Optimization of Andrew Printing

Reallocation of Printing Resources

Keshav Budwal, Charles Tong, Liren Peng, I-Ta Yang

12/21/2011

Abstract

The purpose of the project is to determine whether the current location of CMU printers are optimal and how we can reallocate the printers regarding the demand of each printer, the preference of student and the possible maximum printers at each location in order to increase productivity, minimize queues and potential frustration, and have an all-around better system for focusing on education. This problem has benefits in that it would streamline a commonplace activity that is in the days of most students at Carnegie Mellon. In this project, we create two different modes. First model is using non-linear programming; the other model is using linear programming. However, for both model, we use Brute-Force by Java program to find the optimal solution for reallocating the printer in CMU and the optimal numbers at each printing location.

Introduction

There are thirty-two printers located at 13 locations including academic buildings and dormitories in Carnegie Mellon University. Students can access to thirty-two printers from anywhere by using personal laptops or campus computers by sending their documents online to printers and simply swipe their student ID card at each printing location to print the documents. Therefore, it is not hard to find a close printer to print documents in Carnegie Mellon University. However, sometimes the queues of printer can be long, especially in rush hours, such as the ten minutes period between each class time. Since we can never know how much documents the people in the front are printing; therefore, the waiting time for printer is hard to estimate and sometimes can cause problems to students who are going to class. According to the survey on Surveymokey.com, among 53 CMU students, 92 percent respondents have waited in line for printing and they do not like to wait in line and there are 13.3 percent respondents usually wait in line for more than five minutes. The serious problem of long queues of printer in CMU inspires us to find the optimal numbers of printers at each printing locations. The table below is the current number of printers at locations including academic buildings and dormitories.

Table 1: Current Allocation of Printers

Location	Number of Printers
Baker	2
CFA	2
Cyert	2
Morewood	1
UC	1
Wean	5
West Wing	1
GHC	2
Posner	1
Donner	1
Mudge	1
Hunt	10
Mellon Institute	2

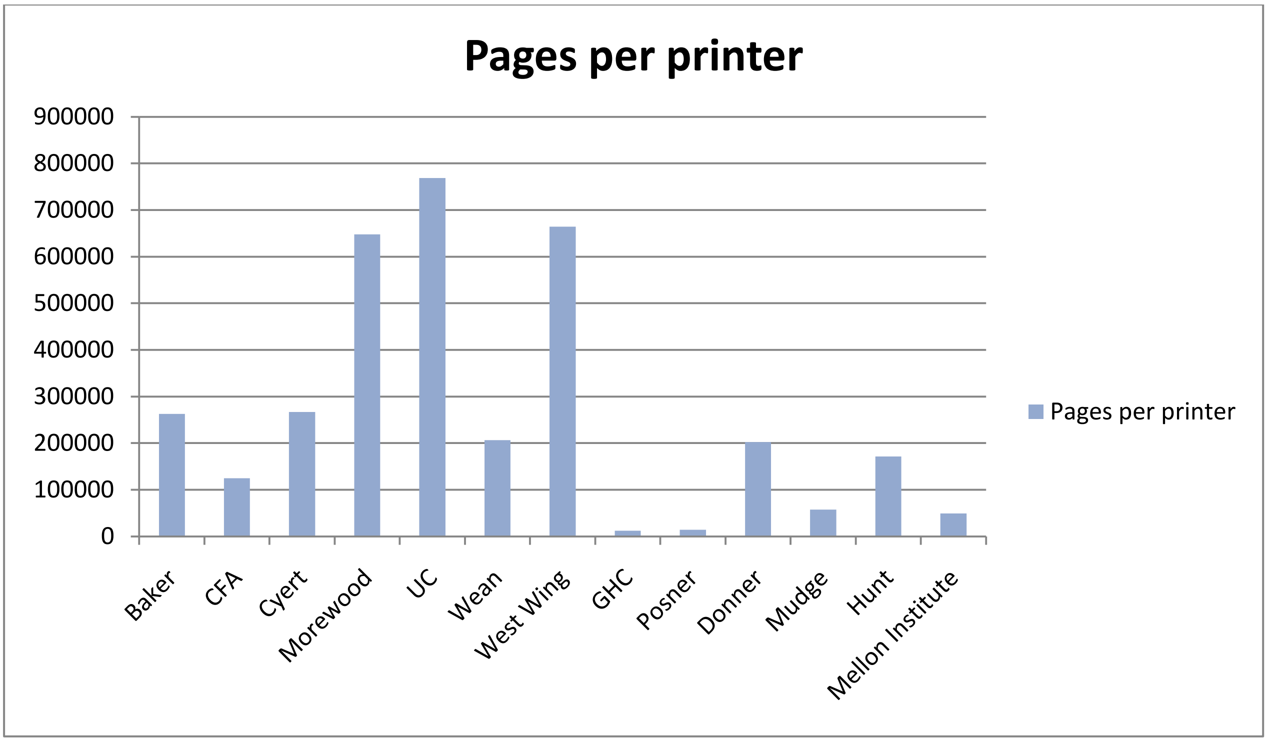
Current Status:

Historical Data:

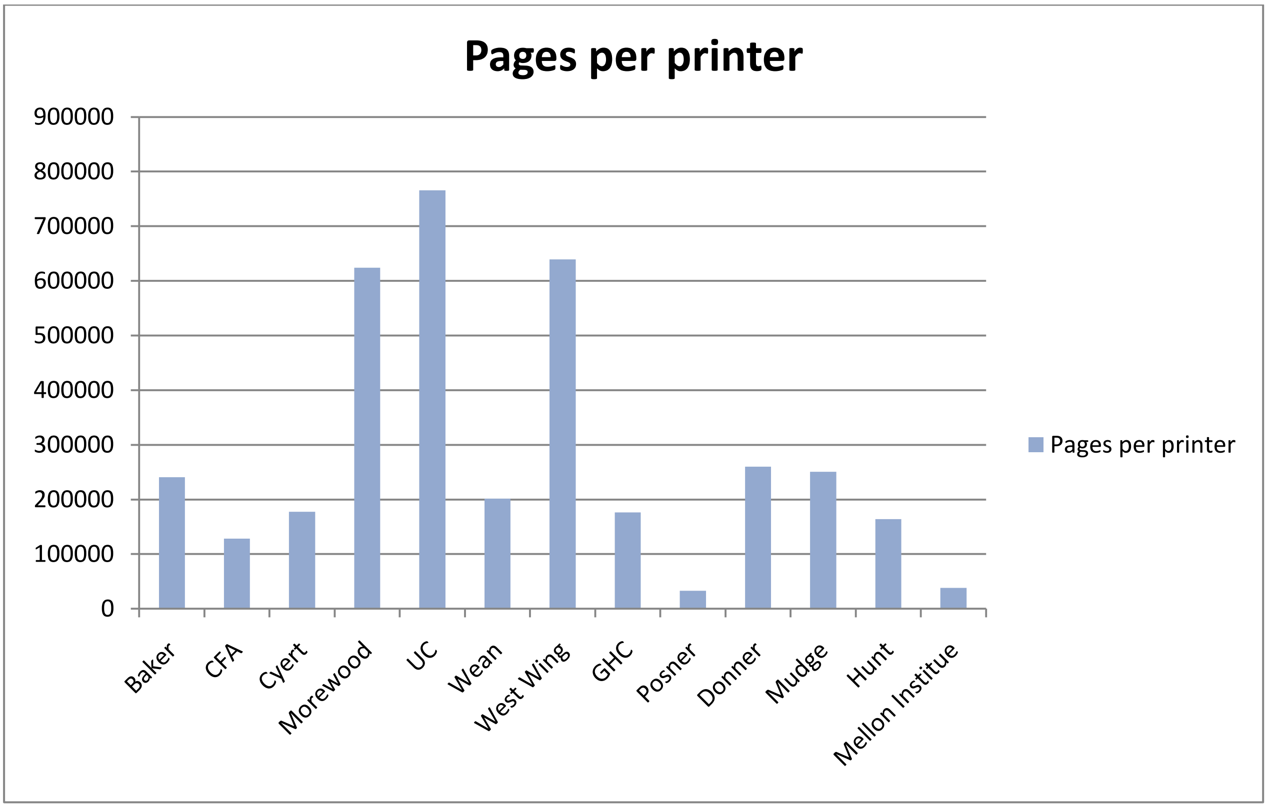
The historical data from 2009 to 2011 for printing demand of each location is shown below. The first bar graph is the demand for 2009 to 2010; the second bar graph is the demand for 2010 to 2011; the third graph is the aggregated demands from the first two graphs which presents the printing demand from 2009 to 2011. The demand is presents as pages per printers at each location, since there might be more than one printer at some locations, such as Baker, CFA and Hunt. We use "pages per printer" in order to compare the demand on the same scale. Looking at the aggregated data (third graph), UC has the highest pages printed per printer, but currently there is only one printer in UC. Morewood and West Wing have the second highest pages printed per printer. Although there are ten printers in Hunt, the printing demand in Hunt is low comparing to UC, Morewood, Westing, Donner, and Baker. From the historical data

(2009-2011), the reallocation must apply to improve the printing system in Carnegie Mellon University.

