0.1 Matroid Theory

Informally: We contruct a solution element by element, at every step, the "locally-best" element. Problems: for which we can find an optimal solution with a greedy algorithm.
Notation: We are given: a ground set S, $I = \{J : J \subseteq S\}$, a family of subsets of S.

$$S = \{\text{triangle, circle, cross, rectangle}\}$$
$$I = \{\{\text{o}\}, \{\text{triangle, square}\}, \{\text{x}\}\}$$

Definition 1. An independence system: I satisfies:

1. $\emptyset \in I$
2. $\forall J \subseteq S, \forall J \in I$, and $J' \subseteq J$, then $J' \in I$

Example 1. $S = E(G)$
$I = \{F \subseteq S : F \text{ has no circuits}\}$
I is an independent system.

Example 2. A matching of $G = (V, E)$ is a subgraph of G such that every node is adjacent at most one edge. S is the edge-set of $G(V, E)$. $I = \{M \subseteq S : M \text{ is the edge-set of a matching of } G\}$

Example 3. S is the columns of a matrix B.
$I = \{J \subseteq S : \text{the columns in } J \text{ are linearly independent}\}$
I is an Independent System, because all subsets of $J \in I$ is linearly independent.

Definition 2. Given an independence system (S, I) and a subset $A \subseteq S$, an independent set $J \in I$ is A-maximal if

1. $J \subseteq A$
2. $\forall e \in A \setminus J, J \cup \{e\} \notin I$

Definition 3. A matroid is an indepence system such that for every $A \subseteq S$, all A-maximal independent sets have the same cardinality.

Example 4. Whatever A, A-maximal independent set are spanning trees in the components of A, so they have the same cardinality, I is set of forest.

1 January 9, 2018

Basis are not unique.

Definition 4. A matroid is an independent system such that for all $A \subset S$, all bases of A have the same cardinality.

Example 5. $I = \{F \subseteq S : F \text{ is a forest}\}$. Let $A \subseteq S$, and $H(W, A)$, the subgraph induced by A. Bases of A are spanning forests of H

Claim: all spanning forests of a graph have the same number of edges

Proof. Let $H_1(W_1, A_1), ..., H_k(w_k, A_k)$, be the connected componennts of H. Spanning forest of H are unions of spanning trees for $H_1, ..., H_k$. Therefore, they all have $(|w_1| - 1) + ... + (|w_k| - 1)$ edges. $\square$

Example 6. $I = \{F \subset S : F \text{ is a matching}\}$, not a matroid though. Some basis don't have the same cardinality

Example 7. $I = \{J : \text{columns in } J \text{ are linearly independent}\}$. Let $A \subset S$, and D a matrix formed with the column in A. How many linearly independent columns of D can we pick? $rank(D)$. So every basis of A has $rank(D)$ columns, hence matroid.

Definition 5. The rank function of a matroid is a function r such that $r(A)$ is a cardinality of all bases of A.

Definition 6. A basis of a matroid (S, I) is a S-maximal.

Problem: Maximum-weight independent set
Let $M = (S, I)$ be a matroid. Given

1. an oracle which $\forall A \subseteq S$ tells you whether $A \in I$ or not
2. weight $w_e, \forall e \in S$

Find a maximum-weight $J \in I$ where $w(J) = \sum_{e \in J} w_e$

1.1 The greedy algorithm for max-weight independent set

WLOG, let $S = \{e_1, ..., e_m\}$ where $w_{e_1} \geq ... \geq w_{e_m}$

```
Initialize J = ∅
for i = 0, ..., m do
    if w_{e_1} > 0 and J ∪ {e_i} ∈ I then
        J = J ∪ {e_i}
    end if
end for
```

2 January 22, 2018

Theorem 1. *The greedy algorithm will return a max-weight independent set.*

Proof. Assume the theorem is false. Let J be the independent set returned by the algorithm. Let J^* be a max-weight independent set. $w(J^*) > w(J)$. Let $J = \{e_1, ..., e_m\}$ where $w_{e_1} \geq ... \geq w_{e_m}$ Let $J = \{e_1, ..., e_m\}$ where $w_{e_1} \geq ... \geq$

w_{e_m}. Let $J^* = \{q_1, ..., q_l\}$ where $w_{q_1} \geq ... \geq w_{q_l}$. Let k be the smallest index such that $w_{e_k} < w_{q_k}$.

$$J; w_{e_1} \geq ... \geq w_{e_k} \geq w_{e_m} \geq 0 \geq ... \geq 0$$
$$J^*; w_{q_1} \geq ... \geq w_{q_k} \geq w_{q_m}$$

We know that $e_k \notin \{q_1, ..., q_k\}$. thus $\forall i \leq k$, either $q_i \in \{e_1, ..., e_{k-1}\}$ or $\{e_1, ..., e_{k-1}, q_i\} \notin I$. Let A be $\{e_1, ..., e_{k-1}, q_1, ..., q_k\}$, then $\{e_1, ..., e_{k-1}\}$ is basis of A. But $\{q_1, ..., q_k\} \subset A$ and $\in I$. There exists another basis basis of A with cardinality $\geq k$. Contradiction $\square$

2.1 Continuous Knapsack problem

Variables: $x_1, x_2, x_3, ..., x_n$
How much of each object we take $\in [0, 1]$?
Formulation: max $\sum c_i x_i$ such that $\sum a_i x_i \leq b$

1: $v_i = \frac{c_i}{a_i}$
2: $w =$ sorted(range(n), key=lambda i: v_i)
3: $l = 0$
4: **for** i in w **do**:
5: $\quad x_i = min\{\frac{b-l}{a_i}, 1\}$
6: $\quad l = l + a_i x_i$
7: **end for**

3 January 24, 2018

3.1 0-1 Knapsack Problem

max $\sum_{j=1}^{n} c_j x_j$ such that $\sum_{j=1}^{n} a_j x_j \leq b$
$a_j, c_j, b > 0$, $a_j, b \in \mathbb{Z}$
Let $f_r(\lambda) = max \sum_{j=0}^{r} c_j x_j$, such that $\sum_{j=1}^{r} a_j x_j \leq \lambda, x_j \in \{0, 1\} \forall j$
Could we give the value of $f_r(\lambda)$ in terms of $f_s(\mu)$ where $s \leq r$ and $\mu < \lambda$. Let x^* be an optimal solution. Looking at the last object either $x_r^* = 0$ or $x_r^* = 1$
If $x_r^* = 0$, $f_r(\lambda) = f_{r-1}(\lambda)$
If $x_r^* = f_r(\lambda) = c_r + f_{r-1}(\lambda)$
$f_r(\lambda) = max\{f_{r-1}(\lambda), c_r + f_{r-1}(\lambda - a_r)\}$
$f_0(\lambda) = 0 \forall \lambda$ and $f_r(\lambda) = 0 \forall \lambda \leq 0$

3.2 Dynamic Programming approach

Compute all $f_r(\lambda)$ for $r = 1, ..n$ and $\lambda = 1, ..., b$ and store the results. How to obtain x^*? keep track of whih term of max was larger

Example 8. $max 10x_1 + 7x_2 + 25x_3 + 24x_4$
$2x_1 + x_2 + 6x_3 + 5x_4 \leq 7$

Complexity: (arithmetic model): $O(n * b)$

3.3 Integer Knapsack

$max \sum_{j=1}^{n} c_j x_j$ such that $\sum_{j=1}^{n} a_j x_j \leq b$, $x \in \mathbb{Z}_+^n$, $a, b, c > 0$, $a_j, b \in \mathbb{Z}, \forall j$
Let $g_r(\lambda) = max \sum_{j=1}^{n} c_j x_j$ such that $\sum_{j=1}^{r} a_j x_j \leq \lambda, x \in \mathbb{Z}_+^r$
Let x^* be an optimal solution,

$$
\begin{aligned}
if x_r* = 0 g_r(\lambda) = g_{r-1}(\lambda) \\
if x_r* = 1 g_r(\lambda) = 1c_r + g_{r-1}(\lambda - a_r) \\
if x_r* = 2 g_r(\lambda) = 2c_r + g_{r-1}(\lambda - 2a_r) \\
... \\
if x_r* = \lfloor \frac{\lambda}{a_r} \rfloor \\
g_r(\lambda) = \frac{\lambda}{a_r} c_r + g_{r-1}(\lambda \frac{\lambda}{a_r} a_r) \\
g_r(\lambda) = max_{t=0,...,\lfloor \frac{\lambda}{a_r} \rfloor} \{ tc_r + g_{r-1}(\lambda - ta_r) \}
\end{aligned}
$$

$$
\begin{aligned}
if x_r^* \geq 1 \\
g_r(\lambda) = c_r + g_r(\lambda - a_r) \\
if x_r^* = 0 \\
g_r(\lambda) = g_{r-1}(\lambda - a_r) \\
g_r(\lambda) = max g_{r-1} c_r, c_r + g_r(\lambda - a_r)
\end{aligned}
$$

Note1: Complexity $O(nb)$ Note2: What if $a_j, b \in \mathbb{Q}$ $max 3x_1 + 2x_2 + 3x_3$
such that $\frac{3}{4}x_1 + \frac{2}{3}x_2 + 2x_3 \leq \frac{25}{3}$
multiply by the lcm of the denomiators her $lcm(4, 3, 1, 3) = 12$ such that $lcm 9x_1 + 8x_2 + 24 \leq 100$. $x \in \mathbb{Z}_+$
Notes3: $max 3x_1 + 2x_2 + 3x_3$
such that $1000x_1 + 100x_2 + 100x_3 + 200x_4 \leq 10000$
You can constraint by the gcd of the numbers that appear in it:

$$
\begin{aligned}
max... \\
\text{such that} 10x_1 + 1x_2 + 2x_3 \leq 100
\end{aligned}
$$

$x \in \mathbb{Z}_+^3$

4 January 29, 2017

Computational Complexity
Note1 : We use the bit model
Note2: Given an optimization problem, $max\{f(x) : x \in S\}$, its decision version is: "Is there an $x \in S$ with value for $ff(x) \geq k$? where k is also given."

