40.016: The Analytics Edge
Week 12 Lecture 2 (Optional)

PRESCRIPTIVE ANALYTICS WITH JULIA (PART 2)

Term 5, 2022

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Outline

1. Capstone Allocation at SUTD
2. Implementation in Julia, with IJulia, Jupyter, and JuMP

Outline

Capstone Allocation

Key features:

- Course started in 2015 and offered once a year
- Requires students from at least two disciplines (pillars)
- Projects sourced primarily from companies operating in Singapore
- Capstone office works with faculty and companies to scope the projects

Capstone Allocation (cont'd)

Challenges in allocating students to projects:

- **Efficiency** → Eliciting student preferences for projects and using them for allocation is critical for efficiency, rather than just allocating projects randomly
- **Fairness** → Students have equal chances in obtaining their preferred capstone
- **Multidisciplinary projects** → Every project must be multidisciplinary and involve students from at least two pillars
- **Flexibility** → It must be easy to incorporate any additional new constraints that might arise during the allocation process

Network representation

Representation of a Capstone project allocation instance with 6 students, 3 projects and 3 disciplines.

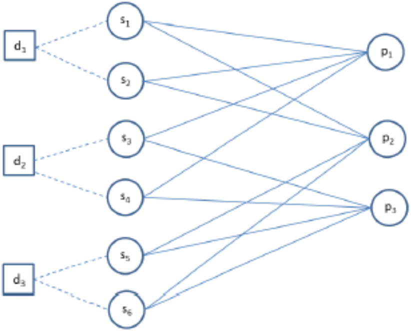

Figure: Source: Magnanti and Natarajan (2018).

Model inputs

- Network representation
- Lower and upper bounds on the number of students of each discipline (needed for each project)
- Student preferences for projects

  $\rightarrow$ The student preferences are converted to utility values on the arcs in the network where the value is set to K (e.g., $= 10$) for each student's topmost preferred project, $K - 1$ for their second most preferred project, down to 1 for the project ranked the lowest in their list

Capstone Allocation

Decision variables: Two sets of binary variables corresponding to

- Which project is assigned to a student
- Which project is launched

Objective function: Total utility, or efficiency, given by the sum of the utilities of the projects assigned to students (maximized)

Constraints:

- Each student must be allocated to a single project
- A student is assigned to a project only if a project is launched
- The number of students of the different disciplines in a project that is allocated lies between the prescribed lower and upper bounds

Formulation

- UB$(p,d) =$ Upper bound on the number of students of discipline d needed for project p
- LB$(p,d) =$ Lower bound on the number of students of discipline d needed for project p
- util$(s,p) =$ Utility of project p for student s in the network

As mentioned above, we have two sets of decision variables:

$$x_{sp} = \begin{cases} 1 & \text{if student } s \text{ is allocated to project } p \\ 0 & \text{otherwise} \end{cases}$$

$$y_p = \begin{cases} 1 & \text{if project } p \text{ is offered} \\ 0 & \text{otherwise} \end{cases}$$

Formulation (cont'd)

- $\mathcal{S}$: set of student nodes
- $\mathcal{P}$: set of project nodes
- $\mathcal{D}$: set of possible types (disciplines) of the students
- $\rightarrow G(\mathcal{S} \cup \mathcal{P}, \mathcal{E})$: bipartite graph
- $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{P}$: set of undirected edges of the graph

Formulation (cont'd)

$$
\begin{aligned}
\max \quad & \sum_{(s,p)\in\mathcal{E}} \mathsf{util}(s,p) x_{sp} & \\
\text{s.t.} \quad & \sum_{p\in\mathcal{P}:(s,p)\in\mathcal{E}} x_{sp} = 1 & \forall s \in \mathcal{S} \\
& x_{sp} \le y_p & \forall (s,p) \in \mathcal{E} \subseteq \mathcal{S} \times \mathcal{P} \\
& \sum_{s\in\mathcal{S}:(s,p)\in\mathcal{E},d(s)=d} x_{sp} \ge \mathsf{LB}(p,d) y_p & \forall d \in \mathcal{D}(p), \forall p \in \mathcal{P} \\
& \sum_{s\in\mathcal{S}:(s,p)\in\mathcal{E},d(s)=d} x_{sp} \le \mathsf{UB}(p,d) y_p & \forall d \in \mathcal{D}(p), \forall p \in \mathcal{P} \\
& x_{sp} \in \{0,1\} & \forall (s,p) \in \mathcal{E} \subseteq \mathcal{S} \times \mathcal{P} \\
& y_p \in \{0,1\} & \forall p \in \mathcal{P}
\end{aligned}
$$

Data

- `UpperBound.csv` [4,61]
- `LowerBound.csv` [4,61]
- `Rank.csv` (utility) [170,61]
- `Pillar.csv` (pillar of each student) [170,4]

So, we have 170 students, 61 potential projects, and 4 Pillars

Outline

1. Capstone Allocation at SUTD
2. Implementation in Julia, with IJulia, Jupyter, and JuMP

IJulia and Jupyter

How to launch Jupyter for Julia?

Option 1: Type `jupyter notebook` in the terminal

Option 2: Type the following commands in Julia (`julia>` prompt)

```
using IJulia
notebook()
```

Workflow for creating a model

- Load JuMP and a solver / optimizer (e.g., GLPK)

```
import Pkg
Pkg.add("JuMP")
Pkg.add("GLPK")
```

  see file `Packages.ipynb`.
- Create a model with the `Model()` function
- Add variables (`@variable()`)
- Add objective (`@objective()`)
- Add constraints, if any (`@constraint()`)

Workflow for solving a problem

- Use the `optimize!()` function
- Check the status of the solution (`termination_status()`)
- Check the obtained objective function (`objective_value()`)
- Check the obtained solution (`value.()`)

Implementation in Julia

Please refer to the file `capstone.ipynb`

References: Julia (with links)

- Julia https://julialang.org/
- IJ https://github.com/JuliaLang/IJulia.jl
- IJulia https://julialang.github.io/IJulia.jl/stable/manual/installation/
- JuMP https://jump.dev/JuMP.jl/stable/

References: Capstone Allocation

- Magnanti, T.L., Natarajan, K. (2018). Allocating students to multidisciplinary capstone projects using discrete optimization. *Interfaces*, 48(3), 204-216.