Optimising the real world, robustly

An introduction to robust optimisation

Dr Andy Harrison
Director, Analytics Consulting
FICO

Agenda

- What is robust optimisation?
  - Why does it matter?
- A real world example
- Some simple examples
  - Shortest path
  - Knapsack

Pre-requisites

- Linear and mixed integer programming
- Modelling and implementation
- Desirable
  - Mosel modelling language
  - Installation of Xpress 7.7 or later

What is robust optimisation?

Why does it matter?

Why does robust optimisation matter?

- Optimisation problems often use data that are subject to uncertainty
  - inaccurate, erroneous, missing measurements
  - data that are not yet known
  - rely on estimates or extrapolations of historic data
  - influenced by external events not captured by the model
- How can we make decisions without knowing the parameters?

FICO

Why does robust optimisation matter?

- Key idea: although some parameters are uncertain, we might know how they vary
  - the accuracy of a given parameter is $\pm\varepsilon = 10^{-4}$
  - a vector of parameters $\boldsymbol{u}$ has a mean $\hat{\boldsymbol{u}}$ and covariance matrix $\hat{\boldsymbol{Q}}$
- Some parameters are uncertain but we have good knowledge of the uncertainty set
- Robust optimisation finds a solution that is feasible **for any/all** values of the parameters in the uncertainty set

Why does robust optimisation matter?

- Robust optimisation is about finding the optimal solution of the worst case realisation of a problem
  - Accounting for data quality and forecast distribution during the optimisation process
- Describing the worst case scenario is not trivial
  - Infinite number of realisations
  - Unknown realisation space

FICO

Why does robust optimisation matter?

- Possible approaches
  - build into the model formulation
  - use suitable solution methods
  - post process results
- Robust optimisation provides a modelling paradigm that offers solutions when **uncertainty in the input data** can be bounded within a well described region
  - finds a solution that is feasible regardless of the realisation of the uncertain values
  - different from Stochastic Optimisation where the expected value is optimised

Robust optimisation in a nutshell (mathematical concepts)

- Deterministic problem
  - $\min_{x \in \mathbb{R}^n} \{c.x \mid A.x \geq b, D.x \geq l\}$
    - A is the structural data of the problem. It is known for sure.
    - D is an approximation of the real world.
- Robust counterpart
  - $\min_{x \in \mathbb{R}^n} \{c.x \mid A.x \geq b, (D + \xi a).x \geq l, \forall \xi \in \mathrm{U}\}$
  - $\min_{x \in \mathbb{R}^n} \left\{c.x \mid A.x \geq b, \min_{\xi \in U} \{(D + \xi a).x\} \geq l\right\}$
    - U is the set of perturbations of non-structural data
    - ξa is the error term
    - x must be feasible for all possible perturbation
    - U must be carefully designed to be tractable

FICO

Robust optimisation in a nutshell (constraints)

- Deterministic constraint

$$a_1 x_1 + \ldots + a_{k-1} x_{k-1} + a_k x_k + \ldots + a_n x_n \leq b$$

where

a_i – coefficient

x_i – decision variable

Robust optimisation in a nutshell (constraints)

- Deterministic constraint

$$a_1x_1 + \dots + a_{k-1}x_{k-1} + a_kx_k + \dots + a_nx_n \le b$$

where

a_i – coefficient

x_i – decision variable

- Robust constraint

$$a_1x_1 + \dots + a_{k-1}x_{k-1} + (a_k + u_k)x_k + \dots + (a_n + u_n)x_n \le b$$

where

u_i – uncertainty on coefficient values

Robust optimisation in a nutshell (constraints)

- Deterministic constraint

$$a_1 x_1 + \ldots + a_{k-1} x_{k-1} + a_k x_k + \ldots + a_n x_n \leq b$$

where

a_i – coefficient

x_i – decision variable

- Robust constraint

$$a_1 x_1 + \ldots + a_{k-1} x_{k-1} + (a_k + u_k) x_k + \ldots + (a_n + u_n) x_n \leq b$$

where

u_i – uncertainty on coefficient values

- Requirement: the uncertainty must be bounded

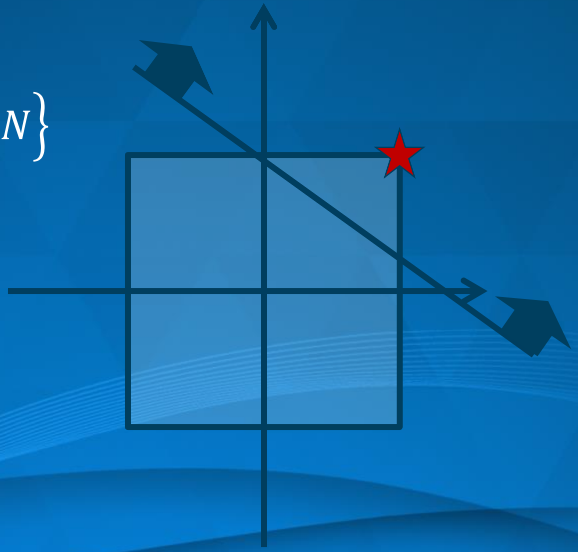
Robust optimisation in a nutshell (terminology)

