Feasible delivery systems of medical services

Alan Aw, Luke Kim

1 Background on transportation

EMS, which is a short form for "Emergency Medical Services", is a term that will be used throughout the paper, since our primary concern in this section is to formulate an efficient model for delivering medical services to those who are in need, specifically Ebola patients. From the question prompt, we are aware that we have to propose a model that can assist in optimizing the eradication of Ebola, and in this regard, transportation (i.e. the delivery of medical service to the infected) plays a critical role. By being able to quickly receive the responses of patients (or the infected) before the disease advances further, the probability for which we can save the lives of the infected increases. Also, by receiving responses from patients we are able to distinguish the infected, and successfully quarantine/separate them from the non-infected, but susceptible population of the people.This again reduces the chances of Ebola spreading and will contribute greatly to the optimization of eradicating Ebola. One important assumption is that people infected by Ebola, who began to show symptoms, are generally reluctant to go to hospitals or places that offer medical services by themselves and require the help of EMS (for example, ambulances).As such, the supply of EMS is of prime importance under this assumption.

The fact that efficient transportation of medical services is apparent, but we need a systematic method of delivering medical service to individuals. The underlying principle in Economics, that demands are infinite but supplies are finite, also holds for this situation. In other words, the number of demands for medical help may easily exceed the number of ambulances we have in a particular region, so it will be necessary for us to devise a methodology that efficiently allocates and dispatches ambulances to the incident that requires medical service. In fact, this problem of ambulance reallocation and dispatch is a classical problem that has been studied by many mathematicians and engineers and there are numerous literature available for this topic. Despite the fact that these papers have diverse approaches, most of them have the following assumption as to the procedure of EMS delivery:

$\boxed{\text{Arrival of emergency request}} \rightarrow \boxed{\text{Operator dispatches nearest available amublance}}$
$\rightarrow \boxed{\text{Ambulance reaches incident location}} \rightarrow \boxed{\text{Ambulance transports victim to hospital}}$
$\rightarrow \boxed{\text{Ambulance returns to base station}}$

For most papers, the above procedure is established, and it is then discussed how to reallocate ambulances efficiently, typically using methods such as **Mixed Integer Linear Program** or **Greedy Algorithm**. In fact, the general idea of sending the nearest ambulance to a particular location is called the **myopic policy** and it is assumed by many EMS suppliers that this is the most efficient and convenient method of dispatching ambulances because it greatly minimizes the response time. These approaches appear to be sound theoretically, but in terms of practice, we believe that this is not the most efficient method of allocating and dispatching ambulances to the emergency spot since it does not take into account the urgency of the patients. Taking into account the degree of urgency, or the priority of the calls from the patients is more significant in reality for two reasons:

1. The first reason is quite intuitive. If there are two calls A and B from two different patients, and let's assume that call A came first before call B. Also we have only one available ambulance at the moment, and say that this ambulance is closer to A. However, from the call, we are aware that B is in a life-threatening situation, whereas A is not. Even if this is the case, the myopic policy asserts that the ambulance should be sent to A instead of B, which definitely reduces the survival chance of B since B has to wait longer for EMS to arrive.

2. A more general way to explain this is by the following: we consider two units x and y, and it is assumed that both units have equally many responsibilities. However, x's area has a significantly higher call frequency. In this case, the mean response time will decrease if y is allowed to respond to some of the calls for which x is the closest unit. Actually, this result may be generalized for cases involving more than two units, and it was done by Cunningham-Green at 1988. Also, it may be better to send another unit z to take x's call when x already is busy, than to send the closer unit y. This holds true if the call frequency in y's primary district is higher than that of z's.

