CHANDPUR ENTERPRISES LIMITED
STEEL DIVISION

RAW MATERIALS REQUIREMENT
FOR AUGUST PRODUCTION

CONSULTING REPORT

ENGEL, LILY
LEE, MYUNGHWAN
LIU, ALICE
RICHARD, JESSICA

IEOR 162 – LINEAR PROGRAMMING AND NETWORK FLOWS

PROFESSOR DORIT S. HOCHBAUM

SPRING 2019
UNIVERSITY OF CALIFORNIA, BERKELEY

Table of Contents

Part I. Consulting Report

1.1 Executive Summary

Chandpur Enterprises Limited (CEL) is a diversified manufacturing company based in India that instituted its steel business in 2006. However, the division quickly suffered from various administrative decisions under the managing director, Akshay Mittal, that revolved around observing the most optimal utilization of raw materials for the August production. Steel can be created from any combinations, known as a "batch", of the seven raw materials in proper ratios of iron content: tasla, rangeen, sponge, local scrap, imported scrap, HC, and pig iron. But, material prices could fluctuate anywhere up to 20% in one month, so it is important to optimize each month. Other factors include rate/ton (in rupees), recovery percentage (weight of iron content over the total weight of raw material), batch limits, and monthly limits with a finished product size constraint of 4,000 kilograms (or 4 tons). The production of steel is also conditional on the total weight of raw material because both electricity (in kWh) and time (in hours) consumption costs are linearly dependent.

Linear programming (LP) is a mathematical modeling technique designed to help maximize monthly profits with respect to asset allocation through the maintenance of all these factors and additional variable costs, as well as the minimization of plant downtime. After a series of formulations, the consulting team can conclude both the optimal batch and a batch which will optimize monthly profits. The optimal batch has profit INR5421.07 with a monthly profit of INR1,460,640.995. The batch which optimizes monthly profits will bring in INR1,788,704.747 over the whole month. Suggestions for how to further increase profits include increasing regulatory size and increasing the amounts of certain preferred materials.

1.2 Results

There are two types of profit maximizations, which are to maximize the profit of producing only one batch and to maximize the monthly profit. These optimal solutions were solved using AMPL and the results are explained below.

If only one batch is produced, the best batch that will give optimal profit[1] is shown in Table 1 below. The profit associated with producing this batch is INR5421.07.

Table 1. Amount of Each Raw Material to Produce One Optimal Batch

Type of Raw Materials	Amount Used (Tons)
Tasla	1.39179
Rangeen	1.39179
Sponge	0.55672
Local Scrap	0.83507
Imported Scrap	0
HC	1.11343
Pig Iron	0.27836

From the results above, it can be concluded that not all raw materials have to be used, as the managing director did in the past. In this case of maximizing the profit of making only one batch, Imported Scrap is not used.

One of the constraints in finding the solution in Table 1 is given the regulation for the finished product to be less than 4 tons. However, based on the analysis we have done, the shadow price associated with this constraint is INR3395.27,[2] meaning that for every unit (1 ton) increase allowed of the finished product, the profit increases by INR3395.27 (Section 2.1.3). Thus, this right-hand-side (RHS) constraint hampers our ability to make more profit, and it is recommended for the director to seek regulatory approval to increase this limit.

In the past, the steel division of CEL had used all of the raw materials, but based on the results in Table 1, we can observe that to reach optimality, all raw materials should be used except for the Imported Scrap. The reduced cost of Imported Scrap was reported as -INR2931.38

[1] Task 1
[2] Task 2

per ton,[3] meaning that for every unit (1 kg) increase of Imported Scrap used, the profit decreases by INR2.93138.[4] Thus, if he uses at least 1 kg of Imported Scrap to make the vendor happy despite the fact that it may not be optimal, he will lose INR2.93138 (Section 2.1.3).

By creating only one optimal batch stated previously in Table 1, the profit for the month while still satisfying all previously stated constraints can be determined. With the calculations shown in Section 2.1.4, the profit over the month is INR1,460,640.995[5], which is achieved by producing 269.438 batches with the recipe shown in Table 1.

