AI Driven Decision Making

Introduction to Optimisation and Linear Programming

Today’s Outline

- Introduction to Optimisation
- Introduction to Linear Programming
- LP examples
- Understanding LP solutions graphically
- Solving LPs:
  - Simplex method
  - Interior point methods
- Software for Solving LPs

Introduction to Optimisation

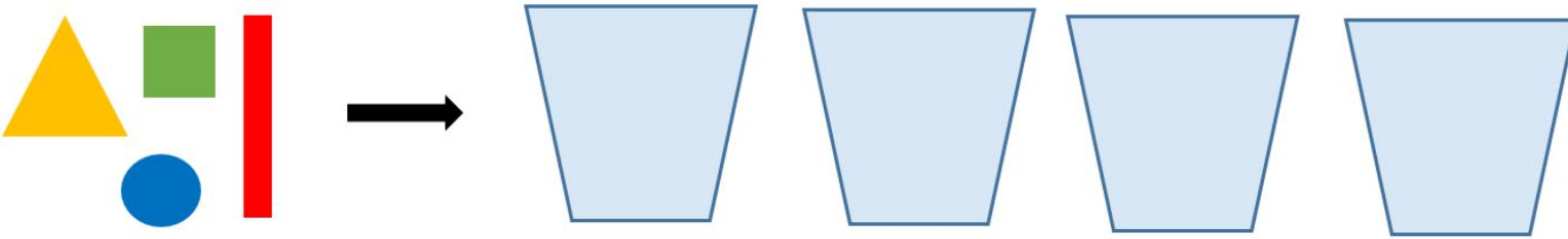
- What do we mean by “optimisation”?
  - It could mean a lot of things, for example speeding up:
    - computer programs
    - database queries
    - etc.

- This module: what we mean is *finding solutions to optimisation problems*.

- We are concerned with finding the optimal (or best) solution to a problem

Introduction to Optimisation

- Example:
    - Use the minimum number of bins to pack the shapes below



Introduction to Optimisation

- Both of the solutions below are valid solutions, but only figure 2 is optimal

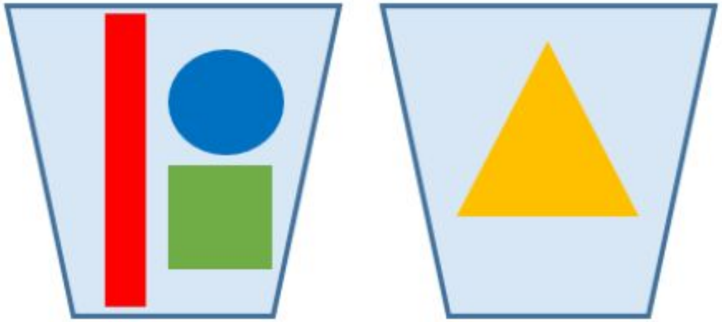


Fig 1: Solution A

Fig 2: Solution B (optimal)

Introduction to Optimisation

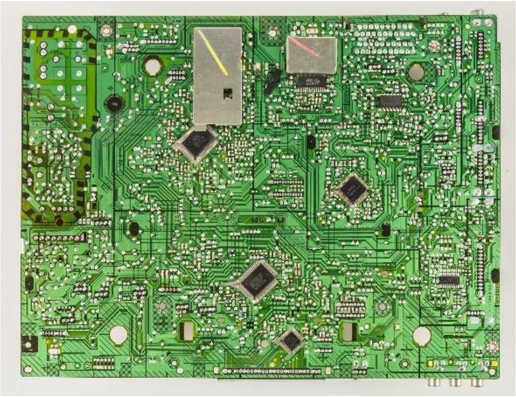
- As another example, suppose we want to timetable lectures for a university department. We want to assign each lecture to a room and a time.

