An introductory look at the antibandwidth problem

Jennifer Scott, STFC Rutherford Appleton Laboratory

in collaboration with Yifan Hu, AT&T

Sparse Days at CERFACS, 6th September 2011

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Outline

- Background: definitions and motivation
- Existing work
- Level-set approach
- Hill climbing for refinement
- Concluding remarks and future work

Bandwidth minimization problem

Minimizing the bandwidth of a sparse symmetric matrix $A = \{a_{ij}\}$ is a well-known problem ie permute the rows and columns of A to minimize the maximum distance b from diagonal

$$b = min_i\{max_j\{|i-j| : a_{ij} \neq 0\}\}$$

Equivalently, label (or order) the nodes of graph $\mathcal{G}(A)$ to minimize the maximum the difference between a node i and its neighbours (minimize the length of the longest edge).

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Bandwidth minimization history

- Bandwidth minimization problem originated in 1950s when structural engineers first analyzed steel frameworks using matrices (wanted small bandwidth to factorize matrix).
- The problem was posed independently for graphs (Harper 1964 and Harary 1967).
- Problem is NP-Complete (Papadimitriou 1976, Johnson, Garey, Graham and Knuth 1978).
- Many papers published in the literature.
- Best-known algorithm for computing a good band ordering is that of Cuthill-McKee (eg `rcm` in MATLAB, `MC61` in HSL).

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Antibandwidth problem

The antibandwidth maximization problem is to permute the rows and columns of A to maximize the minimum distance ab from diagonal

$$ab = max_i\{min_j\{|i-j| : i \neq j \text{ and } a_{ij} \neq 0\}\}.$$

Equivalently, label the nodes of $\mathcal{G}(A)$ such that the length of the shortest edge is maximized.

Science & Technology Facilities Council
Rutherford Appleton Laboratory

A bit of history

- Introduced by Leung, Vornberger, Witthoff in *On some variants of bandwidth minimization problem*, SIAM Journal of Computing 13, 1984.
- Antibandwidth is also known as separation number or dual bandwidth.
- Also NP-Complete.
- Some theoretical results for special graphs.

Applications: frequency assignment problem

Hale, *Frequency assignment: theory and applications*, Proceedings IEEE 60, 1980:

Given n transmitters and n frequencies find a bijective frequency assignment where the interfering transmitters have frequencies that are as different as possible.

Graph model:

- transmitters = nodes
- interferencies = edges between interfering transmitters
- frequency assignment = optimal antibandwidth labelling

Science & Technology Facilities Council
Rutherford Appleton Laboratory

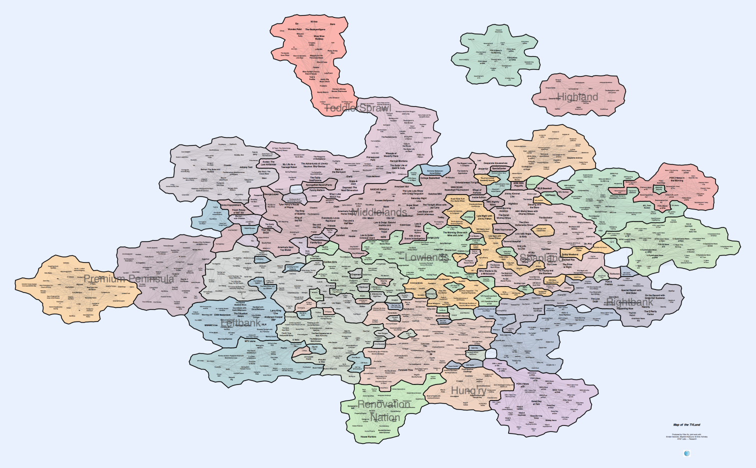
Obnoxious facilities location problem

- Let nodes represent sensitive facilities or chemicals
- **Aim:** locate them as far from each other as possible