Note3: If we can solve a decision problem, then we can also solve its optimization version (by bisection on k), assuming there are known finite bounds on $f(x)$ for $x \in S$.

Definition 7. NP is the class of decision problem with the property that: for any instance whose answer is YES, there exist:

1. a certificate
2. a polynomial alogrithm that given the certificate, can the YES answer.

Example 9. 0-1 Knapsack(decision version): Is there $x \in S = \{x \in \{0,1\}^n : \sum_j a_j x_j \leq b\}$ for which $\sum_j c_j x_j \geq k$. If YES, certificate $x^* \in S$. proof: check $\sum_j a_j x_j^* \leq b$ and $\sum_j c_j x_j^* \geq k$ and $x* \in \{0,1\}^n$ 0-1 Knapsack is in NP

Definition 8. P is the complexity class of problems in NP such that there exists a polynomial algorithm to solve them

Example 10. 0-1 Knapsack is in NP.
Encoding size: $L = \sum_j \log a_j + \sum_j \log c_j + \log b + \log k$
Dynamic Programming: $O(n * b * L) = b = O(2^L)$
$O - 1$ Knapsack is not known to be P

Definition 9. Give $H, Q \in NP$, H is polynomially reducible to Q if all instance of H' can be converted into an instance of Q in polynomial time.

Example 11. SAT (satisfiablity): Does there exists boolean values $x_1, ...x_n$ such that a given boolean expression $f(x_1, ..., x_n)$ is true. ex: $f(x) = (x_2 \wedge x_3) \vee \neg x_1 : YES$, $x = (FALSE, FALSE, FALSE)$

Example 12. BiP: (binary integer programming): Does there exist values for $x \in \{0,1\}^n$ such that $Ax \geq b$.

Reduction of SAT to BiP:
Put $f(x_1, ..., x_n)$ in conjuctive normal form $f = \wedge_{i=1,..,m}(\vee_{j \in C_i} x_j)(\vee_{j \in D_i} \not{x}_j)$, C_i, D_i where C_i where positive and D_j in the not form. We are able to do that in polynomial time. Then express SAT as

$$\sum_{j \in C_j} x_j + \sum_{j \in D_i} (1 - x_j) \geq 1$$
$$\forall i = 1, ..., m$$

Definition 10. A problem $H \in NP$ is $NP-complete$ if all $q \in Np$ are polynomially reducible to H.

SAT is NP-complete

4.1 January 31, 2018

Theorem 2. *SAT is NP-complete*

Question: Given H in NP, is H in P? Is H NP-complete?
Proposition: Given H, Q in NP, if H, if H is polynomially reducible to Q and Q in P, then H is in P
Proposition: Given H, Q in NP, if Q is NP-complete and and Q is polynomially reducible to H, then H is also NP-problem.

Example 13. We have seen that SAT is polynomially reducible to BIP, SAT NP-complete, so BIP is also NP-complete

Conjecture: $P \neq NP$
Proposition: If there existed H NP-complete and H in P then all problems in NP are also in P ($P = NP$)

4.2 Linear Programming

$minc^T x$ such that $Ax = b, x \geq 0, x \in \mathbb{R}^n$

Theorem 3. *The set $\{x \in \mathbb{R}^n : Ax = b\}$ is a polyhedron.*

Theorem 4. *If $min\{c^T x : Ax = b, x \in \mathbb{R}^n_+\}$ has optimal solution, then least one of them is a vertex*

Definition 11. A basis of $\{x \in \mathbb{R}^n_+ : Ax = b\}$ is a subset of the columns of A such that the corresponding submatrix is invertible.

Theorem 5. *Each vertex $\bar{x}$ of $\{x \in \mathbb{R}^n : Ax = b\}$ corresonds to one (or more) basis. If B are the basic colums, N are the non-basic ones, B is the basis matrix, then $\bar{x}_N = 0, \bar{x}_B = B^{-1}b$*

Definition 12. Let $\bar{x}$ be constructed as $\bar{x}_B = B^{-1}b$, $\bar{x}_N = 0$, then $\bar{x}$ is a baisc solution. If, in addition, $\bar{x} \geq 0$, then $\bar{x}$ is a basic feasible solution (a vertex)

Proposition 1. *If B is invertible, then $\{x \in \mathbb{R}^n : Ax = b\} = \{x \in \mathbb{R}^n_+ : B^{-1}Ax = B^{-1}b\}$*

Proposition 2. *Let $f(x) = c^T x$ and $g(x) = (c^T - c_B^T B^{-1} A)x$. Then $f(x) = g(x) + K$ where $K \in \mathbb{R}$ is constant.*

Proof. $g(x) = (c^T - c^T B^{-1} A)x = c^T x - c^T B^{-1} Ax = f(x) - c_B^T B^{-1} b$ which is constant □

Proposition 3. *$min\{c^T x : Ax = b, x \in \mathbb{R}^n_+\}$, is equivalent to $min\{(c^T - c_B^T B^{-1} A)x : B^{-1}Ax = B^{-1}b, x \in \mathbb{R}$, is equivalent to for any B invertible.*

4.3 February 2, 2018

$min\{c^T x : Ax = b, x \geq 0\}$ is equivalent to $min\{\bar{c}^T x : \bar{A}x = \bar{b}\}$ where $\bar{A} = B^{-1}A, \bar{b} = B^{-1}b, \bar{c}^T = c^T - c_B B^{-1} A$
Note: that $A = [BN], c^T = [c_B c_N]$. Then,

$$\begin{aligned} \bar{A} = B^{-1}A = B^{-1}[BN] &= [IB^{-1}N] \\ \bar{c}^T &= [c_B^T c_N^T] - c_B B^{-1} A \\ &= [c_B^T c_N^T] - c_B^T [IB^{-1}N] \\ &= [0 c_N^T - c_B^T B^{-1} N] \end{aligned}$$

What happens if we change the basis? Either x_4 or x_5 will become basic. If x_4 enters the basis, it will go from zero(nonbasic) to some $\lambda \geq 0$ (to stay feasible), x_5 stays zero (nonbasic) and $x_1, x_2 x_3$ will change. Thus $2x_4 - x_5$ will increase (to 2λ) which is not what we want. Instead, if x_5, enters the basis then $2x_4 - x_5$. Now assume x_5 enters the basis, then x_5 increases (from zero), what happens to x_1, x_2, x_3? We have

$$\begin{aligned} x_1 &= 2 + 2x_5 \geq 0 \rightarrow \text{ok} \\ x_2 &= 2 - x_5 \geq 0 \rightarrow x_5 \leq 2 \\ x_3 &= 3 - x_5 \geq 0 \rightarrow x_5 \leq 3 \end{aligned}$$

If we want x_1, x_2, x_3 to stay ≥ 0, then we need $x_5 \geq 2$, Setting x_5 to 2, x_2 becomes zero. This gives us a new basis $\{x_1, x_3, x_5\}$.
We will get $\bar{x} = (60102), \bar{z} = x_2 = 0$
New basis $\{x_1, x_3, x_5\}$
Simplex method: Start with a basis B such that $\bar{x} \geq 0$ While $\exists \bar{c}_j < 0$: x_j enters the basis, $i = argmin_i\{\frac{b_i}{A_{ij}} \mid A_{ij} > 0\}$, x_{B_i} leaves the basis
Simplex method with general bounds Consider

$$\begin{aligned} Ay &= b \\ y &\geq l \\ y &\leq u \end{aligned}$$

Let $x = y - l$ (i.e. $x \geq 0$) s $y = x + l$

$$Ax = b - Alx \geq 0x \leq u - l$$

Note: $\alpha \leq \beta$ is equivalent to $\alpha + S = \beta$, $s \geq 0$

$$\begin{aligned} Ax &= b' \\ x + s' &= u' \\ x, s &\in \mathbb{R}^n_+ \end{aligned}$$

$$\begin{bmatrix} A & 0 \\ I & I \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} = \begin{bmatrix} b' \\ u' \end{bmatrix} x, s \geq q0$$

Proposition 1: IF s_j is nonbasic (thus $s_j = 0$, so $x_j = u_j$) then x_j is basic.

