Report: Managing Operations
A Case Study in Operations Research

Zhengze Yu,* & Xinpeng Jin†
University of Chinese Academy of Sciences

October 24, 2019

*yuzhengze@ict.ac.cn 201928013229043 The School of Computer Science and Technology
†jinxinpeng@mails.ucas.ac.cn 201828015651001 College of Advanced Agricultural Science

1 Description of Problem

Golf-Sport is a small-sized company that produces high-quality components for people who build their own golf clubs and prebuilt sets of clubs. There are five components: steel shafts, graphite shafts, forged iron heads, metal wood heads, and metal wood heads with titanium inserts; these products are produced in three plants: Chandler, Glendale, and Tucson. Each plant can produce any of the components, although each plant has a different set of individual constraints and unit costs. These constraints cover labor and packaging machine time (the machine is used for all components). Note that even though the components are identical in the three plants, different production processes are used, and therefore the products use different amounts of resources in different plants.

Besides component sales, the company takes the components and manufactures sets of golf clubs. Each set requires 13 shafts, 10 iron heads, and 3 wood heads. All of the shafts in a set must be the same type (steel or graphite), and all of the wood heads must be the same type (metal or metal with inserts). Each plant has unique assembly times. Furthermore, each plant of Golf-Sport has a retail outlet to sell components and sets, and the specific plant is the only supplier for its retail outlet. The minimum and maximum amounts of demand for each location are unique.

This planning problem is for two months. The material, production, and assembly costs increase by 12% for the second month, and production times are stationary. Inventory costs are based on end-of-period inventory for each product set and cost out at 8% of the cost values of material, production and assembly. Additionally, each product generates different revenue depending on where it is made. Initially, there is no inventory.

The corporation controls the capital available or expenses; the cash requirements for each product are given. There is a total of $20,000 available for advertising for the entire system during each month, and any money not spent in a month is not available the next month. The corporation also controls graphite. Each shaft requires 4 ounces of graphite; a total of 1,000 pounds is available for each of the two months.

Your job is to determine a recommendation for the company. A recommendation must include a plan for production and sales. In addition, you should also address the following sensitivity-analysis issues in your recommendation:

- If you could get more graphite or advertising cash, how much would you like, how would you use it, and what would you be willing to pay?
- At what site(s) would you like to add extra packing machine hours, assembly hours, and/or extra labor hours? How much would you be willing to pay per hour and how many extra hours would you like?
- Marketing is trying to get Golf-Sport to consider an advertising program that promises a 50% increase in their maximum demand. Can we handle this with the current system or do we need more resources? How much more is the production going to cost if we take on the additional demand?

The solution will presented as a linear programming form with a single objective function to be maximized the profit and constraints based on the production process ,inventory balanced mentioned in the description of the problem. The recommendation will addressed by sensitivity-analysis after solving the linear programming problem.

2 Formulation & Model

2.1 Observation of Problem

In this problem, the decision variables can be considered into the following parts: production factory and period.

- PRODUCTION = { Basic Components, Assembled Sets } Where
  Basic Components include:

Product
Steel Shafts
Graphite Shafts
Forged Iron Heads
Metal Wood Heads
Titanium Insert Wood Heads

  And the Basic Components can be further descried as:
  Basic Components = { shafts,iron-heads, wood-heads }

  Assembled Sets include:

Product
Steel Shafts
Set, Steel Shafts, Metal Heads
Set, Steel Shafts, Insert Heads
Set, Graphite Shafts, Metal Heads
Set, Graphite Shafts, Insert Heads

- FACTORY = { Chandler,Glendale,Tuson }
- PERIOD = { 0,1,2 }

  There are three specific aspects that the production should be took into account:

  - Manufacture: $M = m_{ijt}$
  - Sale :$S = x_{ijt}$
  - Inventory: $I = v_{ijt}$

  Where

  $i \in PRODUCTION, j \in FACTORY, t \in PERIOD$

Besides decision variables, there are two types of constraints in this problem: local and global. Local constraints confined within identical production manufactured at certain factory in particular period(e.g. the minimum and maximum amounts of demand for each plant-product pair) while global constraints sets the whole limitation in at least one of the three dimensions: production, factory and period(e.g. graphite and advertisement control).

2.2 Variables

Based on the observation, we can formulate the variables in the model.

Variable	Description	Type
m_{ijt}	Product *i* manufactured in plant *j* at month *t*	number
x_{ijt}	Product *i* sold in plant *j* at month *t*	number
v_{ijt}	Inventory *i* remained in plant *j* at month *t*	number
M_t	Manufacture Matrix at month*t*	matrix
B_t	Basic components manufacture Matrix at month*t*	matrix
S_t	Assemble components manufacture Matrix at month*t*	matrix
X_t	Sale Matrix at month*t*	matrix
I_t	Inventory Matrix at month*t*	matrix
$Basic$	Basic Components = $\{ s, g, f, m, t \}$	enumerate set
$Assemble$	Assembled Sets = $\{ ssm, ssi, sgm, sgi \}$	enumerate set
shafts	shafts = $\{ s, g \}$	enumerate set
iron-heads	iron-head = $\{ f \}$	enumerate set
wood-heads	wood-head = $\{ m, t \}$	enumerate set
factory	factory = {Chandler,Glendale,Tuson }	enumerate set

Where $i \in \{s, g, f, m, t, ssm, ssi, sgm, sgi\}$ $j \in \{Chandler, Glendale, Tuson\}$ $t = 0, 1, 2$

2.3 Objective Function

The objective of this model is to maximize profit, represented as the difference between sales revenue and cost. For the purposes of this function let *R, H, C* and *A* represent revenue, product cost, and inventory cost, and advertisement cost respectively. The function can be expressed as:

$$\max Z = Revenue - Inventory \quad cost - Production \quad cost - Advertisement$$

$$= \sum_{t=1}^{2} \text{tr}(R^{\text{T}} X_t) - \sum_{t=1}^{2} \text{tr}(H_t^{\text{T}} I_t) - \sum_{t=1}^{2} \text{tr}(C_t^{\text{T}} M_t) - \sum_{t=1}^{2} \text{tr}(A^{\text{T}} B_t) \tag{1}$$

(Note: tr(M) is the trace of a matrix M)

The total revenue from all products produced and sold, minus the cost of manufacture, inventory and advertising is represented in this function.

