International Journal of Scientific Research and Management (IJSRM)
||Volume||06||Issue||12||Pages||EC-2018-137-147||2018||
Website: www.ijsrm.in ISSN (e): 2321-3418
Index Copernicus value (2015): 57.47, (2016):93.67, DOI: 10.18535/ijsrm/v6i12.ec01

Intelligent Recommendation module for more than one emergency vehicles node to destination node

Emad I. Abdul Kareem [1], Farah Younus Safar [2]

[1]Department of Computer Science, Mustansiriyah University, Iraq,
[2]Department of Computer Science, Mustansiriyah University, Iraq

Abstract: Traffic congestion may impede the power of emergency vehicles to arrive at the site of a traffic incident or other catastrophe in a fast and true way. Id's may not be a short one, but it is the optimal path that leads to the scene without passing in con-gested streets. Streets may be congestion for a long time and lead to not reach emergency vehicles to the destination place while other emergency vehicles can reach to the same destination place in fast which is far from the destination with uncrowded streets. This makes the emergency vehicle to reach the destination in fast. In this paper, The proposed module rep-resented in a map to show the algorithm for more than one source nodes to the destination node. In addition, a case study of a real city in Baghdad has been used to evaluate the proposed module. The results show the proposed module ability to find the optimal path with 27 to 204 seconds to give a clear recommendation for which vehicle must be moved to the destination node.

Keywords: Emergency vehicles, Traffic congestion, Simulated annealing,Optimization algorithms,Transportation systems.

1. Introduction

Congestion may prevent the ability of emergency vehicle services to reach destination in fast and at the earliest time. If there is more than one emergency vehicle in the same section, it is able that the far vehicle can reach the destination scenes faster than the nearest one. Often, this case can happen if there is a congestion in the shortest path. For those reasons the shortest path does not always lead to optimal path, emergency vehicles need to well-planned recommendation services to choose properly the emergency vehicle that must move and the appropriate route before [1].

1.1 Simulated annealing algorithm

Simulated annealing inspired by physical phenomena which is annealed in metallurgy is a metaheuristic algorithm, simulated annealing is used to search for feasible solutions and converge to an optimal solution. The law of thermodynamics state that at a temperature, *t*, the probability of an increase in energy, ΔE, is given by

$$P(\Delta E) = \exp(-\Delta E / kt) \quad (1)$$

Where *k* is Boltzmann's constant. The simulation in the Metropolis algorithm calculates the new energy of the system. If the energy has increased then the new state is accepted using the probability by the formula (1). While if the energy has decreased, then the system moves to this state. At each iteration a certain number are carried out at temperature and then the temperature is decreased. This is repeated until the system cools into a steady state. This equation is used in simulated annealing, the Boltzmann constant ignored. Therefore, the probability of accepting a worse state is given by the equation

$$P = \exp(-\Delta E/T) > r \quad (2)$$

Where
ΔE = the change in the evaluation function
T= the current temperature
r = a random number between 0 and 1

The probability of accepting a bad move is a function of both the temperature of the system and of the change in the cost function. It can be appreciated that as the temperature of the system decreases the probability of accepting a worse move is decreased. This is like the gradually moving to a cool state in physical annealing. Note, that when the temperature is zero, then only better moves will be accepted which effectively makes simulated annealing act [7].

Algorithm (1) pseudocode for simulated annealing [8].

Input: problem size, iteration $_{max}$,T $_{max}$
Output :S $_{best}$
1 S $_{current}$ ← create an initial solution(problem size);
2 S $_{best}$ ← S $_{current}$;;
3 for i = 1 to iteration $_{max}$ do
4 S_i ← CreateNeighborSolution(S $_{current}$);

```
T curr ← CalculateTemperature(i, T max);
   if Cost(S i) ≤ Cost(S current) then
   S current ← S i;
         if Cost(S i) ≤ Cost(S best) then
               S best ← S i;
         end if
     else if exp( Cost(S current) Cost(S ) T i ) > r () then
      S current ← S i;
      end
end for
```

2. Literature Review

Some researchers introduced an algorithm based on ant colony optimization algorithm to discover the best route that minimizes the time while taking into consideration the status of problems that can appear each time such as traffics, catastrophes natural, etc. [1]. Also the optimization of EMS vehicle fleet allocation and base station location used the genetic algorithm (GA) with an integrated EMS simulation model and introduced the optimization heuristic and objective function used in conjunction with the simulation model [2]. Some researchers introduced most comfortable route, which is not necessarily mean to be the shortest one, they prepare an early research on the path recommendation system based on street view images [3]. Emergency medical service (EMS) providers participating in vehicle crash-induced incident management aims to supply a full and effective coverage as possible to satisfy the need for accident responses effectively; however, the design and provision of efficient and cost-effective services are tough issues faced by emergency management authorities. Used a double standard model (DSM), along with a genetic algorithm (GA) for assigning EMS fleet from vehicle locations to intersection vehicle crash sites [4]. They present recommendations for multiple-unit dispatch to multiple call priorities based on simulation, optimization and heuristics [5] . Intelligent Transportation System technology is the application of information technology that is used to develop the efficiency of transportation systems. Solve the problems of traffics, enhancement of safety and mobility for transportation where it decreases the environmental impact of transportation. It depends on a wide range of technologies and functions such as: camera system, digital mapping, communication systems, data acquisition and exchange, detection and classification, artificial vision and In-vehicle system [6]. In this paper a simulated annealing algorithm has been used. The main reason of using simulated annealing over other methods is its ability to escape from become trapped in local minima. It uses a stochastic search which not only accepts several changes that decrease the objective function but also its accept several changes that increase it. Thus, in this paper, the result shows that the proposed module with more than one emergency vehicle able to recommended the optimal vehicle and the optimal path. In addition, a case study have been taken to evaluate the module performance in a real city in Baghdad.

