Some hard graph problems
in
telecoms

Keith Briggs

Keith.Briggs@bt.com

http://keithbriggs.info

Warwick Complexity Forum - 2006 Oct 31 1515

graph_problems_Warwick_2006_Oct_31.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

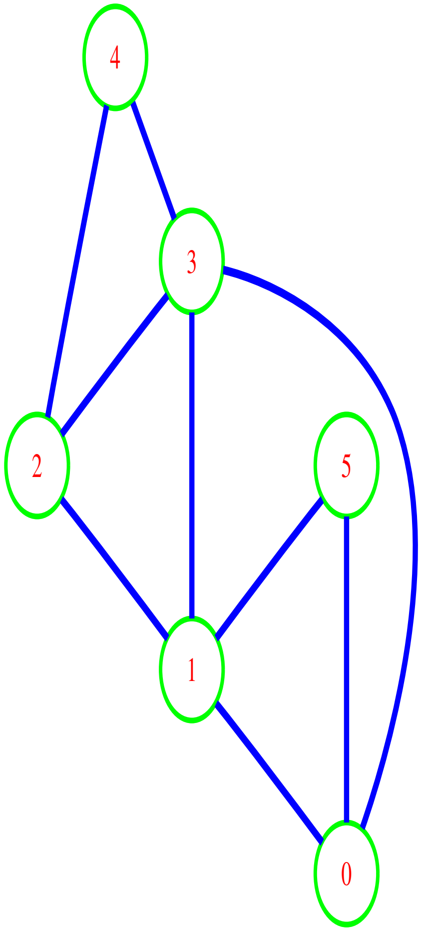
Mathematics in telecoms

- ★ 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
- ★ chromatic number and clique number
- ★ a real problem - channel allocation
- ★ theme - How to balance (exact) theory with (real) practice?
- ★ this is about *algorithmic* complexity, not complex systems as usually understand. But perhaps there are some overlaps . . .

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

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)