Proof. Look at the $(m + j)$-th of the constraint matrix. Only the j-th and $(n + j)$-th columns are nonzero. If both are non-basic, the basis has only zeros in that row, so it is not invertible → Contradiction ☐

Proposition 2: Let k be the number of s_j that are nonbasic, then we have $m+k$ basic x_j.

Proof. WE have $n - k$ basic s_j out of $m + n$ total basic columns. So we have $(m + n) - (n - k) = m + k$ basic x_j. ☐

Corollary 1. *For any given feasible basic of LP' we have*

1. *kx_j basic variables at upper bound $x_j = u'_j$*
2. *m other x_j with $0 \leq x_j \leq u'_j$*
3. *$n - k - m$ x_j nonbasic with $x_j = 0$*

We can partition x into (x_B, x_L, x_U) corresponding to $(2), (3), (1)$ respectively with $x_L, x_U = u'_U$. Equivalently, we can partition y into y_B, y_L, y_U with $y_L = l_L, y_U = u_U$ and $A = [BN_LN_U]$. Then $Ay = b$ gives $By_B + N_Ll_L + N_Uu_U = b, y_B = B^{-1} - N_Ll_L - N_Uu_U$
Note: if $\lambda \geq u_j - l_j$, then B does not change and x_j flips bounds.

5 February 7, 2018

5.1 Dual simplex method

$minc^Tx : Ax = b, x \geq 0, x \in \mathbb{R}^n$
$maxb^Ty : A^Ty \leq c, y \in \mathbb{R}^m$

Let us try to apply the simplex method on the dual.
$min - b^Ty : A^Ty + s = c, s \geq 0$. We can apply the simplex method with generalized bounds $-\infty \leq y \leq +\infty$
As long as the problem is bounded, all y have to be basic. WLOG, we reorder the rows of A^T so that the last n columns are basic. Let us partition this matrix. The dual becomes

$$min - b^Ty \tag{1}$$
$$N^Ty + s_N = c_N \tag{2}$$
$$B^Ty + s_B = c_B \tag{3}$$

We now compute (a) the reduced costs and (b) the bais feasible solution, for y, s_N basic and s_B non-basic.
Reduced cost
We want the costs for y and s_N to be zero, so we want the objective expressed in terms of s_B only (plus a constraint)

$$y = B^{T*-1}(c_B - s_B)$$
$$-b^T y = -b^T B^{T*-1}(c_B - s_B) = -b^T B^{T*-1} c_B + b^T B^{-1*T} s_B$$

The objective becomes $min(B^{-1}b)^T s_B$
The dual basic solution:
s_B nonbasic so $\bar{s_B}$

$$\bar{y} = B^{-1T}(c_B - \bar{s}_B) = B^{T*-1} c_B$$
$$\bar{s} = c_N - N^T y = c_N - N^T B^{T*-1} c_B$$
$$s^T = [\bar{s}_B^T \bar{s}_N^T] =$$
$$[0^t c_n^t - c_b^t b^{-1} n]$$
$$[c_B^T - c_B^T B^{-1} B c_n^t - c_b^t b^{-1} n]$$
$$= c^T - c_B^T B^{-1} A$$

Conclusion: There is a 1 to 1 correspondence between primal and dual basis. Given a primal basis B,

- The primal basic solution is $\bar{b} = \bar{x}_B = B^{-1}b$
- The primal reduced costs are $c^T = c^T - c_B^T B^{-1} A$
- The dual basic solution is $\bar{y}^T = c_B^T B^{-1}$, $\bar{s}^T = \bar{c}^T = c^T - c_B^T B^{-1} A$
- the dual reduced costs are $\bar{b} = B^{-1}b$

Dual simplex method:
Start B such that $\bar{c} \geq 0$ while $\bar{b}_i < 0$ x_{B_i} leaves the basis $j = argmin_j\{\bar{c}_j / -\bar{a_i j} | a_{ij} < 0\}$ x_j enters the basis

5.2 February 9, 2018

WE have seen that, in the dual, the reduced costs corresponding to the basis, for the (dual) nonbasic variables s_B, are $\bar{b} = B^{-1}b$. s_{B_i} entering the dual basis is equivalent to x_{B_i} leaving the primal basis.
A dual basis is feasible if $\bar{c} \geq 0$ (i.e. B is dual feasible)

Example 14.

$$
\begin{aligned}
min x_3 + 3x_4 + x_5 \\
x_1 - x_3 - x_4 + x_5 = -1 \\
x_3 + 2x_4 + 2x_5 + x_6 = 4 \\
x_2 + x_3 + x_4 + x_5 = 4 \\
x \geq 0
\end{aligned}
$$

Remarks:

- The (primal) basis is $\{x_1, x_6, x_3\}$
- it is dual feasible because $\bar{c} \geq 0$
- primal solution: $\bar{x} = [-140003]^T$, $obj = 2$

Apply the dual simplex method

- We select a dual reduced cost < 0, i.e. $\bar{b_1} = -1 \leq 0$. s_{b_1} enters the dual basis, x_1 leaves the primal basis
- We will update $\bar{c}$ to refect the pivot (they become $\bar{c}'$) $\bar{c_1}'$ can be $\neq 0$, one of $\bar{c}'_3, \bar{c}'_4, \bar{c}'_5$ must be zero. We achieve by adding a multiple of the pivot row, $x_1 - x_3 - x_4 + x_5$ to the objective function.

If we choose

$$
\begin{aligned}
\bar{c}'_3 = 0, (1)' = (1) + (2) \text{ we get } min x_1 + 2x_4 + 2x_5 \\
\bar{c}'_4 = 0, (1)' = (1) + 3(2) \text{ we get } min 3x_1 - 2x_3 + 4x_5 \\
\text{The next basis is not dual feasible} \\
\bar{c}'_5, (1)' = (1) - (2), \text{we get } min - x_1 + 2x_3 + 4x_4 \\
\text{not dual feasible} \\
\text{(prima) entering column} argmin_j \{ \frac{\bar{c}_j}{-\bar{a}_{ij}} \bar{a}_{ij} < 0 \}
\end{aligned}
$$

New basis $\{x_3, x_6, x_2\}$

$$
\begin{aligned}
min x_1 + 2x_4 + 2x_5 + k \\
-x_1 + x_3 + x_4 - x_5 = 1 \\
-x_1 + x_4 + 3x_5 + x_6 = 2 \\
x_1 + x_2 + 2x_5 = 3 \\
x* = [031002]^T, obj* = 3
\end{aligned}
$$

Dual simplex method with general bounds

$$
\begin{aligned}
minc^T x \\
Ax = b \\
l \leq x \leq u
\end{aligned}
$$

We partition the variables/columns of A into $A = [BLU]$, $Ax = b$ so $[BLU][x_b x_L x_U]^T = b$

$$
\begin{aligned}
Bx_B + Lx_L + Ux_U = b \\
\bar{x}_B = B^{-1}(b - Lx_L - Ux_U) \\
x_L = l_L \\
x_U = u_U \\
\bar{A} = B^{-1}A \\
\bar{c}^T = c^T - c_B^T B^{-1}A
\end{aligned}
$$

Start with B such that $\bar{c}_L \geq 0, \bar{c}_U \leq 0$, while either $\bar{x}_{B_i} < l_{B_i}$ or $x\bar{_{B_i}} > u_{B_i}$
x_{B_i} leaves the (primal) basis B, to its violated bound $j = argmin_j\{\frac{\bar{c}}{-\bar{a}_{ij}}|\bar{a}_{ij} < 0, x_j$at LB, x_{b_I} goes to LB$\}$ $j = argmin_j\{\frac{\bar{-c}}{\bar{a}_{ij}}|\bar{a}_{ij} > 0, x_j$at UB, x_{b_I} goes to LB$\}$ $j = argmin_j\{\frac{\bar{c}}{\bar{a}_{ij}}|\bar{a}_{ij} > 0, x_j$at LB, x_{b_I} goes to UB$\}$ $j = argmin_j\{\frac{\bar{-c}}{\bar{-a}_{ij}}|\bar{a}_{ij} < 0, x_j$at UB, x_{b_I} goes to UB$\}$
x_j enters the basis.

