Author: Tim Baron

December 17, 2018

Ambulance Position Optimization

Short Abstract:

This project attempts to optimize ambulance positions in relation to both random and chosen demand points. The goal is to determine the most ideal positions for the ambulances. Microsoft Excel is used to create a non-liner programming to solve the problem, but in the end could not solve it.

Medium Abstract:

There are a limited number of ambulances on duty to serve demand points at any given time. In this project both randomly chosen and carefully chosen demand points are used to run a non-linear optimization program to determine the most effective placement of ten ambulances in an area that is 80 miles by 80 miles.

Microsoft Excel is used to create the non-linear program. Extensive testing and debugging was done using demand points that had obvious optimal solutions. Various initial starting points for the ambulances were used. Solver was unable to reliable return the obvious results. The problem is non-convex because we can switch sets of ambulances and get the same objective function value. The standard Solver is likely unsuitable for this problem because of non-convexity, however multi-start Solver may produce better results.

Introduction:

There are a limited number of ambulances available in any given area. In order to best serve demand points, ambulances must be optimally positioned to respond to emergencies. The goal of this project is to create a non-linear program to determine the optimal positions for the ambulances to be located to best serve the demand points.

An artificial area will be created for analysis. Randomly chosen demand points will be created to simulate a realistic situation. Excel will be used for all calculations.

The main goals of this project are:

- Create a non-linear program to find the solution.
- Alter the environment to debug and test the program.
- Determine whether Excel Solver can find a solution.
- Determine the optimized solution.

Methods, Models, and Calculations:

Software Used:

Excel Solver will be used for the analysis.

Creation of the Artificial Area:

This project analyzes an artificially created area that is 80 miles by 80 miles, has 50 randomly chosen demand points, has demand points that are weighted by population, has 10 ambulances available, and has a total population of 2,585,576 people. The 50 demand points each have an **x** and **y** coordinate and were determined using the following Excel functions: for **x** coordinates: =rand()*horizontal spread, for **y** coordinates: = rand()*vertical spread.

The **weights** were determined using the Excel function: =rand()*100,000. The total population is the sum of the weights. The data for the demand points and weights was then frozen by copying it to another sheet.

Below is an example of the first 10 demand points with their weights:

Demand Point	demand x	demand y	Population
1	46.8	13.6	9987
2	22.6	72.4	89785
3	32.6	21.7	64736
4	23.7	19.3	32005
5	74.2	61.4	74923
6	69.9	28.0	90977
7	30.4	19.6	28347
8	65.5	3.2	90813
9	30.9	41.1	19815
10	23.1	78.9	3995

Methods/Model:

The model finds the distances for the closest and second closest ambulance to a given demand point. The second closest ambulance is used in the case that we want to assume a backup ambulance is necessary because the primary ambulance is already handling an emergency. The minimum distance is then multiplied by the demand point population to determine the weighted distance of the demand point.

The 10 ambulance positions are the decision variables for the program and are listed as such (random positions in this case):

Ambulance x =	0.0	18.5	24.1	43.6	22.6	48.9	19.5	9.0	68.7	65.5
Ambulance y =	36.1	8.2	20.4	13.2	72.4	52.8	54.4	3.6	59.7	13.2

To determine the minimum distance, taxicab distance formulas are used as follows:

=abs(demand point **x** – ambulance **x**)^p + (demand **y** – ambulance **y**)^p

An exponent cell, “p”, was added to give the option of calculating the squared taxicab distances. Using squared distances makes the problem easier for Solver to solve.

An example of these calculations is as follows:

Demand Point	demand x	demand y	Population	Ambulance 1	Ambulance 2	Ambulance 3	....	Ambulance 10
1	46.8	13.6	9987	75365.3	75365.3	75365.3		75365.3
2	22.6	72.4	89785	99657.1	99657.1	99657.1		99657.1
3	32.6	21.7	64736	59334.4	59334.4	59334.4		59334.4
4	23.7	19.3	32005	84022.3	84022.3	84022.3		84022.3
5	74.2	61.4	74923	8516.3	8516.3	8516.3		8516.3
6	69.9	28.0	90977	74040.1	74040.1	74040.1		74040.1
7	30.4	19.6	28347	37.4	37.4	37.4		37.4
8	65.5	3.2	90813	32237.4	32237.4	32237.4		32237.4
9	30.9	41.1	19815	30739.8	30739.8	30739.8		30739.8
10	23.1	78.9	3995	76989.8	76989.8	76989.8		76989.8

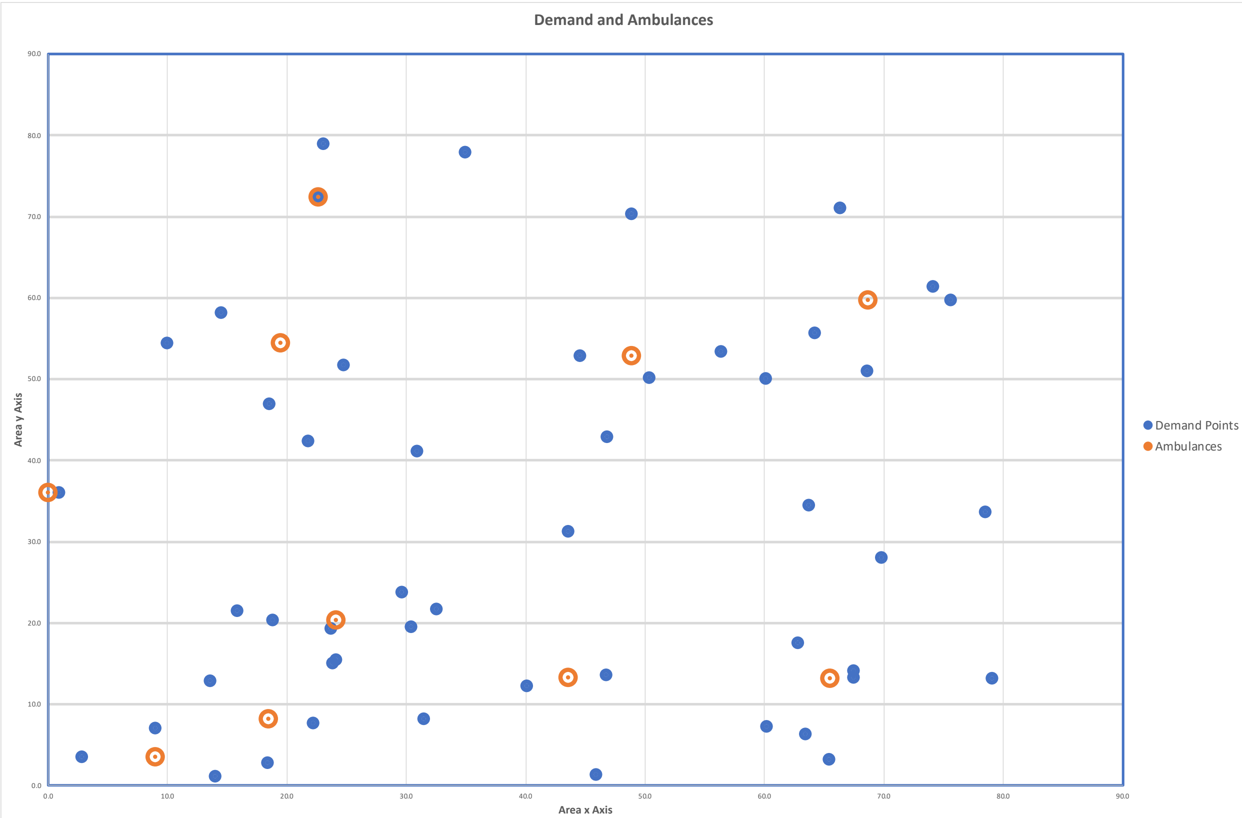
Demand Point	Min Distance Ambulance (No Weight)	Min Distance 2nd Ambulance (No Weight)	Weighted Min Distance Ambulance	Weighted Min Distance 2nd Ambulance	Best Ambulance	Backup Ambulance
1	3.5	19.1	35069.6	191133.8	4	10
2	0.0	21.1	508.5	1898535.1	5	7
3	9.8	19.5	635305.2	1263799.5	3	4
4	1.5	16.3	48065.8	523126.5	3	2
5	7.2	33.8	537457.8	2535965.9	9	6
6	19.2	32.8	1748301.9	2988288.9	10	9
7	7.1	19.5	200116.3	553292.0	3	4
8	10.0	31.9	904040.1	2895113.6	10	4
9	24.7	27.5	490153.6	545520.3	7	3
10	7.0	28.1	27940.3	112440.4	5	7

