Algorithm Engineering for Matching in Bipartite Graph Streams

Lasse Kliemann
contact: http://lasse-kliemann.name

Christian-Albrechts-Universität zu Kiel
Kiel, Germany

Joint work with:

Sebastian Eggert, Ole Kliemann, Peter Munstermann
Anand Srivastav
Selected related work

**Problem:** maximum-cardinality matching in bipartite graphs

**Model:** semi-streaming, i.e., $O(n \cdot \text{poly log } n \cdot \text{poly } 1/\varepsilon)$ edges

**Approximation:** $(1 + \varepsilon) \cdot |M| \geq |M^*|$

### Solutions

<table>
<thead>
<tr>
<th>Graphs</th>
<th>Passes</th>
<th>Technique</th>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>general</td>
<td>$O((1/\varepsilon)^{1/\varepsilon})$</td>
<td>aug’path, random.</td>
<td>McGregor 2005</td>
</tr>
<tr>
<td>bipartite</td>
<td>$O(1/\varepsilon^5)$ and $O(n/\varepsilon)$</td>
<td>aug’path, determ.</td>
<td>we 2009/2011</td>
</tr>
<tr>
<td>bipartite$^1$</td>
<td>$O(1/\varepsilon^2 \cdot \log \log 1/\varepsilon)$</td>
<td>LP-based, determ.</td>
<td>Ahn &amp; Guha 2011</td>
</tr>
<tr>
<td>general$^2$</td>
<td>$O(1/\varepsilon^4 \cdot \log n)$</td>
<td>LP-based, determ.</td>
<td>Ahn &amp; Guha 2011</td>
</tr>
</tbody>
</table>

$^1$ works also for weighted matching in $O(1/\varepsilon^2 \cdot \log 1/\varepsilon)$ passes.

$^2$ works also for weighted matching.
Selected related work

Problem: maximum-cardinality matching in bipartite graphs
Model: semi-streaming, i.e., $O(n \cdot \text{poly log } n \cdot \text{poly } 1/\epsilon)$ edges
Approximation: $(1 + \epsilon) \cdot |M| \geq |M^*|$

| Solutions |
|-----------------|-----------------|----------------|-----------|
| **Graphs** | **Passes** | **Technique** | **Author(s)** |
| general | $O((1/\epsilon)^{1/\epsilon})$ | aug’path, random. | McGregor 2005 |
| bipartite | $O(1/\epsilon^5)$ and $O(n/\epsilon)$ | aug’path, determ. | we 2009/2011 |
| bipartite$^1$ | $O(1/\epsilon^2 \cdot \log \log 1/\epsilon)$ | LP-based, determ. | Ahn & Guha 2011 |
| general$^2$ | $O(1/\epsilon^4 \cdot \log n)$ | LP-based, determ. | Ahn & Guha 2011 |

$^1$ works also for weighted matching in $O(1/\epsilon^2 \cdot \log 1/\epsilon)$ passes.
$^2$ works also for weighted matching.
The following is based on:

- Eggert, Kliemann, Munstermann, Srivastav
  “Bipartite Matching in the Semi-Streaming Model”
  ESA 2009 / Algorithmica 2012

- Kliemann
  “Matching in Bipartite Graph Streams in a Small Number of Passes”
  SEA 2011

All documents available for free at
http://www.informatik.uni-kiel.de/~lki
or via http://lasse-kliemann.name
Augmenting paths
Approximation theory for graph matching

- Let $k \in \mathbb{N}_{\geq 1}$. We aim for a $1 + \frac{1}{k}$ approximation, i.e.,
  \[(1 + \frac{1}{k}) |M| \geq |M^*|,\]
  where $M^*$ is an optimal matching and $M$ the computed matching.

- Given $\lambda \in \mathbb{N}$, we call a path a $\lambda$ path if it has length at most $2\lambda + 1$.

