Group 16 Application

Jonathan De Los Santos Safwan Hasan Daniel Gerhart

Case Study 1: Capacity Concerns

Safwan Hasan

Ryan Miller, either the founder or the CEO of CommuniCorp, is stressed about the financials of his business. CommuniCorp, the manufacturer of pagers, stocks plummeted within the last 12 months, losing about 25% of the pager market. The ticker price is also at an all-time low in the previous 52 weeks. Mr. Miller found that it was due to the lack of transparency and communication between the marketing and warehouse executives.

He turned to the head of Corporate Information Management, Emily Jones, to request her to help him get his company back on track since CommuniCorp was behind on orders, even though their warehouse had adequate inventory.

(a) Emily Jones first decided to evaluate the number and type of servers to purchase on a month-to-month basis. Here, the key parameter is the number of employees in the company, which is 330, who need access to the intranet. An IP model to determine which servers Emily should purchase in a particular month to minimize costs in that month and support all the new users within 5 months, assuming there are 3 server manufacturers (Intel Pentium, SGI, and Sun):

Month 1:

Since we are limited in the budget to purchase a server in the first two months, we cannot spend over \$9500. SGI is also willing to give us a discount of 10 percent off each server purchased, but only if purchased in the first or second month. So SGI Workstation for month 1 would be \$9000 and can support all Sales department employees by month 2, and also a few of the manufacturing department employees. Sun, on the other hand, is willing to give us a 25 percent discount on all servers purchased in the first two months but that is unaffordable since \$18750 is still higher than the \$9500 budget for the first two months. So we will not buy any servers from Sun manufacturer. Therefore, the model is as follows:

Decision variables:

- x1-> the number of Standard Intel Pentium PC servers purchased in month 1
- x2 -> the number of Enhanced Intel Pentium PC servers purchased in month 1
- x3 -> the number of SGI Workstation servers purchased in month 1
- x4 -> the number of Sun Workstation servers purchased in month 1

Constraints:

- The total cost of servers purchased in month 1 cannot exceed $9,500: 2500*x1 + 5000*x2+ 9000*x3 + 18750*x4 <= $9500

