Advanced Techniques for Combinatorial Algorithms: Approximation Algorithms

Gianluca Della Vedova

Univ. Milano–Bicocca
https://gianluca.dellavedova.org

May 11, 2020

NPO

Optimization problem

- Infinite set $\mathcal{I}$ of instances. The set $\mathcal{I}$ is recognizable in polynomial time
- For each instance $I \in \mathcal{I}$, the set $F(I)$ of feasible solutions. Each set $F(I)$ is recognizable in polynomial time. The set of all feasible solutions is $\mathcal{F}$
- An objective function $w : \mathcal{I} \times \mathcal{F} \mapsto \mathbb{Q}^+$. w is a partial function — $w(i, x)$ can be undefined if $x \notin F(i)$. w is computable in polynomial time
- Goal: to minimize or to maximize

Approximation factor

$$\frac{APX}{OPT}$$

APX: value of (approximate) feasible solution, OPT: value of best feasible solution

Min Vertex Cover

Instance

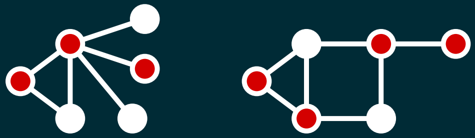
Undirected graph $G = \langle V, E \rangle$

Feasible solutions

A set $C \subset V$ such that for each edge $e \in E$ at least one endpoint of e belongs to C

Objective function

$|C|$



Max Clique

Instance

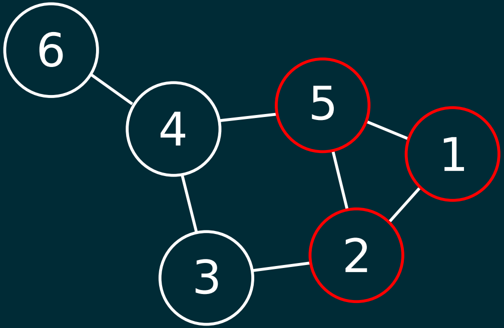
Undirected graph $G = \langle V, E \rangle$

Feasible solution

Find a set $C \subset V$ such that all pairs of vertices in C are connected by an edge

Objective function

$|C|$

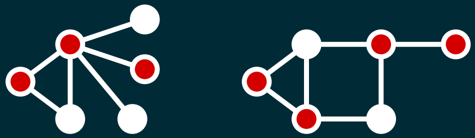


Max Independent Set

Instance
Undirected graph $G = \langle V, E \rangle$

Feasible solution
Find a set $I \subset V$ such that no two vertices in K are connected by an edge

Objective function
$|K|$



Max Cut

Instance

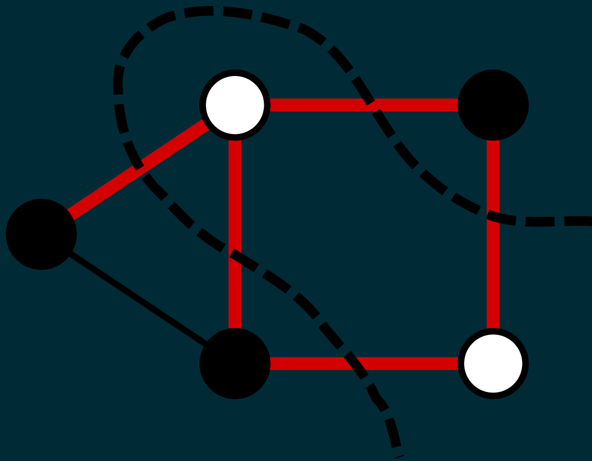
A weighted undirected graph $G = \langle V, E \rangle$, $w : E \mapsto \mathbb{Q}^+$

Feasible solution

a bipartition (V_1, V_2) of V

Objective function

$\sum_{v_1 \in V_1, v_2 \in V_2} w(v_1, v_2)$



Min Traveling Salesperson (TSP)

Instance

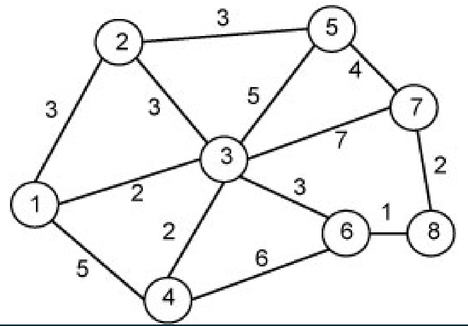
A weighted undirected graph $G = \langle V, E \rangle$, $w : E \mapsto \mathbb{Q}^+$

Feasible solution

Find a cycle C that visits each vertex $v \in V$ exactly once.

