ALMA
ALGORITHMS

Fall 2016

Ioannis Milis

Approximation Algorithms
LP Rounding

Intro

Optimization problems: Find a solution that is

i. Feasible: satisfies certain constraints, and

ii. Best possible (optimal), with respect to some well-defined criterion, among all feasible solutions

Linear programming: A broad class of optimization problems, where both the constraints and the optimization criterion are linear functions.

In other words:

Assign values to a set of variables $x_1, x_2, \dots, x_n$, so as:

i. satisfy a set of linear equations/ inequalities (constraints) on them

ii. Maximize/minimize a given linear objective function of them.

Almost all the problems we have seen so far

Intro

Example: A company has two products 1 and 2 with profits \$1 and \$6.

The daily demand for their products are 200 pieces of product 1 and 300 pieces of product 2. They can also produce 400 pieces of both products per day.

How much of each should they produce to maximize their profit?

Variables: x_1 and x_2 pieces of products

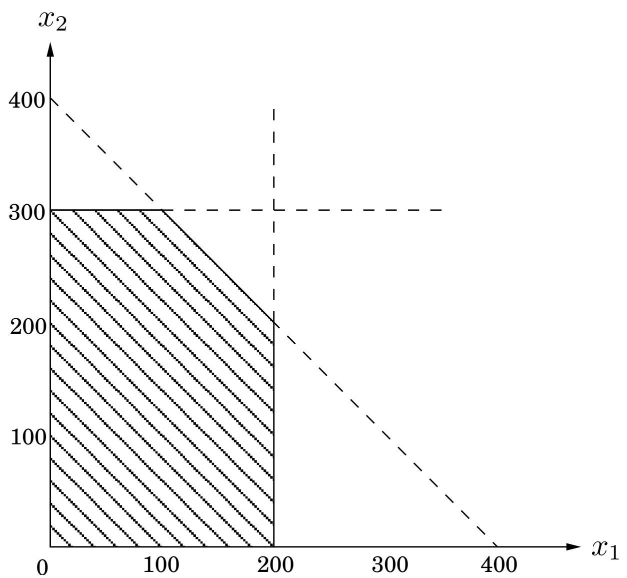
Linear program:

$$\begin{aligned}
&\textbf{Objective function} \quad && \max x_1 + 6x_2 \\
&\textbf{Constraints} && x_1 \leq 200 \\
& && x_2 \leq 300 \\
& && x_1 + x_2 \leq 400 \\
& && x_1, x_2 \geq 0
\end{aligned}$$

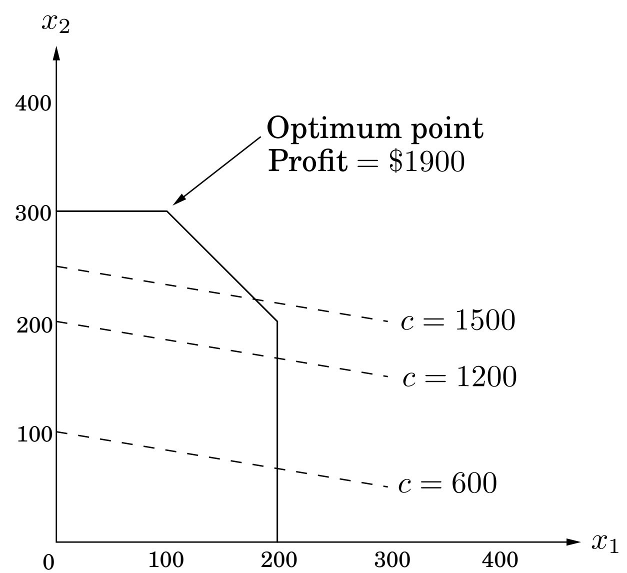
Intro

Geometry

$$\begin{array}{ll} \textbf{Objective function} & \max x_1 + 6x_2 \\ \textbf{Constraints} & x_1 \leq 200 \\ & x_2 \leq 300 \\ & x_1 + x_2 \leq 400 \\ & x_1, x_2 \geq 0 \end{array}$$



Feasible region

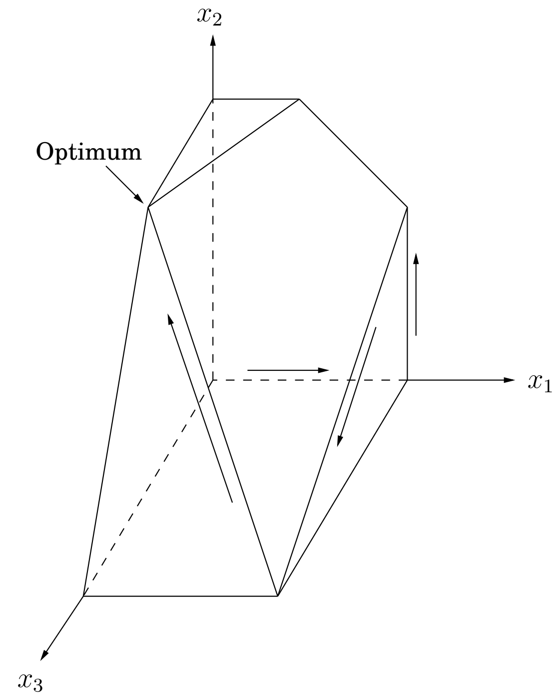


Profits

Intro

More products

$$\max \quad x_1 + 6x_2 + 13x_3$$
$$x_1 \leq 200$$
$$x_2 \leq 300$$
$$x_1 + x_2 + x_3 \leq 400$$
$$x_2 + 3x_3 \leq 600$$
$$x_1, x_2, x_3 \geq 0$$



