Graph Theory and Optimization
Approximation Algorithms

Nicolas Nisse

Université Côte d'Azur, Inria, CNRS, I3S, France

October 2018

Thank you to F. Giroire for some of the slides

Motivation

- Goal:
  - Find "good" solutions for difficult problems (NP-hard).
  - Be able to quantify the "goodness" of the given solution.
- Presentation of a technique to get approximation algorithms: fractional relaxation of integer linear programs.

Outline

1. Approximation Algorithms
2. Example: Max. Matching vs. Min. Vertex Cover
3. Approximation algorithms using Fractional Relaxation
   - Vertex Cover
   - Set Cover

Approximation Algorithms

Π a maximization Problem

c-Approximation for Π $1 < c$ constant or depends on input length

- deterministic polynomial-time algorithm $\mathscr{A}$
- for any input I, $\mathscr{A}$ returns a solution with value at least $OPT(I)/c$.

Π a minimization Problem

c-Approximation for Π $1 < c$ constant or depends on input length

- deterministic polynomial-time algorithm $\mathscr{A}$
- for any input I, $\mathscr{A}$ returns a solution with value at most $c \cdot OPT(I)$.

Approximation Algorithms

Definition: An approximation algorithm produces

- in polynomial time
- a feasible solution
- whose objective function value is close to the optimal OPT, by close we mean within a guaranteed factor of the optimal.

Example: a factor 2 approximation algorithm for the cardinality vertex cover problem, i.e. an algorithm that finds a cover of cost $\leq 2 \cdot OPT$ in time polynomial in $|V|$.

Outline

Approx: Max. Matching vs. Min. Vertex Cover

Let $G=(V,E)$ be a graph

Matching: set M of pairwise disjoint edges in a graph $(M \subseteq E)$

Compute a Max. Matching is polynomial-time solvable [Edmonds 1965]

Vertex Cover: set $K \subseteq V$ such that $\forall e \in E$, $e \cap K \neq \emptyset$
set of vertices that "touch" every edge

Compute a Min. Vertex Cover is NP-complete [Garey,Johnson 1979]

Approx: Max. Matching vs. Min. Vertex Cover

Let $G = (V,E)$ be a graph

Matching: set M of pairwise disjoint edges in a graph $(M \subseteq E)$

Compute a Max. Matching is polynomial-time solvable [Edmonds 1965]

Vertex Cover: set $K \subseteq V$ such that $\forall e \in E$, $e \cap K \neq \emptyset$

set of vertices that "touch" every edge

Compute a Min. Vertex Cover is NP-complete [Garey,Johnson 1979]

Exercise: Prove that for any graph G,

$$maxMatching(G) \leq minCover(G) \leq 2 \cdot maxMatching(G)$$

Deduce a (polynomial-time) 2-approximation algorithm for computing $minCover(G)$

Approx: Max. Matching vs. Min. Vertex Cover

Solution of previous exercise

Theorem: for any graph G

$$maxMatching(G) \leq minCover(G) \leq 2 \cdot maxMatching(G)$$

Proof: Let $K \subseteq V$ be a cover of G and $M \subseteq E$ be a matching of G.
By definition of K: $K \cap e \neq \emptyset$ for any $e \in M$
Moreover, by definition of M, $e \cap f = \emptyset$ for any $e, f \in M$

$$\Rightarrow |M| \leq |K|.$$

Let $M \subseteq E$ be a maximum matching of G
Then $K = \{v \mid \exists e \in M, v \in e\}$ is a cover of G (if not, M is not maximum)

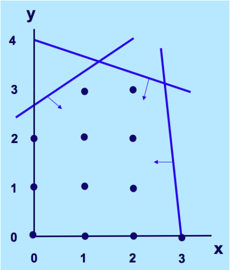
$$\Rightarrow minCover(G) \leq |K| = 2 \cdot |M|$$

Outline

1. Approximation Algorithms
2. Example: Max. Matching vs. Min. Vertex Cover
3. Approximation algorithms using Fractional Relaxation
   - Vertex Cover
   - Set Cover

Approximation via Fractional Relaxation

- Reminder:
  - Integer Linear Programs often hard to solve (NP-hard).
  - Linear Programs (with real numbers) easier to solve (polynomial-time algorithms).
- Idea:
  - 1- Relax the integrality constraints;
  - 2- Solve the (fractional) linear program and then;
  - 3- Round the solution to obtain an integral solution.