6 February 12, 2018

To start the dual simplex method, we need a partition of the columns, $\{1, ..., n\}$ into (B, L, U) such that B is a basis, $\bar{c}_L \geq 0$, $\bar{c}_U \leq 0$.
Given a basis B, if all the variables have finite bounds, it is easy: Compute $\bar{c}^T = c^T - c_B^T B^{-1}A$. B_y construction $\bar{c}_B = 0$ For every j nonbasic,

1. if $\bar{c}_j \geq 0$, put x_j at LB
2. if $\bar{c}_j < 0$, put x_j at UB

Then, how to find the basis, i.e. a $m \times m$ submatrix of A that is invertible?
In pratice, add m variables fixed to zero to problem.

Example 15.

$$
\begin{aligned}
min - 2x_1 - 3x_2 + x_3 \\
suchthat x_1 - 2x_2 - x_3 + x_4 = 2 \\
-x_1 - x_2 + 2x_3 \\
0 \leq x_1, x_2, x_3 \leq 1 \\
0 \leq x_1, x_2 \text{UB} \\
x_3 \leq 1\text{LB} \\
0 \geq x_4, x_5 \leq 0
\end{aligned}
$$

Theorem 6 (Khachiyan, 1979). *(LPD) is in LPD is in P*

The proof is constructive: it gives an algorithm (the "ellipsoid method") that solves (LP) in $O(n^4 L)$. where L is the input size, size of coefficient, size of the matrix. The algorithm also gives an optimal solution x^*.

Theorem 7 (Karmakkar, 1984). *There is an algorithm ("interior point method") that solves (LP) in $O(n^{3.5}L)$ time. Note: contrary to the previous one, this algorithm works well in practice.*

6.1 What about the simplex method?

1. There are many variants of the simplex method, depending on how $\bar{c}_j$ (primal) or $\bar{b_i}$ (dual) is chosen
2. All practical variants of the simplex method have been shown to have exponents running time in some bad case.
3. The simplex method performs well on practical instances, although the interior point method tend to be faster on large problem.

6.2 Warm-starting the dual simplex method

Given an optimal tableau (B, L, U), changing bounds of the variables, or right-hand side b_i maintains dual feasibility. ($\bar{c}^T = c^T - c_B^T B^{-1} A$ is unchanged).
We can apply the dual simplex method after the change, starting from the previously-optimal tableau, and since most of the problem remained the same, we will need few iterations in practice.

6.3 Floating-point arithmetic

In theory, integers are represented as a string of binary digits (bits). In practice, those strings have a fixed number of bits. On modern CPU: 64-bits

Example 16. nonnegative integers in base 10, with 3 digits: 000, 001, 002, ..., 999 $10^3 = 1000$. nonnegative integers in base 2, with 64 bits. $00...0, 00...01, ..., 11...1$, $2^6 4 \approx 1.84 * 10^{19}$

7 February 14, 2018

In theory, we can represent rational numbers as fraction of two (arbitrarily large) integers. In practice, real numbers are represented in floating-point notation:
1 bit for sign 52 bits mantisa 1 bit for sign in exponent 10 bits exponent
After every operation +, -, $\times, /$, the result is rounded to the nearest representable floating-point number

$$(a + b) + c \neq a + (b + c)$$

Example 17. In base 10, with 3 digits after decimal point

$$\begin{aligned}
&(1002 - 1001) + 0.03 \\
&= (1.002 * 10^3 - 1.001 * 10^3) + 0.03 \\
&= 1.000 + 3.000 * 10^{-2} \\
&= 1.030 * 10^0
\end{aligned}$$

$$\begin{aligned}
&1002 - (1001 - 0.003) \\
&= 1.002 * 10^3 - (1.001 * 10^3 + 3.000 * 10^{-2}) \\
&1001 - 0.03 = 1000.97 \\
&1.00097 * 10^3 \\
&1.001 * 10^3 \\
&= 1.002 * 10^3 - 1.001 * 10^3 = 1.000 * 10^0
\end{aligned}$$

Computers can represent rational numbers exactly as a fraction of two integers (of arbitrary length), but these integers will grow larger than 64 bits rapidly

Example 18. $\frac{a}{b} + \frac{c}{d} = \frac{ad+bc}{bd}$

It is possible to represent int larger than 64 bits by stringing multiple 64-bits, but it is typically 10-1000x slower than 64-bit integers.
Note 3: rigorous study of floating-point error is possible (but difficult)

7.1 Assumptions in the simplex method

(Not always true, but most of the time)
For every floating-point number $\tilde{a}$) computed as an approximation of an exact number $\bar{a}$, we assume $\bar{a} \in [\tilde{a} - \epsilon, \tilde{a} + \epsilon]$ with $\epsilon = 10^{-6}$ (tolerance)
Feasibility

$$\begin{aligned}
\bar{x}_j \geq l_j &\rightarrow \tilde{x}_j \geq l_j - \epsilon \\
\bar{x}_j \leq u_j &\rightarrow \tilde{x}_j \leq u_j + \epsilon \\
\bar{c}_j \geq 0 &\rightarrow \tilde{c}_j \geq -\epsilon \\
\bar{c}_j \leq 0 &\rightarrow \tilde{c}_j \leq epsilon
\end{aligned}$$

Ratio test

$$\begin{aligned}
\bar{a}_{ij} > 0 &\rightarrow \tilde{a}_{ij} > +\epsilon \\
\bar{a}_{ij} < 0 &\rightarrow \tilde{a}_{ij} < -\epsilon
\end{aligned}$$

Note: in the dual simplex method, if the ratio test becomes $argmin\{\}$, then the problem is infeasible (dual unbounded)

7.2 Integer Programming

$$
\begin{aligned}
\text{IP:}\min\ & c^T x \\
Ax &= b \\
x &\in \mathbb{Z}^n_+
\end{aligned}
$$

Computational complexity: The decision version of IP is NP-complete.

Proof. BIP is NP-complete, and there is a polynomial reduction from BIP to IP. (trivial: BIP is a special case of IP) □

Same with knapsack
How to solve (IP)?
Let's consider

$$
\begin{aligned}
\text{IP:}\min\ & c^T x \\
Ax &= b \\
x &\in \mathbb{R}^n_+
\end{aligned}
$$

Every feasible solution to IP is feasible for LP, hence (LP) is called is a relaxation of IP. (LP relaxation). But (LP) has more feasible solution: where solution where some $x_j \notin \mathbb{Z}$

8 February 26, 2018

8.1 Integer programming (cont'd)

$$
\begin{aligned}
\min\ & c^T x \\
s.t. Ax &= b \\
x &\in \mathbb{R}^n_+
\end{aligned}
$$

Every feasible solution of IP is also feasible for LP. If we minimize, $z^{LP} \le z^{IP}$.
Divide and conquer:

Let LP be the LP-relaxation of IP
if LP is infeasible **then return** empty
end if
Let x^{LP} be an optimal solution to LP
if $x^{LP} \in \mathbb{Z}^n$ **then return** $\{x^{LP}\}$
else
    let j be such that $x_j^{LP} \notin \mathbb{Z}$
    $IP_0 = minc^T x, Ax = b, x_j \le \lfloor Lx_J^{LP} \rfloor$
    $IP_1 = minc^T x, Ax = b, x_j \le \lceil Lx_J^{LP} \rceil$
    $x^0 = \text{SOLVE}(IP_0)$

$x^1 = \textsc{Solve}(IP_1)$
return best of x^0, x^1
end if

Observation: Recall that if (LP) is the LP-relaxation of IP, then $z^{IP} \geq z^{LP}$. Assume that for some node of the tree, we have an LP objective. $\hat{z}$, and $z^{LP}_{(node)} \geq \hat{z}$. Thus discard that node("pruning")
Note: $\hat{x}$, the solution with value $\hat{z}$ is called the incumbent solution the best IP solution found so far.

8.2 Branch and bound

Global variable: $\hat{z} = +\infty$
function $\textsc{Solve}$(IP)
  Consider the LP-relaxation (LP) of (IP)
  if (LP) is infeasible **then return** $\{\emptyset\}$
  end if
  Let x^{lp} be an optimal solution of L with $c^T x^{LP} = z^{LP}$
  if $x^{LP} \in \mathbb{Z}$ **then**
    if $z^{LP} < \hat{z}$ **then**
      $\hat{z} := z^{LP}$
    else
      return $\{\emptyset\}$
    end if
  end if
  Let j: $x_j^{LP} \notin \mathbb{Z}$
  $x^0 = \textsc{Solve}(IP_0)$
  $x^1 = \textsc{Solve}(IP_1)$
  return best of x^0, x^1
end function

8.3 February 28, 2018

Let $S = \{x \in \mathbb{Z}^n_+ : Ax = b\}$. There are infinitely many $A' \in \mathbb{R}^{m \times n}, b' \in \mathbb{R}^m$ such that $S = \{x \in \mathbb{Z}^n_+ : A'x = b'\}$.

Definition 13. Let $P = \{x \in \mathbb{R}^n_+ : Ax = b\}, P$ is a formulation of S if $P \cap \mathbb{Z}^n = S$.

