Representing Graph Metrics with Fewest Edges

T. Feder *, A. Meyerson **, R. Motwani ***, L. O'Callaghan †, and R. Panigrahy ‡

Carnegie-Mellon University and Stanford University

Abstract. We are given a graph with edge weights, that represents the metric on the vertices in which the distance between two vertices is the total weight of the lowest-weight path between them. Consider the problem of representing this metric using as few edges as possible, provided that new "steiner" vertices (and edges incident on them) can be added. The compression factor achieved is the ratio k between the number of edges in the original graph and the number of edges in the compressed graph. We obtain approximation algorithms for unit weight graphs that replace cliques with stars in cases where the cliques so compressed are disjoint, or when only a constant number of the cliques compressed meet at any vertex. We also show that the general unit weight problem is essentially as hard to approximate as graph coloring and maximum clique.

1 Introduction

Suppose we are given a finite metric space, represented as a graph $G = (V, E)$ on n nodes, with positive edge weights $l(e)$. We wish to find a graph $G' = (V', E')$, where $V \subseteq V'$, such that $|E'|$ is substantially smaller than E while ensuring that the metric is preserved exactly (i.e., pairwise distances for the vertices in V remain the same). The compression achieved by an algorithm is the ratio $k = |E|/|E'|$. If V' is constrained to be exactly V (i.e., if we are not allowed to add any vertices), we can find the edge-minimal graph in polynomial time. If we are allowed to add new vertices, the problem becomes more complex. We give polynomial-time algorithms which find graphs that are approximately edge-minimal.

* Email: tomas@theory.stanford.edu

** Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213. Research supported by NSF Grant CCR-0122581 and ARO Grants DAAG-55-98-1-0170 and DAAG-55-97-1-0221. Email: awm@stanford.edu

*** Department of Computer Science, Stanford University, Stanford, CA 94305. Research supported by NSF Grant IIS-0118173, an Okawa Foundation Research Grant, and Veritas. Email: rajeev@cs.stanford.edu.

† Department of Computer Science, Stanford University, Stanford, CA 94305. Research supported by an NSF Graduate Fellowship, an ARCS Fellowship, and NSF Grants IIS-0118173, IIS-9811904, and EIA-0137761. Email: loc@cs.stanford.edu

‡ Cisco Systems. Email: rinap@cisco.com

Main Techniques The key tool in our algorithms is the following. Consider the case in which we are given a collection of weighted graphs H_i called compressions, where distinct H_i may only share vertices in G. The candidate weighted graphs G' for the compression problem described above are obtained by selecting some of the compressions H_i, taking their union, and adding some edges from G itself. We show that if some such G' achieves a compression ratio k, then we can find one such G' achieving compression ratio at least $k/\log k$.

Suppose we are given a graph G in that has unit edge weights. We show that we achieve the best compression by replacing cliques by stars — the vertices of each replaced clique become the leaves of a star whose center is a new vertex, and each edge has weight $1/2$. In general, it is hard to select an appropriate collection of stars H_i to which to apply the compression algorithm. We show that this can be done if G satisfies certain degree constraints. We also study the case where G is weighted and sparse, and the H_i are trees.

Summary of Results The unit weight problem varies in hardness depending on whether we consider very special optima and solutions or more general ones. We summarize the results that we have obtained. At one end of the spectrum, when we consider the compression of a single clique, we obtain constant factor positive and negative approximation results. At the other end of the spectrum, where we are considering the compression of an arbitrary number of cliques that intersect arbitrarily, we cannot hope to obtain approximation algorithms, since the problem is essentially as hard to approximate as graph coloring and maximum clique. Three intermediate levels exhibit intermediate hardness in approximation: With respect to an optimum that compresses arbitrarily many disjoint cliques, the approximation factor achieved is the logarithm of the optimum compression; for the compression of two cliques that are not necessarily disjoint, the approximation factor achieved is the square root of the optimum compression; and with respect to an optimum that compresses arbitrarily many cliques that are not necessarily disjoint but where each clique compressed meets only a constant number of the other cliques compressed, the approximation factor achieved exceeds the square root of the optimum compression by a logarithm. Thus five levels of generality in the problem give five levels of hardness of approximation.

We describe the results in more detail now. In the unit weight case, we look for algorithms that perform well relative to an optimal solution in which the cliques corresponding to the H_i share no vertices. For the problem of finding a single clique, we give a linear-time 2-approximate algorithm, and show that the problem is as hard to approximate as vertex cover, hence hard to approximate within $7/6$ by the result of Håstad [15] and within 1.3606 by the result of Dinur and Safra [8]. More generally, if some r disjoint cliques achieve compression ratio k, then we can find n cliques such that some r of them achieve compression ratio at least $k/3$. The earlier algorithm applied to these n cliques achieves compression at least $k/(3\log k)$.

