NC Algorithms for Popular Matchings in One-Sided Preference Systems

Changyong Hu, Vijay K. Garg
Department of Electrical and Computer Engineering
University of Texas at Austin
colinhu9@utexas.edu, garg@ece.utexas.edu

Popular matching

- For a vertex v, we say v prefers M to M' if v prefers $p_M(v)$ to $p_{M'}(v)$.
- $P(M, M')$ = the set of vertices that prefers M to M'. (Voting over matchings)
- "more popular than" relation $\succ$ on M:

$$M' \succ M \text{ if } |P(M', M)| > |P(M, M')|.$$

Definition

A matching M is popular if there is no matching M' such that $M' \succ M$.

Housing allocation model

- The stable matching problem only makes sense in two-sided preference systems.
- The housing allocation model:
  - Bipartite graph G
  - $\mathcal{A} \cup \mathcal{H}(\mathcal{P})$ = set of agents(applicants) and set of houses(posts)
  - E = set of edges
  - for each agent in $\mathcal{A}$, there is a strictly ordered preference lists over acceptable houses. Houses do not have preferences over agents.
- We will refer houses as posts and agents as applicants.

NC model

- The set of decision problems decidable in polylogarithmic time on a parallel computer with a polynomial number of processors.
- Equivalently, a problem is in NC if there exist constants c and k such that it can be solved in time $O(log^c n)$ using $O(n^k)$ parallel processors.
- $NC \subseteq P$

Characterizing popular matchings

- f-post: for each agent a, let $f(a)$ denote the first-ranked post on a's preference list.
- s-post: let $s(a)$ be the first non-f-post on a's preference list.
- l-post: last-resort post.

a_1 :	p_1	p_4	p_5	p_2	p_6	l_1		
a_2 :	p_4	p_5	p_7	p_2	p_8	l_2		
a_3 :	p_4	p_1	p_3	p_8	l_3			
a_4 :	p_1	p_7	p_4	p_3	p_9	l_4		
a_5 :	p_5	p_1	p_7	p_2	p_6	l_5		
a_6 :	p_7	p_6	l_6					
a_7 :	p_7	p_4	p_8	p_2	l_7			
a_8 :	p_7	p_4	p_1	p_5	p_9	p_3	l_8	

Figure: A popular matching instance I

Characterizing popular matchings

Theorem (Abraham, Irving, Kavitha and Mehlhorn)

A matching M is popular if and only if

1. *every f-post is matched in M, and*
2. *for each applicant a, $M(a) \in \{f(a), s(a)\}$.*

High level ideas

- For each applicant a, only need to consider a's f-post and s-post. Hence we can obtain a reduced graph that is sparse.
- Find a matching that is complete for the applicants.
- Locally rearrange to make sure every f-post is matched.

Algorithm: popular matching

Algorithm 1: Popular Matching

Input: Graph $G = (\mathcal{A} \cup \mathcal{P}, E)$.
Output: A popular matching M or determine that no such matching.

G' := reduced graph of G;
if G' admits an applicant-complete matching M then
    for each f-post p unmatched in M in parallel do
        let a be any applicant in $f^{-1}(p)$;
        promote a to p in M;
    return M;
else
    return "no popular matching";

Algorithm: Applicant-Complete Matching

- Finding an applicant-complete matching sequentially is easy through augmenting path.
- The degree of each agent is exactly two.
- Finding maximal paths can be done in NC.

Algorithm: Applicant-Complete Matching

Algorithm 2: Applicant-Complete Matching

```
Input: Graph G′ = (𝒜 ∪ 𝒫, E′).
Output: An applicant-complete matching M or determine that no such
matching exists.
M := ∅;
while some post p has degree 1
    For all such p, find maximal paths that end at p;
    for each edge (p′, a′) at an even distance from some p in parallel do
        M := M ∪ {(p′, a′)};
        G′ := G′ − {p′, a′};
end while
for each post p has degree 0 in parallel do
    G′ := G′ − p
// Every post now has degree at least 2;
// Every applicant still has degree 2;
if |𝒫| < |𝒜| then
    return "no applicant-complete matching";
else
    // G′ decomposes into a family of disjoint even cycles
    M′ := any perfect matching of G′;
    return M ∪ M′;
```

Complexity

Lemma

The while loop (line 4 in Algorithm 2) runs $O(\log(n))$ number of rounds.