- The feasible region of the **uncertains** is called the **uncertainty set**, it is modelled by means of constraints on the uncertains
  - the type of a robust constraint is defined by the type of uncertainty set (or sets) on its uncertains
  - solvability depends on whether the robust constraints can be transformed into a form (the so-called **robust counterpart**) that can be solved by the available mathematical solvers

FICO

Robust optimisation in a nutshell (robust counterparts)

- The Box
  - Lower and upper bounds on possible value range
    - $U = \left\{\xi : abs(\xi_i) \leq d, \xi_i \in \left[\underline{\xi_i}, \overline{\xi_i}\right], \forall i \in N\right\}$
  - Worst error term in a greater-than-equal constraint
    - $rc(x) = \min_{\xi \in U} \left\{(\xi a).x : abs(\xi_i) \leq d,\ \xi_i \in \left[\underline{\xi_i}, \overline{\xi_i}\right], \forall i \in N\right\}$
    - $x \geq 0, d \geq 0$
  - Closed form solution
    - $rc(x) = (\tilde{\xi} a).x$
    - With $\tilde{\xi}_i = \min\left(\max\left(\underline{\xi_i}, -d\right), \min\left(\overline{\xi_i}, d\right)\right)$



Trivial

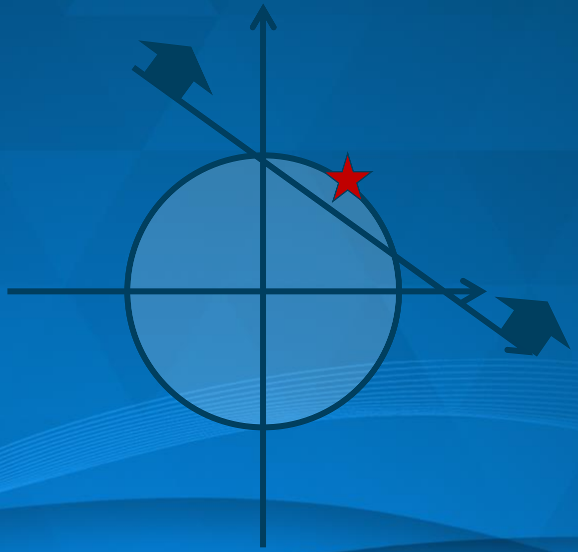
FICO

Robust optimisation in a nutshell (robust counterparts)

- The Ellipsoid
  - Maximum distance or deviation
    - $U = \left\{\xi : \sum_i {\xi_i}^2 \le d^2, \forall i \in N\right\}$

  - Worst error term in a greater-than-equal constraint
    - $\mathrm{rc}(x) = \min_{\xi \in U}\left\{(\xi a).x : \sum_i {\xi_i}^2 \le d^2\right\}$
    - $x \ge 0$

  - Closed form solution
    - $rc(x) = -d\sqrt{\sum_i (a_i.x_i)^2}$

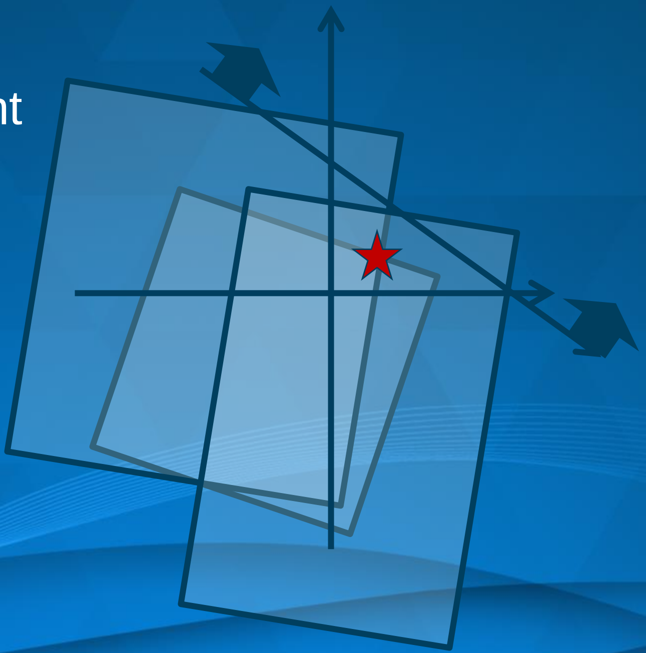


Trivial

FICO

Robust optimisation in a nutshell (robust counterparts)

