{2} k_{a,b} \qquad (4)$$

$$\mathsf{O}_{j-1} \geq \sum \binom{a}{2} k_{a,b} \qquad (3) \qquad\qquad \mathsf{S}_{j-1} \geq \tfrac{1}{2} \sum a k_{a,b} \qquad (5)$$

Equality (2) is the definition of O_j. Inequality (3) holds because the right hand side represents the profit of the optimal clustering of $\bar{B}_j$ restricted to $\bar{B}_{j-1}$, so it cannot exceed the optimal profit O_{j-1} for $\bar{B}_{j-1}$. Similarly, inequality (4) holds because the right hand side is the profit of the optimal clustering of $\bar{B}_j$ restricted to B_j, while Δ_j the optimal profit of B_j. The last bound (5) follows from the fact that the strategy does not have any singleton clusters in $\bar{B}_{j-1}$. This means that in the strategy's clustering of $\bar{B}_{j-1}$ (which has $\sum a k_{a,b}$ vertices) each vertex has an edge included in some cluster, so the number of these edges must be at least $\frac{1}{2} \sum_{a \geq 0} a k_{a,b}$.

We can also bound $\varDelta_j$, the strategy's profit increase, from above. We have $\varDelta_0 = 1$ and for each phase $j \geq 1$

$$\varDelta_j < \gamma^j + \sqrt{2}\gamma^{j/2} + 2 - \sqrt{2}. \tag{6}$$

To show (6), suppose that phase j ends at step t (that is, right after v_t is revealed). Consider the optimal partitioning $\mathcal{P}$ of B_j, and let the cluster C of v_t in $\mathcal{P}$ have size $p+1$. If we remove v_t from this partitioning, we obtain a partitioning of the batch after step $t-1$, whose profit must be strictly smaller than γ^j. So the profit of $\mathcal{P}$ is smaller than $\gamma^j + p$. In this new partitioning, cluster $C - \{v_t\}$ has size p. We thus obtain that $\binom{p}{2} < \gamma^j$, which gives us $p < \sqrt{2}\gamma^{j/2} + 2 - \sqrt{2}$, thus proving (6).

From (6), by adding up all profits from phases $0, \ldots, j$, we obtain an upper bound on the total profit of the strategy:

$$\mathsf{S}_j < \frac{\gamma^{j+1} - 1}{\gamma - 1} + \sqrt{2} \cdot \frac{\gamma^{(j+1)/2} - \gamma^{1/2}}{\gamma^{1/2} - 1} + (2 - \sqrt{2})j. \tag{7}$$

When phase 0 ends we have $\mathsf{O}_0 = \mathsf{S}_0 = 1$. As explained earlier, for $j \geq 1$ the worst case ratio occurs right before phase j ends. At this point, OCC has accrued a profit of S_{j-1}, since all vertices released during phase j are put into singleton clusters. The optimal solution, on the other hand, is bounded by O_j. The ratio $\mathsf{R}_j = \mathsf{O}_j / \mathsf{S}_{j-1}$ is therefore also an upper bound on the competitive ratio throughout phase j. Our goal now is to upper bound R_j, for all j. We will use the following technical lemma.

Lemma 1. *For any pair of non-negative integers a and b, the inequality*

$$\binom{a+b}{2} \leq (x+1)\binom{a}{2} + \frac{x+1}{x}\binom{b}{2} + a$$

holds for any $0 < x \leq 1$.

Proof. Define the function

$$\begin{aligned} F(a,b,x) &= 2x(x+1)\binom{a}{2} + 2(x+1)\binom{b}{2} + 2ax - 2x\binom{a+b}{2} \\ &= a^2x^2 - ax^2 + 2ax + b^2 - b - 2abx = (b - ax)^2 + ax(2-x) - b, \end{aligned}$$

i.e., twice x times the difference between the right hand side and the left hand side of the inequality above. It is sufficient to show that $F(a,b,x)$ is non-negative for integers $a, b \geq 0$ and $0 < x \leq 1$.

