LP/SDP Extended formulations: lower bounds and approximation algorithms

Ph.D. Thesis Proposal

Aurko Roy
September 14, 2016
ACO PhD Student
How powerful are Linear and Semidefinite Programs?

- Linear Programming (LP)
  - a general algorithmic paradigm
  - efficient in theory and practice
  - large body of approximation algorithms

- Semidefinite programming (SDP):
  - generalizes linear programming
  - also efficient in theory
  - covers much of tractable convex optimization
  - better approximations for hard problems!
How powerful are Linear and Semidefinite Programs?

Linear Programming (LP)

- a general algorithmic paradigm
- efficient in theory and practice
- large body of approximation algorithms

Semidefinite programming (SDP):

- generalizes linear programming
- also efficient in theory
- covers much of tractable convex optimization
- better approximations for hard problems!
How powerful are Linear and Semidefinite Programs?

Linear Programming (LP)

- a general algorithmic paradigm
- efficient in theory and practice
- large body of approximation algorithms

Semidefinite programming (SDP):

- generalizes linear programming
- also efficient in theory
- covers much of tractable convex optimization
- better approximations for hard problems!
Section I: Lower bounds
Definition (Optimization problem)

An *optimization problem* \( P = (S, \mathcal{I}, \text{val}) \) consists of

- a set \( \mathcal{I} \) of *instances*,
- a set \( S \) of *feasible solutions*,
- and a real valued objective \( \text{val} : \mathcal{I} \times S \to \mathbb{R} \).

\( \text{val}_I(s) \) is the quality of a solution \( s \in S \) w.r.t. instance \( I \in \mathcal{I} \).

\( \text{OPT}(I) := \min_{s \in S} \text{val}_I(s) \)
Definition (Optimization problem)

An optimization problem \( \mathcal{P} = (S, \mathcal{I}, \text{val}) \) consists of

- a set \( \mathcal{I} \) of instances,
- a set \( S \) of feasible solutions,
Definition (Optimization problem)

An *optimization problem* $\mathcal{P} = (S, \mathcal{I}, \text{val})$ consists of

- a set $\mathcal{I}$ of *instances*,
- a set $S$ of *feasible solutions*,
- and a real valued objective $\text{val}: \mathcal{I} \times S \rightarrow \mathbb{R}$.
An abstract view of optimization problems

Definition (Optimization problem)

An optimization problem $\mathcal{P} = (S, \mathcal{I}, \text{val})$ consists of

- a set $\mathcal{I}$ of instances,
- a set $S$ of feasible solutions,
- and a real valued objective $\text{val}: \mathcal{I} \times S \to \mathbb{R}$.

- $\text{val}_\mathcal{I}(s)$: quality of a solution $s \in S$ w.r.t instance $\mathcal{I} \in \mathcal{I}$
- $\text{OPT} (\mathcal{I}) := \min_{s \in S} \text{val}_\mathcal{I}(s)$
Fractional optimization problems

- An optimization problem where the objective $\text{val}_I$ is of the form $\text{val}^n_I / \text{val}^d_I$.
Fractional optimization problems

- An optimization problem where the objective $\text{val}_{\mathcal{I}}$ is of the form $\text{val}_{\mathcal{I}}^{n} / \text{val}_{\mathcal{I}}^{d}$.
- Efficient LP based algorithms are used to find an optimal value of a linear combination of $\text{val}_{\mathcal{I}}^{n}$ and $\text{val}_{\mathcal{I}}^{d}$.
Fractional optimization problems

- An optimization problem where the objective $v_{\mathcal{I}}$ is of the form $v_{\mathcal{I}}^n / v_{\mathcal{I}}^d$.
- Efficient LP based algorithms are used to find an optimal value of a linear combination of $v_{\mathcal{I}}^n$ and $v_{\mathcal{I}}^d$.

**Example**

The **Sparsest Cut** problem

- $c : E(K_n) \rightarrow \mathbb{R}_{\geq 0}$, called the *capacity* function
- $d : E(K_n) \rightarrow \mathbb{R}_{\geq 0}$ called the *demand* function

The objective function for a cut $S$ is to minimize $\frac{\sum_{i \in S, j \notin S} c(i,j)}{\sum_{i \in S, j \notin S} d(i,j)}$. 
LP formulation for a fractional problem

A linear program $Ax \leq b$ with $x \in \mathbb{R}^r$ s.t.:

**Feasible solutions** as vectors $x^s \in \mathbb{R}^r$ for every $s \in S$

satisfying

$$Ax^s \leq b \quad \text{for all } s \in S,$$
LP formulation for a fractional problem

A linear program $Ax \leq b$ with $x \in \mathbb{R}^r$ s.t.:

