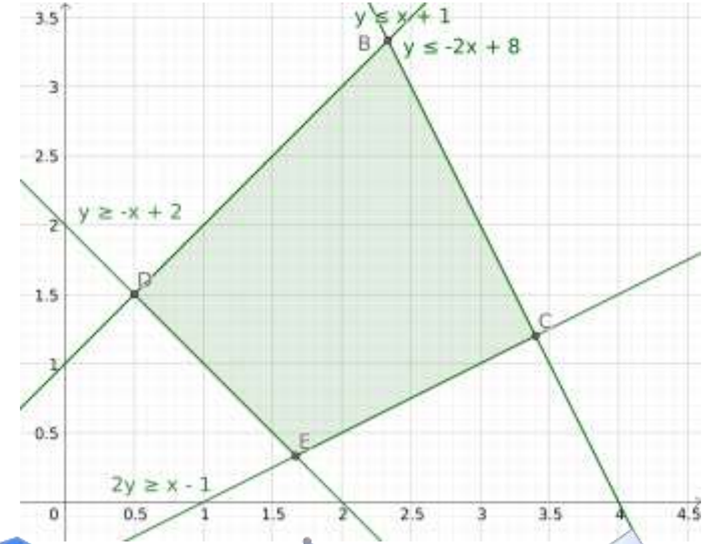
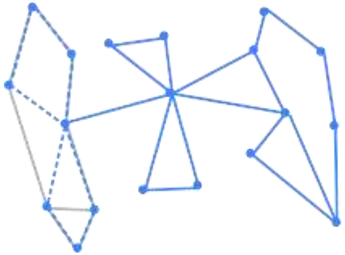
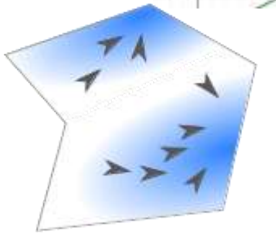
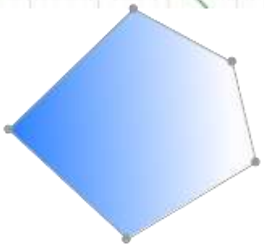
Linear Programming 5

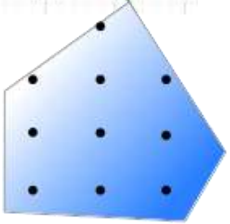
COMP4691 / 8691

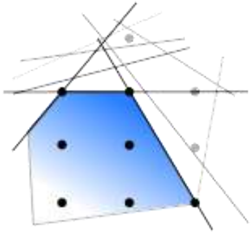


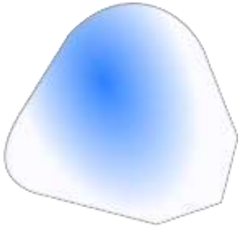












LP Topic Outline

- LP Introduction
- Modelling and solving
- Feasible region and convexity
- Simplex algorithm
- **Relaxations and approximations**
    - Approximations
    - Convex relaxations
    - Battery scheduling example
    - Optimality gap
- **The dual of a linear program**

≈ Approximations ≈

“ **But we all know the world is non-linear** ” - Hotelling 1948*

Indeed we will often want to resort to other approaches to solve problems with non-linearities and combinatorial problems.

Or we can attempt to **trade solution accuracy / quality** off with the favourable computational performance of linear programming. If we do this the right way for the right problem, we can even **sometimes get an exact solution** to the original problem! If not, we can often still learn something useful about the problem.

*A comment from the audience on Dantzig’s first major presentation of his theory of linear programming to the academic world.

Relaxing the Feasible Set

A relaxation is an approximation where we **increase the feasible set.**

$x \in X$	the feasible set
$x \in X'$	a relaxed feasible set
$X \subset X'$	only a relaxation if we haven't lost any feasible points

Can be done constraint by constraint, e.g., $x \geq 10$ to $x \geq 9$ or over a collection of constraints. May include manipulating, removing or replacing constraints.

Can we draw a relationship between the optimal solutions over the two feasible regions?

$$\min_{x \in X} f(x) \geq \min_{x \in X'} f(x)$$

A lower bound on the original problem.

Convex Relaxations

We often want **convex relaxations,** e.g., turn a non-convex feasible region into an easier to solve convex one.

E.g., turning integer variables into continuous variables (relaxing the integrality requirement):

$$z \in \{0, 1, 2, 3, 4\} \longrightarrow z \in [0, 4]$$

non-convex, disjoint convex relaxation

Occasionally we will end up with a solution that is actually integer feasible (more on this in the MIP part of the course).

Restricting the Feasible Set

Similarly we can restrict the feasible set or individual constraints if we reduce the feasible set of points.

$X^\dagger \subset X$ If the tightened set of points is a subset of the original feasible set we can expect:

$$\min_{x \in X} f(x) \leq \min_{x \in X^\dagger} f(x)$$

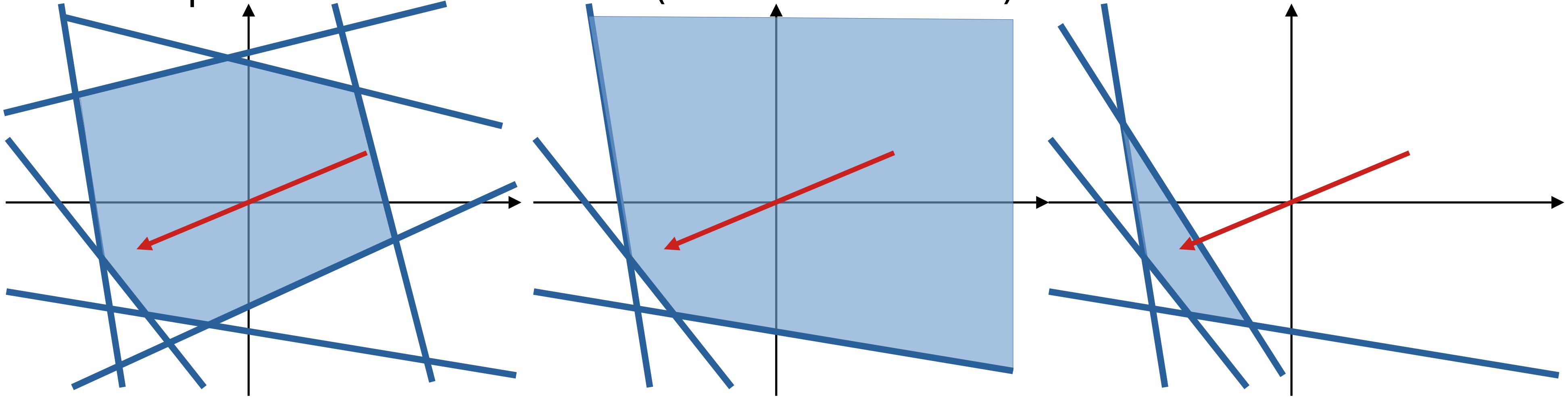
If this is done by adding linear inequalities, this is known as adding **cuts** (cutting off chunks of the feasible region).

Exact Solution / Objective

On some occasions our modified problem can have an optimal objective equal to the original problem (**exact objective value**) and sometimes even the same optimal variable values (**exact solution**).

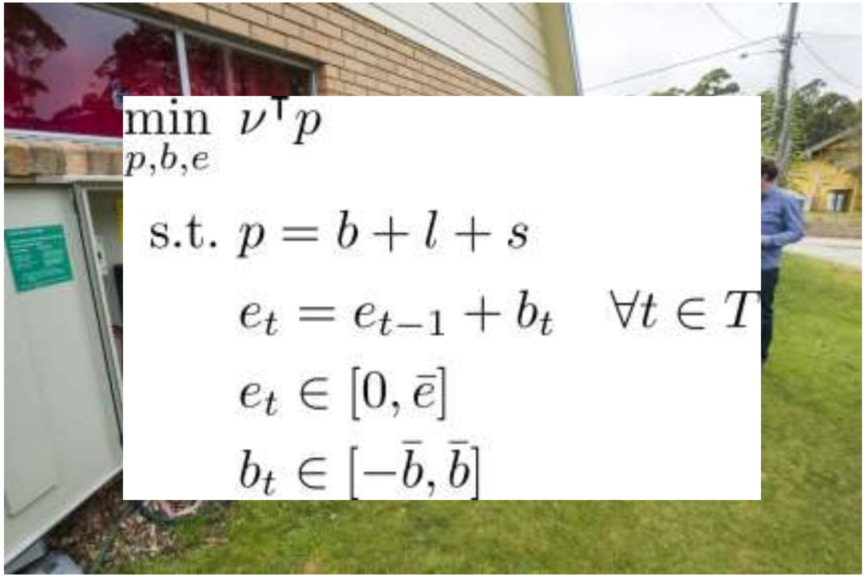
$$\min_{x \in X} f(x) = \min_{x \in X'} f(x) \qquad \arg\min_{x \in X} f(x) = \arg\min_{x \in X'} f(x)$$

For a convex problem we can relax or remove any constraint that is not active at the optimal solution. Similarly we can tighten any constraint up until the point it becomes active (remove the slack).



Battery Scheduling

Let’s demonstrate some of these points. Battery scheduling is a non-combinatorial scheduling problem, where we want to minimise our electricity costs over time.

