Skills: Intro to Software in OR

Zohaad Fazal

1/21/2020

A0: Investment Plan I

Q1: Linear program formulation and solution

Fox Enterprises needs to make a decision which of 6 available projects they want to invest in over a period of 4 years. They can decide to fully fund, partially fund, or not fund a project at all. Yearly cash outlays are given per project per year, available funds per year are also given. At the end of the 4 years each project pays out an expected return which Fox Enterprises hopes to maximize. Note that if a project is partially funded, only a fraction is paid per year and the return at the end is proportional to this fraction.

Fox Enterprises would like to find the optimal project mix to maximize expected returns.

- **parameters**
  - r_i: expected return for project $i \in P$.
  - c_{ij}: cash outlay for project $i \in P$ in year $j \in Y$.
  - f_j: available funds in year $j \in Y$.
- **variables**
  - x_i: fraction of project $i \in P$ to undertake.
- **objective function**

$$\max \sum_{i \in P} r_i x_i \tag{1}$$

- **constraints**

$$\sum_{i \in P} c_{ij} x_i \leq f_j, \ \forall j \in Y \tag{2}$$

$$x_i \in [0,1] \quad \text{for } i \in P \tag{3}$$

  - (1): Maximizes the fractional returns.
  - (2): We can't exceed the yearly available funds.
  - (3): The percentage, or fraction, of each project must lie between 0 and 1.
- **output**

The optimal value is 176.88651 (rounded), and the optimal project mix is given in the following table:

Table 1: Optimal project mix

Variable	**Fraction**
x_1	1
x_2	1
x_3	1
x_4	1
x_5	0.84
x_6	0

To maximize expected returns Fox Enterprises needs to fully invest in projects 1 through 4, invest 84% in project 5 and not invest in project 6. The optimal expected return is then $176,886.51.

Q2

(a) Binding constraints

To find the binding constraints we check which constraints have 0 slack.

Table 2: Slack values

Slack variable	**Value**
w_1	13.452
w_2	9.803
w_3	3.217
w_4	0

So the 4th constraint is binding.

(b) Increasing RHS of binding constraints

The gain for increasing the binding constraint with one unit is the shadow price. The shadow price for the 4th constraint is 3.21746 (rounded).

Note that since one unit represents $1000, we would gain $3,217.46.

(c) Increasing project 6's return

To find out how much we can increase the return of project 6 without changing the optimal solution we look at the upper bound of the optimality range for project 6. This is 6.06349, so we can increase the return of project 6 to $6,063.49.

Q3: Additional features

To implement the additional feature that if a portion of project 2 is undertaken then at least an equal portion of project 6 must be undertaken:

$$x_6 \geq x_2 \tag{4}$$

To implement the additional feature that last year's remaining funds can be used for the current year, we reformulate the constraints as follows

$$\begin{aligned}
\text{for } j = 1: \quad & \sum_i c_{i1} x_i & & + w_1 = f_1 \\
\text{for } j > 1: \quad & \sum_i c_{ij} x_i & -w_{j-1} & + w_j = f_j
\end{aligned} \tag{5}$$

Where $w_j \in \mathbb{R}_+$ represents the slacks, and we add the slack of last year's constraint to the current year's constraint's right hand side. In (5) we subtract it from the left hand side, which is equivalent to adding it to the right hand side.

Note that since we manually added the slack variables, the constraints become equality constraints.

The optimal value is now 201.62757 (rounded), and the optimal project mix is given in the following table:

Table 3: Optimal project mix

Variable	**Fraction**
x_1	1
x_2	1
x_3	1
x_4	1
x_5	0.77
x_6	1

The optimal expected return is now \$201,627.57. Fox Enterprises needs to invest in all projects fully except project 5; it needs to invest 77%.

B0: Capacitated Facility Location I

Q1: Mixed integer linear program formulation

TrendCoats is a manufacturer that needs to make several decisions to meet an increasing demand of jeans in the North American markets. The first decision is whether to open a new production plant in Wichita or to increase the capacity of the existing plant in Denver. This is an exclusive disjunction, so they may not do both. The second decision is whether to increase the capacity of their warehouses in Chicago and Salt Lake City.

TrendCoats would like to minimize total shipping costs with the addition of the costs associated to the aforementioned decisions.

