Problems related to analysis of some models of distributed computations and social networks

Nikolay Kuzyurin

1. Grid, Computatinal Clusters

scheduling parallel tasks on a group of clusters

Model: Multiple Strip Packing.

on-line algorithms

1. Grid, Computatinal Clusters

scheduling parallel tasks on a group of clusters

Model: Multiple Strip Packing.

on-line algorithms

2. Probabilisric graph models of social networks.

Some optimization problems

Strip packing problem

Input:

- $I = (R_1, \ldots, R_N)$ — list of rectangles
- i-th rectangle:
  - $h(R_i)$ — height,
  - $w(R_i)$ — width

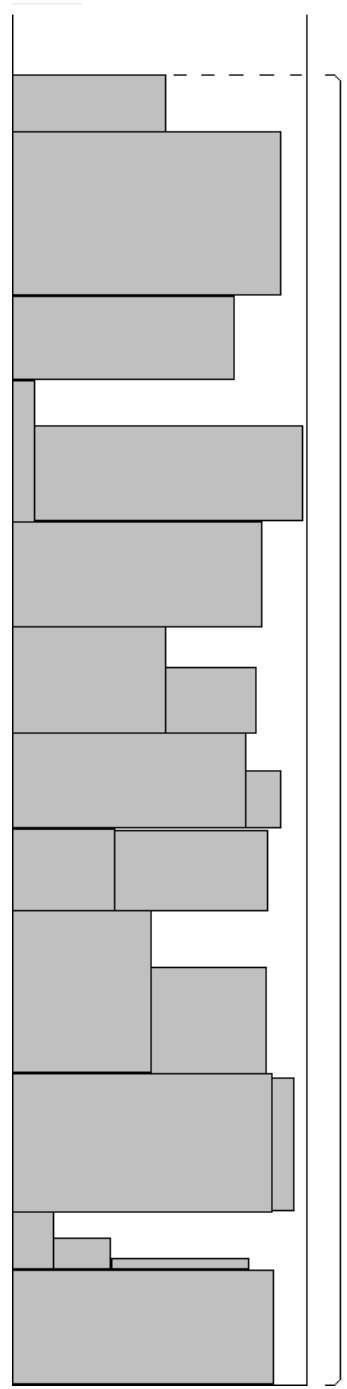
Objective: Find orthogonal packing of I inside a unit width strip without rotations and intersections so that the height of packing is minimal.

Applications

- VLSI design
- Cutting stock problem
- Scheduling of parallel jobs on a cluster

Packing example

N = 20



H - height of packing

Strip packing: approximation algorithms

Strip packing is NP-hard (1980)

$\Rightarrow$ Approximation algorithms

Approximation ratio

$$R_A = \sup_I \left\{ \frac{A(I)}{OPT(I)} \right\}$$

Asymptotic approximation ratio

$$R_A^\infty = \lim_{k \to \infty} \sup_I \left\{ \frac{A(I)}{OPT(I)} \mid OPT(I) \geq k \right\}$$

Strip packing: on-line algorithms. Worst case analysis

On-line algorithms with **asymptotic** approximation ratios

1983 Baker, Schwarz, Shelf algorithms, $R_A^\infty \leq 1.7 + \varepsilon$

1997 Csirik, Woeginger $R_A^\infty \leq 1.69103$

2007 Han, Iwama, Ye, Zhang $R_A^\infty \leq 1.58889$

Lower bound

- van Vliet $R_A^\infty \geq 1.54$

Average case analysis of algorithms

Standard probabilistic model: $h(R_i)$, $w(R_i)$ are independent random variables uniformly distributed in $[0, 1]$

Denote uncovered area of a strip as

$$S = H - \sum_i h(R_i) w(R_i)$$

The goal is to minimize $\mathbf{E}\, S$

Best known results in terms of average-case analysis

1993 $\mathbf{E}\,S = O(N^{1/2})$ — **Off-line algorithm**, Coffman, Shor.

1993 $\mathbf{E}\,S = O(N^{2/3})$ — **Closed-end on-line algorithm** (the number of rectangles N is known in advance), Coffman, Shor.

2010 $\mathbf{E}\,S = O(N^{2/3})$ — **Open-end on-line** (an algorithm does not know the number of rectangles), Kuzyurin, Pospelov.

