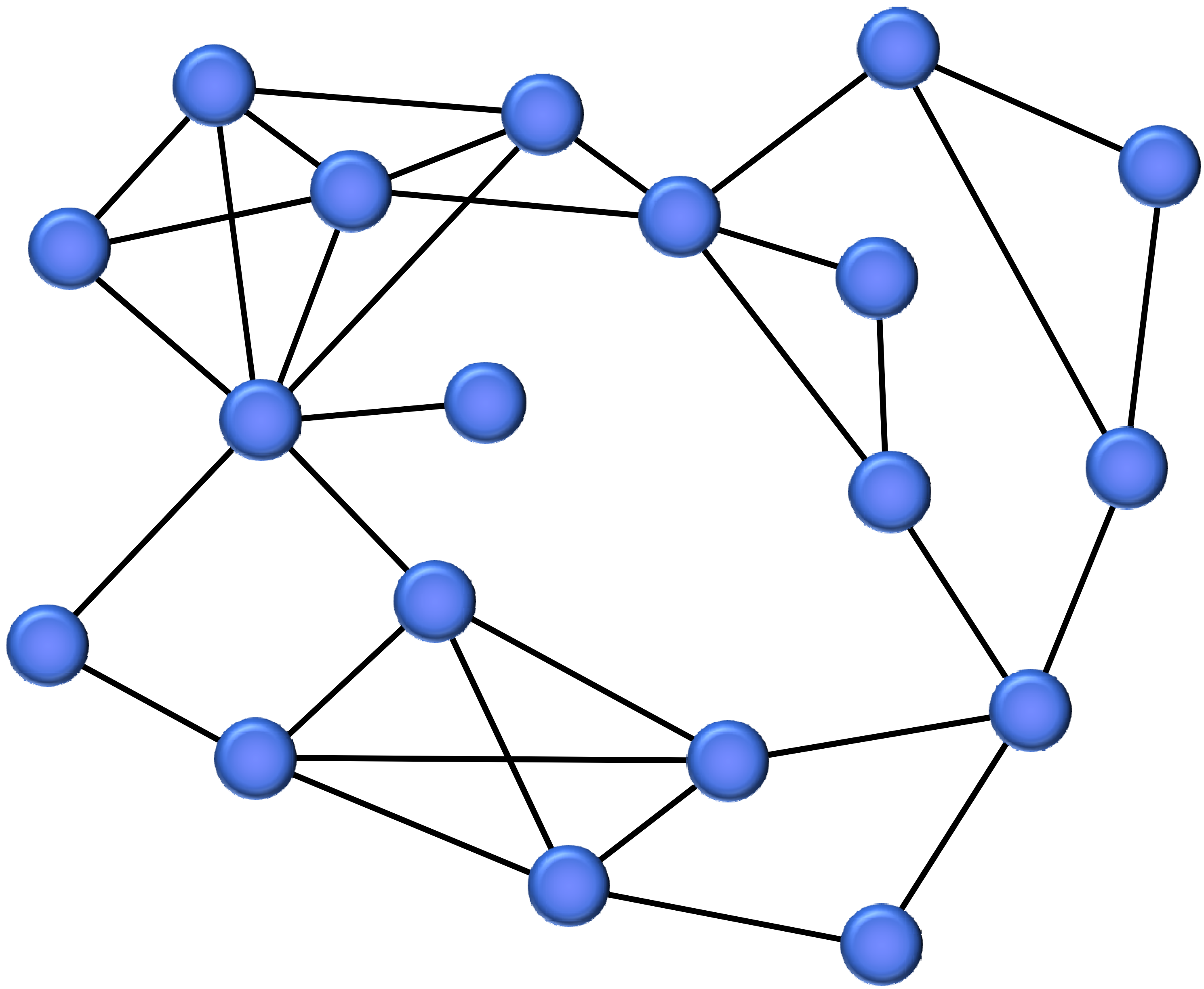
On Local Fixing

Michael König
Roger Wattenhofer

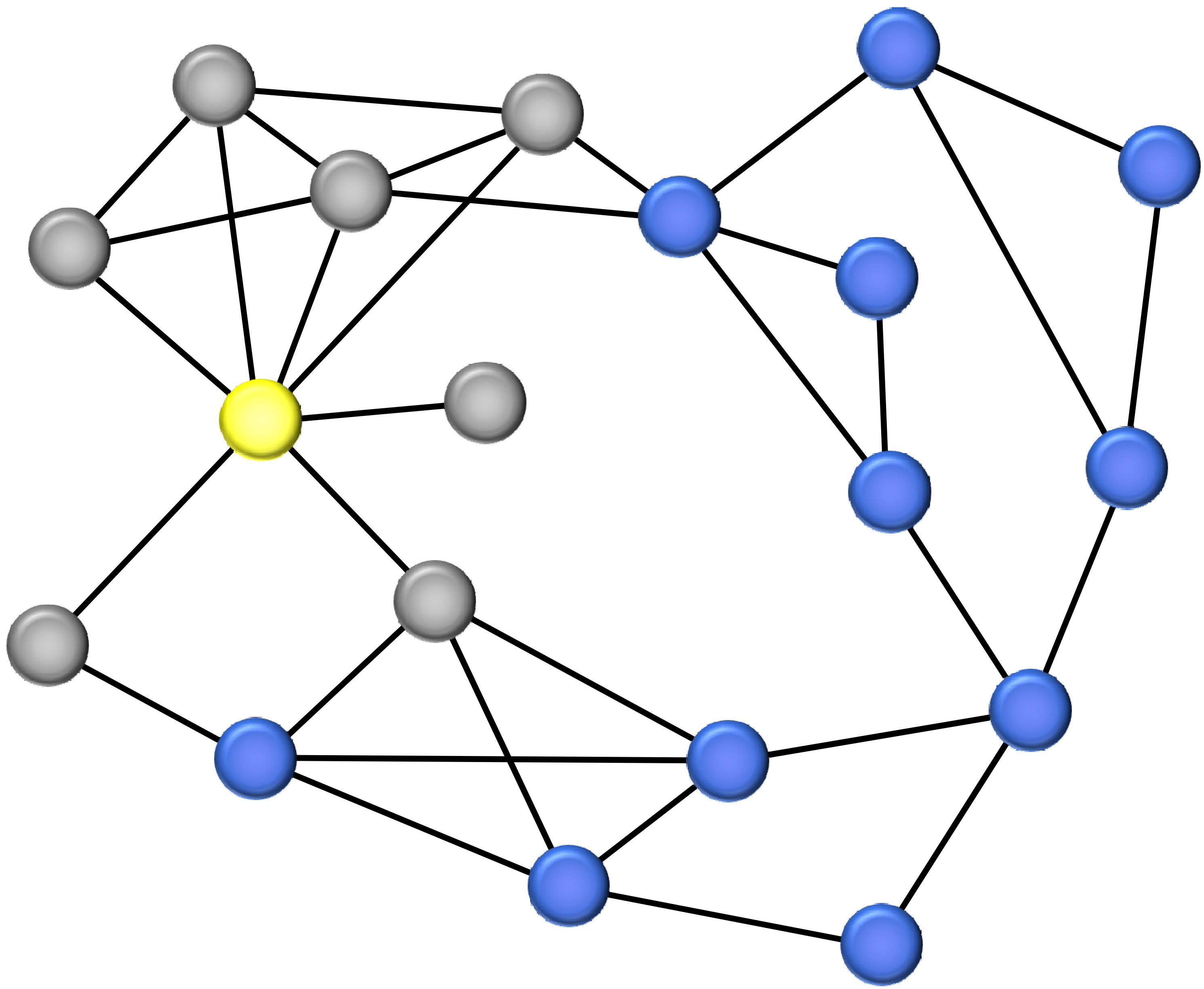
Motivation

Motivation

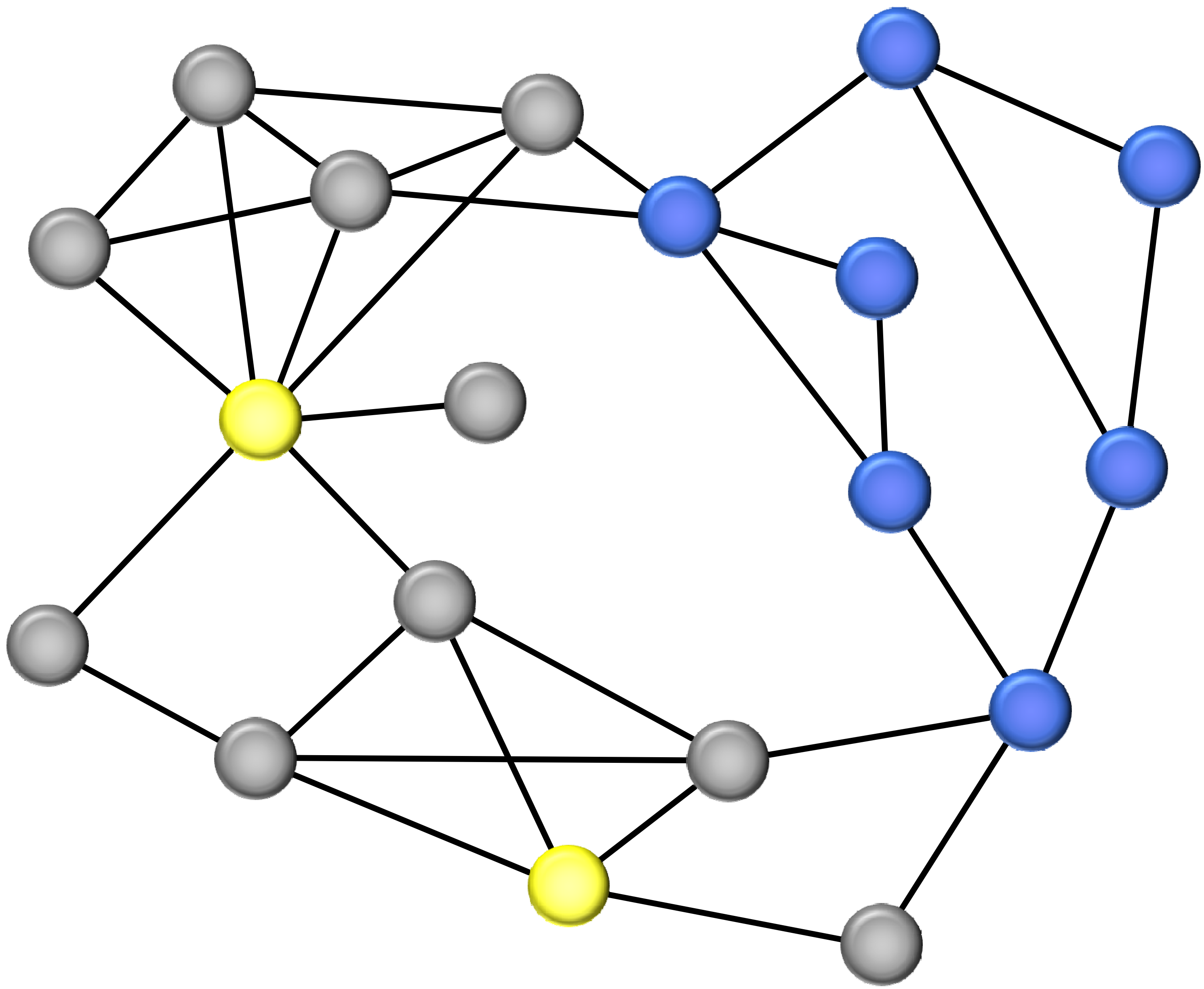
Motivation by Example



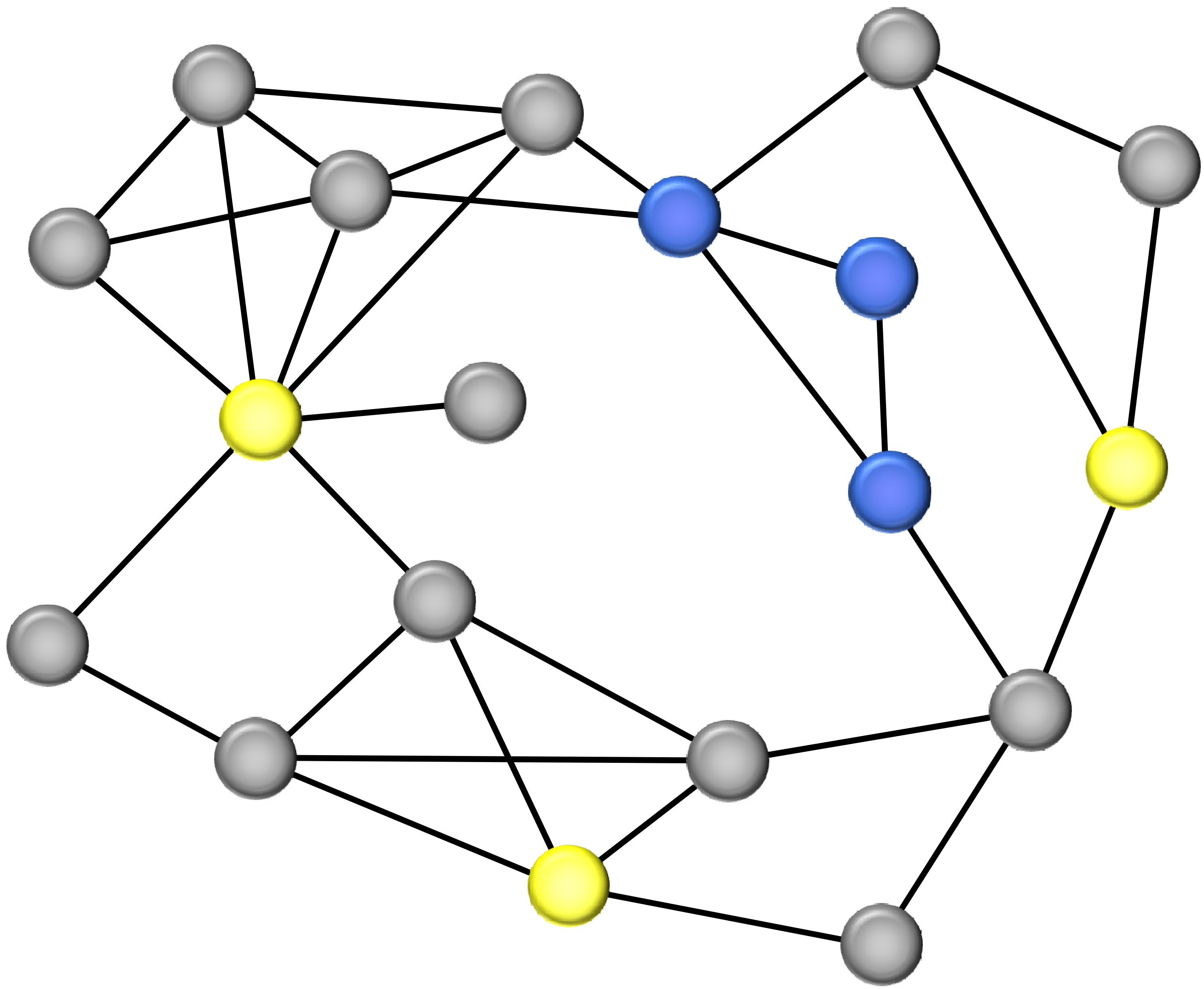
Motivation by Example



