The Dense k-Subgraph Problem *

Uriel Feige † Guy Kortsarz ‡ David Peleg §

April 13, 1999

Abstract

This paper considers the problem of computing the dense k-vertex subgraph of a given graph, namely, the subgraph with the most edges. An approximation algorithm is developed for the problem, with approximation ratio $O(n^{\delta})$, for some $\delta < 1/3$.

1 Introduction

We study the *dense k-subgraph* (DkS) maximization problem, of computing the dense k-vertex subgraph of a given graph. That is, on input a graph G and a parameter k, we are interested in finding a set of k vertices with maximum average degree in the subgraph induced by this set. As this problem is NP-hard (say, by reduction from Clique), we consider approximation algorithms for this problem. We obtain a polynomial time algorithm that on any input (G, k) returns a subgraph of size k whose average degree is within a factor of at most n^{δ} from the optimum solution, where n is the number of vertices in the input graph G, and $\delta < 1/3$ is some universal constant. Unfortunately, we are unable to present a complementary negative result giving evidence that for some $\epsilon > 0$, achieving an approximation ratio of $O(n^{\epsilon})$ is NP-hard. In fact, we do not even know whether achieving an approximation ratio of $(1+\epsilon)$ is NP-hard, though we conjecture that this is indeed the case.

*A preliminary version of this paper appeared in the proceedings of the 34th Annual Symposium on Foundations of Computer Science, 1993, published by IEEE Computer Society Press, pages 692–701.

†Department of Applied Math and Computer Science, The Weizmann Institute, Rehovot 76100, Israel. Incumbent of the Joseph and Celia Reskin Career Development Chair. Yigal Alon Fellow.

‡Department of Computer Science, The Open University, Tel Aviv, Israel.

§Department of Applied Mathematics and Computer Science, The Weizmann Institute, Rehovot 76100, Israel. Supported in part by a Walter and Elise Haas Career Development Award and by a grant from the Basic Research Foundation.

Our problem is related to several other problems. We mention two of them.

- The *Densest Subgraph* (DS) problem concerns choosing a subset V' (of arbitrary size) such that the vertex induced subgraph has maximum average degree. This problem can be solved polynomially using flow techniques (cf. Chapter 4 of [Law76]). The fastest algorithm known for *DS* is given in [GGT89] and runs in time $O(mn\log(n^2/m))$. One may hope that some algorithmic techniques used in solving the DS problem can help approximate the DkS problem, but there seem to be major difficulties involved. Consider for example the case of regular graphs. The densest subgraph of a regular graph is the graph itself, and hence no algorithmic ideas are involved in solving this DS problem. On the other hand, finding the dense k-subgraph remains NP-hard (proof omitted).

- The *Minimum Flux Cut* (FLUX) problem concerns choosing a cut C with minimum ratio between the number of edges that cross the cut and the number of vertices in the smaller side of the cut. This is a measure of the edge expansion of the graph. The FLUX problem on regular graphs is related to the DkS problem in the following sense: by solving the DkS problem optimally for all values of k in a regular graph, one can deduce the optimal solution to the FLUX problem on this graph. The FLUX problem can be approximated within a factor of $O(\log n)$ [LR88].

We mention two special cases of the dense k-subgraph problem that make it easier to approximate. First, if $k = \Omega(n)$ and the number of edges is $\Omega(n^2)$, then the problem has a polynomial time approximation scheme (PTAS) [AKK95]. Secondly, if the input graph is a complete graph with edge weights that obey the triangle inequality, then it is shown in [RRT91] that a greedy algorithm achieves an approximation ratio of 4 for the *dispersion* problem, which asks for the k-vertex subgraph of maximum total edge weight, and an approximation ratio of 2 is given in [HRT].

Recently, Goemans (private communication) showed that using semidefinite programming (SDP) one can obtain an approximation ratio arbitrarily close to n/k for DkS. For some graphs and large values of k, this approximation ratio is better by a constant factor than that of the greedy algorithm (see Section 3.2). However, for small values of k, algorithms based on SDP are not known to perform as well as our combinatorial approximation algorithm. For example, when $k \simeq n^{1/3}$, it appears that the SDP approach cannot distinguish between graphs that have cliques of size k and graphs that only have k-vertex subgraphs with $O(k)$ edges [FS] (in particular, excluding an approximation ratio better than $n^{1/3}$).

