Agenda today

- ❑ Introduction to gurobi package in Python
- ❑ Demonstration of gurobi through Example 2: Distribution system design
- ❑ Assignment 2: Prescriptive model for optimizing production

Aalto University
School of Business

Part 1: Introduction to gurobi

Data Science for Business II

Pekka Malo & Eeva Vilkkumaa

Gurobi algorithms

- Continuous optimization
  - LP: primal and dual simplex, sifting, parallel barrier, concurrent
  - QP: primal and dual simplex, parallel barrier
  - QCP: parallel barrier

- Discrete optimization (MIP, MIQP, MIQCP)
  - Parallel branch-and-bound
  - Relaxations solved with continuous algorithms
  - Cutting planes and heuristics

- General
  - Automatic presolve
  - Identifying irreducible inconsistent subsystems (IIS)
  - Automated parameter tuning tool
  - …

Interfacing with Gurobi

- Command-line tool
- Full-featured interactive shell
- Matrix-oriented APIs
  - C, MATLAB, R
- Object-oriented APIs
  - C++, Java, .NET, Python
- Modeling systems
  - Commercial: AMPL, Frontline Solvers, …
  - Free: JuliaOpt, PuLP, Pyomo, SolverStudio, …

Connecting Python and Gurobi

- Basic Python interpreter: gurobi.bat / gurobi.sh
  - Included and configured when you install Gurobi Optimizer

- Anaconda Python – automated installation

```
conda config --add channels   http://conda.anaconda.org/gurobi
conda install gurobi
```

- Manual installation for other Python distributions
  - Change to GUROBI_HOME directory
  - Run `python setup.py install`

- For details, see the Gurobi Quick Start Guide: `http://www.gurobi.com/documentation/`

Solving a simple MIP model

```
This example formulates and solves the following simple MIP model:
 maximize
       x +   y + 2 z
 subject to
       x + 2 y + 3 z <= 4
       x +   y       >= 1
 x, y, z binary
```

.lp file

Python API script

```
# Create a new model
m = Model("mip1")

# Create variables
x = m.addVar(vtype=GRB.BINARY, name="x")
y = m.addVar(vtype=GRB.BINARY, name="y")
z = m.addVar(vtype=GRB.BINARY, name="z")

# Set objective
m.setObjective(x + y + 2 * z, GRB.MAXIMIZE)

# Add constraint: x + 2 y + 3 z <= 4
m.addConstr(x + 2 * y + 3 * z <= 4, "c0")

# Add constraint: x + y >= 1
m.addConstr(x + y >= 1, "c1")

m.optimize()
```

More information on gurobi

- Familiarize yourself with the material provided on the MyCourses page for week 2
  - User's guide documents
  - Python-codes for examples 1, 2, and 3

- Python API documentation (i.e. user manual for Python) can be found at: http://www.gurobi.com/documentation/8.1/refman/index.html

Aalto University
School of Business

Part 2: Demonstration of gurobi

Distribution system design

Example 2: Distribution system design

The Martin-Beck Company operates a plant in St. Louis with an annual capacity of 30,000 units. Product is shipped from the plant to regional distribution centers located in Boston, Atlanta, and Houston. Because of an anticipated increase in demand, Martin-Beck plans to increase capacity by constructing a new plant in one or more of the following cities: Dallas, Fort Worth, Denver, or Kansas City. Dallas and Fort Worth are very close to one another, whereby the company does not want to have a plant in both cities. The estimated annual fixed costs and capacities for the four proposed plants are as follows:

Proposed plant	Annual Fixed Cost	Annual Capacity
Dallas	$175,000	10,000
Fort Worth	$300,000	20,000
Denver	$375,000	30,000
Kansas City	$500,000	40,000

Source: Anderson et al. 2000, An Introduction to Management Science – Quantitative Approaches to Decision Making, South-Western College Publishing.