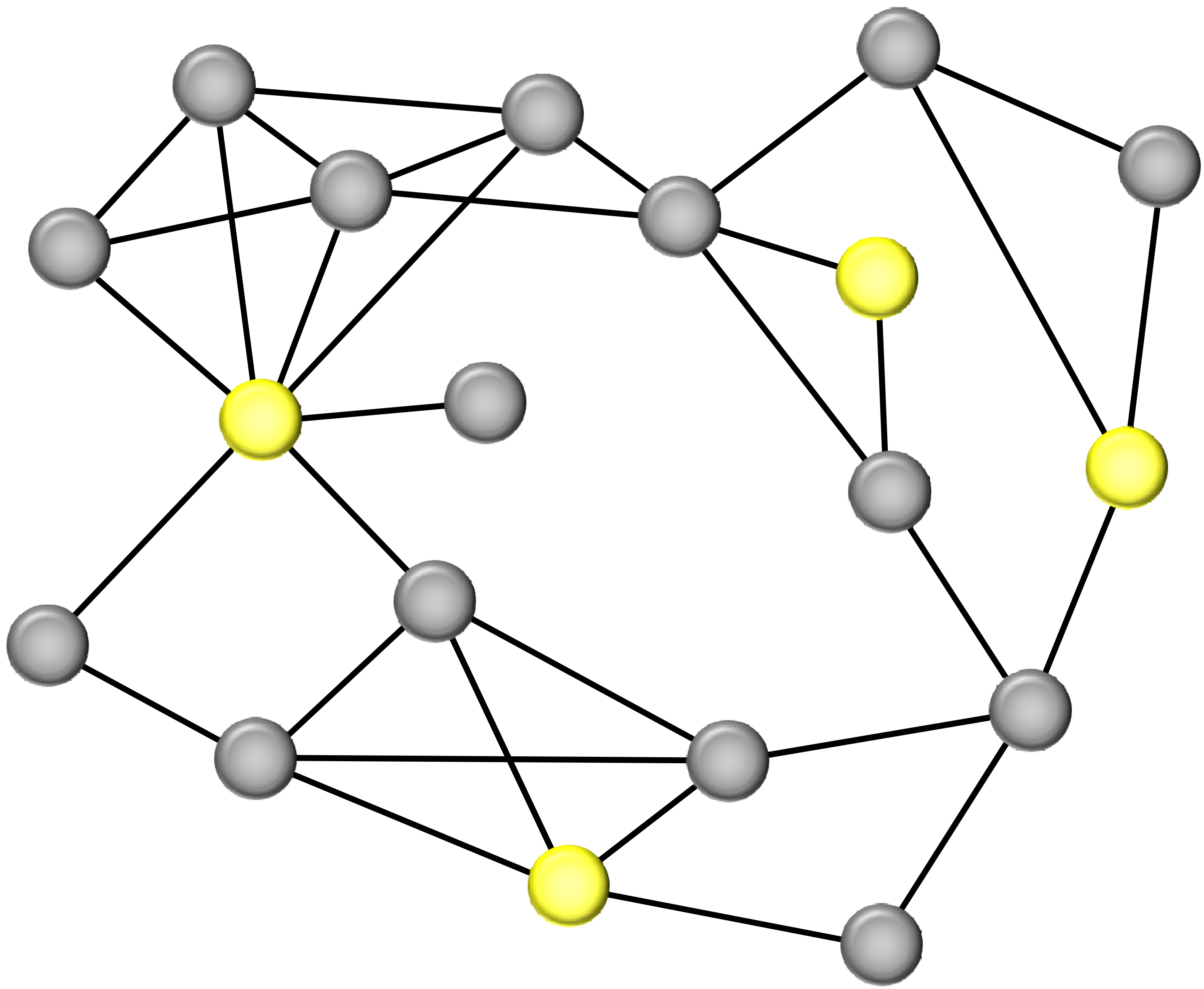
Motivation by Example



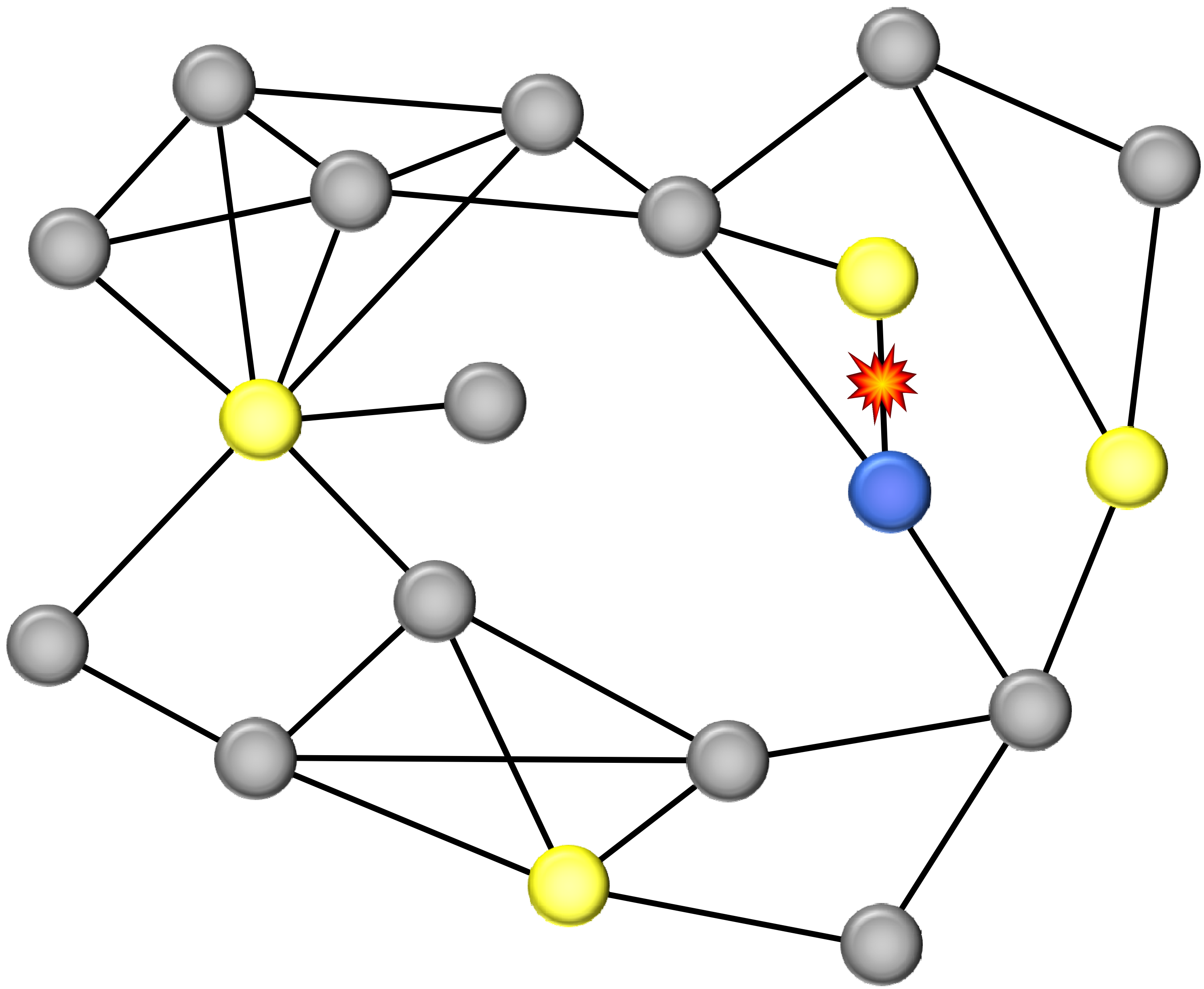
Motivation by Example



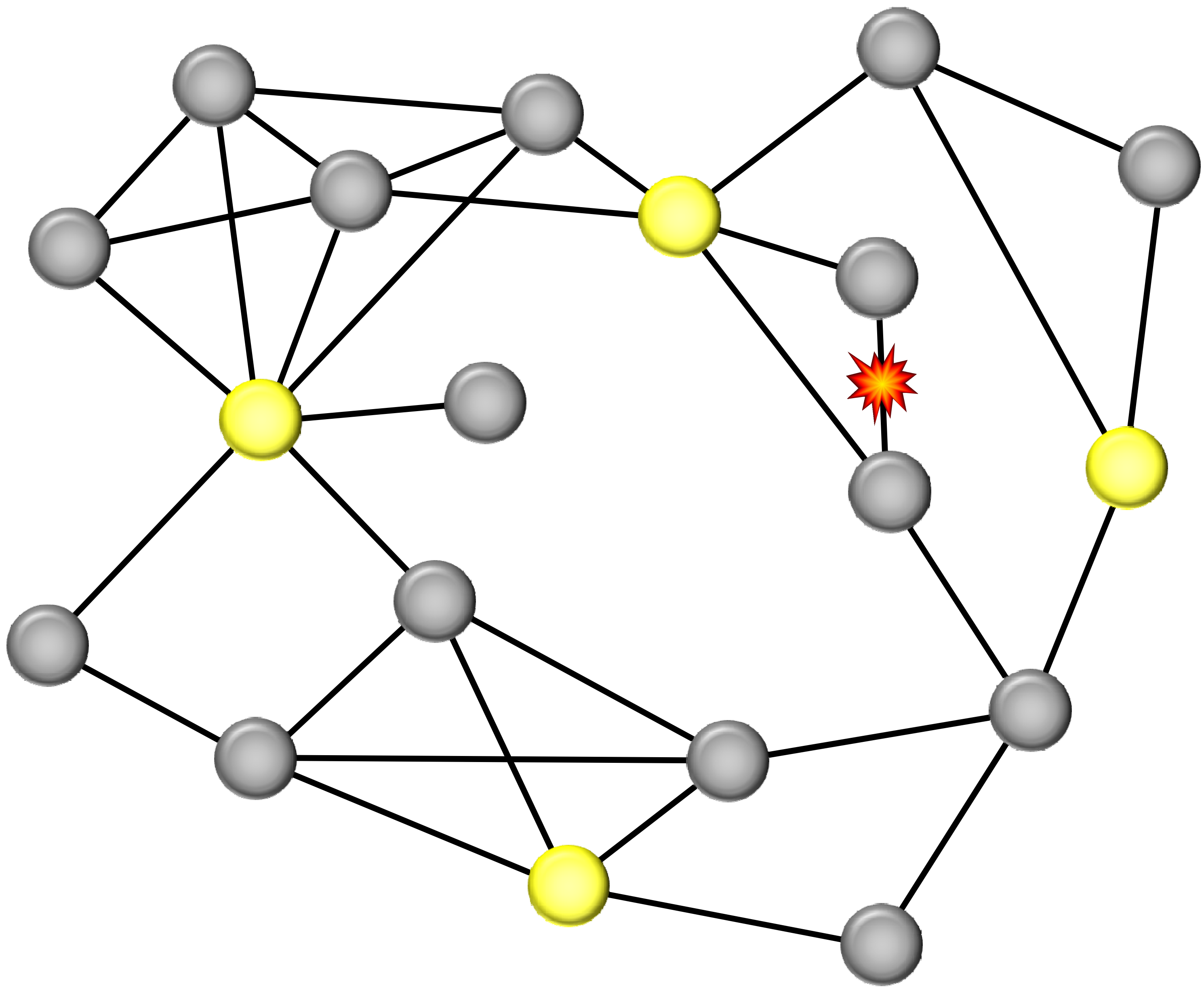
Motivation by Example



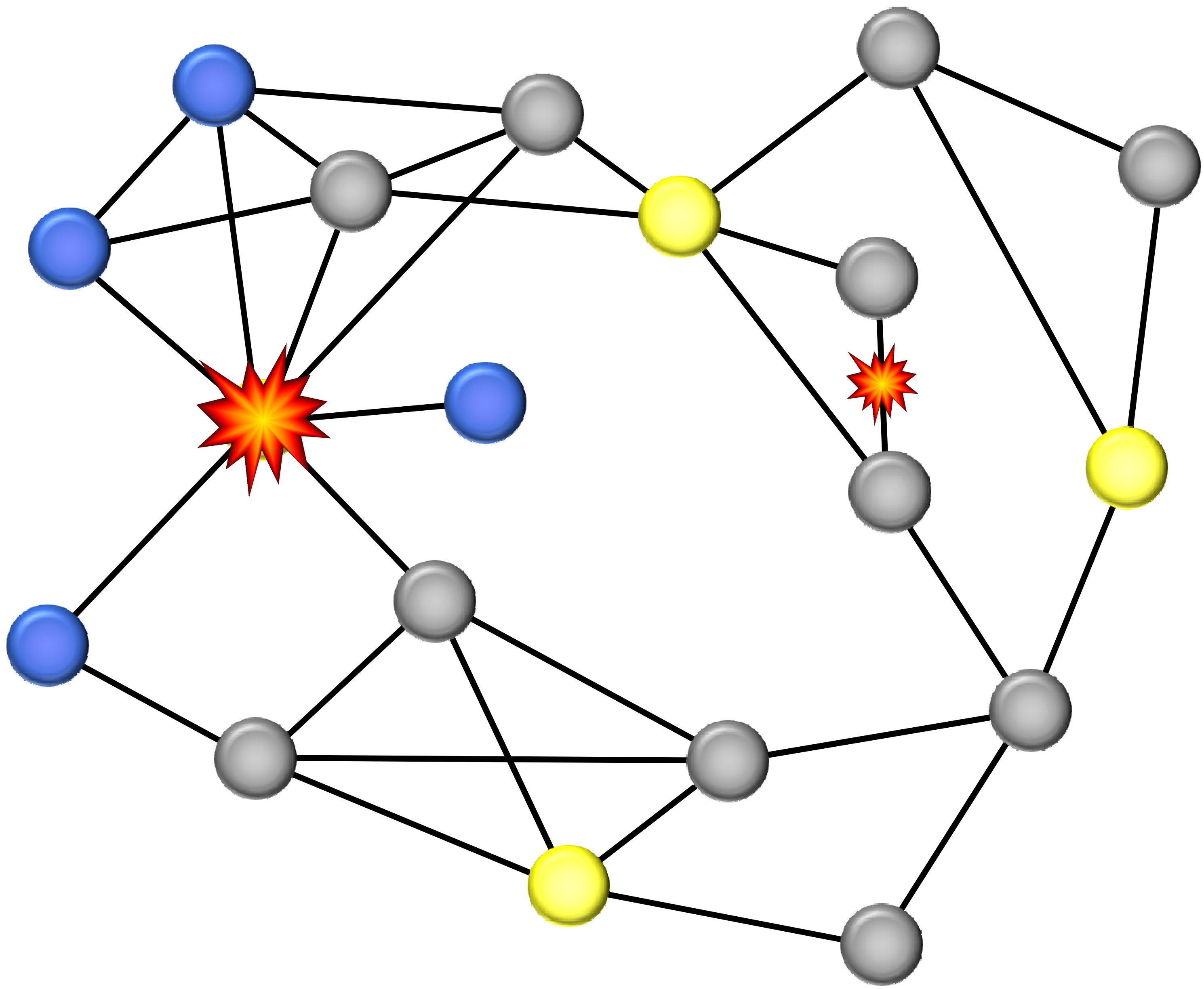
Motivation by Example



Motivation by Example

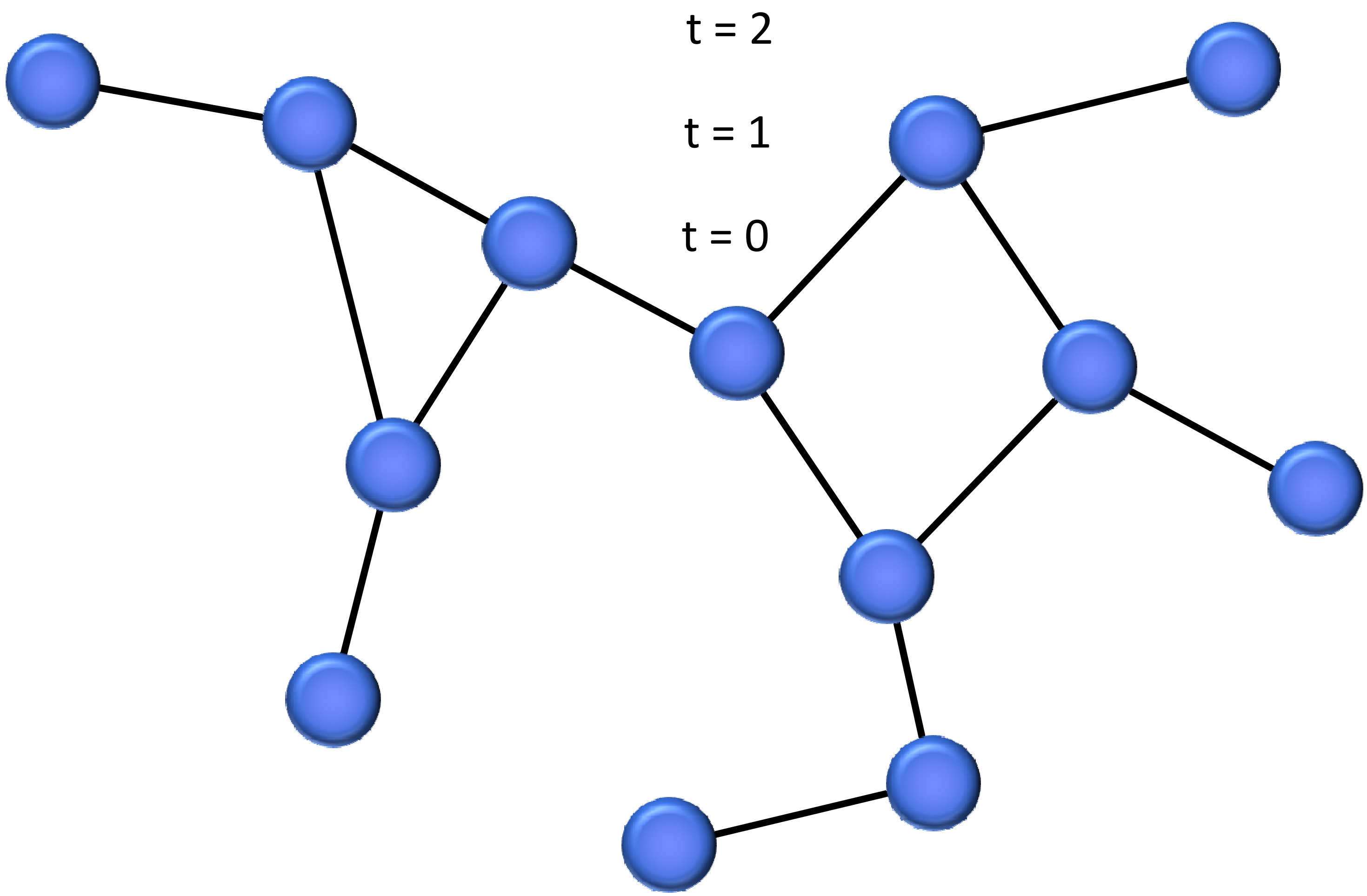


