Approximation Algorithms and Linear Programming

Daya Gaur[1]

[1]Department of Computer Science and Engineering
Indian Institute of Technology Ropar
Thapar University IGGA

28 Oct 2010

Vertex cover

$$IP : minimize \sum_{v \in V} w_v x_v$$
$$x_u + x_v >= 1 \ \forall (u, v) \in E$$
$$x_v \in \{0, 1\} \ \forall v \in V$$

$$LP : minimize \sum_{v \in V} w_v x_v$$
$$x_u + x_v >= 1 \ \forall (u, v) \in E$$
$$x_v \geq 0 \ \forall v \in V$$

Matching

$$D-LP : minimize \sum_{(u,v)\in E} y_{uv}$$

$$\sum_{v\in N(u)} y_{uv} \leq w_u \text{ for all } u \in V$$

$$y_{uv} \geq 0 \;\forall v \in V$$

$$M : minimize \sum_{(u,v)\in E} y_{uv}$$

$$\sum_{v\in N(u)} y_{uv} \leq w_u \text{ for all } u \in V$$

$$y_{uv} \in \{0,1\} \;\forall v \in V$$

Set cover

$$IP : minimize \sum_{s \in S} w_s x_s$$

$$\sum_{s:v \in s} x_s >= 1 \;\forall v \in U$$

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

$$LP : minimize \sum_{v \in V} w_s x_s$$

$$\sum_{s:v \in s} x_s >= 1 \;\forall v \in U$$

$$x_s \geq 0 \;\forall s \in S$$

Set cover: Dual

$$LP : maximize \sum_{u \in V} y_u$$
$$\sum_{v \in s} y_v \leq w_s \ \forall s \in S$$
$$y_v \geq 0 \ \forall v \in V$$

$$ILP : maximize \sum_{u \in V} y_u$$
$$\sum_{v \in s} y_v \leq w_s \ \forall s \in S$$
$$y_v \in \{0,1\} \ \forall v \in V$$

Shortest paths in digraphs

$$IP : minimize \sum_{e \in E} w_e x_e$$

$$\sum_{e \in N(v)} x_e = \begin{cases} 1 & \text{if } v = s \\ -1 & \text{if } v = t \\ 0 & \text{otherwise} \end{cases}$$

$$x_e \in \{0, 1\} \ \forall e \in E$$

Vertex, Edge incidence matrix. (u, e) is 1 is edge goes out from u, -1 otherwise.
LP relaxation polytope is integral (integrality gap is 1).

Constrained shortest paths

$$IP: minimize \sum_{e \in E} w_e x_e$$

$$\sum_{e \in N(v)} x_e = \begin{cases} 1 & \text{if } v = s \\ -1 & \text{if } v = t \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{e \in E} d_e x_e \leq D$$

$$x_e \in \{0, 1\} \forall e \in E$$

Constrained shortest paths

$$IP_1 : min \sum_{e \in E} w_e x_e + \lambda(\sum_{e \in E} d_e x_e - D)$$

$$\sum_{e \in N(v)} x_e = \begin{cases} 1 & \text{if } v = s \\ -1 & \text{if } v = t \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{e \in E} d_e x_e \leq D$$

$$x_e \in \{0, 1\} \; \forall e \in E$$

Let $x*$ be the optimal integral solution to the constrainted shortest path problem. $v(IP, x*) \geq v(IP_1, x*)$ for $\lambda \geq 0$.

Constrained shortest paths

$$IP_L : min \sum_{e \in E} w_e x_e + \lambda(\sum_{e \in E} d_e x_e - D)$$

$$\sum_{e \in N(v)} x_e = \begin{cases} 1 & \text{if } v = s \\ -1 & \text{if } v = t \\ 0 & \text{otherwise} \end{cases}$$

$$x_e \in \{0,1\} \; \forall e \in E$$

Let x' be the optimal integral solution to IP_L.
$v(IP_1, x*) \geq v(IP_L, x') \; \forall \lambda \geq 0$.

Theorem

The value of optimal solution to IP_L is a lower bound on the value of $x*$.