Where: R is the matrix indicate revenue generated by each product.

$$\mathbf{R} = \begin{bmatrix} 10 & 10 & 12 \\ 25 & 25 & 30 \\ 8 & 8 & 10 \\ 18 & 18 & 22 \\ 40 & 40 & 45 \\ 290 & 290 & 310 \\ 380 & 380 & 420 \\ 560 & 560 & 640 \\ 650 & 650 & 720 \end{bmatrix}$$

C is the matrix indicate costs generated by manufacturing: $C = [C_1, C_1 \times 1.12]$

$$\mathbf{C_1} = \begin{bmatrix} 6 & 5 & 7 \\ 19 & 18 & 20 \\ 4 & 5 & 5 \\ 10 & 11 & 12 \\ 26 & 24 & 27 \\ 294 & 291 & 325 \\ 362 & 363 & 409 \\ 492 & 502 & 580 \\ 580 & 604 & 684 \end{bmatrix}$$

$$\mathbf{C_2} = \begin{bmatrix} 6.72 & 5.60 & 7.84 \\ 21.28 & 20.16 & 22.40 \\ 4.48 & 5.60 & 5.60 \\ 11.20 & 12.32 & 13.44 \\ 29.12 & 26.88 & 30.24 \\ 297.6 & 294.24 & 325.36 \\ 365.84 & 366.96 & 411.16 \\ 495.96 & 507.16 & 582.88 \\ 586.60 & 613.48 & 691.08 \end{bmatrix}$$

H is the matrix indicate costs generated by hiring warehouse for inventory.
$H_t = C_t \times 0.08$

$$\mathbf{H_1} = \begin{bmatrix} 0.48 & 0.40 & 0.56 \\ 1.52 & 1.44 & 1.60 \\ 0.32 & 0.40 & 0.40 \\ 0.80 & 0.88 & 0.96 \\ 2.08 & 1.92 & 2.16 \\ 2.40 & 2.16 & 0.24 \\ 2.56 & 2.64 & 1.44 \\ 2.64 & 3.44 & 1.92 \\ 4.40 & 6.32 & 4.72 \end{bmatrix}$$

$$\mathbf{H_2} = \begin{bmatrix} 0.5376 & 0.4480 & 0.6272 \\ 1.7024 & 1.6128 & 1.7920 \\ 0.3584 & 0.4480 & 0.4480 \\ 0.8960 & 0.9856 & 1.0752 \\ 2.3296 & 2.1504 & 2.4192 \\ 2.6880 & 2.4192 & 0.2688 \\ 2.8672 & 2.9568 & 1.6128 \\ 2.9568 & 3.8528 & 2.1504 \\ 4.9280 & 7.0784 & 5.2864 \end{bmatrix}$$

A is the matrix indicate investment in advertisements.

$$\mathbf{A} = \begin{bmatrix} 1.0 & 1.1 & 1.3 \\ 1.5 & 1.1 & 1.3 \\ 1.1 & 1.1 & 1.3 \\ 1.5 & 1.2 & 1.3 \\ 1.9 & 1.9 & 1.9 \end{bmatrix}$$

2.4 Constraints

2.4.1 Local constraints

Labor(workforce):

$$diag(W^{\mathrm{T}}B) \leq W_c \tag{2}$$

where:

$$\mathbf{W} = \begin{bmatrix} 1.0 & 3.5 & 3.0 \\ 1.5 & 3.5 & 3.5 \\ 1.5 & 4.5 & 4.0 \\ 3.0 & 4.5 & 4.5 \\ 4.0 & 5.0 & 5.5 \end{bmatrix}$$

$$\mathbf{W_c} = [12,000 \quad 15,000 \quad 22,000]$$

Packing machine:

$$diag(P^{\mathrm{T}}B) \leq P_c \tag{3}$$

Where:

$$\mathbf{P} = \begin{bmatrix} 4.0 & 7.0 & 7.5 \\ 4.0 & 7.0 & 7.5 \\ 5.0 & 8.0 & 8.5 \\ 6.0 & 9.0 & 9.5 \\ 6.0 & 7.0 & 8.0 \end{bmatrix}$$

$$\mathbf{P_c} = [20,000 \quad 40,000 \quad 35,000]$$

Assembly time:

$$T^{\mathrm{T}} \sum_{j \in factory} S_j \leq T_c \tag{4}$$

Where:

$$\mathbf{T} = [65 \quad 60 \quad 65]$$

$$\mathbf{T_c} = [5,500 \quad 5,000 \quad 6,000]$$

Demand:

$$L \leq M \leq U \tag{5}$$

Where:

$$\mathbf{L} = \begin{bmatrix} 0 & 0 & 0 \\ 100 & 100 & 50 \\ 200 & 200 & 100 \\ 30 & 30 & 15 \\ 100 & 100 & 100 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{U} = \begin{bmatrix} 2000 & 2000 & 2000 \\ 2000 & 2000 & 2000 \\ 2000 & 2000 & 2000 \\ 2000 & 2000 & 2000 \\ 2000 & 2000 & 2000 \\ 200 & 200 & 200 \\ 100 & 100 & 100 \\ 300 & 300 & 300 \\ 400 & 400 & 400 \end{bmatrix}$$

Production process and inventory constraints:

To interpret "Each set requires 13 shafts, 10 iron heads, and 3 wood heads. All of the shafts in a set must be the same type (steel or graphite), and all of the wood heads must be the same type (metal or metal with inserts)." in a mathematical way, we have:

$$13sht + 10iron + 3wood = set$$
$$(sht \in shafts, iron \in iron - heads, wood \in wood - heads, set \in Assemble)$$

i.e.

$$F_1 M_t + F_2 S_t + F_3 I_{t-1} + F_4 I_t = 0 \tag{6}$$

To be specific, for every $j \in \{Chandler, Glendale, Tuson\}\, t = 1, 2$

$$13m_{ssm,j,t} + 13m_{ssi,j,t} + x_{s,j,t} + v_{s,j,t} = m_{s,j,t} + v_{s,j,t-1}$$

$$13m_{sgm,j,t} + 13m_{sgi,j,t} + x_{g,j,t} + v_{g,j,t} = m_{g,j,t} + v_{g,j,t-1}$$

$$10 \sum_{i \in Assembly} m_{i,j,t} + x_{f,j,t} + v_{f,j,t} = m_{f,j,t} + v_{f,j,t-1}$$

$$3m_{sgm,j,t} + 3m_{ssi,j,t} + x_{m,j,t} + v_{m,j,t} = m_{m,j,t} + v_{m,j,t-1}$$

$$3m_{ssm,j,t} + 3m_{ssi,j,t} + x_{t,j,t} + v_{t,j,t} = m_{t,j,t} + v_{t,j,t-1}$$

For every $i \in \{ssm, ssi, sgm, sgi\}$ $j \in \{Chandler, Glendale, Tuson\}\, t = 1, 2$

$$m_{i,j,t} + x_{i,j,t} - v_{i,j,t-1} + v_{i,j,t} = m_{i,j,t}$$

2.4.2 Global constraints

Advertising constraints:

$$\text{tr}(A^{\text{T}} B_t) \leq A_c \tag{7}$$
$$A_c = [20,000]$$

Graphit constraints:

$$\sum_{j \in factory} m_{g,j,t} \leq G_c \tag{8}$$
$$G_c = [4,000]$$

3 Implementation & Recommendation

3.1 Implementation & Solution

In order to encoding subscript i, j in a more effective way, we encode subscript i, j based on one-hot encoding, i.e. for index i:

$$s \to 1, g \to 2, f \to 3, m \to 4, t \to 5, ssm \to 6, ssi \to 7, sgm \to 8, sgi \to 9,$$

for index j:

$$Chandler \to 1, Glendale \to 2, Tuson \to 3,$$

For example, product *Steel Shafts* manufactured at Tuson in the second month is expressed as: $m_{1,3,2}$.With these tricks mentioned above, Problem was successfully coded in Lingo, yielding a maximized profit of $221659.9 for Golf-Sport as a company in the two month period.

$$\max Z = 221659.9$$

3.2 Plan for Production and Sales

The optimal decision variables indicate the plan for production and sales, which is showed in the following tables.

