Graph-Theoretical Approaches in Big Optimization

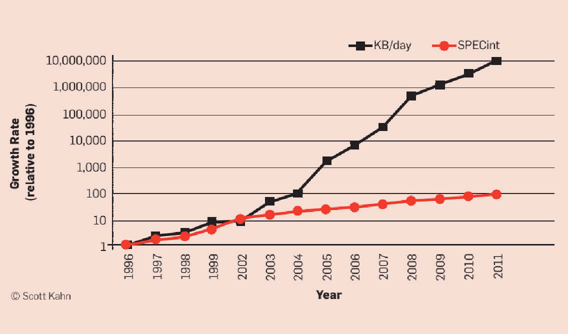
Nurminski E.A.
nurminskiy.ea@dvfu.ru

Far Eastern Federal University, Vladivostok

Optmization Methods and Applications
Irkutsk, June 30 – July 06, 2014

Big Data vs Big Computing

Figure 1. Next-gen sequence data size compared to SPECint.



Source: Comm. ACM, vol. 57(7), 2014.

Optimization dream: to solve $\infty \times \infty$ problem

Optimization dream: to solve $\infty \times \infty$ problem

Successive approximations:

Optimization dream: to solve $\infty \times \infty$ problem

Successive approximations:

- Megabyte-optimization: $10^6 - 10^8$ variables/constraints;

Optimization dream: to solve $\infty \times \infty$ problem

Successive approximations:

- Megabyte-optimization: $10^6 - 10^8$ variables/constraints;
- Gigabyte-optimization: $10^9 - 10^{11}$ variables/constraints;

Optimization dream: to solve $\infty \times \infty$ problem

Successive approximations:

- Megabyte-optimization: $10^6 - 10^8$ variables/constraints;
- Gigabyte-optimization: $10^9 - 10^{11}$ variables/constraints;
- Terabyte-optimization: $10^{12} - 10^{14}$ variables/constraints;

Optimization dream: to solve $\infty \times \infty$ problem

Successive approximations:

- Megabyte-optimization: $10^6 - 10^8$ variables/constraints;
- Gigabyte-optimization: $10^9 - 10^{11}$ variables/constraints;
- Terabyte-optimization: $10^{12} - 10^{14}$ variables/constraints;
- etc ...

Simple algorithms 1

Coordinate descent:

- Y. Nesterov, Efficiency of coordinate descent methods on huge-scale optimization problems, SIAM Journal on Optimization, vol. 22, no. 2, pp. 341-362, 2012.
- Z. Qin, K. Scheinberg, and D. Goldfarb, Efficient block-coordinate descent algorithms for the group lasso, Mathematical Programming Computation, vol. 5, pp. 143-169, June 2013.
- I. Necoara and D. Clipici, Efficient parallel coordinate descent algorithm for convex optimization problems with separable constraints:application to distributed MPC, Journal of Process Control, vol. 23, no. 3, pp. 243-253, March 2013
- etc ...

Simple algorithms 2

Gradient-type algorithms:

- Y. Nesterov, Gradient methods for minimizing composite functions, Mathematical Programming, vol. 140, pp. 125-161, 2013.
- M.A.T. Figueiredo, R.D. Nowak, S.J. Wright Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems IEEE J. Sel.Topics in Signal Processing
- etc . . .

Simple algorithms 3

Projection methods.

- Bauschke H., Borwein J. Projection Methods, SIAM J. Optimization, 1996
- D. Henrion and J. Malick. Projection methods for conic feasibility problems; application to sum-of-squares decompositions Optimization Methods and Software, 26(1):23-46, 2011.
- D. Henrion, J. Malick Projection methods in conic optimization Optimization Online.
- J. Nie Regularization methods for sum of squares relaxations in large scale polynomial optimization. Technical report, ArXiv, 2009.
- And many others ...

Projection methods. Notation

Our main tool:
$\Pi_X : E \to X$ — the orthogonal projection operator.

$$\Pi_X(x) = \operatorname{argmin} \min_{z \in X} \|x - z\|^2.$$

Basic property: non-expansive and commonly contractive or can be easily modified to such.