New algorithm for closed-end SP

M. Trushnikov[1] proposed new **on-line** algorithm for **closed-end** strip packing.

Experimentally he showed that

- $\mathbf{E}\,S = CN^{1/2}$

N	C
80 000	1.5655
150 000	1.5716
500 000	1.5798
1 000 000	1.5798
4 000 000	1.5878
15 000 000	1.5975
30 000 000	1.5897
100 000 000	1.5934
300 000 000	1.6006
800 000 000	1.5912
1 000 000 000	1.6044
1 500 000 000	1.6027
2 000 000 000	1.5949

[1]Proceedings of ISP RAS, 2012, v. 22

The idea of new algorithm (Trushnikov)

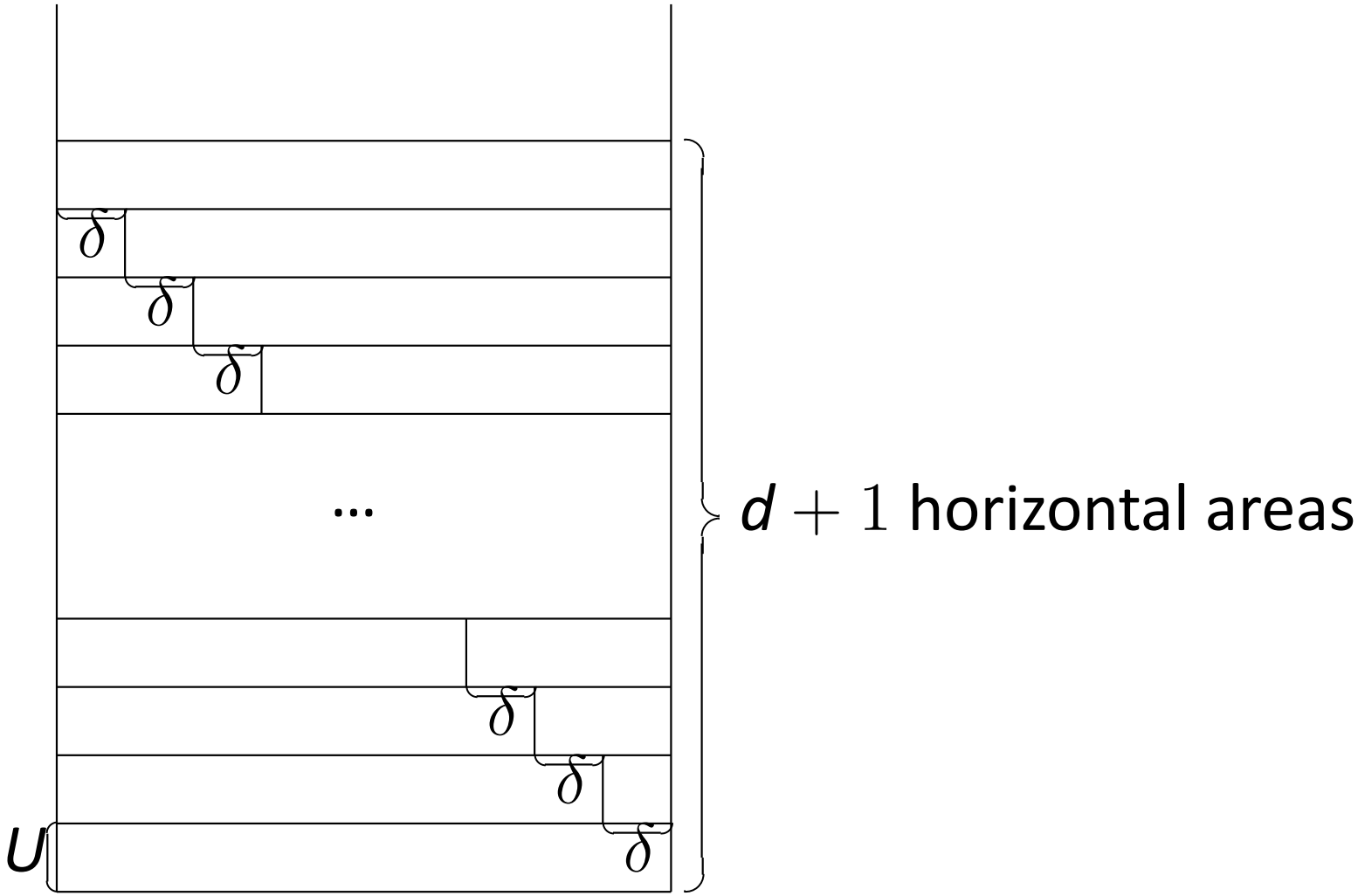
Notations

$$d = \left\lfloor \frac{N/4}{\sqrt{N}} \right\rfloor, \quad \delta = \frac{1}{d}$$

$$U = \frac{N/4}{d} = \sqrt{N} + O(1).$$

At the bottom of the strip we introduce $d + 1$ horizontal areas (called containers) each of height U (see the picture below).