- **parameters**
  - d_m: the demand for market $m \in M$, $M = \{\text{Seattle, Sacramento, Houston, Toronto, Miami, Detroit}\}$, $d_m \in \mathbb{N}_0$.
  - c_{pw}: the cost per 1000 units to transport from plant $p \in P$ to warehouse $w \in W$, $P = \{\text{Denver, Wichita}\}$, $W = \{\text{Chicago, Salt Lake City}\}$.
  - c_{wm}: the cost per 1000 units to transport from warehouse $w \in W$ to market $m \in M$.
- **variables**
  - $f_{p,w}$: the atomic flow from plant $p \in P$ to warehouse $w \in W$, $f_{p,w} \in \mathbb{N}_0$.
  - $f_{w,m}$: the atomic flow from warehouse $w \in W$ to market $m \in M$, $f_{w,m} \in \mathbb{N}_0$.
  - z_D: the decision whether to increase the Denver plant's capacity, $z_D \in \{0, 1\}$.
  - z_W: the decision whether to open a new plant in Wichita, $z_W \in \{0, 1\}$.
  - z_w: the decision whether to increase warehouse capacity for warehouse $w \in W$, $z_w \in \{0, 1\}$. Also, we use $z_C = z_{\text{Chicago}}$ and $z_S = z_{\text{Salt Lake City}}$.
- **objective function**

$$\min \sum_{p \in P} \sum_{w \in W} c_{p,w} f_{p,w} + \sum_{w \in W} \sum_{m \in M} c_{w,m} f_{w,m} + 1500000 z_D + 2000000 z_W + 250000 z_C + 200000 z_S \quad (6)$$

- **constraints**

$$z_D + z_W \le 1 \quad (7)$$

$$\sum_{w \in W} f_{\text{Denver},w} - 1000 z_D \le 1500 \quad (8)$$

$$\sum_{w \in W} f_{\text{Wichita},w} - 1500 z_W \le 0 \quad (9)$$

$$\sum_{m \in M} f_{w,m} - 1000 z_w \leq 1000, \ \forall w \in W \tag{10}$$

$$\sum_{p \in P} f_{p,w} - \sum_{m \in M} f_{w,m} = 0, \ \forall w \in W \tag{11}$$

$$\sum_{w \in W} f_{w,m} = d_m, \ \forall m \in M \tag{12}$$

$$f_{p,w} \in \mathbb{N}_0, \ \forall p \in P, \ \forall w \in W \tag{13}$$

$$f_{w,m} \in \mathbb{N}_0, \ \forall w \in W, \ \forall m \in M \tag{14}$$

$$z_D, z_W, z_C, z_S \in \{0, 1\} \tag{15}$$

- (6): Minimize over the cost from the plants to the warehouses and the warehouses to the markets. Add \$1.5MM if the Denver plant's capacity is increased, add \$2MM if the plant in Wichita is opened. Finally add \$250k if the Chicago warehouse's capacity is increased and \$200k if the Salt Lake City warehouse's capacity is increased.
- (7): This ensures that the exclusive disjunction is satisfied, either to increase the Denver plant's capacity or to open a plant in Wichita, but not both.
- (8): Capacity constraint for the Denver plant, if it's capacity is increased (measured by z_D), we add 1000 thousand's of units.
- (9): Capacity constraint for the Wichita plant, if it is opened (measured by z_W), we set it's capacity to 1500 thousand's of units, otherwise 0.
- (10): Capacity constraints for the warehouses, if a warehouse's capacity is increased (measured by z_w), we add 1000 thousand's of units.
- (11): At a warehouse, the incoming flow from the plants must equal the outgoing flow to the markets.
- (12): Demand constraints for the markets, all the incoming flow must equal the demand for a given market.
- (13): Flows from plants to warehouses must be non-negative integers.
- (14): Flows from warehouses to markets must be non-negative integers.
- (15): The decision variables must be binary, note that this also imples that $z_w \in \{0, 1\}, \ \forall w \in W$.

Q2: Solution

The optimal objective value found is 2,139,490, that is, the cheapest shipping costs for TrendCoats that satisfy the demand are \$2,139,490.

The decisions TrendCoats needs to make are:

Table 4: Decisions to achieve optimum

Optimal decision variables	**Choice**	**Decision**
$z_D^* = 1$	increase Denver plant capacity	True
$z_W^* = 0$	open Wichita plant	False
$z_C^* = 0$	increase Chicago warehouse capacity	False
$z_S^* = 1$	increase Salt Lake City warehouse capacity	True

The flows from the plants to the warehouses are:

Table 5: Flows from pants to warehouses

Flows in 1000's of units	Chicago	Salt Lake City	**Capacity**
Denver	820	1570	2500
Wichita	0	0	0
Capacity	1000	2000	

The flows from the warehouses to the markets are:

Table 6: Flows from warehouses to markets

Flows in 1000's of units	Seattle	Sacramento	Houston	Toronto	Miami	Detroit
Chicago	0	0	0	500	0	320
Salt Lake City	480	420	220	0	450	0

C0: The Bin Packing Problem

Q1: Integer linear program formulation and relaxation (Q3)

For a given set of items $I = \{1, \ldots, n\}$ each having size $s_i \in [0, 1]$ for $i \in I$ we must determine an allocation of items to bins of size 1, that minimizes the number of bins used.

