Polynomial variants of the Densest/Heaviest k-subgraph Problem

Maria Liazi, Ioannis Milis & Vassilis Zissimopoulos

mliazi@di.uoa.gr, milis@aueb.gr, vassilis@di.uoa.gr

Department of Informatics and Telecommunications

University of Athens

Outline of the talk

- Previous work
- The HkS on chains
- The HkS on graphs with maximal degree 2
- The DkS on a subclass of proper interval graphs
- Open Questions

Definitions

- **Densest k-subgraph** (DkS):
  Input: $G = (V, E)$, $|V| = n$, and an integer k,
  $3 \leq k \leq n$
  Output: Finding a k-subgraph with maximum number of edges.

- **Heaviest k-subgraph** (HkS):
  The weighted version of DkS,
  the edges of the given graph have non negative weights and the goal is to find the k-vertices induced subgraph with the maximum total edge weight.

NP-hardness

DkS and HkS are NP-hard by a direct reduction from the *Clique* problem:
a graph G has a clique of k vertices iff G has a k-vertex subgraph with at least $\binom{k}{2}$ edges.

The problems remain NP-hard even for bipartite graphs of maximal degree 3.
[Feige et al. "On the densest k- subgraph problem"]

Applications:

- random generation of test instances,
- representation of the Web graph.

DkS: Approximation Algorithms

- $\frac{n}{k}$-approximation algorithm, (semidefinite programming).
  (gap of the optimum value: $\Omega(n^{\frac{1}{3}})$).
- $O(n^{\delta})$, for some $\delta < \frac{1}{3}$
  (performs well for all values of k).

  [Feige et al. "The dense k-subgraph problem"]
- When $k = \Omega(n)$ and the number of edges is $\Omega(n^2)$ there is a PTAS.

  [Arora et al. "PTASs for dense instances of NP-hard problems"]

HkS: Approximation Algorithms

- R-approximation algorithm, (greedy):

  $(1/2 + n/2k)^2 - O(n^{-1/3}) \leq R \leq (1/2 + n/2k)^2 + O(1/n)$, $n/3 \leq k \leq n$

  $2(n/k - 1) - O(1/k) \leq R \leq 2(n/k - 1) + O(n/k^2)$, $k < n/3$.

  [Asahiro et al. "Greedily finding a subgraph"]

- $O(n^{\frac{1}{3}} \log n)$-approximation algorithm.
- $\frac{n}{k}$-approximation algorithm, ($k \simeq n/3$, semidefinite programming).

  [Srivastav et al. "Finding dense subgraphs with semidefinite programming"]

- 2-approximation algorithm on a complete graph where the weights satisfy the triangle inequality, (greedy).

Polynomial special cases

For the DkS on graphs of maximum degree two there is an $O(kn)$ algorithm.

proof:
Consider the following KNAPSACK problem:

$$max \sum_{i=1}^{m} e_i x_i \text{ such that } \sum_{i=1}^{m} n_i x_i \leq k \text{ and } x_i \in \{0,1\}$$

where n_i is the number of vertices and e_i is the number of edges of the connected component $g_i,\ 1 \leq i \leq m$ of G, (g_i is either a cycle or a chain).

complexity: $O(kn)$

The HkS problem on chains

Theorem
There is an $O(nk^2)$ dynamic programming algorithm for solving the HkS problem on trees.
[Maffioli "Finding a best subtree of a tree"]

!!! Restriction: the optimal solution is connected.

??? Without the restriction of connectivity.

An outline of the proof

Reduction from HkS to POK.

Partially-Ordered Knapsack (POK):
Input: Knapsack + a partial order $P = (N, \prec)$ on the set of items
Output: a subset S of the items with the maximum value + for every item all its predecessors in P are also in S.

The POK problem

POK is strongly NP-hard for general bipartite posets.

$dimP$=minimum number of linear extensions needed to define P.

2-dimensional posets can be recognized in polynomial time,

recognizing whether $dimP = d$ for any $d \geq 3$ is NP-complete.

Example of the reduction

where $p_{u_i} = 0$, $w_{u_i} = 1$ and $p_{e_i} = w(u_i, u_{i+1})$, $w_{e_i} = 0$

The bipartite poset of the above example is 2-dimensional:

The two linear extensions are:
$u_1u_2e_1u_3e_2u_4e_3$ and $u_4u_3e_3u_2e_2u_1e_1$.

The HkS problem on chains

Proposition
The bipartite poset obtained by reducing a HkS instance on a chain to a POK instance is a 2-dimensional one.

Theorem
There is an $O(n^2W)$ pseudopolynomial algorithm for the POK problem on 2-dimensional posets.

[Kolliopoulos et al. "Partially-ordered knapsack and applications to scheduling"]

Theorem
There is an $O(kn^2)$ algorithm for the HkS problem on chains.

!!!For a set of chains, we first connect them into a single one by adding fictive edges between them.

Does this approach work for cycles?

!!! dimension > 2,

!!! the bipartite complement of its comparability graph is not a chordal bipartite graph.

Does this approach work for trees?

!!! dimension > 2 and ...

The HkS problem on graphs of maximal degree two

Let $g_i,\ 1 \leq i \leq m,$ a connected component (either a cycle or a chain) of a graph of maximal degree two.
n_i = the number of vertices of g_i
e_i = the number of edges of g_i, $(\sum_{i=1}^{m} n_i = n)$.

Lemma
There is an $O(kn^3)$ algorithm for the HkS problem on a cycle of n vertices.

