Assignment and Scheduling Problems

Junaid Hasan

Math 381 Lecture 4

Assignment Problem: High Level Idea

- A typical assignment problem look like:
- A list I of m workers and a list J of m Jobs.
- Consider $x[i,j] = 1$ if worker i is assigned job j and 0 otherwise.
- Furthermore, for each (i,j) either a cost $c[i,j]$ or a preference $p[i,j]$.
- Goal is to *maximize* preference $\sum_{i \in I} j \in Jp[i,j] \cdot x[i,j]$ or *minimize* cost $\sum_{i \in I} j \in Jc[i,j] \cdot x[i,j]$.
- subject to
- Each job is assigned to one worker $\sum_{i \in I} x[i,j] = 1$ for all $j \in J$.
- Each worker doing one job $\sum_{j \in J} x[i,j] = 1$ for all $i \in I$.

Modifications

Scheduling Problem: Example

A hospital supervisor needs to create a schedule for four nurses over a three-day period, subject to the following:

- Each day is divided into three 8-hour shifts.
- Every day, each shift is assigned to a single nurse.
- No nurse works more than one shift on a given day.

Question

- How do we convert this problem in an LP?
- What do we maximize/minimize?

Preference

- Suppose nurses have a preference for each day and each shift.
- `rating[(n,d,s)] = 0` if nurse n cannot work day d, shift s.
- `rating[(n,d,s)] = 1` if nurse n can work day d, shift s, but not preferred.
- `rating[(n,d,s)] = 2` if nurse n prefers to work day d, shift s.
- Then our goal is to maximize rating.
- Alternatively there may be a cost per nurse for a day, shift (transport, training, skills etc) and the hospital may want to minimize that.

Data

```
nurses = [1,2,3,4]
shifts = [1,2,3]
days = [1,2,3]
rating = {(1,1,1):0, (1,1,2):1, (1,1,3): 2, ...
}
```

- Refer to Python Notebook for the full code. ## Model
- We create a model `model = Model("Nurse Scheduling")`
- Shift data `shift_data[(i,j,k]) = 1` if nurse i is scheduled on day j and shift k, it is 0 otherwise.
- Constraints:
- One nurse per shift: `quicksum(shift_data[(n,d,s)] for n in nurses) == 1` for all days and shifts.
- At most one shift per day `quicksum(shift_data[(n,d,s)] for s in shifts) <=1` for all nurses.

Contd..

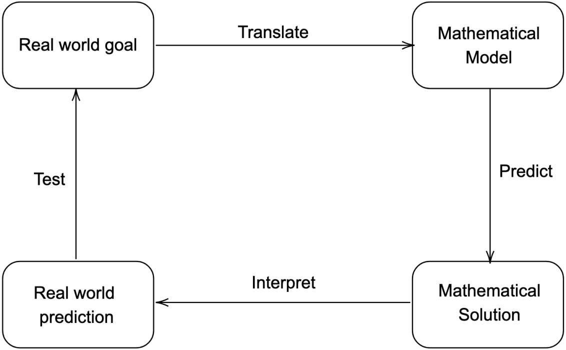
- Objective Function
- Maximize

$$\sum_{d \in \text{days}} \sum_{s \in shifts} \sum_{ninnurses} shiftdata[(n, d, s)] \cdot [rating[n, d, s]]$$

- Solve the Model `model.optimize()`

4 steps of Modelling

- Recall from lecture 1.



-

Tweaks

- Suppose we want to enforce that nurses cannot work 0 rated shifts.
- Each nurse is assigned at least two shifts during the three day period.
- We do this by
- `shift_data[(n,d,s)] <= rating[(n,d,s)]` for all n,d,s.
- This ensures that shift_data is 0 when rating is 0.
- `quicksum(shift_data[(n,i,j)] for i in days for j in shifts)>=2` for all nurses n.

Major Changes

- Suppose now that the union has passed a resolution that shifts cannot be more than 4 hours long. Therefore 6 shifts per day.
- Nurses cannot have more than 2 shifts per day.
- Since the number of shifts now is 18 for a three day schedule.
- There are 4 nurses, therefore each nurse must work at least 4 shifts in three days.

Model

```
nurses = ["A","B","C","D"]
shifts = ["early morning","morning","day","afternoon","ever
days = [1,2,3]
```

- Constraints
- At least 4 total shifts `quicksum(shift_data[(n,i,j)] for i in days for j in shifts) >=4` for all n,d,s.
- One nurse per shift `quicksum(shift_data[(n,d,s)] for n in nurses) == 1`
- At most two shifts a day `quicksum(shift_data[(n,d,s)] for s in shifts) <=2`
- Cannot work 0 rated `shift_data[(n,d,s)] <= rating[(n,d,s)]`

Model vs Real World

- Even with this schedule we may not be satisfied
- It may happen that one a single day nurse B gets early morning shift and evening shift, which may be undersirable because she may prefer consecutive shifts.
- Homework: Think about how you would enforece this constraint in the model.
- Specifically suppose we want that if a nurse gets two shifts then they must be consecutive (1 apart), or at most 2 apart.

Takeaways

- Models help us solve a mathematical problem which may resemble a real world problem.
- However a real world problem may be far more involved and we may have to change the model multiple times.
- Even with lots of changes the model is still a mathematical representation.
- It is essential to get information from the model, but not to lose the sight of the real world problem.
- There may be real world constraints which may be difficult/impossible to state/solve mathematically.