Our algorithm can be extended to handle the weighted version of the DkS problem, incurring an additional $O(\log n)$ factor. This is done in section 5.2.

2 Definitions

Definition 2.1 *The* density d_G *of a graph* $G = G(V, E)$ *is its average degree. That is,* $d_G = 2|E|/|V|$. *When* G *is clear from the context, we shall denote the density by* d.

There is a polynomial time algorithm for finding the densest vertex induced subgraph of an input graph. We study the parameterized version of this problem.

Definition 2.2 *The* dense k-subgraph *(DkS) problem has as input a graph* $G = G(V, E)$ *(on* n *vertices), and a parameter* k. *The output is* G^*, *a subgraph of* G *induced on* k *vertices, such that* G^* *is of maximum density. We denote this density by* $d^*(G, k)$.

Clearly, the problem DkS is NP-hard, by reduction from Clique.

We are interested in polynomial time approximation algorithms for DkS. On input (G, k), such an algorithm outputs a list of k vertices. Let $A(G, k)$ denote the density of the vertex induced subgraph returned by algorithm A on input (G, k). We wish to devise polynomial time algorithms with $A(G, k)$ as close as possible to $d^*(G, k)$. We shall bound $A(G, k)$ as a function of n (the number of vertices in G), k and $d^*(G, k)$.

Notation. $\Delta(G)$ is the maximum degree of graph G. d_H is the average degree of the $k/2$ vertices of highest degree in G. Note that $\Delta(G) \geq d_H \geq d^*(G, k)$. $deg(v, S)$ is the number of edges connecting vertex v to vertices in the set S. $cut(A, B)$ is the number of edges connecting vertices in set A and vertices in set B. A *walk* of length ℓ is a sequence of $\ell + 1$ vertices in which consecutive vertices are adjacent (hence the walk follows ℓ edges). The vertices of a walk need not be distinct. $W_\ell(u, v)$ denotes the number of walks of length ℓ that start at vertex u and end at vertex v. Matrix multiplication (raising the adjacency matrix of the graph to the ℓth power) can be used in order to compute $W_\ell(v_i, v_j)$ for all pairs of vertices simultaneously.

3 An approximation ratio of $O(n^{1/3})$

Theorem 3.1 *There is a polynomial time algorithm* A *that approximates DkS within a factor of* $2n^{1/3}$. *That is, for every graph* G *and every* $1 \leq k \leq n$, $A(G, k) \geq d^*(G, k)/2n^{1/3}$.

Algorithm A employs three different procedures (A_1, A_2 and A_3) to select a dense subgraph, and returns the densest of the three subgraphs that are found.

3.1 A trivial procedure

Without loss of generality, we can assume that G contains at least $k/2$ edges.

Procedure 1 *Select $k/2$ arbitrary edges from G. Return the set of vertices incident with these edges, adding arbitrary vertices to this set if its size is smaller than k.*

Clearly,

$$A_1(G,k) \geq 1 \ .$$

3.2 A greedy procedure

Procedure 2 *Sort the vertices by order of their degree. Let H denote the $k/2$ vertices with highest degrees in G (breaking ties arbitrarily). Sort the remaining vertices by the number of neighbors they have in H. Let C denote the $k/2$ vertices in $G \setminus H$ with largest number of neighbors in H. Return $H \bigcup C$.*

Recall that d_H denotes the average degree (with respect to G) of a vertex in H.

Lemma 3.2 *Procedure 2 returns a vertex induced subgraph satisfying*

$$A_2(G,k) \geq kd_H/2n \ .$$

Proof: Let m_1 denote the number of edges both of whose endpoints lie in H. Then $cut(H, V \setminus H) = d_H|H| - 2m_1 = d_H k/2 - 2m_1 \geq 0$. By the greedy rule for selecting C, at least a $|C|/|V \setminus H| > k/2n$ fraction of these edges are contained in $H \bigcup C$. Thus the total number of edges in the subgraph induced by $H \bigcup C$ is at least

$$(d_H k/2 - 2m_1)k/2n + m_1 \geq d_H k^2/4n$$

and the proof of the lemma follows. $\square$

As $d_H \geq d^*(G,k)$, the greedy procedure approximates $d^*(G,k)$ within a ratio of at most $2n/k$. A different greedy procedure which also has an approximation ratio of $O(n/k)$ is analyzed in [AITT].