The problem where the cliques for the optimal choice of H_i may share vertices is harder to approximate. If compressing two cliques achieves ratio k, then we can find two cliques that give compression $\sqrt{k}/4$. We look for algorithms that perform well relative to an optimal solution in which each clique compressed meets at most r of the other cliques compressed; call such a solution an *r-sparse* solution. If an r-sparse solution with r constant achieves compression ratio k, then we find a solution achieving compression at least $\sqrt{k}/c\log k$, where c depends on r.

We then consider a related problem, where G is bipartite, and the stars H_i must have leaves forming a complete bipartite subgraph of G. Feder and Motwani [10] considered this problem for dense graphs and achieve compression factor $\log n/\log(n^2/m)$ on graphs with n vertices and m edges. The case of a single star H_i again has a 2-approximation algorithm, this time based on parametric flows. Here also, if disjoint stars achieve compression k, then we find a collection of stars containing a subset achieving ratio $k/3$, and we can then find a subset achieving ratio $k/(3\log k)$. With respect to an optimal r-sparse solution with r constant, we also get an algorithm with compression at least $\sqrt{k}/(c\log k)$ in the bipartite case.

Finally, we show that the unit weight compression problem is hard to approximate within a factor $n^{\frac{1}{2}-\epsilon}$ on instances with optimum compression at most $n^{\frac{1}{2}}$ for any constant $\epsilon > 0$ unless NP=ZPP, and for $\epsilon = 1/(\log n)^{\gamma}$ for some constant $\gamma > 0$ unless NP $\subseteq$ ZPTIME$(2^{(\log n)^{O(1)}})$.

Related Work. In 1964, Hakimi and Yau [14] defined the *optimal realization* of a distance matrix — a graph that preserves shortest-path distances while minimizing the *total sum of edge weights*. They give a solution in the case where the optimal realization is a tree. Since then, there has been substantial work on this problem [4, 9, 17, 18], and Althofer [1] has established the NP-hardness of finding the optimal realization if the matrix entries are integral. Chung *et al.* [6] give an algorithm to find a graph that is approximately optimal in the above sense, and in which the shortest-path distances are no shorter than in the given distance matrix.

Under the model of Arya *et al.* [2], Das *et al.* [7], and Rao and Smith [19], the vertices are points in Euclidean space, and the goal is to find *spanners* — subgraphs of the complete Euclidean graph that approximately preserve shortest path distances and have approximately minimal total weight (i.e., the sum of weights of edges in the subgraph). In a similar vein, Gupta [13] shows that some vertices can be removed with only constant distortion to distances on the remaining vertex set. Other related work includes that of Bartal [3], who introduces the idea of probabilistically approximating metric spaces by distributions over sets of other metric spaces, and of Charikar *et al.* [5], who derandomize this algorithm.

2 Generic Compression Algorithm

A *weighted graph* is a graph G with a positive weight on each edge. The distance $d(x, y)$ between two vertices x, y in G is the minimum over all paths p from x to y in G of the sum of the weights on p. If for three distinct vertices x, y, z we have $d(x, y) = d(x, z) + d(z, y)$, then the edge (x, y) is redundant and can be removed from the graph. If G is initially a complete graph on n vertices, and G'

is obtained from G by removing redundant edges so that G' has only m edges, then we can obtain G' from G in $O(n(m + n \log n))$ time.

A *compression* for G is a weighted graph H such that $V = V(G) \cap V(H)$ is nonempty, with $d_H(x, y) \geq d_G(x, y)$ for all x, y in V, and such that H does not have an edge (x, y) with both x, y in V. We say that G' is the weighted graph obtained by applying compression H to G if G' is obtained from $G \cup H$ by removing all edges (x, y) in G with both x, y in V such that $d_H(x, y) = d_G(x, y)$. Let C be a set of compressions for G. We can obtain a weighted graph G' by successively applying each of the compressions in C, starting with G. We say that C compresses m to m/k if $|E(G)| = m$ and $|E(G')| = m/k$. We also say that C has *compression factor* k.

Suppose we are given a set C of candidate compressions, and suppose that some subset of C has compression factor k. Theorem 1 establishes that we can find a subset of C with compression factor at least $k/(1 + \log k)$. The next step is to determine which weighted graphs H should be used as compressions. We focus on the *unit weight* case, where every edge of G has weight 1. Theorem 2 shows that we can assume without loss of generality that H is a star with edges of weight $1/2$ whose leaves form a clique in G.

Theorem 1 *Assume G has arbitrary weights, and let C be a given set of candidate compressions. Suppose that some subset of C compresses m to m/k. Then we can find in polynomial time a subset of C that compresses m to at most $\frac{m}{k}(1 + \log k)$.*

Proof Sketch: Each compression H_i we apply replaces $\lambda_i p_i$ edges with p_i new edges, for some $\lambda_i > 1$. [1] Let r be the number of edges from G that we do not replace in this way. Then the original graph G has $r + \sum_i \lambda_i p_i = m$ edges, while the graph G' obtained by applying the compressions H_i has $r + \sum_i p_i = m'$ edges.