The primary minimum ambulance distances were determined using the Excel =min() function. The backup ambulance distances were determined using the =small(array of all distances, 2^{nd} highest value).

We do not have decision variables in which ambulances are assigned to demand points. The decision variables do not tell us which demand points are covered by which ambulances. The assignments in this project are determined by which ambulance is closest and second closest as the (x,y) location of each ambulance is moved during the optimization calculation.

The identity of the best (primary) and backup ambulances were determined using the Excel formula: =match(minimum distance sought, array of ambulance distances, [match_type zero for exact match]). Identifying these ambulances was for the purpose of determining ambulance load, not optimization.

The points are also graphed to provide a visual representation of both the demand points and ambulances in the area. An example of the graph of a solution is as follows.



The objective function of the non-linear program is the sum of the column of weighted minimum primary ambulance distances added to an assignable percentage multiplied by the sum of the column of weighted minimum distances for the backup ambulance. This allows us to determine the minimized distance for just the primary ambulances and add the distances of backup ambulances if we choose to be assigning a chosen percentage (our options in the project were 0%, 5%, and 10%). Adding the percentages for the backup ambulances gives us an indication of the nonlinearity in the problem. Increasing the percentage above zero increases the objective function value. Each sum is divided by the total population, which gives us the weighted mean distance between each demand point and its ambulance, which makes it easy to make sense of the optimized value calculated by Solver. This also allows us to easily compare the answer to the mean, min, max, median, and standard deviation.

The formula for the objective function is:

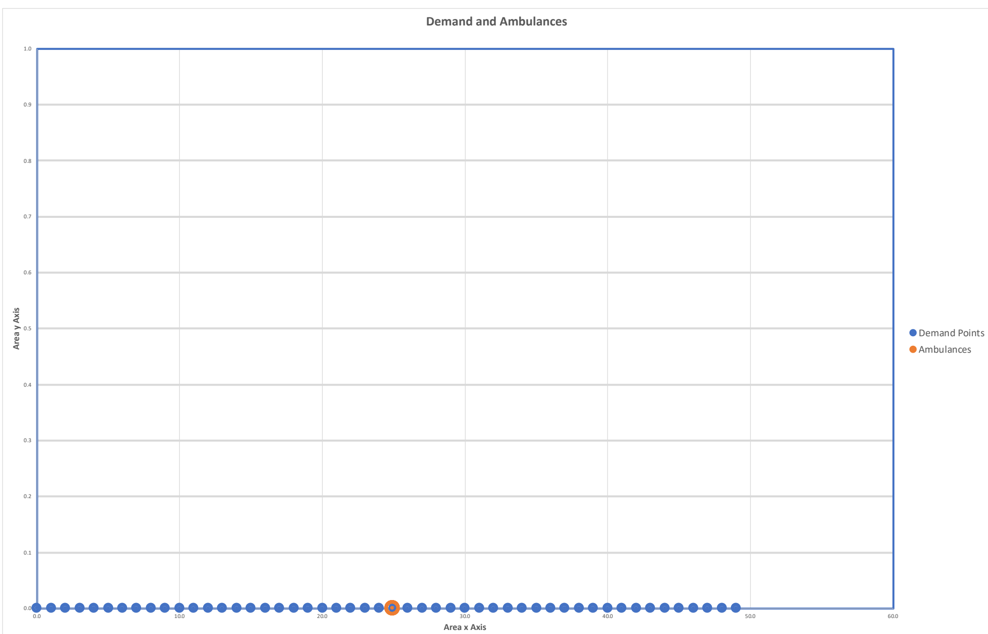
=sum(column of primary ambulance minimum distances)/(total population) + (chosen percentage)*sum(column of backup ambulance minimum distances)/(total population)

No constraints were used.

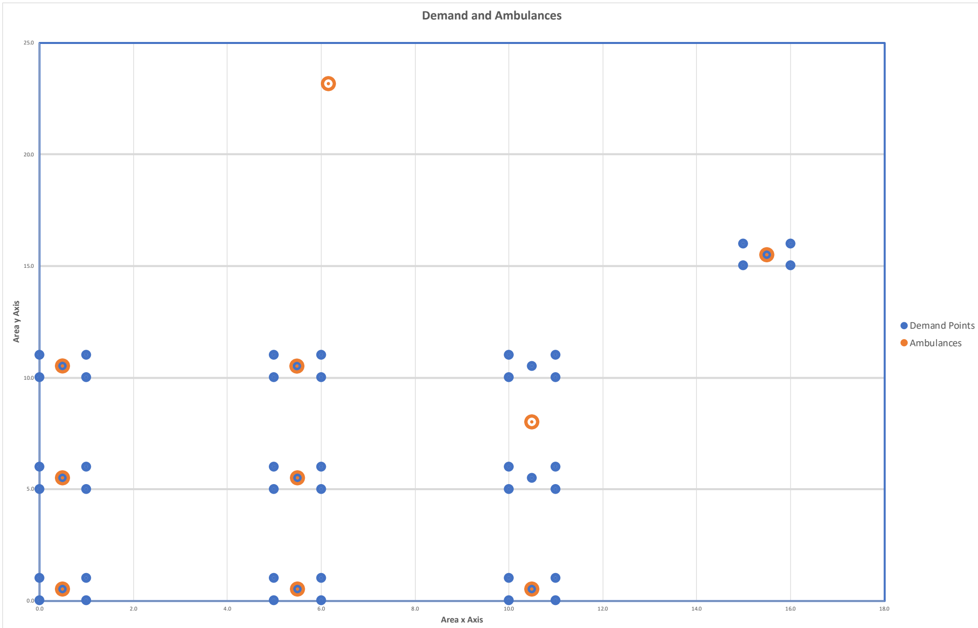
<u>Alternate Versions of the Sheet for Testing/Debugging:</u>

Alternate versions of the sheet were created using non-random demand points. These sheets allowed for debugging/testing of the non-linear program with obvious/expected solutions.

One sheet had demand points arranged in a horizontal line along the x axis using demand points (0,0), (1,0), (2,0), ……. (50,0). We expect a solution of (x = 2, 7, 12, 17, 22, 27, 32, 37, 42, 47 and all y = 0)



The other sheet had demand points arranged in 10 clusters.