Besides maximizing the profit of making only one batch, the optimal monthly profit can also be determined with a more holistic view of the director's goal. Although there is no restriction in the types of batches the director can make, we can make the assumption that optimality can be achieved through making only one type of batch, rather than multiple batches. This is because other variable costs like transportation and time consumption (of raw materials) will prove to be more efficient when only a handful of materials are being ordered. Further evidence would be needed to prove this assumption, but we used the assumption moving forward.

After creating the model in AMPL, the maximum monthly profit is INR1,788,704.747, which is achieved by making 367.074 batches of only one type with the amount of each material shown in Table 2.[6]

Table 2. Amount of Each Raw Material to Produce One Type of Batch to Maximize the Monthly Profit

Type of Raw Materials	Amount Used in 1 Month (Tons)	Amount Used in 1 Batch (Tons)
Tasla	789.878	2.15182
Rangeen	438.821	1.19545
Sponge	175.528	0.47818
Local Scrap	263.293	0.71727
Imported Scrap	0	0

[3] Task 3
[4] 1 metric ton = 1000 kg
[5] Task 4
[6] Task 5

HC	0	0
Pig Iron	87.7642	0.23909

Based on the results above of the optimal ordering, it can be seen that not only is Imported Scrap disregarded, but HC is as well. This is a difference we can observe from Task 4, when monthly profit was calculated from optimizing only on one batch. But with this new model, we can observe a higher monthly profit.

In conclusion, the best ordering strategy for Chandpur Enterprises Limited in August is to make 367.074 batches with recipe shown in Table 2 above.

1.3 Recommendations

The first suggestion[7], as already evidenced from Task 2, includes the possibility of increasing the regulatory size of 4 tons. Changing this would increase the profit by INR3395.27 per ton. As indicated through sensitivity analysis, avoiding the use of Imported Scraps and HC would further yield in additional monthly profits.

As briefly seen in Task 5, we made the assumption that making one batch was more optimal than making multiple. However, in the future, this could be actively accounted for through the maximization of the available 600 working hours, meaning reducing wastages in raw material and overall production time. Taken into account all of our given constraints, we can conclude through linear programming that the best ordering strategy for Chandpur Enterprises Limited is to maximize monthly profit from producing one type of batch. However, we must consider that there may be a more optimal solution that could be found from using another programming technique, possibly one that allows for nonlinearity.

In conclusion, Chandpur Enterprises Limited should increase regulatory size to maximize monthly profits. Also, since the optimal batch for the maximum profit is different per batch than per month, they should also consider investing in the limiting material so that the optimal batch can be used for the whole month. Another option would be to make the optimal batch to its full capacity, and then in the remaining time make a different batch. However, this still would not be the full optimum as if making the optimal batch for the whole month.

[7] Task 6

Part II. Appendix

2.1 Model 1

2.1.1 Linear Program Development

Table 3. Constraints for Raw Materials

	Rate / Ton (in rupees)	Recovery	Minimum per Batch (% of RM)	Maximum per Batch (% of RM)	Maximum per Month (in tons)
Tasla	17,000	0.84	0%	50%	800
Rangeen	13,600	0.74	0%	25%	500
Sponge	17,800	0.85	10%	50%	1,000
Local Scrap	20,000	0.94	15%	80%	1,000
Imported Scrap	23,000	0.97	0%	80%	1,500
HC	2,500	0.25	0%	20%	300
Pig Iron	20,400	0.95	5%	10%	500

