Optimisation

Shing Tak Lam

May 11, 2022

Contents

1 Definitions

In the context of this course, an optimisation problem has the form

$$\text{minimise } f \text{ subject to } x \in \mathcal{X}$$

or

$$\text{minimise } f \text{ subject to } x \in \mathcal{X}, h(x) = b$$

- f is known as the objective function.
- The components of x are known as decision variables.
- $h(x) = b$ is a functional constraint
- $x \in \mathcal{X}$ is a regional constraint

Definition 1.1 (Feasible)
For an optimisation problem, the feasible set is

$$\mathcal{X}(b) = \{x \in \mathcal{X} : h(x) = b\}$$

The problem is feasible if $\mathcal{X}(b) \neq \varnothing$, and bounded if $f(\mathcal{X}(b))$ is bounded below.

Definition 1.2 (Optimal)
$x^* \in \mathcal{X}(b)$ is optimal if it minimises f over $mcX(b)$. The value $f(x^*)$ is known as the optimal cost.

1.1 Slack

Sometimes, for the functional constraint we have an inequality instead of an equality. Slack can be used to transform the inequality into a regional constraint. In particular, given

$$\text{minimise } f \text{ subject to } x \in \mathcal{X}, h(x) \leq b$$

We can write this as

$$\text{minimise } f \text{ subject to } x \in \mathcal{X}, s \geq 0, h(x) + s = b$$

Therefore, we will now (unless otherwise specified) assume that the functional constraint is an equality.

2 Convexity

Definition 2.1 (Convex set)
$S \subseteq \mathbb{R}^n$ is convex if for all $x, y \in S$, $\lambda \in [0, 1]$, we have that

$$(1 - \lambda)x + \lambda y \in S$$

Definition 2.2 (Convex function)
For a convex set S, a function $f : S \to \mathbb{R}$ is convex if for all $x, y \in S$, $\lambda \in [0, 1]$, we have that

$$f((1 - \lambda)x + \lambda y) \leq (1 - \lambda)f(x) + \lambda f(y)$$

It is strictly convex if the inequality is strict on $(0, 1)$.

2.1 Conditions for convexity

Theorem 2.3 (First order conditions for convexity)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable. Then f is convex if and only if for all x, y, we have that

$$f(y) \geq f(x) + (y - x) \cdot \nabla f(x)$$

Proof. First suppose if f is convex. We consider the $n = 1$ case first. Then we have that

$$f(x + t(y - x)) \leq (1 - t)f(x) + tf(y)$$

Rearranging, we get

$$f(y) \geq \frac{f(x + t(y - x) - (1 - t)f(x))}{t} = f(x) + \frac{f(x + t(y - x) - f(x))}{t}$$

Taking $t \to 0$ we get that $f(y) \geq f(x)+(y-x)f'(x)$ as required. For the general case, let $g(t) = f(x+t(y-x))$. Then since f is convex, so is g, and $g'(t) = (y-x) \cdot \nabla(f(x+t(y-x)))$. Using the result for $n = 1$, we get that

$$f(y) = g(1) \geq g(0) + g'(0) = f(x) + (y-x) \cdot \nabla f(x)$$

Now suppose the first order condition holds. Let $x_t = x + t(y-x)$. Then we have that

$$f(x) \geq f(x_t) + t(y-x) \cdot \nabla f(x_t) \tag{1}$$
$$f(y) \geq f(x_t) - (1-t)(y-x) \cdot \nabla f(x_t) \tag{2}$$

Then $(1-t) \cdot (1) + t \cdot (2)$ gives the required result. □

Definition 2.4 (Hessian)
Define the Hessian of a function $f : \mathbb{R}^n \to \mathbb{R}$ by

$$(\nabla^2 f)_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

Theorem 2.5 (Second order condition for convexity)**.** Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable. Then f is convex if $\nabla^2 f$ is positive semidefinite.

Proof. Using the multivariate Taylor theorem, we have that

$$f(y) = f(x) + (y-x) \cdot \nabla f(x) + \frac{1}{2}(y-x) \cdot \nabla^2 f(z) \cdot (y-x)$$

where $z = x + t(y-x)$. Then as the Hessian is positive semidefinite, we have that

$$f(y) \geq f(x) + (y-x) \cdot \nabla f(x)$$

So by the first order conditions, f is convex. □

2.2 Extreme points

Proposition 2.6. For a convex set C, $f : \mathbb{C} \to \mathbb{R}$ convex, if $z \in \operatorname{Int} C$, then $f(z)$ cannot be the maximum of $f(x)$ in C.

Proof. Take any segment $[x, y]$ such that $z \in [x, y]$. □

Definition 2.7 (Extreme point)
For a convex set C, $x \in \mathbb{C}$ is an extreme point if it cannot be written as

$$(1-\delta)y + \delta z$$

for some $\delta \in (0, 1)$, $y, z \in C$ distinct.

Corollary 2.8. The maximum of a convex function must occur at an extreme point.

2.3 Smoothness

Definition 2.9 (β-smooth)
For a C^1 function $f : \mathbb{R}^n \to \mathbb{R}$, we say that f is β-smooth if ∇f is β-Lipschitz. That is,

$$\left\|\nabla f(y) - \nabla f(x)\right\| \leq \beta\left\|y - x\right\|$$

Proposition 2.10. If f is C^2, then β-smoothness implies that

$$\nabla^2 f(x) \preceq \beta I \iff \beta I - \nabla^2 f(x) \text{ is positive semidefinite}$$

which means that all eigenvalues of $\nabla^2 f$ are less than β. Furthermore, we have that for all $u \in \mathbb{R}^n$,

$$u^\top \nabla^2 f(x) u \leq u^\top (\beta I) u = \beta ||u||^2$$

Proof. We have from the definition of the Hessian as the second derivative that

$$\nabla f(y) - \nabla f(x) = \nabla^2 f(x)(y - x) + \mathcal{O}\left(\left\|y - x\right\|^2\right)$$