This recommendation module uses to find the optimal path from two (or more than two) emergency vehicles source to destination. Figure (1) shows the flowchart of the recommended module from two different sources to one destination. It uses modified simulated annealing algorithm which is considered as two simulated annealing with the same iteration loop and it can use more than one simulated annealing according to the number of vehicle that is used in the module, to determine the optimal path for each vehicle and choosing the appropriate vehicle that should be moved to the destination node avoiding any crowded streets. In this recommendation module the nearest vehicle may not be able to reach the destination node in the optimal path. The recommended module will help to choose the best vehicle. Algorithm (2) illustrates the optimal path using a simulated annealing algorithm from two different emergency vehicles, sources to one destination and determine the optimal emergency vehicle that reach faster to the destination scene.

3. Model formulation

This recommendation module uses to find the optimal path from two (or more than two) emergency vehicles source to destination. Figure (1) shows the flowchart of the recommended module from two different sources to one destination. It uses modified simulated annealing algorithm which is considered as two simulated annealing with the same iteration loop and it can use more than one simulated annealing according to the number of vehicle that is used in the module, to determine the optimal path for each vehicle and choosing the appropriate vehicle that should be moved to the destination node avoiding any crowded streets. In this recommendation module the nearest vehicle may not be able to reach the destination node in the optimal path. The recommended module will help to choose the best vehicle. Algorithm (2) illustrates the optimal path using a simulated annealing algorithm from two different emergency vehicles, sources to one destination and determine the optimal emergency vehicle that reach faster to the destination scene.

Algorithm (2) of the proposed module

Input: nodes (matrix of traffic intersections)
b (matrix of streets)

v1 (vehcile1 the source1node),v2(vehcile2 the source2node), snk(destination node)
Temp (temperature) ,T0 (initial temperature) , alpha (cooling factor)
Maxite (the number of iteration in each tempretaure)
asegments(matrix of street distances)
Crowd (matrix of (0,1) data from module monitor , 0 for empty streets ,1 for crowded street)
Penalty (add to distance of crowded street)
Output: optimalpath

Step1: pth1 = random path from v1 to snk
Step2: for i=1 to end element of pth
If pth consist of crowd street, then
Add penalty to the distance of crowded path
dp1= dp1 + distance of street + penalty
Else dp1 = dp1 + distance of street
End if
End for
Step3: determine bestcost1=dp1, optimalpath1=pth

Step 4: ppth = random path from v2 to snk

Step 5: for i=1 to end element of ppth
If ppth consist of crowded streets, then
add penalty to the distance of crowded path
dpp= dpp + distance of street + penalty

Else dpp= dpp + distance of street
End if
End for
bestcost2=dpp
optimalpath2=ppth

Step 6: applying SA algorithm
While Temp > T0
For m=1 to Maxite

Step 7: pth2= random path from v1 to snk
for j=1to end element of pth2
If pth2 consist of crowded streets then add penalty to the distance of crowd street
dp2=dp2+distance of street+ penalty
Else dp2=distance of street + penalty
End if
End for

Step8: delta1 = dp2- dp1

Step9: if (dp2 < dp1) and exp (-delta1) < random [0,1], then
pth=pth2 , dp=dp2
End if

Step 10: while dp < bestcost1
optimalpath1=pth
bestcost1=dp
End while
End If

Step 11: ppth2= random path from v2 to snk

Step 12: for j=1to end element of pth2
If ppth2 consist of crowded streets then add penalty to the distance of crowd street
dpp2=dpp2+distance of street+ penalty
Else dpp2=dpp2+distance of street
End if
End for

Step13: delta = dpp2- dpp

Step14: if (dpp2 < dpp) and exp (-delta1) < random [0,1], then
ppth=ppth2 , dpp=dpp2
End if

Step15: while dpp < bestcost2
optimalpath2=ppth
bestcost2=dpp
End while
End If
End for

Step16: Temp=Temp*alpha
End while

Step 17: select the lowest between bestcost1 and bestcost2
If bestcost1 < bestcost2, then
Optimal vehicle that must move from source to snk is (optimalpath1)
Else Optimal vehicle that must move from source to snk is (optimalpath2)
End if

Step 18: end

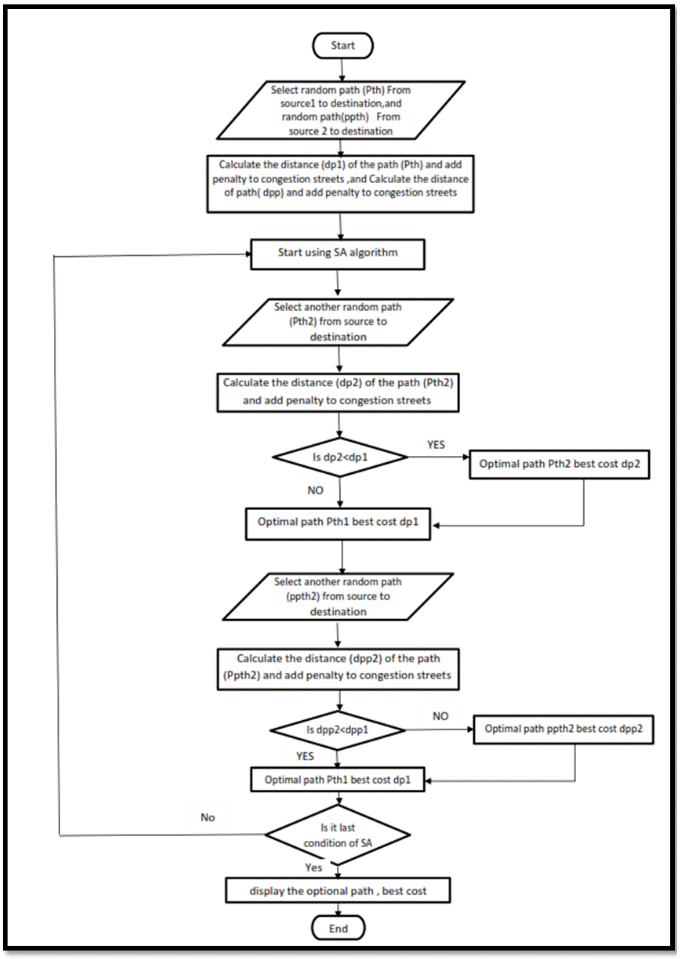

Fig. 1: Flowchart for the proposed recommendation module from more than one source node to destination node.