Objective function

$\sum_{e \in C} w(e)$



Approximation Goal

Complexity classes

- NPO: Optimization problems in NP
- FPTAS: Fully polynomial-time approximation scheme. Guaranteed error ratio $(1+\epsilon)$ or $(1-\epsilon)$, for any $\epsilon > 0$. Time complexity polynomial in n and $\frac{1}{\epsilon}$
- PTAS: Polynomial-time approximation scheme. Guaranteed error ratio $(1+\epsilon)$ or $(1-\epsilon)$, for any $\epsilon > 0$. Time complexity polynomial in n — can be exponential in $\frac{1}{\epsilon}$, e.g. $O(n^{1/\epsilon})$
- APX: $O(1)$ approximation ratio, polytime
- MAX SNP: Definition based on logic and L-reduction. MAX SNP is included in APX

Min Set Cover

Instance

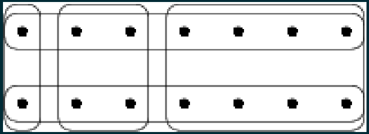
Universe set U, collection $\mathcal{S} = \{S_1, \ldots, S_n\}$of subsets of U. Weight $w : \mathcal{S} \mapsto \mathbb{Q}^+$

Feasible solutions

A cover, that is a subcollection $\mathcal{C}$ of $\mathcal{S}$ that covers all elements of U

Objective function

$\sum_{C \in \mathcal{C}} w(C)$



Min Set Cover

Algorithm 1: greedy-set-cover

```
C, D ← ∅;
while C ≠ U do
    X ← the set in 𝒮 minimizing w(X)/|X \ C|;
    α = w(X)/|X\C|;
    Add X to D;
    For each e ∈ C \ X, p(e) ← α;
    C ← C ∪ X
Output D
```

Lemma

$p(e_k) \leq \frac{OPT}{n-k+1}$

Corollary

Approximation factor is $1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \leq O(\log n)$

Min Metric Steiner Tree

Instance

A weighted undirected graph $G = \langle V, E \rangle$, $w : E \mapsto \mathbb{Q}^+$, w with triangle inequality. V partition into R (required) and S (steiner)

Feasible solution

A subtree T of G that includes all required vertices.

Objective function

$\sum_{e \in T} w(e)$

Approximation

Spanning tree T of G

2-approximation

Euler tour of optimal solution T^*

Min Metric Traveling Salesperson (TSP)

Instance

A weighted undirected graph $G = \langle V, E \rangle$, $w : E \mapsto \mathbb{Q}^+$, w with triangle inequality.

Feasible solution

Find a cycle C that visits each vertex $v \in V$ exactly once.

Objective function

$\sum_{e \in C} w(e)$

2-approximation

Euler tour

$\frac{3}{2}$-approximation

Matching on odd-degree vertices of a spanning tree T

Shortest Superstring

Instance

$s_1, \ldots, s_m$: strings of length n.

Feasible solution

A superstring T, that is each s_i is a substring of T

Objective function

$|T|$

Shortest Superstring

Prefix graph

Arc s_i, s_j with weight $pref(s_i, s_j)$

Length of superstring

Cycle of prefix graph + overlap last and first string

Assignment problem = cycle cover

From $G = \langle V, E \rangle$ to G_2 with two copies U, W of V. For each edge $(v_i, v_j) \in E$, add two edges (u_i, w_i), (w_i, u_i) to G_2

Algorithm

- Concatenate all cycle covers
- 4-approximation

Knapsack

Instance

Universe set U, size $s : U \mapsto \mathbb{Z}^+$, profit $p : U \mapsto \mathbb{Z}^+$, capacity $B \in \mathbb{Z}^+$

