Some graph theory applications

to

communications networks

Keith Briggs

Keith.Briggs@bt.com

`http://keithbriggs.info`

Computational Systems Biology Group, Sheffield - 2006 Nov 02 1100

`graph_problems_Sheffield_2006_Nov_02.tex` TYPESET 2006 OCTOBER 27 14:08 IN PDFLATEX ON A LINUX SYSTEM

BT Research at Martlesham, Suffolk

- ★ Cambridge-Ipswich high-tech corridor
- ★ 2000 technologists
- ★ 15 companies
- ★ UCL, Univ of Essex

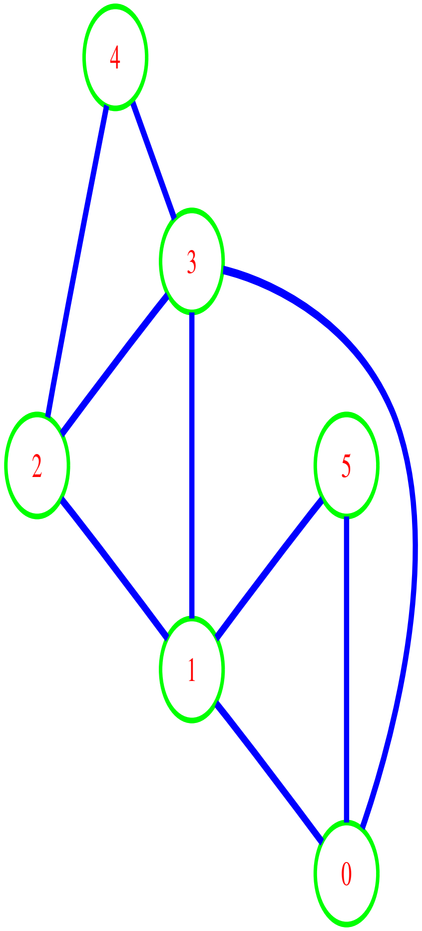
Mathematics in telecoms

- graph theory - network models
- optimization of network topology
- information theory
- Markov chains & queuing theory
- coding, compression, and cryptography
- packet protocols & traffic characteristics
- asynchronous distributed algorithms
- caching and data distribution strategies
- optimization of dynamic processes on networks (typically convex but non-smooth)
- business modelling & financial forecasting
- complex systems?

Talk outline

- ★ graph concepts and problems
- ★ connectivity
- ★ chromatic number and clique number
- ★ channel allocation
- ★ the challenge - to balance (exact) theory with (real) practice

Graph concepts



- *clique* - a complete subgraph
- *maximal clique* - a clique that cannot be extended to a larger one
- *lonely set* - a pairwise disjoint set of nodes (stable set, independent set)
- *colouring* - an assignment of colours to nodes in which no neighbours have the same colour
- *chromatic number* χ - the number of colours in a colouring with a minimal number of colours
- *loneliness* α - the number of nodes in a largest lonely set
- *clique number* ω - the number of nodes in a largest maximal clique

The Bernoulli random graph model $G\{n,p\}$

- let G be a graph of n nodes
- let $p = 1-q$ be the probability that each possible edge exists
- edge events are independent
- let $P(n)$ be the probability that $G\{n,p\}$ is connected
- then $P(1) = 1$ and $P(n) = 1-\sum_{k=1}^{n-1}\binom{n-1}{k-1}P(k)q^{k(n-k)}$ for $n = 2, 3, 4, \ldots$.

$$\begin{aligned}
P(1) &= 1 \\
P(2) &= 1-q \\
P(3) &= (2\,q+1)\,(q-1)^2 \\
P(4) &= \left(6\,q^3+6\,q^2+3\,q+1\right)(1-q)^3 \\
P(5) &= \left(24\,q^6+36\,q^5+30\,q^4+20\,q^3+10\,q^2+4\,q+1\right)(q-1)^4
\end{aligned}$$

- as $n \to \infty$, we have $P(n) \to 1-nq^{n-1}$.

Probability of connectivity - the $G(n,m)$ model

- problem: compute the numbers of connected labelled graphs with n nodes and $m=n-1,n,n+1,n+2,\dots$ edges
- exponential generating function for all labelled graphs:

$$g(w,z)=\sum_{n=0}^{\infty}(1+w)^{\binom{n}{2}}z^n/n!$$

- i.e., the number of labelled graphs with m edges and n nodes is $[w^m z^n]g(w,z)$
- exponential generating function for all connected labelled graphs:

$$\begin{aligned} c(w,z) &= \log(g(w,z)) \\ &= z+w\frac{z^2}{2}+(3w^2+w^3)\frac{z^3}{6}+(16w^3+15w^4+6w^5+w^6)\frac{z^4}{4!}+\dots \end{aligned}$$