For Step1, The proposed method will get a random path from a source node to destination node for the first vehicle (v1). In Step 2, the proposed method will calculate the summation of each street distance and adding the penalty value to the distance of the street if it is crowded. dp1 is the summation of streets distances of the path (pth1). Step3, determine the bestcost1 as dp1 and optimalpath1 as (pth). For Step4, the proposed method will get a random path from a source node to destination node for the second vehicle (v2). In step 5, the proposed method will calculate the summation of each street distance and adding the penalty value to the distance of the street if it is crowded. dpp is the summation of streets distances of the path (ppth). Determine the bestcost2 as dpp and optimalpath2 as ppth. For Step 6, the proposed method start using SA algorithm to obtain an optimal path for vehicle (V1) and for vehicle (V2), it starts with a high temperature and minimizing it slowly with iteration in each temperature to get optimization solution. In step7, (pth2) is another random path from source to destination, it uses to compare it with the previous path (pth1) and calculate the distance of (pth2), and it must add a penalty to the distance of the street which it crowded. dp2 is the summation distances of (pth2) streets. In step 8, calculate the differences between dp2 and dp1 (delta). In step9, there is a condition, if the new cost (dp2) is lower than (dp1) and exp (-delta / temperature) is lower than the random number between zero and one ,then accepted new cost and change the values of dp1 and pth1 to new. In step10, comparing dp1 with the bestcost1, if it is lower than bestcost1 then changes the value of bestcost1 to the value of dp and determine the optimal path. In step 11- (ppth2) is another random path from source to destination to compare it with the previous path (ppth). For step 12, calculate the distance of (ppth2), it must add a penalty to the distance of the street which it crowded. dpp2 is the summation distances of (ppth2) streets. Step 13, calculate the differences between dpp2 and dpp (delta). For step 14, there is a condition if the new cost (dpp2) is lower than (dpp) and exp (-delta / temperature) is lower than the random number between zero and one, then accepted new cost and change the values of dpp and ppth to new.In step 15, comparing dpp with the bestcost2, if it is lower than bestcost2 then change the value of bestcost2 to the value of dpp.and determine the optimalpath. Repeat step 6, step 7, step 8, step 9, step 10, step 11, step 12, step 13, step 14 and step 15 until the iteration of Maxite is completed. For step 16, minimize the temperature slowly, repeat step 6, step 7, step 8, step 9, step 10, step 11, step 12, step 13, step 14, step15 and step 16 until the temperature cooling and reach to the value of T0. In Step 17, comparing between bestcost1 and bestcost2 to determine the appropriate emergency vehicle that must move to the destination node. This algorithm used to appear path for vehicle V1 and path for vehicle V2, the modified simulated annealing algorithm work efficiently, which it gets the result of two paths for two vehicles and to determine the appropriate vehicle that must move to the destination. This algorithm can use for more than two vehicles by adding the values of any addition vehicle in the algorithm in the same loop of simulating annealing algorithm.

4. Result and analysis

The algorithm of an intelligent recommendation module has been evaluated using three maps to determine the optimal path. Figure (2) illustrates three tests for three different maps, figure 2.A, 2.B and 2.C shows three maps with 12,15 and 30 traffic intersections respectively. Figure (4) shows a test map from three source node to the destination node. The result appears in a few seconds. The time that it takes to appear results depends on the number of iterations and the value of temperature that used in simulated annealing, also the time depend on the number of vehicles used in a test map. The total time for the map with 12 traffic intersection is 5.19 seconds and the optimal vehicle is vehicle v2 while the total time of the map with 30 traffic intersection is 8.24 seconds and the optimal vehicle is vehicle v1. In addition the spent time of three vehicles is 15.82 seconds.

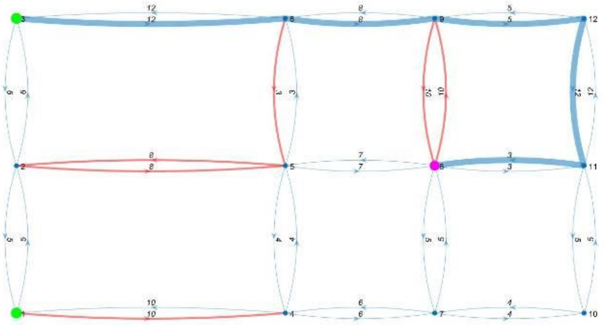

Figure. 2.A: Map of 12 traffic intersections.

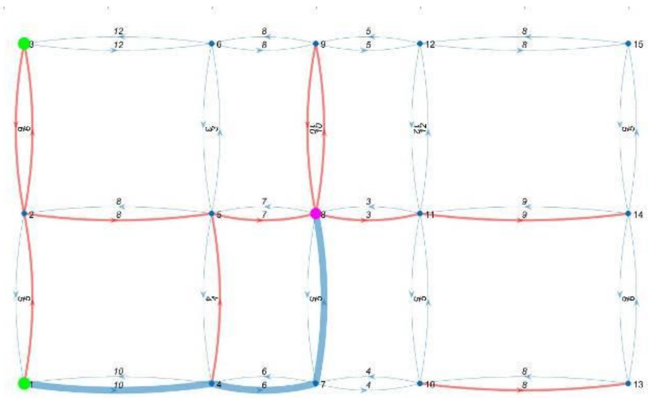

Figure: 2.B: Map of 15 traffic intersections.

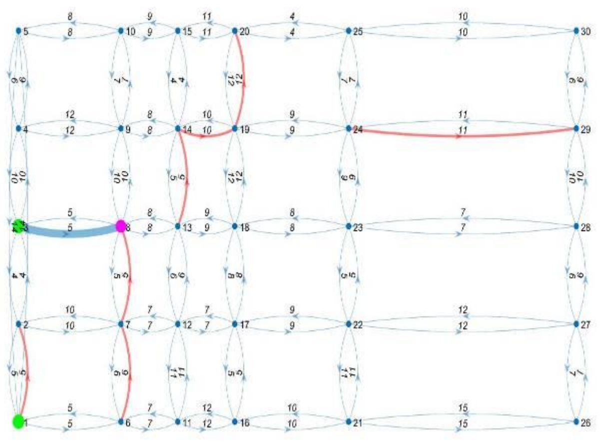

Figure:2.C: Map of 30 traffic intersections.