Definition 14. Let P_1, P_2 be two polyhedra such that $S = P_1 \cap \mathbb{Z}^n$, $S = P_2 \cap \mathbb{Z}^N$, P_1 is stronger formulation than P_2 if $P_1 \subset P_2$,

Stronger formulation are better because they can elad to small *b&b* tree. The strongest formulation formulation for S is $conv(S)$

Theorem 8. *Let $P = \{x \in \mathbb{R}^n_+ : Ax = b\}$ where $A \in \mathbb{Q}^{m \times n}$, $b \in \mathbb{Q}^m$, and $S = P \cap \mathbb{Z}^n$*

1. $conv(S)$ is a polyhedron

2. $conv(S) \subseteq P$

3. every vertex of $conv(S)$ is integer.

Observation: Let $S = \{x \in \mathbb{Z}^n_+ : Ax = b\}$. Solving $minc^T x$ such that $x \in S$ (IP) is NP-hard.
Let $P = conv(S)$, solving $minc^T$ such that $x \in P$ (LP) is in P and it gives a solution to IP. However, computing $conv(S)$ has exponential complexity (as far as we know), actually $conv(S)$ has exponentially many constraints (in the worst case)

8.4 Cutting Plane

Definition 15. Let P be any set in $\mathbb{R}^n$. An inequality $\alpha^T x \geq \beta$ is valid for P if $\alpha^T x \geq \beta$ for every $x \in P$.

Definition 16. Let $P = \{x \in \mathbb{R}^n : Ax = b\}$. A constraint $\alpha^T x = \beta$ is implied by $Ax = b$ if $\alpha = \mu^T A$,$\beta = \mu^T b$ for some $\mu \in \mathbb{R}^m$ for some $\mu \in \mathbb{R}$

Definition 17. Let $P = \{x \in \mathbb{R}^n : Ax \leq b\}$. A constraint $\alpha^T x \leq \beta$ is implied by $Ax \leq b$ if $\alpha = \mu^T A$,$B \leq \mu^T b$ for for some $\mu \in \mathbb{R}^m$ and $\mu \geq 0$

Definition 18. Let P be a polyhedron and $S = P \cap \mathbb{Z}^n$. A cutting plane is constraint that is valid for S but not for valid for P

Example 19. $P = \{x \in \mathbb{R}^5 : 3x_1 - 4x_2 + 2x_3 - 3x_4 + x_5 \leq -2; 0 \leq x_1, ..., x_5\}$. Set $x_2 = x_4 = 0$. The constraint becomes $x_1 + 2x_3 + x_5 \leq -2$. $x_2 = x_4 = 0$ is impossible!. Either x_2 or x_4 are 1. $x_1 + x_4 \geq 1$ is valid for S.

is $x_2 + x_4 \geq 1$ valid for P. (is it implied). Observe that $x = (0, \frac{1}{2}, 0, 0, 0) \in P$. But $x_2 + x_4 = \frac{1}{2} \not\geq 1$ So $x_2 + x_4$ is a cutting plane

9 March 2, 2018

9.1 Cutting plane(continued)

Let $P \subseteq \mathbb{R}^n$ be polyhedron and $S = P \cap \mathbb{Z}^n$. Assume that

$$
\begin{aligned}
2x_1 = 1.3x_2 &\leq 5.8 && (4)\\
2x_1 = \lfloor 1.3 \rfloor x_2 &\leq 5.8 && (5)\\
2x_1 = 1x_2 &\leq 5.8 && (6)\\
2x_1 = 1x_2 &\leq 5 && (7)\\
& && (8)
\end{aligned}
$$

Since the left-hand side is always integer.
Remark 1: (2) is "weaker" than (1), but (3) is "stronger" than (2) so it is (potentially) a cutting plane.

Definition 19. An inequality $\alpha^T x \leq \beta$ dominates another inequality $\alpha^T x \leq b$, if $\{x \in \mathbb{R}^n, : \alpha^T x \leq \beta\} \subset \{x \in \mathbb{R}^n : ax \leq b\}$.

Remark 2: We can divide (3) by 2, then

$$x_1 + \frac{1}{2}x_2 \leq \frac{5}{2}$$
$$x_1 \leq 2$$

Where did (1) come from?
(1) can be the constraint from the formulation.
(1) can wbe any valid combination of such constraints
(1) can come from a previous cut
Chvatal-Gomor cuts: Let $P = \{x \in \mathbb{R}^n_+ : Ax \leq b\}$, and let $S = P \cap \mathbb{Z}^n$.

1. For any $u \in \mathbb{R}^m_+$, let $d^T = u^T A$, and, $h = u^T b$, Then, $d^T x \leq h$ is valid for P, i.e. $\sum_j d_j x_j \leq h$.
2. Thus, $\sum_j \lfloor d_j \rfloor x_j \leq h$. is also valid for P
3. Since $\sum_j \lfloor d_j \rfloor x_j \leq \lfloor h \rfloor$ is valid for S.

Remark: If P has equality constraints $P = \{x \in \mathbb{R}^n_+ : Ax = b\}$, then $P = \{x \in \mathbb{R}^n_+ : Ax \leq b, Ax \geq b\}$ then $P = \{x \in \mathbb{R}^n_+ : Ax \leq b, -Ax \leq -b\}$

Theorem 9. *Every valid inequality for S can be derived by applying the Chvatal-Gomory procedure a finite number (note: finite, but exponential in the worst case, since you get $conv(S)$)*

9.2 Mixed-integer rounding(MIR)

$$S^1 = \{y \in \mathbb{Z}, x \in \mathbb{R}^+ : y \leq b + x\}$$
$$\text{cut: } y \leq \lfloor b \rfloor + \frac{x}{1 - f_b}$$
$$f_b = b - \lfloor b \rfloor$$

MIR with 2 integer variables
Attempt 1:

$$\begin{aligned}
&S^2 = \{y_1, y_2 \in \mathbb{Z}_+, z \in \mathbb{R}_+ : y_1 + a_2 y_2 \leq b + z\} \\
&y_1 + \lfloor a_2 \rfloor y_2 + f_2 y_2 \leq b + z \\
&y_1 + \lfloor a_2 \rfloor y_2 \leq b + z && f_2 y_2 \geq 0 \\
&\text{cut: } y_1 + \lfloor a_2 \rfloor y_2 \leq \lfloor b \rfloor + \frac{z}{1 - f_b}
\end{aligned}$$

Attempt 2:

$$y_1 + \lfloor a_2 \rfloor y_2 \leq b + z - f_2 y_2$$

It's a failed attempt, there exists z, y_2, such that $x = z - f_2 y_2 \ngeq 0$.
Attempt 3:

$$\begin{aligned}
y_1 + \lceil a_2 \rceil y_2 - (1 - f_2) y_2 &\leq b + z \\
y_1 + \lceil a_2 \rceil y_2 &\leq b + z + (1 - f_2) y_2 \\
\text{cut}: y_1 + \lceil a_2 \rceil y_2 &\leq \lfloor b \rfloor + \frac{1}{1 - f_b}(z + (1 - f_2) y_2) \\
y_1 + (\lceil a_2 \rceil - \frac{1 - f_2}{1 - f_b}) y_2 &\leq \lfloor b \rfloor + \frac{z}{1 - f_b}
\end{aligned}$$

Assume that $f_2 > 0$

$$\begin{aligned}
y_1 + (\lfloor a_2 \rfloor + 1 - \frac{1 - f_2}{1 - f_b}) y_2 &\leq \lfloor b \rfloor + \frac{z}{1 - f_b} \\
y_1 + (\lfloor a_2 \rfloor + \frac{f_2 - f_b}{1 - f_b}) y_2 &\leq \lfloor b \rfloor + \frac{z}{1 - f_b}
\end{aligned}$$

Note: if $f_2 > f_b$, then Attempt 3 cut is stronger than Attempt 1 cut
if $f_2 < f_b$, then Attempt 3 cut is weaker than Attempt 1 cut

9.3 MIR with many integer and continuous variables

$$S^n = \{y_B \in \mathbb{Z}, y \in \mathbb{Z}^{|I|}, x \in \mathbb{R}_+^{|C|} : y_B + \sum_{j \in I} a_j y_j + \sum_{j \in C} a_j x_j = b\}$$

1. Dealing with the continuous variables

$$\begin{aligned}
y_B + \sum_{j \in I} a_j y_j + \sum_{j \in C, a_j > 0} a_j x_j + \sum_{j \in C, a_j > 0} a_j x_j &= b \\
y_B + \sum_{j \in I} a_j y_j &\leq b + \sum_{j \in C, a_j < 0} (-a_j) x_j
\end{aligned}$$