Connections to optimization

Convex optimization problem:

$$f(x^\star) = \min_{x \in X} f(x)$$

Equivalent formulations:

- Variational optimality condition

$$g(x - x^\star) \geq 0$$
$$\forall x \in X, \; g \in \partial f(x^\star)$$

- Fixed point problem for the projection operator:

$$x^\star = \Phi_{X,\lambda}(x^\star),$$

where $\Phi_{X,\lambda}(x) = \Pi_X(x - \lambda g), \; g \in \partial f(x)$, $\Pi_X(\cdot)$ is the orthogonal projection operator, $\lambda > 0$ step multiplier.

Algorithmika

Iteration method:

$$x^{k+1} = \Phi_{X,\lambda}(x^k), \; k = 0, 1, \dots$$

or

$$x^{k+1} = \Pi_X(x^k - \lambda g^k), \; g^k \in \partial f(x^k), \; k = 0, 1, \dots$$

Properties:

+ Theoretically sound: based on contraction property of $\Pi_X(\cdot)$);
+ Suitable for Big O with respect to memory requirements;
+ Provides decomposition/parallelization opportunities.
- Slow (linear) convergence;
- Nontrivial sub-problem to solve.

Reduction to projection

Seems to be a folklore [1]

$$\begin{array}{ccc} \min cx & = & \min cx \\ Ax \leq b & & x \in X \end{array} \rightarrow \Pi_X(x^0 - \theta c)$$

for arbitrary x^0 and large enough $\theta > 0$.
As

$$\Pi_X(x^0 - \theta c) = \Pi_{X - x^0 + \theta c}(0) = \Pi_{\bar{X}}(0)$$

it amounts to the least norm problem for the set $\bar{X}$

$$\bar{X} = \{x : Ax \leq \bar{b}\}$$

where $\bar{b} = b - A(x^0 - \theta c)$.

[1] But true for polyhedral sets only ...

Least Norm Problem

Basic steps:

- Step 1. Change to uniform constraints
- Step 2. Use exact penalty function
- Step 3. Dualize
- Step 4. Project onto polyhedral cone

Step 1. Change to uniform constraints

We convert to cone constraints by adding extra $(n+1\text{-th})$ variable:

$$X = \{x : Ax \leq b\} \rightarrow \bar{X} = \{\bar{x} : \bar{A}\bar{x} \leq 0\}, \bar{x}e^{n+1} = 1$$

where

- $\bar{x} = (x, \xi) \in E^{n+1}$
- $\bar{A} = \|A \quad - b\|$

Step 2. Use exact penalty function

We use nondifferentiable exact penalty function to get rid of the all but one constraint:

$$\min_{\substack{\bar{x} \in \bar{X} \\ \bar{x}e^{n+1} = 1}} \frac{1}{2}\|\bar{x}\|^2 = \min_{\bar{x}e^{n+1} = 1} \{\frac{1}{2}\|\bar{x}\|^2 + \gamma|\bar{A}\bar{x}|_\infty^+\}$$

for $\gamma > 0$ large enough, $|\cdot|_\infty^+ = \max\{0, \cdot\}$.

Notice that

$$|\bar{A}\bar{x}|_\infty^+ = \max_{\lambda \in \Delta_{m+1}} (\lambda_0 0 + \textstyle\sum_{i=1}^m \lambda_i \bar{A}_i)\bar{x} = \max_{z \in \bar{\mathcal{A}}} z\bar{x} = (\bar{\mathcal{A}})_{\bar{x}}$$

— the support function of the set $\bar{\mathcal{A}} = \mathrm{co}\{0, \bar{A}_i, i = 1, 2, \ldots, m\}$.