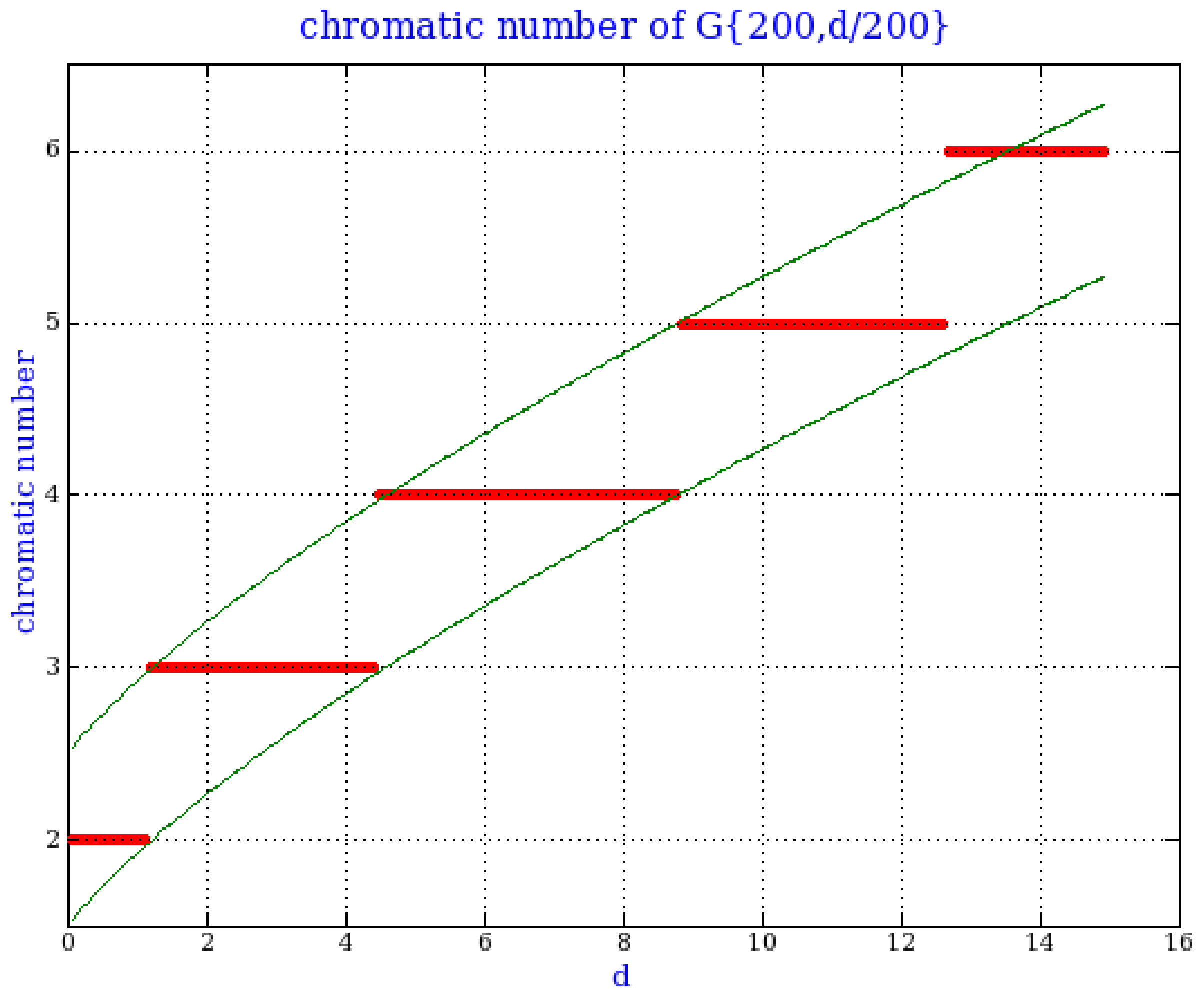
Achlioptas & Naor

- *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 $\lceil d/(2W(d/2)) \rceil$

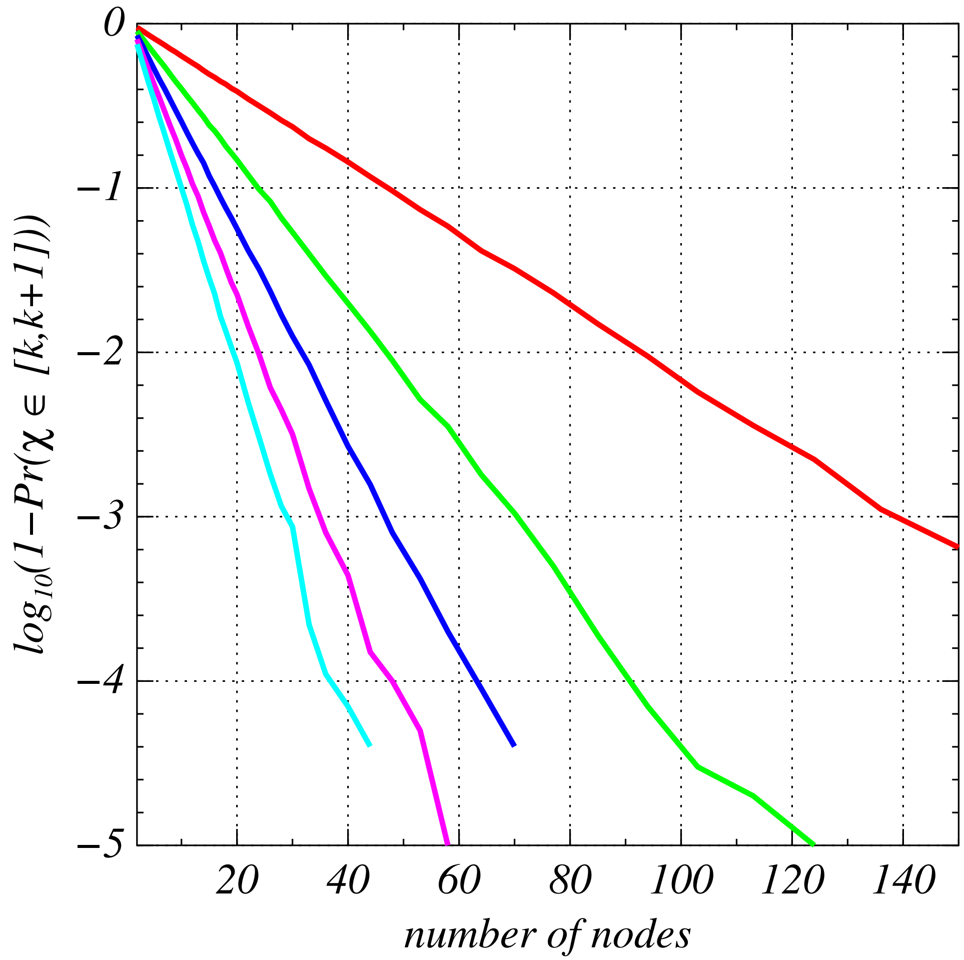
- $G\{n, p\}$ means the random graph on n nodes and each possible edge appears independently with probability p

Achlioptas & Naor cotd.



