Team Stop_NATO

PRANAV KANNAN - 2019111033 - pranav.kannan@research.iiit.ac.in
KSHITIJ MISHRA - 2019111014 - kshitij.m@research.iiit.ac.in
YASH AMIN

Algorithmic disaster preparedness for Roadways and Railways.

April 20, 2022

Algorithmic disaster preparedness for Roadways.

Overview

India ranks first in the number of road accident deaths across the 199 countries and accounts for almost 11% of the accident related deaths in the World. A total number of 449,002 accidents took place in the country during the calendar year 2019 leading to 151,113 deaths and 451,361 injuries.

In such a scenario, having a preparedness strategy for immediate disaster response plays a crucial role in saving numerous lives.

The most important functional factor of emergency response vans is having a preparedness plan for dealing with disasters and emergencies such as road traffic injuries. There is a gap in the design and the development of a valid and reliable tool to evaluate the levels of Emergency response van preparedness during road traffic injuries with mass casualty.

The aim of this study was to explore a valid and reliable tool for measurement of Emergency response van preparedness in road traffic injuries with mass casualties.

Motivation

Road accidents are unplanned and sudden mishaps that result in major causalities, damage or even death. Hyderabad is a bustling city and on every day basis hundreds of people lose their lives in car, bike, bus and other forms of road accidents.

Accidents can’t be prevented but immediate response can most certainly be provided to the affected people.

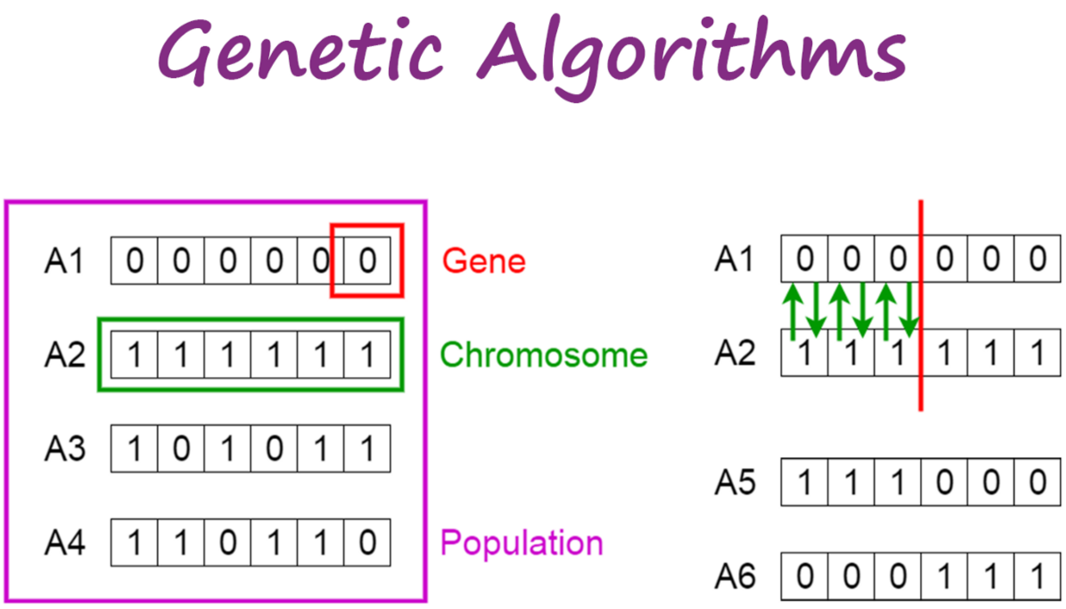
Let’s take the example of Delhi, Delhi’s total road length is about 28,504 Km’s creating a network of emergency response vans is not an easy task. Managing that network is also not an easy task to execute. In such a case figuring out an effective disaster response point is the first step towards solving the problem.

The disaster response point can be either a headquarter or even a hospital for that case to help in immediate response in case of a road accident or even a fire hazard that has broken loose.

Theory

GENETIC ALGORITHM

A genetic algorithm is a search heuristic that is inspired by Charles Darwin's theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.



