Standard Form

Minimisation
Min to Max: Invert the Objective Function
Equality Constraints
$\leq$ Inequality Constraints: + Slack Variables ($+s_i$)
$\geq$ Inequality Constraints: - Excess Variables ($-s_i$)
Non-Negative RHSs
Positive to Negative RHS: Constraint x-1 (Changes Sign of Inequality)
Non-Negative Decision Variables
Substitute $x_j = x_j^+ - x_j^-$ with $x_j^+, x_j^- \geq 0$, or
Use any equality contraitnt involving x_j to eliminate x_j (substitution)

Index Sets, Basis, & Solutions to LP

Assuming LP in Standard Form:
$min\ z = c^T x$
$s.t.\ Ax = b$
$x \geq 0$
$A \epsilon \mathbb{R}^{m \times n}$ $B \epsilon \mathbb{R}^m$ $c \epsilon \mathbb{R}^n$
rank(A) = m & rows linearly independent
Index Set:
Set I containing indexes of m independent columns of A
Basis:
$B \epsilon \mathbb{R}^{m \times m}$ consisting of the columns of A in I (full rank)
Basic Solution: formed by setting $n - m$ NBVs to 0 & solving for the remaining m BVs
n: total no. of variables (no. columns of A)
m: total no. of constraints (no. rows of A)
BVs: m variables with index in I
NBVs: $n - m$ variables with an index not in I
Feasible Solution: solutions that satisfy all the constraints of the LP (including non-negativity constraints)
Set of all Feasible Solutions defines the Feasible Region
Optimal Solution lies within this region (if exists)
Basic Feasible Solution: a solution that is both basic & feasible - satisfying all constraints
Formed by setting $n - m$ variables to 0 & solving for the remaining m variables (vertexes of feasible region)
If an optimal solution exists - there must be at least one optimal solution that is a BFS
BFSs used to locate the Optimal Solution
Unique Basic Solution:
$Bx_B = b$ has unique solution $x_b = B^{-1}b$
Basic Solution corresponding to Index Set I
Vector of m unique solutions & $n - m$ 0s
Finding the Unique Basic Solution:
Take Standard Form LP & Select m columns of A that are linearly independent
Form Indices into Index Set I & form columns into Basis B, giving $Bx_B = b$
Find Unique Solution to the Basis B, solving $x_B = B^{-1}b$

Finding Initial BFS

3 LP Cases when Finding Initial BFS:
All $\leq$ constraints: introduce slack variables to each constraint & easily find BFS by setting slacks to BVs
Some $\leq$ & some $=$ constraints: problem may be infeasible - can use slack variables you have - may need to run Phase 0
All $\geq$ constraints: introduce excess variables to each constraint - cannot choose all excess as BVs jointly - must run Phase 0

Unbounded & Infeasible LPs

Unbounded: (objective can be decreased indefinitely)
If no non-negative ratios exist when performing ratio test
If NBV exists with positive objective coefficient & all non-positive entries in column (in Tableau)
Infeasability: (no solution exists to satisfy all constraints)
No Initial BFS can be found
BFS found in Simplex with a negative BV & no pivoting can eliminate negativity

Pivoting

Idea: Move between vertices in direction that is most promising (gives most decrease per unit of movement) - until stopped by most binding constraint (given smallest non-negaitve ratio)
Transforms basis by removing a BV x_p & adding a NBV x_q
Variable into Basis (x_q): NBV with largest positive objective coefficient (if tie - smallest index) (largest negative r)
Variable out of Basis (x_p): Smallest Non-Negative Ratio (value of BV / coefficient of entering variable in BV row)
Note: Ignore Ratios with 0 / Negative Coefficient
Operation:
Divide every element of pivot row by pivot element - pivot row normalised & pivot element becomes 1 (replace x_p with x_q)
For each other row - eliminate all non-zero coefficients in pivot column by subtracting pivot row
Leaves pivot column with 1 in pivot row & 0 in other rows
Repeat Pivoting until all NBVs with negaitve coefficients in objective row - positive reduced cost

