Some Results on Matchings

Kurt Mehlhorn

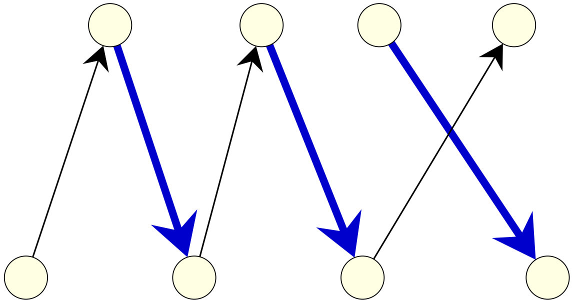
MPI für Informatik

Saarbrücken

Germany

Matchings

- $G = (A \dot{\cup} B, E)$ bipartite graph
- males and females, persons and jobs, families and houses, medical students and hospitals, students and lab sessions
- matching M = subset of edges no two of which share an endpoint



- participants express preferences
  - either by assigning profits to the edges
  - or by ordering the edges (I prefer x over y)
- optimize quality (cardinality and/or total happyness) of M

Overview

- Average Case Behavior of Matching Algorithms

  STACS 04, joint work with H. Bast, G. Schäfer, and H. Tamaki

- Strongly Stable Matchings

  STACS 04, joint work with T. Kavitha, D. Michail, and K. Paluch

- Rank Maximal Matchings

  SODA 04, joint work with R. Irving, T. Kavitha, D. Michail, and K. Paluch

- Pareto-Optimal Matchings

  joint work with D. Abraham, K. Cechlárová, D. Manlove

- papers can be found on my web-page

feel free to interrupt me at any time

Average Case Behavior of Matching Algs

Algorithms of Hopfcroft/Karp and Micali/Vazirani compute maximum cardinality matchings in bipartite or general graphs in time $O(\sqrt{n}m)$

observed behavior seems to be much better

number of phases seems to grow like $\log n$ ($n \leq 10^6$ in experiments)

Motwani(JACM, 94): running time is $O(m \log n)$ with high probability for random graphs in the $G_{n,p}$ model provided that $p \geq (\ln n)/n$.

Our result: running time is $O(m \log n)$ with high probability for random graphs in the $G_{n,p}$ model provided that $p \geq c_0/n$.

$$c_0 = \begin{array}{ll} 9.6 & \text{for bipartite graphs} \\ 35.1 & \text{for general graphs} \end{array}$$

Open problem: what happens for p with $0 \leq p \leq c_0$?

STACS 04, joint work with H. Bast, G. Schäfer, and H. Tamaki

Random Graphs, the $G_{n,p}$ model

- every potential edge is present with probability p, independent of other edges
- bipartite graphs with n nodes on each side: n^2 potential edges
- general graphs with n nodes: $n(n-1)/2$ potential edges
- expected degree is pn for bipartite graphs
- expected degree is $p(n-1)$ for general graphs

The Result

- $G =$ random graph in $G_{n,p}$ model
- $p \geq c_0/n$,
- $c_0 = \begin{array}{rl} 9.6 & \text{for bipartite graphs} \\ 35.1 & \text{for general graphs} \end{array}$
- with high probability G has the property that every non-maximum matching has an augmenting path of length $O(\log n)$
- algs of Hopcroft/Karp and Micali/Vazirani compute maximum matchings in expected time $O(m \log n)$

  because running time is $O(m \cdot L)$, where L is length of longest shortest augmenting path with respect to any non-maximum matching
- Motwani (JACM 94) proved the result for $p \geq (\ln n)/n$

Pareto-Optimal Matchings

- given a bipartite graph $G = (A \dot{\cup} B, E)$
- the nodes in A linearly order their incident edges
- a matching M is pareto-optimal if there is no matching M' in which at least one $a \in A$ is better off and no $a' \in A$ is worse off
- can compute maximum cardinality pareto-optimal matching in time $O(\sqrt{n}m)$.
- minimum cardinality problem is NP-complete.

