THE OPTIMAL WEDDING (WITH PYOMO)

Writing and solving constrained optimization problems in Python

Tarik Berrada, TRAYPORT

TRAYPORT
A TMX COMPANY

EUROPYTHON 2021 Jul 26 - Aug 1 Online

THE OPTIMAL WEDDING (WITH PYOMO)

Git-repository:

https://gitlab.com/Rikerl/optimalwedding

Notebook:

OptimalWedding.ypnb

A SEATING ARRANGEMENT PROBLEM

- 16 guests
- 4 tables with 4 seats each
- Each guest is characterized by:
  - Corona index `ci`
  - Gender (binary, for simplicity)

“How can we arrange the guests so that people with relatively close opinions with regard to Corona sit at the same table ?”

Full Name	Corona Index	Gender
Louis Benefici	-0.49	1
Marion Brandon	4.33	1
Saffron Cade	4.03	0
Sandrine Flippet	4.91	1
Corin Gillian	4.85	1
Barbara Jewell	-3.00	1
Julius Krazinski	-4.03	0
Samantha Krazinski	-3.88	1
Bria Lakhanpal	2.95	1
Lucius Motti	-4.98	0
Hilbert Nguyen	-1.58	0
Sandra Nguyen	-1.24	1
Lovel Praji	0.23	0
Claren Serence	-2.36	0
Guido van Rossum	0.00	0
Lorena Vindi	-4.86	0

Table Name	Number of Seats
Avocado	4
Banana	4
Coconut	4
Dates	4

CORONA DISTANCE

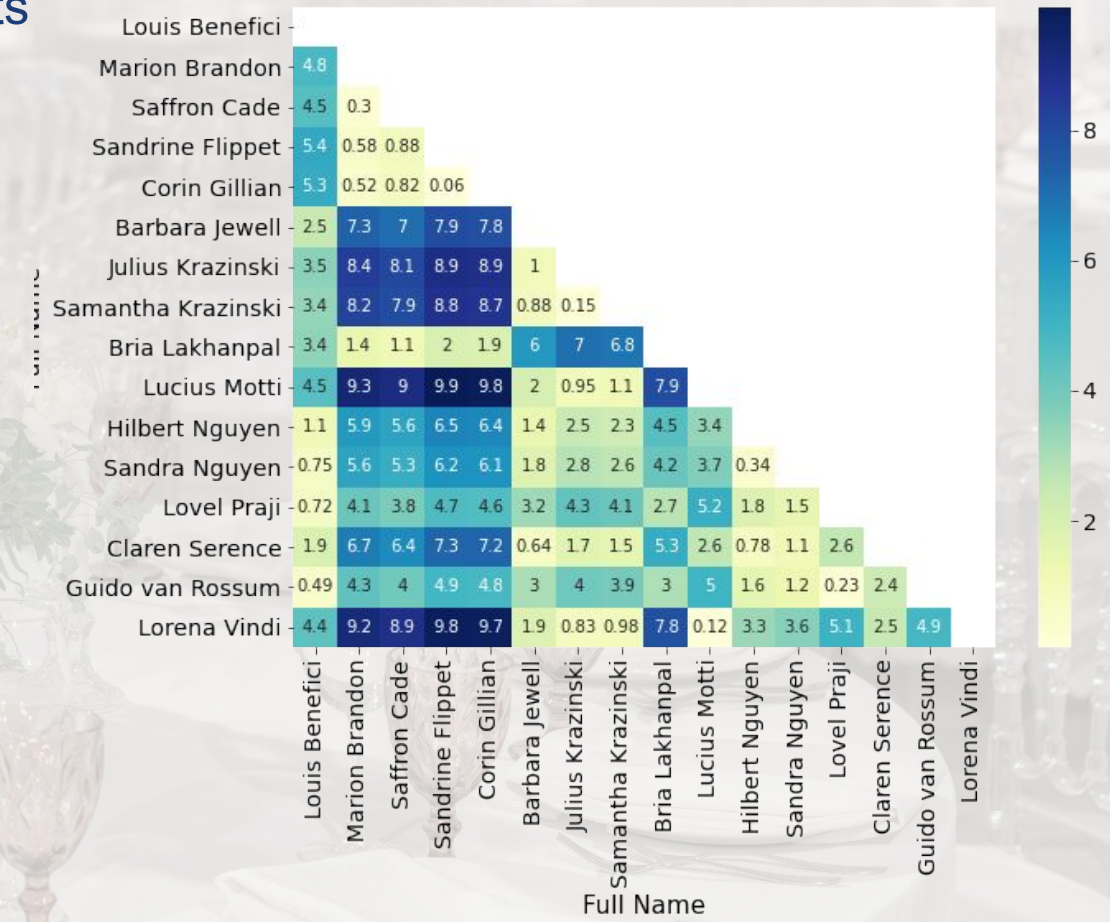
- Compute "**corona distance**" for each pair of guests (g1,g2):

```
d(g1, g2) = |ci[g1] - ci[g2]|
```

→**distance matrix**

"How can we arrange the guests to minimize the total corona distance on each table?"

```
D(table) = sum(0.5 * d(g1,g2)
           for g1 at table
           for g2 at table)
```



