ingredients of a general optimisation problem

Def: convex function

Def: convex set, operations...

convexity preserving operations for functions

Def: extreme point

fundamental theorem of linear programming

Def: feasible solution / feasible region

Def: objective value / optimal value

Def: unbounded problem

standard equality form of a linear problem

standard inequality form of a linear problem

basis of an LP

|

|

|

Problem where arbitrary good solutions are feasible

- unbounded feasible region
- objective function increases towards unbounded direction

maximise $\mathbf{c}^T\mathbf{x}$
subject to $\mathbf{Ax} = \mathbf{b}$
and $\mathbf{x} \geq \mathbf{0}$

x, c: n•1, **A**: m•n, **b**: m•1
n: number of variables
m: number of constraints

maximise $\mathbf{c}^T\mathbf{x}$
subject to $\mathbf{Ax} \leq \mathbf{b}$
and $\mathbf{x} \geq \mathbf{0}$

x, c: n•1, **A**: m•n, **b**: m•1
n: number of variables
m: number of constraints

A basis of an LP in SEF with matrix **A** is a set of n indices $\{b_1, \ldots, b_n\}$ such that the corresponding columns of **A** are linearly independent.

---

point in a corner of a set (for polyhedron: unique solution of n equations)

$$x = \lambda x_1 + (1-\lambda) x_2$$
$$\Rightarrow \lambda = 0 \text{ or } 1$$

|

If a linear problem has a feasible (optimal, respectively) solution, it also has a feasible (optimal, resp.) extreme point solution.

|

- a solution that satisfies all constraints
- set of all feasible solutions

|

- value of the objective function for a feasible solution
- objective value of the optimal solution

- objective function: f(x) either to maximise or minimise
- decision variables: x_1, x_2, …
- constraints: $g_i(x) \geq b_i$
  ≥, ≤ and =
- parameters?

$$f(\lambda x_1 + (1-\lambda) x_2) \leq$$
$$\lambda f(x_1) + (1-\lambda) f(x_2)$$
$$\forall \lambda \in [0,1]$$

Function graph lies above tangent.
The value of the average is smaller than the average of the values ;-)

- A set is convex if all averages of elements are also in the set
- intersection of convex sets is convex
- a polyhedron is convex
- a set described by convex constraints $g_i(x) \leq b_i$ is convex

- A positive linear combination of convex functions f, g is convex.
- max{f, g} is convex.

|

|

|

basic solution	Theorem: relationship between an extreme point solution and a basis	Def: entering variable	Def: leaving variable
tableau at the beginning of the simplex method	tableau at the end of the simplex method	Def: tableau	ratio test for simplex and revised simplex method
Talking so much about the simplex method, what's a *simplex* anyway?	two-phase simplex method	revised simplex method: How to compute reduced costs?	revised simplex method: How to compute the minimum ratio test?

A *simplex* is an n-dimensional analogue of a triangle. It's the complex hull of n+1 affinely independent points.
A 0-simplex is simply a point. Higher dimensions are a line segment, triangle, tetrahedron, pentachoron, ...
The feasible region of an LP is a simplex.

1. Transform the problem to SEF with non-negative RHS.
2. For each row that has no unit column, introduce an artificial variable.
3. Set the obj. function to -∑(artificial vars)
4. Optimise until all artificial vars = 0. If this is not possible, the problem is infeasible.

First row of Tableau:
$\mathbf{yD} - \mathbf{c} = -\bar{c}$

y: $\mathbf{B^{-1}c_B}$
D: (**AI**) = augmented constraint matrix
c: original cost vector (0 for slack variables)
$\mathbf{B^{-1}}$: slack variable coefficients

RHS: $\hat{\boldsymbol{b}}_j = \boldsymbol{B}^{-1} \cdot \boldsymbol{b}$
The j-th column of the tableau at any time:
$(\hat{\boldsymbol{D}})_j = \boldsymbol{B}^{-1} \cdot (\boldsymbol{D})_j$.

D: (**AI**) = augmented constraint matrix
b: original right hand side
$\mathbf{B^{-1}}$: slack variable coefficients

-c	**0**	0
A	**I**	**b**
decision variables	slack variables	RHS

c: reduced costs
A: constraints (m•(n-m))
I: identity matrix (m•m)
b: right-hand-side coefficients (positive)

y*A-c	**y***	Z*
SA	**S**	**b***
decision variables	slack variables	RHS

y*: shadow costs
Z*: optimal value = **y*b**
S: Slack var. matrix = $\mathbf{B}^{-1}$
now the entire first row is positive (except Z* maybe)

A tableau (for a basis B) is a tabular representation of the objective function and the constrains, satisfying

- Obj. function expressed in terms of nonbasics,
- Rows corresponding to B form an identity matrix.