The algorithm is greedy: Repeatedly select the compression H_i with $\lambda_i > 1$ largest. Define the compression factor of an edge e in G to be the value $s(e) = 1/\lambda_i$ when the algorithm uses a compression H_i to replace $\lambda_i p_i$ edges including e with just p_i edges; let $s(e) = 1$ otherwise. In the end, the number of edges in G' will be $\sum_{e \in E} s(e)$.

When $s > m/k$ edges remain that have not been removed by applying a compression, since the optimal solution compresses them to at most m/k edges, the compression factor for the edges replaced when the next H_i is applied by the algorithm is at most $m/(ks)$. Therefore $\sum_e s(e) \leq \frac{m}{k} + \sum_{\frac{m}{k} < s \leq m} \frac{m}{ks} = \frac{m}{k}(1 + H_m - H_{\frac{m}{k}}) \leq \frac{m}{k}(1 + \log k)$. □

The following theorem implies that the question, "Can we reduce the number of edges by p by adding a new vertex?" is NP-complete.

Theorem 2 *In the unit weight case, one can assume without loss of generality that each compression H used is a star with edges of weight $1/2$ whose leaves form a clique in G.*

[1] Note that if we use two compressions H_i and H_j that would both replace a common edge e, and we apply H_i first, then we will credit only H_i for the replacement of e.

Proof Sketch: Let H be a compression for G. Suppose H has an edge (x_0, y) with x_0 in G and $d_H(x_0, y) < 1/2$. Then there is no vertex $x \neq x_0$ in $V(G) \cap V(H)$ such that $d_H(x, y) \leq 1/2$; otherwise we would have $d_G(x_0, x) \leq d_H(x_0, x) < 1$. Consequently, if $d_H(x', y) + d_H(x'', y) = 1$ for some x', x'' in G, such that the edge (x', x'') can be removed from G, then one of x', x'' must be x_0. We can obtain a smaller H' by removing y and its incident edges (y, y') from H and adding edges (x_0, y') for each such $y' \neq x_0$, with $d_{H'}(x_0, y') = d_H(x_0, y) + d_H(y, y')$.

Therefore, we can assume that if H has an edge (x, y) with x in G, then $d_H(x, y) \geq 1/2$. An edge (x', x'') in G can thus only be removed if x' and x'' have a common neighbor y in H $(y \notin V)$ with $d_H(x, y) = d_H(x', y) = 1/2$. That is, the compression H is a union of stars with edges of weight $1/2$ whose leaves form a clique in G. $\square$

3 Compression and the Disjoint Optimum

We continue to assume our graph G has unit weights. We have seen that in this case we can assume that each compression corresponds to a clique in G. It remains to determine which cliques should be chosen for compression. We consider here a comparision with a compression that compresses either a single clique or disjoint cliques.

Theorem 3 *In the unit weight case, compression by selecting a single clique has a 2-approximation algorithm that runs in $O(m)$ time.*

Proof Sketch: We consider the unit weight case with a single additional vertex, and give a 2-approximation algorithm. If the maximum clique has size k, then the optimal compression is from m to $m + k - \binom{k}{2} = \alpha\binom{k}{2}$. With no compression, we have $m \leq (\alpha+1)\binom{k}{2}$ edges, for an approximation factor of $(\alpha+1)/\alpha = 1 + 1/\alpha$, giving the result for $\alpha \geq 1$. We now focus on the case $\alpha < 1$.

We repeatedly remove vertices of degree at most $k-2$, until every vertex has degree at least $k-1$. There are $m - \binom{k}{2} = \alpha\binom{k}{2} - k$ edges not in the clique of size k. If there are v vertices not in the clique, then the number of edges not in the clique is at least $(k-1)v/2$, which gives $v \leq \alpha(k-1)$. The number of edges not in the clique is also at least $(k-1)v - \binom{v}{2}$. The inequality $(k-1)v - \binom{v}{2} \leq \alpha\binom{k}{2} - k$ yields $v \leq (k-1) + \frac{1}{2} - \sqrt{((1-\alpha)(k-1)^2 + (3-\alpha)(k-1) + 9/4)}$, which implies $v \leq (1 - \sqrt{1-\alpha})(k-1)$. The v vertices form a vertex cover in the complement graph. Since vertex cover has a 2-approximation algorithm by means of a maximal matching, we can obtain a vertex cover with at most $2v$ vertices, and the vertices not in the vertex cover give a clique in the original graph with $l \geq k - v$ vertices. Compressing this clique, the resulting number of edges is $m + l - \binom{l}{2} = (\alpha+1)\binom{k}{2} - k + l - \binom{l}{2} \leq (\alpha+1)\binom{k}{2} - \binom{l}{2} \leq 2\alpha\binom{k}{2}$. This last inequality follows from the equivalent inequality $\binom{l}{2} \geq (1-\alpha)\binom{k}{2}$, since $l - 1 \geq \sqrt{1-\alpha}(k-1)$ and $l \geq \sqrt{1-\alpha}(k)$.

