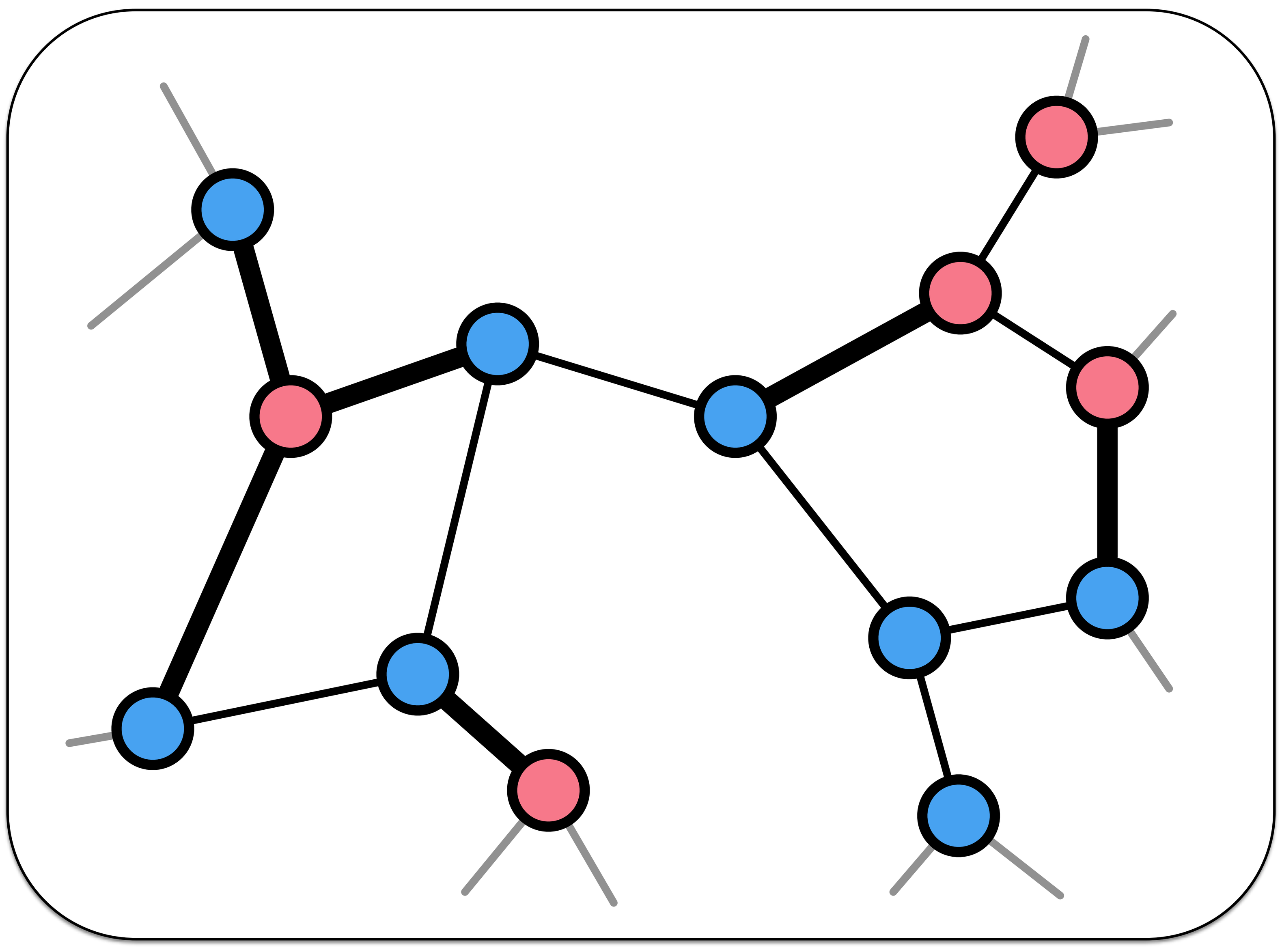
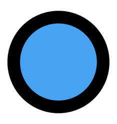
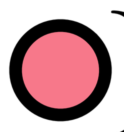
Large Cuts with Local Algorithms