Let $(\mathbf{A})_j$ be the tableau column of the entering variable.
$\forall i \in [1, m]$ such that $a_{ij} > 0$, calculate $d_i = \frac{b_i}{a_{ij}}$. Take as leaving variable the one with smallest d_i.
If no d_i exists, the problem is unbounded.

A basic solution determined by a basis B is the unique solution of **Ax = b** with $x_i = 0 \, \forall i \notin B$.

The following are equivalent:

- **x** is an extreme point solution of an LP in SIF
- **x** (augmented) is a basic feasible solution of the augmented problem in SEF.

A non-basic variable with positive reduced cost. It is increased to a positive value (except in degenerate cases), thus it enters the basis.

A basic variable with minimal ratio b_i / A_{ij} (where x_j is the entering variable). Increasing the entering variable forces it to become zero, thus to leave the basis.

dual problem (SIF)

Theorem: weak duality

Theorem: strong duality

Theorem: symmetry property of LP duality

Theorem: complementary slackness, a.k.a. complementary basic solutions property

duality: feasible, bounded and unbounded problems

duality: equality constraints, unbounded variables

dual two-phase method

sensitivity analysis: change in the right-hand side $\boldsymbol{b} \rightarrow \boldsymbol{b} + \boldsymbol{\Delta b}$

sensitivity analysis: change in the objective function $\boldsymbol{c} \rightarrow \boldsymbol{c} + \boldsymbol{\Delta c}$

sensitivity analysis: adding a constraint

sensitivity analysis: adding a variable

Affects feasibility, but not dual feasibility.

$\bar{c}$ unchanged $\Rightarrow$ B remains optimal (if it's still feasible)

If B becomes infeasible (negative RHS), use dual simplex method to compute the new B.

Affects dual feasibility, but not feasibility.

If basic components of **c** are changed, we need to re-express **c** in terms of non-basics. Still positive? Then B remains optimal.

Otherwise use simplex method to re-optimise.

Adding a ≤ constraint also adds a basic slack variable with reduced cost 0.

If the constraint is satisfied, we're fine.

Otherwise B is still dual feasible, so use the dual simplex method to re-optimise.

Adding a non-negative variable also adds a ≥ constraint to the dual problem.

Primal feasibility is not affected.

If the new variable's c is negative, we need to re-optimise.

Each basic solution **x** has a corresponding basic solution **y** with same objective value.

x_i basic $\Leftrightarrow$ $(\mathbf{A_i})^T \mathbf{y} = c_i$ (equality, the i^{th} slack variable is zero)
$(\mathbf{A_j})^T \mathbf{y} > c_j$ $\Leftrightarrow$ $x_j = 0$ (j^{th} slack non-zero corresponds to non-basic x_j)

Primal	**Dual**
feasible & bounded	feasible & bounded
infeasible	infeasible or unbounded
unbounded	infeasible

Primal	**Dual**
max. $\mathbf{c^T x}$	min. $\mathbf{b^T y}$
≤ constraint	$y_i \geq 0$
= constraint	$y_i \in \Re$
≥ constraint	$y_i \leq 0$
$x_i \geq 0$	≥ constraint
$x_i \in \Re$	= constraint
$x_i \leq 0$	≤ constraint

Given: an LP in SEF having some negative RHS values
1. Set **c** = 0. This gives a dual feasible tableau for some LP'.
2. Use the dual simplex method to solve LP'. Optimality $\Rightarrow$ RHS non-negative
3. Found primal feasible basis! Reset **c** to original costs.

---

minimise $\mathbf{b^T y}$
subject to $\mathbf{A^T y} \geq \mathbf{c}$
and $\mathbf{y} \geq \mathbf{0}$

y, b: m•1, **A**: m•n, **c**: n•1
n: number of primal variables / dual constraints
m: number of primal constraints / dual variables

Every feasible dual solution **y** gives an upper bound for the primal objective value.

$$c^T x \leq y^T A x \leq y^T b$$

If **x*** is an optimal solution for an LP, then it's dual problem has an optimal solution **y*** with

$$c^T x^* = b^T y^*$$

The primal problem is the dual of its dual.

