Implementation of the asymptotically optimal approach to polynomial time solving some hard discrete optimization problems

Edward GIMADI

Sobolev Institute of Mathematics, Novosibirsk, Russia
Novosibirsk State University, Russia

SUMMARY

An overview of research topics on constructing polynomial time asymptotically optimal (exact) algorithms for several hard optimization problems. Relevant research results are mostly obtained in different years (since 1969) by the author and his younger colleagues in the laboratory "Discrete Optimization in Operations Research" of the Institute of Mathematics SB RAS, Novosibirsk, Russia.

PRELIMINARIES

Notation

- $F_A(I)$ is the objective by algorithm A on input I;
- ε_n^A is the resulting relative error of A;
- $\delta_n^A \in (0,1)$ is the failure probability of A;
- $Pr\{\cdot\}$ is the probability of the corresponding event.

PRELIMINARIES

Algorithm with performance guarantees $(\varepsilon_n^A,\ \delta_n^A)$

We say that Algorithm A admits estimates $(\varepsilon_n^A,\ \delta_n^A)$ in the class of n-sized optimization problems if, for every size n and input I, we have

$$Pr\left\{\left|\frac{F_A(I) - OPT(I)}{OPT(I)}\right| > \varepsilon_n^A\right\} \leq \delta_n^A.$$

It is clear that

- The event within the brackets is not desirable.
- The algorithm is better than the smaller its estimates ε_n^A and δ_n^A.

PRELIMINARIES

Asymptotically optimal algorithm

The algorithm is called asymptotically optimal (exact) on the class of n-sized problems, if it admits a pair of estimates $(\varepsilon_n^A,\ \delta_n^A)$, s.t.

$$\varepsilon_n^A \to 0; \quad \delta_n^A \to 0 \ \text{ as } n \to \infty.$$

Thus, the asymptotically optimal algorithm becomes more exact and relieble with increasing the size problem.

This contrasts with the well-known concept of "curse of dimensionality" which appeared at the beginning of the second half of the previous century (Bellman R., Hughes G.F.).

Definitions of "With high probability" term

An event, which occurs with high probability (w.h.p.)

means that an event B_n in the sequence $\{B_n\}$ occurs with probability $\to 1$ as $n \to \infty$.

I.e.: algorithm solves a problem of size n **w.h.p.**, or with probability $1 - \delta_n$, where the failure probability $\delta_n \to 0$ as $n \to \infty$.

Asymptotically optimal approach

Examples of implementations:

- TSP and some its variations: Max TSP on Euclidean Space $\mathbb{R}^k$, Euclidean Max PSP, TSP and m-PSP on random distances bounded and unbounded from above;
- m-Cycles Covering Problem;
- Max (Min) total Weight Vector Subset Problem;
- MST Problem with a bounded below diameter;
- Degree Constrained Connected Subgraph Problem;
- Random Multi-index AP;
- Bin and Strip Packing Problem;
- Random p-median Problem;
- Random VRP with limited number of clients per rout;
- PS Problem with limited accumulative resources;
- CFLP on random input distances

Firs texemple of the algorithm with proven guarantees

A. Borovkov. On probabilistic target setting two economic problems // Reports of Acad. Sc. USSR. 1962. Vol. 146, N 5, P. 983-986.

TSP& AP

n points in bounded simply connected domain Ω in Euclidean space $\mathbf{R^k}$ are considered. Distribution function of points inside Ω are distributed with continuous everywhere non-zero density function.

Borovkov's Algorithm:

- Cut k-dimensional region Ω into a system of cylindrical stripes.

(Each of these stripes gives $(k-1)$-dimensional cube in section $x_1 =$const)

- Connect points of every stripe by broken line such that along this line quantity x_1 are changing monotonously.

Statement [Borovkov, 1962].

For uniform distributed points in unit square Algorithm finds Hamiltonian circuit in time $O(n \log n)$ with estimates

$$\varepsilon_n = 0.48,$$

$$\delta_n \to 0 \text{ as } n \to \infty.$$

On asymptotical optimality of some algorithms for solving TSP and its generalizations.

The classical TSP is to find the minimum length route through n cities.
A good survey on TSP can be found in

"TSP and its Variations"

[(2002) Gutin G., Punnen A. P. (eds.), Kluver Academic Publishers, Dordrecht].

"Nearest City" Algorithm (NCA)

A number of results concerning NCA is obtained under the assumption
independent random variables. In the case of discrete distr. f.

$$p_k = Pr\{c_{ij} = k\},\ k = 1, \dots, r_n,$$

NCA is asymptotically optimal if

$$\sum_{k=1}^{r_n} \frac{1}{\sum_{i=1}^{k} p_i} = o(n).$$

[Gimadi and Perepelitsa, 1969]

$c_{ij} \in [a_n, b_n], \quad a_n > 0 \; a_n \le c_{ij} \le b_n, \; a_n > 0$, NCA is asymptotically optimal if

$$\boxed{\begin{gathered} b_n/a_n = o\left(\frac{n}{\max\{n\gamma_n, J_n\}}\right), \\ J_n = \int_{\gamma_n}^{1} \frac{dx}{P(x)} \to \infty, \end{gathered}}$$

where $P(x) = Pr\left\{\frac{c_{ij}-a_n}{b_n-a_n} < x\right\}, \quad P(\gamma_n) = 1/n.$

[Gimadi & Perepelitsa, 1974]

Probabilistic inequalities

Results mentioned above were established using the classical Chebyshev's probability inequality.

Better approximation guarantees can be obtained using the techniques initiated by [Chernov, 52] and generalized by [Hoeffding, 63].

Also good bounds are investigated using like-wise Bernstein probability inequalities [Petrov 74].

Petrov's Theorem.

Consider i.r.v. $X_1, \dots, X_n$.
Let there be constants T and $g_1, \dots, g_n > 0$ s.t. for all $1 \le k \le n,\ t \in [0, T]$

$$\mathbf{E}e^{tX_k} \le \exp\Big\{\frac{g_k t^2}{2}\Big\}.$$

Put $S = \sum_{k=1}^{n} X_k$ and $G = \sum_{k=1}^{n} g_k$. Then

$$\mathbf{Pr}\{S > y\} \le \begin{cases} \exp\{-\frac{y^2}{2G}\}, & 0 \le y \le GT, \\ \exp\{-\frac{Ty}{2}\}, & y \ge GT. \end{cases}$$

By Petrov's Theorem performance bounds for NCA

$$\varepsilon_n = O\Big(\frac{b_n/a_n}{n/\ln n}\Big), \quad \delta_n = O\left(\frac{1}{n}\right).$$

Then the condition of asymptotic optimality is

$$\frac{b_n}{a_n} = o\Big(\frac{n}{\ln n}\Big) \; .$$

Euclidean TSP

TSP is called Euclidean (**ETSP**), if

- vertexes in graph correspond to points in Eucliden space $\mathbf{R^k}$, and
- edge weights equal to lenghts of relative intervals.
- ETSP$_{max}$ in space $\mathbf{R^k}$ is NP-hard when $k \geq 3$.
- For $k = 2$ hardness status of ETSP$_{max}$ is open.

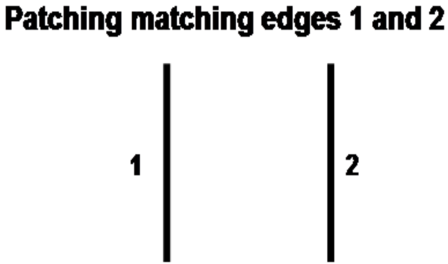
[Fekete&Barvinok]

Asymptotical optimal algorithms for ETSP$_{max}$

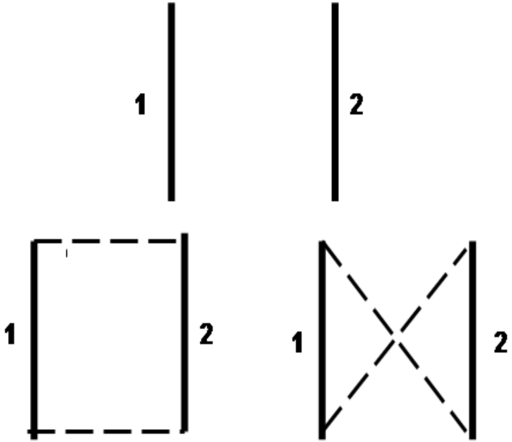
Serdyukov 1987;
Gimadi 2001;
Baburin&Gimadi 2002.