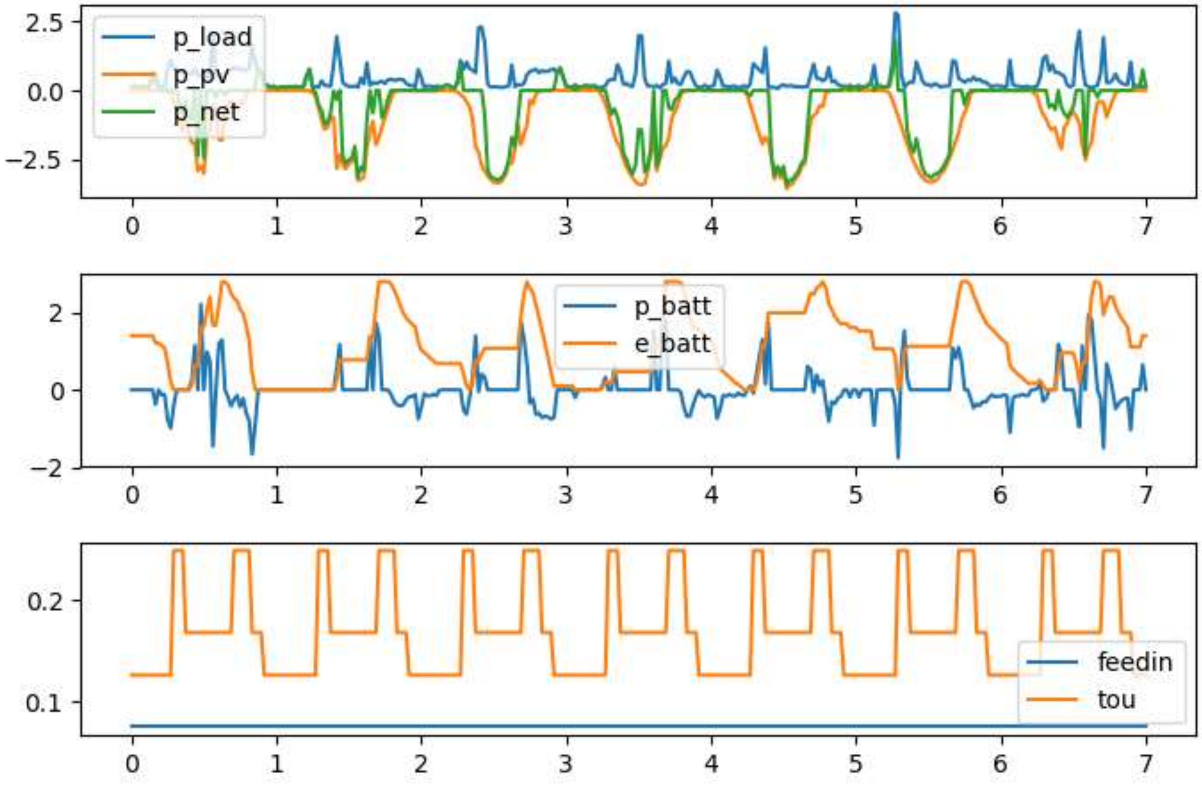


$$
\begin{aligned}
\min_{p,b,e} \quad & \nu^\intercal p \\
\text{s.t.} \quad & p = b + l + s \\
& e_t = e_{t-1} + b_t \quad \forall t \in T \\
& e_t \in [0, \bar{e}] \\
& b_t \in [-\bar{b}, \bar{b}]
\end{aligned}
$$

All vectors in $\mathbb{R}^{|T|}$

- ν electricity tariff ($/kWh)
- p total house power (kW)
- b battery power (kW)
- l load power (kW)
- s solar power (kW)
- e battery state of charge (kWh)

7 Day Optimal Battery Schedule

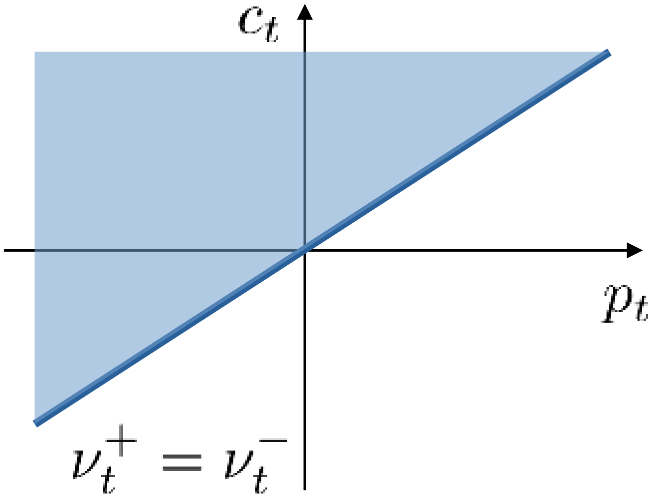


What if we want to have separate tariffs for buying and selling power?

Asymmetric Tariffs

$$\min \sum_t c_t \qquad c_t = \begin{cases} \nu_t^+ p_t & \text{if } p_t \geq 0 \\ \nu_t^- p_t & \text{otherwise} \end{cases} \qquad \nu_t^+, \nu_t^- \geq 0$$

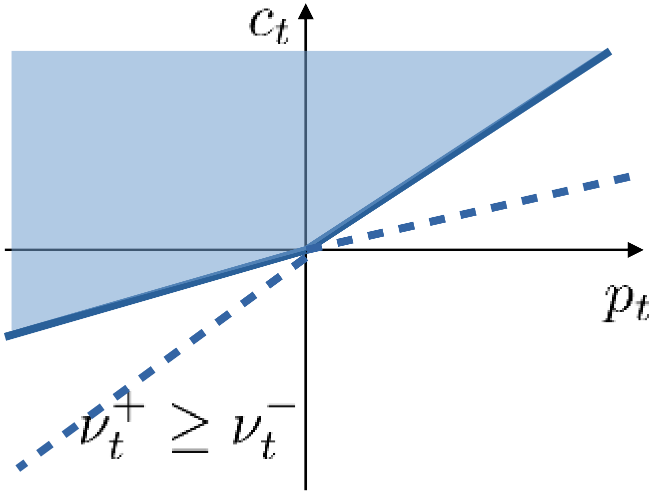
(plotting the 2 dimensions for this constraint)



$c_t = \nu_t^+ p_t$

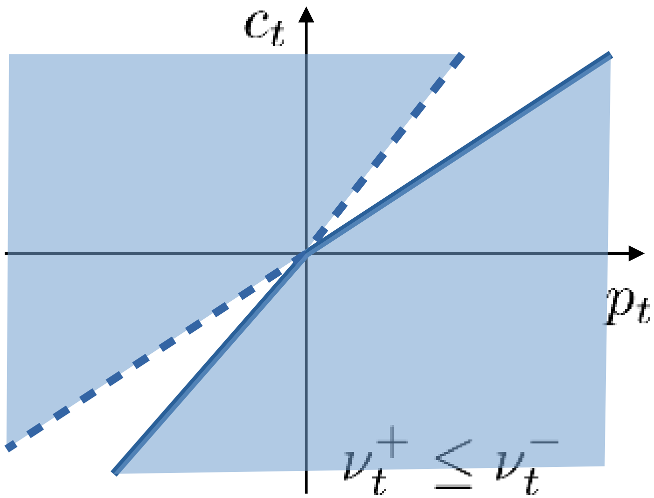
Still exact if relax:

$$c_t \geq \nu_t^+ p_t$$



Can't write linear equality (convexity?) but a gain exact answer if we relax:

$$c_t \geq \nu_t^+ p_t \qquad c_t \geq \nu_t^- p_t$$



Not a relaxation:

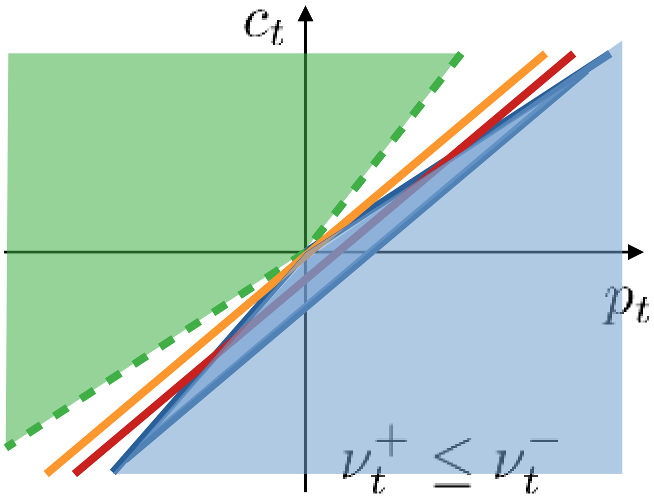
$$c_t \geq \nu_t^+ p_t \qquad c_t \geq \nu_t^- p_t$$

Unbounded result:

$$c_t \leq \nu_t^+ p_t \qquad c_t \leq \nu_t^- p_t$$

Asymmetric Tariffs

Fortunately, in practice the sell price is less than the buy price so we can relax the problem and getting an exact answer. What could we do if that were not the case?



- Use mixed-integer programming to model this non-convexity, and solve the problem exactly
- Find an approximate solution where the objective solution is an:
  - Upper bound
  - Better upper bound
  - Just approximate
  - Lower bound...

We need to “strengthen” our initial relaxation, maybe we can bound $\underline{p} \leq p \leq \bar{p}$ using $p = b + l + s$ and some extra info about b, l, s

Adding Battery Efficiency

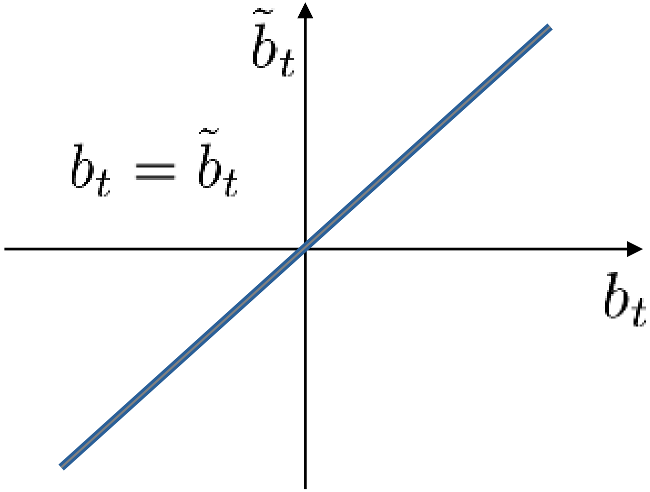
b external battery power (kW)

$\tilde{b}$ internal battery power (kW)

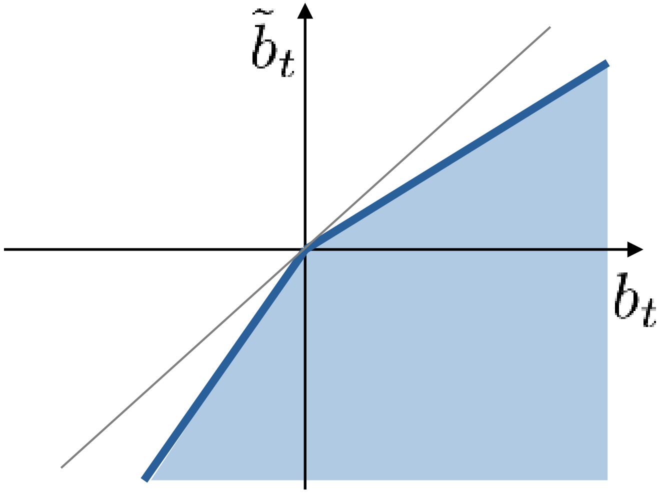
$$p = b + l + s$$

$$e_t = e_{t-1} + \tilde{b}_t$$

Efficiency = 100%



0% < Efficiency < 100%



$$\tilde{b}_t \leq \eta b_t$$

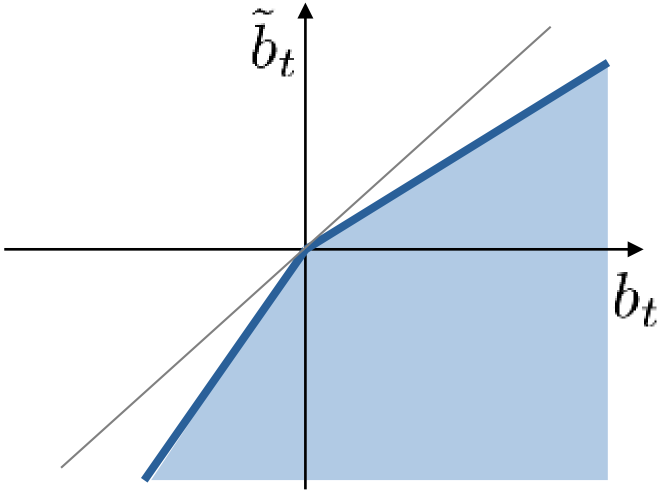
$$\tilde{b}_t \leq \frac{1}{\eta} b_t$$

"Roundtrip" efficiency is:

$$\eta^2$$

Adding Battery Efficiency

0% < Efficiency < 100%



$$\tilde{b}_t \leq \eta b_t$$

$$\tilde{b}_t \leq \frac{1}{\eta} b_t$$

"Roundtrip" efficiency is:

$$\eta^2$$

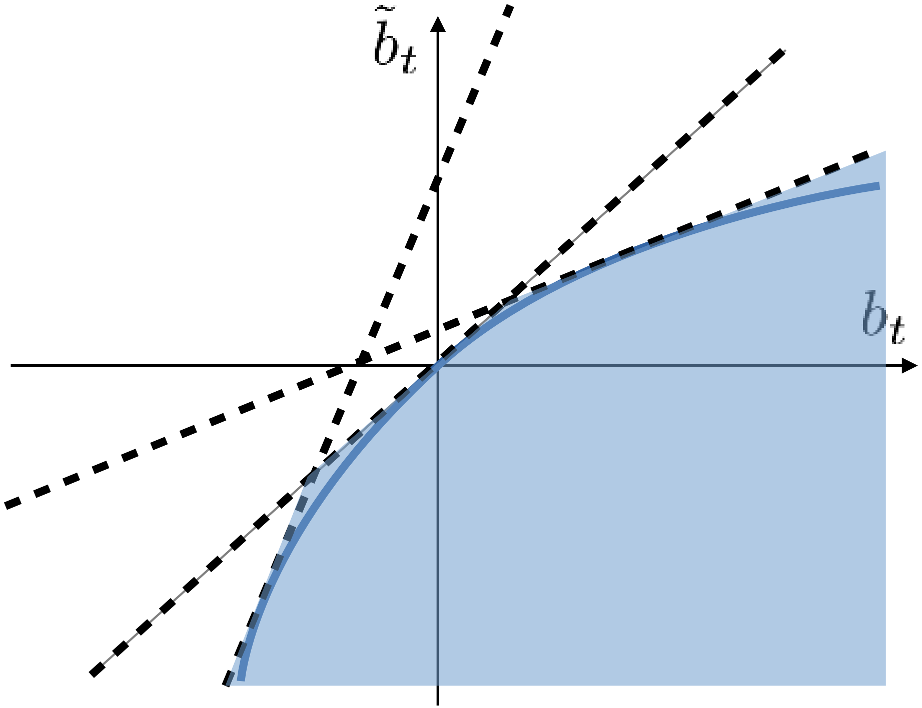
Will this lead to an exact solution?

So long as power always has some value to it. In our problem so long as:

- $\nu > 0$
- and there are enough time steps for the battery to empty if it starts off with an non-zero state of charge

Non-Constant Efficiency

What if instead the battery efficiency is non-constant, say if efficiency is monotonically decreasing with charge / discharge rate:



Relax the constraint by replacing it with linear inequalities tangent to surface at various points (choosing more constraints = more accuracy)

Approximate Solutions

We have an approximate solution, now what?

The first thing we can try to do is see if it is a **feasible solution to the original problem,** and if not, try to convert it to one (may or may not be easy: compare doing so for the asymmetric tariffs vs the battery efficiency).

If we get a feasible solution this way, it provides an upper bound on the true optimal solution.

Optimality Gap

Assume a minimisation problem.

LB / UB = lower / upper bound on objective

A = objective from approximate solution

F = objective of a feasible solution to the original problem, either obtained directly from one of the bound or approximate solutions, or somehow converted from them. These are also upper bounds.

Obtained	We know
LB	The optimal is at least this big
UB	The optimal is at most this big
A	Not much, optimal may or may not be similar
LB = F	We have an optimal solution!
LB < UB ∨ LB < F	The optimal objective must be in this range, the difference is known as the **optimality gap**.

Optimality Gap = F - LB If it is small, we might be happy with solution F

LP Topic Outline

- LP Introduction
- Modelling and solving
- Feasible region and convexity
- Simplex algorithm
- **Relaxations and approximations**
- **The dual of a linear program**
    - LP duality
    - Primal – Dual relationship
    - Interpretation
    - Dual variables from

The Dual of an LP

Every LP has a **dual** problem, which is equivalent in some important ways

A more general form of duality, **Lagrangian duality**, will be presented later on in the convex optimisation section

Duality can make it easier to solve particular problems, interpret the results in different ways, and enable new algorithms

Bounding an LP

Assuming the following problem has an optimal solution, let's try to derive lower bounds on the objective value:

$$\begin{aligned} \min_x \ & x_1 + 3x_2 \\ \text{s.t.}\ & 2x_1 + 9x_2 \geq 14 \\ & 2x_1 + 3x_2 \geq 10 \\ & x_1 + 2x_2 \geq 1 \\ & x \geq 0 \end{aligned}$$

Because the variables are nonnegative:

$$x_1 + 3x_2 \geq x_1 + 2x_2 \geq 1$$

Therefore objective is at least 1!

Divide both sides of constraint by 2:

$$x_1 + 3x_2 \geq x_1 + \frac{3}{2}x_2 \geq 5$$ Better!

Dividing by 3:

$$x_1 + 3x_2 \geq \frac{2}{3}x_1 + 3x_2 \geq \frac{14}{3} = 4.6\ldots$$

Nothing we didn't already know!

Bounding an LP

Assuming the following problem has an optimal solution, let's try to derive lower bounds on the objective value:

$$\begin{aligned} \min_x \quad & x_1 + 3x_2 \\ \text{s.t.} \quad & 2x_1 + 9x_2 \geq 14 \\ & 2x_1 + 3x_2 \geq 10 \\ & x_1 + 2x_2 \geq 1 \\ & x \geq 0 \end{aligned}$$

Combining first and second constraints and dividing by 4:

$$x_1 + 3x_2 \geq \frac{2+2}{4} x_1 + \frac{9+3}{4} x_2 \geq \frac{1}{4}(14+10) = 6$$

This turns out to be not just a lower bound, but the optimal value for the problem!

We can generalise this process of calculating the best lower bound for the objective, and it turns out it can be represented as ….. another LP!

We call this new LP the **dual problem**, with the original labelled the **primal problem**.

