

An Optimized Ambulance Dispatching Solution for Rescuing Injures After Disaster

Hazan Dağlayan, Murat Karakaya*

Department of Computer Engineering, Atilim University, İncek, 06836, Ankara, Turkey

Copyright ©2016 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Abstract In this work, we propose a Genetic Algorithm (GA) for effectively scheduling ambulances aftermath of a disaster. Given the limited number and capacity of ambulances, we aim to minimize the number of ambulance tours and the average time to take all injured people to a hospital. Both of these goals require that the total tour length of all routes should be minimized as well. This problem can be considered as an extension of the well-known Capacitated Vehicle Routing Problem (CVRP). We developed a Genetic Algorithm (GA) and tested using some of the CVRP benchmark files. For the possible number of injures at each location, we define three different scenarios. The proposed GA aims to minimize the tour lengths of the ambulances while respecting all real life constraints given in these scenarios. In order to evaluate the proposed GA, we also developed a rival method based on the Nearest Neighbor (NN) heuristic. The results of extensive simulation tests NN heuristic.

Keywords Genetic Algorithm, Vehicle Routing Problem, Capacitated Vehicle Routing Problem, Ambulance Routing, Nearest Neighbor Heuristic

1 Introduction

The disaster can happen naturally or due to human effects. These disasters can damage both the existing infrastructures and human beings at a larger scale. In those situations, effective logistic operations are hard to be carried out, and, thus, efficient medical aid may not be provided on time, which, in turn, causes more losses. Therefore, after a disaster, rescue and aid operations must be well planned and must be conducted efficiently [1].

In an emergency situation, firstly injured people are to be transferred to hospitals. To perform this task efficiently, we have to know the location and number of the injured people, the number and the capacity of the ambulances since all these values are the input to the problem solution. After collecting these inputs, we have to develop a ambulance scheduling so that injures can be transferred to hospitals in a short time.

The above mentioned problem is related with the Capacitated Vehicle Routing Problem (CVRP) [2] which is a type

of Vehicle Routing Problem (VRP). In VRP, there are costumers and their demands. The aim is to minimize the total distance for transmitting goods according to the customer demands to all customer location from the depot location by a vehicle. The vehicle is loaded with goods from the depot, and the goods are distributed to customers. If the carried goods are run out off, the vehicle returns to the depot. If the vehicle's carrying capacity is limited with a given quantity, the VRP is called Capacitated Vehicle Routing Problem (CVRP). There are various types of solutions are proposed for CVRP: exact solutions, heuristic, or meta-heuristic methods. Some researchers propose methods to find the exact solution. However, these methods may be infeasible for large number of customers because the CVRP is defined as a non-deterministic polynomial problem (NP-hard). Therefore, meta-heuristic and heuristic methods are usually implemented to solve CVRP with larger number of customers efficiently such as Simulated Annealing [3], Particle Swarm Optimization [4], Ant Colony [5], Genetic Algorithm [6] etc. For these types of optimization problems, the solution qualities of different meta-heuristics are not found to be drastically different [7]. In this paper, we opt to use the GA for its matured methodology and proven robustness [8].

To compare the proposed GA, we also implement a heuristic method called Nearest Neighbor (NN). We tested both solutions using many CVRP benchmarks under different scenarios. In each scenario, the number of injured people at each location is determined randomly using some threshold values. We simulated all the scenarios and solutions via MATLAB tool. The results indicates the successo the proposed GA solution over the NN heuristic.

The remainder of this paper is organized as follows. The related work of the ambulance routing problem is given in Section 2. The GA and its operators are described in details in Section 3. In Section 4, the adaptation of NN algorithm to our problem is presented while the implementation of the proposed GA is provided in Section 5. The simulation test results of the solutions are given in Section 6, and finally, Section 7 covers conclusions and implications.

2 Related Work

Various approaches are proposed to solve the problem of efficiently delivering injured people to hospital. Gong et. al.

deal with picking up injured peoples information from different locations and delivering them to the hospital after an earthquake disaster [9]. Gong et. al. proposed that data fusion center helps to improve dispatching efficiency by decreasing travel time to hospital. In their work, the considered key factors are patient priority, cluster information, and distance. Thus, the authors assume that ambulance capacity is only one. However, in this work, we consider the fact that there can be some ambulances with larger capacities. In another work, Nordin et. al. attempt to solve the problem by finding the shortest path of the ambulance with the A* algorithm [10]. In the experiments, they used the data collected at a clinic. However, the calculated path is from a specified ambulance station to a specified emergency site. Thus, the researchers do not consider multiple emergency sites when creating paths. In our study, we consider a specified hospital but multiple locations of injured people.