Achlioptas & Naor - my conjecture

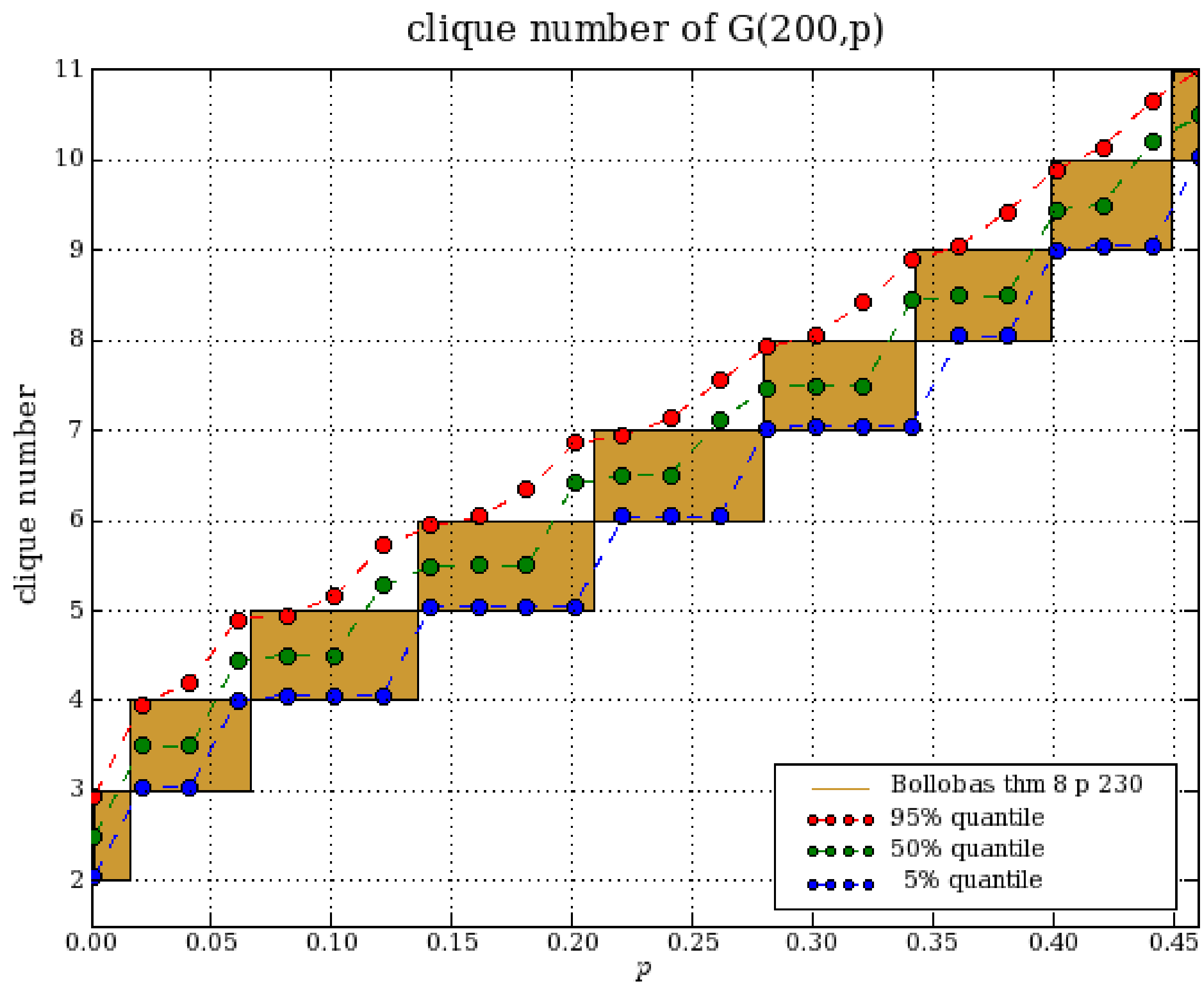
- the next graph (each point is the average of 1 million trials) suggests that for small d, we have Pr $[\chi \in [k, k+1]] \sim 1-\exp(-dn/2)$



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/`)

Counting graphs cotd.

Number of graphs on n nodes with clique number k:

k \ n =	1	2	3	4	5	6	7	8	9	10	
2	0	1	2	6	13	37	106	409	1896	12171	A052450
3	0	0	1	3	15	82	578	6021	101267	2882460	A052451
4	0	0	0	1	4	30	301	4985	142276	7269487	A052452
5	0	0	0	0	1	5	51	842	27107	1724440	A077392
6	0	0	0	0	0	1	6	80	1995	112225	A077393
7	0	0	0	0	0	0	1	7	117	4210	A077394
8	0	0	0	0	0	0	0	1	8	164	
9	0	0	0	0	0	0	0	0	1	9	
10	0	0	0	0	0	0	0	0	0	1	

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

A *real-world* hard probelm

- 802.11b spectral characteristics & interference
- the channel allocation problem
- the minimize maximum interference problem
- randomly placed nodes
- hexagonal lattices

802.11b spectral characteristics

- a *channel assignment* is a vector $x \in \mathbb{Z}^n$, meaning that x_i is the channel used by node i
- the 802.11b spectral envelope is $ch(n,f) \equiv (f-2412-s(n-1))/22$

$$
\begin{aligned}
s(f) &\equiv |\sin(2\pi f)/(2\pi f)| \\
\mathsf{flt}(x) &\equiv 1/(1+(2.6x)^6) \\
\mathsf{ol}(n,m,x) &\equiv \mathsf{flt}(ch(n,x))s(ch(n,x))\mathsf{flt}(ch(m,x))s(ch(m,x)) \\
k_o &\equiv \int_{2200}^{2700} \mathsf{ol}(1,1,x)\,\mathsf{d}x \approx 9.265481882 \\
\mathsf{olf}_k &\equiv \int_{2200}^{2700} \mathsf{ol}(1,k+1,x)/k_o\,\mathsf{d}x.
\end{aligned}
$$

