C9 ✓

Executive Summary

LLBG Consulting was asked to analyze Disco's employment costs in order to suggest improvements to which of their current factories needed to remain open, and how many employees were needed in each of these to meet the expected demands. An integer-programming model was formulated and executed using OPL 3.5 software. From the results that we obtained, Disco's employment budget will be reduced to $5,812,520 from the previous year's budget of $8,000,000. Based on last year's performance, this 24.37% reduction in the employment budget will in turn, lead to a 437.5 % profit increase.

too many #s in intro

After carefully reviewing the results, the most significant change to Disco's current procedures is the closing of the Cincinnati factory. This result was not surprising considering the fact that Cincinnati's fixed costs of operation was the highest among the four factories while simultaneously, its' labor capacity was the lowest of the four.

In order to tackle the problem, we carefully reviewed the data and developed the following conclusions:

1. Due to tax laws that constrain the factories, Cincinnati is most cost inefficient factory as well as the optimal factory to close.
2. Using the other three factories, all labor demand and capacity constraints can be achieved, all while decreasing the employment budget by $2,187,480.
3. The deleted operational costs from the Cincinnati factory are more than sufficient to support the pay for overtime hours, training new employees and the compensation owed to workers who have been terminated in the other three factories; all while increasing profits by 437.5%.

Recommendation:

From our conclusions above, LLBG wholeheartedly supports the validity of the optimal solution that was obtained from the integer-programming model. Although this solution is the most cost efficient solution to the problem that Disco faces, there are definitely drawbacks to the solution that apply to real life that the software does not take into consideration. One of these factors is that there are several occasions where workers are terminated and then shortly thereafter, more are hired and trained in order to meet constraints. Another drawback to the solution is that in many cases, there are workers who put in quite a bit of overtime, which may carry on for substantial ~~amounts of time~~. Taking into account these drawbacks and the probable presence of others, LLBG recommends that if immediate action is necessary, no workers shall be fired after the beginning of the year and that the Cincinnati factory is to be closed. We support this recommendation because we believe that high turnover is detrimental to a company in many ways. That being said, LLBG recommends additional research necessary to develop other feasible solutions in which Disco's integrity as a company remains unscathed by the changes applied. In conclusion, LLBG would like to further extend its services to Disco for the opportunity to resolve the problems at hand and help make it the company that it is striving to be.

wordy

Technical Report

The employment problem that Disco presented LLBG consulting was a multi faceted problem. The first aspect that needed to be researched was whether or not it would be beneficial to close a certain factory or factories. For the second part of the problem, an employment procedure was in need to determine how many people should be hired, trained, fired and how many hours of overtime each factory will use for the upcoming year. To solve this problem, LLBG created an integer-programming model and formulated it into OPL 3.5 software. From the results obtained, Disco will benefit from a $2,187,480 employment budget decrease, which will in turn lead to a 437.5% profit increase based on the previous year's assessments.

Assumptions:

In assessing the problem that is facing Disco, there were assumptions that had to be made. One assumption is that people can only be hired and/or fired at the beginning of the month. Another assumption that had to be made is that no employees would become dissatisfied enough to quit. The model that we developed also assumes that everyone on schedule shows up and can provide overtime hours whenever they are needed. Next, we had to assume that employees are guaranteed salary for a full months wage no matter how many hours they are needed to work per month.

Range:

Setting up a range of data was our first objective. Our range for months stretched from 1 to 12, symbolizing January to December for the upcoming year. Next, our range for factories stretched from 1 to 4, symbolizing factories located in Kansas City, Memphis, Cincinnati and Phoenix. Finally, our range for years stretched from 0 to 12, symbolizing that we needed to include last December for certain aspects.

Variables:

Choosing variables is the next step in solving this problem. Below is a list of our variables and a description of what they stand for.