While some studies classify the injured people according to their emergency as being serious or moderate when they search for an optimal path, some other studies consider risk properties involved in the given problem [1, 11]. For example, Kula [11] uses the medium-integer linear programming to find the path after classifying locations according to their risk evaluations aftermath of an earthquake. The problem resembles Dynamic Vehicle Routing Problem. Talarico [1] proposes a large neighborhood search meta-heuristic to the dynamic and uncertainty conditions for ambulance routing after a disaster. The method categorizes peoples situations about being serious or slightly serious.

In a previous work, we propose a simplified solution to a version of this problem [12]. In this work, we extended the solution by improving crossover operators and conducted extensive simulation test using CVRP benchmarks. Moreover, we analyzed the effect of the GA parameter values such as generation number, population size, and location numbers.

3 Genetic Algorithm

Genetic Algorithm (GA) is a meta-heuristic depends on the evolutionary approach. Each possible solution is encoded as a chromosome which consists of genes. GA begins with solving a problem with an initial population of chromosomes. Each chromosome has a fitness value computed according to the quality of the encoded solution in itself. By applying crossover and mutation operators on the existing population, GA creates a new generation. A selection method picks up two chromosomes as parents from the population. Crossover operator exchanges genes of these parents to create two offsprings. Mutation operator modifies the constructed chromosome structure. Until the given termination condition is met, GA continues to create new generations and improves the quality of found solution. When the termination condition is satisfied, GA outputs the best chromosome with the highest fitness value as the optimum solution. Below, we provide the details of the GA operators and procedures.

3.1 Fitness Value

Fitness value is calculated after each new individual is created. It evaluates the quality of the generated solution according to the given optimization goal. According to the elitism principle, the individual with the maximum fitness value in the current population is maintained for the next generation.

3.2 Selection Operator

The selection operator is used to choose two chromosomes as the parents from the current population to create new individual for the next generation by applying crossover or mutation operators.

Some well-known selection operators are tournament selections, roulette wheel selection, and steady-state approach. The tournament selection helps to find the best individual of tournament size number individual [13, 14, 15]. For instance, if the tournament size is 2, which is called binary tournament size, two individuals are selected randomly, and better one of the two will be the winner.

The Roulette Wheel Selection is proposed to select the chromosomes considering their fitness value: Greater the fitness value is, greater the probability of being selected as a parent is [16, 14].

The steady-state approach ensures that two parents and two offsprings are compared; the best two chromosomes of them are selected [17].

3.3 Crossover

Crossover is the main process of the GA to vary the chromosomes from one generation to the next. Two selected parents produce two offsprings according to the given crossover operator. The main idea is to combine parents to create offsprings whose fitness values are higher. Some most popular crossover operators are Edge Recombination Crossover (ERX) [17, 18, 19], one point or two points crossover [20, 21] and order crossover [22].

3.4 Mutation

Mutation operator is used to getting rid of the local optimum points. When mutation is not used, the local optimum solution can be reported as global optimum solution. Swap [18, 21, 23], insertion [23] and inversion [18, 23] mutations are well-known mutation types.

4 Adaptation of NN Heuristic

Nearest Neighbor (NN) is a heuristic method which is proposed in Travelling Salesman Problem (TSP) and Vehicle Routing Problem (VRP) [24]. In NN method, the nearest unvisited node to the current one is selected as the next node until all nodes are visited.

To observe the success of the proposed GA solution, we adopted the NN heuristic as a solution to the problem as follows. A route starts with the depot (in our case, a hospital), and then the nearest node (in this case the location of injures) to the depot is selected. If this node has no injured people, the node is not visited. Otherwise, until all the injured people are evacuated, the node is re-visited. Later, if the vehicle has adequate remaining capacity, the next nearest node is selected and this process continues. Otherwise, the vehicle (ambulance) returns to depot to deliver all picked-up injures. The process continues until all injured people are transferred to the hospital.