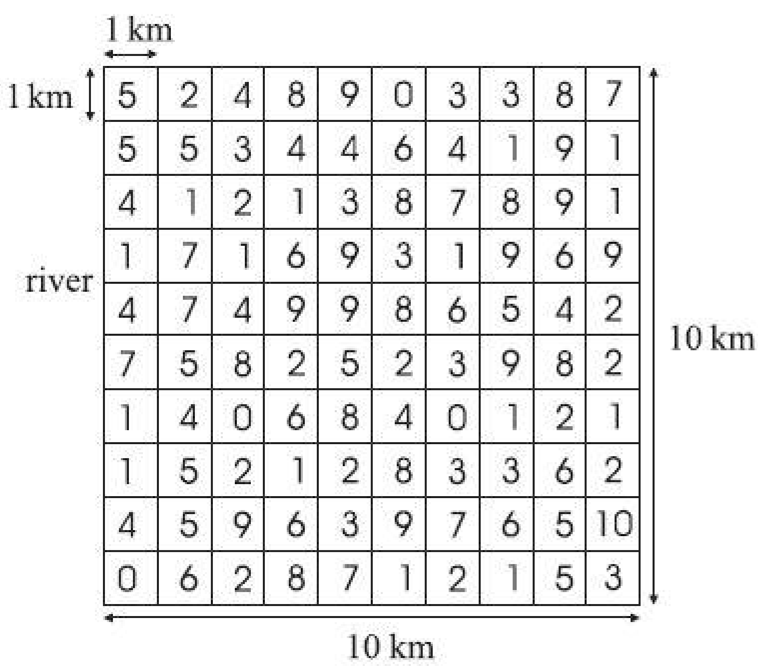
Notion of Natural Selection

The process of natural selection starts with the selection of fittest individuals from a population. They produce offspring which inherit the characteristics of the parents and will be added to the next generation. If parents have better fitness, their offspring will be better than parents and have a better chance at surviving. This process keeps on iterating and at the end, a generation with the fittest individuals will be found.

Our proposed genetic algorithm consists of

- Initial population
- Selection
- Crossover
- Mutation

Proposed Method



Consider this grid as a system of roads with the grid representing the city as a whole; each square in the grid has a specific value assigned to it. This specific value represents the accident/disaster frequencies which are associated with that particular region of the grid.

For this simulation we assign the squares random accident frequencies and we calculate the cost of each square.

$$cost = \sum_{n=1}^{100} w_n \sqrt{(x_n - x_{fs})^2 + (y_n - y_{fs})^2}$$

where

(x_n, y_n) = coordinates of the center of square n
(x_{fs}, y_{fs}) = coordinates of the proposed emergency response unit

Using this cost function we calculate an aggregate sum of all the possible paths of a square. This aggregate sum represents the cost of the square. This cost is what we are trying to minimize using genetic algorithms.

How does GA work here ?

Genetic algorithm works in 3 principles namely Selection, Mutation, Crossover.

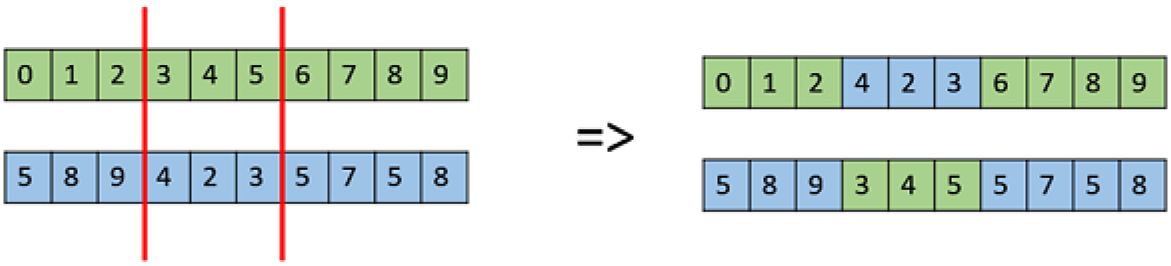
Selection

In the first pass we will determine the costs of N different squares this N can be controlled by the code.

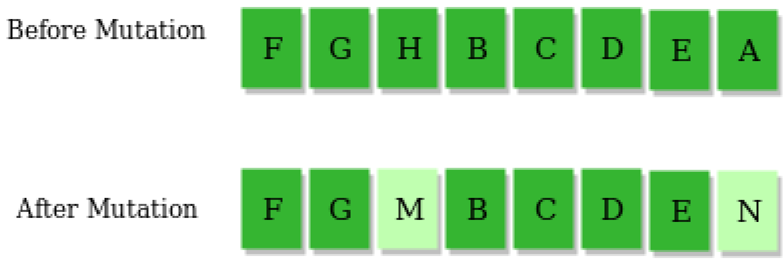
After the first pass we sort out the costs in ascending manner, in this way we get our best seed vectors on the top. Now we perform crossover along with mutation.

