The Routing Model

Previous and Related Work

Computational Results

Structural Results

CCPP

New Results On Routing Via Matchings

Indranil Banerjee
with Dana Richards

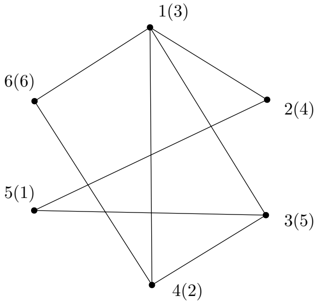
George Mason University

richards@gmu.edu

December 1, 2017

Definitions

- $G(V, E)$ is an undirected graph. $V = \{1, 2, 3, \ldots, n\}$.
- A pebble at vertex i is labeled $\pi(i)$ if it is to be routed to vertex $\pi(i)$, for a given permutation π.
- Permutations written using cycle notation.



$\pi = (135)(24)(6)$

Figure: G with 6 nodes

Definitions

- A matching is a vertex disjoint subset of the edges.
- Swapping pebbles across the matched edges advances to a new permutation (stop at the identity permutation).
- *Routing time*, $rt(G, \pi)$, $\#$ of matchings necessary for π
- The maximum routing time over all permutations is called the *routing number* of G, $rt(G)$.
- If G is not connected, $rt(G) = \infty$

An Example

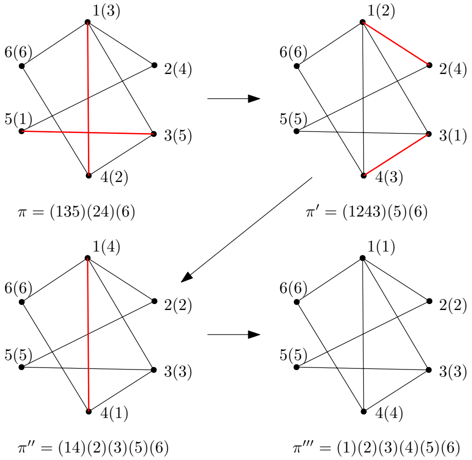


Figure: A 3-step routing scheme for (G, π)

The General Model

- This routing model was first introduced by Alon et. al.(*)
- Which is a special case of the minimum generator sequence (MGS) problem for permutation groups (G).
- Given a set of generators S, the MGS problem asks one to determine the minimum number of generators required to generate every element of G (from the identity element).
- This problem was shown to be PSPACE-complete (even with only generators of order 2).

(*) Alon, N., Chung, F. R., & Graham, R. L. (1994). Routing permutations on graphs via matchings. SIAM J Disc Math, 7(3), 513-530.

Routing Numbers of Familiar Graphs

- Every connected graph, has a spanning tree.
- Trivially, we can pick a pebble whose destination is some leaf vertex.
- Move it to its destination sequentially, then solve for the rest of the tree independently. Takes $O(n^2)$ steps.
- However we can do it faster ($O(n)$).

Tree Routing

First partition the spanning tree around its centroid.

1. Route between the subtrees through the centroid using a matching chosen based on a simple odd-even greedy strategy.
2. Then route within the subtrees recursively (in parallel).

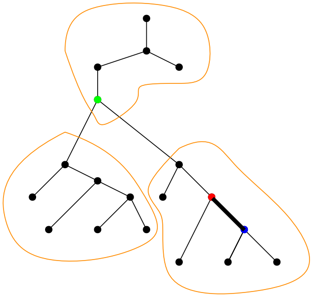


Figure: This strategy gives a $\leq 3n$ routing scheme

Tree Routing

- Current best upper bound for any tree is $3n/2 + O(\log n)$.
- The best lower bound of $\lceil 3n/2 \rceil + 1$ is for the start graph.

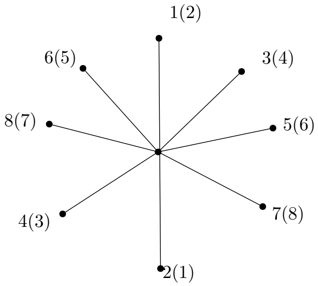


Figure: A matching is just a singleton edge, the permutation $\pi = (12)(34)\dots(2m-1, 2m)$, $n = 2m$ takes $\lceil 3n/2 \rceil + 1$ steps.