Consider first the cases when $a \in \{0,1\}$ or $b \in \{0,1\}$. $F(0,b,x) = b(b-1) \geq 0$, for any non-negative integer b and any x. $F(a,0,x) = ax(ax - x + 2) \geq ax(ax + 1) > 0$, for any positive integer a and $0 < x \leq 1$. $F(a,1,x) = x^2 a(a-1) \geq 0$, for any positive integer a and any x. $F(1,2,x) = 2 - 2x \geq 0$, for $0 < x \leq 1$, and $F(1,b,x) = b^2 - b + 2x - 2bx \geq b^2 - 3b \geq 0$, for any integer $b \geq 3$ and $0 < x \leq 1$.

Thus, it only remains to show that $F(a, b, x)$ is non-negative when both $a \geq 2$ and $b \geq 2$. The function $F(a, b, x)$ is quadratic and hence has one local minimum at $x_0 = \frac{b-1}{a-1}$, as can be easily verified by differentiating F in x. Therefore, in the case when $a \leq b$, $F(a, b, x) \geq F(a, b, 1) = (b-a)^2-(b-a) \geq (b-a)-(b-a) = 0$, for $0 < x \leq 1$. In the case when $a > b$, we have that $F(a, b, x) \geq F(a, b, \frac{b-1}{a-1}) = \frac{(a-b)(b-1)}{a-1} > 0$, which completes the proof. □

Now, to find an upper bound on all R_j's, we will establish a recurrence relation for the sequence $\mathsf{R}_1, \mathsf{R}_2, \ldots$. The value of R_1 is some constant (its exact value is not important since we are interested in the asymptotic ratio). Suppose that $j \geq 2$ and fix some parameter x, $0 < x < 1$, whose value we will determine later. Using Lemma 1 and the bounds (2)-(5) we obtain

$$\begin{aligned}
\mathsf{R}_j \mathsf{S}_{j-1} = \mathsf{O}_j &= \sum \binom{a+b}{2} k_{a,b} \\
&\leq (x+1) \sum \binom{a}{2} k_{a,b} + \frac{x+1}{x} \sum \binom{b}{2} k_{a,b} + \sum a k_{a,b} \\
&\leq (x+1) \mathsf{O}_{j-1} + \frac{x+1}{x} \Delta_j + 2\mathsf{S}_{j-1} \qquad (8) \\
&= (x+1) \mathsf{R}_{j-1} \mathsf{S}_{j-2} + \frac{x+1}{x} \Delta_j + 2\mathsf{S}_{j-1}.
\end{aligned}$$

Thus R_j satisfies the recurrence

$$\mathsf{R}_j \leq \frac{x+1}{x\mathsf{S}_{j-1}} \left[x\mathsf{S}_{j-2}\mathsf{R}_{j-1} + \Delta_j \right] + 2. \qquad (9)$$

From inequalities (6) and (7), we have $\Delta_i = \gamma^i(1 + o(1))$ and $\mathsf{S}_i = \frac{\gamma^{i+1}(1+o(1))}{\gamma-1}$ for all i. We use the notation $o(1)$ to denote any function that tends to 0 as the number of phases goes to infinity. Substituting into the above recurrence, we get

$$\mathsf{R}_j \leq \frac{(x+1)(1+o(1))}{\gamma} \mathsf{R}_{j-1} + \frac{(x+1)(\gamma-1)}{x} + 2 + o(1). \qquad (10)$$

Assuming that $x + 1 < \gamma$, (10) implies that the sequence R_j converges and, denoting its limit by $\mathsf{R} = \lim_{j\to\infty} \mathsf{R}_j$, we then get

$$\mathsf{R} \leq \frac{\gamma(\gamma x + x + \gamma - 1)}{x(\gamma - x - 1)}. \qquad (11)$$

This expression is minimized for parameters $x = (5 - \sqrt{13})/2 \approx 0.697$ and $\gamma = (3 + \sqrt{13})/2 \approx 3.303$, yielding the asymptotic competitive ratio

$$\mathsf{R} \leq \tfrac{1}{6}(47 + 13\sqrt{13}) \approx 15.645.$$

Summarizing this analysis, we obtain the following theorem.

