POPULAR MATCHINGS

D.J. Abraham, R.W. Irwing,T.Kavitha,K.Melhorn

2007, Society for Industrial and Applied Mathematics

November 1, 2019

Content

Problem Statement

Problem Statement

Input

A bipartite graph G = (A U P, E). Set A : Applicants , Set P: Posts.
Each applicant has a preference list, ranking a nonempty subset of posts.

Some Definitions

- We say that an applicant a prefers matching M' to M if
  (i) a is matched in M' and unmatched in M or
  (ii) a is matched in both M' and M and a prefers M'(a) to M(a).
- M' is more popular than M, by M > M', if the number of applicants that prefer M' to M exceeds the number of applicants that prefer M to M'.
- A matching M is popular iff there is no matching M' that is more popular than M'.

Output

Output the maximum cardinality popular matching, if one exists for the instance.
Else Output No. (Does it always exist?)

Problem Statement

Applications

Some real-world markets, allocation of graduates to training positions, families to government-owned housing etc.

Some Previous works

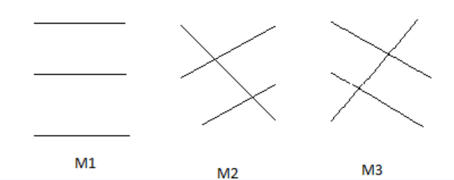
- Gardenfors showed that, when preference lists are strictly ordered, every stable matching is popular.
- He also showed that, when preference lists contain ties, there may be no popular matching.
- No polynomial time algorithms were known previously.

Problem Statement

Existence

Consider the below example:
a1 : p1 p2 p3
a2 : p1 p2 p3
a3 : p1 p2 p3

Matchings in the above example



Matchings

Since M1 < M2 , M2 < M3 , and
M3 < M1, none of these is popular.
This says that the "more popular than " relation is not transitive.

Problem Statement

Input Adjustments and Assumptions

We create a last resort post l(a) for each applicant a and assign the edge (a, l(a)) higher rank than any edge incident on a.
We also assume that ranks are ordered without gaps.

What are we going to see

An O(m+n) time algorithm if the input preference lists are strictly ordered.
If there are ties then an O(m$\sqrt{n}$) *algo*.

Observation

By the above mentioned definition of popular matching,potentially requires an exponential number of comparisons to even check that a given matching is popular.
So,let us find some interesting properties of such matchings.

Matchings with strict preference lists

Some more Definitions

f-posts and s-posts

- For each applicant a, let f(a) denote the first-ranked post on a's preference list. We call any such post p an f-post and denote by f(p) the set of applicants a for which f(a) = p.
- For each applicant a, let s(a) denote the first non-f-post on a's preference list. We call any such post p an s-post and remark that f-posts are disjoint from s-posts.

Example

Consider the example below:
a1 : p1 p2 p3 l1
a2 : p1 p5 p4 l2
a3 : p2 p1 p3 l3
a4 : p2 p3 p6 l4
a5 : p2 p6 p4 l5
a6 : p3 p2 p5 l6
Find f(a) and s(a) for all a's?

Characterization of Popular matchings

Important property

A matching M is popular if and only if
(i) every f-post is matched in M, and
(ii) for each applicant a, M(a) = f(a) or M(a) = s(a).

Sketch of proof

Necessity:
This can be proved by the following lemmas which can be proved easily by contradiction.Let M be any popular matching. Then,

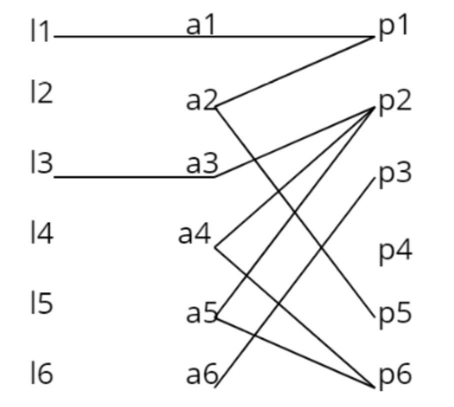
1. For every f-post p, (i) p is matched in M, and (ii) f(p) contains M(p).
2. For every applicant a, M(a) can never be strictly between f(a) and s(a) on a's list.
3. For every applicant a, M(a) is never worse than s(a) on a's preference list.