The algorithm is as follows:

1. Find a sequence of graphs G_t, where $G_1 = G$, and if v_t is a vertex of minimum degree d_t in G_t, then $G_{t+1} = G_t \setminus \{v_t\}$ is obtained by removing v_t and its incident edges from G_t.
2. Find a maximal matching M_t in the complement graph $\overline{G}_t$ for each t. A maximal matching M_t in $\overline{G}_t$ can be obtained from a maximal matching M_{t+1} in $\overline{G}_{t+1}$ by letting $M_t = M_{t+1} \cup \{(v_t, u)\}$ if v_t has a neighbor u in $\overline{G}_t$ such that u is not in an edge of M_{t+1}; otherwise $M_t = M_{t+1}$.
3. The vertices in G_t not incident to an edge of M_t form a clique Q_t — compress the largest such clique Q_t.

□

Theorem 4 *If compression by selecting a single clique has an α-approximation algorithm with $\alpha < 2$, then vertex cover has an $(\alpha + \epsilon)$-approximation algorithm for all $\epsilon > 0$.*

Proof Sketch: Let G be an instance of vertex cover with n vertices and minimum vertex cover of size b. We can assume $b > 1/\epsilon$, since otherwise the minimum vertex cover can be found by considering all subsets of size b. Let G' be the graph on N vertices obtained by adding at least n/ϵ vertices to G, with no new edges.

Consider the complement graph $\overline{G}'$ as an instance of the single clique problem. The maximum clique in $\overline{G}'$ has size $N - b$, and after compressing this clique we have $\mathrm{OPT} \leq N(b+1)$ edges in the compressed graph.

Use the α-approximation algorithm to find a solution that compresses a clique of size $N - a$, thus giving a vertex cover of size a in the original graph. We have $\alpha\mathrm{OPT} \geq \mathrm{SOL} \geq (N - a - 1)(a - b) + \mathrm{OPT}$, implying that $(N - a - 1)(a - b) \leq (\alpha - 1)\mathrm{OPT} \leq (\alpha - 1)N(b+1)$. This gives $(N - a)a \leq \alpha N(b+1)$, implying that $a \leq \frac{1+\frac{1}{b}}{1-\frac{a}{N}}\alpha b \leq \frac{1+\epsilon}{1-\epsilon}\alpha b \leq (1 + 3\epsilon)\alpha b$. We have an $\alpha + \epsilon'$ approximation if we let $\epsilon = \epsilon'/(3\alpha)$. □

Theorem 5 *In the unit weight case, suppose r disjoint cliques compress m to m/k. Then we can find n cliques such that some r of them compress m to at most $3m/k$.*

Proof Sketch: Consider the r disjoint cliques Q_i of size q_i. Let $d_i + q_i - 1$ be the minimum degree of a vertex in Q_i. Then there are at least $d_i q_i$ edges coming out of Q_i so $\frac{m}{k} \geq \sum_i \frac{d_i q_i}{2}$. If $d_i \geq q_i$, then not compressing Q_i costs at most $\binom{q_i}{2} \leq \frac{d_i q_i}{2}$ extra edges. Suppose next $d_i < q_i$. Let v_i be a vertex of degree $d_i + q_i - 1$ in Q_i. Let G_i be the graph induced by v_i and its neighbors. The complement graph $\overline{G}_i$ has a vertex cover of size d_i consisting of the d_i vertices not in Q_i. We can find a maximal matching M_i on $\overline{G}_i$. The matching M_i will have at most d_i edges, and involve at most d_i vertices in Q_i. The vertices of G_i not incident to an edge of M_i give a clique R_i that has at least $q_i - d_i$ vertices in Q_i. Compressing R_i leaves at most $d_i(q_i - 1)$ edges of Q_i not compressed. Furthermore R_i has at most d_i more vertices than Q_i, so the total extra cost is at most $d_i(q_i - 1) + d_i \leq d_i q_i$ extra edges. The total extra cost for the entire graph is therefore at most $\sum_i d_i q_i \leq 2m/k$ extra edges. □

The next result follows immediately from Theorems 1 and 5.

Theorem 6 *In the unit weight case, suppose r disjoint cliques compress m to m/k. Then we can find a compression from m to at most $3m/k(1+\log(k/3))$.*

4 Compression and the NonDisjoint Optimum

It is more difficult to obtain algorithms that perform as well with respect to the optimum which can compress cliques that are not necessarily disjoint.

Theorem 7 *In the unit weight case, suppose two (not necessarily disjoint) cliques compress m to m/k. Then we can find two cliques that compress m to at most $4m/\sqrt{k}$.*

Proof Sketch: Suppose the two cliques Q_1 and Q_2 have a vertices in common, and are of size $a+b_1$ and $a+b_2$ respectively, with $b_2 \leq b_1$. Let r be the number of edges not in the two cliques, and write $r = d_1 b_1 = d_2 b_2$.