Optimal Chandler manufacture		
Item	**Month 1**	**Month 2**
Steel Shafts	0	0
Graphite Shafts	867.14	100
Iron Heads	200	200
Metal Wood Heads	588.57	1100
Titanium Insert Wood Heads	2000	2000
Set, Steel, Metal Heads	0	0
Set, Steel, Titanium Heads	0	0
Set, Graphite, Metal Heads	0	0
Set, Graphite, Titanium Heads	0	0

Optimal Chandler Sales		
Item	**Month 1**	**Month 2**
Steel Shafts	0	0
Graphite Shafts	867.14	100
Iron Heads	200	200
Metal Wood Heads	588.57	1100
Titanium Insert Wood Heads	2000	2000
Set, Steel, Metal Heads	0	0
Set, Steel, Titanium Heads	0	0
Set, Graphite, Metal Heads	0	0
Set, Graphite, Titanium Heads	0	0

Optimal Glendale manufacture		
Item	**Month 1**	**Month 2**
Steel Shafts	0	0
Graphite Shafts	1132.86	1132.86
Iron Heads	200	200
Metal Wood Heads	30	30
Titanium Insert Wood Heads	2000	2000
Set, Steel, Metal Heads	0	0
Set, Steel, Titanium Heads	0	0
Set, Graphite, Metal Heads	0	0
Set, Graphite, Titanium Heads	0	0

Optimal Glendale Sales		
Item	**Month 1**	**Month 2**
Steel Shafts	0	0
Graphite Shafts	1132.86	1132.86
Iron Heads	200	200
Metal Wood Heads	30	30
Titanium Insert Wood Heads	2000	2000
Set, Steel, Metal Heads	0	0
Set, Steel, Titanium Heads	0	0
Set, Graphite, Metal Heads	0	0
Set, Graphite, Titanium Heads	0	0

Optimal Tuson manufacture		
Item	**Month 1**	**Month 2**
Steel Shafts	0	0
Graphite Shafts	2000	2000
Iron Heads	100	100
Metal Wood Heads	331.58	331.58
Titanium Insert Wood Heads	2000	2000
Set, Steel, Metal Heads	0	0
Set, Steel, Titanium Heads	0	0
Set, Graphite, Metal Heads	0	0
Set, Graphite, Titanium Heads	0	0

Optimal Tuson Sales		
Item	**Month 1**	**Month 2**
Steel Shafts	0	0
Graphite Shafts	2000	2000
Iron Heads	100	100
Metal Wood Heads	331.58	331.58
Titanium Insert Wood Heads	2000	2000
Set, Steel, Metal Heads	0	0
Set, Steel, Titanium Heads	0	0
Set, Graphite, Metal Heads	0	0
Set, Graphite, Titanium Heads	0	0

3.3 Sensitive-analysis & Recommendation

After solving the linear programming problem, recommendations are made towards the issues proposed in the description of problem based on the solution report.

- **issue:** If you could get more graphite or advertising cash, how much would you like, how would you use it, and what would you be willing to pay?
- **recommendation:** We selected several essential indexes from lingo lindo's default solution and range reports dual price and allowable range to further analysis. The shadow price and surplus of advertising and graphite is 0.67 and 0 respectively, suggesting investing 209.46 in graphite in the first month, yielding $ 213,544.6 profit.

range reports					
month	**Item**	**RHS**	**Allowable increase**	**Allowable decrease**	**Dule price**
Month 1	Graphite	1000	209.4643	191.7857	0.67
Month 2	Graphite	1000	INF	191.7857	0
Month 1	Advertising	20,000	INF	1533.233	0
Month 2	Advertising	20,000	INF	1916.855	0

- **issue:** At what site(s) would you like to add extra packing machine hours, assembly hours, and/or extra labor hours? How much would you be willing to pay per hour and how many extra hours would you like?

- **recommendation:** According to Righthand Side Ranges of labor and packaging constraints showed above, adding extra labor time doesn't work for factory Chandler and Tucson whereas increasing profit for factory Glendale at $1.64 in the first month and $1.069 the second month per minute yielding profit $4,403 and $2,870, respectively.

  When it comes to extra packing and assembly time, report showed extra packing time increasing$1.083 and$0.916 in the first month for Chandler and Tucson respectively,$0.883 and $0.764 for the second. We notice that extra assembly time make no difference for any of three factory in any period.

  In conclusion, we suggest adding extra labor and packing time as long as it is not greater than its corresponding shadow price.

Labor					
month	**Factory**	**RHS**	**Allowable increase**	**Allowable decrease**	**Dule price**
Month 1	Chandler	12,000	INF	633.57	0
Month 1	Glendale	15,000	2,685	2,932.5	1.64
Month 1	Tucson	22,000	INF	2,107.895	0
Month 2	Chandler	12,000	INF	250	0
Month 2	Glendale	15,000	2,685	3,615	1.069
Month 2	Tucson	22,000	INF	2,107.9	0

Packing					
month	**Factory**	**RHS**	**Allowable increase**	**Allowable decrease**	**Dule price**
Month 1	Chandler	20,000	1,267.14	3,351.43	1.083
Month 1	Glendale	40,000	INF	16,200	0
Month 1	Tucson	35,000	4,450	3,007.5	0.916
Month 2	Chandler	20,000	500	6,420	0.883
Month 2	Glendale	40,000	INF	16,200	0
Month 2	Tucson	35,000	4,450	3,007.5	0.764

Assembly					
month	**Factory**	**RHS**	**Allowable increase**	**Allowable decrease**	**Dule price**
Month 1	Chandler	5,500	INF	5,500	0
Month 1	Glendale	5,000	INF	5,000	0
Month 1	Tucson	6,000	INF	6,000	0
Month 2	Chandler	5,500	INF	5,500	0
Month 2	Glendale	5,000	INF	5,000	0
Month 2	Tucson	6,000	INF	6,000	0

- **issue:** Marketing is trying to get Golf-Sport to consider an advertising program that promises a 50% increase in their maximum demand. Can we handle this with the current system or do

we need more resources? How much more is the production going to cost if we take on the additional demand?

