Traveling at the Speed of Life: Optimizing the Efficiency of Ambulances in a Multizonal County

Team # 4498

November 10, 2013

1 Summary

In tackling the problem, we considered the ideal distributions of ambulances among the zones depending on the number of ambulances that are available. To do this, we had to develop criteria to be able to compare distributions. To do this, a simplifying assumption was first made and then removed.

If we assume that only one emergency call will happen at a time (two calls will not happen in close succession so that the response to one affects the possibilities of response to the other), we can consider the percent of the population that can be reached by an ambulance in under 8 minutes as a preliminary statistic. Using a computer based search of all of the possible distributions of algorithms, we found that there were 13 distributions of 3 ambulances which cover 100% of the population in under 8 minutes. Therefore, to compare these distributions, we calculated the average response time for each distribution. There was a clearly superior distribution: stationing an ambulance in each of zones 1, 2, and 5.

We repeated this search for an ideal allocation of two ambulances. We found that only one distribution of 2 ambulances resulted in 100% of the population being reached in under 8 minutes: positioning one in region 2 and the other in region 5. Finally, we considered one ambulance, and found that the ideal zone to place it in was zone 2 so as to reach the maximum number of people in under 8 minutes.

Additionally, given these ideal distributions, it was considered how much time was required to move between them given that an ambulance suddenly became available or unavailable. However, for ambulances that become unavailable for short, predictable periods of time (such as would happen from taking a call) it was considered on a case-by-case basis whether it would be beneficial to move the remaining ambulances or simply wait for the ambulance to become available again.

However, sometimes multiple calls occur in close succession, and can be modeled as happening simultaneously for purposes of response. Using a Poisson distribution, we calculated estimates as to the likelihood of some number of calls happening within a 10 minute time period. In these situations, if there are less than 3 calls (which will happen roughly 99.8% of the time), our available resources are still sufficient to deal with the situation. An event where two emergency calls happen simultaneously was modeled with the 13 distributions previously found, and all things considered, the previous distribution of ambulances was found to still be the best. Although there are three or more incidents extremely infrequently, it is possible, and our resources would be overwhelmed in these cases.

Contents

2 Introduction

The availability of speedy and effective emergency services is imperative in times of crisis: for the majority of citizens, an ambulance is the only source of aid during emergencies. There are 240 million 9-1-1 calls each year in the United States [1]. But ambulances are limited in their ability to reach people quickly, and in life or death situations, the extra minute of delay caused by various inconveniences can be the difference between a heroic rescue and a tragic loss. Unfortunately, very little can be done about random occurrences such as traffic and weather. But where an ambulance is when a call comes in is something completely within our control, and is arguably the most significant factor in reducing response times. Obviously, an ambulance 10 miles from you is far less useful to you than one down the block. The problem then is to position ambulances so that the greatest number of people can be reached in the smallest amount of time possible.

In this paper, we have developed a mathematical model to optimize the usefulness of available ambulances in a given urban area. This area consists of 6 zones, and travel times between the zones, as well as populations per zone, are shown below.

Travel Times in Minutes						
Zones	1	2	3	4	5	6
1	1	8	12	14	10	16
2	8	1	6	18	16	16
3	12	18	1.5	12	6	4
4	16	14	4	1	16	12
5	18	16	10	4	2	2
6	16	18	4	12	2	2

Zone	Population
1	50,000
2	80,000
3	30,000
4	55,000
5	35,000
6	20,000
Total	270,000

Our model determines the optimum placement of 1, 2, and 3 ambulances, as well as the county's ability to respond to catastrophic situations. This was accomplished by exhaustively checking all cases using modern software, since there are so few ambulances and so few zones that this is extremely computationally feasible.

3 Assumptions

Assumption 1. Maximizing the number of people covered takes priority over minimizing response time.

Justification. Studies have shown that 8 minutes is the critical time by which individuals in need of assistance must be reached [2]. Below 8 minutes, variations in arrival time are much less important, and so our model strives to maximize the number of people reached in 8 minutes rather than reducing times further below 8 minutes, since there are diminishing returns for decreasing time past 8 minutes.

Assumption 2. The position of the ambulance in each zone is irrelevant to the average time taken to reach people.

Justification. Since we have no information as to the geographic layout inside each zone, it will be considered that the only reference to position that is significant for each ambulance is the zone that it is in.

Assumption 3. Random deviations from the average travel time are small enough that they are negligible.

Justification. We have no data on the variability of travel times, and any variations in travel time would most likely uniformly affect all ambulances, so we may assume that these times are fixed.

Assumption 4. Emergency calls are equally likely to come from any one individual.

Justification. While this may not actually be the case in real life, unless there is a group of high-risk individuals in some geographic zone (and there is nothing to suggest that this is the case), differences in individual probabilities will balance each other out geographically and so we can assume that all individuals are equally likely to place an emergency call that needs to be responded to. Additionally, since we don not have data on the frequency of emery calls from each each area, taking variances in zonal needs into account is impossible, and so we must assume population is the only factor affecting how important it is that a zone be quickly serviced.

Assumption 5. Deviations from "semi-perfect conditions" will affect all transportation times proportionally.

Justification. The given travel times are small enough that we may assume that this county is relatively small, and factors such as weather and traffic will most likely impact travel between all zones equally. This means that the results from semi-perfect conditions will be roughly generalizable to all conditions.

Assumption 6. No two emergencies occur at the same time.[1]

Justification. This means that each ambulance is summoned from its designated location, and not another location where it is responding to an emergency. This assumption is acceptable because the probability of one emergency is already low, and the probability of two occurring simultaneously is so small it must be negligible.

Assumption 7. More than one ambulance can be placed in the same zone.

Justification. This is clearly possible in real life, and although seemingly counterproductive, placing multiple ambulances in the same zone may decrease response times.

Assumption 8. The rate of calls that require medical attention is the same as the U.S. national average

Justification. There is no indication that the rate of calls is higher or lower than the U.S. national average, so therefore we will use this rate as the best approximation to the number of calls that can be expected to require medical attention.