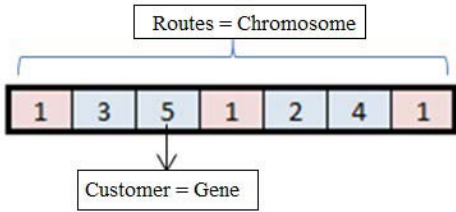


Figure 1. Representation of the relation between CVRP and GA

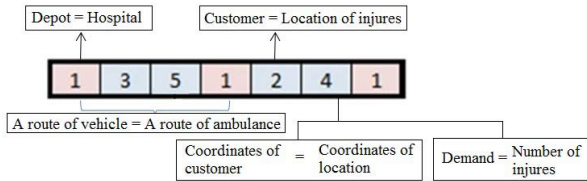


Figure 2. Representation of the relation between CVRP and the problem

5 Implementation of GA

5.1 Representation of the Problem

The stated problem has similar properties to the CVRP problems. Each of the customer location is represented as an injured people location while customers demands are represented as the number of injured people in each location. Moreover, the capacity of the vehicle in CVRP corresponds to the number of the injured people that an ambulance can take, and the depot is represented as the hospital.

The CVRP can be encoded into GA as follows. The customer locations in the CVRP are represented as genes in chromosomes. That is, in our problem, the locations of injured people are represented in genes by the location number. Moreover, the routes which consist of these locations are chromosomes in GA and all possible routes construct the population. Besides, the total distance among all locations in a chromosome is defined as the fitness value. By applying the proposed crossover and mutation operators to the chromosomes at each generation, the minimal distance (maximum fitness value) is found.

5.1.1 Encoding Schema

In the CVRP, the location of customers and depot is represented with a number. These numbers are represented as genes in a chromosome while the proposed solution which consists of these numbers is represented as the chromosome in GA illustrated in the Figure 1. Moreover, the depot in CVRP is encoded as hospital and the customers are represented as the location of injures. Each customer has coordinates (x, y) and demands in CVRP whereas the locations have the information of coordinates and the number of injures in our problem shown in Figure 2

5.2 Genetic Algorithm Operations

Below, we explain how we modified the GA to adopt our problem. In each cycle, a new generation is constructed and off-springs are created applying crossover and mutation operators. Termination condition is a fixed number for the gen-

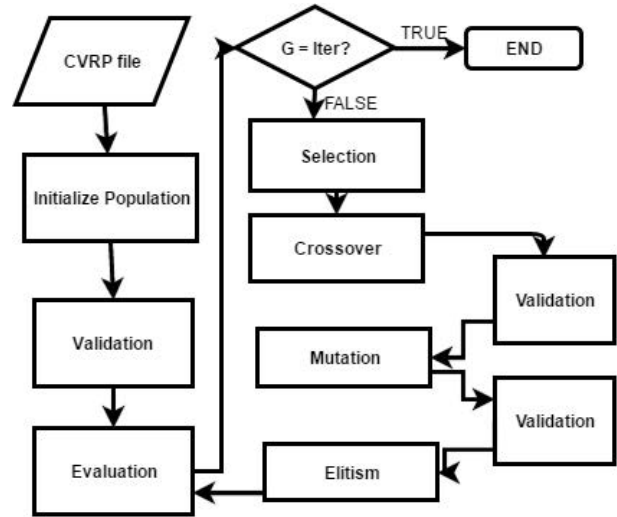


Figure 3. Flowchart of the proposed GA solution

erated generations. The flowchart of the algorithm is given in Figure 3.

5.2.1 Validation

Each produced chromosome is tested against validation rule. Basically, a chromosome is valid if the total number of injures picked-up from the locations as in the order of the genes does not violate the capacity constraint of ambulance. That is, when the ambulance is fully loaded with the injures, it must return to the hospital. In addition, the ambulance does not return to the hospital when it is not fully occupied unless all injures already transferred. When a chromosome is not validated, it is trimmed such that the ambulance returns to hospital. In other words, an invalid chromosome can be fixed by inserting the hospital location into proper order of the genes. Thus, modified chromosome becomes a valid one. To solve the validation problem, all 1's (depots) are removed from the chromosome and then according to demands, 1's are inserted when the capacity is fulfilled.

5.2.2 Initialization

The initial population is created randomly. After randomization, the population must be validated according to process described above.

5.2.3 Fitness Value

In our problem, fitness value of a chromosome is the total route distance. The minimum value of the distance is maintained for the next generation. The fitness function F can be formulated as

$$F = \sum_{i=1}^n \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (1)$$

where (x_i, y_i) is the coordinate of the gene i and n is the number of total genes in a chromosome.

