A Type-2 Fuzzy Genetic Approach to Uncertain & Dynamic Resilient Routing within Telecommunications Networks

Lewis Veryard, Hani Hagras
The Computational Intelligence Centre
School of Computer Science and Electronic Engineering
University of Esesx, Colchester, UK

Anthony Conway, Gilbert Owusu
BT Labs
BT plc, Adastral Park, UK

***Abstract*—Data and voice connections are an important part of any modern society, thus it is important for telecoms companies to provide resilience for network routing. Network conditions can fluctuate, redefining an optimal path through a network, which can introduce high levels of uncertainty. In this paper, we will present a type-2 fuzzy genetic system that can provide multiple routes between two locations with some ability to resist network fluctuations and uncertainty. We have tested the proposed system against a real world telecommunications data set with high levels of uncertainty, and compared this to a type-1 fuzzy system counterpart and the crisp system employed in the industry. The results indicate an average reduction in the occurrence of unique routes by 13%. Moreover, on average it increases the occurrence of a route occurring more than once by 26% when compared to a crisp version of the system. With Dijkstra's algorithm and A* being unable to perform the resilient routing task effectively.**

Keywords- Type-2 fuzzy logic, telecommunications, resilience routing, genetic algorithms.

I. INTRODUCTION

Throughout the corporate and academic world accurate modelling has become an integral part of complex or intelligent systems. One such complex system is the resilient routing system in traditional network systems. Two points on a network can usually be connected through multiple paths, which in turn indicates there is a shortest path between two locations. In a perfect world this would be our solution to real world routing problems, but this is not always the case as sometimes a connection in a route will fail. In this failure state there must be some kind of back up route available, allowing for a constant and seamless transmission to continue. This back up route is referred to as a *Resilient Route*. There are a couple of constraints imposed on *Resilient Routing* to ensure there is not a single point of failure within a solution. The routes must not intersect at any point other than the start and finish location. If it is possible to produce a *Resilient Route* which will result in a significant increase in cost, it must be taken.

There are a wide variety of shortest path algorithms used within routing problems [1] [2]. Dijkstra's Algorithm is the most commonly used due to its simplicity and effectiveness [3]. Dijkstra's Algorithm has been used in a wide variety of problems that require an effective and reliable routing algorithm. These problems range from robotics [4] to gas source selection within a network [5], and optimal parking strategies [6]. Over the years since Dijkstra's inception there has been various attempts to improve the algorithm, one of the most famous improvements being the A* algorithm [7]. A* addresses one of Dijkstra's major flaws, where it becomes very slow on larger networks. Dijkstra's Algorithm has a worst case performance of $O(|E| + |V|\log|V|)$ compared to A*s $O(|E|)$. Dijkstra's Algorithm is slow as it must visit every vertex in a graph. A* on the other hand doesn't visit every node although it requires an accurate heuristic. A* has found a home within the video game industry as a simple and light weight routing tool for agents [8], [9]. It has also been used to route on real road networks [10].

The traditional routing tends to be greedy in their solutions, a characteristic feature of the shortest path problem. By using traditional routing methods to find multiple paths the first path will consume all of the cheapest vertices leaving more expensive and therefore less desirable paths for the proceeding routes. When performing *Resilient Routing*, this is undesirable as we need to have the lowest cost for the entire solution not just for the *Primary Route*. This refers to the first route calculated. Data uncertainty can be an issue in real world networks as it can be difficult to ensure the accuracy of measurements or historical data. On the other hand, data that can be accurately measured may be highly susceptible to change and thus be volatile. This is unacceptable for telecommunications routing as this can cause an array of issues, in particular, as it can be expensive to keep changing routes. If many routes across a network are updated in parallel this can have an adverse effect on the accuracy of data. Finally, it can reduce user confidence in a system's accuracy. In order to address these issues a more complex strategy is required.

Optimisation algorithms such as Genetic Algorithms (GA's) [11] Particle Swarm Optimisation [12], Ant Colony Optimisation [13] and Simulated Annealing [14] are all very successful optimisers. There is no guarantee that these methods will present the perfect solution. That being said, they can be used to optimise a diverse selection of problems, ranging from Antenna Design [15], routing problems [16], [17], workforce optimisation [18] and network design [19].