Each constraint corresponds
to a face of the polyhedron

Intro

- The optimum is achieved at a vertex of the feasible region
- The only exceptions are cases in which there is no optimum

1. The LP is **infeasible**

   too tight constraints; impossible to satisfy all of them

   e.g. $x_1 \le 1$, $x_1 \ge 2$

2. The LP is **unbounded;**

   too loose constraints; the feasible region is unbounded

   e.g. arbitrarily high objective values

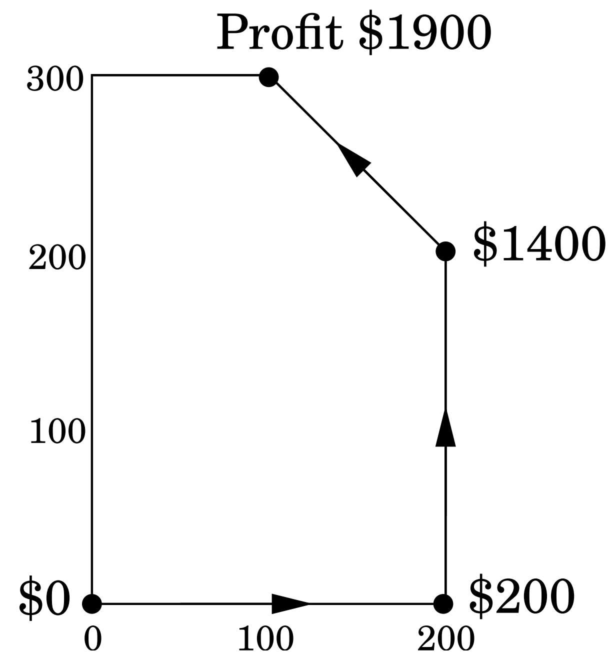
   $\max x_1 + x_2$

   $x_1, x_2 \ge 0$

Intro

LP's can be solved by the **Simplex Method** [G. Dantzig, 1947]

- Starts at a vertex, say (0, 0)
- Repeatedly looks for an adjacent vertex
  of better objective value
- Halts upon reaching a vertex that has
  no better neighbor and declares it as optimal



Does *hill-climbing* on the vertices of the polygon, from neighbor to neighbor so as to steadily increase profit along the way (Local Search)

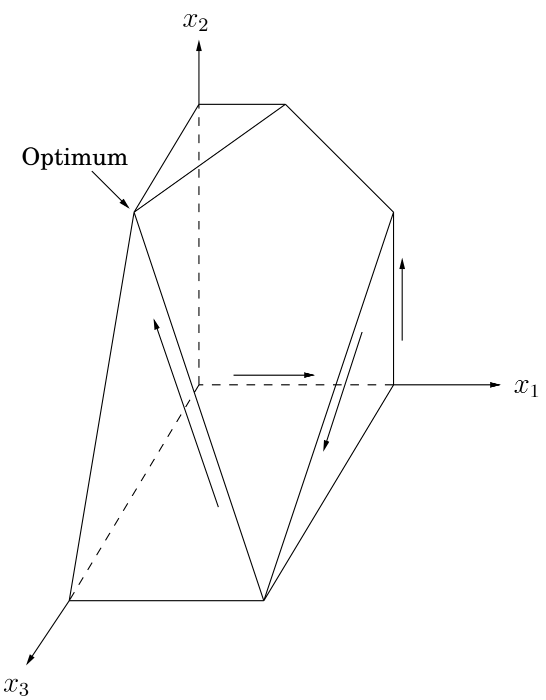
Why does its *local* test imply *global* optimality?

By simple geometry—think of the profit line passing through this vertex. Since all the vertex's neighbors lie below the line, the rest of the feasible polygon must also lie below this line.

Intro

Simplex Method [G. Dantzig, 1947].

More products



It would move from vertex to vertex,
along edges of the polyhedron,
increasing profit steadily.

Again by basic geometry,
if all the vertex's neighbors
lie on one side of the profit-plane,
then so must the entire polyhedron

A possible trajectory

Vertices:	(0,0,0) →	(200,0,0) →	(200,200,0) →	(200,0,200) →	(0,300,100)
Profits	\$0	\$200	\$1400	\$2800	\$3100

Intro

Variants of LP's

- The objective can be Maximization or Minimization
- The constraints can be equations and/or inequalities.
- The variables can be restricted to be nonnegative,
  but they can also be unrestricted in sign

All LP's variants can be reduced to one another

- Maximization $\rightarrow$ Minimization (or vice versa):
  Multiply the coefficients of the objective function by −1
- Inequality constraint $\rightarrow$ equality constraint

$$\sum_{i=1}^{n} a_i x_i \leq b$$

$$\begin{aligned} \sum_{i=1}^{n} a_i x_i + s &= b \\ s &\geq 0. \end{aligned}$$

s is a new variable called slack variable for the inequality

Intro

All LP's variants *can be reduced to one another*

- Equality constraint → Inequalities

  Rewrite an equality constraint as an equivalent pair of inequality constraints

  $ax = b$ $\quad\quad$ $ax \le b$

  $\quad\quad\quad\quad\quad\;$ $ax \ge b$

- Unrestricted in sign variable x → nonnegative variable(s)

  Introduce two nonnegative variables, $x^+, x^- \ge 0$

  Replace x, wherever it appears

  by $x^+ - x^-$

  Thus, x can take any value by appropriately adjusting x^+ and x^-.