Step 3. Dualization

Finally the problem is dualized with respect to the single non-uniform constraint:

$$\min_{\bar{x}e^{n+1} = 1} \left\{ \frac{1}{2}\|\bar{x}\|^2 + \gamma \left(\bar{\mathcal{A}}\right)_{\bar{x}} \right\} =$$

$$\max_u \min_{\bar{x}} \left\{ \frac{1}{2}\|\bar{x}\|^2 + \gamma \left(\bar{\mathcal{A}}\right)_{\bar{x}} + u(\bar{x}e^{n+1} - 1) \right\} = \max_u \{ \phi_\gamma(u) - u \}$$

where

$$\phi_\gamma(u) = \min_{\bar{x}} \left\{ \frac{1}{2}\|\bar{x}\|^2 + \gamma \left(\bar{\mathcal{A}}\right)_{\bar{x}} + u\bar{x}e^{n+1} \right\} =$$
$$\min_{\bar{x}} \left\{ \frac{1}{2}\|\bar{x}\|^2 + \left(\gamma\bar{\mathcal{A}} - ue^{n+1}\right)_{\bar{x}} \right.$$

As it happened the function $\phi_\gamma(u)$ becomes trivial for large γ.

Smth from Convex Analysis

Not too well-known but useful formula from convex analysis

$$\Pi_{Ax\leq b}(0) = \min_{Ax \leq b} \frac{1}{2}\|x\|^2 = -\min_u\{\frac{1}{2}\|u\|^2 + (Ax \leq b)_u\}$$

where $(Z)_u = \sup_{z\in Z} uz$ — the support function of the set Z.

Easy to derive from $Z = \mathrm{co}\{\hat{z}^1, \hat{z}^2, \ldots, \hat{z}^N\}$ for polyhedral Z and for a general convex set in the finite dimensional case thanks to Karatheodori theorem.

Step 4. Projection on the polyhedral cone

The final result of the transformations above was to hide all complexities into

$$\phi_\gamma(u) = \min_{\bar{x}}\{\frac{1}{2}\|\bar{x}\|^2 + \left(\gamma\bar{\mathcal{A}} - ue^{n+1}\right)_{\bar{x}}\}$$

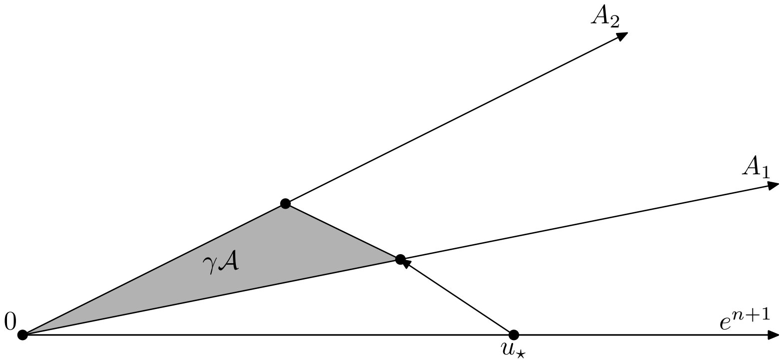
but from the previous slide

$$\phi_\gamma(u) = \min_{\bar{x} \in \gamma\bar{\mathcal{A}} - ue^{n+1}} \frac{1}{2}\|\bar{x}\|^2 =$$

$$\min_{\bar{x} \in \gamma\bar{\mathcal{A}}} \frac{1}{2}\|\bar{x} - ue^{n+1}\|^2 = \Pi_{\gamma\bar{\mathcal{A}}}(ue^{n+1})$$

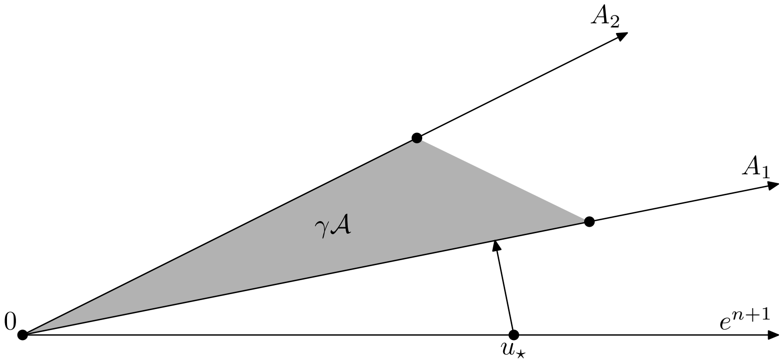
and this is very simple problem for large γ.