Proof.

- In round r, suppose we have t vertices of degree 1.
- Such vertices have degree at least 3 in round $r-1 \implies \geq 2t$ vertices are deleted.
- Totally $\geq (2^r - 1)t$ vertices deleted $\implies$ at most $\lceil \log(n) \rceil + 1$ rounds.

□

Theorem

There is an NC algorithm to find a popular matching, or determine that no such matching exists.

Maximum-cardinality popular matching

- Popular matchings may have different sizes.
- Switching graph captures all the possible popular matchings.

Switching graph

Definition

Given a popular matching M, the switching graph G_M of M is a directed graph with a vertex for each post p, and a directed edge (p_i, p_j) for each agent a, where $p_i = M(a)$ and $p_j = O_M(a)$. $O_M(a)$ is the unmatched post in $\{f(a), s(a)\}$.

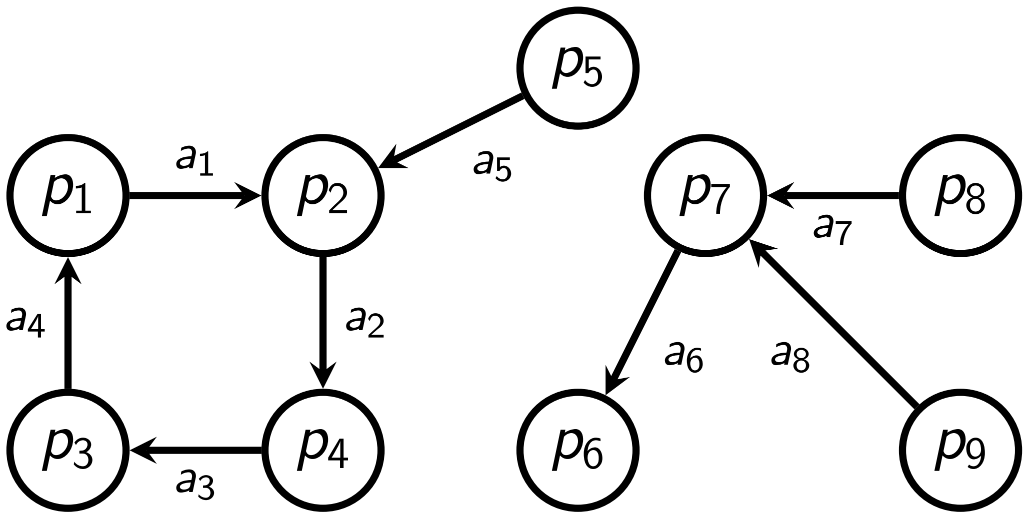

Figure: The switching graph G_M for popular matching M

Switching graph

Lemma (McDermid and Irving, Lemma 1)

Let M be a popular matching for an instance of $G = (\mathcal{A} \cup \mathcal{P}, E)$, G_M be the switching graph of M. Then

1. *Each vertex in G_M has outdegree at most 1.*
2. *The sink vertices of G_M are those vertices corresponding to posts that are unmatched in M, and are all s-post vertices.*
3. *Each component of G_M contains either a single sink vertex or a single cycle.*

Cycle components and tree components

- A component of a switching graph G_M is called a *cycle component* if it contains a cycle, and a *tree component* if it contains a sink vertex.
- Each cycle in G_M is called a *switching cycle*.
- If T is a tree component of G_M with sink vertex p, and if q is another s-post vertex in T, the unique path from q to p is called a *switching path*.
- Note that each cycle component of G_M has a unique switching cycle, but each tree component may have zero or multiple switching paths.