  (any feasible solution to the original LP involving x can be mapped to a feasible solution of the new LP involving x^+ , x^-, and vice versa)

Intro

Any LP (maximization or minimization, with both inequalities and equations, and with both nonnegative and unrestricted variables) can be transformed to an equivalent standard form LP in which the variables are all nonnegative, the constraints are all equations, and the objective function is to be minimized

$$\begin{aligned} \max\ x_1 + 6x_2 & \\ x_1 &\le 200 \\ x_2 &\le 300 \\ x_1 + x_2 &\le 400 \\ x_1, x_2 &\ge 0 \end{aligned} \quad \Longrightarrow \quad \begin{aligned} \min\ -x_1 - 6x_2 & \\ x_1 + s_1 &= 200 \\ x_2 + s_2 &= 300 \\ x_1 + x_2 + s_3 &= 400 \\ x_1, x_2, s_1, s_2, s_3 &\ge 0 \end{aligned}$$

The original here is also in a useful form:

maximize an objective subject to certain inequalities

Matrix-vector notation of LP's

$$\max x_1 + 6x_2$$
$$s.t. \quad x_1 \le 200$$
$$x_2 \le 300$$
$$x_1 + x_2 \le 400$$
$$x_1, x_2 \ge 0$$

$$\max \; \mathbf{c}^T \mathbf{x}$$
$$\mathbf{A}\mathbf{x} \le \mathbf{b}$$
$$\mathbf{x} \ge 0.$$

$$\mathbf{c} = \begin{pmatrix} 1 \\ 6 \end{pmatrix} \text{ and } \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\begin{array}{rcl} x_1 & \le & 200 \\ x_2 & \le & 300 \\ x_1 + x_2 & \le & 400 \end{array} \Longrightarrow \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{\mathbf{x}} \underbrace{\le}_{\le} \underbrace{\begin{pmatrix} 200 \\ 300 \\ 400 \end{pmatrix}}_{\mathbf{b}} .$$

n: # of variables
m: # of constraints

LP is polynomial

Method	Typical cost	Worst case cost
Simplex	$O(n^2 m)$	Very bad - **Not polynomial**
Ellipsoid	$O(n^8)$	$O(n^8)$

Everything you need to know about solving linear programs

Integer (Linear) Programming (IP)

$$
\begin{aligned}
\max\ & x_1 + 6x_2 \\
s.t.\ & x_1 \le 200 \\
& x_2 \le 300 \\
& x_1 + x_2 \le 400 \\
& x_1, x_2 \in N
\end{aligned}
$$

$$
\begin{aligned}
\max\ & \mathbf{c}^T\mathbf{x} \\
& \mathbf{A}\mathbf{x} \le \mathbf{b} \\
& \mathbf{x} \ge 0.
\end{aligned}
$$

$$\mathbf{c} = \begin{pmatrix} 1 \\ 6 \end{pmatrix} \text{ and } \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$
\begin{array}{rcl}
x_1 & \le & 200 \\
x_2 & \le & 300 \\
x_1 + x_2 & \le & 400
\end{array}
\Longrightarrow
\underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}}_{\mathbf{A}}
\underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{\mathbf{x}}
\underbrace{\le}_{\le}
\underbrace{\begin{pmatrix} 200 \\ 300 \\ 400 \end{pmatrix}}_{\mathbf{b}}.
$$

The variables are restricted to be integers

IP is NP-complete; via a simple reduction from 3-SAT

All the NP-complete problems we have seen can be written as IP's, so, they all reduce to IP

Integer Programming Formulations

To formulate a problem as an **integer** program (IP), in general, we assign a binary variable x_i to items to be included in a solution

$$x_i = \begin{cases} 1, & \text{if item } i \text{ is in a solution} \\ 0, & \text{otherwise} \end{cases}$$

<u>0-1 KNAPSACK</u>

$$\max \quad \sum_{s \in S} v(s) x_s$$

$$s.t. \quad \sum_{s \in S} w(s) \cdot x_s \leq W$$

$$x_s \in \{0,1\}, \forall s \in S$$

Integer Programming Formulations

To formulate a problem as an **integer** program (IP), in general, we assign a binary variable x_i to items to be included in a solution

$$x_i = \begin{cases} 1, & \text{if item } i \text{ is in a solution} \\ 0, & \text{otherwise} \end{cases}$$

CLIQUE, $G=(V,E)$

$$\max \sum_{u \in V} x_u$$
$$s.t. \quad x_u + x_v \leq 1, \forall (u,v) \notin E$$
$$x_u \in \{0,1\}, \forall u \in V$$