Motivation by Example



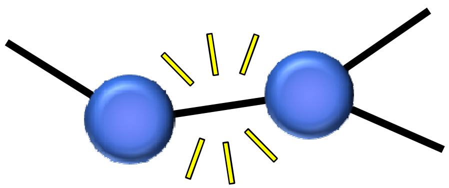
Model

- Neighborhood model
  - Unbounded message sizes
  - Synchronous rounds

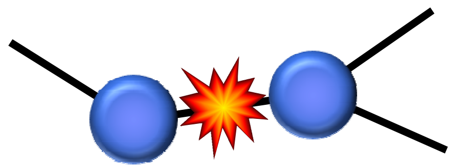


Model (cont.)

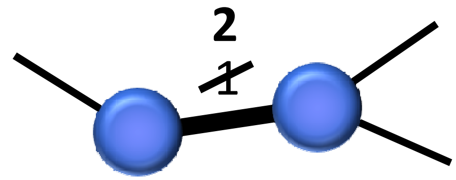
- We say a problem can be ***fixed locally*** if any solution can be fixed within O(1) rounds after a graph change.



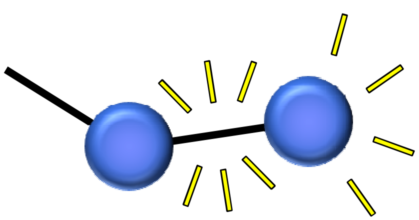
(+e)
Edge Insertion



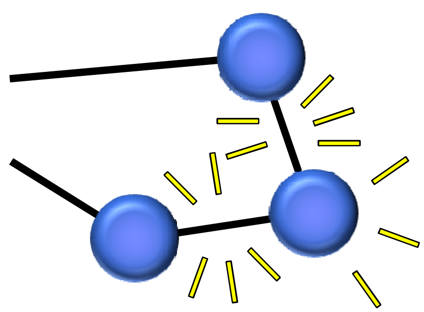
(-e)
Edge Deletion



(w → w')
Weight Change

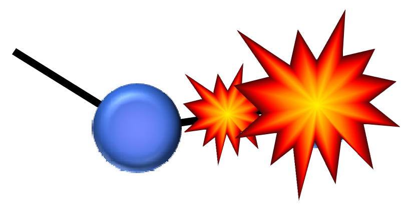


($+v_1$)

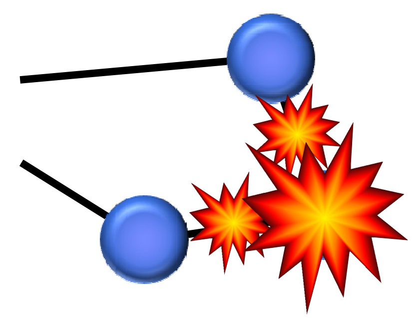