- We have *constraints* such as
  - 2 lectures can't be given in the same room at the same time
  - a lecturer or student can't be in 2 rooms at the same time
  - every lecture must occur somewhere at some time
  - Dr Smith can't lecture on Tuesday mornings

- We often have something we want to optimise, usually called an *objective function* (it's our "objective" to optimise this function), for example
  - minimise the number of lectures given on Monday morning

Introduction to Optimisation

- There are *many* problems like this in industry, business, science, etc. Some example are:
  - Organizing a factory production line
  - Designing a circuit board
  - Delivering packages
  - Managing inventories...



Introduction to Optimisation

- We may want to minimise cost or risk, maximise profit, minimise delivery time, etc.

- All these are solved by optimisation techniques.

- Very many optimisation techniques developed by researchers in several fields

- But at the end, you should be able to model a wide range of problems and solve them using downloadable software, or write your own software for some types of problem.

Historical Perspectives

- The original optimisation field is Operations Research (OR).
- Began in the late 1930s in a systematic fashion
- World War 2
  - 1936: British RAF used OR to exploit large amount radar data being generated
  - 1939: Pre-war air defense exercise involving:
    - 33000 men
    - 300 aircraft
    - 110 anti-aircraft guns
    - 700 searchlights
    - 100 barrage balloons

  - The contribution made by the OR team was so impressive that teams were set up at other RAF commands

Historical Perspectives

- During WWII they continued to make great contributions, solving hard problems such as:
  - How many fighter squadrons should be sent to France
    - (none, which turned out to be the right decision as they were used instead to defend Britain)
  - Organizing flying and maintenance to make best use of squadron resources
    - 61% improvement by reorganizing the system
  - Improvement of “attack kill probability” on U-boats
    - Improvement from 2-3% to 40%, by considering the setting for depth charge explosion, the lethal radius of a depth charge, aiming errors etc)

Historical Perspectives

- The successes during WW2 led to OR being adopted in many non-military domains such as business, manufacturing and economics.

  - https://en.wikipedia.org/wiki/Operations_research

Linear Programming

- A linear program is a problem with a special form:

$$
\begin{aligned}
\text{maximise or minimise} \quad & \textstyle\sum_{j=1}^{n} c_j x_j \\
\text{subject to} \quad & \textstyle\sum_{j=1}^{n} a_{ij} x_j \{\leq, =, \geq\} b_i \\
& x_j \leq u_j \; (j = 1 \ldots n) \\
& x_j \geq 0 \; (j = 1 \ldots n)
\end{aligned}
$$

- It has:

1. Objective:
   - Goal statement
   - E.g. maximise profit or minimise cost

2. Constraints:
   - Rules that we must obey
   - E.g. don't make more items than we can sell, deliver all packages, etc.

Linear Programming

- A linear program is a problem with a special form:

$$
\begin{aligned}
\text{maximise or minimise} \quad & \textstyle\sum_{j=1}^{n} c_j x_j \\
\text{subject to} \quad & \textstyle\sum_{j=1}^{n} a_{ij} x_j \{\leq, =, \geq\} b_i \\
& x_j \leq u_j \ (j = 1 \ldots n) \\
& x_j \geq 0 \ (j = 1 \ldots n)
\end{aligned}
$$

- It has:

3. Decision variables:
    - Used to represent the decisions taken

4. Sets and Parameters:
    - Elements used in the objective and/or the constraints
    - E.g. the set of all variables.

Linear Programming

- We must choose values for the *decision variables* x_j that *satisfy* (do not *violate*) the linear *constraints*, the *upper bounds*, and the *non-negativity restrictions*, so that we get the greatest or least (depending on the problem) value for the *objective function*.

- The objective function may represent cost, profit, yield, or something else that we're trying to optimise.

- The constraints are things that must be true in all solutions, and may be inequalities ($\geq$, $\leq$) or equalities ($=$)

- The non-negativity restrictions occur in many problems, but sometimes we might have unrestricted variables. Some variables might also have no upper bound, though again they're commonly used.

- The a_{ij} , b_j , c_j , u_j are called the *parameters* of the problem, and are given: we have no control over them.

- A large number of real problems can be expressed in this way. Also, a strength of this approach is its simplicity: highly specialized software has been developed that can be used to solve LPs: we just have to model them.