Time complexity is determined by a procedure of searching maximum weight matching in given graph

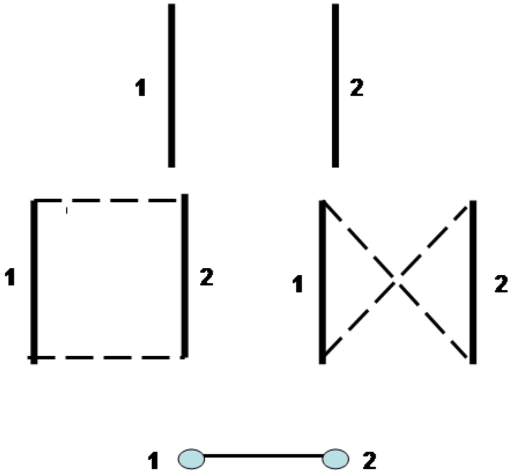
Patching intervals



Patching matching edges 1 and 2



Patching matching edges 1 and 2



Max angle between two intervals [1987]

Let $\alpha \leq \pi/2$ be an angle between two intervals $A = (a_1, a_2),\ B = (b_1, b_2)$ from $\mathbf{R}^{\mathbf{k}}$. Then

$$(|A| + |B|) \geq$$

$$\geq \max\begin{cases} |a_1, b_1| + |a_2, b_2|, \\ |a_1, b_2| + |a_2, b_1| \end{cases}$$

$$\geq (|A| + |B|) \cos\frac{\alpha}{2}$$

Min solid angle between t intervals [1987].

Let $\alpha_k(t)$ be a min solid angle between t intervals in $\mathbf{R^k}$. Then

$$\sin\frac{\alpha_k(t)}{2} \le \frac{\gamma_k}{t^{\frac{2}{k-1}}},$$

where γ_k depends on dimension of set $\mathbf{R^k}$.
So

$$\alpha_k(t) \to 0$$

as $t \to \infty$.

Degree constrained connected subgraph problem

Theorem [Baburin and Gimadi, 2005]

.

Let $d = d_{min}$ be minimal vertex degree in subgraph chosen, M be the number of edges. Then DCCSP_{max} can be solved in time $O(Mn^2)$ with relative error $\varepsilon \leq \frac{2}{d^2+d}$.
In the metric case the error is two times less.

The condition of asymptotical optimality for DCCSP_{max} is $\quad d = \phi_n \to \infty.$

d-regular DCCSP

Generalization of TSP, when $d = 2$.
For d-regular d-regular DCCSP$_{max}$ the conditions of asymptotic optimality were established in the case when edge weights are i. r. v. from $[a_n, b_n]$ with i.u.d.f. of minorized type.

uniform.pdf

• $TSPmax_i$ is solved for each distance matrix $D_i,\ 1 \leq i < d$, using FC-heuristic.
• For every matrix $H_i,\ 1 \leq i \leq d-2$, we pick out i elements in each row and one element in each column by greedy heuristic.
• The elements symmetric to the ones that are chosen above are also chosen.
So in each line of the matrix (w_{ij}) we have exactly d chosen elements that correspond to a d-regular subgraph in given graph.

$$\varepsilon_n \le \frac{3\ln(m+1)+1}{m}, \quad \delta_n = \frac{1}{m+1},$$

where $m = n/(d-1)$.

So the condition of asymptotic optimality is

$$d = o(n).$$

[Baburin&Gimadi, 2005]

Now analogous results were established also without the requirement of integer representation of m and for d-regular DCCSP$_{min}$.

Euclidean problem of finding a maximum total weight subset of vectors

P1: Consider the finite family of vectors $V = \{v_1, \dots, v_n\}$ in Euclidean space R^k and positive integer $m < n$. Find a subset of vectors X of cardinality m such that the norm of the sum of the vectors from X is maximum.

P2: Given the finite family $V = \{v_1, \dots, v_n\}$ of vectors in Euclidean space R^k and positive integers l, m such that $lm < n$ find a subset of vectors $X = \{v_{a_1}, \dots, v_{a_m}\}$, such that the norm of the sum of the vectors from X is maximum under the restriction $a_{i+1} - a_i \geq l$ for $i = 1, 2, \dots, m-1$.

The problems are NP-hard.

The problem P2 is connected in a signal search in a sequence of impulses with additional noise problem. This problem has applications in electronic reckoning, radiolocation, telecommunication, geophysics, medical and technical diagnostics, etc.

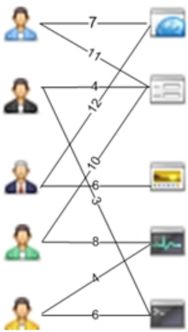
On latest results

1. The m-layered 3-index planar assignment problem on single cycle permutations **or**
   the m-Peripatetic Salesman Problem with different weight functions (m-PSP-DW)
2. The m-Peripatetic Salesman Problem with identical weight functions (m-PSP)
3. The m-Cycles Cover Problem (m-CCP)
4. The m-index axial assignment problem on single-cycle permutations (m-AAPC)

The classic Assignment Problem

Given $n \times n$ cost matrix $C = (c_{ij})$.

	1	2	3	4	5
1	7	11	$+\infty$	$+\infty$	$+\infty$
2	$+\infty$	4	$+\infty$	$+\infty$	3
3	12	$+\infty$	6	$+\infty$	$+\infty$
4	$+\infty$	10	$+\infty$	8	$+\infty$
5	$+\infty$	$+\infty$	$+\infty$	4	6



Find a permutation $\pi \in S_n$, such that:

$$\sum_{i=1}^{n} c_{i\pi(i)} \to \min_{\pi \in S_n}$$

Multi-index Assignment Problem (MAP)

To minimize the sum of the chosen elements.

NP-hard for the number of indexes at least three in axial and planar cases both [Karp,Frieze].

Axial MAP

In the case of the axial MAP n elements must be selected in the multi-dimensional matrix such that in every "cross-section" exactly one element is chosen. (The *"cross-section"* is such set of matrix elements when one index is fixed).

Conditions of asymptotic optimality established are likewise for TSP .

Selection of n^2 elements in a cubic matrix (c_{ijk}). Exactly one element in each line is chosen. (*A line* is the set of n elements with two fixed indexes).

Conditions of asymptotic optimality were established when the number of layers of the matrix (c_{ijk}) is at most $O(n^\theta)$, $0 < \theta < 1$

Motivation of the research

Complexity

The considered problems:

- Multi-index assignment problems on single-cycle permutations
- The m-Peripatetic Salesman Problem
- The m-Cycles Cover Problem

are **strong NP-hard.**

Motivation: numerous practical applications

Multi-index axial assignment problems

- Data association problem in multitarget tracking.
- Problems of classification and pairing of human chromosomes.

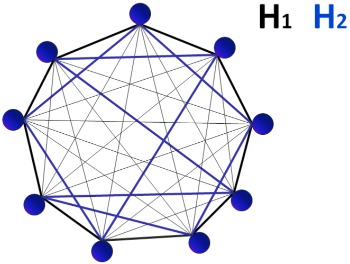
Multi-index planar assignment problems

- Related to the Latin squares, which has applications in combinatorics, statistics, cryptography, quasigroups studies in algebra.
- Scheduling problems
- Conflict-free access to parallel memories.
- Design of error-correcting codes.

1. The m-Peripatetic Salesman Problem

Given a complete graph $G = (V, E)$ with weight functions of edges $w_i : E \to \mathbb{R}_+$, $1 \le i \le m$.
The problem is to find m edge-disjoint Hamiltonian cycles $H_1, \ldots, H_m \subset E$:

$$W(H_1, \ldots, H_m) = \sum_{i=1}^{m} \sum_{e \in H_i} w_i(e) \to \min(\max).$$



- The weight functions w_i can be **different** – m-PSP-DW
- Or **identical** $w_1 = \ldots = w_n$ – classic m-PSP

The problem was first described in [Krarup 1974].