Discarding the second order term, we get that

$$(y - x) \cdot \nabla^2 f(x)(y - x) = (y - x) \cdot (\nabla f(y) - \nabla f(x)) \leq \left\|y - x\right\|\left\|\nabla f(y) - \nabla f(x)\right\| \leq \beta\left\|y - x\right\|^2$$

□

Proposition 2.11. Let f be C^2, convex and β-smooth. Then

$$f(y) \leq f(x) + (y - x) \cdot \nabla f(x) + \frac{\beta}{2}\left\|x - y\right\|^2$$

Proof. By Taylor's theorem we have that

$$\begin{aligned} f(y) &= f(x) + (y - x) \cdot \nabla f(x) + \frac{1}{2}(y - x)^\top \nabla^2 f(z)(y - x) \quad \text{for some} \quad z \in [x, y] \\ &\leq f(x) + (y - x) \cdot \nabla f(x) + \frac{\beta}{2}\left\|y - x\right\|^2 \end{aligned}$$

□

Corollary 2.12.

$$f\left(x - \frac{1}{\beta}\nabla f(x)\right) \leq f(x) - \frac{1}{2\beta}\left\|\nabla f(x)\right\|^2$$

Proof. Define $g(y) = f(x) + (y - x)^\top \nabla f(x) + \frac{\beta}{2}\left\|y - x\right\|^2$. Minimising g as a function of y, we find that

$$0 = \nabla g(y) = \nabla_y \left(f(x) + (y - x)^\top \nabla f(x) + \frac{\beta}{2}\left\|y - x\right\|^2\right) = \nabla f(x) - \beta\left\|y - x\right\|$$

Letting $y = x - \frac{1}{\beta}\nabla f(x)$ in the preceding proposition yields the required result. □

Proposition 2.13 (Improved first order condition). For a convex, β-smooth function, we have that

$$f(y) \geq f(x) + (y-x)^{\mathsf{T}} \nabla f(x) + \frac{1}{2\beta} \left\| \nabla f(x) - \nabla f(y) \right\|^2$$

Proof. For any z, we have that

$$f(x) + (z-x)^{\mathsf{T}} \nabla f(x) \leq f(z) \leq f(y) + (z-y)^{\mathsf{T}} \nabla f(y) + \frac{\beta}{2} \left\| z - y \right\|^2$$

Which implies that

$$f(x) - f(y) \leq (x-z)^{\mathsf{T}} \nabla f(x) + (z-y)^{\mathsf{T}} \nabla f(y) + \frac{\beta}{2} \left\| z - y \right\|$$

Minimising the right hand side in terms of z yields the required result. □

2.4 Strong convexity

Definition 2.14 (α-strongly convex)
$f : \mathbb{R}^n \to \mathbb{R}$ is α-strongly convex if

$$f(y) \geq f(x) + (y-x)^{\mathsf{T}} \nabla f(x) + \frac{\alpha}{2} \left\| y - x \right\|^2$$

Proposition 2.15. If f is C^2 and α-strongly convex, then

$$\alpha I \preceq \nabla^2 f$$

Proposition 2.16. Suppose f is α strongly convex, and let

$$p^* = \min_y f(y)$$

Then we have that for all x,

$$p^* \geq f(x) - \frac{1}{2\alpha} \left\| \nabla f(x) \right\|^2$$

Proof. We have (by assumption) that

$$f(y) \geq f(x) + (y-x)^{\mathsf{T}} \nabla f(x) + \frac{\alpha}{2} \left\| y - x \right\|^2$$

Minimising the left and right hand sides with respect to y, we find that

$$p^* \geq f(x) + \min_y \left((y-x)^{\mathsf{T}} \nabla f(x) + \frac{\alpha}{2} \left\| y - x \right\|^2 \right)$$

Setting ∇ of the RHS to be zero, and substituting we get the required result. □

Proposition 2.17. Suppose f is α strongly convcex, and let

$$x^* = \arg\min_x f(x)$$

Then for any x, we have that

$$\left\| x - x^* \right\| \leq \frac{2}{\alpha} \left\| \nabla f(x) \right\|$$

Proof. By Cauchy-Schwarz, we have that

$$f(x^*) \geq f(x) + (x^* - x)^{\mathsf{T}} \nabla f(x) + \frac{\alpha}{2}\left\|x^* - x\right\|^2 \geq f(x) + \left\|x^* - x\right\|\left\|\nabla f(x)\right\| + \frac{\alpha}{2}\left\|x^* - x\right\|^2$$

Rearranging and using the fact that $f(x^*) - f(x) \leq 0$ gives the required result. □

2.5 Gradient descent

One method for finding the minimum of a function is known as gradient descent.

Algorithm 1: Gradient descent

```
t ← 0;
repeat
    $v_t$ ← DescDir() ;                    // Choose a descending direction
    $\eta_t$ ← StepSize() ;                      // Choose a step size
    $x_{t+1} \leftarrow x_t + \eta_t v_t$;
    t ← t + 1;
until condition met, e.g. $\nabla f(x_t) = 0$, or t large;
```

where the descending direction v_t is chosen such that $v_t \cdot \nabla f(x_t) < 0$.

Theorem 2.18. For f α-strongly convex and β-smooth, gradient descent with $v_t = -\nabla f(x_t)$ and $\eta_t = 1/\beta$ satisfies

$$f(x_T) - f(x^*) \leq e^{-\frac{\alpha T}{\beta}} \frac{\beta}{2}\left\|x^* - x_0\right\|^2$$

