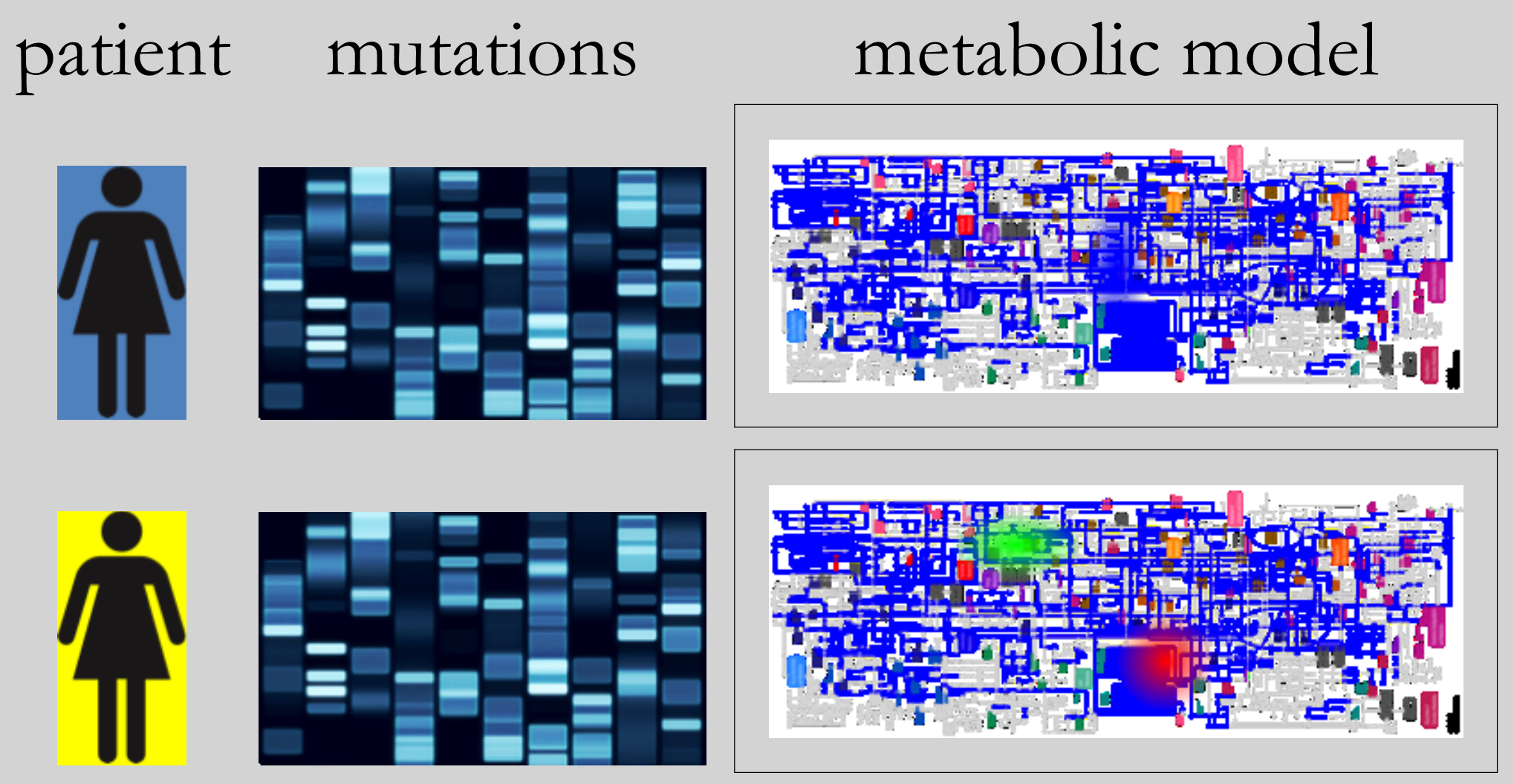
Finding dense subgraphs in relational graphs

Vinay Jethava, Niko Beerenwinkel

vinay.jethava@bsse.ethz.ch, niko.beerenwinkel@bsse.ethz.ch

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Coherent sub-networks in genome-scale metabolic models



http://www.metabolicatlas.org/

- 917 patients
- $|V| \sim 4000$
- $|E| \sim 15000$

Aim Find **dense common subgraphs** in patients with specific markers e.g. $mutbrca = 1$ and $mutp53 = 1$