- The Polyhedron
  - Linear dependency and bounds of error term
    - $U = \{\xi : \sum_i \xi_i \leq d, \forall i \in N\}$
  - Worst error term in a greater-than-equal constraint
    - $\mathrm{rc}(x) = \min_{\xi \in U}\{\xi . x : \sum_i \xi_i \leq d\}$
    - $x \geq 0$
  - Robust counterpart
    - Assume $x \geq 0$, and fixed
    - Apply LP Strong duality



Non-trivial

FICO

A real-world case: air products and chemicals[1]

- Production planning at a liquid oxygen/nitrogen plant, a very energy-intensive operation
  - Interruptible Load Contract (ILC): power company can suspend supply in periods of high demand (summer)
  - At most k interruptions each month (8 hours each)
  - **Cheaper** (per kWh) than with uninterrupted contract
- The power supplier won't tell us when the interruptions will be
  - Treat the interruptions as uncertains
  - Plan production so that even with the most evil-placed k interruptions we satisfy customer demand

[1] Latifoglu, C., Belotti, P., Snyder, L.V. (2013). Models for production planning under power interruptions. *Naval Research Logistics* **60**(5):413-431.

FICO

Original model

```
declarations
          produce: array (PERIODS, GASES) of mpvar
          inventory: array (PERIODS, GASES) of mpvar

end-declarations
forall(t in PERIODS, g in GASES) do
          inventory(0, g) + sum(tp in PERIODS | tp <= t)
                    produce(tp, g) - DEMAND(tp, g)) >= 0
          inventory(t, g) <= INV_CAP (g)
          produce(t, g) <= PROD_CAP (g)
end-do

minimize(sum (t in PERIODS, g in GASES)
                    (PROD_COST * produce(t,g) + INV_COST * inventory(t,g)))
```


Robust model

```
declarations
        produce: array (PERIODS, GASES) of mpvar
        inventory: array (PERIODS, GASES) of mpvar
        interrupt: array (PERIODS) of uncertain
end-declarations
forall(t in PERIODS, g in GASES) do
        inventory(0, g) + sum(tp in PERIODS | tp <= t)
                ((1 - interrupt(tp)) * produce(tp, g) - DEMAND(tp, g)) >= 0
        inventory(t, g) <= INV_CAP (g)
        produce(t, g) <= PROD_CAP (g)
end-do
sum(t in PERIODS) interrupt (t) <= MAX_NINTERR
minimize(sum (t in PERIODS, g in GASES)
                (PROD_COST * produce(t,g) + INV_COST * inventory(t,g)))
```

The user vs. opponent

User vs. opponent perspective

- Robustness implies we are prepared against ***any*** realisation of $\boldsymbol{u}$.
  - We (the user) have power over the decision variables x
  - Uncertain parameters $\boldsymbol{u}$ are ***not*** in our control
  - An opponent controls $\boldsymbol{u}$:
    - nature
    - competitor, supplier or customer
    - market
  - Akin to a *leader-follower game*: we (leader) make a decision on $\boldsymbol{x}$ and the opponent (follower) gets to choose $\boldsymbol{u}$ after we made our decision
    - the opponent has a PhD in optimisation and will pick $\boldsymbol{u}$ that violate the user's constraints