Theorem 1. *The asymptotic competitive ratio of* OCC *is at most* 15.645.

Table 1. Some initial upper bound values for the absolute competitive ratio.

Phase	1	2	3	4	5	6	7	8
Bound	10.000	17.493	23.157	24.854	24.521	22.539	20.474	18.793

Absolute Competitive Ratio. In fact, for parameters $x = (5 - \sqrt{13})/2$ and $\gamma = (3 + \sqrt{13})/2$, Strategy OCC has a low absolute competitive ratio as well. We show that this ratio is at most 24.854.

When phase 0 ends, the competitive ratio is 1. For $j \geq 1$, let O'_j be the optimal profit right before phase j ends. (Earlier we used O_j to estimate this value, but O_j also includes the profit for the last step of phase j.) It remains to show that for phases $j \geq 1$ we have $\mathsf{R}'_j \leq 24.854$, where $\mathsf{R}'_j = \mathsf{O}'_j / \mathsf{S}_{j-1}$.

By analyzing the behavior of Strategy OCC in phase 1 and exhaustively enumerating the possible configurations, given that $\gamma \approx 3.303$, we can establish that $\mathsf{R}'_1 = 10$.

For phases $j \geq 2$, we can tabulate upper bounds for R'_j by explicitly computing the ratios $\mathsf{O}'_j / \mathsf{S}_{j-1}$ using a modification of recurrence (9), where we take advantage of the fact that some quantities in inequalities (6) and (7) are integral, so their estimates can be rounded down. We show the first few estimates in Table 1.

To bound the sequence $\{\mathsf{R}'_j\}_{j>0}$ we use (9), (6) and (7), to obtain the recurrence

$$\mathsf{R}'_j \leq (x+1)\alpha_j \mathsf{R}'_{j-1} + \beta_j,$$

where $\alpha_j \leq \dfrac{\gamma^{j-1} + \sqrt{5\gamma^j} + 3j/2}{\gamma^j - 1}$ and $\beta_j \leq \dfrac{(x+1)(\gamma - 1)}{x} \cdot \dfrac{\gamma^j + \sqrt{2\gamma^j} + 1}{\gamma^j - 1} + 2.$

For $j \geq 6$ it is not hard to show that $\beta_j \leq 8$. Consider the denominator $\gamma^j - 1$ of α_j. We have that $\gamma^j - 1 > \frac{9}{10}\gamma^j$ for $j \geq 2$. Hence, $\mathsf{R}'_j \leq \hat{\mathsf{R}}_j$, where $\hat{\mathsf{R}}_j$ is given by the recurrence

$$\hat{\mathsf{R}}_j \leq \frac{10(x+1)(\gamma^{j-1} + \sqrt{5}\gamma^{j/2} + 3j/2)}{9\gamma^j}\hat{\mathsf{R}}_{j-1} + 8 \leq \frac{3}{5}\hat{\mathsf{R}}_{j-1} + 8 = 20 - 19\left(\frac{3}{5}\right)^j$$

for $j \geq 8$. The sequence $\{\hat{\mathsf{R}}_j\}_{j \geq 0}$, with $\hat{\mathsf{R}}_0 = 1$, grows monotonically to the limit $\lim_{j \to \infty} \hat{\mathsf{R}}_j = 20$ and hence $\hat{\mathsf{R}}_j \leq 20$ for every $j \geq 8$. Combining this with the earlier bounds, we see that the largest bound on R'_j is 24.854, given in Table 1 for $j = 4$. We can thus conclude that the absolute competitive ratio is at most 24.854.

