Approximation Algorithms

Martin Böhm (University of Bremen)

April 30, 2019

Linear programming and rounding

Imagine a tool...

- where any problem you define automatically has a PTIME algorithm;
- that gives you not only a solution, but also a certificate with which you can use to quickly verify optimality;
- that is P-complete – it can model any PTIME decision problem;
- that can be used *(surprise)* to design approximation algorithms for some hard optimization problems.

That tool? **Linear programming.**

Definition of an LP

Definition

A linear program is a model of an optimization problem over real-valued variables using linear constraints and a linear objective function.

- A linear function: $c_1 \cdot x_1 + c_2 \cdot x_2 + c_3 \cdot x_3 + \cdots + c_n x_n$. c_i are *constants*, x_i are *variables*.
- A linear constraint: $c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \leq b$. b is a *right-hand-side constant*.

In other words...

Definition

A linear program is a model of an optimization problem over real-valued variables using linear constraints and a linear objective function.

Typical LP:

$$
\begin{aligned}
\min \quad & c_1x_1 + c_2x_2 + \ldots + c_nx_n \\
\text{s.t.} \quad & a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n \geq b_1 \\
& a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,n}x_n \geq b_2 \\
& \ldots \\
& a_{m,1}x_1 + a_{m,2}x_2 + \ldots + a_{m,n}x_n \geq b_n \\
& \forall i\colon x_i \geq 0, x_i \in \mathbb{R}
\end{aligned}
$$

Compact form:

$$
\begin{aligned}
\min \ & \boldsymbol{c}^T \boldsymbol{x} \\
\text{s.t.} \ & A\boldsymbol{x} \geq \boldsymbol{b} \\
& \boldsymbol{x} \geq 0
\end{aligned}
$$

Compact form:

$$\begin{aligned} &\min \boldsymbol{c}^T \boldsymbol{x} \\ &\text{s.t. } A\boldsymbol{x} \geq \boldsymbol{b} \\ &\quad\quad \boldsymbol{x} \geq 0 \end{aligned}$$

Theorem

An optimal vector x^* to a linear program can be computed in polynomial time with respect to the program size.

Notes.

- We can use any linear inequalities we want – $\geq, \leq$ and $=$.
- We can use maximum instead of minimum.
- If our model has no feasible solution, the solver will inform us about this also in polynomial time.
- For optimization problems that arise from NP-complete problems, it is unlikely we will find a linear programming formulation. *However!*

- For optimization problems that arise from NP-complete problems, it is unlikely we will find a linear programming formulation. However, there is a very close modelling tool for those:

Definition (Integer linear program)

Compact form:

$$\begin{aligned} \min\ & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.}\ & A\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{x} \geq 0, \boldsymbol{x} \in \mathbb{N} \end{aligned}$$

Theorem

Deciding whether a given integer linear program has at least one feasible solution is NP-complete.

How to model with LPs

There is no *correct* way to model a problem with a linear program.
In fact, many problems have several useful LP formulations.
There is no guarantee that a *nice* (readable, reasonable, etc.) model even exists.

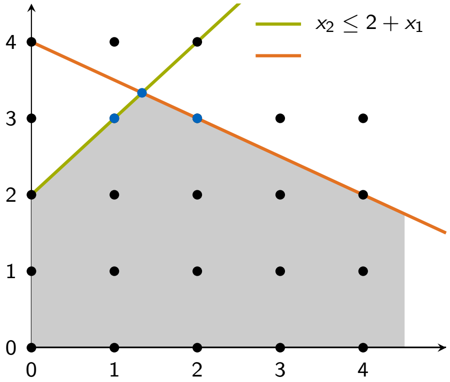
Useful questions at the beginning:

1. **Q:** What should our variables $\boldsymbol{x}$ be?
2. **Q:** How should we model the objective function?

Example 0: A numerical problem

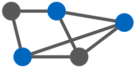
LP relaxation

$$
\begin{aligned}
\max \quad & x_2 \\
\text{s.t.} \quad & x_2 \le 2 + x_1 \\
& x_2 \le 4 - \tfrac{1}{2}x_1 \\
& x_i \in \mathbb{N} && \forall i \in \{1, 2\}
\end{aligned}
$$



Example 1: Vertex Cover

Problem WEIGHTED VERTEX COVER:



Input: An undirected, weighted graph $G = (V, E)$.
Every vertex $v \in V$ has an associated weight $w_v \geq 0$.

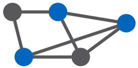
Output: A cover $C \subseteq V$ such that every edge $uv \in E$ has either u or v *(or both)* in C.

Goal: Find a cover C minimizing the total weight $\sum_{v \in C} w_v$.

Q: What should our variables $\boldsymbol{x}$ be?