- this gives (taking $20\log_{10}(\mathsf{olf}_k)$ to get dB) the vector of overlap factors as: $[0, -2.767, -11.329, -28.525, -45.296, -61.560, -74.686, \ldots$

802.11b interference

- the *interference at node j caused by node i* is $I_{ij} = r_{ij} + c(|x_i - x_j|)$ where $r_{ij} = T_j - (P_{\text{ref}} + 10m\log_{10}(d_{ij}))$ dBm is the received power at node i from node j.
- d_{ij} is the distance from node i to node j
- the $\log$ factors are due the conversions to and from dB units
- T_j is the transmit power, typically 20dBm (100mW)
- P_{ref} is the reference loss at 1m, typically 40.2dB
- m is the path loss exponent, typically about 2.86

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 (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

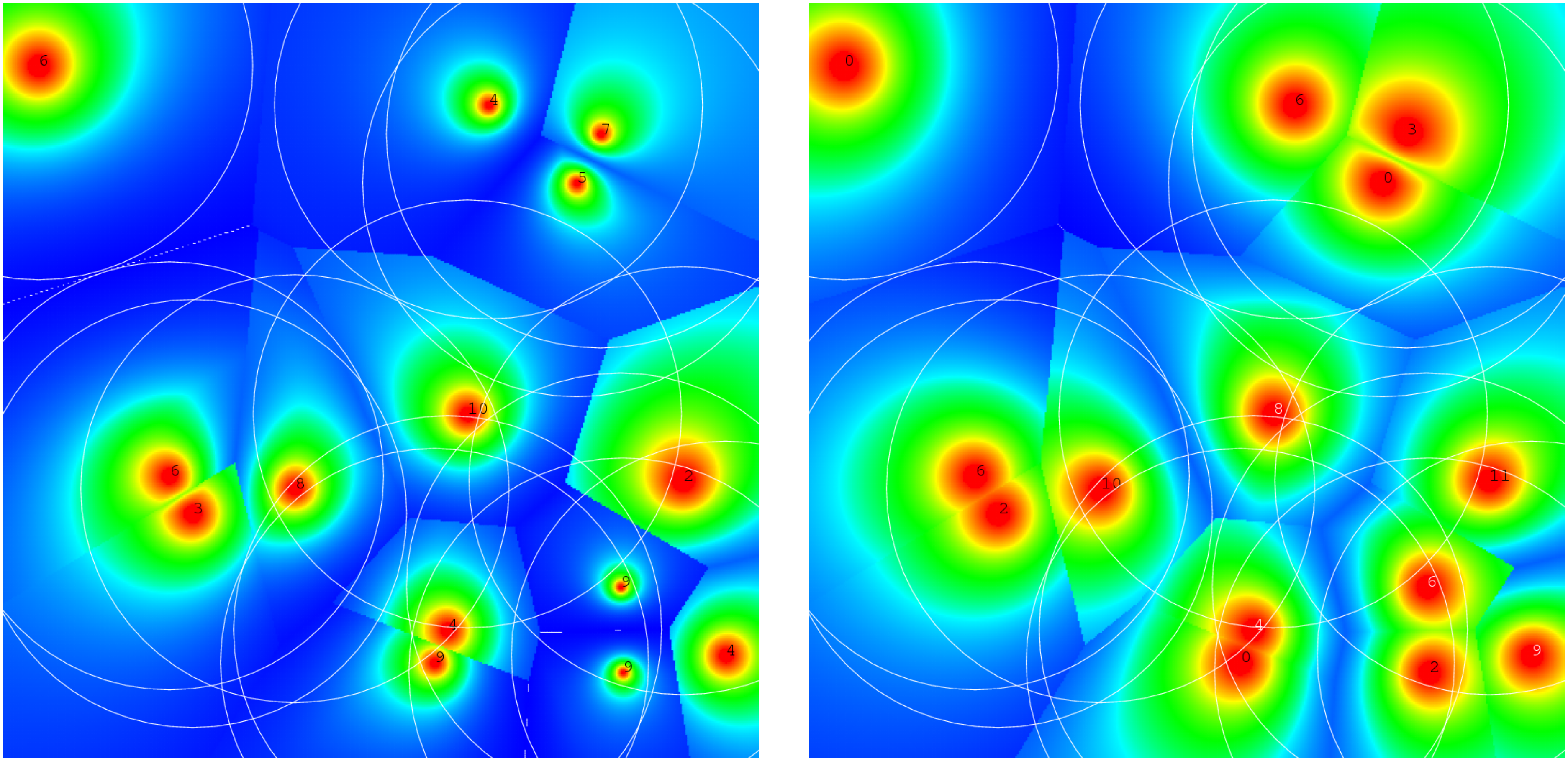
★ calculation of minimum separations from interference matrix

  ▷ *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.