Example 2: Distribution system design

The company's long-range planning group has developed forecasts of the anticipated annual demand at the distribution centers as follows:

Distribution center	Annual demand
Boston	30,000
Atlanta	20,000
Houston	20,000

The shipping cost per unit from each plant to each distribution center are as follows:

	Distribution centers		
Plant site	Boston	Atlanta	Houston
Dallas	5	2	3
Fort Worth	4	3	4
Denver	9	7	5
Kansas City	10	4	2
St. Louis	8	4	3

In which city/cities should the Martin-Beck Company construct its new plant/plants?

Example 2: Distribution system design

MILP formulation:

Minimize total transportation & operational costs

$$\min_{x_t, i_t} \quad 5x_{11} + 2x_{12} + 3x_{13} + 4x_{21} + 3x_{22} + 4x_{23} + 9x_{31} + 7x_{32} + 5x_{33} + 10x_{41} + 4x_{42} + 2x_{43} + 8x_{51} + 4x_{52} + 3x_{53} + 175y_1 + 300y_2 + 375y_3 + 500y_4$$

s.t.

$$\begin{aligned}
x_{11} + x_{12} + x_{13} - 10y_1 &\le 0 \\
x_{21} + x_{22} + x_{23} - 20y_2 &\le 0 \\
x_{31} + x_{32} + x_{33} - 30y_3 &\le 0 \\
x_{41} + x_{42} + x_{43} - 40y_4 &\le 0 \\
x_{51} + x_{52} + x_{53} &\le 30
\end{aligned}$$

Plant capacity constraints

$$\begin{aligned}
x_{11} + x_{21} + x_{31} + x_{41} + x_{51} &= 30 \\
x_{12} + x_{22} + x_{32} + x_{42} + x_{52} &= 20 \\
x_{13} + x_{23} + x_{33} + x_{43} + x_{53} &= 20
\end{aligned}$$

Demand constraints

$$y_1 + y_2 \le 1$$

No plant in both Da & FW

$$x_{ij} \ge 0, \quad i = 1, \dots, 5, \quad j = 1, \dots, 3$$

$$y_i \in \{0,1\}, \quad i = 1, \dots, 5$$

Non-negativity & binary constraints

Python implementation

- Load the following files from the course website:
  - Example-2.py
  - Ex2-plant_options.txt
  - Ex2-distribution_centers.txt
  - Ex2-shipping_costs.txt
- Open file Example-2.py in PyCharm / Jupyter, familiarize yourself with the code, and run it
- Assume that the Martin-Beck Company wants to build one of the plants in either Dallas or Fort Worth (but not both). Amend the model and rerun it. How do the results change?

Results

Solution to the original problem:

- Optimal plant locations: $(y_1, y_2, y_3, y_4) = (0,0,0,1) \rightarrow x_{ij} = 0$ for all i=1,2,3

- Optimal shipping amounts: $(x_{41}, x_{42}, x_{43}) = (0, 20, 20)$ and $(x_{51}, x_{52}, x_{53}) = (30, 0, 0)$

- Optimal cost: $860,000

Solution to the modified problem:

- Optimal plant locations: $(y_1, y_2, y_3, y_4) = (1,0,1,0) \rightarrow x_{ij} = 0$ for all i=2,4

- Optimal shipping amounts: $(x_{11}, x_{12}, x_{13}) = (0, 10, 0)$, $(x_{31}, x_{32}, x_{33}) = (30, 0, 0)$, and $(x_{51}, x_{52}, x_{53}) = (0, 10, 20)$