Feasible solutions

A subset $K \subseteq U$, such that $\sum_{k \in K} s(k) \leq B$

Objective function

$\sum_{k \in K} p(k)$, to maximize

Knapsack

Algorithm

- Dynamic programming
- NP-hard
- $K(i, b)$: uses only $\{u_1, \ldots, u_i\}$, total size b
- pseudo-polynomial time
- Transform it into an approximation algorithm
- Scale down profits $p_1(u) = \lfloor p(u)\frac{n}{\epsilon \max\{p(u)\}} \rfloor$, move to dual problem
- Approximation factor $1 - \epsilon$, $\forall \epsilon > 0$
- Time polynomial in n and $\frac{1}{\epsilon}$
- *FPTAS*

Linear Programming

Basic facts

- The primal has finite optimum iff the dual has finite optimum
- Let x, y be two feasible solution of the primal and dual. Then x are y re both optimal if:
  1. $\forall j$: either $x_j = 0$ or $\sum_i a_{i,j} y_i = c_j$
  2. $\forall i$: either $y_i = 0$ or $\sum_j a_{i,j} x_j = b_i$

Min Vertex Cover

Integral version

$$\begin{aligned} \min \sum x_v & \quad \text{subject to} \\ x_v + x_w \geq 1 & \quad \forall (v, w) \in E \\ x_v \in \{0, 1\} & \quad \forall v \in V \end{aligned} \tag{1}$$

Fractional version

$$\begin{aligned} \min \sum x_v & \quad \text{subject to} \\ x_v + x_w \geq 1 & \quad \forall (v, w) \in E \\ 0 \leq x_v \leq 1 & \quad \forall v \in V \end{aligned} \tag{2}$$

Integrality ratio

$$sup_I \frac{OPT(I)}{OPT_f(I)} \tag{3}$$

over all instances I, where OPT is the integral optimum, OPT_f is the fractional optimum

Lemma

An LP-based approach cannot outperform the integrality ratio

Half Integrality of Vertex Cover

Fractional version

$$
\begin{aligned}
\min \sum x_v \qquad & \text{subject to} \\
x_v + x_w \quad & \forall (v, w) \in E \\
x_v \geq 0 \quad & \forall v \in V
\end{aligned}
\tag{4}
$$

Lemma

There exists an optimal solution with $x_v \in \{0, 1, \frac{1}{2}\}$

Half Integrality of Vertex Cover

Lemma

There exists an optimal solution with $x_v \in \{0, 1, \frac{1}{2}\}$

$$y_v = \begin{cases} x_v + \epsilon, \dfrac{1}{2} < x_v < 1 \\ x_v - \epsilon, 0 < x_v < \dfrac{1}{2} \end{cases} \quad (5) \qquad z_v = \begin{cases} x_v - \epsilon, \dfrac{1}{2} < x_v < 1 \\ x_v + \epsilon, 0 < x_v < \dfrac{1}{2} \end{cases} \quad (6)$$

Proof

$x = \frac{1}{2}(y + z)$. Choose ϵ sufficiently small, then y and z are both feasible

Dual Fitting for Greedy Set Cover

$C, D \leftarrow \emptyset$;
while $C \neq U$ **do**
    $X \leftarrow$ set in $\mathcal{S}$ with min $w(X)/|X \setminus C|$;
    $\alpha = \frac{w(X)}{|X \setminus C|}$;
    Add X to D;
    For each $e \in C \setminus X$, $p(e) \leftarrow \alpha$;
    $C \leftarrow C \cup X$
Output D

ILP

$$\min \sum_{S \in \mathcal{S}} w(S) \qquad \text{subject to}$$

$$\sum_{S : e \in S} x_S \geq 1 \quad \forall e \in U \qquad (7)$$

$$x_S \in \{0, 1\} \quad \forall S \in \mathcal{S}$$

Dual Fitting for Greedy Set Cover

Primal

$$
\begin{aligned}
\min \sum_{S \in \mathcal{S}} w(S) & \qquad \text{subject to} \\
\sum_{S : e \in S} x_S \geq 1 & \quad \forall e \in U \\
x_S \geq 0 & \quad \forall S \in \mathcal{S}
\end{aligned}
\tag{8}
$$