Note that in the worst case, $s_i \in (\frac{1}{2}, 1]$ for all $i \in I$, then n bins will be used. This is an upper bound so in the worst case we need n bins. Let us denote the bins as $B = \{1, \ldots, n\}$, each having size 1.

- **parameters**
  - s_i: size of item $i \in I$, $s_i \in [0, 1]$.
- **variables**
  - b_j: decision variable whether bin $j \in B$ is used, $b_j \in \{0, 1\}$.
  - x_{ij}: decision variable whether item $i \in I$ is placed in bin $j \in B$, $x_{ij} \in \{0, 1\}$.
- **objective function**

$$\min \sum_{j \in B} b_j \tag{16}$$

- **constraints**

$$\sum_{i \in I} s_i x_{ij} - b_j \leq 0, \ \forall j \in B \tag{17}$$

$$\sum_{j \in B} x_{ij} = 1, \ \forall i \in I \tag{18}$$

$$b_j \in \{0, 1\}, \ \forall j \in B \tag{19}$$

$$x_{ij} \in \{0, 1\}, \ \forall i \in I, \ \forall j \in B \tag{20}$$

  - (16): minimizes the number of bins used.
  - (17): ensures both that if a bin is used, its contents don't exceed the capacity of 1, and if a bin is not used, its contents don't exceed the capacity of 0.
  - (18): ensures that every item is placed in exactly one bin.
  - (19): whether a bin is used is a binary variable.
  - (20): whether an item is placed in a bin is a binary variable.

The linear program relaxation is done by relaxing (19) and (20) to:

$$b_j \in [0, 1], \ \forall j \in B \tag{21}$$

$$x_{ij} \in [0, 1], \ \forall i \in I, \ \forall j \in B \tag{22}$$

Q2-Q5: Results

The limit of the computational time of CPLEX was set to 3600. These are the results we obtained:

Table 7: Results

Instance	Optimal value (ILP)	Run time ILP (ms)	Optimal value (LP)	Run time LP (ms)	GAP
1	4	108	3.53	7	0.1175
2	5	13	4.21	4	0.158
3	16	3701	14.12	11	0.1175
4	15	110	14.47	10	0.0353
5	25	626	23.82	31	0.0472

Q6: Context of instance 1's solution

For instance 1, in the ILP, we obtain the optimal objective value of 4 and the following item placements:

Table 8: Item placements instance 1

rows: items columns: bins	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	1	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	1	0	0	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0	0
9	0	0	1	0	0	0	0	0	0	0
10	0	0	0	0	1	0	0	0	0	0

So we can see that bin 1 has items 4, 6, and 8. Bin 3 has items 3, 7, and 9. Bin 5 has items 2, and 10. Finally, bin 6 has items 1 and 5. The rest of the bins are not used.

D0: Facility Location

Q1: Algorithm for solving shortest paths

We used the Floyd-Warshall algorithm for solving for all pairs shortest paths. The pseudo code is given:

```
let dist be a |V| × |V| array of minimum distances initialized to infinity
for each edge (u, v) do
    dist[u][v] ← w(u, v)  // The weight of the edge (u, v)
for each vertex v do
    dist[v][v] ← 0
for k from 1 to |V|
    for i from 1 to |V|
        for j from 1 to |V|
            if dist[i][j] > dist[i][k] + dist[k][j]
                dist[i][j] ← dist[i][k] + dist[k][j]
            end if
```

The asymptotic running time of this algorithm is $\Theta(|V|^3)$.

The pseudo code for the Floyd-Warshall algorithm is retrieved from (Wikipedia 2020) without further references provided, due to it being original work.

Q2: Problem in the literature

A problem like this is called the uncapacitated facility location problem in the literature (Sviridenko 2002). A key difference with Sviridenko (2002) is that our potential facility locations $\mathcal{F}$ are the same as the demand points $\mathcal{D}$, that is, $\mathcal{F} = \mathcal{D}$.

Another key difference is that fixed costs are equal for each potential facility location. Hence we are able to omit $\sum_{i \in \mathcal{F}} f_i y_i$ from our objective function, where f_i is the fixed cost for potential facility $i \in \mathcal{F}$, and y_i is the decision variable whether facility $i \in \mathcal{F}$ is opened. y_i is analogous to x_j for $j \in R$ in our integer linear program formulation, where R is the set of regions. This integer linear program formulation is described in the next section.