FICO

Robust shortest path

A simple example

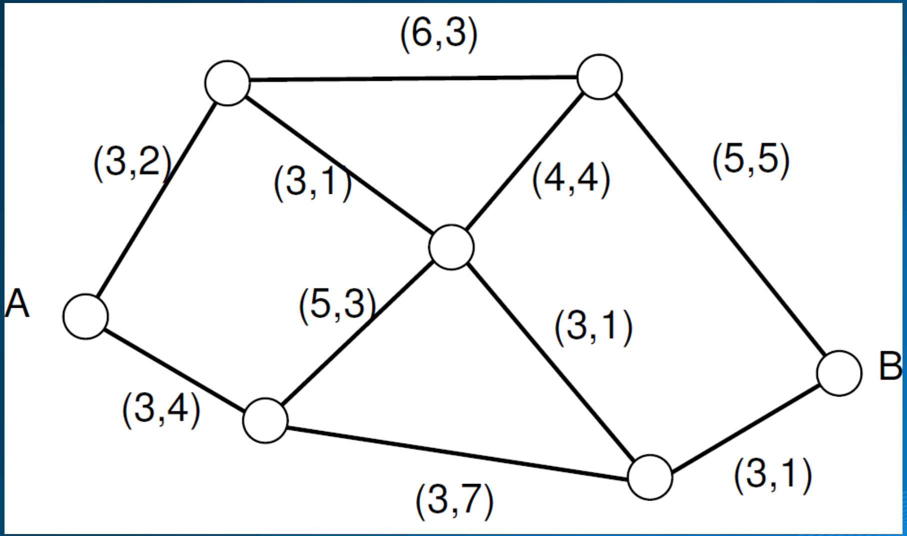
Robust shortest path

- Find the shortest route from A to B on the city's road network
  - it takes c_e minutes to drive on road e
  - unless there's construction work, and then it's $c_e + d_e$
  - we don't know where the construction work is
  - but we know it is on at most k roads
- **Decider**: the user
- **Opponent**: city's contractors, with k crews working every day

FICO

Robust shortest path

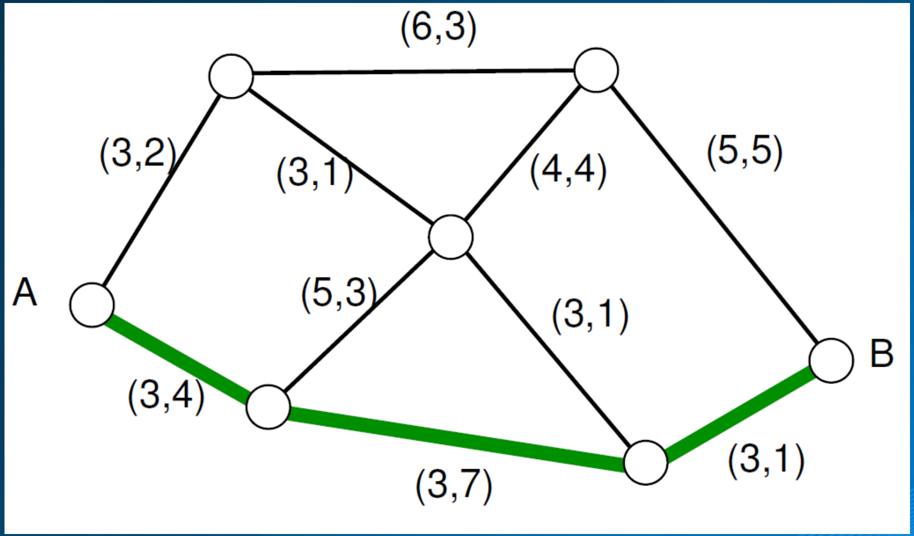
- Each link e has (c_e, d_e). Suppose at most $k = 2$ construction zones.



- How long to get from A to B?

Robust shortest path

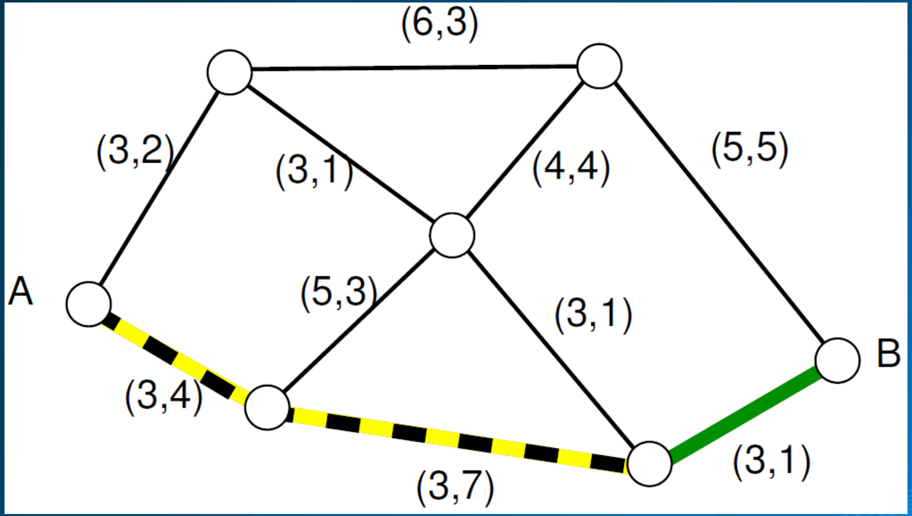
- Each link e has (c_e, d_e). Suppose at most k = 2 construction zones.



- How long to get from A to B?

