Tools for Kernelizing Graph Cut Problems

Stefan Kratsch      Magnus Wahlström

Max Planck Institute for Informatics

September 3, 2011
Graph cut problems

For many graph cut problems, the existence of polynomial kernels is/was unknown:

- **Multiway Cut** – separate terminals $T$ by removing $k$ vertices
- **Directed Feedback Vertex Set** – hit all directed cycles
- **Multicut** – fulfill all cut requests $(s_i, t_i)$ using $k$ vertices

Also related problems (graph cut problems in disguise):

- **Graph Bipartization (OCT)** – hit all odd cycles
- **Almost 2-SAT**, a.k.a. 2-CNF Deletion – remove $k$ variables/ clauses to make $F$ satisfiable
Graph cut problems

Many found attackable by matroid theory:

This talk:
- Graph Bipartization (OCT)

Other problems:
- Almost 2-SAT, a.k.a. 2-CNF Deletion
- Multiway Cut – restricted cases
- Directed Feedback Vertex Set – unknown
- Multicuts – unknown
Outline

Matroids 1: Encoding Terminal Cuts
  Matroid introduction
  Encoding Terminal Cuts

Application: OCT kernel

*Matroids 2: Irrelevant vertices
  Application sketches
Matroids 1:
Encoding Terminal Cuts
Matroids

*Matroid theory is for the concept of dependence what group theory is for symmetry — (unknown)*

A matroid $M = (U, I)$, $I \subseteq 2^U$, is an independence system with independent sets $I$ satisfying:

1. The empty set is independent
2. A subset of an independent set is independent
3. **Augmentation property:** If $A, B$ are independent and $|B| > |A|$, then there is some $b \in B - A$ such that $A + b$ is independent

**Rank** $r(X)$: Size of largest independent subset of $X$
Matroids

Matroid theory is for the concept of dependence what group theory is for symmetry — (unknown)

A matroid $M = (U, \mathcal{I})$, $\mathcal{I} \subseteq 2^U$, is an independence system with independent sets $\mathcal{I}$ satisfying:

1. The empty set is independent
2. A subset of an independent set is independent
3. Augmentation property: If $A, B$ are independent and $|B| > |A|$, then there is some $b \in B - A$ such that $A + b$ is independent

Rank $r(X)$: Size of largest independent subset of $X$
Examples

Canonical examples:

1. **Graphic matroids**: Let \( G = (V, E) \) be a graph.
   - \( M = (E, I) \), \( I \) contains cycle-free edge sets
   - Rank: number of vertices minus number of components

2. **Linear matroids** \( M = (U, I) \):
   - \( U \) is a collection of vectors in \( \mathbb{F}^d \) for some field \( \mathbb{F} \)
   - Independence concept is linear independence
   - Rank: dimension

Linear matroids more conveniently represented by \( d \times |U| \) matrix. Many tools work only for linear matroids (our matroids are linear).
Examples

Canonical examples:

1. **Graphic matroids**: Let $G = (V, E)$ be a graph.
   - $M = (E, I)$, $I$ contains cycle-free edge sets
   - Rank: number of vertices minus number of components

2. **Linear matroids** $M = (U, I)$:
   - $U$ is a collection of vectors in $\mathbb{F}^d$ for some field $\mathbb{F}$
   - Independence concept is linear independence
   - Rank: dimension

**Linear** matroids more conveniently represented by $d \times |U|$ matrix. Many tools work only for linear matroids (our matroids are linear).
Gammoids

Let $G = (V, E)$ be a graph. Say that $T \subseteq V$ is linked into $S$ if there are $|T|$ vertex-disjoint paths from $S$ to $T$ (not only internally vertex-disjoint).

The gammoid defined by $G$ and $S$ is $M = (V, \mathcal{I})$ where:

- $\mathcal{I}$ contains all sets linked into $S$
- The rank $r(X)$ equals the size of an $(S, X)$-cut
- **Augmentation property**: see next slide

Also works for digraphs.
Gammoids: augmentation property

Let $G = (V, E)$, $S \subseteq V$, $A$ and $B$ linked into $S$, with $|B| > |A|$. There exists a vertex $v \in B - A$ such that $A + v$ is linked into $S$.

1. Let $C$ be the minimum $(S, A)$-cut closest to $S$. Claim: if $A + v$ is dependent then $C$ cuts $v$ from $S$.
   - $A + v$ dependent $\Rightarrow$ cut $(S, A + v)$ of size $< |A| + 1$
   - $A$ independent $\Rightarrow$ cut has size $|A| = |C|$, dominated by $C$

2. $|B|$ paths from $S$ to $B$, $|B| > |C|$ $\Rightarrow$ some $b \in B$ is not cut by $C$. 
Gammoids: augmentation property

Let $G = (V, E)$, $S \subseteq V$, $A$ and $B$ linked into $S$, with $|B| > |A|$. There exists a vertex $v \in B - A$ such that $A + v$ is linked into $S$.