Probability of connectivity for $G(n,m)$

- $\frac{P(n,n-1)}{2^n e^{2-n} n^{-1/2} \xi} \sim \frac{1}{2} - \frac{7}{8} n^{-1} + \frac{35}{192} n^{-2} + \frac{1127}{11520} n^{-3} + \frac{5189}{61440} n^{-4} + \frac{457915}{3096576} n^{-5} + \frac{570281371}{1857945600} n^{-6} + \frac{291736667}{495452160} n^{-7} + O\left(n^{-8}\right)$
  - ***check:*** $n = 10$, ***exact***=0.1128460393, ***asymptotic***=0.1128460359
- $\frac{P(n,n+0)}{2^n e^{2-n} \xi} \sim \frac{1}{4}\xi - \frac{7}{6} n^{-1/2} + \frac{1}{3}\xi n^{-1} - \frac{1051}{1080} n^{-3/2} + \frac{5}{9}\xi n^{-2} + O\left(n^{-3}\right)$
  - ***check:*** $n = 10$, ***exact***=0.276, ***asymptotic***=0.319
- $\frac{P(n,n+1)}{2^n e^{2-n} n^{1/2} \xi} \sim \frac{5}{12} - \frac{7}{12}\xi n^{-1/2} + \frac{515}{144} n^{-1} - \frac{28}{9}\xi n^{-3/2} + \frac{788347}{51840} n^{-2} - \frac{308}{27}\xi n^{-5/2} + O\left(n^{-3}\right)$
  - ***check:*** $n = 10$, ***exact***=0.437, ***asymptotic***=0.407
  - ***check:*** $n = 20$, ***exact***=0.037108, ***asymptotic***=0.037245
  - ***check:*** $n = 100$, ***exact***=2.617608×10^{-12}, ***asymptotic***=2.617596×10^{-12}

Hard graph problems

- finding χ, α and ω is proven to be NP-complete
  - *this means that it unlikely that any algorithm exists which runs in time which is a polynomial function of the number of nodes*

- we therefore have two options:
  - *use a heuristic, which is probably fast but may give the wrong answer*
  - *use an exact algorithm, and try to make it as fast as possible by clever coding*

- the theory is well developed and presented in many places, but little practical experience gets reported

- therefore, ti is interesting to try exact algorithms for these problems to determine how big the problems can be in practice, and compared the timings with approximate (relaxed) algorithms