Crossover

After the first selection we pick out the top M vectors and perform crossover on them.



Point (x,y) on crossovers with point (x1,y1) will yield in point (x,y1) and (x1,y) ; these are the 2 new children seeds. We can also perform **mutation** on them by playing around with the x,y values. For eg in our case we've played around like (x1*3%49,y*3%49) and (x*3%49,y1*3%49).



Now using the newly formed children, we again calculate their costs using the cost function. These costs will now again be arranged in an ascending manner along with the parents and then again the top M vectors will be chosen. This process is called generation. In our simulation we generate 15 generations worth of seeds and determine the lowest cost node in the city.

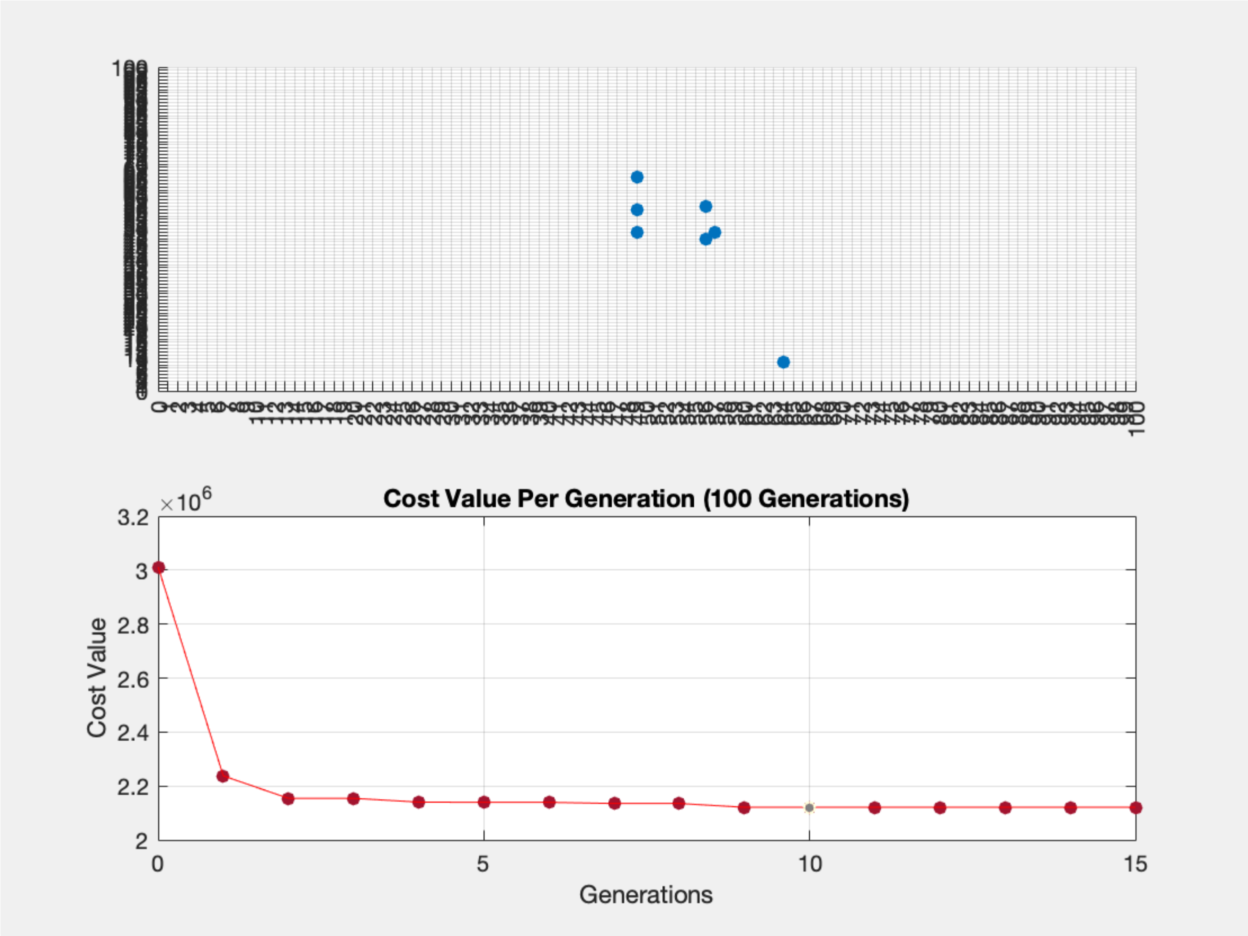
Output

Using Truncation Selection, the most efficient location will be found right at generation 1. A simpler approach in solving this problem is to evaluate all the locations, then sort them by efficiency. Since Truncation Selection uses sorting, it makes sense that it will get the best location right at generation 1.

After we have received 15 generations our final output will be indicative of the cell which has the least value of cost. Due to computational costs right now we are only running a grid of 50*50.

But the real advantage of this algorithm is when the number of cells are very high in number and computation time is something that we can't afford because of a disaster impending.

Results



Here is the result from our simulation with 2500 points on the top with the specific points highlighted in each of the generations.

The graph below indicates the decrease in the cost value with each iteration. Cost comes to a saturation after a few generations as we converge to the least cost vector.

Generations	ProposedCoordinates		CostValue
0	64	9	3.0118e+06
1	49	66	2.2377e+06
2	56	57	2.1543e+06
3	56	57	2.1543e+06
4	57	49	2.1403e+06
5	56	47	2.1393e+06
6	56	47	2.1393e+06
7	49	56	2.1351e+06
8	49	56	2.1351e+06
9	49	49	2.1212e+06
10	49	49	2.1212e+06
11	49	49	2.1212e+06
12	49	49	2.1212e+06
13	49	49	2.1212e+06
14	49	49	2.1212e+06
15	49	49	2.1212e+06

This is the output after the GA algorithm, here each generation along with it's best fit vector is shown along with their costs.

We can observe that after a few iterations the entire algorithm converges to one specific point because it's cost is the least of the lot.

Algorithmic disaster preparedness for Railways.