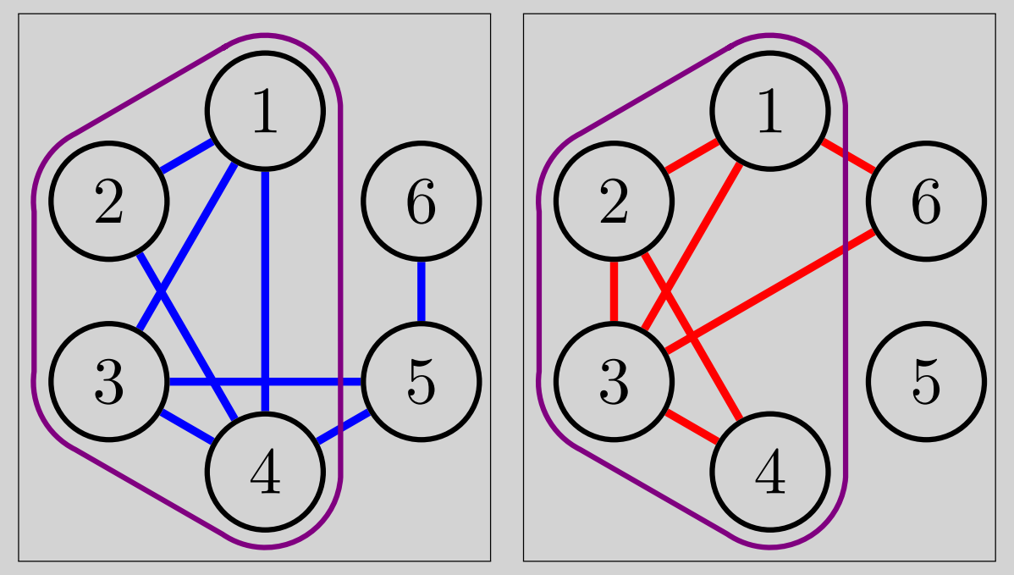
Existing methods do not scale [Jiang and Pei 2009; Li et al. 2011]

Dense Common Subgraph (DCS) problem

DCS Given relational graph set $G^{(1)} = (V, E^{(1)}), G^{(2)} = (V, E^{(2)}), \ldots,$

$$\delta_{DCS} = \max_{S \subseteq V} \min_{G^{(m)}} \frac{\#\{\text{edges induced by } S \text{ in } G^{(m)}\}}{|S|}$$

Example



S	$\delta^{(1)}$	$\delta^{(2)}$	$\delta^{\min}$
123456	1.33	1.17	1.17
12345	1.4	1.0	1.0
12346	1.0	1.4	1.0
1234	1.25	1.25	**1.25**
⋮	⋮	⋮	⋮