Genetic Algorithms are good at optimising solutions with a complex set of requirements. GA's are able to travel the solution space in order to exploit the potentially best solution. *Resilient Routing* is a problem with a simple set of individual requirements. It is important to keep a perspective of the industrial use case of such a problem thus ensuring the solutions are applicable to the real world. In the telco domain, it is important to deliver a high quality of service to the end users at a low price, with minimal down time.

Resilient Routing becomes more challenging with the challenge of handling uncertainty. One technique that has had success dealing with uncertainty is fuzzy logic [20], [21], [22], [23]. Fuzzy Logic has had success when used as a control part of a control architecture in uncertain environment [24], [25], [26]. Genetic algorithms and fuzzy

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

logic have been combined together in various ways in the past for tasks including workforce optimisation [27] and financial classification [28].The *Uncertain Resilient Routing* problem [29],[30] has been tackled in the past using type-1 fuzzy logic and compared to Simulated Annealing. In this paper we look at the possible improvement presented by the use of type-2 fuzzy logic. Improvements in performance outlined from type-1 to type-2 fuzzy logic have been seen in control systems [31].

The rest of this paper is organised as follows. In Section II we will explain the *Uncertain Resilient Routing* problem in greater detail when applied to the telco domain. In Section III we will give a brief overview of Genetic Algorithms. In Section IV we will present the proposed type-2 fuzzy genetic approach to uncertain resilient routing. Section V presents the experiments and results. Finally, in Section VI, we conclude the paper and outline the future work.

II. Overview of the Uncertain Resilient Routing Problem in Telecommunications Networks

Telecommunication companies provide end to end routing for a wide array of different customers.. Customers expect a very high quality service with little to no downtime. One way to help maintain this this connection is to provide multiple routes between desired endpoints. This is known as *Resilient Routing*. *Resilient Routing* comprises of a primary route and one or more backup routes, known as resilient routes. To ensure that there is not a single point of failure, none of the routes within a solution can intersect except at the start and finish locations.

Dijkstra's Algorithm is known to be able to solve shortest path problems within a graph, that being said it is unsuitable for *Resilient Routing*. This unsuitability comes from its inherent greedy approach to routing where it consumes all the best vertices to reach its goal. Fig. 1 demonstrates the issue of using Dijkstra for this problem. The *Primary Route* as denoted in red (A, B, C, I) has a cost of 29 and the *Resilient Route* as denoted in blue (A, F, G, H, I) has a cost of 40, which brings the total cost to 69. When the *Primary Route* takes a less greedy approach, and thus increasing its cost, it allows the *Resilient Route* to take a more optimal route thus decreasing its cost. The optimal solution does just this, as seen in Fig. 2 the primary route as shown in red (A, B, D, E, I) has a cost of 32 and the *Resilient Route* as seen in blue (A, F, C, I) has a cost of 34 reducing the total cost to 66.

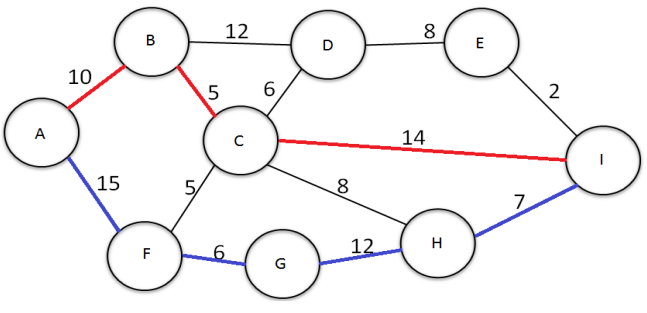

Fig. 1. Dijkstra resilience routing non-blocking example.

On the other hand, Dijkstra could prevent the existence of a *Resilient Routing* through its greedy nature, this situation is outlined in Fig. 3. The *Primary Route* as show in red (A, B, D, F, G) with a cost 14 is the only route available when using Dijkstra, which doesn't fulfil the requirement of having multiple independent routes between the start and finish locations.

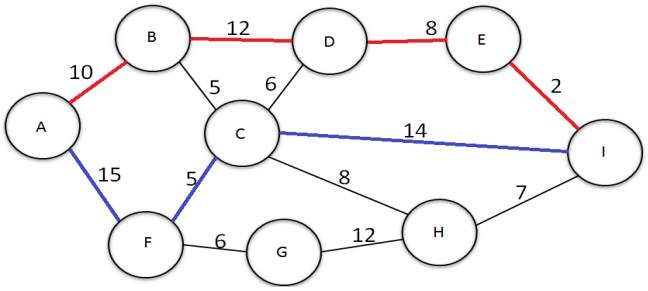

Fig. 2. Optimal resilience routing non-blocking example.