Maximum differential graph colouring

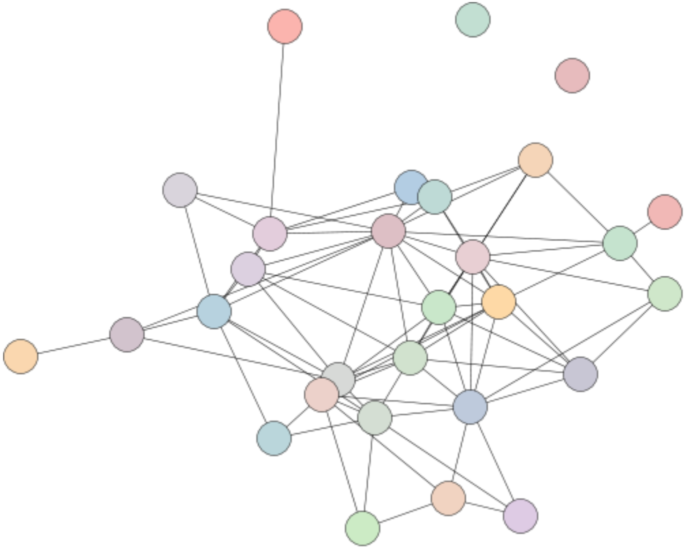
- Four Colour Theorem states only 4 colours needed to colour a map so that no neighbouring countries share same colour.
- **But** assumes each country forms a contiguous region.
- If **not**, need unique colour for each country.
- countries = nodes
- edge between two countries if they share non-trivial boundary
- maximize colour distance between nodes that share an edge = optimal antibandwidth labelling

Map with original colours



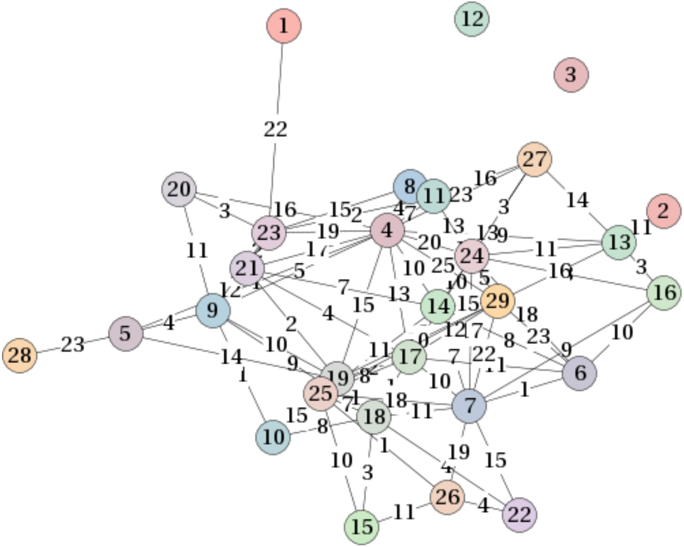
Science & Technology Facilities Council
Rutherford Appleton Laboratory

Representation as graph



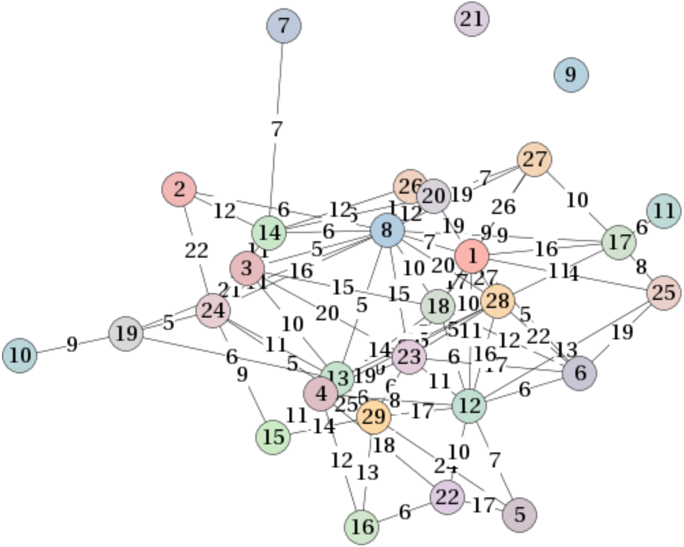
Assign numbers to colours

Random, 2−norm objective fun =23634.