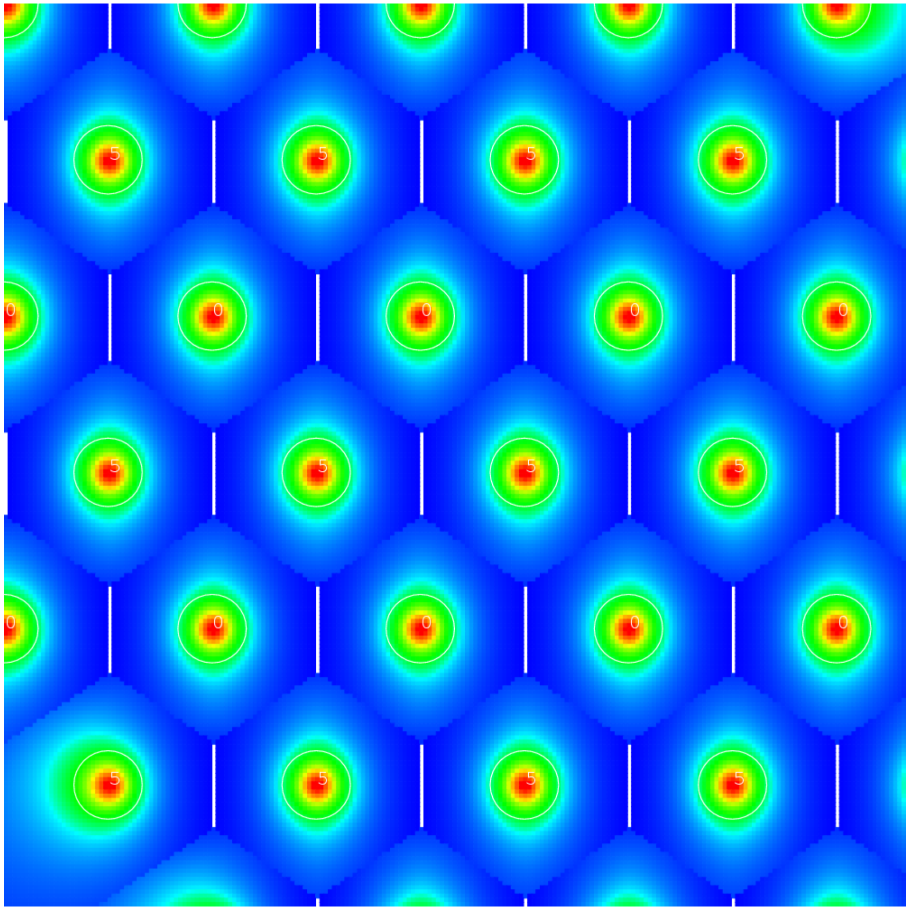
★ by using the k-means heuristic to locate clusters and analysing these first pruning, become much more effective

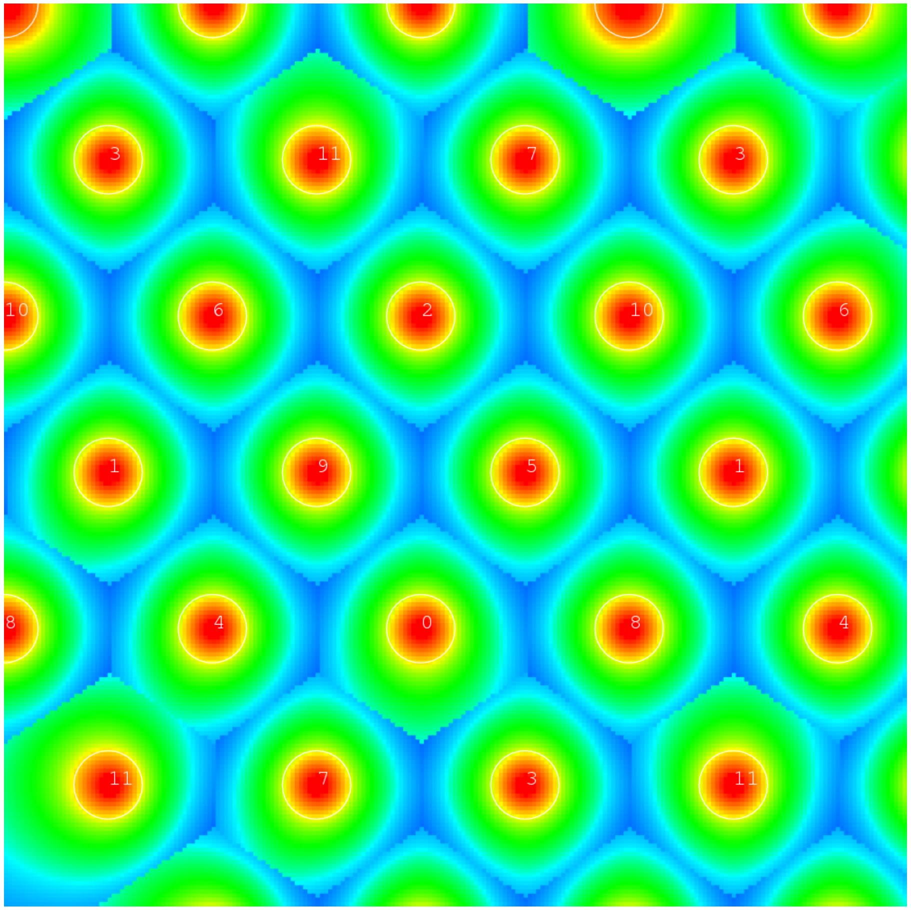
Randomly placed nodes: before & after optimization