[1] This assumption is revisited in the "Crisis" part of the model

Assumption 9. The time an ambulance spends on a call (ignoring travel time) is in the neighborhood of 10 minutes

Justification. This is an order of magnitude estimate for the amount of time an ambulance will be occupied on average by a call for the purpose of rough calculations.

Assumption 10. The probability of an emergency call being placed does not change with time.

Justification. Although in reality there may be minor differences in the probability of an emergency call happening at any one time, given lack of data, we must assume that 9-1-1 calls are equally likely to happen at any time.

Assumption 11. Emergency calls are independent of each other (i.e. one call happening will not in any way influence other calls).

Justification. For the most part, emergency calls are unrelated to one another, so it is a safe assumption that they will be independent events.

4 Model

Purpose

The purpose of this model is to determine how to place 3 or fewer ambulances so as to best serve the people of the county.

Ranking Arrangements

In order to decide which arrangement is best, we must somehow sort them based on"value." According to our assumptions, the best way to do this is by favoring arrangements that serve more people over arrangements that serve fewer people. If multiple arrangements can serve the same number of people, then the arrangement with the higher average response time is superior.

Algorithm

We first write a program that will help us in determining the ideal placement of ambulances (the code, written in Java, can be found in appendix A). First, the code defines the time threshold of importance to be 8 minutes.

Additionally, the code defines a method that will return the time it takes to travel between two input zones, the starting zone and destination zone. It also makes a method that will return a population (in thousands) given a zone number. Finally, it employs two standard methods that will return the least of 2 or 3 numbers respectively.

With these methods, the program can implement a method that returns the population an arrangement of some number of ambulances can reach. As inputs, it takes some number of zone numbers each one corresponding to the location of an ambulance. It then creates a variable initially equal to 0 representing the population that can be reached in less than or equal to threshold minutes. For each zone i from 1 to 6, the program checks that the minimum time from each of the input zone numbers to the zone i is less than or equal to the threshold. If it is, then the population of that zone is added to the population reached; otherwise, nothing happens. Then the method returns the number of people reached. There are three versions of this method, one for one of 1, 2, or 3 ambulances.

Also, the program has a method to determine the expected time it will take to reach a random person with a given configuration of ambulances. As inputs, it takes some number of zone numbers that correspond to the positions of ambulances. It then defines a variable that will correspond to the average time. For each of the 6 zones, it adds to the variable the minimum of the times the ambulances will make from each of the input zones, times the population of the current zone it is checking. It then divides this value by total population and returns it. This method also works for 1, 2, or 3 ambulances.

With all the necessary methods defined, the program's main method runs as follows. Using 3 cycled loops, it checks all possible configurations of 3 ambulances. For each one, if it reaches all of the population in less than 8 minutes, it prints:

- The configuration
- The average time it takes to reach a random individual
- The average of the population reached in under 8 minutes, for each of the 3 subsets of two ambulances of the configuration

It then checks (using two nested loops) all possible configurations of 3 ambulances. Fore each one, if it reaches all of the population in under 8 minutes, it prints the configuration and the average time it will take to reach a random individual.

Finally, using a single loop, it checks all possible configurations of one ambulance and prints the population that it can reach in under 8 minutes.

Three Ambulances

Since there are less than $6^3 = 216$ possible arrangements of the ambulances, we used the previously described program. It determined that there are 13 arrangements that are able to service the entire population. They are listed below, where the ordered triple (a, b, c) denotes an ambulance in zones a, b, and c (the same number appearing multiple times in one triple denotes putting multiple ambulances in one zone):

$$(1,2,5) \quad (1,3,4) \quad (1,3,5) \quad (1,4,5) \quad (1,4,6) \quad (1,5,6)$$

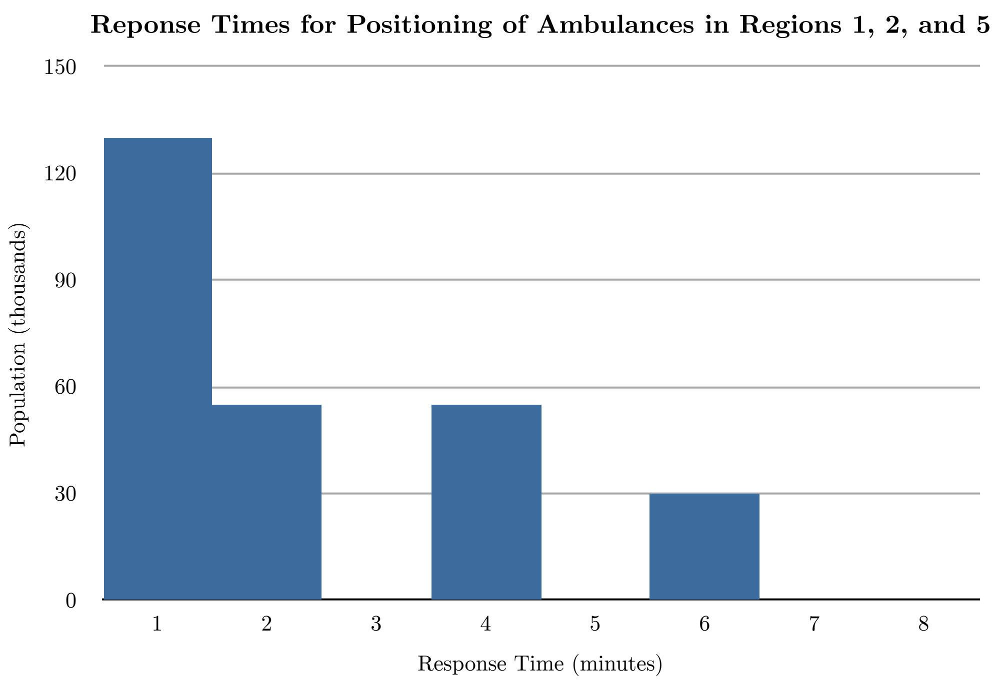
$$(2,2,5) \quad (2,3,4) \quad (2,3,5) \quad (2,4,5) \quad (2,4,6) \quad (2,5,5) \quad (2,5,6)$$

Since there are multiple configurations that can reach 100% of people, the program then calculated the average time taken to reach an individual for each of the 13 configurations. The results of this analysis are shown below.