Reduced Graph

Construction

The reduced graph G' is the subgraph of G containing two edges for each applicant a: one to f(a) and the other to s(a).
(Does G' need to admit an applicant complete matching??).

Reduced Graph for the example



Motivation towards algorithm

Theorem

M is a popular matching of G if and only if
(i) every f-post is matched in M, and
(ii) M is an applicant-complete matching of the reduced graph G'

For the example

We can see that there are 4 popular matchings possible.
M1 = $\{(a1, p1), (a2, p5), (a4, p2), (a5, p6), (a6, p3)\}$
$M2 = \{(a1, p1), (a2, p5), (a4, p6), (a5, p2), (a6, p3)\}$
$M3 = \{(a2, p1), (a4, p2), (a5, p6), (a6, p3)\}$
$M4 = \{(a2, p1), (a4, p6), (a5, p2), (a6, p3)\}$

Algorithm

Finding Popular Matching

1. G' := reduced graph of G

2.If G' do not admit an applicant-complete matching M

return “no popular matching”

3.Else

for each f-post p unmatched in M

- let a be any applicant in f(p)
- promote a to p in M

return M

Algorithm

Finding applicant complete matching in G'

M := Empty Set;
while some post p has degree 1

- a := unique applicant adjacent to p;
- M := M U (a, p);
- Remove a and p from G'

while some post p has degree 0

- Remove p from G'
  // 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 cycles
M' := any maximum-cardinality matching of G';
return M U M'

Analysis

Analysis

- The reduced graph G' of G can be constructed in O(n + m) time.
- As degree of a vertex is atmost 2, the loop requires O(n) time.
- The task remains, is to find max. cardinality matching in G' in linear time. (How about Hopcroft-Karp??)
- In the last part,
  Now, if |P| < |A|, G' cannot admit an applicant-complete matching by Hall's marriage theorem.
- Otherwise, we have that |P| >= |A|, and 2|P| <= degree sum =2|A|. Hence, it must be the case that |A| = |P|, and every post has degree exactly 2. G' therefore decomposes into disjoint cycles, and walk over these cycles, choosing every second edge- O(m+n).

Preference lists with ties

Introduction

Why do we don't expect linear time algorithm

Hint : Establish an equivalence with Maximum matching.

Recall

Recall Partition of vertices into Even, Odd, Unreachable w.r.t a matching for a given bipartite graph.

f-post

f-post remains the same as defined above. Just that f(a) is not singleton. But s(a) changes. (Why??)

Gallai–Edmonds decomposition

Theorem

Let E,O,U be the partition of vertices of a graph G into Even, Odd, Unreachable sets w.r.t maximum matching M.
(i) E,O,U are pairwise disjoint.
(ii) E,O,U are invariant of the maximum matching.
(iii) O vertex is always matched with E vertex, U vertex is always matched with U vertex. $|M|=|O|+|U|/2$.
(iv) No O-O edges in M, no E-U edges in G.

Terminologies

First Choice Graph

The First Choice Graph is a subgraph of G containing all rank one edges. It is denoted by G1.

Theorem

Let M be a popular matching in G. Then E(G1) $\cap$ *M is a maximum matching of G*1.

Defining s(a)

Construct G1.
Find M1 = maximum matching in G1.
Let E = set of even vertices of G1 w.r.t M1.
We define s(a) to be the set of top-ranked posts in a's preference list that belongs to E1.

Properties of s(a)

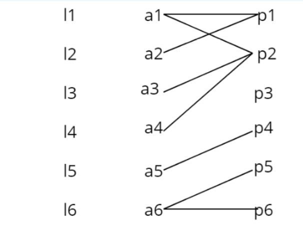
Does s(a) depend on M1?? (Recall Gallai–Edmonds decomposition)
Is s(a) singleton??
Is s(a) disjoint with f(a)??

