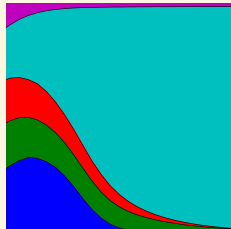


Why you are here.

@drvinceknight

@opcambell

@alexwlchan



David Maclver - Hypothesis.works - @DRMaclver

<http://bit.ly/hard-problems>

maximise: $16x + 10y$

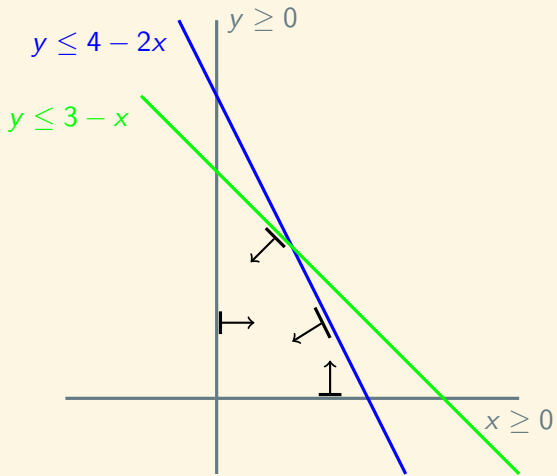
such that:

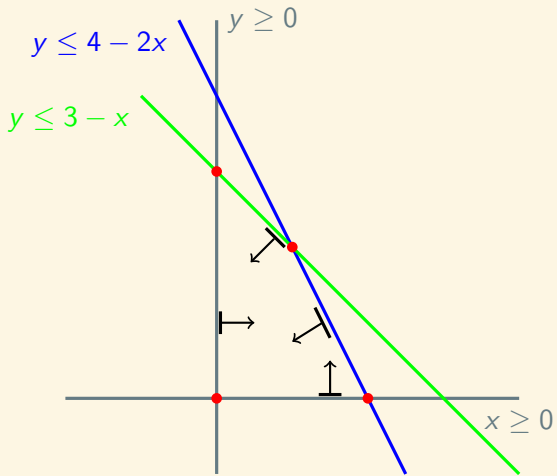
$$2x + y \leq 4$$

$$2x + 2y \leq 6$$

$$x \geq 0$$

$$y \geq 0$$







```
>>> import numpy as np
>>> import scipy as sp
>>> assert sp.__version__ == '0.19.0'
>>> from scipy import spatial
>>> halfspaces = np.array([[2, 1, -4],
...                        [2, 2, -6],
...                        [-1, 0, 0],
...                        [0, -1, 0]])
>>> feasible_point = np.array([1/2, 1/2])
>>> hs = spatial.HalfspaceIntersection(
...     halfspaces=halfspaces,
...     interior_point=feasible_point)
>>> hs.intersections
array([[ 0.,  0.],
       [ 2.,  0.],
       [ 0.,  3.],
       [ 1.,  2.]])
```


$$X = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\sum_{\text{columns}} X_{ij} = 1$$

$$\sum_{\text{rows}} X_{ij} \leq 1$$

conference-scheduler.readthedocs.io/

```
>>> from conference_scheduler.resources import Slot, Event
```

Physics

$$C_s = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$X_{ij} \leq C_{sij}$$

Physics, Ethics, Life

$$C_e = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$S_j = \{1 \leq j' \leq N \mid \text{if } j \text{ and } j' \text{ are at the same time}\}$$

$$X_{ij} + X_{i'j'} \leq 1 + C_{ei'j'} \text{ for all } j' \in S_j$$

```
>>> from datetime import datetime

>>> slots = [Slot(venue='Big',
...               starts_at=datetime(2016, 9, 15, 9, 30),
...               duration=30,
...               session="A",
...               capacity=200),
...          Slot(venue='Big',
...               starts_at=datetime(2016, 9, 15, 10, 0),
...               duration=30,
...               session="A",
...               capacity=200)]
```



```
>>> events = [Event(name='Talk 1', duration=30,
...                 unavailability=[slots[0]],
...                 demand=50),
...           Event(name='Talk 2', duration=30,
...                 demand=130)]
>>> events[0].add_unavailability(events[1])
```

maximise: $16x + 10y$

such that:

$$2x + y \leq 4$$

$$2x + 2y \leq 6$$

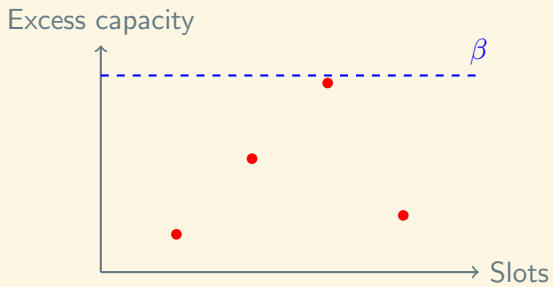
$$x \geq 0$$

$$y \geq 0$$

Excess capacity



Slots



minimize: β

such that:

$$\text{demand} - \text{capacity} \leq \beta$$

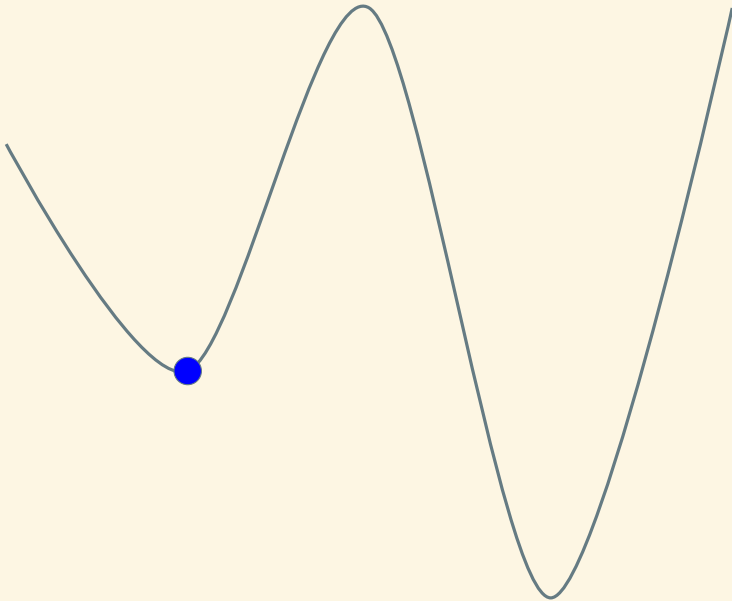


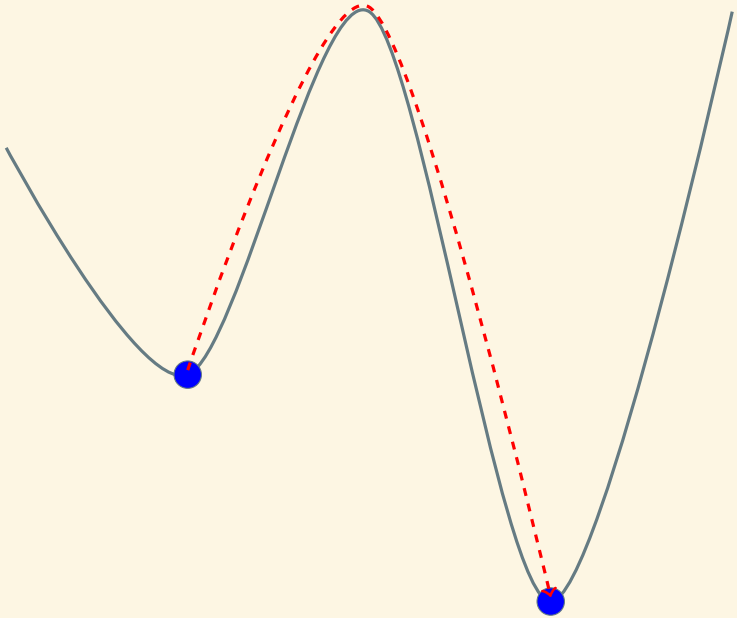
europython

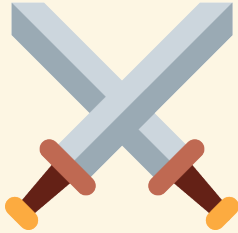
9-16 JULY 2017

Rimini





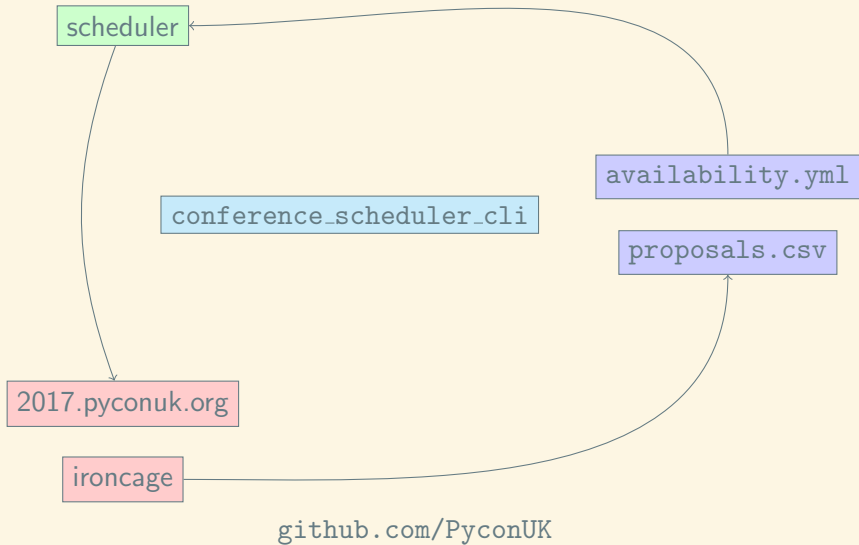




(CC-BY Twitter)

conference_scheduler/heuristics/simulated_annealing.py:

```
34 while current_energy > lower_bound and iterations <= max_iterations:
35
36     iterations += 1
37     candidate = element_from_neighbourhood(X)
38     candidate_energy = objective_function(candidate)
39
40     delta = candidate_energy - current_energy
41
42     if (candidate_energy < best_energy and
43         (acceptance_criteria is None or
44          acceptance_criteria(candidate) <= acceptance_bound)):
45
46         best_energy = candidate_energy
47         best_X = candidate
48
49
50     if delta < 0 or (temperature > 0 and
51                    np.random.random() < np.exp(-delta / temperature)):
52         X = candidate
53         current_energy = candidate_energy
54
55     temperature *= (cooldown_rate) ** iterations
```



conference-scheduler.readthedocs.io
github.com/PyconUK
@drvinceknight