Findings cycles in Pseudoforest

Definition

A pseudoforest is an undirected graph in which every connected component has at most one cycle. A directed pseudoforest is a directed graph in which each vertex has at most one outgoing edge, i.e., it has outdegree at most one.

Transitive closure

Theorem (Hirschberg)

The transitive closure of a directed graph with n vertices can be computed in $O(\log^2 n)$ time, using $O(n^\omega \log n)$ operations on a CREW PRAM, where n^ω is the best known sequential bound for multiplying two $n \times n$ matrices over a ring.

We compute the transitive closure G_P^* and for any two vertices i and j s.t. $i \neq j$ in G_P, if $G_P^*(i,j) = 1$ and $G_P^*(j,i) = 1$, then both i and j are in the unique cycle C. Hence we can identify the cycle C by checking each pair of vertices in parallel.

Algorithm: maximum-cardinality popular matching

Definition

Let Δ be the margin of applying a switching cycle C (resp. switching path P) to M, i.e.

$$\Delta = \sum_{a \in C(\text{resp.} P)} \mathbb{1}_{M \cdot C(a)} - \mathbb{1}_{M(a)}$$

where $\mathbb{1}_p$ is an indicator function of posts

s.t. $\mathbb{1}_p := \begin{cases} 1 & \text{if } p \text{ is not } l\text{-post} \\ 0 & \text{if } p \text{ is } l\text{-post} \end{cases}$

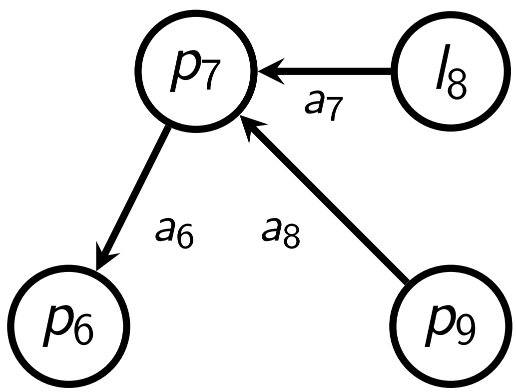

Figure: A tree component in switching graph

Algorithm: maximum-cardinality popular matching

Algorithm 3: Maximum-Cardinality Popular Matching

Input: Reduced graph $G' = (\mathcal{A} \cup \mathcal{P}, E')$ and a popular matching M.
Output: A maximum-cardinality popular matching M'.

$G_M :=$ switching graph of M and G'.
Find all weakly connected components of G_M;
for each cycle component in parallel do
    Find the unique switching cycle;
for each switching cycle in parallel do
    Compute the margin of applying this switching cycle;
for each cycle component in parallel do
    if the margin Δ of switching cycle is positive
        Apply this switching cycle to M;
return M';

Correctness

Theorem (McDermid and Irving, Corollary 1)

Let the tree components of G_M be $T_1, \cdots, T_k$, and the cycle components of G_M be $C_1, \cdots, C_l$. Then the set of popular matchings for G consists of exactly those matchings obtained by applying at most one switching path in $T_i$and by either applying or not applying the switching cycle in C_i.

- Any popular matching can be obtained from M by applying at most one switching cycle or switching path per component of the switching graph G_M.

Theorem

For each tree component T, applying the switching path in T with the largest positive margin; similarly, for each cycle component C, applying the switching cycle in C with positive margin, the new matching is the maximum-cardinality popular matching.

Complexity

- The switching graph G_M can be constructed from G' and M in constant time in parallel.
- All weakly connected components of G_M can also be found in polylog time.
- All switching cycles and switching paths can be found in polylog time. Each switching cycle and switching path can be applied to matching M easily in parallel since they are vertex-disjoint in G_M.

Theorem

There is an NC algorithm to find a maximum-cardinality popular matching, or determine that no such matching exists.