Hence it is important for the ambulance dispatcher to maintain an adequate **preparedness** for serving calls with various **priorities**. In terms of priorities, we mean the degree of urgency of the patients giving EMS the call. Since we are considering Ebola, we may apply the following criterion: when people infected by Ebola call EMS, they explain the symptoms they are showing. Depending on how many symptoms they have, we decide the priorities. Another important assumption here is that the people calling for help from EMS are only those that are infected by Ebola and not any other medical emergencies. (i.e. people calling from this particular region are calling because they want help regarding the cure of Ebola). Quick research shows that Ebola includes the following symptoms:

1. Fever greater than 101.5 degrees Fahrenheit or (38.6) degrees Celsius

2. Chills

3. Severe headache

4. Sore Throat

5. Muscle Pain

6. Weakness
7. Fatigue
8. Rash
9. Abdominal Pain
10. Diarrhea
11. Vomiting
12. Bleeding from the mouth and rectum
13. Bleeding from eyes, ears, and nose
14. Organ Failure

Since we are not medical professionals, this criterion may not be the most accurate, but this can late be changed. Symptoms 1 11 are early symptoms of Ebola, and among the symptoms that the patient explains, if it includes ≤ 5 of the symptoms in the list, we classify it as priority3 or $P3$. If it includes ≥ 6 and ≤ 11, then we classify it as priority2 or $P2$. If any of symptoms $12, 13, 14$ is suspected from the patient, we place the top priority on that patient and classify that patient as priority1 or $P1$. As it will be explained later, depending on the classification of the priorities of the call, the dispatch and allocation of the ambulances will be affected.

The algorithm that we employ to deliver EMS will work like the following: based on the priority of the call, ambulance dispatch will be decided. Once an ambulance is dispatched, it will affect the value of a constant named **Preparedness** of that particular region. When the value of Preparedness is under a certain threshold for that region, the ambulance dispatcher asks for more ambulance supplies such that it will increase the value of Preparedness to at least the threshold level. In this way, we can effectively cope with both the dispatch and the reallocation of ambulances. This model that we are using (originally established by Granberg and Varbrand) was once employed for major European cities (such as Stockholm) and was deemed successful, thus we find it reasonable to employ this method and adapt it to the context of efficiently delivering medical services to Ebola victims.

Before we end this section, here is the summary of some important assumptions that are made:
• Ebola patients are in need of EMS to arrive at medical service providers.
• Only Ebola patients will be contacting EMS under this setting, so that our criterion for determining the priority will be valid.
• The calls given by patients probably follow a Poisson distribution with an average of λ equal to a constant per minute, but the model we are employing is greatly simplified (but still efficient) and it does not require the use of this.
• The demand for ambulances probably follows some distribution that varies with time, and there have been papers that attempted to solve this problem. It is also known that even without changing the number of ambulances, it is possible to plan relocation of the existing units to match the demand. This is

called the location problem, and it can be solved for each time period under consideration. However, in terms of practicality, it is more common to dynamically relocate ambulance units, which is why we propose the dynamic ambulance relocation algorithm.
• From the prompt, Ebola patients have diseases that are **not advanced**, which probably means that instances of $P1$ calls will be rare. However, we still consider $P1$ as a possible situation that can arise in our model.

2 Algorithm for ambulance relocation

2.1 The Dispatching Algorithm

Before we delve into the algorithm for the dispatch of ambulances, we define Preparedness, or P as an important constant used in order to manage the ambulance relocation process. It is possible that the notion of Preparedness may vary among different people, depending on experience and personality. Furthermore the standard for the number of ambulances for a region to be considered 'Prepared' for emergency circumstances may vary widely. In order to eliminate these discrepancies, we may divide the region we are considering into a number of zones. For each zone j, we say a weight w_j is associated, which reflects the demand for ambulances in the zone. This weight can be defined such that it is proportional to the number of calls served in the zone during a specific time period. (this condition is quite loose: we may define the weight to be proportional to other factors too, such as the number of people currently residing in that zone). The preparedness in zone j is then calculated as

