Linear Programming

17 October 2018 11:36

Standard Form:

- aim is to minimise
- equality constraints as linear functions
  - RHS are non-negative
- decision variables are non-negative

$z = c_1x_1 + \cdots + c_nx_n$

$a_{11}x_1 + \cdots + a_{1n}x_n = b_1$

$\cdots$

$a_{1m}x_1 + \cdots + a_{1m}x_n = b_m$

- $b_i \geq 0$, $find\ x_i$

Place $a_{ij} into\ a\ matric\ A$; $b, x, c\ into\ vectors$

LP - **minimise subject to:**

- $\boldsymbol{z = c^T x}$
- $\boldsymbol{Ax = b}$
- $\boldsymbol{x \geq 0}$

If Rank(A) is full, then only unique point to satisfy LP

- If this is not feasible, feasible set is empty

Decision Variables - Value taken by variables we control
Objective Function - Optimise goal which depends on controlled (decision) variables
Feasible Set - Set of values of Decision Variables that meet constraints

LP Model must be bounded
Can solve using **Simplex** algorithm to find optimal vertex

Standardising LPs:

- Max -> Min
  - $new\ obj = -\boldsymbol{f}(\boldsymbol{x})$
  - **change final sign**
- Negative RHS (b's)
  - **multiply constraint by -1**
  - $-b_i \geq 0$
  - **inequality flips**
- <= Inequality -> Equality
  - introduce new slack variables $s_m \geq 0$
  - $LHS \leq RHS \rightarrow LHS + S_i = RHS$
  - $\boldsymbol{Ax + s = b}$
- >= Inequality -> Equality
  - introduce new surplus variables $s_m \geq 0$
  - $LHS \leq RHS \rightarrow LHS - S_i = RHS$
  - $\boldsymbol{Ax - s = b}$
- Free Variables
  - 1st approach
    - **replace free variable $\boldsymbol{x_i}$ $\boldsymbol{with}$ $\boldsymbol{x_i^+ - x_i^-}$**
    - $x_i^+ \geq 0\ and\ x_i^- \geq 0$
    - not a 1-1 mapping
  - 2nd approach
    - **eliminate the free variable**
    - use any equality constraint involving the free var
      - substitute to remove it

LP Models

23 October 2018 14:44

Demand: must meet
Supply: Must not go over

Resource Allocation Models

- variables - how many resources each use requires
- constraints - resource availability

Blending Models

- similar to allocation - but focus on **combining resources**
- variables - how much to include of each resource
- constraints - limit on properties in the blend

Operations planning models

- variables - products, activities, processing facilities
- constraints - balance between input and output of the activities

Shift scheduling models

- variable - employees that start shift at certain date/time
- constraints - allocate workers to cover requirements

Time-phased models

- variables - returns (revenue) and state at given time
- constraints - balance based on a time in future

Flow Conservation:

- When using intermediate locations
  - Like a warehouse
- Flow into the intermediate has to be same as flow out of intermediate

Basic Feasible Solutions

17 October 2018 16:40

Basic Solutions:

- assume an LP in standard form
  - number of variables **n >=** number of equations m
  - rows of A are linearly independent: **rank(A)=m**
- select a subset of **m** columns from A that are **linearly independent**
  - **To test - row reduction - no free vars**
  - index set I - set of the indexes of these columns
  - **Basis** - **matrix B** - matrix of these columns
  - if x_i = 0 for i **not in** I, the **x** is a **basic solution (BS)** to Ax=b with respect to index set I
- **Feasible solution (FS)**
  - **x** satisfying both Ax=b and x >= 0
  - if also basic, then this becomes a basic feasible solution (**BFS**)
    - which are the vertices of the feasible set
    - optimal always achieved at a BFS
- columns in B are linearly independent
  - $B\boldsymbol{x}_B = b$ is invertible
  - $\boldsymbol{x}_B = B^{-1}b$ - unique solution
  - back to Ax=b, covering the columns not in $\boldsymbol{x_B}, to\ be\ 0\ in\ \boldsymbol{x}$
    - **x** is the **unique basic solution** to Ax=b with respect to I

Fundamental theorem of LP:

For an LP in standard form, with rank(A)=m <=n

- if there is a feasible solution, there must be a BFS
- if there is an optimal solution, there must be an optimal BFS

At most $\binom{n}{m} = \frac{n!}{m!(n-m)!}$ BFS's

- usually less as B(I) may be singular (non-invertible) and B may not be feasible

Basic Variables:

- $i \in I, \boldsymbol{x}_i is\ a\ BV$
- $i \notin I, \boldsymbol{x}_i is\ a\ NBV\ (non\ basic\ variable)(all = 0)$

Basic representations:

- express the objective function value (z) and each BV **as a linear function of the NBVs**
- $A\boldsymbol{x} = b, \qquad A\boldsymbol{x} = B\boldsymbol{x}_B + N\boldsymbol{x}_N$
- $z = c^T\boldsymbol{x} = c_B^T\boldsymbol{x}_B + c_N^T\boldsymbol{x}_N$

- since **B is invertible**, can rearrange to get

- $z - \left(c_N - N^T B^{-T} c_B\right)^T \boldsymbol{x}_N = c_B^T B^{-1} b$

- $\boldsymbol{x}_B + B^{-1}N\boldsymbol{x}_N = B^{-1}b$

  - $c_N - N^T B^{-T} c_B\ - reduced\ cost\ vector\ - sensitivity\ of\ z\ to\ NBV$
    - helps to determine if BFS is optimal, I ff r>=0

Simplex Algorithm

17 October 2018 16:40

Simplex Tableau:

BV	All Variables	RHS
z	$-\left(c_N - N^T B^{-T} c_B\right)^T$ (β's)	$c_B^T B^{-1} b$
$\boldsymbol{x}_B$	$(B^{-1}N)$	$B^{-1}b$

- top row - called **objective row**
- all $RHS \geq 0$, apart from objective row
- coefficients of NBV -> **negative** reduced costs

Pivoting:

- if vertex x is not optimal, a neighbouring vertex will be more optimal
- x_p leaves the basis, x_q enters the basis
  - row operations
- the pair (p,q) is called a pivot ($\neq 0$)

1. divide row p by pivot element y_{pq}
   a. relabel as row q
2. for other rows in I
   a. subtract original $row\ p\ * \frac{y_{iq}}{y_{pq}}$
3. objective row
   a. subtract original $row\ p\ * \frac{y_{zq}}{y_{pq}}$

Pivot Selection:

- **Entering value** - better obj value
  - new obj vale < current obj value
  - need any NBV where $\beta > 0$
    - if one exists, pick this one
    - if more than one exists,
      - pick one with largest β_q
    - if several have same largest,
      - pick one with smallest I
- **Leaving variable** - feasibility
  - RHS of row ≥ 0
  - Express as leaving variable $(as\ \lambda)$
    - Choose one which gives smallest λ
      - Can't make others negative

BV	z	x_1	$\cdots$	x_p	$\cdots$	x_q	$\cdots$	x_n	RHS
z	1	β_1	$\cdots$	β_p	$\cdots$	β_q	$\cdots$	β_n	β_0
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$		$\vdots$	$\vdots$
x_p	0	y_{p1}	$\cdots$	y_{pp}	$\cdots$	y_{pq}	$\cdots$	y_{pn}	y_{p0}
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$		$\vdots$	$\vdots$

Simplex Algorithm:

0. Find initial BFS and its basic representation
1. STOP if $\beta_i \leq 0\ for\ all\ i \notin I$
2. STOP if $\beta > 0\ for\ any\ NBV\ j$
   a. AND for **ALL** BV, i, $y_{ij} \leq 0$
   b. As no finite min exists
3. Choose entering NBV, q, with largest β
4. Choose exiting value
   a. Express in terms of q $(as\ \lambda)$
5. Pivot on y_{pq}
   a. Now go back to step 1

Finite Termination & Degeneracies

23 October 2018 18:06

Degenerate BS: one or more BV are zero
Non-Degenerate BS: all BV's are non-zero

Finite Termination Theorem:

- if all *BFS's are non-degenerate*
  - simplex algorithm must *terminate after a finite number of steps*
    - with an *optimal solution*
    - or with a proof that the problem is *unbounded*
- Proof
  - RHS for BV are > 0 if non-degenerate
  - then new objective value is always decreasing
    - unless optimal or unboundness if caught in STEP 1 or 2
  - will terminate - no BS will ever be repeated
    - at most, **n choose m** solutions
    - cannot run indefinably

Degeneracy and Simplex:

- if pivot on (p,q) where $y_{p0}=0$
  - new BFS obj value same
    - no improvement
  - finite termination theorem breaks
  - pivot is called degenerate

Cycling and **Bland's Rule**:

- can cause cycling if using degenerate pivots
  - necessary, but not sufficient
- **Bland's Rule:**
  - choose the **left-most** column **q** with positive cost
    - From initial tableau
    - And do until end
    - $(\beta_j > 0)$
  - choose **p** same as you would normally
  - removes cycling
    - finite

Two Phase Simplex

29 October 2018 01:28

Initial BFS:

- All slack variables
  - Not always an obvious initial BFS
    - If we have <=, >= inequalities etc.
- **Artificial variables**
  - For == and >= , add ξ_{av}
  - Now have a BS, with the AVs acting as BVs
  - Auxiliary LP:
    - Minimize $\zeta = sum\ \xi_{av}$
      - Add equations involving AVs to obj
    - Solve using Simplex
    - Must terminate, with two cases
    - $\zeta = 0 \rightarrow provides\ BFS\ for\ original\ LP$
    - $\zeta > 0 \rightarrow original\ LP\ is\ infeasible$

Two Phase Simplex Method:

- Phase 1:
  - Modify for negative RHS
  - **Identify** equality and >= inequalities
  - Modify for inequalities (slack variables)
  - Add Artificial variables (for those in Step 2)
  - Get Auxiliary LP
  - Solve using Simplex
  - Type equation here.
- Phase 2:
  - Case 1:
    - $\zeta 1 > 0\ - original\ LP\ is\ infeasible$
  - Case 2:
    - $\zeta = 0\ and\ all\ \xi_i\ are\ nonbasic\ (= 0)$
    - Remove all AVs from tableau
    - Derive basic representation for z
      - Using I from end of Phase 1
    - Now solve using Simplex algorithm
  - Case 3:
    - $\zeta = 0\ and\ and\ at\ least\ one$ ξ_i is basic
      - Since $\zeta = 0\ then\ all\ \xi_i = 0$
      - So we have a basic variable =0
      - Phase 1 - given a **degenerate BFS**
        - Pivot on $y_{pq}, \xi_p\ and\ x_q\ while\ keeping\ \zeta = 0$
      - Can now remove all ξ_i
      - Solve using Simplex

Extensions of LP

30 October 2018 20:09

Non-linear programs can sometimes be reformulated into:

- a single LP
- a sequence of LP problems

Min-Max (**game theory**):

- $MM: \ y_i(x) = c(i)^T x + d(i)$
  - $set\ \phi(x) = \max c(i)^T x + d(i)$
  - $\min \phi(x)$
  - $s.t. Ax = b, x \geq 0$
- LP: $\min z$
- $s.t$

  $z \geq \ c(i)^T x + d(i)\ \forall i$

  $Ax = b$

  $x \geq 0, z\ free$
- **optimal in LP:** $(\boldsymbol{x_{LP}^*}, \boldsymbol{z_{LP}^*})$
- **then MM has optimal value** $\boldsymbol{\phi(x_{LP}^*) = z_{LP}^*}$

Proof by contradiction:

- let (x_{LP}^*, z_{LP}^*) be optimal in LP
- assume not optimal in MM
  - then $\phi(x_{MM}^*) < \phi(x_{LP}^*)$
- but then x_{MM}^* must be feasible in LP
- $by\ constraints\ of\ the\ LP, z_{LP}^* \geq \phi(x_{LP}^*)$
  - $z_{LP}^* \geq \phi(x_{LP}^*) > \ \phi(x_{MM}^*)$
  - (x_{MM}^*) would give a better obj value in LP
  - but this can't happen as x_{LP}^* is optimal
    - so other point cannot exist

Min-Min:

- $MM: \ y_i(x) = c(i)^T x + d(i)$
  - $set\ \psi(x) = \min c(i)^T x + d(i)$
  - $\min \psi(x)$
  - $s.t. Ax = b, x \geq 0$
- LP'**s**:

  min $\quad z_i = c(i)^T x(i) + d(i) \quad LP(i)$

  s.t. $\quad Ax(i) = b$

  $x(i) \geq 0$
  - $z_j^* = \min z_i \qquad LP(j)$
    - then $x^*(j)$ is optimal in MM
    - $\psi^* = \psi\left(x^*(j)\right) = z_j^*$

Goal Programming:

- replace absolute values with a summation
- $| \ldots | = U - V$
- $in\ obj, U + V$
  - then add new constraint

Replace Max function Max(A,x)

- new variable V
- V>A

Fractional LP:

- Assume that feasible set is bounded
- Denominator of objective function is strictly positive
- **Homogenisation:**
  - For each $x_i\ create\ a\ y_i and\ a\ y_0$
    - $\alpha - num\ coeff$
    - $\beta - den\ coeff$
  - $\min \alpha_0 y_0 + \alpha_1 y_1 + \cdots + \alpha_n y_n$
  - $s.t\ \ \beta_0 y_0 + \sum_{j=1}^{n} \beta_j y_j = 1$
  - $Ay = by_0$
  - $y_{0\ldots n} \geq 0$

Integer Programming

29 October 2018 08:23

Discrete **Integer** Values
Can mix - MILP

For min problem,
$f(x)_{MILP} \geq f(x)_{LP}$
- either the same or worse

MILP Standard form:
- slack and excess in MILP are **continuous**

Pure IP Standard form:
- apply LP rules apart from slack and excess
- scale to get integer coefficients
- insert slack and excess variables

Finite-Valued Variables
- $x_j \in \{p_1, \dots, p_m\}$
- replace x_i
  - with z_{ij}
  - $x_i = p_j z_{ij}$
  - $j \in \{1, \dots, m\}$

Knapsack Problem:
- items have weight w_j , $j \in \{1, \dots, n\}$
- items have value v_j , $j \in \{1, \dots, n\}$
- $x_j = 1\ if\ items\ is\ packed$
- 'knapsack' has capacity $\boldsymbol{W}$
- $\max z = \sum_{j=1}^{n} v_j x_j$
- $s.t.$

  $\sum_{j=1}^{n} w_j x_j \leq W$

  $x_j \in \{0,1\} \qquad \forall j \in \{1, \dots, n\}$

Bin-Packing Problem:
- items have weight w_j , $j \in \{1, \dots, n\}$
- k bins of capacity $\boldsymbol{W}$
  - $y_i = 1\ if\ bin\ i\ is\ used$
- $x_{ij} = 1\ if\ item\ j\ assigned\ to\ bin\ i$
- minimise number of bins
- $\min z = \sum_{i=1}^{k} y_i$
- $s.t.$

  $\sum_{j=1}^{n} w_j x_{ij} \leq W y_i$

  $\sum_{i=1}^{k} x_{ij} = 1 \qquad \forall j \in \{1, \dots, n\}$

  $x_{ij}, y_i \in \{0,1\} \qquad \forall i \in \{1, \dots, k\}, \forall j \in \{1, \dots, n\}$

Cutting Plane

10 November 2018 21:35

Continuous Relaxation:

- LP relaxation:
  - replace all integer variables with continuous
  - Solve
- if optimal is integer - done
- else - tighten LP relaxation and repeat

Cutting Plane Algorithm:

0. Write ILP in standard form
1. Solve LP relaxation
2. if optimal is integer - STOP
3. Generate a cut
   a. knocks out optimal vertex
   b. only removes non-integer solutions
4. Add cut to LP relaxation - Step 1

Gomory mixed-integer cuts:

- $f = x - \underline{x}$
- RHS for a BV in tableau is not integer
- for LHS get f part * x_j (NBV)
  - sum of these >= fraction part of RHS
    - add this as new row
      - after standardising
    - use two-phase to solve
      - $\boldsymbol{+\xi_1 - x_i}$

Knapsack cover cuts:

- add constraints to get rid of infeasible solutions
  - when BV set to 1 and not 0
  - $SUM\ i\ x\ terms \leq i - 1$
    - min cover cut - break down above
- e.g. $5x_2 + 4x_3 \leq 8$
  - $x_2 + x_3 \leq 1$

Branch & Bound

10 November 2018 21:35

Finite bound for variables, min and max

For MILP

<u>Divide and Conquer to MILP:</u>

Notation:

- P_i: $ith\ subproblem$
  - drop ILP, each are LPs
- $x^*(P_i)$: $optimal\ solution\ to\ the\ ith\ subproblem$

Application:

- Solve LP relaxation of problem $P_0 \Rightarrow x^*(P_0)$ $(original\ MILP)$
- Stop if solution satisfies integrality constraints
- Otherwise choose non-integer $x_p^* \in x^*(P_0), p \in Z$ $(integer\ x's)$
  - **Divide**: (branch)
    - create two subproblems by adding constraints to P_0
      - $P_1: x_p \leq \underline{\underline{x_p^*}}$ $(floor)$
      - $P_2: x_p \geq \overline{\overline{x_p^*}}$ $(ceiling)$
  - **Conquer**: (bound)
    - if optimal for subproblem is worse than any known feasible solution for P_0 - stop with that branch
    - as - any solution of P_0 satisfying integrality constraints is feasible in **one of** P_1 and P_2 - hence can solve P_0 by solving P_1 **and** P_2

Branch and Bound Steps:

1. Initialise
   a. $OPT = \infty$
   b. Solve P_0
   c. if $x^*(P_0)$ is feasible, STOP
   d. else list of problems = {P_0}
2. Problem Selection
   a. from list of problems, extract P_i having opt < OPT
   b. if non exists, STOP
3. Variable Selection
   a. choose x_p to divide on
4. Branch
   a. create subproblems P' and P''
      i. (see left)
5. Bound P':
   a. find $x^*(P')$
   b. if opt < OPT
      i. if feasible for P_0 (integer..)
         1) OPT = opt
      ii. else add P' to list of problems
6. Bound P''
7. go back to step 2

Duality

15 November 2018 22:30

Dual LP:

Primal LP	Dual LP
$\max_{x_1,x_2} z = x_1 + 6x_2$	$\min_{y_1,y_2,y_3} 200y_1 + 300y_2 + 400y_3$
s.t. $x_1 \leq 200$	s.t. $y_1 + y_3 \geq 1$
$x_2 \leq 300$	$y_2 + y_3 \geq 6$
$x_1 + x_2 \leq 400$	$y_1, y_2, y_3 \geq 0$
$x_1, x_2 \geq 0$	

$$\max \left\{c^T x \,:\, Ax \leq b,\, x \geq 0\right\} \qquad \min \left\{b^T y \,:\, A^T y \geq c,\, y \geq 0\right\}$$

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

P_{max}	D_{min}
const	R var
var	const

P_{min}	D_{max}
const	var
var	R const

Comment on rules when transforming

Weak Duality:

- both Primal and Dual are feasible
- $\boldsymbol{c^T x \leq b^T y}$
  - Dual is upper bound for Primal
- for P max

Strong Duality:

- both Primal and Dual are feasible
- **B** is optimal basis for Primal
- $y^* = (B^{-1})^T c_B$
- $\boldsymbol{c^T x^* = b^T y^*}$

Direct Conversion:

1.
   a. primal constraint -> dual variable
   b. primal variable -> dual constraint
2.
   a. constraints coefficients given by A^T
   b. dual costs = primal RHS
   c. dual RHS = primal costs
3. If primal max, dual is min
   a. primal constraint - $[\geq, =, \leq]$
      i. dual variable - $[y \leq 0, y \in \mathcal{R}, y \geq 0]$
   b. primal variable - $[x \geq 0, x \in \mathcal{R}, x \leq 0]$
      i. dual constraint - $[\geq, =, \leq]$
4. If primal min, dual is max
   a. primal constraint - $[\geq, =, \leq]$
      i. dual variable - $[y \geq 0, y \in \mathcal{R}, y \leq 0]$
   b. primal variable - $[x \geq 0, x \in \mathcal{R}, x \leq 0]$
      i. dual constraint - $[\leq, =, \geq]$

