Primal - Dual Algorithms

(P & S, Chapter 5. Hochbaum, Chapter 4 by Goemans & Williamson)

$$\begin{array}{llll} \min & c^T x & & \\ \text{s.t.} & Ax = b & \quad (P) & \\ & x \geq 0 & & \end{array} \qquad\qquad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y \leq c \end{array}$$

Complementary slackness: $x_i (c_i - A_i^T y) = 0 \quad \forall i.$

Algorithm:

→ Know dual feasible y.

Search for x satisfying complementary slackness, by solving a restricted primal.

If unsuccessful, use the optimal dual solution to the restricted primal to update y

Typically (RP) can be solved using combinatorial techniques.

Eg: Assignment problem:

$$\begin{array}{lll} \min & c^T x & \\ \text{s.t.} & \sum_i x_{ij} = 1 & \forall j \\ & \sum_j x_{ij} = 1 & \forall i \\ & x_{ij} \geq 0 & \end{array} \quad (P) \qquad\qquad \begin{array}{ll} \max & \sum u_j + \sum v_i \\ \text{s.t.} & u_j + v_i \leq c_{ij} \end{array} \quad (D).$$

Assume $c_{ij} \geq 0$, to simplify the notation.

Initialize with $u_j = 0, \forall j, \quad v_i = 0 \ \forall i$ (Better choices available)

Restricted primal:

$$\min \quad \sum s_j + \sum z_i$$

$$\text{s.t.} \quad \sum_i x_{ij} + s_j = 1 \quad \forall j$$

$$\sum_j x_{ij} + z_i = 1 \quad \forall i \qquad \text{(RP).}$$

$$x_{ij} = 0 \quad \text{if} \quad u_j + v_i < c_{ij}$$

$$x_{ij} \geq 0, \; s_j \geq 0, \; z_i \geq 0$$

This is a <u>maximum matching</u> problem in a bipartite graph, so can be solved by combinatorial techniques. Optimal value is # unmatched verts.
If the optimal value is zero, we have primal feasibility, so we are done.

Otherwise, get optimal soln s_j^*, z_i^*.

Dual to (RP) is:

$$\max \quad \sum \bar{u}_j + \sum \bar{v}_i$$

$$\text{s.t.} \quad \bar{u}_j + \bar{v}_i \leq 0 \quad \text{if} \quad u_j + v_i = c_{ij}$$

(RD)

(no constraint if $u_j + v_i < c_{ij}$, ∵ eliminate x_{ij} from (RP))

$$\bar{u}_j \leq 1$$

$$\bar{v}_i \leq 1$$

Can also be solved combinatorially.

Update: $(u,v) \leftarrow (u,v) + \alpha(\bar{u},\bar{v})$ for some $\alpha > 0$, chosen using a minimum ratio test.

Can be used, eg, for network design problems:
Which edges do we build?

Constraints of the form

$$\sum_{e \in \delta(S)} x_e \geqslant f(S)$$ for subsets S of the vertices.

Eg: each vertex must be connected to the rest of the graph,

so $$\sum_{e \in \delta(v)} x_e \geqslant 1 \quad \forall v \in V.$$

Integer programming formulation:

$$\min \sum c_e x_e$$

$$\text{s.t.} \quad \sum_{e \in \delta(S)} x_e \geqslant f(S) \quad (P) \quad \text{for subsets } S \subseteq V$$

$$x_e \text{ binary.}$$

Dual of LP relaxation: (provided all $f(S) \leq 1$, so don't need $x_e \leq 1$ explicitly)

$$\max \sum f(S) y_S \qquad (D)$$

$$\text{s.t.} \quad \sum_{S: e \in \delta(S)} y_S \leq c_e \qquad e \in E$$

$$y_S \geqslant 0.$$

Goemans & Williamson:

Can use the primal-dual algorithm to get approximation schemes with provable bounds, for some choices of $f(S)$.

Primal dual method gives y_S.