Small γ



The value of the function $\phi_\gamma(u)$ depends on γ for a given $u_\star$ for (small) changes in γ.

Big γ



The value of the function $\phi_\gamma(u)$ does not depend on further increase of γ for a given $u_\star$.

Formal proof

- Let $K_{\mathcal{A}} = \cup_{\lambda \geq 0} \lambda \mathcal{A}$, of course convex cone.
- Then $dist(ue^{n+1}, K_{\mathcal{A}}) = \phi(u) = \phi(1)u^2 \leq \phi_\gamma(u)$ due to linearity and $\lambda\mathcal{A} \subset K_{\mathcal{A}}$.
- Let $u_\star$ solves $\min_u \{\phi(u) - u\}$. Actually $u_\star = 1/2\phi(1)$. Then $\phi(u_\star) = \|z^\star - u_\star e^{n+1}\|^2$ with $z^\star \in K_{\mathcal{A}}$ and therefore $z^\star \in \lambda\mathcal{A}$ for all λ greater then certain $\lambda_\star$.
- It leads to $\phi_\lambda(u_\star) \leq \phi(u_\star)$ for such $\lambda \geq \lambda_\star$ and hence $\phi_\lambda(u_\star) = \phi(u_\star)$.
- Finally from $\phi_\gamma(u) - u \geq \phi(u) - u$ for all u and

$$\phi_\gamma(u_\star) - u_\star = \phi(u_\star) - u_\star \leq \phi(u) - u \leq \phi_\gamma(u) - u$$

follows that $u_\star$ in fact minimizes $\phi_\lambda(u) - u$.

Projection on the polyhedral cone

The key computational problem:

$$\phi(1) = \min_{z \in K(\mathcal{A})} \|z - e^{n+1}\|^2 = \Pi_{K(\mathcal{A})}(e^{n+1}),$$

where $K(\mathcal{A}) = \mathrm{Co}\{\bar{A}_i, i = 1, 2, \ldots, m\}$.

Closed form solution $z^\star = \bar{A}'(\bar{A}\bar{A}')^{-1}\bar{A}e^{n+1}$ is unrealistic however for Big O as well as the chain rule

$$v = \bar{A}e^{n+1} \rightarrow (\bar{A}\bar{A}')w = v \rightarrow z^\star = \bar{A}'w$$

Decomposition

We can use the property of the conical hull

$$K(\mathcal{A}) = \mathrm{Co}\{\bar{A}_i, i = 1, 2, \ldots, m\} = \mathrm{Co}\{\mathrm{Co}\{\bar{\mathcal{A}}_k\}, k = 1, 2, \ldots, K\},$$

where $\bar{\mathcal{A}}_k = \{\bar{A}_i, i \in I_k\}$ and index sets I_k cover the whole range of rows $1, 2, \ldots, m$.

In turn $\Pi_{K(\bar{\mathcal{A}})}(e^{n+1})$ can be reduced to separate (and parallel) projections $\Pi_{K(\bar{\mathcal{A}}_k)}(e^{n+1})$ (with slight modifications).

Sketch of Decomposition-Coordination

General idea: iterate between two steps Coordination (C) and Decomposition (D):

C: Get proposals $\bar{z}^k, k = 1, 2, \ldots, K$ from each of sub-problems and solve the coordination problem

$$\|\bar{z} - e^{n+1}\|^2 = \min \|z - e^{n+1}\|^2, z \in \mathrm{Co}\{z^k, k = 1, 2, \ldots, K\}$$

D: For each of sub-problems modify the feasible cone $\tilde{K}_k = \mathrm{Co}\{\bar{z}, K(\bar{\mathcal{A}}_k)\}$ and solve for $k = 1, 2, \ldots, K$ sub-problems

$$\|\bar{z}^k - e^{n+1}\|^2 = \min \|z - e^{n+1}\|^2, z \in \tilde{K}_k\}$$

to get new proposals $\bar{z}^k, k = 1, 2, \ldots, K$.

