Asychronous distributed graph coloring

Keith Briggs & James Evans
keith.briggs@bt.com http://keithbriggs.info

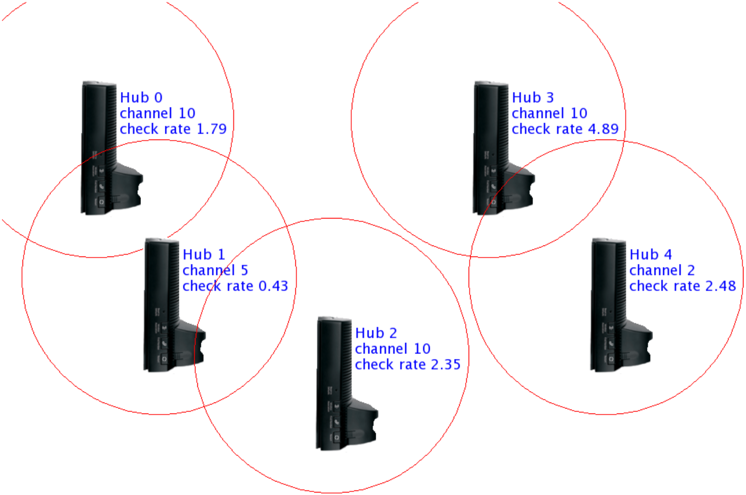
Mobility Research, BT Innovate & Design
Department of Mathematics, University of York

2010-08-19 16:00

Abstract

One of the original motivations for studying graph coloring problems was channel assignment for fixed broadcast radio systems. With the advent of homehubs, there is renewed interest, but now with the difference that the system cannot be planned as a whole before it is deployed, and optimal channel assignments cannot be computed centrally. Therefore, autonomous, decentralized heuristics are required, and this study is a comparison of the efficiency of various competing heuristics in this general class.

Home hubs


Hub 0
channel 10
check rate 1.79
Hub 3
channel 10
check rate 4.89
Hub 1
channel 5
check rate 0.43
Hub 4
channel 2
check rate 2.48
Hub 2
channel 10
check rate 2.35

Principles

- Think of interference graph
- No central control
- Nodes know only about neighbours
- Nodes may switch on and off, or new nodes may enter the system
- Simulated annealing — bad!
- How far can we reduce conflicts (interference) under these restrictions?
- How closely can we approach the chromatic number?

Heuristics

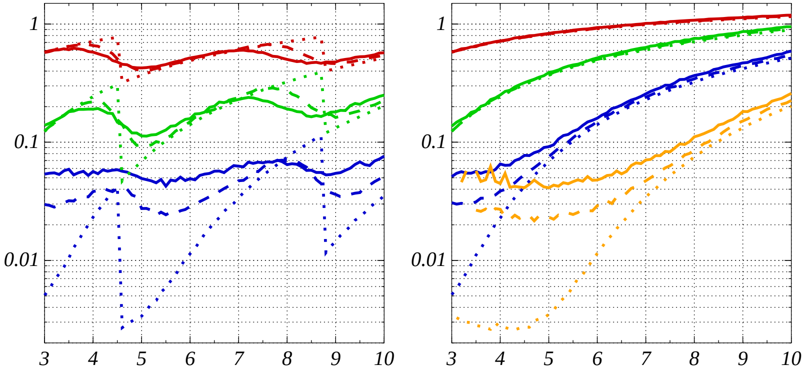
- rate of checking for conflicts — fixed/variable — F/V
- resolution — complete/partial/mixed — C/P/M
- choice of new channel — deterministic/stochastic — D/S

- τ is next wake-up time

Heuristic FCD

```
if clock_time=τ then
    col_neigh ←{colour of neighbours}
    if col_neigh=∅ then
        our_color←0
    else
        for col ∈ colours do
            if #conflicts(col, col_neigh)=0 then
                our_color← col
                break for
    τ←τ+expovariate(1)
```