Algorithm



Algorithm

Each even rectangle we will pack in the first pyramid and each odd one in the second.

Rectangles which constitute the pyramid we will call **containers**.

Enumerate containers inside the pyramid by numbers from 1 up to d such that the i th one has width $i\delta$.

Rectangles inside containers will be packed one by one: the first at the bottom, next one above the first and so on.

The steps of the Algorithm

Let we obtain as input current rectangle of width w.

- Find i, such that $(i-1)\delta < w \leq i\delta$. We will call this rectangle be *assigned* to the i th container.
- Then find minimal j such that $i \leq j \leq d$ and in the j th container it is enough room to pack the rectangle.
- If such j esists we pack the rectangle into the j th container.
- If no, then put the rectangle above current packing.

Such rectangles we will call **unpacked**.

Theorem (Trushnikov)

Theorem. The expected wasted area of packing obtained by the Algorithm is

$$\mathbf{E}\, S = \tilde{O}(\sqrt{N}) = O\left(N^{1/2}(\log N)^{3/2}\right)$$

Outline of the proof

Let Σ is the square of all N rectangles. Obviously $\mathbb{E}\Sigma = N/4$.

The height of the pyramids is

$$(d+1)U = N/4\left(\frac{d+1}{d}\right) = N/4 + \frac{N}{4\lfloor\frac{N/4}{\sqrt{N}}\rfloor} = N/4 + O(N^{1/2}).$$

We will consider only one of the two pyramids and only $\lfloor N/2 \rfloor$ rectangles packed into this pyramid.
Let us enumerate these $\lfloor N/2 \rfloor$ rectangles by numbers from 1 up to $\lfloor N/2 \rfloor$ in the order of arriving rectangles.

Let $\mathbb{M}\{n_1, n_2\}$ be the expectation of the number of **unpacked** rectangles when the Algorithm packs rectangles with numbers from the interval $[n_1, n_2]$

It is sufficient to prove that $\mathbb{M}\{1, \lfloor N/2 \rfloor\} = O(N^{1/2}(\log N)^{3/2})$.

Main results

Define two numbers k_0 and k_1:

$$k_0 = \lfloor N/2 \rfloor - \lfloor N^{3/4}\sqrt{\log N} \rfloor, \quad k_1 = \lfloor N/2 \rfloor - \lfloor N^{1/2} \rfloor.$$

Obviously

$$\mathbb{M}\{1, \lfloor N/2 \rfloor\} = \mathbb{M}\{1, k_0\} + \mathbb{M}\{k_0 + 1, k_1\} +$$

$$\mathbb{M}\{k_1 + 1, \lfloor N/2 \rfloor\}$$

Main results

Lemma 1. $\mathbb{M}\{k_1 + 1, \lfloor N/2 \rfloor\} = O(N^{1/2}).$

Lemma 2. $\mathbb{M}\{1, k_0\} \to 0, \quad N \to \infty,$

Lemma 3. $\mathbb{M}\{k_0 + 1, k_1\} = O(N^{1/2}(\log N)^{3/2})$

Open questions

Process. The are n enumerated urns, each can contain at most n balls and there are n^2 balls.

At the beginning all urns are empty.

At the current step the current ball goes to any urn with probability n^{-1}.

Process. If the urn is not full (contains less than n balls), the ball will be packed into this urn.

In opposite case it moves to the urn with number less by 1. If it is not full the ball will be packed into this urn, else it moves to the next urn with number less by 1.

Problem

If the ball was moved to the urn with number 1 and the urn is full, the ball is **unpacked**.

Question: *Is it true that the expectation of the unpacked balls is* $O(n)$?