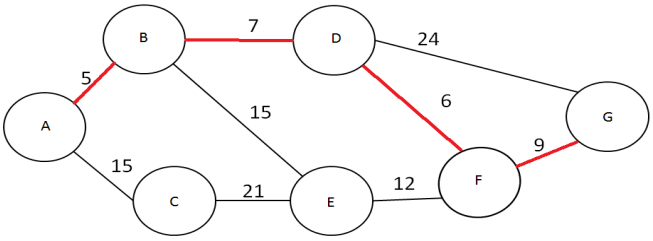

Fig. 3. Dijkstra resilience routing blocking example.

This is not the case in the optimal solution where the *Primary Route* takes a less optimal, and thus a more expensive route to the solution. Fig. 4 demonstrates this with *Primary Route* in red (A, B, D, G) costing 36 and the *Resilient Route* in blue (A, C, E, F, G) costing 57, which gives an increase total cost of 93 whilst satisfying a resilience requirement.

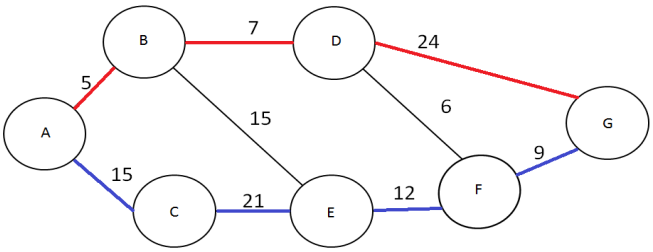

Fig. 4. Optimal resilience routing blocking example.

This problem becomes increasingly more complex when you consider a situation where the cost metrics are subject to minor changes over a period of time. If the difference in costs between routes is very small, and the cost functions are dynamically changing, it becomes very difficult to determine which route is the best over a given period of time. In an ideal world the routes would just be changed every time there is a new optimal solution for a given route. This is not actually a good idea in the real world for two reasons. Firstly, there is a cost with updating the routes for end to end service. Secondly, by changing the routes that are taken across a network, the related metrics tare impacted. For example, if all network traffic is routed based on latency then any location with a low latency will have a lot of traffic routed through it. Thus increasing its latency which reduces the latency elsewhere in the network. This type of uncertainty is ideal for a Fuzzy Logic System (FLS).

III. A Brief Overview of Genetic Algorthims

Genetic Algorithms (GA) take inspiration from nature i.e., the ideas of evolution, selection, and mutation known as operators. GA's have a set of chromosomes known as a population, which change across the course of many generations. The chromosomes are ranked through survival of the fittest, meaning the individuals most suited to the environment are ranked the best.

Each chromosome represents a solution to a pre-defined problem, where each chromosome is made up of a collection of genes. Each gene has a specific encoding strategy that allows it to store information. Genetic information is passed from generation to generation allowing the best gene to thrive. This information is passed on through the process known as crossover. When two chromosomes are selected to combine their genes they produce two offspring. These offspring are a combination of the partners with some genes from each of them. All four chromosomes are now returned to the population ready for the next generation of selection and crossover.

Just like in nature, sometimes a mutation can occur in the offspring when they are created. This mutation will change the chromosome in some way allowing for some genetic diversity to emerge within the population. A mutation will usually change one of the genes within an offspring chromosome. There are many different mutation strategies, most of which are problem and encoding dependant.

These chromosomes are not usually mated at random, they are chosen through a process known as selection. The most widespread selection methods are tournament and roulette wheel.

Chromosomes are ranked through some kind of pre-defined metric that measures the effectiveness of the represented solution. Through the use of this metric and the operators a dominant solution will rise to the top of the population. After a pre-defined number of generations, the GA will stop returning the solution with the best fitness score.

IV. The propsed Type-2 Fuzzy Genetic Approach to uncertain & Dynamic Resilient Routing within Telecommunivations Networks

The proposed system as shown in Fig. 5 is broken down into two main sections; Dijkstra's Algorithm and the GA with a Type-2 Fuzzy evaluator. A solution within the system is regarded as a *Primary Route* and N number of *Resilient Routes*. For the simplicity, let's assume that there is only one *Resilient Route*.