Routing Numbers of Familiar Graphs

- $rt(P_n) = 2\lfloor n/2 \rfloor$ (path graph).
- $rt(K_n) = 2$ (complete graph)
- $rt(K_{n,n}) = 4$ (complete bipartite graph)
- $rt(Q_n) \leq 2n - 3$ (the n-cube with 2^n vertices)
- $rt(M_{n,n}) = O(n)$ ($n \times n$ mesh)
- If G is a bounded degree expander then $rt(G) = O(\log^2 n)$

Hypercube

- It is known that:

$$rt(G\Box H) \leq 2\min(rt(G), rt(H)) + \max(rt(G), rt(H))$$

- Since $Q_n = K_2 \Box Q_{n-1}$
- The upper bound $rt(Q_n) \leq 2n - 3$ follows. (the n-cube with 2^n vertices)
- It is also the best known.
- Lower bound $\geq n + 1$
- It has been conjectured that $rt(Q_n) \leq n + 1 + o(n)$.

Hypercube

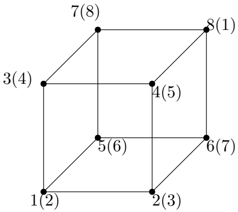


Figure: A bad permutation. The cycle crosses many non-adjacent vertices.

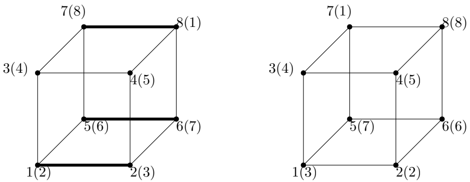


Figure: Step - 1

Hypercube

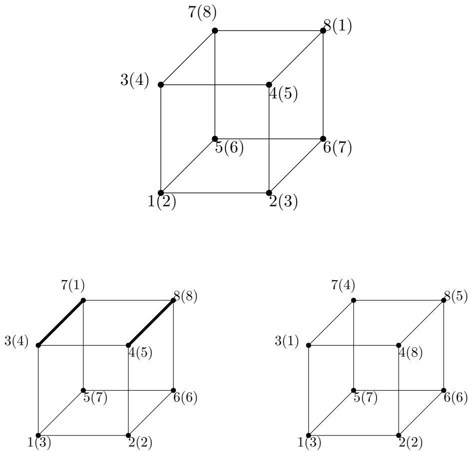


Figure: Step - 2

Hypercube

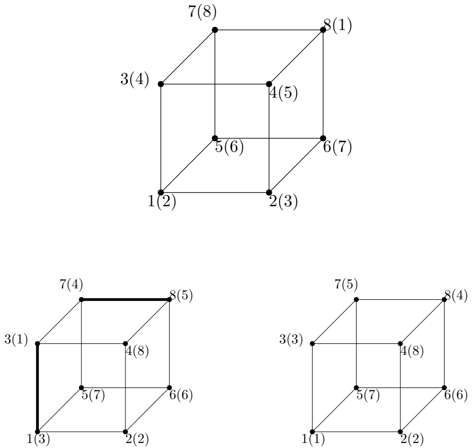


Figure: Step - 3

Hypercube

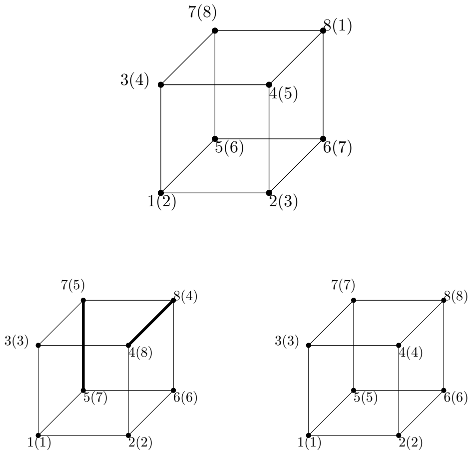


Figure: Step - 4

Computational Results

Our results:

- Deciding if $rt(G, \pi) \leq 2$ can be done in polynomial time
- Determining $rt(G, \pi)$ is NP-complete
- It remains so when G is 2-connected and π is an involution

Later we show

- Decision version of MaxRoute is also NP-complete
- Connected colored partition problem (CCPP) is NP-complete
- An $O(n \log \log n / \log n)$-approximation algorithm for MaxRoute on a degree bounded graph.