We then want

$$\sum_{e \in \delta(S)} x_e = f(S) \quad \text{if} \quad y_S > 0$$

Impose through objective $\rightarrow$ $$\sum x_e = 0 \quad \text{if} \quad \sum_{S: e \in \delta(S)} y_S < c_e$$

If these are not satisfied, find minimal sets $\{S\}$ that are violated.

Increase the corresponding y_S until one of the constraints

$\sum_{S: e \in \delta(S)} y_S \leq c_e$ becomes active.

Loop.

MATROIDS

Prototypes of independence systems with "nice" properties.

Defn Let $N = \{1, \ldots, n\}$ be a finite set, and let F be a set of subsets of N. $I = (N, F)$ is an INDEPENDENCE SYSTEM if $F_1 \in F$, $F_2 \subseteq F_1 \Rightarrow F_2 \in F$. Elements of F are called INDEPENDENT SETS, and the remaining subsets of N are called DEPENDENT SETS.

Eg: (i) Acyclic subgraphs
(ii) Linearly independent vectors (on homework)
(iii) Matchings

Def Given an independence system $I = (N, F)$, we say that F_i is a maximal independent set if $F_i \cup \{j\} \notin F$ for any $j \in N \setminus F_i$. A maximal independent set T is maximum if $|S| \leq |T|$ for all S in F.

Defn $M = (N, F)$ is a MATROID if M is an independence system in which for any subset $T \subseteq N$, every independent set in T that is maximal in T has the same cardinality.

omit ... on hold

Eg: (i), (ii) YES (i) is called GRAPHIC MATROID.

(iii) NO

(iv) Uniform matroids: Every subset of size $\le k$ is independent.

(v) Partition matroids: on disjoint finite sets E_i.
$E = \cup E_i$
$F \subseteq E$ indep if $|F \cap E_i| \le 1$ for each i
Eg: on bipartite graph G, set of edges covering vertices

Eg: Need maximal = maximum "FOR EVERY SUBSET" because, eg matching on C_4 vs matching on the subset 

Associate weights with elements of base set N

Theorem. Greedy algorithm solves the matroid problem

~~$\max \{ \sum_{i=1}^{n} c_i x_i : x$ is incidence vector of independent set $\}$.~~

$\max \{ \sum_{j \in S} c_j : S$ indep set $\}$. Eg: spanning tree

Theorem If (N, F) is an indep. sys but not a matroid then there exists a weight function for which greedy fails.

Eg: matching

Theorem Let $r(T)$ be the size of the largest independent set in $T \subseteq N$.

Then $r(T) + r(S) \ge r(T \cup S) + r(T \cap S)$. (~~Submodular function~~ A function satisfying this condition is SUBMODULAR.)

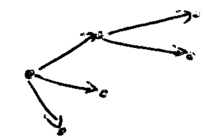
Eg: Spanning tree.

k-matroid intersection theorem

Given k matroids $M_i = (N, F_i)^{i=1,\dots,k}$ on base set N, and weight vector c_i, $i \in N$, solve

$$\max_S \left\{ \sum_{j \in S} c_j : S \in \bigcap_{i=1}^{k} F_i \right\}$$

Application for $k=2$: Find a branching in digraph, i.e. a spanning tree where at most one edge enters each vertex



Solvable in poly time for $k=2$.

NP-hard for $k=3$.

Reduction from HAMILTONIAN PATH ON DIRECTED GRAPH.

Dual Matroids:

[Call the independent sets of maximum cardinality the BASES OF A MATROID]

Then all the complements of the bases forms ~~another~~ the bases of another matroid: the dual matroid.

Eg: • Uniform matroid: Dual matroid is also uniform.

• Partition matroid: ~~Ind~~ Independent provided don't take all of a set C_i

• Acyclic subgraphs: Bases are complements of spanning trees

[Minimal dependent sets are circuits of a matroid.]

Dual ~~ind~~ independent sets: sets of edges that do not disconnect the graph. COCYCLE MATROID

Why does greedy work?

Assume there is a better soln.

There is something in this better soln that is not in the greedy soln.

Find the smallest dependent set when we add this extra element to the greedy soln.

Remove ~~the past~~ worst element from the dependent set, giving a new indep set, better than greedy.

Greedy should have found this instead of the one it did.