Proof. We prove the stronger result

$$f(x_T) - f(x^*) \leq \left(1 - \frac{\alpha}{\beta}\right)^T (f(x_0) - f(x^*))$$

from which the required result can be shown. We have that

$$\begin{aligned}
f(x_{t+1}) - f(x^*) &\leq f(x_t) - f(x^*) - \frac{1}{2\beta}\left\|\nabla f(x_t)\right\|^2 \\
&\leq f(x_t) - f(x^*) - \frac{\alpha}{\beta}(f(x_t) - f(x^*)) \\
&= \left(1 - \frac{\alpha}{\beta}\right)(f(x_t) - f(x^*))
\end{aligned}$$

and the result follows by induction. □

2.6 Newton's method

Gradient descent is a first order method. A second order method is known as Newton's method, and is given by

$$x_{t+1} = x_t - \left(\nabla^2 f(x_t)\right)^{-1} \nabla f(x_t)$$

3 Lagrange multipliers

Consider the minimisation problem

$$\text{minimise } f(x) \text{ subject to } h(x) = b, x \in \mathcal{X}$$

We can convert this into an unconstrained problem by minimising the Lagrangian

$$L(x, \lambda) = f(x) - \lambda^{\mathsf{T}}(h(x) - b)$$

Theorem 3.1 (Lagrange sufficiency)**.** Suppose we can find λ^* such that

$$\min_{x\in\mathcal{X}} L(x,\lambda^*) = L(x^*,\lambda^*)$$

for some x^* (i.e. the minimum is attained). Furthermore, suppose $x \in \mathcal{X}(b)$. Then x^* is optimal for f. That is,

$$\min_{x\in\mathcal{X}(b)} f(x) = f(x^*)$$

Proof. Since $x^* \in \mathcal{X}(b)$, we already have that $\min_{x\in\mathcal{X}(b)} f(x) \leq f(x^*)$. So suffices to show the reverse inequality.

$$\begin{aligned}
\min_{x\in\mathcal{X}(b)} &= \min_{x\in\mathcal{X}(b)} \left(f(x) - (\lambda^*)^{\mathsf{T}}(h(x)-b)\right) \\
&\geq \min_{x\in\mathcal{X}} \left(f(x) - (\lambda^*)^{\mathsf{T}}(h(x)-b)\right) \quad \text{as} \quad \mathcal{X}(b)\subseteq\mathcal{X} \\
&= \min_{x\in\mathcal{X}} L(x,\lambda^*) \\
&= L(x^*,\lambda^*) \\
&= f(x^*)
\end{aligned}$$

□

3.1 Slack variables

Given a problem of the form[1]

$$\text{minimise } f \text{ subject to } x\in\mathcal{X}, s\geq 0, h(x)+s=b$$

The Lagrangian is of the form

$$L(x,\lambda,s) = f(x) - \lambda^T(h(x)+s-b)$$

and to find the optimal solution, we first consider the set

$$\Lambda = \left\{\lambda : \inf_{x\in\mathcal{X},s\geq 0} L(x,\lambda,s) > -\infty\right\}$$

For each $\lambda\in\Lambda$, we find $x^*(\lambda)$ and $s^*(\lambda)$ such that the infimum is attained. Then we find $\lambda^*\in\Lambda$ such that $x^*(\lambda^*)$ and $s^*(\lambda^*)$ are feasible.

3.2 Complementary slackness

The definition of the set Λ, and finding $x^*(\lambda)$ and $s^*(\lambda)$ involves the minimisation problem

$$\text{minimise } f(x) - \lambda^{\mathsf{T}}(h(x)-b) - \lambda^{\mathsf{T}}s \text{ subject to } x\in\mathcal{X}, s\geq 0$$

Proposition 3.2. For any $\lambda\in\Lambda$, $\lambda\leq 0$.

Proof. Suppose if for some i, $\lambda_i > 0$. Then taking $s_i\to\infty$ we see that the problem is not bounded. □

Proposition 3.3 (Complementary slackness)**.** At the optimum (x,λ,s), for each i, $\lambda_i s_i = 0$.

Proof.

$$-\lambda^{\mathsf{T}}s = -\sum\lambda_i s_i = \sum(-\lambda_i)s_i$$

is a sum of nonnegative terms, therefore the minimum is attained when they are all zero. □

[1] We have already converted the inequality into a regional constraint.

3.3 Duality

Definition 3.4 (Dual problem)
Given a minimisation problem

$$\text{minimise } f(x) \text{ subject to } x \in \mathcal{X}, h(x) = b$$

with Lagrangian $L(x, \lambda) = f(x) - \lambda^{\mathsf{T}}(h(x) - b)$. Define

$$g(\lambda) = \inf_{x \in \mathcal{X}} L(x, \lambda)$$

and

$$\text{maximise } g(\Lambda) \text{ subject to } \lambda \in \Lambda$$

is known as the dual problem. The original problem is known as the primal problem.

Theorem 3.5 (Weak duality).

$$\inf_{x \in \mathcal{X}(b)} f(x) \geq \sup_{\lambda \in \Lambda} g(\lambda)$$

Proof. For any $\lambda \in \Lambda$, we have that

$$\begin{aligned} \inf_{x \in \mathcal{X}(b)} f(x) &= \inf_{x \in \mathcal{X}(b)} L(x, \lambda) \\ &\geq \inf_{x \in \mathcal{X}} L(x, \lambda) \\ &= g(\lambda) \end{aligned}$$

□

Definition 3.6 (Duality gap, Strong duality)
Define the duality gap to be

$$\inf_{x \in \mathcal{X}(b)} f(x) - \sup_{\lambda \in \Lambda} g(\lambda)$$

If this is zero we say strong duality holds.

Remark 3.7. If the Lagrangian method works, then we have λ^* such that

$$\inf_{x \in \mathcal{X}(b)} L(x, \lambda^*) = \inf_{x \in \mathcal{X}} L(x, \lambda^*)$$

and we get equality in the above. So strong duality holds. The converse is also true.

Definition 3.8 (Value function)
For a minimisation problem, define the value function

$$\phi(c) = \inf_{x \in \mathcal{X}(c)} f(x)$$

Definition 3.9 (Supporting hyperplane)
A function ϕ has a supporting hyperplane at b if there exists λ such that for all c,

$$\phi(c) \geq \phi(b) + \lambda^{\mathsf{T}}(c-b)$$

Theorem 3.10. Strong duality holds if and only if the value function ϕ has a supporting hyperplane at b.