Dual

$$
\begin{aligned}
\max \sum_{e \in U} y_e & \qquad \text{subject to} \\
\sum_{e : e \in S} y_e \leq c(S) & \quad \forall S \in \mathcal{S} \\
y_e \geq 0 & \quad \forall S \in \mathcal{S}
\end{aligned}
\tag{9}
$$

Algorithm — ILP

$p(e) = y_e$

Not dual feasible

Dual Fitting for Greedy Set Cover

Fitting

$y_e = \frac{p(e)}{H_n}$, $H_n = \sum_{i=1}^{n} \frac{1}{i}$

Lemma

x, y are both feasible

Proof

Let $S \in \mathcal{S}$, $|S| = k$. Let $e_1, \ldots, e_k \in S$, same order as the algorithm. When inserting e_i, there are are least $k - i + 1$ uncovered elements of S. By choice of S, $p(e_i) \leq c(S)/(k - i + 1)$, hence $y_e = \frac{p(e_i)}{H_n} \leq \frac{c(S)/(k-i+1)}{H_n}$. Checking the constraint: $\sum_{i=1}^{k} y_{e_i} \leq \frac{c(S)}{H_n} \sum_{i=1}^{k} \frac{1}{i} = c(S)$

Max Cut

Integral version

$$
\begin{aligned}
\max \frac{1}{2} \sum_{i,j} w_{i,j} d_{i,j} & \qquad \text{subject to} \\
d_{i,j} \leq x_i + x_j & \quad \forall (v_i, v_j) \in E \\
d_{i,j} \leq 2 - (x_i + x_j) & \quad \forall (v_i, v_j) \in E \\
x_v, d_{i,j} & \in \{0, 1\}
\end{aligned} \tag{10}
$$

Max Cut

Second version

$$\max \frac{1}{2}\sum_{i,j} w_{i,j}(1 - x_i x_j) \qquad \text{subject to}$$
$$x_v^2 = 1 \quad \forall v \in V$$
$$x_v, d_{i,j} \in \{0,1\} \tag{11}$$

Fractional version

$$\max \frac{1}{2}\sum_{i,j} w_{i,j}(1 - x_i x_j) \qquad \text{subject to}$$
$$x_v^2 = 1 \quad \forall v \in V$$
$$-1 \leq x_v \leq 1 \quad \forall v \in V \tag{12}$$

Semidefinite programming

Vector version

$$
\begin{aligned}
\max \frac{1}{2} \sum_{i,j} w_{i,j}(1 - x_i \cdot x_j) \qquad & \text{subject to} \\
x_v \cdot x_v = 1 \quad & \forall v \in V \\
x_v \in \mathbb{R}^n \quad & \forall v \in V
\end{aligned}
\tag{13}
$$

How to solve?

Can be solved approximately (additive error ϵ) via interior point

Problem

From vector (fractional) solution to bipartition

Semidefinite programming

Solution

- Random hyperplane
- Contribution of vertices x_i, x_j is $\frac{w_{i,j}}{2}(1 - cos\theta_{i,j})$, where $\theta_{i,j}$ is the angle between the two vectors x_i, x_j
- Probability of separation: $\frac{\theta_{i,j}}{\pi}$

Approximation Factor

$$\alpha = \frac{2}{\pi} \min_{\theta} \frac{\theta}{1 - cos\theta} > 0.878 \tag{14}$$

Max Sat

Instance

A set of boolean clauses $C = \{c_1, \dots, c_m\}$ made of disjunctions over variables $X = \langle x_1, \dots, x_n \rangle$ A weight $w : C \mapsto \mathbb{Q}^+$ of each clause.

Feasible solution

A truth assignment Y to the variables in X

Objective function

$\sum_{c \in D} w(c)$, where D is the set of clauses of C that are made true by Y

Example

$c_1 = x_1 \vee x_3 \vee \neg x_5$

$c_2 = \neg x_1 \vee \neg x_2$

$c_3 = x_4$

Probabilistic Approach for Max Sat

Random assignment

- Each variable x_i is true is probability $1/2$
- $E[w(c)] = w(c) \cdot Pr[c \text{ is satisfied}]$
- Depends on $size(c) = k_c$
- $E[w(c)] = w(c)\left(1 - 2^{-k_c}\right) = \alpha_k w_c$, for $\alpha_k = \left(1 - 2^{-k_c}\right)$
- Since $\alpha_k \geq \frac{1}{2}$, then $\frac{1}{2} - approximation$ (expected)