- **recommendation:**When we increase the maximum demand limit by 50% in our model i.e.

$$\mathbf{U}' = \begin{bmatrix} 3000 & 3000 & 3000 \\ 3000 & 3000 & 3000 \\ 3000 & 3000 & 3000 \\ 3000 & 3000 & 3000 \\ 3000 & 3000 & 3000 \\ 300 & 300 & 300 \\ 150 & 150 & 150 \\ 450 & 450 & 450 \\ 600 & 600 & 600 \end{bmatrix}$$

which exceeds the allowable change causing optimal value Z' comes to \$241,175 suggesting that more resource is needed. We consider the production cost no more than \$27,770 in total.

4 Contributions

Yu

- Modeling.
- Writing report.

Jin

- Checking correctness of model.
- Programming.
- Sensitivity Analysis.

5 Appendix

Listing 1: Lingo code

```
model:

  sets:

    month/1,2/;
    factory/1,2,3/:labor_limit, packing_limit, assembly_limit;
    production/1, 2, 3, 4, 5, 6, 7, 8, 9/;
    pairi(month,factory);
    pair(factory,production):price, inventory_cost,
       advertise_cost, labor_time, packing_time, assembly_time;
    pairs(month,factory,production):sold, manufacture, inventory,
        cost;

  endsets

  !Objective function;

  max =
  @sum(pairs(i,j,k): sold(i,j,k)*price(j,k)) -
  @sum(pairs(i,j,k): inventory_cost(i,j,k)*inventory(i,j,k)) -
  @sum(pairs(i,j,k): cost(i,j,k)*sold(i,j,k)) -
  @sum(pairs(i,j,k)|k#le#5:advertise_cost(j,k)*manufacture(i,j,k)
     );


  !Labor;
  @for(pairi(i,j):@sum(pair(j,k)|k#le#5:labor_time(j,k)*
     manufacture(i,j,k)) <= labor_limit(j));
  !Packing;
  @for(pairi(i,j):@sum(pair(j,k)|k#le#5:packing_time(j,k)*
     manufacture(i,j,k)) <= packing_limit(j));
  !Assembly
  @for(pairi(i,j):@sum(pair(j,k)|k#gt#5:assembly_time(j,k)*
     manufacture(i,j,k)) <= assembly_limit(j));
  !Graphit;
  @for(pairi(i,j):@sum(pair(j,k)|k#eq#2:manufacture(i,j,k)) <=
     4000);
  !Advertisement;
  @for(pairi(i,j):@sum(pair(j,k)|k#le#5:advertise_cost(j,k)*
     manufacture(i,j,k)) <= 20000);
```

```
!productionuction process and inventoryentory balanced
   constraints;

manufacture(1,1,1)=inventory(1,1,1)+sold(1,1,1)+manufacture
   (1,1,6)*13+manufacture(1,1,7)*13;
manufacture(1,1,2)=inventory(1,1,2)+sold(1,1,2)+manufacture
   (1,1,8)*13+manufacture(1,1,8)*13;
manufacture(1,1,3)=inventory(1,1,3)+sold(1,1,3)+manufacture
   (1,1,6)*10+manufacture(1,1,7)*10+manufacture(1,1,8)*10+
   manufacture(1,1,9)*10;
manufacture(1,1,4)=inventory(1,1,4)+sold(1,1,4)+manufacture
   (1,1,6)*3+manufacture(1,1,8)*3;
manufacture(1,1,5)=inventory(1,1,5)+sold(1,1,5)+manufacture
   (1,1,7)*3+manufacture(1,1,9)*3;

manufacture(1,1,6)=sold(1,1,6)+inventory(1,1,6);
manufacture(1,1,7)=sold(1,1,7)+inventory(1,1,7);
manufacture(1,1,8)=sold(1,1,8)+inventory(1,1,8);
manufacture(1,1,9)=sold(1,1,9)+inventory(1,1,9);

manufacture(1,2,1)=inventory(1,2,1)+sold(1,2,1)+manufacture
   (1,2,6)*13+manufacture(1,2,7)*13;
manufacture(1,2,2)=inventory(1,2,2)+sold(1,2,2)+manufacture
   (1,2,8)*13+manufacture(1,2,8)*13;
manufacture(1,2,3)=inventory(1,2,3)+sold(1,2,3)+manufacture
   (1,2,6)*10+manufacture(1,2,7)*10+manufacture(1,2,8)*10+
   manufacture(1,2,9)*10;
manufacture(1,2,4)=inventory(1,2,4)+sold(1,2,4)+manufacture
   (1,2,6)*3+manufacture(1,2,8)*3;
manufacture(1,2,5)=inventory(1,2,5)+sold(1,2,5)+manufacture
   (1,2,7)*3+manufacture(1,2,9)*3;

manufacture(1,2,6)=sold(1,2,6)+inventory(1,2,6);
manufacture(1,2,7)=sold(1,2,7)+inventory(1,2,7);
manufacture(1,2,8)=sold(1,2,8)+inventory(1,2,8);
manufacture(1,2,9)=sold(1,2,9)+inventory(1,2,9);

manufacture(1,3,1)=inventory(1,3,1)+sold(1,3,1)+manufacture
   (1,3,6)*13+manufacture(1,3,7)*13;
manufacture(1,3,2)=inventory(1,3,2)+sold(1,3,2)+manufacture
   (1,3,8)*13+manufacture(1,3,8)*13;
manufacture(1,3,3)=inventory(1,3,3)+sold(1,3,3)+manufacture
   (1,3,6)*10+manufacture(1,3,7)*10+manufacture(1,3,8)*10+
   manufacture(1,3,9)*10;
```

```
manufacture(1,3,4)=inventory(1,3,4)+sold(1,3,4)+manufacture
   (1,3,6)*3+manufacture(1,3,8)*3;
manufacture(1,3,5)=inventory(1,3,5)+sold(1,3,5)+manufacture
   (1,3,7)*3+manufacture(1,3,9)*3;

manufacture(1,3,6)=sold(1,3,6)+inventory(1,3,6);
manufacture(1,3,7)=sold(1,3,7)+inventory(1,3,7);
manufacture(1,3,8)=sold(1,3,8)+inventory(1,3,8);
manufacture(1,3,9)=sold(1,3,9)+inventory(1,3,9);

inventory(2,1,1)=inventory(1,1,1)+manufacture(2,1,1)-
   manufacture(2,1,6)*13-manufacture(2,1,7)*13-sold(2,1,1);
inventory(2,1,2)=inventory(1,1,2)+manufacture(2,1,2)-
   manufacture(2,1,8)*13-manufacture(2,1,9)*13-sold(2,1,2);
inventory(2,1,3)=inventory(1,1,3)+manufacture(2,1,3)-
   manufacture(2,1,6)*10-manufacture(2,1,7)*10-manufacture
   (2,1,8)*10-manufacture(2,1,9)*10-sold(2,1,3);
inventory(2,1,4)=inventory(1,1,4)+manufacture(2,1,4)-
   manufacture(2,1,6)*3-manufacture(2,1,8)*3-sold(2,1,4);
inventory(2,1,5)=inventory(1,1,5)+manufacture(2,1,5)-
   manufacture(2,1,7)*3-manufacture(2,1,9)*3-sold(2,1,5);

inventory(2,1,6)=inventory(1,1,6)+manufacture(2,1,6)-sold
   (2,1,6);
inventory(2,1,7)=inventory(1,1,7)+manufacture(2,1,7)-sold
   (2,1,7);