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

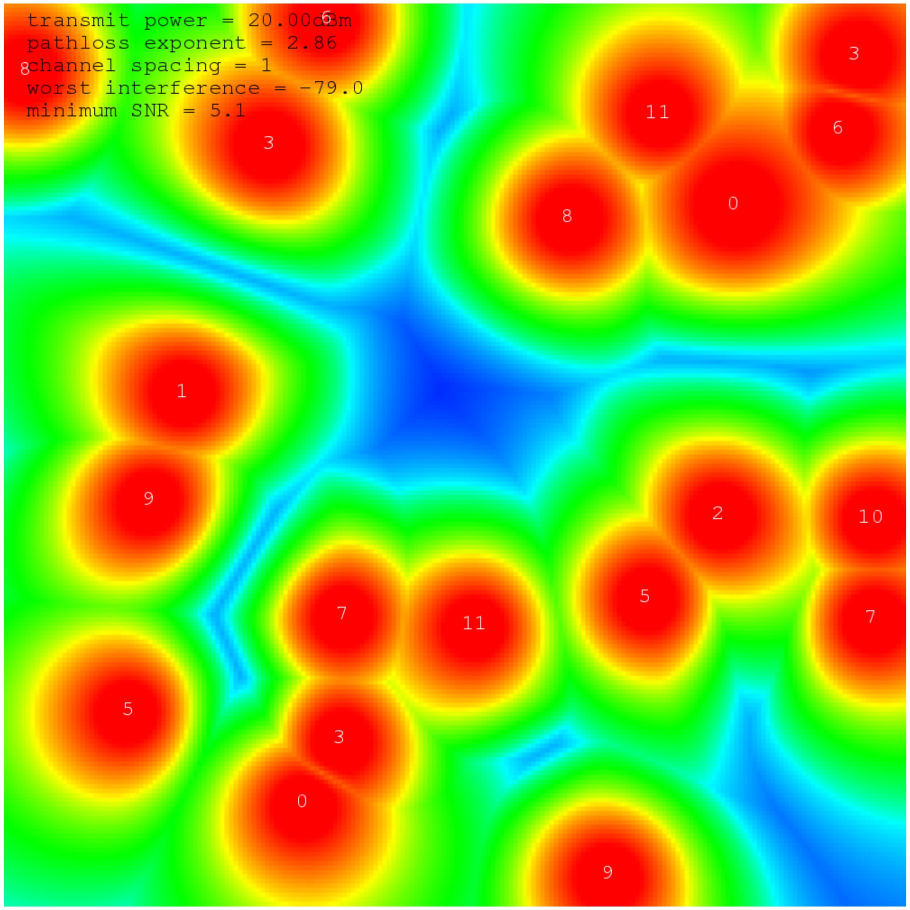
Hexagonal lattice - 3 and 12 channels

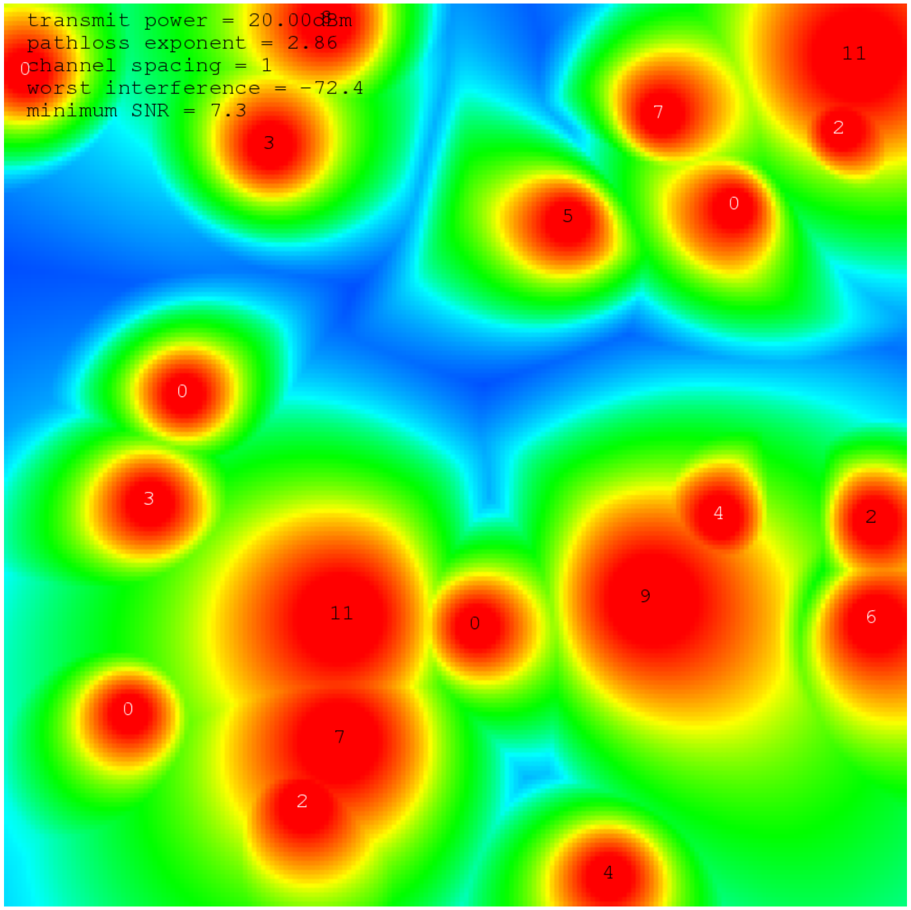




typical improvement: 12Mbps coverage goes from 26% to 100%.

