Discrete Algorithms - Linear Programming

Mateusz Pełechaty

12 April 2023

1 Exercise

1.1 Model Description

- $c_{i,j}$ - cost of heart of fuel, i - airport, j - company
- Uc_j - upper bound of fuel in company i
- La_i - lower bound of needed fuel at airport j

Decision Variables

- $x_{i,j} \geq 0$ - how much company j should provide hearts of fuel on airport j

Constraints

- $\sum_i x_{i,j} \leq Uc_j$ for every j
- $\sum_j x_{i,j} \leq La_i$ for every i

Objective Function

- MIN $\sum_{i,j} x_{i,j} \cdot c_{i,j}$

1.2 Testing, Analysis, Interpretation

What's the total min cost of deliveries?

You can see it by using

```
@show objective_value(model)
```

How much every company delivered?

```
per_company_deliveries = [
    sum(
        value(x[airport, i])
        for airport in 1:amount_airports
    )
    for i=1:amount_companies
]
per_company_deliveries
```

Is every company delivering fuel?

```
all(per_company_deliveries .> 0)
```

Are possibilities of delivering fuel exploited?

```
per_company_deliveries .== companies_upper_bounds
```

2 Exercise

Variation of shortest path problem with model with cost and time and T.
Goal is to minimizer cost with time constraint

2.1 Model Description

- N - vertexes
- A - edges
- $c_{i,j}$ - cost of moving across edge i, j
- $t_{i,j}$ - time of moving across edge i, j
- T - Total available time
- S - start vertex
- G - goal vertex

Decision Variables

- $x_{i,j}$ answers $1/0$ question if edge i, j belongs to shortest path between S, G

Constraints

$$d_i = \sum_j x_{j,i} - \sum_j x_{i,j}$$

- $x_i \geq 0$
- $d_i = 1$ if $i == G$
- $d_i = -1$ if $i == S$
- $d_i = 0$ others
- $\sum_{i,j \in A} x_{i,j} \cdot t_{i,j} \leq T$ sum of times on path less than max

Objective Function

- MIN $\sum_{i,j \in A} x_{i,j} \cdot c_{i,j} \leq T$ minimize cost of path

2.2 Testing, Analysis, Interpretation

Is Integer Programming needed?

No. It is not. Sketch proof would look like this:
Assume that Linear Programming has different and better solution than Integer Programming. That means that the path has separated at some point into N not separating paths. Now we can compare path in IP and LP and from there concur, that at least one of the separated paths had lower cost than path in IP. But it would mean that IP path was not the best.

Is Time constraint needed? Would it give acceptable result

Depends what we mean by acceptable result. In the point above I proved that IP solution == LP solution without using Time constraint so proof stays here too. But Solution without time constraint will just MIN cost path.

3 Exercise

3.1 Model Description

- $l_{d,s}$ - lower bounds for amount of cars in given district and shift
- $u_{d,s}$ - upper bounds for amount of cars in given district and shift
- ls_s - lower bound for amount of cars at given shift
- ds_d - lower bound for amount of cars at given district

Decision Variables

-

Constraints

-

Objective Function

-

4 Exercise

4.1 Model Description

- N - board height
- M - board width
- k - Camera vision distance
- C - Set of tuple describing row and column

Decision Variables

- $\forall_{i=1...N,j=1...M} x_{i,j}$ - 1 if camera is on tile (i, j) else 0.

Constraints

- $\forall_{(i,j)\in C} x_{i,j} = 0$
- There is at least one camera at k distance in straight line from any container

Objective Function

$$MIN \sum x_{i,j}$$

5 Exercise

5.1 Model Description

- P_i - Product i, $1 \leq i \leq N$
- M_j - Machine j, $1 \leq j \leq M$
- Ip_i - Income from selling P_i
- Cp_i - Cost of buying raw materials for P_i
- Cm_i - Cost of hour of work of M_i
- d_i - demand for P_i. Effectively upper bound for produce
- T - Time for which every machine is booked in hours
- $A_{i,j}$ time use in minutes for kg of processing of P_i on M_j

Decision Variables

- x_i - processed x_i kgs of P_i

Constraints

- $\forall_{i=1...N} d_i >= x_i >= 0$
- $\forall_{j=1...M} \sum_{i=1}^{N} x_i \cdot A_{i,j} <= 60 \cdot T$

Objective Function

Total Income:

$$TI = \sum_{i=1}^{N} x_i \cdot (Ip_i - Cp_i)$$

Total Machine Cost

$$TMC = \sum_{j=1}^{M} \frac{Cm_j}{60} \cdot \sum_{i=1}^{N} x_i \cdot A_{i,j}$$

Objective function:

$$MAX(TI - TMC)$$