Approximation Algorithms

Maurice Queyranne

Lecture 4

August 6, 2009

Taking advantage of the structure of an LP relaxation

[Vazirani, Section 14.3, pp 136-137]

Weighted Vertex Cover: given a graph $G=(V,E)$ node weights $w_j > 0$ find a cover $C \subseteq V$ of all the edges by vertices with minimum total weight $w(C) = \sum_{j \in C} w_j$

IP formulation: $x_j = \begin{cases} 1 & \text{if vertex } j \text{ is selected} \\ 0 & \text{o/w} \end{cases}$

VCIP
$$\begin{cases} \min \sum_{j \in V} w_j x_j \\ \text{s.t.} \quad x_i + x_j \geq 1 \qquad \forall e = \{i,j\} \in E \\ \qquad x \geq 0, \text{ integer} \end{cases}$$

Theorem: Every extreme optimum solution to the LP relaxation of VCIP is half-integral (i.e., all its components are integer multiples of 1/2)

Proof: It suffices to prove that every extreme point of the polyhedron
$$P = \{x \in \mathbb{R}^n : x_i + x_j \geq 1 \ \forall \{i,j\} \in E, \ x \geq 0\}$$
is half-integer. Recall that a point x is an extreme point of a set P if x cannot be expressed as a convex combination of other points in P, i.e., there do not exist $x', x'' \in P \setminus \{x\}$ and real number $\lambda \in (0,1)$ such that $x = \lambda x' + (1-\lambda) x''$

Let x be an extreme point of P, so $0 \leq x \leq 1$
$$V_+ := \{j : 1/2 < x_j < 1\} \quad \text{and} \quad V_- := \{j : 0 < x_j < 1/2\}$$
If $V_+ \cup V_- \neq \emptyset$ then for $\varepsilon > 0$ define x' and x''

$$x'_j := \begin{cases} x_j + \varepsilon & \text{if } j \in V_+ \\ x_j - \varepsilon & \text{if } j \in V_- \\ x_j & \text{o/w} \end{cases} \qquad x''_j := \begin{cases} x_j - \varepsilon & \text{if } j \in V_+ \\ x_j + \varepsilon & \text{if } j \in V_- \\ x_j & \text{o/w} \end{cases}$$

so $x = \frac{1}{2}x' + \frac{1}{2}x''$. We have $x', x'' \geq 0$ for $\varepsilon > 0$ small enough

for any edge $\{i,j\} \in E$

if $x_i + x_j > 1$ then $\left.\begin{array}{l} x'_i + x'_j \geq 1 \\ \text{and } x''_i + x''_j \geq 1 \end{array}\right)$ for $\varepsilon > 0$ small enough

else $x_i + x_j = 1$ then if one of i or j is in V_+ or V_-

then the other must be in V_- or V_+ respectively, and

$x'_i + x'_j = x''_i + x''_j = x_i + x_j = 1$

Therefore $x' \in P$ and $x'' \in P$. Since $x = \frac{1}{2}x' + \frac{1}{2}x''$,

x cannot be an extreme point of P, a contradiction. QED

Then the Rounding Algorithm (method 1, or 2) constructs the solution x^R with

$$x^R_j = \begin{cases} 1 & \text{if } x_j \in \{1, 1/2\} \\ 0 & \text{o/w} \end{cases}$$

this gives a vertex cover, with weight $w(x^R) \leq 2\,w(x) \leq 2\,OPT$

MAX SAT: given a logical expression in n Boolean variables X_j $j = 1..n$ in conjunctive normal form, i.e., the conjunction ("and") of clauses, each of which is a disjunction ("or") of literals, each of which is either a variable X_j or its negation not X_j

- weights $w_c > 0$ for each clause c

Example: a clause $C_1 = X_2$ or (not X_3) or X_4

a logical expression: C_1 and C_2 and C_3 ... and C_m

$$\big(\underbrace{X_2 \text{ or } (\text{not } X_3) \text{ or } X_4}_{C_1}\big) \text{ and } \big(\underbrace{X_1 \text{ or } X_2 \text{ or } X_3}_{C_2}\big) \text{ and } \big(\underbrace{X_3 \text{ or } (\text{not } X_5)}_{C_3}\big)$$