Objective function: Maximize efficiency: 2500*x1 + 5000*x2+ 9000*x3 + 18750*x4

In [2]:

```
import pulp
from pulp import *

# Create the problem
prob = LpProblem("Capacity concerns", LpMaximize)

# Define the decision variables
x1 = LpVariable("Standard", lowBound=0, cat = "Integer")
x2 = LpVariable("Enhanced", lowBound=0, cat = "Integer")
x3 = LpVariable("SGI", lowBound=0, cat = "Integer")
x4 = LpVariable("Sun", lowBound=0, cat = "Integer")

# Define the objective function
prob += 2500*x1 + 5000*x2+ 9000*x3 + 18750*x4

# Define the constraints
prob += 2500*x1 + 5000*x2+ 9000*x3 + 18750*x4 <= 9500


# Solve the problem
prob.solve()

# Print the solution status
print("Status:", LpStatus[prob.status])

# Print the optimal solution
print("Optimal Solution:")
print("Standard =", value(x1))
print("Enhanced =", value(x2))
print("SGI =", value(x3))
print("Sun=", value(x4))
print("Optimal Cost =", value(prob.objective))
```

```
Welcome to the CBC MILP Solver
Version: 2.10.3
Build Date: Dec 15 2019

command line - /Users/jdmini/datapad/lib/python3.9/site-packages/pulp/solve
rdir/cbc/osx/64/cbc /var/folders/1n/nvr79nb55tz9j4lsbrw6hsf80000gn/T/1c871a
2b441e44438a273635796f6f3b-pulp.mps max timeMode elapsed branch printingOpt
ions all solution /var/folders/1n/nvr79nb55tz9j4lsbrw6hsf80000gn/T/1c871a2b
441e44438a273635796f6f3b-pulp.sol (default strategy 1)
At line 2 NAME          MODEL
At line 3 ROWS
At line 6 COLUMNS
At line 23 RHS
At line 25 BOUNDS
At line 30 ENDATA
Problem MODEL has 1 rows, 4 columns and 4 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Continuous objective value is 9500 - 0.00 seconds
Cgl0003I 0 fixed, 0 tightened bounds, 1 strengthened rows, 0 substitutions
```

```
Cgl0003I 0 fixed, 0 tightened bounds, 1 strengthened rows, 0 substitutions
Cgl0004I processed model has 1 rows, 3 columns (3 integer (2 of which binar
y)) and 3 elements
Cutoff increment increased from 1e-05 to 500
Cbc0038I Initial state - 1 integers unsatisfied sum - 0.210526
Cbc0038I Solution found of -7500
Cbc0038I Cleaned solution of -7500
Cbc0038I Before mini branch and bound, 2 integers at bound fixed and 0 cont
inuous
Cbc0038I Full problem 1 rows 3 columns, reduced to 0 rows 0 columns
Cbc0038I Mini branch and bound did not improve solution (0.00 seconds)
Cbc0038I Round again with cutoff of -8139.47
Cbc0038I Reduced cost fixing fixed 1 variables on major pass 2
Cbc0038I Pass   1: suminf.    0.07105 (1) obj. -8139.47 iterations 1
Cbc0038I Pass   2: suminf.    0.00000 (0) obj. -9000 iterations 2
Cbc0038I Solution found of -9000
Cbc0038I Cleaned solution of -9000
Cbc0038I Before mini branch and bound, 1 integers at bound fixed and 0 cont
inuous
Cbc0038I Mini branch and bound did not improve solution (0.00 seconds)
Cbc0038I After 0.00 seconds - Feasibility pump exiting with objective of -9
000 - took 0.00 seconds
Cbc0012I Integer solution of -9000 found by feasibility pump after 0 iterat
ions and 0 nodes (0.00 seconds)
Cbc0001I Search completed - best objective -9000, took 1 iterations and 0 n
odes (0.00 seconds)
Cbc0035I Maximum depth 0, 1 variables fixed on reduced cost
Cuts at root node changed objective from -9394.74 to -7500
Probing was tried 0 times and created 0 cuts of which 0 were active after a
dding rounds of cuts (0.000 seconds)
Gomory was tried 0 times and created 0 cuts of which 0 were active after ad
ding rounds of cuts (0.000 seconds)
Knapsack was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)
Clique was tried 0 times and created 0 cuts of which 0 were active after ad
ding rounds of cuts (0.000 seconds)
MixedIntegerRounding2 was tried 0 times and created 0 cuts of which 0 were
active after adding rounds of cuts (0.000 seconds)
FlowCover was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)
TwoMirCuts was tried 0 times and created 0 cuts of which 0 were active afte
r adding rounds of cuts (0.000 seconds)
ZeroHalf was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)

Result - Optimal solution found

Objective value:                9000.00000000
Enumerated nodes:               0
Total iterations:               1
Time (CPU seconds):             0.00
Time (Wallclock seconds):       0.00
```

```
Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.00

Status: Optimal
Optimal Solution:
Standard = 0.0
Enhanced = 0.0
SGI = 1.0
Sun= 0.0
Optimal Cost = 9000.0
```

According to the results, Emily should purchase 1 SGI workstation since that is the optimal number of servers within an optimal budget of $9000 to support all Sales department employees by month 2, and also a few of the manufacturing department employees by month 3.