Suppose first $d_2 \leq b_2$. Then $d_1 \leq b_1$ as well. For $i = 1, 2$, some vertex v_i out of the b_i vertices in clique Q_i but not in the other clique has at most d_i edges incident to v_i and not in Q_i.

As in the proof of Theorem 5, we can find a clique R_i contained in the graph induced by v_i and its neighbors, so that when we compress it, the extra number of edges is at most $d_i(a+b_i)$. The total number of edges after both R_1 and R_2 are compressed is thus at most $\frac{m}{k} + d_1(a+b_1) + d_2(a+b_2) \leq 3\frac{m}{k} + 2d_2 a$ with $(d_2 a)^2 = d_2^2 a^2 \leq r(2m) \leq \frac{2m^2}{k}$, so that $d_2 a \leq \frac{\sqrt{2}m}{\sqrt{k}}$. Suppose next $d_2 > b_2$. If we only compress Q_1, the total number of edges resulting is at most $\frac{m}{k} + b_2(a+b_2) \leq 2\frac{m}{k} + b_2 a$ with $(b_2 a)^2 = b_2^2 a^2 \leq r(2m) \leq \frac{2m^2}{k}$, so that $b_2 a \leq \frac{\sqrt{2}m}{\sqrt{k}}$. We can find a 2-approximation to compressing Q_1 by Theorem 3.

In both cases, the bound is at most $4\frac{m}{k} + \frac{2\sqrt{2}m}{\sqrt{k}} = (2\sqrt{2} + \frac{4}{\sqrt{k}})\frac{m}{\sqrt{k}}$.

If $k \leq 16$, then the m original edges give a $4m/\sqrt{k}$ bound; if $k \geq 16$ then the above bound is at most $(2\sqrt{2}+1)m/\sqrt{k}$ and the result follows. □

Consider a solution involving some l compressed cliques H_i, such that each H_i intersects at most r other H_i, for constant r. Suppose this solution has compression factor k. We define *sectors* so that two vertices are in the same sector if and only if the set of H_i to which they belong is same. The number of sectors within a clique H_i is at most s for some $s \leq 2^r$. We define an *associated graph* whose vertices are the sectors S_j; two sectors are adjacent if they belong to the same clique H_i for some i. The max-degree in the associated graph is $d \leq rs$.

Theorem 8 *There is a constant c such that we can find a collection of at most $n^{\lfloor d/2 \rfloor + 1}$ cliques containing a subcollection of at most ls^2 cliques that achieve compression factor at least $\sqrt{k}/(cd^4)$.*

Proof Sketch: Consider two adjacent sectors S_1 and S_2, and allow $S_1 = S_2$. Consider the sector S_3 adjacent to both S_1 and S_2 that has the largest number v of vertices in it; we allow S_3 to be S_1 or S_2 as well. Suppose the number of edges joining S_1 to S_2 is at most $v^2/\sqrt{k}$. We charge these edges to the $v^2/2$ edges of S_3. The sector S_3 will be so charged at most d^2 times, so the total charge is at most $md^2/\sqrt{k}$.

Conversely, suppose the number of edges joining S_1 to S_2 is at least $v^2/\sqrt{k}$. Then both S_1 and S_2 have at least $v/\sqrt{k}$ vertices. Let Q be a maximal clique in the associated graph containing S_1, S_2, S_3, such that each sector in Q has at least $v/\sqrt{k}$ vertices. For each sector S_i in Q, let u_i be the vertex in S_i that has the smallest number t_i of edges incident to it going to vertices in sectors S_j such that S_i and S_j are not adjacent. Let H be the induced subgraph whose vertices are all the vertices w that are either equal to some u_i or adjacent to all u_i.

We can find a single clique that gives a 2-approximation in H as in Theorem 5. The bound on the number of edges of Q not compressed plus the number of additional edges in the compression is fq, where f is the number of vertices in H that are not in Q, and q is the size of Q. We bound f and q. Clearly $q \leq dv$.

There are at most d sectors adjacent to all sectors in Q but not in Q, and each such sector has at most $v/\sqrt{k}$ vertices, for a total of $dv/\sqrt{k}$ vertices. Multiplying this quantity by q gives at most $d^2v^2/\sqrt{k}$ edges, which can be charged again to the $v^2/2$ edges of S_3. The sector S_3 will be so charged at most d^2 times, so the total charge is at most $md^4/\sqrt{k}$.

The remaining t vertices have at least one neighbor u_i such that their edge to u_i is not compressed in the optimum. Thus, $t \leq dt_i$ for some d; multiplying this quantity by q gives at most d^2vt_i edges, which can be charged to the $\frac{v}{\sqrt{k}}t_i$ edges not compressed coming out of S_i. Again S_i will be charged at most d^2 times, for a total charge of $d^4\sqrt{k}$ per edge not compressed in the optimum. Since the number of edges not compressed in the optimum is at most $\frac{m}{k}$, the total charge is at most $\frac{md^4}{\sqrt{k}}$.