Baburin, Gimadi, 2010

An $O(n^3)$ algorithm for the m-PSP$_{max}$ in k-dimensional Euclidean space with relative error

$$\frac{m}{n}^{2/(k+1)}.$$

The algorithm is asymptotically optimal for $m = o(n)$.

Stage $i = 1, \ldots, m < n/4$**.**

Consider graph G with weight function of edges w_i and build Hamiltonian cycle H_i :

Step i0 Randomly select the first edge for H_i.

Step i1 Starting from the endpoint of the first edge build a partial path, according to the principal "go to the nearest unvisited vertex" $n - 4i$ times.

Step i2 Convert the path into a Hamiltonian cycle H_i using an extension-rotation procedure P_H.

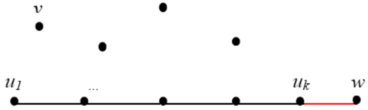
Delete all edges that belong to H_i from G, so these edges won't be used in Hamiltonian cycles $H_{i+1}, \ldots, H_m$.

1. Extension-rotation procedure P_H for the m-PSP

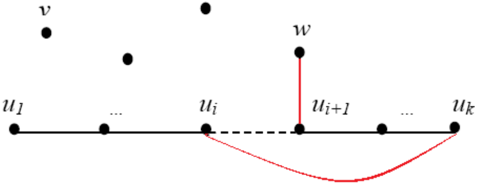
Given graph (V_H, E_H) with vertex degree $> |V_H|/2$, find a Hamiltonian path P with given endpoints u and v in $O(|V_H|^2)$ running-time.

- Let $P = \{u = u_1, \ldots, u_k\}$ be the constructed path

Extension: If possible, add an edge $\{u_k, w\}$, $w \notin P \cup \{v\}$ to the path.



Rotation: Else, for an arbitrary $w \notin P$, add edges $\{u_k, u_i\}$ and $\{u_{i+1}, w\}$ into P, and delete the edge $\{u_i, u_{i+1}\}$



- We assume that the weights of the edges $w_i(e)$ are i.i.d. random reals.
- The weight of an edge $w_i(e_s^i)$ chosen at Step i1 is estimated from above as minimum of $n-2i-s+2$ elements of random input.
- The weights of edges chosen at Step i2 has the same distribution function as the elements of random input.
- All random variables $w_i(e_s^i)$ are **independent**.
- The key element when estimating

$$Pr\Big\{\sum_{i=1}^{m}\sum_{s=1}^{n} w_i(e_s^i) > (1+\varepsilon_{\widetilde{A}})OPT\Big\} \leq \delta_{\widetilde{A}}$$

is the Petrov's theorem.

Definition

The distribution function $\mathcal{F}'(x)$ is a function of $\mathcal{F}$-majorizing type if

$$\mathcal{F}'(x) \geq \mathcal{F}(x) \quad \text{for every x}$$

Definition

The distribution function $\mathcal{F}'(x)$ is a function of $\mathcal{F}$-majorizing type if

$$\mathcal{F}'(x) \geq \mathcal{F}(x) \quad \text{for every x}$$

Distribution functions considered

We have considered the random inputs for the m-PSP-DW with distribution functions:

Definition

The distribution function $\mathcal{F}'(x)$ is a function of $\mathcal{F}$-majorizing type if

$$\mathcal{F}'(x) \geq \mathcal{F}(x) \quad \text{for every x}$$

Distribution functions considered

We have considered the random inputs for the m-PSP-DW with distribution functions:

- of UNI$[a_n, b_n]$-majorizing type,
  where UNI$[a_n, b_n]$ is uniform distribution in the interval $[a_n, b_n]$, $0 < a_n < b_n$;

Definition

The distribution function $\mathcal{F}'(x)$ is a function of $\mathcal{F}$-majorizing type if

$$\mathcal{F}'(x) \geq \mathcal{F}(x) \quad \text{for every x}$$

Distribution functions considered

We have considered the random inputs for the m-PSP-DW with distribution functions:

- of UNI$[a_n, b_n]$-majorizing type,
  where UNI$[a_n, b_n]$ is uniform distribution in the interval $[a_n, b_n]$, $0 < a_n < b_n$;
- of $Exp(x)$-majorizing type,
  where $Exp(x)$ is exponential distribution with parameters β_n, a_n:
  $$Exp(x) = 1 - \exp\left(\frac{x - a_n}{\beta_n}\right), \; x \geq a_n > 0.$$

Theorem

If the weights of edges are i.i.d. random reals with distribution UNI$[a_n, b_n], 0 < a_n < b_n$, then the $O(mn^2)$ algorithm $\widetilde{A}$ is asymptotically optimal with the following performance guarantees:

$$\varepsilon_n = O\left(\frac{b_n/a_n}{n/lnn}\right),\ \delta_n = n^{-9}, \quad \textit{for} \quad 2 \le m \le \ln n,\ \frac{b_n}{a_n} = o\left(\frac{n}{\ln n}\right).$$

$$\varepsilon_n = O\left(\frac{b_n/a_n}{n^\theta}\right), \delta_n = n^{-9}, \quad \textit{for} \quad \begin{array}{c} \ln n < m \le n^{1-\theta},\ \frac{b_n}{a_n} = o(n^\theta), \\ \theta \in (0, 1 - \frac{\ln\ln n}{\ln n}). \end{array}$$

Note

The obtained performance guarantees are true for any distribution function that dominates (majorates) UNI$[a_n, b_n]$.

For the maximum m-PSP-DW, use a similar algorithm $\widetilde{A}_1$, that choose the longest edge at Step i1.

Note

If the weights of edges are i.i.d. UNI$[a_n, b_n], 0 < a_n < b_n$ *random reals, then the* $O(mn^2)$ *algorithm* $\widetilde{A}_1$ *for the maximum m-PSP-DW is asymptotically optimal with the following performance guarantees:*

$$\varepsilon_n = O\left(\frac{lnn}{n}\right),\ \delta_n = n^{-9}, \quad \textit{for} \qquad 2 \le m \le \ln n.$$

$$\varepsilon_n = O(n^{-\theta}), \delta_n = n^{-9}, \quad \textit{for} \quad \ln n < m \le n^{1-\theta}, \theta \in (0, 1 - \frac{\ln\ln n}{\ln n}).$$

Theorem

If the weights of edges are i.i.d. random reals having distribution $Exp(x)$ *with parameters* β_n, $a_n > 0$, *then the* $O(mn^2)$ *algorithm* $\widetilde{A}$ *is asymptotically optimal with the following performance guarantees:*

$$\varepsilon_n = O\left(\frac{\beta_n/a_n}{n/lnn}\right), \ \delta_n = n^{-(3/4m+6/4)}, \quad \textit{for} \quad \begin{array}{c} 2 \le m \le \ln n, \\ \beta_n/a_n = o\left(\frac{n}{\ln n}\right); \end{array}$$

$$\varepsilon_n = O\left(\frac{\beta_n/a_n}{n^\theta}\right), \ \delta_n = n^{-(3/4m+6/4)}, \quad \textit{for} \quad \begin{array}{c} \ln n < m \le n^{1-\theta}, \\ \beta_n/a_n = o(n^\theta), \\ \theta \in (0, 1 - \frac{\ln\ln n}{\ln n}). \end{array}$$

Note

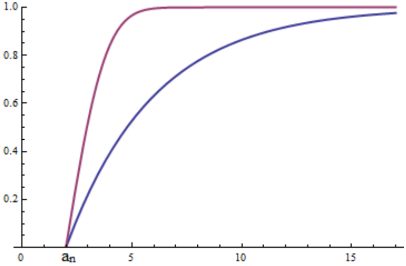
The obtained performance guarantees are true for any distribution function that dominates (majorates) Exp(x).

Note

The truncated normal distribution with $a_n, \sigma_n = \beta_n/2$*:*

$$\mathcal{N}_{a_n,\sigma_n^2}(x) = \frac{2}{\sqrt{2\pi\sigma_n^2}} \int_{a_n}^{x} \exp\left(-\frac{(t-a_n)^2}{2\sigma_n^2}\right) dt, \ \ , 0 < a_n \le x \,,$$

dominates the shifted exponential distribution with parameters a_n, β_n.