Configurations of 3 Ambulances Sorted by ETA

Distribution	ETA (min.)
(1,2,5)	2.370
(2,4,5)	2.833
(2,4,6)	2.833
(2,3,5)	3.166
(2,3,4)	3.222
(2,5,6)	3.444
(1,4,5)	3.611
(1,4,6)	3.611
(2,2,5)	3.666
(2,5,5)	3.666
(1,3,5)	3.944
(1,3,4)	4.000
(1,5,6)	4.222

This clearly shows that if only 3 ambulances are available, they should be placed in zones 1, 2, and 5. This will minimize response time, and serve everyone in the county. A histogram of response times, as well as statistics on response time for this configuration are below.

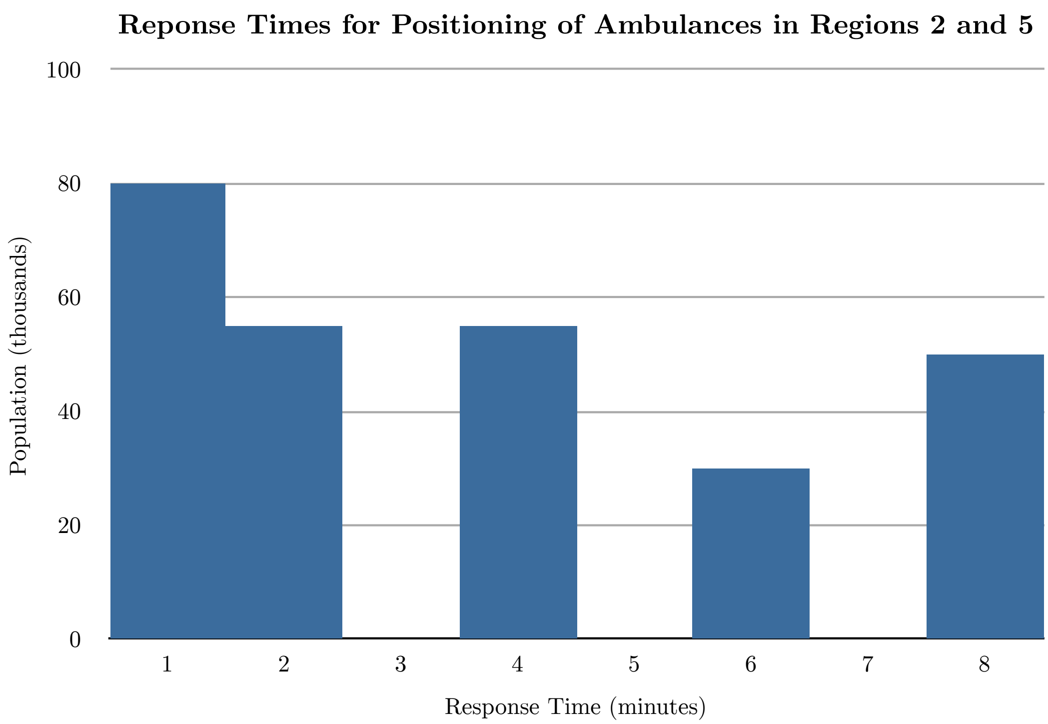


Statistics for Response Time Distribution

$\overline{x}$	Min	Q1	Median	Q3	Max
2.370	1	1	2	4	6

Two Ambulances

We have determined the optimal positioning of three ambulances, but we now consider the problem of what to do if only two ambulances are available. We repeat the procedure that we used for 3 ambulances, but our task is greatly simplified this time: for two ambulances, this results in only one arrangement that can cover all of the population in under 8 minutes: an ambulance in zone 2 and another in zone 5. There is thus no need to consider response times, since we place priority on serving the maximum number of people. The histogram of response times, and statistics for this configuration are shown below.



Statistics for Response Time Distribution

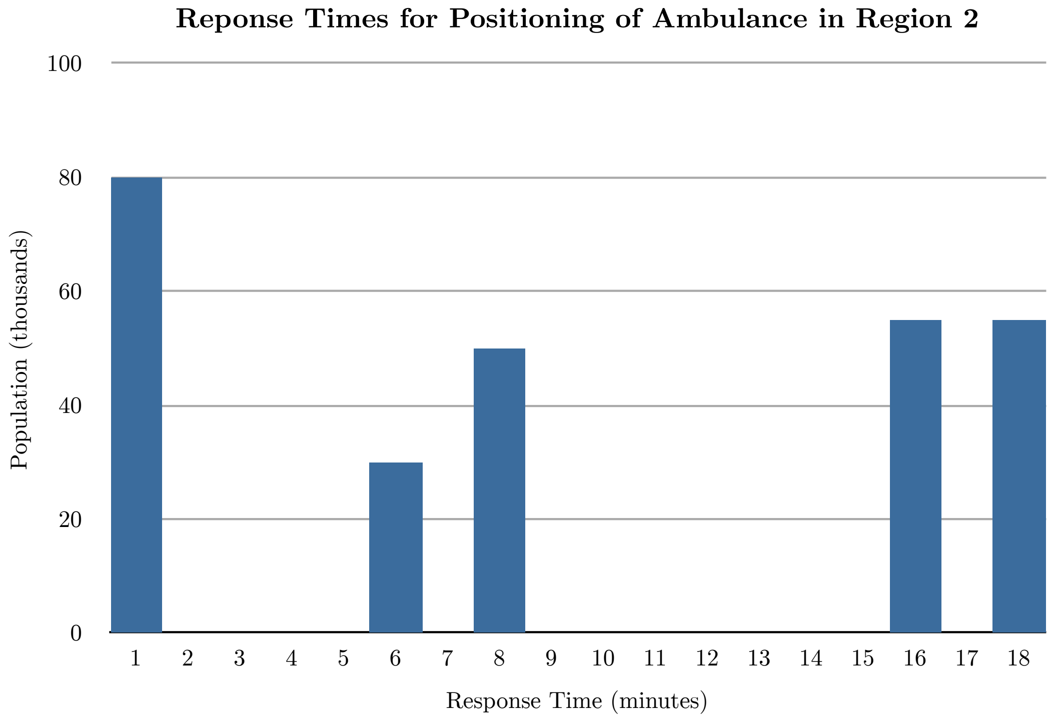
$\overline{x}$	Min	Q1	Median	Q3	Max
3.667	1	1	3	6	8

One Ambulance

We recycled the approach used for three and two ambulances for this case. Shown below is the number of people reached in under 8 minutes by placing the single ambulance in each zone.