3 A Lower Bound of 6

We now prove that any deterministic online strategy $\mathcal{S}$ for the clique clustering problem has competitive ratio at least 6. We present the proof for the absolute competitive ratio; later we explain how to extend it to the asymptotic ratio. The lower bound is established by showing, for any constant $R < 6$, an adversary

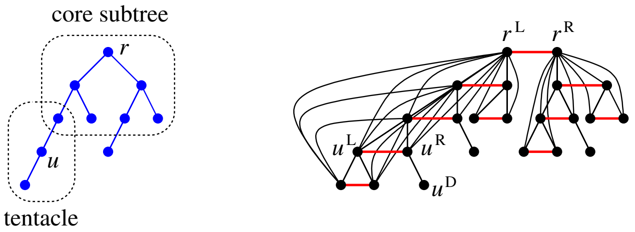

Fig. 1. On the left, an example of a skeleton tree $\mathcal{T}$. The core subtree of $\mathcal{T}$ has depth 2 and two tentacles, one of length 2 and one of length 1. On the right, the corresponding graph $\mathcal{G}_{\mathcal{T}}$.

strategy for constructing an input graph on which the optimal profit is at least R times the profit of $\mathcal{S}$.

Fix some $R < 6$ and let D be a non-negative integer (that depends on R) whose value will be specified later. It is convenient to describe the graph constructed by the adversary in terms of its underlying *skeleton tree* $\mathcal{T}$, which is a rooted binary tree. The root of $\mathcal{T}$ will be denoted by r. For a node $v \in \mathcal{T}$, define the *depth* or *level* of v to be the number of edges on the simple path from v to r. The adversary will only use skeleton trees of the following special form: each non-leaf node at depths $0, 1, \ldots, D-1$ has two children, and each non-leaf node at levels at least D has one child. Such a tree can be thought of as consisting of its *core subtree*, which is a complete binary tree of depth D, with paths attached to its leaves at level D. The nodes of $\mathcal{T}$ at depth D are the leaves of the core subtree. If v is a node of the core subtree of $\mathcal{T}$ then the path extending from v down to a leaf of $\mathcal{T}$ is called a *tentacle* – see Figure 1. (Thus v belongs both to the core subtree and to a tentacle attached to v.) The length of a tentacle is the number of its edges. The nodes in the tentacles are all considered left children of their parents.

The graph represented by a skeleton tree $\mathcal{T}$ will be denoted by $\mathcal{G}_{\mathcal{T}}$. We differentiate between the *nodes* of $\mathcal{T}$ and the *vertices* of $\mathcal{G}_{\mathcal{T}}$. The relation between $\mathcal{T}$ and $\mathcal{G}_{\mathcal{T}}$ is illustrated in Figure 1. $\mathcal{G}_{\mathcal{T}}$ is obtained from $\mathcal{T}$ as follows:

- For each node $u \in \mathcal{T}$ we create two vertices $u^{\rm L}$ and $u^{\rm R}$ in $\mathcal{G}_{\mathcal{T}}$, with an edge between them. This edge $(u^{\rm L}, u^{\rm R})$ is called the *cross edge* corresponding to u.
- Suppose that $u, v \in \mathcal{T}$. If u is in the left subtree of v then $(u^{\rm L}, v^{\rm L})$ and $(u^{\rm R}, v^{\rm L})$ are edges of $\mathcal{G}_{\mathcal{T}}$. If u is in the right subtree of v then $(u^{\rm L}, v^{\rm R})$ and $(u^{\rm R}, v^{\rm R})$ are edges of $\mathcal{G}_{\mathcal{T}}$. These edges are called *upward edges*.
- If $u \in \mathcal{T}$ is a node in a tentacle of $\mathcal{T}$ and is not a leaf, then $\mathcal{G}_{\mathcal{T}}$ has a vertex $u^{\rm D}$ with edge $(u^{\rm D}, u^{\rm R})$. This edge is called a *whisker*.