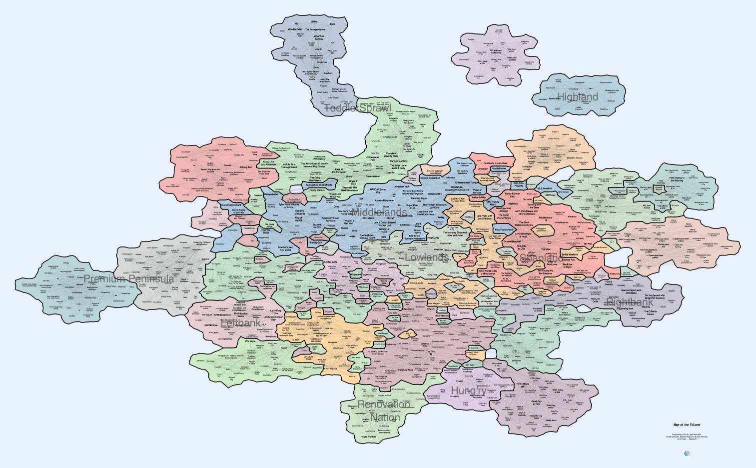
Relabel graph

Spectral+Greedy, 2−norm objective fun = 31446.



Science & Technology Facilities Council
Rutherford Appleton Laboratory

Map with new colours



Science & Technology Facilities Council
Rutherford Appleton Laboratory

Theoretical results

Optimal solutions are known for a few special classes and for others bounds have been proved.

In particular, for a 2D $m \times k$ mesh with $m \geq k \geq 2$

$$ab = \left\lceil \frac{k(m-1)}{2} \right\rceil$$

(Miller and Pritkin 1989, Raspaud et al. 2009).

For 3D $m \times m \times m$ mesh (Török and Vrt'o 2010)

$$ab = \frac{4m^3 - 3m^2}{8} + \mathcal{O}(m).$$

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Numerical methods

Duarte, Marti, Resende and Silva (2010) first proposed heuristics aimed at obtaining high-quality solutions on general graphs.

- Use an integer programming formulation and then CPLEX (optimization package) to solve.
- **But** too expensive ($>$ 24 hours for problems with $n = \mathcal{O}(10)$).
- So then propose metaheuristics based on GRASP (Greedy Randomized Adaptive Search Procedure).
- Each GRASP iteration constructs a trial solution and then applies a local search.

Some good results but still costly ... several minutes for relatively small Harwell-Boeing matrices.

Can we do better? Retain quality but faster orderings?

Can we exploit knowledge and experience from the bandwidth minimization problem to develop algorithms for the antibandwidth maximization problem?

Cuthill-McKee algorithm for bandwidth reduction

- Works with the adjacency graph $\mathcal{G}$ of A.
- $\mathcal{G}$ is an undirected graph that has a node for each row (or column) of the matrix and node i is a neighbour of node j if a_{ij} (and by symmetry a_{ji}) is an entry (nonzero) of A.
- Choose starting node s and relabel nodes of $\mathcal{G}$ by order of increasing distance from s.
- Notes:
  - Assume $\mathcal{G}$ is connected (otherwise, procedure repeated from an s in each component).
  - Reversing the ordering (RCM) reduces the profile of A (the average distance between first entry in a row and the diagonal) but does**not** effect the bandwidth.