3.3 Walks of length 2

For vertices v, w and integer $\ell \geq 1$, recall that $W_\ell(v, w)$ denotes the number of walks of length ℓ from v to w.

Procedure 3 *Compute $W_2(u, v)$ for all pairs of vertices. Construct a candidate graph $\mathcal{H}^v$ for every vertex v in G, as follows: Sort the vertices of G by non-increasing order of their number of length-2 walks to v, $W_2(v, w_1) \geq W_2(v, w_2) \geq \ldots$ Let P_h^v denote the set $\{w_1, \ldots, w_{k/2}\}$. Compute for every neighbor x of v the number of edges connecting x to P_h^v,* $\deg(x, P_h^v)$*, and construct a set B^v containing the $k/2$ neighbors of v with highest* $\deg(x, P_h^v)$*. Let $\mathcal{H}^v$ denote the subgraph induced on $P_h^v \cup B^v$. (If $\mathcal{H}^v$ still contains less than k vertices then it is completed to size k arbitrarily.) Select the densest candidate graph $\mathcal{H}^v$ as the output.*

We now analyze the approximation ratio of this procedure. Let us first note that the number of length-2 walks within the optimum subgraph G^* is at least $k(d^*(G, k))^2$. This is because each $v \in G^*$ contributes $(\deg^*(v))^2$ to this sum, and $\sum_{v \in G^*} (\deg^*(v))^2 \geq k(d^*(G, k))^2$ by convexity. (Here we used $\deg^*(v)$ to denote the degree of v in G^*. See also remark in the appendix.)

It follows that there is a vertex v which is the endpoint of at least $(d^*(G, k))^2$ length-2 walks in G^*. By the greedy construction of P_h^v, there are at least $(d^*(G, k))^2/2$ walks of length 2 between this v and vertices of P_h^v. The vertices of B^v have at least $(d^*(G, k))^2/2$ edges connecting them to P_h^v if $\deg(v) \leq k/2$, and at least $(d^*(G, k))^2 k/4 \deg(v)$ edges connecting them to P_h^v otherwise. Since we do not require P_h^v and B^v to be disjoint, each edge may have been counted twice. Hence, altogether, $\mathcal{H}^v$ contains at least $\min[(d^*(G, k))^2/4, (d^*(G, k))^2 k/8\Delta(G)]$ edges, where $\Delta(G)$ denotes the maximum degree in the graph.

This guarantees:

$$A_3(G, k) \geq (d^*(G, k))^2/2 \max[k, 2\Delta(G)] \ .$$

3.4 Algorithm A

Algorithm A applies the three procedures described above, and outputs the densest of the three subgraphs obtained by each of these procedures. Procedures 1 and 2 are applied to the original input graph G. Procedure 3 however is applied to the graph G_ℓ induced on the vertices of $V \setminus H$, where H is the set of $k/2$ vertices of highest degree in G, as defined in Procedure 2. Hence $\Delta(G_\ell) \leq d_H(G)$.

For the following lemma to make sense, we assume that $k \leq 2n/3$. This assumption can be made without loss of generality, because for $k \geq 2n/3$ the greedy procedure approximates DkS within a ratio not worse than 3 (see end of Section 3.2).

Lemma 3.3 *The graph G_ℓ contains a k-vertex induced subgraph with average degree at least $d^*(G,k) - 2d_2$, where $d_2 = A_2(G,k)$.*

Proof: Let m denote the number of edges of G^* with both endpoints in H, and let ℓ denote the number of edges of G^* with one endpoint in H. Hence G_ℓ contains a k-vertex induced subgraph with at least $d^*(G,k)k/2 - m - \ell$ edges. To prove the lemma, we need to show that Procedure 2 returns a solution with at least $(m+\ell)/2$ edges. In fact, the solution has at least $m + \ell/2$ edges. This is because it clearly contains the m edges internal to $V(G^*) \cap H$, and there must be at least $l/2$ edges between C and H, since at least one possible choice for C offers this many edges (namely, taking C to contain the $k/2$ vertices of $V(G^*) \setminus H$ with the highest number of edges into H). $\square$

It follows from the performance guarantees on the three procedures that

$$A(G,k) \geq \max[1, d_2, \frac{kd_H}{2n}, \frac{(d^*(G,k) - 2d_2)^2}{2\max[k, 2d_H]}] \ .$$