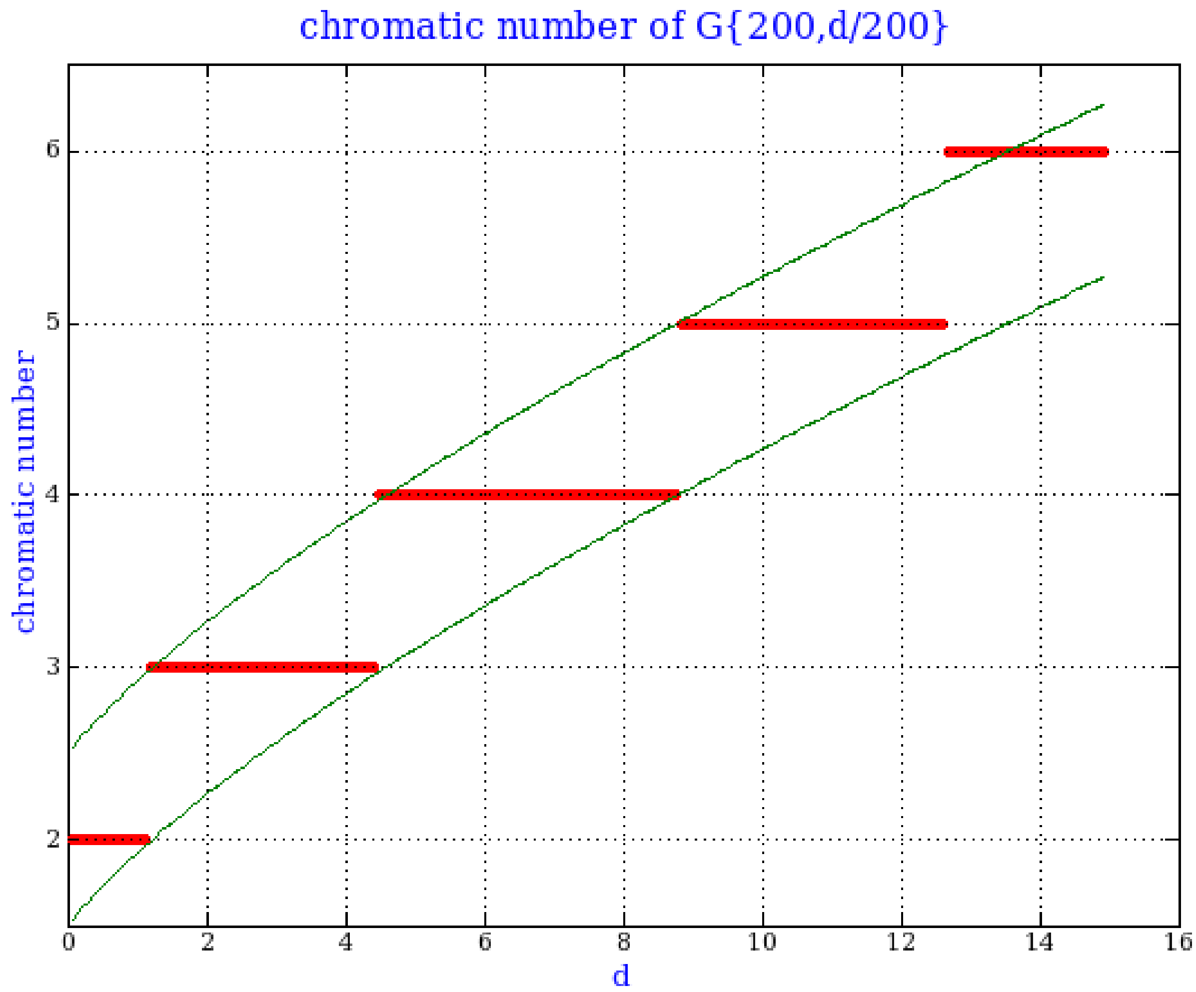
Chromatic number χ

- many papers appeared in the 1980s about backtracking (branch-and-bound) methods. Some had errors
  - *idea: start to compute all colourings, but abort one as soon as it is worse than the best so far*
- can be combined with heuristics (greedy colourings) and exact bounds like $\omega \leqslant \chi \leqslant \Delta+1$, where Δ is the maximum degree
- tradeoff in using heuristics depends on type of graph
- in practice (with a well-written C program), up to 100 nodes is ok, and up to 200 for very sparse or very dense graphs
- best results are in a PhD by Chiarandini (Darmstadt 2005) `http://www.imada.sdu.dk/~marco/public.php`
- determining χ may be easy for many real-world graphs with specific structures (Coudert, DAC97)

Achlioptas & Naor

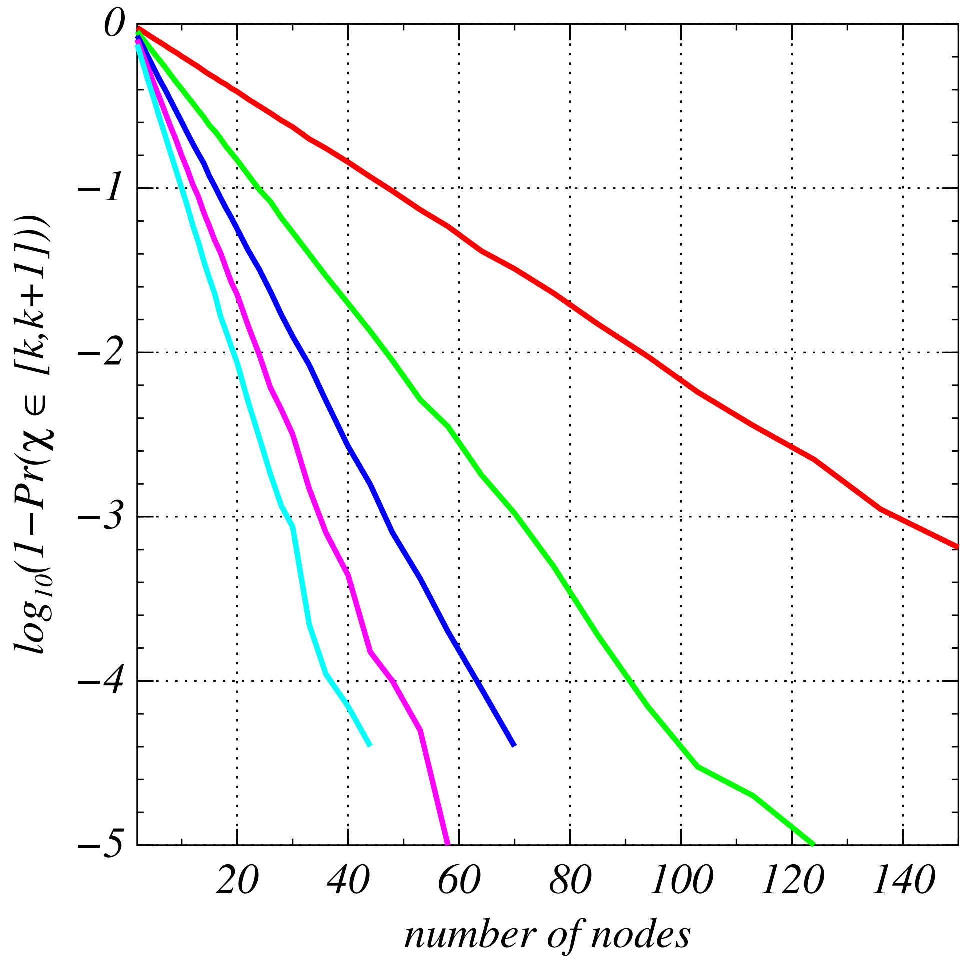
- *The two possible values of the chromatic number of a random graph* Annals of Mathematics, **162** (2005) `http://www.cs.ucsc.edu/~optas/`
- the authors show that for fixed d, as $n \to \infty$, the chromatic number of $G\{n, d/n\}$ is either k or $k+1$, where k is the smallest integer such that $d < 2k\log(k)$. In fact, this means that k is given by $\lceil d/(2W(d/2)) \rceil$
- $G\{n, p\}$ means the random graph on n nodes and each possible edge appears independently with probability p

Achlioptas & Naor cotd.