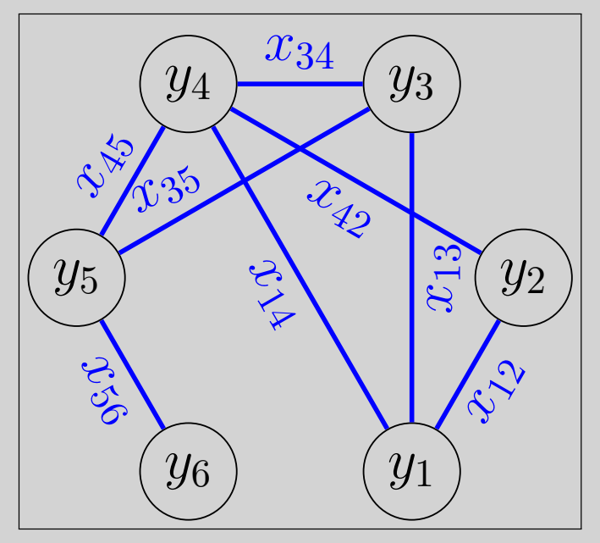
$\delta_{DCS} = 1.25$

Background: Charikar's algorithm for Dense Subgraph

If single graph, **DCS** is equivalent to Dense Subgraph problem,

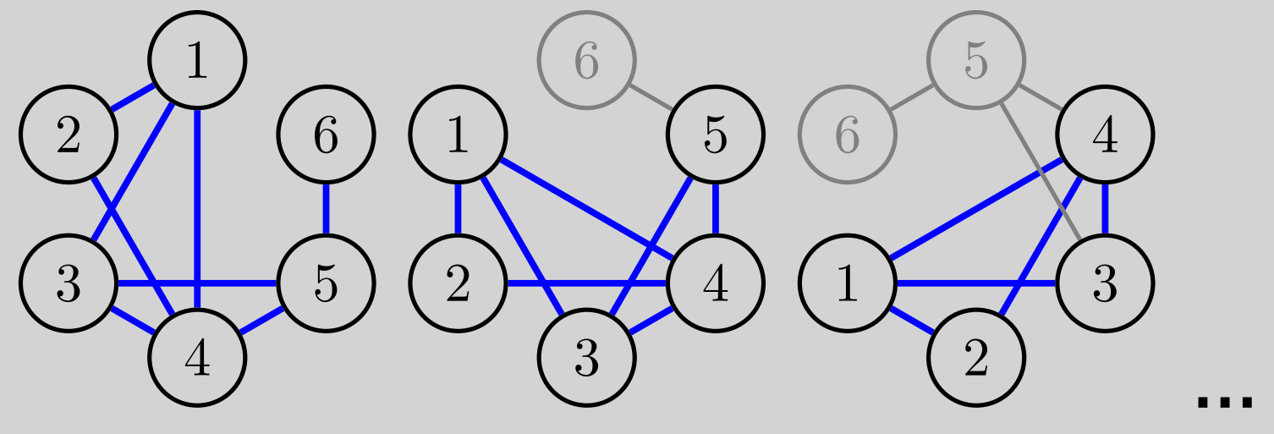
$$\delta = \max_{S \subseteq V} \frac{|E(S)|}{|S|}$$

- Exact solution [Goldberg 1984; Charikar 2000]



$$\delta = \max_{x,y} \sum_{ij \in E} x_{ij}$$
$$\text{s.t.} \sum_i y_i \leq 1$$
$$x_{ij} \leq \min(y_i, y_j) \; \forall ij \in E$$
$$x_{ij} \geq 0, y_i \geq 0$$

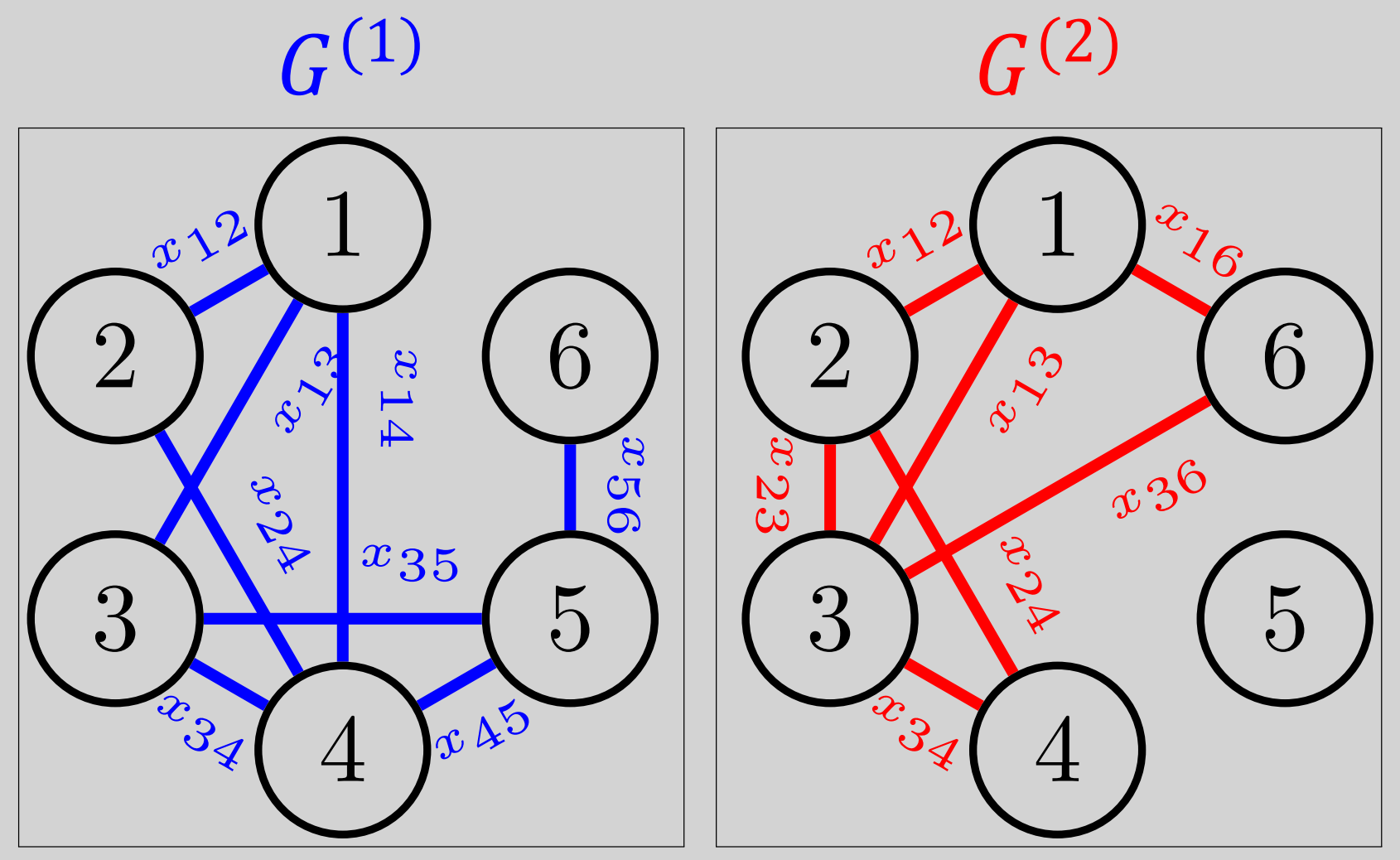
- Greedy 2-approximation: `remove least degree node and return subgraph with highest average degree`