2-Approximation for Vertex Cover using LP

Let $G = (V, E)$ be a graph

Vertex Cover: set $K \subseteq V$ such that $\forall e \in E$, $e \cap K \neq \emptyset$

set of vertices that "touch" every edge

Integer Linear programme (*ILP*):

$$\begin{array}{llcll} \text{Min.} & \sum_{v \in V} x_v & & & \\ \text{s.t.:} & x_v + x_u & \geq & 1 & \forall \{u,v\} \in E \\ & x_v & \in & \{0,1\} & \forall v \in V \end{array}$$

Fractional relaxation (*LP*):

$$\begin{array}{llcll} \text{Min.} & \sum_{v \in V} x_v & & & \\ \text{s.t.:} & x_v + x_u & \geq & 1 & \forall \{u,v\} \in E \\ & x_v & \geq & 0 & \forall v \in V \end{array}$$

Exercise: Prove that the LP has an half-integral optimal solution (i.e., $x_v \in \{0, 1/2, 1\}$)

Exercise: Deduce a 2-approximation algorithm for Min. Vertex Cover

2-Approximation for Vertex Cover using LP

Let $G=(V,E)$ be a graph

Vertex Cover: set $K \subseteq V$ such that $\forall e \in E$, $e \cap K \neq \emptyset$

set of vertices that "touch" every edge

Integer Linear programme (*ILP*):

$$\begin{array}{llcll} \text{Min.} & \sum_{v \in V} x_v & & & \\ \text{s.t.:} & x_v + x_u & \geq & 1 & \forall \{u,v\} \in E \\ & x_v & \in & \{0,1\} & \forall v \in V \end{array}$$

Fractional relaxation (*LP*):

$$\begin{array}{llcll} \text{Min.} & \sum_{v \in V} x_v & & & \\ \text{s.t.:} & x_v + x_u & \geq & 1 & \forall \{u,v\} \in E \\ & x_v & \geq & 0 & \forall v \in V \end{array}$$

Exercise: Prove that the LP has an half-integral optimal solution

(i.e., $x_v \in \{0, 1/2, 1\}$)

Exercise: Deduce a 2-approximation algorithm for Min. Vertex Cover

2-Approximation for Vertex Cover using LP

Let $G=(V,E)$ be a graph

Vertex Cover: set $K \subseteq V$ such that $\forall e \in E$, $e \cap K \neq \emptyset$

set of vertices that "touch" every edge

Integer Linear programme (*ILP*):

$$\begin{array}{llcrr} \text{Min.} & \sum_{v \in V} x_v & & & \\ \text{s.t.:} & x_v + x_u & \geq & 1 & \forall \{u,v\} \in E \\ & x_v & \in & \{0,1\} & \forall v \in V \end{array}$$

Fractional relaxation (*LP*):

$$\begin{array}{llcrr} \text{Min.} & \sum_{v \in V} x_v & & & \\ \text{s.t.:} & x_v + x_u & \geq & 1 & \forall \{u,v\} \in E \\ & x_v & \geq & 0 & \forall v \in V \end{array}$$

Exercise: Prove that the LP has an half-integral optimal solution

(i.e., $x_v \in \{0, 1/2, 1\}$)

Exercise: Deduce a 2-approximation algorithm for Min. Vertex Cover

2-Approximation for Vertex Cover using LP

Theorem: Fractional Vertex Cover has an half-integral optimal solution

Proof: $\mathbf{y}$: optimal solution with the largest number of coordinates in $\{0,1/2,1\}$.
For purpose of contradiction: $\mathbf{y}$ not half-integral: Set
$\varepsilon = \min\{y_v, |y_v - \frac{1}{2}|, 1 - y_v \mid v \in V \text{ and } y_v \notin \{0,1/2,1\}\}$.
Consider $\mathbf{y}'$ and $\mathbf{y}''$, feasible solutions, defined as follows:

$$y'_v = \begin{cases} y_v - \varepsilon, & \text{if } 0 < y_v < \frac{1}{2}, \\ y_v + \varepsilon, & \text{if } \frac{1}{2} < y_v < 1, \\ y_v, & \text{otherwise.} \end{cases} \quad \text{and } y''_v = \begin{cases} y_v + \varepsilon, & \text{if } 0 < y_v < \frac{1}{2}, \\ y_v - \varepsilon, & \text{if } \frac{1}{2} < y_v < 1, \\ y_v, & \text{otherwise.} \end{cases}$$