$P_j = \frac{1}{w_j} \sum_{l=1}^{L_j} \frac{\gamma^l}{t_j^l}$

Here, L_j is the number of ambulances that contribute to the preparedness in zone j, and t_j^l is the travel time for the ambulance l to zone j, γ^l is the contribution factor for the ambulance l (this is loosely defined as a constant of proportionality). These variables satisfy the following constraints: $t_j^1 \leq t_j^2 \leq ... \leq t_j^{L_j}$ and $\gamma^{L_j} < ... < \gamma^2 < \gamma^1$. From the equation for P_j, we can deduce that when L_j closest ambulances travel to zone j, P_j decreases as the travel time to the zone increases. Moreover, as the ambulance moves closer to the zone, t_j^l decreases and this increases P_j of the zone. If the call frequency increases, w_j increases and this decreases P_j.

Now that we have established the notion of P_j, we can explain how our ambulance dispatching algorithm operates: When the patient demands for medical help, the ambulance dispatcher classifies the call as either $P1, P2$ or $P3$. If the call is $P1$, then the ambulance with the shortest expected travel time to the call site (i.e. the nearest) is dispatched. We can say that we are resorting to myopic policy when we have a $P1$ call. Otherwise, if the call is $P2$ or $P3$, then we can assign an ambulance with a longer travel time. It is our assumption that an implementation of the preparedness measure involves a list of closest ambulances for each zone that is sorted according to the expected travel time, so it won't require too much work to find the closest ambulance to a certain region. Overall, if we were to illustrate this process mentioned in a flowchart, it

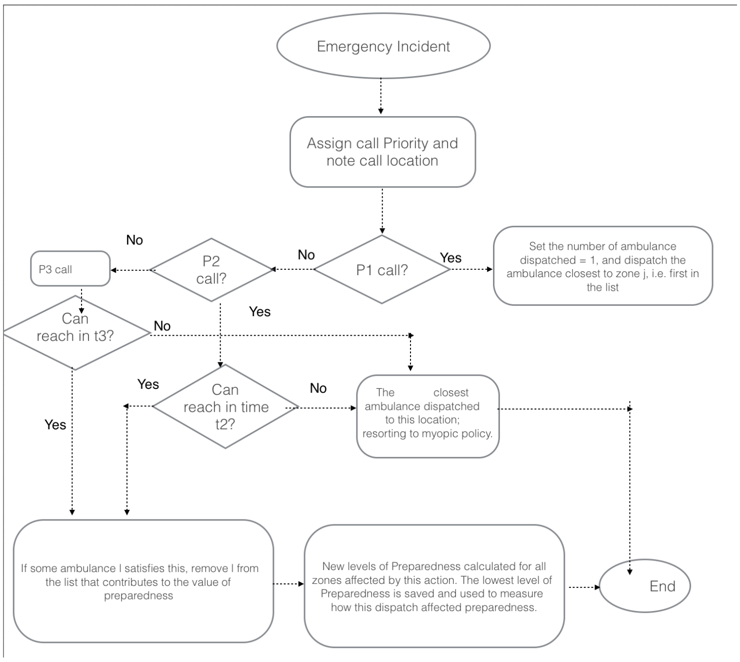

Emergency Incident
Assign call Priority and note call location
P1 call?
Yes
Set the number of ambulance dispatched = 1, and dispatch the ambulance closest to zone j, i.e. first in the list
No
P2 call?
No
P3 call
Yes
Can reach in t3?
No
Yes
Yes
Can reach in time t2?
No
The closest ambulance dispatched to this location; resorting to myopic policy.
If some ambulance l satisfies this, remove l from the list that contributes to the value of preparedness
New levels of Preparedness calculated for all zones affected by this action. The lowest level of Preparedness is saved and used to measure how this dispatch affected preparedness.
End

Figure 1: This flowchart illustrates the process of our dispatching algorithm depending on the priority of the call.It is produced by the candidate using pages.

would look like the one in Figure 1.

