International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 56, No. 2, April, 2023, pp. 301-307
Journal homepage: http://iieta.org/journals/jesa

Enhancing Path Planning of Assistive Robots in Complex Environments Using Geno-Fuzzy Algorithm

Aws Hazim Saber Anaz[1*], Omar A. Ibrahim[2], Ghazwan Alwan[3]

[1] Mechatronics Engineering Department, Engineering Collage, University of Mosul, Mosul 41002, Iraq
[2] Department of Electrical Engineering, Tikrit University, Tikrit, Saladin 34001, Iraq
[3] Department of Mechanical Engineering, Tikrit University, Tikrit, Saladin 34001, Iraq

Corresponding Author Email: aws.anaz@uomosul.edu.iq

https://doi.org/10.18280/jesa.560215

Received: 1 March 2023
Accepted: 15 April 2023

Keywords:
path planning, genetic algorithm, Assistive Robot, Geno-fuzzy

ABSTRACT

The path-planning algorithm is the central part of most v. The algorithm should consider fixed obstacles, furniture and building style, dynamic obstacles, humans, and pets. assistive robots encounter a challenging and complex environment with various obstacles during daily work. In addition, to maximize the service per hour, the robot has to select the optimum path. These challenges motivate the work toward an efficient path-planning algorithm that can handle complex environments. The proposed algorithm employs a designed genetic algorithm to look for the best path that maximizes the service area per hour. This genetic algorithm is then combined with a dynamic obstacle detection fuzzy system. This system relies on fuzzy membership zones. The algorithm decides whether the obstacle is dynamic or static according to speed, direction, and size. The Geno-fuzzy path planning algorithm is implemented in an assistive robot and tested in an actual environment. The algorithm implementation in a simulated environment of 100 BED hospitals in Iraq reveals a high-performance result. The test on a large scale without obstacles shows the ability of the algorithm to deal with more than 300 service points successfully. The local experiment on Webots proved the algorithm's performance to overcome dynamic obstacles and achieve safe traveling.

1. INTRODUCTION

Assistive Robotics is a challenging research field with a significant market and social impact [1]. Assistive robots are widely used in domestic and industrial applications. Consequently, increasingly new products are introduced on the market, products with new or enhanced tasks. Modern Assistive Robots (ARs) are designed with intelligent traveling techniques primarily used in domestic areas with indoor coverage and minimum possible time-consuming. These design challenges highlight the utilization of advanced path-planning algorithms. Smart mobile AR generally needs a coverage path planning algorithm. Coverage Path Planning (CPP) determines the route that covers all areas of an area of interest while avoiding obstacles [2]. Various ARs have been studied and improved in the last three decades for many service applications.

For cleaning and monitoring applications, Hu et al. [3] used a network of ARs with an algorithm for path planning to achieve fast cleaning. Their proposed algorithm was tested using five ARs in an uncomplicated environment without any obstacles for validation. However, their idea requires many ARs to achieve fast cleaning. In 2021, Ruan et al. [4] used the A* algorithm as a global path planning algorithm combined with a dynamic window algorithm (DWA) for local path planning. Their robot is mainly used for autonomous indoor cleaning and disinfection work. DWA is low-efficiency in the case of local trajectory planning since the robot cannot perceive the density of the obstacle [5]. sTetro robot used Zig-Zag CPP to clean the staircases with an RGBD camera to detect the staircases [6].

The robotic wheelchairs are studied and developed by Demetriou [7] and Gillham et al. [8] so that the corresponding person can be independent or semi–independent to create autonomy for persons who have limited mobility. In addition, Rao et al. [9] introduced an intelligent wheelchair supervised by Human-Robot Interaction denoted by HRI. Perrin et al. [10] implemented semi–autonomous robotic-based learning user behaviors system. The proposed system was stated on dynamical assumption based on network basics to predict the user's path at day and night times and hence assign a combination between the user's response and the chosen path to realize user habits.