1. Let $C$ be the minimum $(S, A)$-cut closest to $S$. Claim: if $A + v$ is dependent then $C$ cuts $v$ from $S$.
   
   $- A + v$ dependent $\Rightarrow$ cut $(S, A + v)$ of size $< |A| + 1$
   $- A$ independent $\Rightarrow$ cut has size $|A| = |C|$, dominated by $C$

2. $|B|$ paths from $S$ to $B$, $|B| > |C| \Rightarrow$ some $b \in B$ is not cut by $C$. 
**Gammoids: augmentation property**

Let $G = (V, E)$, $S \subseteq V$, $A$ and $B$ linked into $S$, with $|B| > |A|$. There exists a vertex $v \in B - A$ such that $A + v$ is linked into $S$.

1. Let $C$ be the minimum $(S, A)$-cut closest to $S$. Claim: if $A + v$ is dependent then $C$ cuts $v$ from $S$.
   - $A + v$ dependent $\Rightarrow$ cut $(S, A + v)$ of size $< |A| + 1$
   - $A$ independent $\Rightarrow$ cut has size $|A| = |C|$, dominated by $C$

2. $|B|$ paths from $S$ to $B$, $|B| > |C|$ $\Rightarrow$ some $b \in B$ is not cut by $C$. 
Gammoids: augmentation property

Let \( G = (V, E) \), \( S \subset V \), \( A \) and \( B \) linked into \( S \), with \( |B| > |A| \).
There exists a vertex \( v \in B - A \) such that \( A + v \) is linked into \( S \).

1. Let \( C \) be the minimum \((S, A)\)-cut closest to \( S \). Claim:
   if \( A + v \) is dependent then \( C \) cuts \( v \) from \( S \).
   - \( A + v \) dependent \( \Rightarrow \) cut \((S, A + v)\) of size \( < |A| + 1 \)
   - \( A \) independent \( \Rightarrow \) cut has size \( |A| = |C| \), dominated by \( C \)

2. \( |B| \) paths from \( S \) to \( B \), \( |B| > |C| \) \( \Rightarrow \) some \( b \in B \) is not cut by \( C \).
Gammoids: small representation

1. Let $G = (V, E)$, $S, T \subseteq V$. We can represent the subsets of $T$ which are linked into $S$ in space $O(|S| \cdot |T| \cdot |S| \log |T|)$.

2. Source-free form: Let $G = (V, E)$, $X \subseteq V$ a set of terminals. We can represent the flow from $S$ to $T$ in $G - R$ for arbitrary $S, T, R \subseteq X$ in space $O(|X|^3)$.

Still works for digraphs. Randomized polynomial time with one-sided error (underestimates flow only).
Application:

Polynomial Kernel for OCT (Graph Bipartization)
OCT algorithm

Recall OCT iterative compression algorithm\(^1\) \((G_i = G[v_1, \ldots, v_i])\)

1. Start with graph \(G_1\), empty solution \(X = \emptyset\)
2. For \(i = 2 \ldots n:\)
   2.1 Have solution \(X\) of size \(k\) for \(G_{i-1}\)
   2.2 \(X' = X + v_i\) is solution of size \(k + 1\) for \(G_i\)
   2.3 Use \(X'\) to find optimal solution for \(G_i\)

\(^1\)Reed, Smith, Vetta 2004
OCT algorithm

Recall OCT iterative compression algorithm\(^2\) \((G_i = G[v_1, \ldots, v_i])\)

1. Start with graph \(G_1\), empty solution \(X = \emptyset\)
2. For \(i = 2 \ldots n\):
   2.* Compress \(X + v_i\) to optimal solution \(X\)

Kernelization order:
1. Create approximate solution \(X\) (size \(k^c\))
2. Feed \(X\) to compression step
3. Kernelize resulting graph cut problem

\(^2\)Reed, Smith, Vetta 2004
OCT algorithm

Recall OCT iterative compression algorithm\(^2\) \((G_i = G[v_1, \ldots, v_i])\)

1. Start with graph \(G_1\), empty solution \(X = \emptyset\)
2. For \(i = 2 \ldots n:\)
   2.* Compress \(X + v_i\) to optimal solution \(X\)

Kernelization order:
1. Create approximate solution \(X\) (size \(k^c\))
2. Feed \(X\) to compression step
3. Kernelize resulting graph cut problem

\(^2\)Reed, Smith, Vetta 2004
**OPT^c** approximation

1. Run FPT algorithm with parameter $k = \log n \left(3^k n^{O(1)} \rightarrow n^{O(1)}\right)$
2. Run $O(\sqrt{\log n})$-approximation;\(^3\) observe $\log n < k$

$O(OPT^{1.5}) = O(k^{1.5})$-sized solution.

\(^3\)Agarwal, Charikar, Makarychev, Makarychev, STOC 2005
Algorithm sketch (compression step)

Let $u-v$ be a normal edge ($u \neq v$), $u-v$ an equality edge ($u = v$).