Month 2

The server purchased in month 1 supports 200 employees, so all 50 Sales employees are supported and can already start to use the intranet. However, she is now left with a budget of \$500, which is not enough to buy any servers. Also, since we used SGI’s discount on month 1, we will have to pay the entire \$10,000 for another SGI server. Let’s still look at what our output might be with such complicated constraints:

Decision variables:

- x1-> the number of Standard Intel Pentium PC servers purchased in month 2
- x2 -> the number of Enhanced Intel Pentium PC servers purchased in month 2
- x3 -> the number of SGI Workstation servers purchased in month 2
- x4 -> the number of Sun Workstation servers purchased in month 2

Constraints:

- The total cost of servers purchased in month 2 cannot exceed $500: 2500*x1 + 5000*x2 + 10000*x3 + 18750*x4 <= $500

Objective function: Maximize efficiency: 2500*x1 + 5000*x2 + 10000*x3 + 18750*x4

In [3]:

```python
import pulp
from pulp import *

# Create the problem
prob = LpProblem("Capacity concerns", LpMaximize)

# Define the decision variables
x1 = LpVariable("Standard", lowBound=0, cat = "Integer")
x2 = LpVariable("Enhanced", lowBound=0, cat = "Integer")
x3 = LpVariable("SGI", lowBound=0, cat = "Integer")
x4 = LpVariable("Sun", lowBound=0, cat = "Integer")

# Define the objective function
prob += 2500*x1 + 5000*x2+ 10000*x3 + 18750*x4

# Define the constraints
prob += 2500*x1 + 5000*x2+ 10000*x3 + 18750*x4 <= 500

# Solve the problem
prob.solve()

# Print the solution status
print("Status:", LpStatus[prob.status])

# Print the optimal solution
print("Optimal Solution:")
print("Standard =", value(x1))
print("Enhanced =", value(x2))
print("SGI =", value(x3))
print("Sun=", value(x4))
print("Optimal Cost =", value(prob.objective))
```

```
Welcome to the CBC MILP Solver
Version: 2.10.3
Build Date: Dec 15 2019

command line - /Users/jdmini/datapad/lib/python3.9/site-packages/pulp/solve
rdir/cbc/osx/64/cbc /var/folders/1n/nvr79nb55tz9j4lsbrw6hsf80000gn/T/57e262
dc69484cc3bd697875f02b9170-pulp.mps max timeMode elapsed branch printingOpt
ions all solution /var/folders/1n/nvr79nb55tz9j4lsbrw6hsf80000gn/T/57e262dc
69484cc3bd697875f02b9170-pulp.sol (default strategy 1)
At line 2 NAME          MODEL
At line 3 ROWS
At line 6 COLUMNS
At line 23 RHS
At line 25 BOUNDS
At line 30 ENDATA
Problem MODEL has 1 rows, 4 columns and 4 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Continuous objective value is 500 - 0.00 seconds
Cgl0004I processed model has 0 rows, 0 columns (0 integer (0 of which binar
y)) and 0 elements
```

```
Cbc3007W No integer variables - nothing to do
Cuts at root node changed objective from 0 to -1.79769e+308
Probing was tried 0 times and created 0 cuts of which 0 were active after a
dding rounds of cuts (0.000 seconds)
Gomory was tried 0 times and created 0 cuts of which 0 were active after ad
ding rounds of cuts (0.000 seconds)
Knapsack was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)
Clique was tried 0 times and created 0 cuts of which 0 were active after ad
ding rounds of cuts (0.000 seconds)
MixedIntegerRounding2 was tried 0 times and created 0 cuts of which 0 were
active after adding rounds of cuts (0.000 seconds)
FlowCover was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)
TwoMirCuts was tried 0 times and created 0 cuts of which 0 were active afte
r adding rounds of cuts (0.000 seconds)
ZeroHalf was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)

Result - Optimal solution found

Objective value:                0.00000000
Enumerated nodes:               0
Total iterations:               1
Time (CPU seconds):             0.00
Time (Wallclock seconds):       0.01

Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.02

Status: Optimal
Optimal Solution:
Standard = 0.0
Enhanced = 0.0
SGI = 0.0
Sun= 0.0
Optimal Cost = 0.0
```

According to the results, it also shows CommuniCorp cannot buy any server because of the budget constraint.