$\sum_{v \in V} y_v = \frac{1}{2}(\sum_{v\in V} y'_v + \sum_{v \in V} y''_v)$. $\mathbf{y}'$ and $\mathbf{y}''$ are also optimal solutions.
By choice of ε, $\mathbf{y}'$ and $\mathbf{y}''$ has more coordinates in $\{0,1/2,1\}$ than $\mathbf{y}$, a contradiction.

Theorem: 2-Approximation of Vertex Cover

Proof: First solve Fractional Vertex Cover and derive an half-integral optimal solution $\mathbf{y}^f$ to it. Define $\mathbf{y}$ by $y_v = 1$ if and only if $y^f_v \in \{1/2;1\}$, i.e., $y_v = \lceil y^f_v \rceil$
Clearly, $\mathbf{y}$ is an admissible solution of Vertex Cover. Moreover, by definition

$$\sum_{v \in V} y_v \leq 2 \sum_{v \in V} y^f_v = 2 \cdot \nu^f(G) \leq 2 \cdot \nu(G).$$

Set Cover

- Problem: Given a universe U of n elements, a collection of subsets of U, $\S = S_1, ..., S_k$, and a cost function $c : S \rightarrow Q^+$, find a minimum cost subcollection of S that covers all elements of U.
- Model numerous classical problems as special cases of set cover: vertex cover, minimum cost shortest path...
- Definition: The frequency of an element is the number of sets it is in. The frequency of the most frequent element is denoted by f.
- Various approximation algorithms for set cover achieve one of the two factors $O(\log n)$ or f.

Fractional relaxation

Write a linear program to solve set cover.

Var.:	$x_S = 1$ if S picked in $\mathscr{C}$,	
	$x_S = 0$ otherwise	
min	$\sum_{S \in \S} c(S) x_S$	
s. t.		
	$\sum_{S : e \in S} x_S \geq 1$	$(\forall e \in U)$
	$x_S \in \{0,1\}$	$(\forall S \in \S)$

Var.:	$1 \geq x_S \geq 0$	
min	$\sum_{S \in \S} c(S) x_S$	
s. t.		
	$\sum_{S : e \in S} x_S \geq 1$	$(\forall e \in U)$
	$x_S \geq 0$	$(\forall S \in \S)$

Fractional relaxation

Write a linear program to solve set cover.

Var.:	$x_S = 1$ if S picked in $\mathscr{C}$,
	$x_S = 0$ otherwise
min	$\sum_{S \in \S} c(S) x_S$
s. t.	
	$\sum_{S : e \in S} x_S \geq 1 \quad (\forall e \in U)$
	$x_S \in \{0,1\} \quad (\forall S \in \S)$

Var.:	$1 \geq x_S \geq 0$
min	$\sum_{S \in \S} c(S) x_S$
s. t.	
	$\sum_{S : e \in S} x_S \geq 1 \quad (\forall e \in U)$
	$x_S \geq 0 \quad (\forall S \in \S)$

Fractional relaxation

Write a linear program to solve set cover.

Var.:	$x_S = 1$ if S picked in $\mathscr{C}$, $x_S = 0$ otherwise	
min	$\sum_{S \in \S} c(S) x_S$	
s. t.	$\sum_{S: e \in S} x_S \geq 1$	$(\forall e \in U)$
	$x_S \in \{0,1\}$	$(\forall S \in \S)$

Var.:	$1 \geq x_S \geq 0$	
min	$\sum_{S \in \S} c(S) x_S$	
s. t.	$\sum_{S: e \in S} x_S \geq 1$	$(\forall e \in U)$
	$x_S \geq 0$	$(\forall S \in \S)$

Fractional relaxation

- The (fractional) optimal solution of the relaxation is a lower bound of the optimal solution of the original integer linear program.