Calculations, Tests, and Results:

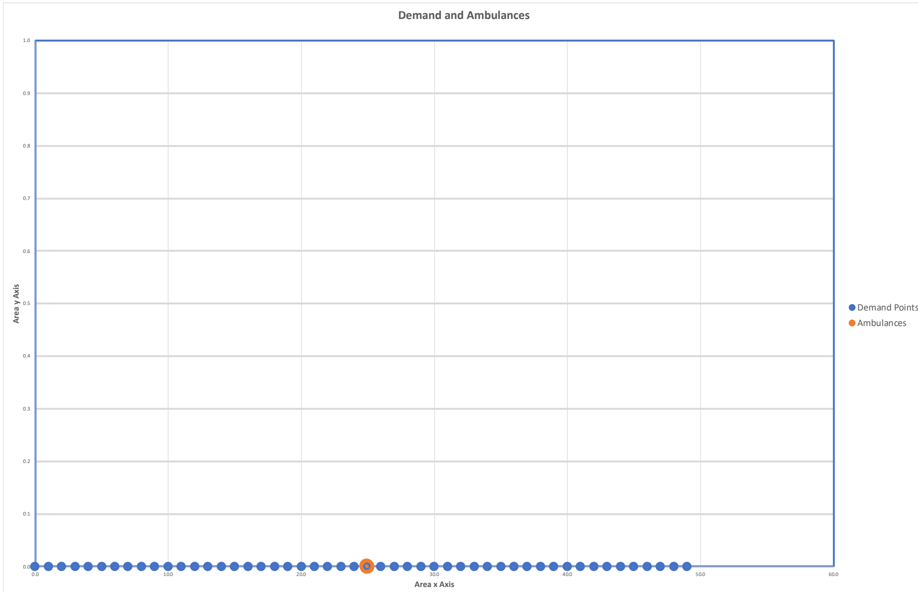
Various starting positions for the 10 ambulances were used for testing the program. Several tests were run using the alternate sheets to determine if Solver could find the expected answers. Both GRG Nonlinear and Evolutionary solving methods were tested. The Evolutionary solving method never produced acceptable results for this project. In an attempt to get better results Solver was run first with the taxicab distances squared (see formula above) and then those results were left as the decision variables and Solver was re-run with an exponent of ^1. This did not provide sufficiently improved results. When Solver was started with optimal solutions in the tests, it returned the same optimal solutions.

Tests Starting with All Ambulances at the Origin:

These tests were started with all ambulances at the origin. In each test case in which all ambulances were started at the origin non-optimal results were reached.

In the case of the horizontal line on the x-axis alternate sheet, Solver appears to give a saddle point with all ambulances near the halfway point returning values of:

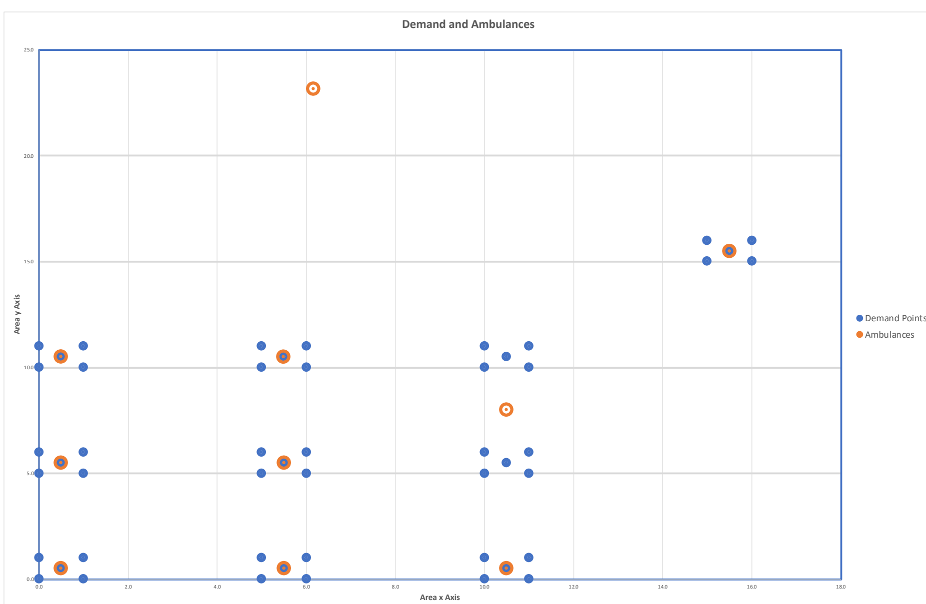
(24.9, 0.0), (24.9, 0.0), (24.9, 0.0), (24.9, 0.0), (24.9, 0.0), (24.9, 0.0), (24.9, 0.0), (24.9, 0.0), (24.9, 0.0), and (24.9, 0.0) for ambulances one through ten, respectively.



obj. function: 12.5

In the case of the clustered alternate sheet (in which case it would seem the ambulances would be optimized with an ambulance at the center of each cluster), Solver returned:

(0.5, 0.0), (5.5, 0.5), (5.5, 5.5), (0.5, 5.5), (10.5, 0.5), (10.5, 8.0), (5.5, 10.5), (0.5, 10.5), (6.2, 23.1), and (15.5, 15.5) for ambulances one through ten, respectively.



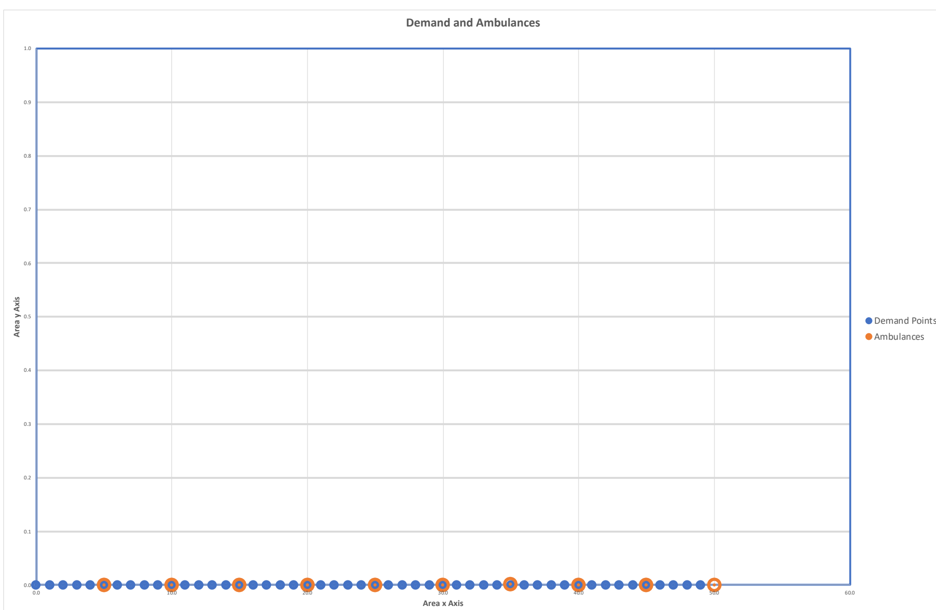
obj. function:	1.220002

Tests Starting with Ambulances Starting at a Horizontal Line Through the Area:

These tests were started with all ambulances in a horizontal line through the area. These results were better, but still not optimal.