Month 3

Month 3:

By now, we currently have an SGI Workstation server that already supports up to 200 employees. 50 of the Sales department employees and 150 out of the 180 Manufacturing employees are already covered. So for month 3, we would need to purchase a server to cover the remaining 30 Manufacturing employees while keeping the costs down (not necessarily a constraint that needs to be defined). We also have a new constraint: the manufacturing employees need three of the more powerful servers to communicate using the intranet. This means that we cannot purchase the Standard Intel Pentium PC server. We also lost the 25% off of Sun Workstation servers since it's already month 3, and have to pay the full price of $25000 for a Sun Workstation server. We are not, however, constrained by the budget. So our model would be as follows:

Decision variables:

- x1-> the number of Standard Intel Pentium PC servers purchased in month 3
- x2 -> the number of Enhanced Intel Pentium PC servers purchased in month 3
- x3 -> the number of SGI Workstation servers purchased in month 3
- x4 -> the number of Sun Workstation servers purchased in month 3

Constraints:

- At least one of the 3 more powerful servers: x2 + x3 + x4 >= 1
- Cover at least 30 of Manufacturing department employees: 30*x1 + 80*x2 + 200*x3 + 2000*x4 >= 30

Objective function: Minimize: 2500*x1 + 5000*x2 + 10000*x3 + 25000*x4

In [4]:

```
import pulp
from pulp import *

# Create the problem
prob = LpProblem("Capacity concerns", LpMinimize)

# Define the decision variables
x1 = LpVariable("Standard", lowBound=0, cat = "Integer")
x2 = LpVariable("Enhanced", lowBound=0, cat = "Integer")
x3 = LpVariable("SGI", lowBound=0, cat = "Integer")
x4 = LpVariable("Sun", lowBound=0, cat = "Integer")


# Define the objective function
prob += 2500*x1 + 5000*x2+ 10000*x3 + 25000*x4

# Define the constraints
prob += x2 + x3 + x4 >= 1
prob += 30*x1 + 80*x2 + 200*x3 + 2000*x4 >= 30

# Solve the problem
prob.solve()

# Print the solution status
print("Status:", LpStatus[prob.status])

# Print the optimal solution
print("Optimal Solution:")
print("Standard =", value(x1))
print("Enhanced =", value(x2))
print("SGI =", value(x3))
print("Sun=", value(x4))
print("Optimal Cost =", value(prob.objective))
```

```
Welcome to the CBC MILP Solver
Version: 2.10.3
Build Date: Dec 15 2019

command line - /Users/jdmini/datapad/lib/python3.9/site-packages/pulp/solve
rdir/cbc/osx/64/cbc /var/folders/1n/nvr79nb55tz9j4lsbrw6hsf80000gn/T/3bb6ce
769eff4630bf5cdabb9f3f8229-pulp.mps timeMode elapsed branch printingOptions
all solution /var/folders/1n/nvr79nb55tz9j4lsbrw6hsf80000gn/T/3bb6ce769eff4
630bf5cdabb9f3f8229-pulp.sol (default strategy 1)
At line 2 NAME          MODEL
At line 3 ROWS
At line 7 COLUMNS
At line 27 RHS
At line 30 BOUNDS
At line 35 ENDATA
Problem MODEL has 2 rows, 4 columns and 7 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Continuous objective value is 5000 - 0.00 seconds
```

```
Cgl0003I 0 fixed, 4 tightened bounds, 1 strengthened rows, 0 substitutions
Cgl0003I 0 fixed, 0 tightened bounds, 1 strengthened rows, 0 substitutions
Cgl0003I 0 fixed, 0 tightened bounds, 1 strengthened rows, 0 substitutions
Cgl0004I processed model has 2 rows, 4 columns (4 integer (0 of which binar
y)) and 7 elements
Cutoff increment increased from 1e-05 to 2500
Cbc0012I Integer solution of 5000 found by greedy cover after 0 iterations
and 0 nodes (0.01 seconds)
Cbc0001I Search completed - best objective 5000, took 0 iterations and 0 no
des (0.01 seconds)
Cbc0035I Maximum depth 0, 0 variables fixed on reduced cost
Cuts at root node changed objective from 5000 to 5000
Probing was tried 0 times and created 0 cuts of which 0 were active after a
dding rounds of cuts (0.000 seconds)
Gomory was tried 0 times and created 0 cuts of which 0 were active after ad
ding rounds of cuts (0.000 seconds)
Knapsack was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)
Clique was tried 0 times and created 0 cuts of which 0 were active after ad
ding rounds of cuts (0.000 seconds)
MixedIntegerRounding2 was tried 0 times and created 0 cuts of which 0 were
active after adding rounds of cuts (0.000 seconds)
FlowCover was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)
TwoMirCuts was tried 0 times and created 0 cuts of which 0 were active afte
r adding rounds of cuts (0.000 seconds)
ZeroHalf was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)

Result - Optimal solution found

Objective value:                5000.00000000
Enumerated nodes:               0
Total iterations:               0
Time (CPU seconds):             0.00
Time (Wallclock seconds):       0.01

Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.01

Status: Optimal
Optimal Solution:
Standard = 0.0
Enhanced = 1.0
SGI = 0.0
Sun= 0.0
Optimal Cost = 5000.0
```