2. The m-PSP with identical weight functions

Given a complete graph $G = (V, E)$ with weight function of edges $w : E \to \mathbb{R}_+$, $1 \le i \le m$.
The problem is to find m edge-disjoint Hamiltonian cycles $H_1, \ldots, H_m \subset E$:

$$W(H_1, \ldots, H_m) = \sum_{i=1}^{m} \sum_{e \in H_i} w(e) \to min(max).$$

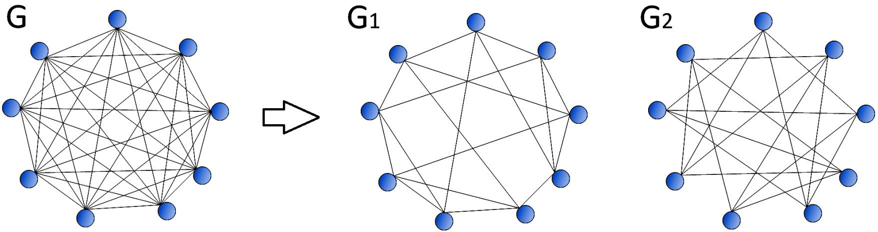
The analysis of algorithm $\widetilde{A}$ for m-PSP-DW strongly depends on

1. the Petrov's theorem
2. the independence of the weight functions $w_i, 1 \le i \le m$.

In the classic m-PSP the weight functions are **dependent:**

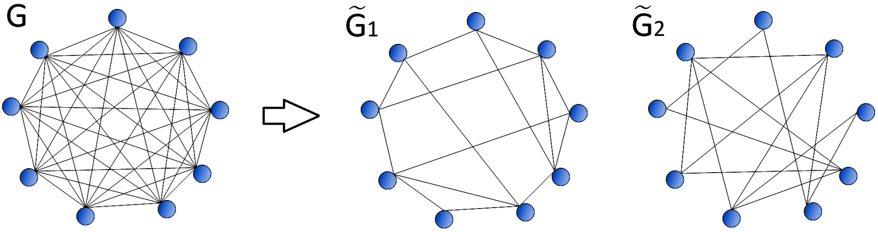
$$w_1 = w_2 = \ldots = w_m$$

Step 1 Uniformly split the initial complete n-vertex graph G into spanning subgraphs $G_1, \ldots G_m$:



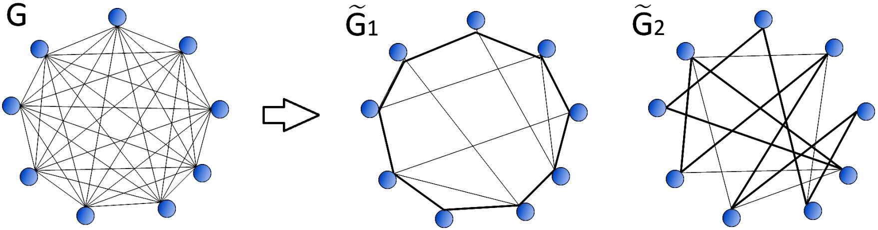
- G_i is a random graph where each edge exists with probability $1/m$

Step 2 Construct subgraphs $\widetilde{G}_1, \dots, \widetilde{G}_m$ deleting all edges in G_i, $1 \le i \le m$, which are heavier than w^*.



- $\widetilde{G}_i$ is a random graph where each edge exists with probability $\frac{f(w^*)}{m}$
- $f(x)$ - the distribution function of weights of edges

Step 3 In each subgraph $\widetilde{G}_i$ build a Hamiltonian cycle, using polynomial algorithms, that **w.h.p.** find a Hamiltonian cycle in a sparse random graph.



We will use here algorithms by Gimadi-Perepelitsa (1973) and Angluin-Valiant (1979).

- Steps 1 and 2 takes $O(n^2)$,
- At Step 3 the chosen algorithm with time complexity $T(n)$ runs m times.

The total time complexity of the approach is $O(n^2 + mT(n))$.

- Time complexity of the approach with Gimadi-Perepelitsa algorithm is $O(mn^2)$.
- Time complexity of the approach with Angluin-Valiant algorithm is $O(n^2 + mn\log^2 n)$.

Random graphs

Two concepts of a random graph:

1. G_p – n-vertex graph, where each edge exists with probability p, independently of other edges.

2. G_N – graph with n vertices and exactly N edges, chosen uniformly from the set of all such graphs.

In [Angluin-Valiant, 1979] it was shown that these two concepts are interchangeable for appropriate values N and p.

Existence of the HC in a random graph

1952 Dirac: A simple graph with n vertices ($n \geq 3$) is Hamiltonian if every vertex has degree $n/2$ or greater.

1959 Erdos and Renyi: for any $\varepsilon > 0$ if the number of edges $N < (1/2 - \varepsilon) n \log n$, then w.h.p. the graph contains isolated vertices (w.h.p. no HC).

1976 Posa: an undirected graphs with $N \geq cn \log n$ edges w.h.p. contains a Hamiltonian cycle. Non-algorithmic proof.

1983 Komlos and Szemeredi and independently Korshunov: the required density may be reduced to $N \geq 1/2(n \log n + n \log \log n + Q(n))$ edges, where $Q(n) \to \infty$ as $n \to \infty$.

Algorithms

1973 Gimadi and Perepelitsa: randomized algorithm A_{GP}, which in $O(n^2/\ln n)$ time finds w.h.p. a HC in a random graph with $N \geq n\sqrt{n \ln n}$ edges.

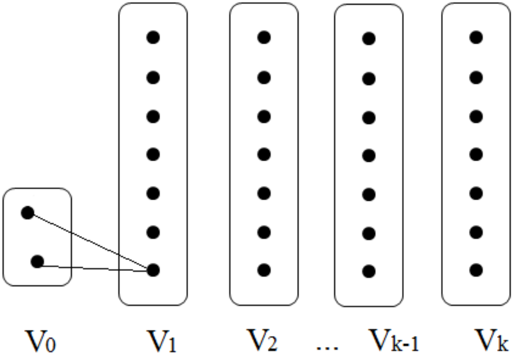
1979 Angluin and Valiant: randomized algorithm A_{AV}, which in $O(n \ln^2 n)$ time with probability $1 - O(n^{-\alpha})$ finds a HC in a random graph with $N \geq c_\alpha n \ln n$ edges, where $c_\alpha \sim 100 + \alpha$

1987 Bollobas, Fenner and Frieze: deterministic $O(n^{3+o(1)})$ algorithm that w.h.p. finds a HC in a random undirected graph with $N \geq 1/2(n \log n + n \log \log n + c_n n)$ edges.

1988 Frieze: algorithm with running-time $O(n^{1,5})$ for directed random graphs.

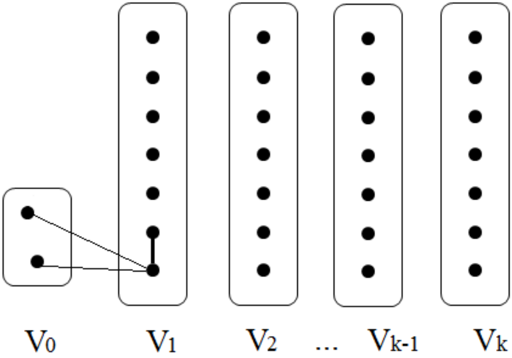
2015 Frieze and Haber: algorithm that in $O(n^{1+o(1)})$ time w.h.p finds a HC in a random graph with vertex degree ≥ 3 and $N = cn$ edges, where c is a sufficiently large constant.