Generalized multiple-strip packing

- MSP: Multiple strip packing problem
- there are **M** strips of unit width instead of one.
- Generalized MSP (Initially addressed by Zhuk, 2006): There are **M** strips of widths $w_1, \ldots, w_M$.

$$w_1 \geq w_2 \geq \ldots \geq w_M$$

Generalized multiple-strip packing

There are examples of inputs for Generalized MSP such that very natural heuristics give

$$R_A^{\infty} \to \infty$$

Generalized multiple-strip packing

Zhuk proved (2007) for generalized MSP that

- there is an on-line algorithm A

$$R_A^\infty \leq 2e$$

- For any on-line algorithm A:

$$R_A^\infty \geq e$$

Notations.

Define $A(T)$ as a vector $y = (y_1, \ldots, y_m)$, where y_k is the sum of squares of rectangles from T packed by algorithm A into the k th strip.

$h(T)$ efficiently computable function

$h(T)$ is the lower bound of the height of optimal packing

$$OPT(T) \geq h(T)$$

An idea of balancing. Concrete rule: Let a set of rectangles T was packed and $A_r(T) = (y_1, \ldots, y_m)$. Next rectangle R will be packed as follows:

1. Compute $h = h(T + \{R\})$.
2. Find k, such that

$$k = \max i : w(R) \leq w_i \text{ and } \frac{y_i}{w_i} \leq eh.$$

If such k exists we pack R into the k th strip.

Directions for future work

Special cases: all strips have equal widths (MSP)

strips have widths of special form (say, powers of 2)

strips have constant number of different widths

MSP: on-line vs off-line

Off-line

- AFPTAS, 2009, Bougeret, Dutot, Jansen, Otte, Trystam
- $R_A \leq 2$ 2009, Bougeret, Dutot, Jansen, Otte, Trystam

MSP: on-line vs off-line

Off-line

- AFPTAS, 2009, Bougeret, Dutot, Jansen, Otte, Trystam
- $R_A \leq 2$ 2009, Bougeret, Dutot, Jansen, Otte, Trystam

On-line

- $R_A \leq 3 + \delta_m$, Ye, Han, Zhang, 2009
- $R_A \leq 2.7 + \delta_m$, Ye, Han, Zhang, 2009 randomized on-line algorithm

$$\delta_m \to 0,\ m \to \infty$$

Multiple Strip Packing: average case

MSP – all strips have equal widths

Our results on average case analysis for MSP

Multiple Strip Packing: average case

MSP – all strips have equal widths

Our results on average case analysis for MSP

Modified T-algorithm: every new rectangle we place on the emptiest strip and then use Trushnikov's algorithm.

Theorem

$\mathbf{E}\, S_{max} = \tilde{O}(N^{1/2})$ for $M = const$.

Experiments show that $\mathbf{E}\, S_{max} = O(N^{1/2})$ even for $M = N^{1/3}$

$$\mathbf{E}\, S_{max} = CN^{1/2}$$

M	N	C
21	10 000	1.663
34	40 000	1.6415
54	160 000	1.6937
86	640 000	1.7065
136	2 560 000	1.7238
273	20 480 000	1.5822
434	81 920 000	1.6312
547	163 840 000	1.7506
689	327 680 000	1.7396
868	655 360 000	1.6455
1000	1 000 000 000	1.5631

Experiments (average case) for MSP

For $M = N^{1/2}$ average waste grows faster than $N^{1/2}$

M	N	C
200	40 000	3.0043
400	160 000	3.7113
800	640 000	4.8146
1131	1 280 000	5.1267
1600	2 560 000	4.7967
2262	5 120 000	3.9807
3200	10 240 000	5.321
4525	20 480 000	5.4551
6400	40 960 000	7.5701
9050	81 920 000	8.067
12800	163 840 000	9.3379
18101	327 680 000	7.6747
31623	1 000 000 000	16.4354

Resume and future work

- New closed-end on-line algorithm for strip packing
- It is shown experimentally that $\mathbf{E}\,S = O(N^{1/2})$.
- It is proved that the algorithm provides $\mathbf{E}\,S = \tilde{O}(N^{1/2})$

Resume and future work

- New closed-end on-line algorithm for strip packing
- It is shown experimentally that $\mathbf{E}\,S = O(N^{1/2})$.
- It is proved that the algorithm provides $\mathbf{E}\,S = \tilde{O}(N^{1/2})$

Future work: improve analysis of new algorithm ($\mathbf{E}\,S = O(N^{1/2})$) and adapt it to MSP.