Zone	Population
1	130,000
2	160,000
3	85,000
4	85,000
5	110,000
6	85,000

This shows that an ambulance placed in zone 2 will serve the largest number of people, and is the best configuration. This histogram and statistics of response time for a single ambulance placed here are shown.



Statistics for Response Time Distribution

$\overline{x}$	Min	Q1	Median	Q3	Max
9.370	1	1	8	16	18

Changing Number of Ambulances

In the case that one ambulance may be suddenly needed to deal with an emergency for an indefinite period of time or may finish dealing with an emergency and is now free, the number of ambulances available to help at any given time will constantly change. To address this, the number of ambulances that are presently available should redeploy to the ideal configuration of the number of ambulances that are available.

Losing an Ambulance

In the case that we have three ambulances and one becomes unavailable, we consider what zone that ambulance was covering, and what should be done fix any gap in coverage of the county:

Zone	Movement	Time to Redeploy
1	None (ambulances are in ideal position)	0 minutes
2	Ambulance in zone 1 moves to zone 2	8 minutes
5	Ambulance in zone 1 moves to zone 5	10 minutes

Alternatively, if we only had two ambulances to begin with and we lose one, we consider what should be done depending on what zone the newly occupied ambulance was covering

Zone	Movement	Time to Redeploy
2	Ambulance in zone 5 moves to zone 2	16 minutes
5	None (ambulance is in ideal position)	0 minutes

Gaining an Ambulance

In the case that we have an ambulance in zone 2, we consider what we should do with a newly available ambulance depending on the zone in which it is located:

Zone	Movement	Time to Redeploy
1	Move new ambulance to zone 5	10 minutes
2	Move new ambulance to zone 5	16 minutes
3	Move new ambulance to zone 5	6 minutes
4	Move new ambulance to zone 5	16 minutes
5	Move ambulance within zone 5	2 minutes
6	Move new ambulance to zone 5	2 minutes

Additionally, in the case that we have two ambulances in zones 2 and 5, we consider what we should do with a newly available ambulance depending on the zone in which it is located:

Zone	Movement	Time to Redeploy
1	Move new ambulance within zone 1	1 minute
2	Move new ambulance to zone 1	1 minute
3	Move new ambulance to zone 1	12 minutes
4	Move new ambulance to zone 1	16 minutes
5	Move new ambulance to zone 2 and move zone 2 ambulance to zone 1	16 minutes
6	Move new ambulance to zone 1	16 minutes

Short-lived Fluctuations

Whilst these movements are necessary to provide ideal coverage for the county if an ambulance is going to be unavailable for an indefinite period of time, they may be counterproductive if an ambulance is going to be unavailable for only a short period of time, in which it may return to it's coverage of a zone before any adjustments can take place. In the cases where the ambulance can return to its position faster than could be achieved by removing and adding the ambulance successively using the above procedures, it is preferable to only change the position of the ambulance that was called.

For this, we consider 3 ambulances in their ideal locations in zones 1, 2, and 5. For an emergency call in any one zone, we consider the ambulance that responds and the time it would take for the ambulance to travel to the zone with the emergency and back. Additionally, we consider the time it would take for the ambulance to go to to the emergency zone, wait (if necessary) for the ambulances to redeploy (as per the section "losing an ambulance"), and then be added back to the active ambulances (as per the procedures in the section "gaining an ambulance").

Emergency Zone	Response Ambulance Zone	Round-trip time	Time to Redeploy
1	1	2 minutes	2 minute
2	2	2 minute	9 minutes
3	2	24 minutes	20 minutes
4	5	20 minutes	26 minutes
5	5	4 minutes	26 minutes
6	5	4 minutes	26 minutes

Since redeploying is a trivial action if the emergency is in zone 1 (the unoccupied ambulances already represent an ideal positioning), a round-trip for the ambulance is the same as redeploying, and the ideal movement for this case.

For an emergency is zone 2, redeploying presents a significantly larger time than that of a round-trip, and as such a round-trip for the occupied ambulance is preferable.

For an emergency in zone 3, redeploying presents a slightly shorter time (and more coverage in the meanwhile) and as such presents the more preferable option.

For an emergency in any zone 4, a round-trip presents a slightly shorter time (with a negligible loss of coverage since we are assuming that calls last in the neighborhood of 10 minutes).

For an emergency in zone 5 or 6, the great time difference between a round-trip and the time to redeploy means that the round-trip greatly preferable.

Crisis Situations

We interpreted a crisis situation as one that violates the assumption that no two emergencies happen at the same time, and attempted to determine the best course of action in such a situation.

Likelihood

In considering the likelihood of what we define to be a crisis situation (more than one emergency at a time) we must first define how closely spaced two calls should be to consider them happening at the same time. In our ideal positioning of 3 ambulances, given the loss of any one ambulance, the other two ambulances can position themselves ideally in a maximum of 10 minutes. Therefore, we define two calls placed within a 10 minute time period to be considered a crisis.