This is an algorithm produced by Granberg and Varbrand that explains the above process:

1. Let j be the zone to where an ambulance needs to be dispatched, and $l = 1, ..., L_j$ an ordered list of ambulances that contribute to the preparedness in j. Let $A = \emptyset$ be the ambulance that is dispatched and let $p_{min} = 0$.

2. If $P1$: Set $A = 1$ and dispatch A. So we are dispatching the ambulance closest to zone j and therefore first in the list.

3. If $P2orP3$: Check for the ambulances in the list, beginning with the closest: For $l = 1, ..., L_j$, If $t_j^l < T_2$ (or $T3$ depending on if the call is $P2$ or $P3$): remove ambulance l from the list of ambulances contributing to the preparedness, and recalculate the preparedness P_i, in all zones that are affected by this action.

4. If $min_{i \in N} P_i > P_{min}$, then $P_{min} = min_{i \in N} P_i, A = l$.

5. If we don't have any $l = 1, ..., L_j$ that satisfy just dispatch A.

The value of the absolute minimum threshold for P_{min} has to be determined around 0.923 for Stockholm, but this value has to be calibrated depending on the which region we are considering. In general finding P_{min} should involve some statistical analysis, we would have to run numerous simulations with the given data sets and find the minimum value of P that occurs. Similarly, finding the value of γ would require statistical analysis. The value of $\gamma^l = \frac{1}{2^{l-1}}$ according to the Swedish researchers who originally devised the algorithm, but this result must have relied heavily on real data sets and **distribution fitting** to come up with an appropriate contribution constant γ.

In order to fully showcase the computation of the dispatch algorithm, by considering a small data set. Assume we have only two zones to consider, so $j = 2$. Let $w_1 = 3$ calls per hour and $w_2 = 4$ calls per hour. Also, the list of ambulances is $\{1, 2, 3\}$.In other words, there are a maximum of 3 ambulances used to calculate the preparedness of a zone.Using the Sweden example, we let $\gamma^1 = 1$, $\gamma^2 = \frac{1}{2}$ and $\gamma^3 = \frac{1}{4}$. Now, $t_1^1 = 15$ minutes, $t_1^2 = 20$ minutes, $t_2^1 = 25$ minutes, $t_2^2 = 30$ minutes, $t_3^1 = 22$ minutes, $t_3^2 = 18$ minutes. Lastly, in case there is a $P2$ or a $P3$ call, each of the ambulance must reach the site in time less than $T2 = 27$ minutes and $T3 = 33$ minutes. We first attempt to calculate the initial values of $P1$ and $P2$, when there are no ambulances dispatched yet.

$P1 = \frac{1}{3}(\frac{1}{15} + \frac{\frac{1}{2}}{20} + \frac{\frac{1}{4}}{22}) = 0.034$
$P2 = \frac{1}{4}(\frac{1}{20} + \frac{\frac{1}{2}}{30} + \frac{\frac{1}{4}}{18}) = 0.020$
(each to three decimal places)

Now assume that there was a $P2$ call from zone2. We then check each of the ambulances $1, 2$ and 3, to see that only ambulances 1 and 3 qualify to arrive in zone 2 in less than 27 minutes. Since it is a priority2 call, we send the slower ambulance, which is ambulance1. Now ambulance1 is removed from the list of ambulances, and we recalculate the preparedness for each of the zones:

$P1 = \frac{1}{3}(\frac{\frac{1}{2}}{20} + \frac{\frac{1}{4}}{22}) = 0.012$
$P2 = \frac{1}{4}(\frac{\frac{1}{2}}{30} + \frac{\frac{1}{4}}{18}) = 0.007$
(each to three decimal places)