Proof. Suppose ϕ has a supporting hyperplane at b, say $\phi(c) \geq \phi(b) + \lambda^{\mathsf{T}}(c-b)$. Then we have that

$$\begin{aligned} g(\lambda) &= \inf_{x \in \mathcal{X}} \left(f(x) - \lambda^{\mathsf{T}}(h(x)-b)\right) \\ &= \inf_{c} \inf_{x \in \mathcal{X}(c)} \left(f(x) - \lambda^{T}(h(x)-c) - \lambda^{T}(c-b)\right) \\ &= \inf_{c} \left(\phi(c) - \lambda^{T}(c-b)\right) \\ &\geq \phi(b) = \inf_{x \in \mathcal{X}(b)} f(x) \end{aligned}$$

Since weak duality gives us the reverse inequality, we must in fact have equality here. Conversely, if strong duality holds, then we have λ such that $g(\lambda) = \phi(b)$. Then

$$\begin{aligned} \phi(b) &= g(\lambda) \\ &= \inf_{x \in \mathcal{X}} \left(f(x) - \lambda^{\mathsf{T}}(h(x)-b)\right) \\ &= \inf_{x \in \mathcal{X}} \left(f(x) - \lambda^{\mathsf{T}}(h(x)-c) - \lambda^{\mathsf{T}}(c-b)\right) \\ &\leq \phi(c) - \lambda^{\mathsf{T}}(c-b) \end{aligned}$$

by weak duality. □

Theorem 3.11. A function $\phi : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if it has a supporting hyperplane at every $b \in \mathbb{R}^n$.

Proof. Omitted. □

Theorem 3.12. Consider the minimisation problem

$$\text{minimise } f(x) \text{ subject to } h(x) \leq b, x \in \mathcal{X}$$

with value function ϕ. Then ϕ is convex if $\mathcal{X}$ is convex, $f : \mathbb{R}^n \to \mathbb{R}$ convex, and $h_j : \mathbb{R}^n \to \mathbb{R}$ convex for all j.

Proof. Suppose $b, c \in \mathcal{X}$, and the minima are attained at x_b and x_c respectively. Let $x = (1-\lambda)x_b + \lambda x_c$. Then $x \in \mathcal{X}$ by convexity. Furthermore, by convexity $h(x) \leq (1-\lambda)h(x_b) + \lambda h(x_c) \leq (1-\lambda)b + \lambda c$, so x is in $\mathcal{X}((1-\lambda)b + \lambda c)$.

Thus, we have that

$$\phi((1-\lambda)b + \lambda c) \leq f(x) \leq (1-\lambda)f(x_b) + \lambda f(x_c) = (1-\lambda)\phi(b) + \lambda\phi(c)$$

□

4 Linear programs

Definition 4.1 (Linear program)
A linear program is an optimisation problem where the objective function, functional and regional constraints are all linear. In particular, we have

$$\text{minimise } c^{\mathsf{T}} x \text{ subject to } \textit{conditions}$$

where *conditions* include inequalities and equalities involving linear combinations of the decision variables.

Remark 4.2. We note that $\leq$ constraints can be written in the above form by negating, and equality constraints can be written as a pair of inequalities. Regional constraints of the form $x_j \geq 0$ can also be written in the above form.

Definition 4.3 (General form)
The general form of the above linear program is

$$\text{minimise } c^{\mathsf{T}} x \text{ subject to } Ax \geq b$$

where A is an $m \times n$ matrix, and $b \in \mathbb{R}^n$, with

$$A = \begin{pmatrix} \cdots & a_1^{\mathsf{T}} & \cdots \\ & \vdots & \\ \cdots & a_m^{\mathsf{T}} & \cdots \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$

Definition 4.4 (Standard form)
A linear program is in standard form if it is of the form

$$\text{minimise } c^{\mathsf{T}} x \text{ subject to } Ax = b, x \geq 0$$

Proposition 4.5. Every linear program in general form can be written in standard form.

Proof. By adding slack variables, we can transform the inequality constraint into an equality and a regional constraint on the slack variables. For the decision variables, let $x_j = x_j^+ - x_j^-$, where $x_j^+, x_j^- \geq 0$. Then we get

$$\text{minimise } c^{\mathsf{T}}(x^+ - x^-) \text{ subject to } A(x^+ - x^-) = b; x^+, x_i \geq 0$$

Letting $z = (x^+, x^-)$, we get

$$\text{minimise } (c^{\mathsf{T}}, = c^{\mathsf{T}})z \text{ subject to } (A, -A)z = b, z \geq 0$$

□

4.1 Basic solutions

In this section we will be considering linear programs in standard form. That is,

$$\text{minimise } c^{\mathsf{T}} x \text{ subject to } Ax = b, x \geq 0$$

where A is an $m \times n$ matrix, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$.

Definition 4.6 (Basic solution)

A point x is a basic solution if $Ax = b$, and x has at most m nonzero entries.

Definition 4.7 (Basic feasible solution)

A point x is a basic feasible solution if x is a basic solution and $x \geq 0$.

Finding basic solutions

Remark 4.8. From now on, we make the additional assumptions that

1. The rows of A are linearly independent.
2. Every m columns of A are linearly independent.
3. Every basic solution has exactly m nonzero entries (nondegeneracy).

Choose $1 \leq b_1 \leq \cdots \leq b_m \leq n$ to be the entries where x is nonzero. Then

$$\begin{pmatrix} \vdots & & \vdots \\ A_1 & \cdots & A_n \\ \vdots & & \vdots \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ x_{b_1} \\ \vdots \\ x_{b_m} \\ \vdots \\ 0 \end{pmatrix} = \underbrace{\begin{pmatrix} \vdots & & \vdots \\ A_{b_1} & \cdots & A_{b_m} \\ \vdots & & \vdots \end{pmatrix}}_{B} \begin{pmatrix} x_{b_1} \\ \vdots \\ x_{b_m} \end{pmatrix} = b$$

Hence we have that

$$\begin{pmatrix} x_{b_1} \\ \vdots \\ x_{b_m} \end{pmatrix} = B^{-1} b$$

Definition 4.9 (Basis)

B is called the basis matrix, $x_B = (x_{b_1}, \dots, x_{b_m})$ are called basis variables, and $A_{b_1}, \dots, A_{b_m}$ are known as basis columns.