- Optimal cost: $940,000

Aalto University
School of Business

Part 3: Assignment 2

Prescriptive model for optimizing production

Problem description in brief

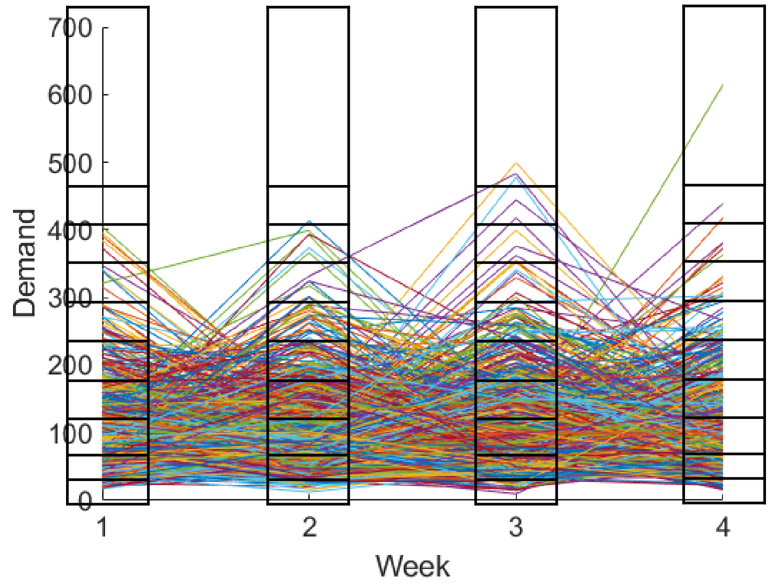
- In the assignment for week 1, you
  - Developed a time series model to predict product demand for the next four weeks
  - Simulated trajectories of this demand

- In the assignment for week 2, you will utilize the predictive model from week 1 to optimize
  - The marketing campaign scheme and
  - Production plan for the next four weeks

From simulated trajectories to scenario trees

- To optimize the production plan, the evolution of the uncertain demand is modeled as a scenario tree (cf. Example 3: Stochastic optimization)

- 50,000 simulated demand trajectories are converted into a scenario tree by
  - Discretizing the range of demand values
  - Approximating scenario probabilities by the shares of trajectories whose values are within the discretization ”bins”



Small sample of the scenario tree data

	level0me	level	sequence	freq	p	sales
1	start	1	0	50000	1	0
2	<U+00A6>--	2	1	9341	0.18682	25
3	<U+00A6>	3	1, 1	2007	0.214859	25
4	<U+00A6>	4	1, 1, 1	422	0.210264	25
5	<U+00A6>	5	1, 1, 1, 1	85	0.201422	25
6	<U+00A6>	5	1, 1, 1, 2	193	0.457346	75
7	<U+00A6>	5	1, 1, 1, 3	120	0.28436	150
8	<U+00A6>	5	1, 1, 1, 4	19	0.045024	250
9	<U+00A6>	5	1, 1, 1, 5	3	0.007109	350
10	<U+00A6>	5	1, 1, 1, 6	1	0.00237	450
11	<U+00A6>	5	1, 1, 1, 7	1	0.00237	550
12	<U+00A6>	5	1, 1, 1, 8	0	0	650
13	<U+00A6>	5	1, 1, 1, 9	0	0	750
14	<U+00A6>	5	1, 1, 1, 10	0	0	2900
15	<U+00A6>	4	1, 1, 2	929	0.46288	75
16	<U+00A6>	5	1, 1, 2, 1	156	0.167922	25
17	<U+00A6>	5	1, 1, 2, 2	405	0.435953	75
18	<U+00A6>	5	1, 1, 2, 3	324	0.348762	150
19	<U+00A6>	5	1, 1, 2, 4	36	0.038751	250
20	<U+00A6>	5	1, 1, 2, 5	6	0.006459	350
21	<U+00A6>	5	1, 1, 2, 6	2	0.002153	450
22	<U+00A6>	5	1, 1, 2, 7	0	0	550
23	<U+00A6>	5	1, 1, 2, 8	0	0	650
24	<U+00A6>	5	1, 1, 2, 9	0	0	750
25	<U+00A6>	5	1, 1, 2, 10	0	0	2900

Aalto University
School of Business

Task description in brief

- Download the following files from the course website and familiarize yourself with them
  - DSFB2_A2_2018.pdf: A detailed description of the problem and tasks. Contains the formulation of the optimization problem.
  - Campaign.Xscenario-tree.txt, X=1,...,4: Contain the data for 4 campaign-specific scenario tree models for the uncertain demand.
  - OptimalProduction.py: Contains an incomplete Python-code for optimizing the production plan.

- Your task is to
  - Complete the Python-code by building the LP model using gurobi,
  - Use the code to find the optimal marketing campaign scheme and production plan,
  - Modify the code to study the impact of changes in model parameters on your results.

Aalto University
School of Business

Questions or comments?