Therefore, all relationships between them must be symmetric ((x,c) and (y,b) are interchangeable).

graph terminology and definitions

number of nodes and edges in a graph

Def: (minimum) spanning tree

network flow terminology

residual network

Ford-Fulkerson algorithm (maximum flow)

duality and maximum flow

minimum cost flow problem formulated as LP

network simplex method: number of constraints, variables

Theorem: fundamental theorem of the network simplex method

solving a spanning tree given by basis B

network simplex method

n: number of nodes
m: number of arcs

n node constraints
+ m capacity constraints
⌁ one is redundant!
therefore m+n–1 constraints

2m variables x_{uv}, t_{uv}

Every collection of variables B determines a collection of arcs:

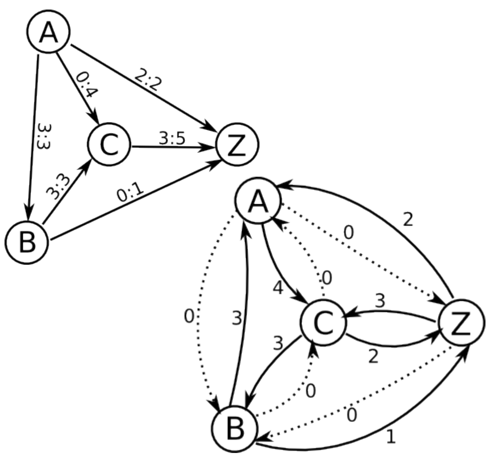
$$T_B = \{arcs\, uv \mid x_{uv} \in B \text{ and } t_{uv} \in B\}$$

Fundamental theorem:
B is a basis
⇔
T_B is a spanning tree

1. assign zero flow to all arcs where x_{uv} is non-basic.
2. assign maximum flow to all arcs where t_{uv} is non-basic (reverse arcs).
3. Since T_B is a tree, ∃ a leaf node with only one unsolved connection. Calculate the flow for this connection using the node's outflow.
4. Repeat step 3...

1. Start with a feasible basis, solve it's spanning tree.
2. Find a non-basic arc with negative cost, i.e. the sum of costs in its cycle is negative.
3. Increase the flow through that arc as much as possible.
4. Repeat 2-3...

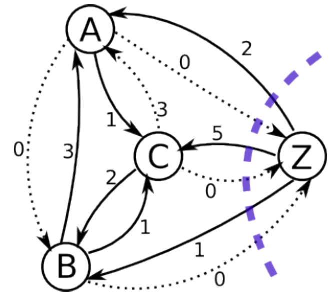
⌁ reverse arcs can be treated like normal arcs: reverse them and negate the cost.



Repeatedly search for an augmenting path in the residual network from the source to the sink, and use it.

Then update the (residual) network.

Stop when there is no more augmenting path.



cut: division of graph in two
size of a cut: sum of inflowing arc capacities
Duality: max. flow = min. cut

minimise $\sum_{arcs\, uv} x_{uv} c_{uv}$
subject to
$$\sum_{arcs\, uv} x_{uv} - \sum_{arcs\, vu} x_{vu} = b_u \;\forall\, nodes\, u$$
and $0 \le x_{uv} \le u_{uv} \;\forall\, arcs\, uv$

c_{uv}: cost of arc from u to v
u_{uv}: capacity of that arc
b_u: outflow of node u
x_{uv}: flow from u to v
t_{uv}: corresponding slack var.

---

graph: set of *nodes* (vertices) and *arcs* (edges)
arcs: *directed* or *undirected*
Arcs are *incident* to two *adjacent* nodes.
path: sequence of arcs
cycle: closed path
connected graph: ∃ a path between any pair of nodes
tree: connected network without cycles.

general planar graph:
n + e = a + c
in particular for a tree:
n = a + 1

n: number of nodes
a: number of arcs
e: number of eyes (closed surfaces)
c: number of connected components

Spanning tree: a subtree of a network connecting all the nodes. That is, only some arcs are removed from the network.

Minimum spanning tree: a spanning tree such that the sum of all arc weights is minimal.

arc *capacity*: maximum flow through an arc
arc *cost*: penalty per unit of flow through that arc
supply / *demand* / *transshipment* node: node with inflow < / > / = outflow
conservation of flow: in a network,
total inflow = total outflow