Remark 4.10. If $B^{-1}b \geq 0$, then we have found a BFS.

Theorem 4.11. x is an extreme point of the feasible set $\mathcal{X}(b) = \{x : Ax = b, x \geq 0\}$ if and only if it is a BFS.

Proof. First suppose x is a BFS and y, z are feasible points, with $x = (1-\delta)y + \delta z$, $\delta \in (0,1)$. Let $b_1, \dots, b_m$ be the indices where x is nonzero. Then since $y, z \geq 0$, we must have that $y_j = z_j = 0$ for $j \notin \{b_1, \dots, b_m\}$. Since y, z feasible, we must have $Ay = Az = b$. But this means that $By_B = Bz_B = b$, so $y_B = z_B = x_B \implies x = y = z$.

Now suppose if x is not a BFS. Let the indices where $x \neq 0$ be $i_1, \dots, i_r$, where $r > m$. Since $\operatorname{rank}(A) = m$, we must have that $\{A_{i_1}, \dots, A_{i_r}\}$ is linearly dependent. So we have $w_{i_1}, \dots, w_{i_r}$ such that

$$w_{i_1} A_{i_1} + \cdots + w_{i_r} A_{i_r} = 0$$

Define

$$w_i = \begin{cases} w_{i_j} & \text{if } i = i_j \\ 0 & \text{otherwise} \end{cases}$$

Then $Aw = 0$. Then for any ε, we have that

$$A(x + \varepsilon w) = Ax + \varepsilon Aw = Ax$$

Hence if we have $\varepsilon > 0$ such that $x + \varepsilon w, x - \varepsilon w$ are feasible, then

$$x = \frac{1}{2}(x + \varepsilon w) + \frac{1}{2}x - \varepsilon w$$

is not an extreme point. □

4.2 Duality

Theorem 4.12. If a linear program is bounded and feasible, then it satisfies strong duality.

Proof. A linear program is convex. □

Proposition 4.13 (Dual of a linear program in standard form)**.** For a linear program

$$P : \text{minimise } c^\mathsf{T} x \text{ subject to } Ax = b, x \geq 0$$

the dual is

$$D : \text{maximise } \lambda^\mathsf{T} b \text{ subject to } \lambda^T A \leq c^T$$

Proof. The objective function in the dual is

$$g(\lambda) = \inf_{x \geq 0} L(x, \lambda) = \inf_{x \geq 0} \left(c^\mathsf{T} x - \lambda^T (Ax - b)\right) = \inf_{x \geq 0} (c^\mathsf{T} - \lambda^\mathsf{T} A)x + \lambda^T b$$

For the infimum to exist, we must have that

$$\Lambda = \left\{\lambda : c^\mathsf{T} - \lambda^\mathsf{T} A \geq 0\right\} = \left\{\lambda : \lambda^\mathsf{T} A \leq c^\mathsf{T}\right\}$$

and the minimum occurs for $x = 0$, i.e. $g(\lambda) = \lambda^\mathsf{T} b$. □

Proposition 4.14 (Dual of linear program in general form)**.** For a linear program

$$P : \text{minimise } c^\mathsf{T} x \text{ subject to } Ax \geq 0$$

the dual is

$$D : \text{maximise } \lambda^\mathsf{T} b \text{ subject to } \lambda^\mathsf{T} A = c^\mathsf{T}, \lambda \geq 0$$

Proof. Similar to the case of the standard form. □

4.3 Optimality conditions

Theorem 4.15 (Fundamental theorem of Linear programs)**.** Let x and p be feasible solutions to the primal and dual problems[a] respectively. Then x and p are optimal if and only if complementary slackness holds. That is,

$$p^\mathsf{T}(Ax - b) = 0 \quad \text{and} \quad (c - A^\mathsf{T} p)^\mathsf{T} x = 0$$

[a]In this case we have stated/proven this for linear programs in standard form, but this holds for any linear program. See Lecturer's

notes for proof.

Proof. If x and p are feasible, then we have from weak duality that $c^\top x \leq p^\top b$. Thus, if complementary slackness holds, then we have equality. So strong duality holds and they are optimal.

Conversely, if we have optimal solutions, then by strong duality $c^\top x = p^\top b$. Since x is feasible, we have that $Ax = b$. Combining these gives us the result required. □

Reduced costs

For a linear program in standard form, the above gives us the following optimality conditions.

1. x is primal feasible, that is, $Ax = b, x \geq 0$.
2. λ is dual feasible, that is, $A^\top \lambda \leq c$.
3. Complementary slackness[2], $x^\top(c - A^\top \lambda) = 0$.

Suppose x_B is a BFS. Substituting into 3., we find that

$$x_B(c_B - B^T \lambda) = 0$$

Since for a BFS $x_B > 0$, we must have $c_B = B^\top \lambda$, and $\lambda = (B^\top)^{-1} c_B$. If this is dual feasible, then we have an optimal solution. That is, if

$$\bar{c} = c - A^\top (B^\top)^{-1} c_B \geq 0$$

$\bar{c}$ is known as the reduced costs.

Theorem 4.16. Let x be a BFS, B be the basis matrix. Let $\bar{c}$ be the reduced costs. Then x is optimal if and only if $\bar{c} \geq 0$.

Proof. By optimailty conditions, since the corresponding λ gives us a dual feasible solution that satisfies complementary slackness. □

4.4 Simplex method

Feasible direction

Definition 4.17 (Feasible direction)
Let $x \in \mathcal{X}(b) = \{x : Ax = b, x \geq 0\}$. Then $d \in \mathbb{R}^n$ is a feasible direction if there exists $\theta > 0$ such that $x + \theta d \in \mathcal{X}(b)$.

Definition 4.18 (j-th basic direction)
Let x be a BFS. Without loss of generality, assume $b_i = i$. Then the j-th basic direction is the vector

$$d = (d_{b_1}, \dots, d_{b_m}, 0, \dots, 1, \dots, 0)$$

where the 1 is at the j-th entry, and $d_B = -B^{-1}A_j$, where A_j is the j-th column of A.