Cuthill-McKee algorithm

Choosing starting node s and label as node 1

Set $l_1 = \{s\}$; $i = 1$

do $k = 2, 3, \ldots$ **until** $i = n$

    $l_k = \{\}$

    do for each $v \in l_{k-1}$ in label order

        do for each neighbour u of v that has not been labelled,

                in order of increasing degree

            add u to l_k; $i = i + 1$; label u as node i

        end do

    end do

end do

Science & Technology Facilities Council
Rutherford Appleton Laboratory

- Ordering the nodes in this way groups them into level sets.
- Nodes in level set l_k can have neighbours only in level sets l_{k-1}, l_k, and l_{k+1}.
- Therefore desirable that the level sets be small, which is likely if there are many of them.
- Algorithms for finding a good starting node are usually based on finding a pseudo-diameter of $\mathcal{G}$.

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Level-set approach for antibandwidth problem

Recall: for 2D $m \times k$ mesh with $m \geq k \geq 2$

$$ab = \left\lceil \frac{k(m-1)}{2} \right\rceil .$$

Miller and Pritikin describe how this bound can be achieved.

Science & Technology Facilities Council
Rutherford Appleton Laboratory

25	26	27	28	29	30
19	20	21	22	23	24
13	14	15	16	17	18
7	8	9	10	11	12
1	2	3	4	5	6

Start at a corner: $l_1 = \{1\}$, $l_2 = \{2, 7\}$, $l_3 = \{3, 8, 13\}$, and so on.

25	26	27	28	29	30
19	20	21	22	23	24
13	14	15	16	17	18
7	8	9	10	11	12
1	2	3	4	5	6

Start at a corner: $l_1 = \{1\}$, $l_2 = \{2, 7\}$, $l_3 = \{3, 8, 13\}$, and so on.

Order l_2, l_4, l_6, ..., and then l_1, l_3, ...

20	7	25	12	29	15
3	21	8	26	13	30
17	4	22	9	27	14
1	18	5	23	10	28
16	2	19	6	24	11

Science & Technology Facilities Council
Rutherford Appleton Laboratory

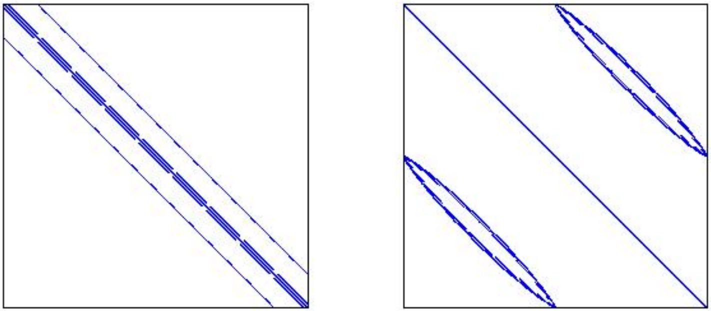
Level-based antibandwidth algorithm

Initialise: $sweep = 0$; $flag(1:n) = 0$; $i = 0$.
Given s, construct $L(s) = \{l_1, l_2, ..., l_h\}$.
do until $i = n$
    $sweep = sweep + 1$
    do $r = 1, ..., h$
        do for each unlabelled $u \in l_r$
            if $(flag(u) = sweep)$ **cycle**
            $i = i + 1$; label u as node i
            Set $flag(v) = sweep$ for each unlabelled neighbour v of u
        end do
    end do
end do

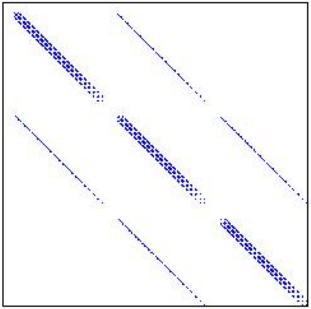
Notes on level-based approach

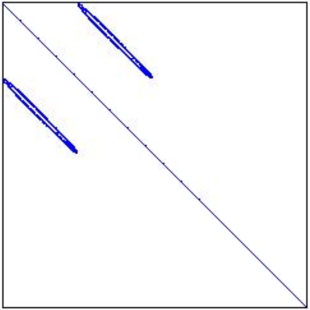
- Algorithm comprises a number of sweeps through level-set structure $L(s)$.
- On each sweep, a node can only be labelled if none of its neighbours has already been labelled on the current sweep.
- On uniform mesh, reduces to ordering the even numbered sets and then the odd numbered set (Miller and Pritikin).
- Success will depend on good starting node and getting well-balanced level sets.