A SIMPLE SOLUTION

- Simple idea: sort guest by corona index and fill tables accordingly
- But what if we add **new constraints** ?
  - 1) At least one female/one male guest per table
  - 2) Julius Krazinski and Samantha Krazinski may **NOT** sit at the same table
- Simple sorting doesn't work any more
- **Brute force iteration** through all possible seating arrangements?
  - ~ 21,000 billion ways of seating 16 people on 16 chairs
  - **~ 63 million distinct table arrangements**
  - For each arrangement,
    - Check if valid
    - If valid + corona distance is below smallest one so far: **save solution as best one so far**

Full Name	Corona Index	Gender	table
Lucius Motti	-4.98	0	Avocado
Lorena Vindi	-4.86	0	Avocado
Julius Krazinski	-4.03	0	Avocado
Samantha Krazinski	-3.88	1	Avocado
Barbara Jewell	-3.00	1	Banana
Claren Serence	-2.36	0	Banana
Hilbert Nguyen	-1.58	0	Banana
Sandra Nguyen	-1.24	1	Banana
Louis Benefici	-0.49	1	Coconut
Guido van Rossum	0.00	0	Coconut
Lovel Praji	0.23	0	Coconut
Bria Lakhanpal	2.95	1	Coconut
Saffron Cade	4.03	0	Dates
Marion Brandon	4.33	1	Dates
Corin Gillian	4.85	1	Dates
Sandrine Flippet	4.91	1	Dates

Brute-force estimate: > 1500 seconds

Would work, but quite time-consuming

SEATING ARRANGEMENT AS AN OPTIMIZATION MODEL UNDER CONSTRAINTS

- **Minimize** total corona distance on each table ← **Objective function**

- **subject to**: } **Constraints**
  - Each guest sits at one (and only one) table
  - There are no more guests than seats on each table
  - There is at least one male guest at each table
  - There is at least one female guest at each table
  - Julius and Samantha Krazinski do not sit at the same table

An **optimization model** consists of:

- **optimization variables**: “decision” variables (to be optimized)
- **parameters**: given
- **constraints**: equations between variables and parameters
- an **objective function**: to be maximized or minimized
- (**sets**: to index the other model components on)

→express in “standardized” form and pass to **optimization solver**

WRITING DOWN THE MODEL IN PYTHON

- **Pyomo:** *"Python-based open-source software package that supports a diverse set of optimization capabilities for* ***formulating, solving, and analyzing optimization models****."*
  http://www.pyomo.org/
  - Turn equations into code
  - Send problem to solver
  - Collect results

- **COIN-OR CBC solver**: **Branch-and-Cut** solver from the COmputational INfrastructure for Operations Research program (https://www.coin-or.org/)

AN EXEMPLARY PYOMO MODEL (simplified syntax)

```
model = Model()

# add set:
model.tables = Set([“Avocado”, “Banana”, “Coconut”, ...])

# add (indexed) parameter:
model.table_capacity = Param(model.tables, {“Avocado”: 4, ...})

# add (indexed) optimization variable:
model.guest_seats_at = Var(model.guests * model.tables, domain=Binary)

# add (indexed) constraint:
model.one_table_per_guest = Constraint(model.guests,
rule=sum(model.guest_seats_at[guest, table] for table in model.tables) == 1)

# add objective function:
model.corona_distance = Objective(rule=total_corona_distance, sense=minimize)

# solve model:
solver.solve(model)

# retrieve optimal values:
model.guest_seats_at[“Julius Krazinski”, “Coconut”]
>>> 1.0
model.guest_seats_at[“Samantha Krazinski”, “Coconut”]
>>> 0.0  # Great relief!
```

DEMO: JUPYTER NOTEBOOK

- https://gitlab.com/RikerI/optimalwedding
- Jupyter notebook: OptimalWedding.ypnb

HAPPY END

- Optimal seating arrangement found after **~10 seconds** on my laptop
  - brute-force iteration : ~ 1500 seconds
  - In practice: interrupt solving after `<time_up>` and keep best solution so far

- **Pyomo**: leveraging the power of Python as a programming language
  - Load data from Python
  - Return result into Python
  - Alternatives: **PuLP**, scipy...

- Many real-life industrial situations can be expressed as optimization models
  - Logistics
  - Economics/finance
  - **Energy systems**
  - ...

Full Name	Corona Index	Gender	Table
Barbara Jewell	-3.00	1	Avocado
Julius Krazinski	-4.03	0	Avocado
Lucius Motti	-4.98	0	Avocado
Lorena Vindi	-4.86	0	Avocado
Marion Brandon	4.33	1	Banana
Saffron Cade	4.03	0	Banana
Sandrine Flippet	4.91	1	Banana
Corin Gillian	4.85	1	Banana
Samantha Krazinski	-3.88	1	Coconut
Hilbert Nguyen	-1.58	0	Coconut
Sandra Nguyen	-1.24	1	Coconut
Claren Serence	-2.36	0	Coconut
Louis Benefici	-0.49	1	Dates
Bria Lakhanpal	2.95	1	Dates
Lovel Praji	0.23	0	Dates
Guido van Rossum	0.00	0	Dates

… meet us at our VIRTUAL BOOTH!

THANK YOU! QUESTIONS?

TRAYPORT
A TMX COMPANY

info@trayport.com www.trayport.com

LONDON 7th Floor, 9 Appold Street, London EC2A 2AP, United Kingdom +44 (0)20 7960 5500
SINGAPORE One Raffles Place, Office Tower 1, #31-02, Singapore, 048616 +65 6411 4700
VIENNA Lemböckgasse 49/1A/5.OG, 1230 Vienna, Austria +43 1 609 2290