To initialize the process one can use of course an arbitrary $\bar{z}^k \in K(\bar{\mathcal{A}}_k)$ or to think of smth not that stupid.

Historical remarks

This idea can be traced back at least as far as Demyanov V.F., Malozemov V.N. *Introduction to Minmax*, M.: Nauka, where it was used in its simplest form.

In its current form in was proposed by Nurminski E. (IzVuz, somewhere in '90).

Some improvements due to Nurminski E, Dolgy D. in Korean Univ publication, 2012.

Probably there are many similar proposals, pls let me know.

Caveats

- $\mathcal{A}_k$ should not be too big or ill-structured to complicate solutions of sub-problems.
- $\mathcal{A}_k$ should not be too small to make the coordination problem too big or complicated.

Definition Vectors a and b from E^n are called structurally orthogonal (or independent) if $a_i b_i = 0$ for all $i = 1, 2, \ldots, n$. [2]

Definition An $m \times n$ matrix A is called structurally orthogonal if its rows $A_i, i = 1, 2, \ldots, m$ are structurally orthogonal.

For such matrices the projection problem has linear complexity as AA' is a diagonal matrix.

[2] Of course it is more restrictive than orthogonality (implies it) and less restrictive than complementarity (no sign constraints). Somewhere in between.

Structurally Orthogonal Decomposition

Select I_k such that the corresponding $\bar{A}_k$ are structurally (s-) orthogonal.

Define graph $G_A = (V_A, E_A)$, where

- V_A — the set of rows of A,
- E_A — the set of edges $e = (v_1, v_2) \in V_A \times V_A$ such that v_1 is *NOT* s-orthogonal to v_2.

The set of mutually s-orthogonal rows is a set of *independent* nodes in graph G.

Problem: Decompose the graph into minimal number of independent components (coloring).

Specifics: Graphs are huge, but sparse. Solutions with relatively small (up to 10^4) uncolored reminders are acceptable.

Heuristics

Greedy:

- try to build from the current graph an independent set as big as possible;
- delete this set from the graph (with incidents edges of course);
- continue with the rest of the graph.

Allows for many variants.

LP-specifics

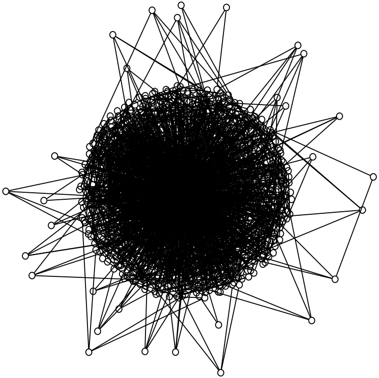
Simplest cases:

- Sign constraints: $x \geq 0$ — all constraints are s-orthogonal;
- Two-sided constrains: $l \leq x \leq u$ — 2 s-orth sets, 2^n variants, any combination of lower-upper constraints;
- Transportation problem: 2 s-orth sets, supply balances and demand balances or any combination;
- Canonical $m \times n$, $m << n$ LP:

$$\min cx$$
$$Ax = b; x \geq 0$$

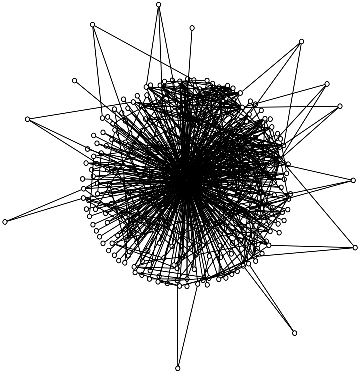
1 s-ort set (sign constraints), general equality constraints as "reminder".