($+v_*$)

Node Insertion



($-v_1$)

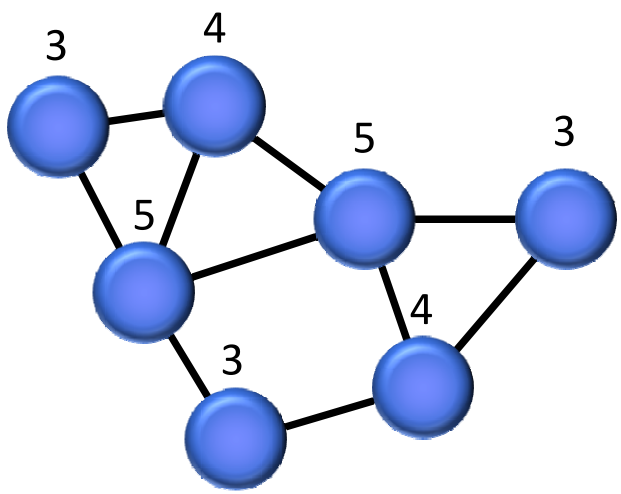


($-v_*$)

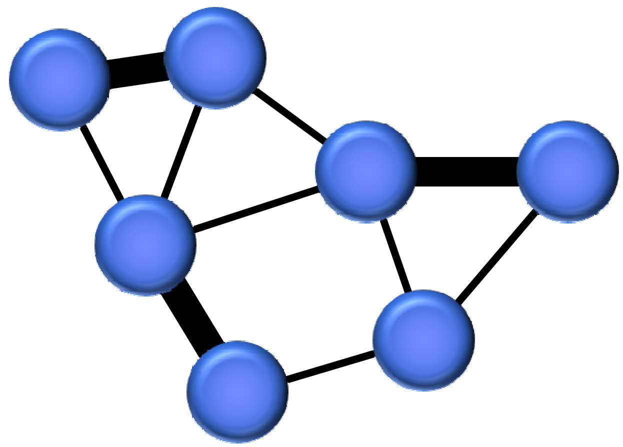
Node Deletion

Computation vs. Fixing

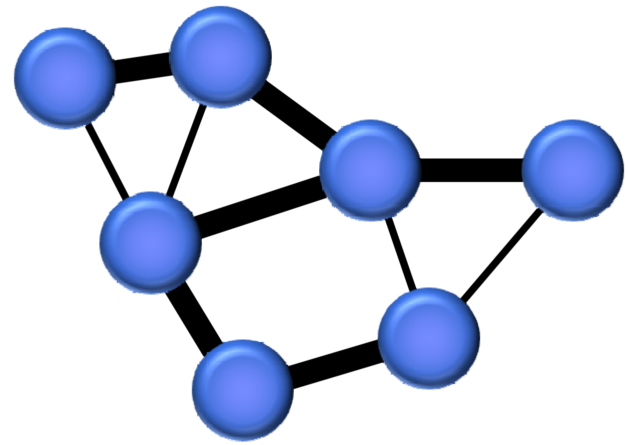
- Computing solutions is very well-studied
- Different "complexity classes" have been defined:



Γ_1-Count
"local"



Maximal Matching
"polylog"



Spanning Tree
"global"

- Previous work on local fixing
  - 1995: Maximal Independent Sets (MIS), by Kutten and Peleg
  - 2007: O(1)-Maximum Weighted Matchings, by Lotker, Patt-Shamir and Rosén

Problems

	LBound	UBound
Γ_1-Count	Ω(1)	O(1)
o(n)-MDS	Ω(1)	O(1)
MIS	Ω(√log(n))	O(log(n))
O(1)-MWM	Ω(√log(n))	O(log(n))
MM	Ω(√log(n))	O(log(n))
2-MVC	Ω(√log(n))	O(log(n))
$\Gamma_{\log(n)}$-Count	Ω(log(n))	O(log(n))
ST	Ω(D)	O(D)
MST	Ω(D)	O(D)
SPT	Ω(D)	O(D)
Flow	Ω(D)	O(D)
Leader	Ω(D)	O(D)
Count	Ω(D)	O(D)

Problems

	LBound	UBound	+e	-e	$w \to w'$	$+v_1$	$-v_1$	$+v_*$	$-v_*$
Γ_1-Count	$\Omega(1)$	$O(1)$	✓	✓		✓	✓	✓	✓
o(n)-MDS	$\Omega(1)$	$O(1)$	✓	✓		✓	✓	✓	✓
MIS	$\Omega(\sqrt{\log(n)})$	$O(\log(n))$	✓	✓		✓	✓	✓	✓
O(1)-MWM	$\Omega(\sqrt{\log(n)})$	$O(\log(n))$	✓	✓	✓	✓	✓	✓	✓
MM	$\Omega(\sqrt{\log(n)})$	$O(\log(n))$	✓	✓		✓	✓	✓	✓
2-MVC	$\Omega(\sqrt{\log(n)})$	$O(\log(n))$	✓	✓		✓	✓	✓	✓
$\Gamma_{\log(n)}$-Count	$\Omega(\log(n))$	$O(\log(n))$	✓	✓		✓	✓	✓	✓
ST	$\Omega(D)$	$O(D)$	✓	✓		✓	✓	✓	✓
MST	$\Omega(D)$	$O(D)$	✓	✓	✓	✓	✓	✓	✓
SPT	$\Omega(D)$	$O(D)$	✓	✓	✓	✓	✓	✓	✓
Flow	$\Omega(D)$	$O(D)$	✓	✓	✓	✓	✓	✓	✓
Leader	$\Omega(D)$	$O(D)$	✓	✓		✓	✓	✓	✓
Count	$\Omega(D)$	$O(D)$	✓	✓		✓	✓	✓	✓

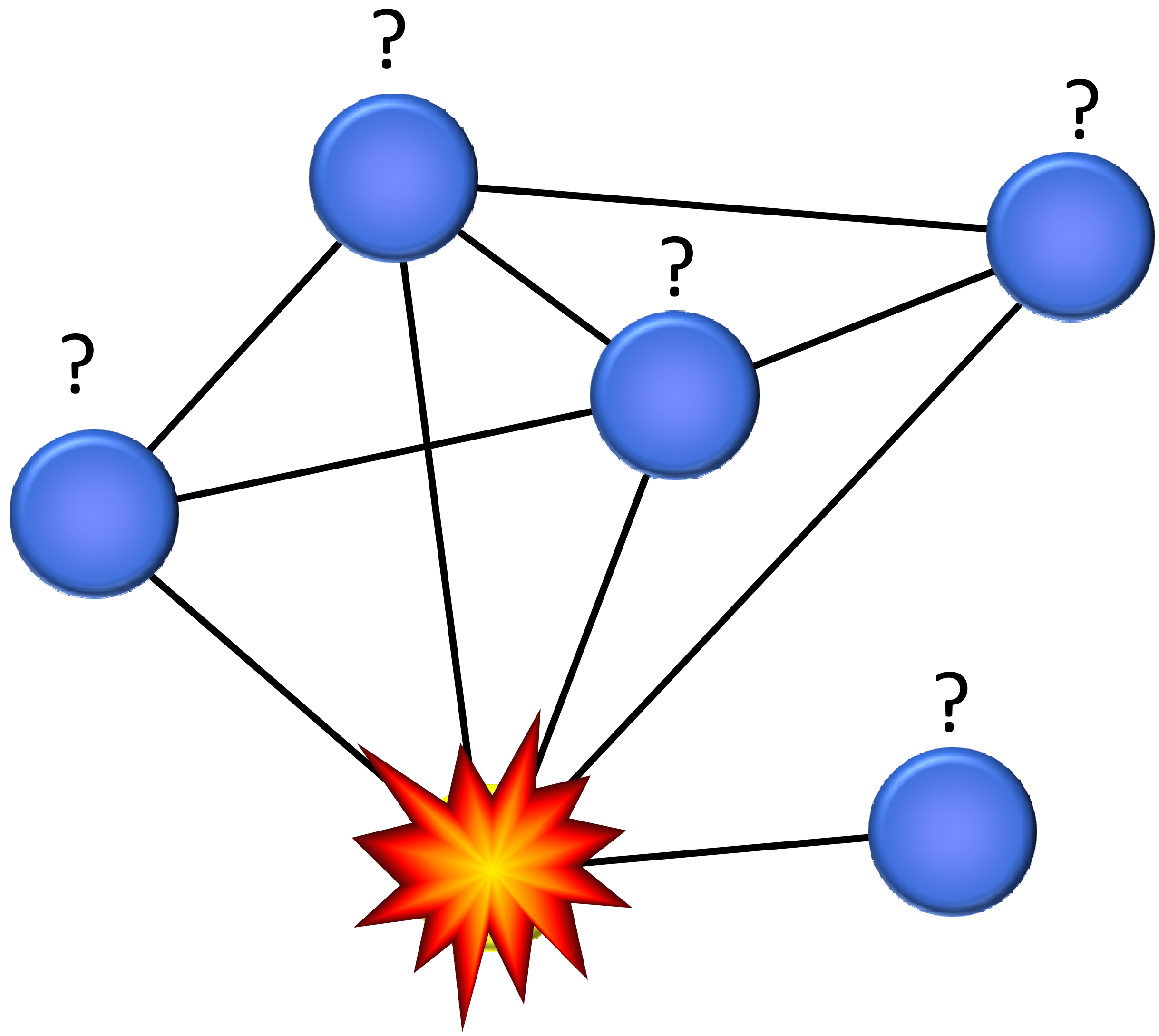
Problems

	LBound	UBound	+e	-e	$w \to w'$	$+v_1$	$-v_1$	$+v_*$	$-v_*$
Γ_1-Count	$\Omega(1)$	$O(1)$	✓	✗		✓	✗	✓	✗
$o(n)$-MDS	$\Omega(1)$	$O(1)$	✗	✓		✗	✓	✗	✓
MIS	$\Omega(\sqrt{\log(n)})$	$O(\log(n))$	✓	✗		✓	✗	✓	✗
$O(1)$-MWM	$\Omega(\sqrt{\log(n)})$	$O(\log(n))$	✗	✓	✗	✗	✓	✗	✓
MM	$\Omega(\sqrt{\log(n)})$	$O(\log(n))$	✓	✗		✓	✗	✓	✗
2-MVC	$\Omega(\sqrt{\log(n)})$	$O(\log(n))$	✗	✓		✗	✓	✗	✓
$\Gamma_{\log(n)}$-Count	$\Omega(\log(n))$	$O(\log(n))$	✓	✗		✓	✗	✓	✗
ST	$\Omega(D)$	$O(D)$	✓	✗		✓	✗	✓	✗
MST	$\Omega(D)$	$O(D)$	✗	✓	✗	✗	✓	✗	✓
SPT	$\Omega(D)$	$O(D)$	✓	✗	✓	✓	✗	✓	✗
Flow	$\Omega(D)$	$O(D)$	✗	✓	✗	✗	✓	✗	✓
Leader	$\Omega(D)$	$O(D)$	✓	✗		✓	✗	✓	✗
Count	$\Omega(D)$	$O(D)$	✗	✓		✗	✓	✗	✓