Figure (2.A) represents the optimal path in a map consist of 12 traffic intersection. There are two vehicles (two source nodes), one from node 3 which is represents vehicle one, and the other from node 1, the optimal path represented as blue path and the red paths represented crowded streets. Vehicle number one is the best to move to the destination node 8 due to track less distance and no crowded streets will pass. Figure (2.B) represents the optimal path in a map consist of 15 traffic intersection, two vehicles one from node 3 represents vehicle one, and the other from node 1, the optimal path represents that vehicle two is the best to move to the destination node 8 due to track less distance and no crowded streets will pass. Figure (2.C) represents the optimal path in a map consist of 30 traffic intersection, and 100 streets. There are two vehicles (two source nodes), one from node 3 which is represents vehicle one, and the other from node 1. The optimal path represents that vehicle one is the best to move to the destination node 8 due to track less distance and no crowded streets will pass. The table (1) represents two vehicles all maps in figure (2) with the traffic intersections, random number of crowded streets, the source node (1), different values of destination nodes, different optimal paths for each vehicle, distances of optimal path for each vehicle and the total time for each traffic intersection. The time is different for each traffic intersection. The total time for the map of 12 traffic intersection is 5.19 second while the total time for the map of 30 traffic intersection is 8.24 second. The time will be increased as soon as the number of traffic intersections will be increased.

Table 1: The Performance of all maps, v1 and v2 represent vehicle1and vehicle2

Traffic intersection no.	Destination Node	V1	Optimal path v1	Distance v1	V2	Optimal path v2	Distance V2	Total time/ Sec
12	8	3	3-6-9-12-11-8	51	1	1-2-3-6-9-12-11-8	40	5.19
15	8	3	3-6-5-4-7-8	30	1	1-4-7-8	21	6
30	8	3	3-8	10	1	1-6-11-12-13-8	37	8.24

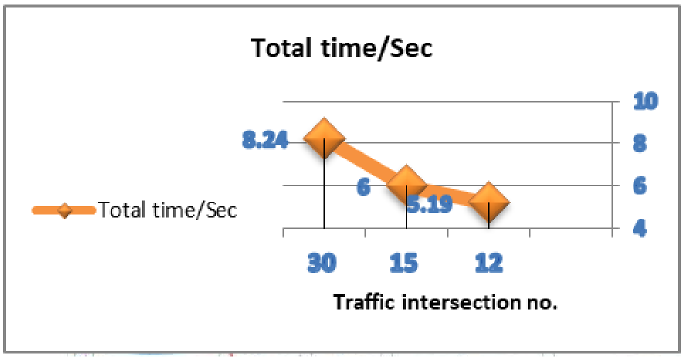

Figure 3: Illustrates performance criteria for all test maps

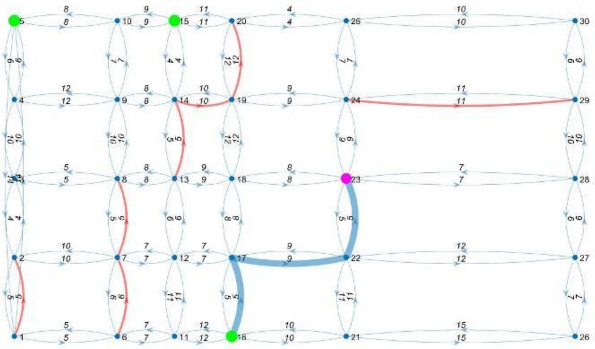

Figure 4: **A map with Three emergency vehicles.**

4.2 Case study: Sadr city

A big city in Baghdad, which it Al-Sader City has been taken as a real case study. Figure (5,6) are maps of Sadr city which it got from Google map. It consist of 89 traffic intersections and 312 streets. The city includes two hospitals (Jwader hospital) and (Martyr Sadr hospital). Only these hospitals have emergency vehicles, a recommendation module with two source point has been used to obtain the optimal vehicle and path from source (hospital) to destination scene. Sadr city has been simulated with crowded street started from 10% crowded streets to 90% crowded streets.

Figure 5: **Sadr city from google map.**

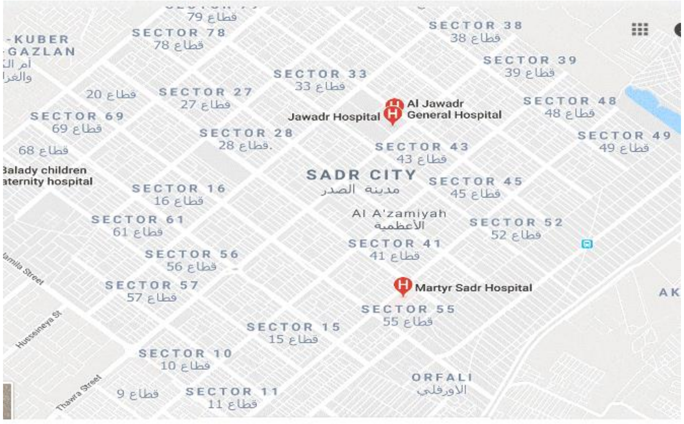

Figure 6: Sadr city from google map.

Figure (7) represents 10% crowded streets. Thus, there are 31 crowded streets, source nodes are (37, 72), the random destination node is 10. The blue path represents the optimal source node and path, table (2) in the Appendix illustrates the optimal path of source nodes to the destination node. Depending on the distance of each node, the module determines the optimal path as well as a better emergency vehicle with its optimal path. Figure (8) represents 20% crowded streets. Thus, there are 62 random crowded streets, table (3) in the Appendix illustrates the optimal path of source nodes to random destination node such as node 23. The blue path represents the optimal source node and path.