With this definition in hand, we consider that in 2008, "fire departments nationwide responded to 15.7 million total medical aid calls" [7]. Also, according to the United States Census, the population of the United States in 2008 was approximately 314 million. With this information, we can find our expected rate of responses for this county (in expected number of responses per 10 minutes):

$$\frac{15.7 \times 10^6 \text{ responses}}{314 \times 10^6 \text{ people} \cdot \text{year}} \cdot \frac{1 \text{ year}}{365 \text{ days}} \cdot \frac{1 \text{ day}}{1440 \text{ minutes}} (270,000 \text{ people})(10 \text{ minutes}) = 0.2568$$

Given this information for the average number of calls in a 10 minute period, and the assumptions, we can model the number of calls that occur in a 10 minute period as a Poisson distribution with $\lambda = 0.2568$. This means that we can create the probability distribution:

Number of Calls	Probability
0	0.7735
1	0.1987
2	0.0255
3	0.0022
more than 3	0.0001

From this information, we can see that 97.22% of the time, there will only be one emergency call that requires attention in a given 10-minute time period. However, 2.55% of the time, there will be two emergency calls in a time period. Since any greater number of events will only happen less than 0.23% of the time, it is very unlikely that we get more than two calls and we can consider these cases too rare to be of importance.

Therefore, we concern ourselves with ensuring that our distribution of ambulances is the best possible to deal with the possibility of 2 events happening at the same time.

The Algorithm

To tackle this new situation, we must write a new program (which can be found in appendix b). For this program, we once again define the threshold value at the beginning and set it equal to 8 minutes.

First, the program redefines the methods that were defined in the first program. A method is defined that will return the time it takes to travel from a source zone to a destination zone, both of which are given as input. Additionally, a method is defined that will return the population of a region given the region number as input.

The program then defines a method that will, given 3 zone numbers for ambulances, and 2 destination numbers for emergencies, determine the maximum number of the emergencies that can be reached by an ambulance in under threshold minutes. It is assumed that once an ambulance is going to one emergency, it can not help the other. The method first defines a variable that is going to be the maximum number of the emergencies that can be helped and sets it equal to 0. It then considers all possible pairings of the emergencies with an ambulance, and if for any one of these, one location can be reached in under threshold minutes, the variable is set equal to 1. It then considers all possible matchings, and if any one of them results in both emergencies being reached in under threshold minutes, the variable is set equal to 2. The variable is then returned.

Also, the program defines a method that will give the population that can be reached in under threshold minutes if emergencies happen two at a time. The method takes in as input three numbers corresponding to the locations of the three ambulances. A variable is defined to be the population that can be reached. The method cycles through all possible pairs of regions in which emergencies happen using two nested loops. For each pair the variable is incremented by the product of the two regions times the output of the previously defined method given the two emergencies and three ambulance locations. The variable is then returned.

With the definition of all the necessary methods, the main method can be defined. The main method cycles through all possible configurations of ambulances, and if a configuration is one of the 13 that covers all of the population (as determined from the previous program) the program outputs the configuration along with the percentage of the population that is covered if two incident happen at the same time (the previous method for the configuration divided by (total population squared times two)).

Three Ambulances

In order to compare all possible arrangements of three ambulances by this criteria, for each arrangement, we will consider all possible combinations of two calls. Two simplify calculations, we consider that the origin of the first and the second call are independent (meaning that theoretically both could will result from the same person); given the 1 in 270,000 chances of this happening in the model, it provides negligible error of $\frac{1}{270,000} \approx \pm 3.7 \times 10^{-6}$ in calculations.

For every possible origin of two different calls, we then consider the percentage of those calls could be reached in under 8 minutes (0%, 50%, or 100%) given some distribution of ambulances. Having these values, we average them to get the percentage of the time that the ambulances will reach their destinations given that 2 emergencies happen in the same time interval. We can test the 13 solutions that we had previously found that reach 100% of the population if only one emergency happens at a time (using the program found in appendix B), to get the following results:

Distribution	Percent Covered
(2,2,5)	92.81%
(1,2,5)	92.20%
(2,3,5)	87.45%
(2,4,5)	87.45%
(2,5,6)	87.45%
(1,3,5)	86.83%
(1,5,6)	86.83%
(1,4,5)	86.83%
(1,3,4)	85.37%
(1,4,6)	85.37%
(2,3,4)	85.37%
(2,4,6)	85.37%
(2,5,5)	83.55%

This has the surprising result of showing us that that placing an ambulance in each of zones 1, 2, and 5 is not by this metric the best (but still the second best) arrangement. However, upon further investigation, we see that the difference between placing two ambulances in zone 2 and one in zone 5, and an ambulance in each of zones 1, 2, and 5 is about 0.6% of the population 2.55% of the time which pales in comparison to an average time of arrival difference of about 1 minute 20 seconds.

Therefore, all things considered, placing an ambulance in each of zones 1, 2, and 5 is still the best distribution when taking into account that emergencies can happen at roughly the same time.

Real World Preparations

In real life, urban areas employ a variety of strategies to prepare themselves for large scale disasters. Recently, disaster planning has shifted from forming a single plan for all disasters to forming individual plans for specific disasters. In an effort to plan for specific catastrophes, local state officials often use a technique known as gap analysis [5]. This is able to account and provide a basis for planning for catastrophes. By looking at the amount of resources the city/county possesses, the amount of resources needed in order for the county/city to be able to successfully respond to the disaster, and the gap between these two measures, planners are able to see how prepared the city/county is to handle the catastrophe at hand. Using this analysis, planners can see what needs to be done in order to be fully prepared for the disaster at hand.

As another means of preparing for these catastrophic events, cities/counties generally implement what are known as early warning systems [4]. These systems serve to provide a warning to the population of an area that a catastrophe is about to occur. Even a warning a couple minutes early can allow the population to remove themselves from dangerous positions, moving to a safer position, and be more prepared when the disaster hits. By reducing the number of people in jeopardy during a disaster, cities/counties are also able to reduce the scope of the necessary response, and thus the resources necessary in the case of emergency.