SHELL_0 (www.netlib.org) 537 cnst, 1775 vars, 4900 nz (0.5%)



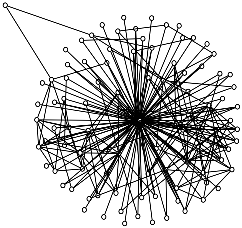
Graph stat: 537 nodes, 2210 arcs.

SHELL_1 (www.netlib.org) $|IS| = 278$



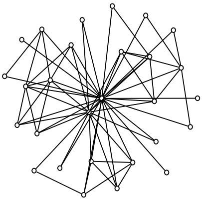
Graph stat: 257 nodes, 781 arcs

SHELL_2 (www.netlib.org) $|IS| = 159$



Graph stat: 96 nodes, 254 arcs

SHELL_3 (www.netlib.org) $|IS| = 68$

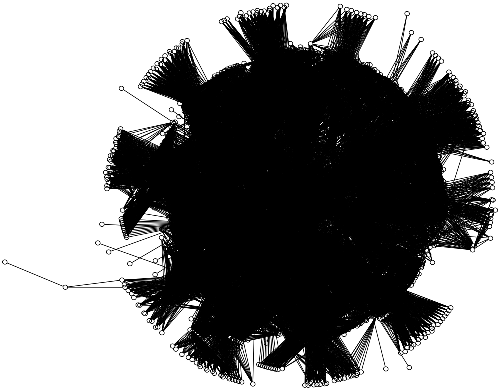


Graph stat: 28 nodes, 65 arcs

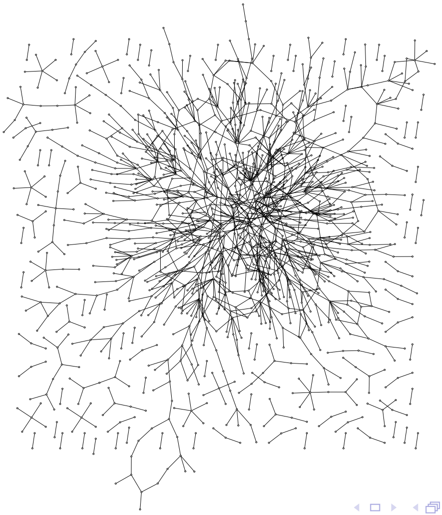
SHELL (www.netlib.org) Summary of the selection process

Desc	Nodes	Arcs	Av.degree	Indp. set
SHELL_0	537	2210	4.1155	278
SHELL_1	257	781	3.0389	159
SHELL_2	96	254	2.6458	68
SHELL_3	28	65	2.3214	

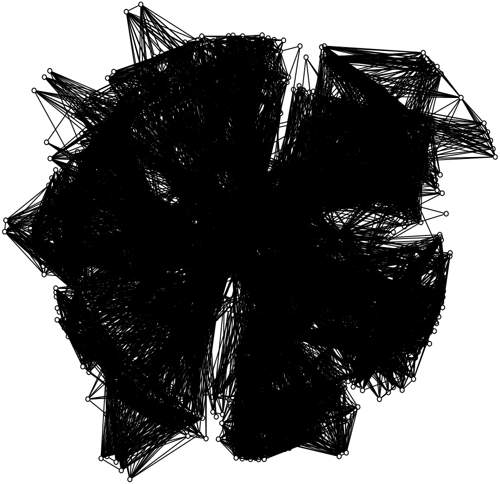
GREENBEA, www.netlib.org 2374x5323x30230 (0.24%)



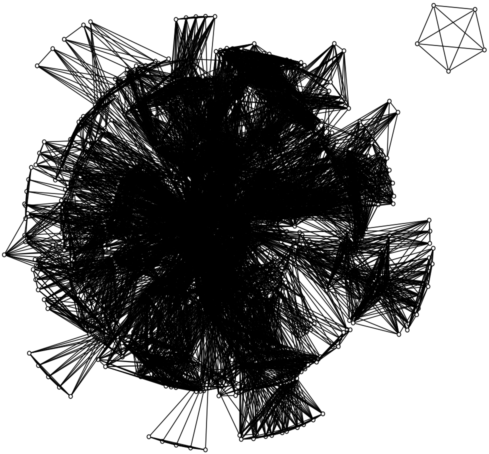
GREENBEA/20 – the giant core (1105 constraints)