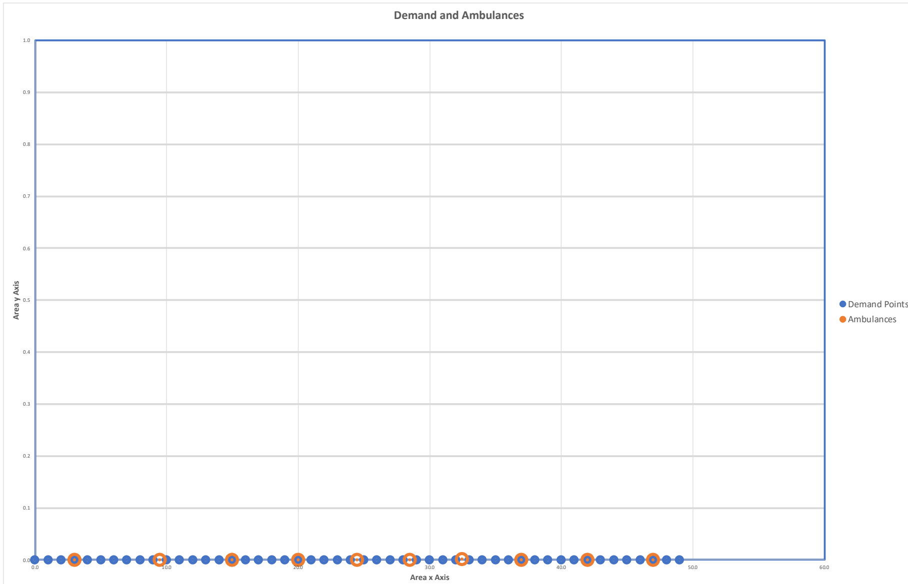
Start (horizontal line on x-axis alternate sheet):

(5.0, 0.0), (10.0, 0.0), (15.0, 0.0), (20.0, 0.0), (25.0, 0.0), (30.0, 0.0), (35.0, 0.0), (40.0, 0.0), (45.0, 0.0), and (50.0, 0.0) for ambulances one through ten, respectively.



In the case of the horizontal line on the x-axis alternate sheet, Solver returned:

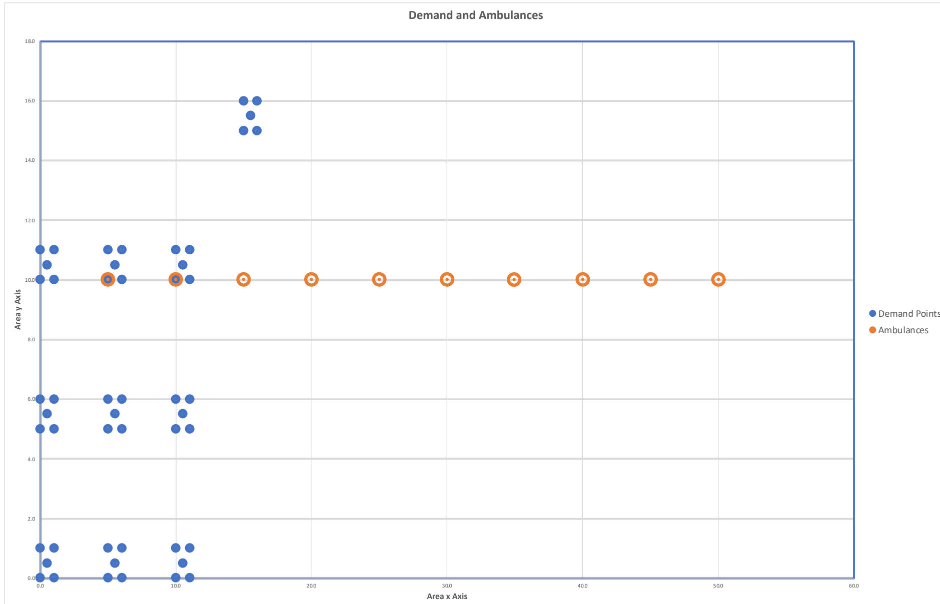
(3.0, 0.0), (9.5, 0.0), (15.0, 0.0), (20.0, 0.0), (24.5, 0.0), (28.5, 0.0), (32.5, 0.0), (37.0, 0.0), (42.0, 0.0), and (47.0, 0.0) for ambulances one through ten, respectively.



obj. function:	1.4

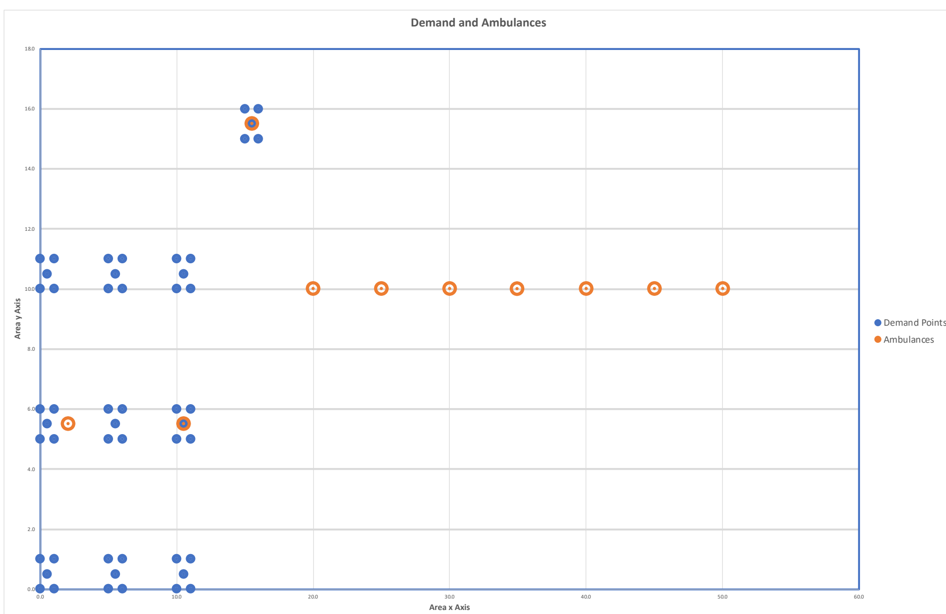
Start (horizontal line on clustered alternate sheet):

(5.0, 10.0), (10.0, 10.0), (15.0, 10.0), (20.0, 10.0), (25.0, 10.0), (30.0, 10.0), (35.0, 10.0), (40.0, 10.0), (45.0, 10.0), and (50.0, 10.0) for ambulances one through ten, respectively.



In the case of the horizontal line on the clustered alternate sheet, Solver returned:

(2.0, 5.5), (10.5, 5.5), (15.5, 15.5), (20.0, 10.0), (25.0, 10.0), (30.0, 10.0), (35.0, 10.0), (40.0, 10.0), (45.0, 10.0), and (50.0, 10.0) for ambulances one through ten, respectively.

