CO343 Operations Research

1 Linear Programming

Linear program

- Optimises a linear *objective function* $f : \mathbb{R}^n \to \mathbb{R}$.
- Over a *feasible set* $\chi \subseteq \mathbb{R}^n$ described by (in)equality *linear constraints*.
- The feasible region is a *convex polyhedron*, and the *vertices* contain a solution.

Standard form

$$\begin{aligned} \text{minimise } & z = c^\top x \\ \text{subject to } & Ax = b \\ & x \geq 0 \end{aligned}$$

1. *Maximisation* $\to$ *Minimisation*: invert objective.
2. *Inequalites* $\to$ *Equalities*: add slack/excess variables.
3. *Negative* $\to$ *Non-negative RHS*: multiply constraint by -1.
4. *Free decision variables* $\to$ *Non-negative variables*:
    - Substitute $x_j = x_j^+ - x_j^-$ with $x_j^+, x_j^- \geq 0$, or
    - Use an equality constraint to eliminate it (sub into all other constraints).

Examples

1. *Resource allocation models*: split a resource. E.g. find assignment of CPU share to maximise completion rate.
    - Variables: how much of each resource allocated to each use.

- Constraints: on resource availability.

2. *Blending models*: combine resources. E.g. find most economical diet meeting nutritional requirements.
   - Variables: how much of each resource to use in the mix.
   - Constraints: express composition of output.
3. *Operations planning models*: decide organisational strategy. E.g. minimise cost of shipping goods.
   - Variables: identify products, activities, processing facilites, etc.
   - Constraints: *balance* inputs and outputs of activities.
4. *Shift scheduling models*: allocate workforce to tasks. E.g. minimise cost of shifts.
   - Variables: number of employees.
   - Contraints: allocate enough workers to cover activities.
5. *Time-phased models*: address circumstances that vary over time.
   - Variables: express returns or state at given time.
   - Constraints: time-phase balance constraints.

Linear independence Linear independence of rows in A implies either:

1. Contradictory constratints.
2. Redundant constraints.

Index set Set of indexes for columns which are linearly independent.

Basis Matrix consisting of columns referenced by the index set.

Basic solution

1. A solution to $Ax = b$ with $x_i = 0$ for all $i \notin I$ is a *basic solution* (vertices).
   - A basic solution corresponding to an index set I is unique since the columns referenced by the index set are linearly independent, and so the basis B is invertible.
2. A solution to $Ax = b$ with $x \geq 0$ is a *feasible solution* (in feasible set).
3. A *basic feasible solution* is both basic and feasible (vertices of the feasible set).

Shadow prices Objective coefficient of x_s for slack vars.

$$\Pi = \left(B^{-1}\right)^\top c_B$$

May be more than one optimal basis $\implies$ shadow prices need not be unique.

$$v(p) = v(b) + \Pi^\top (p-b)$$

if $B^{-1}p \geq 0$ (still feasible). In general:

$$v(p) \geq v(b) + \Pi^\top (p-b)$$

$-\Pi$ gives the maximum price one should pay for an additional unit.

4 Game Theory

Payoff matrix What RP gains and CP loses.

Dominance

1. Row where some other row has all values greater than or equal to us.
2. Column where some other column has all values less than or equal to us.

Nash equilibrium

$$\max_{i=1,\dots,m} \alpha_i = \min_{j=1,\dots,n} \beta_j$$

Without an equilibrium

$$V_{CP} = \min_{q_1,\dots,q_n} \max_{p_i,\dots,p_m} \sum_{i=1}^{m} \sum_{j=1}^{n} p_i q_j a_{ij}$$

subject to $\sum_{j=1}^m q_j = 1$ and $\sum_{i=1}^n p_i = 1$ with all ps and qs ≥ 0.

We can rewrite this as the linear program:

$$\begin{array}{rl} \min & \tau \\ \text{s.t} & \tau \geq \sum_{j=1}^n q_j a_{ij} \quad \forall i = 1, \dots, m \\ & \sum_{j=1}^n q_j = 1 \\ & q_j \geq 0 \end{array}$$

Fundamental theorem of LP

1. $\exists$ a feasible solution $\implies$ $\exists$ a BFS.
2. $\exists$ an optimal solution $\implies$ $\exists$ an optimal BFS.