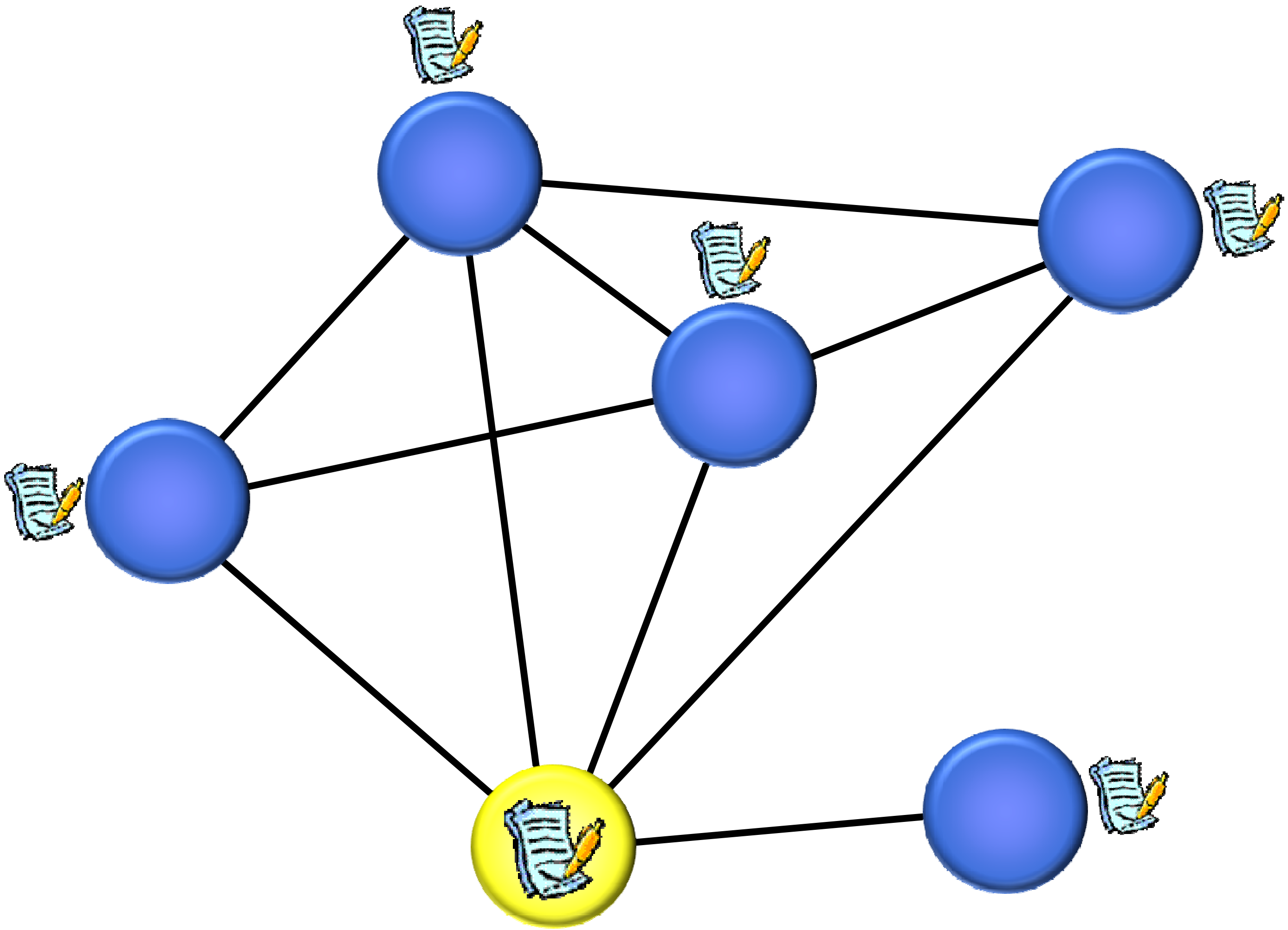
Results

	LBound	UBound	+e	-e	$w \rightarrow w'$	$+v_1$	$-v_1$	$+v_*$	$-v_*$
Γ_1-Count	$\Omega(1)$	$O(1)$	✓	✓		✓	✓	✓	✓
$o(n)$-MDS	$\Omega(1)$	$O(1)$	✗	✗		✗	✗	✗	✗
MIS	$\Omega(\sqrt{\log(n)})$	$O(\log(n))$	✓	✓		✓	✓	✓	✓
$O(1)$-MWM	$\Omega(\sqrt{\log(n)})$	$O(\log(n))$	✓	✓	✓	✓	✓	✓	✓
MM	$\Omega(\sqrt{\log(n)})$	$O(\log(n))$	✓	✓		✓	✓	✓	✓
2-MVC	$\Omega(\sqrt{\log(n)})$	$O(\log(n))$	✓	✓		✓	✓	✓	✓
$\Gamma_{\log(n)}$-Count	$\Omega(\log(n))$	$O(\log(n))$	✗	✗		✗	✗	✗	✗
ST	$\Omega(D)$	$O(D)$	✓	✗		✓	✓	✓	✗
MST	$\Omega(D)$	$O(D)$	✗	✗	✗	✓	✓	✗	✗
SPT	$\Omega(D)$	$O(D)$	✗	✗	✗	✓	✓	✗	✗
Flow	$\Omega(D)$	$O(D)$	✗	✗	✗	✓	✓	✗	✗
Leader	$\Omega(D)$	$O(D)$	✓	✓		✓	✓	✓	✓
Count	$\Omega(D)$	$O(D)$	✓	✓		✗	✗	✗	✗

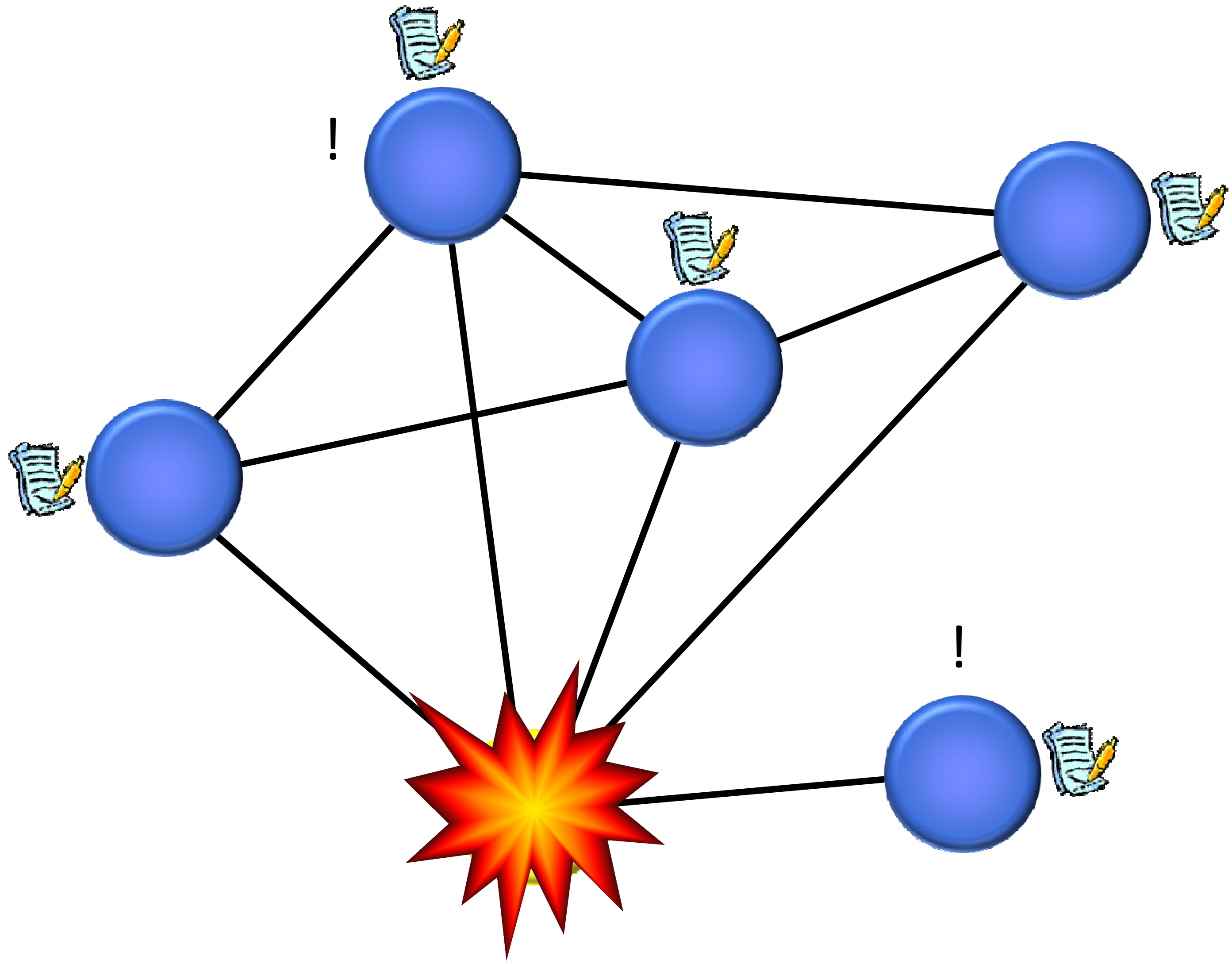
Maximal Independent Set (MIS)



