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 $p_{M'}(v)$.
- $P(M, M') = \text{the set of vertices that prefers } M \text{ to } M'. \text{ (Voting over matchings)}$
- “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 $M' \succ M$. 
The stable matching problem only makes sense in two-sided preference systems.

The housing allocation model:
- Bipartite graph $G$
- $\mathcal{A} \cup \mathcal{H}(P) = \text{set of agents(applicants) and set of houses(posts)}$
- $E = \text{set of edges}$
- 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.

**Figure**: A popular matching instance $I$

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

1. Input: Graph $G = (A \cup P, E)$.
2. Output: A popular matching $M$ or determine that no such matching.
3. $G' :=$ reduced graph of $G$;
4. if $G'$ admits an applicant-complete matching $M$ then
5. for each $f$-post $p$ unmatched in $M$ in parallel do
6. let $a$ be any applicant in $f^{-1}(p)$;
7. promote $a$ to $p$ in $M$;
8. return $M$;
9. else
10. 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 2: Applicant-Complete Matching

1. Input: Graph $G' = (A \cup \mathcal{P}, E')$.
2. Output: An applicant-complete matching $M$ or determine that no such matching exists.
3. $M := \emptyset$;
4. while some post $p$ has degree 1
   5. For all such $p$, find maximal paths that end at $p$;
   6. for each edge $(p', a')$ at an even distance from some $p$ in parallel do
      7. $M := M \cup \{ (p', a') \}$;
      8. $G' := G' - \{ p', a' \}$;
   9. end while
10. for each post $p$ has degree 0 in parallel do
11.   $G' := G' - p$
12. // Every post now has degree at least 2;
13. // Every applicant still has degree 2;
14. if $|\mathcal{P}| < |A|$ then
15.   return “no applicant-complete matching”;
16. else
17.   // $G'$ decomposes into a family of disjoint even cycles
18.   $M' :=$ any perfect matching of $G'$;
19.   return $M \cup M'$;
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 $\{f(a), s(a)\}$.

![Switching graph](image)

**Figure:** The switching graph $G_M$ for popular matching $M$.
Lemma (McDermid and Irving, Lemma 1)

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

- Each vertex in $G_M$ has outdegree at most 1.
- The sink vertices of $G_M$ are those vertices corresponding to posts that are unmatched in $M$, and are all $s$-post vertices.
- Each component of $G_M$ contains either a single sink vertex or a single cycle.
A component of a switching graph $G_M$ is called a \textit{cycle component} if it contains a cycle, and a \textit{tree component} if it contains a sink vertex.

Each cycle in $G_M$ is called a \textit{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 \textit{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 $n^\omega$ 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 $\Delta$ 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 3: Maximum-Cardinality Popular Matching

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

1. $G_M :=$ switching graph of $M$ and $G'$.
2. Find all weakly connected components of $G_M$.
3. for each cycle component in parallel do
4.   Find the unique switching cycle;
5. for each switching cycle in parallel do
6.   Compute the margin of applying this switching cycle;
7. for each cycle component in parallel do
8.   if the margin $\Delta$ of switching cycle is positive
9.     Apply this switching cycle to $M$;
10. return $M'$;
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.
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.