Achlioptas & Naor - my conjecture

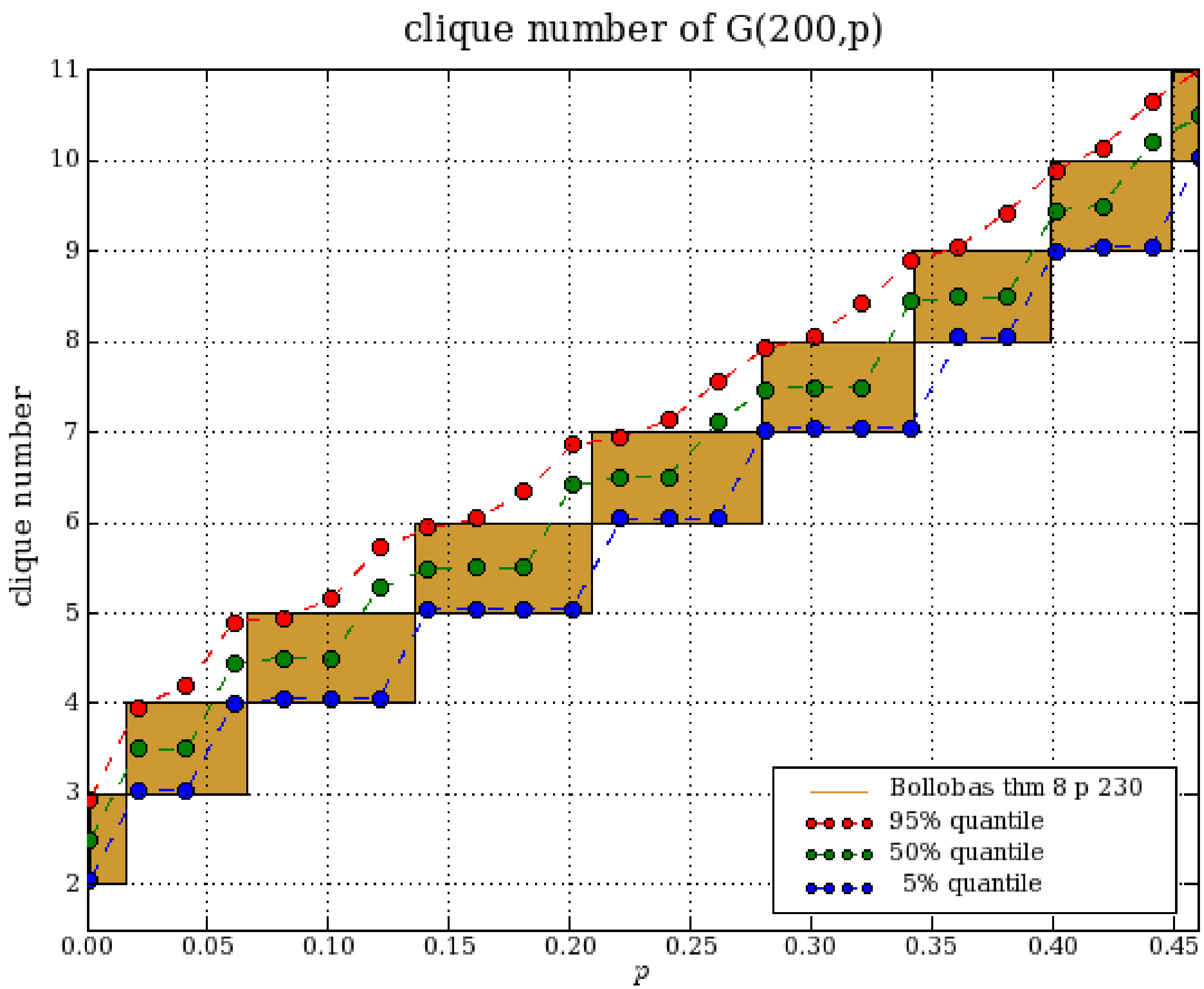
- the next graph (each point is the average of 1 million trials) suggests that for small d, we have Pr $[\chi \in [k, k{+}1]] \sim 1{-}\exp(-dn/2)$



Clique number

- In *Modern graph theory*, page 230, Bollobás shows that the clique number of $G(n,p)$ as $n \to \infty$ is almost surely d or $d+1$, where d is the greatest natural number such that $\binom{n}{d} p^{\binom{d}{2}} \geqslant \log(n)$
- How accurate is this formula when n is small?
- We have $d = 2\log(n)/\log(1/p) + \mathcal{O}(\log\log(n))$.

Clique number - simulation results



Counting graphs

Number of graphs on n nodes with chromatic number k:

k \ n =	1	2	3	4	5	6	7	8	9	10	
2	0	1	2	6	12	34	87	302	1118	5478	A076278
3	0	0	1	3	16	84	579	5721	87381	2104349	A076279
4	0	0	0	1	4	31	318	5366	155291	7855628	A076280
5	0	0	0	0	1	5	52	867	28722	1919895	A076281
6	0	0	0	0	0	1	6	81	2028	115391	A076282
7	0	0	0	0	0	0	1	7	118	4251	
8	0	0	0	0	0	0	0	1	8	165	
9	0	0	0	0	0	0	0	0	1	9	
10	0	0	0	0	0	0	0	0	0	1	
11	0	0	0	0	0	0	0	0	0	0	

(A-numbers from `http://www.research.att.com/~njas/sequences/`)

A *real-world* hard probelm

- I use *real* 802.11b spectral characteristics & interference behaviour
- the channel allocation problem is to minimize the maximum interference problem
- randomly placed nodes
- hexagonal lattices

The channel allocation problem

- choose x such that some objective function is minimized
- This is a combinatorial optimization problem, so to find the exact solution we must explicitly enumerate and evaluate all channel assignments
- the number of assignments grows as $(\text{number of nodes})^{\text{number of channels}}$ and becomes infeasible to do a complete search beyond about 12 channels and 12 nodes
- so we use branch and bound method for the maximum interference problem.
  - *we build a tree showing all possible assignment vectors with the depth of tree representing the number of nodes being considered and each leaf a different complete assignment. We do this by testing partial solutions and disregarding ones worse than the best so far.*

