SUPPLIER SELECTION ASSIGMENT

LOGISTICS MANAGEMENT 704
PROF. RUBEN PROAÑO

AMANDA JOHNSON
MILWINDA MONTERO
NICK MORRIS

Assignment Description:

Implement the model demonstrated in the paper, *Supplier Selection with Multiple Criteria in Volume Discount Environments*, by Weijung Xia and Zhiming Wu. It dealt with determining optimal purchasing policy from multiple suppliers, taking many factors into account. Specific factors mentioned were price, quality, and service. Quality and service had multiple sub-criteria. Price varied along with a quantity discount with multiple levels.

Results:

Table 1. Suppliers' final weights

Supplier's Final Weights									
Supplier	Price	Technical level	Defects	Reliability	On-time delivery	Supply capacity	Repair turnaround	Warranty period	Supplier Weights
A	0.3380	0.2000	0.0732	0.2121	0.2392	0.2564	0.2353	0.1818	0.2273
B	0.2503	0.2000	0.1463	0.2242	0.2392	0.1805	0.2353	0.3636	0.2274
C	0.2209	0.3000	0.1951	0.2727	0.2738	0.2018	0.1765	0.2727	0.2404
D	0.1908	0.3000	0.5854	0.2909	0.2478	0.3613	0.3529	0.1818	0.3049
Global Weight	0.2961	0.1867	0.1658	0.0553	0.0804	0.0676	0.0466	0.1015	

After the calculation of the suppliers' weights (see table 1) following the AHP with the entropy method, was needed the use of AMPL to solve the problem of the supplier selection adapting the integer program created by the authors to the requirements of the assignment.

In the specific instance of our implementation, it was determined that we should only purchase from suppliers C and D, as this allowed us to take advantage of each supplier's highest level of discount (10%). The specific quantities to purchase were 356 from supplier C with a business volume purchased of $223,644 and 444 from supplier D with a business volume purchased of $323,556. For our work, please see the Appendix on the next page and the attached excel file.

Lessons Learned:

We gained insight into one method of developing a strategy for selecting suppliers based on multiple criteria. The model could easily be modified to rely on other input criteria and further sub-levels if needed. It is both simple enough to be well understood and flexible enough to stand up to multiple situations for use. Finally, this is an integer program, not a linear one, making it computationally intensive in case of large data.

Appendix:

```
#Suppliers selection model


#Sets

set J; #set of Suppliers

set R; #Set of discount interval of supplier j


#Parameters

param w{j in J}; #final weight of supplier j

param b{j in J, r in R}; #Upper limit in interval r of supplier......

param d{j in J, r in R}; #discount coefficient associated with interval r.....

param P{j in J}; #unit price of item quoted by supplier j

param q{j in J}; #defective rate of item  offered by supplier j

param Q; #the buyer's maximum acceptable defective rate of item

param t{j in J}; #on-time delivery rate of item  offered by supplier j

param T; #the buyer's minimum acceptable on-time delivery rate of item

param C{j in J}; #maximum supply capacity of item  offered by supplier j

param D; #total demand of item


#Variables

var X{j in J}>=0; #units of item to purchase from supplier j

var V{j in J,r in R}; #business volume purchased from supplier j in discount interval
r

var y{j in J,r in R} binary; #binary integer variable .....


#Objective Function

maximize Z1: sum{j in J} (w[j]*X[j]);

minimize Z2: sum{j in J, r in R: r in 1..3} (1-d[j,r])*V[j,r];

minimize Z3: sum{j in J} (q[j]*X[j]);

maximize Z4: sum{j in J} (t[j]*X[j]);
```

#Constraints

```
 s.t. cons1{j in J}: sum{r in R: r in 1..3}(V[j,r]) =(P[j]*X[j]);

 s.t. cons2{j in J}: X[j] <= C[j]; #Capacity constraint

 s.t. cons3{j in J, r in R: r in 1..3}: b[j,r-1]*y[j,r] <=V[j,r]; #Discount
Constraint 1

 s.t. cons31{j in J, r in R: r in 1..3}: V[j,r] <= b[j,r]*y[j,r];

 s.t. cons4{j in J}: sum{r in R: r in 1..3} y[j,r] <=1; #Discount Constraint 2

 s.t. cons5: sum{j in J} X[j]= D; # Demand Constraint

 s.t. cons6: sum{j in J} (q[j]*X[j])<= Q*D; #Quality constraint

 s.t. cons7: sum{j in J} ((1-t[j])*X[j])<=(1-T)*D # Delivery constraint
```

#Data

```
set J:= 1 2 3 4;

set R:= 0 1 2 3;


param w:=

1 0.2273

2 0.2274

3 0.2474

4 0.3049;


param b: 0 1 2 3 :=

1 0 10000 20000 100000000

2 0 10000 20000 100000000

3 0 10000 20000 100000000

4 0 10000 20000 100000000;


param d: 1 2 3:=

1 0 0.10 0.20
```

```
2 0 0.10 0.20
3 0 0.10 0.20
4 0 0.10 0.20 ;

param P:=
1 411
2 555
3 629
4 728;

param q:=
1 0.08
2 0.04
3 0.03
4 0.01;
param Q:= 0.02;
param t:=
1 0.83
2 0.83
3 0.95
4 0.86;

param T:= 0.90;

param C:=
1 648
2 456
3 510
4 913;

param D:= 800;
```

#Run

```
;
reset;
model trial.txt;
data data1.txt;
option solver cplex;
solve;


display X;
display Z1, Z2, Z3, Z4;
display y;
display V;
```

```
ampl: include run1.txt;
CPLEX 12.6.2.0: optimal integer solution; objective 223.4755556
0 MIP simplex iterations
0 branch-and-bound nodes
No basis.
Objective = Z1
X [*] :=
1     0
2     0
3   355.556
4   444.444
;

Z1 = 223.476
Z2 = 437760
Z3 = 15.1111
Z4 = 720

y :=
1 0   0
1 1   1
1 2   0
1 3   0
2 0   0
2 1   1
2 2   0
2 3   0
3 0   0
3 1   0
3 2   0
3 3   1
4 0   0
4 1   0
4 2   0
4 3   1
;

V :=
1 0         0
1 1         0
1 2         0
1 3         0
2 0         0
2 1         0
2 2         0
2 3         0
3 0         0
3 1         0
3 2         0
3 3    223644
4 0         0
4 1         0
4 2         0
4 3    323556
;
```