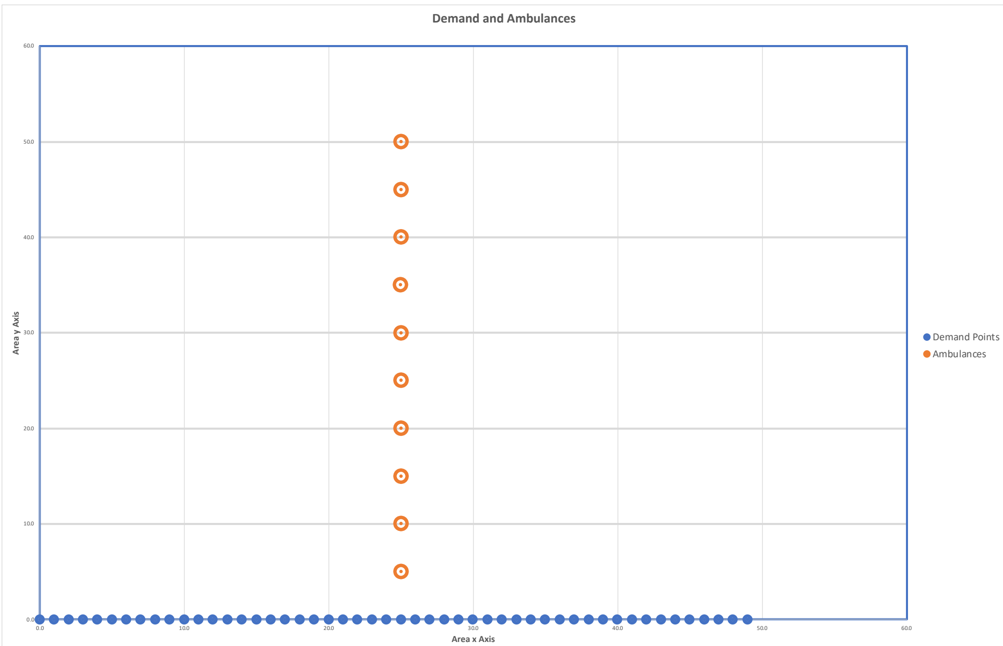


obj. function:	**4.820002**

<u>Tests Starting with Ambulances Starting at a Vertical Line Through the Area:</u>

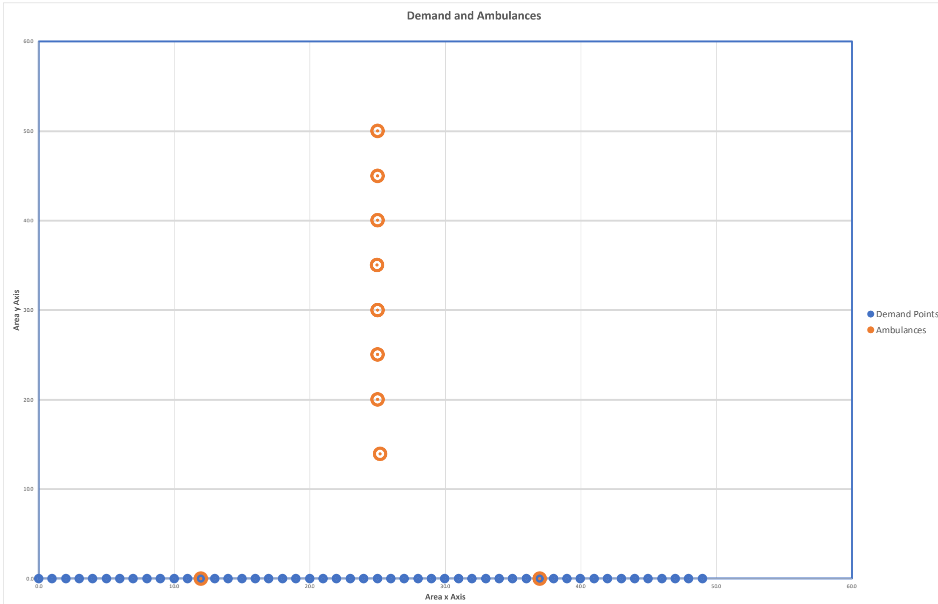
Start (vertical line on x-axis alternate sheet):

(25.0, 5.0), (25.0, 10.0), (25.0, 15.0), (25.0, 20.0), (25.0, 25.0), (25.0, 30.0), (25.0, 35.0), (25.0, 40.0), (25.0, 45.0), and (25.0, 50.0) for ambulances one through ten, respectively.



In the case of the vertical line on the x-axis alternate sheet, Solver returned:

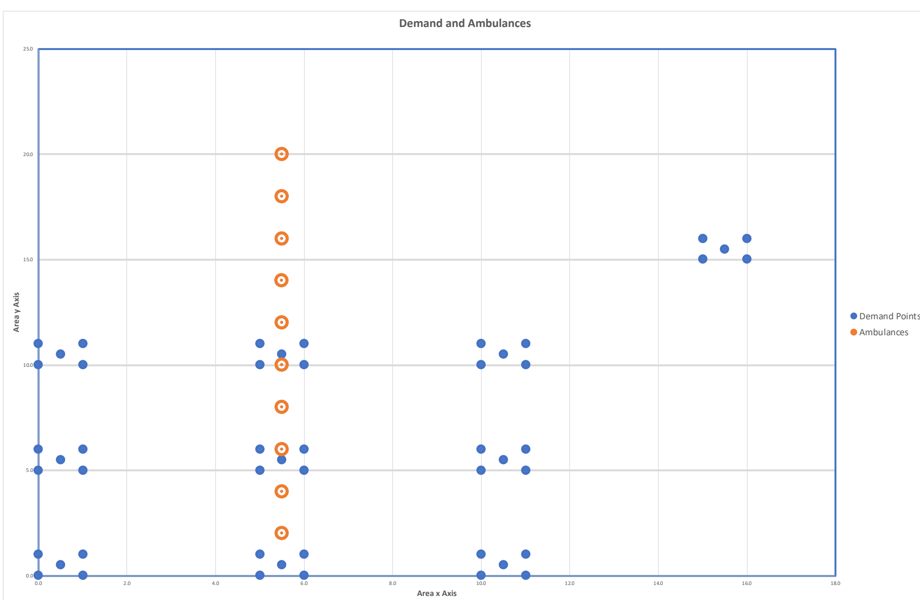
(12.0, 0.0), (37.0, 0.0), (25.2, 13.9), (25.0, 20.0), (25.0, 25.0), (25.0, 30.0), (25.0, 35.0), (25.0, 40.0), (25.0, 45.0), and (25.0, 50.0) for ambulances one through ten, respectively.



obj. function: 6.2

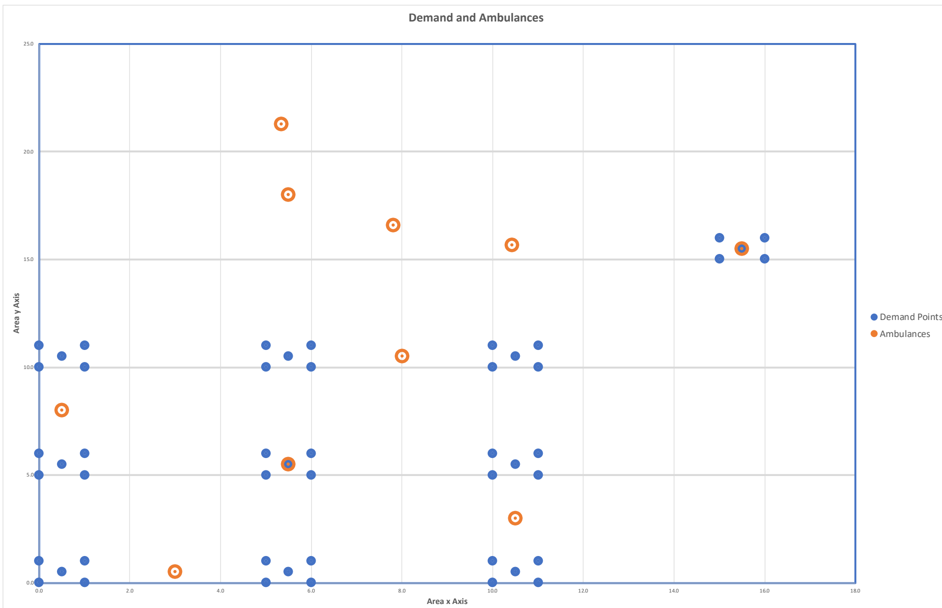
Start (vertical line on clustered alternate sheet):

(5.5, 2.0), (5.5, 4.0), (5.5, 6.0), (5.6, 8.0), (5.5, 10.0), (5.5, 12.0), (5.5, 14.0), (5.5, 16.0), (5.5, 18.0), and (5.5, 20.0) for ambulances one through ten, respectively.