According to the results, Emily should purchase 1 Enhanced Intel Pentium PC server since that is the optimal number of servers to support the remaining 30 employees from the Manufacturing department, all 30 employees from the Warehouse department, and 20 out of 70 employees from the Marketing department. So now, the remaining number of employees is 50 who do not need to use the intranet until month 5 and belong to the Marketing department.

Month 4

The server purchased in month 3 supports the remaining 30 employees from the Manufacturing department, all 30 employees from the Warehouse department, and 20 out of 70 employees from the Marketing department. The remaining 50 employees from the Marketing department do not need the intranet until month 5, and hence there are no constraints this month. So Emily shouldn’t need to purchase any servers this month but let’s still see an LP model:

Decision variables:

- x1-> the number of Standard Intel Pentium PC servers purchased in month 4
- x2 -> the number of Enhanced Intel Pentium PC servers purchased in month 4
- x3 -> the number of SGI Workstation servers purchased in month 4
- x4 -> the number of Sun Workstation servers purchased in month 4

Constraints: None

Objective function: Minimize: 2500*x1 + 5000*x2 + 10000*x3 + 18750*x4

In [5]:

```
import pulp
from pulp import *

# Create the problem
prob = LpProblem("Capacity concerns", LpMaximize)

# Define the decision variables
x1 = LpVariable("Standard", lowBound=0, cat = "Integer")
x2 = LpVariable("Enhanced", lowBound=0, cat = "Integer")
x3 = LpVariable("SGI", lowBound=0, cat = "Integer")
x4 = LpVariable("Sun", lowBound=0, cat = "Integer")

# Define the objective function
prob += 2500*x1 + 5000*x2+ 10000*x3 + 18750*x4

# Solve the problem
prob.solve()

# Print the solution status
print("Status:", LpStatus[prob.status])

# Print the optimal solution
print("Optimal Solution:")
print("Standard =", value(x1))
print("Enhanced =", value(x2))
print("SGI =", value(x3))
print("Sun=", value(x4))
print("Optimal Cost =", value(prob.objective))
```

```
Welcome to the CBC MILP Solver
Version: 2.10.3
Build Date: Dec 15 2019

command line - /Users/jdmini/datapad/lib/python3.9/site-packages/pulp/solve
rdir/cbc/osx/64/cbc /var/folders/1n/nvr79nb55tz9j4lsbrw6hsf80000gn/T/524366
5230c7451b8d3db9db47f07332-pulp.mps max timeMode elapsed branch printingOpt
ions all solution /var/folders/1n/nvr79nb55tz9j4lsbrw6hsf80000gn/T/52436652
30c7451b8d3db9db47f07332-pulp.sol (default strategy 1)
At line 2 NAME          MODEL
At line 3 ROWS
At line 5 COLUMNS
At line 18 RHS
At line 19 BOUNDS
At line 24 ENDATA
Problem MODEL has 0 rows, 4 columns and 0 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Problem is unbounded - 0.00 seconds
Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.00

Status: Unbounded
Optimal Solution:
Standard = 0.0
Enhanced = 0.0
SGI = 0.0
Sun= 0.0
Optimal Cost = 0.0
```

According to the results, it also shows the optimal number of servers needed is 0.