Is $rt(G,\pi) \leq 2$?

$G[V_c]$ = induced subgraph over the vertices in cycle c
"Self-routing" a cycle c of π uses only using $G[V_c]$ in two steps.

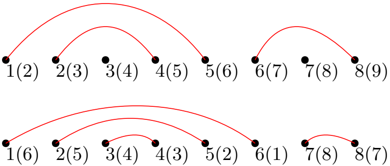


Figure: One way to route a simple cycle $c = (12345678)$ in two steps. There are 8 possible ways on a complete graph

For a sparser graph there may not be 8 options.
Can determine if there is at least one way in linear time.

Is $rt(G, \pi) \leq 2$? Contd.

"Mutual routing" of a pair of cycles c_1, c_2 in π uses only edges of the induced bipartite subgraph $G[V_{c_1}, V_{c_2}]$, in two steps.

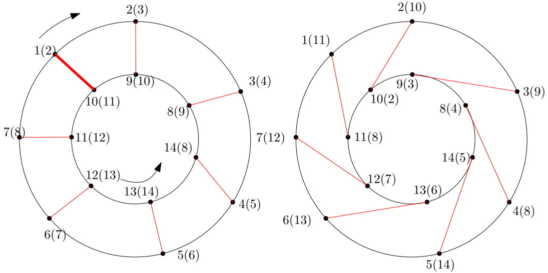


Figure: One way to route two cycles $c_1 = (1\ 2\ 3\ 4\ 5\ 6\ 7)$ and $c_2 = (8\ 9\ 10\ 11\ 12\ 13\ 14)$ in two steps.

Can determine if there is at least one way in linear time.

Is $rt(G, \pi) \leq 2$? Contd.

1. For each cycle we can determine if it can be self-routed
2. For each pair we can determine their mutual-routability
3. Create a graph G_{cycle} with:
   - a vertex for each cycle of π
   - edges and self-loops for mutual- and self-routability
4. Then $rt(G, \pi) = 2$ iff G_{cycle} has a perfect matching.
5. All this can be carried out in the time it takes compute a maximum matching.

Hardness Proof: Reduction from 3-SAT

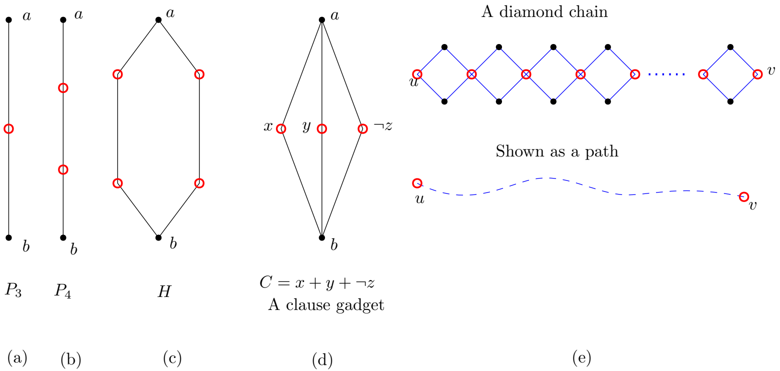


Figure: The involution (ab) takes at least three steps to route for the graphs in figures (a)-(d)

A clause can be routed in 3 steps iff a vertex from $\{x, y, z\}$ is available, i.e. not used to route any other pebbles.

Hardness Proof Contd.

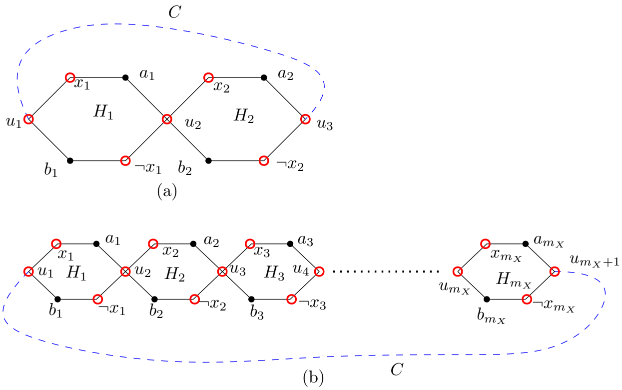


Figure: Variable gadget.