Need Names

-1/2

Variable Name	Description
loc	Where the factory is located.
workers	Number of workers that have been working for at least 1 month.
overtime	Number of overtime hours that workers will use in the present month.
training	Number of people hired and training in the present month.
compensation	Number of workers fired and given compensation in the present month.
hire	Number of people hired for a given month. (Equal to training)
employee	Total number of employees in the present month. (workers + hire)

For the location variable (loc), there were four locations. Each city represents a location from 1 to 4, therefore, Kansas City is loc [1], Memphis is loc [2], Cincinnati is loc [3], and Phoenix is loc [4]. The location variable is also set up to have either an open or close option, equal to 1 if it is open, 0 if it is not (binary).

The next variable was named workers [i,j]. As with this variable and the rest to follow, the "j" stands for which factory you are currently located, and "i" is the current month. When looking at the output data, realize that the workers variable pertains only to those employees that have worked at Disco for at least one month. The total number employees needed for a given month could be calculated by adding the variables workers [i,j] plus training [i,j].

Overtime [i,j] is the third variable listed in the chart, and as the name would suggest, represents the overtime hours worked at a given factory for a given month. This variable is needed so that the time and a half charge that goes with overtime can be put into the objective function.

The training [i,j] variable represents the amount of people that need to be trained that month. It also represents the number of workers [i,j] that have to conduct the training. As noted above, the training [i,j] variable is not part of the workers [i,j] variable.

The next variable created in the program was named compensation [i,j]. Compensation represents the number of people fired at the end of the previous month that need to be paid compensation for the current month's standard hours as stated by the union.

The last two variables, hire [i,j] and employee [i,j], are not needed for the program to run. These variables are merely there to make are output easier to comprehend. The hire variable is set equal to the training variable in order to state the fact that any person hired in any given month will also be trained in that same month. The employee variable was created in order to total up the number of people who are currently working in the factory as a worker, trainee or trainer. Nice explanation

Constraints:

Below is a list of constraints and how each of them was programmed into OPL.

1 This constraint initializes the number of employees in month 0 (December) to the data we received from Disco when we accepted to do this project. This constraint was needed so that we did not have to hire all the help for the first month.

- forall(j in factory) workers[0,,j]=employees[j]

2 To satisfy Disco's request that the total number of hours worked be less than the given capacity at a plant, this constraint was made. Since workers (those who have worked for more than a month) are the only ones that can be productive, they are the only ones that use plant capacity time. New hires do not take up plant capacity time.

The most important concept for this constraint is multiplying the right hand side by the binary location variable. Doing this enables you to get rid of the labor capacity of one factory if it need be closed.

- forall(i in month) forall(j in factory) (workers[i,j]*160 + overtime[i,j])<=(laborcapacity[j]*loc[j])

3 Making sure that we reach expected demand is vital, this constraint was constructed to achieve this goal. Taking workers times the available working time gives the total time to work without overtime. However, workers that are training (the training variable) are able to be productive for half of the time, or 80 hours. Subtracting out 80 times training gives the hours of productive work without overtime. The last step in this constraint was adding in overtime and setting all of this greater than or equal to our labor demand.

- forall(i in month) sum(j in factory) (workers[i,j]*160 - training[i,j]*80 + overtime[i,j])>=labordemand[i]

4 In order to train newly hired employees Disco had to make sure that there were enough workers that had been there a month to train these new employees. That is why making sure your training variable was less than or equal to workers is necessary.

- forall(i in month) forall(j in factory) training[i,j]<=workers[i,j]

5 In order to limit the overtime to 40 hours per worker, overtime is set less than or equal to 40 times the number of workers.

- forall(i in month) forall(j in factory) overtime[i,j]<=40*workers[i,j]

6 To calculate the number of workers in a current month, it is necessary to set it equal to the workers plus the employees that were trained minus those that were fired all in the previous month.

- forall(i in month) forall(j in factory) workers[i,j]=workers[i-1,j]+training[i-1,j]-compensation[i,j]

7 The first location constraint makes sure that the Kansas City factory and the Cincinnati factory are either both closed or both open.