- Also, the techniques used to solve such problems have been generalized to other problems, for example with non-linear constraints.

Decision Variables	
Continuous	$x \geq 0$
Integer	$y \in \{0, 1, 2, 3, 4, 5\}$
Mixed	$x \in [0, 10]$, $y \in \{0, 1\}$

Objectives and Constraints	
Linear	$3x + 2y = 0$
Quadratic	$x^2 - x + 1 \geq 0$
Convex	$\min e^x + e^y$
Non-convex	$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq 5$

An example

- A company produces 3 products P,Q,R. How much should they produce each week to maximise profits?

- Each product has different profit per unit sold, and only so much of each product can be sold in a given week:

	P	Q	R
profit per unit	45	60	50
maximum sales	100	40	60

An example

- To make things more complicated, the company can't simply make as much of any product as they want as they only have 4 machines:

machine	unit processing time (min) P	Q	R	machine availability (min)
A	20	10	10	2400
B	12	28	16	2400
C	15	6	16	2400
D	10	15	0	2400

- That is, each

An example

- One more piece of information: the plant costs 6000 euro a week to operate, which must be subtracted from our profits.

- The problem is to decide how much of each product to make during the week.

Modelling the problem

- First, what are the variables?
  - All we want to know here is how much of each product to make, so let's use 3 variables and call them P,Q,R.
  - If we decide to make 30 units of product P this week then variable P = 30, and so on.

- Objective?
  - We want to maximise total profit
  - Total Profit = Gross profit - Expenses
  - $45P + 60Q + 50R - 6000$

Modelling the problem

- So the (linear) objective is:

$$\text{maximise } 45P + 60Q + 50R$$

- Note that the 6000 isn't used, as it's not a function of the decision variables:
  - maximising $45P + 60Q + 50R - 6000$

  gives the same values for P,Q,R as:

  - maximising $45P + 60Q + 50R$.
- But we'll use it later to work out the actual profit (or loss).

Modelling the problem

- Constraints?
  - The market limitations (maximum sales above) tell us that we shouldn't make too much of any product, as we won't be able to sell them:

$$P \leq 100$$
$$Q \leq 40$$
$$R \leq 60$$

- Non-negativity restrictions?
  - We can't make negative amounts of products

$$P \geq 0$$
$$Q \geq 0$$
$$R \geq 0$$

Modelling the problem

- More constraints:
  - Machine limitations:
  - It takes machine A 20 minutes to make one unit of product P
  - P units take $20 \times P$ minutes to make

- Similarly for the other products and machines. Each machine can only make products for 2400 minutes in the week, so we write constraints:

$$
\begin{aligned}
20P + 10Q + 10R &\leq 2400 \quad \text{[machine A]} \\
12P + 28Q + 16R &\leq 2400 \quad \text{[machine B]} \\
15P + 6Q + 16R &\leq 2400 \quad \text{[machine C]} \\
10P + 15Q + 0R &\leq 2400 \quad \text{[machine D]}
\end{aligned}
$$

Our Model

$$
\begin{aligned}
\text{maximise} \quad & 45P + 60Q + 50R \\
\text{subject to} \quad & 20P + 10Q + 10R \leq 2400 \\
& 12P + 28Q + 16R \leq 2400 \\
& 15P + 6Q + 16R \leq 2400 \\
& 10P + 15Q + 0R \leq 2400 \\
& P \leq 100 \quad Q \leq 40 \quad R \leq 60 \\
& P \geq 0 \quad Q \geq 0 \quad R \geq 0
\end{aligned}
$$

- Notice that all the constraints and the objective function are linear.
- Solving with standard optimisation software gives:

$$
\begin{aligned}
P &= 81.82 \\
Q &= 16.36 \\
R &= 60
\end{aligned}
$$

Interpreting our results

- What do these figures mean for our problem?
  - If we feed these 3 values into the objective function we get 7664 euro. The actual profit is this figure minus 6000 (the plant cost) giving 1664 euro profit.
  - Assuming that all our assumptions were correct (about markets etc.) this is guaranteed to be the optimal solution.
- We can also use the 3 figures to check the machine usages.
  - For example machine A is used for $20P + 10Q + 10R$ units, giving exactly 2400 minutes of usage.
  - So is machine B.
  - But machine C is only used for 2285 minutes, and machine D for 1064 minutes.
  - So machines C and D are idle for some of the week.