A: For every vertex $v \in V$, we have $x_v \in \{0, 1\}$ – whether v is in the cover or not.

Problem Weighted Vertex Cover:



Input: An undirected, weighted graph $G = (V, E)$.
Every vertex $v \in V$ has an associated weight $w_v \geq 0$.

Output: A cover $C \subseteq V$ such that every edge $uv \in E$ has either u or v *(or both)* in C.

Goal: Find a cover C minimizing the total weight $\sum_{v \in C} w_v$.

1. Objective function: $\min \sum_{v \in V} w_v x_v$.
2. Constraints: For every edge $ij \in E(G)$, one constraint: $x_i + x_j \geq 1$.
3. Variable space for integer linear program: $x_v \in \{0, 1\}$.
4. Variable space for linear relaxation: $x_v \geq 0, x_v \leq 1, x_v \in \mathbb{R}$.

$$\begin{aligned} \text{LP relaxation:} \qquad & \min \sum_{v \in V} w_v x_v \\ \text{s.t.} \qquad & \forall ij \in E(G)\colon x_i + x_j \geq 1 \\ & x_v \geq 0, x_v \leq 1, x_v \in \mathbb{R} \end{aligned}$$

Theorem

There exists a 2-approximation algorithm for WEIGHTED VERTEX COVER.

Approach:

1. Compute the optimum LP solution x^*. This solution may (will) be fractional.
2. Round x^* up or down to integer $\{0, 1\}$ values $\rightarrow y$.
3. Prove that y is a feasible solution (a valid vertex cover).
4. Prove that $y \leq 2x^*$. This implies $y \leq 2OPT$.

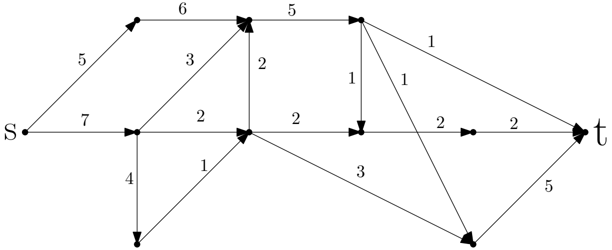
Example 2: Maximum flow

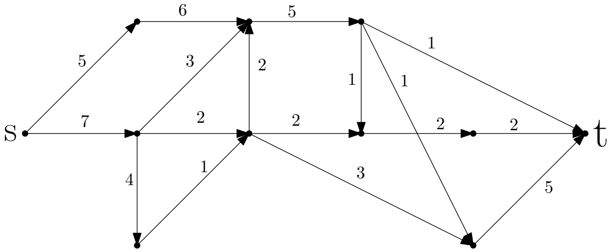
Problem MAXIMUM FLOW:

Input: An edge-weighted, directed graph $G = (V, E, c)$ (a network). Every directed edge has an associated capacity $c_e \geq 0$. Additionally, there are two special vertices – source s and sink t.

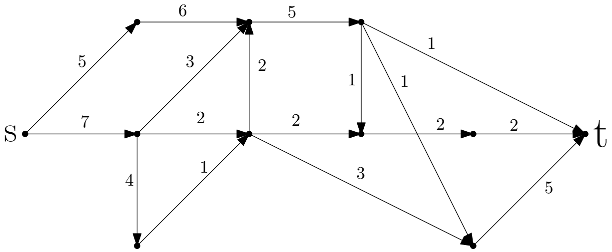
Output: Any flow – function $f : E \to \mathbb{R}_0^+$ which *obeys capacities* ($f(e) \leq c_e$) and *follows Kirchhoff's law* – in every non-special vertex, fluid coming in = fluid coming out.

Goal: Find a flow that sends as much fluid as possible from s to t.





Q: What should our variables $\boldsymbol{x}$ be?
A: Flow on one given edge $x_{\vec{e}}$, $x_{\vec{e}} \in \mathbb{R}_0^+$.
Q: How should we model "*as much fluid as possible*"?
A: $\max \sum_{\vec{si}} x_i$.



$$\max \sum_{\vec{si}} x_{\vec{si}}$$

$$\begin{aligned} \text{s.t.} \quad & \forall \vec{ij} \in E(G): & x_{\vec{ij}} \leq c_{\vec{ij}} \\ & \forall v \in V(G) \setminus \{s,t\}: & \sum_{\vec{vi}} x_{\vec{vi}} - \sum_{\vec{jv}} x_{\vec{jv}} = 0 \\ & \forall \vec{ij} \in E(G): & x_{\vec{ij}} \geq 0, x_{\vec{ij}} \in \mathbb{R} \end{aligned}$$

Theorem

MAXIMUM FLOW *is solvable in polynomial time.*