To prove Theorem 3.1, we can assume that $d_2 \leq d^*(G,k)/n^{1/3}$ (otherwise, the output of procedure 2 achieves the desired ratio of approximation). Hence, for procedure 3, we have that $d^*(G,k) - 2d_2 \simeq d^*(G,k)$, with a negligible error term. The performance guarantee of algorithm A is at least the geometric mean of the performance guarantee of the three procedures 1, 2, and 3. Hence

$$A(G,k) \geq (1 \cdot \frac{kd_H}{2n} \cdot \frac{(d^*(G,k))^2}{2\max[k, 2d_H]})^{1/3} \geq \frac{d^*(G,k)}{2n^{1/3}} \ ,$$

where the last inequality follows from the fact that $k \geq d^*(G,k)$ and $d_H \geq d^*(G,k)$.

4 Improving over $O(n^{1/3})$

The approximation ratio for algorithm A was upper bounded as a geometric mean of three approximation ratios. In order for Algorithm A to give an approximation ratio as bad as $\Omega(n^{1/3})$, it must hold that all three procedures, 1, 2, and 3, give an approximation ratio of $\Theta(n^{1/3})$. This happens only if $d^*(G,k) = \Theta(n^{1/3})$, $kd_H = \Theta(n)$, and $\max[k, d_H] = \Theta(n^{2/3})$. If any of the above three conditions is violated by as much as n^ϵ, then the approximation ratio is $O(n^{1/3-\epsilon/2})$. The above worst case conditions are satisfied only in the two cases below:

1. $d^*(G,k) = \Theta(n^{1/3})$. $k = \Theta(n^{1/3})$. $d_H = \Theta(n^{2/3})$.

2. $d^*(G,k) = \Theta(n^{1/3})$. $k = \Theta(n^{2/3})$. $d_H = \Theta(n^{1/3})$.

We present two additional procedures, each giving an approximation ratio better than $O(n^{1/3})$ in one of the above cases. Together with algorithm A, this guarantees an approximation ratio of $O(n^{1/3-\epsilon})$, for some $\epsilon > 0$, for the DkS problem.

Theorem 4.1 *There is a polynomial time algorithm B that approximates DkS within a factor of $n^{1/3-\epsilon}$, for some $\epsilon > 0$. That is, for every graph G and for every $1 \le k \le n$, $B(G,k) \ge d^*(G,k)/n^{1/3-\epsilon}$.*

A unifying theme of the two new procedures is the use of the following lemma. Recall that $W_\ell(v_i, v_j)$ denotes the number of walks of length ℓ from v_i to v_j.

Lemma 4.2 *Let G be a graph with n vertices and average degree d. There exist two vertices $v_i, v_j \in V$ such that*

$$W_\ell(v_i, v_j) \ge \frac{d^\ell}{n} .$$

A proof of Lemma 4.2 appears in the appendix.

In section 4.1 we treat case 1. In section 4.2 we treat case 2. In both cases, we assume that the following step has been performed:

Remove H, the set of $k/2$ vertices of highest degree, and remain with the graph G_ℓ.

We shall use the fact that $\Delta(G_\ell) \le d_H$. We further assume that $d^*(G,k)$ remains virtually unchanged by the step above. This assumption can be made without loss of generality, because it fails to hold only if Procedure 2 achieves an approximation ratio better than $n^{1/3-\epsilon}$ (see Lemma 3.3 and the discussion that follows it). We let $G_\ell{}^*$ denote the k-vertex induced subgraph of highest density in G_ℓ.

4.1 Walks of length 3

We first present a procedure that handles case 1 above ($d^*(G,k) = \Theta(n^{1/3})$, $k = \Theta(n^{1/3})$, $d_H = \Theta(n^{2/3})$). Its analysis is based on the following lemma.

Lemma 4.3 *There exist two vertices (not necessarily distinct) $v_i, v_j \in V$ such that the subgraph of $G_\ell{}^*$ induced by $N(v_i) \cup N(v_j)$ has at least $(d^*(G,k))^3/2k$ edges.*

Proof: Consider Lemma 4.2 with $\ell = 3$ applied to ${G_\ell}^*$, and let v_i, v_j be two vertices with $W_3[v_i, v_j] \geq (d^*(G,k))^3/k$. Consider the multiset of middle edges of all length three walks between v_i and v_j. An edge may appear in this multiset at most twice (e.g., once as (v_k, v_ℓ), and once as (v_ℓ, v_k), if both v_k and v_ℓ are in $N(v_i) \cap N(v_j)$). The proof follows. $\square$

We can now present procedure 4.