Robust shortest path

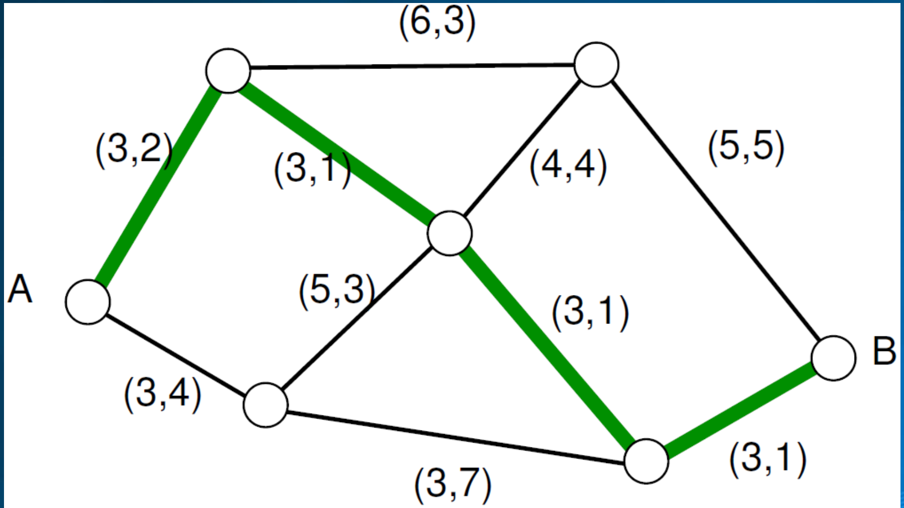
- Each link e has (c_e, d_e). Suppose at most $k = 2$ construction zones.



- How long to get from A to B?

Robust shortest path

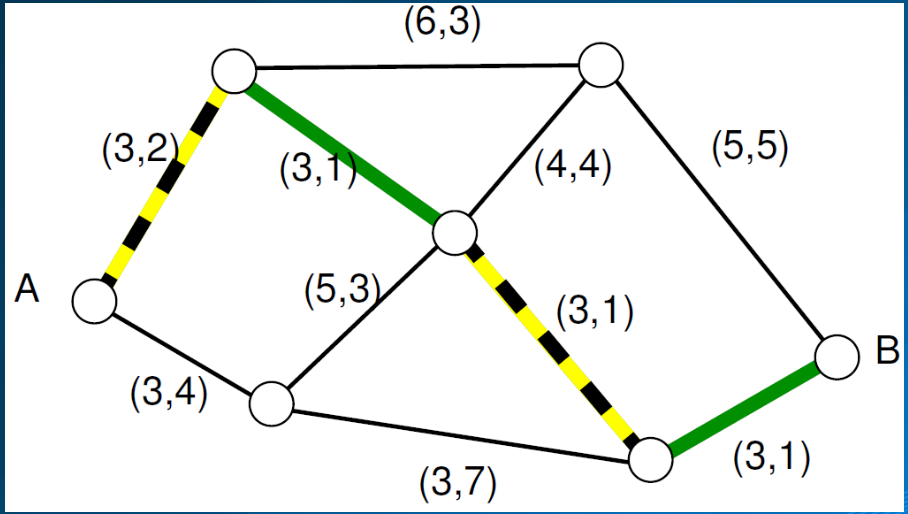
► Each link e has (c_e, d_e). Suppose at most $k = 2$ construction zones.



► How long to get from A to B?

Robust shortest path

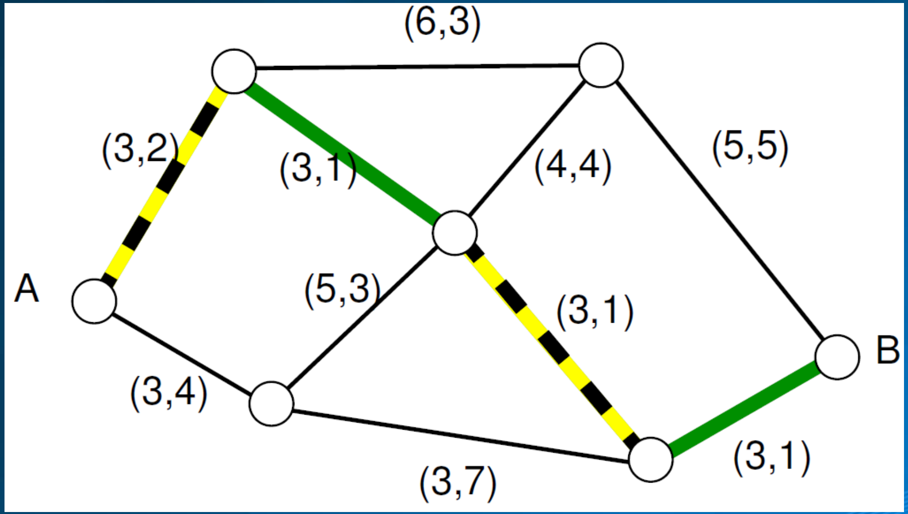
► Each link e has (c_e, d_e). Suppose at most $k = 2$ construction zones.



► How long to get from A to B?