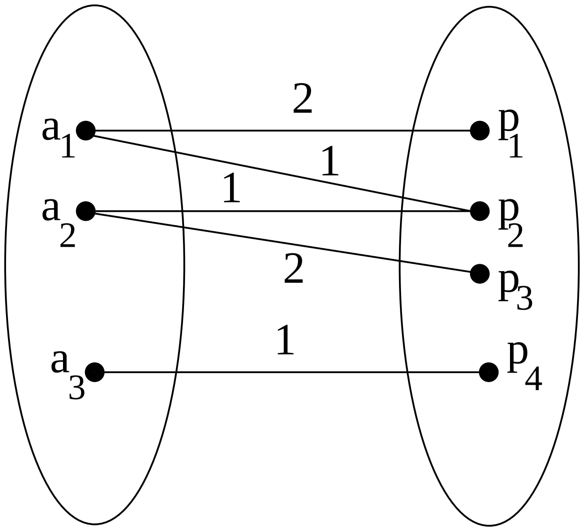
joint work with D. Abraham, K. Cechlárová, D. Manlove

Rank-Maximal Matchings

- given a bipartite graph $G = (A \dot{\cup} P, E)$
- the nodes in A rank their incident edges (ties allowed)
- rank of a matching $M = (\#$ of rank 1 edges, $\#$ of rank 2 edges, $\ldots)$
- compute a matching of maximal rank
- time $O(\min(r \cdot n^{1/2} \cdot m, n \cdot m))$ and space $O(m)$, where r is the maximal rank of any edge in the optimal matching
- previous best was $O(r^3 m^2)$ by R. Irving
- problem can also be solved by reduction to weighted matchings: assign weight n^{r-i} to the edges of rank i, see below

SODA 04, joint work with R. Irving, T. Kavitha, D. Michail, and K. Paluch

Ideas underlying our Algorithm



- $s_1 = 2$ and p_2, a_3, and p_4 must be matched by rank 1 edges
- delete all rank > 1 edges incident on them
- add the remaining rank 2 edges and extend matching by augmentation

  here we do not distinguish ranks 1 and 2
- p_2, a_3, and p_4 stay matched and hence stay matched via rank 1 edges
- a_1 and a_2 must be matched by rank ≤ 2 edges.
- delete all higher rank edges incident on them
- then a_1 and a_2 will stay matched in later phases of the algorithm and will stay matched via rank ≤ 2 edges.

Variations of the Problem

- $s_i(M)$ = number of rank i edges in M
- rank-maximal, i.e., maximize $(s_1, s_2, \ldots, s_r)$
- or a maximal cardinality matching of maximal rank
- or a maximal cardinality matching minimizing $(s_r, \ldots, s_2, s_1)$
- all problems above reduce to weighted matching
- for the first problem, assign weight n^{r-i} to the edges of rank i
- edge weights and node potentials require space $O(r)$,
  arithmetic ops on weights and potentials take time $O(r)$
- general weighted matching requires $O(nm \log n)$ arithmetic ops;
  this results in time $O(rnm \log n)$ and space $O(rm)$
- scaling algs (Gabow/Tarjan,Ahuja/Orlin) for integer weights require $O(\sqrt{n}m \log(nC))$ arithmetic ops, where C is the largest weight;
  this results in time $O(r^2 \sqrt{n}m \log n)$ and space $O(rm)$
- the space requirement is a killer

Scaling with Implicitly Given Weights

- assume weights are integers in $[0, C]$, here $C = n^r$
- assume that one can compute individual bits of weights efficiently
- we modify the scaling algs
  - algs compute near-optimal matching for weights $w_\ell(e)$, where $w_\ell(e)$ consists of first ℓ bits of $w(e)$ and $1 \le \ell \le \log C$
  - dual solution (= node potentials) is used for next iteration
  - node potentials are not stored, only the reduced costs of edges
  - edges are priced out as soon as their reduced costs exceeds $4n$
  - in this way only numbers up to $O(n)$ need to be handled
  - scaling up uses the weight-oracle to obtain the next bit of every edge weight
- $O(\sqrt{n}m \log(nC))$ arithmetic ops on polynomially bounded integers and space $O(m)$; as before C is the largest weight
- time $O(r\sqrt{n}m \log n)$ and space $O(m)$

Strongly Stable Matchings

- given a bipartite graph $G = (A \dot\cup B, E)$
- the nodes rank their incident edges (ties allowed)
- a matching M is stable if there is no blocking edge
- an edge $(a,b) \in E \setminus M$ is *blocking* if
  - a would prefer to match up with b and b would not object, i.e.,
  - a prefers b over her partner in M or is unmatched in M
  - b prefers a over her partner in M or is indifferent between them or is unmatched in M
- decide existence of a stable matching and compute one

- we do so in time $O(nm)$, even if nodes in B have capacities
- previous best was $O(m^2)$ by R. Irving
- Irving's algorithm is used to match medical students and hospitals
- open problem: deal with couples

An Instance without a Stable Matching

$$\begin{array}{llcll} x_1: & w_1, w_2 & \qquad & w_1: & x_2, x_1 \\ x_2: & \{w_1, w_2\} & & w_2: & x_2, x_1 \end{array}$$

- both woman prefer x_2 to x_1.
- man x_1 prefers w_1 to w_2 and x_2 is indifferent between the women.
- every man ranks every woman and vice versa and hence any strongly stable matching must match all men and all women.
- $\{(x_1, w_1), (x_2, w_2)\}$ is not strongly stable since w_1 prefers x_2 to x_1 and x_2 is indifferent between w_1 and w_2.
- $\{(x_1, w_2), (x_2, w_1)\}$ is not strongly stable since w_2 prefers x_2 to x_1 and x_2 is indifferent between w_1 and w_2.