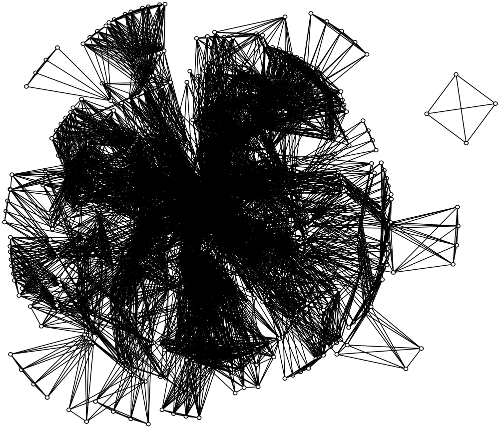
GREENBEA_1 1313 cnst, $|IS| = 1075$, 18148 ars



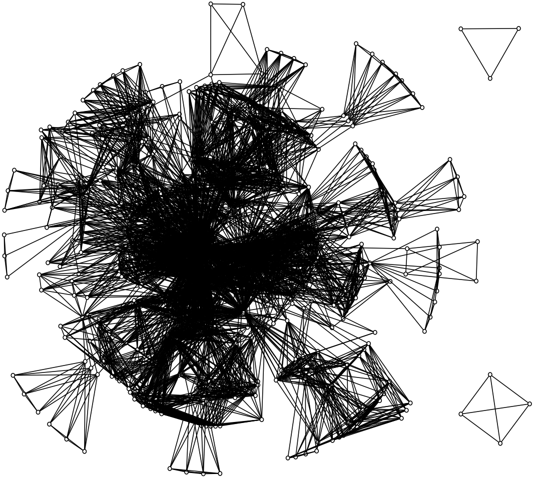
GREENBEA_2 866 cnst, $|IS| = 447$, 9938 arcs



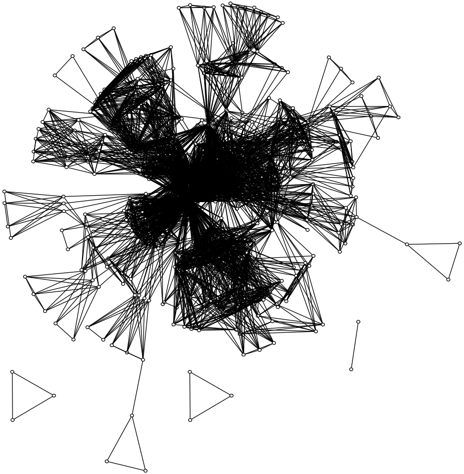
GREENBEA_3 685 cnst, $|IS| = 181$, 7415 arcs



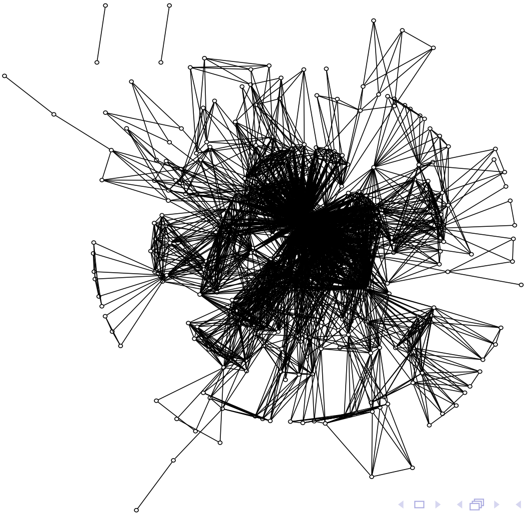
GREENBEA_4 549 cnst, $|IS| = 136$, 5490 arcs