The adversary constructs $\mathcal{G}_{\mathcal{T}}$ gradually, in response to $\mathcal{S}$'s choices. Initially, $\mathcal{T}$ is a single node r, and thus $\mathcal{G}_{\mathcal{T}}$ is a single edge $(r^{\rm L}, r^{\rm R})$. At this time, $\mathsf{profit}_{\mathcal{S}}(\mathcal{T}) = 0$ and $\mathsf{O}(\mathcal{T}) = 1$, so $\mathcal{S}$ is forced to collect this edge (that is, it creates a 2-clique $\{r^{\rm L}, r^{\rm R}\}$).

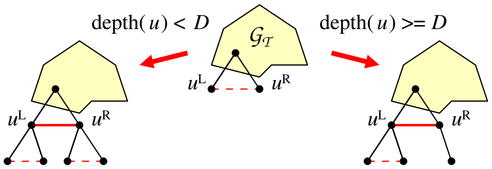

Fig. 2. Adversary moves. Upward edges from new vertices are not shown, to avoid clutter. Dashed lines represent cross edges that are not collected by $\mathcal{S}$, while thick lines represent those that are already collected by $\mathcal{S}$.

In general, the strategy will be able to collect only cross edges. Suppose that, at some step, $\mathcal{S}$ collects a cross edge $(u^{\mathrm{L}}, u^{\mathrm{R}})$, corresponding to node u of $\mathcal{T}$. If u is at depth less than D, the adversary extends $\mathcal{T}$ by adding two children of u. If u is at depth at least D, the adversary only adds the left child of u, thus extending the tentacle ending at u. In terms of $\mathcal{G}_{\mathcal{T}}$, the first move adds two triangles to u^{L} and u^{R}, with all corresponding upward edges. The second move adds a triangle to u^{L} and a whisker to u^{R} (see Figure 2).

Thus the adversary will be building the core binary skeleton tree down to level D, and from then on, it will extend the tentacles. Our objective is to prove that after each step the ratio between the adversary profit and the strategy's profit is at least $6 - O(1/D)$. This is enough to prove the lower bound. The reason is this: If the strategy stops collecting edges at some point, the ratio is $6 - O(1/D)$, and we are done. Otherwise, suppose that the game lasts for a very long time, and since D is fixed, then at least one tentacle will grow without bound. But the optimal cost is at least quadratic with respect to the maximum tentacle length s, while $\mathcal{S}$'s profit is only linear in s. Thus eventually the adversary can simply stop playing, and even if the strategy collects the remaining cross edges (and there will be at most $2^D \cdot s$ of those), the ratio will be larger than 6.

Denote by $\mathcal{T}_v$ the subtree of $\mathcal{T}$ rooted at v. To simplify the computation of the adversary (or optimal) profit, we will assume that the adversary computes his clustering recursively, as follows:

(opt1) If x is a leaf of $\mathcal{T}$, then x^{L} and x^{R} are in the same cluster.

(opt2) Suppose that x is an internal node of $\mathcal{T}$ and let y be the left child of x. Assume that the clustering of $\mathcal{T}_y$ is already computed. If x has a right child, let z be this child and assume that the clustering of $\mathcal{T}_z$ is already computed. Then

(opt2.a) x^{L} is added either to the cluster of $\mathcal{T}_y$ containing y^{L} or to the cluster containing y^{R}. (When we estimate the adversary profit, we will specify which choice we use.) This is correct, since all neighbors of y^{L} and y^{R} that correspond to nodes in $\mathcal{T}_y$ are also neighbors of x^{L}. Note that in the special case when y is a leaf, the clusters of y^{L} and y^{R} are the same.

(opt2.b) If x has the right child z, then the rule for adding x^{R} to the clustering of $\mathcal{T}_z$ is symmetric to (opt2.a). If x does not have the right child (so x is in a tentacle), then we create the "whisker" cluster consisting of two vertices x^{R} and x^{D}.

Observe that, in particular, all clusters, except for the whisker clusters, have at least three vertices.