The Classical Algorithm (No Ties, Complete Instances)

$M = \emptyset$;
while $\exists$ a free man x **do**
  let $e = (x, w)$ be the top choice of x;
  if w is free or prefers x over her current partner **then**
    dissolve the current marriage of w (if any);
    add e to M;
    delete all edges (x', w) which w ranks strictly after e.
  else
    discard e;
  end if
end while

- once a woman is matched it stays matched
- and to better and better partners
- alg constructs a complete and stable matching (man-optimal)
- alg for general case is similar, but more involved

Average Case Analysis, Some Details

- $G =$ random graph in $G_{n,p}$ model
- $p \geq c_0/n$,
- $c_0 = \begin{array}{rl} 9.6 & \text{for bipartite graphs} \\ 35.1 & \text{for general graphs} \end{array}$
- with high probability G has the property that every non-maximum matching has an augmenting path of length $O(\log n)$
- algs of Hopcroft/Karp and Micali/Vazirani compute maximum matchings in expected time $O(m \log n)$
- because running time is $O(m \cdot L)$, where L is length of longest shortest augmenting path with respect to any non-maximum matching
- Motwani (JACM 94) proved the result for $p \geq (\ln n)/n$

Notation and Basic Facts

- $G = (V, E)$, graph
- matching = subset of edges no two of which share an endpoint
- maximum matching = matching of maximum cardinality
- $M \subseteq E$, matching
- matching edge = edge in M
- non-matching edge = edge outside M
- matched node = node incident to an edge in M
- free node = non-matched node
- alternating path $p = (e_1, e_2, \ldots, e_k)$ with $e_i \in M$ iff $e_{i+1} \notin M$
- augmenting path = alternating path connecting two free nodes
- if p is augmenting, $M \oplus p$ has one larger cardinality than M
- if M is non-maximum, there is augmenting path relative to it
- $S \subseteq V$, $\Gamma(S) =$ neighbors of the nodes in S

Motwani's Argument

- non-maximum matchings in expander graphs have short augmenting paths because alternating trees are bushy and hence reach all nodes after $\log n$ levels
- expander graph: $|\Gamma(S)| \geq (1+\varepsilon)|S|$ for every node set S with $|S| \leq n/2$
- for $p \geq (\ln n)/n$: random graphs are essentially expander graphs
- sparse random graphs are far from being expander graphs
  - constant fraction of nodes is isolated
  - constant fraction of nodes has degree one
  - there are chains of length $O(\log n)$
  - nevertheless, our proof also uses the concept of expansion

Two Probabilistic Lemmas

An *alternating path tree* is a rooted tree of even depth, where each vertex in $Odd(T)$ has exactly one child.

We use $Even(T)$ to denote the nodes of even depth excluding the root. Then $|Odd(T)| = |Even(T)|$.

Lemma 1 (Expansion Lemma) *For each $\varepsilon > 0$ and $\beta > 1 + \varepsilon$, there are constants α and c_0 such that a random graph $G \in G(n, n, c/n)$, where $c \geq c_0$, with high probability has the following property:*
for each alternating path tree T with $\alpha \cdot log n \leq |Even(T)| \leq n/\beta$, it holds that

$$|\Gamma(Even(T))| \geq (1+\varepsilon) \cdot |Even(T)|$$

Lemma 2 (Large Sets Lemma) *For every $\beta > 1$, a random graph $G \in G(n, n, c/n)$, where $c \geq 2\beta(1 + \ln \beta)$, with high probability has the property that for every two disjoint sets of vertices, both of size at least n/β has an edge between them.*

$\varepsilon = 0.01$, $\beta = 2.6$, $c_0 = 9.6$

The Proof of the Main Theorem: Bipartite Case

- M non-maximum matching, p augmenting path, endpoints f_1 and f_2
- grow alternating trees T_1 and T_2 rooted at f_1 and f_2, respectively
  - suppose we have constructed even nodes at level $2j$
  - put their unreached neighbors into level $2j+1$
  - stop if one of the new nodes is free or
    belongs to other tree
  - put mates of new nodes into level $2j+2$
- grow the trees in phases: in each phase add two levels to both trees
- Claim: process ends after a logarithmic number of phases