The maximum interference problem

- the *maximum interference at node i* is

$$w_i = \max_{\substack{j=1,\dots,n \\ j\neq i}} I_{ij}$$

- the *objective function* is $w(x) = \max_i w_i(x)$; that is, the worst maximum interference at any AP
- the *optimization problem* is

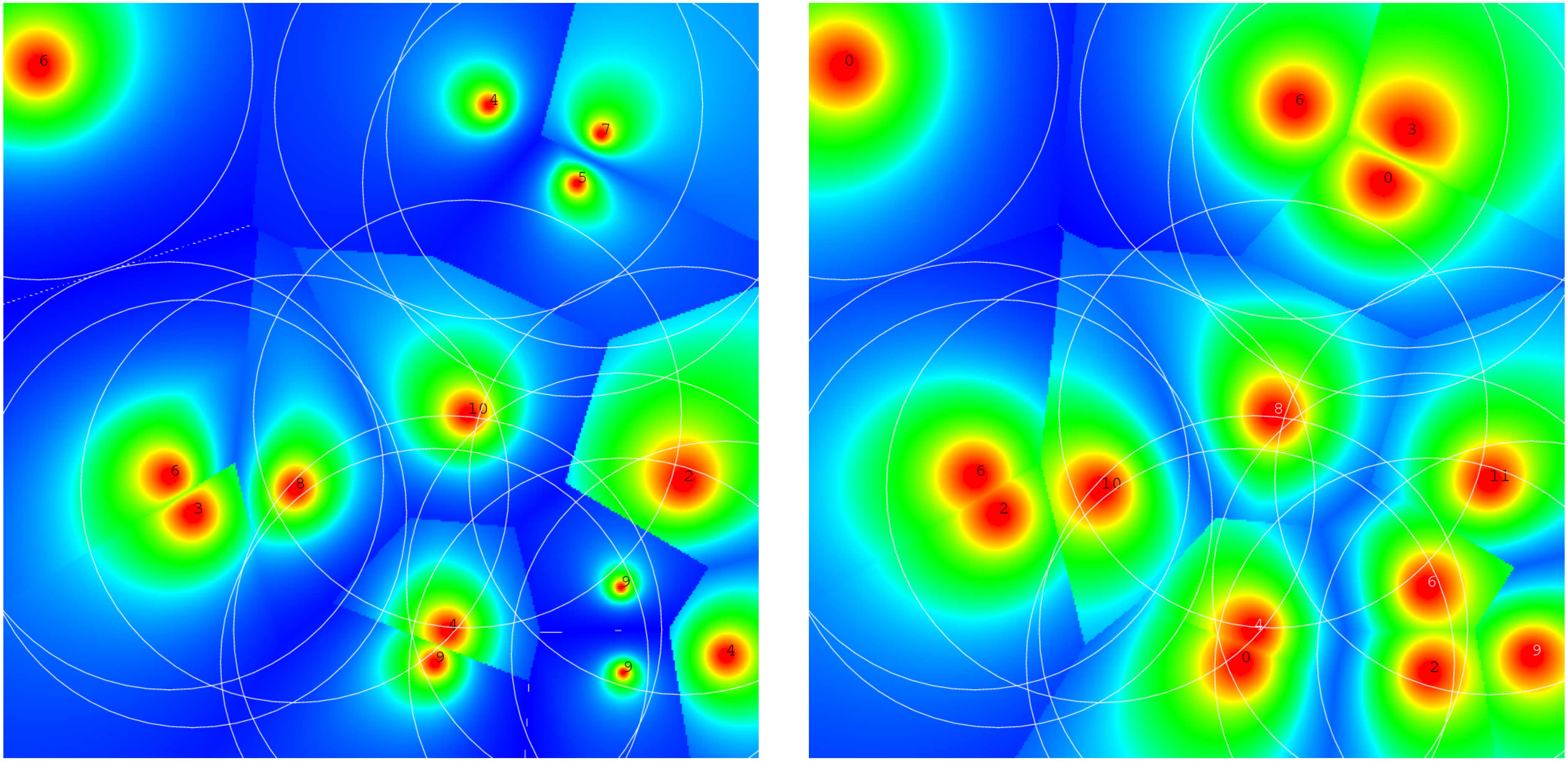
$$\min_x w(x);$$

  that is, we aim to minimize the worst maximum interference
- this is feasible to solve exactly if good pruning strategies can be found