Lagrangian relaxation

- IP_L is the Lagrangian relaxation.
- We want the best possible lower bound, therefore find λ such that optimal to IP_L maximized.
  - Largarangian can be solved using subgradient methods (note that the function might not be differentiable).
  - Or using column generation (Dantzig-Wolfe decomposition).
  - Lagrangian bound is atleast as good as the linear programming bound, for integral polytopes the two bounds coincide

Greedy Algorithm for Set Cover

```
R = {}
Sol = {}
while R != U
       s : min { w(s)/|s \ R| }
       Sol <- Sol union s
       R <- R union (s \ R)
       for all e in (s \ R)
           p(e) =  w(s)/|s \ R|
```

Analysis

Proof.

$$\sum_{e \in U} p(e) = w(Sol)$$

Consider a set $s = (s_1, \ldots, s_k)$

For all $i, p(s_i) \leq \dfrac{w(s)}{k-i}$

$$\sum_{s_i \in s} p(s_i) \leq w(s)H(k)$$

$p(e)/H(n)$ is feasible in the dual

By weak duality the performance ratio is $H(n)$.

$s_i's$ are ordered in the order they are covered by the greedy.

□

LP rounding

1. Construct an integer program for the problem.
2. Relax the integrality constraints.
3. Solve the linear programming relaxation (in polynomial time).
4. Construct an integral solution from the optimal LP solution.

Vertex Cover

$$IP : minimize \sum_{v \in V} w_v x_v$$
$$x_u + x_v >= 1 \ \forall (u,v) \in E$$
$$x_v \in \{0,1\} \ \forall v \in V$$

$$LP : minimize \sum_{v \in V} w_v x_v$$
$$x_u + x_v >= 1 \ \forall (u,v) \in E$$
$$x_v \geq 0 \ \forall v \in V$$

Let x^* be the optimal LP solution.

$$x_v = \begin{cases} 1 & \text{if } x_v^* >= \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases}$$

Vertex Cover

- Each constraint contains atleast one variable with value $\geq 1/2$ in x^*.
- Rounding gives a feasible integral solution.
- Value of the integral solution is at most double the value of the optimal LP solution.

Current best approximation ratio for Vertex Cover. Long standing problem to either improve or show that vertex cover cannot be approximated better than 2 (absolute constant).

Half integrality of Vertex Cover

Definition

A solution to an LP is an extreme point if it cannot be expressed as a convex combination of two other feasible solutions.

Lemma

Every extreme point solution is half integral i.e., $x_v \in \{0, 1/2, 1\}$.

Proof.

$$V_p = \{v \mid x_v^* > 1/2\} \qquad V_n = \{v \mid x_v^* < 1/2\}$$

$$a_v = \begin{cases} x_v^* + \varepsilon & \text{if } v \in V_p \\ x_v^* - \varepsilon & \text{if } v \in V_n \\ x_v^* & \text{otherwise.} \end{cases} \qquad b_v = \begin{cases} x_v^* - \varepsilon & \text{if } v \in V_p \\ x_v^* + \varepsilon & \text{if } v \in V_n \\ x_v^* & \text{otherwise.} \end{cases}$$

□

Multi processor scheduling

Definition

Input: n jobs and m machines, p_{ij} is the processing time for job i on machine j.
Problem: Assign each job to a machine so as to minimize the makespan (time by which all the machines finish).

$$IP : minimize \ t$$
$$\sum_{j \in M} x_{ij} = 1 \quad \forall i \in J$$
$$\sum_{i \in J} p_{ij} x_{ij} <= t \quad \forall j \in M$$
$$x_{ij} \in \{0,1\} \quad \forall \ i,j$$

$$LP : minimize \ t$$
$$\sum_{j \in M} x_{ij} = 1 \quad \forall i \in J$$
$$\sum_{i \in J} p_{ij} x_{ij} <= t \quad \forall j \in M$$

Integrality gap

$i_T = \{j \mid p_{ij} <= T\}$

$$
\begin{aligned}
IP(T): \mathit{minimize}\ 0 & \\
\sum_{j \in i_T} x_{ij} = 1 & \quad \forall i \in J \\
\sum_{i \in J} p_{ij} x_{ij} <= T & \quad \forall j \in M \\
x_{ij} \in \{0,1\} & \quad \forall\ i,j
\end{aligned}
$$

$$
\begin{aligned}
LP(T): \sum_{j \in i_T} x_{ij} = 1 & \quad \forall i \in J \\
\sum_{i \in J} p_{ij} x_{ij} <= T & \quad \forall j \in M \\
x_{ij} \geq 0 & \quad \forall\ i,j
\end{aligned}
$$

Plan

1. Guess the optimal makespan T
2. Construct an optimal extreme point solution to LP(T).
3. Assign fractionally scheduled jobs to machines such that no machines receives more than 1 job.