Procedure 4 *1. For all pairs of vertices $v_i, v_j \in G_\ell$, apply algorithm $A(N(v_i) \cup N(v_j), k)$.*

2. Return the densest of the subgraph returned by any of the $O(n^2)$ applications of algorithm A.

Lemma 4.4 *The performance guarantee of procedure 4 satisfies $A_4(G_\ell, k) \geq A(G', k)$, where G' is a graph on at most $n' = 2d_H$ vertices that contains a k-vertex subgraph of average degree at least $d' = (d^*(G,k))^3/k^2$.*

Proof: Let v_i and v_j be the two vertices in ${G_\ell}^*$ to which Lemma 4.3 applies. Then $N(v_i) \cup N(v_j)$ contains a k-vertex induced subgraph with at least $(d^*(G,k))^3/2k$ edges, implying average degree at least $(d^*(G,k))^3/k^2$. Moreover, $|N(v_i) \cup N(v_j)| \leq 2d_H$. $\square$

For case 1, we get $A_4(G_\ell, k) \geq A(G', k)$ with $n' = O(n^{2/3})$ and $d' = \Theta(n^{1/3}) = \Theta(d^*(G,k))$. Algorithm A achieves an approximation ratio of $O((n')^{1/3})$ (and in fact even better, for these parameters), which is certainly better than $O(n^{1/3})$.

4.2 Walks of length 5

We handle case 2, with parameters $d^*(G,k) = \Theta(n^{1/3})$, $k = \Theta(n^{2/3})$, $d_H = \Theta(n^{1/3})$ (and hence $\Delta(G_\ell) = O(n^{1/3})$). These parameters are fixed throughout this section. We present an outline of procedure 5 that is used in this case. We shall later fill in the missing details (how step 1 is performed). In what follows, $\epsilon > 0$ is a small universal constant.

Procedure 5 *1. Select a subgraph induced over $O(n^{2/3})$ vertices, with average degree $\Omega(n^\epsilon)$. Remove it from G_ℓ to obtain a new graph.*

2. Repeat the above step of selecting subsets of vertices and removing them from the input graph until one of the following stopping conditions occur:

(a) A total of $n^{2/3}$ vertices have been selected.

(b) One can deduce (by the fact that Claim 4.5 below fails to hold) that the remaining graph no longer contains a $\Theta(n^{2/3})$-vertex induced subgraph with average degree $\Omega(n^{1/3})$.

3. *Return the subgraph induced on the union of the vertices selected by applications of step 1 above. If stopping condition 2b occurred, complete to $n^{2/3}$ vertices in a greedy way, similar to the selection of C in Section 3.2.*

For the parameters of case 2, procedure 5 guarantees an approximation ratio of $O(n^{1/3}/n^{\epsilon})$. This is clearly the case if the first stopping condition occurs, because then the average degree of the subgraph found is $\Omega(n^{\epsilon})$. The same also applies to the case that the second stopping condition occurs after $n^{2/3}/2$ vertices have been selected. The only nontrivial case is when the second stopping condition occurs before $n^{2/3}/2$ vertices are selected, but then the approximation ratio can be shown to be a constant. The reason is that in this case, all but a small fraction of the edges of ${G_\ell}^*$ have at least one endpoint in the selected vertices. As long as there are $\Omega(n)$ edges of ${G_\ell}^*$ that do not have both endpoint in the selected vertices, there must be some vertex of ${G_\ell}^*$ that was not yet selected and has $\Omega(n^{1/3})$ neighbors in the selected vertices. The greedy rule for choosing C then ensures that a vertex of degree $\Omega(n^{1/3})$ will be chosen. The average degree of the final subgraph is $\Omega(n^{1/3})$.

The main unexplained part of procedure 5 is step 1. It uses analysis based on walks of length 5. For the parameters of case 2, applying Lemma 4.2 to ${G_\ell}^*$, we obtain:

Claim 4.5 *There exist two vertices u, v in G_ℓ with*

$$W_5(u,v) \geq (d^*(G,k))^5/k = \Omega(n) \ .$$

Let u and v be two vertices with $\Omega(n)$ length 5 walks from u to v. Let N_1, (N_2, N_3, N_4, respectively) denote the sets of vertices that are first (second, third, fourth, respectively) along these walks. Let F denote the subgraph induced by the union of these sets.