Dual LP Example

Primal problem:

$$\begin{aligned} \min_x \quad & x_1 + 3x_2 \\ \text{s.t.} \quad & 2x_1 + 9x_2 \geq 14 \\ & 2x_1 + 3x_2 \geq 10 \\ & x_1 + 2x_2 \geq 1 \\ & x \geq 0 \end{aligned}$$

Dual problem:

$$\begin{aligned} \max_y \quad & 14y_1 + 10y_2 + y_3 \\ \text{s.t.} \quad & 2y_1 + 2y_2 + y_3 \leq 1 \\ & 9y_1 + 3y_2 + 2y_3 \leq 3 \\ & y \geq 0 \end{aligned}$$

Introduce a non-negative scaling factor for each primal constraint, and combine the constraints (a linear combination of the constraints):

$$(2x_1 + 9x_2)y_1 + (2x_1 + 3x_2)y_2 + (x_1 + 2x_2)y_3 \geq 14y_1 + 10y_2 + y_3$$

We want y values such that:

$$x_1 + 3x_2 \geq (2x_1 + 9x_2)y_1 + (2x_1 + 3x_2)y_2 + (x_1 + 2x_2)y_3$$

which only holds true when y is appropriately constrained. We need to ensure the primal variable coefficients on the RHS are less or equal to the primal objective coefficients:

$$\begin{aligned} 2y_1 + 2y_2 + y_3 &\leq 1 \\ 9y_1 + 3y_2 + 2y_3 &\leq 3 \end{aligned}$$

The **dual problem**: find the largest lower bound subject to these constraints.

More Generally

Primal problem of form:

$$\begin{aligned} \min_x \ & c^\intercal x \\ \text{s.t. } & Ax \geq b \\ & x \geq 0 \end{aligned}$$

Into **Dual problem** of form:

$$\begin{aligned} \max_y \ & b^\intercal y \\ \text{s.t. } & A^\intercal y \leq c \\ & y \geq 0 \end{aligned}$$

We used the fact that when y is appropriately constrained: $c^\intercal x \boxed{\geq} y^\intercal A x \boxed{\geq} y^\intercal b$

What if $Ax \leq b \longrightarrow y \leq 0$ To swap inequality

What if $Ax = b \longrightarrow y \in \mathbb{R}^m$ Equality so the sign can be either

What if $x \leq 0 \longrightarrow A^\intercal y \geq c$ Coeffs of x in middle expr now need to be greater for first inequality to hold true

What if $x \in \mathbb{R}^n \longrightarrow A^\intercal y = c$ If x is free the only way for first inequality to hold is if x coeffs are exactly the same

What if max not min?

Rules for Getting the Dual

Dual of the dual of an LP is the primal.

The dual of a **min** i misation problem:

$$\min_x c^\intercal x \longleftrightarrow \max_y b^\intercal y$$

$$\begin{array}{lcl} x \geq 0 & \longleftrightarrow & A^\intercal y \leq c \\ x \leq 0 & \longleftrightarrow & A^\intercal y \geq c \\ x \in \mathbb{R}^n & \longleftrightarrow & A^\intercal y = c \end{array}$$

swap ineq.

$$\begin{array}{lcl} Ax \leq b & \longleftrightarrow & y \leq 0 \\ Ax \geq b & \longleftrightarrow & y \geq 0 \\ Ax = b & \longleftrightarrow & y \in \mathbb{R}^m \end{array}$$

same ineq.

The dual of a **max** imisation problem:

$$\boxed{\max_x} c^\intercal x \longleftrightarrow \boxed{\min_y} b^\intercal y$$

$$\begin{array}{lcl} x \geq 0 & \longleftrightarrow & \boxed{A^\intercal y \geq c} \\ x \leq 0 & \longleftrightarrow & \boxed{A^\intercal y \leq c} \\ x \in \mathbb{R}^n & \longleftrightarrow & A^\intercal y = c \end{array}$$

same ineq.

$$\begin{array}{lcl} Ax \leq b & \longleftrightarrow & \boxed{y \geq 0} \\ Ax \geq b & \longleftrightarrow & \boxed{y \leq 0} \\ Ax = b & \longleftrightarrow & y \in \mathbb{R}^m \end{array}$$

swap ineq.

Or remember one side, and convert max to min or min to max before taking dual

Primal and Dual Forms

Some more example forms for quick reference

Primal Problem $x \in \mathbb{R}^n$ primal variables	$\min_x \; c^\intercal x$ s.t. $Ax \leq b$	$\min_x \; c^\intercal x$ s.t. $Ax = b$, $x \geq 0$	$\min_x \; c^\intercal x$ s.t. $Ax \leq b$, $x \geq 0$
Dual Problem $y \in \mathbb{R}^m$ dual variables	$\max_y \; b^\intercal y$ s.t. $A^\intercal y = c$, $y \leq 0$	$\max_y \; b^\intercal y$ s.t. $A^\intercal y \leq c$	$\max_y \; b^\intercal y$ s.t. $A^\intercal y \leq c$, $y \leq 0$

Primal-Dual Relationships

Weak duality: Any feasible dual solution provides a lower bound on primal solutions, and any feasible primal solution provides an upper bound on dual solutions (assuming primal is minimising). This holds in general.

Primal SolutionObjective $\geq$ **Dual** SolutionObjective

Unbounded dual $\Rightarrow$ infeasible primal

Unbounded primal $\Rightarrow$ infeasible dual

It is also possible for both to be infeasible at the same time

Strong duality: Optimal solutions to primal and dual produce same objective value. Strong duality holds true for LPs (Farkas' Lemma for proof).

Primal OptimalObjective = **Dual** OptimalObjective

Some Intuition

$$\min_x \; c^\intercal x \qquad\qquad \max_y \; b^\intercal y$$

$$\text{s.t. } Ax \leq b \qquad\qquad \text{s.t. } A^\intercal y = c$$

$$y \leq 0$$

Intuitively, at the optimal solution, an inactive constraint should have no influence on the objective:

$A_i x^* < b_i$ so for dual problem $b_i y_i^* = 0 \quad (\implies y_i^* = 0)$

Conversely, if the objective is impacted the constraint must be active:

$b_i y_i^* \neq 0 \quad (\implies y_i^* < 0)$ so for primal problem $A_i x^* = b_i$

Therefore, when a constraint is actually influencing the optimal solution we expect a non-zero dual variable.