In the case of the vertical line on the clustered alternate sheet, Solver returned:

(3.0, 0.5), (10.5, 3.0), (5.5, 5.5), (0.5, 8.0), (7.8, 16.6), (8.0, 10.5), (10.4, 15.6), (15.5, 15.5), (5.5, 18.0), and (5.3, 31.3) for ambulances one through ten, respectively.

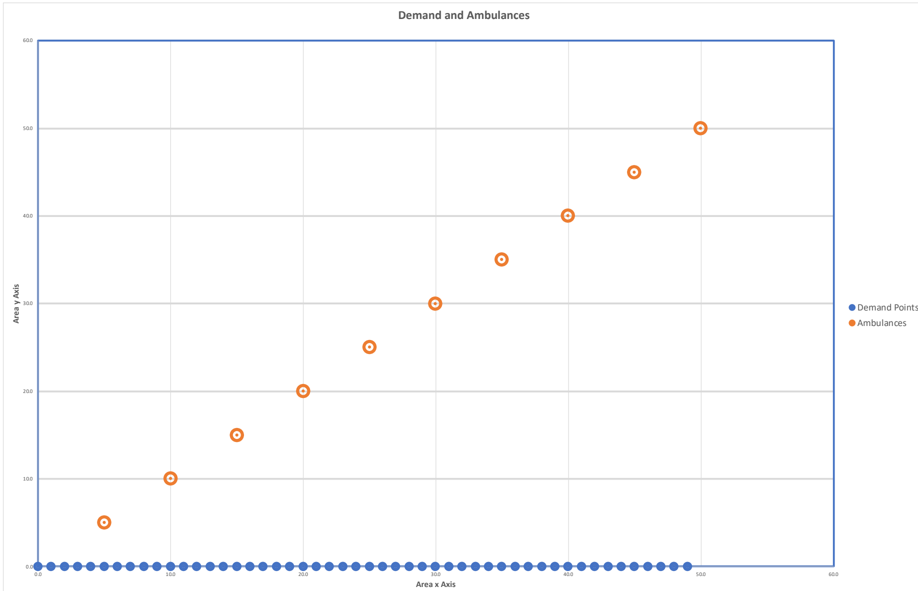


obj. function: 2.480001

<u>Tests Starting with Ambulances Starting at a Diagonal Line Through the Area:</u>

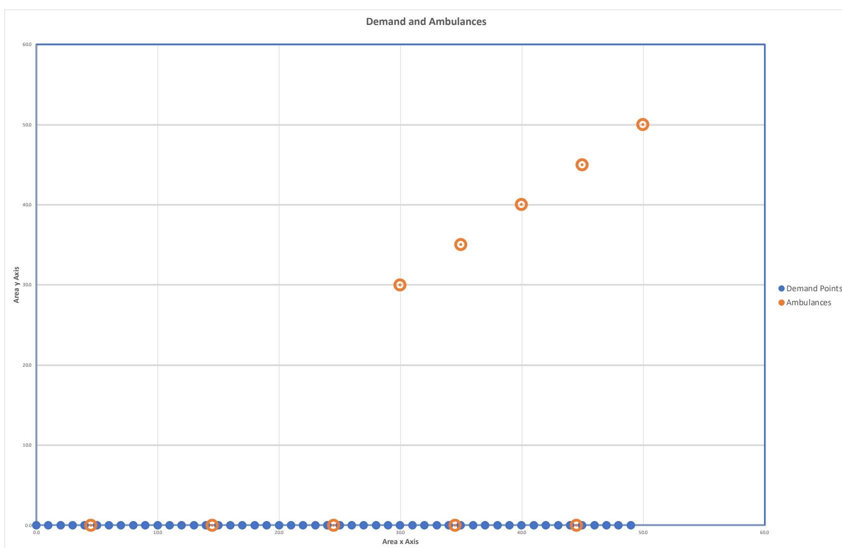
Start (diagonal line on x-axis alternate sheet):

(5.0, 5.0), (10.0, 10.0), (12.0, 15.0), (20.0, 20.0), (25.0, 25.0), (30.0, 30.0), (35.0, 35.0), (40.0, 40.0), (45.0, 45.0), and (50.0, 50.0) for ambulances one through ten, respectively.



In the case of the diagonal line on the x-axis alternate sheet, Solver returned:

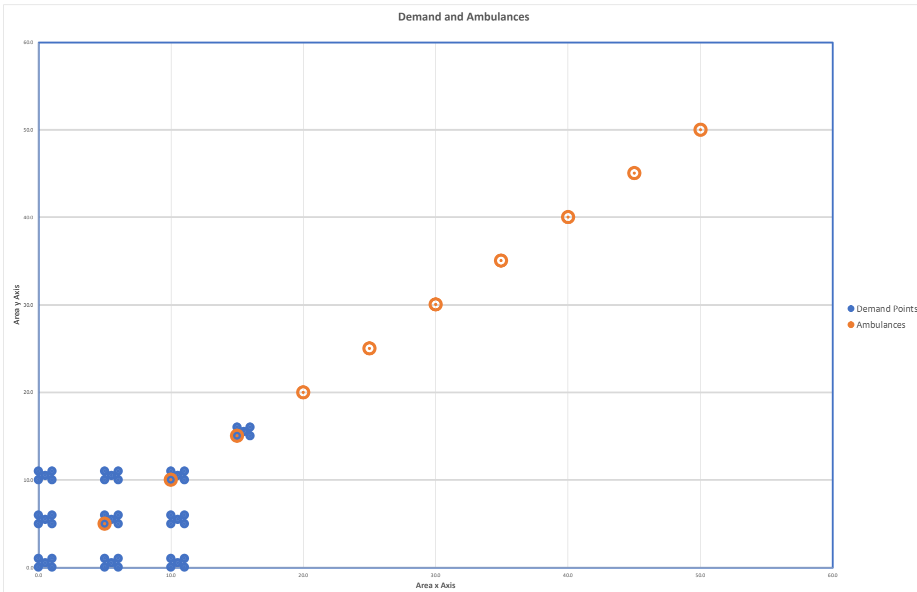
(4.5, 0.0), (14.5, 0.0), (24.5, 0.0), (34.5, 0.0), (44.5, 25.0), (30.0, 30.0), (35.0, 35.0), (40.0, 40.0), (45.0, 45.0), and (50.0, 50.0) for ambulances one through ten, respectively.