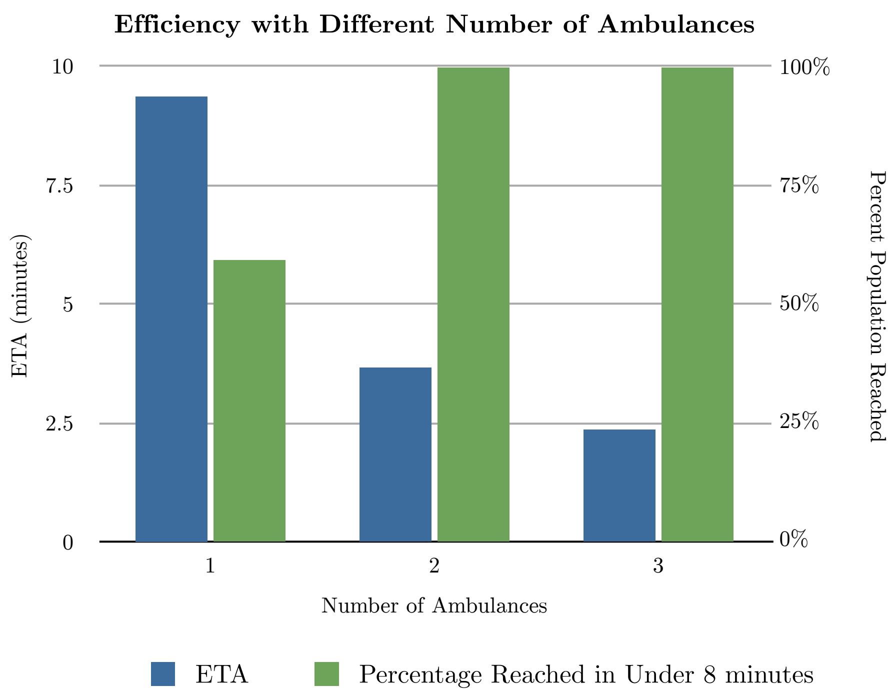
5 Conclusions

The mathematical analysis, along with the program, allowed us to answer each question presented in the problem statement.

If three ambulances are available, then they should be placed in zones 1, 2, and 5. They will be able to reach the entire population in less than 8 minutes, and will take, on average, about two-and-a-half minutes. If two ambulances are available, then they should be placed in zones 2 and 5. This still serves the population of the entire county in less than 8 minutes, but has an average response time of about three-and-a-half minutes. If only one ambulance is available, it should be placed in zone 2. Unfortunately, this leaves 110,000 people unable to be reached in less than 8 minutes, but has an average response time of about 9 minutes.

In the event of a crisis situation with multiple calls at the same time, ambulances in zones 1, 2, and 5 are still the optimal configuration. However, if there are more than 3 call simultaneously, which is unlikely but possible, the cities' emergency response system would be overwhelmed.

Our model is strong in that because it exhaustively checks all cases, we know that it gave the optimal configuration given our assumptions. Also, because we gathered data on the frequency of calls in the US, and the given numbers for population are reasonable, this model could realistically be applied to a metropolitan area. Unfortunately, some of our assumptions may be violated in real life. Travel times can vary based on weather, and it maybe difficult to divide a large city into only 6 areas with consistent travel times from one zone to another.



6 Memo to ESC

Memorandum
To: County Emergency Service Personnel
Date: November 10, 2013
Re: Ambulance Allocation

The speedy and effective response of the countys limited supply of ambulances is crucial to the safety of citizens in peril. As a result of the need to tend to those who depend on us for help, we must properly allocate our emergency services three ambulances throughout the county's zones so that we can be prepared to respond to a call from any zone within the generally-accepted 8-minute ALS/BLS standard. With the help of extensive computer simulation and in-depth analysis of the average response time data on record, we have concluded that an ambulance must be placed in zone 1; another one in zone 2; and a third in zone 5 in order to achieve maximal speed, minimum distance, and greatest coverage.

With all three ambulances available for dispatch, on average, first responders can address any of the 270,000 of our residents in about two-and-a-half minutes. This is outperforms the vast majority of over two hundred possible ambulance orientations. If the ambulance in zone 1 is already responding to an emergency, on average, the other two ambulances will be able to reach a given county resident within three minutes and forty seconds. Despite the incapacity of ambulance 1, our emergency medical staff will still be perfectly capable of reaching all of the county's citizens, highlighting the potentially life-saving efficacy of this new distribution of ambulances.

If ambulances 1 and 5 are already responding to emergencies, ambulance 2 will still be able to reach 160,000 residents (59.3% of the population) within 8 minutes and, on average, a given county resident in 9.27 minutes. The ability to reach nearly 60% of our citizens is noteworthy and surpasses the extension of any of the other ambulances. Moreover, while an average overall response time of 9.37 seconds may not be within the 8-minute ALS/BLS standard, it still lies within the time to respond to the majority of emergencies as the ALS/BLS standards are based on the window for treating cardiac arrest, which only has a survival rate of 2.2% when consciousness revival treatment is first applied by ambulance personnel [3].

In a crisis situation, if there are no more than three calls occurring simultaneously, we will be able to respond using the same ambulance placements as in non-crisis situations. However, if there is an excessive number of overlapping calls, external resources must be taken advantage of to effectively handle the situation. Some popular methods of handling larger crises include gap analysis, and awareness programs.

7 References

[1] 911 Statistics. (n.d.). Retrieved November 10, 2013, from National Emergency Number Association website: http://www.nena.org/?page=911Statistics

[2] Saadah, R. (n.d.). The Eight-Minute ALS Response Time Standard. Retrieved November 10, 2013, from Federal Emergency Management Agency website: http://www.usfa.fema.gov/pdf/efop/efo37506.pdf

[3] Nolan, J. (n.d.). Resuscitation Council (UK) comments on compression-only CPR study published in The Lancet 17 March 2007. Retrieved November 10, 2013, from Resuscitation Council (UK) website: http://www.resus.org.uk/pages/compCPRs.htm

[4] Earthquake Preparedness. (n.d.). Retrieved November 10, 2013, from City of Temecula website: http://www.cityoftemecula.org/Temecula/Residents/PublicSafety/earthquakes.htm

[5] Catastrophe Readiness and Response. (n.d.). Retrieved November 10, 2013, from Federal Emergency Management Agency website: http://training.fema.gov/emiweb/edu/docs/crr/CAT

[6] U.S. and World Population Clock. (n.d.). Retrieved November 10, 2013, from United States Census Bureau website: http://www.census.gov/popclock/