**Feasible solutions** as vectors $x^s \in \mathbb{R}^r$ for every $s \in S$ satisfying

$$Ax^s \leq b \quad \text{for all } s \in S,$$

**Instances** as a pair of affine functions $w^n_\mathcal{I}, w^d_\mathcal{I} : \mathbb{R}^r \to \mathbb{R}$ for all $\mathcal{I} \in \mathcal{I}^S$ satisfying

$$w^n_\mathcal{I}(x^s) = \text{val}^n_\mathcal{I}(s)$$
$$w^d_\mathcal{I}(x^s) = \text{val}^d_\mathcal{I}(s)$$
LP formulation for a fractional problem

A linear program $Ax \leq b$ with $x \in \mathbb{R}^r$ s.t.:

**Feasible solutions** as vectors $x^s \in \mathbb{R}^r$ for every $s \in S$ satisfying

$$Ax^s \leq b \quad \text{for all } s \in S,$$

**Instances** as a pair of affine functions $w^n_I, w^d_I : \mathbb{R}^r \to \mathbb{R}$ for all $I \in \mathcal{I}^S$ satisfying

$$w^n_I(x^s) = \text{val}^n_I(s)$$

$$w^d_I(x^s) = \text{val}^d_I(s)$$

**(C, S) approximation guarantee** If $\text{OPT}(I) \geq S(I)$

$$Ax \leq b \Rightarrow \begin{cases} w^d_I(x) \geq 0 \\ w^n_I(x) \geq C(I)w^d_I(x) \end{cases}$$
Example of an LP formulation for a fractional problem

Example

A common LP relaxation for the \textsc{SparsestCut} problem with capacity function \(c\) and demand function \(d\) is the following

\[
\begin{align*}
\min \sum_{i,j} c(i,j)x_{ij} \\
\sum_{i,j} d(i,j)x_{ij} &\geq \alpha \sum_{i,j} d(i,j) \\
1 &\geq x_{ij} \geq 0
\end{align*}
\]

This is the LP used by \cite{GTW13}; the value of \(\alpha\) is found by binary search.
Example

A common LP relaxation for the \textsc{SparsestCut} problem with \textit{capacity} function $c$ and \textit{demand} function $d$ is the following

$$
\begin{align*}
\min & \quad \sum_{i,j} c(i, j) x_{ij} \\
\text{s.t.} & \quad \sum_{i,j} d(i, j) x_{ij} \geq \alpha \sum_{i,j} d(i, j) \\
& \quad 1 \geq x_{ij} \geq 0
\end{align*}
$$

This is the LP used by [GTW13]; the value of $\alpha$ is found by binary search.
Slack matrix for fractional optimization problems

Definition

(C, S)-approximate slack matrix for a fractional optimization problem:

\[ M_{\mathcal{P}, C, S} = \begin{bmatrix} M_{\mathcal{P}, C, S}^{(d)} \\ M_{\mathcal{P}, C, S}^{(n)} \end{bmatrix} \]

where \( M_{\mathcal{P}, C, S}^{(d)} \), \( M_{\mathcal{P}, C, S}^{(n)} \) are nonnegative \( \mathcal{I}^S \times S \) matrices with entries

\[ M_{\mathcal{P}, C, S}^{(d)}(\mathcal{I}, s) := \text{val}_{\mathcal{I}}^d(s) \]
\[ M_{\mathcal{P}, C, S}^{(n)}(\mathcal{I}, s) := \text{val}_{\mathcal{I}}^n(s) - C(\mathcal{I}) \text{val}_{\mathcal{I}}^d(s). \]
Low rank non-affine reductions between fractional problems

Definition

A reduction from $\mathcal{P}_1$ to $\mathcal{P}_2$ consists of

- Two maps $\ast : \mathcal{I}_1 \to \mathcal{I}_2$ and $\ast : S_1 \to S_2$
Definition

A reduction from $\mathcal{P}_1$ to $\mathcal{P}_2$ consists of

- Two maps $\ast : \mathcal{I}_1 \to \mathcal{I}_2$ and $\ast : \mathcal{S}_1 \to \mathcal{S}_2$
- Four nonnegative $\mathcal{I}_1 \times \mathcal{S}_1$ matrices $M_1^{(n)}$, $M_1^{(d)}$, $M_2^{(n)}$, $M_2^{(d)}$

Matrices $M_1^{(n)}$, $M_1^{(d)}$, $M_2^{(n)}$, $M_2^{(d)}$ can be arbitrary functions of the solutions and instances.
Definition