Month 5:

We need to buy at least one server to support the remaining 50 employees from the Marketing department while keeping the costs down. So the model is as follows:

Decision variables: x1-> the number of Standard Intel Pentium PC servers purchased in month 5 x2 -> the number of Enhanced Intel Pentium PC servers purchased in month 5 x3 -> the number of SGI Workstation servers purchased in month 5 x4 -> the number of Sun Workstation servers purchased in month 5

Constraints: Cover at least 50 Marketing department employees: 30*x1 + 80*x2 + 200*x3 + 2000*x4 >= 50

Objective function: Minimize: 2500*x1 + 5000*x2 + 10000*x3 + 25000*x4

In [6]:

```
import pulp
from pulp import *

# Create the problem
prob = LpProblem("Capacity concerns", LpMinimize)

# Define the decision variables
x1 = LpVariable("Standard", lowBound=0, cat = "Integer")
x2 = LpVariable("Enhanced", lowBound=0, cat = "Integer")
x3 = LpVariable("SGI", lowBound=0, cat = "Integer")
x4 = LpVariable("Sun", lowBound=0, cat = "Integer")


# Define the objective function
prob += 2500*x1 + 5000*x2+ 10000*x3 + 25000*x4

# Define the constraints
prob += 30*x1 + 80*x2 + 200*x3 + 2000*x4 >= 50

# Solve the problem
prob.solve()

# Print the solution status
print("Status:", LpStatus[prob.status])

# Print the optimal solution
print("Optimal Solution:")
print("Standard =", value(x1))
print("Enhanced =", value(x2))
print("SGI =", value(x3))
print("Sun=", value(x4))
print("Optimal Cost =", value(prob.objective))
```

```
Welcome to the CBC MILP Solver
Version: 2.10.3
Build Date: Dec 15 2019

command line - /Users/jdmini/datapad/lib/python3.9/site-packages/pulp/solve
rdir/cbc/osx/64/cbc /var/folders/1n/nvr79nb55tz9j4lsbrw6hsf80000gn/T/0cf52a
21520e4dae976eb4b18a78b50f-pulp.mps timeMode elapsed branch printingOptions
all solution /var/folders/1n/nvr79nb55tz9j4lsbrw6hsf80000gn/T/0cf52a21520e4
dae976eb4b18a78b50f-pulp.sol (default strategy 1)
At line 2 NAME          MODEL
At line 3 ROWS
At line 6 COLUMNS
At line 23 RHS
At line 25 BOUNDS
At line 30 ENDATA
Problem MODEL has 1 rows, 4 columns and 4 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Continuous objective value is 625 - 0.00 seconds
Cgl0003I 0 fixed, 4 tightened bounds, 1 strengthened rows, 0 substitutions
```

```
Cgl0003I 0 fixed, 0 tightened bounds, 1 strengthened rows, 0 substitutions
Cgl0003I 0 fixed, 0 tightened bounds, 1 strengthened rows, 0 substitutions
Cgl0004I processed model has 1 rows, 4 columns (4 integer (0 of which binar
y)) and 4 elements
Cutoff increment increased from 1e-05 to 2500
Cbc0012I Integer solution of 7500 found by greedy cover after 0 iterations
and 0 nodes (0.00 seconds)
Cbc0012I Integer solution of 5000 found by DiveCoefficient after 0 iteratio
ns and 0 nodes (0.00 seconds)
Cbc0001I Search completed - best objective 5000, took 0 iterations and 0 no
des (0.00 seconds)
Cbc0035I Maximum depth 0, 0 variables fixed on reduced cost
Cuts at root node changed objective from 4166.67 to 4166.67
Probing was tried 0 times and created 0 cuts of which 0 were active after a
dding rounds of cuts (0.000 seconds)
Gomory was tried 0 times and created 0 cuts of which 0 were active after ad
ding rounds of cuts (0.000 seconds)
Knapsack was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)
Clique was tried 0 times and created 0 cuts of which 0 were active after ad
ding rounds of cuts (0.000 seconds)
MixedIntegerRounding2 was tried 0 times and created 0 cuts of which 0 were
active after adding rounds of cuts (0.000 seconds)
FlowCover was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)
TwoMirCuts was tried 0 times and created 0 cuts of which 0 were active afte
r adding rounds of cuts (0.000 seconds)
ZeroHalf was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)

Result - Optimal solution found

Objective value:                5000.00000000
Enumerated nodes:               0
Total iterations:               0
Time (CPU seconds):             0.00
Time (Wallclock seconds):       0.00

Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.01

Status: Optimal
Optimal Solution:
Standard = 2.0
Enhanced = 0.0
SGI = 0.0
Sun= 0.0
Optimal Cost = 5000.0
```