Note that a vertex w may appear in several of these sets. E.g., w may be a neighbor of v but also may lie in a path of length 2 from v. This fact may cause some edges to be counted several times and affects the constants in our analysis. This effect is taken care of by the O, Ω, Θ notation that we use.

From the assumption that $d_H = O(n^{1/3})$, step 1 of procedure 5 is applied on graphs with maximum degree $\Delta = O(n^{1/3})$. It follows that $|N_1|, |N_4| = O(n^{1/3})$ and $|N_2|, |N_3| = O(n^{2/3})$.

4.2.1 Some easy subcases

We make some assumptions regarding the structure of F. Each assumption is justified by the fact that it can either be enforced on F, or otherwise a subgraph of average degree $\Omega(n^{\epsilon})$ is found (and hence step 1 is completed).

Assumption 1: $cut(N_2, N_3) < n^{2/3+\epsilon}$.
Justification: Otherwise, take $N_2 \bigcup N_3$.

Assumption 2: For every $w \in N_2$, $W_3(w, v) \leq n^{1/3+\epsilon}$, and for every $w \in N_3$, $W_3(w, u) \leq n^{1/3+\epsilon}$.
Justification: Consider the case that $w \in N_2$ and $W_3(w, v) > n^{1/3+\epsilon}$. Observe that all the length 3 walks between w and v must pass through N_3 and N_4. Consider the graph induced by the neighbors of w in N_3, and the set N_4. Since w has $O(n^{1/3})$ neighbors in N_3, this graph contains $O(n^{1/3})$ vertices, and $\Omega(n^{1/3+\epsilon})$ edges. Hence step 1 is completed.

Assumption 3: Every edge between N_2 and N_3 lies in at least $\Omega(n^{1/3-2\epsilon})$ walks from v to u.
Justification: remove any edge between N_2 and N_3 that lies in less than $n^{1/3-2\epsilon}$ length 5 walks from v to u. Since the number of edges between N_2 and N_3 is less than $n^{2/3+\epsilon}$, (see assumption 1) we "kill" at most $O(n^{1-\epsilon})$ walks, maintaining $W_5(u, v) = \Omega(n)$.

4.2.2 The remaining subcase

Let $e = (w, z)$ be an arbitrary edge between $w \in N_2$ and $z \in N_3$. By assumption 3, e lies in $p = \Omega(n^{1/3-2\epsilon})$ walks from u to v.

Clearly,

$$p = deg(w, N_1) \cdot deg(z, N_4).$$

Thus, either $deg(w, N_1) \geq n^{1/6-\epsilon}$ or $deg(z, N_4) \geq n^{1/6-\epsilon}$. If $deg(z, N_4) \geq n^{1/6-\epsilon}$, call z the "good" vertex of e. Otherwise, call w the good vertex.

Now initiate the following process. The process chooses two subsets $S_2 \subseteq N_2$, $S_3 \subseteq N_3$ of "good" vertices, i.e., vertices of high degrees. Repeat the following 3 steps:

1. Choose an edge e between N_2 and N_3. Let w be its good vertex.

2. If $w \in N_2$, add w to S_2, otherwise, add w to S_3.

3. Remove from F all the edges between N_2 and N_3 that touch w.

Observe that in step 3 above, we only discard the length 5 walks from v to u in F that *go through* w. Assume w.l.o.g. that $w \in N_2$. By assumption 2, $W_3(w, v) \leq n^{1/3+\epsilon}$. The number of walks between u and w (which equals $deg(w, N_1)$) is bounded above by $n^{1/3}$. Thus, the number of walks between v and u that go through w, is bounded by $O(n^{1/3} \cdot n^{1/3+\epsilon}) = n^{2/3+\epsilon}$.

Since we have $\Omega(n)$ walks between v and u, and each iteration removes only $n^{2/3+\epsilon}$ of them, the number of iterations can be chosen to be $\Theta(n/n^{2/3+\epsilon}) = \Theta(n^{1/3-\epsilon})$. Thus the total number of "'good" vertices, found by the algorithm is $\Theta(n^{1/3-\epsilon})$.