[7] Kavilanz, P. (n.d.). 911 abuse: Calling with the sniffles. Retrieved from CNN website: http://money.cnn.com/2009/08/24/news/economy/healthcare_911_abuse/

8 Appendix A: Placement Code

```
// ************************************************************************
// OneEmergencyAtATime.java                         Author: Team #4498
//
// Provides a comparison of different ambulance locations
// assuming that only one emergency will happen at a time
// ************************************************************************

public class model
{
    final static double THRESHOLD = 8;
    public static void main(String[] args)
    {
        for (int a = 1; a<=6; a++)
        for (int b = a; b<=6; b++)
        for (int c = b; c<=6; c++)
        {
            if (populationReached(a,b,c) == 270)
                {
                    System.out.print("(" + a + ", " + b + ", " + c + ")");
                    System.out.println(" Expected time of arrival: " +
                    expectedTime(a,b,c));
                    System.out.println((populationReached(a,b) +
                     populationReached(a,c) + populationReached(b,c))/3.0);
                }
        }

        System.out.println();

        for (int a = 1; a<=6; a++)
        for (int b = a; b<=6; b++)
        {
            if (populationReached(a,b) == 270)
            {
                System.out.println("(" + a + ", " + b + ") Expected time: " +
                 expectedTime(a,b));
            }
        }

        System.out.println();

        for (int a = 1; a<=6; a++)
        {
            System.out.println(a + ": population reached: " + populationReached(a));
        }


    }

    /*
     * returns the time it takes to go from zone #"from" to
     * zone #"to"
     */
```

```
public static double timeTo(int from, int to)
{
    double[][] vals = {{1, 8, 12, 14, 10, 16}, {8, 1, 6, 18, 16, 16},
     {12, 18, 1.5, 12, 6, 4}, {16, 14, 4, 1, 16, 12},
      {18, 16, 10, 4, 2, 2}, {16, 18, 4, 12, 2, 2}};
    return vals[from-1][to-1];
}

/*
 * returns the population in thousands of zone #"zone"
 */
public static int population(int zone)
{
    if (zone == 1) return 50;
    else if (zone == 2) return 80;
    else if (zone == 3) return 30;
    else if (zone == 4) return 55;
    else if (zone == 5) return 35;
    else if (zone == 6) return 20;
    else return 0;
}

/*
 * returns the population a configuration of 3 ambulances
 * reaches in less than THRESHOLD minutes
 */
public static int populationReached(int a, int b, int c)
{
    int reached = 0;
    for (int i = 1; i <=6; i++)
    {
        if (min(timeTo(a,i), timeTo(b,i), timeTo(c,i)) <= THRESHOLD)
            reached += population(i);
    }
    return reached;
}

/*
 * returns the expected value of the time it takes
 * for an ambulance to get to a random person for
 * a given configuration of 3 ambulances
 */
public static double expectedTime(int a, int b, int c)
{
    double sum = 0;
    for (int i = 1; i <=6; i++)
        sum += min(timeTo(a,i), timeTo(b,i), timeTo(c,i))*population(i);
    sum /= 270;
    return sum;
}

/*
 * returns the expected value of the time it takes
 * for an ambulance to get to a random person for
```

```
     * a given configuration of 2 ambulances
     */
    public static double expectedTime(int a, int b)
    {
        double sum = 0;
        for (int i = 1; i <= 6; i++)
            sum += min(timeTo(a,i), timeTo(b,i))*population(i);
        sum /= 270;
        return sum;
    }

    /*
     * returns the population a configuration of 2 ambulances
     * reaches in less than THRESHOLD minutes
     */
    public static int populationReached(int a, int b)
    {
        int reached = 0;
        for (int i = 1; i <=6; i++)
        {
            if (min(timeTo(a,i), timeTo(b,i)) <= THRESHOLD)
                reached += population(i);
        }
        return reached;
    }

    /*
     * returns the population a configuration of 1 ambulance
     * reaches in less than THRESHOLD minutes
     */
    public static int populationReached(int a)
    {
        int reached = 0;
        for (int i = 1; i <= 6; i++)
        {
            if (timeTo(a,i) <= THRESHOLD)
                reached += population(i);
        }
        return reached;
    }

    /*
     * returns the minimum of 3 values
     */
    public static double min(double a, double b, double c)
    {
        if (a <= b && a <= c) return a;
        else if (b <= a && b <= c) return b;
        else return c;
    }

    /*
     * returns the minimum of 2 values
     */
```

```
    public static double min(double a, double b)
    {
        if (a <= b) return a; return b;
    }
}
```