We stress that the profit of the clustering computed as above (even for the way we specify the adversary choices in (opt2.a) and (opt2.b)) may not be actually maximized, but this does not matter, since for the purpose of our proof we only need a lower bound on the adversary profit.

We now claim that before the core tree reaches its target height D the ratio is at least 6. Indeed, consider one step, when $\mathcal{S}$ collects an edge $(u^{\mathrm{L}}, u^{\mathrm{R}})$. (See Figure 2.) The strategy's profit increases by 1. As for the adversary, he can increase his profit as follows:

(i) Create a new clique that is a triangle consisting of u^{R} and two new vertices, increasing the profit by 3.

(ii) In the current clique that contained u^{L} and u^{R}, replace u^{R} by the two new vertices connected to u^{L}. This current clique had size at least 3 (the adversary will maintain the invariant that in his clustering each cross edge is in a clique of size at least 3) and its size increases by 1, so its profit increases by at least 3.

Overall, the adversary's profit increases by at least 6, proving the claim.

Thus from now on it is sufficient to analyze skeleton trees of height strictly larger than D, namely trees that have at least one tentacle already started. Let $\mathcal{T}$ be such a skeleton tree. We will focus on analyzing the profits of the adversary and the strategy on such trees $\mathcal{T}_v$, where v is a node in the core subtree of $\mathcal{T}$. If $\mathcal{T}_v$ ends at depth $D+1$ or more, we call it a *bottom subtree*. The *core depth* of a bottom subtree $\mathcal{T}_v$ is defined as the depth of the part of $\mathcal{T}_v$ within the core subtree of $\mathcal{T}$. If h and s are, respectively, the core depth of $\mathcal{T}_v$ and its maximum tentacle length, then $0 \leq h \leq D$ and $s \geq 1$.

For a subtree $X = \mathcal{T}_v$, let $\mathsf{O}(X)$ be the optimal profit in X, computed according to the description above, and $\mathsf{S}(X)$ be $\mathcal{S}$'s profit (the number of cross edges). The lemma below is key in our argument.

Lemma 2. *Let X be a bottom subtree of height $h \geq 0$ and maximum tentacle length $s \geq 1$. Then*

$$\mathsf{O}(X) + 2(h+s) \geq 6 \cdot \mathsf{S}(X).$$

Before proving the lemma, let us argue first that this lemma is sufficient to establish our lower bound. Indeed, since we are now considering the case when $\mathcal{T}$ is a bottom subtree itself, the lemma implies that $\mathsf{O}(\mathcal{T}) + 2(D+s) \geq 6 \cdot \mathsf{S}(\mathcal{T})$, where s is the maximum tentacle length of $\mathcal{T}$. But $\mathsf{O}(\mathcal{T})$ is at least quadratic in $D+s$. So for large D the ratio $\mathsf{O}(\mathcal{T})/\mathsf{S}(\mathcal{T})$ approaches 6.

So now we prove Lemma 2. The proof is by induction on h, the core height of X. Consider first the base case, for $h = 0$ (when X is just a tentacle). The adversary has one clique of $s+2$ vertices, namely all x^{L} vertices in the tentacle

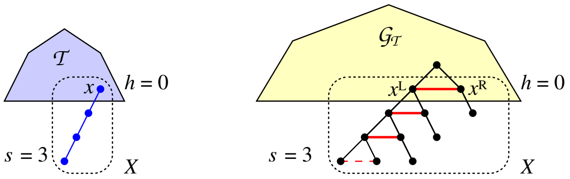

Fig. 3. Illustration of the inductive proof, the base case. Subtree X on the left, the corresponding subgraph on the right.

(there are $s+1$ of these), plus one z^{R} vertex for the leaf z. He also has s whiskers, so his profit for X is $\binom{s+2}{2} + s = \frac{1}{2}(s^2 + 5s + 2)$. The strategy collects only s cross edges, namely all cross edges in X except last. (See Figure 3.) Solving the quadratic inequality and using the integrality of s, we get $\mathsf{O}(X) + 2s \geq 6s = 6 \cdot \mathsf{S}(X)$. Note that this inequality is in fact tight for $s = 1, 2$.