- loc[1]=loc[2];

8 Making sure that at least the Memphis or Cincinnati factory is open, if not both, is satisfied by this constraint. This is true because setting location 2 plus location 3 greater than or equal to one requires that at least one of the factories be open.

- loc[2]+loc[3]>=1

9 The constraint that required the Phoenix factory to be closed if the Kansas City and Cincinnati plants are open and at least one plant had to be closed was very difficult to formulate. In order to return the optimal solution, we ran the program twice, one time using the Phoenix factory and the other time not using it.

- loc[1]=1, loc[2]=1,loc[3]=1 and loc[4]=1
- loc[1]=1, loc[3]=1 and loc[4]=0

Objective Function:

The goal of Disco's request was to minimize the cost of labor and the operational costs associated with their factories. Making sure not to over look any cost is obviously the most crucial step of writing the objective function. In Disco's situation, there were five different types of cost to take into account.

1) Fixed factory (operational) costs
2) Worker salaries for the month
3) Overtime costs
4) Training costs
5) Compensation to terminated employees

Another important factor is to make sure that the correct pay is going to the correct variable. For example, the workers variable should be multiplied by standard union hours (160) and multiplied again by the wage they receive ($14). Below is the objective function that was placed into the program designed for Disco.

```
MINIMIZE    loc[1]*1100000 + loc[2]*1500000 + loc[3]*2100000 +
            loc[4]*1700000 +sum(i in month) sum(j in factory) workers[i,j]*160*14
            + sum(i in month) sum(j in factory) overtime[i,j]*14*1.5 +
            sum(i in month) sum(j in factory) training[i,j]*10*160 +
            sum(i in month) sum(j in factory) compensation[i,j]*160*14
```

Results:

In evaluating the optimal solution, we were able to establish firm conclusions. First of all, our solution will provide a $2,187,480 employment budget decrease. By saving this amount of money, profits for the company will more than quadruple. To achieve this, the Cincinnati factory will be shut down and the other three factories will fulfill the demanded worker hours. By accepting this solution, Disco will hire a total of 32 workers throughout the year and will fire a total of 43. The workers will be paid for 37,800 hours more than are demanded while the total capacity of the workers is underachieved by 19,540 hours.

Recommendation:

The result that OPL provided Disco is the minimum cost with which the company can operate. Although this result entails closing the Cincinnati factory, after looking at the data, we feel that this is the best decision. There are two significant reasons why Disco needs to close this plant. First, its fixed cost for being open is the highest of the four factories by a half million dollars. Secondly, it has the smallest capacity for working hours. We find these aspects reasonable grounds to shut the facility down.

As with all computer-generated results, there were drawbacks to the optimal solution that apply to real life challenges. The first problem that we noticed is that there are only 100 accessible hours left to work within the month of July (July has the third highest demand); these 100 hours are including your overtime hours. So to combat this problem, we recommend that four employees are hired in the month of May instead of the model's projected amount of two. By doing this, Disco is allotted another 400 hours of labor that might be needed to meet an unexpected spike in demand. Another problem that we discovered with the output model was located at each factory, but was most prevalent in the Phoenix facility. In August, nine workers are to be fired. However, in the next month, ten employees are to be hired. We recommend to Disco that such a fluctuation in employees not happen. The reasons for this are simple. First, it puts a lot of pressure on management to hire ten new employees. The other reason is that that many workers (employees that have been working for more than a month) will have to train them; these workers become half as productive.

With regards to the closing of the Cincinnati factory and the drawbacks associated with our optimal solution, we would also like to recommend that no workers are to be fired (unless justified by actions) after the first of the year. It is our belief that less turnover leads to a more unified and happier workforce. The difference in the objective value between the OPL solution and what LLBG is recommending is only $14220. We believe this extra cost is worth it to Disco in order to not have to scramble to ~~hire~~ employees. An additional bonus to this recommendation is that it puts less pressure on management to hire new employees. This in turn allows them to spend more time working on ways to improve the company from other angles.