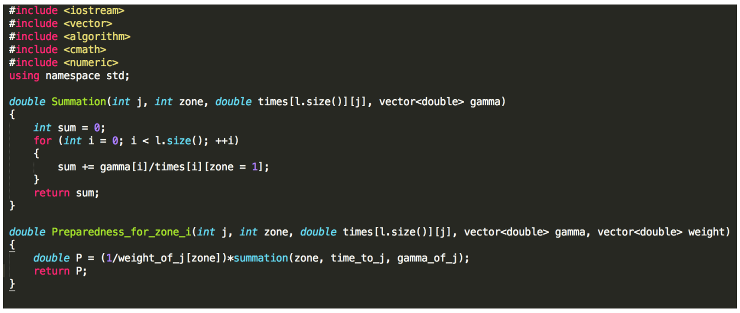
As such we can observe that the value of Preparedness for each zone decreases, and we compare the two values and obviously $0.007 < 0.012$, so $P_{min} = 0.007$.So when this value goes below a certain threshold, then we use the **Dynamic Ambulance Relocation** method that we will introduce in the next section so that we can drag up P_min to a value greater than the threshold.

Now it is relatively easy for us to type up some C++ code to simulate the dispatch algorithm. I will be presenting C++ code to highlight some important functions that may used to simulate the dispatch algorithm.

Before we begin writing code however, let us look at how we can calculate the Preparedness first. For the sample calculations we have provided earlier, note that we can organize the time data as in Table 1. Where we have the times for ambulances 1,2,3 to reach zone1 and zone2 respectively. Then it suggests that to calculate $\sum_{l=1}^{L_j} \frac{\gamma^l}{t_j^l}$, we have to create a two dimensional array that can

Zone1	Zone2
15	20
25	30
22	18

Table 1: A table illustrating our example case.



```
#include <iostream>
#include <vector>
#include <algorithm>
#include <cmath>
#include <numeric>
using namespace std;

double Summation(int j, int zone, double times[l.size()][j], vector<double> gamma)
{
    int sum = 0;
    for (int i = 0; i < l.size(); ++i)
    {
        sum += gamma[i]/times[i][zone = 1];
    }
    return sum;
}

double Preparedness_for_zone_i(int j, int zone, double times[l.size()][j], vector<double> gamma, vector<double> weight)
{
    double P = (1/weight_of_j[zone])*summation(zone, time_to_j, gamma_of_j);
    return P;
}
```

Figure 2: C++ code produced by candidate.

store the values for time. The code is shown in Figure 2. In the code, l is the set of ambulances for a particular region, and it is a vector that will be defined globally. times is a two dimensional array that stores time, weight is the vector that stores the value of weights and gamma is a vector that stores the values of gamma.

So we are now able to compute P easily as long as we have all the given parameters. However, another action that we have to take care of is the dispatch action. Let us say that one ambulance is dispatched to a zone,so this means that we remove an element of the vector l. This is done easily by l.erase(iter), where iter is the iterator (vector¡int¿::iterator iter) that points to the index of the element of l that has to be removed. If the i^{th} index of the vector l was removed, we simply subtract gamma[i]/times[i][zone - 1] from the value returned by the function Summation(int j, int zone, double times[l.size()][j], vector¡double¿ gamma). Although there are more technicalities to be covered in this program, the above is the gist of the computation of this program. The running time of this program is $O(size of vector l)$, because of the for loop, and this means that the running time increases linearly as the size of the vector of the ambulance increases. However, C++ is a compiler language, and even if the numbers become larger (as in something like 10000), the program can most likely compile within a few seconds.

2.2 Procedure for Dynamic Ambulance Relocation

Now that we have established a method of how we manage the value of Preparedness, in this section, we explain method of ”response”. In other words, how one will respond in the reallocation of ambulances to those zones/regions where the value of Preparedness is below the minimum/threshold level. Note that because of the time constraint, as well as the problem of the lack of precise real world data, we did not create a programming simulation or a sample calculation section, but instead only explained the concepts. Although we did not present concrete calculations, this model is feasible (also proved by Granberg and Varbrand) and if time allows, we can program this model as well using a **Depth first search** approach of exploring the tree.