Step 3, can be performed because of the special structure of the extreme point solution.

Step 3 at most doubles the makespan of the fractional schedule.

Therefore the performance ratio is 2.

Extreme point solutions

Lemma

At most $n + m$ variables are fractional in any extreme point solution.

Proof.

Let r be the total number of linearly independent constraints, whose solution is the extreme point solution. Of these r, atleast $r - (n + m)$ should be of the type $x_{ij} \geq 0$, $(x_{ij} = 0)$. Therefore atmost $n + m$ variables are non-zero. □

Extreme point solutions

Definition

Connected graph G on vertex set V has property T, if it has atmost $|V|$ edges. Graph G has property T is every connected component has property T. $G = (\{J, M\}, \{(i,j) \mid 1 > x_{ij} > 0\})$

Lemma

G has property T.

Proof.

Extreme point solution restricted to connected component is also extreme point solution to LP(T) restricted to the connected component. Else x is not extreme point. Result follows from the previous Lemma. □

Extreme point solutions

Definition

Let F be the set of fractionally assigned jobs. H is subgraph of G on $F \cup M$

Lemma

H has property T.

Proof.

Equal number of job vertices and incident edges are removed from G to obtain H. □

Extreme point solutions

Lemma

H has a perfect matching.

Proof.

Every job vertex has atleast two edge incident on it, therefore each leaf is a machine. Pair a machine on the leaf with a job (along the fractional edge). Remove the vertices and the edge from H, subgraph still has property T. If left with even cycle (biparite graph) then pair off alternate edges. □

Rectangle Stabbing

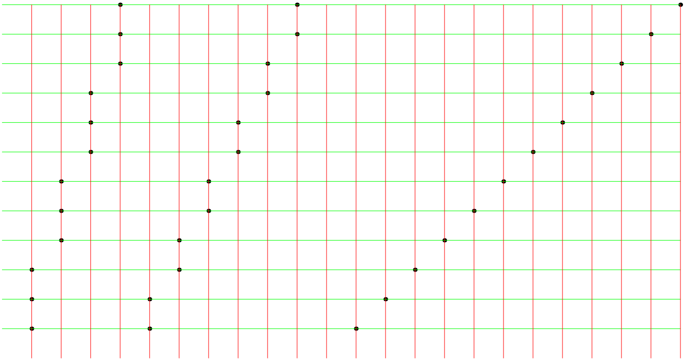
Definition

Given a set of n rectangles the objective is to stab all the rectangles using minimum number of axis parallel lines.

Natural Greedy Algorithm (restricted to points)

- Pick the line that covers the maximum number of points in each iteration.
- Approximation ratio is $O(\log n)$.
- Same analysis as for the set cover.

An example



All the horizontal lines (12) comprise the optimal solution, and the greedy algorithm will pick all the vertical lines (22). The example can be generalized.

Point Stabbing (min-max duality)

Theorem

The size of the minimum vertex cover is the same as the size of the maximum matching in a bipartite graph.

Point stabbing can be solved optimally.

Rectangle stabbing

L is set of all the lines. $r(H)$ is the set of horizontal lines that intersect rectangle r, $r(V)$ is the set of vertical lines that intersect rectangle r,

$$\begin{array}{rc} min & \sum_{l\in L} w_l x_l \\ & \sum_{l\in r(H)} x_l + \sum_{l\in r(V)} x_l \geq 1 \quad \forall\ r \\ & x_l \in \{0,1\} \\ P: min & \sum_{l\in L} w_l x_l \\ & \sum_{l\in r(H)} x_l + \sum_{l\in r(V)} x_l \geq 1 \quad \forall\ r \\ & x_l \geq 0 \end{array}$$

Rectangle stabbing

x^* is the optimal LP solution.

$$r(*) = \begin{cases} r(V) & \text{if } \sum_{l \in r(H)} x_l^* \geq \frac{1}{2} \\ r(H) & \text{otherwise.} \end{cases}$$

$$M : \min \quad \sum_{l \in L} w_l x_l$$
$$\sum_{l \in r(*)} x_l \geq 1 \quad \forall\ r$$
$$x_l \geq 0$$

Polytope associated with M is integral. $2x^*$ is feasible in M.

Set cover

Let x^* be the optimal solution.