5.2.4 Selection

Selection operator is used to determining parents on which crossover or mutation operator is applied to create off-

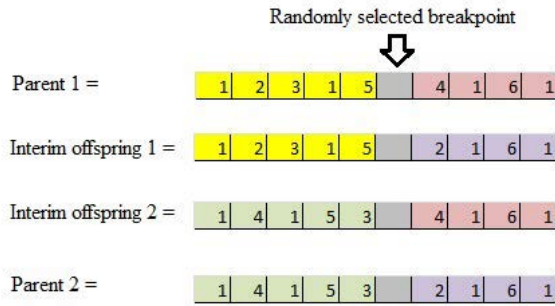


Figure 4. First step of a sample of One Point Crossover

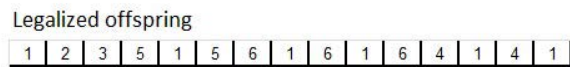


Figure 5. Legalized offspring of a sample of One Point Crossover

springs. In the proposed algorithm, the tournament selection is applied as described in Section 3.3.

5.2.5 Crossover

In proposed algorithm, we use two different crossover methods to observe the possible impact of the selected crossover method. In the first crossover method, a cut point is selected randomly. The first offspring consists of the first part of the first parent and the second part of the second parent. The second off-spring is constructed with remaining parts of the parents. Then, the constructed off-springs are validated according to above validation. The following example demonstrates the process. Assume that selected two parents are as follows: $P1 = (123154161)$ and $P2 = (141532161)$. Moreover, the number of injures at locations are (012637) . That is, the first location which is the hospital does not have any injury, but the location 2 has 1 injury and location 2 has 2 injures. furthermore, assume that the capacity of the ambulance is 4 injures.

After the interim off-spring is constructed as above, until all demands supply the vehicle visits the depot. In the example, in the starting location 1 (hospital), the initial capacity is 4 and after location 2 is visited, remaining capacity is 3. Then, location 3 where 2 injures exit is visited. With the remaining capacity 1, location 5 is visited and only 1 injured is picked up. Since capacity is full, the ambulance must return to the hospital (location 1) and the location 5 must be re-visited for collecting up the remaining injured people. All locations of the constructed off-spring are visited and the injures located there are picked up. The remaining locations of the first part of the second parent are used as explained above.

In the second crossover method, Periere’s crossover is modified to our problem [26]. First, a sub-route is selected randomly from first parent and the next point of the first point of the sub-route is found from second parent. Then, the sub-route is inserted to the next point and all duplicated points are removed. Finally, validation process is run over the off-springs.

The following example shows how the second crossover method operates. Assume that $(5,6)$ sub-route is selected and

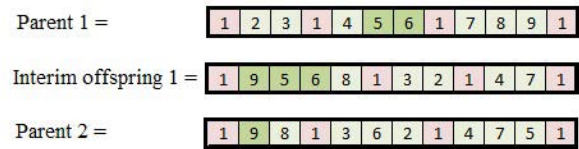


Figure 6. Legalized offspring of a sample of Pereira’s Crossover

9 is the next location to the location 5. The sub-route is inserted to the point 9.

In the crossover methods, two offspring are constructed from two parents. The process explained above is also applied to the second off-spring as a reverse of the first off-spring.

5.2.6 Mutation

In the proposed GA, we use swap, insertion and inversion mutations. In swap mutation, two points are selected randomly and they swap. In insertion mutation, a gene is selected randomly. Then, it is removed from the position and reinserted to a new position while in inversion mutation, the points between two positions, which are selected randomly are reversed. Then, the mutated individuals are validated as described above.

5.2.7 Elitism

The chromosomes with higher fitness value may be kept intact by not applying the crossover and mutation operators. Therefore, these chromosomes are carried to next generations without any changes. We apply elitism by selecting the half of best individuals of the new and old generations as the member of the new generation.

6 Simulation Results

We have implemented the proposed GA and NN solutions on MATLAB and their experimental results are compared using well-known CVRP benchmarks. Moreover, we define three scenarios considering number of injured people at locations as given in Table 1. In the first scenario, we assume that disaster has a light damage and each location has less number of injures than the given capacity of the ambulance. In the second one, we assume that damage is heavy and the number of injured people at each location is more than the given capacity. In the last scenario, we consider a medium damage in which the locations have medium number of injured people such that the number of injures at some location is less than capacity of ambulance whereas the number of injures at the rest is more than capacity of ambulance. For each scenario, the numbers of injured people are assigned randomly and the ambulance capacity is given 4 injures. The parameters and their default values which are used in the simulation are given in Table 2. Each scenario are executed ten times and their average values are presented.