2. Dealing with integer variables

$$
\begin{gathered}
y_B + \sum_{j \in I, f_j \le f_b} \lfloor a_j \rfloor y_j + \sum_{j \in I, f_j \le f_b} f_j y_j + \\
\sum_{j \in I, f_j > f_b} \lceil a_j \rceil y_j - \sum_{j \in I, f_j > f_b} (1 - f_j) y_j \\
\le b + \sum_{j \in C, a_j < 0} (-a_j) x_j \\
\text{Cut: } y_B + \sum_{j \in I, f_j \le f_b} \lfloor a_j \rfloor y_j + \sum_{j \in I, f_j > f_b} \lceil a_j \rceil y_j \\
\le \lfloor b \rfloor + \frac{1}{1 - f_b} \Big(\sum_{j \in I, f_j > f_b} (1 - f_j) y_j + \sum_{j \in C, a_j < 0} (-a_j) x_j \Big) \\
y_B + \sum_{j \in I, f_j < f_b} \lfloor a_j \rfloor y_j + \sum_{j \in I, f_j > f_b} \Big(\lceil a_j \rceil + \frac{f_j - 1}{1 - f_b} \Big) y_j \\
+ \sum_{j \in C, a_j < 0} \frac{a_j}{1 - f_b} x_j \le b
\end{gathered}
$$

From the original equality constraint and divide by f_b (assuming > 0)

$$
\sum_{j \in I, f_j \le f_b} \frac{f_j}{f_b} y_j + \sum_{j \in I, f_j > f_b} \frac{1 - f_j}{1 - f_b} y_j + \sum_{j \in C, a_j \ge 0} \frac{a_j}{f_b} x_j - \sum_{j \in C, a_j < 0} \frac{a_j}{1 - f_b} x_j \ge 1
$$

9.4 Gomory's mixed-integer cut(GMi)

Given a row of simplex tableau

$$
\begin{gathered}
x_{B_i} + \sum_{j \in I} a_j x_J + \sum_{j \in C} a_j x_j = b \\
x_{B_i} \in \mathbb{Z}_+ \\
x_j \in \mathbb{Z}_+ \forall j \in I \\
x_j \in \mathbb{R}_+ \forall j \in C \\
b \notin \mathbb{Z}
\end{gathered}
$$

the corresponding GMi cut is

$$
\sum_{j \in I, f_j \le f_b} \frac{f_j}{f_b} x_j + \sum_{j \in I, f_j > f_b} \frac{1 - f_j}{1 - f_b} x_j + \sum_{j \in C, a_j \ge 0} \frac{a_j}{f_b} x_j - \sum_{j \in C, a_j < 0} \frac{a_j}{1 - f_b} x_j \ge 1
$$

Observation 1: IN the basic solution $\bar{x}$ associated to the tableau $\bar{x}_{B_i} = b, \bar{x}_j = 0, \forall j \in I \cup C$. If we evaluate the GMi cut at $\bar{x}$, we get $0 \ge 1$. Thus if the basis is feasible, the GMi inequality cuts off at least one vertex.

9.5 Split Cuts

Let $P \subseteq \mathbb{R}^n$ be a polyhedron, let $I \subseteq \{1, ..., n\}, C = \{1, ..n\} \setminus I$, let $S = \{x \in P : x_j \in \mathbb{Z} \forall j \in I\}$ Consider any $\pi \in \mathbb{Z}^n$: $\pi_j = 0, \forall j \in C$. For every $x \in S, \pi^T x \in \mathbb{Z}$. In particular, for every $f \in \mathbb{Z}$ either $\pi^T x \leq f$ or $\pi^T x \geq f + 1$. This is called a split disunction.

Definition 20. Let

$$\pi_1 = \{x \in P : \pi^T x \leq f\}$$
$$\pi_2 = \{x \in P : \pi^T x \geq f + 1\}$$

We define $P^{(\pi,f)} = conv(\pi_1, \pi_2)$

Observation 1: $S \subset \pi_1 \cup \pi_2$, so $S \subseteq P^{(\pi,f)}$
Observation 2: $P^{(\pi,f)}$ is a polyhedron
Observation 3: In the example, $P^{(\pi,f)} = P \cap \{\alpha^T x \leq \beta\}$ for some α, β. This is not always the case (we sometimes need zero or ≥ 2 additional inequalities)

Definition 21. A split cut for P is an inequality that is valid for $P^{(\pi,f)}$ for some $\pi \in \mathbb{Z}^n : \pi_j = 0 \forall j \in C, f \in \mathbb{Z}$

Definition 22. The split closure P' of P is

$$P' = \cap_{\pi,f} P^{(\pi,f)}$$

If P is defined by rational constraints, then P' is a polyhedron.

Theorem 10. *If P is defined by rational constraints, then P' is a polyhedron.*

Theorem 11. *Every undominated split cut can be obtained as a GMi cut from an integer combination of the row of some tableau of P*

Observation[C, K, S]: Chvatal-Gomory Cut are split cuts for which either $pi_1 =$ or $\pi_2 =$.
Corollory: Chvatal-Gomory cuts are a subset of split cuts.

10 0-1 knapsack cover inequalities

Consider the knapsack $S = \{x \in \{0,1\}^n : \sum_j a_j x_j \leq b\}$, where $a_j > 0, \forall j$.
Note: if $a_j < 0$, let $x'_j = 1 - x_j, a'_j = -a_j, b' = b - a_j$

Definition 23. A set $C \subseteq \{1, ..., n\}$ is a cover for S if $\sum_{j \in C} a_j > b$. It is minimal if $C \setminus \{j\}$ is not a cover, for all $j \in C$.

Proposition 4. *If C is a cover for S, the inequality, $\sum_{j \in C} x_j \leq |C| - 1$ is valid for S*

Proof. If $\sum_{j\in C} x_j = |C|$, then we take all objects in C which is impossible since $\sum_{j\in C} a_j > b$. $\square$

Example 20. $S = \{x \in \{0,1\}^7 : 11x_1+6x_2+6x_3+5x_4+5x_5+4x_6+1x_7 \leq 19\}$. Some cover inequalites:

$$\begin{aligned}
C = \{1,2,3\} &\to x_1 + x_2 + x_3 \leq 2 \\
C = \{1,2,6\} &\to x_1 + x_2 \leq 2 \\
C = \{1,5,6\} &\to x_1 \leq 2 \\
C = \{1,2,5,6\} &\to x_1 + x_2 + x_5 + x_6 \leq 3 \\
C = \{3,4,5,6\} &\to x_3 + x_4 + x_5 + x_6 \leq 3
\end{aligned}$$

Question: Dominated?
Cover inequalities from non-minimal covers are dominated by those from minimal cover

Proposition 5. *If C is a cover for S, then the extended cover inequality*

$$\sum_{j\in E(C)} x_j \leq |C| - 1$$

is valid for S where $E(C) = C \cup \{j : a_j \geq a_i \forall i \in C\}$

Remark: $E(C) = C \cup \{j : a_j \geq a^{max}\}$ where $a^{max} = max_{j\in C} a_j$

Proof. Let $R = \{j \in E(C) : x_j = 1\}$. If $\sum_{j\in E(C)} x_j \geq |C|$, then $|R| \geq |C|, |R_C| + |R_E| \geq |C|$, where $R_C = R \cap C, R_E R \setminus R_C$, so $|R_E| \geq |C| - |R_C|$.

$$\begin{aligned}
\sum_{j\in E(C)} a_j x_j &= \sum_{j\in R} a_j \\
&= \sum_{j\in R_c} a_j + \sum_{j\in R_E} a_j \\
&\geq \sum_{j\in R_c} a_j + |R_E| a^{max} \\
&\geq \sum_{j\in R_C} a_j + (|C| + |R_C|) a^{max} \\
&= \sum_{j\in R_C} a_j + \sum_{j\in C\setminus R_c} a^{max} \\
&\geq \sum_{j\in C} a_j > b
\end{aligned}$$

$\square$

Example 21. Let $C = \{3,4,5,6\}$, then $E(C) = C \cup \{j : a_j \geq 6\}$
$E(C) = C \cup \{1,2\}$ So $x_1 + x_2 + x_3 + x_4 + x_5 + x_5 + x_6 \leq 3$ is valid Note: $2x_1 + x_2 + x_3 + x_4 + x_5 + x_5 + x_6 \leq 3$ is valid be proven to be valid for S

11 Strengthening cover inequalities

Proposition 6. *Given two inequalites $\alpha^T \leq \beta$ and $\gamma^T \leq \beta$. valid for $S \subseteq \mathbb{R}^n_+$,*

1. *If $\alpha \geq \gamma$, then $\alpha^T x \leq \beta$ implies $\gamma^T x \leq \beta$*
2. *In addition, if $\alpha \neq \gamma$, then $\alpha^T x \leq \beta$ dominates $\gamma^T x \leq \beta$.*

Proof. 1. If $\forall x \in S : \alpha^T x \leq \beta$, then $\gamma^T x \leq \alpha^T x \leq \beta$,

2. If $\alpha = \gamma$. the cuts are the same

□

Example 22.

$$S = \{x \in \{0,1\}^7 : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + 1x_7 \leq 19\}$$

Start from the cover inequality $x_3 + x_4 + x_5 + x_6 \leq 3$ which is valid for:

$$S^4 = \{x \in \{0,1\}^4 : 6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 19\}$$

Let's find the values α_1 for which $\alpha_1 x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$ is valid for:

$$S^5 = \{x \in \{0,1\}^5 : 11x_1 + 6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 19\}$$

If $x_1 = 0$, always valid whatever α_1.
If $x_1 = 1$, it is valid

- iff $\alpha_1 + x_3 + x_4 + x_5 + x_6 \leq 3$ is valid for

$$\{x \in \{0,1\}^4 : 11 * 1 + 6x_3 + 5x_4 + 5x_5 + 5x_5 + 4x_6 \leq 19\}$$
$$\{x \in \{0,1\}^4 : 6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 8\}$$

- iff $x_3 + x_4 + x_5 + x_6 \leq 3 - \alpha_1$ is valid for

$$\{x \in \{0,1\}^4 : 6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 8\}$$

- iff $x_3 + x_4 + x_5 + x_6 \leq 1 \leq 3 - \alpha_1$, i.e. $\alpha_1 \leq 2$.