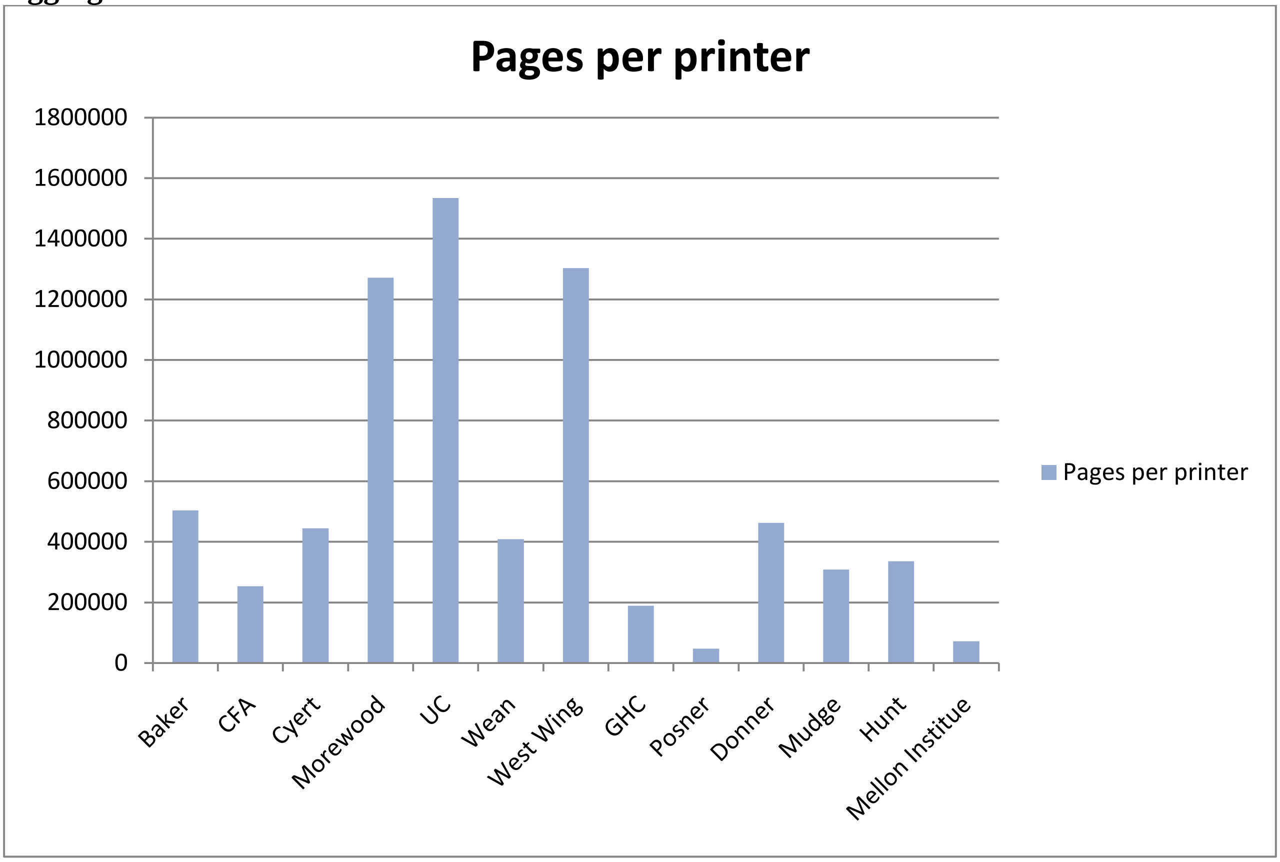
2009-2010



2010-2011

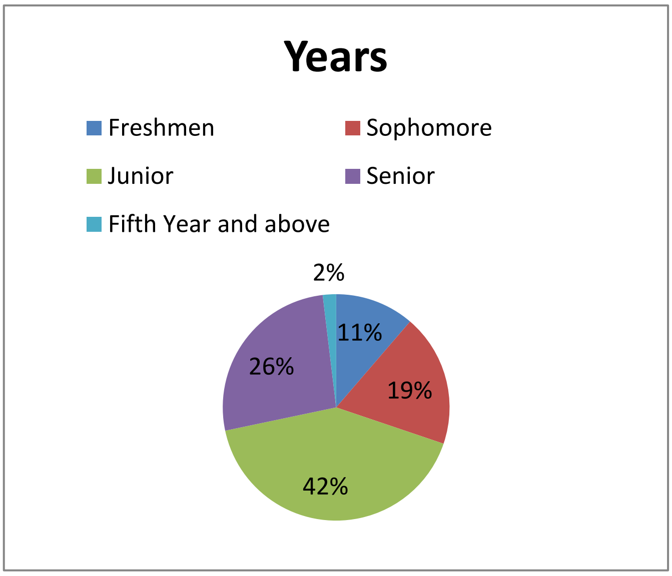


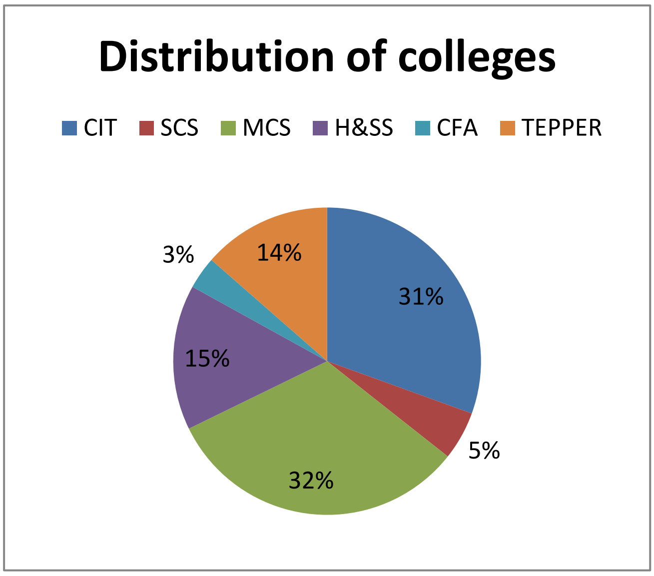
Aggregated

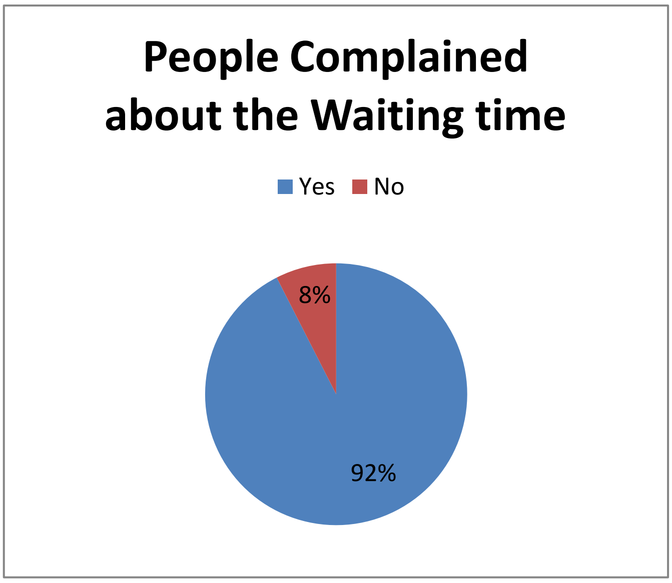


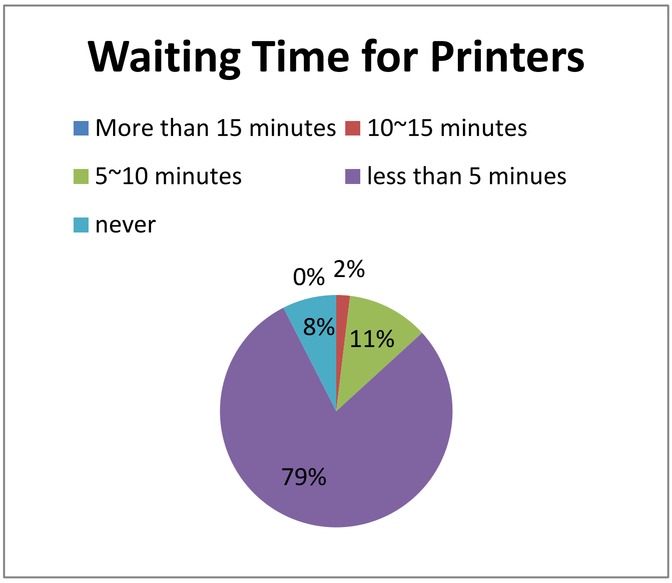
Survey Data:

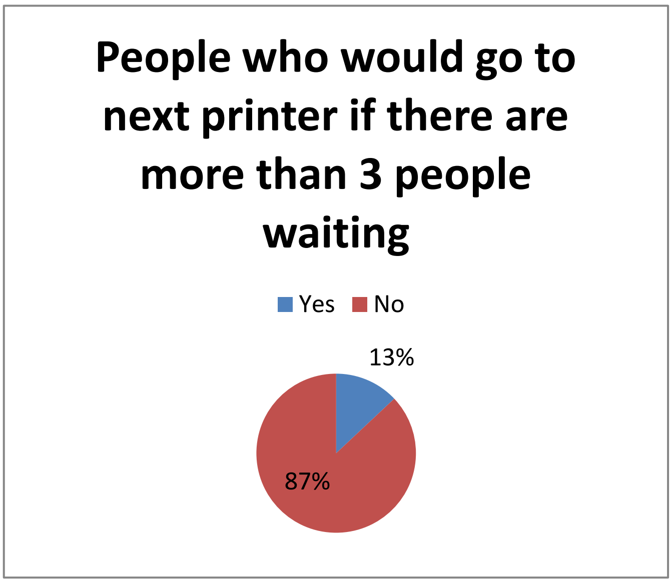
53 CMU students answered the survey from surveymokey.com. Most of the respondents (42%) are in junior year and (26%) senior year. 32 % respondents are from Mellon college of science, and 31% are from Carnegie Institute of Technology. Among 53 respondents, there are 92% people complaining about the waiting time for printers and about 13% respondents usually wait for more than 5 minutes. 13% respondents would go to next printer if they see that there are more than three people waiting in line. For our non-linear integer model, 13 percent from the survey is the percentage of demand which would be reallocated for each printing location. Comparing the preference of printing locations from all the respondents, the best printing location is Wean (24%); second printing location is Hunt (20%); third is Morewood and UC (14% each). The least prefer printing locations are Mudge(2%), Cyert (2%), and Mellon Institute (0%).