In order to compare solutions we define a metric called Improvement. The improvement defines the decrease in the total length of the solution generated by the GA over the solution created by the NN. The improvement metric is calculated according to the following equation.

Table 1. Number of injures at each scenario

Scenario	# injures at each location
Light Damage	1.. Ambulance capacity
Medium Damage	1..8
Heavy Damage	(Ambulance capacity + 1)..8

Table 2. GA parameters and their values

Parameter	Default Value	Range
Population Size	200	-
Crossover Probability	1	-
Mutation Probability	0.5	-
Generation number	2000	500-2000
Crossover Operator	Periere	Periere, One Point

$$Improvement = \frac{|NN| - |GA|}{\frac{|NN| + |GA|}{2}} * 100 \quad (2)$$

where $|NN|$ is the total tour length of the solution by Nearest Neighbour method and $|GA|$ is the total tour length of the solution value of Genetic Algorithm.

The algorithm is implemented on an Intel Core i5 3.2 Ghz, 4 RAM machine with MATLAB version R2016A. The duration of the solution changes according the nodes and scenario of the problem. For instance, when the ulysses-n22-k4 is selected as the benchmark and the considered scenario is light, the GA solution takes around 2 minutes. On the other hand, when the number of node is more and the scenario is heavy, the solution takes longer time. For example, when P-n101-k4 is selected as the benchmark, the solution generation takes about 20 minutes on the average.

Below, we present the results of the simulation tests under two main sets. In the first one, we compare the performance of the proposed GA and the adapted NN heuristic. Next, we investigate the proposed GA closer by observing the effect of setting different values as generation number and implementing different crossover operators.

6.1 Results of GA and NN solutions

We conducted extensive simulation tests to compare the performance of both solutions. The obtained results of these tests are presented in Tables 3 and 4, and at Figure 7.

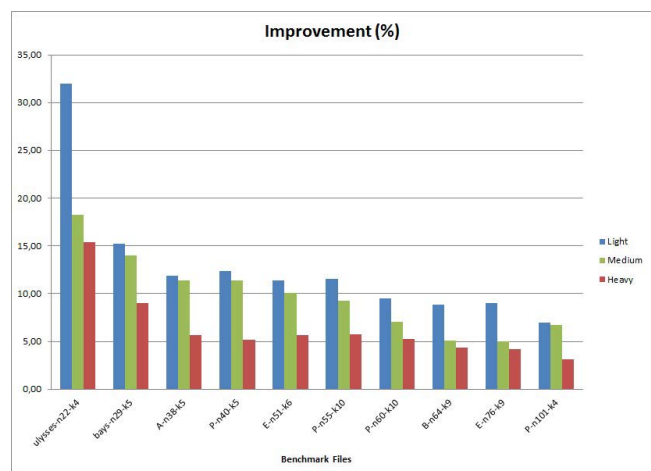
The best values obtained at the experiments for the proposed GA and NN solutions are given in Table 3. For each scenario, all 10 benchmarks are tested and their results show that GA has better values on all dataset and all scenarios. Moreover, Figure 7 shows the success of GA is reported more clearly.

When we consider the case in which the probable number of injures at each location is less than the capacity of ambulances (light scenario), we observe in Figure 7 that the improvement of the proposed algorithm is better than those of the other scenarios (i.e. heavy and medium).

When all the benchmark files in light scenarios are considered, the average improvement is about %12.89. On the other hand, the average improvements of the heavy and medium scenarios are %6.36 and %10.36, respectively. Similarly, in the medium hand scenarios the obtained performance results are better compared to the results observed in heavy scenarios.