The first model we developed solves tasks 1-4. It's main objective is to determine the optimal batch possible if only making one batch. Thus, the decision variable is x_i, i=1..7, which represent tons of each raw material i used per batch. x_i refers to the i^{th} item in Table 3 of the project instructions, so x_1 represents tasla, x_2 represents rangeen, etc. We used the parameters r_i, c_i, $minim_i$, and $maxim_i$ to represent recovery, rate/ton, minimum per batch, and maximum per batch, respectively, provided in Table 3 for each raw material i, i=1..7. The objective function of this linear program is thus to maximize the profit associated with making this one batch. First, it should be noted that the sum of all x_i represents the total mass of raw materials put into one batch. This has a different meaning than the total mass of the batch after it is processed in the furnace, because there are certain recovery values for each raw material. So, the mass of x_i after it is processed is $r_i x_i$. In the objective we have to include the price/ton of finished product (INR29,000), rate/ton of each raw material (c_i), electricity per batch $4.3*(1200+700*\sum x_i)$, salary/batch (INR3,000), and other consumables/ton (INR2,000). So, we end up with the following objective function (in INR):

$$29000 \sum_{i=1}^{7} r_i * x_i - \sum_{i=1}^{7} c_i * x_i - 4.3(1200 + 700 \sum_{i=1}^{7} x_i) - 3000 - 2000 \sum_{i=1}^{7} r_i * x_i \qquad \text{(Equation 1)}$$

The constraints for this linear program can mostly be found from Table 3. First of all, the finished mass per batch cannot exceed 4000 kg (or 4 tons). Then, we have to be sure that each raw material stays within its maximum and minimum percentage limitations of the total amount of raw materials put into the batch (before processing). See the following equations. Nonnegativity is included with the $minim_i$ constraints. Note that all of the constraints below are in tons.

$$\sum_{i=1}^{7} r_i * x_i \le 4 \qquad \text{(Equation 2)}$$

$$minim_j * \sum_{i=1}^{7} x_i \le x_j \le maxim_j * \sum_{i=1}^{7} x_i, \quad \forall j$$

(Equation 3)

2.1.2 Code

```
#Group Project Problem 1
#IEOR 162
#Lily Engel, Ben Lee, Alice Liu, Jessica Richard

#Load project1.dat before running this program

var x{1..7}>=0; #tons each raw material per batch

param r{1..7}; #recovery each raw material
param c{1..7}; #cost each raw material
param minim{1..7}; #minimum %RM per batch
param maxim{1..7}; #maximum %RM per batch

maximize profit: 29000*sum{i in 1..7} r[i]*x[i] - sum{i in 1..7} c[i]*x[i] -
4.3*(1200+700*sum{i in 1..7} x[i]) - 3000 - 2000*sum{i in 1..7} r[i]*x[i];

subject to

capacity: sum{i in 1..7} r[i]*x[i] <= 4;  #4000 kg finished product per batch

max_lims {j in 1..7}: x[j] <= maxim[j]*sum{i in 1..7} x[i];  #max limit each raw material per batch
min_lims {j in 1..7}: x[j] >= minim[j]*sum{i in 1..7} x[i];  #min limit each raw material per batch
```