Since our problem assumes fixed costs are equal for every potential facility $j \in R$ and don't longer consider these costs, this problem reduces to the k-median problem (Pedroso et al. 2012). Pedroso et al. (2012) explain that the k-median problem is a variant of the uncapacitated facility location problem.

Q3: Integer Linear Program

We have n regions, with an $n \times n$ distance matrix containing all pairs shortest paths: $\{d_{ij}\}_{i,j=1}^{n}$. We must decide where to place $p < n$ facilities to minimize the total distance traveled over customers in all regions.

- **parameters**
  - n: number of regions. Let the set of regions be $R = \{1, \ldots, n\}$.
  - p: number of facilities that must be placed.
  - d_{ij}: distance from region $i \in R$ to region $j \in R$, $d_{ij} \in [0, \infty)$. Note that $d_{ii} = 0$ for all $i \in R$.
- **variables**
  - x_j: the decision whether to open a facility in region $j \in R$, $x_j \in \{0, 1\}$.
  - c_{ij}: the decision whether customers in region $i \in L$ go to the facility in region $j \in R$, $c_{ij} \in \{0, 1\}$.
- **objective function**

$$\min \sum_{i \in R} \sum_{j \in R} d_{ij} c_{ij} \tag{23}$$

- **constraints**

$$\sum_{j \in R} x_j = p \tag{24}$$

$$c_{ij} - x_j \leq 0, \ \forall j \in R, \ \forall i \in R \tag{25}$$

$$\sum_{j \in R} c_{ij} = 1, \ \forall i \in R \tag{26}$$

$$x_j \in \{0, 1\}, \ \forall j \in R \tag{27}$$

$$c_{ij} \in \{0, 1\}, \ \forall i \in R, \ \forall j \in R \tag{28}$$

  - (23): minimizes over the total distance from all customers at regions i to regions j.
  - (24): ensures that there are exactly p facilities placed.
  - (25): ensures that customers can only travel from region i to region j if there is a facility placed at region j.
  - (26): ensures that a customer in region i travels to precisely one region j.
  - (27): whether a facility is openend in region j is a decision variable.
  - (28): whether customer in region i travels to region j is a decision variable.

Q4: Results

These are the results we obtained, only the running time for `cplex.solve()` was measured.

Table 9: Results

	Optimal value	Run time (ms)
Instance 1		
$p = 2$	20.5	28
$p = 3$	15.4	23
$p = 4$	10.0	7
Instance 2		
$p = 2$	69.5	39
$p = 3$	51.4	22
$p = 4$	41.5	16
Instance 3		
$p = 2$	31.8	40
$p = 3$	21.5	18
$p = 4$	16.7	21
Instance 4		
$p = 2$	57.6	75
$p = 3$	49.3	62
$p = 4$	42.5	225
Instance 5		
$p = 2$	75.9	133
$p = 3$	62.9	174
$p = 4$	54.8	148

Q5: Context of instance 1's solution

We provide the optimal value, that is, the minimized traveling distance. We also describe the placements of the facilities along with the customer allocations to these facilities that minimize the total traveling distance.

For instance 1, with $p = 2$, we obtain the optimal objective value of 20.5. Facilities are opened at regions 1 and 9. The customers of region 1 go to the facility at region 1, and the customers of the other regions go to the facility at region 9.

For instance 1, with $p = 3$, we obtain the optimal objective value of 15.4. Facilities are opened at regions 1, 6, and 9. The customers of region 1 go to the facility at region 1, customers of region 6 go to the facility at region 6, and the customers of the other regions go to the facility at region 9.

For instance 1, with $p = 4$, we obtain the optimal objective value of 10.0. Facilities are opened at regions 1, 2, 6, and 10. Customers from regions 1, 6, and 10 go to the facilities at their respective regions, and the customers of the other regions go to the facility at region 2.

The customer allocation matrices from where this is derived can be found by running the code.

References

Pedroso, João Pedro, Abdur Rais, Mikio Kubo, and Masakazu Muramatsu. 2012. "Facility location problems — Mathematical Optimization: Solving Problems using SCIP and Python." https://scipbook.readthedocs.io/en/latest/flp.html#the-k-median-problem.

Sviridenko, Maxim. 2002. "An Improved Approximation Algorithm for the Metric Uncapacitated Facility Location Problem." In *Integer Programming and Combinatorial Optimization*, edited by William J. Cook and Andreas S. Schulz, 240–57. Berlin, Heidelberg: Springer Berlin Heidelberg.

Wikipedia. 2020. "Floyd-Warshall algorithm — Wikipedia, the Free Encyclopedia." http://en.wikipedia.org/w/index.php?title=Floyd%E2%80%93Warshall%20algorithm&oldid=931174635#Algorithm.