Pruning and preprocessing

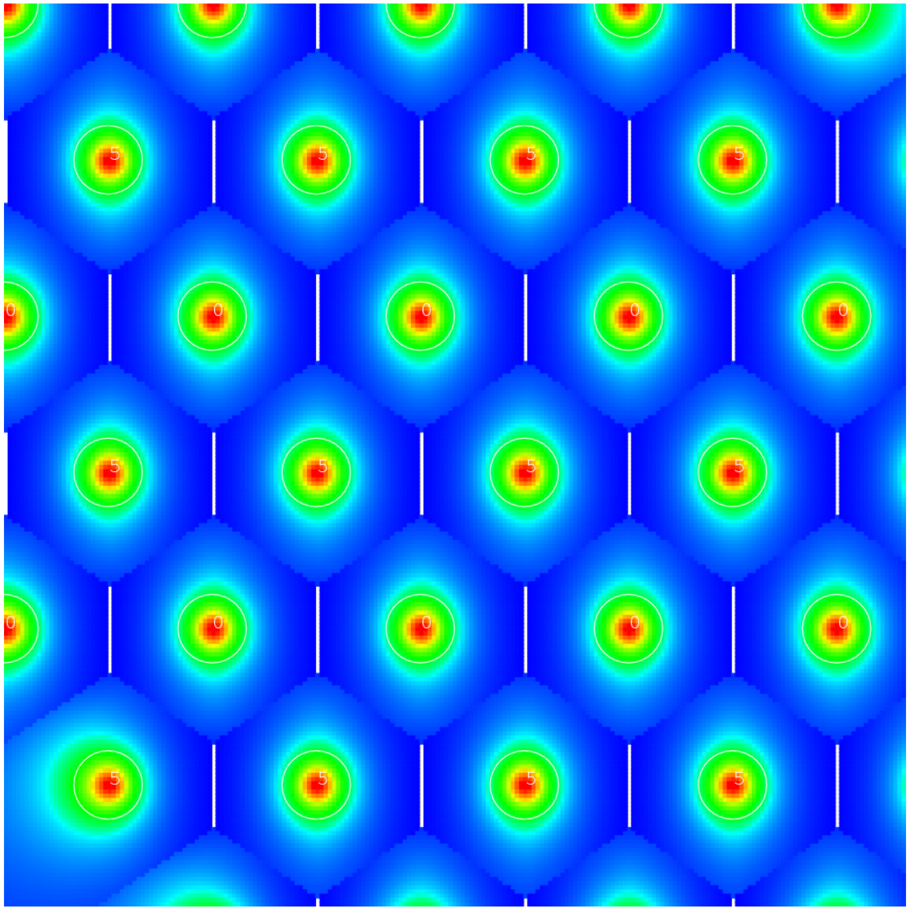
- to have any advantage over complete enumeration efficient pruning strategies must be found
- testing of partial solutions to determine possible good solutions
  - *in a typical example the number of function calls can drop from* 6.10^6 *to about* 6000
- calculation of minimum separations from interference matrix
  - *this can usually give a further* $50-75\%$ *reduction in function calls*
- while branch and bound is powerful on its own it is sensitive to the order in which the nodes are considered.
- by using the k-means heuristic to locate clusters and analysing these first pruning, become much more effective

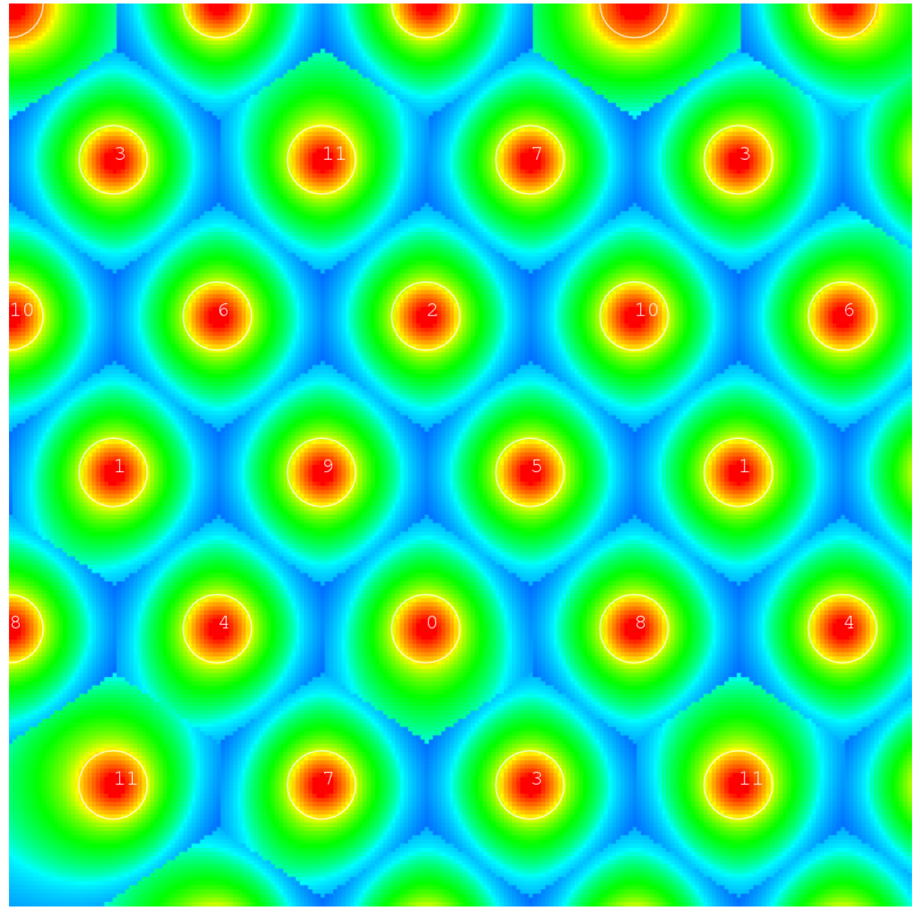
Randomly placed nodes: before & after optimization



typical improvement: 2Mbps coverage goes from 50% to 90%.