IndSet, $G=(V,E)$

$$\max \sum_{u \in V} x_u$$
$$s.t. \quad x_u + x_v \leq 1, \forall (u,v) \in E$$
$$x_u \in \{0,1\}, \forall u \in V$$

Integer Programming Formulations

To formulate a problem as an **integer** program (IP), in general, we assign a binary variable x_i to items to be included in a solution

$$x_i = \begin{cases} 1, & \text{if item } i \text{ is in a solution} \\ 0, & \text{otherwise} \end{cases}$$

(W)VC, $G = (V, E)$

$$\begin{aligned} \min \quad & \sum_{u \in V} w(u) \cdot x_u \\ s.t. \quad & x_u + x_v \geq 1, \forall (u,v) \in E \\ & x_u \in \{0,1\}, \forall u \in V \end{aligned}$$

(W)SC

$$\begin{aligned} \min \quad & \sum_{S \in F} w(S) x_S \\ s.t. \quad & \sum_{S \in F : u \in S} x_S \geq 1, \forall u \in U \\ & x_S \in \{0,1\}, \forall s \in S \end{aligned}$$

Integer Programming Formulations

To formulate a problem as an **integer** program (IP), in general,
we assign a binary variable x_i to items to be included in a solution

$$x_i = \begin{cases} 1, & \text{if item } i \text{ is in a solution} \\ 0, & \text{otherwise} \end{cases}$$

KNAPSACK

$$\begin{aligned} \max \quad & \sum_{s \in S} v(s) x_s \\ s.t. \quad & \sum_{s \in S} w(s) \cdot x_s \le W \\ & x_s \in \{0,1\}, \forall s \in S \end{aligned}$$

CLIQUE, $G=(V,E)$

$$\begin{aligned} \max \quad & \sum_{u \in V} x_u \\ s.t. \quad & x_u + x_v \le 1, \forall (u,v) \notin E \\ & x_u \in \{0,1\}, \forall u \in V \end{aligned}$$

IndSet, $G=(V,E)$

$$\begin{aligned} \max \quad & \sum_{u \in V} x_u \\ s.t. \quad & x_u + x_v \le 1, \forall (u,v) \in E \\ & x_u \in \{0,1\}, \forall u \in V \end{aligned}$$

(W)VC, $G=(V,E)$

$$\begin{aligned} \min \quad & \sum_{u \in V} w(u) \cdot x_u \\ s.t. \quad & x_u + x_v \ge 1, \forall (u,v) \in E \\ & x_u \in \{0,1\}, \forall u \in V \end{aligned}$$

(W)SC

$$\begin{aligned} \min \quad & \sum_{S \in F} w(S) x_S \\ s.t. \quad & \sum_{S \in F : u \in S} x_S \ge 1, \forall u \in U \\ & x_S \in \{0,1\}, \forall s \in S \end{aligned}$$

LP based approximation algorithms

Linear Programming Relaxations

Relax the integer constraint

KNAPSACK

$$\max \sum_{s\in S} v(s)x_s$$
$$s.t. \quad \sum_{s\in S} w(s)\cdot x_s \le W$$
$$x_s \in [0,1], \forall s \in S$$

CLIQUE, $G=(V,E)$

$$\max \sum_{u\in V} x_u$$
$$s.t. \quad x_u + x_v \le 1, \forall (u,v) \notin E$$
$$x_u \in [0,1], \forall u \in V$$

IndSet, $G=(V,E)$

$$\max \sum_{u\in V} x_u$$
$$s.t. \quad x_u + x_v \le 1, \forall (u,v) \in E$$
$$x_u \in [0,1], \forall u \in V$$

(W)VC, $G=(V,E)$

$$\min \sum_{u\in V} w(u)\cdot x_u$$
$$s.t. \quad x_u + x_v \ge 1, \forall (u,v) \in E$$
$$x_u \in [0,1], \forall u \in V$$

(W)SC

$$\min \sum_{S\in F} w(S)x_S$$
$$s.t. \quad \sum_{S\in F: u\in S} x_S \ge 1, \forall u \in U$$
$$x_S \in [0,1], \forall s \in S$$

Rounding and Integrality gap

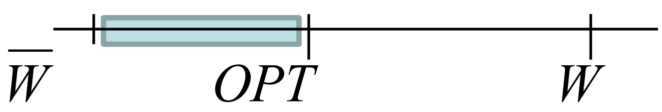
Rounding

Solve the LP-relaxation in $O(poly|I|)$ time

$\rightarrow$ fractional solution $\overline{x}$ of cost $\overline{W}$

Round $\overline{x}$ to an integral solution to the IP problem of cost W

But,



$$\overline{W} \le OPT \le W \quad \text{(min problems)}$$

We bound the ratio $\frac{W}{\overline{W}} \le \rho$, that is, $\frac{W}{OPT} \le \frac{W}{\overline{W}} \le \rho$

Let $\gamma = \frac{OPT}{\overline{W}}$, then $\gamma = \frac{OPT}{\overline{W}} \le \frac{W}{\overline{W}} \le \rho$

γ is called integrality gap and always $\gamma \le \rho$

Deterministic rounding for WVC

(W)VC, $G=(V,E)$

$$\begin{array}{lll} \min & \sum_{u\in V} w(u)\cdot x_u & \\ s.t. & x_u + x_v \geq 1, \forall (u,v)\in E \\ & x_u \in [0,1], \forall u \in V \end{array}$$