arXiv:1402.2543

Juho Hirvonen
Joel Rybicki
Jukka Suomela
Aalto University

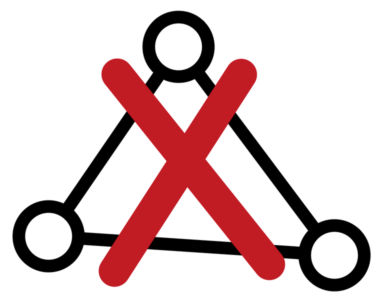
Stefan Schmid
TU Berlin & T-Labs



$c\colon V \to \{$ , $\}$

Input graph

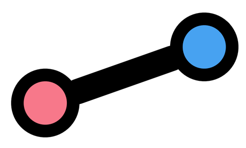
- *d*-regular
- triangle-free



Model of computing

- *n* nodes = processors
- *m* edges = communication links

Finding large cuts

- maximise number of *cut edges* 

Main result

- *one-round* randomized algorithm
- expected number of cut edges is

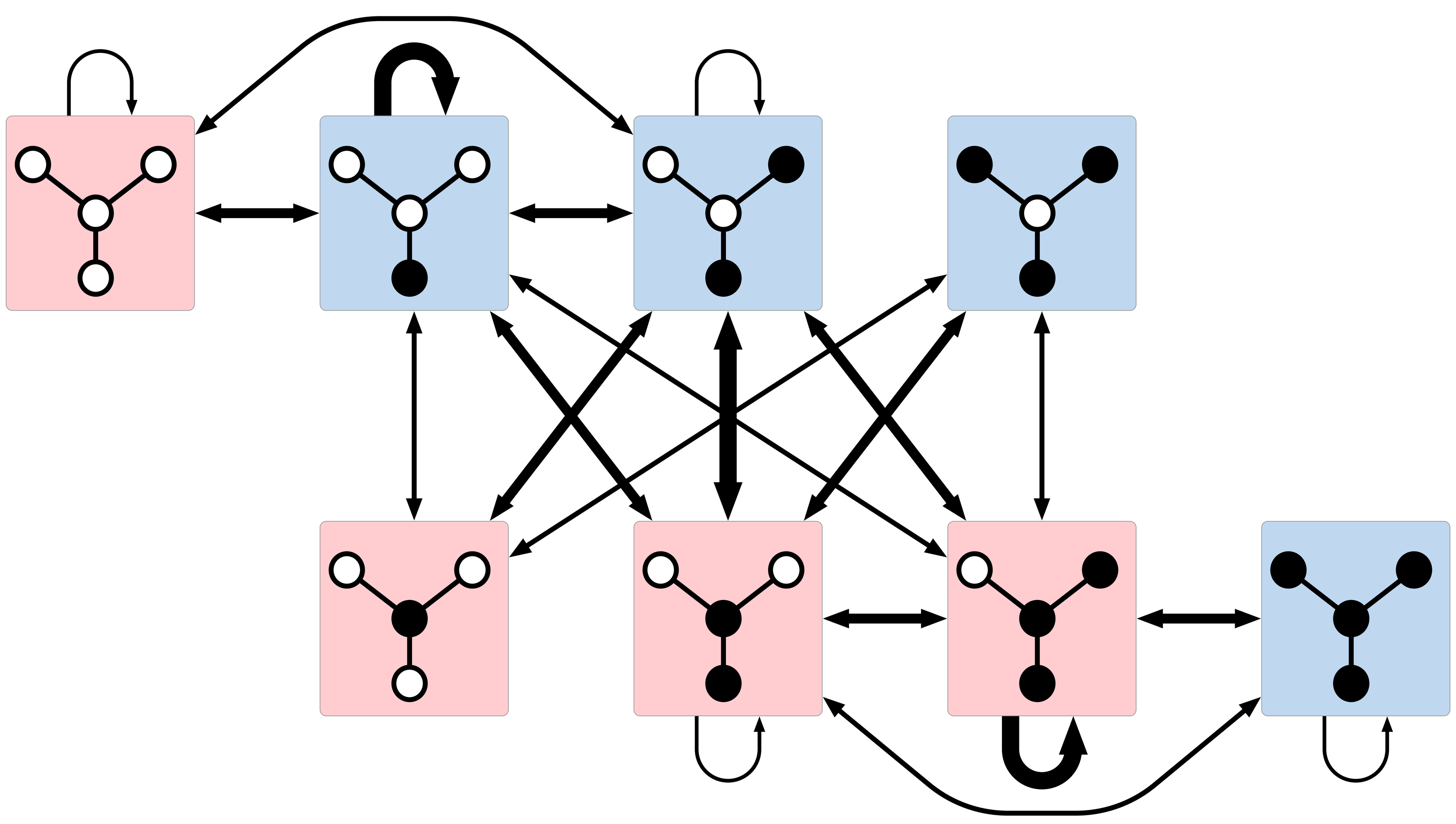
$$\left(\frac{1}{2} + \frac{0.28125}{\sqrt{d}}\right) m$$

Lower bound

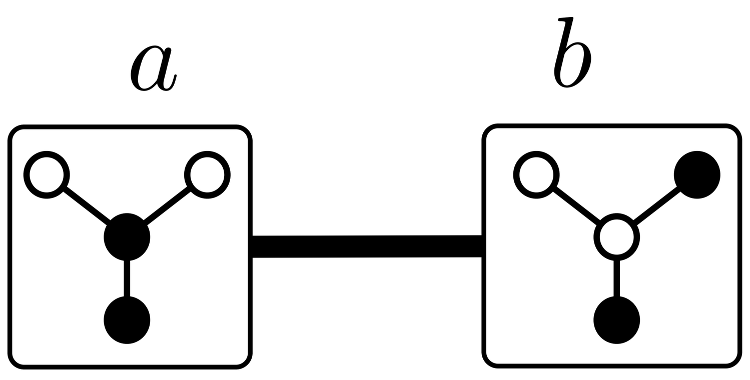
- there exist graphs where largest cut has size

$$\approx \left(\frac{1}{2} + \frac{1}{\sqrt{d}}\right) m$$

Computational Algorithm Design



Neighbourhood graphs
heavy cut = good local algorithm



$(x_a \vee x_b)\colon w(a,b)$
$(\overline{x}_a \vee \overline{x}_b)\colon w(a,b)$

Weighted MAX-SAT
find *optimal* algorithms
for any fixed *d*