“Last Will”: during settling



“Last Will”: in action



o(n)-Minimum Dominating Set

- We know algorithms which compute a o(n)-MDS in constant time [Kuhn et al., 2005]
- But we can construct o(n)-MDS solutions which cannot be fixed locally!

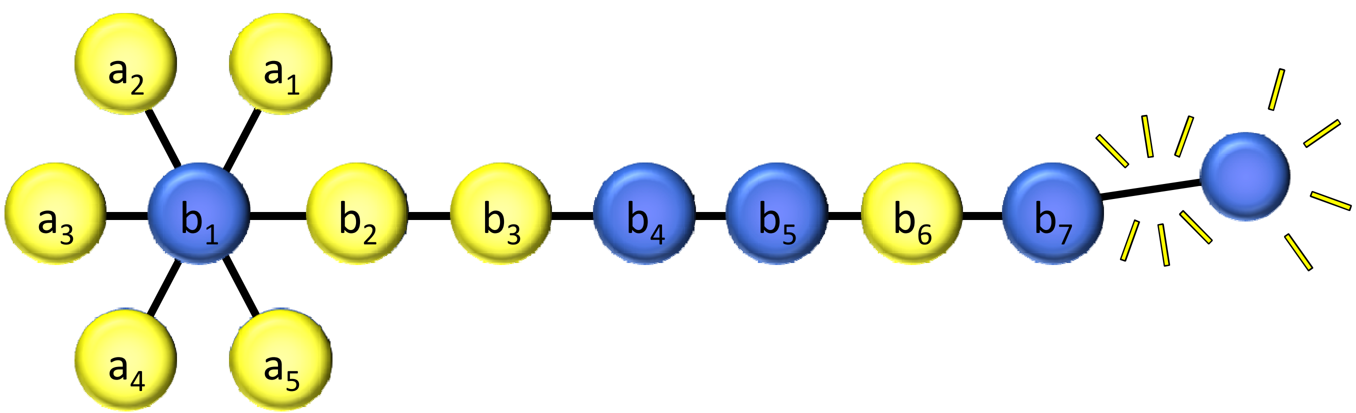
Recipe for a **k**-MDS which cannot be fixed within **c** steps:

$x = \lfloor (k - 1)(c + 1) \rfloor$, $y = 3c + 1$

$V = (a_1, a_2, \ldots, a_x, b_1, b_2, \ldots, b_y)$

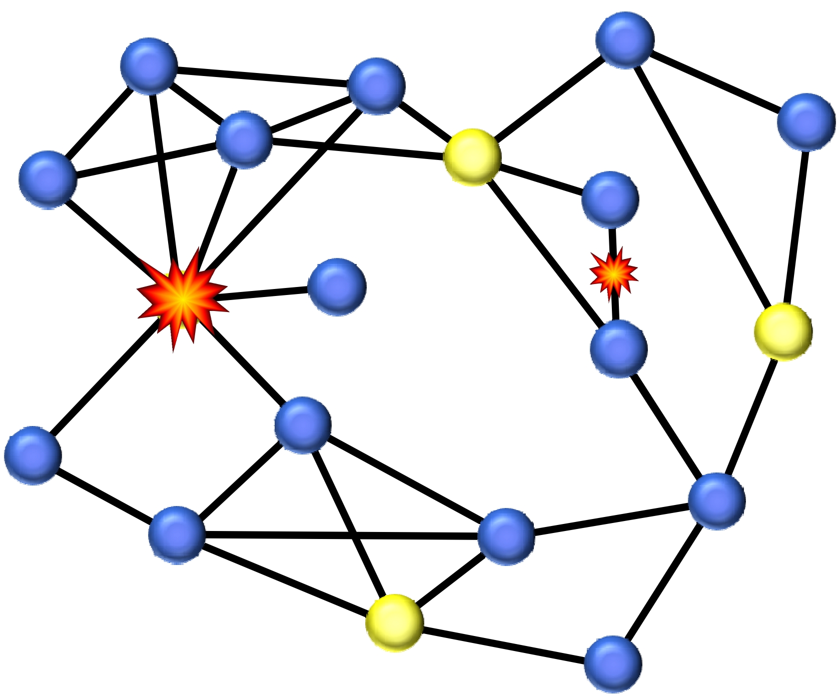
$E = \{(a_i, b_1) \mid 1 \leq i \leq x\} \cup \{(b_i, b_{i+1}) \mid 1 \leq i \leq y\}$

For k = 2.8, c = 2, x = 5 and y = 7:



Conclusion

- Traditional distributed complexity classes don't tell the whole story.
- A set of orthogonal "fixing complexity" classes may be interesting!



	LBound	UBound	+e	-e	w → w'	$+v_1$	$-v_1$	$+v_*$	$-v_*$
Γ_1-Count	$\Omega(1)$	$O(1)$	✓	✓		✓	✓	✓	✓
o(n)-MDS	$\Omega(1)$	$O(1)$	✗	✗		✗	✗	✗	✗
MIS	$\Omega(\sqrt{\log(n)})$	$O(\log(n))$	✓	✓		✓	✓	✓	✓
O(1)-MWM	$\Omega(\sqrt{\log(n)})$	$O(\log(n))$	✓	✓	✓	✓	✓	✓	✓
MM	$\Omega(\sqrt{\log(n)})$	$O(\log(n))$	✓	✓		✓	✓	✓	✓
2-MVC	$\Omega(\sqrt{\log(n)})$	$O(\log(n))$	✓	✓		✓	✓	✓	✓
$\Gamma_{\log(n)}$-Count	$\Omega(\log(n))$	$O(\log(n))$	✗	✗		✗	✗	✗	✗
ST	$\Omega(D)$	$O(D)$	✓	✗		✓	✓	✓	✗
MST	$\Omega(D)$	$O(D)$	✗	✗	✗	✓	✓	✗	✗
SPT	$\Omega(D)$	$O(D)$	✗	✗	✗	✓	✓	✗	✗
Flow	$\Omega(D)$	$O(D)$	✗	✗	✗	✓	✓	✗	✗
Leader	$\Omega(D)$	$O(D)$	✓	✓		✓	✓	✓	✓
Count	$\Omega(D)$	$O(D)$	✓	✓		✗	✗	✗	✗

Thanks!

Questions & Comments?