Rounding

Solve the LP-relaxation in $O(poly|I|)$ time

$\rightarrow$ fractional solution $\bar{x}_u$ of cost $\overline{W}$

If $\bar{x}_u \geq \frac{1}{2}$ then $x_u = 1$ else $x_u = 0$ (pick all vertices with $\bar{x}_u \geq \frac{1}{2}$)

Algorithm Rounding achieves an approximation ratio of 2 for the WVC problem

Deterministic rounding for WVC

Algorithm Rounding achieves an approximation ratio of 2 for WVC

(W)VC, $G = (V, E)$

$$\min \quad \sum_{u \in V} w(u) \cdot x_u$$
$$s.t. \quad x_u + x_v \geq 1, \forall (u,v) \in E$$
$$x_u \in [0,1], \forall u \in V$$

Proof:
Let C be the collection of sets picked

i) C is a valid VC

Assume that there is an edge $(u, v) \in E$ s.t. $u, v \notin C$

That is, $\bar{x}_u < 1/2$ and $\bar{x}_v < 1/2$

Hence, $\bar{x}_u + \bar{x}_v < 1$, a contradiction,

as this violates the LP constraint for edge (u, v)

Hence, either $u \in C$ or $u \in C$ and C is a valid VC

Deterministic rounding for WVC

Algorithm Rounding achieves a 2-approximation ratio for WVC

Proof:
Let C be the collection of sets picked

(W)VC, $G=(V,E)$

$$\min \sum_{u \in V} w(u) \cdot x_u$$
$$s.t. \quad x_u + x_v \geq 1, \forall (u,v) \in E$$
$$x_u \in [0,1], \forall u \in V$$

ii) $\frac{W}{OPT} \leq 2$

Recall that $\bar{x}_u \geq \frac{1}{2}$ for each $u \in C$

$$W = \sum_{u \in C} w(u) \overset{\bar{x}_u \geq \frac{1}{2}}{\leq} \sum_{u \in C} w(u) \cdot \bar{x}_u \cdot 2 \leq 2 \sum_{u \in C} w(u) \cdot \bar{x}_u$$

$$\leq 2 \sum_{u \in V} w(u) \cdot \bar{x}_u = 2 \cdot \bar{W} \leq 2 \cdot OPT \qquad (\bar{W} \leq OPT \leq W)$$

Deterministic rounding for WVC

Integrality gap

K_n=(V, ExE), |V|=n,
w(u)=1, for each u $\in$ V

- OPT= n-1 (all vertices but one)
- $\bar{x}_u = \frac{1}{2}, \forall u \in V \text{(due to symmetry)} \Rightarrow \overline{W} = \frac{n}{2}$

$$\gamma = \frac{OPT}{\overline{W}} = \frac{n-1}{n/2} \to 2$$

$$\gamma = 2 = \frac{OPT}{\overline{W}} \leq \frac{W}{\overline{W}} \leq \rho$$

(W)VC, $G = (V, E)$

$$\begin{array}{ll} \min & \sum_{u \in V} w(u) \cdot x_u \\ s.t. & x_u + x_v \geq 1, \forall (u,v) \in E \\ & x_u \in [0,1], \forall u \in V \end{array}$$

Deterministic rounding for WSC

(W)SC

$$\min \sum_{S \in F} w(S) x_S$$
$$s.t. \sum_{S \in F : u \in S} x_S \geq 1, \forall u \in U$$
$$x_S \in [0,1], \forall s \in S$$

Rounding

Solve the LP-relaxation in $O(poly|I|)$ time
$\rightarrow$ fractional solution $\overline{x}_S$ of cost $\overline{W}$

If $\overline{x}_S \geq \frac{1}{f}$ then $x_S = 1$ else $x_S = 0$ (pick all sets with $\overline{x}_S \geq \frac{1}{f}$)

Algorithm Rounding achieves an approximation ratio of f for the WSC problem

Deterministic rounding for WSC

Algorithm Rounding achieves an approximation ratio of f for WSC

Proof:
Let C be the collection of sets picked

i) C is a valid SC

Assume that there is $u \in U$ s.t. $u \notin C$

For each set $S_i \in F$, s.t $u \in S_i$, we have $\bar{x}_S < 1/f$

That is, $\sum_{S:u \in S} \bar{x}_S < \frac{1}{f} |\{S : u \in S\}| = \frac{1}{f} f_u \leq \frac{1}{f} f = 1$

A contradiction, as this violates the LP constraint for element u

Hence, $u \in C$ and C is a valid SC

(W)SC

$$\min \sum_{S \in F} w(S) x_S$$

$$s.t. \sum_{S \in F : u \in S} x_S \geq 1, \forall u \in U$$

$$x_S \in [0,1], \forall s \in S$$

Deterministic rounding for WSC

Algorithm Rounding achieves an f-approximation ratio for WSC

Proof:

Let C be the collection of sets picked

ii) $\frac{W}{OPT} \leq f$

(W)SC

$$\min \sum_{S \in F} w(S) x_S$$

$$s.t. \quad \sum_{S \in F : u \in S} x_S \geq 1, \forall u \in U$$

$$x_S \in [0,1], \forall s \in S$$

Recall that $\bar{x}_S \geq \frac{1}{f}$ for each $S \in C$

$$W = \sum_{S \in C} w(S) \overset{\bar{x}_S \geq \frac{1}{f}}{\leq} \sum_{S \in C} w(S) \cdot \bar{x}_S \cdot f \leq f \sum_{S \in C} w(S) \cdot \bar{x}_S$$

$$\leq f \sum_{S \in F} w(S) \cdot \bar{x}_S = f \cdot \bar{W} \leq f \cdot OPT \qquad (\bar{W} \leq OPT \leq W)$$

Randomized rounding for WSC

Randomized Rounding

Solve the LP-relaxation in O(poly|I|) time

→ Fractional solution x^* of cost Z^*_{LP}

For each subset S_j set $x_j = 1$ with probability x^*_j, independently

X_j : random variable that is 1 if subset S_j is taken and 0 otherwise

$\Pr[X_j = 1] = x^*_j$

Then the expected value of the solution is:

$$E\left[\sum_{j=1}^{m} w_j X_j\right] = \sum_{j=1}^{m} w_j \Pr[X_j = 1] = \sum_{j=1}^{m} w_j x_j^* = Z_{LP}^*,$$

Randomized rounding for WSC

Is such a solution a set cover ?

What is the probability that an element e_i is not covered?

$$\begin{aligned} \Pr[e_i \text{ not covered}] &= \prod_{j:e_i \in S_j} (1 - x_j^*) \\ &\le \prod_{j:e_i \in S_j} e^{-x_j^*} \\ &= e^{-\sum_{j:e_i \in S_j} x_j^*} \\ &\le e^{-1}, \end{aligned}$$

$$1 - x \le e^{-x}$$

$$\sum_{j:e_i \in S_j} x_j^* \ge 1$$

constant and too high …

Randomized rounding for WSC

Produce a cover whp

Repeat c ln n times: For each subset S_j set $x_j = 1$ with probability x^*_j

Return the union of the sets taken

Now,

$$\begin{aligned}
\Pr[e_i \text{ not covered}] &= \prod_{j:e_i \in S_j} (1 - x_j^*)^{c \ln n} \\
&\leq \prod_{j:e_i \in S_j} e^{-x_j^*(c \ln n)} \\
&= e^{-(c \ln n) \sum_{j:e_i \in S_j} x_j^*} \\
&\leq \frac{1}{n^c},
\end{aligned}$$

and for $c \geq 2$ we get a cover whp as

$$\Pr[\text{there exists an uncovered element}] \leq \sum_{i=1}^{n} \Pr[e_i \text{ not covered}] \leq \frac{1}{n^{c-1}}.$$

Randomized rounding for WSC

Bound the cost of the solution

As we repeat c ln n we have $\Pr[X_j=1] \le (c \ln n)\, x^*_j$, and

$$E\left[\sum_{j=1}^{m} w_j X_j\right] = \sum_{j=1}^{m} w_j \Pr[X_j = 1] \le \sum_{j=1}^{m} w_j (c \ln n) x_j^* = (c \ln n) \sum_{j=1}^{m} w_j x_j^* = (c \ln n) Z_{LP}^*$$

But we are interested in the cost of the solution

given that a cover is produced whp (fact F), that is

$$E\left[\sum_{j=1}^{m} w_j X_j\right] = E\left[\sum_{j=1}^{m} w_j X_j \,\middle|\, F\right] \Pr[F] + E\left[\sum_{j=1}^{m} w_j X_j \,\middle|\, \bar{F}\right] \Pr[\bar{F}] \qquad (*)$$

Randomized rounding for WSC

Bound the cost of the solution

From (*) we get

$$E\left[\sum_{j=1}^{m} w_j X_j \,\middle|\, F\right] = \frac{1}{\Pr[F]}\left(E\left[\sum_{j=1}^{m} w_j X_j\right] - E\left[\sum_{j=1}^{m} w_j X_j \,\middle|\, \bar{F}\right]\Pr[\bar{F}]\right)$$

$$\leq \frac{1}{\Pr[F]} \cdot E\left[\sum_{j=1}^{m} w_j X_j\right]$$

$$\leq \frac{(c \ln n) Z^*_{LP}}{1 - \frac{1}{n^{c-1}}}$$

$$\leq 2c(\ln n) Z^*_{LP}$$

for $n \geq 2$ and $c \geq 2$.

$$E\left[\sum_{j=1}^{m} w_j X_j \,\middle|\, \bar{F}\right] \geq 0$$

$$\Pr[F] \geq 1 - \frac{1}{n^{c-1}};$$

That is a randomized O(log n)-approximation algorithm !