Where the variable X is in $m_X =$ clauses.

Hardness Proof Contd.

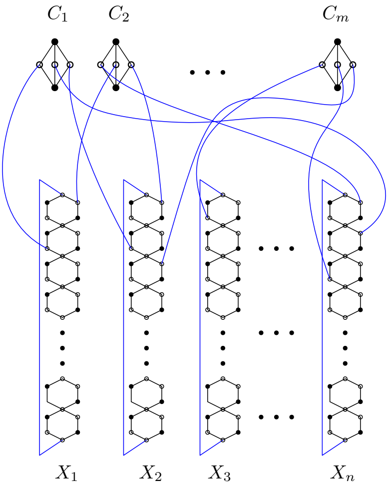


Figure: The entire G_ϕ that is built.

Hardness Proof: Observations

- $rt(G_\phi, \pi) = 3$ iff ϕ is satisfiable.
- The graph G_ϕ built in the reduction is 2-connected.
- The permutation π in the reduction is an involution.

The other hardness proof in this work extend this reduction.

Approximate/Partial Routing

Define the MaxRoute problem (partial routing) as follows:

- Given a graph G, a permutation π and number of steps k route the most pebbles to their destination within k steps.
- $mr(G, \pi, k)$ is the max number of pebbles routed.
- The decision version of this problem is to determine if $mr(G, \pi, k) \geq t$.

Approximating MaxRoute

We give an approximation algorithm for the restricted case where $\Delta^k = O(\log^2 n)$, $\Delta =$ max degree of G.

- Our approximation algorithm is based on a reduction to the MaxClique problem.
- The best known approximation factor for MaxClique is $O(n \log\log n / (\log n)^3)$

Approximating Contd.

1. We enumerate all walks of length k for each pebble on G.
2. A pair of walks is “compatible” if:
   a. The walks belong to different pebbles.
   b. They do not intersect (same place at the same time).
   c. The pebbles reach their destinations at the end.
3. Build graph G' with a vertex for each walk and edges for compatible pairs

A clique in G' gives a set of mutually compatible walks.

Structural Results

Three structural results

- If G is a h-connected graph and H is any h-vertex induced subgraph of G then $rt(G) = O((n/h)rt(H))$.
- If G has a clique of size at least κ then $rt(G) = O(n - \kappa)$.
- Routing number of the pyramid graph $\triangle_{m,d}$ is $O(dN^{1/d})$

$$N = \frac{2^{md} - 1}{2^d - 1}$$

h-Connectivity

- Let A, B be a bi-partition of V for some min-cut of size h.
- Then it takes at least $\Omega(\min(|A|, |B|)/h)$ to move all pebbles between A and B.
- For some graphs this is $\Omega(n/h)$.

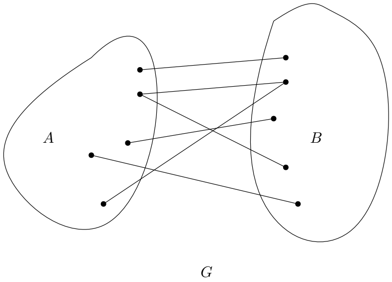


Figure: Lower bound.

h-Connectivity, Contd

The Gyori-Lovasz theorem: for all h-connected graphs and for any set of h vertices there is a partition:

- Where each of the h vertices is in a distinct block,
- We can insist the size of the blocks are nearly equal,
- Each block induces a connected subgraph.

This set of h vertices will induce a subgraph H of G.
We can assume H is a subgraph which minimizes $rt(H)$.

h-Connectivity, Contd

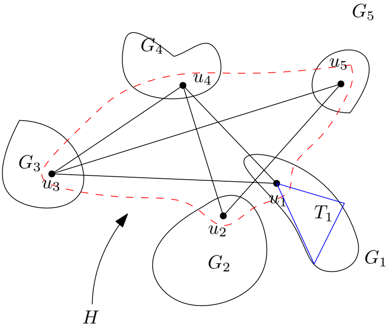


Figure: A partition of G, with $h = 5$. Since each induced subgraph G_i is connected, there is a spanning tree T_i of G_i rooted at u_i.

h-Connectivity

Let each G_i have a distinct "color".