Overview

Indian Railways has an organized system of relief for managing accidents and disaster with its own resources with more than 6 thousand trained breakdown staff, 176 Accident Relief Trains (ARTs) and 86 Accident Relief Medical Vans (ARMVs) made up of passenger coaches. IR is a set of sixteen Zonal railways. Every Zone of IR is responsible for readiness and dispatching ARMVs in accidents occurring in their area of control. The present study will examine the current ARMV asset management of North Eastern Railway (one of the Zone of IR) which is the share of 3032 route-kms of the Indian Railways network.

Due to suboptimal stationing of ARTs/ARMVs, it takes much more time than the minimum stipulated time. Measurement of emergency preparedness plans to handle any unpredicted disaster and delivering rescue operations by all means within a stipulated time is a desirable index for a developed state. The situation of medical infrastructure and facilities in developing and emerging countries like India is quite different from developed countries where emergency medical services such as ambulances are scarce available only to cities and towns so Railway transporter, like Indian Railways (IR) takes care of train accident victims with its own resources of railway train-ambulance(ARMV) and doctors.

Methodology

In graph theory, a network is depicted with a graph, where the nodes are the vertices and the links are the edges of that graph. Graph is an ordered pair $G = (V, E)$, where $V = V(G) \neq \Phi$ is the set of vertices and $E = E(G)$ is the set of edges such that for any two vertices $u, v \in V$ there exists $e \in E$ which connects the two endpoints u, v. Two connected vertices are adjacent vertices and we say that the edge and its endpoint are incident. The number of edges incident to a given vertex is the degree of that vertex.

In the graph NE Railway having 41 nodes and 44 edges the edges are weighted as distances. Edge can be weighed by many other factors besides distance. Its purpose is to cluster the graph so that one ARMV can be provisioned for each cluster. The motive is to cluster the graph so that for each cluster one ARMV can be provisioned. Presently NER is using 8 ARMVs based at eight locations so 8 clusters have been iterated using Girvan-Newman Algorithm, and then based on the criteria provided in the code, we suggest the most suitable locations in the clusters for deployment of ARMV.