Figure (9) represents 30% crowded streets.Thus, there are 93

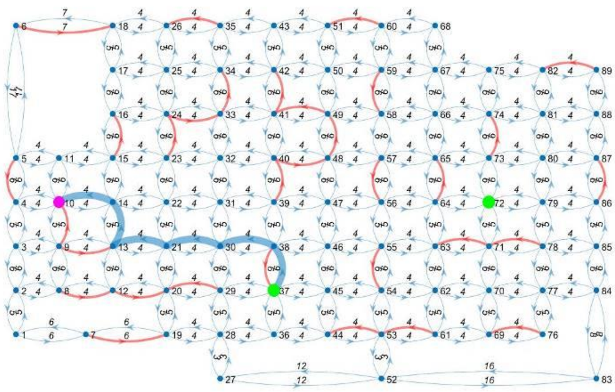

Figure7: Al-Sader city with 10% crowded streets.

random crowded streets, table (4) in the Appendix illustrates the optimal path of source nodes to random destination node such as node 57. Figure (10) represents 40% crowded streets. Thus, there are 124 random crowded streets, table (5) in the Appendix illustrates the optimal path of source nodes to random destination node such as node 57. The blue path represents the optimal source node and path. Figure (11) represents 50% crowded streets. Thus, there are 155 random crowded streets, table (6) in the Appendix illustrates the optimal path of source nodes (37, 72) to random destination node such as node 22. The blue path represents the optimal source node and path.

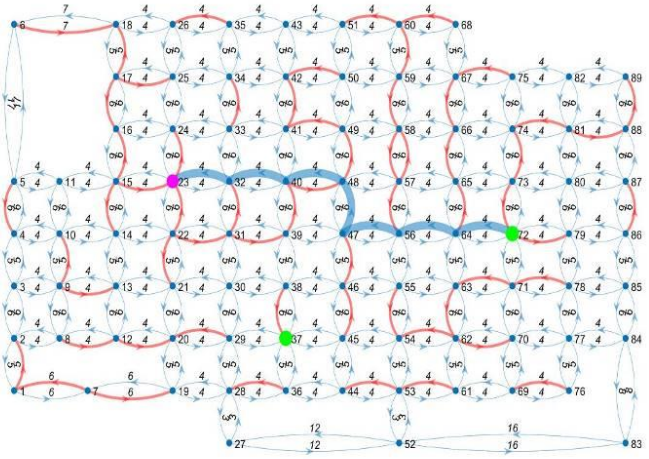

Figure 8: Al-Sader city with 20% crowded streets.

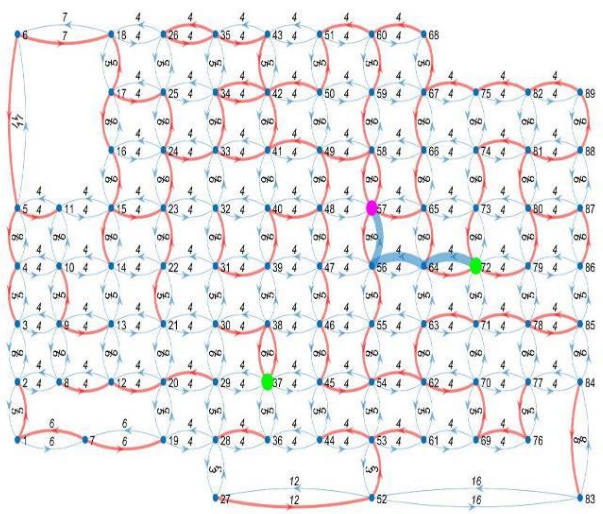

Figure 9: Al-Sader city with 30% crowded streets

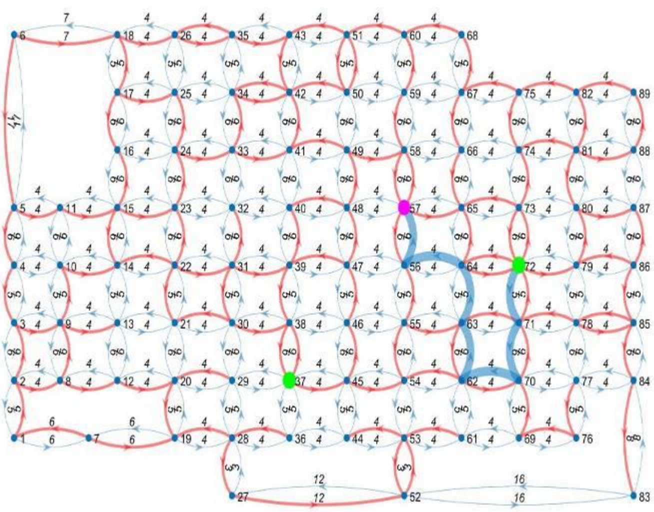

Figure 10: Al-Sader city with 40% crowded streets.

Figure (12) represents 60% crowded streets. Thus, there are 186 random crowded streets, table (7) in the Appendix illustrates the optimal path of source nodes to random destination node such as node 50. Figure (13) represents 70% crowded streets. Thus, there are 217 random crowded streets, table (8) in the Appendix illustrates the optimal path of source nodes to random destination node such as node 55. Figure (14) represents 80% crowded streets. Thus, there are 249 random crowded streets, table (9) in the Appendix illustrates the optimal path of source nodes to random destination node such as node 22. Figure (15) represents 90% crowded streets. Thus, there are 280 random crowded streets, table (10) in the Appendix illustrates the optimal path of source nodes to random destination node such as node 46. The blue path represents the optimal source node and path, depending on the distance of each path, the module determines the optimal path as well as a better emergency vehicle with its optimal path.