9 Appendix B: Code for Crisis

```
// ***************************************************************************
// Crisis.java                        Author: Team #4498
//
// Provides a comparison of different ambulance locations
// assuming that two emergencies happen at a time
// ***************************************************************************

public class catastrophe
{
        final static int THRESHOLD = 8;

        public static void main(String[] args)
        {
                for (int a = 1; a<=6; a++)
                for (int b = 1; b<=6; b++)
                for (int c = 1; c<=6; c++)
                {

                if ((a == 1 && b == 2) && c == 5)
                        System.out.println("(" + a + ", " + b + ", " + c + "): " +
                         (1.0*populationReached(a,b,c) /(2*270*270)))
                if ((a==2 && b == 4) && c == 5)
                        System.out.println("(" + a + ", " + b + ", " + c + "): " +
                         (1.0*populationReached(a,b,c) / (2*270*270)));
                if ((a==2 && b==4) && c == 6)
                        System.out.println("(" + a + ", " + b + ", " + c + "): " +
                         (1.0*populationReached(a,b,c) / (2*270*270)));
                if ((a==2 && b ==3) && c==5)
                        System.out.println("(" + a + ", " + b + ", " + c + "): " +
                         (1.0*populationReached(a,b,c) / (2*270*270)));
                if ((a==2 && b ==3) && c ==4)
                        System.out.println("(" + a + ", " + b + ", " + c + "): " +
                         (1.0*populationReached(a,b,c) / (2*270*270)));
                if ((a==2 && b==5) && c==6)
                        System.out.println("(" + a + ", " + b + ", " + c + "): " +
                         (1.0*populationReached(a,b,c) / (2*270*270)));
                if ((a==1 && b==4) && c==5)
                        System.out.println("(" + a + ", " + b + ", " + c + "): " +
                         (1.0*populationReached(a,b,c) / (2*270*270)));
                if ((a==1 && b ==4) && c ==6)
                        System.out.println("(" + a + ", " + b + ", " + c + "): " +
                         (1.0*populationReached(a,b,c) / (2*270*270)));
                if ((a==2 && b ==2) & c==5)
                        System.out.println("(" + a + ", " + b + ", " + c + "): " +
                         (1.0*populationReached(a,b,c) / (2*270*270)));
                if ((a==2 && b ==5) && c ==5)
                        System.out.println("(" + a + ", " + b + ", " + c + "): " +
                         (1.0*populationReached(a,b,c) / (2*270*270)));
                if ((a==1 && b ==3) && c==5)
                        System.out.println("(" + a + ", " + b + ", " + c + "): " +
                         (1.0*populationReached(a,b,c) / (2*270*270)));
                if ((a==1 && b ==3) && c==4)
```

```
                System.out.println("(" + a + ", " + b + ", " + c + "): " +
                 (1.0*populationReached(a,b,c) / (2*270*270)));
        if ((a==1 && b ==5) & c==6)
                System.out.println("(" + a + ", " + b + ", " + c + "): " +
                         (1.0*populationReached(a,b,c) / (2*270*270)));
        }

}

/*
 * returns the time it takes to go from zone #"from" to
 * zone #"to"
 */
public static double timeTo(int from, int to)
{
        double[][] vals = {{1, 8, 12, 14, 10, 16}, {8, 1, 6, 18, 16, 16},
        {12, 18, 1.5, 12, 6, 4}, {16, 14, 4, 1, 16, 12}, {18, 16, 10, 4, 2, 2},
        {16, 18, 4, 12, 2, 2}};
        return vals[from-1][to-1];
}

/*
 * returns the population in thousands of zone #"zone"
 */
public static int population(int zone)
{
        if (zone == 1) return 50;
        else if (zone == 2) return 80;
        else if (zone == 3) return 30;
        else if (zone == 4) return 55;
        else if (zone == 5) return 35;
        else if (zone == 6) return 20;
        else return 0;
}

/*
 * returns the population a configuration of 3 ambulances
 * reaches in less than THRESHOLD minutes
 */
public static int populationReached(int a, int b, int c)
{
        int reached = 0;
        for (int i = 1; i <=6; i++)
        for (int j = 1; j<=6; j++)
        {
                reached+= population(i)*population(j)*numReached(a,b,c,i,j);
                        reached += population(i);
        }
        return reached;
}

public static int numReached(int a, int b, int c, int i, int j)
{
        int max = 0;
```

```
        if (timeTo(a,i)<=THRESHOLD || timeTo(b,j)<=THRESHOLD) max = 1;
        if (timeTo(a,i)<=THRESHOLD || timeTo(c,j)<=THRESHOLD) max = 1;
        if (timeTo(b,i)<=THRESHOLD || timeTo(a,j)<=THRESHOLD) max = 1;
        if (timeTo(b,i)<=THRESHOLD || timeTo(c,j)<=THRESHOLD) max = 1;
        if (timeTo(c,i)<=THRESHOLD || timeTo(a,j)<=THRESHOLD) max = 1;
        if (timeTo(c,i)<=THRESHOLD || timeTo(b,j)<=THRESHOLD) max = 1;
        if (timeTo(a,i)<=THRESHOLD && timeTo(b,j)<=THRESHOLD) max = 2;
        if (timeTo(a,i)<=THRESHOLD && timeTo(c,j)<=THRESHOLD) max = 2;
        if (timeTo(b,i)<=THRESHOLD && timeTo(a,j)<=THRESHOLD) max = 2;
        if (timeTo(b,i)<=THRESHOLD && timeTo(c,j)<=THRESHOLD) max = 2;
        if (timeTo(c,i)<=THRESHOLD && timeTo(a,j)<=THRESHOLD) max = 2;
        if (timeTo(c,i)<=THRESHOLD && timeTo(b,j)<=THRESHOLD) max = 2;
        return max;
}

/*
 * returns the expected value of the time it takes
 * for an ambulance to get to a random person for
 * a given configuration of 3 ambulances
 */
public static double expectedTime(int a, int b, int c)
{
        double sum = 0;
        for (int i = 1; i <=6; i++)
                sum += min(timeTo(a,i), timeTo(b,i), timeTo(c,i))*population(i);
        sum /= 270;
        return sum;
}

/*
 * returns the expected value of the time it takes
 * for an ambulance to get to a random person for
 * a given configuration of 2 ambulances
 */
public static double expectedTime(int a, int b)
{
        double sum = 0;
        for (int i = 1; i <= 6; i++)
                sum += min(timeTo(a,i), timeTo(b,i))*population(i);
        sum /= 270;
        return sum;
}

/*
 * returns the population a configuration of 2 ambulances
 * reaches in less than THRESHOLD minutes
 */
public static int populationReached(int a, int b)
{
        int reached = 0;
        for (int i = 1; i <=6; i++)
        {
                if (min(timeTo(a,i), timeTo(b,i)) <= THRESHOLD)
                        reached += population(i);
```

```
        }
        return reached;
    }

    /*
     * returns the population a configuration of 1 ambulance
     * reaches in less than THRESHOLD minutes
     */
    public static int populationReached(int a)
    {
        int reached = 0;
        for (int i = 1; i <= 6; i++)
        {
            if (timeTo(a,i) <= THRESHOLD)
                reached += population(i);
        }
        return reached;
    }

    /*
     * returns the minimum of 3 values
     */
    public static double min(double a, double b, double c)
    {
        if (a <= b && a <= c) return a;
        else if (b <= a && b <= c) return b;
        else return c;
    }

    /*
     * returns the minimum of 2 values
     */
    public static double min(double a, double b)
    {
        if (a <= b) return a; return b;
    }

}
```