Table 3. Comparison of GA and NN solutions

Benchmark	Scenario	Total Tour Length		Improvement(%)
		NN	GA	
ulysses-n22-k4	light	250	181	32.01
	heavy	518	444	15.38
	medium	371	309	18.24
Average				21.88
bays-n29-k5	light	35934	30840	15.26
	heavy	74051	67666	9.01
	medium	41650	36187	14.04
Average				12.77
A-n38-k5	light	2220	1971	11.88
	heavy	5078	4798	5.67
	medium	3218	2871	11.40
Average				9.65
A-n40-k5	light	1529	1351	12.36
	heavy	3105	2949	5.15
	medium	2221	1982	11.37
Average				9.63
E-n51-k6	light	1943	1733	11.43
	heavy	4066	3842	5.67
	medium	3422	3094	10.07
Average				9.05
P-n55-k10	light	1999	1781	11.53
	heavy	4131	3900	5.75
	medium	3130	2853	9.26
Average				8.85
P-n60-k10	light	2109	1918	9.49
	heavy	4615	4379	5.25
	medium	3452	3216	7.08
Average				7.27
B-n64-k9	light	3419	3128	8.89
	heavy	7149	6846	4.33
	medium	5301	5036	5.13
Average				6.12
E-n76-k9	light	2717	2483	9.00
	heavy	6020	5773	41.89
	medium	4366	4151	5.05
Average				6.08
P-n101-k4	light	3619	3374	7.01
	heavy	8120	7868	3.15
	medium	5526	5167	6.71
Average				5.62

**Figure 7.** Improvement of the proposed GA for each injured distribution scenario and benchmark file

In Figure 7, the effect of location numbers in the benchmark files on improvements is also observed. The proposed GA sustains its success over the NN heuristic for all considered benchmark files. However, as the number of locations is increased the improvement of the GA algorithm decreases as shown in Figure 8, considering all scenarios together.

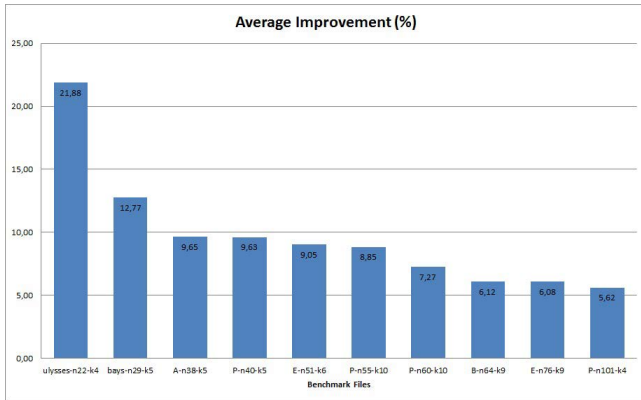


Figure 8. Average improvement(%) for each benchmark

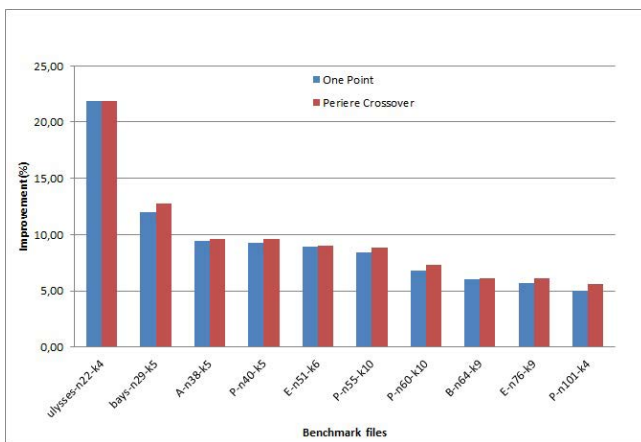


Figure 9. Average improvement (%) achieved by two different crossover operators

Table 4. Total Tour Lengths for various Generation Number

Benchmark file	Scenario	Total Tour Length			
		Iter.500	Iter.1000	Iter.1500	Iter.2000
ulysses-n22-k4	light	181	181	181	181
	heavy	444	444	444	444
	medium	309	309	309	309
bays-n29-k5	light	30840	30840	30840	30840
	heavy	67771	67771	67666	67666
	medium	36187	36187	36187	36187
A-n38-k5	light	1971	1971	1971	1971
	heavy	4798	4798	4798	4798
	medium	2761	2761	2761	2761
A-n40-k5	light	1351	1351	1351	1351
	heavy	2949	2949	2949	2949
	medium	2127	2127	2127	2127
E-n51-k6	light	1733	1733	1733	1733
	heavy	3891	3852	3842	3842
	medium	3108	3105	3094	3094
P-n55-k10	light	1798	1781	1781	1781
	heavy	3925	3900	3900	3900
	medium	2857	2853	2853	2853
P-n60-k10	light	1918	1918	1918	1918
	heavy	4396	4384	4384	4379
	medium	3264	3227	3216	3216
B-n64-k9	light	3133	3128	3128	3128
	heavy	6878	6857	6854	6846
	medium	5038	5036	5036	5036
E-n76-k9	light	2514	2485	2483	2483
	heavy	5811	5783	5775	5773
	medium	4153	4152	4151	4151
P-n101-k4	light	3410	3380	3375	3374
	heavy	7998	7944	7883	7868
	medium	5254	5193	5173	5167