inventory(2,1,8)=inventory(1,1,8)+manufacture(2,1,8)-sold
   (2,1,8);
inventory(2,1,9)=inventory(1,1,9)+manufacture(2,1,8)-sold
   (2,1,9);

inventory(2,2,1)=inventory(1,2,1)+manufacture(2,2,1)-
   manufacture(2,2,6)*13-manufacture(2,2,7)*13-sold(2,2,1);
inventory(2,2,2)=inventory(1,2,2)+manufacture(2,2,2)-
   manufacture(2,2,8)*13-manufacture(2,2,9)*13-sold(2,2,2);
inventory(2,2,3)=inventory(1,2,3)+manufacture(2,2,3)-
   manufacture(2,2,6)*10-manufacture(2,2,7)*10-manufacture
   (2,2,8)*10-manufacture(2,2,9)*10-sold(2,2,3);
inventory(2,2,4)=inventory(1,2,4)+manufacture(2,2,4)-
   manufacture(2,2,6)*3-manufacture(2,2,8)*3-sold(2,2,4);
inventory(2,2,5)=inventory(1,2,5)+manufacture(2,2,5)-
   manufacture(2,2,7)*3-manufacture(2,2,9)*3-sold(2,2,5);

inventory(2,2,6)=inventory(1,2,6)+manufacture(2,2,6)-sold
   (2,2,6);
```

```
inventory(2,2,7)=inventory(1,2,7)+manufacture(2,2,7)-sold
   (2,2,7);
inventory(2,2,8)=inventory(1,2,8)+manufacture(2,2,8)-sold
   (2,2,8);
inventory(2,2,9)=inventory(1,2,9)+manufacture(2,2,8)-sold
   (2,2,9);

inventory(2,3,1)=inventory(1,3,1)+manufacture(2,3,1)-
   manufacture(2,3,6)*13-manufacture(2,3,7)*13-sold(2,3,1);
inventory(2,3,2)=inventory(1,3,2)+manufacture(2,3,2)-
   manufacture(2,3,8)*13-manufacture(2,3,9)*13-sold(2,3,2);
inventory(2,3,3)=inventory(1,3,3)+manufacture(2,3,3)-
   manufacture(2,3,6)*10-manufacture(2,3,7)*10-manufacture
   (2,3,8)*10-manufacture(2,3,9)*10-sold(2,3,3);
inventory(2,3,4)=inventory(1,3,4)+manufacture(2,3,4)-
   manufacture(2,3,6)*3-manufacture(2,3,8)*3-sold(2,3,4);
inventory(2,3,5)=inventory(1,3,5)+manufacture(2,3,5)-
   manufacture(2,3,7)*3-manufacture(2,3,9)*3-sold(2,3,5);

inventory(2,3,6)=inventory(1,3,6)+manufacture(2,3,6)-sold
   (2,3,6);
inventory(2,3,7)=inventory(1,3,7)+manufacture(2,3,7)-sold
   (2,3,7);
inventory(2,3,8)=inventory(1,3,8)+manufacture(2,3,8)-sold
   (2,3,8);
inventory(2,3,9)=inventory(1,3,9)+manufacture(2,3,8)-sold
   (2,3,9);

! Demand;

@bnd(0,sold(1,1,1),2000);
@bnd(100,sold(1,1,2),2000);
@bnd(200,sold(1,1,3),2000);
@bnd(30,sold(1,1,4),2000);
@bnd(100,sold(1,1,5),2000);
@bnd(0,sold(1,1,6),200);
@bnd(0,sold(1,1,7),100);
@bnd(0,sold(1,1,8),300);
@bnd(0,sold(1,1,9),400);

@bnd(0,sold(2,1,1),2000);
@bnd(100,sold(2,1,2),2000);
@bnd(200,sold(2,1,3),2000);
@bnd(30,sold(2,1,4),2000);
@bnd(100,sold(2,1,5),2000);
```

```
@bnd(0,sold(2,1,6),200);
@bnd(0,sold(2,1,7),100);
@bnd(0,sold(2,1,8),300);
@bnd(0,sold(2,1,9),400);

@bnd(0,sold(1,2,1),2000);
@bnd(100,sold(1,2,2),2000);
@bnd(200,sold(1,2,3),2000);
@bnd(30,sold(1,2,4),2000);
@bnd(100,sold(1,2,5),2000);
@bnd(0,sold(1,2,6),200);
@bnd(0,sold(1,2,7),100);
@bnd(0,sold(1,2,8),300);
@bnd(0,sold(1,2,9),400);

@bnd(0,sold(2,2,1),2000);
@bnd(100,sold(2,2,2),2000);
@bnd(200,sold(2,2,3),2000);
@bnd(30,sold(2,2,4),2000);
@bnd(100,sold(2,2,5),2000);
@bnd(0,sold(2,2,6),200);
@bnd(0,sold(2,2,7),100);
@bnd(0,sold(2,2,8),300);
@bnd(0,sold(2,2,9),400);

@bnd(0,sold(1,3,1),2000);
@bnd(50,sold(1,3,2),2000);
@bnd(100,sold(1,3,3),2000);
@bnd(15,sold(1,3,4),2000);
@bnd(100,sold(1,3,5),2000);
@bnd(0,sold(1,3,6),200);
@bnd(0,sold(1,3,7),100);
@bnd(0,sold(1,3,8),300);
@bnd(0,sold(1,3,9),400);

@bnd(0,sold(2,3,1),2000);
@bnd(50,sold(2,3,2),2000);
@bnd(100,sold(2,3,3),2000);
@bnd(15,sold(2,3,4),2000);
@bnd(100,sold(2,3,5),2000);
@bnd(0,sold(2,3,6),200);
@bnd(0,sold(2,3,7),100);
@bnd(0,sold(2,3,8),300);
@bnd(0,sold(2,3,9),400);
```

```
data:

   price =
   10, 25, 8, 18, 40, 290, 380, 560, 650,
   10, 25, 8, 18, 40, 290, 380, 560, 650,
   12, 30, 10, 22, 45, 310, 420, 640, 720;

   cost =
   6, 19, 4, 10, 26, 30, 32, 33, 55,
   5, 18, 5, 11, 24, 27, 33, 43, 79,
   7, 20, 5, 12, 27, 3, 18, 24, 59,
   6.72, 21.28, 4.48, 11.2, 29.12, 33.60, 35.84, 36.96, 61.6,
   5.6, 20.16, 5.6, 12.32, 26.88, 30.24, 36.96, 48.16, 88.48,
   7.84, 22.4, 5.6, 13.44, 30.24, 3.36, 20.16, 26.88, 66.08;

   inventory_cost =
   0.48, 1.52, 0.32, 0.80, 2.08, 2.40, 2.56, 2.64, 4.40,
   0.40, 1.44, 0.40, 0.88, 1.92, 2.16, 2.64, 3.44, 6.32,
   0.56, 1.60, 0.40, 0.96, 2.16, 0.24, 1.44, 1.92, 4.72,
   0.5376, 1.7024, 0.3584, 0.8960, 2.3296, 2.6880, 2.8672,
      2.9568, 4.9280,
   0.4480, 1.6128, 0.4480, 0.9856, 2.1504, 2.4192, 2.9568,
      3.8528, 7.0784,
   0.6272, 1.7920, 0.4480, 1.0752, 2.4192, 0.2688, 1.6128,
      2.1504, 5.2864;

   advertise_cost =
   1.0, 1.5, 1.1, 1.5, 1.9, , , , ,
   1.1, 1.1, 1.1, 1.2, 1.9, , , , ,
   1.3, 1.3, 1.3, 1.3, 1.9, , , , ;

   labor_time =
   1.0, 1.5, 1.5, 3.0, 4.0, , , , ,
   3.5, 3.5, 4.5, 4.5, 5.0, , , , ,
   3.0, 3.5, 4.0, 4.5, 5.5, , , , ;

   labor_limit = 12000, 15000, 22000;

   packing_time =
   4, 4, 5, 6, 6, , , , ,
   7, 7, 8, 9, 7, , , , ,
   7.5, 7.5, 8.5, 9.5, 8, , , , ;

   packing_limit = 20000, 40000, 35000;
```

```
    assembly_time =
    , , , , , 65, 65, 65, 65,
    , , , , , 60, 60, 60, 60,
    , , , , , 65, 65, 65, 65;

    assembly_limit = 5500, 5000, 6000;

  enddata

end

\end{lstlisting}
```