Fundamental Theorem of LP

Theorem 1:
For an LP in Standard Form with rank(A) = m:
If there exists a FS, there exists a BFS
If there exists optimal solution, there exists optimal BFS
To find a optimal solution - check the vertices of FR
Partitioning:
$Ax = Bx_B + Nx_N$ & $c^T x = c_B^T x_B + c_N^T x_N$
B: includes columns with indices in I (Basic)
N: includes columns with indices not in I (Non-Basic)
Basic Representation:
Expressing the Objective Function & BVs as linear function of the NBVs
$z = c_B^T B^{-1} b + (c_N - N^T B^{-T} c_B)^T x_N$
$x_B = B^{-1}b - B^{-1}Nx_N$
Shows how z & x_B change when NBVs increase
Setting $x_N = 0$:
Basic Solution: $x = (x_b, x_n) = (B^{-1}b, 0)$
Objective Value: $z = c_B^T B^{-1} b$
Optimality & Feasibility:
Reduced Cost Vector: $r = c_N - N^T B^{-T} c_B$
r contains the reduced costs associated with each NBV
r change in objective when NBVs are set to non-0 to obtain BS (i.e. quantity gained by setting NBVs to non-0/moving from one vertex to another)
Optimalality: all r positive (for minimisation) & objective increases if any NBV set to non-0 (i.e. put into basis & become BVs) Any r negative: setting that NBV to non-0 reduces objective value (improves minimisation)
Find BFS with lower objective: increase NBV with negative r
Feasibility: solution satisfies all constraints (including non-negativity)
& $x_B = B^{-1}b \geq 0$
Theorem 2:
If there exists a feasible solution & objective function is bounded - there exists an optimal solution
Geometrically:
BFSs at Vertices of Feasible Region
Optimum always achieved at a Vertex/BFS
Turning constraint on corresponds to a variable being set to 0

Simplex

Given LP in Standard Form:
A finite subset of Feasible Solutions are BFSs
BFS: vertex of the FR (& at least one optimal BFS)
Each BFS associated with a Basic Representation
BR gives reduced costs r - indicates best NBV to increase to minimise objective
Simplex Idea: iterate from vertex-to-vertex until optimal
Simplex Tableau: (for a given Basic Representation)
Rearrange BR: variables on LHS & constants on RHS

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

RHS of objective row: objective value of current BS
RHS of BV rows: value of BV at current BS
Coefficients of NBVs in objective: negative reduced costs
All NBVs have negative objective coefficients: optimal
Columns of BVs have all 0s & 1 in their row
Current BS feasible if & only if all BV RHS non-negative

Dual Simplex

Apply Simplex to Dual - Operating Directly on Primal Tableau
Simplex applied to Dual: $\max\ b^T y$ s.t. $A^T y + w = c,\ w \geq 0$
Simplex: enforces primal feasibility & steps to primal optimality
Dual Simplex: enforces dual feasibility (primal optimally) & steps to dual optimally (primal feasibility)
In Primal Tableau, starts Optimal & Infeasible - moves to Feasible & Optimal **Dual Basis:**
Fixing Primal Basis B - Dual Basis $\hat{B}$:

$$\hat{B} = \begin{bmatrix} B^T & 0 \\ N^T & I \end{bmatrix} \quad \hat{B}^{-1} = \begin{bmatrix} B^{-T} & 0 \\ -N^T B^{-T} & I \end{bmatrix}$$

If $\hat{B}$ feasible for (D) - then B optimal for (P)
Proof: $\hat{(B)}$ feasible for (D) implies $w_N \geq 0$ i.e. $w_N = -N^T B^{-T} c_B + C_N \geq 0 \leftrightarrow r \geq 0$
If $\hat{B}$ optimal for (D) - then B feasible for (P)
Proof: $\hat{B}$ optimal for (D) implies dual $r \leq 0$ (as maximisation)

$$0 - \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} B^{-1} & -B^{-1}N \\ 0 & I \end{bmatrix} \begin{bmatrix} b \\ 0 \end{bmatrix} = -B^{-1}b < 0 \iff$$