Interpreting our results

- Also, notice that only product R is manufactured at a level that matches its market limit of 60.

- Fewer units of products P and Q (81.82 and 16.36) are made than the market could stand (100 and 40).

- The availability for machines A and B are *only just* satisfied by the optimal solution: that is, the left hand side of each isn't just to the right hand side, it's actually equal.

- In other words, if any more of P or Q were manufactured then constraints would be violated, and we would no longer have a *feasible solution* (satisfying all constraints) to the problem. (In LP any assignment to all variables is a *solution*.)
    - We call these constraints *tight* (or *active*).

- Similarly, the market constraint on R is tight: we could not manufacture any more of R without violating its market constraint.

- In contrast, the other constraints are not tight: we could make more of P, for example, and still have a feasible solution (though it wouldn't be an optimal solution).

Interpreting our results

- The tight constraints here represent bottlenecks for the manufacturing process.

- They tell us that, if the company could increase the availabilities of machines A and B, or increase the market for product R, then more profit might be made.

- On the other hand, increasing the availability of machines C and D, or the market for products P and Q, would not be worthwhile.

Abstractions

To model this problem as an LP we made some simplifying assumptions (“abstractions”):

- Integrality:
  - We cant manufacture a fraction of a product (e.g. 81.82 *units)*
  - Integrality constraints relaxed (more on this later)

- Linearity:
  - Real situations might be nonlinear (later)
  - Buying raw materials in bulk could reduce the amount spent

- Forecasting market demand with certainty:
  - Real-life market demands are uncertain and modelled using probability distributions, not exact figures.
  - Handling uncertainty is will be discussed later

- Many Other Things:
  - Even this simple model has highlighted some interesting aspects of the problem: the bottlenecks on product R, and machines A and B.
  - All models are wrong, but some are useful - **George E. P. Box**

Understanding LP solutions graphically

- Higher number of variables means higher dimensions, which is more difficult to visualize

- Visualisation is easier to understand in 2 dimensions, so lets keep only 2 variables, and set R to 60 (as in the optimal solution above)

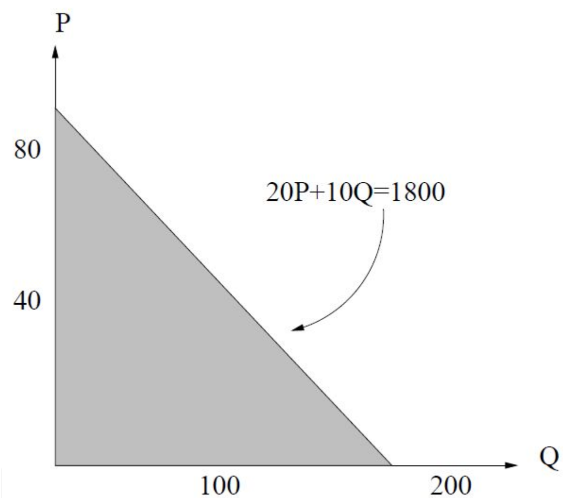
Understanding LP solutions graphically

- Remaining problem with only 2 variables P and Q:

$$
\begin{array}{lll}
\text{maximise} & Z = 45P + 60Q & \\
\text{subject to} & 20P + 10Q \leq 1800 & \text{(machine A)} \\
 & 12P + 28Q \leq 1440 & \text{(machine B)} \\
 & 15P + 6Q \leq 1440 & \text{(machine C)} \\
 & 10P + 15Q \leq 2400 & \text{(machine D)} \\
 & P \leq 100 \quad Q \leq 40 & \text{(market constraints)} \\
 & P \geq 0 \quad Q \geq 0 & \text{(non-negativity constraints)}
\end{array}
$$