Conditional expectation

Derandomize

1. $E[Y] = \sum_{c \in C} \alpha_k w_c$
2. $E[Y] = \frac{1}{2}(E[Y|x_1 = T] + E[Y|x_1 = F])$
3. Pick the best between $E[Y|x_1 = T]$ and $E[Y|x_1 = F]$

LP Approach for Max Sat

ILP for Max Sat

$$
\begin{aligned}
\max_{c \in C} w_c z_c & \qquad \text{subject to} \\
\sum_{i \in S_c^+} y_i + \sum_{i \in S_c^-} (1 - y_i) \geq z_c & \quad \forall c \in C \\
y_i \in \{0, 1\} & \quad \forall i \\
z_c \in \{0, 1\} & \quad \forall c \in C
\end{aligned}
\tag{15}
$$

S_c^+: boolean variables non-negated in c_i, S_c^-: boolean variables negated in c_i

LP Approach for Max Sat

ILP relaxation

$$
\max_{c \in C} w_c z_c \qquad \text{subject to}
$$
$$
\sum_{i \in S_c^+} y_i + \sum_{i \in S_c^-} (1 - y_i) \geq z_c \quad \forall c \in C
$$
$$
0 \leq y_i \leq 1 \quad \forall i
$$
$$
0 \leq z_c \leq 1 \quad \forall c \in C \tag{16}
$$

y^*, z^*: fractional optimum

Algorithm

$x_i \leftarrow T$ with probability y_i^*

Lemma

$E[W] \geq \beta_k w_c z_c^*$, with $\beta_k = 1 - \left(1 - \frac{1}{k}\right)^k$

Proof

- c_i satisfied if at least a variable is T
- $E[W_c] = 1 - \Pi_{i=1}^k (1 - y_i) \geq 1 - \left(\frac{\sum_{i=1}^k (1-y_i)}{k}\right)^k \geq 1 - \left(1 - \frac{z_c^*}{k}\right)^k$
- $1 - \left(1 - \frac{z_c^*}{k}\right)^k \geq \beta_k$ for $0 \leq z_c^* \leq 1$

LP Approach for Max Sat

Approximation

$\beta_k = 1 - \left(1 - \frac{1}{k}\right)^k \geq 1 - \frac{1}{e}$

Notice

β_k monotone increasing

Better approximation for Max Sat

Algorithm

Pick the better of the solutions of the two algorithms

Approximation Factor

- $E[W_c] = \alpha_k w_c + \beta_k w_c z_c^*$
- $z_c^* \leq 1$, hence $E[W_c] = \alpha_k w_c z_c^* + \beta_k w_c z_c^* = (\alpha_k + \beta_k) w_c z_c^*$
- Prove $\alpha_k + \beta_k \geq \frac{3}{4}$
- $\frac{3}{4}$-approximation

Max multicommodity flow

Instance

Undirected graph $G = \langle V, E \rangle$. Capacity c_e for each edge E. A set $\{(s_1, t_1), \ldots, (s_k, t_k)\}$ of source-sink pairs (commodity)

Feasible solution

A flow that respects the maximum capacity and satisfies flow conservation.

Objective function

$\sum_{(s_i,t_i)}$ flow from s_1 to t_i

Max multicommodity flow

Primal

$$\max \sum_{p \in P} f_p \qquad \text{subject to}$$
$$\sum_{p: e \in p} f_p \leq c_e \quad \forall e \in E \tag{17}$$
$$f_p \geq 0 \quad \forall \text{ paths } p \in P$$

Dual

$$\min \sum_{e \in E} c_e d_e \qquad \text{subject to}$$
$$\sum_{e \in p} d_e \geq 1 \quad \forall p \in P \tag{18}$$
$$d_e \geq 0 \quad \forall e \in E$$

d_e

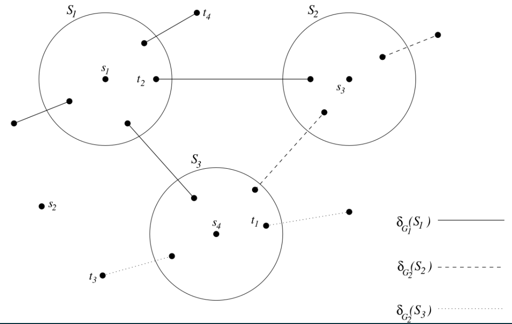
distance between vertices.