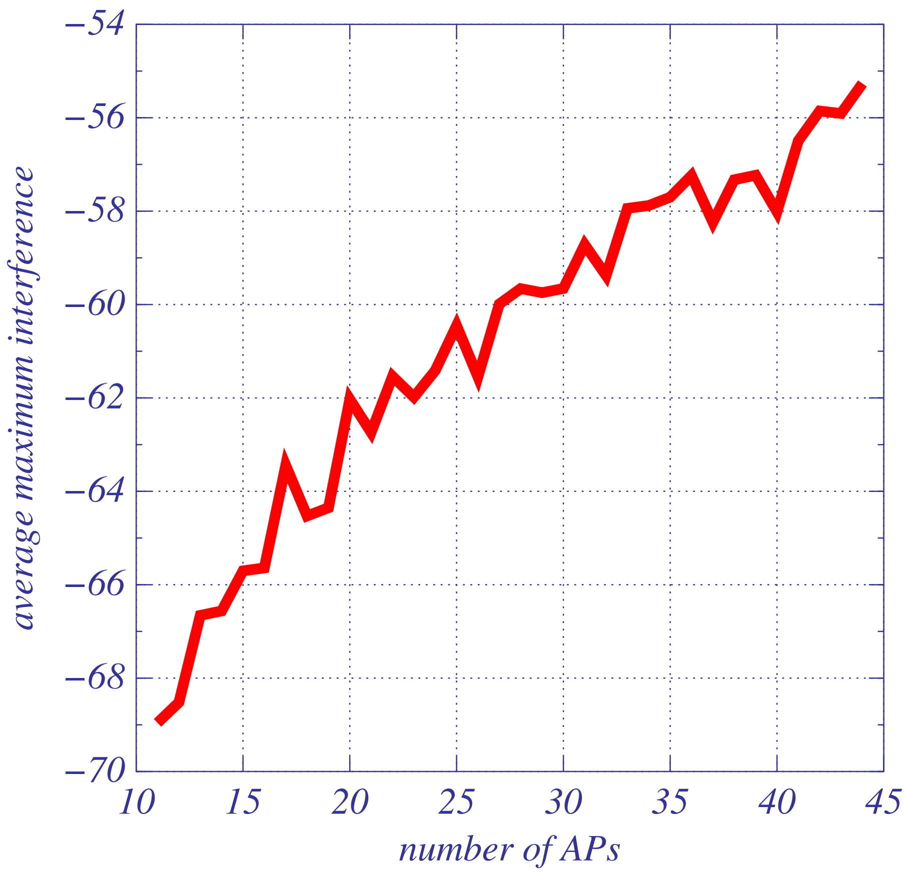
Two-network optimization

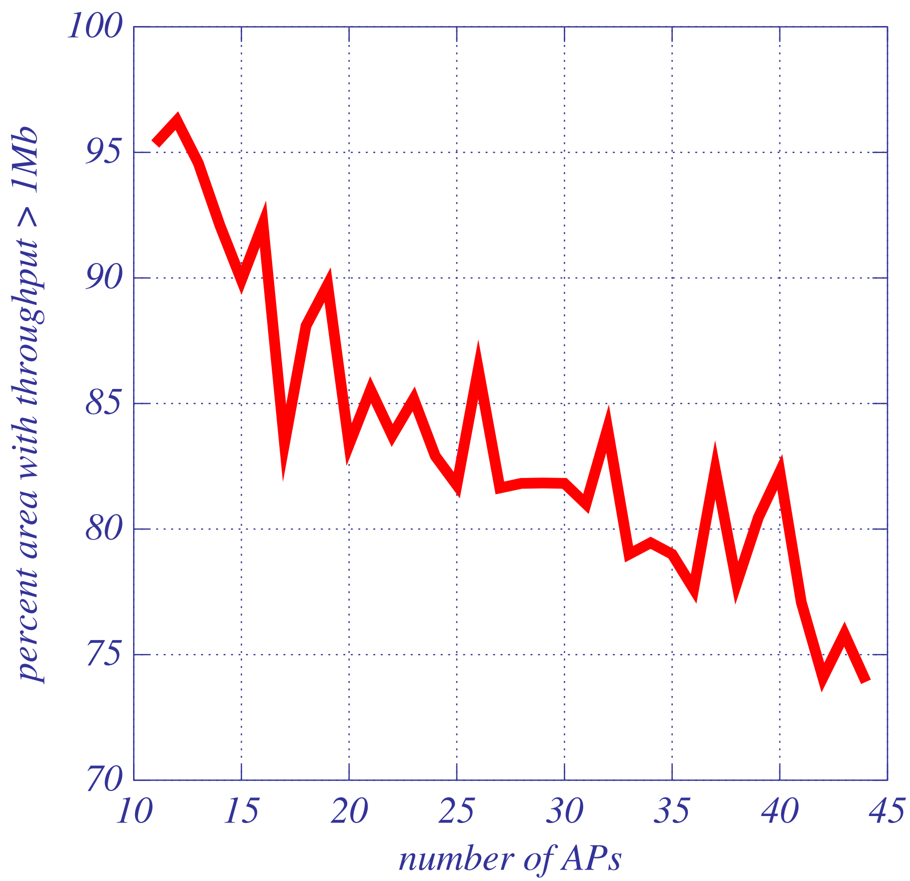




First optimize all 20 nodes, then imagine the first 10 nodes belong to a competitor's network and are optimized and then frozen, and then we come in with the second 10 nodes. How is our coverage and SNR affected by the competitor's network? (Answer: only about 2dB.)

Scaling of interference & throughput with node density

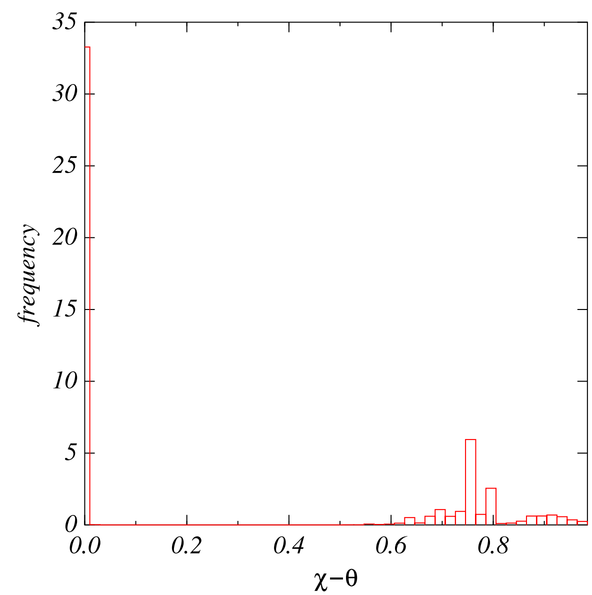




Results here are averaged over many instances of Poisson point process.

Relaxations and semidefinite programming

- idea: formulate as an integer linear program (still hard), then relax constraints to obtain a semidefinite program (SDP, easy to solve)
- SDP: provides the *Lovász θ number* for a graph. This number is an upper bound for the clique number of a graph, and a lower bound for the chromatic number
- best SDP code: DSDP5.8 by Benson `http://www-unix.mcs.anl.gov/DSDP/`
- distribution of $\chi - \theta$ on $G\{n, p\}$:



LP formulation of chromatic number and clique number

