Data Science Indy Meetup
10 August 2016

Workshop Preference Optimization

Solutions in Python/PuLP and XPRESS Optimization Suite

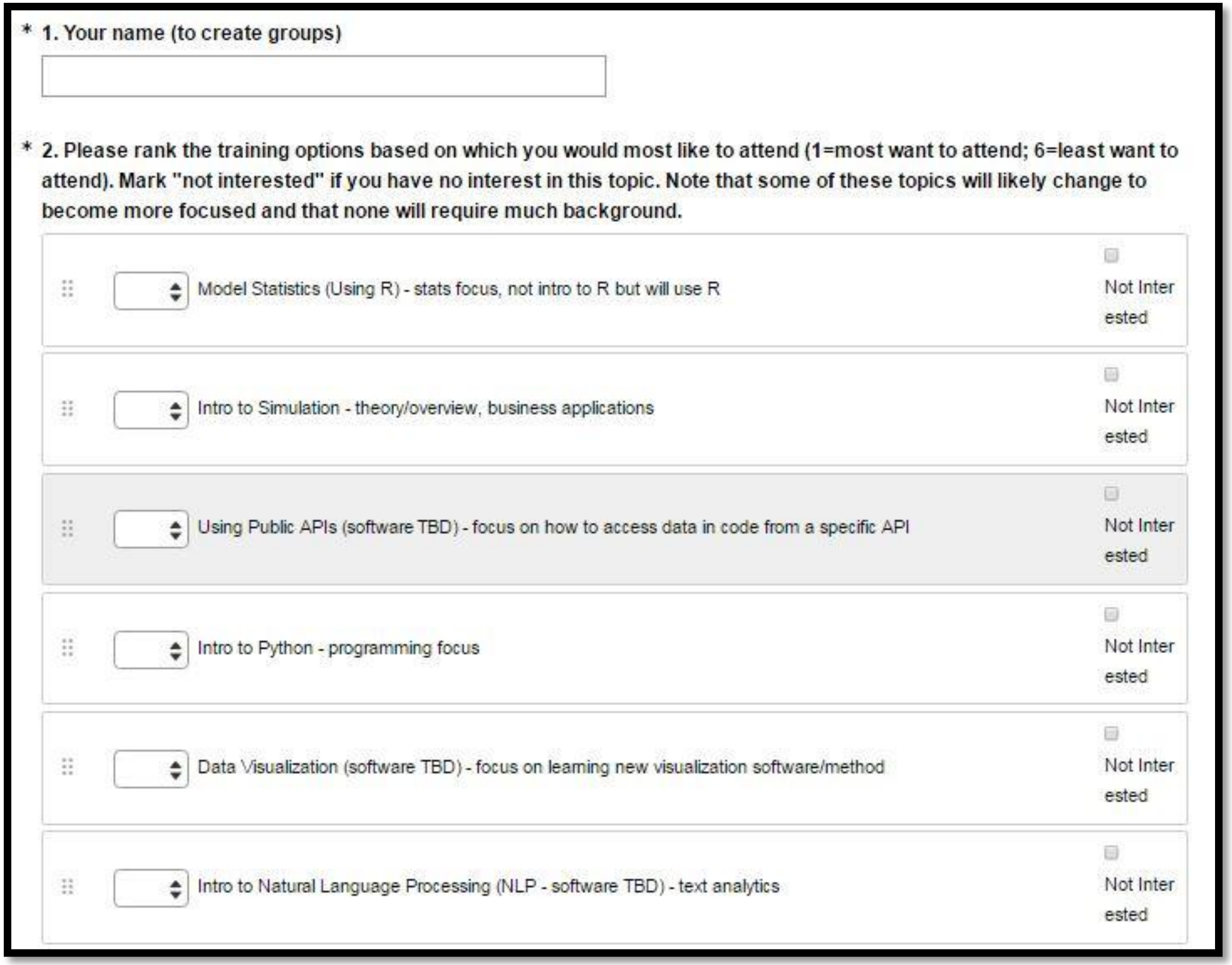
Alexandra Sullivan
Wyndham Destination Network