Robust shortest path

- Each link e has (c_e, d_e). Suppose at most $k = 2$ construction zones.



- An example implementation: roadworks.mos

Robust knapsack

Project selection

Robust knapsack – project selection

- Select among five projects such that
  - total profit is maximised (each project has profit p_i)
  - total cost is within budget B
  - cost of each project: c_i = a_i + u_i , with u_i uncertain

minimise $p_1 x_1 + p_2 x_2 + p_3 x_3 + p_4 x_4 + p_5 x_5$

s.t. $c_1 x_1 + c_2 x_2 + c_3 x_3 + c_4 x_4 + c_5 x_5 \leq B$

$x_1, x_2, x_3, x_4, x_5 \in \{0,1\}$

Robust knapsack – project selection

- Suppose we know $U = \left\{u_i \geq 0, i = 1,2, \ldots, 5, \sum_{i=1}^{5} u_i \leq 0.04\right\}$
  - then we want to solve the following robust counterpart

$$\begin{array}{ll} \text{minimise} & \sum_{i=1}^{5} p_i x_i \\ \text{s.t.} & max\left\{\sum_{i=1}^{5} (a_i + u_i) x_i\right\} \leq B \\ & x_1, x_2, x_3, x_4, x_5 \in \{0,1\} \end{array}$$

- Alternatively, $\boldsymbol{u}$ has mean 0 with covariance matrix Q and confidence level α, i.e. $\boldsymbol{u}^T Q \boldsymbol{u} \leq \alpha$

$$\begin{array}{ll} \text{minimise} & \sum_{i=1}^{5} p_i x_i \\ \text{s.t.} & max_{\boldsymbol{u}:\boldsymbol{u}^T Q \boldsymbol{u} \leq \alpha} \left\{\sum_{i=1}^{5} (a_i + u_i) x_i\right\} \leq B \\ & x_1, x_2, x_3, x_4, x_5 \in \{0,1\} \end{array}$$

Robust knapsack – project selection

- Another alternative: we don't have a model for $\boldsymbol{u}$, but we have historical data: values for $\boldsymbol{u}$ for the past 12 years: $\boldsymbol{u}^{2013}$, $\boldsymbol{u}^{2012}$, …, $\boldsymbol{u}^{2002}$.
  - We at least require that our constraint be satisfied for the past values of **u**.

$$
\begin{array}{ll}
\text{minimise} & \sum_{i=1}^{5} p_i x_i \\
\text{s.t.} & \sum_{i=1}^{5} (a_i + u_i{}^{2013}) x_i \leq B \\
& \sum_{i=1}^{5} (a_i + u_i{}^{2012}) x_i \leq B \\
& \vdots \\
& \sum_{i=1}^{5} (a_i + u_i{}^{2002}) x_i \leq B \\
& x_1, x_2, x_3, x_4, x_5 \in \{0,1\}
\end{array}
$$

- Some example implementations:
  - knapsack_basic.mos, knapsack_ellipsoid.mos, knapsack_scenario.mos

FICO

To summarise

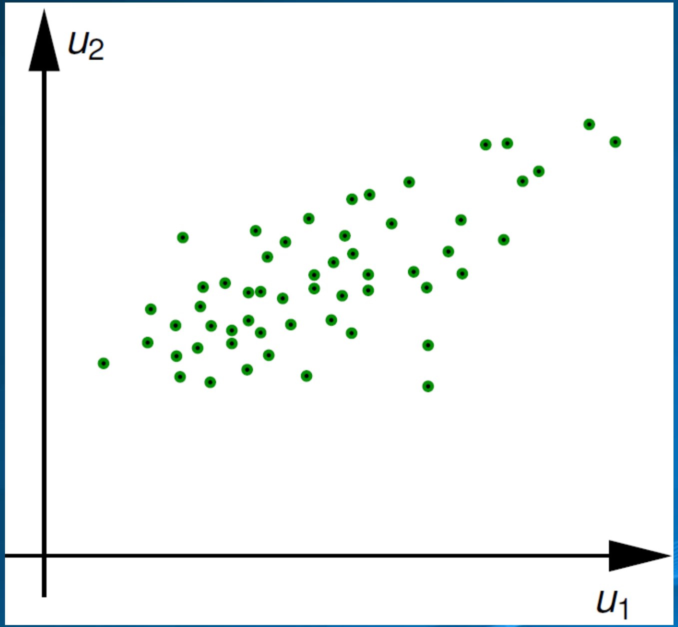
- There are several classes of uncertainty set
  - **polyhedral**: a system of linear equations/inequalities on $\boldsymbol{u}$
  - **ellipsoidal**: a quadratic constraint $\boldsymbol{u}^T Q \boldsymbol{u} \leq \alpha$
  - **scenarios**: a list of historical values of $\boldsymbol{u}$
- To solve the problem
  1. construct the **robust counterpart**, a robust version of the problem
  2. solve the robust counterpart
  3. return the optimal solution of the robust counterpart as the solution of the original problem.

