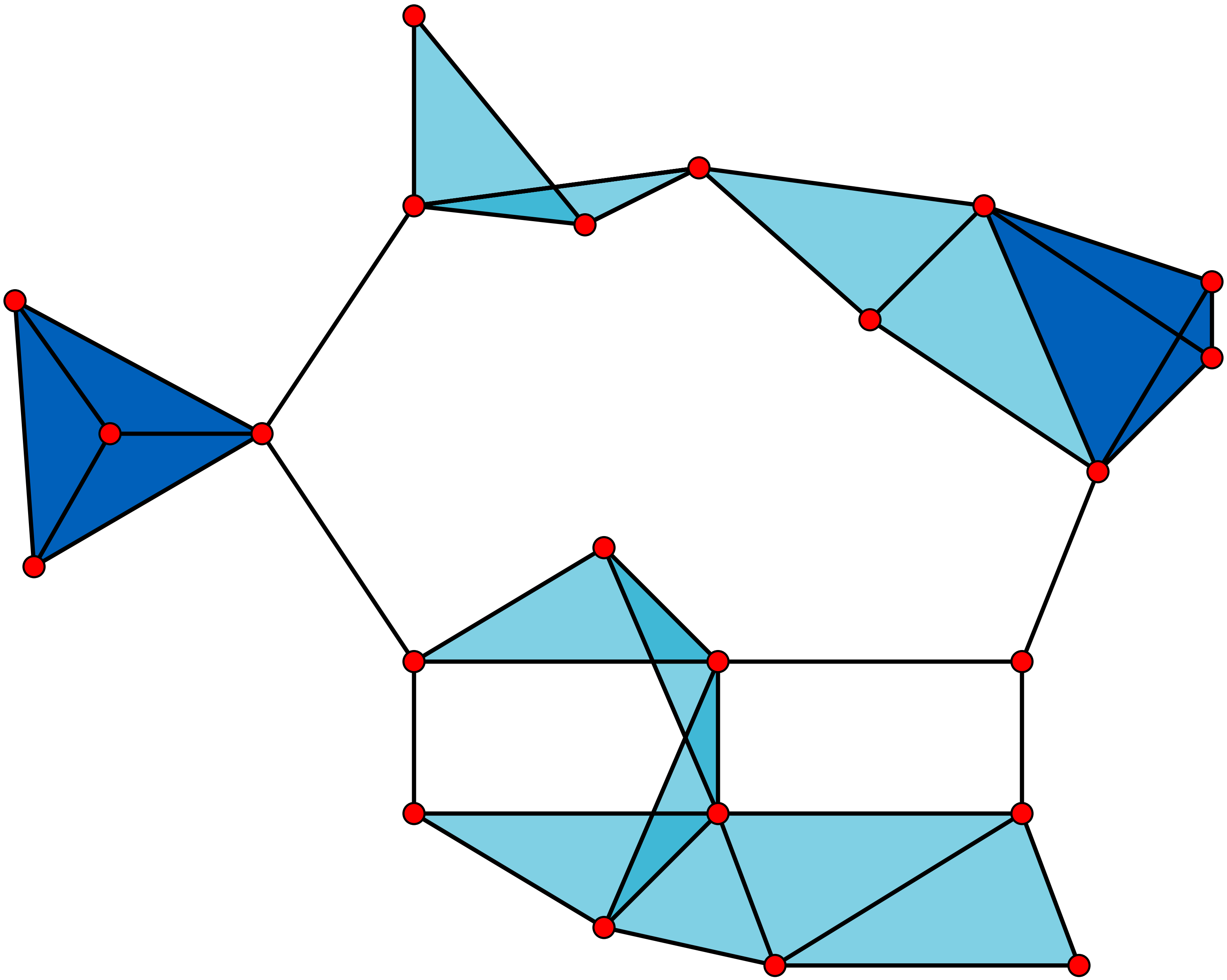
Finding Cliques Quickly: An Update

David Eppstein

(includes joint work with Darren Strash and Maarten Löffler)

Clique: subset of vertices in a social network that are all pairwise connected
Maximal clique: can't add any more vertices and retain complete connectivity



A few milestones

1949: Duncan Luce and Albert Perry name the clique problem, and apply it to the problem of finding well-connected subsets of people in a social network

1957: Harary and Ross describe the first clique-finding algorithm

1965: Moon and Moser show that every n-vertex graph has at most $3^{n/3}$ maximal cliques, and that some graphs have this many

1973: Bron and Kerbosch describe a simple backtracking procedure for listing all maximal cliques that works well in practice

... many more algorithms and papers of lesser importance ...

2006: Tomita, Tanaki, and and Takahashi partially explain the success of the Bron–Kerbosch algorithm by showing its worst case running time exactly matches the Moon–Moser bound

A paradox?

The Bron–Kerbosch algorithm
has the fastest possible
worst-case time bound...

...but this time bound
is far too slow
to explain its good practicality

Our resolution of the paradox [E, Löffler, Strash, ISAAC 2010]

Instead of using only the number of vertices, use “degeneracy”
(minimum d such that every subgraph has a vertex of degree ≤ d)

Prove that every graph has at most $3^{d/3}$ n maximal cliques
and that some graphs have this many (analogous to Moon–Moser)

Find a variant of the Bron–Kerbosch algorithm
(based on carefully sequencing the recursive subproblems)
whose running time is $O(d3^{d/3} n)$, almost optimal

Linear for graphs of bounded degeneracy
as we expect to be true for many social networks

But does our variant work well in practice?

Results for Stanford data sets

graph	n	m	d	tomita	maxdegree	hybrid	degen
road-CA	1,965,206	2,766,607	3	*	2.00	5.34	5.81
road-PA	1,088,092	1,541,898	3	*	1.09	2.95	3.21
road-TX	1,379,917	1,921,660	3	*	1.35	3.72	4.00
amazon	403,394	2,443,408	10	*	3.59	5.01	6.03
email-EuAll	265,214	364,481	37	*	4.93	1.25	1.33
email-Enron	36,692	183,831	43	31.96	2.78	1.30	0.90
web-Google	875,713	4,322,051	44	*	9.01	8.43	9.70
wiki-Vote	7,115	100,762	53	0.96	4.21	2.10	1.14
slashdot	82,168	504,230	55	*	7.81	4.20	2.58
cit-Patents	3,774,768	16,518,947	64	*	28.56	49.22	58.64
Epinions1	75,888	405,740	67	*	27.87	9.24	4.78
wiki-Talk	2,394,385	4,659,565	131	*	$> 18,000$	542.28	216.00
berkstan	685,231	6,649,470	201	*	76.90	31.81	20.87

Summary

New algorithm is fast...
...faster than the Tomita et al version
...maybe fast enough that our new bounds don't fully explain its speed

Additionally, because it uses significantly less memory than Tomita et al.,
it can be applied to much larger graphs

We presented these experiments at the 2011 Symp. on Experimental Algorithms

Our paper was judged to be one of the three best from the symposium,
invited to a special issue of J. Experimental Algorithms

We intend to make the code available within R
so that it can be incorporated into actual social network analysis
next phase of research, not yet complete