Problem Introduction

Image Source
https://pixabay.com/en/traffic-sign-directory-skills-can-809006/

Gathering Data

SurveyMonkey®

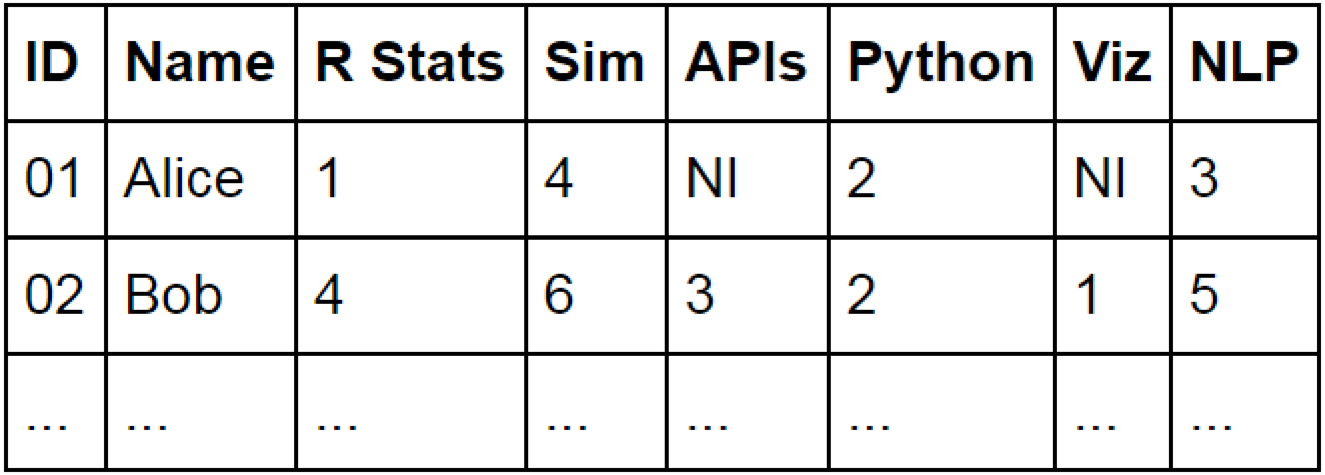


* 1. Your name (to create groups)

* 2. Please rank the training options based on which you would most like to attend (1=most want to attend; 6=least want to attend). Mark "not interested" if you have no interest in this topic. Note that some of these topics will likely change to become more focused and that none will require much background.

- Model Statistics (Using R) - stats focus, not intro to R but will use R — □ Not Interested
- Intro to Simulation - theory/overview, business applications — □ Not Interested
- Using Public APIs (software TBD) - focus on how to access data in code from a specific API — □ Not Interested
- Intro to Python - programming focus — □ Not Interested
- Data Visualization (software TBD) - focus on learning new visualization software/method — □ Not Interested
- Intro to Natural Language Processing (NLP - software TBD) - text analytics — □ Not Interested

Gathering Data



ID	Name	R Stats	Sim	APIs	Python	Viz	NLP
01	Alice	1	4	NI	2	NI	3
02	Bob	4	6	3	2	1	5
...	...	...	...	...	...	...	...

Decision Variables

$x_{student,\ workshop}$	Binary variable that indicates whether the student is enrolled in the given workshop (1 = yes, 0 = no)
$y_{workshop}$	Binary variable that indicates whether the given workshop is open or closed (1 = open, 0 = closed)

Objective

IDEAL WORLD

Everybody gets
their first choice!

NEXT BEST THING

Assign everyone to their highest choice…
given some constraints.

$$\underset{x,y}{\text{minimize}} \sum_{workshops} \sum_{students} preference \cdot x_{student,workshop}$$

Constraints – Total Workshops

We have six available workshops, but having all of these would be excessive given our 30-person department. Three workshops seemed like a good number to provide a smaller group feel without making too many people prepare material.