obj. function:	**2.5**

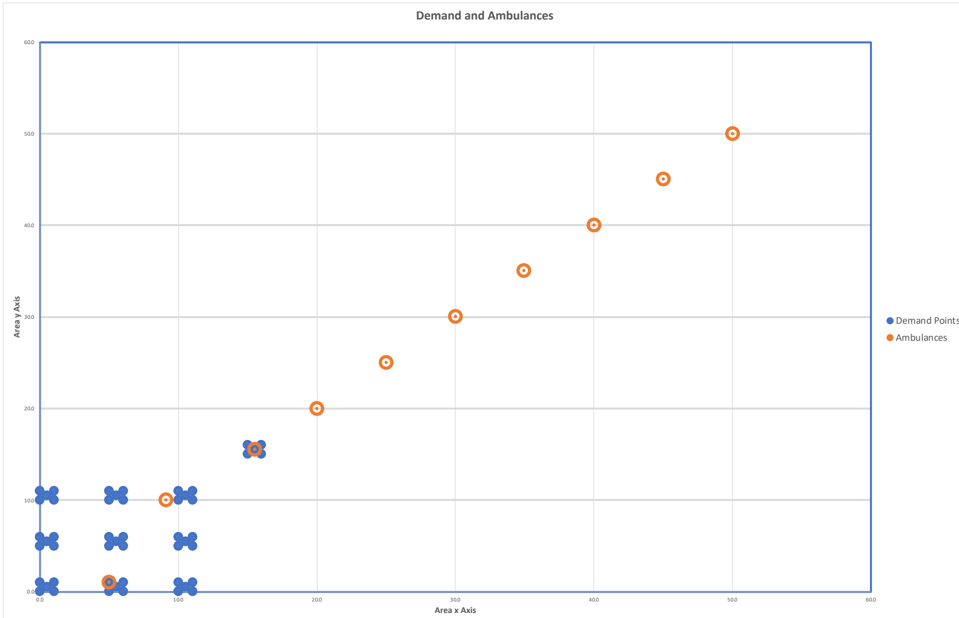
Start (diagonal line on clustered alternate sheet):

(5.0, 5.0), (10.0, 10.0), (12.0, 15.0), (20.0, 20.0), (25.0, 25.0), (30.0, 30.0), (35.0, 35.0), (40.0, 40.0), (45.0, 45.0), and (50.0, 50.0) for ambulances one through ten, respectively.



In the case of the vertical line on the clustered alternate sheet, Solver returned:

(5.0, 1.0), (9.1, 10.0), (15.5, 15.5), (20.0, 20.0), (25.0, 25.0), (30.0, 30.0), (35.0, 35.0), (40.0, 40.0), (45.0, 45.0), and (50.0, 50.0) for ambulances one through ten, respectively.



obj. function: 4.780655

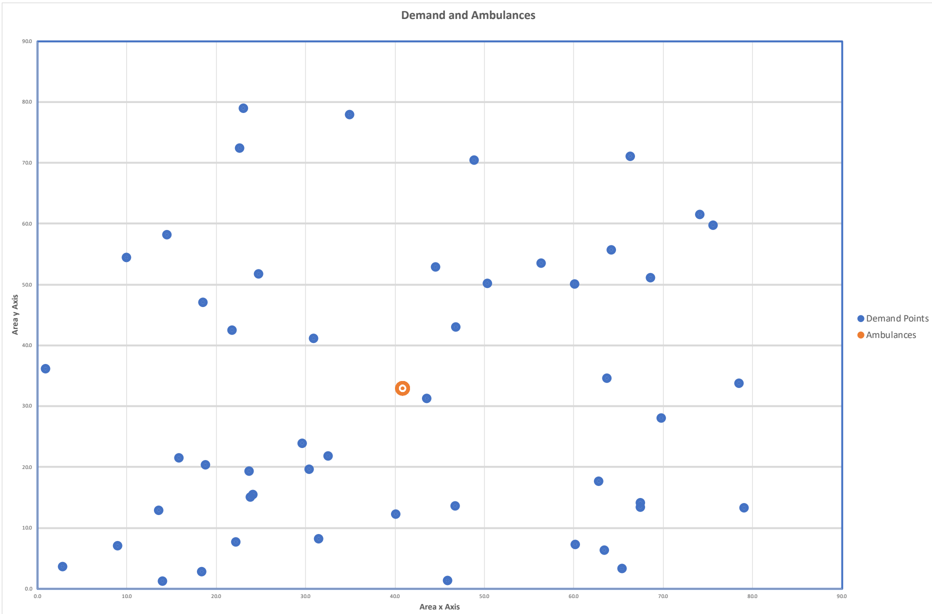
Testing of the Random Demand Points and Ambulances:

The debugs/tests did not prove reliable for returning optimal answers and we know the problem is non-convex because we can switch sets of ambulances and get the same objective function value. Solver is likely not working for this problem because of non-convexity. However, the main problem for this project must be run.

I decided to test the same four scenarios and see which provided the best objective function value.

The Origin Start Test Returned:

(40.9, 32.8), (40.9, 32.8), (40.9, 32.8), (40.9, 32.8), (40.9, 32.8), (40.9, 32.8), (40.9, 32.8), (40.9, 32.8), (40.9, 32.8), and (40.9, 32.8) for ambulances one through ten, respectively.

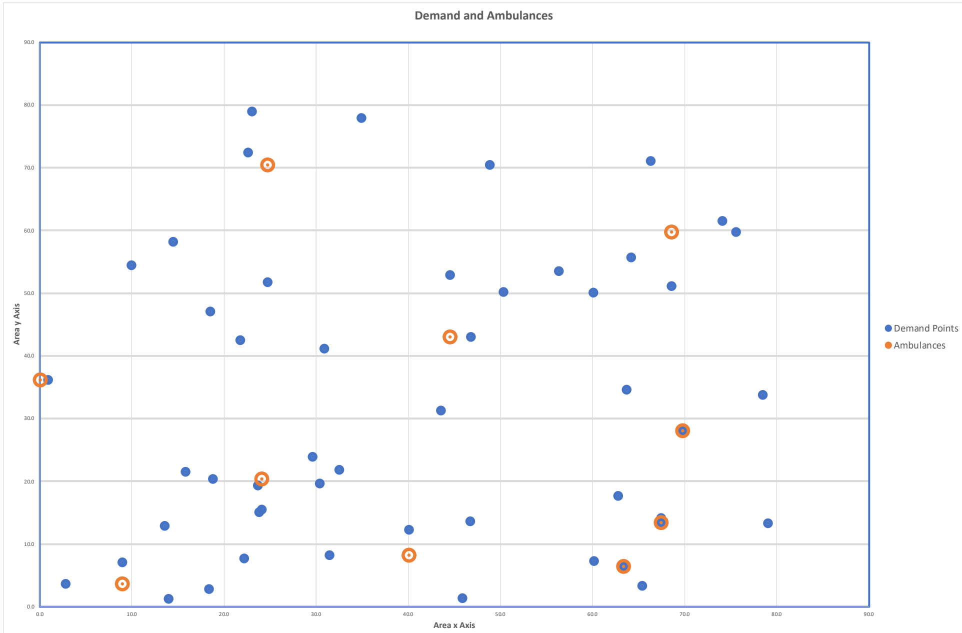


obj. function:	**41.4**

Max =	67.3
Median Distance =	37.9
Mean Distance =	40.3
SD =	14.5

The Horizontal Line Test Returned:

(0.1, 36.1), (9.0, 3.6), (40.1, 8.2), (24.1, 20.4), (44.6, 42.9), (24.8, 70.4), (63.5, 6.3), (67.5, 13.3), (69.9, 28.0), and (68.7, 59.7) for ambulances one through ten, respectively.