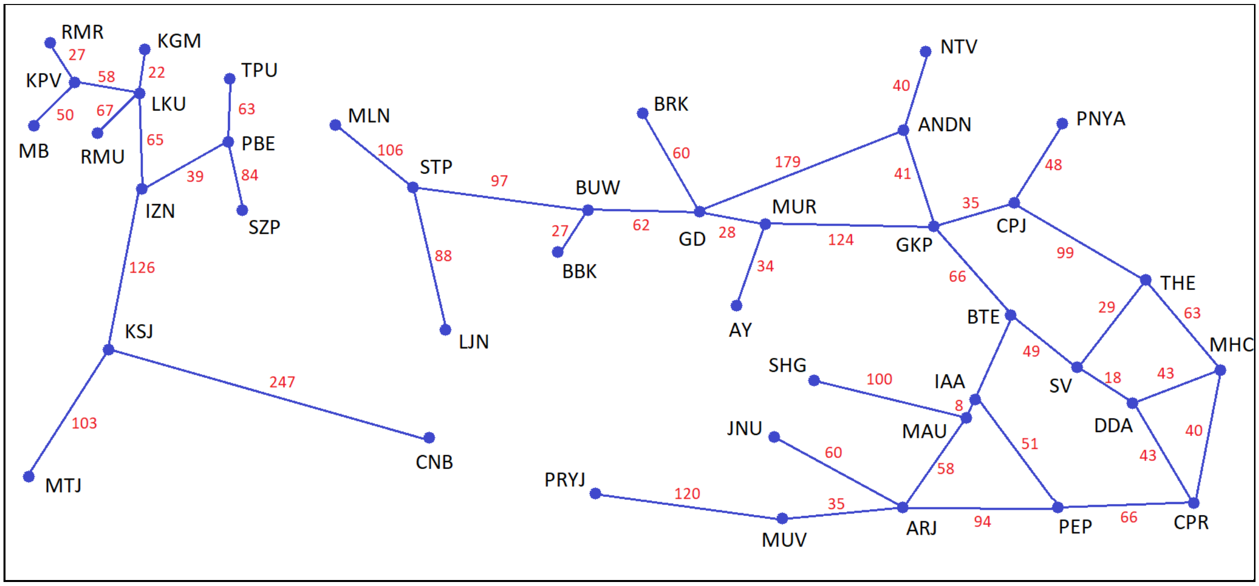
Girvan-Newman Algorithm

Since clustering algorithms work on determining the communities by taking edge weight into account, it is necessary to choose significant factors that might influence edge weight. The most important factor turns out to be the length of the track between two nodes. Greater length implies more necessity for ARMV on either end of the edge. This is intuitive but a suitable argument for this to be true is non availability of medical relief facilities between two nodes. Other factors include age of tracks, maintenance schedules, accident history, traffic density, average age of locomotives, wagons and coaches running on track, single/double tracks, diesel/electric traction and maximum permissible speed among others. If the weight of tracks is w and factors affecting them are x, then, (where a, are constants for unit and range adjustments).

$$w = \sum_{i=0}^{n} a_i x_i$$

Most of the graph clustering algorithms like K-means and K-medians take node weights into account while we require algorithms that make clusters based on edge weights. One most suitable algorithm is the Girvan-Newman algorithm which is a hierarchical method for detecting communities in networks.

The Girvan-Newman algorithm is run for one less time than the number of ARMVs available. This creates clusters equal to the number of ARMVs.



The algorithm removes the least important edge from the graph and hence converts the fully connected graph to a disjoint graph. This disjoint graph consists of 2 connected graphs and on subsequent application of the algorithm this graph gets further disjoint. For N-1 iterations, the graph yields N disjoint graphs which we call clusters.

The algorithm works by first making the clusters using Girvan-Newman method and finding the suitable location in the clusters by finding the node with maximum strength. In a weighted network, the strength of nodes is calculated by summing the weights of its adjacent edges.

If two or more nodes turn out to have the same strength then node with higher degree is chosen. If the cluster has only one node, then it is directly taken to be the preferred node. The pseudo code in figure 2 was run using Python3. The Networkx library is used for running the Girvan-Newman algorithm. G is the parameter of graph which is defined using the Networkx library as G = networkx.graph(). The graph is then initialised with the elements i.e., the nodes as G.nodes() and the edges as G.edges() between the nodes.