Translated into math:

$$Total\ workshops: \quad \sum_{workshops} y_{workshop} = 3$$

$$Only\ valid\ workshops: \quad x_{student,workshop} \leq y_{workshop} \quad \forall \quad students, workshops$$

Constraints – Min Students Per Workshop and Min Workshops Per Student

It would be silly to prepare a workshop for a group of two people, so we want a constraint to keep this from happening:

$$Min\ students\ per\ workshop: \quad \sum_{students} x_{student,workshop} \geq 7y_{workshop} \quad \forall\ \ workshops$$

And we want to make sure we assign everybody to at least one workshop, so we assign a constraint for this:

$$Workshops\ per\ student: \quad \sum_{workshops} x_{student,workshop} = 1 \quad \forall\ \ students$$

Full Formulation

Objective:

$$\underset{x,y}{\text{minimize}} \sum_{workshops} \sum_{students} preference \cdot x_{student,workshop}$$

Subject to:

$$(1)\ Binary\ variables:\quad x_{student,workshop}\ ,\ y_{workshop}\ \in\ \{0,1\}\ \forall\ students, workshops$$

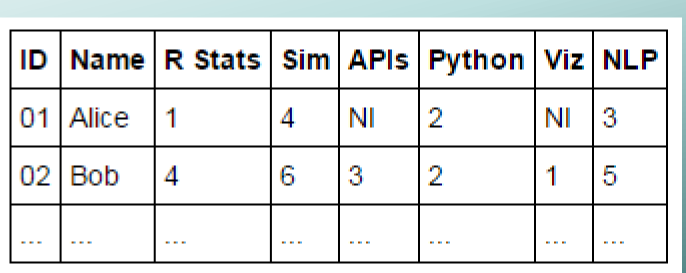
$$(2)\ Total\ workshops:\quad \sum_{workshops} y_{workshop} = 3$$

$$(3)\ Workshops\ per\ student:\quad \sum_{workshops} x_{student,workshop} = 1\ \forall\ students$$

$$(4)\ Min\ students\ per\ workshop:\quad \sum_{students} x_{student,workshop} \geq 7y_{workshop}\ \ \forall\ workshops$$

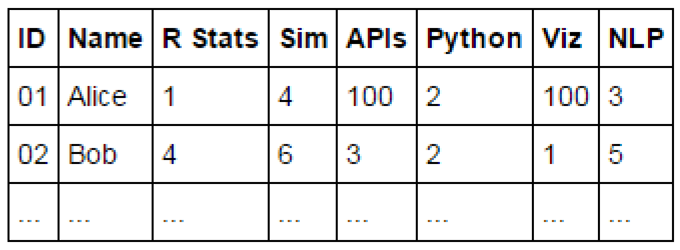
$$(5)\ Only\ valid\ workshops:\quad x_{student,workshop} \leq y_{workshop}\ \ \forall\ students, workshops$$

Data Cleaning



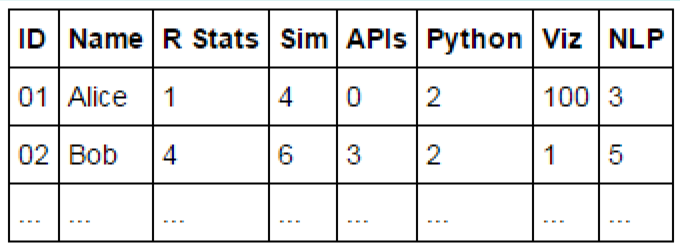
ID	Name	R Stats	Sim	APIs	Python	Viz	NLP
01	Alice	1	4	NI	2	NI	3
02	Bob	4	6	3	2	1	5
...	...	...	...	...	...	...	...

Original



ID	Name	R Stats	Sim	APIs	Python	Viz	NLP
01	Alice	1	4	100	2	100	3
02	Bob	4	6	3	2	1	5
...	...	...	...	...	...	...	...