MAX SAT is the problem of finding a true/false assignment to the variables to maximize the total weight of the true clauses

The size(C) of a clause C is its number of literals

"Large clauses" Randomized algorithm:

for each variable X_j assign it the value True with prob. 1/2
False — - 1/2

independently for all variables

the probability that a clause C is false is $\left(\frac{1}{2}\right)^{\text{size}(C)}$

$\Rightarrow$ Prob (C is true) $= 1 - \left(\frac{1}{2}\right)^{\text{size}(C)}$

Its expected total weight $E[W] = \sum_C w_c \left(1 - \left(\frac{1}{2}\right)^{\text{size}(C)}\right) \geq \frac{1}{2} \sum_c w_c$

$\geq \frac{1}{2}$ OPT

Derandomization: Turning the randomized algorithm into a deterministic algorithm without any loss in approximation ratio

Method of Conditional Expectations

$$E[W] = \text{Prob}\{X_j \text{ true}\}\, E[W \mid X_j \text{ true}] + \left(1 - \text{Prob}\{X_j \text{ true}\}\right) E[W \mid X_j \text{ false}]\big)$$

$$\leq \max\left(E[W \mid X_j \text{ true}],\, E[W \mid X_j \text{ false}]\right)$$

Since the problem obtained after fixing any subset of variables to any True/False values has the same structure, and we can compute the corresponding conditional expectation in polytime, the following deterministic algorithm finds, in polytime, a solution with weight at least as large as $E[W]$:

for $j = 1 \dots n$ do
- compute $E[W \mid x_j \text{ true}]$ and $E[W \mid x_j \text{ false}]$
- fix x_j to the value corresponding to the larger of these two conditional expectations

Indeed
$$E[W \mid x_j \text{ true}] = \sum_{C:\, x_j \text{ in } C} w_C + \sum_{\substack{C \text{ does} \\ \text{not contain} \\ x_j \text{ nor } (\text{not } x_j)}} w_C \left(1 - \left(\tfrac{1}{2}\right)^{\text{size}(C)}\right) + \sum_{C:\, (\text{not } x_j) \text{ in } C} w_C \left(1 - \left(\tfrac{1}{2}\right)^{\text{size}(C)-1}\right)$$

and similarly for $E[W \mid x_j \text{ false}]$

LP Rounding for MAX SAT:

IP formulation
$$y_j := \begin{cases} 1 & \text{if we choose } x_j = \text{true} \\ 0 & \text{———— } x_j = \text{false} \end{cases}$$

$$z_C := \begin{cases} 1 & \text{if clause } C \text{ is satisfied by this assignment} \\ 0 & \text{o/w} \end{cases}$$

$$(IP) \begin{cases} \max \sum_C w_C z_C \\ \text{s.t.} \quad z_C \le \sum_{j:\, x_j \text{ in } C} y_j + \sum_{\substack{j:\, (\text{not } x_j) \\ \text{in } C}} (1 - y_j) \quad \forall \text{clauses } C \\ 0 \le z \le 1 \quad z \text{ (integer)} \\ 0 \le y \le 1 \quad y \text{ integer} \end{cases}$$

Let (y^{LP}, z^{LP}) denote an optimum solution to the LP relaxation of (IP)

Randomized algorithm: set each x_j = True with prob. y_j^{LP}
False ——— $1-y_j^{LP}$

independently for all j

For integer k let $\beta_k = 1-\left(1-\frac{1}{k}\right)^k$

$E[W_c] = w_c \, \mathrm{Prob}\{\text{clause } C \text{ is true}\}$

<u>Lemma</u>: If size$(C) = k$ then $E[W_c] \geq \beta_k w_k z_k^{LP}$

proof: wlog let $C = x_1 \text{ or } x_2 \text{ or} \ldots \text{or } x_k$ (to simplify notation)