In the inductive step, consider a bottom subtree $X = \mathcal{T}_u$. Let Y and Z be its left and right subtrees, respectively. Without loss of generality, we can assume that Y is a bottom tree with height $h-1$ and the same maximum tentacle length s as X, while Z is either not a bottom tree (that is, it has no tentacles), or it is a bottom tree with maximum tentacle length at most s.

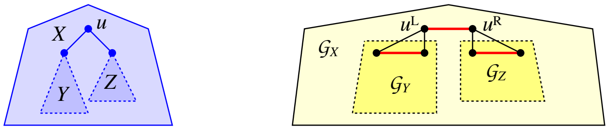

Fig. 4. Illustration of the inductive proof, the inductive step. Subtrees X, Y, Z on the left, the corresponding subgraphs on the right.

By the inductive assumption, we have $\mathsf{O}(Y)+2(h-1+s) \geq 6\cdot\mathsf{S}(Y)$. Regarding Z, if Z is not a bottom tree then $\mathsf{O}(Z) \geq 6 \cdot \mathsf{S}(Z)$, and if Z is a bottom tree (necessarily of height $h-1$) then $\mathsf{O}(Z) + 2(h-1+s') \geq 6 \cdot \mathsf{S}(Z)$, where s' is Z's maximum tentacle length, such that $1 \leq s' \leq s$.

Consider first the case when Z is not a bottom tree. Note that

$$\mathsf{S}(X) = \mathsf{S}(Y) + \mathsf{S}(Z) + 1 \quad \text{and} \quad \mathsf{O}(X) \geq \mathsf{O}(Y) + \mathsf{O}(Z) + h + s + 4$$

The first equation is trivial, because for X the strategy gets all cross edges in Y and Z, plus one more cross edge $(u^{\mathrm{L}}, u^{\mathrm{R}})$. The second inequality holds because u^{L} can be added to Y's largest cluster which has $(h-1) + s + 2 = h + s + 1$ vertices, and u^{R} can be added to Z's largest cluster that has at least 3 vertices. Then we get (since $h, s \geq 1$):

$$\begin{aligned} \mathsf{O}(X) + 2(h+s) &\geq [\mathsf{O}(Y) + \mathsf{O}(Z) + h + s + 4] + 2(h+s) \\ &= [\mathsf{O}(Y) + 2(h-1+s)] + \mathsf{O}(Z) + 6 \\ &\geq 6 \cdot \mathsf{S}(Y) + 6 \cdot \mathsf{S}(Z) + 6 = 6 \cdot \mathsf{S}(X). \end{aligned}$$

The second case is when Z is a bottom tree (of the same core height $h-1$) and maximum tentacle length s', where $1 \leq s' \leq s$. As before, we have $\mathsf{S}(X) = \mathsf{S}(Y) + \mathsf{S}(Z) + 1$. The optimum profit satisfies (by a similar argument as before. applied to both Y and Z):

$$\mathsf{O}(X) \geq \mathsf{O}(Y) + \mathsf{O}(Z) + 2h + s + s' + 2.$$

Then we get (using $s \geq s'$):

$$\begin{aligned} \mathsf{O}(X) + 2(h+s) &\geq [\mathsf{O}(Y) + \mathsf{O}(Z) + 2h + s + s' + 2] + 2(h+s) \\ &\geq [\mathsf{O}(Y) + 2(h-1+s)] + [\mathsf{O}(Z) + 2(h-1+s')] + 6 \\ &\geq 6 \cdot \mathsf{S}(Y) + 6 \cdot \mathsf{S}(Z) + 6 = 6 \cdot \mathsf{S}(X). \end{aligned}$$

This completes the proof of Lemma 2, for the case of the absolute competitive ratio.