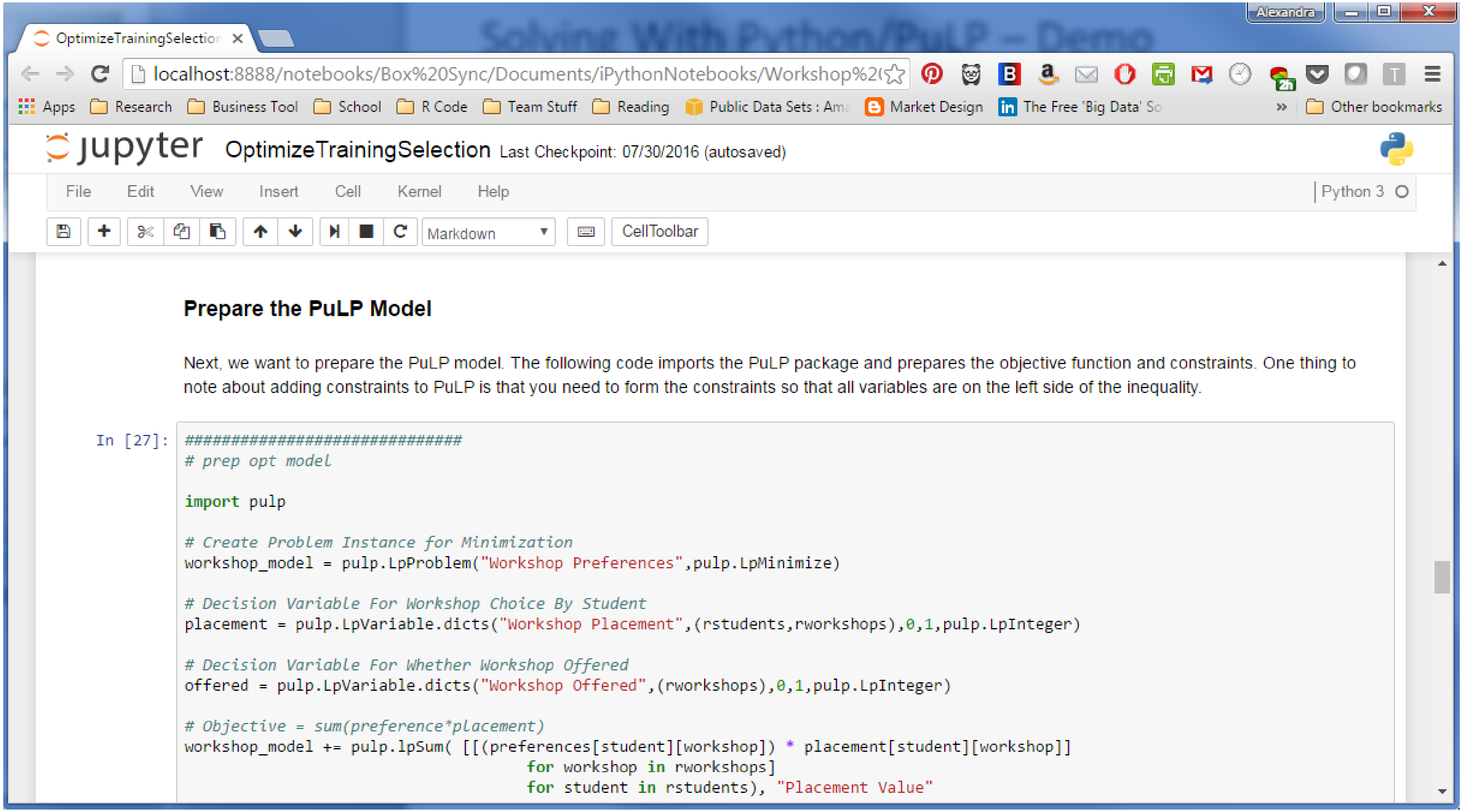
Replace “Not Interested”



ID	Name	R Stats	Sim	APIs	Python	Viz	NLP
01	Alice	1	4	0	2	100	3
02	Bob	4	6	3	2	1	5
...	...	...	...	...	...	...	...

Instructor Interest

Solving With Python/PuLP – Demo



Prepare the PuLP Model

Next, we want to prepare the PuLP model. The following code imports the PuLP package and prepares the objective function and constraints. One thing to note about adding constraints to PuLP is that you need to form the constraints so that all variables are on the left side of the inequality.