HB/nos7 before and after LB reordering

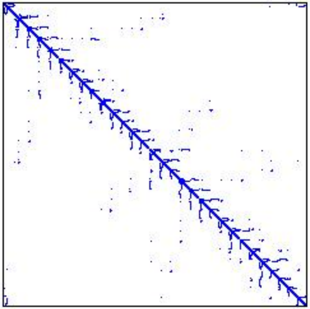


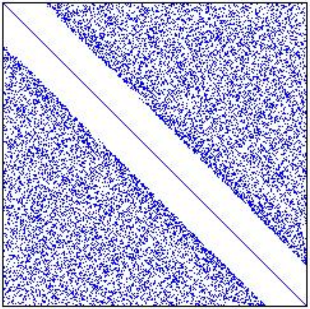
HB/sherman4 before and after LB reordering



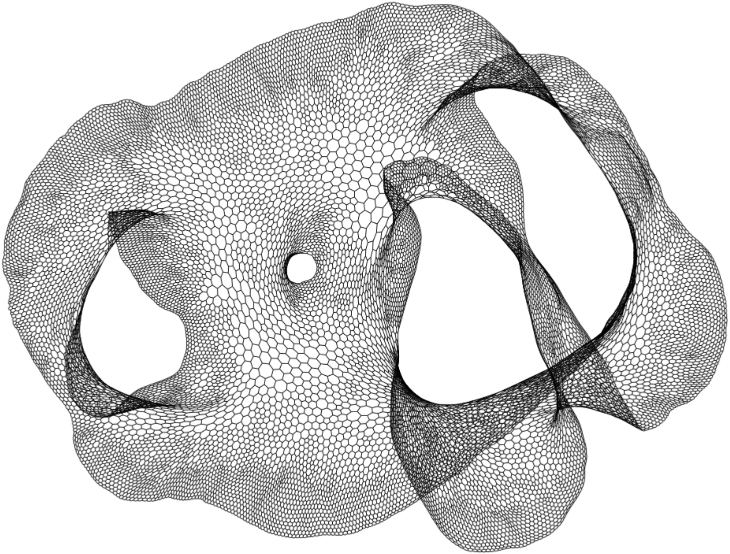


HB/lshp2614 before and after LB reordering

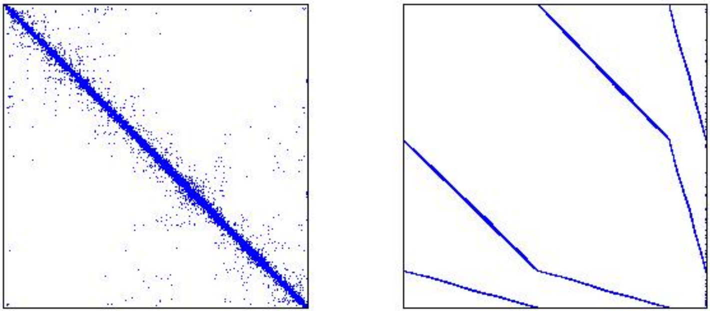




Finite-element mesh for `AG-Monien/big_dual`



AG-Monien/big_dual before and after LB reordering



Story so far ...

- Level-based approach generalises that of Miller and Pritikin.
- It works well on many grid-based problems.
- **BUT** while it increases average distance from the diagonal, can get some rows towards end of ordering where minimum distance to diagonal is very small (possibly even $ab = 1$) eg `AG-Monien/big_dual`.

Can we improve *ab* using a local refinement strategy?