According to the results, Emily needs to purchase two Standard Intel Pentium PC servers that cover 60 employees and we only needed to cover 50, all within an optimal budget of $5000.

To summarize, Emily needs to purchase 1 SGI Workstation in month 1, 1 Enhanced Intel Pentium PC server in month 3, and 2 Standard Intel Pentium PC servers in month 5 which costs a total of $19,000 altogether.

(b) Emily decides to take a different approach to this dilemma of capacity concerns. She, therefore, decides to evaluate the number and type of servers to purchase over the entire planning period.

The model code is as follows:

In [7]:

```
from pulp import *

# Define the problem
problem = LpProblem("Server Purchase problem", LpMinimize)

# Define decision variables
x = {(month, server): LpVariable(f"x{month}{server}", lowBound=0, cat="Integ

# Define objective function
objective_function = lpSum([
    2500 * x[month, 1] + 5000 * x[month, 2] + (9000 if month in [1, 2] else
    for month in range(1, 6)
])
problem += objective_function

# Define constraints
problem += x[1, 1] + x[1, 2] + x[1, 3] + x[1, 4] == 0, "Month 1 Constraint"
problem += lpSum([2500 * x[1, 1] + 5000 * x[1, 2] + 9000 * x[1, 3] + 18750 *
problem += x[2, 1] + x[2, 2] + x[2, 3] + x[2, 4] >= 2, "Month 2 Constraint"
problem += x[3, 1] + x[3, 2] + x[3, 3] + x[3, 4] >= 6, "Month 3 Constraint"
problem += x[4, 1] + x[4, 2] + x[4, 3] + x[4, 4] >= 1, "Month 4 Constraint"
problem += x[5, 1] + x[5, 2] + x[5, 3] + x[5, 4] >= 3, "Month 5 Constraint"
problem += x[3, 2] + x[3, 3] + x[3, 4] >= 1, "Manufacturing Requirement"

# Solve the problem
problem.solve()

# Print the results
for month in range(1, 6):
    print(f"Month {month}:")
    for server in range(1, 5):
        print(f"Number of server type {server} to purchase: {int(x[month, se
print(f"Total cost: ${value(problem.objective)}")
```

```
Welcome to the CBC MILP Solver
Version: 2.10.3
Build Date: Dec 15 2019
```

```
command line - /Users/jdmini/datapad/lib/python3.9/site-packages/pulp/solve
rdir/cbc/osx/64/cbc /var/folders/1n/nvr79nb55tz9j4lsbrw6hsf80000gn/T/90c67e
96cfff48b5990db088d43e275d-pulp.mps timeMode elapsed branch printingOptions
all solution /var/folders/1n/nvr79nb55tz9j4lsbrw6hsf80000gn/T/90c67e96cfff4
8b5990db088d43e275d-pulp.sol (default strategy 1)
At line 2 NAME          MODEL
At line 3 ROWS
At line 12 COLUMNS
At line 104 RHS
At line 112 BOUNDS
At line 133 ENDATA
Problem MODEL has 7 rows, 20 columns and 31 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Continuous objective value is 32500 - 0.00 seconds
Cgl0003I 0 fixed, 2 tightened bounds, 0 strengthened rows, 0 substitutions
Cgl0004I processed model has 3 rows, 4 columns (4 integer (1 of which binar
y)) and 6 elements
Cutoff increment increased from 1e-05 to 2500
Cbc0012I Integer solution of 32500 found by DiveCoefficient after 0 iterati
ons and 0 nodes (0.00 seconds)
Cbc0001I Search completed - best objective 32500, took 0 iterations and 0 n
odes (0.00 seconds)
Cbc0035I Maximum depth 0, 0 variables fixed on reduced cost
Cuts at root node changed objective from 32500 to 32500
Probing was tried 0 times and created 0 cuts of which 0 were active after a
dding rounds of cuts (0.000 seconds)
Gomory was tried 0 times and created 0 cuts of which 0 were active after ad
ding rounds of cuts (0.000 seconds)
Knapsack was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)
Clique was tried 0 times and created 0 cuts of which 0 were active after ad