Some Intuition

$$\min_x \; c^\intercal x \quad \text{s.t. } Ax \le b$$

$$\max_y \; b^\intercal y \quad \text{s.t. } A^\intercal y = c, \quad y \le 0$$

If we add 1 unit to the RHS of the constraint, how much do we "anticipate" the objective to change by?

$A_i x \le b_i + 1$ $\qquad$ y_i Gives the anticipated change in objective

If one constraint is preventing a better solution more so than another, then we expect its corresponding dual variable will be larger in magnitude:

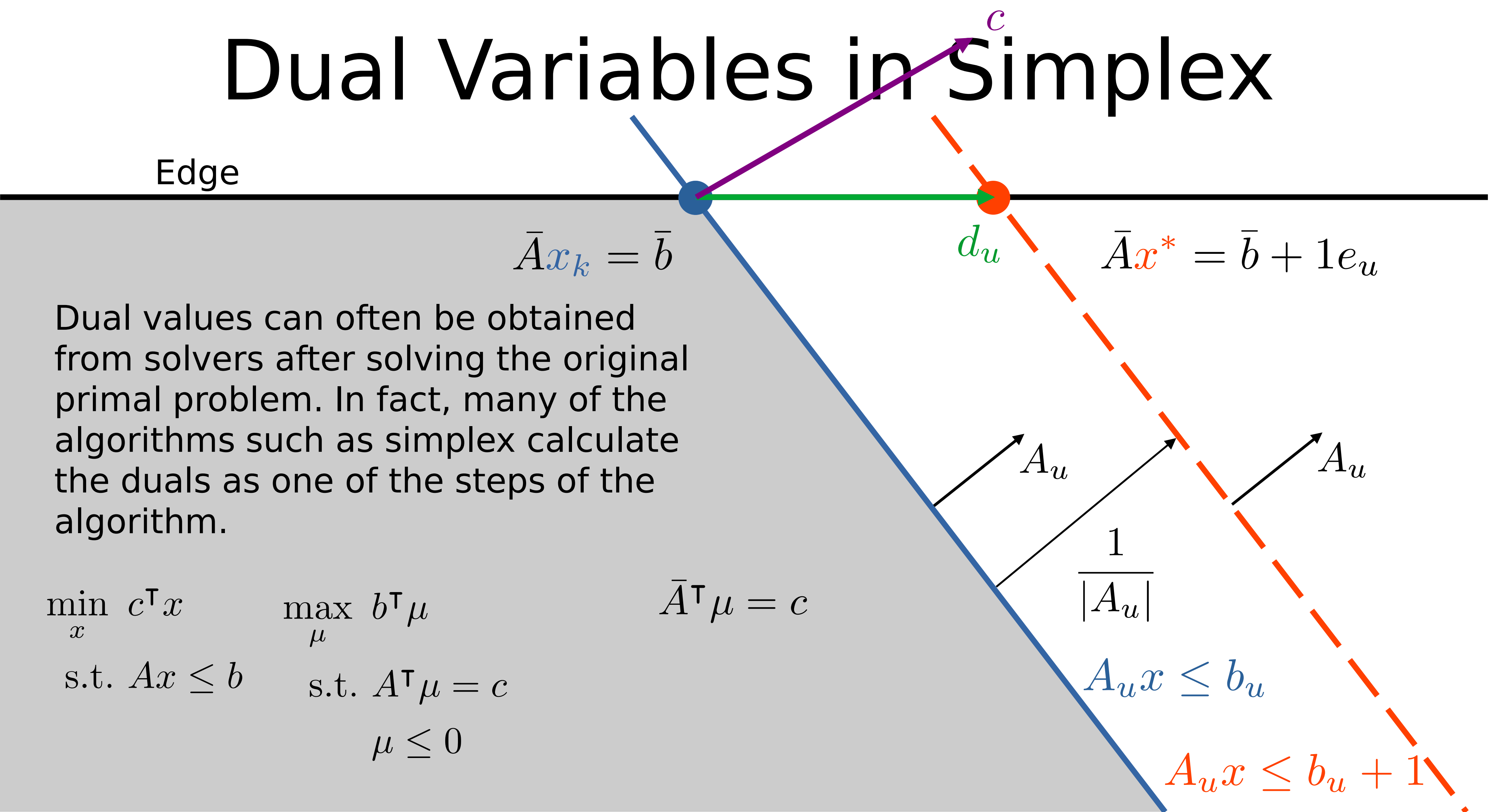
$$\begin{aligned} \min \;& x_1 + 5x_2 \\ \text{s.t. } & x_1 \ge 4 \quad (y_1 = 1) \\ & x_2 \ge 4 \quad (y_2 = 5) \end{aligned}$$

Are we always guaranteed to achieve this?

Imagine another constr. $x_1 \ge 3.9$

These duals are marginal values.

Dual Variables in Simplex



Dual values can often be obtained from solvers after solving the original primal problem. In fact, many of the algorithms such as simplex calculate the duals as one of the steps of the algorithm.

