40.016 The Analytics Edge

Prescriptive analytics with Julia (Part 2)

Stefano Galelli

SUTD

Term 6, 2020

Outline

- Capstone Allocation at SUTD
- Implementation in Julia, with IJulia, Jupyter, and JuMP

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

Challenges in allocating students to projects:

1. **Efficiency** → Eliciting student preferences for projects and using them for allocation is critical for efficiency, rather than just allocating projects randomly
2. **Fairness** → Students have equal chances in obtaining their preferred capstone
3. **Multidisciplinary projects** → Every project must be multidisciplinary and involve students from at least two pillars
4. **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.

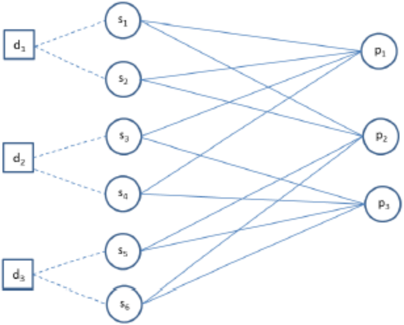

Figure 1: Source: Magnanti and Natarajan (2018).

Capstone Allocation

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)

Capstone Allocation

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

Capstone Allocation

Formulation

$\mathrm{UB}(p, d) =$ Upper bound on the number of students of type d needed for project p

$\mathrm{LB}(p, d) =$ Lower bound on the number of students of type d needed for project p

$\mathrm{util}(s, p) =$ Utility of project p for student s in the network

Capstone Allocation

Formulation

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

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

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

Capstone Allocation

Formulation

$\mathcal{S}$: set of student nodes

$\mathcal{P}$: set of project nodes

$\mathcal{D}$: set of possible types (disciplines) of the students

$\rightarrow G(\mathcal{S} \cup \mathcal{P}, \varepsilon)$: bipartite graph

$\varepsilon \subseteq \mathcal{S} \times \mathcal{P}$: set of undirected edges of the graph

Capstone Allocation

Formulation

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

Capstone Allocation

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

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()
```

JuMP

Workflow for creating a model:

1. Load JuMP and a solver / optimizer (e.g., GLPK)
2. Create a model with the `Model()` function
3. Add variables (`@variable()`)
4. Add objective (`@objective()`)
5. Add constraints, if any (`@constrain()`)

JuMP

Workflow for solving a problem:

1. Use the `optimize!()` function
2. Check the status of the solution (`termination_status()`)

Implementation in Julia

Please refer to the file *capstone.ipynb*

References

Julia (with links)

- *Julia*
- *IJ*
- *IJulia*
- *JuMP*

References

Capstone Allocation

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