Ranges in which the basis is unchanged:

Objective Coefficient Ranges:

Variable	Current Coefficient	Allowable Increase	Allowable Decrease
ADVCO(1, 6)	0.000000	0.000000	INFINITY
ADVCO(1, 7)	0.000000	0.000000	INFINITY
ADVCO(1, 8)	0.000000	0.000000	INFINITY
ADVCO(1, 9)	0.000000	0.000000	INFINITY
ADVCO(2, 6)	0.000000	0.000000	INFINITY
ADVCO(2, 7)	0.000000	0.000000	INFINITY
ADVCO(2, 8)	0.000000	0.000000	INFINITY
ADVCO(2, 9)	0.000000	0.000000	INFINITY
ADVCO(3, 6)	0.000000	0.000000	INFINITY
ADVCO(3, 7)	0.000000	0.000000	INFINITY
ADVCO(3, 8)	0.000000	0.000000	INFINITY
ADVCO(3, 9)	0.000000	0.000000	INFINITY
LABTI(1, 6)	0.000000	0.000000	INFINITY
LABTI(1, 7)	0.000000	0.000000	INFINITY
LABTI(1, 8)	0.000000	0.000000	INFINITY
LABTI(1, 9)	0.000000	0.000000	INFINITY
LABTI(2, 6)	0.000000	0.000000	INFINITY
LABTI(2, 7)	0.000000	0.000000	INFINITY
LABTI(2, 8)	0.000000	0.000000	INFINITY
LABTI(2, 9)	0.000000	0.000000	INFINITY
LABTI(3, 6)	0.000000	0.000000	INFINITY
LABTI(3, 7)	0.000000	0.000000	INFINITY
LABTI(3, 8)	0.000000	0.000000	INFINITY
LABTI(3, 9)	0.000000	0.000000	INFINITY
PAKTI(1, 6)	0.000000	0.000000	INFINITY
PAKTI(1, 7)	0.000000	0.000000	INFINITY
PAKTI(1, 8)	0.000000	0.000000	INFINITY
PAKTI(1, 9)	0.000000	0.000000	INFINITY
PAKTI(2, 6)	0.000000	0.000000	INFINITY
PAKTI(2, 7)	0.000000	0.000000	INFINITY
PAKTI(2, 8)	0.000000	0.000000	INFINITY
PAKTI(2, 9)	0.000000	0.000000	INFINITY
PAKTI(3, 6)	0.000000	0.000000	INFINITY
PAKTI(3, 7)	0.000000	0.000000	INFINITY
PAKTI(3, 8)	0.000000	0.000000	INFINITY
PAKTI(3, 9)	0.000000	0.000000	INFINITY
ASSTI(1, 1)	0.000000	0.000000	INFINITY
ASSTI(1, 2)	0.000000	0.000000	INFINITY
ASSTI(1, 3)	0.000000	0.000000	INFINITY
ASSTI(1, 4)	0.000000	0.000000	INFINITY
ASSTI(1, 5)	0.000000	0.000000	INFINITY
ASSTI(2, 1)	0.000000	0.000000	INFINITY
ASSTI(2, 2)	0.000000	0.000000	INFINITY
ASSTI(2, 3)	0.000000	0.000000	INFINITY
ASSTI(2, 4)	0.000000	0.000000	INFINITY
ASSTI(2, 5)	0.000000	0.000000	INFINITY
ASSTI(3, 1)	0.000000	0.000000	INFINITY
ASSTI(3, 2)	0.000000	0.000000	INFINITY
ASSTI(3, 3)	0.000000	0.000000	INFINITY