W.l.o.g. assume that $|S_2| \geq |S_3|$. Now, consider the subgraph induced by $S_2 \cup N_1$. It contains $O(n^{1/3})$ vertices, out of which $\Theta(n^{1/3-\epsilon})$ vertices have degree at least $deg(w, N_1) \geq n^{1/6-\epsilon}$. Thus the average degree is $\Omega(n^{1/6-2\epsilon}) \geq n^{\epsilon}$, for $\epsilon \leq 1/18$. Hence we obtain:

Lemma 4.6 *For the parameters $k = \Theta(n^{2/3})$, $d^*(G,k) = \Theta(n^{1/3})$, and $d_H = \Theta(n^{1/3})$, procedure 5 achieves an approximation ratio of $O(n^{5/18})$.*

4.3 Algorithm B

Algorithm B applies algorithm A and the two procedures 4 and 5. To see that it obtains an approximation ratio of $O(n^{1/3-\epsilon})$, for some $\epsilon > 0$, observe that the analysis of procedures 4 and 5 can withstand small changes in the input parameters. For example, if $d_H \leq n^{1-6\epsilon}$ and $d^*(G,k) \geq k/n^{\epsilon/3}$ (implying $d' \geq k/n^{\epsilon}$) then procedure 4 has approximation ratio $O(n^{1/3-\epsilon})$.

We have made no attempt to compute the best value of ϵ that can be obtained by algorithm B, other than verify that $\epsilon > 0$.

5 Extensions

5.1 Better approximation when $d = \Omega(k)$

In the case that $d^*(G,k) = \Omega(k)$ it is possible to alternate between our procedures 2 and 4 and obtain an algorithm that for any given ϵ finds an $O(n^{\epsilon})$ approximation to DkS, with time complexity $n^{O(1/\epsilon)}$. In each iteration the algorithm first applies procedure 2, and stops if it produces a subgraph with average degree k/n^{ϵ}. If procedure 2 fails to produce such a subgraph, the $k/2$ vertices of highest degree are removed, and procedure 4 is applied. This results in $O(n^2)$ new DkS problems to be solved, but in each one of them the number of vertices has been reduced by a factor of $\Omega(n^{\epsilon})$. At least one of these smaller problems must contain a vertex induced subgraph of density roughly $\Theta(d^*(G,k))$. Now the process can be repeated on each of the smaller problems. After $O(1/\epsilon)$ iterations, we remain with DkS problems on graphs with k vertices or fewer, and we take the densest of these graphs. The details are omitted.

A simplified version of the above argument is used in [FS] to show that if G contains a clique on k vertices, then for every $\epsilon > 0$ a k-subgraph with average degree $(1-\epsilon)(k-1)$ can be found in time $n^{O(\frac{1}{\epsilon}\log\frac{n}{k})}$.

5.2 Arbitrary edge weights

In the weighted version of the DkS problem, edges have nonnegative weights, and the goal is to find the k-vertex induced subgraph with maximum total weight of edges. This problem can be reduced to the unweighted DkS problem with a loss of at most $O(\log n)$ in the approximation ratio. We sketch how this is done.

1. Scale edge weights such that the maximum edge weight is n^2.
2. Round up each edge weight to the nearest (nonnegative) power of two.
3. Solve $2\log n$ DkS problems, one for each edge weight (with all other edges removed).
4. Select the best of the $O(\log n)$ solutions.

References

[AFWZ] N. Alon, U. Feige, A. Wigderson and D. Zuckerman. Derandomized graph products. *Computational Complexity* 5 (1995), 60–75.

[AKK95] S. Arora, D. Karger and M. Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. *27th Symp. on Theory of Computing*, ACM, 284–293, 1995.

[AITT] Y. Asahiro, K. Iwama, H. Tamaki, T. Tokuyama. Greedily finding a dense subgraph. In *SWAT 96, LNCS 1097, 136–148.*

[Bi] Biggs. *Algebraic graph theory.* Cambridge University Press.

[FS] U. Feige and M. Seltser. On the densest k-subgraph problem. *Technical report CS97-16 of the Weizmann Institute, 1997.* Available at http://www.wisdom.weizmann.ac.il.

[GGT89] G. Gallo, M.D. Grigoriadis, and R.E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM J. on Comput.*, 18:30–55, 1989.

[HRT] R. Hassin, S. Rubinstein and A. Tamir. Approximation algorithms for Maximum Dispersion. To appear in *Operations Research Letters.*

[Law76] E.L. Lawler. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart and Winston, 1976.