δ: 1.33 **1.40** 1.25 ...

$$\boxed{\delta_{opt} \leq 2\delta_{greedy}}$$

DCS_LP Linear Program for Dense Common Subgraph



DCS_LP

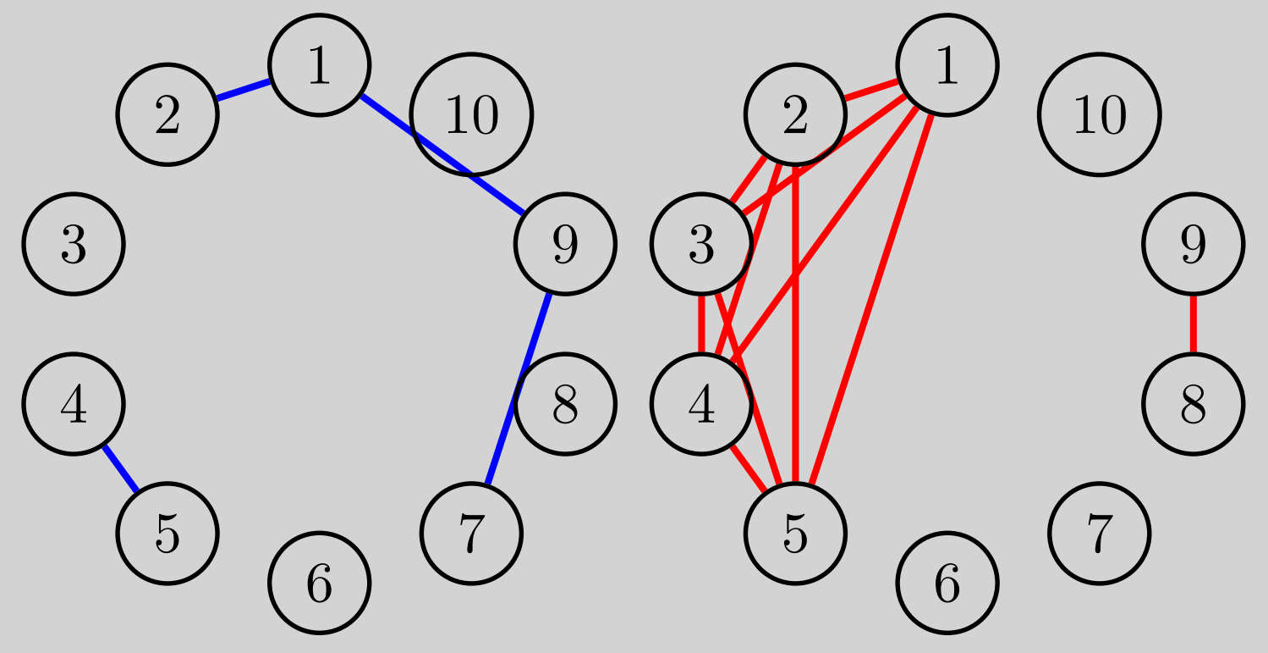
$$\max_{x,y,t} \; t$$
$$\sum_{ij \in E^{(m)}} x_{ij}^{(m)} \geq t \quad \forall G^{(m)}$$
$$x_{ij}^{(m)} \leq \min(y_i, y_j) \; \forall ij \in E^{(m)}$$
$$\sum_i y_i \leq 1$$
$$x_{ij}^{(m)} \geq 0, y_i \geq 0$$

How good is DCS_LP -- Is $t^* = \delta_{DCS}$? Can one recover optimal S_{DCS} from LP solution?