(P) feasible
To resolve problem for different constraints - can reuse initial BFS to Dual (no need to run phase 0 again for Dual)
Running Dual Simplex on Primal Tableau:
Identical Steps with exception of:
Initial Point $\rightarrow$ Start from Point with a Positive Reduced Cost - Ensuring Optimality (rather than Feasibility) (as point has to be optimal rather than feasible $\rightarrow$ constraints do not have to be satisfied - therefore can reverse inequalities & add slack variables - which do not satisfy non-negativity - to easily find Initial BFS - as long as reduced cost remains positive)
Variable out of Basis $\rightarrow$ Row Corresponding to Most Negative RHS (worst w.r.t Primal Feasibility)
Variable into Basis $\rightarrow$ Smallest Positive Ratio over Columns (Coefficient in Objective / Coefficient in Leaving BV Row) $\rightarrow$ Most Binding Constraint (if tie - smallest index)
Halting Test $\rightarrow$ Test for Feasibility (rather than Optimality)

Phase 0 & Phase 1 Simplex

Given LP in Standard Form:
Phase 0:
Consider augmented problem with objective $min\ \zeta = \epsilon$ (minimise sum of ϵ's)
Add an ϵ (artificial variable) to constraints without slack variables
Set slack variables & ϵ's as BVs (NBVs = 0)
Find BR for $\zeta = \epsilon$ (using constraints to rewrite ζ as a function of NBVs)
Form a Tableau from BR & constraints to minimise ϵ (with z & ζ row)
z : original objective & ζ : augmented objective
$\rightarrow$ keeping z row allows for easily finding BR of Initial BFS
Perform Pivoting Operations (considering augmented problem ζ)
If RHS of $\zeta = 0$ ($\zeta = \epsilon = 0$) $\rightarrow$ Original LP is feasible
Note: if some ϵ (=0) in BVs of BFS (i.e. degenerate point) - BFS does not represent original problem & need to pivot ϵ out & find another degenerate point (ϵ cannot be in the basis of the original LP)
When $\zeta = \epsilon = 0$ & no ϵ in BVs - BFS found is an initial BFS to original LP
(Tableau objective row has all 0s, 1 in z, & -1 in ϵ's)
Otherwise, (some ϵ non-0) original problem infeasible

Phase 1:
Initial BFS given by Phase 0 (use Tableau for Phase 0 without ϵ & ζ)
After Phase 0 - z row will be Basic Representation of NBVs
Pivot & stop at optimal solution (positive reduced cost & all NBVs with negative coefficient in objective row)

Degeneracies

Degenerate BS:
1 or more BVs are 0 (giving more than $m - n$ 0s)
Overlapping constraints & multiple index sets/basis generate same BS
Simplex: NBV to non-0 (constraint off) & BV to 0 (constraint on) may be overlapping constraints & stays at same vertex
Finite Termination:
If all BFSs are non-degenerate - Simplex terminates after finite steps, with either an optimal solution or proof of unboundedness
(Strictly Decreasing Objective Function)
If some degenerate BFSs - may fail convergence
Simplex:
If there is a degenerate BV - 2 Index Sets give the same BFS & different BR
Pivoting on a degenerate BV (0 RHS of Tableau): new BFS identical to previous BFS (same vertex)
Objective Value unchanged & Finite Termination breaks down
Simplex = [Sequence of Degenerate Pivots] Non-Degenerate Pivot [Sequence of Degenerate Pivots]
Sequences of Degenerate Pivots are finite if no I is repeated
Cycle: Pivoting on a Degenerate Pivot causes return to the same Index Set (& never terminates)
Bland's Rule: select x_q (entering basis) as the leftmost NBV for which the objective coefficient is positive
May no longer move towards local optimum (slows down)
Replacement: can replace RHS 0 with very small $\epsilon \leq 10^{-3}$