- A_{GP} is trying to find a HC in a given random graph $G = (V, E)$
- If it cannot produce some step it stops and returns "failure"



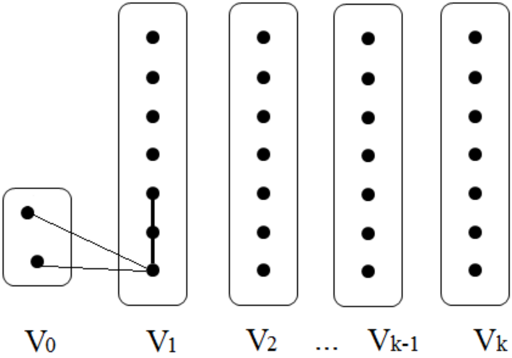
$|V_0| \sim 0.3\sqrt{n \ln n}; \qquad |V_1| = \ldots = |V_k| \sim \frac{n}{\ln n} - 0.3\sqrt{\frac{n}{\ln n}};$
$k = \lfloor \ln n/2 \rfloor.$

- A_{GP} is trying to find a HC in a given random graph $G = (V, E)$
- If it cannot produce some step it stops and returns "failure"



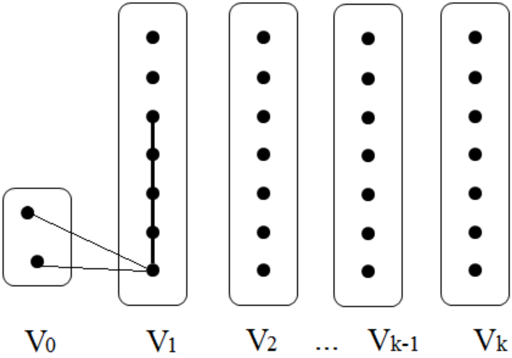
$|V_0| \sim 0.3\sqrt{n \ln n}$; $\quad |V_1| = \ldots = |V_k| \sim \frac{n}{\ln n} - 0.3\sqrt{\frac{n}{\ln n}}$;
$k = \lfloor \ln n/2 \rfloor$.

- A_{GP} is trying to find a HC in a given random graph $G = (V, E)$
- If it cannot produce some step it stops and returns "failure"



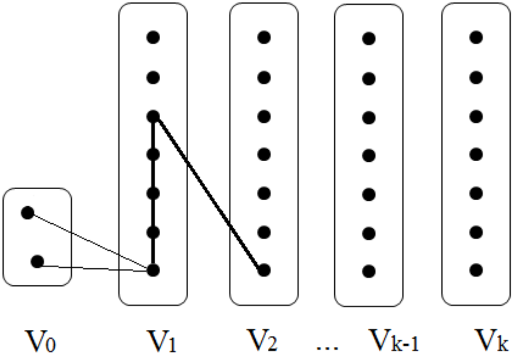
$|V_0| \sim 0.3\sqrt{n \ln n}$; $\quad |V_1| = \ldots = |V_k| \sim \frac{n}{\ln n} - 0.3\sqrt{\frac{n}{\ln n}}$;
$k = \lfloor \ln n/2 \rfloor$.

- A_{GP} is trying to find a HC in a given random graph $G = (V, E)$
- If it cannot produce some step it stops and returns "failure"



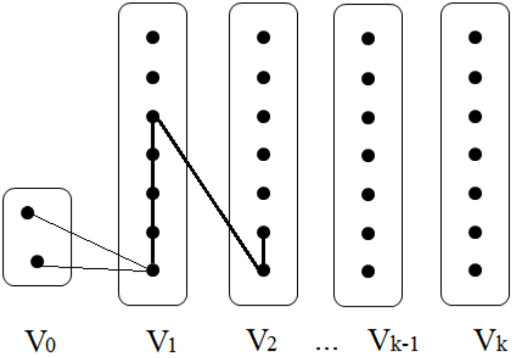
$|V_0| \sim 0.3\sqrt{n \ln n}$; $|V_1| = \ldots = |V_k| \sim \frac{n}{\ln n} - 0.3\sqrt{\frac{n}{\ln n}}$;
$k = \lfloor \ln n/2 \rfloor$.

- A_{GP} is trying to find a HC in a given random graph $G = (V, E)$
- If it cannot produce some step it stops and returns "failure"



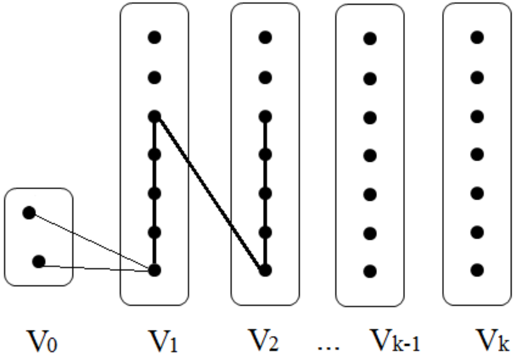
$|V_0| \sim 0.3\sqrt{n \ln n}$; $\quad |V_1| = \ldots = |V_k| \sim \frac{n}{\ln n} - 0.3\sqrt{\frac{n}{\ln n}}$;
$k = \lfloor \ln n/2 \rfloor$.

- A_{GP} is trying to find a HC in a given random graph $G = (V, E)$
- If it cannot produce some step it stops and returns "failure"



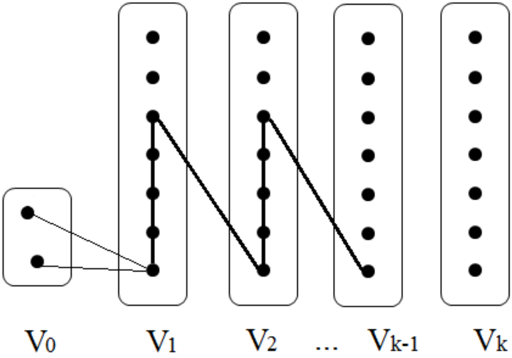
$|V_0| \sim 0.3\sqrt{n \ln n}$; $\quad |V_1| = \ldots = |V_k| \sim \frac{n}{\ln n} - 0.3\sqrt{\frac{n}{\ln n}}$;
$k = \lfloor \ln n/2 \rfloor$.

- A_{GP} is trying to find a HC in a given random graph $G = (V, E)$
- If it cannot produce some step it stops and returns "failure"



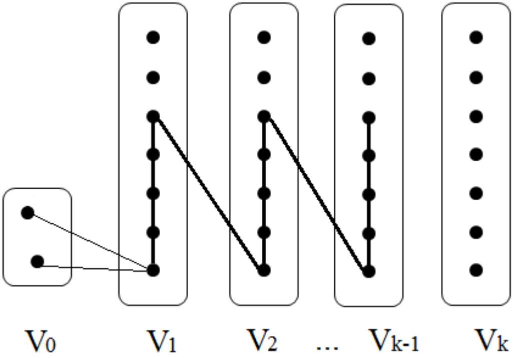
$|V_0| \sim 0.3\sqrt{n \ln n}$; $|V_1| = \ldots = |V_k| \sim \frac{n}{\ln n} - 0.3\sqrt{\frac{n}{\ln n}}$;
$k = \lfloor \ln n/2 \rfloor$.

- A_{GP} is trying to find a HC in a given random graph $G = (V, E)$
- If it cannot produce some step it stops and returns "failure"



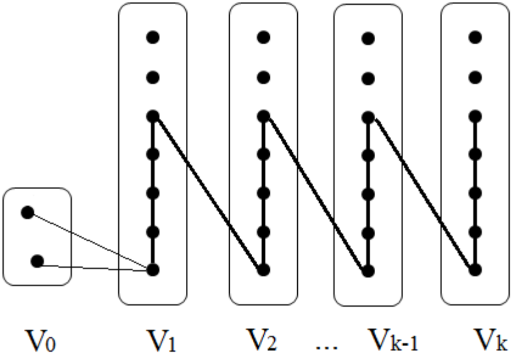
$|V_0| \sim 0.3\sqrt{n \ln n}$; $\quad |V_1| = \ldots = |V_k| \sim \frac{n}{\ln n} - 0.3\sqrt{\frac{n}{\ln n}}$;
$k = \lfloor \ln n/2 \rfloor$.

- A_{GP} is trying to find a HC in a given random graph $G = (V, E)$
- If it cannot produce some step it stops and returns "failure"



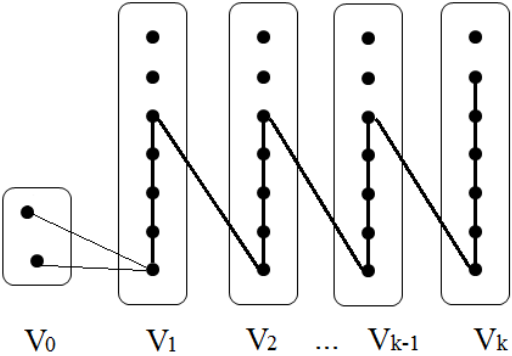
$|V_0| \sim 0.3\sqrt{n \ln n}$; $|V_1| = \ldots = |V_k| \sim \frac{n}{\ln n} - 0.3\sqrt{\frac{n}{\ln n}}$;
$k = \lfloor \ln n/2 \rfloor$.