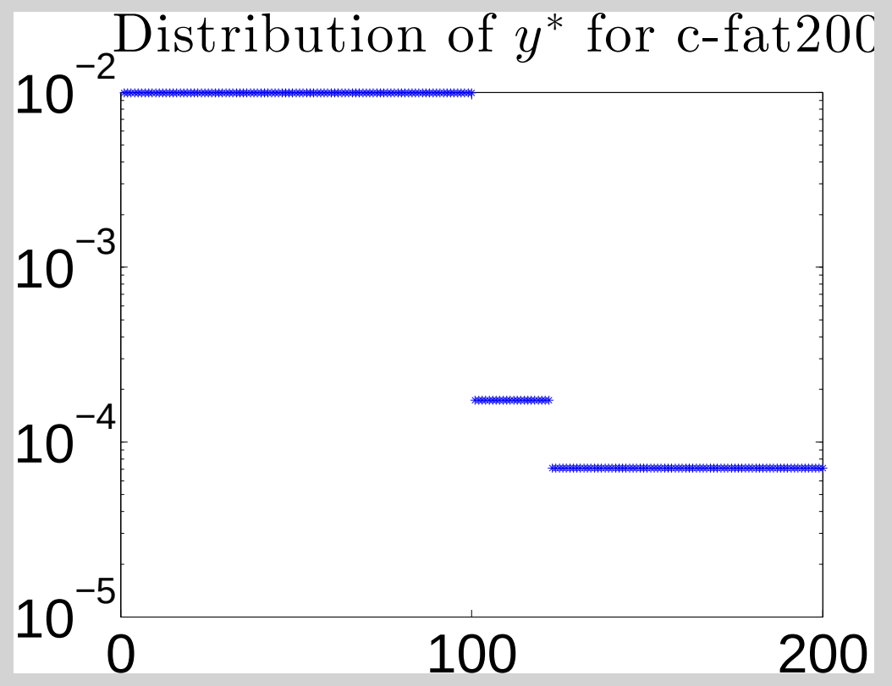
If $y^* = [\underbrace{\tfrac{1}{n}, \ldots, \tfrac{1}{n}}_{n}, 0, \ldots, 0]$, then $t^* = \delta_{DCS}$ and $S_{DCS} = \{i : y_i^* > 0\}$

No in general!

- Integrality gap $\delta_{DCS} < t^*$
- Cannot always recover S_{DCS} from LP solution