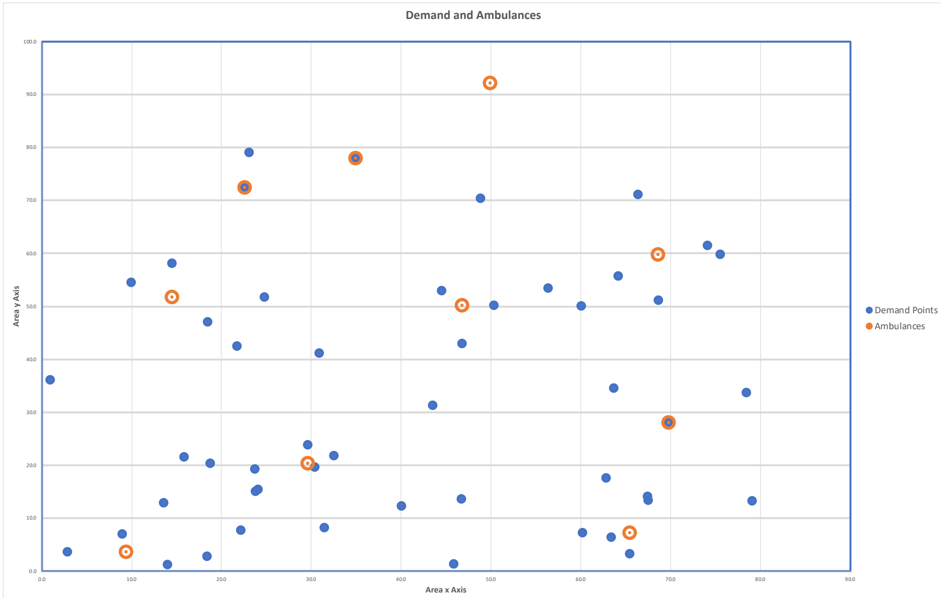


obj. function:	**9.4**

Max =	29.3
Median Distance =	9.2
Mean Distance =	10.3
SD =	7.2

The Vertical Line Test Returned:

(9.5, 3.6), (65.5, 7.2), (29.7, 20.4), (69.9, 28.0), (14.5, 51.7), (46.9, 50.1), (68.7, 59.7), (22.6, 72.4), (35.0, 77.9), and (50.0, 92.1) for ambulances one through ten, respectively.

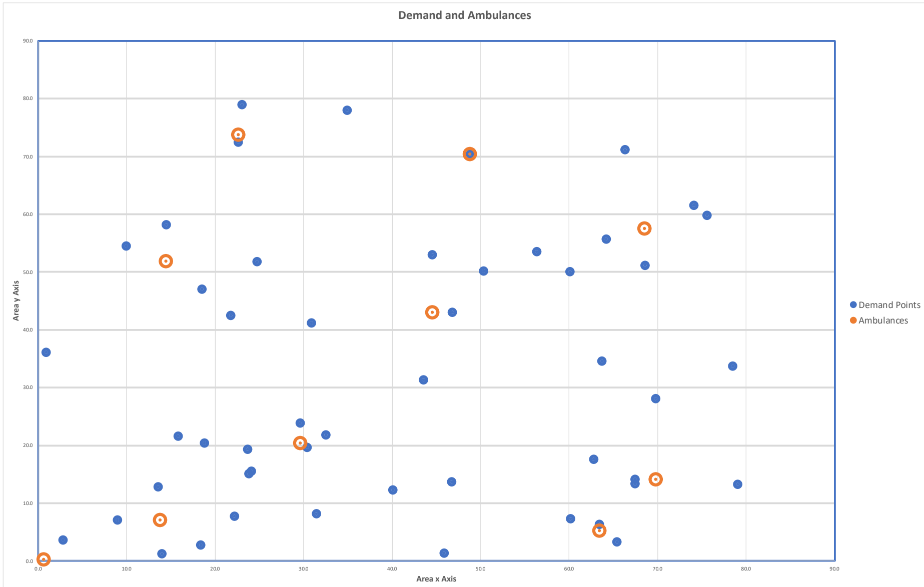


obj. function:	**9.5**

Max =	29.2
Median Distance =	8.8
Mean Distance =	10.6
SD =	6.9

The Diagonal Line Test Returned:

(0.7, 0.2), (13.8, 7.0), (29.7, 20.4), (63.4, 5.2), (44.6, 42.9), (69.8, 14.1), (14.5, 51.8), (68.6,57.5), (48.9, 70.4), and (22.6, 73.7) for ambulances one through ten, respectively.



obj. function:	**9.5**

Max =	29.2
Median Distance =	9.4
Mean Distance =	10.4
SD =	7.1

Conclusion:

Solver is unable to solve this non-linear program with repeatability. It fails to solve the test cases with known/obvious answers. The problem is non-convex because we can switch sets of ambulances and get the same objective function value. Solver is likely not working for this problem because of non-convexity. Multi-Solver or Python may produce better results.

Annotated Bibliography:

A model for optimally dispatching ambulances to emergency calls with classification errors in patient priorities.
Laura A. McLay & Maria E. Mayorga

Annotation: The goal of this article is to create a model for determining how to optimally dispatch servers to prioritized customers. It takes into account that dispatchers make classification errors when assessing customer priorities. It uses a Markov Decision Process (MDP) model. It was useful to my project because it was similar and had priorities (where as I had population weights). It was helpful to see an approach by professional researcher for an ambulance problem. The objective was different because they sought to determine how to optimally dispatch ambulances to patients to maximize the long-run average utility of the system. They conclude that their model should be applicable to real life situations. I like that they had an example of what they did.

The Vehicle Mix Decision in Emergency Medical Service Systems. Manufacturing & Service Operations Management 18:3, pages 347-360. Kenneth C. Chong, Shane G. Henderson, Mark E. Lewis. (2016

Annotation: The goal of this paper was to look at the problem of selecting the number of Advanced Life Support (ALS) and Basic Life Support (BLS) ambulances, which the describe as "the vehicle mix". They look at the problem under a budget constraint. It was useful to my project because they were doing a similar study, but they had more constraints. They use a Markov Decision process and an Integer Program in their research. This differed from my project because I used a non-linear program. They conclude that their research was successful given reasonable inputs. I like that they described details about what happens on ambulance calls and how it relates to a project like this. This is something that was not considered in my project.

Optimal ambulance location with random delays and travel times
Armann Ingolfsson, Susan Budge, Erhan Erkut

Annotation: This paper involves an ambulance location optimization model that minimizes the number of ambulances needed to provide a specified service level. They measure service level as the fraction of calls reached within a given time standard. They have random delay and random travel time elements in their model. They also add uncertainty in the ambulance availability when determining response time. This paper was the most similar to what I was doing of the papers I read, but it had added variables/constraints that I did not use. I like that they used actual data to do their model.

Ambulance location and relocation problems with time-dependent travel times
Verena Schmid and Karl F.Doerner

Annotation: In this paper they use a model that takes into account time-varying coverage areas. Their model allows ambulances to be repositioned to maintain a certain coverage standard. They use a mixed integer program for the problem to optimize coverage at various points in time simultaneously. This paper was useful because it had an interesting overview of models that have been used on ambulance problems. They solve the problem using what they call a variable neighborhood search. This one had a few too many symbols and the diagrams were small and difficult to see.

Introduction to Operations Research, Hillier-Lieberman, 10th Edition,

Annotation: An introductory level operations research book. I used this to look at examples of non-linear programming problems. There are good examples in here and it's straightforward.

Operations Research – Applications and Algorithms, Wayne L. Winston, 2004

Annotation: This is a highly regarded operations research book. It's very extensive and goes into more depth than the Hillier-Lieberman book. I used this to look at examples of non-linear programming problems.

https://www.mathworks.com/discovery/nonlinear-model.html

Annotation: An interesting website with links to videos/links about non-linear modeling.

https://www.firerescue1.com/fire-products/traffic-control/articles/1123548-Scene-safety-Emergency-vehicle-placement-tips/

Annotation: Although this was not about mathematical modeling, it talks about some interesting information regarding EMS services and arrival on the scene. Some of the information would need to be taken into account when modeling real world ambulance situations.