So we get $2x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$.
Apply the same for x_2 (S^6) and x_7 ($S^7 = S$), we get:

$$\alpha_2 = 1$$
$$\alpha_7 = 0$$
$$2x_2 + x_2 + x_3 + x_4 + x_5 + x_6 + 0x_7 \leq 3$$
$$2x_2 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$$

11.1 Lifting procedure

Start from $\sum_{i=0}^{t} \alpha_j x_j \leq \beta$ is valid for $\{x \in \{0,1\}^t : \sum_{j=0}^{t} a_j x_j \leq b\}$.
Compute $z_{t+1} = max \sum_{j=1}^{t} \alpha_j x_j$ s.t. $\sum_{j=0}^{t} a_j x_j \leq b - a_{t+1}$, $x \in \{0,1\}^t$
Set $\alpha_{t+1} := \beta - z_{t+1}$
Set $t := t + 1$
Repeat

Remark 1.

$$\alpha_{t+1} \leq \beta - z_{t+1} \leq \beta - \sum_{j=1}^{t} \alpha_j x_j$$

We find the least upper bound of $\{x : \beta - \sum_{j=1}^{t} \alpha_j x_j\}$ to find the maximum α_{t+1} can take. β remain unchanged through iterations.

Separating Cover inequalites: Given x^* fractional, find a violated cover.
Observation:

$$\sum_{j \in C} x_j \leq |C| - 1$$

is equivalent to

$$\sum_{j \in C} (1 - x_j) \geq 1$$

Thus let us find C that is violated by x^*

$$min_{C \subseteq \{1,\ldots,n\}} \{\sum_{j \in C} (1 - x_j^*)\} < 1$$

Then $\forall j \in \{1, ..., n\}$,

$$z_j = \begin{cases} 0 & \text{if } j \notin C \\ 1 & \text{if } j \in C \end{cases}$$

Find

$$\begin{aligned} \min \ & \sum_j (1 - x_j^*) z_j \\ \text{s.t.} \ & \sum_j a_j z_j > b \end{aligned}$$

If the minimum is < 1, then z gives a violated cover.

Example 23. S same as before

$$\begin{aligned} \max\ & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \\ \text{i.e. } & 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + 1x_7 \le 19 \\ & x \in \mathbb{R}^7_+ \end{aligned}$$

Solving this LP gives us, $x^* = [00\frac{2}{3}1111]^T$, let's find the cover

$$\begin{gathered} \min 1z_1 + 1z_2 + \frac{1}{3}z_3 \\ 11z_1 + 6z_2 + 6z_3 + 5z_4 + 5z_5 + 4z_6 + 1z_7 > 19 \\ z \in \mathbb{B}^7 \end{gathered}$$

To solve this LP, we can convert it to knapsack problem.

$$\begin{gathered} y_j = 1 - z_j \\ \max \frac{2}{3}y_3 + y_4 + y_5 + y_6 + y_7 \\ \text{s.t. } 11y_1 + 6y_2 + 6y_3 + 5y_4 + 5y_5 + 4y_6 + 1y_7 \le 18 \end{gathered}$$

Solve with DP, it yields $y* = [1100000]^T$, $z* = [0011111]$, Cover $= \{3, 4, 5, 6, 7\}$

Remark 2. $\max x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$ finds the maximum item, the knapsack can have. Construct $S = \{j : x^*_j > 0\}$. Then we want to find a cover with most element of S.

12 Column Generation

We are given an infinite supply of "stocks" of a fixed size. We want to cut them into a given number of "pieces" of given sizes. Arrange the cuts so as to minimize the number of stocks required.
Consider all possible patterns, i.e., all possible ways to cut a single stack. Let $Q_p = [q_{1p}...q_{mp}]^T$. where we cut q_{ip} pieces of type i. for all possible patterns, $p = 1, 2, ...$
Data:

- a_i length of the pieces type
- b the length of a stock
- d_i the demands for each type piece

Remark: For all p, $\sum_{i=0}^{m} a_i q_{ip} \le b$.
Let $x_p \in \mathbb{Z}_+$ be the number of times which we take pattern p.

Formulation:

$$\min \sum_p x_p$$
$$\sum_p Q_p x_p \geq d$$
$$x_p \geq 0$$
$$x_p \in \mathbb{Z}, \forall p$$

Remark 3. We construct Q_p based on data a_i and, b; d_i goes into the LP

12.1 Column generation approach

Choose a subset of the variables (columns): Implicitly, we are fixing all other variables to zero (nonbasic)

Example 24. Solve $b = 17$, $a = \begin{bmatrix} 3 & 5 & 9 \end{bmatrix}^T$, $d = \begin{bmatrix} 25 & 20 & 15 \end{bmatrix}^T$
Contructing Q_p, we get

$$Q_1 = \begin{bmatrix} 5 & 0 & 0 \end{bmatrix}^T$$
$$Q_2 = \begin{bmatrix} 0 & 3 & 0 \end{bmatrix}^T$$
$$Q_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$
$$Q_4 = \begin{bmatrix} 4 & 1 & 0 \end{bmatrix}^T$$
$$Q_5 = \begin{bmatrix} 2 & 2 & 0 \end{bmatrix}^T$$
$$\dots$$

There are too many distinct patterns, thus we choose a subset of them, usually the simplest ones Q_1, Q_2 and Q_3. This gives the following LP:

$$\min x_1 + x_2 + x_3$$
$$\text{s.t. } Q_1 x_1 + Q_2 x_2 + Q_3 x_3 \geq d$$

Optimal primal solution: $x^* = \begin{bmatrix} 5 & \frac{20}{3} & 15 \end{bmatrix}$
Optimal primal solution: $y^* = \begin{bmatrix} \frac{1}{5} & \frac{1}{3} & 1 \end{bmatrix}$
Recall: $y^{*T} = c_B B^{-1}$ and $\bar{c} = c - c_B B^{-1} A = 1 - y^{*T} Q_p$.
If all $\bar{c} > 0$, then we know x^* is optimal
If $\bar{c}_p < 0$ for some p, x_p needs to enter the basis, add column Q_p, repeat.
How to find $p : \bar{c}_p < 0$? This is equivalent formulating as

$$\min 1 - y^* q$$
$$\text{s.t. } \sum_{i=1}^{m} a_i q_i \leq b$$
$$q \in \mathbb{Z}_+^m$$

$$
\begin{aligned}
1 - \max\ & y^* q \\
\text{s.t. } & \sum_{i=1}^{m} a_i q_i \leq b \\
& q \in \mathbb{Z}_+^m
\end{aligned}
$$

Observe this problem is a knapsack problem.

$$
\begin{aligned}
1 - \max\ & \frac{1}{5} q_1 + \frac{1}{3} q_2 + 1 q_3 \\
\text{s.t. } & \sum_{i=1}^{m} 3 q_i + 5 q_2 + 9 q_3 \leq 17 \\
& q \in \mathbb{Z}_+^m
\end{aligned}
$$

Solving this gives: $\bar{c}_p = -\frac{8}{15} < 0$, and $q^* = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$, add that column to the LP. gives

$$
\begin{aligned}
\min\ & x_1 + x_2 + x_3 + x_4 \\
\text{s.t. } & Q_1 x_1 + Q_2 x_2 + Q_3 x_3 + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} x_4 \geq d
\end{aligned}
$$

Repeat the process until we reach optimal such that $\bar{c} > 0$. The optimal solution x^* may not be integer, then we have to perform branch and bound. In the example $x^* = \begin{bmatrix} 0 & \frac{5}{6} & 0 & 15 & \frac{5}{2} \end{bmatrix}^T$ is optimal.