Social networks

Facebook, Twitter, VKontakte, etc.

Social networks

Facebook, Twitter, VKontakte, etc.

J. Ugander, B. Karrer, L. Bachstrom, C. Marlow,
The anatomy of the Facebook social graph
Conell Univ. Library, arXiv.org>cs>arXiv:
1111.4503

Social networks models

Social networks are sparce random graphs, rapidly growing and rapidly changing

Social networks models

Social networks are sparce random graphs, rapidly growing and rapidly changing

Classical Erdos-Renyi model $G_{n,p}$ (1959)

Erdos-Renyi model

Random graph $G_{n,p}$

- n nodes of a graph
- p probability that edge (v, u) exists $\forall u, v$

Evolution: p is often a function of n

'evolution' of random graphs: the study for what functions $p = p(n)$ the graph change its properties.

If $p = \frac{c}{n}$ then component structure depends on the value of c:

If $c < 1 \Rightarrow$ all components have size $O(\log n)$

If $c \geq 1$ there is one giant component of size $\theta(n)$, and all other componets have size $O(\log n)$.

Erdos and Renyi model is inappropriate for real-life

- do not satisfy power law: the number of nodes of degree *i* is proportional to $i^{-\beta}$

Barabasi-Albert model (1999)

- starting with m_0 of vertices
- at every step:
  - add a vertex v_{new} with m edges
  - P(v_{new} connected v_i) depend on $degree(v_i)$
- After t steps we have $t + m_0$ vertices and mt edges.

Three types of models

1. Heirstical (preferential attachments, forest fire, kronecker graph products,)
2. Bollobas-Riordan - fixed parameter $\beta = 3$ and some generalizations ($2.1 \leq \beta \leq 3$)
3. Models with arbitrary β (Chung-Lu, Luczak-Janson)

New random models (generators)

- Random walk (2003)
- Nearest Neighbor (2003)
- Forest Fire (2005)
- Modifications (2010)
- KronFit (2007)
- DK-2 (2006)

$G(w)$ model Chung-Lu (2006): if average degree $d > 1$ then almost surely (a.s) G has a unique giant component,
the second largest component a.s. has size $O(\log n)$

$G(w)$ model Chung-Lu (2006): if average degree $d > 1$ then almost surely (a.s) G has a unique giant component,
the second largest component a.s. has size $O(\log n)$

Janson, Luczak, Norros (2009): if $0 < \beta < 3$ the largest clique a.s. has size $O(n^{c(\beta)})$,
if $\beta > 3$ then largest clique has size in $\{2, 3\}$

k-core: maximal induced subgraph with minimum degree k.

$G(w)$ model Chung-Lu

k-core: maximal induced subgraph with minimum degree k.

$G(w)$ model Chung-Lu

Fernholz-Ramachandran (2004): for every $k \geq 3$ a random G a.s has a giant k-core if $2 < \beta < 3$,

a.s. has no giant 3-core if $\beta > 3$

Two popular optimization problems in social networks:

influence maximization (IM)

finding communities

Information diffusion in social networks

- Social network as a directed graph $G = (V, E)$

Informally

Starting from few *seed* nodes information can stochastically propagate to new nodes.
Goal: find small subset of nodes that could maximize the spread of influence (influence maximization IM).

Analogies: Epidemies, physics, sociology, ecomics (viral marketing, word-of-mouth effect)

Information diffusion in social networks

2001 Domingos and Richardson: first study of IM as an algorithmic problem.

2003 Kempe, Kleinberg, Tardos — first results for stochastic cascade model:

- IM is NP-hard
- Greedy is (1-1/e)-approximate algorithm

Basic Diffusion Models

Three basic models (Kempe et al, 2003):

IC Independent cascade model

LT Linear threshold model

WC Weight cascade model

Generalizations

Independent cascade (IC) model

1. Start with initial set of nodes (*seed*)
2. Runs until no more activations are possible:
   - if node v becomes active at step t then
     - ★ at step $t+1$ (only!)
     - ★ it can activate each neighbor w of v
     - ★ independently with probability $p(v,w)$

Problem formulation

- Given
  - $G = (V, E)$,
  - number k
  - $p(u, v)$ for all edges $(u, v) \in E$
- Find k nodes maximizing expected number of nodes influenced by these k nodes.