1. Pick each set s in the cover with probability x_s^*.
2. Repeat Step 1. $c \log n$ times.

$$E[cost(setsPicked)] \leq c \log(n)\ cost(x^*)$$

$$P[cost(setsPicked) > 4c \log(n)\ cost(x^*)] \leq 1/4$$

$$LP : minimize \sum_{v \in V} x_s$$

$$\sum_{s: v \in s} x_s >= 1 \quad \forall\ v \in U$$

$$x_s \geq 0 \quad \forall\ s \in S$$

Set Cover

Lemma

Probability that sets picked do not form a cover $< 1/4$.

Proof.

Let $u \in U$ belong to sets $s_1, s_2, \ldots, s_k$, probability that atleast one of $s_1, s_2, \ldots, s_k$ is picked $\sum_{i=1}^{k} x_i^* \geq 1 - (1 - \frac{1}{k})^k \geq 1 - \frac{1}{e}$ as $\sum_i s_i^* \geq 1$.
Probability that u is not covered after $c \log(n)$ iterations $\leq \frac{1}{e^{c \log(n)}} \leq \frac{1}{4n}$.
Probability that picked sets do not form a cover $\leq n \frac{1}{4n} = \frac{1}{4}$. □

Theorem

Probability that picked sets form a cover with cost atmost $c \log(n)$ *times the optimal* $\geq 1/2$.

Pricing Method for Vertex Cover

Definition

$p_e \geq 0$ is the price associated with each edge e.
w_v is the cost associated with each vertex v.
Price p is fair if for every vertex $\sum_{e \text{ on } v} p_e \leq w_v$.

Theorem

A fair price is a lower bound on the cost of any vertex cover.

$$\sum_{e \text{ on } v} p_e \leq w_v$$

$$\sum_{v \in S} \sum_{e \text{ on } v} p_e \leq w(S)$$

Algorithm

Definition

A vertex is saturated if $\sum_{e \text{ on } v} p_e = w_v$
An edge is uncovered if neither of its endpoints are in the cover.

```
price of e = 0
   while there exists an
   uncovered edge e
       raise price on e
       without violating
       fairness
S = { v | v is saturated}
```

Analysis

Theorem

Cost of vertex cover produced is at most twice the fair price.

Proof.

Every vertex in the cover is saturated.

$$\sum_{e \text{ on } v} p_e = w_v$$

$$\sum_{v \in S} \sum_{e \text{ on } v} p_e = w(S)$$

$$2 \sum_{e} p_e \geq w(S)$$

□

Complementary Slackness

$$P: min \quad \sum_{i=1}^{n} c_i x_i$$
$$\sum_{i=1}^{n} a_{ji} x_i \geq b_j, \quad j = 1..m$$
$$x_i \geq 0$$

$$D: max \quad \sum_{j=1}^{m} b_j y_j$$
$$\sum_{j=1}^{m} a_{ji} y_j \leq c_i, \quad i = 1..n$$
$$y_j \geq 0$$

Complementary Slackness

Theorem

Let x, y be primal and dual feasible solutions. x, y are optimal if and only if following conditions are satisfied:

1. $x_i(\sum_{j=1}^{m} a_{ji} y_j - c_i) = 0$ *for all* $1 \leq i \leq n$
2. $y_j(\sum_{i=1}^{n} a_{ji} x_i - b_j) = 0$ *for all* $1 \leq j \leq m$.

Relaxed Complementary Slackness

Definition

Let x, y be primal and dual feasible solutions. x, y are said to satisfy relaxed complementary slackness condition if:

1. $x_i > 0 \implies \frac{c_i}{\alpha} \leq \sum_{j=1}^{m} a_{ji} y_j \leq c_i = 0$ for all $1 \leq i \leq n$
2. $y_j > 0 \implies (b_j \leq \sum_{i=1}^{n} a_{ji} x_i \leq \beta b_j) = 0$ for all $1 \leq j \leq m$.

RCSC

Theorem

If x, y are feasible and satisfy RCSC then $\sum_{i=1}^{n} c_i x_i \leq \alpha\beta \sum_{j=1}^{m} b_j y_j$.