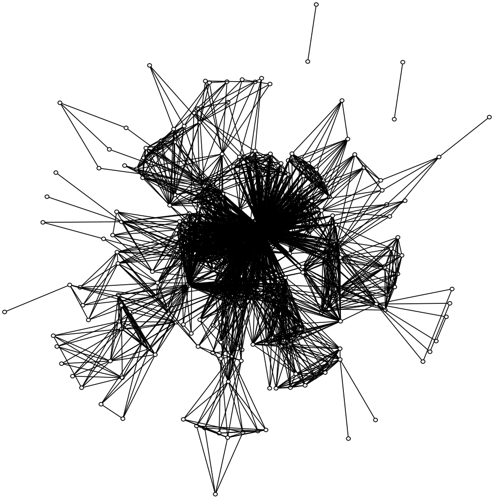
GREENBEA_5 450 cnst, $|IS| = 99$, 4310 arcs



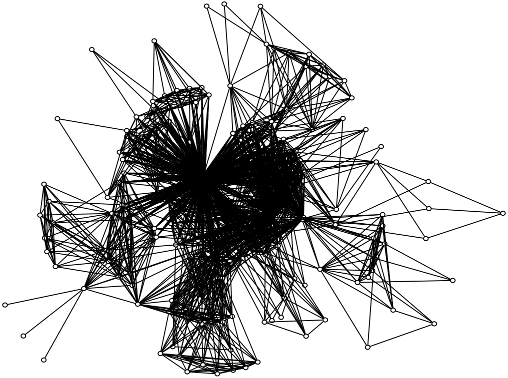
GREENBEA_6 350 cnst, $|IS| = 99$, 3135 arcs



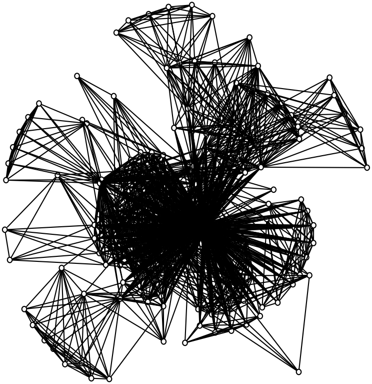
GREENBEA_7 264 cnst, $|IS| = 81$, 2470 arcs



GREENBEA_8 191 cnst, $|IS| = 67$, 1960 arcs

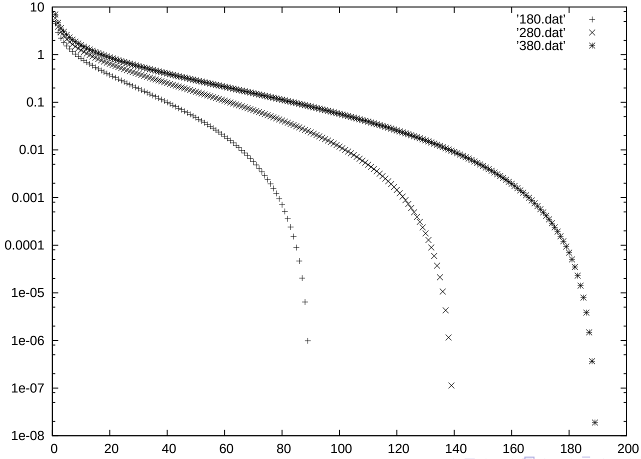


GREENBEA_9 149 cnst, $|IS| = 40$, 1601 arcs



Desc	Nodes	Arcs	Av.degree	Indp. set
GREENBEA_1	2388	34294	6.9633	1075
GREENBEA_2	1313	18148	7.2350	447
GREENBEA_3	866	9938	8.7140	181
GREENBEA_4	685	7415	9.2380	136
GREENBEA_5	549	5490	10.0000	99
GREENBEA_6	450	4310	10.4408	99
GREENBEA_7	350	3135	11.1643	81
GREENBEA_7	264	2470	10.6883	67
GREENBEA_9	191	1960	9.7449	40

Convergence of the projection procedure



Almost Gigabyte-Optimization