A reduction from $\mathcal{P}_1$ to $\mathcal{P}_2$ consists of

- Two maps $\star : \mathcal{I}_1 \to \mathcal{I}_2$ and $\star : \mathcal{S}_1 \to \mathcal{S}_2$
- Four nonnegative $\mathcal{I}_1 \times \mathcal{S}_1$ matrices $M_1^{(n)}$, $M_1^{(d)}$, $M_2^{(n)}$, $M_2^{(d)}$

such that

$$M_{\mathcal{P}_1, \mathcal{I}_1, \mathcal{S}_1}^{(n)} = M_{\mathcal{P}_2, \mathcal{I}_1^*, \mathcal{S}_1^*}^{(n)} \cdot M_1^{(n)}(\mathcal{I}_1, \mathcal{s}_1) + M_2^{(n)}(\mathcal{I}_1, \mathcal{s}_1),$$

$$\text{val}_{\mathcal{I}_1}^{(d)}(\mathcal{s}_1) = \text{val}_{\mathcal{I}_1^*}^{(d)}(\mathcal{s}_1^*) \cdot M_1^{(d)}(\mathcal{I}_1, \mathcal{S}_1) + M_2^{(d)}(\mathcal{I}_2, \mathcal{S}_2),$$
Low rank non-affine reductions between fractional problems

Definition

A reduction from $\mathcal{P}_1$ to $\mathcal{P}_2$ consists of

- Two maps $\star : \mathcal{I}_1 \rightarrow \mathcal{I}_2$ and $\star : \mathcal{S}_1 \rightarrow \mathcal{S}_2$
- Four nonnegative $\mathcal{I}_1 \times \mathcal{S}_1$ matrices $M_{1}^{(n)}, M_{1}^{(d)}, M_{2}^{(n)}, M_{2}^{(d)}$

such that

$$M_{\mathcal{P}_1, c_1, S_1}^{(n)}(\mathcal{I}_1, S_1) = M_{\mathcal{P}_2, c_2, S_2}^{(n)}(\mathcal{I}_1^*, \mathcal{S}_1^*) \cdot M_{1}^{(n)}(\mathcal{I}_1, s_1) + M_{2}^{(n)}(\mathcal{I}_1, s_1),$$

$$\text{val}_{\mathcal{I}_1}^{d}(s_1) = \text{val}_{\mathcal{I}_1^*}^{d}(s_1^*) \cdot M_{1}^{(d)}(\mathcal{I}_1, S_1) + M_{2}^{(d)}(\mathcal{I}_2, S_2),$$

Matrices $M_{1}^{(n)}, M_{1}^{(d)}, M_{2}^{(n)}, M_{2}^{(d)}$ can be arbitrary functions of the solutions and instances.
Theorem

Let \((P_1, C_1, S_1)\) and \((P_2, C_2, S_2)\) be two fractional problems with a reduction from \(P_1\) to \(P_2\). Then

\[
fc_{\text{LP}}(P_1) \leq \text{rk}_{\text{LP}} \begin{bmatrix} M_2^{(n)} \\ M_2^{(d)} \end{bmatrix} + \text{rk}_{\text{LP}} \begin{bmatrix} M_1^{(n)} \\ M_1^{(d)} \end{bmatrix} + \text{rk}_{\text{LP}} \begin{bmatrix} M_1^{(n)} \\ M_1^{(d)} \end{bmatrix} \cdot fc_{\text{LP}}(P_2),
\]

where \(M_1^{(n)}, M_2^{(d)}, M_2^{(n)}, M_2^{(d)}\) are the matrices in the reduction.
## Summary of Results

<table>
<thead>
<tr>
<th>Problem</th>
<th>Factor</th>
<th>Source</th>
<th>Paradigm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MaxCut</strong></td>
<td>15/16 + ( \varepsilon )</td>
<td>Max-3-XOR/0</td>
<td>SDP</td>
</tr>
<tr>
<td><strong>SparsestCut,</strong></td>
<td>2 - ( \varepsilon )</td>
<td>MaxCut</td>
<td>LP</td>
</tr>
<tr>
<td>( \text{tw(supply)} = O(1) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SparsestCut,</strong></td>
<td>16/15 - ( \varepsilon )</td>
<td>MaxCut</td>
<td>SDP</td>
</tr>
<tr>
<td>( \text{tw(supply)} = O(1) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>BalancedSeparator,</strong></td>
<td>( \omega(1) )</td>
<td>UniqueGames</td>
<td>LP</td>
</tr>
<tr>
<td>( \text{tw(demand)} = O(1) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>IndependentSet</strong></td>
<td>( \omega(n^{1-\varepsilon}) )</td>
<td>Max-( k )-CSP</td>
<td>Lasserre</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( O(n^\varepsilon) ) rounds</td>
</tr>
<tr>
<td><strong>Matching, 3-regular</strong></td>
<td>1 + ( \varepsilon/n^2 )</td>
<td>Matching</td>
<td>LP</td>
</tr>
</tbody>
</table>
• **SPARSEST CUT** with bounded tree-width on the demand graph.