- A_{GP} is trying to find a HC in a given random graph $G = (V, E)$
- If it cannot produce some step it stops and returns "failure"



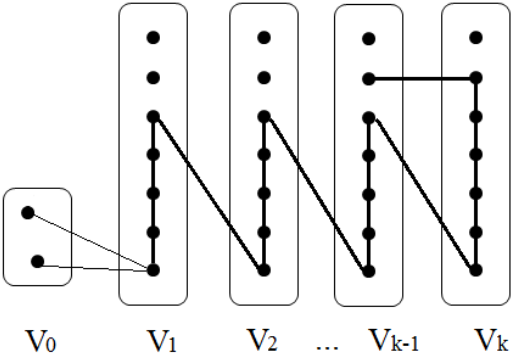
$|V_0| \sim 0.3\sqrt{n \ln n}$; $|V_1| = \ldots = |V_k| \sim \frac{n}{\ln n} - 0.3\sqrt{\frac{n}{\ln n}}$;
$k = \lfloor \ln n/2 \rfloor$.

- A_{GP} is trying to find a HC in a given random graph $G = (V, E)$
- If it cannot produce some step it stops and returns "failure"



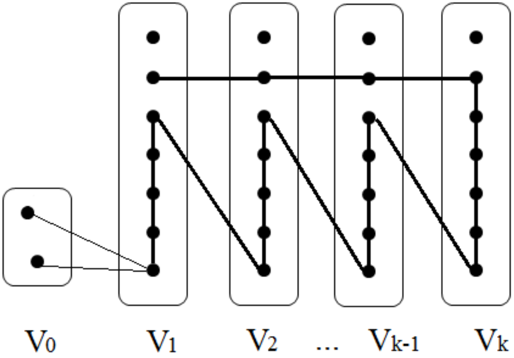
$|V_0| \sim 0.3\sqrt{n \ln n}$; $\quad |V_1| = \ldots = |V_k| \sim \frac{n}{\ln n} - 0.3\sqrt{\frac{n}{\ln n}}$;
$k = \lfloor \ln n/2 \rfloor$.

- A_{GP} is trying to find a HC in a given random graph $G = (V, E)$
- If it cannot produce some step it stops and returns "failure"



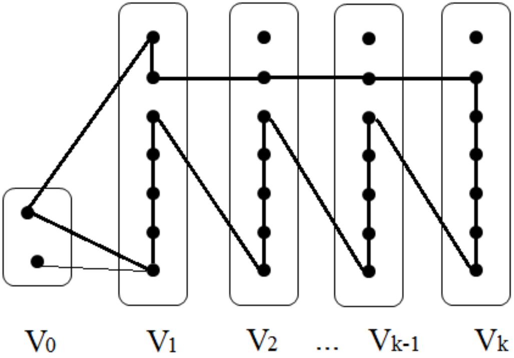
$|V_0| \sim 0.3\sqrt{n \ln n}$; $\quad |V_1| = \ldots = |V_k| \sim \frac{n}{\ln n} - 0.3\sqrt{\frac{n}{\ln n}}$;
$k = \lfloor \ln n/2 \rfloor$.

- A_{GP} is trying to find a HC in a given random graph $G = (V, E)$
- If it cannot produce some step it stops and returns "failure"



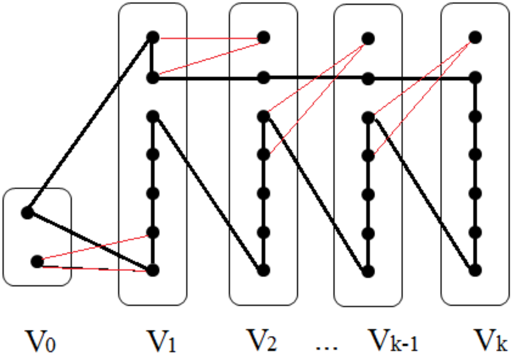
$|V_0| \sim 0.3\sqrt{n \ln n}$; $|V_1| = \ldots = |V_k| \sim \frac{n}{\ln n} - 0.3\sqrt{\frac{n}{\ln n}}$;
$k = \lfloor \ln n/2 \rfloor$.

- A_{GP} is trying to find a HC in a given random graph $G = (V, E)$
- If it cannot produce some step it stops and returns "failure"



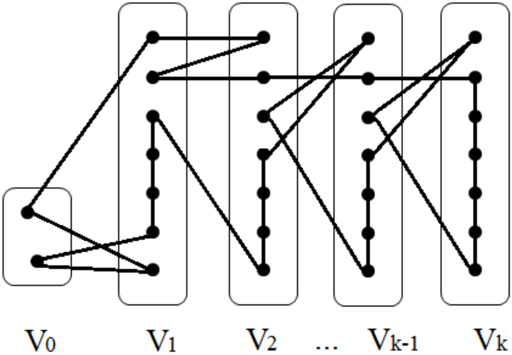
$|V_0| \sim 0.3\sqrt{n \ln n}$; $\quad |V_1| = \ldots = |V_k| \sim \frac{n}{\ln n} - 0.3\sqrt{\frac{n}{\ln n}}$;
$k = \lfloor \ln n/2 \rfloor$.

- A_{GP} is trying to find a HC in a given random graph $G = (V, E)$
- If it cannot produce some step it stops and returns "failure"



$|V_0| \sim 0.3\sqrt{n \ln n}$; $|V_1| = \ldots = |V_k| \sim \frac{n}{\ln n} - 0.3\sqrt{\frac{n}{\ln n}}$;
$k = \lfloor \ln n/2 \rfloor$.

- A_{GP} is trying to find a HC in a given random graph $G = (V, E)$
- If it cannot produce some step it stops and returns "failure"



$|V_0| \sim 0.3\sqrt{n \ln n}$; $\quad |V_1| = \ldots = |V_k| \sim \frac{n}{\ln n} - 0.3\sqrt{\frac{n}{\ln n}}$;
$k = \lfloor \ln n / 2 \rfloor$.

Theorem (Gimadi-Perepelitsa,1979)

Algorithm A_{GP} w.h.p. builds a Hamiltonian cycle in a random n-vertex graph with at least

$$N = n\sqrt{n \ln n}$$

edges. The failure probability of the algorithm is

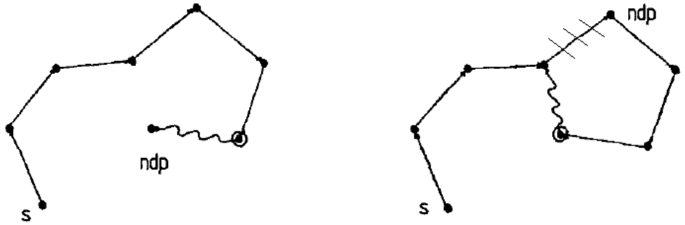
$$\delta_{GP} = O\left(\frac{\sqrt{\ln n}}{n^{1.5-o(1)}}\right) = O\left(\frac{\sqrt{\ln n}}{n^{0.8}}\right).$$

Time complexity of A_{GP} is $O(n^2/\ln n)$.

- − Complicated
- − Requires more than $cn \log n$ edges
- + Doesn't have big constants in the definition of N, thus can be used for small n.