How does robustness change problem difficulty

- If a polyhedral or scenario uncertainty set are added, the problem remains of the same class
  - LP $\rightarrow$ LP
  - MILP $\rightarrow$ MILP
  - MIQCQP $\rightarrow$ MIQCQP
- A quadratic uncertainty set introduces a *second order cone*:
  - LP $\rightarrow$ Second Order Conic Programming (SOCP)
  - MILP $\rightarrow$ MISOCP
  - MIQCQP $\rightarrow$ MIQCQP + MISOCP

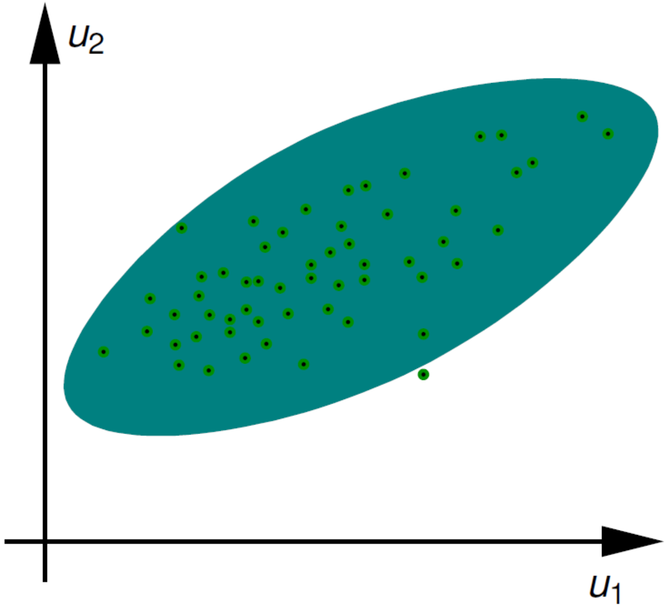
In practice …

► Suppose we have some historical data, but not enough



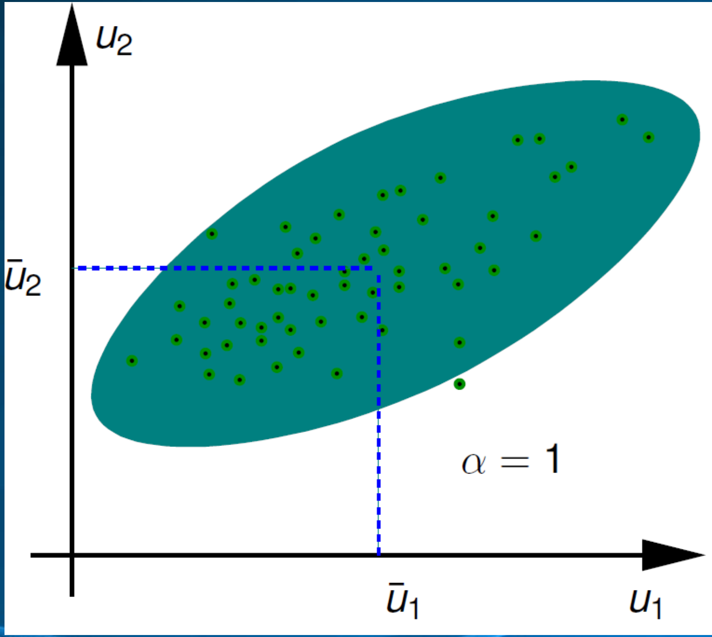
In practice …

- Analytics can be used to yield a pattern (mean/covariance).
- Exploit it!



In practice …

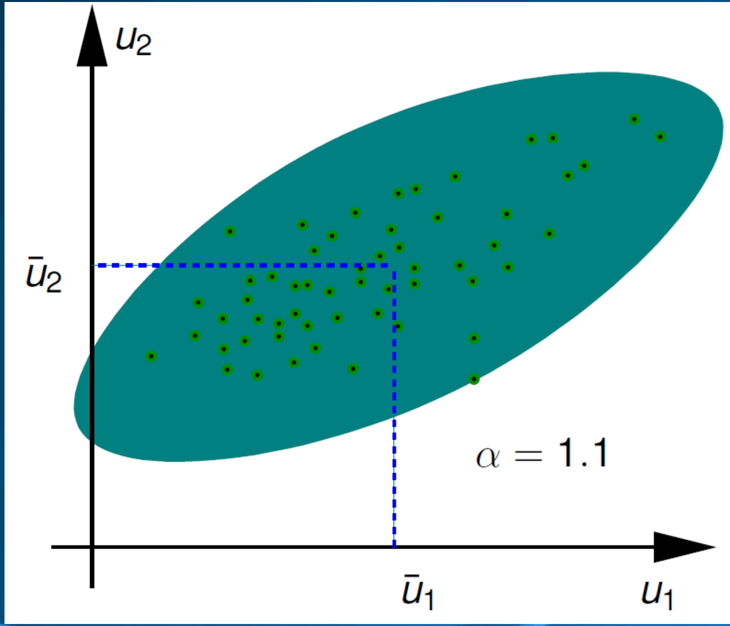
- Quadratic uncertainty: $\boldsymbol{u} = \bar{\boldsymbol{u}} + \tilde{\boldsymbol{u}}$ with $\tilde{\boldsymbol{u}}^T \boldsymbol{Q} \tilde{\boldsymbol{u}} \leq \boldsymbol{\alpha}$