Erdős-Rényi $G(n, p)$ — FCD baseline tests



Vertical: average relative conflicts after stabilization

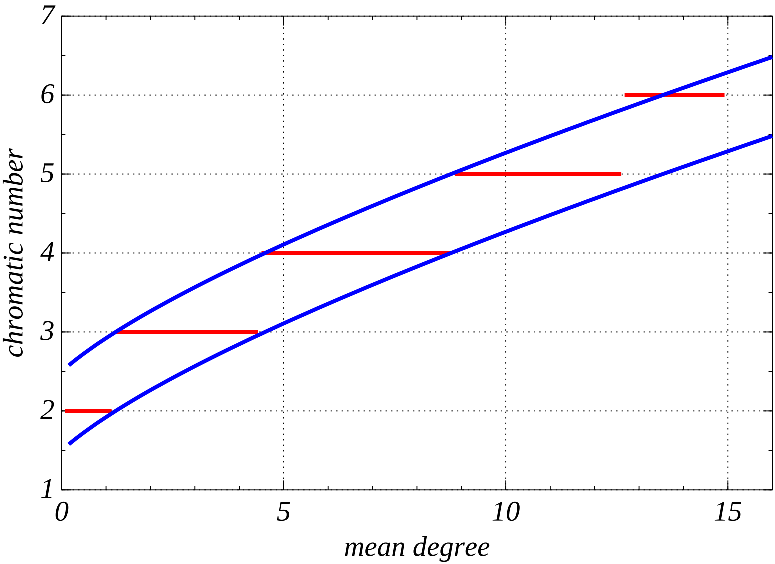
Horizontal: mean degree $\overline{d}$.

Lines: solid $n\!=\!50$, dashed $n\!=\!100$, dotted $n\!=\!1000$.

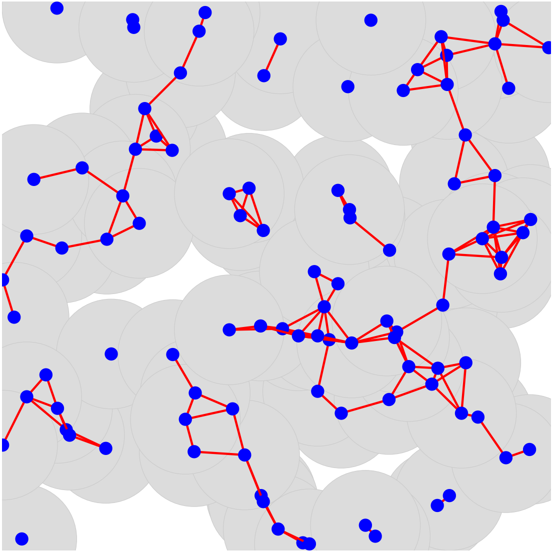
Left: $C\!=\!\chi\!-\!1$, $C\!=\!\chi$, $C\!=\!\chi\!+\!1$.

Right: $C\!=\!2$, $C\!=\!3$, $C\!=\!4$, $C\!=\!5$

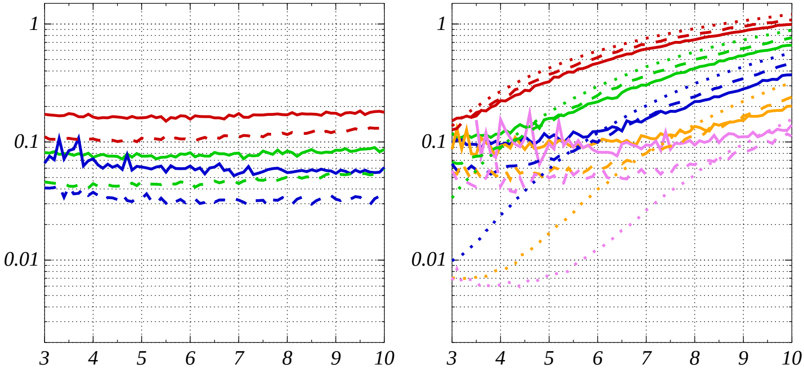
Chromatic number of $G(200, d/200)$



Geometric random graphs



$GRG\{n, \overline{d}\}$ fixed-rate test



Vertical: average relative conflicts after stabilization.

Horizontal: mean degree $\overline{d}$.

Lines: solid $n{=}50$, dashed $n{=}100$, dotted $n{=}1000$.

Left: $C{=}\chi{-}1$, $C{=}\chi$, $C{=}\chi{+}1$.

Right: $C{=}4$, $C{=}5$, $C{=}6$, $C{=}7$, $C{=}8$.

Heuristic VCD

if clock_time$=\tau$ **then**

    col_neigh $\leftarrow$\{colour of neighbours\}

    if col_neigh$=\emptyset$ **then**

        our_color$\leftarrow 0$

        conflicts $\leftarrow$ 0

    else

        for col $\in$colours$\setminus$\{our_color\} **do**

            conflicts $\leftarrow$#conflicts(col, col_neigh)