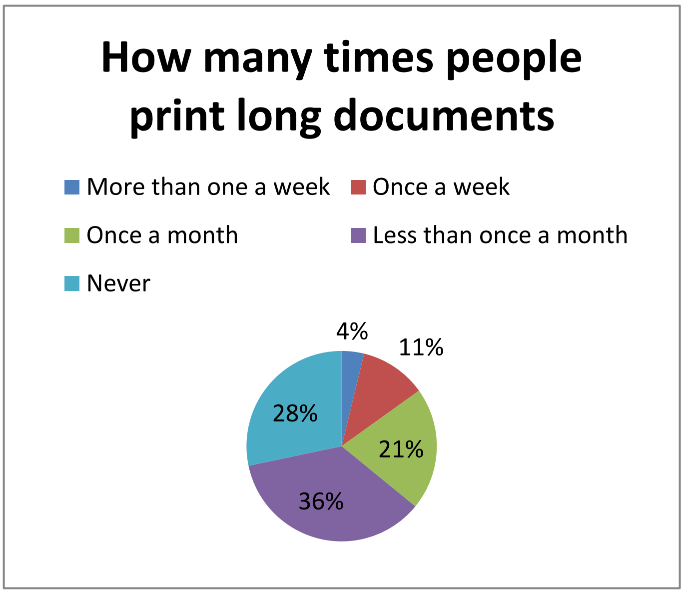

Years
Freshmen
Sophomore
Junior
Senior
Fifth Year and above
2%
11%
26%
19%
42%

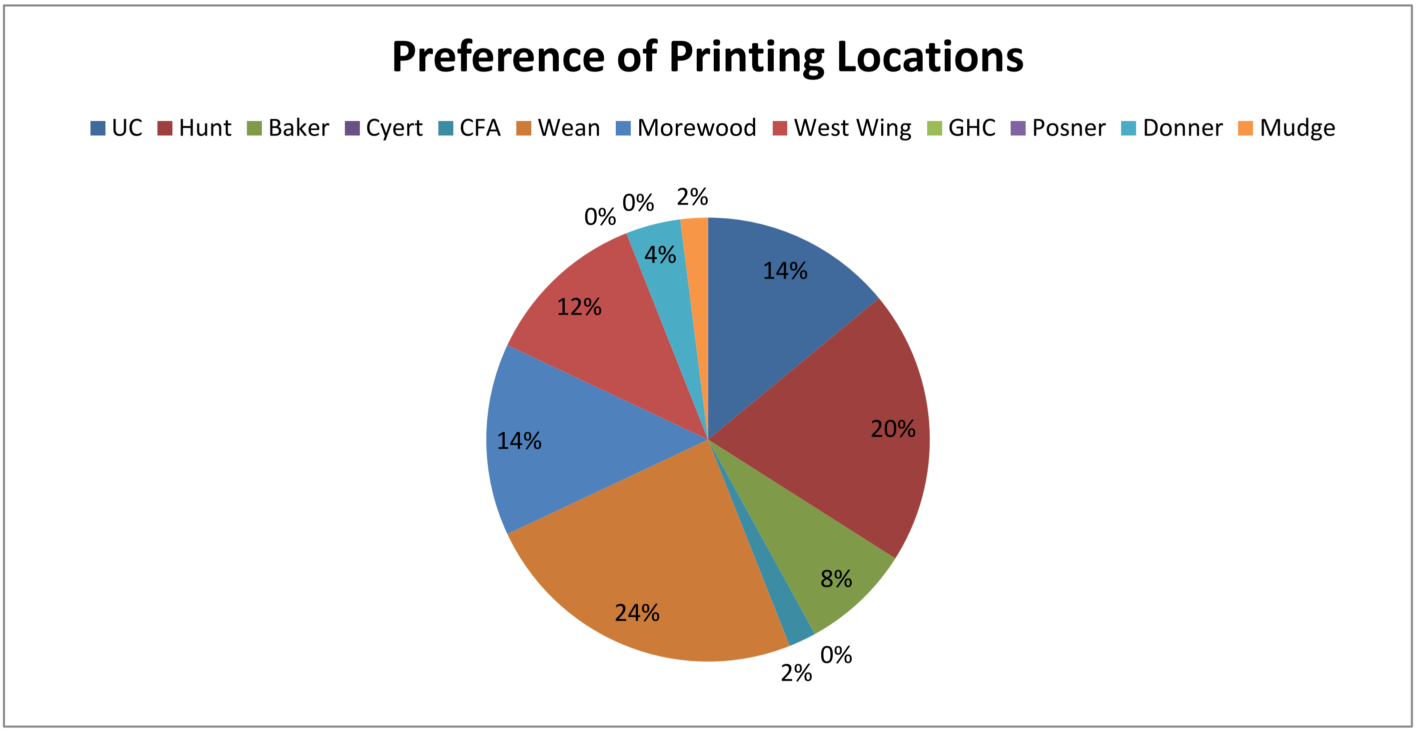

Distribution of colleges
CIT
SCS
MCS
H&SS
CFA
TEPPER
3%
14%
31%
15%
5%
32%


People Complained about the Waiting time
Yes
No
8%
92%


Waiting Time for Printers
More than 15 minutes
10~15 minutes
5~10 minutes
less than 5 minues
never
0%
2%
8%
11%
79%


People who would go to next printer if there are more than 3 people waiting
Yes
No
13%
87%


How many times people print long documents
More than one a week
Once a week
Once a month
Less than once a month
Never
4%
11%
28%
21%
36%



Modeling the Problem

In the course of deciding how to model the printing system, we developed two models, the first of which is presented here. For the final solution, we only ended up implementing this primary model for reasons discussed later in 'Model Selection'.

When deciding how to model printing, one of the major concerns was that demand at each location was affected by the current printer set-up. That is, that people change their printing habits or locations based on the current allocation of printers, this means that our objective function must also take into account potential changes in demand for each candidate reallocation of printers.

The first model deals with this by lumping potential effects into a linear factor of the number of printers at the location of interest, whereas the secondary model has a more complicated system of interdependencies between each printer and those closest to it. The primary advantage of lumping the potential effects into a linear factor is of making the model simple enough to actually optimize and solve the problem. In equation form, lumping the effecct into a linear factor of the actual number of printers at each location means that the actual demand at each location is $d_i + bx_i$ where d_i represents a "natural demand" at each location and x_i represent the number of printers at each location. The term b was determined by noting that overall demand must be preserved, hence we had a variable that indicated what fraction of demand was variable and had b equal to this amount of demand divided by the total number of printers. From the survey data, we approximated the percentage of demand by the number of people willing to move given sufficient queue lengths, hence 13% of demand.