Finally, each vertex is involved in at most d^2 cliques Q, giving at most $nd^2 \leq \frac{md^2}{k}$ new edges.

The algorithm is thus as follows: For each choice of at most d vertices u_i forming a clique, find a single clique in the graph of the common neighbors of the u_i as in Theorem 5. We can reduce the number of chosen vertices to $\lfloor d/2 \rfloor + 1$ as follows. Either Q has at most this many sectors, or the number of sectors not in Q adjacent to a chosen sector in Q is at most $\lceil d/2 \rceil - 1$. We will need to choose at most $\lceil d/2 \rceil - 1$ extra sectors in Q to rule out the neighbor sectors that are not common neighbors of all sectors in Q, for a total of $\lceil d/2 \rceil \leq \lfloor d/2 \rfloor + 1$ chosen vertices. □

Theorem 9 follows immediately from Theorems 1 and 8.

Theorem 9 *We can find a collection of cliques achieving compression factor* $\sqrt{k}/(cd^4 \log k)$.

In general, it is not possible to find all the cliques and apply the generic compression algorithm. However, this can be done if all vertices have degree $O(\log n)$, or for slighly smaller cliques in a graph of a slighly larger degree. We consider again the sequence of graphs G_t, where $G_1 = G$, and if v_t is a vertex of minimum degree d_t in G_t, then $G_{t+1} = G_t \setminus \{v_t\}$ is obtained by removing v_t

and its incident edges from G_t. Every clique in G is a clique containing v_t in G_t for some t.

Theorem 10 *If v_t has degree $d_t \leq f_t \log n$ in G_t, then we can find the polynomially many cliques containing v_t in G_t of size $O(\log n / \log f_t)$. These are all the cliques if d_t is $O(\log n)$.*

In the weighted case, for a vertex v and a constant c, we denote by $N^c(v)$ the set of vertices joined by a path with at most c edges to v in G. A *good compression* H has all its vertices from G inside $N^c(v)$ for some v.

Theorem 11 *In the weighted case, there are polynomially many good compressions by trees of size $O(\log n / \log \log n)$ in a graph of maximum degree $O(\log^d n)$, and these can be found in poly-time.*

5 Bipartite Compression

Consider the following situation. We have identified two cliques R_1, R_2 to be compressed, and every additional clique Q that we may compress has vertices in either R_1 or R_2. We may assume $V(R_1) \cap V(R_2) = \emptyset$ and $V(R_1) \cup V(R_2) = V(G)$. Then $G' = G \setminus (R_1 \cup R_2)$ is a bipartite graph $G' = (V(R_1), V(R_2), E)$. Compressing a clique Q in G corresponds to compressing a complete bipartite subgraph of G', which we refer to as a *bi-clique*.

We consider here the case where G' has a collection of r bi-cliques sharing no vertices giving a compression factor k, and obtain three results analogous to the three results in Section 3. We consider then the optimum where every bi-clique compressed meets at most a constant number of the other bi-cliques compressed, and has compression k. We obtain results analogous to those in Section 4.

Theorem 12 *Compression by a single bi-clique has a 2-approximation algorithm.*

Proof Sketch: Suppose the optimal bi-clique Q has q_1 vertices in R_1 and q_2 vertices in R_2. The optimal compression is thus from m to $m + q_1 + q_2 - q_1 q_2$. Let v_1 be a vertex in $Q \cap R_1$ of minimum degree $q_2 + d_2$, and let v_2 be a vertex in $Q \cap R_2$ of minimum degree $q_1 + d_1$. The number of edges not compressed incident to Q is at least $s = q_1 d_2 + q_2 d_1$. Let $\hat{G}$ be the subgraph induced by the $q_1 + d_1$ neighbors of v_2 and the $q_2 + d_2$ neighbors of v_1.

We consider first the case where $q_1 = q_2 = q$. Find a maximal matching M in the bipartite complement of $\hat{G}$. The veritces not in M form a bi-clique T in $\hat{G}$. Note that M has at most d_2 vertices in $Q \cap R_1$ and at most d_1 vertices in $Q \cap R_2$. Thus the number of edges in Q but not in T is at most $qd_2 + qd_1 = s$. This gives a 2-approximation when we compress T instead of Q.

When q_1 and q_2 are not necessarily equal, we can define $\hat{G}'$ obtained from $\hat{G}$ by making q_2 copies of each vertex in $\hat{G} \cap R_1$ and q_1 copies of each vertex in $\hat{G} \cap R_2$. Now Q gives a bi-clique Q' in G' with $q_1 q_2 = q_2 q_1 = q$ vertices in each side.