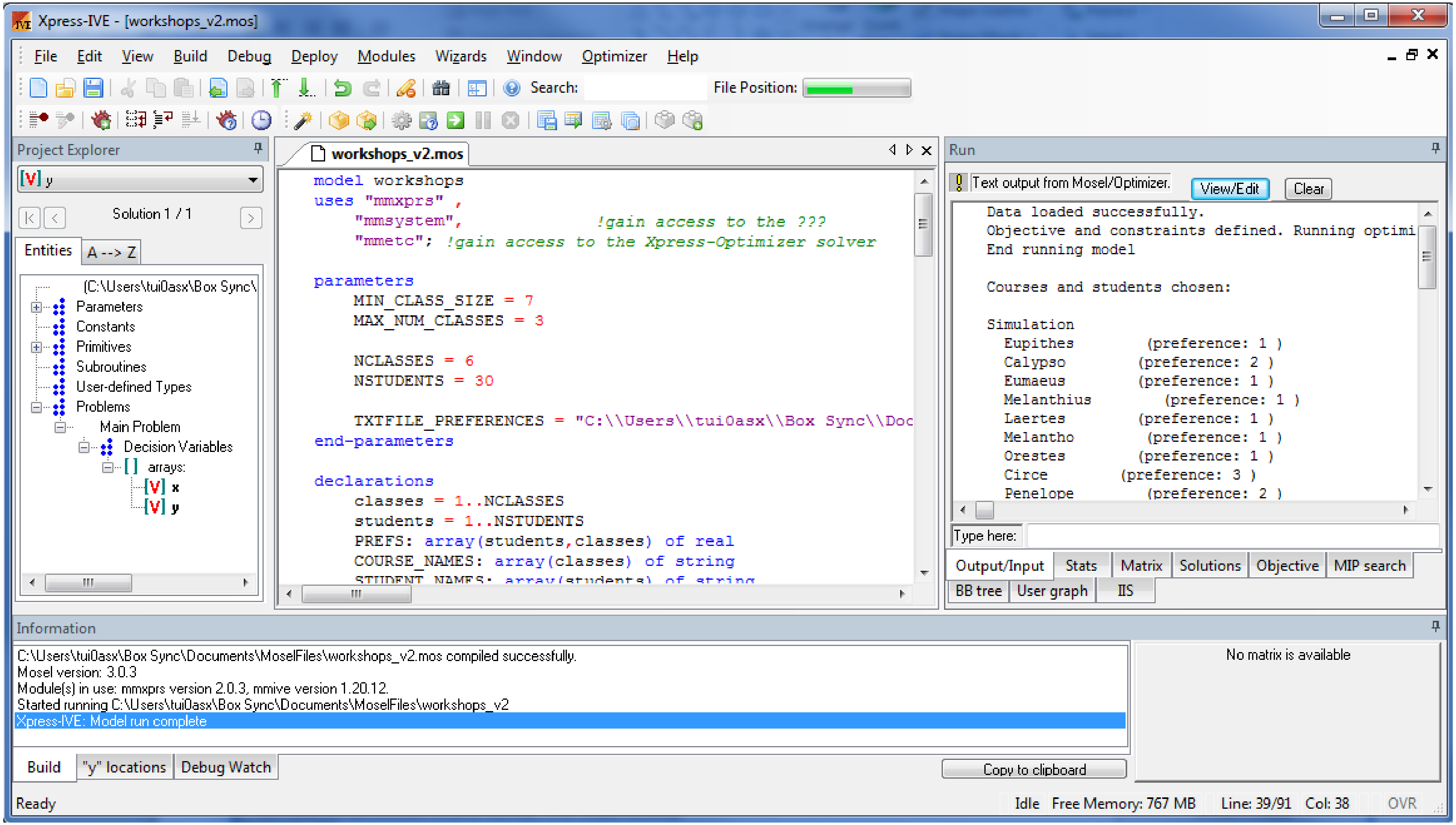
In [27]:

```
################################
# prep opt model

import pulp

# Create Problem Instance for Minimization
workshop_model = pulp.LpProblem("Workshop Preferences",pulp.LpMinimize)

# Decision Variable For Workshop Choice By Student
placement = pulp.LpVariable.dicts("Workshop Placement",(rstudents,rworkshops),0,1,pulp.LpInteger)

# Decision Variable For Whether Workshop Offered
offered = pulp.LpVariable.dicts("Workshop Offered",(rworkshops),0,1,pulp.LpInteger)

# Objective = sum(preference*placement)
workshop_model += pulp.lpSum( [[(preferences[student][workshop]) * placement[student][workshop]]
                                     for workshop in rworkshops]
                                     for student in rstudents), "Placement Value"
```