The system begins by passing the selected start and finish locations into Dijkstra's algorithm. If no route can be found, then the system stops and returns nothing. If a route is found it is stored as a *Primary Route* and removed from the network. At this stage Dijkstra is run again. If another route is found it stores it as a *Resilient Route*. These routes are then passed into the GA, if only one route is found then only one route is passed on.

The GA will always manipulate N-1 chromosomes where N represents the desired number of routes. The final route is always found through the use of Dijkstra's algorithm. The GA starts by generating an initial population. This is achieved by taking one of the starting routes and randomly removing one third of the nodes of that route from the network and re-running the Dijkstra algorithm. This will produce a new *Primary Route*, where all the nodes are then returned to the graph and the new *Resilient Route* is found in the same fashion as the initial *Resilient Route*.

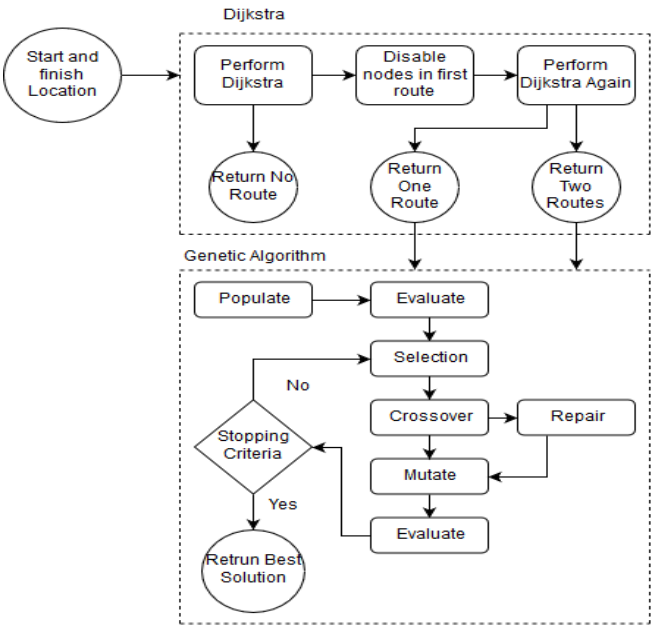

Fig. 5. Outline of the proposed system.

The GA is made up of five operators: *Selection*, *Crossover*, *Repair*, *Mutation* and *Evaluation*. The selection process of the GA is a best ranking survival of the fittest tournament. Solutions are ranked by the evaluation process. Next, a 10% random sample is taken from the population and the best ranked is selected as the first parent. The sample is then returned to the population without the first parent, and a second sample is taken and ranked, with the best one being taken as the second parent. The crossover operation takes the two parents and produces two offspring. The crossover point of the two parents is the centre point, so offspring one will have the left of parent one and the right of parent two as shown in Fig. 6.

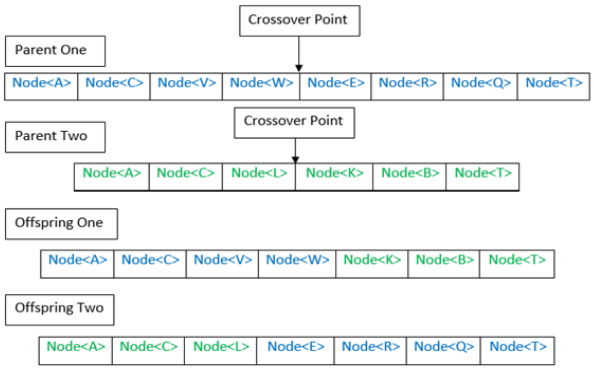

Fig. 6. GA Crossover example.

Once the crossover is completed the chromosomes are checked to see if they are valid routes. If they are valid, they are added to the population. If they are invalid, the repair operator is run again. Any chromosome that cannot be repaired is not added to the population. Additionally, if identical chromosomes are found within the population, they will not be added in order to keep genetic diversity amongst

the population. These two criteria were added to keep the GA focused on valid solutions.

The repair operator attempts to make invalid routes valid. There are two ways that a route can be invalid. Firstly, if a route uses the same node more than once, thus creating a cycle within our network. Secondly, if there are two nodes in a route that are not connected. The first is simple to fix, the cycle must be removed from the route, and this is achieved by removing all the nodes between the repeated nodes and one of the repeated nodes. The second scenario is harder to fix. Looking at offspring one in Fig.6, let's assume node W and node K are not connected. We use the Dijkstra's algorithm to connect them by one node. Now let's say that we ran Dijkstra's algorithm and it was connected by two nodes. The first scenario would be acceptable, but the second would not as it would give the offspring a length of nine, which is now longer than its longest parent. This final criterion was added in order to stop the chromosomes getting too long and thus losing the ability to converge into solution.