With the modeling of actual demand done, our next task was to come up with an objective function to accurately represent the qualities we were looking to reduce. Assuming we knew the natural demand at each location, d_i , we were initially tempted to optimize over the objective function where we define an average queue length as simply $\frac{d_i+bx_i}{x_i}$ at each location, however for each additional person at a queue, we note that the cost is the same, when in reality, the longer the queue the more costly the wait. Hence we determined that we needed a concave cost function for each location in terms of this ratio and exponentiated each term, this meant that now the objective function would seek to lower average queues over all locations rather than allowing for simple redistributions. This gave us our primary model's objective function:

$$z = \sum e^{\frac{d_i+bx_i}{x_i}}$$

Our last problem with the model was to ascertain the constraints under which we were to optimize. Given that our only decision variables are the number of printers at each location, we simply constrained this by having each printer location have at least one printer but then have local maximum constraints which we determined personally by evaluating each location on a case by case basis. This actually allows for a relatively simple brute force approach wherein we can use nested for-loops for each location. The actual numbers for the maximum number of printers at each location that we determined can be easily found in the commented code attached in the appendix as the upper-bound on each for-loop.

Secondary Model

We also have constructed a secondary model for our optimization problem. In this secondary model, we use the same method of iterating through all feasible placements of printers. However, in our secondary model, instead of directly calculating an estimate of queue times, we construct linear programming problems within each iteration aimed at modeling the behavior of students given a placement of printers. In each iteration, solving our linear programming problem tells us how students on campus re-allocate from printer to printer in response to hypothetical printer congestion. With this model, we hope to provide a more accurate and detailed analysis of how successful any given placement of printers will be in achieving our goal of minimizing printing queue times and satisfying the needs of students.

Before going into the structure of the model itself, it is worth going over the parameters that are involved in this model. The first is the minimum and maximum numbers of printers that can be placed at any location, $\overrightarrow{\mathrm{Pmin}}$ and $\overrightarrow{\mathrm{Pmax}}$. This, with our total number of printers, P_{total}, provides our constraints for feasible solutions to the main problem of optimizing printer placement.

Next, we have a "true demand" $\vec{D}$, for each printing location. This parameter is not any kind of concrete value, but rather an index indicating how many people have each printing location as their preferred location when congestion at the location is not taken account. We can create a rough estimate of this parameter by observing historical printing rates and adjusting based off of survey results. Printing locations that are reported to be more congested will have a higher "true demand" than historical results would suggest. This is to account for students who have the location as their preferred printing location but re-allocate to other printers due to over-congestion.

Our last parameter for modeling student re-allocation is distance between each location. This is represented in our matrix P, where each element pij summarizes the distance from printing location i to printing location j. Note that for i = j, pij is equal to zero as there is zero distance between a location and itself. Note also, that certain elements of our matrix have been omitted. This is because certain re-allocations are impractical and unlikely to occur. For example, students who wish to print at the Mellon Institute are unlikely to re-allocate to other locations regardless of how congested the Mellon Institute printing location becomes. These omitted elements may also be thought of as having an M value, as they have a prohibiting cost.

Table 2: Printing Cluster Distances

	Baker	**CFA**	**Cyert**	**Morewood**	**UC**	**Wean**	**West Wing**	**GHC**	**Posner**	**Donner**	**Mudge**	**Hunt**	**Mellon**
Baker	0	~	~	~	~	7.1	~	~	~	~	~	5.7	~
CFA	~	0	~	~	8.5	~	8.5	~	3.4	~	~	3.7	~
Cyert	~	~	0	7.4	6.4	~	~	3.9	~	~	~	~	~
Morewood	~	~	7.4	0	10.8	~	~	~	~	~	10.7	~	~
UC	~	8.5	6.4	10.8	0	~	4.3	7.9	11.1	~	~	10.3	~
Wean	7.1	~	~	~	~	0	~	6.9	~	~	~	~	~
West Wing	~	8.5	~	~	4.3	~	0		~	7.8	7.3	~	~
GHC	~	~	3.9	~	7.9	6.9	~	0	~	~	~	~	~
Posner	~	3.4	~	~	11.1	~	7.8	~	0	8.8	~	4.4	~
Donner	~	~	~	~	~	~	7.3	~	8.8	0	~	~	~
Mudge	~	~	~	10.7	~	~	~	~	~	~	0	~	~
Hunt	5.7	3.7	~	~	10.3	~	~	~	4.4	~	~	0	~
Mellon	~	~	~	~	~	~	~	~	~	~	~	~	0

Now that we have our parameters described, we can begin describing our model. In the main problem, our decision variables are the number of printers we put at each location, $\vec{X}$. Our constraints are that the number of printers at each location must lie between the minimum and maximum number of printers at each location and that the total number of printers at all locations must equal the number of printers we have been given to allocate.

Within each iteration of the main problem, our decision variables for our linear programming are the number of students allocated to each printer from each set of students who have a specific printer as their preferred printer. In other words, our decision variables can be represented by a matrix Y where each element yij represents the number of students with preferred printer i who go to location j to print. Our sole constraint is that the full demand from each printer must be fully allocated.

Our cost function has two parts. The first part of the cost function is a cost associated with the expected congestion at each printer. The second part of the cost function is a cost associated with re-allocating students away from their preferred printer. This cost is proportional to the number of students re-allocated and to the distance from their preferred printer to the printer that they are allocated to.

Mathematically we can represent our main problem as follows (L is the set of all locations):

$$\mathrm{pmin}_i \leq x_i \leq \mathrm{pmax}_i \quad i \in L$$

$$\sum_{i \in L} x_i = p_{total}$$

Our second problem:

$$Z = \sum_{i \in L} \left(C\left(\frac{\sum_{j \in L} y_{ij}}{p_i}\right)\right) + \alpha \sum_{i \in L} \sum_{j \in L} p_{ij}\, y_{ij}$$

$$\sum_{i \in L} y_{ij} = d_i$$

In this model, Z is our cost function that we are trying to minimize, both in the main problem and in the sub-problem. After we iterate through all feasible solutions of the main problem, we take the solution that had the smallest Z after solving for our demand allocation. The function C is a concave function to represent the congestion cost, ideally a piecewise linear function to keep this a linear programming problem. α meanwhile is a constant that relates the re-allocation cost to the congestion cost. Note that for students who are allocated to their preferred printer (i.e. elements of P where i=j), their re-allocation cost is zero. This makes intuitive sense, since there is no downside to students who get to use their preferred printer.

Model Selection

In the end, we did not choose to implement or code this secondary model. This model has a few downsides, all of which would have required more time to deal with. The first problem with our model is that many of the parameters are subjective or difficult to estimate. To get an accurate feel for "true demand" would require more extensive data. Because of the great variety of printing needs amongst students on campus (especially between students from different colleges), it is imperative to have a large sample size of student opinions of printer congestion. Furthermore, if our survey receives too many responses from students of a certain college, this can heavily skew our analysis of which printers are currently experiencing congestion.

Second, our congestion cost function C and our relating constant α are both very subjective. In the end, while we are trying to minimize student inconvenience, this is not an easily measured quantity. It is difficult to model the structure of congestion costs, though this model is certainly a step up from our previous model. However, we introduce the potential of misevaluating the relationship between congestion costs and re-allocation costs. The two quantities have no clear mathematical relationship, so it is up to us to arbitrarily set the constant α to relate the two costs. If we set α too high, we risk over-prioritizing re-allocation costs and vice-versa.