2.1.3 Output

```
ampl: reset;
ampl: option solver cplex;
ampl: option cplex_options 'sensitivity';
ampl: option presolve 0;
ampl: model Project1.mod;
ampl: data Project1.dat;
ampl: solve;
CPLEX 12.8.0.0: sensitivity
CPLEX 12.8.0.0: optimal solution; objective 5421.071677
6 dual simplex iterations (1 in phase I)

suffix up OUT;
suffix down OUT;
suffix current OUT;
ampl: display profit;
profit = 5421.07

ampl: display x;
x [*] :=
1  1.39179
2  1.39179
3  0.556715
4  0.835073
5  0
6  1.11343
7  0.278358
;

ampl: display _varname, _var.rc, _var.down, _var.current, _var.up;
: _varname       _var.rc       _var.down   _var.current   _var.up     :=
1   'x[1]'       1.42109e-13     2108         2670        3391.28
2   'x[2]'       5.96856e-13     2365.43      3370       1e+20
3   'x[3]'      -2.27374e-13    -4840.51      2140        2704.74
4   'x[4]'      -1.27898e-13    -2283.67      2370        3023.16
5   'x[5]'   -2931.38           -1e+20         180        3111.38
6   'x[6]'       1.13687e-13      747.651     1240       1e+20
7   'x[7]'       0              -11721        2240       3049.68
;

ampl: display _conname, _con, _con.slack, _con.up, _con.current, _con.down;
# $5 = _con.current
:      _conname         _con       _con.slack      _con.up     $5     _con.down
 :=
1    capacity        3395.27     8.88178e-16   1e+20          4        0
2    'max_lims[1]'      0        1.39179       1e+20          0      -1.39179
3    'max_lims[2]'   1039.53     2.77556e-17       1.44196    0      -1.34499
4    'max_lims[3]'      0        2.22686       1e+20          0      -2.22686
5    'max_lims[4]'      0        3.61865       1e+20          0      -3.61865
6    'max_lims[5]'      0        4.45372       1e+20          0      -4.45372
7    'max_lims[6]'    573.208    6.93889e-17       1.75131    0      -0.956366
8    'max_lims[7]'      0        0.278358      1e+20          0      -0.278358
9    'min_lims[1]'      0        1.39179           1.39179    0   -1e+20
10   'min_lims[2]'      0        1.39179           1.39179    0   -1e+20
11   'min_lims[3]'   -563.953   -6.93889e-18       1.38696    0      -0.557491
12   'min_lims[4]'   -639.527    3.46945e-17       1.34499    0      -0.852878
13   'min_lims[5]'      0        0                 0          0   -1e+20
14   'min_lims[6]'      0        1.11343           1.11343    0   -1e+20
15   'min_lims[7]'   -803.479   -5.55112e-17       0.276243   0      -0.280505
;
```

2.1.4 Interpreting the Output

The output of this code gives the profit per batch as well as the tons of each raw material in the one optimal batch. To solve task 4, which asks for the profit associated with this optimal batch over the month, we first need to determine the amount of time this batch will take to make.

$$Time\ of\ one\ batch\ =\ 0.2 + 0.3 * \sum_{i=1}^{7} x_i = 0.2 + 0.3(5.56715) = 1.87\ hours$$

So, out of the 25 working days per month and 24 hours per day, there are 600 hours available per month. To solve the maximum amount of batches that can be made per month:

$$600\ \frac{hours}{month} * \frac{1\ batch}{1.87\ hours}\ =\ 320.83\ \frac{batches}{month}$$

However, if we multiply x_i by the number of batches to get the amount of each raw material used per month, we obtain:

Table 4. Amount of Each Raw Material of Optimal Batch in One Month

Type of Raw Materials	Amount Used in 1 Month (Tons)
Tasla	446.527
Rangeen	446.527
Sponge	178.611
Local Scrap	267.916
Imported Scrap	0
HC	357.222
Pig Iron	89.3055

If we compare this to the maximum amount of each material per month in Table 3, x_6 is in violation. So, this is our limiting material, and we can only make the number of batches before this limiting amount is reached. Since that limiting amount is 300 tons and all of the other materials are under their maximums, the maximum number of batches that can be made with this recipe is:

$$300\ \frac{tons}{month}\ /\ x_6\ \frac{tons}{batch} = 300\ \frac{tons}{month}/1.11343\ \frac{tons}{batch}\ =\ 269.438\ \frac{batches}{month}$$

So, the monthly profit with this batch is

$$269.438\ \frac{batches}{month} * \frac{INR\ 5421.07}{batch} = INR1,460,640.995\ .$$

Thus, since the factory is not being used to its full time potential, there must be a better batch to maximize monthly profits.

2.1.5 Sensitivity Analysis

The sensitivity analysis of this program shows which constraints are binding, the reduced costs of each variable, and how we can fluctuate either the coefficient of each variable in the objective function or the right hand sides constraints and maintain the same basis as well as the shadow price. Thus, we can interpret which changes could bring better profit.

In Section 2.1.3 above, the column named '_var.rc' is the reduced cost of each variable, which gives the change in the objective function value with every unit increase of the optimal variable. For example, _var.rc of x_5 is -2931.38. It means that for every additional unit of x_5 used, the optimal objective function (profit of making one batch) is reduced by INR2931.38. '_var.current' is the current objective coefficients. '_var.down' and '_var.up' are the range of the objective function coefficients that can decrease or increase while maintaining the optimality of the current basis.