Proof.

$$\sum_{i=1}^{n} c_i x_i \leq \alpha \sum_{i=1}^{n} \left(\sum_{j=1}^{m} a_{ij} y_j \right) x_i$$

$$\sum_{j=1}^{m} \left(\sum_{i=1}^{n} a_{ij} x_i \right) y_j \leq \beta \sum_{j=1}^{m} b_j y_j$$

□

pricing method for vertex cover revisited

$\alpha = 1, \beta = 2$
$x_v > 0 \implies \sum_{e \text{ on } v} p_e = w_v$
$p_e > 0 \implies x_u + x_v \leq 2$

- PCSC: Pick only saturated vertices in the cover.
- DCSC: From every edge pick atmost two vertices in the cover.

PSCS satisfied by pricing algorithm, DCSC condition satisfied automatically

Minimum Knapsack

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} c_i x_i \\
& \sum_{i=1}^{n} d_i x_i \geq D \\
& x_i \in \{0,1\}
\end{aligned}
$$

$$
\begin{aligned}
D : \max \quad & y_1 D \\
& d_i y_1 \leq c_i \ \forall i \\
& y_i \geq 0
\end{aligned}
$$

Flow cover inequalities

Let F be the set of all the items. Given a set $A \subseteq F$ of items $d(A) = \sum_{a \in A} d_a$, and the residual demand $D(A) = D - d(A)$. Define $d_i(A) = \min\{d_i, D(A)\}$. Items in $F \setminus A$ form another knapsack problem, d_i replaced with $d_i(A)$.

$$\begin{array}{rl} min & \sum_{i \in F} c_i x_i \\ \displaystyle\sum_{i \in F \setminus A} d_i(A) x_i \geq D(A) & \forall A \subseteq F \\ & x_i \geq 0 \end{array}$$

$$\begin{array}{rl} max & \sum_{A \subseteq} D(A) y_A \\ \displaystyle\sum_{A \subseteq F : i \notin A} d_i(A) y_A \leq c_i & \forall i \in F \\ & y_A \geq 0 \end{array}$$

Primal-dual algorithm

- PCSC: $x_i > 0 \implies \sum_{A \subseteq F : i \notin A} d_i(A) y_A = c_i$
- DSCP: $y_A > 0 \implies \sum_{i \in F \setminus A} d_i(A) x_i \leq 2D(A)$

```
y_{} <- 0
A <- {}
while D(A) > 0 do
        raise y_A until some
        constraint is tight
        x_i <- 1
        A <- A union {i}
end while
S <- A
```

Relaxed complementary slackness

- Algorithm can raise y_A without violating packing constraint.
- Primal condition is satisfied because algorithm only picks tight elements.
- Let elements in A be $\{1, 2, \ldots, k\}$ in the order they are picked.
- At the start $A = \phi$, $d_1(\phi)x_1 + \ldots + d_k(\phi)x_k \leq 2D(\phi)$. Consider the point in time when $y_{\{1,2,\ldots,k-1\}}$ was raised, then the first $k-1$ on the LHS in the equation sum to$< D(\phi)$, and the last term is $\leq D(\phi)$.
- The argument holds for any $A = \{1, \ldots, i-1\}$, $d_i(A)x_1 + \ldots + d_k(A)x_k \leq 2D(A)$.

R. Bar-Yehuda and S. Even, *A linear-time approximation algorithm for the weighted cover problem*. Journal of Algorithms, 2:198–203, 1981.

T. Carnes and D. B. Shmoys, *Primal-dual schema for capacitated covering problems*. In IPCO, pages 288–302, 2008.

G. B. Dantzig, L. R. Ford, D. R. Fulkerson, *A primal-dual algorithm for linear programs*, In Linear Inequalities and Related Systems, Editors Kuhn and Tucker, pages 171–181, Princeton University Press, 1956.

T. Gonzalez, editor, *Handbook of Approximation Algorithms and Meta Heuristics*, CRC Press, 2007.

D. R. Gaur and T. Ibaraki and R. Krishnamurti, *Constant Ratio Approximation Algorithms for the Rectangle Stabbing Problem and the Rectilinear Partitioning Problem*, Journal of Algorithms 43(1): 138–152, 2002.

D. Hochbaum, editor, *Approximation Algorithms for NP-hard problems*, PWS Publishing, Boston, 1997.

H. W. Kuhn, *The Hungarian method for the assignment problem*, Naval Research Logistics Quaterly, 2:83–97, 1955.

J. K. Lenstra, D. B. Shmoys and E. Tardos, *Approximation Algorithms for scheduling unrelated parallel machines*, Math. Prog., 46:259–271, 1990.

V. Vazirani, *Approximation Algorithms*, Springer, 2001.