Proposition 4.19.

$$Ad = 0$$

Proof.

$$Ad = Bd_B + A_j = 0$$

□

[2]The other equation holds automatically by condition 1.

Proposition 4.20. d is a feasible direction for a BFS x.

Proof.

$$x + \theta d = (x_{b_1} + \theta d_{b_1}, \dots, x_{b_m} + \theta d_{b_m}, 0, \dots, \theta, \dots, 0) \geq 0$$

for θ sufficiently small, since by non-degeneracy, $x_B > 0$. □

Proposition 4.21. The cost at $x + \theta d$ is $c^\top x + \theta \bar{c}_j$

Proof.

$$c^\top (x + \theta d) = c^\top x + \theta c^\top d = c^\top x + \theta(c_j - c_B^\top B^{-1} A_j) = c^\top x + \theta \bar{c}_j$$

□

Minimising cost using simplex method

From the above, we can see that if $\bar{c}_j < 0$ for any c_j, then we can reduce the cost by travelling along the j-th basis direction d. The amount we can move is given by the largest θ such that $x + \theta d$ is feasible.

Algorithm 2: Simplex method

```
input: a BFS $x$ with basis matrix $B$
$\bar{c} \leftarrow c - A^\top (B^\top)^{-1} c_B$;
while $\bar{c} \not\geq 0$ do
    $j \leftarrow$ ChooseBasicDir() ;                    // $j$ is any index such that $c_j < 0$
    $u \leftarrow B^{-1} A_j$;
    if $u \leq 0$ then
        return $-\infty$ ;          // Problem is unbounded, the minimum cost is $-\infty$
    else
        $\theta^* \leftarrow \min_{i \text{ s.t. } u_i > 0} \frac{x_{b_i}}{u_i}$ ;          // The amount by which we can move
        $l \leftarrow \arg\min_{i \text{ s.t. } u_i > 0} \frac{x_{b_i}}{u_i}$ ;         // The index along which we are moving
        $y_i = \begin{cases} x_{b_j} + \theta^* u_j & \text{if } i = b_j \text{ and } j \neq l \\ 0 & \text{if } i = b_l \\ \theta^* & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$ ;     // The result of moving along the $l$-th
         basis direction by $\theta^*$
        $x \leftarrow y$; ;                                   // and update $B$ accordingly
        $\bar{c} \leftarrow c - A^\top (B^\top)^{-1} c_B$;
    end
end
```

In practice, these computations can be combined into a tableau.

Simplex tableau

Definition 4.22 (Simplex tableau)
The simplex tableau (for a given BFS x and basis matrix B) is a tableau of the form

$$\begin{array}{c|ccc} -c_B^\top x_B & \bar{c}_1 & \dots & \bar{c}_n \\ \hline x_{b_1} & \vdots & & \vdots \\ \vdots & B^{-1}A_1 & \dots & B^{-1}A_n \\ x_{b_m} & \vdots & & \vdots \end{array}$$

where we say the "first" row is the zero-th row.

Algorithm 3: Simplex tableau

```
input: Simplex tableau
while $\bar{c} \ngeq 0$ do
    $j \leftarrow$ ChooseBasicDir() ;                 // $j$ is any index such that $c_j < 0$
    $u \leftarrow B^{-1}A_j$ ;                        // i.e. the $j$-th column of the simplex tableau
    if $u \leq 0$ then
        return $-\infty$
    else
        $\theta^* \leftarrow \min_{i \text{ s.t. } u_i>0} \frac{x_{b_i}}{u_i}$ ;      // The amount by which we can move
        $l \leftarrow \arg\min_{i \text{ s.t. } u_i>0} \frac{x_{b_i}}{u_i}$ ;        // The index along which we are moving
        add to each row (including the zeroth) a constant multiple of the $l$-th row such that $u_l = 1$, all
          other entries of the pivot column are zero.
    end
end
```

5 Applications

5.1 Game theory

Consider a zero-sum two player game, where player 1 (P1) has actions $\{1, \ldots, m\}$, and player 2 (P2) has actions $\{1, \ldots, n\}$.

Definition 5.1 (Payoff matrix)
If P1 plays i and P2 plays j, then P1 wins a_{ij} and P2 loses a_{ij}. The matrix

$$A = \left(a_{ij}\right)_{1\leq i\leq m, 1\leq j\leq n}$$

is known as the payoff matrix.

Minmax

If P1 plays i first, they expect to get $\min_j a_{ij}$. This gives us the problem

$$\text{maximise } \min_{1\leq j\leq n} a_{ij} \text{ subject to } 1 \leq i \leq m$$

and the related problem for P2,

$$\text{minimise } \max_{1\leq i\leq m} a_{ij} \text{ subject to } 1 \leq j \leq n$$

Mixed strategies

If instead we allow P1 and P2 to choose randomly, with probabilities $(p_1, \ldots, p_m)$ and $(q_1, \ldots, q_n)$ respectively. If P2 plays j, then the expected value for P1 is $\sum_i a_{ij}p_i$. This gives us the optimisation problem

$$\text{maximise } v \text{ subject to } A^{\mathsf{T}}p \geq ve, e^{\mathsf{T}}p = 1, p \geq 0$$

where $e = (1, \ldots, 1)$ is a vector of ones.

Remark 5.2. The max min has been turned into an inequality of vectors.

For P2, we have the problem

$$\text{minimise } w \text{ subject to } Aq \leq we, e^\mathsf{T} q = 1, q \geq 0$$

Proposition 5.3. These two problems are dual.