Hill climbing

We define i to be critical if

$$min_k\{|i-k| : i \neq k \text{ and } a_{ik} \neq 0\} = ab.$$

Basic idea: If i is critical, look for non-critical j such that symmetrically permuting i and j (swapping rows i and j and columns i and j) leaves i and j non-critical.

Note: used for bandwidth reduction by Lim et al. (2004) and by Reid and Scott (2006) for unsymmetric matrices.

Hill climbing (HC) algorithm

outer: do
  Form the set V_c of critical nodes
  do until V_c is empty
    if there are nodes $i \in V_c$ and $j \notin V_c$ such that
     swapping i and j leaves both non-critical **then**
     swap i and j and remove i from V_c
    else
     exit outer
    end if
  end do
end do outer

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Hill climbing

Differences between HC for bandwidth and antibandwidth problems:

- definition of a critical node,
- checks that are needed for finding suitable swap.

Looking for swap is simple in bandwidth case (just keep track of first and last entries in each row).

Hill climbing

For antibandwidth, restrict the search eg if row i has a lower critical entry for current antibandwidth ($\exists\ k < i$ such that $i - k = ab$) and j is in range

$$i - 2 * ab \leq j \leq i - 1$$

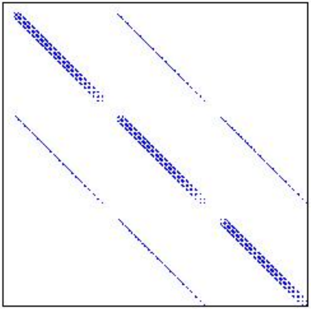
swapping i and j decreases the antibandwidth so reject.

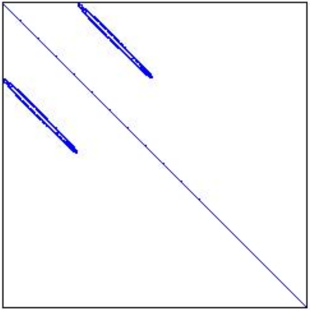
But still necessary to check entries in rows i and j to see if a swap is possible. Not acceptable if

1. $a_{ij} \neq 0$ and $|i - j| = ab$, or
2. $\exists\ l$ such that $a_{il} \neq 0$ and $|l - j| \leq ab$, or
3. $\exists\ k$ such that $a_{kj} \neq 0$ and $|k - i| \leq ab$.

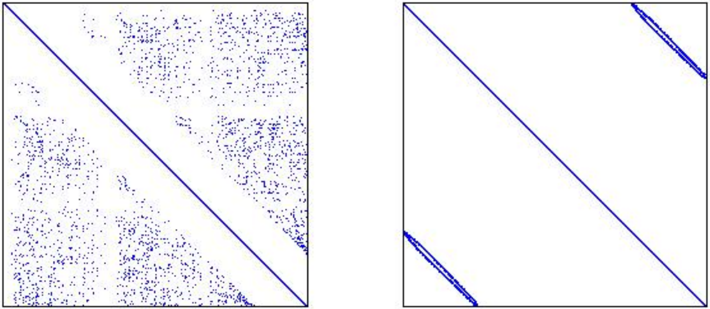
Science & Technology Facilities Council
Rutherford Appleton Laboratory