The Proof of the Main Theorem: Bipartite Case

- M non-maximum matching, p augmenting path, endpoints f_1 and f_2
- grow alternating trees T_1 and T_2 rooted at f_1 and f_2, respectively
  - suppose we have constructed even nodes at level $2j$
  - put their unreached neighbors into level $2j+1$
  - stop if one of the new nodes is free or
    belongs to other tree
  - put mates of new nodes into level $2j+2$
- grow the trees in phases: in each phase add two levels to both trees
- if $|Even(T_i)| \geq n/\beta$ for $i = 1, 2$, the Large Sets Lemma guarantees an edge connecting them and the process stops
- use Expansion Lemma to show that situation of preceding item is reached in a logarithmic number of phases
- Expansion Lemma guarantees expansion of trees with at least logarithmically many levels

The Proof of the Main Theorem: Bipartite Case

MAX-PLANCK-GESELLSCHAFT

- M non-maximum matching, p augmenting path, endpoints f_1 and f_2
- grow alternating trees T_1 and T_2 rooted at f_1 and f_2, respectively
  - suppose we have constructed even nodes at level $2j$
  - put their unreached neighbors into level $2j+1$
  - stop if one of the new nodes is free or belongs to other tree
  - put mates of new nodes into level $2j+2$
- grow the trees in phases: in each phase add two levels to both trees
- consider a phase $2j$ with $j \geq \alpha \log n$: then $|Even(T_i)| \geq \alpha \log n$
- assume $|Even(T_i)| < n/\beta$ and let T_i' be the next tree
- Expansion Lemma guarantees $|\Gamma(Even(T_i))| \geq (1+\varepsilon) \cdot |Even(T_i)|$
- $|Even(T_i')| = |Odd(T_i')| = |\Gamma(Even(T_i))|$
- thus $|Even(T_i')| \geq (1+\varepsilon) \cdot |Even(T_i)|$ and we have exponential growth

Why do Trees expand, if Sets do not?

Motwani used an expansion lemma for sets.

What is probability that some set does not expand, i.e., for some set S, $|S| = s$, we have $|T| \leq \varepsilon s$ where $T = \Gamma(S) \setminus S$?

$$\sum_{t \leq \varepsilon s} \binom{n}{s} \binom{n-s}{t} (1 - c/n)^{s(n-(s+t))}$$

- there are $\binom{n}{s}$ ways to choose S
- and $\binom{n-s}{t}$ ways to choose T
- and we want no edge from S to $V \setminus T$

Why do Trees expand, if Sets do not?

Motwani used an expansion lemma for sets.

What is probability that some set does not expand, i.e., for some set S, $|S| = s$, we have $|T| \leq \varepsilon s$ where $T = \Gamma(S) \setminus S$?

$$\sum_{t \leq \varepsilon s} \binom{n}{s} \binom{n-s}{t} (1 - c/n)^{s(n-(s+t))}$$

we concentrate on a single term and on the case where $s + t \ll n$. Then

$$\approx (\frac{en}{s})^s (\frac{en}{t})^t e^{-cs}$$

we ignore the term involving t and obtain

$$\approx (\frac{en}{se^c})^s$$

In order for this to be small one needs $c = \Omega(\log n)$.

And now for trees

- What happens if we require in addition that G contains a tree on S?
- We get an additional factor

$$s^{s-2}(c/n)^{s-1}$$

- the first factor counts the number of trees (Cayley's theorem)
- the second factor accounts for the fact that the edges of the tree must be present
- if we add this into our previous formula, we obtain

$$\approx (\frac{en}{se^c})^s \; s^{s-2}(c/n)^{s-1} = n/cs^2(\frac{ensc}{sne^c})^s \leq n^3(\frac{ec}{e^c})^s$$

- and this is small if $s = \Omega(\log n)$ and c a sufficiently large constant: logarithmic size trees expand

- of course, the details are slightly more involved

Open Problems

- is the result true for all random graphs?
  - we need $c \geq c_0$, $\quad c_0 = 9.6$ for bipartite graphs, ...
  - result also holds for $c < 1$, since only logarithmic size connected components
  - what happens in between?

- smoothed analysis
  - start with an arbitrary graph G
  - perturb it by deleting/adding a small number of edges
  - if edges are added with probability $p \geq (\ln n)/n$ Motwani's analysis still applies
  - what happens for $p = c/n$ and constant c?