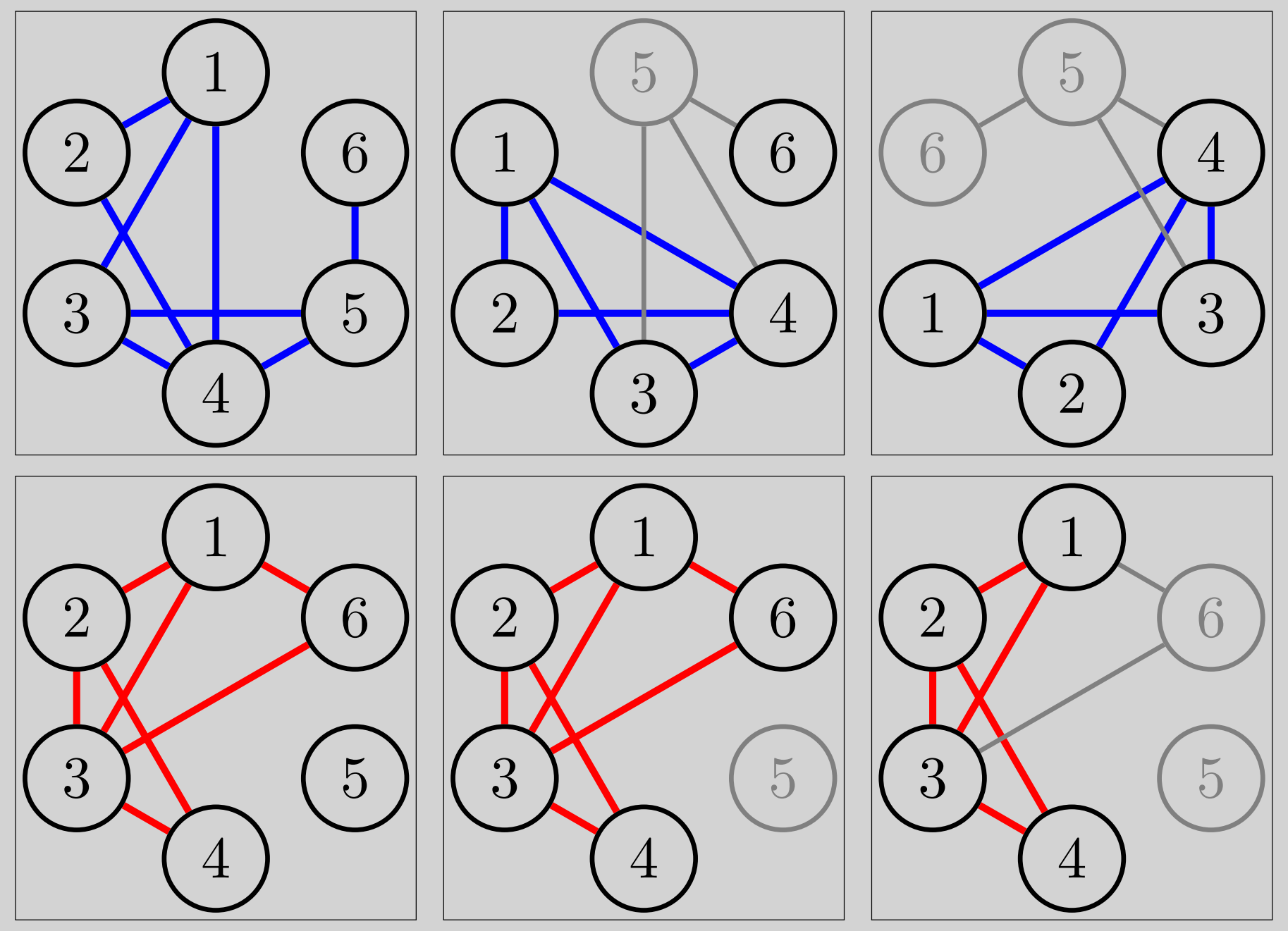
$\delta_{DCS} = 0.67, t^* = 0.7$



$S_{DCS} \neq \{i : y_i^* > 0\}$

Greedy algorithm for DCS

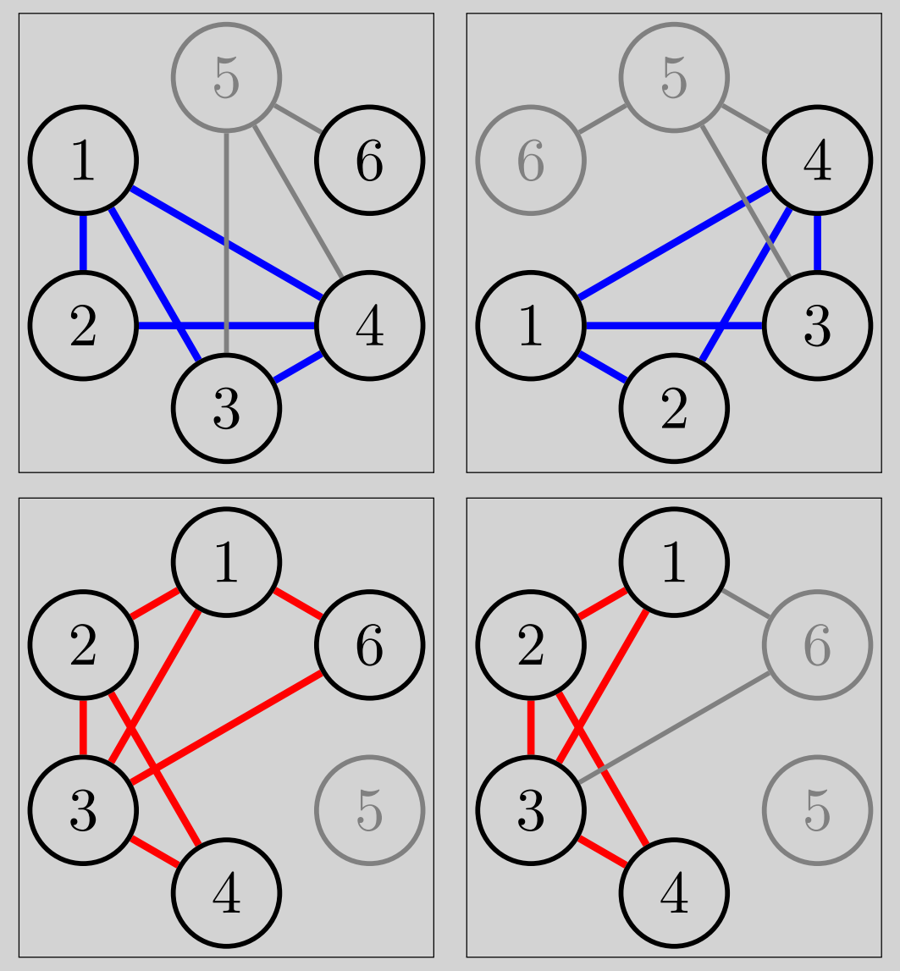
1. Choose least dense graph in relational graph set
2. Find minimum degree node in the least dense graph
3. Remove node from graph set and repeat $1 - 3$.