Algorithm A_{VG} by Angluin and Valiant

- A_{VG} is trying to find a HC in a given random graph $G = (V, E)$
- Works almost like algorithm by Dirac
- If it cannot produce a step it stops and returns "failure"



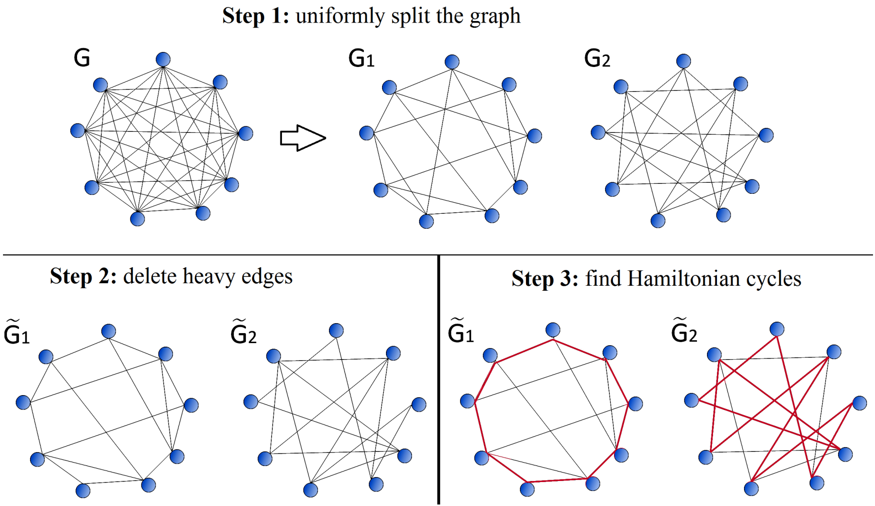
Theorem (Angluin-Valiant, 1979)

Algorithm A_{AV} with probability $1 - O(n^{-\alpha})$ finds a HC in an undirected random graph with

$$N \geq c_\alpha n \ln n$$

edges, where c_α is a sufficiently large constant.
The running-time of A_{AV} is $O(n \log^2 n)$

+ Algorithm is fast and simple.
+ It has a small failure probability $\delta_{AV} = O(n^{-\alpha})$
− Constant $c_\alpha \sim 100 + \alpha$.



Step 1: uniformly split the graph
G
G1
G2
Step 2: delete heavy edges
G̃1
G̃2
Step 3: find Hamiltonian cycles
G̃1
G̃2

Let the weights of the input graph be random reals with a continuous distribution function $f(x)$ defined on $[a_n, b_n]$ or $[a_n, \infty)$, $0 < a_n$. Let N be the required number of edges.

Setting w^*

$$\text{If} \quad w^* = f^{-1}\left(\frac{4m(N+n)}{n(n-1)}\right), \quad \text{then}$$

$$Pr\{\text{graph } \widetilde{G}_i \text{ does not have N edges at Step 3}\} \le e^{-n}$$

The approach gives the following performance guarantees.

The relative error is

$$\varepsilon_n < \frac{w^* - a_n}{a_n},$$

The failure probability is

$$\delta_n = \begin{cases} O(m \ln n / n^{0.8}), & \text{if we use } A_{GP} \text{ at Step 3,} \\ O(m/n^{\alpha}), & \text{if we use } A_{AV} \text{ at Step 3.} \end{cases}$$

Theorem

Let the distribution function of inputs be

- $UNI(x)$ *uniform on* $[a_n, \beta_n]$, $0 < a_n \leq \beta_n$,
- $Exp(x)$ *shifted exponential with parameters* a_n, β_n:

$$Exp(x) = 1 - \exp\left(-\frac{x - a_n}{\beta_n}\right),\ 0 < a_n \leq x\,.$$

The approach gives an asymptotically optimal m-PSP solution if:

- $m \leq n^{0.5-\theta}/4\,,\ 0 < \theta < 0.5,$ *и* $\beta_n/a_n = o\left(\frac{n^\theta}{\sqrt{\ln n}}\right)$, *and we use* A_{GP} *at Step 3.*
- $m \leq n^{1-\theta}/4\,,\ 0 < \theta < 1,$ *и* $\beta_n/a_n = o\left(\frac{n^\theta}{\ln n}\right)$, *and we use* A_{AV} *at Step 3.*

Remark

The obtained results can be extended to the case of any distribution

Theorem

Let the distribution function of the weights of edges of the input graph be discrete and defined on $[a, b]$ or $[a, \infty)$, where a is the the lowest possible weight of an edge. Let the distribution function satisfy:

$$p_a = Pr(X = a) \geq \frac{4m(\sqrt{n \ln n} + 1)}{n - 1}.$$

If $m \leq n^{0.3-\theta}/4$, where $0 \leq \theta < 0.3$, the approach with A_{GP} gives an exact solution of the m-PSP with failure probability $\delta_A = O\left(\frac{\sqrt{\ln n}}{n^{0.5+\theta}}\right)$.

In particular, the result is true for:

- Bernoulli distribution $\mathcal{B}_p$, where $p \leq 1 - \frac{\sqrt{\ln n}}{n^{0.2+\theta}}$;
- Geometric distribution $\mathcal{G}_p$, where $p \geq \frac{\sqrt{\ln n}}{n^{0.2+\theta}}$;
- Poisson distribution Π_λ, where $\lambda \leq \ln\left(\frac{n^{0.2+\theta}}{\sqrt{\ln n}}\right)$.

For the considered continuous distributions on $[a_n, \beta_n]$ or $[a_n, \infty)$:

Performance guarantees of the algorithm $\widetilde{A}$ for m-PSP-DW:

$$\varepsilon = O\left(\frac{\beta_n}{a_n}\frac{\ln n}{n}\right)$$

$$\delta = n^{-(3/4m+6/4)}$$

if

$$2 \leq m < \ln n$$

$$\frac{\beta_n}{a_n} = o\left(\frac{n}{\ln n}\right);$$

Performance guarantees of the new approach+A_{AV} for m-PSP:

$$\varepsilon = O\left(\frac{\beta_n}{a_n}\frac{\ln^2 n}{n}\right)$$

$$\delta = O\left(\frac{m}{n^\alpha}\right), \; \alpha > 1$$

if

$$2 \leq m < \ln n$$

$$\frac{\beta_n}{a_n} = o\left(\frac{n}{\ln^2 n}\right)$$

For the considered continuous distributions on $[a_n, \beta_n]$ or $[a_n, \infty)$:

Performance guarantees of the algorithm $\widetilde{A}$ for m-PSP-DW:

$$\varepsilon = O\Big(\frac{\beta_n}{a_n}\frac{1}{n^\theta}\Big)$$

$$\delta = n^{-(3/4m+6/4)}$$

if

$$\ln n < m \le n^{1-\theta}$$

$$\frac{\beta_n}{a_n} = o(n^\theta)$$

Performance guarantees of the new approach+A_{AV} for m-PSP:

$$\varepsilon = O\left(\frac{\beta_n}{a_n}\frac{\ln n}{n^\theta}\right)$$

$$\delta = O\left(\frac{m}{n^\alpha}\right), \alpha > 1$$

if

$$m \le n^{1-\theta}/4$$

$$\frac{\beta_n}{a_n} = o\left(\frac{n^\theta}{\ln n}\right)$$

The m-Cycles Cover Problem

Given: a complete n-vertex weighted graph $G = (V, E)$,
The problem is to find m vertex-disjoint simple cycles of extreme total weight, that spans all vertices of G.

$$W(\widetilde{C}) = \sum_{i=1}^{m} \sum_{e \in E(C_i)} w(e) \to \min(\max),$$

$$V(C_1) \cup \ldots \cup V(C_m) = V,$$

$$V(C_i) \cap V(C_j) = \varnothing, i \neq j, 1 \leq i, j \leq m.$$

It was first stated in [Khachay, Neznakhina, 2014]

Khachay, Neznakhina, 2014

The problem is NP-hard
2-approximation algorithm for metric m-CCP$_{min}$.

Khachay, Neznakhina, 2015

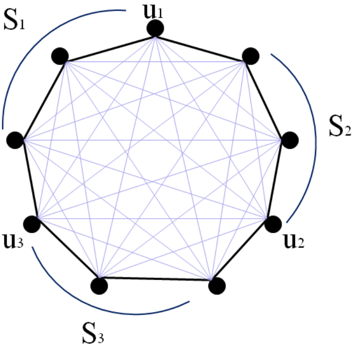
PTAS with relative error $1/c$ and time complexity $O(n^3(\ln n)^{O(c)})$ for Euclidean m-CCP$_{min}$.

Gimadi, Rykov 2015

TSP-approach and

- Asymptotically optimal algorithm for Euclidean m-CCP$_{max}$, if $m = o(n)$.
- Asymptotically optimal algorithm for m-CCP$_{min}$ on random UNI[0,1] inputs, if $m \leq n^{1/3}/\ln n$.

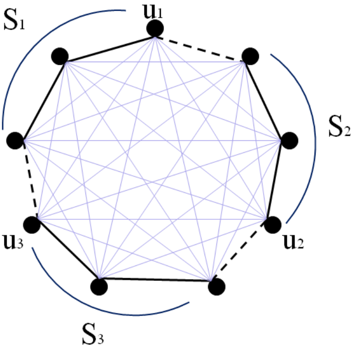
Step 1 Let $H = \{1, \ldots, n\}$ be the approximate TSP solution.