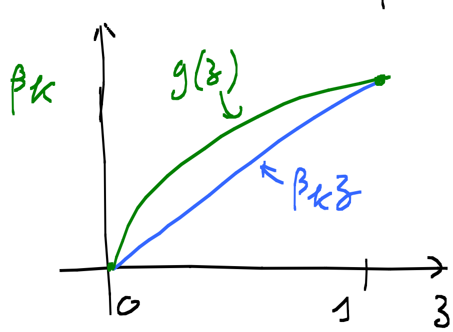
$$\mathrm{Prob}\{C \text{ is true}\} = 1 - \prod_{i=1}^{k}(1-y_i^{LP})$$

$$\geq 1 - \left(\frac{\sum_{i=1}^{k}(1-y_i^{LP})}{k}\right)^k$$ by the Geometric-Arithmetic Mean inequality

$$= 1 - \left(1 - \frac{\sum_{i=1}^{k} y_i^{LP}}{k}\right)^k$$

$$\geq 1 - \left(1 - \frac{z_c^{LP}}{k}\right)^k$$ since (y^{LP}, z^{LP}) feasible for the LP relaxation

$$\geq \beta_k z_c^{LP}$$ because the function g defined by $g(z) := 1-\left(1-\frac{z}{k}\right)^k$ is concave for $0 \leq z \leq 1$, with $g(0) = 0$ and $g(1) = \beta_k$, so $g(z) \geq \beta_k z$



If all clauses have size $\leq k$ ("small clauses") then

$$E[W] = \sum_c E[W_c] \geq \beta_k \sum w_c z_c^{LP} \geq \beta_k \, OPT$$

Remarks:

1) this algorithm can be derandomized

2) $\forall k \qquad \beta_k = 1 - \left(1 - \frac{1}{k}\right)^k > 1 - \frac{1}{e}$

this is an $\frac{e}{e-1}$ approximation for MAXSAT $\left(\frac{e}{e-1} \simeq 1.582\right)$

Combined algorithm (CA)

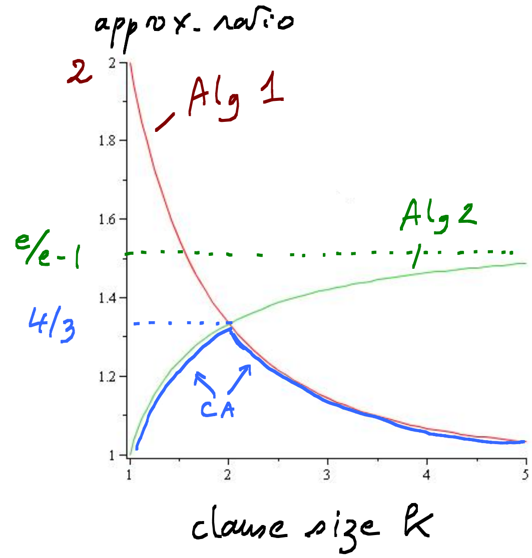
1. Run the "Large Clauses algorithm" (Randomized alg. with prob. ½)

   let X^L be the corresponding solution and W^L its weight

2. Run the "Small Clauses algorithm" (LP Rounding)

   let X^S be the corresponding solution and W^S its weight

3. Return the better of X^L and X^S



<u>Theorem</u>: This Combined Algorithm is a 4/3-approximation for MAXSAT

proof: Consider the **Randomized Combined Algorithm (RCA)**:

flip a fair coin - if Heads run Algorithm 1 (Large Clauses)

else — — 2 (Small Clauses)

<u>Lemma</u>: In RCA, for every clause C, $E[W_c] \geq \frac{3}{4} w_c z_c^{LP}$

proof: let $k = \text{size}(c)$

$$E[W_c \mid \text{Heads}] = \alpha_k w_c \geq \alpha_k w_c z_k^{LP} \quad \text{where } \alpha_k = 1 - 2^{-k}$$

$$E[W_c \mid \text{Tails}] \geq \beta_k w_c z_k^{LP} \quad \text{by previous lemma}$$

so $E[W_c] = \frac{1}{2} E[W_c | \text{Heads}] + \frac{1}{2} E[W_c | \text{Tails}] \geq \frac{\alpha_k + \beta_k}{2} w_c z_c^{LP}$

for $k = 1$ and 2 $\quad \frac{\alpha_k + \beta_k}{2} = 3/4$

$k \geq 3$ $\quad \frac{\alpha_k + \beta_k}{2} \geq \frac{1}{2}\left(\frac{7}{8} + \left(1 - \frac{1}{e}\right)\right) > \frac{3}{4}$ QED