We still need to explain how to extend our proof so that it also applies to the asymptotic competitive ratio. This is quite simple: Choose some large constant M. The adversary will create M instances of the above game, playing each one independently. Our construction above used the fact that at each step the strategy was forced to collect one of the pending cross edges, for otherwise its competitive ratio would exceed ratio R (where R was arbitrarily close to 6). Now, for M sufficiently large, the strategy will be forced to collect cross edges in all except for some finite number of copies of the game, where this number depends on the additive constant in the competitiveness bound.

Note: Our construction is very tight, in the following sense. Suppose that the strategy maintains $\mathcal{T}$ as balanced as possible. Then the ratio is exactly 6 when the depth of $\mathcal{T}$ is 1 or 2. Further, suppose that D is very large and the strategy constructs $\mathcal{T}$ to have depth D or more. Then the ratio is $6 - o(1)$ for $s = 1$ and $s = 2$. The intuition is that when the adversary plays optimally, he will only allow the online strategy to collect isolated edges (cliques of size 2). For this reason, we conjecture that 6 is the optimal competitive ratio.

4 Conclusions

We have shown an improved strategy with competitive ratio 15.645 for the problem of clique clustering where the objective is to maximize the number of edges in the cliques. Our strategy uses doubling to guarantee that the optimal measure does not become significantly larger than the strategy's measure. In fact, it is possible to prove (this result is omitted from this paper because of space constraints) that any strategy that uses doubling in this manner cannot achieve a competitive ratio better than 10.927.

We also prove that no strategy whatsoever can achieve a competitive ratio better than 6. Evidently, tightening these bounds would be of significant interest.

References

1. Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.
2. Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.
3. Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998.
4. Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. *SIAM J. Comput.*, 33(6):1417–1440, 2004.
5. Kamalika Chaudhuri, Brighten Godfrey, Satish Rao, and Kunal Talwar. Paths, trees, and minimum latency tours. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 36–45, 2003.
6. Marek Chrobak and Mathilde Hurand. Better bounds for incremental medians. *Theor. Comput. Sci.*, 412(7):594–601, 2011.
7. Marek Chrobak, Claire Kenyon, John Noga, and Neal E. Young. Incremental medians via online bidding. *Algorithmica*, 50(4):455–478, 2008.
8. Marek Chrobak and Claire Kenyon-Mathieu. SIGACT news online algorithms column 10: competitiveness via doubling. *SIGACT News*, 37(4):115–126, 2006.
9. Anders Dessmark, Jesper Jansson, Andrzej Lingas, Eva-Marta Lundell, and Mia Persson. On the approximability of maximum and minimum edge clique partition problems. *Int. J. Found. Comput. Sci.*, 18(2):217–226, 2007.
10. Aleksander Fabijan, Bengt J. Nilsson, and Mia Persson. Competitive online clique clustering. In *Proc. 8th International Conference on Algorithms and Complexity (CIAC'13)*, pages 221–233, 2013.
11. Andres Figueroa, James Borneman, and Tao Jiang. Clustering binary fingerprint vectors with missing values for DNA array data analysis. *Journal of Computational Biology*, 11(5):887–901, 2004.
12. Guolong Lin, Chandrashekhar Nagarajan, Rajmohan Rajaraman, and David P. Williamson. A general approach for incremental approximation and hierarchical clustering. *SIAM J. Comput.*, 39(8):3633–3669, 2010.
13. Claire Mathieu, Ocan Sankur, and Warren Schudy. Online correlation clustering. In *27th International Symposium on Theoretical Aspects of Computer Science (STACS'10)*, pages 573–584, 2010.
14. Lea Valinsky, Gianluca Della Vedova, Ra J. Scupham, Sam Alvey, Andres Figueroa, Bei Yin, R. Jack Hartin, Marek Chrobak, David E. Crowley, Tao Jiang, and James Borneman. Analysis of bacterial community composition by oligonucleotide fingerprinting of rRNA genes. *Applied and Environmental Microbiology*, 68:2002, 2002.