[LR88] F.T. Leighton and S. Rao. An Approximate Max-FLow Min-Cut Theorem for Uniform Multicommodity Flow Problems with Applications to Approximation Algorithms. *29th Symp. on Foundations of Computer Science.* IEEE, 422-431, 1988.

[MM] M. Marcus and H. Minc. *A survey of matrix theory and matrix inequalities.* Allyn and Bacon, Inc., Boston, 1964.

[RRT91] S.S. Ravi, D.J. Rosenkrantz, and G.K. Tayi. Heuristic and special case algorithms for Dispersion problems. *Operations Research*, Vol. 42, No. 2, 1994, pages 299–310.

Appendix

We restate Lemma 4.2 and present its proof.

Lemma 4.2: *Let G be a graph with n vertices and average degree d. There exist two vertices $v_i, v_j \in V$ such that $W_\ell(v_i, v_j) \geq \frac{d^\ell}{n}$.*

Before proving the above lemma, we recall without proofs some elementary facts from linear algebra and its relation to graphs. For a more detailed treatment, see [Bi, MM].

Let $G(V, E)$ be a graph with $n = |V|$ vertices and $m = |E|$ edges where $V = \{v_1, \ldots, v_n\}$. The adjacency matrix $A(G)$ is the matrix $A(G) = (a_{ij})$ where a_{ij} is defined as

$$a_{ij} = \begin{cases} 1, & (v_i, v_j) \in E \\ 0, & (v_i, v_j) \notin E \end{cases}$$

The matrix is a $0-1$ symmetric matrix with 0 in the diagonal (as we deal with simple graphs).

We denote the eigenvalues of $A(G)$ by $\lambda_0, \ldots, \lambda_{n-1}$ (some eigenvalues may have multiplicity, i.e, the same value may appear many times). Since $A(G)$ is symmetric, all its eigenvalues are real. Without loss of generality assume that $\lambda_i \geq \lambda_{i+1}$ for $0 \leq i \leq n-2$.

For any square matrix $B = (b_{ij})$ we denote by $trace(B)$ the sum of elements in the diagonal of B, i.e., $trace(B) = \sum_{i=n}^{i=1} b_{ii}$. If B has eigenvalues $\mu_0, \ldots, \mu_{n-1}$ then:

$$trace(B) = \sum_{i=n-1}^{i=0} \mu_i \ .$$

(Hence the sum of the eigenvalues of $A(G)$ equals 0.)

The largest eigenvalue of the adjacency matrix of a graph satisfies $\lambda_0 \geq (\sum a_{ij})/n = d$.

When raising a square matrix A with eigenvalues $\lambda_0, \ldots, \lambda_{n-1}$ to some power k, the values of the eigenvalues of A^k are $(\lambda_0)^k, \ldots, (\lambda_{n-1})^k$.

We are now ready to prove Lemma 4.2.

Proof: Consider the adjacency matrix $A(G)$, and put $P = A(G)^\ell$ and $P = (p_{ij})$. Each entry p_{ij} counts the number of walks of length ℓ from v_i to v_j, i.e,

$$p_{ij} = W_\ell(i,j) \ .$$

Now consider the matrix $B = P^2 (= A(G)^{2\ell})$ and put $B = (b_{ij})$. Consider a diagonal element b_{ii}. Since $b_{ii} = \sum_{j=1}^n p_{ij} p_{ji}$, and the graph is undirected, we have that $b_{ii} = \sum_{j=1}^n {p_{ij}}^2$.

It follows that

$$\sum_{i,j} W_\ell(i,j)^2 = trace(B) = trace(A^{2\ell}) = \sum_{i=0}^{i=n-1} (\lambda_i)^{2\ell} \geq (\lambda_0)^{2\ell} \geq d^{2\ell} \ .$$

By averaging, there is a pair (i,j) such that $W_\ell(i,j)^2 \geq d^{2\ell}/n^2$ which gives the proof. □

Remark: Lemma 4.2 also follows from the fact that the total number of length ℓ walks in a graph of average degree d is at least nd^ℓ. A proof of this fact, but only for even values of ℓ, is presented in [AFWZ]. Unfortunately, we need to use this fact with odd values of ℓ. We are not aware of any reference to the corresponding result for odd values of ℓ, except for a recent private communication by Noga Alon.