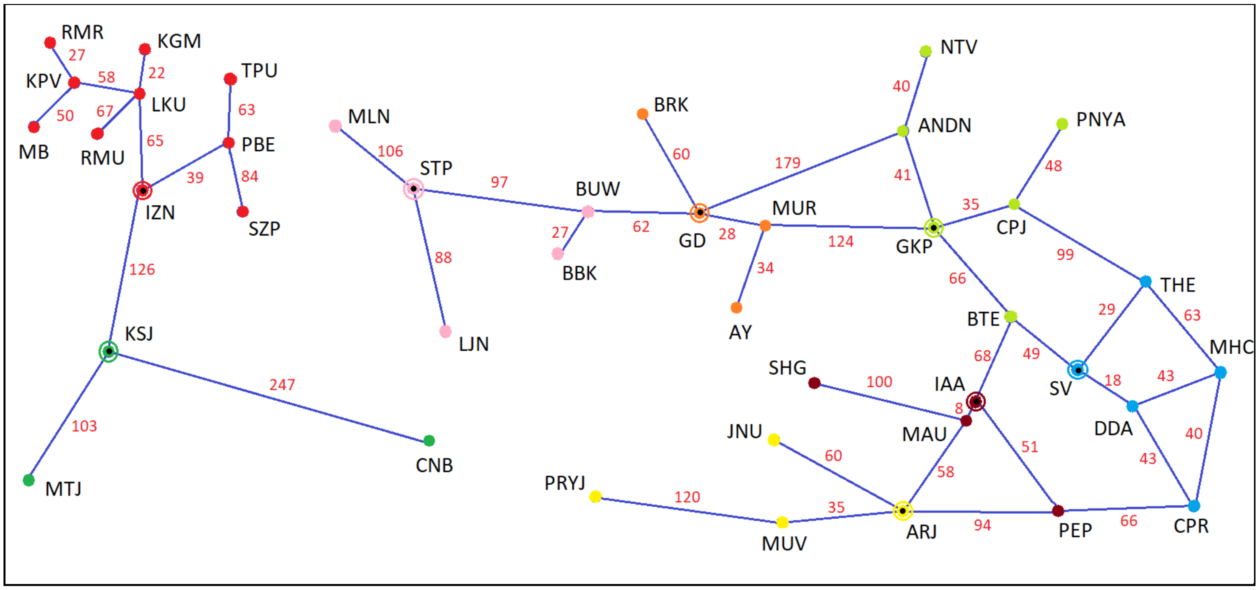
Algorithm: Determining locations for stationing Relief Trains

Input: Stations as nodes, tracks as edges, weight of edges and number of Relief Trains(N)

Output: Nodes suitable for Relief Trains as relief_train_location

```
graph G ⟵ node
graph G ⟵ edge
    edge ⟵ edge_weight
    for i ⟵ 1 to i ⟵ N-1 do
        procedure girvan_newman
            clusters ⟵ girvan_newman(G)
    for j ⟵ 1 to j ⟵ N-1 do
        procedure finding_stations
            if nodes_in_cluster > 1 then
                procedure find_maximum_weight_node
                    relief_train_location ⟵ maximum_weight_node
                if maximum_weight_node > 1 then
                    procedure find_highest_degree_node
                        relief_train_location ⟵ highest_degree_node
            else if nodes_in_cluster = 1 then
                relief_train_location ⟵ node
print relief_train_location
```

Then the edge weight is defined for the edges as the numerical values of the distance between the nodes that are connected. Now since the Girvan-Newman algorithm is an iterative algorithm, we need to run this algorithm on the graph G for N-1 iterations where N is the number of clusters we want depending on the number of ARMVs available, in our case we do 7 iterations to make 8 clusters for the network.