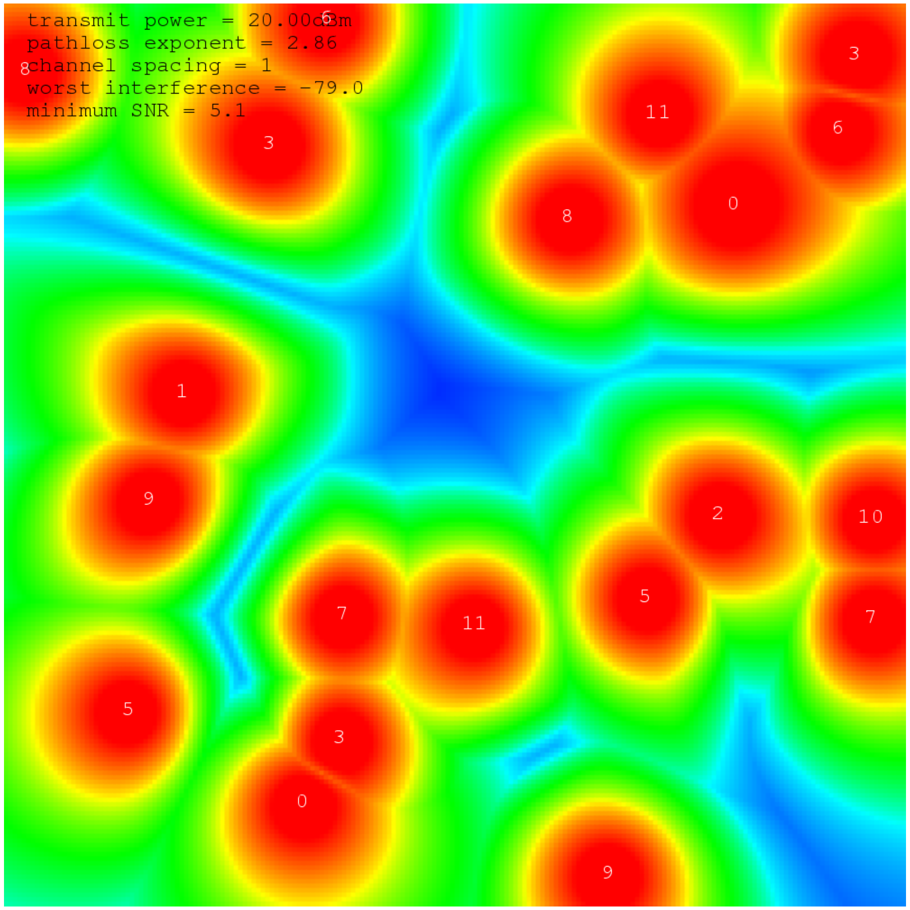
Hexagonal lattice - 3 and 12 channels

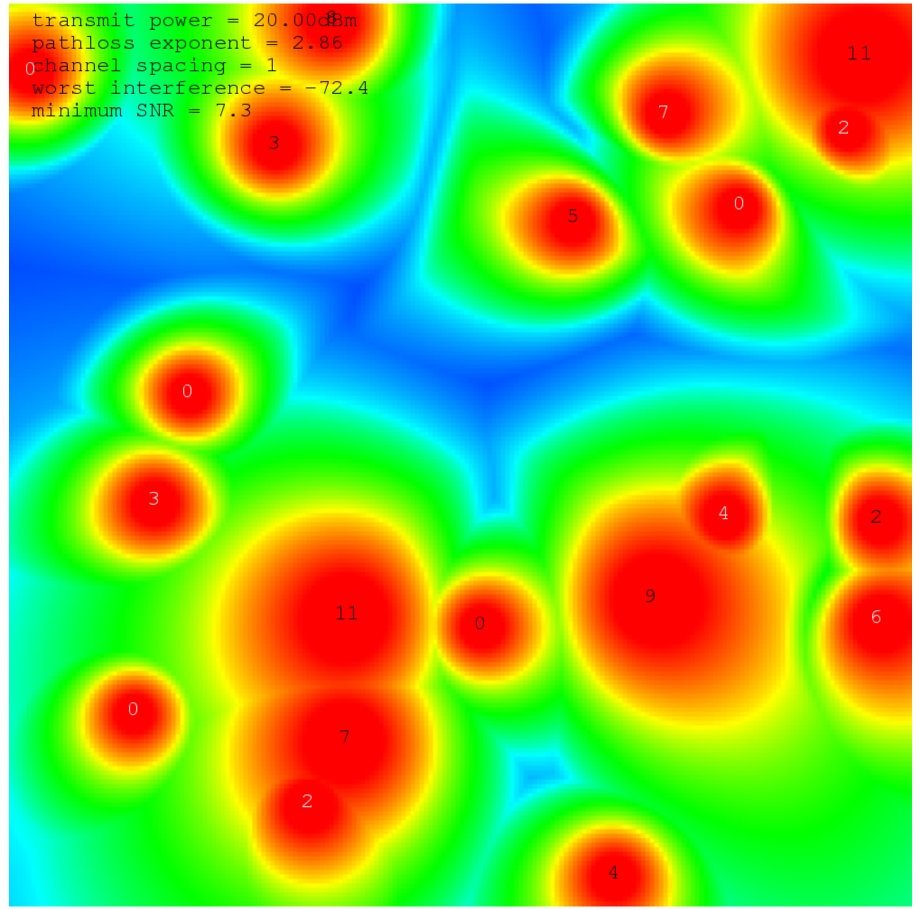




typical improvement: 12Mbps coverage goes from 26% to 100%.