$(\lambda_1, \lambda_2)$ DAP set

- Fix a matching $M$.
- Fix $\lambda_1, \lambda_2 \in \mathbb{N}, \lambda_1 \leq \lambda_2$.
- A set $\mathcal{Y}$ of paths is called a $(\lambda_1, \lambda_2)$ DAP set (shortly: “$\lambda$ DAP set”) if:
  (i) All paths in $\mathcal{Y}$ are $M$ augmenting $\lambda_2$ paths.
  (ii) Any two paths in $\mathcal{Y}$ are vertex-disjoint.
  (iii) We cannot add another $M$ augmenting $\lambda_1$ path to $\mathcal{Y}$ without violating condition (ii).
Existence of small DAP sets guarantees approximation

Let $k \in \mathbb{N}_{\geq 1}$ and $k \leq \lambda_1 \leq \lambda_2$ and

$$
\delta(k, \lambda_1, \lambda_2) := \frac{\lambda_1 - k + 1}{2k\lambda_1(\lambda_2 + 2)} \in (0, 1).
$$

**Lemma**

Let $M$ be an inclusion-maximal matching. Let $\mathcal{Y}$ be a $(\lambda_1, \lambda_2)$ DAP set such that $|\mathcal{Y}| \leq 2\delta|M|$ with $\delta = \delta(\lambda_1, \lambda_2)$.

Then $M$ is a $1 + \frac{1}{k}$ approximation.

N.B.: mere existence of an appropriate $\lambda$ DAP set guarantees optimality – we do not have to compute it!
Given $\delta \in [0, 1]$, an algorithm is called a $(\lambda_1, \lambda_2, \delta)$ DAP approximation algorithm if for any matching $M$ it delivers a result $A$ such that:

- $A$ consists of disjoint $M$ augmenting $\lambda_2$ paths;
- there exists a $(\lambda_1, \lambda_2)$ DAP set $Y$ so that $|Y| \leq |A| + \delta |M|$. 

DAP approximation
Approximation algorithm

Theorem

Assume we have a \((\lambda_1, \lambda_2, \delta)\) DAP approximation algorithm \(\text{DAP}\) for \(\delta = \delta(k, \lambda_1, \lambda_2)\) and \(k \leq \lambda_1 \leq \lambda_2\).

Then the following gives a \(1 + \frac{1}{k}\) approximation to the matching problem.

\[
\begin{align*}
M &:= \text{any inclusion-maximal matching} \\
\text{repeat} &\ \\
A &:= \text{DAP}(M) \\
\text{if } |A| &\leq \delta |M| \text{ then break and return } M \\
\text{augment } M \text{ using } A \\
\text{until } \text{forever}
\end{align*}
\]

Proof.

- Let \(\mathcal{Y}\) be the \((\lambda_1, \lambda_2)\) DAP set guaranteed to exist.
- By the termination criterion: \(|\mathcal{Y}| \leq |A| + \delta |M| \leq 2\delta |M|\).
- Lemma: we have our approximation.
$1 < \ell(m) = \lambda + 1$?
\ell(m) := 1
\[ \ell(m) := 2 \]
$2 < \ell(m_1) = 3?$
\( \ell(m_1) := 2 \) and \( \ell(m_2) := 3 \)
Let $\lambda := \lambda_1 = \lambda_2$. The number of passes is

$$O \left( \frac{k^2\lambda^5}{(\lambda - k + 1)^2} \right) = \begin{cases} 
O(k^5) & \text{if } \lambda = 2k - 1 \text{ (long paths)} \\
O(k^6) & \text{if } \lambda = k + \sqrt{k} - 1 \\
O(k^7) & \text{if } \lambda = k \text{ (short paths)} 
\end{cases}$$
Experimental setup

- Random bipartite graphs (rand)
- Difficult instances (degm, hilo, rbg, rope)
- For example, rope looks like this:

![Graph diagram]

- Hard limit of 8 GiB, that are about $1 \times 10^9$ edges.
- C++ implementation on 64 bit CentOS Linux.
Experimental setup

- Random bipartite graphs (rand)
- Difficult instances (degm, hilo, rbg, rope)
- For example, rope looks like this:

![Diagram of rope structure]

- Hard limit of 8 GiB, that are about $1 \times 10^9$ edges.
- C++ implementation on 64 bit CentOS Linux.
Experimental results

- Fix $k = 9$, that is, we guarantee a 90% approximation.
- $n = 40,000, 41,000, \ldots, 50,000$
- Density is limited by $D_{\text{max}} = \frac{1}{10}$.
- Number of edges ranges up to about $|E| = 62 \times 10^6$.