1. \[ X = \{a, d\} \] Partition $G - X$ as $U \cup V = \{b\} \cup \{c\}$

2. Negate $U$: Toggle crossing edges

3. Split $X$ into positive, negative copies

4. Search for cuts that for $x \in X$ delete $x$ and $\bar{x}$, or split $x \neq \bar{x}$
Algorithm sketch (compression step)

Let $u - v$ be a normal edge ($u \neq v$), $u - v$ an equality edge ($u = v$).

1. \hspace{1cm} X = \{a, d\}. Partition $G - X$ as $U \cup V = \{b\} \cup \{c\}$

2. \hspace{1cm} Negate $U$: Toggle crossing edges

3. \hspace{1cm} Split $X$ into positive, negative copies

4. Search for cuts that for $x \in X$ delete $x$ and $\bar{x}$, or split $x \neq \bar{x}$
Algorithm sketch (compression step)

Let $u-v$ be a normal edge ($u \neq v$), $u-v$ an equality edge ($u = v$).

1. $X = \{a, d\}$. Partition $G - X$ as $U \cup V = \{b\} \cup \{c\}$

2. Negate $U$: Toggle crossing edges

3. Split $X$ into positive, negative copies

4. Search for cuts that for $x \in X$ delete $x$ and $\overline{x}$, or split $x \neq \overline{x}$
Algorithm sketch (compression step)

Let $u-v$ be a normal edge ($u \neq v$), $u-v$ an equality edge ($u = v$).

1. $X = \{a, d\}$. Partition $G - X$ as $U \cup V = \{b\} \cup \{c\}$

2. Negate $U$: Toggle crossing edges

3. Split $X$ into positive, negative copies

4. Search for cuts that for $x \in X$ delete $x$ and $\bar{x}$, or split $x \neq \bar{x}$
OCT kernel, summary

1. Use known tools to get solution $X$ with $|X| = O(k^{3/2})$
2. Create cut problem on auxiliary graph, terminals $X'$
3. Encode terminal cuts over $X'$ into $O(|X|^3)$-size description
4. Yields a $O(k^{4.5})$-sized instance description (compression, a.k.a. generalized kernel/bikernel)

(True (direct) kernel by NP-hardness.)
Matroids 2:
Irrelevant vertices
Direct kernels

### Terminal Cuts Compression

- **Input:** Graph $G = (V, E)$, sets $S, T \subset V$
- **Parameter:** $|S| + |T|$
- **Task:** Reduce $G$ to a small graph $G'$ while preserving sizes of $(A, B)$-cuts, $A \subseteq S, B \subseteq T$

Would give **direct** (combinatorial?) kernels for our problems.
Strategy: Irrelevant vertices

Terminal Cuts Compression
Reduce size of $G$ while preserving $(A, B)$-cuts, $A \subseteq S$, $B \subseteq T$

- A vertex $v$ is essential if for some $A, B$, every minimum $(A, B)$-cut uses $v$
- Otherwise irrelevant

Claim
There are at most $k^4$ essential vertices (and we can find them).
An irrelevant vertex may be removed (lifted) (then iterate).
**Strategy: Irrelevant vertices**

---

**Terminal Cuts Compression**

Reduce size of $G$ while preserving $(A, B)$-cuts, $A \subseteq S$, $B \subseteq T$

- A vertex $v$ is **essential** if for some $A, B$, every minimum $(A, B)$-cut uses $v$
- Otherwise **irrelevant**

---

**Claim**

There are at most $k^4$ essential vertices (and we can find them).

An irrelevant vertex may be removed (lifted) (then iterate).
## Case: Almost 2-SAT kernel

### Digraph Pair Cut

**Input:** Digraph $D = (V, A)$, source vertex $s$, integer $k$, set of pairs $P \subseteq \binom{V}{2}$

**Parameter:** $k$

**Problem:** Remove $k$ vertices such that for every pair $\{u, v\} \in P$, either $u$ or $v$ is not reachable from $s$

- **Claim 1:** Digraph Pair Cut has $\tilde{O}(k^4)$ kernel
- **Claim 2** (omitted): This gives a polynomial kernel for Almost 2-SAT
**Digraph Pair Cut**

Algorithm:

1. Let $T = \emptyset$, $X = \emptyset$

2. While any pair is reachable and $|X| < k$:
   - $\{u, v\}$ reachable: both $u$ and $v$ reachable from $s$ in $G - X$
   - 2.1 Find reachable pair $\{u, v\}$
   - 2.2 Branch on ($T = T + u$) or ($T = T + v$)
   - 2.3 Let $X$ be the min-$(s, T)$-cut closest to $s$

**Claims**

- There are only $k^2$ non-irrelevant pairs $P' \subseteq P$ for step 2.1
- We can encode the problem into *terminal cuts*, size $\tilde{O}(k^2|P'|)$. 
Conclusions

- Parameterized graph cut problems (still) include many open problems for polynomial kernelization
- Matroid theory gives very powerful tools for these problems
  - Encode a problem compactly (as a matrix), in small space
  - Detect irrelevant vertices/objects to remove