ding rounds of cuts (0.000 seconds)
MixedIntegerRounding2 was tried 0 times and created 0 cuts of which 0 were
active after adding rounds of cuts (0.000 seconds)
FlowCover was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)
TwoMirCuts was tried 0 times and created 0 cuts of which 0 were active afte
r adding rounds of cuts (0.000 seconds)
ZeroHalf was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)

Result - Optimal solution found

Objective value:                32500.00000000
Enumerated nodes:               0
Total iterations:               0
Time (CPU seconds):             0.00
Time (Wallclock seconds):       0.00

Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.00
```

```
Month 1:
Number of server type 1 to purchase: 0
Number of server type 2 to purchase: 0
Number of server type 3 to purchase: 0
Number of server type 4 to purchase: 0
Month 2:
Number of server type 1 to purchase: 2
Number of server type 2 to purchase: 0
Number of server type 3 to purchase: 0
Number of server type 4 to purchase: 0
Month 3:
Number of server type 1 to purchase: 5
Number of server type 2 to purchase: 1
Number of server type 3 to purchase: 0
Number of server type 4 to purchase: 0
Month 4:
Number of server type 1 to purchase: 1
Number of server type 2 to purchase: 0
Number of server type 3 to purchase: 0
Number of server type 4 to purchase: 0
Month 5:
Number of server type 1 to purchase: 3
Number of server type 2 to purchase: 0
Number of server type 3 to purchase: 0
Number of server type 4 to purchase: 0
Total cost: $32500.0
```

To summarize, Emily needs to purchase 2 SGI Workstation servers in month 2, 5 Standard Intel Pentium PC servers and 1 Enhanced Intel Pentium PC server in month 3, and 1 Standard Intel Pentium PC servers in month 4, and 3 Standard Intel Pentium PC servers in month 5, which costs a total of $32,500 altogether.

c) The reason why the answer differs between the two methods is that they have different planning horizons. The first method in part (a) involves Emily evaluating the number and type of servers to purchase on a monthly basis, resulting in separate decisions for each month. However, this approach leads to a suboptimal solution as it doesn't consider potential cost savings or capacity requirements for the entire planning period. On the other hand, the second method in part (b) involves Emily evaluating the number and type of servers to purchase over the entire planning period. This approach allows her to make better-informed decisions that minimize the total cost while meeting the capacity requirements for all departments over the entire timeline.

d) Emily may have overlooked some costs in her problem formulation, such as regular maintenance and support costs, varying energy costs due to different power consumption of servers, depreciation costs, setup and installation costs, and training and onboarding costs for employees to use the new intranet system effectively.

e) CommuniCorp's intranet implementation might encounter additional concerns from various departments, including security, usability, integration, scalability, reliability, performance, customization, and flexibility. These concerns encompass protecting sensitive information, ensuring user-friendliness, integrating with existing software, accommodating future growth, providing reliable and efficient operations, and being flexible to meet different department needs.