ASSTI(3, 4)	0.000000	0.000000	INFINITY
ASSTI(3, 5)	0.000000	0.000000	INFINITY
SOLD(1, 1, 1)	4.000000	1.333333	INFINITY
SOLD(1, 1, 2)	6.000000	1.222222	0.1666667
SOLD(1, 1, 3)	4.000000	2.516667	INFINITY
SOLD(1, 1, 4)	8.000000	0.2500000	1.571176
SOLD(1, 1, 5)	14.00000	INFINITY	5.600000
SOLD(1, 1, 6)	-4.000000	162.5000	INFINITY
SOLD(1, 1, 7)	18.00000	141.7000	INFINITY
SOLD(1, 1, 8)	68.00000	177.1667	INFINITY
SOLD(1, 1, 9)	70.00000	20.36667	INFINITY
SOLD(1, 2, 1)	5.000000	1.833333	INFINITY
SOLD(1, 2, 2)	7.000000	4.136667	1.222222
SOLD(1, 2, 3)	3.000000	5.471429	INFINITY
SOLD(1, 2, 4)	7.000000	1.571429	INFINITY
SOLD(1, 2, 5)	16.00000	INFINITY	5.909524
SOLD(1, 2, 6)	-1.000000	200.2619	INFINITY
SOLD(1, 2, 7)	17.00000	186.8190	INFINITY
SOLD(1, 2, 8)	58.00000	234.4286	INFINITY
SOLD(1, 2, 9)	46.00000	68.98571	INFINITY
SOLD(1, 3, 1)	5.000000	3.168421	INFINITY
SOLD(1, 3, 2)	10.00000	INFINITY	1.664912
SOLD(1, 3, 3)	5.000000	4.084211	INFINITY
SOLD(1, 3, 4)	10.00000	2.108889	1.887059
SOLD(1, 3, 5)	18.00000	INFINITY	8.773684
SOLD(1, 3, 6)	-15.00000	14.49120	INFINITY
SOLD(1, 3, 7)	11.00000	213.7105	INFINITY
SOLD(1, 3, 8)	60.00000	277.5544	INFINITY
SOLD(1, 3, 9)	36.00000	82.52105	INFINITY
SOLD(2, 1, 1)	3.280000	1.253333	INFINITY
SOLD(2, 1, 2)	3.720000	1.313333	INFINITY
SOLD(2, 1, 3)	3.520000	1.996667	INFINITY
SOLD(2, 1, 4)	6.800000	0.8903333	0.6780000
SOLD(2, 1, 5)	10.88000	INFINITY	3.680000
SOLD(2, 1, 6)	-7.600000	4.912000	INFINITY
SOLD(2, 1, 7)	14.16000	121.5400	17.02720
SOLD(2, 1, 8)	64.04000	13.56000	17.80667
SOLD(2, 1, 9)	63.40000	13.56000	INFINITY
SOLD(2, 2, 1)	4.400000	0.4400000	INFINITY
SOLD(2, 2, 2)	4.840000	2.304444	0.2555556
SOLD(2, 2, 3)	2.400000	3.508571	INFINITY
SOLD(2, 2, 4)	5.680000	0.3285714	INFINITY
SOLD(2, 2, 5)	13.12000	INFINITY	5.877143
SOLD(2, 2, 6)	-4.240000	1.820800	INFINITY
SOLD(2, 2, 7)	13.04000	130.6943	15.99680
SOLD(2, 2, 8)	52.84000	50.67143	INFINITY
SOLD(2, 2, 9)	36.52000	50.67143	43.59840
SOLD(2, 3, 1)	4.160000	2.871579	INFINITY
SOLD(2, 3, 2)	7.600000	INFINITY	0.5684211
SOLD(2, 3, 3)	4.400000	3.395789	INFINITY
SOLD(2, 3, 4)	8.560000	0.7200000	3.637333
SOLD(2, 3, 5)	14.76000	INFINITY	6.746316
SOLD(2, 3, 6)	-15.36000	15.09120	INFINITY
SOLD(2, 3, 7)	8.840000	184.5695	10.45280
SOLD(2, 3, 8)	57.12000	109.0084	INFINITY
SOLD(2, 3, 9)	28.92000	94.32105	34.20640

PRODC(1, 1, 1)	-1.000000	1.280000	INFINITY
PRODC(1, 1, 2)	-1.500000	1.222222	0.1666667
PRODC(1, 1, 3)	-1.100000	1.320000	INFINITY
PRODC(1, 1, 4)	-1.500000	0.2500000	1.571176
PRODC(1, 1, 5)	-1.900000	3.280000	5.600000
PRODC(1, 1, 6)	0.000000	162.5000	INFINITY
PRODC(1, 1, 7)	0.000000	141.7000	INFINITY
PRODC(1, 1, 8)	0.000000	177.1667	INFINITY
PRODC(1, 1, 9)	0.000000	17.80667	INFINITY
PRODC(1, 2, 1)	-1.100000	1.833333	INFINITY
PRODC(1, 2, 2)	-1.100000	4.136667	1.222222
PRODC(1, 2, 3)	-1.100000	2.962857	INFINITY
PRODC(1, 2, 4)	-1.200000	1.571429	INFINITY
PRODC(1, 2, 5)	-1.900000	4.767619	5.909524
PRODC(1, 2, 6)	0.000000	200.2619	INFINITY
PRODC(1, 2, 7)	0.000000	186.8190	INFINITY
PRODC(1, 2, 8)	0.000000	192.3571	INFINITY
PRODC(1, 2, 9)	0.000000	68.98571	INFINITY
PRODC(1, 3, 1)	-1.300000	1.696842	INFINITY
PRODC(1, 3, 2)	-1.300000	2.903509	1.664912
PRODC(1, 3, 3)	-1.300000	1.688421	INFINITY
PRODC(1, 3, 4)	-1.300000	2.108889	1.887059
PRODC(1, 3, 5)	-1.900000	3.372632	8.773684
PRODC(1, 3, 6)	0.000000	227.5404	INFINITY
PRODC(1, 3, 7)	0.000000	213.7105	INFINITY
PRODC(1, 3, 8)	0.000000	173.3460	INFINITY
PRODC(1, 3, 9)	0.000000	82.52105	INFINITY
PRODC(2, 1, 1)	-1.000000	1.253333	1.280000
PRODC(2, 1, 2)	-1.500000	1.043077	1.369744
PRODC(2, 1, 3)	-1.100000	1.356000	1.320000
PRODC(2, 1, 4)	-1.500000	1.047451	0.7976471
PRODC(2, 1, 5)	-1.900000	9.529600	3.280000
PRODC(2, 1, 6)	0.000000	137.1880	INFINITY
PRODC(2, 1, 7)	0.000000	121.5400	INFINITY
PRODC(2, 1, 8)	0.000000	13.56000	17.80667
PRODC(2, 1, 9)	0.000000	142.2000	INFINITY
PRODC(2, 2, 1)	-1.100000	0.4400000	2.393333
PRODC(2, 2, 2)	-1.100000	2.304444	0.2555556
PRODC(2, 2, 3)	-1.100000	3.508571	2.962857
PRODC(2, 2, 4)	-1.200000	0.3285714	3.442857
PRODC(2, 2, 5)	-1.900000	9.393257	4.767619
PRODC(2, 2, 6)	0.000000	142.4506	INFINITY
PRODC(2, 2, 7)	0.000000	130.6943	INFINITY
PRODC(2, 2, 8)	0.000000	50.67143	192.3571
PRODC(2, 2, 9)	0.000000	143.7343	INFINITY
PRODC(2, 3, 1)	-1.300000	2.871579	1.696842
PRODC(2, 3, 2)	-1.300000	8.385263	0.5684211
PRODC(2, 3, 3)	-1.300000	3.395789	1.688421
PRODC(2, 3, 4)	-1.300000	0.7200000	3.637333
PRODC(2, 3, 5)	-1.900000	10.43288	3.372632
PRODC(2, 3, 6)	0.000000	195.3172	INFINITY
PRODC(2, 3, 7)	0.000000	184.5695	INFINITY
PRODC(2, 3, 8)	0.000000	109.0084	173.3460
PRODC(2, 3, 9)	0.000000	193.4095	INFINITY
INV(1, 1, 1)	-0.4800000	1.280000	INFINITY
INV(1, 1, 2)	-1.520000	2.486667	INFINITY