- $u = \{u_1, \ldots, u_m\}$ is a feasible partition of H into a collection of paths $S_k = [u_{k-1} + 1, u_k]$, if $1 \leq u_1 < \ldots, < u_m \leq n, u_0 = u_m$, and each path S_k contains at least 2 edges.

3. The m-Cycles Cover Problem
The TSP-approach: TSP solution $\to$ m-CCP solution

Step 1 Let $H = \{1, \ldots, n\}$ be the approximate TSP solution.

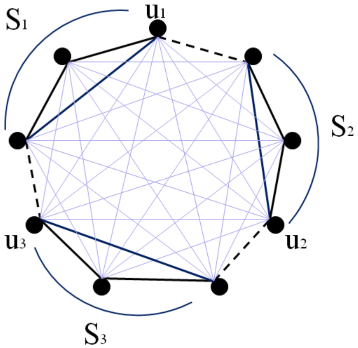


- $u = \{u_1, \ldots, u_m\}$ is a feasible partition of H into a collection of paths $S_k = [u_{k-1}+1, u_k]$, if $1 \le u_1 < \ldots, < u_m \le n, u_0 = u_m$, and each path S_k contains at least 2 edges.

3. The m-Cycles Cover Problem

The TSP-approach: TSP solution $\rightarrow$ m-CCP solution

Step 1 Let $H = \{1, \ldots, n\}$ be the approximate TSP solution.



- $u = \{u_1, \ldots, u_m\}$ is a feasible partition of H into a collection of paths $S_k = [u_{k-1} + 1, u_k]$, if $1 \le u_1 < \ldots, < u_m \le n, u_0 = u_m$, and each path S_k contains at least 2 edges.

Step 2a Find the best feasible partition of H:

$$f(u) = \sum_{k=1}^{m} \Big(w(u_{k-1}+1, u_k) - w(u_k, u_k+1) \Big) \rightarrow \min_{u \in U} (\max_{u \in U}).$$

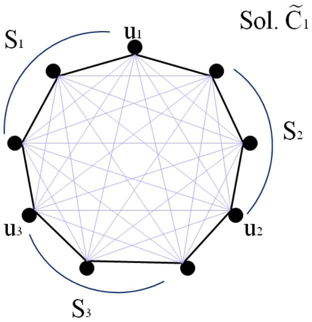
where U is a set of all feasible partitions of H.

- The dynamic programming algorithm gives the exact solution in $O(mn^3)$ time.
- According to the best feasible partition delete separating edges from H, add closing edges and return the obtained m-cycles cover as a result of the approach.

In the case of maximization problem

Step 2b Find the approximately best feasible partition of H in $O(nm)$ time:

- Choose arbitrary u_k, $1 \leq k \leq m$, such that $u_k - u_{k-1} \geq 3$.

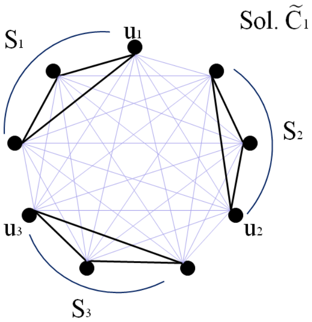


$\widetilde{C} = arg\max_{0 \leq j < n} W(\widetilde{C}_j)$.

In the case of maximization problem

Step 2b Find the approximately best feasible partition of H in $O(nm)$ time:

- Choose arbitrary u_k, $1 \le k \le m$, such that $u_k - u_{k-1} \ge 3$.

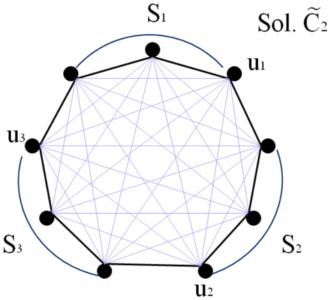


$\widetilde{C} = arg\max_{0 \le j < n} W(\widetilde{C}_j)$.

In the case of maximization problem

Step 2b Find the approximately best feasible partition of H in $O(nm)$ time:

- Choose arbitrary u_k, $1 \leq k \leq m$, such that $u_k - u_{k-1} \geq 3$.

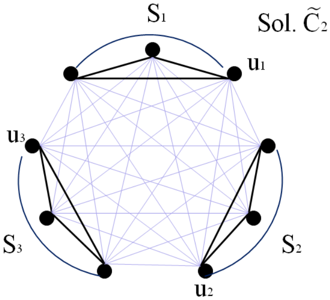


$\widetilde{C} = arg\max_{0 \leq j < n} W(\widetilde{C}_j)$.

In the case of maximization problem

Step 2b Find the approximately best feasible partition of H in $O(nm)$ time:

- Choose arbitrary u_k, $1 \leq k \leq m$, such that $u_k - u_{k-1} \geq 3$.



$\widetilde{C} = arg\max_{0 \leq j < n} W(\widetilde{C}_j)$.

- The TSP-approach **does not** guarantee that the approximation ratio that was true for the TSP algorithm would also hold for the m-CCP.

Lemma (Gimadi, Rykov, 2015)

Let H be a Hamiltonian cycle, and $\widetilde{C}$ be the solution of the MAX m-Cycles Cover problem, obtained by applying procedure from Step 2b to a Hamiltonian cycle H. For the weight of $\widetilde{C}$ we have:

$$W(\widetilde{C}) \geq (1 - m/n)W(H).$$

- It is not the optimal solution of TSP, that important here, but the preliminary constructions (spanning tree, cycle cover, perfect matching) form the approximation algorithm for the TSP.

Lemma: MIN m-CCP

The solution of the m-CCP may contain cycles of odd length, so in the case of minimization problem the weight of the perfect matching M_{min} can be in many times greater than the weight of the m-CCP solution in general, metric or Euclidean cases.

Lemma: symmetric MAX m-CCP

Let C^* be the optimal solution of maximum m-CCP with non-negative edge weights. We assume that a cycle should consist of at least 3 vertices. Then

$$W(C^*) \leq 3W(M).$$

Lemma: metric MAX m-CCP [Gimadi, Rykov, 2015]

Let C^* be the optimal solution of the metric maximum m-CCP. Then:

$$(1 - m/n)W(C^*) \leq 2W(M).$$

Some reasnt papers

1. **Tsidulko O. Yu.** On solvability of the axial 8-index assignment problem on single-cycle permutations// Diskretnyi Analiz i Issledovanie Operatsii, 2013, Vol. 20, No. 5, pp. 66–83.

2. **Gimadi E. Kh. , Glazkov Yu. V., Tsidulko O. Yu.** Probabilistic analysis of an algorithm for the m-planar 3-index assignment problem on single-cycle permutations.// Diskretnyi Analiz i Issledovanie Operatsii, 2014, Vol. 21, No. 1, pp. 15–29

3. **Gimadi E. Kh., Istomin A. M., Rykov I. A., Tsidulko O. Yu.** Probabilistic analysis of an approximation algorithm for the m-peripatetic salesman problem on random instances unbounded from above // Proceedings of the Steklov Institute of Mathematics (Supplementary issues), 2015, 289, suppl. 1, 77–87

4. **Gimadi E. Kh., Glebov A. N., Skretneva A. A., Tsidulko O. Yu., Zambalaeva D. Zh.** Combinatorial algorithms with performance guarantees for finding several Hamiltonian circuits in a complete directed weighted graph. // Discrete Appl. Math. 2015, Vol. 196, pp 54-–61.

5. **Gimadi E. Kh., Istomin A. M. , Tsidulko O. Yu.** On Asymptotically Optimal Approach to the m-Peripatetic Salesman Problem on Random Inputs. // In: Kochetov, Yu. et all (eds.) DOOR-2016. LNCS, vol. 9869, P. 136–148.

6. **Gimadi E., Rykov I., Tsidulko O.** The TSP-Approach to Approximate Solving the m-Cycles Cover Problem. // Book of abstracts of the 2nd International Conference and Summer School "Numerical Computations: Theory and Algorithms" NUMTA-2016, Calabria, Italy, 19-25 June 2016. P. 55.