Approximating MAX SAT

MAX SAT
I: A CNF formula φ
Q: Find an assignment satisfying the maximum number of clauses

$$\begin{cases} 1-\frac{1}{2^k} \geq \frac{1}{2} \\ \frac{\sqrt{5}-1}{2} = 0{,}618 \\ \frac{e-1}{e} = 0{,}632 \\ \frac{3}{4} = 0.75 \end{cases}$$

Randomized algorithms for MAX SAT

Set each variable TRUE with probability p=…

X=# of satisfied clauses

X_j : random variable that is 1 if clause C_j is satisfied and 0 otherwise

$E[X_j]$ = Pr[Clause C_j satisfied]

$$E[X] = E\left[\sum_{j=1}^{m} X_j\right] = \sum_{j=1}^{m} E[Xj] = \sum_{j=1}^{m} \Pr[\text{Clause } C_j \ \text{ satisfied}]$$

Randomized algorithms R1 and R2

Algorithm R1: Set each variable TRUE with probability p=½

k:# of literals in clause C_j

$$E[X_i] = \Pr[\text{Clause } C_j \text{ satisfied}] = 1 - \frac{1}{2^k} = a_k \geq \frac{1}{2} \quad (\text{for } k = 1)$$

$$E[X] = \sum_{j=1}^{m} \Pr[\text{Clause } C_j \text{ satisfied}] \geq \frac{1}{2} m \geq \frac{1}{2} OPT$$

Algorithm R2: Set each variable TRUE with probability p ≥ ½

$$E[X_i] = \Pr[\text{Clause } C_j \text{ satisfied}] = \frac{\sqrt{5}-1}{2} = 0{,}618$$

$$E[X] = \sum_{j=1}^{m} \Pr[\text{Clause } C_j \text{ satisfied}] \geq 0{,}618m \geq 0{,}618 OPT$$

Algorithm LP

for each variable x_i : $y_i = \begin{cases} 0, & FALSE \\ 1, & TRUE \end{cases}$

for each clause C_j : $z_j = \begin{cases} 0, & FALSE \\ 1, & TRUE \end{cases}$

IP binary variables

for a clause c : $\begin{cases} P_c & \text{positive variables} \\ N_c & \text{negative variables} \end{cases}$

$$C_j = \bigvee_{i \in P_j} x_i \vee \bigvee_{i \in N_j} \bar{x}_i$$

for $z_c = 1$ $\begin{cases} \text{at least one variable in } P_c \text{ is } 1 \\ \text{OR at least one variable in } N_c \text{ is } 0 \end{cases}$

Algorithm LP

(IP) $$\max \sum_{j=1}^{m} z_j$$

$$\sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j, \forall C_j$$

$$z_j \in \{0,1\}, \quad \mathrm{j} = 1,2,\ldots,\mathrm{m} \ \ (\forall \text{ clause } C_j)$$

$$y_i \in \{0,1\}, \quad \mathrm{i} = 1,2,\ldots,\mathrm{n} \ \ (\forall \text{ variable } \mathrm{x}_i)$$

Relax y_i, z_j and solve LP

x_i^*, y_j^* : optimal solution of cost $Z_{\mathrm{LP}}^* \geq Z_{\mathrm{IP}}^* = OPT$

Rounding : Set each x_i TRUE with probability y_i^*, independently

Algorithm LP

$$\Pr[\text{clause } C_j \text{ not satisfied}] = \prod_{i \in P_j} (1 - y_i^*) \prod_{i \in N_j} y_i^*$$

$$\sqrt[k]{\prod_{i=1}^{k} a_i} \leq \frac{1}{k} \sum_{i=1}^{k} a_i$$

αi ≥0, i=1,2,…,k

$$\Pr[\text{clause } C_j \text{ not satisfied}] = \prod_{i \in P_j} (1 - y_i^*) \prod_{i \in N_j} y_i^* \leq \left[\frac{1}{l_j} \left(\sum_{i \in P_j} (1 - y_i^*) + \sum_{i \in N_j} y_i^* \right) \right]^{l_j}$$

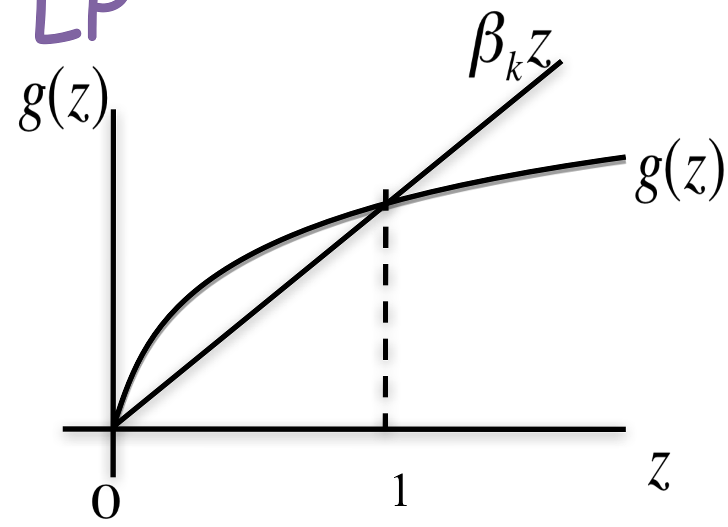
$$= \left[1 - \frac{1}{l_j} \left(\sum_{i \in P_j} y_i^* + \sum_{i \in N_j} (1 - y_i^*) \right) \right]^{l_j}$$

$$\sum_{i \in P_j} y_i^* + \sum_{i \in N_j} (1 - y_i^*) \geq z_j^*$$

$$\leq \left(1 - \frac{z_j^*}{l_j} \right)^{l_j}$$

Algorithm LP

$$\Pr[\text{clause } C_j \text{ satisfied}] \geq 1 - \left(1 - \frac{z_j^*}{l_j}\right)^{l_j}$$



$$g(z) = 1 - \left(1 - \frac{z}{k}\right)^k$$

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

g(z) concave function of z, g(0) = 0, g(1) = β_k $\Rightarrow$ **$g(z) \geq \beta_k z$**, z $\in$[0,1]