$\delta^{(1)}$	$\delta^{(2)}$	v_{remove}
1.33	1.17	5
1.0	1.4	6
1.25	1.25	2
1.0	0.67	4
0.5	0.5	3

$\delta_{dcs\_greedy} = 1.25$

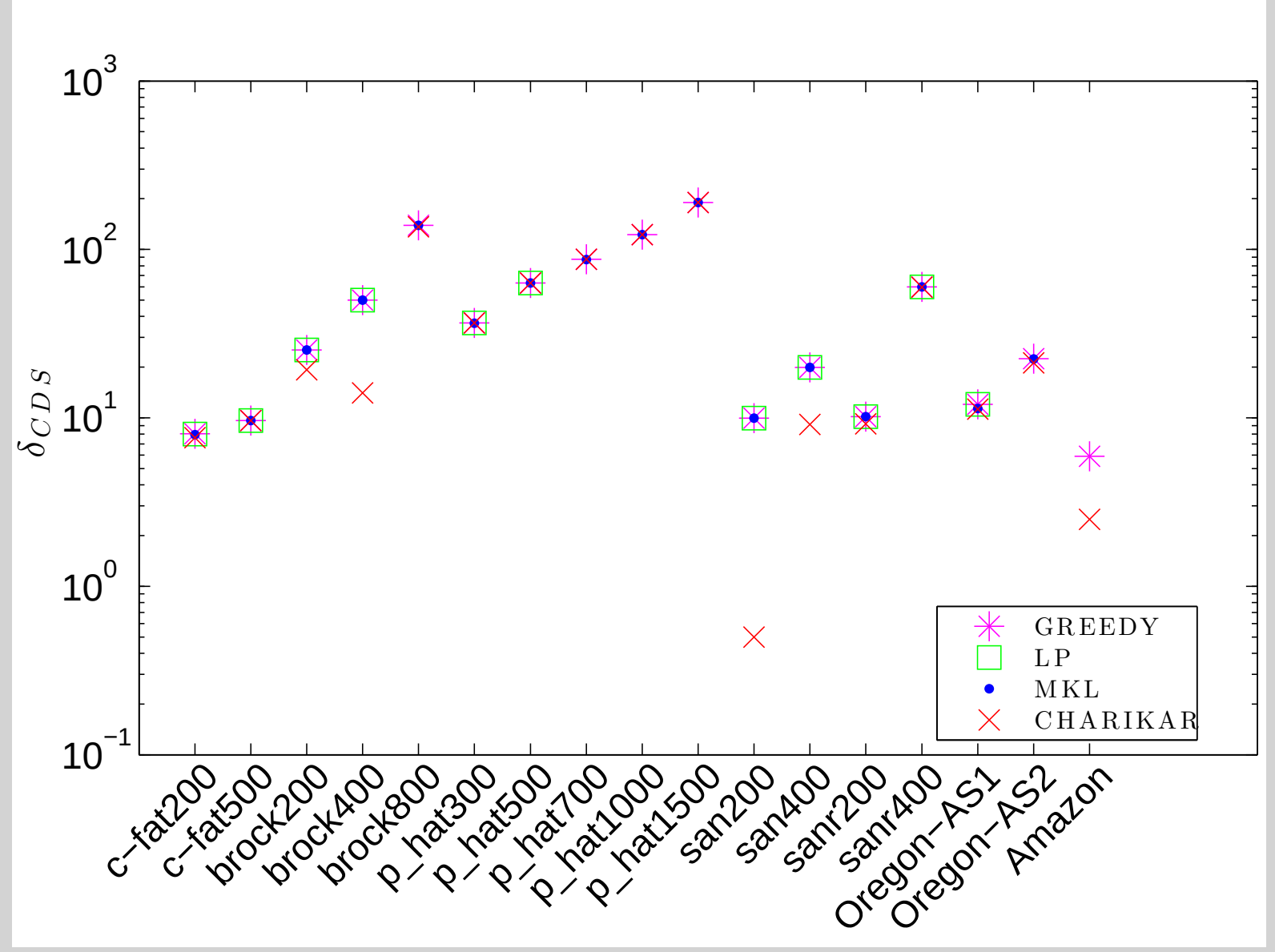
How good is DCS_GREEDY?