The mutation operation is an essential tool for any genetic population and it is no different in the case of the GA. It allows for the GA to explore the solution space in a new way. The mutation used in this GA is a random mutation. It works by randomly selecting a node from the chromosome then removing it and all nodes proceeding it. It then looks at the node prior to the selected one and picks a different node which is connected to it. From this newly selected node it uses the Dijkstra's algorithm to the destination node to find the rest of the route. Once the mutation process is concluded the population is checked to see if it's unique, if so the new route is added to the population.

The evaluation operation allows for solutions to be ranked in order of effectiveness for solving the *Resilient Routing* problem. As this is a routing problem, it is a minimisation task, i.e. we want the solution with the minimum total cost to be our dominant solution. A solution's cost is calculated based on its total cost of all the routes within a solution. The score itself is comprised of three components; a *Discount Factor* (DF), a count of the number of links (NLC) and the *Fuzzy Score* (FS). The DF adds a large value to any evaluated solution that doesn't have the desired number of routes. The NLC is used as a conflict resolution and only adds a small number in comparison to the other two. The fuzzy score is calculated on every connection in a route and summed together, it is the core component of the evaluation.

The fuzzy score is calculated using two inputs, a local input and a global input. As each node is evaluated, a new local membership function is calculated. As seen in Fig. 7, P1 is the minimum connection value, P2 is the average connection value and P3 is the highest connection value. The value D is obtained by computing the 10% of the distance between the centre and the outer points of the triangle. Then the D is added and subtracted from the outer point to obtain the two points that describe the footprint of uncertainty. This 10% is taken from the data that the experiments are performed upon, with 10% being the maximum that any value changes by. For example, the FOU of the lower point of LOW at P2 is calculated by P2 minus 10% the distance between P1 and P2. Whereas, the FOU of the upper point of low at P2 is calculated by P2 plus 10% the distance between P1 and P2.

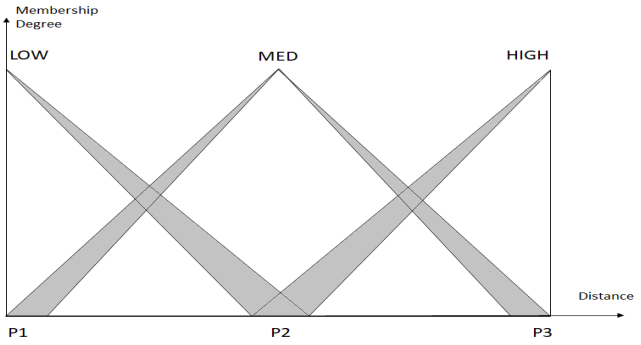

Fig.7. Local input Membership Function.

The *Global Membership* and *Output Membership* functions are identical as shown in Fig. 8 and are mapped to a box plot. The data for the box plot is the connection values of the entire network with P1 being the minimum value, P2 the lower quartile, P3 the median, P4 the upper quartile and P5 the maximum value. This allows for five membership functions to be extracted from the data. The FOU for these data sets is calculated in the same manner as for the local membership functions using 10% of the difference between the points. By using these membership functions and the rules shown in Table I, the fuzzy score is calculated.

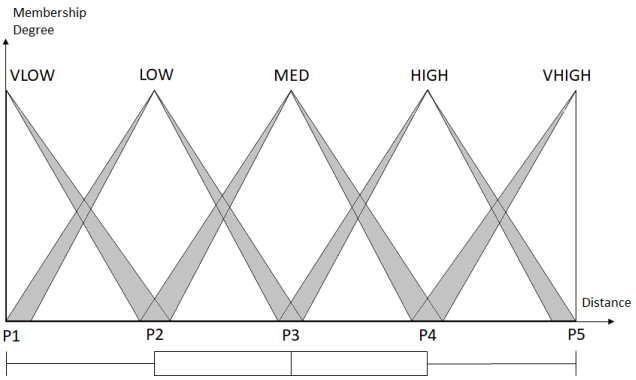

Fig. 8.Global and Output membership functions.