Nonlinear Programs

Min-Max: Worst-Case Optimisation (Robust in Worst-Case)
$y_i(x) = c(i)^T x + d(i)$ & $\phi(x) = max\ y_i(x)$
$min\ \phi(x)\ s.t. Ax = b, x \geq 0$
Minimising Maximum Value in a set of Objective Functions
LP Formulation: $min\ z\ s.t. z \geq c(i)^T x + d(i), Ax = b, x \geq 0, z \epsilon \mathbb{R}$ Turning Set of Objectives into Constraints
Min-Min: Best-Case Optimisation (Optimal in Best-Case)
$\phi(x) = min c(i)^T x + d(i)$
$min\ \phi(x)\ s.t. Ax = b, x \geq 0$
Minimising the Minimum Value in a set of Objective Functions
No LP Reformulation is required
Solution: take minimum value of each objective function & select the smallest
Order of Minimisation Operations Interchangeable
Fractional LPs:
Objective Ratio of Two Linear Functions & Constraints Linear
$min\ \frac{\alpha_0 + \alpha_1 x_1 + \ldots + \alpha_n x_n}{\beta_0 + \beta_2 x_2 + \ldots + \beta_n x_n}\ s.t. Ax = b, x \geq 0$
Assuming Feasible Set is Bounded & Denominator of Objective Function is strictly positive - can linearise program
Homogenisation:
Introduce new variables $y_i \geq 0,\ i = 1..n,\ y_0 > 0$
Set $x_i = \frac{y_i}{y_0}$ & can homogenise FLP:
$\min \frac{\alpha_0 y_0 + \alpha_1 y_1 + \alpha_2 y_2 + \ldots + \alpha_n y_n}{\beta_0 y_0 + \beta_1 y_1 + \beta_2 y_2 + \ldots + \beta_n y_n}$ s.t. $b_i y_0 - \sum_{j=1}^n a_{ij} y_j = 0,\ \forall i = 1, \ldots, m,\ \ y_0 > 0, y_1 \geq 0, \ldots, y_n \geq 0$
For feasible $(y_0 \cdots y_n)$ $\lambda(y_0 \cdots y_n)$ feasible & same objective - find λ such that denominator = 1 (normalising to unity)
Solve Normalised Problem (LP):
$\min \frac{\alpha_0 y_0 + \alpha_1 y_1 + \alpha_2 y_2 + \ldots + \alpha_n y_n}{\beta_0 y_0 + \sum_{j=1}^n \beta_j y_j}$ s.t. $\beta_0 y_0 + \sum_{j=1}^n \beta_j y_j = 1,\ b_i y_0 - \sum_{j=1}^n a_{ij} y_j = 0\ \forall i = 1, \ldots, m,\ y_0 > 0, y_1 \geq 0, \ldots, y_n \geq 0$

Duality

Form for Duality:
$\max\{c^T x : Ax \le b, x \ge 0\},\ c, x \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m.\quad (P)$
$\min\{b^T y : A^T y \ge c, y \ge 0\}$, c, A, b as in (P) and $y \in \mathbb{R}^m.\quad (D)$
No. Decision Variables in Primal = No. Constraints in Dual (& vice versa)
Cost Coefficients in Primal = Constraint RHSs in Dual (& vice versa)
Matrix A of Constraints - Transposed
Dual of (D) = (P) - **Symmetric** (either (P)/(D) considered Primal/Dual)
Weak Duality:
Assuming (P) & (D) Feasible: $c^T x \le b^T y$
For any x & y in FR - (D) Cost Upper Bound on (P) Cost
Strong Duality:
Assume (P) feasible & finite optimum - let B be optimal basis for (P) with optimal basic solution (x^*_B, x^*_N):
$y^* = (B^{-1})^T c_B$ **optimal solution for (D)**
$c^T x^* = b^T y^*$ **(P) & (D) same objective**
& vice versa - can find solution to (P) from (D)
$\Pi = (B^{-1})^T c_B = y^*$ & Shadow Prices of Primal = Optimal Solution to Dual (Π: how changing constraint RHSs affects optimal objective)
Primal Finite Optimal $\leftrightarrow$ Dual Finite Optimal (Strong Duality - (P) feasible & bounded - (D) with same objective)
Primal/Dual Unbounded $\rightarrow$ Dual/Primal Infeasible (Weak Duality - if (P) unbounded $\rightarrow \infty$ & if (D) unbounded $\rightarrow -\infty$ & no longer upper bound - so (D) must be infeasible)
Primal/Dual Infeasible $\rightarrow$ Dual/Primal Unbounded or Infeasible (Unbounded possible as above - Infeasible possible if constraints of Primal & Dual cannot be satisfied)
Indirect Way: Bring LP to form of either (D) or (P)
Replace variables $x_i \epsilon \mathbb{R}$ with $(x_i^+ - x_i^-)$ where $x_i^+, x_i^- \ge 0$
Replace equality constraints with two inequality constraints
Change constraint ($\le, \ge$) or objective by $\times$ -1 if needed
Obtain Dual according to definition: (P) $\rightarrow$ (D) / (D) $\rightarrow$ (P)
Simplify Dual Problem:
Replace variable pairs $y_i, y_j \ge 0, i \ne 0$ that occur in all functions as $\alpha y_i - \alpha y_j$ by single variable $y_k \epsilon \mathbb{R}$
Replace matching inequality constraints by equality constraints
Direct Way:
Apply Duality without detour via (P) or (D)
For every Primal Constraint, create a Dual Variable
For every Primal Variable, create a Dual Constraint
Dual Coefficient Matrix is A^T
Former RHSs b become new costs
Former costs c become new RHSs
If Primal is a Max Problem - Dual is a Min
Primal Constraints $[\ge, =, \le] \rightarrow$ Dual Variables $[y_i \le 0, y_i \epsilon \mathbb{R}, y_i \ge 0]$
Primal Variables $[x_j \ge 0, x_j \epsilon \mathbb{R}, x_j \le 0] \rightarrow [\ge, =, \le]$
If Primal is a Min Problem - Dual is a Max
Primal Constraints $[\ge, =, \le] \rightarrow$ Dual Variables $[y_i \ge 0, y_i \epsilon \mathbb{R}, y_i \le 0]$
Primal Variables $[x_j \ge 0, x_j \epsilon \mathbb{R}, x_j \le 0] \rightarrow [\le, =, \ge]$
(Free Variables in Primal become Equality Constraints & Vice Versa)