Understanding LP solutions graphically

- We can use a graph to represent important features of the problem:



- All points in the space

$$20P + 10Q = 1800$$

and in the top-right quadrant are feasible for the machine A and non-negativity constraints

Understanding LP solutions graphically

- Coordinates for Machine A line are calculated as follows:

- Machine A: $20P + 10Q = 1800$

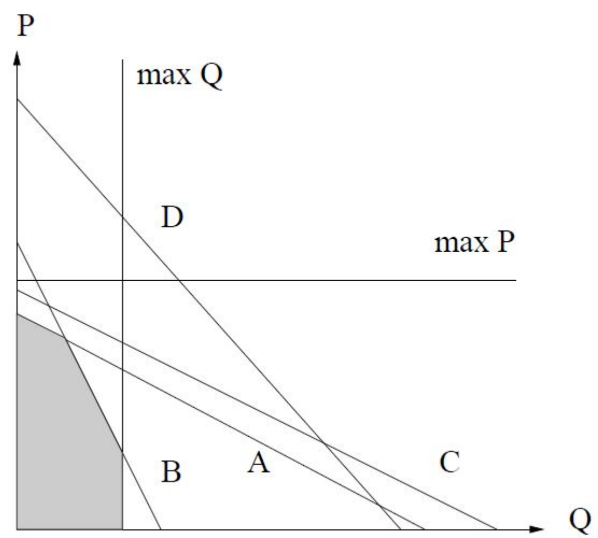
  - When $P = 0$: $Q = \frac{1800}{10} = 180$

  - When $Q = 0$: $P = \frac{1800}{20} = 90$

- Draw a straight line from $(P, Q) = (0, 180)$ to $(P, Q) = (90, 0)$

Understanding LP solutions graphically

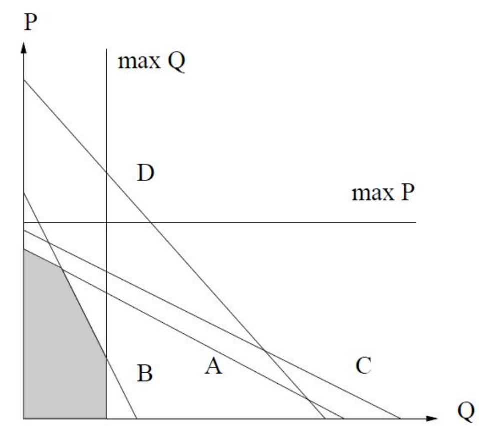
- All constraints can be graphed in this way
- Each constraint in the problem cuts out a different set of points, leaving a ***feasible*** region of the space called a *polyhedron*:



- We want to find a point in the polyhedron with the maximum objective function value, in other words an optimal feasible solution.

Understanding LP solutions graphically

- Notice that only some of the constraints affect the shape of the polyhedron:
    - those for machines A and B,
    - non-negativity
    - the market.



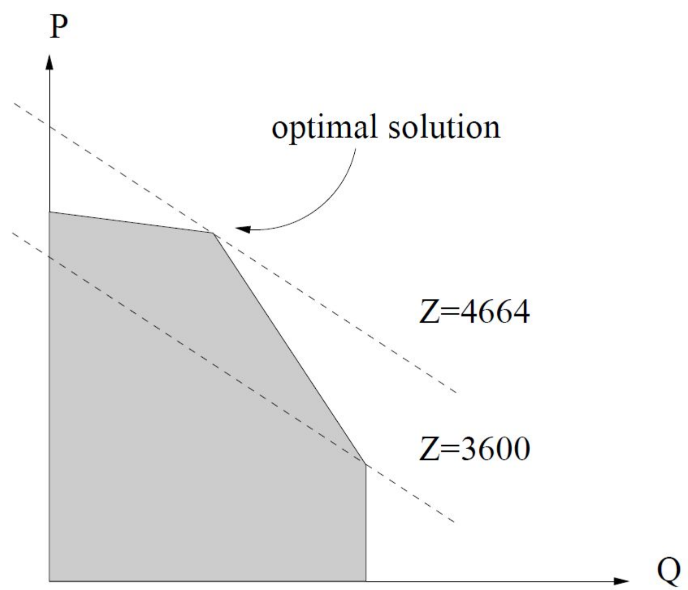
- If we removed the other constraints the polyhedron would be unchanged: we call these *redundant* constraints.
    - (They weren’t redundant in the original problem, but they are in this one.)