The primary part of the evaluation is the fuzzy score which tends to return a high value, something in the range of 1000 – 50000. The NLC returns a small value something in the range of 3 – 100. In addition to this it also has a weighting effect of 25% on it to reduce its influence. Therefore, NLC only has an effect on two solutions that have the same score and works as conflict resolution. The final stage of the evaluator is the DF which adds a large value to any solution that cannot find the desired number of routes. This is achieved by putting the fuzzy score to the power of the number of desired routes (DR) minus the number of actual routes (AR) as shown in equation (1).

$$DF = FS^{DR-AR} \quad (1)$$

The evaluation equation is shown with Equation (2) which combines together the three components; the *Discount Factor* (DF), *Number of Link Count* (NLC), and *Fuzzy Score* (FS) as follows:

$$Score = \frac{1}{DF * FS * (0.25 * NLC)} \quad (2)$$

With Equation (2) the solutions are ranked in ascending order with the best solutions having the smallest combined value.

TABLE I. THE EMPLOYED FLS RULEBASE

Global input	*Local input*	*Output*
VLOW	LOW	VLOW
VLOW	MEDIUM	VLOW
VLOW	HIGH	LOW
LOW	LOW	VLOW
LOW	MEDIUM	LOW
LOW	HIGH	LOW
MEDIUM	LOW	LOW
MEDIUM	MEDIUM	MEDIUM
MEDIUM	HIGH	HIGH
HIGH	LOW	HIGH
HIGH	MEDIUM	HIGH
HIGH	HIGH	VHIGH
VHIGH	LOW	HIGH
VHIGH	MEDIUM	VHIGH
VHIGH	HIGH	VHIGH

V. EXPERIMENTS AND RESULTS

The goal of the system is to be able to perform *Resilient Routing* through a network in real-world scenarios which are characterised by high levels of data uncertainty. The experiments have been set out in such a way that the consistency of routes can be measured. The experiments are run across 10 real world data sets, from similar geographies, where the data was extracted under high levels of uncertainty.

Dijkstra and A* have been tested in three different setting. First, the proposed system with a crisp evaluation method used in industry. Second, the proposed system with a type-1 fuzzy evaluator, and finally the proposed type-2 Fuzzy system as described in Section IV.

Each of these systems are run against 10 data sets. These 10 data sets have the same network layout, with slightly different cost values associated with their edges. The network itself is vast and complex allowing for many different routes between geographic locations to be found.

The results indicate that Dijkstra's algorithm and A* are unable to effectively resolve the resilient routing problem. One of two scenarios appear; either they are unable to produce a resilient route, or they produce a resilient route at a highly inflated cost in comparison to the other three.

The experiments have been run on 10 scenarios with the results summarised in Table II and Table III. There are four metrics that have been used in order to assess the quality of the results. The first metric is the number of *Unique Routes*, i.e. how many times does a route occur once The goal is to minimise this number. The second metric is *Routes of Multiple Occurrence*, ─ this is how many routes occurred more than once. The third metric is the *Max Same Route*, this ─many times the most common route occurred, we want to maximise this number. The final metric is the *Number of Independent Routes* that have occurred, this is how many different routes there are. We want to minimise this number. The most important metric to maximise is the *Max Same Route* and the most important metric to minimise is *Unique Routes*.

As shown in Table II the best scenario for the crisp system presents 7 identical routes and 3 identical routes, with no routes occurring only once. The best scenario for the Type-1 system is slightly better with 8 identical routes and 2 identical routes and no routes occurring only once. The best scenario the Type-2 system presents is the best possible outcome with all 10 routes being identical. The worst case scenario for the crisp system is the worst possible outcome with all routes being different. The worst case scenario for the Type-1 system is only slightly better with 8 of the routes being unique and two of them being the same. The Type-2 system is again slightly better with 6 unique routes and 2 lots of 2 routes being the same.

TABLE II. BEST AND WORST RESULTS FROM THE 10 ROUTING SCENARIOS

System	***Unique Routes***	***Routes of Multiple Occurrence***	***Max Same Route***	***Number of Independent Routes***
Best Crisp	0	2	7	2
Worst Crisp	10	0	1	10
Best Type-1	0	2	8	2
Worst Type-1	8	1	2	8
Best Type-2	0	1	10	1
Worst Type-2	6	2	2	8