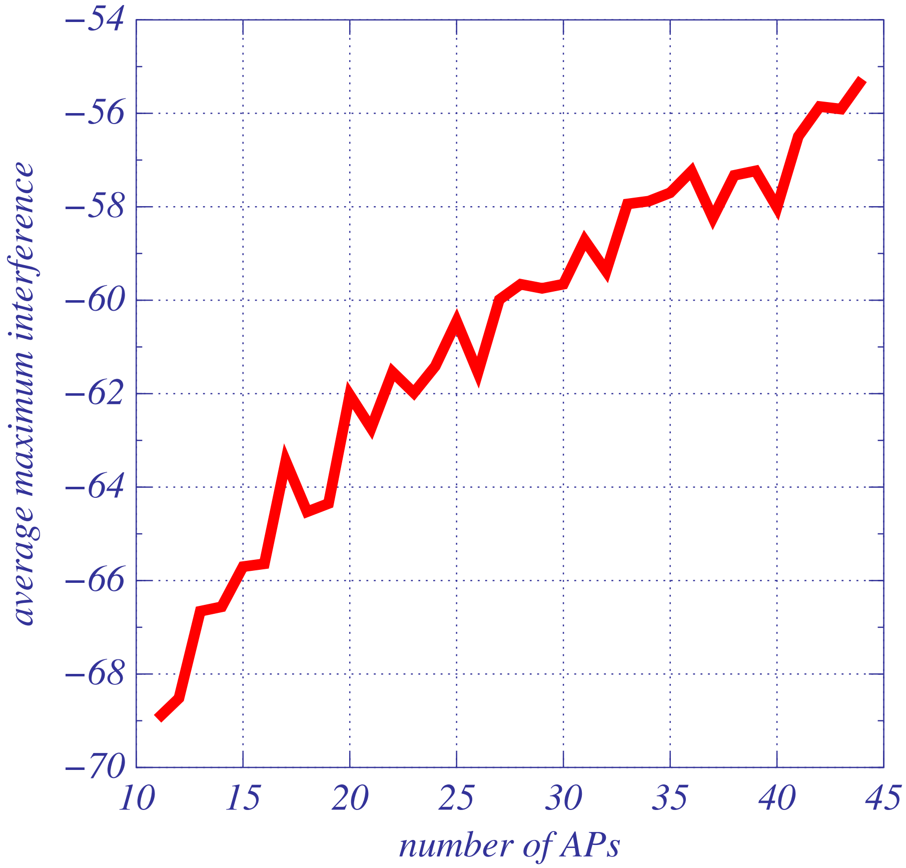
Two-network optimization

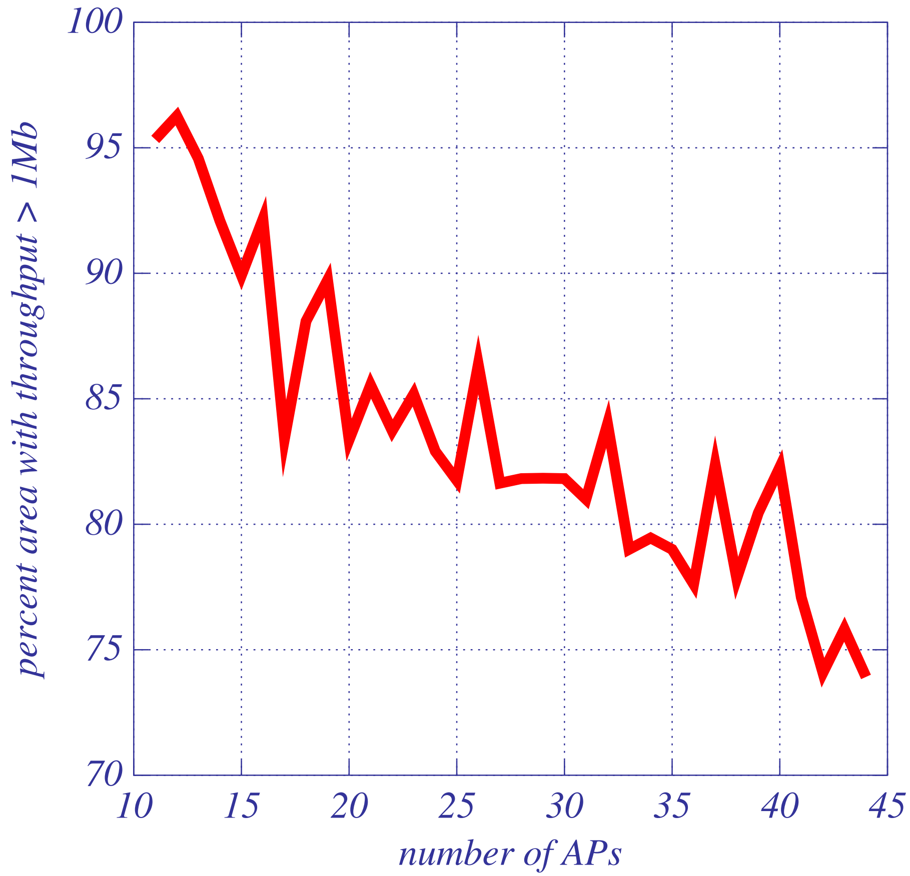




First optimize all 20 nodes, then imagine the first 10 nodes belong to a competitor's network and are optimized and then frozen, and then we come in with the second 10 nodes. How is our coverage and SNR affected by the competitor's network? (Answer: only about 2dB.)

Scaling of interference & throughput with node density





Results here are averaged over many instances of Poisson point process.