Variables

- Variables $\{x_i\}_{i\in I}$ are *basic variables*.
- Variables $\{x_i\}_{i\notin I}$ are *nonbasic variables*.

Basic representation Reformulation of the system $\left(z = c^\top x, Ax = b\right)$ which expresses the objective function and each BV as linear function of the NBVs. This ends up to be something like:

$$\begin{aligned} z =& c_B^\top B^{-1} b + \left(c_N - N^\top B^{-\top} c_B\right)^\top x_N \\ x_B =& B^{-1} b N x_N \end{aligned}$$

where $r = c_N - N^\top B^{-\top} c_B$ is the *reduced cost vector* that tells us:

1. Whether the current BFS is optimal $(r \geq 0)$.
2. Find a new BFS with a lower objective value (increase a nonbasic variable with a negative reduced cost).

Simplex tableau A nicer way of writing out basic representation:

BV	z	$x_B^\top$	$x_N^\top$	RHS
z	1	$0^\top$	$-r^\top$	$c_B^\top B^{-1} b$
x_B	0	I	$B^{-1}N$	$B^{-1}b$

Simplex algorithm Use Qiang's notes...

Degenerate BS One or more basic variables are 0.

- If all BFS's are non-degenerate, then the simplex algorithm terminates after a finite number of steps (solution / unbounded).
  - The sequence of objective values is strictly decreasing.

- The number of solutions is $\leq \left(\begin{array}{c} n \\ m \end{array} \right)$.
- Therefore must terminate after a finite number of iterations.

- A BS x is degenerate iff it is associated with more than one index set.
  - Assume a BS x corresponds to two index sets I_1 and I_2 with $I_1 \neq I_2$, then an NBV x_i in I_1 must be a BV in x_2, we have a BV equal to 0.
  - If we have a BV equal to 0, we can pivot on that value to get a different index set.
- Finite termination theorem breaks down with degeneracy!

Bland's rule

1. Choose the leftmost non-basic column with positive cost.
2. Choose the row with minimal $\overline{x}_{iq}$, choosing the smallest index in case of ties.

Not useful in practice - replacing $y_{i0} = 0$ with $y_{i0} = \epsilon$ is acceptable.

Two-phase simplex algorithm

- Use when there is no obvious initial BFS.
- I.e. "all-slack basis" not possible: $=$ or $\geq$ constraints.

Phase 1

1. Add artificial variables to constraints without slack variables.
2. Minimse the sum of artificial variables ζ:
   (a) Find a basic representation for ζ.
   (b) Find the minimum value for ζ using simplex algorithm.
3. If $\zeta^* > 0$ stop, the LP is infeasible.
4. If $\zeta^* = 0$ but some artificial variable is still basic, we have a degenerate BFS. Pivot to remove artificial variable from the BFS.
5. If $\zeta^* = 0$ with all artifical variables non-basic, go to phase 2.

Finding the Dual Indirect method:

1. Bring the LP to a P or D form:
   (a) Replace free variables.
   (b) Replace equalities with two inequalities.
   (c) Change constraint direction by multiplying with -1 where necessary.
2. Obtain dual according to its definition.
3. Simplify dual problem (optional):
   (a) Replace variable pairs $y_i, y_j \geq 0$ that always occur as $\alpha y_i - \alpha y_j$ by $y_k \in \mathbb{R}$.
   (b) Replace matching inequality constraints with equality constraints.