Table III indicates the average results over the 10 routing scenarios for the three versions of the system. The crisp system has the highest average number of unique routes. The type-1 system has a slightly better result and the type-2 system has an even better average reduction in unique routes. The crisp system has the highest average number of routes with multiple occurrence, with the type-1 fuzzy system having a slight reduction upon this and Type-2 having a further reduction. According to this metric alone, the crisp system is performing the best, but this metric is heavily dependent on the *Max Same Route* metric. The average *Max Same Route* is the most important metric. If this is high it means that the system has found the same routes lots of times across multiple scenarios. The crisp system has relatively low average number of *Max Same Routes* with a slight improvement being shown by the type-1 system, and a much larger improvement by the Type-2 system. This metric indicates that on an average of 62% of cases, the type-2 system will find the same optimal solution. Whereas the type-1 fuzzy system finds the same route in 44% of case and the crisp system find the same solution in 36% of cases. Finally, the crisp system has a relatively high number of independent routes with the Type-1 system having a slight reduction on this and the Type-2 system having a more substantial reduction upon this.

Overall there is an improvement from crisp to Type-1 and again from Type-1 to Type-2. Type-2 has the average number of routes that are the same, and the lowest number of routes that occur only once

TABLE III. AVERAGE ROUTEING RESULTS OVER THE 10 ROUTING SCENARIOS

System	*Average Unique Routes*	*Average Routes of Multiple Occurrence*	*Average Max Same Route*	*Average Number of Independent Routes*
Crisp	3.6	2	3.6	5.6
Type-1	3.4	1.8	4.4	5.4
Type-2	2.3	1.6	6.2	4

VI. CONCLUSIONS & FUTURE WORKS

Consumers, businesses and government organisations are reliant on a stable and reliable network connection, be it a voice or data connection. Routing is not only important to networking, but also manufacturing, construction, traffic and infrastructure systems such as gas and water. In this paper we presented a type-2 Genetic approach to *Resilient Routing* within uncertain and dynamic telecommunications networks. The results showed that by using fuzzy logic the consistency of routes can be improved. Type-1 fuzzy logic on average increases the number of times the same route is found by 8% and on average reduces the number of times one route occurs by 2%. Whereas, Type-2 fuzzy logic on average finds the same route 26% more often than the crisp system, and reduces the number of routes that only occur once by 13% when compared to the crisp system. In the future, a much larger sample must be taken in order to back up these results with more confidence.

REFERENCES

[1] S. Meena and K. N. Geethanjali, "A Survey on Shortest Path Routing Algorithms for Public Transport Travel," *Glob. J. Comput. Sci. Technol.*, vol. 9, no. 2, pp. 73–76, 2010.

[2] J. Kubacki, L. Koszalka, I. Pozniak-Koszalka, and A. Kasprzak, "Comparison of heuristic algorithms to solving mesh network path finding problem," in *4th International Conference on Frontier of Computer Science and Technology, FCST 2009*, 2009.

[3] E. W. Dijkstra, "Solution of a problem in concurrent programming control," *Commun. ACM*, 1965.

[4] S. Julius Fusic, P. Ramkumar, and K. Hariharan, "Path planning of robot using modified dijkstra Algorithm," *2018 Natl. Power Eng. Conf. NPEC 2018*, pp. 1–5, 2018.

[5] Y. Cuan and X. Chen, "Application of improved Dijkstra algorithm in selection of gas source node in gas network," *Proc. 2012 Int. Conf. Ind. Control Electron. Eng. ICICEE 2012*, pp. 1558–1560, 2012.

[6] Yujin and G. Xiaoxue, "Optimal Route Planning of Parking Lot Based on Dijkstra Algorithm," *Proc. - 2017 Int. Conf. Robot. Intell. Syst. ICRIS 2017*, pp. 221–224, 2017.

[7] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, 1968.

[8] K. Khantanapoka and K. Chinnasarn, "Pathfinding of 2D & 3D game real-time strategy with Depth Direction A* algorithm for multi-layer," in *2009 8th International Symposium on Natural Language Processing, SNLP '09*, 2009.

[9] V. Bulitko, Y. Björnsson, N. R. Sturtevant, and R. Lawrence, "Real-time heuristic search for pathfinding in video games," in *Artificial Intelligence for Computer Games*, 2011.

[10] W. Zeng and R. L. Church, "Finding shortest paths on real road networks: The case for A*," *Int. J. Geogr. Inf. Sci.*, vol. 23, no. 4, pp. 531–543, 2009.

[11] J. H. Holland, "Genetic algorithms," *Sci. Am.*, vol. 267, no. 1, pp. 66–72, 1992.