Hence,

$$\Pr[\text{clause } C_j \text{ satisfied}] \geq 1 - \left(1 - \frac{z_j^*}{l_j}\right)^{l_j} \geq \left[1 - \left(1 - \frac{1}{l_j}\right)^{l_j}\right] z_j^* = \beta_j z_j^*$$

$$\beta_j = 1 - \left(1 - \frac{1}{l_j}\right)^{l_j}$$

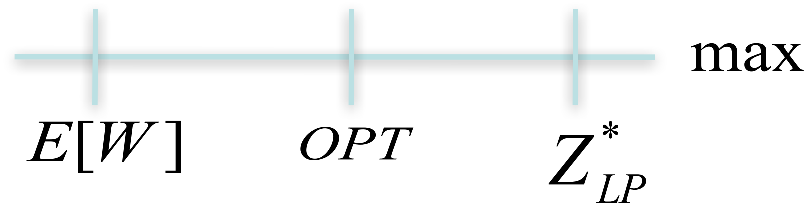
Algorithm LP

$\beta_j = 1 - \left(1 - \frac{1}{l_j}\right)^{l_j}$ is a decreasing function of l_j

Assume that all clauses have at most k literals, i.e. $l_j \le k$

$$E[W] = \sum_{j=1}^{m} E[w_j] = \sum_{j=1}^{m} \Pr[C_j = 1] \ge \beta_j \sum_{j=1}^{m} z_j^* \ge \beta_k Z_{LP}^* \ge \beta_k OPT$$

Ratio $\beta_k = 1 - (1 - \frac{1}{k})^k$



$$(1 - \frac{1}{k})^k < \frac{1}{e}, \forall k \in Z^+, \text{ that is } \beta_k > 1 - \frac{1}{e} = 0{,}632$$

$$E[W] \ge 0.632 OPT$$

Algorithm LP, γ = 3/4

(LP)

$$\max \sum_{j=1}^{m} z_j$$

$$\sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j, \forall C_j$$

$$z_j \in \{0,1\}, \quad \mathrm{j} = 1,2,\ldots,\mathrm{m} \quad (\forall \text{ clause } C_j)$$

$$y_i \in \{0,1\}, \quad \mathrm{i} = 1,2,\ldots,\mathrm{n} \quad (\forall \text{ variable } \mathrm{x}_i)$$

$$\phi = (x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_2})$$

<u>LP returns</u> $y_i^* = 1/2, \forall i, \quad z_j^* = 1, \forall j, \quad Z_{\mathrm{LP}}^* = 4$

<u>But OPT=3 and hence</u> $\gamma = \dfrac{3}{4}$

Algorithms R1 + LP

Run both and return the best (or run either R1 or LP uniformly at random)

Consider a clause C_j containing k literals

$$\text{R1}: \; E[X_j|R1] \geq a_k \geq a_k z_j^* \;\; (\text{as } \; z_j^* \leq 1)$$

$$\text{LP}: \; E[X_j|LP] \geq \beta_k z_j^*$$

$$E[X_j] = \max\{E[X_j \mid R1], E[X_j \mid LP]\}$$

$$\geq \frac{1}{2}(E[X_j \mid R1] + E[X_j \mid LP]\} = \frac{a_k + \beta_k}{2} z_j^*$$

$$\left.\begin{array}{l} a_1 + \beta_1 = a_2 + \beta_2 = \frac{3}{2}, k = 1,2 \\ a_k + \beta_k \geq \frac{7}{8} + (1 - \frac{1}{e}) \geq \frac{3}{2}, \forall k \geq 3 \end{array}\right\} \Rightarrow E[X_j] \geq \frac{3}{4} z_j^*, \forall k$$

Hence, $E[X] = \sum_{j=1}^{m} E[X_j] = \frac{3}{4}\sum_{j=1}^{m} z_j^* = \frac{3}{4} Z_{LP}^* \geq \frac{3}{4} OPT$

Achieving γ without using Algo R1

Non-linear randomized rounding

g(y) a function $1-4^{-y} \le g(y) \le 4^{y-1}$, for $y \in [0,1]$

→ Set x_i to 1 with probability $g(y_i^*)$

If $C = x_1 \vee x_2 \vee \ldots \vee x_k$

analyze g(y)

$$\Pr[C_j = 1] = 1 - \prod_{i=1}^{k}(1 - g(y_i^*)) \ge 1 - \prod_{i=1}^{k} 4^{-y_i^*} = 1 - 4^{(-\sum_{i=1}^{k} y_i^*)}$$

$$\ge 1 - 4^{-z_j^*} \ge 1 - 4^{-1} \ge \frac{3}{4} \ge \frac{3}{4} z_j^*$$

$$E[W] = \sum_{j=1}^{m} \Pr[c=1] \ge \frac{3}{4}\sum_{j=1}^{m} z_j^* = \frac{3}{4} Z_{LP}^* \ge \frac{3}{4} OPT$$

This generalizes to any form of clauses