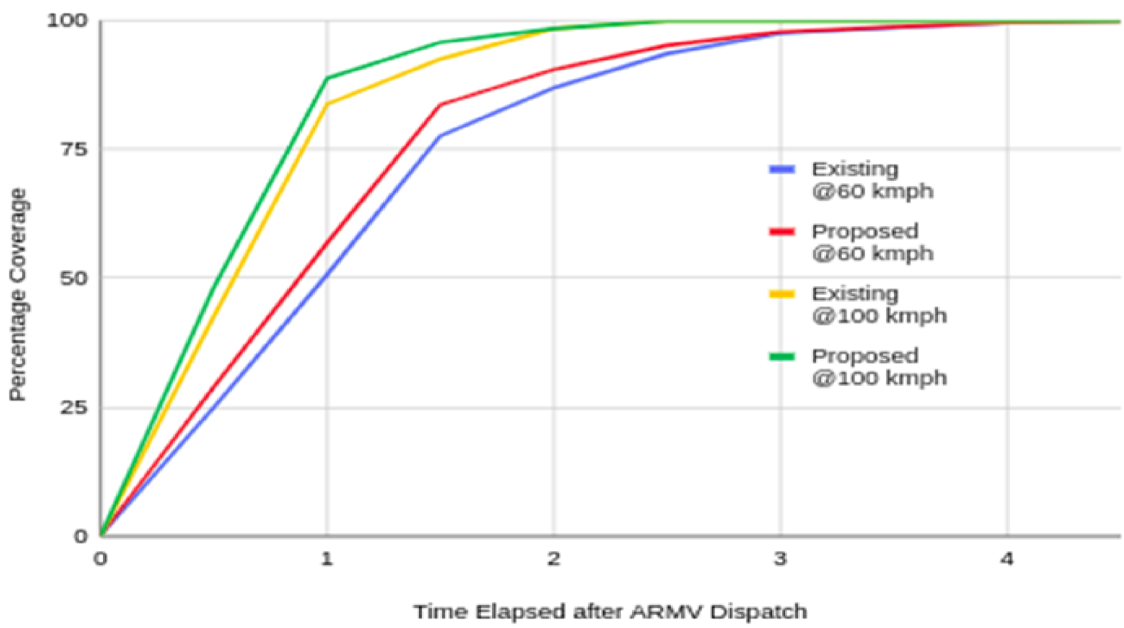
Results

ARMV parking is assigned to only those nodes where crews/drivers are available for 24X7 hours and the distance in KM between these nodes is taken as the edge weight of the graph.

In this paper, we have assumed that accidents occur randomly and probabilistically over a network, but in most real-life problems the assumption of traffic and train density becomes significant factors to be considered as edge weights because the numbers of train services are not homogenous over the network.

Clusters	**Existing Locations**	**Proposed Locations**
MLN, STP, LJN, BUW, BBK	LJN	STP
BRK, GD, MUR, AY	GD	GD
GKP, ANDN, NTV, CPJ, PNYA, BTE	GKP	GKP
SV, THE, DDA, MHC, CPR	CPR	SV
IAA, MAU, SHG, PEP	MAU	IAA
JNU, ARJ, MUV, PRYJ	MUV	ARJ
CNB, KSJ, MTZ	KSJ	KSJ
KPV, MB, RMR, RMU, LKU, PBE, IZN, KGM, TPU, SZP	LKU	IZN

After a disruption occurs, dispatchers and operating personnel need a certain amount of time to investigate the situation and decide on appropriate measures to assess remaining resources. This investigation, response time of rescue staff and decision-making time affects the response time statistics of the ARMV dispatch.



This transition to stable operation phase can be studied to create disruption programs, especially for large infrastructure disruptions. The possibility of incorporating faster reaction times into weights may also be a good area for future research.

Result shows by redeployment and relocation of existing 08 ARMVs to the locations proposed in Cluster Coverage Model, for one hour stipulated time, 12% and 6% more coverage can be enhance by the ARMVs travelling with speed of 60 kmph and 100 kmph respectively,

Solution shows that total 10 ARMVs (@100 kmph speed) are capable for the coverage of 98% of the given network whereas the results of the proposed new cluster coverage model shows that after the proposed relocation, deployment of only 08 number of ARMV will give coverage of 88.84% on the given network with the same speed.

References

Goldberg, J.B. (2004). Operations Research Models for the Deployment of Emergency Services Vehicles. EMS Management Journal, 1(1):20–39, January-March 2004.

Khan, B.S., and Niazi, M.A. (2017). Network Community Detection: A Review and Visual Survey. CoRR. Vol. abs/1708.00977, http://arxiv.org/abs/1708.00977.

Newman, M. E. J. (2006). Fast algorithm for detecting community structure in networks. Phys. Rev. E .Vol 69,66-133 doi = {10.1103/PhysRevE.69.066133}

Newman, M. E. J., and Girvan, M. (2004). Finding and evaluating community structure in networks. Physical Review E - Statistical, Nonlinear and Soft Matter Physics 69, 2 Pt 2, 16.

Report No.13 of 2016 (Railways) https://cag.gov.in/uploads/download_audit_report/2016/Chapter_6_Disaster_Management.pdf