IP Neural Networks

Optimisation Problems on Trained NNs:
Given a trained NN, image $\hat{x}$, & label j - is there an adversary k (misclassification label) within some small perturbation of this image (size of perturbation measured by norms - such as l_1 or l_∞)
Verification: is there an adversary labelled k within a given perturbation
Optimal Adversary: what image within perturbation maximises the prediction difference
Minimally Distorted Adversary: what is the smallest perturbation over which NN predicts label k
Lossless Compression: can I safely remove NN nodes/layers
Decision-Making over a Learned ReLU Neural Network:
ReLU: output is max of 0 & weighted combinations of inputs + bias terms
x^{l-1}: input of layer & x^l: output $\rightarrow x^l = max(0, (w^T x^{l-1} + b))$
Verification:
$\max_{x\in\mathcal{X}} f_k(x^{(L)}) - f_j(x^{(L)})$ s.t. $x_i^{(\ell)} = \max(0, (w_i^{(\ell-1)})^T x^{(\ell-1)} + b_i)\quad \forall \ell \in \{1, \dots, L\}, i \in \text{Node}^{(\ell)}$
f_k & f_j correspond to the $k-th$ & $j-th$ elements of the NN output layer L & X defines domain of perturbations
Maximises the difference between Misclassified Label k & Label j for the NN Output Layer L - subject to ReLU Function & across domain of all possible perturbations
Trying to find example of misclassification (if there is a feasible point with positive objective value) $\rightarrow$ want no feasible solution for verification
Max term appears for as many layers within the Feedforward Neural Network
Can replace equality constraint with a $\le$ & $\ge$ equality constraint
Big-M Formulation of a Learned ReLU NN:
$x^{(\ell)} \ge (w^T x^{(\ell-1)} + b)$, $x^{(\ell)} \le (w^T x^{(\ell-1)} + b) - (1-\sigma)LB^0$, $0 \le x^{(\ell)} \le \sigma UB^0$, $\sigma \in \{0, 1\}$
If $\delta = 1 \rightarrow x^l = (w_T x^{l-1} + b)$ - want UB^0 big enough to permit $0 \le x^{(\ell)} \le \sigma UB^0$ to be slack
If $\delta = 0 \rightarrow x^l = 0$ - want LB^0 to be small enough to permit $x^{(\ell)} \le (w^T x^{(\ell-1)} + b) - (1-\sigma)LB^0$ to be slack