In essence our ambulance relocation problem occurs when one or more zones under consideration goes below P_{min}. The objective is then to reach or exceed P_{min} as soon as possible. In order to achieve this we conduct a tree search based on the following mathematics (proposed by Granberg and Varbrand):
$minz \geq \sum_{j \in N^k} \tau_j^k x_j^k$, where $k = 1, ..., A$ and $\sum_{j \in N^k} x_j^k \leq 1$, where $k = 1, ..., A$. Such that $\sum_{k=1}^{A} \sum j \in N^k x_j^k \leq M$ and $\frac{1}{w_j} \sum_{l=1}^{L_j} \frac{\gamma^l}{t_j^l(x)} \geq P_{min}$ where $j = 1, ..., N$ and $x \in \{0, 1\}$.

To define the variables, z is the maximum travel time for any of the relocated ambulances, and our primary goal is to minimize the variable z. τ_j^k is the time required for ambulance k to reach zone j, and the constraints state that z has to be greater than or equal to this. The variable x_j^k equals 1 if ambulance k is relocated to zone j. Each of the ambulances can be relocated to at most one zone in the set N^k, which is representative of the set of zones to be reached by ambulance k in less than R minutes. We deduce that by setting R to be less, the set N^k and the set of feasible solutions will be smaller, but if R is too small, it could imply that there may be no solutions in some instances. Hence, R is an upper bound on the function of variable z, i.e. if R is set to y minutes, then no ambulance will have a relocation travel time longer than y minutes. Also from the constraints, we are ensured that not more than M ambulances are relocated. Lastly, $t_j^l(x)$ is a function of the variable x, which is the vector form of x_j^k. Thus, the travel time for the l'th closest ambulance to zone j, i.e. t_j^l depends on where the ambulances are located, as is decided by the values on the variable x. With all the variables explained, below is the tree algorithm that evaluates solutions to the dynamic ambulance relocation problem:

1. Let the current solution, i.e. $x_j^k = 0 \forall j, k$ be the root of the tree.
2. Let j = zone with the lowest preparedness
3. Repeat: find the n ambulances, with the minimum travel times such that they can be relocated in a way that ensures that $p_j \geq P_{min}$. We save a maximum of m zones for each of these ambulances that satisfy the conditions above. The ambulances must not have been relocated once already and definitely not more than $M - 1$ ambulances must have been moved.
4. Each of the moves in 4 gives a potential solution.
5. For all new solutions: Check if $p_i \geq P_{min} \forall i = 1, ..., N$ and check the longest travel time against the best solution found so far if this is true. If the new solution is not feasible, we create a new node and connect it to its parent solution.

6. Pick a new node and let j = zone with the lowest preparedness in the new solution.

7. We continue until there are no nodes left to examine, or there is another stop criterion.

3 Concluding remarks and Summary

Due to the lack of real world data, as well as the complexity involved in the calculation of such constants like γ, it was difficult for us to conduct an actual processing of data. However, the method presented above was adapted from an actual ambulance operating system that was deemed to be successful mostly because of two reasons: Firstly, the algorithms can be built and run quickly using programming languages such as C++. Secondly, it takes care of both the ambulance dispatch and the ambulance relocation problem. It is common for many research papers to either focus on the dispatch of EMS or focus on the relocation problem, while the other aspect is neglected. However, in this model, we present a criterion ”preparedness” that is changed depending on the ambulance dispatch actions, and also depending on this value of preparedness we adapt a dynamic ambulance relocation. Of course,there are still rooms for improvements in this model - I believe there are typically two parts: Firstly, in reality we could have an ambulance that is going to serve a P2 or a P3 call be redirected to serve a P1 call, and in this case the dispatch algorithm would become more complicated. Secondly, in terms of the ambulance reallocation problem, we may also try methods such as mixed integer linear program (Lau) or integer programming model(Bandara) and compare which models provide a greater chance of patient survivability and a more efficient response time.

4 bibliography

References

[1] Granberg, T.A and Verbrand, P, ”Decision support tools for ambulance dispatch and relocation” *Linköping University Post Print***1-9**, 195-201, 2007