An outline of the proof

Firstly the algorithm solves optimally the j vertices heaviest subgraph problem for each value j, $1 \leq j \leq \min\{n_i, k\}$ on each connected component g_i of G.

Then it combines one value of j per g_i such that $\sum j = k$.

Theorem

There is an $O(kn^3)$ algorithm for the HkS problem on graphs of maximal degree two.

The DkS problem on a subclass of proper interval graphs

Definitions:

- A graph $G = (V, E)$ is called interval if there is a mapping I of the vertices of G into sets of consecutive integers such that for each pair of vertices $u, v \in V$ the following is true $(u, v) \in E \Longleftrightarrow I(u) \bigcap I(v) \neq \emptyset$.
- A graph $G = (V, E)$ is called proper interval if the set of intervals can be chosen to be inclusion free.

The DkS problem on a subclass of proper interval graphs

Definitions:

- Clique graph of G: a weighted graph whose vertices correspond to the maximal cliques and there is an edge between two vertices iff the corresponding maximal cliques intersect, (Edge weight= number of vertices in the intersection).
- Clique tree of G: maximum weight spanning tree of its clique graph.
- The maximal cliques of an interval graph can be found in linear time.
- The clique tree of an interval graph is a simple path.

The DkS problem on a subclass of proper interval graphs

clique graph $\longmapsto$simple path.

An outline of the proof

The algorithm starts from the clique Q_1 and traverses the clique graph through the cliques Q_j, $j = 2, ..., z$.

An outline of the proof

$f^{m_j}(i)$: the value of the optimal solution.

$$f^{m_j}(i) = \max\{f_0^{m_j}(i), f_1^{m_j}(i,a)\},$$

for every m_j, $i = 1, ..., k$, $a = 1, .., q_j$,
$f_0^{m_j}(i)$ = optimal solution that excludes vertex m_j
$f_1^{m_j}(i,a)$ = optimal solution that includes vertex m_j.

CONNECT(Q_j, Q_{j-1})

$$f_0^{m_j}(i) = \begin{cases} i(i-1)/2, & \text{if } \mathrm{i} < \mathrm{c_j} + \mathrm{q_{j-1}} + \mathrm{q_j} \\ \max\{f^{m_j-1}(i), \max\{I_0^{m_j}(a) + f_1^{m_j-1}(c+d,c) + b(b-1)/2 + a(b+c) \\ \quad +bc : a+b+c+d = i, \\ \quad 0 \le a < q_j, 0 \le b \le c_j, 1 \le c \le q_{j-1}\}\}, & \text{otherwise} \end{cases}$$

$$f_1^{m_j}(i,a) = \begin{cases} i(i-1)/2, & \text{if } \mathrm{i} \le \mathrm{c_j} + \mathrm{q_{j-1}} + \alpha \\ \max\{I_1^{m_j}(a) + f_1^{m_j-1}(c+d,c) + b(b-1)/2 + a(b+c) + bc : \\ \quad a+b+c+d = i, \\ \quad 1 \le a \le q_j, 0 \le b \le c_j, 1 \le c \le q_{j-1}\}, & \text{otherwise} \end{cases}$$

$0 \le d \le \sum_{r=1}^{j-1} c_r + \sum_{l=1}^{j-2} q_l$.

Initialization

I. For every $m_j \in Q_j$ we compute $I_0^{m_j}(i)$ and $I_1^{m_j}(i)$, for $0 \leq i \leq k$.

$$I_0^{m_j}(i) = \begin{cases} i(i-1)/2, & \text{if i < q}_\text{j} \\ -\infty, & \text{otherwise} \end{cases}$$

$$I_1^{m_j}(i) = \begin{cases} i(i-1)/2, & \text{if i} \leq \text{q}_\text{j} \\ -\infty, & \text{otherwise} \end{cases}$$

Initialization

II. For $j = 1$ we compute $f_0^{m_1}(i)$, $1 \leq i \leq k$, and $f_1^{m_1}(i,a)$, $1 \leq i \leq k$, $1 \leq a \leq q_1$.

$$f_0^{m_1}(i) = \begin{cases} i(i-1)/2, & \text{if i} < \text{c}_1 + \text{q}_1 \\ -\infty, & \text{otherwise} \end{cases}$$

$$f_1^{m_1}(i,a) = \begin{cases} i(i-1)/2, & \text{if i} \leq \text{c}_1 + \alpha \\ -\infty, & \text{otherwise} \end{cases}$$

Theorem
There is an $O(nk^4)$ algorithm for the DkS on proper interval graphs having a simple path as clique graph.

Optimal solution: $f^{m_z}(k) = \max\{f_0^{m_z}(k), f_1^{m_z}(k,1)\}$.

Complexity: $O(nk^4)$

Conclusions

- An $O(kn^2)$ algorithm for the HkS problem on chains, (the optimal can be disconnected).
- An $O(kn^3)$ algorithm for the HkS problem on cycles, (the optimal can be disconnected).
- An $O(kn^3)$ algorithm for the HkS problem on graphs of maximal degree 2, (the optimal can be disconnected).
- An $O(nk^4)$ algorithm for the DkS problem on a subclass of proper interval graphs.

Open Questions

- HkS on trees, (disconnected solution)?
- DkS on Intervals, Chordals, ...?
- Is there an $\epsilon > 0$ such that achieving an approximation ratio of $O(n^{\epsilon})$ for the DkS is NP-hard?
- Is there a better than $O(n^{1/3})$ approximation ratio for the DkS problem?