'_con' gives the shadow price of each constraint, which is the change in optimal objective function value with unit change of each constraint. For example, for the first constraint (maximum finished product in 1 batch is 4 tons), the shadow price of this constraint is INR3395.27, thus for every 1 ton increase in this constraint, the objective value (profit of making one batch) will be increased by INR3395.27. '_con.slack' gives the slack in each constraint. '_con.current' (or '$5'), '_con.down' , and '_con.up' give the current value and the range of the right hand side of each constraint where it can decrease or increase while maintaining the optimality of the current basis.

2.2 Model 2

2.2.1 Linear Program Development

For task 5, we needed to develop a new model. This is because now we are trying to maximize the monthly profit as opposed to the profit of only one batch. So, in order to solve that and keep this program linear, we needed to rename our decision variables. This time, x_i represents the tons of each raw material used per month, where i=1..7 and refers to each row in Table 3. We added a new variable 'a', which is the number of batches made per month. The only new parameter added is $maxmon_i$, which is the maximum tonnage of each raw material per month. The rest were the same from Model 1. So, the new objective function had to account for the fact that certain costs were evaluated on a per ton basis and others were done on a per batch basis. The new function is (in INR):

$$29000 \sum_{i=1}^{7} r_i * x_i - \sum_{i=1}^{7} c_i * x_i - 4.3(1200a + 700 \sum_{i=1}^{7} x_i) - 3000a - 2000 \sum_{i=1}^{7} r_i * x_i \qquad \text{(Equation 4)}$$

Also, since the minimum and maximum per batch are determined as a ratio, those constraints can remain the same as in Model 1. The batch constraint needs to be multiplied by a so that it is determined on a monthly scale, and the maxmon limits needs to be accounted for as well. Finally, there is also a time limit for when the factory can be running. There are 25 working days per month and 24 hours per day, so there are 600 hours per month that can be used. The time of one batch in hours is $0.2+(0.3*\sum x_i)$. The constraints for this model are as follows (in tons):

$$\sum_{i=1}^{7} r_i * x_i \leq 4a \qquad \text{(Equation 5)}$$

$$minim_j * \sum_{i=1}^{7} x_i \leq x_j \leq maxim_j * \sum_{i=1}^{7} x_i, \quad \forall j$$

(Equation 6)

$$x_j \leq maxmon_j * \sum_{i=1}^{7} x_i, \quad \forall j \qquad \text{(Equation 7)}$$

$$0.2a + 0.3 \sum_{i=1}^{7} x_i \leq 600 \qquad \text{(Equation 8)}$$

2.2.2 Code

```
#Group Project Problem 5
#IEOR 162
#Lily Engel, Ben Lee, Alice Liu, Jessica Richard

#Load project2.dat before running this program

var x{1..7} >= 0; #tons each raw material per month
var a >= 0; #number batches per month

param r{1..7}; #recovery each raw material
param c{1..7}; #cost each raw material
param minim{1..7}; #minimum %RM per batch
param maxim{1..7}; #maximum %RM per batch
param maxmon{1..7}; #maximum per month (tons)

maximize profit: 29000*sum{i in 1..7} r[i]*x[i] - sum{i in 1..7} c[i]*x[i] -
 4.3*(1200*a+700*sum{i in 1..7} x[i]) - 3000*a - 2000*sum{i in 1..7} r[i]*x[i];

subject to

capacity: sum{i in 1..7} r[i]*x[i] <= 4*a;  #4000 kg finished product per batch, multiply by a for total

max_lims {j in 1..7}: x[j] <= maxim[j]*sum{i in 1..7} x[i];  #max limit each raw material per batch
                                                                #-> per month (equivalent) since ratio
min_lims {j in 1..7}: x[j] >= minim[j]*sum{i in 1..7} x[i];  #min limit each raw material per batch
                                                                #-> per month (equivalent) since ratio
max_mons {j in 1..7}: x[j] <= maxmon[j];  #max limit each raw material per month

time: 0.2*a + 0.3* sum{i in 1..7} x[i] <= 600;  #max limit time per month
```