[12] B. Chopard and M. Tomassini, "Particle swarm optimization," in *Natural Computing Series*, 2018.

[13] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, 1996.

[14] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science (80-.).*, 1983.

[15] G. S. Hornby, A. Globus, D. S. Linden, and J. D. Lohn, "Automated antenna design with evolutionary algorithms," *Collect. Tech. Pap. - Sp. 2006 Conf.*, vol. 1, pp. 445–452, 2006.

[16] E. Alba and B. Dorronsoro, "Solving the Vehicle Routing Problem by Using Cellular Genetic Algorithms," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3004, pp. 11–20, 2004.

[17] C. W. Ahn and R. S. Ramakrishna, "A genetic algorithm for shortest path routing problem and the sizing of populations," *IEEE Trans. Evol. Comput.*, vol. 6, no. 6, pp. 566–579, 2002.

[18] A. J. Starkey, H. Hagras, S. Shakya, and G. Owusu, "A Genetic Algorithm Based Approach for the Simultaneous Optimisation of Workforce Skill Sets and Team Allocation," in *Research and Development in Intelligent Systems XXXIII*, 2016.

[19] M. A. S. Barbosa and M. M. Gouvêa, "Access point design with a genetic algorithm," *Proc. - 2012 6th Int. Conf. Genet. Evol. Comput. ICGEC 2012*, pp. 119–123, 2012.

[20] J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems*. 2017.

[21] A. Pourabdollah, C. Wagner, J. H. Aladi, and J. M. Garibaldi, "Improved Uncertainty Capture for Nonsingleton Fuzzy Systems," *IEEE Trans. Fuzzy Syst.*, 2016.

[22] H. Hagras, C. Wagner, "Introduction to Interval Type-2 Fuzzy Logic Controllers - Towards Better Uncertainty Handling in Real World Applications", The IEEE Systems, an and Cybernetics eNewsletter, Issue 27, June 2009.

[23] J. Andreu-Perez, F.Cao, H. Hagras,G.Yang, "A self-adaptive online brain–machine interface of a humanoid robot through a general type-2 fuzzy inference system", IEEE Transactions on Fuzzy Systems, Vol. 26, No.1, pp. 101-116, Februray 2018

[24] C. Lynch. H. Hagras, V. Callaghan, "Embedded Interval Type-2 Neuro-Fuzzy Speed Controller for Marine Diesel Engines", Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2006), pp. 1340-1347, Paris, France, July 2006

[25] H. Hagras, V. Callaghan, M. Colley, M. Carr-West "A Fuzzy-Genetic Based Embedded-Agent Approach to Learning and Control in Agricultural Autonomous Vehicles", Proceedings of the1999 IEEE International Conference on Robotics and Automation, pp. 1005-1010, Detroit, U.S.A, May 1999

[26] H. Hagras, M. Colley, V. Callaghan, M. Carr-West, "Online Learning and Adaptation of Autonomous Mobile Robots for Sustainable Agriculture", Autonomous Robots, Vol. 13, pp. 37-52, July 2002.

[27] A. Starkey, H. Hagras, S. Shakya, and G. Owusu, "A multi-objective genetic type-2 fuzzy logic based system for mobile field workforce area optimization," *Inf. Sci. (Ny).*, 2016.

[28] M. Antonelli, D. Bernardo, H. Hagras, and F. Marcelloni, "Multiobjective Evolutionary Optimization of Type-2 Fuzzy Rule-Based Systems for Financial Data Classification," *IEEE Trans. Fuzzy Syst.*, 2017.

[29] L. Veryard, H. Hagras, A. Starkey, and G. Owusu, "A Fuzzy Genetic System for Resilient Routing in Uncertain Dynamic Telecommunication Networks," in *IEEE International Conference on Fuzzy Systems*, 2019 June.

[30] L Veryard, H Hagras, A Starkey, A Conway, G Owusu. "NNIR: N-Non-Intersecting-Routing Algorithm for Multi-Path Resilient Routing in Telecommunications Applications" International Journal of Computational Intelligence Systems. 2020 Mar.

[31] A. Sakalli, T. Kumbasar, E. Yesil, H. Hagras, "Analysis of the performances of type-1, self-tuning type-1 and interval type-2 fuzzy PID controllers on the Magnetic Levitation system". Proceedings of the 2014 IEEE International Conference on Fuzzy Systems, Beijing, China, July 2014.