- Example in which a fractional set cover may be cheaper than the optimal integral set cover:
  Input: $U = \{e, f, g\}$ and the specified sets $S_1 = \{e, f\}$, $S_2 = \{f, g\}$, $S_3 = \{e, g\}$, each of unit cost.
  - An integral cover of cost 2 (must pick two of the sets).
  - A fractional cover of cost 3/2 (each set picked to the extent of 1/2).

Fractional relaxation

- The (fractional) optimal solution of the relaxation is a lower bound of the optimal solution of the original integer linear program.
- Example in which a fractional set cover may be cheaper than the optimal integral set cover:
  Input: $U = \{e, f, g\}$ and the specified sets $S_1 = \{e, f\}$, $S_2 = \{f, g\}$, $S_3 = \{e, g\}$, each of unit cost.
  - An integral cover of cost 2 (must pick two of the sets).
  - A fractional cover of cost 3/2 (each set picked to the extent of 1/2).

A simple rounding algorithm

Algorithm:

- 1- Find an optimal solution to the LP-relaxation.
- 2- (Rounding) Pick all sets S for which $x_S \geq 1/f$ in this solution.

- **Theorem:** The algorithm achieves an approximation factor of f for the set cover problem.
- **Proof:** To be proved:
  - 1) All elements are covered.
  - 2) The cover returned by the algorithm is of cost at most $f \cdot OPT$

  Proofs:
  - proof of 1) All elements are covered. e is in at most f sets, thus one of this set must be picked to the extent of at least 1/f in the fractional cover.
  - proof of 2) The rounding process increases x_S by a factor of at most f. Therefore, the cost of $\mathscr{C}$ is at most f times the cost of the fractional cover.

$$OPT_f \leq OPT \leq f \cdot OPT_f$$

- Theorem: The algorithm achieves an approximation factor of f for the set cover problem.
- Proof: To be proved:
  - 1) All elements are covered.
  - 2) The cover returned by the algorithm is of cost at most $f \cdot OPT$

Proofs:

- proof of 1) All elements are covered. e is in at most f sets, thus one of this set must be picked to the extent of at least 1/f in the fractional cover.
- proof of 2) The rounding process increases x_S by a factor of at most f. Therefore, the cost of $\mathscr{C}$ is at most f times the cost of the fractional cover.

$$OPT_f \leq OPT \leq f \cdot OPT_f$$

- Theorem: The algorithm achieves an approximation factor of f for the set cover problem.
- Proof: To be proved:
  - 1) All elements are covered.
  - 2) The cover returned by the algorithm is of cost at most $f \cdot OPT$

Proofs:

- proof of 1) All elements are covered. e is in at most f sets, thus one of this set must be picked to the extent of at least 1/f in the fractional cover.
- proof of 2) The rounding process increases x_S by a factor of at most f. Therefore, the cost of $\mathscr{C}$ is at most f times the cost of the fractional cover.

$$OPT_f \leq OPT \leq f \cdot OPT_f$$

- Theorem: The algorithm achieves an approximation factor of f for the set cover problem.

- Proof: To be proved:
  - 1) All elements are covered.
  - 2) The cover returned by the algorithm is of cost at most $f \cdot OPT$

Proofs:

- proof of 1) All elements are covered. e is in at most f sets, thus one of this set must be picked to the extent of at least 1/f in the fractional cover.

- proof of 2) The rounding process increases x_S by a factor of at most f. Therefore, the cost of $\mathscr{C}$ is at most f times the cost of the fractional cover.

$$OPT_f \leq OPT \leq f \cdot OPT_f$$

Randomized rounding

- Idea: View the optimal fractional solutions as probabilities.
- Algorithm:
  - Flip coins with biases and round accordingly (S is in the cover with probability x_S).
  - Repeat the rouding $O(\log n)$ times.
- This leads to an $O(\log n)$ factor randomized approximation algorithm. That is
  - The set is covered with high probability.
  - The cover has expected cost: $O(\log n)OPT$.

Take Aways

- Fractional relaxation is a method to obtain for some problems:
  - Lower bounds on the optimal solution of an integer linear program (minimization).
    Remark: Used in Branch & Bound algorithms to cut branches.
  - Polynomial approximation algorithms (with rounding).
- Complexity:
  - Integer linear programs are often hard.
  - (Fractional) linear programs are quicker to solve (polynomial time).