Finally, we have execution barriers. Iterating through all feasible solutions of the main problem is already computationally taxing when simply evaluating an explicit function. Having a linear programming problem within a brute force problem is potentially unfeasibly taxing computationally.

The advantage of the second model is that it gives us a more detailed and far more accurate representation of student allocation as opposed to our previous model which makes many simplifying assumptions. However, due to run-time considerations, the secondary model could not be implemented in an effective fashion as a brute force algorithm where the objective function was itself a linear optimization problem.

Implementation of Model

The implementation of the model was done through Java, where the code is attached at the end of this report in an appendix. We were able to effectively code a brute force evaluation algorithm by noting that there are 32 printers at 13 locations, with the constraints of at least one printer at each location and some locations with upper bound constraints. This means that the number of possibilities is less than the number of ways to reallocate 19 excess printers among 13 locations which is simply $\binom{19+12}{19} = \binom{31}{19} = 141120525$. A number of possibilities which is easy to deal with for a computer and a relatively simple objective function, and when we take into account the local upper bounds on the number of printers, the actual number of possibilities is much lower.

Our approach to cycle through these possibilities was to go through every possible number of printers at each location with nested for-loops, check the viability of the allocation (ie. that they sum to 32 printers), and then calculate an objective function value and keep track of a running best allocation. This runs through about 3317760 possibilities. The run-time was negligible and took under five seconds.

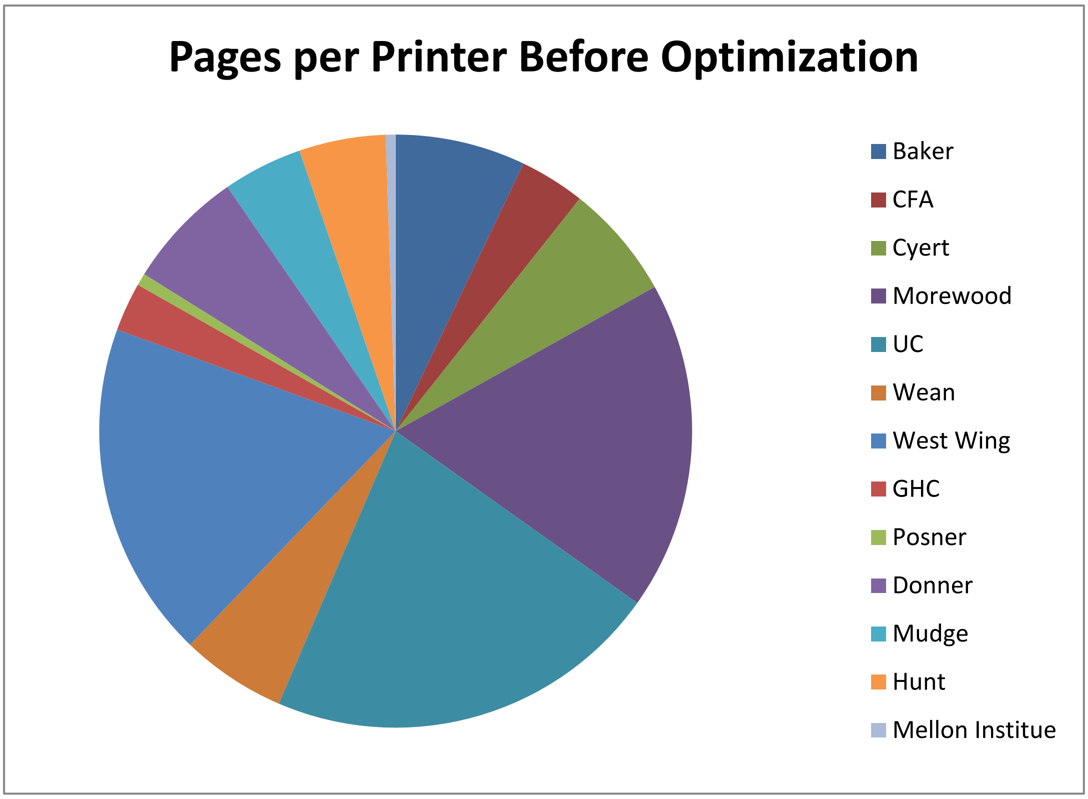
Result and Implementation of Solution

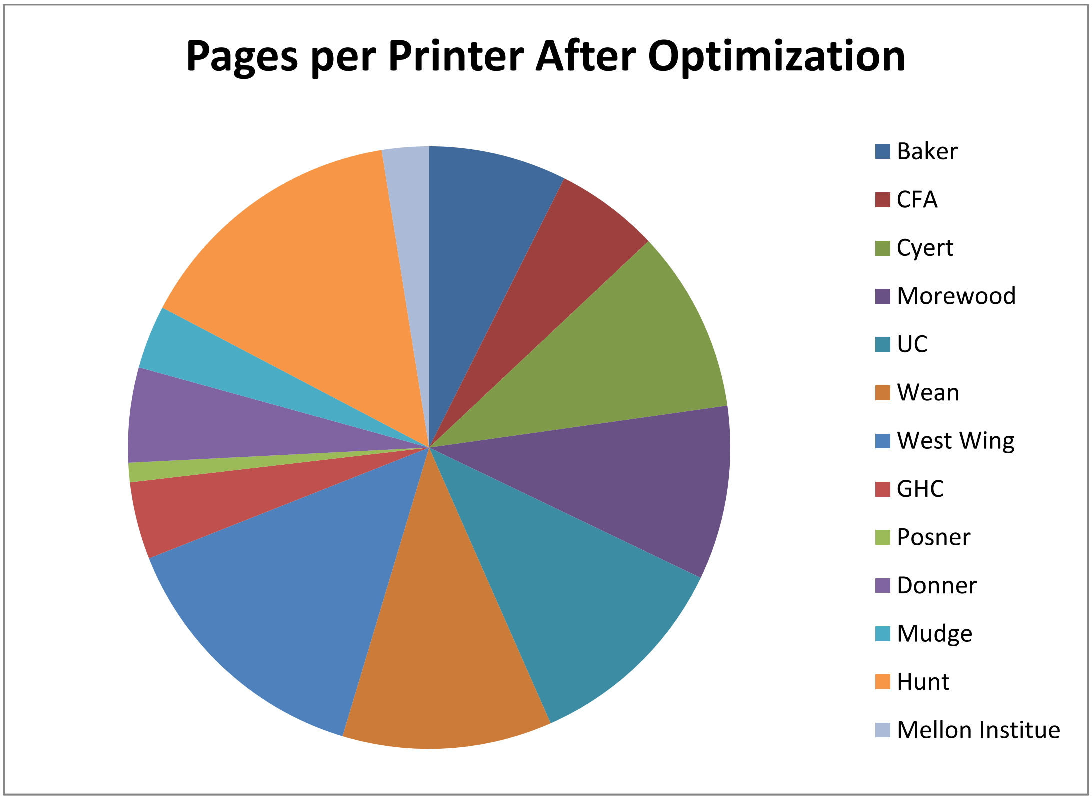
After running the algorithm we described above, we obtain the following result:

Table 3: Final Results

Location	Number of Printers	After Reallocation	Change
Baker	2	3	+1
CFA	2	2	0
Cyert	2	2	0
Morewood	1	3	+2
UC	1	3	+2
Wean	5	4	-1
West Wing	1	2	+1
GHC	2	2	0
Posner	1	1	0
Donner	1	2	+1
Mudge	1	2	+1
Hunt	10	5	-5
Mellon Institute	3	1	-2

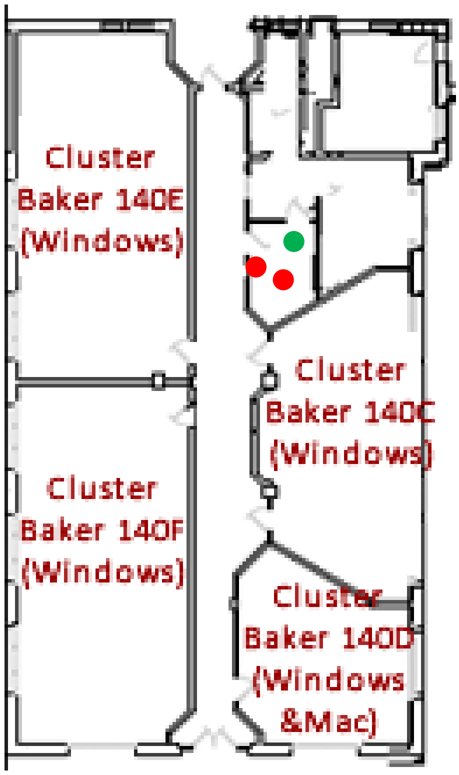
From the result, we can see that we want to add printers in Baker, Morewood, University Center, West Wing, Donner and Mudge. These places are all popular printing locations with limited number of printer available. And we want to remove printers from Wean, Hunt and Mellon Institute. Wean and Hunt has excessive number of printers because of the two libraries. However these library printers are not fully used, it is reasonable to move printers to places with higher demand. We can all see the new allocation will create a more balanced printing system through the Pages per Printer pie chart before and after the optimization. Before the reallocation, the three most popular printing locations, Morewood, University Center and West Wing have much more pages printed per printer compared with other places, but after the reallocation the difference of pages per printer between locations are much smaller.


Pages per Printer Before Optimization
Baker
CFA
Cyert
Morewood
UC
Wean
West Wing
GHC
Posner
Donner
Mudge
Hunt
Mellon Institue


Pages per Printer After Optimization
Baker
CFA
Cyert
Morewood
UC
Wean
West Wing
GHC
Posner
Donner
Mudge
Hunt
Mellon Institue

Here are the detailed implementation maps for some location where space is quite limited:

Baker

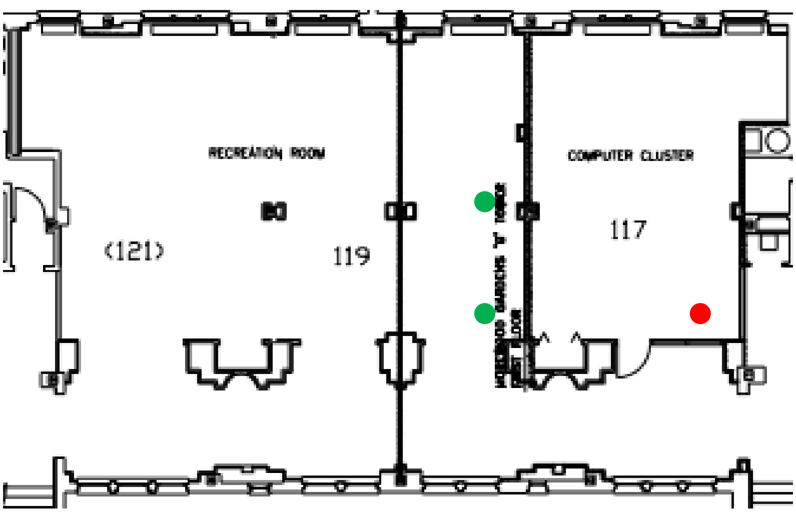


Existing Printer
Available Space

Current number: 2
Optimal number: 3

Adding Cost: There is a empty closet in the printing room, so removing the closet will make enough room for an extra printer. The cost of adding one printer is negligible.

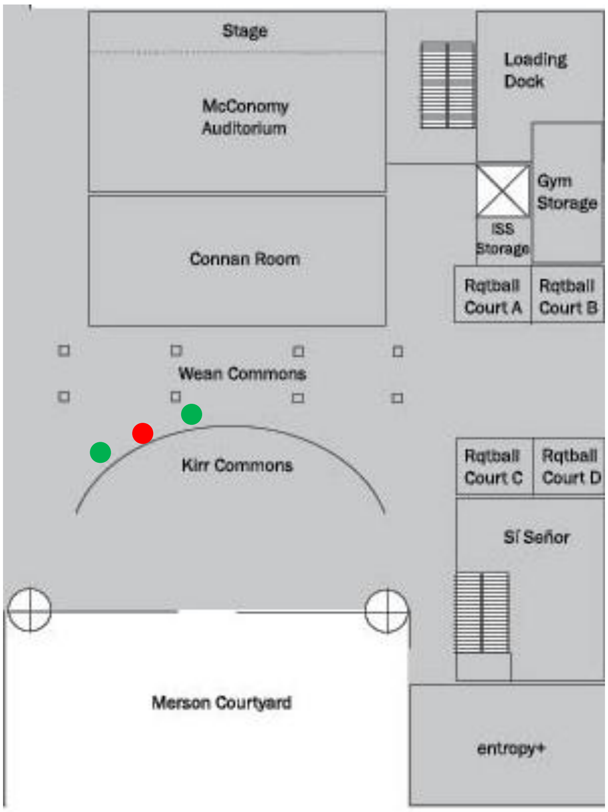
Morewood



Current number: 1
Optimal number: 3

Adding cost: There is no more available space in the cluster room, and removing existing computer is not a very good choice. However, there is plenty of space in the recreation room next door. We just need to rearrange the sofas to make space for two more printers, which is acceptable.