Corollary 5.4. A strategy p is optimal for P1 if there exists q, v such that

1. (Primal feasibility) $A^\mathsf{T} p \geq ve, e^\mathsf{T} p = 1, p \geq 0$
2. (Dual feasibility) $Aq \leq we, e^\mathsf{T} q = 1, q \geq 0$
3. (Complementary slackness) $v = p^\mathsf{T} Aq$

Proof. The complementary slackness equations are

$$(Aq - we)^\mathsf{T} p = 0 \quad \text{and} \quad q^\mathsf{T}(A^\mathsf{T} p - ve)$$

From which we get that $v = w = p^\mathsf{T} Aq$. □

Saddle points and dominating strategies

If in the (non-mixed) strategies we find the same (i, j), then this is known as a saddle point, and gives us an optimum.

If there exists i, i' such that $a_{i'j} \geq a_{ij}$ for all j, then i' dominates i and we can drop row i without loss of generality. Similarly we can find dominating strategies for P2.

5.2 Network flow

Definition 5.5 (Network flow)
Given a graph $G = (V, E)$, where $|V| = n$, we have the following

- $b \in \mathbb{R}^n$, where b_i is the amount of flow entering vertex i. If $b_i > 0$ then i is a source, and if $b_i < 0$ then i is a sink.
- $C \in \mathbb{R}^{n \times n}$, where c_{ij} is the cost per unit of flow from i to j.
- Matrices $\overline{M}, \underline{M}$ are upper and lower bounds on the flow respectively.

The flow is given by a matrix X, where x_{ij} is the amount of flow from i to j. The minimum cost flow is given by the linear program

$$\text{minimise } \sum_{i,j} c_{ij} x_{ij} \text{ subject to } \underline{M} \leq X \leq \overline{M}, b_i + \sum_{(j,i) \in E} x_{ji} = \sum_{(i,j) \in E} x_{ij}, \sum_i b_i = 0$$

Remark 5.6. The second condition can be expressed as a linear equality by defining a matrix $A \in \mathbb{R}^{|V| \times |E|}$, where

$$a_{i,(j,k)} = \begin{cases} 1 & \text{if } i = j \\ -1 & \text{if } i = k \\ 0 & \text{otherwise} \end{cases}$$

5.3 Transport problem

Suppose we have the special case where G is a bipartite graph. Say we have n sources (called suppliers in this case), and m sinks (called customers in this case). The suppliers have capacity $s_1, \ldots, s_n$ (which is just the corresponding entries in b), and the customers have demand $d_1, \dots, d_m$ (which is $-$ of the corresponding entries in b. Furthermore we assume that there is no upper/lower bound on the flow. This gives us the optimisation problem

$$\text{minimise } \sum_{i,j} c_{ij}x_{ij} \text{ subject to } \sum_{j=1}^{m} x_{ij} = s_i, \sum_{i=1}^{n} x_{ij} = d_j, x \geq 0$$

Theorem 5.7. Every minimum cost flow problem with finite capacities, or nonnegative costs, can be written as a transportation problem.

Proof. First, we may assume $\underline{M} = 0$ without loss of generality. Also, we note that if all the costs are nonnegative, for any edge with infinite capacity, we can replace it by a sufficiently large capacity such that the optimal solution is not changed.

For each vertex i in the transportation problem, create a customer with demand

$$\sum_{(i,k)\in E} \overline{m}_{ik} - b_i$$

and for each edge (i, j), create a supplier with capacity

$$\overline{m}_{ij}$$

i.e.

$$\begin{tikzcd}
 & & i \arrow[r, dashed, "\sum_k \overline{m}_{ik} - b_i"] & \bullet \\
\bullet \arrow[r, dashed, "\overline{m}_{ij}"] & (i,j) \arrow[ur, "0"] \arrow[dr, "c_{ij}"] & & \\
 & & j \arrow[r, dashed, "\sum_k \overline{m}_{jk} - b_j"] & \bullet
\end{tikzcd}$$

Suppose x_{ij} flows from (i, j) to j, and $\overline{m}_{ij} - x_{ij}$ into i. Then we note that this satisfies the conservation equations, and the optimal cost of this problem is the same as in the original problem. □

Theorem 5.8 (Optimality conditions for the transport problem)**.** If we have $x \in \mathbb{R}^{n\times m}$ feasible, and dual variables $\lambda \in \mathbb{R}^n$ (suppliers) and $\mu \in \mathbb{R}^m$ (customers). Suppose

- $\lambda_i + \mu_j \leq c_{ij}$ for all $(i, j) \in E$.
- $c_{ij} - (\lambda_i + \mu_j))x_{ij} = 0$.

then x is optimal.

Proof. Consider the Lagrangian

$$\begin{aligned} L(x, \lambda, \mu) &= \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} - \sum_{i=1}^{n} \lambda_i \left(\sum_{j=1}^{m} x_{ij} - s_i\right) - \sum_{j=1}^{m} \mu_j \left(\sum_{i=1}^{n} x_{ij} - d_j\right) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} \left(c_{ij} - \lambda_i - \mu_j\right) x_{ij} + \sum_i \lambda_i s_i + \sum_j \mu_j d_j \end{aligned}$$

Thus, for λ, μ to be dual feasible, we must have that $\lambda_i + \mu_j \leq c_{ij}$. Furthermore,

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\left(c_{ij}-\lambda_i-\mu_j\right)x_{ij}$$

is a sum of nonnegative terms, which must be zero at the minimum, so we get the second equation, which is complementary slackness. Thus, we have primal and dual feasibility, as well as complementary slackness, which means that x is optimal. □

Remark 5.9. The values of the dual variables are not unique. If (λ, μ) are dual feasible, so are $(\lambda+\theta e, \mu-\theta e)$. To reduce the degrees of freedom, we may without loss of generality assume $\lambda_1 = 0$.

Initial assignment

To be able to solve the linear program using (a modified version of) the simplex method, we will need a BFS to start off with. We can do this greedily.

Algorithm 4: Greedy initial assignment

```
i, j ← 1;
while i ≤ $n$ and j ≤ $m$ do
    $x_{ij} \leftarrow \min\left(s_i - \sum_{j'=1}^{j-1} x_{ij'}, d_j - \sum_{i'=1}^{i-1} x_{i'j}\right)$;      // Fill $x_{ij}$ as much as possible
    if $\sum_{j'=1}^{j} x_{ij'} = s_i$ then      // If we have filled i then move on to the next supplier
        i ← i + 1;
    end
    if $\sum_{i'=1}^{i} x_{i'j} = d_j$ then      // If we have filled j then move on to the next customer
        j ← j + 1;
    end
end
```

Proposition 5.10. The set of edges with positive flow forms a spanning tree.