A maximal matching M' in the complement of $\hat{G}'$ has at most $d_2 q_1$ vertices in $Q' \cap R_1'$ and at most $d_1 q_2$ vertices in $Q' \cap R_2'$. The vertices not in M' form a bi-clique T' in $\hat{G}'$. The number of edges in Q' but not in T' is at most $qd_2q_1 + qd_1q_2 = qs$. We can add vertices to T' until the vertices not in T' form a minimal vertex cover in the complement of $\hat{G}'$. Then T' will have either all or none of the q_2 copies of a vertex in R_1, and either all or none of the q_1 copies of a vertex in R_2, so T' corresponds to a bi-clique T in $\hat{G}$, and the number of edges in Q but not in T is at most s.

We may thus try all possible pairs of values q_1, q_2. Alternatively, we can find minimum instead of minimal vertex covers in the complement of $\hat{G}$. For a parameter $0 < \lambda < 1$, assign weight λ to the vertices in R_1 and weight $1 - \lambda$ to the vertices in R_2. We can find a collection of at most $1 + \min(|\hat{G} \cap R_1|, |\hat{G} \cap R_2|)$ weighted minimum vertex covers over all values of λ, by a parametric flow [12]. One of these weighted minimum vertex covers will correspond to $q_1/q_2 = (1 - \lambda)/\lambda$. □

Theorem 13 *Suppose r disjoint bi-cliques compress m to m/k. Then we can find at most n^3 bi-cliques such that some r of them compress m to at most $3m/k$.*

Theorem 14 *Suppose r disjoint bi-cliques compress m to m/k. Then we can find a compression from m to at most $3\frac{m}{k}(1 + \log \frac{k}{3})$.*

Consider a solution involving some l compressed bi-cliques H_i, such that each H_i intersects at most r other H_i, for constant r. Suppose this solution has compression factor k. We define *sectors* so that two vertices in the same R_p ($p = 1, 2$) are in the same sector if and only if the set of H_i to which they belong is the same. The number of sectors within a bi-clique H_i and in the same R_p is at most s for some $s \leq 2^r$. We define an *associated graph* whose vertices are the sectors S_j, where two sectors are adjacent if they belong to the same clique H_i for some i and they are in different R_p. The max-degree in the associated graph is $d \leq rs$.

Theorem 15 *There is a constant c such that we can find a collection of at most n^{d+2} bi-cliques containing a subcollection of at most ls^2 bi-cliques that achieve compression factor at least $\sqrt{k}/(cd^4)$.*

Theorem 16 follows from Theorem 15 by the algorithm of Theorem 1.

Theorem 16 *We can find a collection of bi-cliques achieving compression factor $\frac{\sqrt{k}}{cd^4 \log k}$.*

6 Hardness of approximation

We establish the following result and its corollary.

Theorem 17 *Finding an* $(\frac{r}{4+\log_p n})$*-approximation for the unit weight compression problem, on instances with* n^2 *vertices and optimum compression factor at most* n*, is as hard as finding an independent set of size* n/rp *in a* p*-colorable graph with* n *vertices, where* r*,* p *may depend on* n*.*

Proof Sketch: Let G be a p-colorable graph where we wish to find an independent set of size n/rp. We assume $n = pq$, where p and q are prime numbers. We define a graph H with n^2 vertices of the form (x, y, z), where $0 \leq x < n$, $0 \leq y < p$, and $0 \leq z < q$. We view x, y, z as integers modulo n, p, q respectively.

The graph H has all the edges between two vertices (x, y, z) and (x', y', z') such that $x \neq x'$ and $y \neq y'$. The number of such edges is $M_1 = n(n-1)p(p-1)q^2/2$. In addition, H has all the edges between two vertices (x, y, z) and (x', y, z) such that $x \neq x'$ and (x, x') is not an edge in G. The number of such edges is at most $M_2 = n(n-1)pq/2$.

We exhibit a compression of H, using the fact that G is p-colorable. We shall not compress the M_2 edges, although these edges may belong to compressed cliques. Note that $M_2 \leq \frac{1}{(p-1)q} M_1$.

Let R be the clique consisting of the n vertices $(x, y, 0)$ such that vertex x in G has color y in the p-coloring. For $0 \leq i < p$, $1 < j < p$, and $0 \leq k, l < q$, let R_{ijkl} be the clique consisting of the n vertices $(x, i + jy, k + ly)$ such that $(x, y, 0)$ is in R. These $p(p-1)q^2$ cliques compress all the edges between two vertices (x, y, z) and (x', y', z') with $x \neq x'$ and $y \neq y'$ and such that x and x' have different colors in the p-coloring, and introduce $np(p-1)q^2 \leq \frac{2}{n-1} M_1$ new edges.