PuLP is an open-source Python package that can be used with several open-source and commercial optimization software options

Solving With XPRESS – Demo

FICO Xpress



```
model workshops
uses "mmxprs" ,
    "mmsystem",            !gain access to the ???
    "mmetc"; !gain access to the Xpress-Optimizer solver

parameters
    MIN_CLASS_SIZE = 7
    MAX_NUM_CLASSES = 3

    NCLASSES = 6
    NSTUDENTS = 30

    TXTFILE_PREFERENCES = "C:\\Users\\tui0asx\\Box Sync\\Doc
end-parameters

declarations
    classes = 1..NCLASSES
    students = 1..NSTUDENTS
    PREFS: array(students,classes) of real
    COURSE_NAMES: array(classes) of string
    STUDENT_NAMES: array(students) of string
```

```
Data loaded successfully.
Objective and constraints defined. Running optimi
End running model

Courses and students chosen:

Simulation
  Eupithes         (preference: 1 )
  Calypso         (preference: 2 )
  Eumaeus         (preference: 1 )
  Melanthius         (preference: 1 )
  Laertes         (preference: 1 )
  Melantho         (preference: 1 )
  Orestes         (preference: 1 )
  Circe         (preference: 3 )
  Penelope         (preference: 2 )
```

```
C:\Users\tui0asx\Box Sync\Documents\MoselFiles\workshops_v2.mos compiled successfully.
Mosel version: 3.0.3
Module(s) in use: mmxprs version 2.0.3, mmive version 1.20.12.
Started running C:\Users\tui0asx\Box Sync\Documents\MoselFiles\workshops_v2
Xpress-IVE: Model run complete
```

XPRESS Optimization Suite is a powerful commercial optimization software from FICO

Optimization Software Comparison

Analysis of commercial and free and open source solvers for linear optimization problems (B. Meindl and M. Templ, 2012)

Open source:

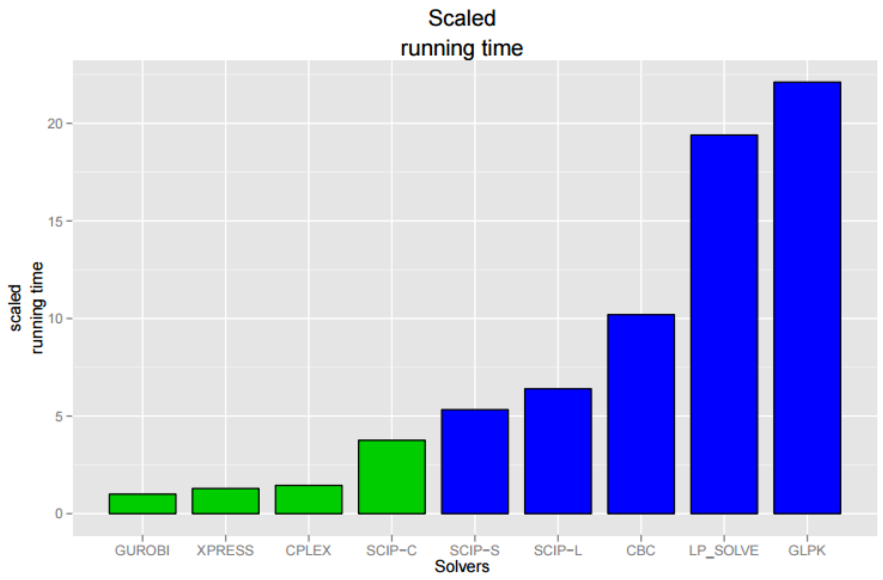
- GLPK
- LP_SOLVE
- CLP/CBC (Coin-OR, default for PuLP(?))