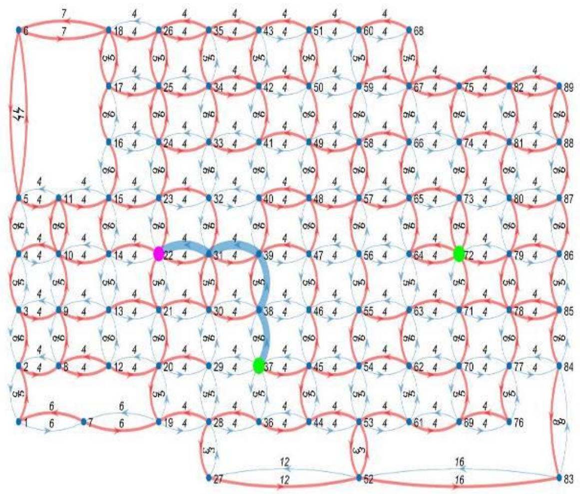

Figure 11: Al-Sader city with 50% crowded streets

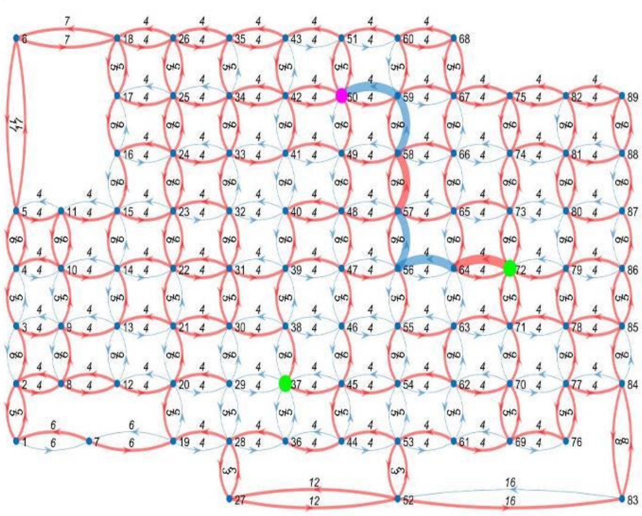

Figure 12: Al-Sader city with 60% crowded streets.

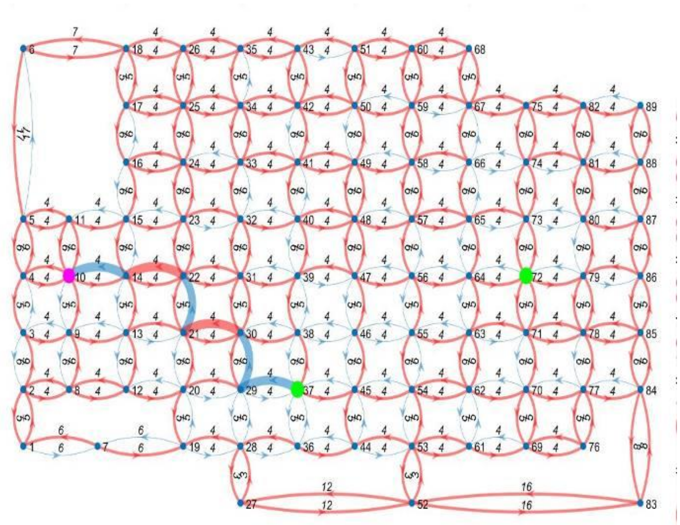

Figure 13: Al-Sader city with 70% crowded streets

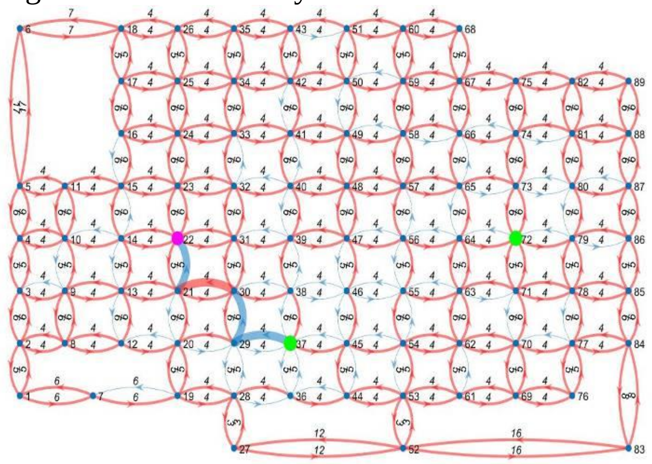

Figure 14: Al-Sader city with 80% crowded streets

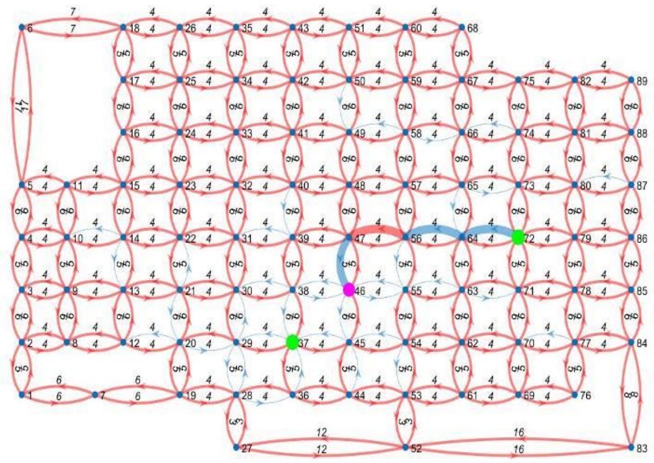

Figure 15: Al-Sader city with 90% crowded streets.

5. Conclusion

Modified simulated annealing has been able to find optimal paths for more than one emergency vehicle on the same map and determine the optimal vehicle which is must move to the destination node passing less crowded streets (as much as it can). The intersection with its streets is mapped in different number of traffic intersection with a random number of crowded streets in each map. Test maps uses to evaluate the recommendation module. The results show the spent time for a map with 12 traffic intersections is 5.11 seconds. While in 30 traffic intersections is 8.24 seconds. In addition the spent time of the case study of a real city in Baghdad, between 27 to 204 seconds. The spent time depends on the destination node place. If the destination node is near to the source vehicle, the time spend is smaller than the farthest from the source node. The large number of traffic intersection nodes takes more processing time than the small number of intersection nodes.

References

[1] Emad I. Abdul Kareem , Farah Younus Safar, Intelligent Recommendation Module for Emergency Vehicles,International Journal of Traffic and Transportation Engineering 2018, 7(3): 63-69.