It remains to compress the edges between two vertices (x, y, z) and (x', y', z') with $x \neq x'$ and $y \neq y'$ and such that x and x' have the same color d in the p-coloring. Let s_d be the number of vertices of color d, so that $\sum_{0 \leq d < p} s_d = n$. There exist $\lceil \log_p s_d \rceil$ p-colorings of the s_d vertices such that every pair of distinct vertices among the s_d vertices gets different colors in at least one such p-coloring. Using the previous argument, we find $p(p-1)q^2 \lceil \log_p s_d \rceil$ cliques of size s_d for the vertices (x, y, z) with x of color d. The number of new edges introduced is $\sum_d s_d p(p-1)q^2 \lceil \log_p s_d \rceil \leq np(p-1)q^2 \lceil \log_p n \rceil \leq \frac{2\lceil \log_p n \rceil}{n-1} M_1$. Thus, we have a compression factor k such that $\frac{1}{k} \leq \frac{1}{(p-1)q} + \frac{2}{n-1} + \frac{2\lceil \log_p n \rceil}{n-1} \leq \frac{6+2\log_p n}{n-1}$. Suppose we can find a compression of H with compression factor l. Then the compression includes a clique with s vertices in H such that $(s-1)/2 \geq l$. This clique gives is a p-coloring of s vertices of G, and hence an independent set of size at least s/p in G. If this independent set is of size smaller than n/rp, then $s \leq n/p$ and $l \leq n/2r \leq (\frac{4+\log_p n}{r})k$. □

Feige and Kilian [11] prove that chromatic number is hard to approximate within factor $n^{1-\epsilon}$ for any constant $\epsilon > 0$, unless NP=ZPP. This implies that it is also hard to find an independent set of size $\frac{n^\epsilon}{p}$ for any constant $\epsilon > 0$, where p is the chromatic number of the graph; otherwise we could repeatedly select large independent sets and get a coloring with $pn^{1-\epsilon} \log n$ colors. Since the graphs in the preceding theorem have size n^2, setting $r = n^{1-\epsilon}$ gives the following.

Theorem 18 *The unit weight compression problem is inapproximable in polynomial time within factor $n^{\frac{1}{2}-\epsilon}$ on instances with optimum compression factor at most $n^{\frac{1}{2}}$ for any constant $\epsilon > 0$, unless NP=ZPP.*

Khot [16] improves the chromatic number result to $\epsilon = \frac{1}{(\log n)^{\gamma}}$ for a constant $\gamma > 0$, if it is not the case that NP $\subseteq$ ZPTIME$(2^{(\log n)^{O(1)}})$. This can be carried over to the above theorem as well.

References

1. I. Althofer. "On optimal realizations of finite metric spaces by graphs." *Discrete Comp. Geom* 3, 1988.
2. S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. H. M. Smid. "Euclidean spanners: short, thin, and lanky." In *Proc. STOC*, 1995.
3. Y. Bartal. "Probabilistic approximation of metric spaces and its algorithmic applications." In *Proc FOCS*, 1996.
4. F. Boesch. "Properties of the distance matrix of a tree." *Quart. Appl. Math.* 26 (1968-69), 607–609.
5. M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. Plotkin. "Approximating a finite metric by a small number of tree metrics." In *Proc. FOCS*, 1998.
6. F. Chung, M. Garrett, R. Graham, and D. Shallcross. "Distance realization problems with applications to internet tomography." Preprint, `http://www.math.ucsd.edu/~fan`.
7. G. Das, G. Narasimhan, and J. Salowe. "A new way to weigh malnourished Euclidean graphs." In *Proc. SODA*, 1995.
8. I. Dinur and M. Safra. Personal communication.
9. A. W. M. Dress. "Trees, tight extensions of metric spaces, and the cohomological dimension of certain groups." Advances in Mathematics 53 (1984), 321–402.
10. T. Feder and R. Motwani. "Clique compressions, graph partitions and speeding-up algorithms." *JCSS* 51 (1995), 261–272.
11. U. Feige and J. Kilian. "Zero-knowledge and chromatic number." In *Proc. Annual Conf. on Comp. Complex.* (1996).
12. G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. "A fast parametric maximum flow algorithm and applications." *SICOMP 18* (1989) 30–55.
13. A. Gupta. "Steiner points in tree metrics don't (really) help." In *Proc. 12th SODA* 2001, pp 220-227.
14. S. L. Hakimi and S. S. Yau. "Distance matrix of a graph and its realizability." *Quart. Appl. Math.* 22 (1964), 305–317.
15. J. Håstad. "Some optimal inapproximability results." In *Proc STOC* (1997) 1–10.
16. S. Khot. "Improved inapproximability results for max clique, chromatic number and approximate graph coloring." In *Proc FOCS* (2001).
17. J. Nieminen. "Realizing the distance matrix of a graph." *Elektron. Informationsverarbeit. Kybernetik* 12(1-2):1976, 29–31.
18. J. Pereira. "An algorithm and its role in the study of optimal graph realizations of distance matrices." *Discrete Math.* 79(3):1990, 299–312.
19. S. B. Rao and W. D. Smith. "Improved approximation schemes for geometrical graphs via spanners and banyans." In *Proc. STOC* (1998), 540–550.