Back to the proof of the Theorem:

$W^{CA} = \max\{W^L, W^S\} \geq \frac{1}{2} W^L + \frac{1}{2} W^S = E[W^{RCA}] \geq \frac{3}{4} OPT$ QED

Local Search

let $\mathcal{S}$ set of feasible solution to a combinatorial optimization problem

$$\max\{f(S) : S \in \mathcal{S}\}$$

a **neighborhood structure** is a collection $(\mathcal{N}(S))_{S \in \mathcal{S}}$ of subsets $\mathcal{N}(S) \subseteq \mathcal{S}$ such that:

(1) the resulting (hyper)graph is connected: for any $S, T \in \mathcal{S}$ there exists a sequence $S = S_0, S_1, \ldots, S_k = T$ of solutions such that $S_i \in \mathcal{N}(S_{i-1})$ for all $i = 1, \ldots, k$

(2) for every $S \in \mathcal{S}$ one can decide in polynomial time whether there exists $T \in \mathcal{N}(S)$ with $f(T) > f(S)$, and find one such **improving solution** if it exists

For example, we could have $|\mathcal{N}(S)| \leq$ polynomial (in the instance size) so we can do (2) by enumerating all solution in $\mathcal{N}(S)$ "small neighborhoods"

or a large neighborhood with an algorithm for (2)

("**very large neighborhood**" when $\mathcal{N}(S)$ is exponential)

Generic Local Search algorithm

- Initialize by finding a feasible solution $S_0 \in \mathcal{S}$, let $S := S_0$
- while S is not a local optimum do
  - find an improving solution $T \in \mathcal{N}(S)$ (i.e., with better obj. value)
  - $S := T$

If $\mathcal{S}$ is finite, this algorithm is finite

Two issues: (1) polynomial time?

(2) quality of the solution?

(1) Given $\varepsilon > 0$, S is an **ε-local optimum** (for a maximization problem)

$$\text{if} \quad f(T) \le (1+\varepsilon) f(S) \quad \forall T \in \mathcal{N}(S)$$

Modified Local Search algorithm (MLS)

- Initialize by finding a feasible solution $S_0 \in \mathcal{S}$. Let $S := S_0$
- while S is not a ε-local optimum do
  - find an $(1+\varepsilon)$-improving solution $T \in \mathcal{N}(S)$, i.e. such that

    $$f(T) > (1+\varepsilon) f(S)$$

  - $S := T$

If $f(S_0) > 0$ then after k iterations of MLS, $f(S) \ge (1+\varepsilon)^k f(S_0)$

If $\log(OPT/f(S_0))$ is polynomial in the instance size then, for any fixed $\varepsilon > 0$ the Modified Local Search algorithm stops with an ε-local optimum after at most $\log(OPT/f(S_0)) / \log(1+\varepsilon)$ iterations.

② How good is a local optimum?

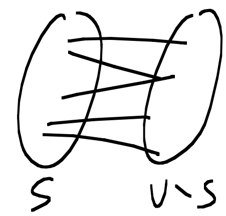
(WEIGHTED) MAX CUT problem

given graph $G=(V,E)$ with edge weights $w_{ij} > 0$

find a partition $(S, V\setminus S)$ of V to maximize the total weight of the edges in the resulting **cut**; i.e., edges with one endpoint in S and the other in $V\setminus S$

Denote a solution by S

and let $f(S)$ = total weight of the cut



Neighborhood structure: move one vertex from one side of the cut to the other side

$$\mathcal{N}(S) = \{T \subseteq V : |S\setminus T| = 1 \text{ or } |T\setminus S| = 1\}$$

$$|\mathcal{N}(S)| = |V| = n$$

so an improving solution can be found in $O(|E|)$ time (actually in $O(n)$ time)

① Is this local search algorithm polytime?