[2] Xiaoyi Fan, Jiangchuan Liu, Zhi Wang, Yong Jiang, and Xue Liu, Crowdsourced Road Navigation:Concept, Design, and Implementation,IEEE Communications Magazine • June 2017.

[3] Muhammad Usman Shahid Khan, Osman Khalid, Ying HuRajiv Ranjan, Fan Zhang, Junwei Cao, haradwaj Veeravalli Samee U. Khan, Keqin Li, Fellow, IEEE, and Albert Y. Zomaya, MacroServ: A Route Recommendation Service for Large-Scale Evacuations, IEEE Transactions on Services Computing 2015.

[4] Long Liu, Jin Xu, Stephen Shaoyi Liao and Huaping Chen, A real-time personalized route recommendation system for self-drive tourists based on vehicle to vehicle communication, 2013 Elsevier.

[5] Yi Liu, Arash M. Roshandeh, Zongzhi , Konstantinos Kepaptsoglou, Harshingar Patel and Xi Lu, Heuristic Approach for Optimizing Emergency Medical Services in Road Safety within Large Urban Networks, 2014American Society of Civil Engineers

[6] Sumit Mallik, *Ahmedabad*, Intelligent Transportation System, International Journal of Civil Engineering Research.ISSN 2278-3652 Volume 5, Number 4 (2014), pp. 367-372.

[7] A. Kaveh, Advances in Metaheuristic Algorithms for Optimal Design of Structures , © Springer International Publishing AG 2014, 2017.

[8] Jason Brownlee,Clever Algorithms Nature-Inspired Programming Recipes, Copyright 2011 Jason Brownlee. All Reserved

Appendix

Table 2: Performance criteria values of Al-Sader city with 10% crowded streets

Destination node	Optimal path,v1	Distance v1	Optimal path v2	Distance V2	Total time/sec
83	37-36-44-53-52-83	32	72-79-80-85-84-83	27	48.25
52	37-45-44-53-52	16	72-71-70-62-54-53-52	27	34.72
70	37-45-54-62-70	16	72-71-70	11	29.61
10	37-38-30-31-22-14-10	27	72-64-56-47-39-31-22-14-10	32	107
50	37-38-46-47-56-57-58	41	72-73-65-66-67-59-50	30	60.66
6	37-29-30-31-33-24-25-17-18-6	53	72-73-65-57-58-59-50	30	196.14
40	37-38-46-47-48-40	25	72-64-56-57-48-40	22	44
25	37-38-39-31-32-33-24-25	37	72-73-65-66-67-59-50-42-34-25	42	89.36
66	37-38-39-47-48-57-65-66	35	72-73-65-66	16	46.45
62	37-45-54-62	12	72-71-63-62	15	27
60	37-38-46-47-56-57-58-59-60	42	72-64-56-57-58-59-60	31	78.20

Table 3: Performance criteria values of Al-Sader city with 20% crowded streets

Destination node	Optimal path,v1	Distance v1	Optimal path v2	Distance V2	Total time/sec
23	37-38-39-47-48-40-32	33	72-73-65-57-48-40-32-23	30	74.95
38	37-38	6	72-64-56-47-39-38	32.48	32.48
18	37-38-39-47-48-40-32-33-24-25-26-18	54	72-64-56-47-48-40-41-33-24-25-26-18	51	131.20
6	37-38-39-31-22-14-10-11-5-6	54	72-64-56-57-48-40-32-33-24-25-26-18-6	58	200
30	37-38-30	10	72-64-63-55-46-38-30	25	44.78
57	37-38-46-55-56-57	25	72-73-65-57	14	36.93
60	37-38-39-47-48-40-41-49-50-51-60	50	72-73-65-66-58-59-50-51-60	39	78.79
22	37-29-30-21-22	19	72-64-56-47-39-31-22	24	67
89	37-45-54-62-63-64-72-79-80-81-82-89	53	72-73-80-81-82-89	26	96.88
33	37-38-39-47-48-40-32-33	32	72-73-65-57-48-40-41-33	35	67.46

Table 4: Performance criteria values of Al-Sader city with 30% crowded streets

Destination node	Optimal path,v1	Distance v1	Optimal path v2	Distance V2	Total time/sec
57	37-36-44-53-61-62-63-64-56-57	33	72-64-56-57	14	36.86
61	37-45-44-53-61	17	72-64-63-62-61	20	36.23
38	37-45-44-53-54-55-56-38	32	72-64-63-55-46-38	21	32.77
80	37-45-44-53-54-62-63-71-72-73-80	37	72-73-80	10	44.23
1	37-29-28-19-20-12-8-2-1	35	72-64-56-47-39-38-30-21-20-12-8-2-1	52	125
20	37-29-28-19-20	18	72-64-56-47-39-38-30-21-20	35	56.22
17	37-36-44-53-54-55-63-64-56-57-48-40-41-33-24-16-17	141	72-64-56-57-48-40-32-33-24-16-17	46	134.57
66	37-36-44-53-54-62-63-71-72-73-65-66	53	72-73-65-66	16	46.83
89	37-45-44-53-54-55-63-71-72-73-80-81-82-89	63	72-73-80-81-82-89	26	100.31
40	37-36-44-53-54-55-63-64-56-57-48-40	51	72-73-65-57-48-40	22	45.70

Table 5: Performance criteria values of Al-Sader city with 40% crowded streets

Destination node	Optimal path,v1	Distance v1	Optimal path v2	Distance V2	Total time/sec
57	37-36-44-53-61-62-63-64-56-57	43	72-71-70-62-63-64-56-57	36	37.1
30	72-71-70-62-63-55-46-38-30	37	37-29-30	10	43.78
1	37-29-20-12-8-2	125	72-71-70-62-63-55-46-38-30-13-12-8-2-1	164	130
41	37-29-30-31-32-40-41	131	72-73-65-66-58-49-41	128	59.15
56	37-36-44-53-54-62-63-64-56	37	72-71-70-62-63-64-56	30	29.2
14	37-29-20-12-13-14	123	72-71-70-62-63-55-46-38-30-31-22-14	150	89.11
68	37-29-30-31-32-40-41-42-43-51-60-68	154	72-73-65-66-67-68	227	110
22	37-29-30-31-22	19	72-71-70-62-63-55-46-38-39-31-22	46	68.19
30	37-29-30	10	72-71-70-62-63-55-46-38-30	37	43.83
52	37-36-44-53-52	116	72-71-70-78-84-83-52	143	34.28
18	37-29-30-31-32-33-24-16-17-18	249	72-73-65-66-58-49-41-33-24-16-17-18	251	133.15