Game Theory

Game: multi-agent optimisation problem
Each player takes a decision from a set of actions $x_i \epsilon X_i$ (i.e. a feasible set for each player) & player i receives a payoff $J_i(x_1, \dots, x_n)$ (i.e. objective function for each player)
Two-Person Zero-Sum Games with Finite Actions:
Zero-Sum: objective functions of two players sum to 0
Finite Actions: feasible set countable & has finitely many choices
Row Player (RP): chooses one of m row strategies
Column Player (CP): chooses one of n column strategies
Payoff Matrix: Matrix containing payoffs (CP to RP) for each possible action - if RP plays i & CP plays j - CP plays a_{ij} to RP (representing whole action space of two objectives together)
Each entry a utility to RP & cost to CP (CP $\rightarrow$ RP)
RP $\rightarrow$ maximising & CP $\rightarrow$ minimising
Assumptions: Each player knows the game setting (available strategies to RP & CP + payoff values)
Both players simultaneously select strategies (not knowing opponent selection)
Each player chooses strategy that enables them to do best - reasoning as if opponent could anticipate their strategy (worst-case opponent)
Both players are rational (maximise their utility & no compassion)
Dominance:
Dominant Row/Column Strategy: A Row/Column Strategy dominates if it is better than all other row/column strategies for all strategies the opponent could take. **Dominant strategies are always played**
Dominant Strategy Equilibrium: if a repeats removal of dominant strategies leads to a game where each player has just one strategy left - strategy pair is a dominant strategy equilibrium (associated payoff = value of the game)
Nash Equilibrium in Pure Strategies:
Assumption: each player chooses a strategy that enables them to do best in face of their worst-case opponent (highest cost for CP or lowest utility for RP)
RP selects the minimum value (worst-case) from each row (α_i) & selects the maximum of these (as RP maximising utility)
CP selects the maximum value (worse-case) from each column (β_i) & selects the minimum of these (as CP minimising cost)
Nash Equilibrium: a strategy pair such that no player has an incentive to unilaterally deviate from their chosen strategy (2 players cannot deviate without risking incurring a loss in worst-state) ($max_i \alpha_i = min_j \beta_j$)
Nash Equilibrium may not always exist in Pure Strategies
Value of the Game = Payoff ($max\ min_j a_{ij} = min\ max_i a_{ij}$)
Nash Equilibrium in Mixed Strategies: (Always Exists)
RP & CP assign probabilities to their strategies (RP plays strategy i with probability p_i & CP plays strategy j with probability p_j)
If $p = 1$ or q_k = 1 - k is a pure strategy
Payoff of Mixed Strategy = $V(p, q) = \sum_{i=1}^m \sum_{j=1}^n p_i q_j a_{ij}$ (expected value - sum of all actions $\times$ their probabilities)
RP seeks probabilities to maximise & CP to minimise payoff
Mixed Nash Equilibrium: Pair of Mixed Strategies (p^*, q^*) such that the RP cannot increase their utility & CP cannot decrease their cost by changing strategy $(V(p, q^*) \le V(p^*, q^*) \le V(p^*, q))$
CP: (trying to select optimal q_j to minimise payoff for worst-case opponent)
Assuming RP selects optimal p_i's for any choice of q_j's - CP selects q_j to minimise the maximum expected payoff
$V_{CP} = \min_{q_1,..,q_n} \max_{p_1,..,p_m} \left(\sum_{i=1}^m \sum_{j=1}^n p_i q_j a_{ij}\right)$
s.t. $\left(\sum_{j=1}^n q_j = 1,\ \sum_{i=1}^m p_i = 1,\ q_j \ge 0,\ p_i \ge 0\right)$
LP Formulation:
$V_{CP} = \min_{\tau, q_1,..,q_n} \tau$
s.t. $\tau \ge \sum_{j=1}^n q_j a_{ij}\ \forall i = 1, .., m \quad \sum_{j=1}^n q_j = 1 \quad q_j \ge 0$
RP: (trying to select optimal q_i)