Table 5. Comparison of Modified One Point Crossover and Modified Periere’s Crossover

Benchmark file	Scenario	Total Travelled Distance			
		One Point	Impro. (%)	Periere	Impro. (%)
ulysses-n22-k4	light	181	32.02	181	32.02
	heavy	444	15.38	444	15.38
	medium	309	18.24	309	18.24
bays-n29-k5	light	30904	15.05	30840	15.26
	heavy	67722	8.93	67666	9.01
	medium	36900	12.09	36187	14.04
A-n38-k5	light	1971	11.88	1971	11.88
	heavy	4798	5.67	4798	5.67
	medium	2890	10.74	2871	11.40
A-n40-k5	light	1351	12.36	1351	12.36
	heavy	2949	5.15	2949	5.15
	medium	2002	10.37	1982	11.37
E-n51-k6	light	1733	11.43	1733	11.43
	heavy	3853	5.38	3842	5.67
	medium	3098	9.94	3094	10.07
P-n55-k10	light	1781	11.53	1781	11.53
	heavy	3915	5.37	3900	5.75
	medium	2882	8.25	2853	9.26
P-n60-k10	light	1920	9.38	1918	9.49
	heavy	4414	4.45	4379	5.25
	medium	3233	6.55	3216	7.08
B-n64-k9	light	3129	8.86	3128	8.89
	heavy	6849	4.29	6846	4.33
	medium	5040	5.05	5036	5.13
E-n76-k9	light	2500	8.32	2483	9.00
	heavy	5791	3.88	5773	4.19
	medium	4155	4.95	4151	5.05
P-n101-k4	light	3382	6.77	3374	7.01
	heavy	7899	2.76	7868	3.15
	medium	5227	5.56	5167	6.71

6.2 Investigating the proposed GA performance

We investigated the impact of selecting different generation numbers on the solution quality. The results presented at Table 4 indicate that increasing number of generations can cause better optimizations for some benchmarks. Especially, the benchmarks with larger number of locations are more sensitive to increase in generation number. For example, for the benchmark P-n101-k4 the total tour length decreases from 7998 to 7868 for the heavy scenario when the generation number changes from 500 to 2000. However, for the benchmarks with relatively less number of locations, we do not observe the considerable change in the obtained results as the number of generation increases. We can explain this observation as follows. The proposed GA can find the optimum solution for small size problems faster (less generation number), but for the larger size problems the GA needs more generations to produce better results.

We also compare the success of two type crossovers: One point crossover and Periere’s crossover. The results of the simulation tests are depicted in Figure 9 and at Table 5. In general, we can argue that Periere’s crossover generates better results compared to One Point crossover. However, as seen in Figure 9, the improvement is slight. This means that crossover type does not affect the quality of the produced solution largely.

7 Conclusions

In this paper, we developed a GA based solution to the efficient routing of ambulances aftermath a disaster. Via extensive simulation experiments, we observed that the proposed

GA generates better solutions than these of the NN method. In all considered scenarios, the GA method achieved shorter total tour distances compared to the NN method. Therefore, we can argue that the GA method is robust against the changes of the underlying topologies, the number of locations, and the number of injured people at each locations. Thus, the GA consistently produces better results even the important problem parameters vary.

We would like to extend this study for multiple ambulances and multiple hospital locations as a future study. Thus, we can apply the proposed solution to the real life problems successfully. Moreover, we would like to implement other meta-heuristics (PSO or ACO) to observe the performance of different meta-heuristics.