Direct method:

1. For every primal constraint ccreate one dual variable, for every primal variable create one dual constraint.
   (a) If primal is *min*:
      i. Constraint $(\geq, =, \leq)$ goes to variable $(y_i \geq 0, y_i \in \mathbb{R}, y_i \leq 0)$.
      ii. Variable $(x_j \geq 0, x_j \in \mathbb{R}, x_j \leq 0)$ goes to constraint $(\leq, =, \geq)$.
   (b) If primal is *max*:
      i. Constraint $(\geq, =, \leq)$ goes to variable $(y_i \leq 0, y_i \in \mathbb{R}, y_i \geq 0)$.
      ii. Variable $(x_j \geq 0, x_j \in \mathbb{R}, x_j \leq 0)$ goes to constraint $(\geq, =, \leq)$.
2. The dual coefficient matrix comes from the transposed primal coefficient matrix.
3. Former RHS become new costs.
4. Former costs become new right hand sides.

Value function

$$v(p) = \min \left\{ z = c^\top x : Ax = p, x \geq 0 \right\}$$

$v(p)$ is *non-increasing*, *convex* and *piecewise linear*.

Knapsack cover cut For a cover S ($\sum_{j\in S} w_j > W$), we can add the cut:

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

Is minimal if no subset of S is a cover.

Branch and bound

1. Solve LP relaxation of problem.
2. If solution is integer, finish.
3. Otherwise, choose non-integer x_p^*:
   (a) Divide, adding constraints $x \leq \lfloor x_p^* \rfloor$ and $x \geq \lceil x_p^* \rceil$.
   (b) Solve recursively. Disregard a branch if a better feasible solution is seen.

3 Duality

Every prial problem,

$$\min \left\{ b^\top y : A^\top y \geq c, y \geq 0 \right\}$$

has a dual problem,

$$\max \left\{ c^\top x : Ax \leq b, x \geq 0 \right\}$$

Weak duality $c^\top x \leq b^\top y$.

Strong duality If P and D are both feasible and B is an optimal basis for P, with optimal basic solution (x_B^*, x_N^*):

1. The optimal solution to the dual problem are the shadow prices: $y^* = \left(B^{-1}\right)^\top c_B$.
2. The objective values coincide: $c^\top x^* = b^\top y^*$.

Phase 2

1. Remove artificial columns from phase 1 tableau.
2. Find a basic representation for z, add to tableau.
3. Find the minimum value for z using simplex algorithm.

Min-max problems Where $\phi(x) = \max_{i=1,\dots,I} \left\{ c(i)^\top x + d(i) \right\}$:

$$\left.\begin{array}{ll} \min & \phi(x) \\ \text{s.t.} & Ax = b, x \geq 0 \end{array}\right\} = \left\{\begin{array}{ll} \min & z \\ \text{s.t} & z \geq c(i)^\top x + d(i) \, \forall i = 1, \dots, I \\ & Ax = b \\ & x \geq 0, z \text{ free} \end{array}\right.$$

Proof: suppose that there is a solution (x_{RHS}^*, z_{RHS}^*) for the RHS, and assume that there is different optimal solution x_{LHS}^* for the LHS with $\phi(x_{LHS}^*) \leq \phi(x_{RHS}^*)$, then find a contradiction.

Min-min problems Where $\psi(x) = \min_{i=1,\dots,I} \left\{ c(i)^\top x + d(i) \right\}$:

$$\left.\begin{array}{ll} \min & \psi(x) \\ \text{s.t.} & Ax = b, x \geq 0 \end{array}\right\} = \left\{\begin{array}{ll} \min & z_i = c^\top(i)x(i) + d(i) \\ \text{s.t} & Ax(i) = b \\ & x(i) \geq 0 \end{array}\right.$$

Then choose $\min_{i=1,\dots,I} z_i$.

Proof: follows directly from $\min_{x\in X} \min_{y\in Y} f(x,y) = \min_{y\in Y} \min_{x\in X} f(x,y)$.

