Previous: (5c) A Transshipment Problem

Next: (5e) A Cutting Stock Problem

(5d) A Facility Location Problem

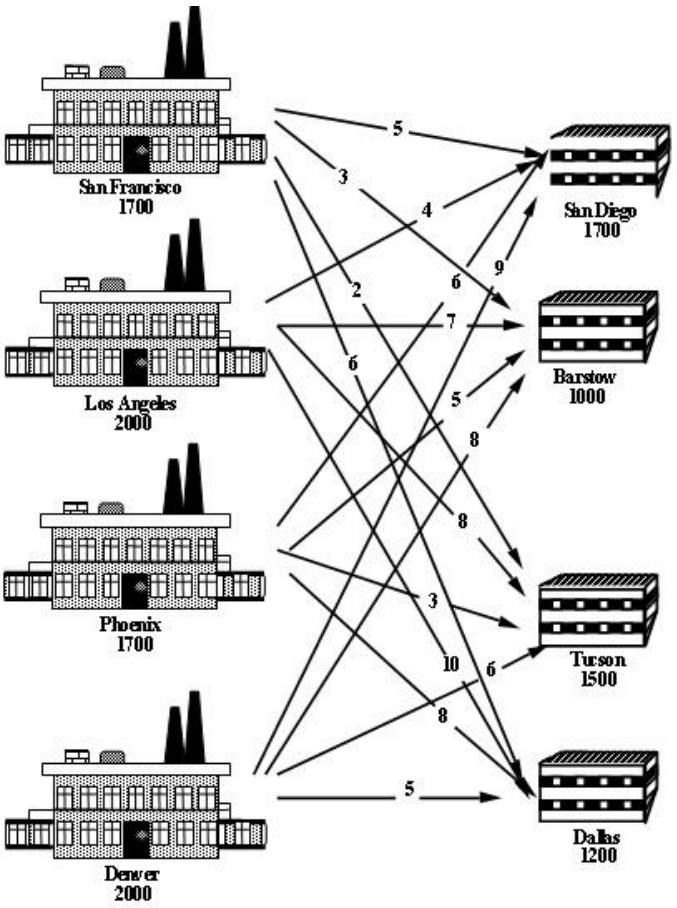
The Computer Plant Problem

Problem Description

Consider the Cosmic Computer company who wish to set up production in America. They are contemplating building production plants in up to 4 locations. Each location has planning restrictions that effectively determine the monthly production possible in each location. The monthly fixed costs of the different locations have been calculated by the accounting department, and are listed below:

Location	Monthly Fixed Cost
San Francisco	$70,000
Los Angeles	$70,000
Phoenix	$65,000
Denver	$70,000

These plants will supply stores at 4 locations. Predicted demand and per-unit transport costs for supplying the 4 production plants to the 4 stores are shown below, along with the quantity each factory can supply:



Where should the company set up their plants to minimise their total costs (fixed plus transportation)?

Formulation

This problem is essentially a transportation problem, but with master-slave constraints at the supply nodes. The lines where this code framework is different/additional to the Transportation Problem code framework will be pointed out.

1. **Identify the Decision Variables**

   The decisions are two-fold:
   1. Where to build the plants?
   2. Where to ship the computers? This is specified by Flow in the Transportation Problem.

2. **Formulate the Objective Function**

   We have already incorporated the transportation cost in the Transportation Problem, now we have to add the fixed (construction + maintenance) cost.

3. **Formulate the Constraints**

   The constraints are the same as in the Transportation Problem, except that the supply is determined by the construction (or not) of a plant:

   Demand at each store must be met. Supply at each factory must be within that plants limitations if it is build, or zero if it is not.

Solution

The introductory commenting and import statement is entered.

```
"""
The Computer Plant Problem for the PuLP Modeller



Authors: Antony Phillips, Dr Stuart Mitchell  2007
"""


#Import PuLP modeller functions
from pulp import *
```

A list of the supply nodes (Plants) is created. The Supply values and Fixed costs of each plant are also entered into a list in a dictionary with the Plant names as the reference keys. (In a Transportation Model, only the Supply values would be entered)

```
# Creates a list of all the supply nodes
Plants = ["San Francisco",
          "Los Angeles",
          "Phoenix",
          "Denver"]



# Creates a dictionary of lists for the number of units of supply at
# each plant and the fixed cost of running each plant
supplyData = {#Plant     Supply  Fixed Cost
          "San Francisco":[1700, 70000],
          "Los Angeles"  :[2000, 70000],
          "Phoenix"      :[1700, 65000],
          "Denver"       :[2000, 70000]
          }
```