Recall: HB/sherman4 before and after LB reordering

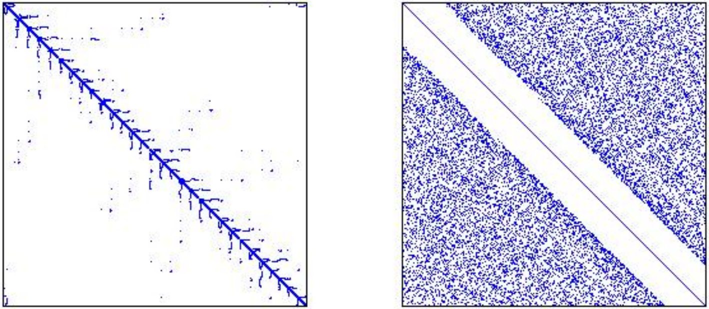




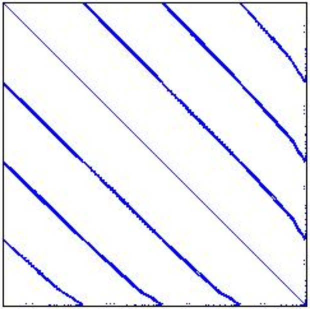
HC applied to HB/sherman4

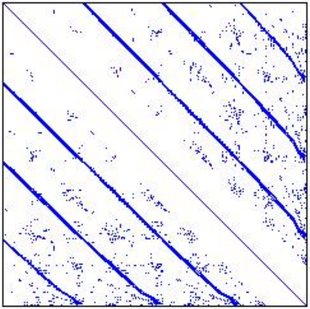


Recall: HB/lshp2614 before and after LB reordering

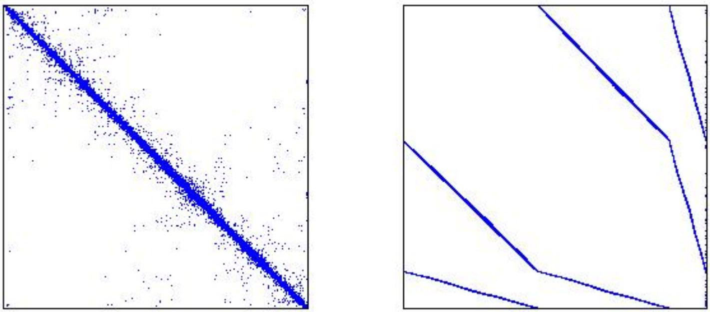


HC applied to HB/lshp2614

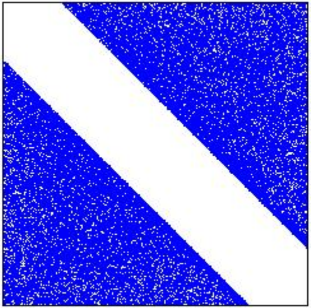


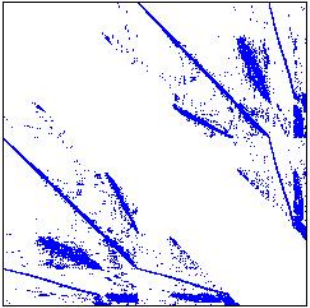


Recall: AG-Monien/big_dual before and after LB reordering



HC applied to AG-Monien/big_dual





Comparisons with GRASP

Problem	n	GRASP	LB+HC
HB/curtis54	54	**12**	9
HB/dwt_234	234	51	**80**
HB/can_445	445	**85**	56
HB/662_bus	662	**220**	163
HB/nos6	675	328	**329**
HB/sherman4	1104	261	**815**

For these small problems, LB+HC takes < 0.01 seconds.
For a problem with $n = 201,822$, time was 100 seconds.

Duarte et al report GRASP requires 300 seconds for HB/sherman4

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Results to date

Bad news:
The antibandwidth maximization problem appears to be tougher than the bandwidth minimization problem.

Some positive results:

- Level-based ordering can work well on grid problems.
- Hill climbing offers local refinement ... final quality depends on initial ordering.
- Hill-climbing is expensive compared with level-based ordering but much faster than GRASP and can be used for larger problems.

Future directions

However, simple 2-step approach is not sufficient for all problems.

Future plans include:

- Developing other algorithms to obtain initial orderings (including a spectral approach and a node centroid algorithm).
- Deriving more sophisticated refinement algorithms.

It seems likely there will always be a compromise between quality and speed.

Thank you!

Note: we are currently looking to recruit a new member of the Numerical Analysis Group at RAL. Contact me if interested.

Science & Technology Facilities Council
Rutherford Appleton Laboratory