INV(1, 1, 3)	-0.3200000	1.320000	INFINITY
INV(1, 1, 4)	-0.8000000	2.000000	INFINITY
INV(1, 1, 5)	-2.080000	3.280000	INFINITY
INV(1, 1, 6)	-2.400000	163.5880	INFINITY
INV(1, 1, 7)	-2.560000	148.1000	INFINITY
INV(1, 1, 8)	-2.640000	183.7667	INFINITY
INV(1, 1, 9)	-4.400000	17.80667	INFINITY
INV(1, 2, 1)	-0.4000000	2.393333	INFINITY
INV(1, 2, 2)	-1.440000	3.600000	INFINITY
INV(1, 2, 3)	-0.4000000	2.962857	INFINITY
INV(1, 2, 4)	-0.8800000	3.442857	INFINITY
INV(1, 2, 5)	-1.920000	4.767619	INFINITY
INV(1, 2, 6)	-2.160000	203.8411	INFINITY
INV(1, 2, 7)	-2.640000	193.4190	INFINITY
INV(1, 2, 8)	-3.440000	192.3571	INFINITY
INV(1, 2, 9)	-6.320000	84.78571	INFINITY
INV(1, 3, 1)	-0.5600000	1.696842	INFINITY
INV(1, 3, 2)	-1.600000	2.903509	INFINITY
INV(1, 3, 3)	-0.4000000	1.688421	INFINITY
INV(1, 3, 4)	-0.9600000	2.400000	INFINITY
INV(1, 3, 5)	-2.160000	3.372632	INFINITY
INV(1, 3, 6)	-0.2400000	227.5404	14.49120
INV(1, 3, 7)	-1.440000	217.3105	INFINITY
INV(1, 3, 8)	-1.920000	173.3460	INFINITY
INV(1, 3, 9)	-4.720000	94.32105	INFINITY
INV(2, 1, 1)	-0.5376000	5.070933	INFINITY
INV(2, 1, 2)	-1.702400	6.735733	INFINITY
INV(2, 1, 3)	-0.3584000	5.875067	INFINITY
INV(2, 1, 4)	-0.8960000	7.696000	INFINITY
INV(2, 1, 5)	-2.329600	9.529600	INFINITY
INV(2, 1, 6)	-2.688000	137.1880	4.912000
INV(2, 1, 7)	-2.867200	17.02720	INFINITY
INV(2, 1, 8)	-2.956800	66.99680	INFINITY
INV(2, 1, 9)	-4.928000	81.88800	INFINITY
INV(2, 2, 1)	-0.4480000	5.288000	INFINITY
INV(2, 2, 2)	-1.612800	6.452800	INFINITY
INV(2, 2, 3)	-0.4480000	6.356571	INFINITY
INV(2, 2, 4)	-0.9856000	6.994171	INFINITY
INV(2, 2, 5)	-2.150400	9.393257	INFINITY
INV(2, 2, 6)	-2.419200	142.4506	1.820800
INV(2, 2, 7)	-2.956800	15.99680	INFINITY
INV(2, 2, 8)	-3.852800	107.3642	INFINITY
INV(2, 2, 9)	-7.078400	43.59840	INFINITY
INV(2, 3, 1)	-0.6272000	7.658779	INFINITY
INV(2, 3, 2)	-1.792000	8.823579	INFINITY
INV(2, 3, 3)	-0.4480000	8.243789	INFINITY
INV(2, 3, 4)	-1.075200	9.635200	INFINITY
INV(2, 3, 5)	-2.419200	10.43288	INFINITY
INV(2, 3, 6)	-0.2688000	195.3172	14.49120
INV(2, 3, 7)	-1.612800	10.45280	INFINITY
INV(2, 3, 8)	-2.150400	168.2788	INFINITY
INV(2, 3, 9)	-5.286400	34.20640	INFINITY

Righthand Side Ranges:

	Current	Allowable	Allowable

Row	RHS	Increase	Decrease
2	12000.00	INFINITY	633.5714
3	15000.00	2685.000	2932.500
4	22000.00	INFINITY	2107.895
5	12000.00	INFINITY	250.0000
6	15000.00	2685.000	3615.000
7	22000.00	INFINITY	2107.895
8	20000.00	1267.143	3351.429
9	40000.00	INFINITY	16200.00
10	35000.00	4450.000	3007.500
11	20000.00	500.0000	6420.000
12	40000.00	INFINITY	16200.00
13	35000.00	4450.000	3007.500
14	5500.000	INFINITY	5500.000
15	5000.000	INFINITY	5000.000
16	6000.000	INFINITY	6000.000
17	5500.000	INFINITY	5500.000
18	5000.000	INFINITY	5000.000
19	6000.000	INFINITY	6000.000
20	1000.000	209.4643	191.7857
21	1000.000	INFINITY	191.7857
22	20000.00	INFINITY	1533.233
23	20000.00	INFINITY	1916.805
24	0.000000	837.8571	0.000000
25	0.000000	767.1429	1132.857
26	0.000000	670.2857	200.0000
27	0.000000	558.5714	1411.429
28	0.000000	558.5714	1411.429
29	0.000000	27.92857	0.000000
30	0.000000	27.92857	0.000000
31	0.000000	29.50549	0.000000
32	0.000000	49.28571	0.000000
33	0.000000	837.8571	0.000000
34	0.000000	1032.857	867.1429
35	0.000000	651.6667	200.0000
36	0.000000	651.6667	30.00000
37	0.000000	586.5000	537.0000
38	0.000000	28.19712	0.000000
39	0.000000	27.79621	0.000000
40	0.000000	24.18060	0.000000
41	0.000000	48.87500	0.000000
42	0.000000	401.0000	0.000000
43	0.000000	401.0000	837.8571
44	0.000000	353.8235	100.0000
45	0.000000	316.5789	1668.421
46	0.000000	375.9375	1981.250
47	0.000000	0.000000	INFINITY
48	0.000000	14.56416	0.000000
49	0.000000	9.748784	0.000000
50	0.000000	27.59174	0.000000
51	0.000000	0.000000	1605.000
52	0.000000	100.0000	767.1429
53	0.000000	200.0000	1284.000
54	0.000000	900.0000	1070.000
55	0.000000	900.0000	250.0000
56	0.000000	INFINITY	0.000000

57	0.000000	100.0000	0.000000
58	0.000000	300.0000	0.000000
59	0.000000	0.000000	53.50000
60	0.000000	0.000000	1032.857
61	0.000000	867.1429	1032.857
62	0.000000	200.0000	803.3333
63	0.000000	30.00000	803.3333
64	0.000000	537.0000	723.0000
65	0.000000	INFINITY	0.000000
66	0.000000	100.0000	0.000000
67	0.000000	0.000000	34.75962
68	0.000000	400.0000	0.000000
69	0.000000	0.000000	401.0000
70	0.000000	2000.000	401.0000
71	0.000000	100.0000	353.8235
72	0.000000	1668.421	316.5789
73	0.000000	1981.250	375.9375
74	0.000000	INFINITY	0.000000
75	0.000000	100.0000	0.000000
76	0.000000	0.000000	14.25355
77	0.000000	400.0000	0.000000