Understanding LP solutions graphically

- We can now see the ***feasible*** solutions: but how do we find an ***optimal feasible*** solution?
- Notice that for any value of the function Z the values of P,Q lie on a straight line, because the function is linear:

$$Z = 45P + 60Q.$$

- Here are the lines for $Z = 3600$ and $Z = 4664$, for example:

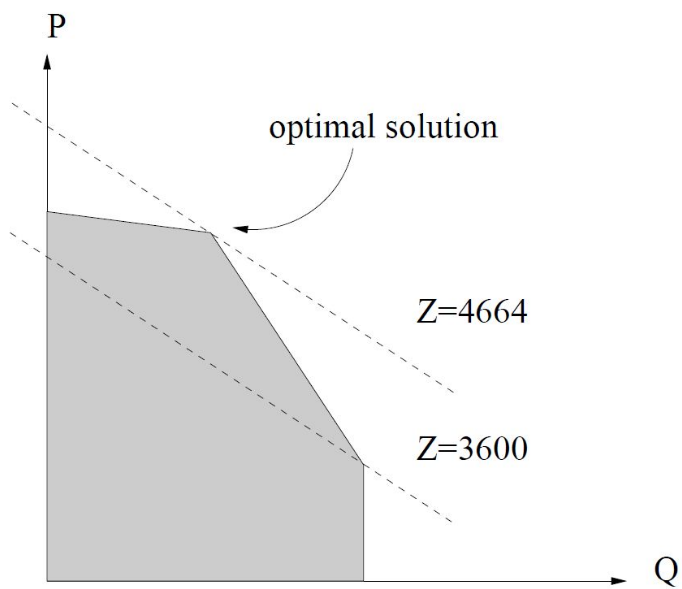


Understanding LP solutions graphically

- Notice that they're parallel: whatever Z we choose, we'll get a line parallel to these, called an ***isovalue contour***.
- Now we want a solution on one of these contours: unfortunately we don't know which one so there are an infinite number of possibilities.
- **But** the solution should have the greatest possible value of Z, so we can slide a contour downwards until it just touches the feasible solution: where it touches will be a feasible optimal solution.
- This fact is the key to solving LP problems: it can be shown that it touches will always touch *one of the vertices of the polyhedron*.
- There may be other optimal feasible solutions, but all we care about is finding one of them.

Understanding LP solutions graphically

- This bit of geometry shows that we need only check the Z value at each vertex of the polyhedron (the *extreme points* of the feasible region):
  - The one with greatest value is feasible and optimal.
- In our example we have only 5 such points. From an infinite number of possibilities, we've reduced the problem to a relatively easy one. (Of course it's more complicated with many variables.)