13 March 19, 2018

Observation 1: Let $P \subseteq \mathbb{R}^n$, $\min\ \{c^T x : x \in P\} \leq \min\ \{c^T x : x \in P \cap \mathbb{Z}^n\}$. Hence $z^{IP} \geq z^{LP}$, since all coefficients of the objective function are integer in the cutting-stock problem, $z^{IP} \geq \lceil z^{LP} \rceil$
Observation 2: Let $P = \{x \in \mathbb{R}^n : Ax \geq b\}$ Since $A_{ij} \geq 0$ for all i, j, then, if $\bar{x} \in P$, then for all $x' \geq \bar{x}, x' \in P$

13.1 Approximation algorithms

For NP-hard problems, we don't know how to find an optimal solution in polynomial time - it is even impossible if $NP \neq P$
Approches:

- Exact: Find an optimal solution (e.g. in exponential time)
- Heuristic: Find any feasible solution in polynomial time without any guarantee on the objective function value
- Approximation algorithm: Find a feasible solution in polynomial time, with a certifed bound on how close it is to optimal.

Definition 24. Let $\alpha \geq 1$, Q be a minimization problem. An α-approximation algorithm computes, in polynomial time for every instance of Q, a feasible solution of value at most of x times the value of an optimal solution of value at most α times the value of an optimal solution

Note: For max problems, $\alpha \leq 1$
Typical values for $\alpha : 2$, $O(1)$, $\log n$, $1 + \epsilon$

Example 25. Given a connected graph $G(V, E)$, find a node subset $C \subseteq V$ of minimum such that every edge in the graph has at least one end in C.

$$min\{|C| : C \subseteq V, \forall uv \in E, \{u, v\} \cap E \neq \emptyset\}$$

Vertex Cover is NP-hard. Example exact algorithm:

1. enumerate all $2^{|v|}$ node subsets
2. check whether they cover all edges
3. select subset of smallest cardinality

α-approximation algorithm:
Let C^* is an optimal solution, find C such that $|C| \leq \alpha|C^*|$
IDEA: We don't know C^*, but we might be able to copute a good lower bound $|C^*|$, i.e. $|C^*| \geq$ ... We could compute any matching M (set of edges that are not pairwise adjacent)
Algorithm:

```
1: Compute a maximal matching
2: Output: C = ∪_{uv∈M} {u, v}
```

Theorem 12. *The above algorithm is a 2-approximation for vertex cover*

14 March 26, 2018

Theorem 13. *The above algorithm is a 2-approximation for vertex cover*

Proof.

1. C is feasible, if it was infeasible, $\exists uv \in E, u \notin C, v \notin C$. But then $M \cup \{uv\}$ is a matching too, which contradicts M is a maximal matching
2. $|C| = 2|M| \leq 2|C^*|$

□

Example 26 (Weighted Verted cover)**.** Given a a connected graph $G(E, V)$, find a node subset $C \subset |V|$ of minimum weight such that every edge in the graph has at least one end in C.

$$min\{\sum_{u \in C} w_u : C \subseteq V, \forall uw \in E, \{v, w\} \cap C \neq\}$$

The previous strategy can arbitrarily bad.

We use an integer programming formulation:

$$\text{var: } x_v = \begin{cases} 1 & \text{if } v \in C \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} \min \sum_{v \in V} w_v x_v \\ \text{s.t. } x_u + x_v \geq 1 : \forall uv \in E \\ 0 \leq x_v : \forall v \in V \end{aligned}$$

IP is NP-hard, but we can solve the LP-relaxation in poly-time. (if it has polynomial size in $|V|$ and $|E|$ which is the case here). The LP-relaxation give us a lower bound on "opt" (the objective function value of an optimal solution). Algorithm:

1: Solve the LP-relaxation, let x^* be an optimal solution.
2: Output: $C = \{v \in V : x_v^* \geq \frac{1}{2}\}$.

Theorem 14. *The above algorithm is a 2-approximation*

Proof. 1. C is feasible, $\forall uv \in E, x_u^* + x_v^* \geq 1$, either $x_u^* \geq \frac{1}{2}$ or $x_v^* \geq \frac{1}{2}$ (or both). Thus either $u \in C$ or $v \in C$ (or both)

2.
$$\begin{aligned} \sum_{v \in C} w_v &= \sum_{v \in C : x_v^* \geq \frac{1}{2}} w_v \lceil x_v^* \rceil + \sum_{v \in C : x^* < \frac{1}{2}} w_v \lfloor x_v^* \rfloor \\ &\leq 2 \sum_{v \in C : x_v^* \geq \frac{1}{2}} w_v x_v^* + 2 \sum_{v \in C : x^* < \frac{1}{2}} w_v x_v^* \\ &\leq 2 \sum_{v \in V} w_v x_v^* \\ &\leq 2 \text{ opt} \end{aligned}$$

□

The weighted VC, LP is half integral.
Remainder: A feasible solution is a vertex iff it cannot be written as a convex combination of two other feasible solution.

Theorem 15. *If x^* is a vertex of the Weighted Vertex Cover Linear Program, then $\forall v \in V : x_v^* \in \{0, \frac{1}{2}, 1\}$*

Proof. Let $V_+ = \{v \in V : \frac{1}{2} < x_v^* < 1\}$, and $V_- = \{v \in V : 0 < x_v^* < \frac{1}{2}\}$
Define

$$y_v = \begin{cases} x_v^* + \epsilon & v \in V_+ \\ x_v^* - \epsilon & v \in V_- \\ x_v^*, & \text{otherwise} \end{cases} \quad z_v = \begin{cases} x_v^* - \epsilon & v \in V_+ \\ x_v^* + \epsilon & v \in V_- \\ x_v^* & \text{otherwise} \end{cases}$$

Then $x^* = \frac{1}{2}(y+z)$ Furthermore, y and z are feasible. Then $x_v^* + x_u^* = 1$,

- $x_u^* = x_v^* = \frac{1}{2}$, then it's ok
- $x_u^* > \frac{1}{2}, x_v^* < \frac{1}{2}$, we can choose ϵ small enough such that $y_u, y_v, x_u, x_v \in [0,1]$, then x^* won't be an optimal solution.

Similarly if $x_u^* + x_v^* > 1$ $\square$

15 March 28, 2018

Example 27. Given a set of n elements $U := \{1, ..., n\}$ and a colletion of sets $S_1, ..., S_m \subset U$ with cost $c(S_i)$, find a subcollection of sets of minimum total cost, covering all elements.

$$opt = min_{I \subset \{1,...,m\}} \{\sum_{i \in I} c(S_I) : \cup_{i \in I} S_i = U\}$$

Greedy algorithm:

$C = \emptyset$
while $C \neq U$ **do**
  let $i = argmin\{\frac{c(S_i)}{|S_i \backslash C|}\}$
  Set $C := C \cup S_i$
end while

Theorem 16. *The greedy algorithm is a $O(\log n)$-approximation algorithm*

Theorem 17. *Given $a_1, ..., a_l > 0$, and $b_1, ..., b_l > 0$, then*

$$min_i \frac{a_i}{b_i} \leq \frac{\sum_i a_i}{\sum_i b_i}$$

Proof.

$$\begin{aligned} \sum_i a_i = \sum_i b_i \cdot \frac{a_i}{b_i} &\geq \sum_i (b_i \cdot min_j \frac{a_j}{b_j}) \\ &= min_j \frac{a_j}{b_j} \sum_i b_i \\ \frac{\sum_i a_i}{\sum_i b_i} &\geq min_j \frac{a_j}{b_j} \end{aligned}$$

$\square$

Proof. Let $\{e_1, ..., e_n\}$ give the element in U in the order they are added to C. Let $p(e_j) = min_i \frac{c(S_i)}{|S_i \backslash C|}$ for C before e_j was added to it. Observe that $cost(C) =$

$\sum_{j=1}^{n} p(e_j)$. Let $I^* \subset \{1, ..., m\}$ be an optimal solution then $opt = \sum_{i \in I^*} c(S_i)$

$$\begin{aligned}
p(e_j) &= min_{i=1,\dots,m}\{\frac{c(S_i)}{|S_i \setminus C|}\} \\
&\leq min_{i \in I^*}\{\frac{c(S_i)}{|S_i \setminus C|}\} \\
&\leq \frac{\sum_{i \in I*} c(S_i)}{\sum_{i \in I*} |S_i \setminus C|} \\
&\leq \frac{opt}{n-j+1} \\
cost(C)_{\text{approx}} &= \sum_{j=1}^{n} p(e_j) \\
&\leq \sum_{j=1}^{n} \frac{opt}{n-j+1} \\
&= opt \sum_{j'-1}^{n} \frac{1}{j'} && j' = n-j+1 \\
&= opt \cdot H_n \\
&\leq opt \cdot (\log_e n + 1) && \text{(Harmonic value)} \\
&\leq opt \cdot (\log n + 1) \\
&= O(\log n) \cdot opt
\end{aligned}$$

□

Is our analysis tight? Yes!