- Flow-cut duality
- Pick a multicut D

Growing regions

Goal

- No region contains a source-sink pair
- region centered on a source
- $c(R) \leq \epsilon wt(R)$, where $c(R)$ is the capacity of the cut



Lemma

Radius $\leq \frac{1}{2}$, the ball has no source-sink pair

Growing regions

Weight distribution

- $wt(s) = \frac{F}{k}$, with s source, F fractional optimum
- q_e: fraction of edge e in the region
- $q_e = \frac{r - dist(s,u)}{dist(s,v) - dist(s,u)}$ for each edge $e = (u, v)$ in the cut
- $wt(R) = wt(s) + \sum_{e \in X} c_e d_e q_e$, with X the set of edges with at least an endpoint in R
- Larger region R, easier $c(R) \leq \epsilon wt(R)$
- $\epsilon \leftarrow 2 \ln(k+1) \Rightarrow$ radius $\leq \frac{1}{2}$
- $\frac{dwt(s(r))}{dr} \geq \sum_e c_e d_e \frac{dq_e}{dr} \geq c(S(r))$

Smooth polynomial programming

Program

$$
\begin{aligned}
\max p(x_1, \dots, x_n) \qquad & \text{subject to} \\
\sum l_i \le p(x_1, \dots, x_n) \le g_i & \\
x_i \in \{0, 1\} \quad & \forall x_i
\end{aligned} \tag{19}
$$

Smoothness

For each degree-d polynomial, each coefficient of each degree i monomial is $\le cn^{d-i}$

Compute a (random) solution with

- Additive error ϵn^d
- degree-f constraints satisfied with additive error ϵn^f
- linear constraints satisfied with additive error $O(\epsilon\sqrt{n \log n})$
- time complexity $O\left(\left(dKn^d\right)^t\right)$, with $t = 4\frac{c^2 e^2 d^2}{\epsilon^2}$ and K the number of constraints

Max Cut

Instance

A weighted undirected graph $G = \langle V, E \rangle$, $w : E \mapsto \mathbb{Q}^+$

Feasible solution

a bipartition (V_1, V_2) of V

Objective function

$\sum_{v_1 \in V_1, v_2 \in V_2} w(v_1, v_2)$

Program

$$\max_{(i,j)\in E} w(i,j)\left(x_i(1-x_j)\right) + \left(x_j(1-x_i)\right) \tag{20}$$

Dense-k-subgraph

Instance
An undirected graph $G = \langle V, E \rangle$

Feasible solution
A subset S of k vertices of G

Objective function
$|E \cap S \times S|$

Denseness
Average degree δ

Random algorithm
Has $\alpha^2\delta^2 n^2/2$ edges.

Dense-k-subgraph

Instance

An undirected graph $G = \langle V, E \rangle$

Feasible solution

A subset S of k vertices of G

Objective function

$|E \cap S \times S|$

Program

$$\begin{aligned} \max_{(i,j)\in E} x_i x_j \qquad & \text{subject to} \\ \sum x_i = k & \\ x_i \in \{0,1\} \quad \forall x_i & \end{aligned} \tag{21}$$

Linear constraint

Move $O(\sqrt{n \log n})$ vertices in/out the set S, at most $O(n\sqrt{n \log n}) = o(n^2)$ edges affected

Attribution

- Vertex Cover figure: By Miym - Own work, CC BY-SA 3.0,
  `https://commons.wikimedia.org/w/index.php?curid=6017739`
- Clique figure: Public Domain,
  `https://commons.wikimedia.org/w/index.php?curid=1072101`
- Max Cut figure: By Miym - Own work, CC BY-SA 3.0,
  `https://commons.wikimedia.org/w/index.php?curid=6002348`
- Set Cover figure: Public Domain,
  `https://commons.wikimedia.org/w/index.php?curid=647030`