Remark 5.11. By complementary slackness, if $x_{ij} > 0$ then $\lambda_i + \mu_j = c_{ij}$. Since we can assume $\lambda_1 = 0$, we will have $n+m-1$ equations for $n+m-1$ variables, which can then be solved.

Transportation tableau

Definition 5.12 (Transportation tableau)

The transportation tableau has the form

	μ_1	$\cdots$	μ_m	
λ_1	x_{11} $\boxed{c_{11}}$	$\cdots$	x_{1m} $\boxed{c_{1m}}$	s_1
$\vdots$	$\vdots$		$\vdots$	$\vdots$
λ_n	x_{n1} $\boxed{c_{n1}}$	$\cdots$	x_{nm} $\boxed{c_{nm}}$	s_n
	d_1	$\cdots$	d_m	

Algorithm 5: Transportation tableau

```
while $\exists i, j$ such that $c_{ij} < \lambda_i + \mu_j$ do
    Choose $i, j$ such that $c_{ij} - \lambda_i - \mu_j$ is least;
    Increase flow along $(i, j)$ as much as possible;
    Recalculate the dual variables;
end
```

For the update step, note that we must have $x_{ij} = 0$, so if we include the edge (i, j), we get a cycle in the graph. For simplicity, suppose we get the cycle

$$\begin{tikzcd}
i \arrow[r, dashed] \arrow[dr] & j \\
i' \arrow[r] \arrow[ur] & j'
\end{tikzcd}$$

Suppose we pass θ along (i, j). Then we get an updated allocation with

$$x_{ij} \leftarrow \theta \quad , \quad x_{ij'} \leftarrow x_{ij'} - \theta \quad , \quad x_{i'j} \leftarrow x_{ij'} - \theta \quad \text{and} \quad x_{i'j'} \leftarrow x_{i'j'} + \theta$$

We see that $\theta = \min(x_{ij'}, x_{i'j})$ is the maximum allowable value, and this makes the corresponding entry zero.

5.4 Max flow, min cut

Now suppose we have one source and one sink in the network, and we wished to maximise the flow through the network. This gives us the optimisation problem

$$\text{maximise } \delta \text{ subject to } \sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} \delta & \text{if } i = 1 \\ -\delta & \text{if } i = n \\ 0 & \text{otherwise} \end{cases}, 0 \leq X \leq C$$

where X is the matrix of allocations, C is the matrix of capacities, and $delta$ is the flow through the network.

Definition 5.13 (Cut)
A cut of a graph $G = (V, E)$ is a partition of the vertices into two sets $S, V \setminus S$.

Definition 5.14 (Capacity of a cut)
The capacity of a cut S is

$$C(s) = \sum_{i \in S, j \in V \setminus S} c_{ij}$$

Theorem 5.15. Let x be a feasible flow, with value δ. Then for any cut $S, V \setminus S$ such that $1 \in S$, $n \in V \setminus S$, we have that

$$\delta \leq C(S)$$

Proof. For any $X, Y \subseteq V$, define

$$f(X, Y) = \sum_{i \in X, j \in Y} x_{ij}$$

Then we have that

$$\begin{aligned}
\delta &= \sum_{i \in S} \left(\sum_j x_{ij} - \sum_j x_{ji} \right) \\
&= f(S, V) - f(V, S) \\
&= f(S, S) + f(S, V \setminus S) - f(S, S) - f(V \setminus S, S) \\
&= f(S, V \setminus S) - f(V \setminus S, S) \\
&\leq f(S, V \setminus S) \\
&= C(S)
\end{aligned}$$

□

Theorem 5.16 (Max flow, min cut). Let δ^* be the value of the maximum flow. Then we have that

$$\delta^* = \min \{C(S) : S \subseteq V, 1 \in S, n \notin S\}$$

Definition 5.17 (Path)
A path $v_0 \dots v_n$ in a graph $G = (V, E)$ such that for all i, either $(v_i, v_{i+1}) \in E$ or $(v_{i+1}, v_i) \in E$[a].

[a]The usual definition of a directed path in a graph is different to this.

Definition 5.18 (Augmenting path)
For an allocation, a path $v_0 \dots v_n$ is augmenting if for all i, either $x_{v_i v_{i-1} < 0}$, or $x_{v_i v_{i+1} < c_{v_i v_{i+1}}}$. That is, either the forwards direction has remaining capacity, or the reverse edge has nonzero flow.

Proof. Note that if we have an augementing path from 1 to n, then the allocation cannot be optimal, since we can increase the flow by θ along the forward edges, and decrease by η along the backward edges, for sufficiently small θ. Now suppose x is optimal. Then let

$$S = \{1\} \cup \{i : \exists \text{ augmenting path from 1 to } i\}$$

Then $n \notin S$ by definition. Recall that $\delta = f(S, V \setminus S) - f(V \setminus S, S)$. We must have that $f(V \setminus S, S) = 0$, since if not, say we have $i \notin S, j \in S$ such that $x_{ij} > 0$. Then concatenate any augmenting path from 1 to j with the edge (j, i), and we get an augmenting path from 1 to i. But $i \notin S$. Contradiction.

Thus $\delta^* = f(S, V \setminus S) = C(S)$. □

Ford-Fulkerson

The above proof also gives us an algorithm for finding the maximum flow.

Algorithm 6: Ford-Fulkerson

```
input: Feasible flow $x$
while Augmenting path exists do
    Increase flow as much as possible along augmenting path
end
```

Proposition 5.19. For integer (and by rescaling rational) capacities, the Ford-Fulkerson algorithm terminates.

Proof. Without loss of generality we may assume the capacities are integers. Then the value is an integer, and in each step, this increases by at least 1. As the value is bounded above, the algorithm must terminate. □