Should provide the profit -1/2

Appendix 1) Model:

```
range month 1..12;
range factory 1..4;
range year 0..12;

int+ labordemand[month]=...;
int+ laborcapacity[factory]=...;
int+ employees[factory]=...;

var int+ loc[factory] in 0..1;
var int+ workers[year,factory] in 0..100;
var float+ overtime[month,factory];
var int+ training[year,factory] in 0..100;
var int+ compensation[month,factory] in 0..100;
var int+ hire[month, factory] in 0..100;
var int+ employee[month, factory] in 0..100;

minimize loc[1]*1100000 + loc[2]*1500000 + loc[3]*2100000 +
loc[4]*1700000 +sum(i in month) sum(j in factory) workers[i,j]*160*14
+ sum(i in month) sum(j in factory) overtime[i,j]*14*1.5 +
sum(i in month) sum(j in factory) training[i,j]*10*160 +
sum(i in month) sum(j in factory) compensation[i,j]*160*14

subject to{
        forall(j in factory) workers[0,j]=employees[j];
        forall(i in month) forall(j in factory) (workers[i,j]*160 +
        overtime[i,j])<=(laborcapacity[j]*loc[j]);
        forall(i in month) sum(j in factory) (workers[i,j]*160 - training[i,j]*80 -
        overtime[i,j])>=labordemand[i];
        forall(i in month) forall(j in factory) training[i,j]<=workers[i,j];
        forall(i in month) forall(j in factory) overtime[i,j]<=40*workers[i,j];
        forall(j in factory) workers[1,j]=workers[0,j]-compensation[1,j];
        forall(i in month) forall(j in factory) workers[i,j]=workers[i-1,j]+training[i-1,j]-
        compensation[i,j];

        forall(i in month) forall(j in factory) hire[i,j] = training[i,j];
        forall(i in month) forall(j in factory) employee[i,j] = workers[i,j] + hire[i,j];

        loc[1]=loc[2];
        loc[2]+loc[3]>=1;
        loc[3]=0 or loc[3]=1
};
```

Appendix 2) Optimal Solution:

Optimal Solution with Objective Value: 5812520.0000

loc[1] = 1
loc[2] = 1
loc[3] = 0
loc[4] = 1

workers[0,1] = 20
workers[0,2] = 15
workers[0,3] = 18
workers[0,4] = 16
workers[1,1] = 20
workers[1,2] = 9
workers[1,3] = 0
workers[1,4] = 16
workers[2,1] = 20
workers[2,2] = 5
workers[2,3] = 0
workers[2,4] = 15
workers[3,1] = 20
workers[3,2] = 5
workers[3,3] = 0
workers[3,4] = 10
workers[4,1] = 20
workers[4,2] = 5
workers[4,3] = 0
workers[4,4] = 10
workers[5,1] = 21
workers[5,2] = 8
workers[5,3] = 0
workers[5,4] = 18
workers[6,1] = 23
workers[6,2] = 8
workers[6,3] = 0
workers[6,4] = 19
workers[7,1] = 23
workers[7,2] = 8
workers[7,3] = 0
workers[7,4] = 19
workers[8,1] = 15
workers[8,2] = 8
workers[8,3] = 0
workers[8,4] = 18
workers[9,1] = 15
workers[9,2] = 8
workers[9,3] = 0
workers[9,4] = 18
workers[10,1] = 20
workers[10,2] = 16
workers[10,3] = 0
workers[10,4] = 22
workers[11,1] = 20
workers[11,2] = 16
workers[11,3] = 0
workers[11,4] = 22
workers[12,1] = 20
workers[12,2] = 16
workers[12,3] = 0
workers[12,4] = 22