2.2.3 Output

```
ampl: reset;
ampl: option solver cplex;
ampl: option cplex_options 'sensitivity';
ampl: option presolve 0;
ampl: model Project2.mod;
ampl: data Project2.dat;
ampl: solve;
CPLEX 12.8.0.0: sensitivity
CPLEX 12.8.0.0: optimal solution; objective 1788704.747
6 dual simplex iterations (0 in phase I)

suffix up OUT;
suffix down OUT;
suffix current OUT;
ampl: display profit;
profit = 1788700

ampl: display x,a;
x [*] :=
1  789.878
2  438.821
3  175.528
4  263.293
5    0
6    0
7   87.7642
;

a = 367.074

ampl: display _varname, _var.rc, _var.down, _var.current, _var.up;
: _varname       _var.rc       _var.down _var.current   _var.up    :=
1   'x[1]'       2.27374e-13     2525.95      2670      3594.99
2   'x[2]'      -2.27374e-13     2454.44      3370      9787.29
3   'x[3]'       1.13687e-13    -8050.4       2140      2691.97
4   'x[4]'      -1.13687e-13    -4423.6       2370      2890.05
```

```
5   'x[5]'   -2774.58          -1e+20          180      2954.58
6   'x[6]'    -138.455         -1e+20         1240      1378.46
7   'x[7]'      -2.27374e-13   -18140.8       2240      2911.34
8   a            3.41061e-13    -9229.55     -8160      1817
;

ampl: display _conname, _con, _con.slack, _con.up, _con.current, _con.down;
:      _conname        _con       _con.slack      _con.up   _con.current    :=
1    capacity        2189.06       6.82121e-13     153.778         0
2    'max_lims[1]'      0         87.7642        1e+20            0
3    'max_lims[2]'    918.906      8.52651e-14      60.9561        0
4    'max_lims[3]'      0        702.114         1e+20            0
5    'max_lims[4]'      0       1140.93          1e+20            0
6    'max_lims[5]'      0       1404.23          1e+20            0
7    'max_lims[6]'      0        351.057         1e+20            0
8    'max_lims[7]'      0         87.7642        1e+20            0
9    'min_lims[1]'      0        789.878          789.878         0
10   'min_lims[2]'      0        438.821          438.821         0
11   'min_lims[3]'   -551.891      5.32907e-15     701.703         0
12   'min_lims[4]'   -518.906      8.88178e-15     738.327         0
13   'min_lims[5]'      0          0                 0            0
14   'min_lims[6]'      0          0                 0            0
15   'min_lims[7]'   -670.796      0                87.6937        0
16   'max_mons[1]'      0         10.1221        1e+20          800
17   'max_mons[2]'      0         61.179         1e+20          500
18   'max_mons[3]'      0        824.472         1e+20         1000
19   'max_mons[4]'      0        736.707         1e+20         1000
20   'max_mons[5]'      0       1500             1e+20         1500
21   'max_mons[6]'      0        300             1e+20          300
22   'max_mons[7]'      0        412.236         1e+20          500
23   time            2981.17       1.13687e-13     607.689       600
;
:     _con.down     :=
1    -12000
2       -87.7642
3       -10.1892

4      -702.114
5     -1140.93
6     -1404.23
7      -351.057
8       -87.7642
9    -1e+20
10   -1e+20
11      -10.1155
12      -10.0559
13   -1e+20
14   -1e+20
15      -10.0494
16      789.878
17      438.821
18      175.528
19      263.293
20        0
21        0
22       87.7642
23        0
;
```

2.2.4 Interpreting the Output

Since the output for this model gives x_i in terms of raw material used per month, the amount of each material per batch can be determined by x_i/a. Similarly, the profit per batch is profit/a since the output of this code gives monthly profit.

2.2.5 Sensitivity Analysis

The sensitivity analysis of this program follows the same format in the output to what we saw for the first model. Refer to section 2.1.5.