- let B be the 0-1 matrix with n rows and whose columns indicate the lonely sets (in practice, ok to use only maximal lonely sets). Finding B is slow
- chromatic number χ is the solution of the 0-1 ILP

$$\begin{aligned} \text{minimize} \quad & 1^T x \\ \text{subject to} \quad & Bx \geqslant 1 \end{aligned}$$

- clique number ω is the solution of the 0-1 ILP

$$\begin{aligned} \text{maximize} \quad & y^T 1 \\ \text{subject to} \quad & y^T B \leqslant 1 \end{aligned}$$

- solving the 0-1 ILPs is hard, so we don't try

Fractional chromatic number χ_f

- ★ used by McDiarmid for a radio channel assignment problem in which the *demand* (required number of channels) at each node varies
- ★ χ_f is the solution of the LP (ordinary LP, so easy)

$$\begin{aligned} \text{minimize} \quad & 1^T x \\ \text{subject to} \quad & Bx \geqslant 1 \\ & x \geqslant 0 \end{aligned}$$

- ★ ω_f is the solution of the LP (ordinary LP, so easy)

$$\begin{aligned} \text{maximize} \quad & y^T 1 \\ \text{subject to} \quad & y^T B \leqslant 1 \\ & y \geqslant 0 \end{aligned}$$