- if all $w_{ij} = 1$ then $OPT \le \frac{n(n-1)}{2}$, $f(S)$ is integer-valued.

  take any $S_0 = \{v\}$ so $f(S_0) \ge 1$, every improvement is by at least 1 unit

  so at most $O(n^2)$ iterations, and $O(n^3)$ altogether

- for general weights $OPT \le \sum_{ij\in E} w_{ij}$

  take $S^0 = \{v\}$ with $v \in \operatorname{argmax}\{f(\{u\}) : u \in V\}$

  $$\text{so } OPT \le \sum_{ij\in E} w_{ij} = \frac{1}{2}\sum_{u\in V} f(\{u\}) \le \frac{n}{2}\max_{u\in V} f(\{u\}) = \frac{n}{2} f(S_0)$$

  and MLS finds an ε-local optimum after a polynomial (actually polylog) number of iterations, for any fixed $\varepsilon>0$

② How good is a local optimum?

Thm: For any local maximum S for this neighborhood structure for MAXCUT

$$f(S) \geq \frac{1}{2} OPT$$ (any local optimum is a 2-approximation)

proof: let S be a local maximum

$\forall v \in S$ it is better to keep v in S than to move it to $V \setminus S$

$$\sum_{\substack{uv \in E \\ u \in V \setminus S}} w_{uv} \geq \sum_{\substack{uv \in E \\ u \in S}} w_{uv}$$

$$\Rightarrow 2 \sum_{u \in V \setminus S} w_{uv} \geq \sum_{u \in S} w_{uv} + \sum_{u \in V \setminus S} w_{uv} = \sum_{\substack{uv \in E \\ u \in V}} w_{uv}$$

$$\Rightarrow \sum_{\substack{uv \in E \\ u \in V \setminus S}} w_{uv} \geq \frac{1}{2} \sum_{\substack{uv \in E \\ u \in V}} w_{uv} = \frac{1}{2} f(\{v\})$$

Similarly for $v' \in V \setminus S$

$$\sum_{\substack{uv' \in E \\ u \in S}} w_{uv'} \geq \sum_{\substack{uv' \in E \\ u \in V \setminus S}} w_{uv'}$$

$$\Rightarrow \sum_{\substack{uv' \in E \\ u \in S}} w_{uv'} \geq \frac{1}{2} \sum_{\substack{uv' \in E \\ u \in V}} w_{uv'} = \frac{1}{2} f(\{v'\})$$

Adding these two inequalities for all $v \in V$

$$2 f(S) = \sum_{v \in S} \sum_{u \in V \setminus S} w_{uv} + \sum_{v' \in V \setminus S} \sum_{u \in S} w_{uv'}$$

$$\geq \frac{1}{2} \sum_{v \in V} f(\{v\}) = \sum_{uv \in E} w_{uv} \geq OPT$$

QED

Semi-Definite Programming (SDP) relaxations

SDP is a generalization of LP where the decision variables are (symmetric) matrices, and the nonnegativity constraints are replaced by the constraint that the variables form a semidefinite matrix

an $n \times n$ matrix A is symmetric if $a_{ij} = a_{ji}$ $\forall i,j$

let $\mathcal{M}_n$ be the set of all symmetric $n \times n$ matrices

$n \times n$ matrix A is positive semi-definite (psd) if

$x^T A x \geq 0$ for all $x \in \mathbb{R}^n$, denoted $A \succeq 0$

Theorem: A symmetric $n \times n$ matrix A is psd iff all its eigenvalues are nonnegative reals

Given two $n \times n$ matrices A, B their Frobenius product

$$A \bullet B := \sum_i \sum_j a_{ij} b_{ij}$$

A SDP problem: given $n \times n$ matrices $C, A_1 \ldots A_m$ and RHS n-vectors $b_1, \ldots, b_m \in \mathbb{R}^n$

find an $n \times n$ matrix X to

$$\begin{cases} \text{maximize} & C \bullet X \\ \text{s.t.} & A_i \bullet X = b_i \quad i = 1..m \\ & X \succeq 0, \; X \in \mathcal{M}_n \end{cases}$$

There exist polytime algorithms to solve SDP problems approximately, within any factor $(1-\varepsilon)$ of the optimum

Lieven Vandenberghe, Stephen Boyd, "Semidefinite Programming", SIAM Review 38, March 1996, 49-95.

Monique Laurent, Franz Rendl, "Semidefinite Programming and Integer Programming", Report PNA-R0210, CWI, Amsterdam, April 2002. (Available at *optimization-online*)