Indirect Conversion:

1. Make into form of P or D
   a. replace $x_i \in \mathbb{R}$ *with* $x_i^+ - x_I^-$
      i. $x_i^+, x_i^- \geq 0$
   b. replace equality with two inequality
   c. if necessary
      i. change directions of inequality
      ii. change obj max/min
2. Obtain Dual
3. Simplify (Optional)
   a. replace variables pairs that appear in all constraints as $\alpha y_i - \beta y_j$
   b. replace matching inequality with equality

Sensitivity Analysis

Standard Form

20 November 2018 11:01

Uncertainty in a RHS in LP: (perturbation - value function)

- Optimal solution is can be affected by this value

- $v(RHS) - optimal\ value\ depending\ on\ p$
  - As RHS increases from 0
  - $v(RHS)\ improves$
  - Piecewise linear

Evaluate from final Simplex tableau:

- If a **row t** had a slack variable $\boldsymbol{x_s}$
- Then $\boldsymbol{\Pi_t = \beta_s}$ from final tableau

Shadow Prices:

- Is a vector
  - $\boldsymbol{\Pi = \left(B^{-1}\right)^T c_B}$
  - Doesn't have to be unique
  - Give info about the **sensitivity** of v(RHS)

Behaviour of value function:

- $v(p) = v(b) + \Pi^T(p-b)$
- If $B^{-1}p \geq 0$
  - $v(p) = c_B^T B^{-1} p + c_B^T B^{-1} b - c_B^T B^{-1} b$
    - $= c_B^T B^{-1} b + c_B^T B^{-1}(p-b)$
    - $v(b) + \Pi^T(p-b)$

Global Behaviour:

- In general
  - $v(p) \geq v(b) + \Pi^T(p-b)\ for\ all\ p$
  - reversed for max problems
  - PRROF>

Zero Sum Games

Focus on Pure Strategy - just pick one strategy always

23 November 2018 11:26

Games:

- Players must guess what other players will do
- Outcome depends on combination of players strategies

Two Person Zero-Sum Games:

- Row Player (RP) and Column Player (CP)
- RP chooses one out of m strategies
- CP chooses one out of n strategies
- Zero-Sum - RP + CP gains = 0 (one has 'negative gain')
- Assumptions
  - Each player knows **game setting**
    - RP and CP available strats
    - Payoff matrix values
  - Moves are made simultaneously
    - Not turn by turn
  - Players play to get best

Payoff Matrix:

- Rows - strats available to RP
- Columns - strats available to CP
- Element i,j - gain for RP
- Eliminate rows/columns that RP/CP should never play
  - Eliminate 'dominated rows'
  - **Dominant Strategy Equilibria:**
    - If after removing all dominated strategies
    - Left with **one strategy for each player**
      - This pair is DSE

When no DSE:

AFTER ELIMINATING DOMINATED R/C

Minimum Row Payoff - maxmin

- $\alpha_i = \min row\ element$
- RP chooses **i** with **max** $\boldsymbol{\alpha_i}$

Maximum Column Payoff - minmax

- $\beta_j = \max column\ element$
- CP chooses **j** with **min** $\boldsymbol{\beta_j}$

Nash Equilibria:

- No player has any incentive to unilaterally deviate from strategy pair **if told** strategy of other player
- Occurs when
  - $\max \alpha_i = \min \beta_j$

Mixed Strategies

25 November 2018 19:54

No Nash Equilibria exists

- So randomly play strategies

<u>Mixed Strategy:</u>

- RP plays strat **i** with prob $\boldsymbol{p_i}$
- CP plays strat **j** with prob $\boldsymbol{q_j}$

- Payoff $= \sum_{i=1}^{m} \sum_{j=1}^{n} p_i q_i a_{i,j}$
  - Non-linear optimization
- RP seeks p_i's that maximize payoff
- CP seeks q_j's that minimize loss

<u>Minimax Theorem:</u>

- For every two-person zero sum games
- The *RP and CP LP's have the same optimal* value
  - $\boldsymbol{V^* = V_{RP}^* = V_{CP}^*}$
  - -> Nash equilibrium always exists for mixed strategies
    - Extends to M players