Cutting Planes

LP Relaxation: LP - replacing all integer variables in an ILP with continuous variables (removing integer constraint) - has better/same optimal as ILP
Cutting Plane: cut off non integer points from the feasible region without excluding any integer feasible points
Cutting Plane Algorithm:
Write the ILP in Standard Form (Step 0) & Solve the LP Relaxation (Step 1)
If the resulting optimal solution x^* is an integer $\rightarrow$ stop - optimal found
Otherwise $\rightarrow$ generate a cut (a constraint satisfied by all feasible integer solutions - but not by solution x^* with non-integer components)
Add cut to the LP Relaxation & go back to step 1 (tightening LP Relaxation - restricting feasible set without excluding the optimal solution to ILP)
Note: after adding a cut to LP - need to -excess to get back to equality constraint - & as solution found no longer feasible with cut - must run Phase 0 to find new Initial BFS
Phase 0: augmented objective $\zeta = \epsilon$ - add ϵ to cut to deal with excess - find new BR for $\zeta = \epsilon$ (rewriting cut constraint) - & create Tableau (taking previous Tableau & adding ζ/excess/ϵ columns + ζ & ϵ rows)
CP Algorithm Terminates after a Finite No. of Iterations
Gomory Cut:
If LP Relaxation gives non-integer optimal solution - must be a BV that is non-integer - row in Tableau with non-integer RHS
Row: $x_i^* + \sum_{j\notin I} y_{ij} x_j^* = y_{i0},\ f_j := y_{ij} - \lfloor y_{ij} \rfloor,\ f := y_{i0} - \lfloor y_{i0} \rfloor.$
$\sum_{j\ne i} f_{ij} x_j \ge f.\quad$ (GC) $\quad x_j$ are the NBVs
Gomory Cut eliminates non-integer parts of variables in the optimal solution of the LP relaxation - excludes the current non-integer solution by adding a new constraint that the solution cannot satisfy & forces the next iteration of cutting plane algorithm to find a new solution closer to being an integer
Note: Floor of Negatives (e.g. Floor of -0.3 = -1 & $f_j = -0.3 - -1 = 0.7$)
Knapsack Cover Cuts:
Cover: Set S of items in a Knapsack Problem where $\sum_{j\in S} w_j > W$
Knapsack Cover Cut (given Cover S): $\sum_{j\in S} x_j \le \mod S - 1$
Minimal Cover Constraint: For all subsets T in S: $\sum_{j\in T} w_j < W$

Integer Programming

Pure ILP: all decision variables (incl. slack/objective) required as integers
Mixed ILP: includes decision variables that are both integer & real
MILP Standard Form: similar to LPs & all slack/excess continuous
Pure ILP Standard Form: slack/excess variables integers - apply LP standard form transformations (except + slacks/excess) - scale equations such that all coefficients are integers - insert integer slack/excess variables
Combinatorial Optimisation: finding an optimal object in finite set
Knapsack Problem: n items of weight w_j & Knapsack of capacity W - item j has value v_j - maximise total value of Knapsack
$\max_x z = \sum_{j=1}^n v_j x_j$ s.t. $\sum_{j=1}^n w_j x_j \le W,\ x_j \in \{0, 1\}\,\forall j \in \{1, \dots, n\}$
Bin-Packing Problem: (multiple knapsacks) n items of weight w_j & k bins of capacity W - $x_{ij} = 1$ if item j assigned to bin i, otherwise 0 - minimise no. bins needed to store all items
$\min_{x,y} z = \sum_{i=1}^k y_i$ s.t. $\sum_{j=1}^n w_j x_{ij} \le W y_i,\ \sum_{i=1}^k x_{ij} = 1\,\forall j \in \{1, \dots, n\},\ x_{ij}, y_i \in \{0, 1\}\,\forall i \in \{1, \dots, k\}, \forall j \in \{1, \dots, n\}$
Logical Operations: can be modelled on constraints with integer variables
Either-Or: (Big-M Constraints)
$a_1^T x \le b_1 + M\delta,\quad a_2^T x \le b_2 + M(1-\delta),\quad \delta \in \{0, 1\}$
δ: turns constraint on/off
$\delta = 0 \rightarrow$ first constraint must be true (& M large enough such that second constraint satisfied for any x)
$\delta = 1 \rightarrow$ second constraint must be true (& M large enough such that first constraint satisfied for any x)
Suitable $M \rightarrow$ upper bound on $x\times$ element-wise max of a1/a2
k-out-of-m: (satisfying at least k out of m constraints)
$a_1^T x \le b_1 + M\delta_1, \dots, a_m^T x \le b_m + M\delta_m,\ \sum_{j=1}^m \delta_j \le m-k,\ \delta_j \in \{0, 1\}, \forall j \in \{1, \dots, m\}$
Sum of $m\ \delta$ must be $\le m-k$ - as maximum of $m-k\ \delta$ can take value 1 (so a minimum of k must take value 0 & at least k constraints satisfied)
Finite-Valued Variables:
Assume variable x_j only takes one of a finite no. of values: $x_j \epsilon p_1, \dots, p_m$
Introduce variables $z_{j1} \dots z_{jm} \epsilon \{0, 1\}$ & add constraint $z_{j1} + \dots + z_{jm} = 1$ & replace $x_j = p_1 z_{j1} + \dots + p_m\text{-}jm$ in objective & all constraints