$\delta^{(1)}$	$\delta^{(2)}$	v_{remove}	$\underline{n}$	$\underline{d}$	$\hat{d}$	r
1.33	$\underline{1.17}$	5	5	2	3	1.8
$\underline{1.0}$	1.4	6	4	2	2	1.25
⋮	⋮	⋮	⋮	⋮	⋮	⋮

$$\delta_{DCS} \leq 2 r_{\max} \delta_{GREEDY}$$

Results on DIMACS and SNAP graph sets

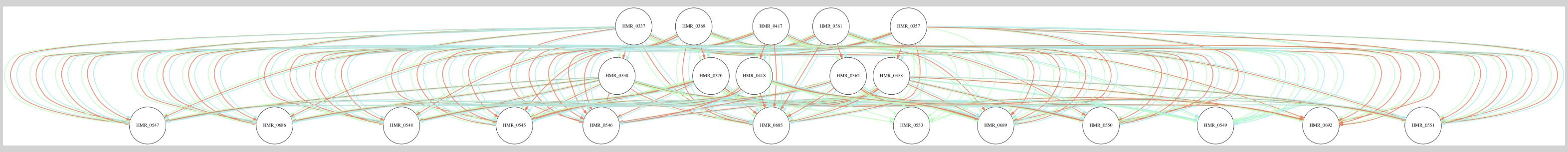


$\mathbb{G}$	n	$n_{\mathbb{G}}$	LP
c-fat200 (3)	200	13242	🟥
c-fat500 (4)	500	83416	🟩
brock200 (4)	200	29753	🟩
brock400 (4)	400	80245	🟩
brock800 (4)	800	447753	■
p_hat300 (3)	300	66251	🟩
p_hat500 (3)	500	188315	🟩
p_hat700 (3)	700	365737	■
p_hat1000 (3)	1000	738798	■
p_hat1500 (3)	1500	1701127	■
san200 (5)	200	17910	🟩
san400 (3)	400	119700	🟩
sanr200 (2)	200	8069	🟩
sanr400 (2)	400	63747	🟩
AS1 (9)	11492	203127	🟩
AS2 (9)	11806	284031	■
Amazon (3)	400727	7157921	■

- LP solution: 🟩=optimal, 🟥=sub-optimal, ■=out of time

Subnetworks in genome-scale metabolic models

- Dense common subnetworks for specific markers
- Method captures altered metabolic pathways



Summary

- Extension of Charikar's algorithm to DCS
- LP solution optimal if $y^* = \frac{1}{n}[\underbrace{1, \ldots, 1}_{n}, 0, \ldots, 0]$
- DCS_GREEDY gives graph-dependent bounds

References

- A. V. Goldberg (1984). *Finding a maximum density subgraph.* Berkeley, CA
- M. Charikar (2000). ``Greedy approximation algorithms for finding dense components in a graph''. In: *APPROX*, pp. 84–95
- D. Jiang and J. Pei (2009). *Mining frequent cross-graph quasi-cliques.* KDD '09
- W. Li et al. (2011). ``Integrative analysis of many weighted co-expression networks using tensor computation''. In: *PLoS Comp Bio* 7.6, e1001106