<u>Column Player's Perspective:</u>

- For all possible combination of q_i
  - Select max $\alpha_i = \sum_1^n q_j a_{ij}$
- Then select the set of q_i with min(max α_i)
- Equivalent LP:
  - $V_{CP} = \min\limits_{\tau, q_1, \dots, q_n} \tau$
  - $\tau \geq \sum\limits_{j=1}^{n} q_j a_{ij} \quad \forall i = 1 \dots m$
  - $\sum\limits_{j=1}^{n} q_j = 1 \qquad q_j \geq 0, \forall j = 1 \dots n$

<u>Row Player's Perspective:</u>

- For all possible combination of p_i
  - Select min $\beta_i = \sum_1^m p_i a_{ij}$
- Then select the set of p_i with max(min β_i)
- Equivalent LP:
  - $V_{RP} = \max\limits_{\tau, p_1, \dots, p_m} \tau$
  - $\tau \leq \sum\limits_{j=1}^{m} p_i a_{ij} \quad \forall j = 1 \dots n$
  - $\sum\limits_{j=1}^{m} p_i = 1 \qquad p_i \geq 0, \forall i = 1 \dots m$

Case Study 1

17 October 2018 17:39

If Upper Bound > Activity, then that row is limited

- In this case,
  - Sulphur and nitrate are the limiting factors for the production

Case Study 2

17 October 2018 17:39

GLPK by default doesn't need std form

Case Study 3 - Klee and Minty

30 October 2018 20:09

Visit all vertices of feasible set before finding Optimal

- Worst case of Simplex

Classical problem:

- Obj value $= -10^{2n-2}$
- $(x_1, \dots, x_n) = (0, \dots, 10^n)$
- has no degenerate BFS

Numerical issues:

- Solution became infeasible due to round-off errors
- Or, LU transformation failed
- Or, cannot fit number, too large

Fix with --exact

- Detrimental to performance

Steepest Edge:

- Select the NBV entering in the pivot process
- No improvement when using --exact

Interior Point method:

- Proportional to polynomial of n

Case Study 4 - Linear Separability

30 October 2018 20:09

Hyper-plane that divides two sets of point in d-dimensions

$a_1 x_1^{(i)} + \cdots + a_d x_d^{(i)} = b$ *splits the two sets*

Then

$a_1 x_1^{(i)} + \cdots + a_d x_d^{(i)} < b \quad \forall i = 1, \dots, N \ (N\ points)$

$a_1 y_1^{(j)} + \cdots + a_d y_d^{(j)} > b \quad \forall j = 1, \dots, M \ (M\ points)$

Add a small positive ϵ

$a_1 x_1^{(i)} + \cdots + a_d x_d^{(i)} \leq b - \epsilon \quad \forall i = 1, \dots, N \ (N\ points)$

$a_1 y_1^{(j)} + \cdots + a_d y_d^{(j)} \geq b + \epsilon \quad \forall j = 1, \dots, M \ (M\ points)$

Divide both sides by ϵ ($\alpha' = \alpha/\epsilon$)

$a'_1 x_1^{(i)} + \cdots + a'_d x_d^{(i)} \leq b' - 1 \quad \forall i = 1, \dots, N$

$a'_1 y_1^{(j)} + \cdots + a'_d y_d^{(j)} \geq b' + 1 \quad \forall j = 1, \dots, M$

Finally

$a'_1 x_1^{(i)} + \cdots + a'_d x_d^{(i)} \quad - b' \leq -1 \quad \forall i = 1, \dots, N$

$-a'_1 y_1^{(j)} - \cdots - a'_d y_d^{(j)} + b' \leq 1 \quad \forall j = 1, \dots, M$

Objective function is constant,

- Just want to find a feasible solution
  - i.e. the tuple $(a_1, \dots, a_d, b)$

Case Study 5 - Totally Unimodular

10 November 2018 21:35

Unimodular matrix => det = +-1
Totally unimodular matrix => all square non-singular sub matrices are unimodular

if A is totally unimodular, the ILP can be solved with LP methods

Case Study 6 - Network flow

15 November 2018 22:30

Max flow = min cut
strong duality

Case Study 7

15 November 2018 22:30

RP and CP LPs have strong duality