REFERENCES

- [1] L. Talarico, F. Meisel, and K. Srensen. Ambulance routing for disaster response with patient groups, *Computers and Operations Research* 56, 120-133,2015.
- [2] G. B. Dantzig, J. H. Ramser. The truck dispatching problem, *Management science* 6.1, 80-91, 1959.
- [3] A. S. Alfa, S. S. Heragu, M. Chen. A 3-opt based simulated annealing algorithm for vehicle routing problems. *Computers & Industrial Engineering*, 21, 635–639. 1991. Elsevier
- [4] J. Ai, K. Voratas. Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. *Computers & Industrial Engineering* 56.1 ,380-387. 2009.
- [5] S. Mazzeo, L. Irene. An ant colony algorithm for the capacitated vehicle routing. *Electronic Notes in Discrete Mathematics* 18, 181-186. 2004.
- [6] J. Berger, M. Barkaoui. A hybrid genetic algorithm for the capacitated vehicle routing problem. *Genetic and evolutionary computation conference*. Springer Berlin Heidelberg, 2003.
- [7] R. K. Ahuja, J. B. Orlin, A. Tiwari. A greedy genetic algorithm for the quadratic assignment problem. *Computers & Operations Research* 27.10 , 917-934. 2000.
- [8] H. Savuran, M. Karakaya. Efficient route planning for an unmanned air vehicle deployed on a moving carrier. *Soft Computing*, 1-16. 2015.
- [9] Q. Gong, A. Jotshi, R. Batta R. Dispatching/Routing of Emergency Vehicles in a Disaster Environment using Data Fusion Concepts. *The 7th International Conference on Information Fusion*, 2004.
- [10] N. A. M. Nordin, Z. A. Zaharudin, M. A. Maasar, N. A. Nordin. Finding Shortest Path of the Ambulance Routing: Interface of A* Algorithm using C# Programming. *Humanities, Science and Engineering Research*, 1, 1569 1573, 2012.
- [11] U. Kula, O. Tozanli, S. Tarakcio. Emergency Vehicle Routing in Disaster Response Operations. *POMS Conference Proceedings*, 2012.
- [12] H. Daglayan, N. Kilinc, S. Aktas, I. Cerci, M. Karakaya. Efficient Routing of Ambulances Aftermath of a Disaster. *The 1st International Conference on Engineering and Natural Sciences (ICENS 2015)*, Skopje, Macedonia. 145. 2015.
- [13] H. Nazif, L. S. Lee. Optimized crossover genetic algorithm for capacitated vehicle routing problem. *Applied Mathematical Modelling*, 36, 2110-2117, 2012.
- [14] M. A. Mohammed, M. S. Ahmad, S. A. Mostafa. Using genetic algorithm in implementing capacitated vehicle routing problem. *Computer and Information Science*, 1, 257-262, 2012.
- [15] C. Bermudez, P. Graglia, N. Stark, C. Salto, H. Alfonso. A Comparison of Recombination Operators for Capacitated Vehicle Routing Problem. *Inteligencia Artificial*, 46, 34-44, 2010.
- [16] A. S. Tasan, M. Gen. A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and delivers. *Computers and Industrial Engineering*, 62, 755761, 2012.
- [17] M. Filipec, D. Skrlec, S. Krajcar. An efficient implementation of genetic algorithms for constrained vehicle routing problem. *Systems, Man, and Cybernetics*, 3, 2231 2236, 1998.
- [18] S. T. Bae, H. S. Hwang, G. S. Cho, M. J. Goan. Integrated GA-VRP solver for multi-depot system. *Computers & Industrial Engineering*, 53, 233240, 2007.
- [19] E. Alba, B. Dorronsoro. Solving the Vehicle Routing Problem by Using Cellular Genetic Algorithms. *Computer Science*, 3004, 11 20, 2004.
- [20] N. LI , S. Wang, Y. Li. A Hybrid Approach of GA and ACO for VRP. *Computational Information Systems*, 7, 4939-4946, 2011.
- [21] B. M. Baker, M.A. Ayechev. A Genetic Algorithm For The Vehicle Routing Problem. *Computers & Operations Research*, 30, 787-800, 2003.
- [22] Y. Chang, L. Chen. Solve The Vehicle Routing Problem With Time Windows Via A Genetic Algorithm. *Discrete Continuous Dynamical Systems*, 1, 240249, 2007.
- [23] E. Alba, B. Dorronsoro. Computing nine new best-so-far solutions for Capacitated VRP with a cellular Genetic Algorithm. *Information Processing Letters* 98, 225230, 2006.
- [24] R. He, W. Xu, Y. Wang, W. Zhan. A Route-Nearest Neighbor Algorithm for Large-Scale Vehicle Routing Problem. *Intelligent Information Technology and Security Informatics*, 1, 390 393, 2010.

- [25] G. B. Dantzig, and J. H. Ramser. The truck dispatching problem. *Management science* 6.1, 80-91, 1959. Conference on Artificial Intelligence and Cognitive Science. 95-102. 2002. Springer
- [26] F. B. Pereira, J. Tavares, P. Machado, E. Costa. GVR: a new genetic representation for the vehicle routing problem. *Irish*