Table 6: Performance criteria values of Al-Sader city with 50% crowded streets**.**

Destination node	Optimal path,v1	Distance v1	Optimal path v2	Distance V2	Total time/sec
22	37-38-39-31-32	19	72-71-70-62-63-55-46-38-39-31-22	46	65
56	37-38-39-47-56	19	72-71-70-62-63-64-56	30	29.17
74	37-38-39-47-56-57-65-66-74	139	72-71-70-62-63-64-65-66-74	142	48.56
47	37-38-39-47	15	72-71-70-62-63-64-56-47	34	32
33	37-38-39-31-32-33	127	72-71-70-62-63-55-46-38-39-40-41-33	154	68.25
66	37-38-39-47-56-57-58-66	135	72-71-70-62-63-64-65-66	138	47.08
64	37-38-39-47-56-64	23	72-71-70-62-63-64	26	28.94
59	37-38-39-47-56-57-58-59	137	72-71-70-62-63-64-56-57-58-59	148	60.86
81	37-38-39-47-56-57-55-73-80-81	143	72-73-80-81	116	54.05
41	37-38-39-40-41	123	72-71-70-62-63-55-46-38-39-40-41	150	56.13

Table 7: Performance criteria values of Al-Sader city with 60% crowded streets**.**

Destination node	Optimal path,v1	Distance v1	Optimal path v2	Distance V2	Total time/sec
30	37-29-30	10	72-64-56-47-46-38-30	125	42.74
5	37-29-20-12-13-9-3-4-5	237	72-64-56-57-48-40-32-23-15-11-5	342	145
28	37-29-28	9	72-64-56-55-46-38-30-29-28	136	32.14
50	37-36-44-53-61-62-63-64-56-57-58-59-50	259	72-64-56-57-58-59-50	230	60.17
57	37-36-44-53-61-62-63-64-56-57	143	72-64-56-57	114	36.59
70	37-36-44-53-61-62-63-71-70	138	72-71-70	111	30.02
61	37-36-44-53-61	117	72-64-63-62-61	120	35.19
62	37-36-44-53-61-62	122	72-71-70-62	115	27.22
8	37-29-20-12-43-9-8	128	72-64-63-55-46-38-30-29-20-12-13-9-8	255	112
80	37-36-44-53-61-62-63-64-65-73-80	247	72-73-80	110	42.87

Table 8: Performance criteria values of Al-Sader city with 70% crowded streets**.**

Destination node	Optimal path,v1	Distance v1	Optimal path v2	Distance V2	Total time/sec
36	37-36	5	72-64-56-47-46-45-37-36	32	36.25
50	37-38-39-40-41-42-50	433	72-64-65-66-67-59-50	230	60.54
57	37-38-46-55-56-57	225	72-64-56-57	14	36.67
48	37-38-39-47-48	221	72-64-56-57-48	18	39.37
38	37-38-	106	72-64-56-47-46-38	21	32.38
85	37-38-46-55-63-71-78-85	430	72-79-86-85	213	39.35
18	37-38-39-31-32-33-24-16-17-18	446	72-64-63-55-46-38-39-31-32-33-24-16-17-18	361	128.46
6	37-29-30-21-13-9-3-4-5-6	354	72-64-56-47-46-38-39-40-32-23-15-11-5-6	369	199.54
8	37-29-20-12-8	216	72-64-63-55-46-38-30-21-20-12-8	243	114.54
30	37-29-30	10	72-64-56-47-46-38-30	25	42.40

Table 9: Performance criteria values of Al-Sader city with 80% crowded streets**.**

Destination node	Optimal path,v1	Distance v1	Optimal path v2	Distance V2	Total time/sec
22	37-29-30-21-22	119	72-64-56-47-46-38-39-31-22	134	142
1	37-29-28-19-7-1	225	72-64-56-47-46-38-37-29-28-19-7-1	252	131.68
36	37-36	5	72-64-56-55-46-45-37-36	32	38.84
55	37-38-46-55	114	72-64-56-55	13	26.99
17	37-29-30-21-22-14-15-16-17	441	72-64-56-47-46-38-39-40-32-33-24-16-17		
77	37-38-46-55-63-71-70-77	332	72-79-78-77	215	33.02
82	37-38-46-55-56-57-65-66-67-75-82	549	72-73-74-81-82	322	76.29
10	37-29-30-21-22-14-10	227	72-64-56-55-46-38-39-31-22-14-10	242	115.63
45	37-45	104	72-84-56-47-46-45	23	28.13
70	37-38-46-55-63-71-70	228	72-71-70	111	31.85

Table 10: Performance criteria values of Al-Sader city with 90% crowded streets**.**

Destination node	Optimal path,v1	Distance v1	Optimal path v2	Distance V2	Total time/sec
30	37-29-30	10	72-64-56-47-46-38-30	25	44.60
80	37-38-46-55-63-71-72-73-80	537	72-79-80	210	43.45
75	37-38-46-55-56-57-58-66-67-75	545	72-73-74-75	318	70.41
55	37-38-46-55	114	72-64-56-55	13	26.31
62	37-45-54-62	312	72-64-63-62	215	27.84
67	37-38-46-55-56-57-58-66-67	541	72-64-65-66-67	322	60
89	37-38-46-55-50-57-58-66-67-75-82-89	753	72-73-74-81-88-89	426	102
44	37-29-28-36-44	117	72-64-56-55-46-45-44	128	36.56
25	37-29-30-21-22-23-24-25	437	72-64-56-55-46-38-30-21-22-23-24-25	452	102.09
46	37-38-46	110	72-64-56-47-46	17	28.61