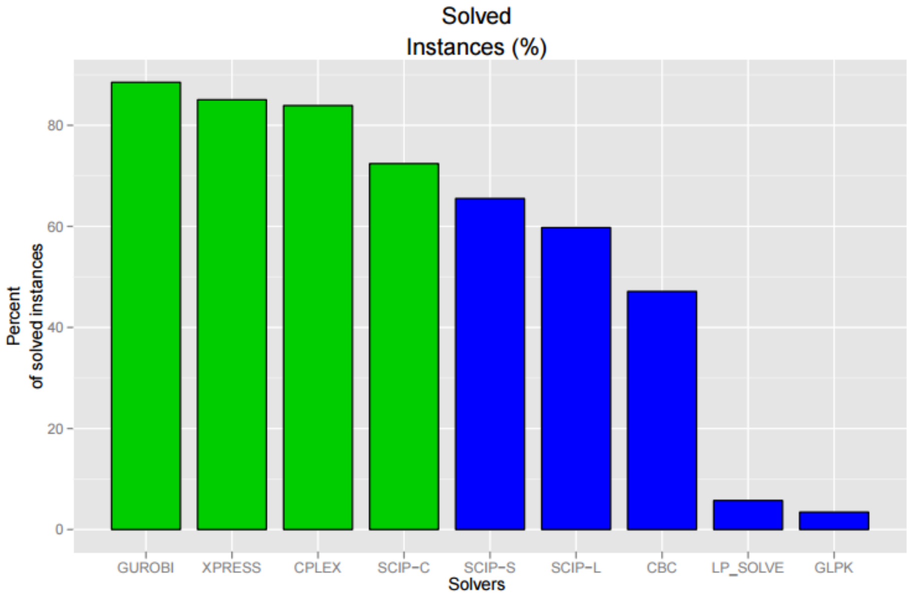
Free for non-commercial use:

- SoPlex
- SCIP
  - -C = CPLEX solver
  - -L = CLP solver
  - -S = SoPlex solver

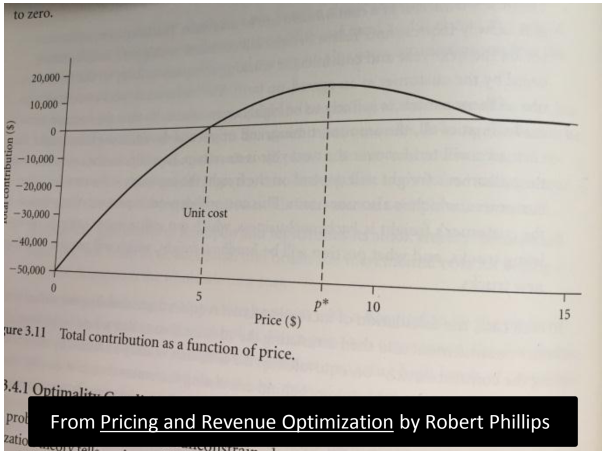
Commercial:

- CPLEX
- XPRESS Optimization Suite
- Gurobi





Real-World Use Case: Price Optimization for Hotels



ure 3.11 Total contribution as a function of price.

3.4.1 Optimality

From Pricing and Revenue Optimization by Robert Phillips

Formulation:

- Data inputs: supply, demand, and price response (estimates/forecasts)
- Business constraints: for example, size ordering or minimum prices
- Decision variable: price for each SKU
- Objective: maximize revenue!

Millions of combinations, daily changes to inputs – how do we solve this efficiently and easily?

Optimization software!

Resources

- Python Hosted’s PuLP Sudoku Example (how to use PuLP)
- Julian Hall’s Mosel (XPRESS) Documentation Summary (how to use XPRESS)
- Wikipedia List of Optimization Software (other software options)
- Pricing and Revenue Optimization by Robert Phillips (how to use optimization in pricing)
- Analysis of commercial and free and open source solvers for linear optimization problems (B. Meindl and M. Templ, 2012)
- PuzzlOR (source for OR puzzles – for your next PuLP project ☺)