An Example

Example

In the below example:
a1 : (p1 p2) p4 l1
a2 : p1 (p2 p5) l2
a3 : p2 (p4 p6) l3
a4 : p2 p1 p3 l4
a5 : p4 p3 p2 l5
a6 : (p5 p6) p1 l6

Constructing G1



An Example

Example

In the below example:
a1 : (p1 p2) p4 l1
a2 : p1 (p2 p5) l2
a3 : p2 (p4 p6) l3
a4 : p2 p1 p3 l4
a5 : p4 p3 p2 l5
a6 : (p5 p6) p1 l6

Finding f(a)

f(a1) = $\{p1, p2\}$
$f(a2) = \{p1\}$
$f(a3) = f(a4) = f(a5) = \{p2\}$

Computing s(a)

Even Set in G1 (E) = $\{a1, a2, a3, a4, p3, p5, p6, l1, l2, l3, l4, l5, l6\}$
$s(a1) = \{l1\}$
$s(a2) = s(a3) = \{p5\}$
$s(a4) = s(a5) = \{p3\}$
$s(a6) = \{p5, p6\}$

Characterizing Popular Matchings

Theorem

A matching M is popular in G if and only if
(i) M $\cap$ $E(G1)$ *is a maximum matching of* $G1$, *and*
(ii) *for each applicant* a, $M(a) \in f(a) \cup s(a)$.

Proof

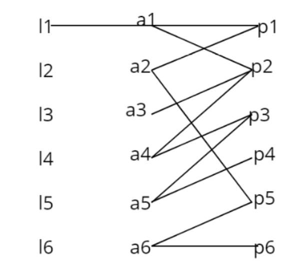
In similar lines to the strict preferences case, we prove the necessity by proving the below lemmas
1.For every applicant a, M(a) is never worse than s(a) on a's preference list.
2.For every applicant a, M(a) can never be strictly between f(a) and s(a) on a's preference list.

Reduced Graph

Reduced Graph Construction

G' is the subgraph of G containing edges from each applicant a to posts in f(a) U s(a).

Reduced Graph for above example



Algorithm

Theorem

M is a popular matching of G if and only if
(i) M $\cap$ $E(G1)$ *is a maximum matching of* $G1$, *and*
(ii)*M is an applicant complete matching of the reduced graph* G'

Popular Matchings for the above example

M1 = $\{(a1, p1), (a2, p5), (a3, p2), (a4, p3), (a5, p4), (a6, p6)\}$
$M2 = \{(a1, p2), (a2, p1), (a3, p6), (a4, p3), (a5, p4), (a6, p5)\}$
$M3 = \{(a2, p1), (a3, p2), (a4, p3), (a5, p4), (a6, p5)\}$
$M4 = \{(a2, p1), (a3, p2), (a4, p3), (a5, p4), (a6, p6)\}$
$M5 = \{(a2, p1), (a3, p6), (a4, p2), (a5, p4), (a6, p5)\}$

Algorithm

Finding Popular Matching

1.Construct G' .
2.Compute a maximum matching M1 in G1.
3.Delete all edges in G' connecting two nodes in the set O or a node in O with a node in U, where O and U are the sets of odd and unreachable nodes of G1.
4.Determine a maximum matching M in the modified graph G' by augmenting M1.
5. If M is not applicant complete, return "No popular matching",else return M.

Correctness

(i).By the previous theorem , we need to only prove that
M $\cap E(G1)$ *is a maximum matching in* $G1$.
(ii).*This can be proved by Gallai Edmonds decomposition theorem that by step* 3, *we are ensuring M has atleast* $|O| + |U|/2$ *edges*.

Analysis

Step 1 – Linear time.
Step 2 – O(m$\sqrt{n}$)(*Hopcroft Karp Algo*).
Step 3 – *Linear time*.
Step 4 – $O(m\sqrt{n})$ (*Hopcroft Karp Algo*).
Step 5 – *Linear time*.

Developments

Authors' observations

Probability

The authors performed simulations on popular matchings on bipartite graphs which are randomly constructed.
They concluded that Popular matchings exist with good probability when the chance of ties in the preference lists is high.

Further Developments

(i). Mahdian has shown that a popular matching exists with high probability, when (i) preference lists are randomly constructed and (ii) the number of posts is a small multiplicative factor larger than the number of applicants.
(ii) Manlove and Sng have also considered the case in which each post having a capacity (number of applicants it can accommodate). They gave an
O$((\sqrt{C})n) + m)$ *algo for no ties case* ,
$O((\sqrt{C} + n)m)$ *algo considering ties.(Here C is the sum of capacities of all posts)*

Unpopularity

Problem

- McCutchen defined the problem of finding a least-unpopular matching, where the unpopularity of a matching M is defined as the maximum ratio over all matchings M' of the number of applicants preferring M' to
- M to the number of applicants preferring M to M'.
  This version is proved to be NP-hard, and also there are other versions of unpopularity.