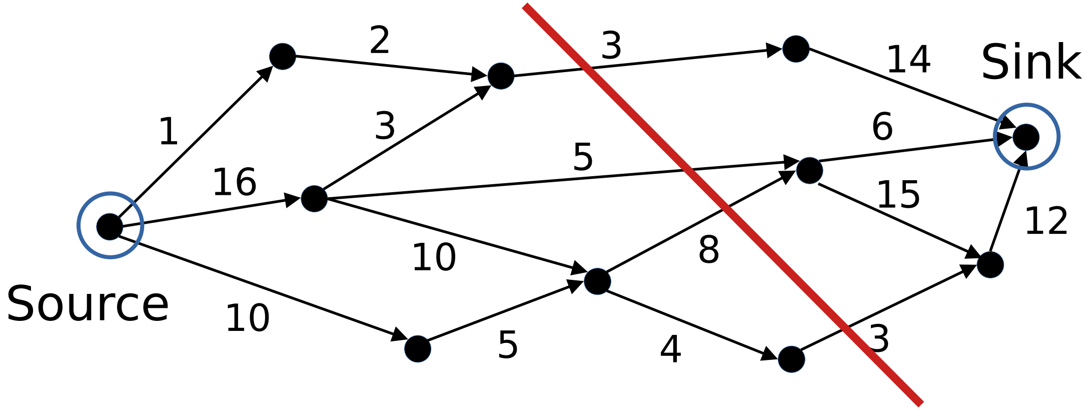
$$\min_x \; c^\top x \quad \text{s.t.} \; Ax \leq b$$

$$\max_\mu \; b^\top \mu \quad \text{s.t.} \; A^\top \mu = c, \quad \mu \leq 0$$

$$\bar{A}^\top \mu = c$$

Max Flow – Min Cut

The dual of some problems has a clear interpretation.



In max flow we just need to find the bottleneck in the network, which is equivalent to finding the cheapest edges to remove when capacity = cost of cutting edge

What is the dumbest UB (after ∞) we can easily calculate for max flow?

Sum all capacities

What is a better one?

Sum all capacities directly connected to one of the source / sink vertices

Max Flow: maximum amount of flow from a source to a sink over a graph where the edges have capacities.

Min Cut: minimum cost to separate a source from a sink by removing edges where each edge has a cost

Max Flow – Min Cut

Directed graph: $G := (V, E)$ Maximum flow between two vertices: (s, n)

Assuming there are no edges directly between the source and sink, and that the source only has exiting edges and the sink entering edges (reduces the conditions we need to consider for the dual formulation).

$$\max_x \sum_{(s,j)\in E} x_{s,j}$$

$$\text{s.t.} \sum_{(k,j)\in E} x_{k,j} - \sum_{(i,k)\in E} x_{i,k} = 0 \quad \forall k \in V \setminus \{s, n\} \qquad (y_k)$$

$$0 \leq x_{i,j} \leq c_{i,j} \quad \forall (i,j) \in E$$

$$(z_{i,j})$$

Graph is directed, and only allow flow in one direction (can be generalised)

Max Flow – Min Cut

After taking the dual:

$$\min_{y,z} \sum_{(i,j)\in E} c_{i,j} z_{i,j}$$

$$\text{s.t.} \quad -y_j + z_{s,j} \geq 1 \quad \forall (s,j) \in E$$

If uncut j must belong to source partition

$$y_i + z_{i,n} \geq 0 \quad \forall (i,n) \in E$$

If uncut i must belong to sink partition

$$y_i - y_j + z_{i,j} \geq 0 \quad \forall (i,j) \in E | i,j \neq s,n$$

If uncut receiving vertex can be no larger than sending vertex

$$z_{i,j} \geq 0 \quad \forall (i,j) \in E$$

$z_{i,j}$ represents the amount to cut the edge by:
0 is an uncut edge, 1 is a fully cut edge

y_k indicates whether a vertex belongs to the source or sink partition:
a value ≤ -1 is the source partition, ≥ 0 the sink partition

Economic Interpretation

If our objective is in $, then the dual variables will be some form of price:

$$\begin{aligned} \max \quad & 1.5x_{\text{cherry}} + 1.4x_{\text{lemon}} \\ \text{s.t.} \quad & x_{\text{cherry}} \leq 100 \\ & x_{\text{lemon}} \leq 200 \\ & x_{\text{cherry}} + x_{\text{lemon}} \leq 150 \end{aligned}$$

y_1 $ / cherry

y_2 $ / lemon

y_3 $ / can

What are the dual variable values?

$$\boxed{\begin{aligned} y_1 &= 0.1 \\ y_2 &= 0 \\ y_3 &= 1.4 \end{aligned}}$$

The dual variables may be referred to as **shadow** or **marginal prices.**
They indicate:

- the constraints that are most costly
- where we should alter the real world system to get the best marginal improvement

In more general contexts they might be called **Lagrange** or **KKT multipliers** (we will come to this later).

Hidden Duals

$$\min_{x \in \mathbb{R}^n} \quad c^\intercal x$$
$$\text{s.t.} \quad Ax \geq b$$
$$x \geq 0$$

The non-negativity requirements are constraints!
Why don't they also have dual values?

They do, but they disappear from the formulation.

$$\max_{y \in \mathbb{R}^m} \quad b^\intercal y$$
$$\text{s.t.} \quad A^\intercal y \leq c$$
$$y \geq 0$$

Why do they disappear from the objective of the dual problem?

Their corresponding "b" value is zero.

What about in the constraints?

$A^\intercal y \leq c$ introduce slack variables: $z \in \mathbb{R}^n_{\geq 0}$ $A^\intercal y + z = c$

There they are! $z = c - A^\intercal y$ (recall that if x was free $A^\intercal y = c$)

They are often called **reduced costs,** to distinguish them.

Dual Variables from Solver

In PuLP "pi" gives the dual of a constraint, "dj" gives the dual of a variable with bounds (reduced costs), "slack" gives the slack in a constraint:

```
c1 = x + y <= 10
m += c1

m. solve()

print ((c1. pi, c1.slack, x.dj))
```

Note that PuLP allows arbitrary lower and upper bounds to be defined for a variable, so the "dj" reduced costs are defined a bit differently from what we have presented. They are positive if the variable lower bound is active and negative when the upper bound is active.