- Each pebble knows the color of its destination block.
- By Hall's theorem there is a set of permutations $\pi_1, \pi_2, \ldots, \pi_h$, one for each subgraph, such that each $(\pi_1(i), \pi_2(i), \ldots, \pi_h(i))$ contains h distinct colors.
- Hence each $(\pi_1(i), \pi_2(i), \ldots, \pi_h(i))$ is a permutation which we can route using only H in $rt(H)$ steps.

h-Connectivity: Routing Algorithm

Routing proceeds in three stages

1. During the first stage we move pebbles within each T_i according to π_i. (This takes $O(n/h)$ steps in parallel)
2. We use H to route pebbles between the connected blocks using colors, n/h times. ($O((n/h)rt(H))$ steps)
3. Finally we move pebbles within each T_i to their final position. ($O(n/h)$ steps)

Conjecture

If G is h-connected then there is a H (as above) having $g(h)$ vertices with $rt(H)/g(h) = o(1)$.

Routing and Clique Number

- Recall that $rt(K_n) = 2$.
- Intuitively having a large clique should results in a smaller routing number
- However this dependency is not multiplicative:

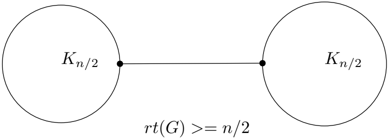


Figure: The barbell graph, although it has two large cliques, its routing number is still $\Omega(n)$

So there is a $\Omega(n - \kappa)$ bound for such graph families.

Routing and Clique Number, Contd

- Let H be a clique of size κ
- $G_{\backslash H}$ is the minor of G after contracting H to the vertex v
- T is a spanning tree of $G_{\backslash H}$

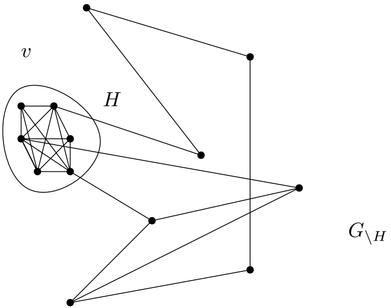


Figure: The (super) vertex v acts as any other vertex in $G_{\backslash H}$, with the exception that pebbles exchanges takes three time steps.

Routing and Clique Number, Contd

1. In the first stage we route all pebbles that belong in the super vertex v into v. (Takes at most $3(n-\kappa)+O(1)$ steps).
2. Next we route the pebbles within T, treating v as any other vertex, using any optimal tree routing algorithm. (Takes $\leq 3(3/2)(n-\kappa)+o(n)$)
3. Finish up within v in two steps.

Hence it takes $O(n-\kappa)$ steps to route any permutation on G.

Routing Number of △

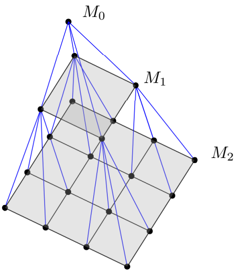


Figure: A pyramid $△_{3,2}$ with 3 layers.

Routing Number of △

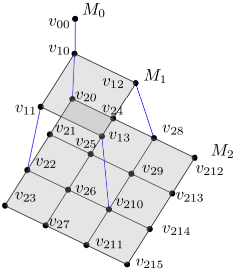


Figure: A multi-grid formed after stripping way some edges from $\triangle_{3,2}$

Use vertical paths of length k to move pebbles up to level k (from the base).

CCPP

Connected Colored Partition Problem

This arises in the analysis of some approximation algorithms.

Given a graph G and a vertex coloring with at most k colors, the problem asks whether there is a partition of the vertices such the following holds:

- Each block of the partition induces a connected subgraph.
- No color spans two blocks.
- Each block is of size $\leq p$

CCPP, Contd

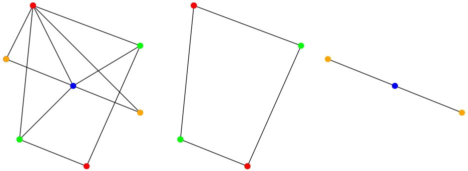


Figure: An example using two blocks.

- We reduce from 3-SAT.
- The reduction is similar to the routing time proof.
- If (ab) is a 2-cycle of π then the vertices corresponding to a, b are assigned the same color.
- Vertices with fixed pebbles are assigned a unique color.

Questions?