<table>
<thead>
<tr>
<th>maximum</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>rand</td>
<td>degm</td>
</tr>
<tr>
<td>$O(k^5)$</td>
<td>11,886</td>
</tr>
<tr>
<td>$O(k^6)$</td>
<td>7,817</td>
</tr>
<tr>
<td>$O(k^7)$</td>
<td>7,121</td>
</tr>
</tbody>
</table>

Theoretical bounds with all constants:

$$
\begin{align*}
O(k^5) & = 14,211,355 \\
O(k^6) & = 16,215,343 \\
O(k^7) & = 57,193,291
\end{align*}
$$
Growing trees
\ell \ell \ell \ell m_1 m_3 m_2 m_4 \ell p m_i q : \lambda`
\ell(m_i) := \lambda + 1
\[\ell_1^2 > \ell_2^2\]

\[m_1 > m_2 > m_3 > m_4\]

\[\lambda_{14}\]
Theoretical analysis

- $1 + \frac{1}{k}$ approximation
- $O(kn)$ passes
- $\Omega(n)$ passes
- Running in parallel to original version, we inherit $O(k^{O(1)})$ bound.
### Experimental results

<table>
<thead>
<tr>
<th>$n$</th>
<th>rand</th>
<th>degm</th>
<th>hilo</th>
<th>rbg</th>
<th>rope</th>
<th>$O(kn)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>8,192</td>
<td>35</td>
<td>21</td>
<td>51</td>
<td>43</td>
<td>53</td>
<td>18,433</td>
</tr>
<tr>
<td>16,384</td>
<td>25</td>
<td>26</td>
<td>52</td>
<td>44</td>
<td>57</td>
<td>36,865</td>
</tr>
<tr>
<td>32,768</td>
<td>37</td>
<td>19</td>
<td>54</td>
<td>48</td>
<td>59</td>
<td>73,729</td>
</tr>
<tr>
<td>65,536</td>
<td>39</td>
<td>23</td>
<td>55</td>
<td>50</td>
<td>64</td>
<td>147,457</td>
</tr>
<tr>
<td>131,072</td>
<td>28</td>
<td>28</td>
<td>57</td>
<td>51</td>
<td>65</td>
<td>294,913</td>
</tr>
<tr>
<td>262,144</td>
<td>39</td>
<td>42</td>
<td>58</td>
<td>53</td>
<td>64</td>
<td>589,825</td>
</tr>
<tr>
<td>524,288</td>
<td>27</td>
<td>26</td>
<td>58</td>
<td>55</td>
<td>61</td>
<td>1,179,649</td>
</tr>
<tr>
<td>1,048,576</td>
<td>44</td>
<td>26</td>
<td>62</td>
<td>57</td>
<td>53</td>
<td>2,359,297</td>
</tr>
<tr>
<td>2,097,152</td>
<td>41</td>
<td>31</td>
<td>60</td>
<td>57</td>
<td>59</td>
<td>4,718,593</td>
</tr>
<tr>
<td>4,194,304</td>
<td>30</td>
<td>29</td>
<td>62</td>
<td>58</td>
<td>54</td>
<td>9,437,185</td>
</tr>
</tbody>
</table>

Data based on roughly 800,000 instances in total.
Lower bound $\Omega(n)$
Lower bound $\Omega(n)$
Lower bound $\Omega(n)$
Lower bound $\Omega(n)$
Lower bound $\Omega(n)$
Lower bound $\Omega(n)$
Current work

- Beat the adversary by randomization (experimental study).
- Modify path/tree-growing approach for general graphs!