Sensitivity Analysis

Compute LP Solution with some parameters & then analyse what happens when parameters changed **(Analysing Sensitivity to Parameter Values)**
Value Function:
$v(p_i)$ - expresses optimal value of LP as function of the parameter p_i
$v(p_i)$ vs p_i: Non-Increasing, Convex, Piecewise Linear Function (Gradient = Π)
Considering $p = b + \epsilon$ (reference parameter + perturbation)
Value Function: $v(p) = minz = c^T x\ s.t.\ Ax = p, x \ge 0$
Reference Problem (Original LP): $minz = c^T x\ s.t.\ Ax = b, x \ge 0 = v(b)$
Shadow Prices:
Π: how optimal value (Primal) changes with changed RHS parameters
$\Pi = (B^{-1})^T c_B = y^*$ - Shadow Prices of Primal Problem = Values of Dual Variables at Optimality - $B = B(I)$ is the optimal basis
Information about the sensitivity of the value function $v(p)$ at $p = b$
Behaviour of Value Function:
With Reference Problem & Π - can find any Value Function
$v(p) = v(b) + \Pi^T(p - b), \forall p \in \mathbb{R}^m$ with $B^{-1}p \ge 0$
If x_B remains feasible ($B^{-1}p \ge 0$) B remains optimal basis for new parameter & r is not affects changing b to p & can find optimal value at p
If x_B does not remain feasible - can only find a lower bound of $v(p)$ **Tableau:** For a $\le$ constraint - Π objective entry of first slack introduced
For a $\ge$ constraint - Π negative of objective entry first slack introduced

Branch & Bound

Complete Enumeration: Loop through all values of integer variables solving LPs in the continuous variables (given integer variables are few)
Branch & Bound Notation:
P_i : i^{th} subproblem - $x^*(P_i)$ optimal solution for LP Relaxation of P_i - $c^T x^*(P_i)$ optimal value for LP Relaxation of P_i - OPT optimal objective function value thus far (= ∞ at beginning for minimisation) - x_{OPT} solution producing best OPT - x^* feasible for MILP if satisfies integrality constraints
Branch & Bound Algorithm (MILP):
Initialisation: Initialise $OPT = \infty$ (minimisation)
Solve LP Relaxation of original problem P_0 ($\rightarrow x^*(P_0)$)
If Optimal Solution satisfies all integrality constraints of MILP (feasible) $\rightarrow$ STOP (OPT = $c^T x^*(P_0)$ & $x_{OPT} = x^*(P_0)$)
Branch: choose non-integer $x_p^* \in x^*(P_0)$ that must be an integer in P_0 & create two subproblems P_1&P_2 adding to P_0 constraints $x_p \le \lfloor x_p^* \rfloor$ & $x_p \ge \lceil x_p^* \rceil$
Bound: solve LP Relaxation for P_1 & P_2 - if $c^T x^*(P) < OPT$ & $x^*(P)$ feasible (satisfies all integer constraints) $\rightarrow$ update OPT = $c^T x^*(P)$ & $x_{OPT} = x^*(P)$
Pruning: Branching STOPs if: Optimal Value ¿ OPT, Optimal Value Feasible & satisfies all integer constraints, Subproblem Infeasible (Prunes/Eliminates Branches that cannot yield a better solution that the OPT found so far)
Repeat Branch & Bound Steps - Stopping if Pruning Conditions Met Output: OPT = ∞ if P_0 infeasible - OPT ¡ ∞ if P_0 feasible & OPT optimal
B & B + Cover Cuts can be used together (Cover Cuts are useful when the Relaxation gives solution far from integer feasibility as they can cut down the Feasible Space & then B & B solves for smaller space)