Opener Problems
Open Problems

- \textsc{SparsestCut} with bounded tree-width on the demand graph.

- Stronger lower bound for \textsc{IndependentSet}. There is an $O(n)$ sized LP due to [FJ14], hence cannot expect $\Omega(n^{1-\epsilon})$ as in the PCP theorem.
Open Problems

- **SPARSEST CUT** with bounded tree-width on the demand graph.

- Stronger lower bound for **INDEPENDENT SET**. There is an $O(n)$ sized LP due to [FJ14], hence cannot expect $\Omega(n^{1-\varepsilon})$ as in the PCP theorem.

- LP and SDP equivalent of *gap amplification* [Din07].
Section II: Upper bounds
Hierarchical clustering

Definition

- Data set $V$ and a \textit{pairwise similarity} $\kappa : V \times V \rightarrow \mathbb{R}_{\geq 0}$
- \textit{Hierarchical clustering}: tree $T$ with root $r$ such that $\text{leaves}(T) = V$.

Cost function due to [Das16]:

$$\text{cost}(T) := \sum_{\{i,j\} \in E(K_n)} \kappa(i, j) \cdot |\text{leaves}(T[lca(i, j)])|.$$
Hierarchical clustering

Definition

- Data set $V$ and a *pairwise similarity* $\kappa : V \times V \rightarrow \mathbb{R}_{\geq 0}$
- *Hierarchical clustering*: tree $T$ with root $r$ such that $\text{leaves}(T) = V$.

Cost function due to [Das16]:

$$\text{cost}(T) := \sum_{\{i,j\} \in E(K_n)} \kappa(i,j) |\text{leaves}(T[lca(i,j)])|.$$  

Top-down algorithm using $\text{SPARSESTCUT}$ gives $O(\log^{3/2} n)$ approximation using [ARV09].
A combinatorial characterization of ultrametrics

- **Ultrametric**: $d(i, j) \leq \max\{d(i, k), d(k, j)\}$.
- $d(i, j) := |\text{leaves}( T[\text{lca}(i, j)] )| - 1$ is an *ultrametric* on $V$.
- **Natural question**: can we characterize such ultrametrics?

**Definition**

*Non-trivial* ultrametric:

1. $\forall S \subseteq V, \exists i, j \in S$ such that $d(i, j) \geq |S| - 1$.
2. $\forall t$ if $S_t$ is an *equivalence class* of $V$ under the relation $i \sim j 
\iff d(i, j) \leq t$, then $\max_{i,j \in S_t} d(i, j) \leq |S_t| - 1$. 

An ILP for non-trivial ultrametrics

\[
\begin{align*}
\min_{t=1}^{n-1} & \sum_{\{i,j\} \in E(K_n)} \kappa(i, j) x_{ij}^t \\
\text{s.t.} & \quad x_{ij}^t \geq x_{ij}^{t+1} \\
& \quad x_{ij}^t + x_{jk}^t \geq x_{ik}^t \\
& \quad \sum_{i,j \in S} x_{ij}^t \geq 1 \\
& \quad \sum_{i,j \in S} x_{ij}^{|S|} \leq |S| \left( \sum_{i,j \in S} x_{ij}^t + \sum_{i \in S} \sum_{j \notin S} (1 - x_{ij}^t) \right) \\
x_{ij}^t & \in \{0, 1\}
\end{align*}
\]
Rounding an LP relaxation

- Use iterative rounding scheme using the idea of *sphere growing* in a layer-wise manner.
- Use combinatorial interpretation of feasible solutions for ILP to demonstrate feasibility.

**Theorem**

The rounding algorithm computes a hierarchical clustering $T$ of $V$ satisfying $\text{cost}(T) \leq O(\log n) \min_{T' \in \mathcal{T}} \text{cost}(T')$ in time polynomial in $n$ and $\log \left( \max_{i,j \in V} \kappa(i,j) \right)$. 
• Improve approximation to $O(\sqrt{\log n \log \log n})$ due to similarities with minimum linear arrangement problem.
Open Problems

- Improve approximation to $O(\sqrt{\log n \log \log n})$ due to similarities with the minimum linear arrangement problem.

- Matching lower bounds for this problem (so far we can show $\Omega(1)$ inapproximability for LPs and SDPs).
Open Problems

- Improve approximation to $O(\sqrt{\log n \log \log n})$ due to similarities with *minimum linear arrangement* problem.

- Matching lower bounds for this problem (so far we can show $\Omega(1)$ inapproximability for LPs and SDPs).

- Combinatorial algorithm instead of Ellipsoid using the framework of [PST95, You95].
Thank you!
References