Goal programming

$$\min \; \sum_{i=1}^I \left| c(i)^\top x - d(i) \right| \; \Bigg\} = \left\{\begin{array}{ll} \min & \sum_{i=1}^I z_i^+ + z_i^- \\ \text{s.t} & c(i)^\top x - d(i) = z_i^+ - z_i^- \\ & z_i^+, z_i^- \geq 0 \end{array}\right.$$

Fractional linear programming

$$\left.\begin{array}{ll} \min & \frac{\alpha_o + \alpha_1 x_1 + \cdots + \alpha_n x_n}{\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n} \\ \text{s.t.} & Ax = b, x \geq 0 \end{array}\right\} = \left\{\begin{array}{ll} \min & \frac{\alpha_o y_0 + \alpha_1 y_1 + \cdots + \alpha_n y_n}{\beta_0 y_0 + \beta_1 y_1 + \cdots + \beta_n y_n} \\ \text{s.t} & b_i y_0 - \sum_{j=1}^n a_{ij} y_j = 0 \forall i = 1, \dots, m \\ & y_0 > 0, y_1 \geq 0, \dots, y_n \geq 0 \end{array}\right.$$

Then we can always multiply any y by $\lambda \implies$ we can make the denominator 1:

$$\begin{array}{rl} \min & \alpha_o y_0 + \alpha_1 y_1 + \cdots + \alpha_n y_n \\ \text{s.t} & \beta_0 y_0 + \sum_{j=1}^{n} \beta_j y_j = 1 \\ & b_i y_0 - \sum_{j=1}^{n} a_{ij} y_j = 0 \forall i = 1, \ldots, m \\ & y_0 > 0, y_1 \geq 0, \ldots, y_n \geq 0 \end{array}$$

- Need to prove that the denominator of the objective function is strictly positive for all feasible x.

2 Integer Programming

- *Mixed Integer Linear Programming*: some variables are integers.
- *Pure Integer Linear Programming*: all variables (including slack, objective val) are integers.
- *Binary Linear Programming*: all variables are binary.
- *Mixed Integer Binary Porgramming*: integer variables are binary.

Standard form

1. Get to LP standard form (except slack/excess vars):
   (a) *Maximisation → Minimisation.*
   (b) *Negative → Non-negative RHS.*
   (c) *Free decision variables → Non-negative variables.*
2. Scale the equations to make all coefficients integers.
3. Insert integer slack and/or excess vars.

Examples

1. *Knapsack problem.*
2. *Bin-packing problem.*
3. *Capital budgeting*: which project to undertake given limited resources?
4. *Facility location*: which centres should be built and how should demand be satisfied to minimise costs?
5. *Crew scheduling*: which sequences to operate such that costs are minimal and all flights have crew?

Logical operations Using Big-M:

- *Or*: $a_1^\top x \leq b_1 \vee a_2^\top x \leq b_2$:

$$\begin{array}{rcl} a_1^\top x & \leq & b_1 + M\delta \\ a_2^\top x & \leq & b_2 + M(1-\delta) \\ \delta & \in & \{0,1\} \end{array}$$

- *k-out-of-m*: At least k of $a_1^\top x \leq b_1, a_2^\top x \leq b_2, \ldots, a_m^\top x \leq b_m$:

$$\begin{array}{rcll} a_1^\top x & \leq & b_1 + M\delta_1 & \\ a_2^\top x & \leq & b_2 + M\delta_2 & \\ & \vdots & & \\ a_m^\top & \leq & b_m + M\delta_m & \\ \sum_{j=1}^{m} \delta_j & \leq & m - k & \\ \delta_j & \in & \{0,1\} & \forall j \in \{1..m\} \end{array}$$

Finite-valued variables $x_j \in \{p_1, p_2, \ldots, p_m\}$:

1. Introduce variables $y_{j1}, y_{j2}, \ldots, y_{jm} \in \{0,1\}$.
2. Add the constraint $y_{j1} + y_{j2} + \cdots + y_{jm} = 1$.
3. Replace x_j with $p_1 y_{j1} + p_2 y_{j2} + \cdots + p_m y_{jm}$.

Cutting plane algorithm

1. Write ILP in standard form.
2. Solve LP relaxation.
3. If solution is integer, finish.
4. Otherwise, introduce cuts and repeat.

Gomory cut Where $f_j := y_{ij} - \lfloor y_{ij} \rfloor$ and $f := y_{i0} - \lfloor y_{i0} \rfloor$, we can always add the cut:

$$\sum_{j \notin I} f_j x_j^* \geq f$$