FICO

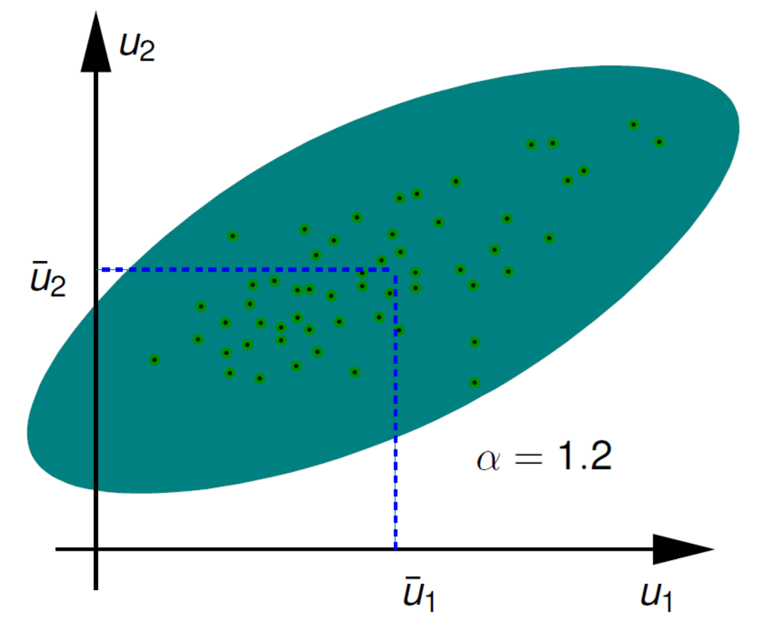
In practice …

▶ Our level of uncertainty (conservativeness) is given by α



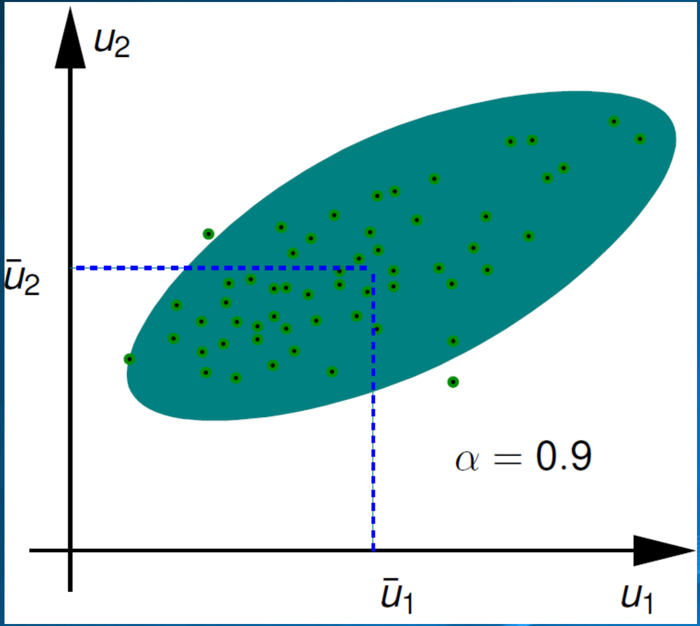
In practice …

- Our level of uncertainty (conservativeness) is given by α



In practice …

- Our level of uncertainty (conservativeness) is given by α



FICO

Reference material

- White paper Robust Optimization with Xpress
  - Explains the underlying concepts and documents the Robust Optimisation examples distributed in the Xpress release

- Ben-Tal, A., El Ghaoui, L., Nemirovski, A. (2009). *Robust optimization*. Princeton University Press.
- Bertsimas, D., Sim, M. (2004). The price of robustness. *Operations Research*, **52**(1), 35-53

Thank You

Dr Andy Harrison
+44(0)7808 777 340
AndyHarrison@fico.com