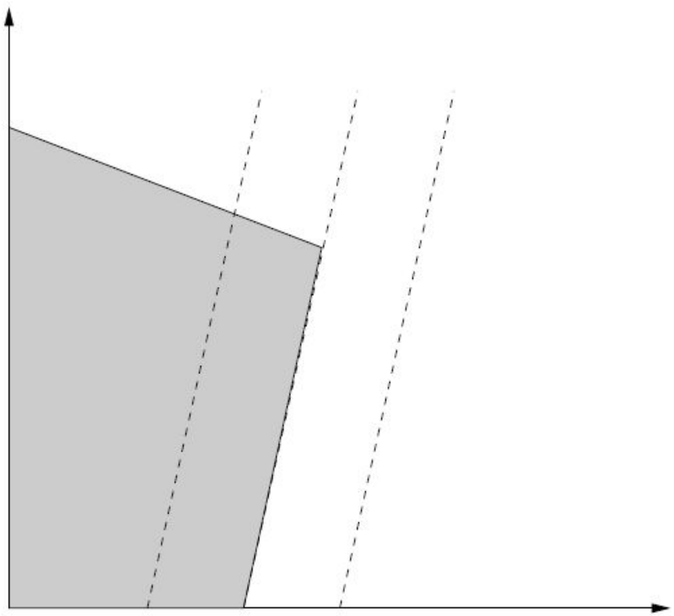
More graphical examples

- There are several things that might happen when we try to find an optimal feasible solution.
- One of them is as above: there are a finite number of such solutions, at the extreme points.
- Here's another possibility: the isovalue contours are parallel to one of the polyhedron edges. This occurs in the LP:

$$\begin{array}{ll} \text{maximise} & Z = 3X_1 - X_2 \\ \text{subject to} & 15X_1 - 5X_2 \leq 30 \\ & 10X_1 + 30X_2 \leq 120 \\ & X_1 \geq 0 \quad X_2 \geq 0 \end{array}$$

More graphical examples

- Here there are an infinite number of feasible optima, lying along the constraint boundary. But two of these are at vertices, so we still need to check only the vertices.

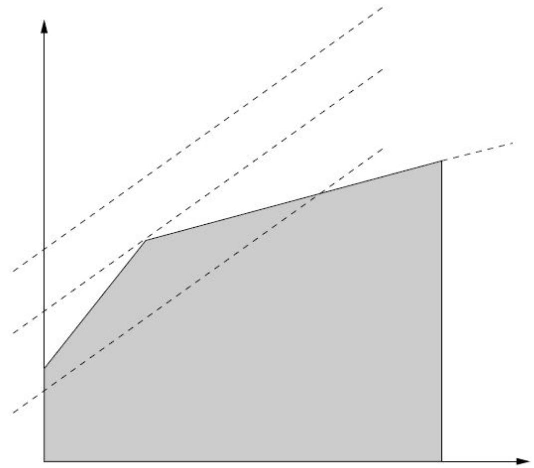


More graphical examples

- Another case is:

$$\begin{array}{ll} \text{maximise} & Z = -X_1 + X_2 \\ \text{subject to} & -X_1 + 4X_2 \leq 10 \\ & -3X_1 + 2X_2 \leq 2 \\ & X_1 \geq 0 \quad X_2 \geq 0 \end{array}$$

- Here the feasible region is unbounded, but the optimal solution is finite.

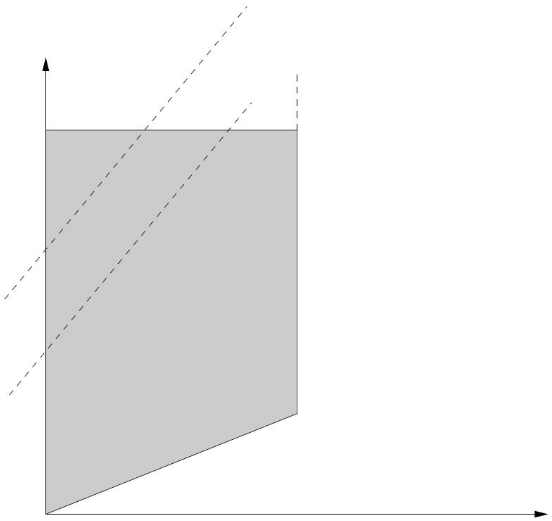


More graphical examples

- Another case:

$$
\begin{aligned}
\text{maximise} \quad & Z = -X_1 + X_2 \\
\text{subject to} \quad & -X_1 + 4X_2 \leq 10 \\
& X_1 \leq 4 \\
& X_1 \geq 0 \quad X_2 \geq 0
\end{aligned}
$$

- The optimal solution is unbounded: X1 has an upper bound but X2 doesn’t. (If all variables are bounded then the feasible solution is always finite.)