A list of the demand nodes (Stores) is created. The Demand values are also entered into a dictionary with the Store names as the reference keys.

```
# Creates a list of all demand nodes
Stores = ["San Diego",
          "Barstow",
          "Tucson",
          "Dallas"]



# Creates a dictionary for the number of units of demand at each store
demand = { #Store    Demand
          "San Diego" :1700,
          "Barstow"   :1000,
          "Tucson"    :1500,
          "Dallas"    :1200}
```

The costs of each of the transportation routes are entered as 4 lists (one for each plant), in a large list. Each of the sub lists contains the transport costs to each of the stores from that plant.

```
# Creates a list of costs for each transportation path
costs = [  #Stores
         #SD BA TU DA
         [5, 3, 2, 6], #SF
         [4, 7, 8, 10],#LA    Plants
         [6, 5, 3, 8], #PH
         [9, 8, 6, 5]  #DE
         ]
```

A list of tuples containing all the arcs in the problem is created.

```
# Creates a list of tuples containing all the possible routes for transport
Routes = [(p,s) for p in Plants for s in Stores]
```

The `supplyData` dictionary is split up into two dictionaries. (additional to the Transportation Problem)

```
# Splits the dictionaries to be more understandable
(supply,fixedCost) = splitDict(supplyData)
```

The cost data is then made into a Dictionary so that 'costs["Phoenix"]["Barstow"]' will return the cost: 5.

```
# The cost data is made into a dictionary
costs = makeDict([Plants,Stores],costs,0)
```

The problem variables are created. This differs from the Transportation Model since two sets of variables must be made, one as usual for the flow and one for whether or not we are to build each plant.

```
# Creates the problem variables of the Flow on the Arcs
flow = LpVariable.dicts("Route",(Plants,Stores),0,None,LpInteger)



# Creates the master problem variables of whether to build the Plants or not
build = LpVariable.dicts("BuildaPlant",Plants,0,1,LpInteger)
```

The variable `prob` is created.

```
# Creates the prob variable to contain the problem data
prob = LpProblem("Computer Plant Problem",LpMinimize)
```

The Objective Function is added to `prob`. This is simply the addition of the transport costs and the fixed costs (the difference from the Tranportation Problem).

```
# The objective function is added to prob - The sum of the transportation costs and the building fixed costs
prob += lpSum([flow[p][s]*costs[p][s] for (p,s) in Routes])+lpSum([fixedCost[p]*build[p] for p in Plants]),"Total Costs"
```

The constraints are added to `prob`. The supply constraint has the 'build[p]' variable in it, since if the plant is not built,

the flow will have to be zero. The demand constraint is standard, ensuring that all demands are met.

```
# The Supply maximum constraints are added for each supply node (plant)
for p in Plants:
    prob += lpSum([flow[p][s] for s in Stores]) <= supply[p]*build[p], "Sum of Products out of Plant %s"%p



# The Demand minimum constraints are added for each demand node (store)
for s in Stores:
    prob += lpSum([flow[p][s] for p in Plants]) >= demand[s], "Sum of Products into Stores %s"%s
```

Following this is the prob.writeLP statement and the standard end to the formulation. The full file is available here. The Optimal Solution for the objective function is $228,100.

Presentation of Solution and Analysis

Your management summary should contain:

1. A brief summary of the problem description;
2. A construction plan for the plants;
3. A transportation plan that describes the shipments from plants to stores;
4. A description of your post-optimal analysis:
    - A brief description of the analysis undertaken (and why);
    - A summary of the results of your analysis;

Implementation and Ongoing Monitoring

Before implementation, sensitivity analysis of both the fixed and transportation costs could identify any data that needs to be confirmed before proceeding. This could also identify any problems should construction costs run over budget.

Ongoing monitoring of the transportation costs, production levels (supply) and demands will produce optimal shiiping plans. If the fixed cost at any plant rises significantly, relocation may be considered (this can also take the form of a facility location problem).


Previous: (5c) A Transshipment Problem


Next: (5e) A Cutting Stock Problem