            if conflicts$=0$ **then**

                our_color$\leftarrow$ col

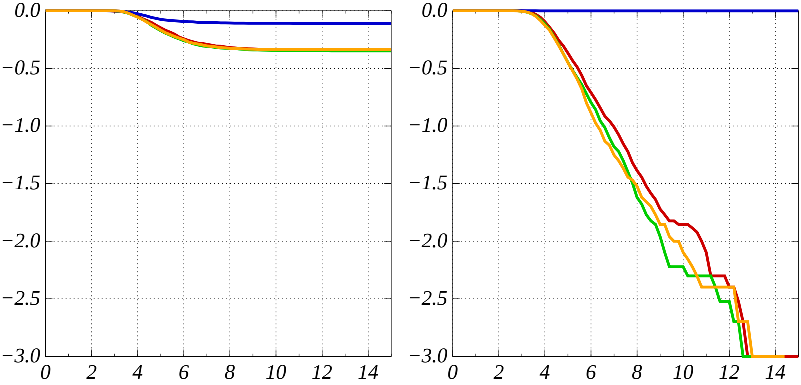
                break for

    $\tau \leftarrow \tau +$expovariate$(1+$conflicts$)$

Heuristic VCS

```
if clock_time=τ then
    col_neigh ←{colour of neighbours}
    if col_neigh=∅  then
        our_color←0
        conflicts ← 0
    else
        for col ∈ colours  do
            conflicts ←#conflicts(col, col_neigh)
            if conflicts=0 then
                our_color← col
                break for
        if conflicts>0 then
            our_color←uniform_discrete[0, C−1]
    τ←τ+expovariate(1+conflicts)
```

Street graph ($n \approx 400$) test



Horizontal: time.

Vertical: $\log(1-$fraction of runs finding a proper coloring$)$.

Left: FCD (check rate $\lambda = 1$), VCD (check rate $\lambda = 1 + \#$conflicts), VCD & terminate after resolving all conflicts, VCD, $\exp(\#$conflicts$)$.

Right: same with stochastic choice (FCS, VCS).

Heuristic VPS

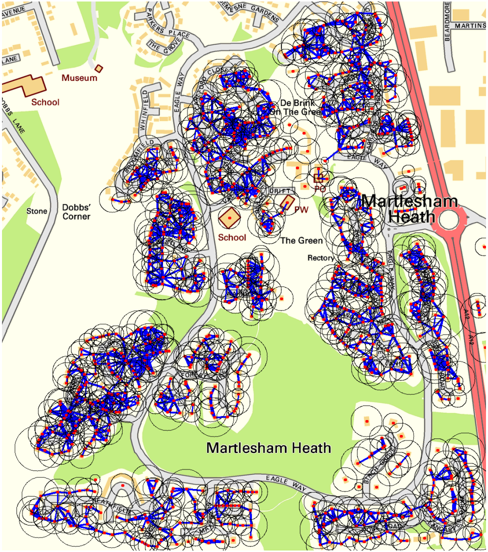
```
if clock_time=τ then
    col_neigh ←{colour of neighbours}
    cur_col ←{our_color}
    cur_conf ←#conflicts(our_color, col_neigh)
    for col ∈colours\{our_color} do
        conflicts ←#conflicts(col, col_neigh)
        if conflicts=0 then
            our_color← col
            break for
        else if conflicts < cur_conf then
            cur_col, cur_conf ← {col}, conflicts
        else if cur_conf = conflicts then
            cur_col ← {cur_col, col}
    if cur_conf>0 then
        our_color←uniform_random{cur_col}
    τ←τ+expovariate(1+#conflicts(our_color, col_neigh))
```

Heuristic VMS

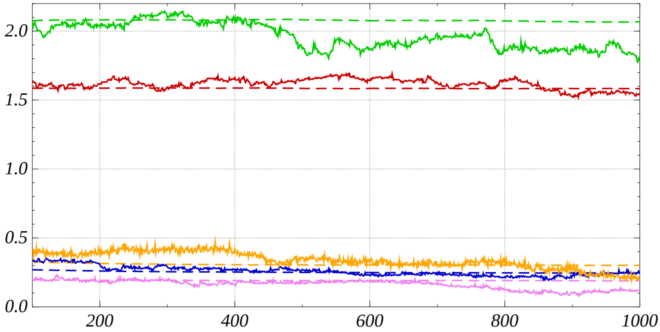
```
if clock_time=τ then
    col_neigh ←{colour of neighbours}
    change ← false
    cur_conf ← #conflicts(our_color, col_neigh)
    for col ∈ colours\{our_color} do
        conflicts ← #conflicts(col, col_neigh)
        if conflicts=0 then
            our_color← col
            break for
        else if conflicts < cur_conf then
            our_color, cur_conf ← col, conflicts
            change ← true
    if not change and cur_conf>0 then
        our_color←uniform_discrete[0, C−1]
    τ←τ+expovariate(1+#conflicts(our_color, col_neigh))
```

Martlesham Heath — matching APs to measurements

Real example — Martlesham Heath



Martlesham Heath



Homehubs are randomly switching on and off.

Vertical: average interference per homehub. FCD with check rate of 1; random colourer; VMS with check rate of λ=conf(i); VPD with check rate of λ=conf(i); VPS with λ=exp(conf(i)).