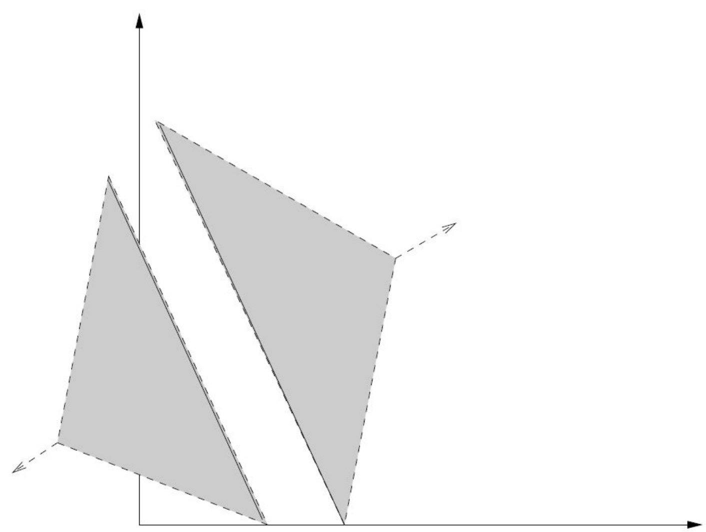


More graphical examples

- Another case:

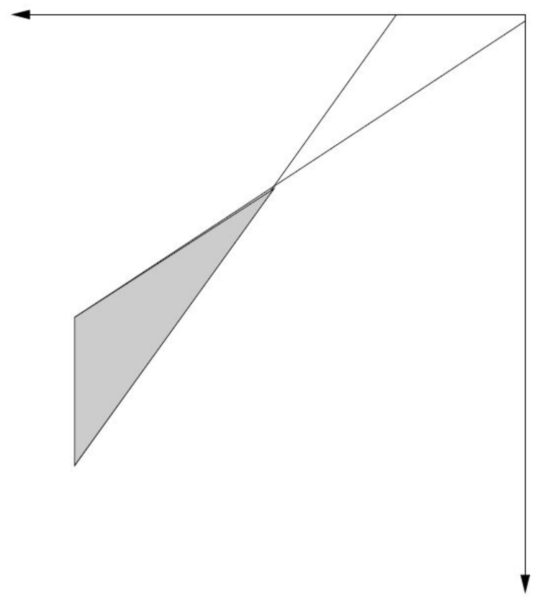
$$\begin{array}{ll} \text{maximise} & Z = -X_1 + X_2 \\ \text{subject to} & 3X_1 + X_2 \geq 6 \\ & 3X_1 + X_2 \leq 3 \\ & X_1 \geq 0 \quad X_2 \geq 0 \end{array}$$

- The constraints are inconsistent, and there's no feasible region



More graphical examples

- Something else that can occur is that the constraints, apart from the non-negativity ones, are consistent but not in the positive quadrant: (X1,X2 positive).



More graphical examples

- If these last 3 cases occur in practice, the reason is probably an error by the modeller: omitted constraints, typos, etc.

- Real problems are usually feasible and bounded by resource limits. But it can be hard to spot these cases when there are more than 2 variables, as we can’t visualise them: the feasible region is an n-dimensional polyhedron, and its boundary is a hyperplane.

- But we still find optimal solutions at the extreme points of the feasible region, if they exist.

Solving LPs

- There are several algorithms, but the best-known algorithm is the ***Simplex algorithm***. Its development in the 1940s was a landmark in OR and it's been greatly improved since.

- Simplex uses the method mentioned already: it searches the boundary of the feasible region to find an optimal feasible solution, which must occur at an extreme point.

- The latest algorithms don't do this: instead they follow a path through the feasible region to an optimum. These ***interior point methods*** have an important theoretical property: they always solve the LP in a time that's polynomial in the problem size.

- It's not known if Simplex is always polynomial, but in practice Simplex is often most efficient, and is used in almost all commercial optimisation packages.

- Interior point methods seem to be best on large, *sparse* problems (in which each constraint contains only a few of the variables, so that the matrix of coefficients is sparse).

Software

MATLAB

Maple™

GUROBI OPTIMIZATION

AIMMS

GAMS

AMPL

SciPy

IBM ILOG CPLEX Optimization Studio