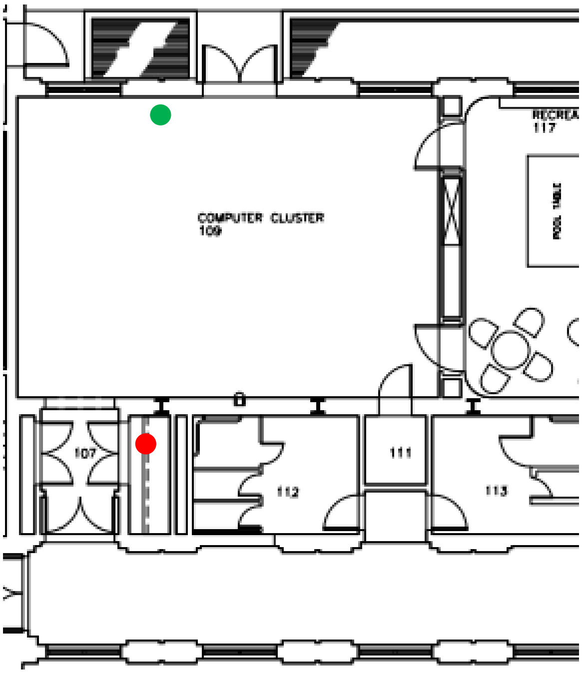
University Center



Current number: 1
Optimal number: 3

Adding cost: The space in University Center is very limited. In order to add two more printers, we need to remove a web station and a bench. But considering the extremely high demand in this location, we think that the cost is still within reasonable range.

West Wing



Current number: 1
Optimal number: 2

Adding cost: There is still some space in the cluster. It is possible to make space for one printer if we carefully rearrange the sofas. Even if we have to remove one sofa, the cost is still acceptable comparing with the benefit from an extra printer.

Conclusion/Discussion

Our final results are posted above, for the most part they seem very reasonable and follow our intuition regarding the allocation of more printers at places with the highest pages per printer. In general, there seemed to be a trend for less printers at academic institutions like Hunt library and Wean Hall, and more printers at dormitories such as West Wing, Morewood and Muge. After the reallocation suggested we do see that the pages per printer are distributed more evenly which indicate that pritners are being used more efficiently.

There are a few areas where our model could improve. One thing it does not take into account certain possibilities such as removing possible locations or adding new ones at locations where there are none, such as Stever House or the Hill. This would require something similar to the secondary model and a much greater understanding of the underlying natural demand at each location – this kind of information would have to be obtained through intensive surveys and require more time. Another problem that was not covered here would be the possibility of adding or removing some number of printers to the system, this complicates the problem somewhat more as there would have to be an assessment of comparing the price of printers to the 'price' of a long queue. Color printing was also not considered as it has a much more limited scope. In the future, however, these considerations are good candidates for extending this model's applicability and relevance if given additional resources and time.

Overall, we feel that the model at hand has given relevant considerations as to the reallocation of printers for Carnegie Mellon University. The same or similar techniques could be used to apply this to other universities' printing systems or even other systems alltogether, such as the placement of checkout kiosks in a sprawling market. Collectively, we look forward to using and applying those lessons learned here to springboard our investigation into future operations research problems.

Appendix: Algorithm Code

```
// Operation Research 2
// Printing Project
// Group D


import java.util.Arrays;


public class OptimizePrinters21{

 /**
  * @param args
  */
  public static void main(String[] args) {
  int[] printers = new int[13];

  double currentCost = 0;
  double optimal = 100000000;
  int[] solution = new int[13];
  // manually set the percentage of students who are affected by
printer location
  double redistP = .13;
  double totalPrinters = 32;

  int totalDemand = 13315684;
  // calculate coefficient b
  double b = (totalDemand*redistP)/(totalPrinters*1.0);

  // the current demand in each location
  double[] demand = new double[13];
  demand[0] = 1007655;
  demand[1] = 506146;
  demand[2] = 888371;
  demand[3] = 1271963;
  demand[4] = 1534213;
  demand[5] = 2041168;
  demand[6] = 1303271;
  demand[7] = 376843;
```

```
  demand[8] = 47151;
  demand[9] = 462603;
  demand[10] = 308262;
  demand[11] = 3354570;
  demand[12] = 213468;

  double[] a = new double[13];

  // Baker
  for(printers[0] = 1; printers[0] <= 6; printers[0]++){
    // CFA
    for(printers[1] = 1; printers[1] <= 4; printers[1]++){
      // Cyert
      for(printers[2] = 1; printers[2] <= 4; printers[2]++){
        // Morewood
        for(printers[3] = 1; printers[3] <= 4; printers[3]++){
          // UC
          for(printers[4] = 1; printers[4] <= 4; printers[4]++){
            // Wean
            for(printers[5] = 1; printers[5] <= 6;
printers[5]++){
              // West Wing
              for(printers[6] = 1; printers[6] <= 2;
printers[6]++){
                // GHC
                for(printers[7] = 1; printers[7] <= 6;
printers[7]++){
                  // Posner
                  for(printers[8] = 1; printers[8] <= 3;
printers[8]++){
                    // Donner
                    for(printers[9] = 1; printers[9] <= 2;
printers[9]++){
                      // Mudge
                      for(printers[10] = 1; printers[10] <= 2;
printers[10]++){
                        // Hunt
```

```
                        for(printers[11] = 1; printers[11] <= 10;
printers[11]++){
                          // Mellon Institue
                          for(printers[12] = 1; printers[12] <=
3; printers[12]++){
                            if((printers[0] + printers[1] +
printers[2] + printers[3] + printers[4] + printers[5]
                                  + printers[6] + printers[7] +
printers[8] + printers[9] + printers[10] + printers[11] +
printers[12]) == 32){
                              currentCost = 0;
                              // find the demand fuction and
calculate the cost
                              for (int i = 0; i < 13; i++){
                                a[i] = (demand[i] -
b*printers[i]);

                                currentCost +=
(a[i]/printers[i]);
                              }
                              // check for optimality
                              if (currentCost < optimal) {
                                optimal = currentCost;
                                for (int i = 0; i < 13; i++){
                                  solution[i] = printers[i];
                                }
                              }
                            }
                          }
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
```

```
        }
    }
    System.out.println("Optimal Allocation of Printers:");
    System.out.println("Baker: " + printers[0]);
    System.out.println("CFA: " + printers[1]);
    System.out.println("Cyert: " + printers[2]);
    System.out.println("Morewodd: " + printers[3]);
    System.out.println("UC: " + printers[4]);
    System.out.println("Wean: " + printers[5]);
    System.out.println("West Wing: " + printers[6]);
    System.out.println("GHC: " + printers[7]);
    System.out.println("Posner: " + printers[8]);
    System.out.println("Donner: " + printers[9]);
    System.out.println("Mudge: " + printers[10]);
    System.out.println("Hunt: " + printers[11]);
    System.out.println("Mellon Institue: " + printers[12]);
    }
}
```