overtime[1,1] = 0.0000
overtime[1,2] = 0.0000
overtime[1,3] = 0.0000
overtime[1,4] = 0.0000
overtime[2,1] = 100.00
overtime[2,2] = 0.0000
overtime[2,3] = 0.0000
overtime[2,4] = 0.0000
overtime[3,1] = 0.0000
overtime[3,2] = 0.0000
overtime[3,3] = 0.0000
overtime[3,4] = 0.0000
overtime[4,1] = 0.0000
overtime[4,2] = 0.0000
overtime[4,3] = 0.0000
overtime[4,4] = 0.0000
overtime[5,1] = 20.000
overtime[5,2] = 0.0000
overtime[5,3] = 0.0000
overtime[5,4] = 0.0000
overtime[6,1] = 420.00
overtime[6,2] = 320.00
overtime[6,3] = 0.0000
overtime[6,4] = 760.00
overtime[7,1] = 820.00
overtime[7,2] = 320.00
overtime[7,3] = 0.0000
overtime[7,4] = 760.00
overtime[8,1] = 0.0000
overtime[8,2] = 0.0000
overtime[8,3] = 0.0000
overtime[8,4] = 0.0000
overtime[9,1] = 0.0000
overtime[9,2] = 0.0000
overtime[9,3] = 0.0000
overtime[9,4] = 0.0000
overtime[10,1] = 720.0
overtime[10,2] = 0.000
overtime[10,3] = 0.000
overtime[10,4] = 0.000
overtime[11,1] = 800.0
overtime[11,2] = 0.000
overtime[11,3] = 0.000
overtime[11,4] = 420.0
overtime[12,1] = 20.00
overtime[12,2] = 0.000
overtime[12,3] = 0.000
overtime[12,4] = 0.000

training[0,1] = 0
training[0,2] = 0
training[0,3] = 0
training[0,4] = 0
training[1,1] = 0
training[1,2] = 0
training[1,3] = 0
training[1,4] = 0
training[2,1] = 0
training[2,2] = 0
training[2,3] = 0
training[2,4] = 0
training[3,1] = 0
training[3,2] = 0
training[3,3] = 0
training[3,4] = 0
training[4,1] = 1
training[4,2] = 3
training[4,3] = 0
training[4,4] = 8
training[5,1] = 2
training[5,2] = 0

Appendix 3) Employment Procedure Spreadsheet

Disco's Optimal Employment Procedure

	Workers	[illegible] Hired	Fired	Workers	[illegible]	[illegible]	[illegible]	[illegible] Hired	Hired	Workers	[illegible] Hired	[illegible]
January	14	0	6	15	0	0	0	0	18	16	0	0
February	11	0	3	14	0	1	0	0	0	15	0	1
March	11	0	0	14	0	0	0	0	0	10	0	5
April	11	2	0	14	0	0	0	0	0	10	10	0
May	13	2	0	14	1	0	0	0	0	20	0	0
June	15	0	0	15	0	0	0	0	0	20	0	0
July	15	0	0	15	0	0	0	0	0	20	0	0
August	15	0	0	15	0	0	0	0	0	11	0	9
September	15	7	0	15	0	0	0	0	0	11	10	0
October	22	0	0	15	0	0	0	0	0	21	0	0
November	22	0	0	15	0	0	0	0	0	21	0	0
December	22	0	0	15	0	0	0	0	0	21	0	0
Year Totals		11	9		1	1		0	18		20	15

	Regular Hours	Overtime	Total		Demand			[illegible]			Workable Hours	Unused Hours
January	7200	0	7200	>=	7200			0			9000	1800
February	6400	100	6500	>=	6500			0			8000	1500
March	5600	0	5600	>=	3700			1900			7000	3300
April	4640	0	4640	>=	4600			40			6040	1440
May	7280	20	7300	>=	7300			0			9160	1860
June	8000	1500	9500	>=	9500			0			10000	500
July	8000	1900	9900	>=	9900			0			10000	100
August	6560	0	6560	>=	6000			560			8200	2200
September	5200	0	5200	>=	5000			200			6840	1840
October	9280	720	10000	>=	10000			0			11600	1600
November	9280	1220	10500	>=	10500			0			11600	1100
December	9280	20	9300	>=	9300			0			11600	2300
						[illegible]	[illegible]	37800		Totals	109040	19540