Complexity

Kempe at al (2003):

For both models the problem is NP-hard (worst case).

Approximation algorithms

- Greedy: $(1 - 1/e)$-approximation for any input
- $(1 - \frac{1}{e} - \varepsilon)$-approximation is NP-hard for any $\varepsilon > 0$ (Chen et al, 2010)

Approximation algorithms: Greedy

- $I \subseteq V$ subset of nodes $G = (V, E)$
- $f(I)$ the expectation of the size of influenced nodes (this function is submodular)

Greedy

Set $I := \emptyset$

At each step choose v such that $f(I \cup v)$ is maximum and set $I := I \cup v$ until $|I| = k$

Difficulties with Greedy. Finding v such that $f(I \cup v)$ is maximum is #P-hard (2010)

Experiments

To find next node with maximum expectation of additional influence it is necessary to do about 10000 random iterations (smaller values decrease the quality of solution).

As a consequence the classical greedy algorithm is too slow for relatively large networks (hundreds thousands of nodes or more)

Fast heuristics:

- Random
- Single discount
- Degree
- Dedgree discount
- Centrality

General observations

1. The basic Greedy heuristics is the best among known algorithms with respect to quality (experimental results) (Kempe at al., 2003, Chen et al., 2009).

But! Such algorithms can not be used for large enough networks

General observations

1. The basic Greedy heuristics is the best among known algorithms with respect to quality (experimental results) (Kempe at al., 2003, Chen et al., 2009).
But! Such algorithms can not be used for large enough networks

2. Different fast heuristics (say, Random, Single discount, Degree, Dedgree discount) are fast enough for large networks but cannot achieve the quality of solution obtained by Greedy

New problem formulation

- Given
  - $G = (V, E)$,
  - number k
  - $p(u, v)$ for all edges $(u, v) \in E$
  - subset $H \subseteq V$ of nodes
- Find k nodes maximizing expected number of nodes in H influenced by these k nodes.

Analogy: spanning tree vs Steiner tree

Communities: definitions

Graph $G = (V, E)$

- **k-clique** — complete subgraph on k nodes
- **a-near-k-clique** (or a-dense k-subgraph) — subgraph S on k nodes with $2|E(S)|/(k(k-1) \geq a$

Communities: definitions

Graph $G = (V, E)$

- **k-clique** — complete subgraph on k nodes
- **a-near-k-clique** (or a-dense k-subgraph) — subgraph S on k nodes with $2|E(S)|/(k(k-1) \geq a$
- **subset of nodes** S: $|E(S)| > |E(S, V \setminus S)|$
- **subset of nodes** S: $|E(S, V \setminus S)|/|E(S)| \leq a$

Computational hardness

The most formulations of communities problems (maximum clique, densest subgraph of given size, etc.) are NP-hard.

Computational hardness

Maximum Clique is hard to approximate within $|V|^{1-\delta}$ (Hastad, 1996)

This problem is difficult even in random graphs ($G_{n,p}$ Erdos-Renyi model) — (Karp, 1976)

Finding large hidden clique in $G_{n,p}$ is hard (Alon, Krivelevich, Sudakov, 1994)

(k, γ)-community

Subset $S \subseteq V$, $|V| = k$ is a (k, γ)-community if

$$|E(S, V \setminus S)|/|E(S)| \leq \gamma$$

Given $G = (V, E)$ and k find $S \subseteq V$ of size k minimizing γ such that

$$|E(S, V \setminus S)|/|E(S)| \leq \gamma$$

Given $G = (V, E)$ and k find $S \subseteq V$ of size k minimizing γ such that

$$|E(S, V \setminus S)|/|E(S)| \leq \gamma$$

The problem is NP-hard. Moreover it cannot be approximated under UGC (Uniques games conjecture): distingwish between $\gamma \leq \delta$ and $\gamma \geq 1 - \delta$

Raghavendra-Steurer, STOC-2010, Graph expansion and the Unique Games Conjecture

Problems

Properties of random graphs in power law models

Achieve quality of Greedy for IM by more efficient heuristics

Approximation algorithm for IM with objective set of nodes $H \subseteq V$

Algorithms for finding (k, γ)-communities in random power law graphs

nnkuz@ispras.ru