In addition, AR is implemented widely for the persons with heavy motion to control their mobility independently. Kosuge Lab [11] developed a walk support system-based obstacle avoidance strategy to direct the user to reach the targeted location and select a suitable suit. Care-O-bot II is an AR that helps elderly persons in their ADL, the system configured by Fraunhofer Institute Manufacturing Engineering and Automation IPA. The development processes started by establishing a share control point scheme to analyze user inputs throughout the assistance. On the other hand, Morris et al. [12] presented a developed assistive robotic walker to provide guidance technology and iCane intelligent cane robot introduced by Di et al. [13] to assist patients that have muscle weakness in their lower limbs. Dubowsky et al. [14], which offered several services such as walk support, navigation, and a health monitoring system extensionally, Lyu et al. [15] complemented the system by adding more functions like a call to come service. Furthermore, Song and Jiang [16] presented a developed model of a walking assistive robot that detects the next motion of the users and supports them in slippery and unstable areas.

Additionally, ARs are used for Health care in older people-based environments. Threatt et al. [17] exposed an argumentation to study the ARs that depend mainly on environment type and how to support the medical environment such as nursing rooms. In the study [18], the innovation has inspired the designers to include furniture schemes to be a part of the environment to help older people get used to assistive robotic models. Furthermore, the home care project offered by Sugano and Shirai [19] under the given name WABOT-HOUSE depends strategically on applying interaction between the ARs and an environment (house) to monitor and protect user health conditions. Green et al. [20] invented an architectural robot named comfortable, which allowed older people to use a convenient room with several healthcare services. Lauretti et al. [21] suggested and implemented a motion planning system based on robotic devices to be adapted to assistive rehabilitation behavior. The strategy allows performing the personal motion of the sensor with high accuracy and efficiently considers the object's new position. Further, the system needs to be provided with real-time information about the environment and evaluate more patients to reach better accuracy during rehabilitation.

ARs are suitable for people who have impaired vision. AR for obstacle detection helps impaired vision by detecting the obstacle, processing the detection through an intelligent system, and giving an alarm to the user by vibration or sound. Mustapha et al. [22] designed a low-cost wearable shoe for visually impaired people to alarm obstacles. Smart Cane, presented by Dubowsky et al. [14], is widely used to provide a safe and independent movement for people suffering from vision disability, besides its help in lower limb disability. Sharma and Sharma [23] reviewed and discussed the challenges facing Smart Cane, Wearable Obstacle Detection System, E- Drive, and handheld computer-based Tour Guide, which are the primary tools used for impaired vision people.

In summary, ARs work in different environments for various applications, having the common need for flexible and general, large-scale path planning and obstacle avoidance algorithms. In this paper, a large-scale path planning algorithm is designed and implemented for assistive robots that can handle the complex environment during daily working activity challenges with the optimum path to maximize the service area per hour along with safe traveling ability.

The designed algorithm clusters the large-scale environment and then combines the genetic algorithm for the best path in the cluster with a fuzzy inference system for smooth and safe obstacle avoidance. Based on this summary, the paper is discussed.

2. METHODOLOGY

The proposed Geno-fuzzy system is mainly a large-scale path-planning genetic algorithm (GA) combined with a fuzzy controller for smooth traveling in complex environments and includes fuzzy membership of obstacles to dangerous zones in the path cost calculations. This combination of the genetic algorithm and the fuzzy membership is the reason for our suggested term, the Geno-fuzzy system path planning algorithm. Our algorithm is heavily based on K-Means, hierarchical clustering, and GA. The following sections summarize the main algorithms and methods for implementing our approach.

2.1 Hierarchical Clustering

Hierarchical Clustering (HC) allows clustering points without centroids, making it different from other clustering algorithms such as K-Means. Because HC does not rely on randomly chosen centroids, it produces the same clusters for a given dataset every time it is run.

To initialize the algorithm, the pairwise distance matrix needs to be calculated. Each cell represents the distance between the point denoted by row 'i' and the point denoted by column 'j'. When executing HC, it is helpful to consider each point as a singleton (each point as a cluster). The clustering is carried out as follows (Figure 1).

1. *Search the table to find the shortest distance between clusters C1 and C2.*
2. *Now merge the two clusters C1 and C2, into one new cluster Cnew.*
3. *Change any distance between Cnew and the remaining clusters Ci to… min(||C1-Ci||2, ||C2-Ci||2).*
4. *Repeat until N clusters remain.*

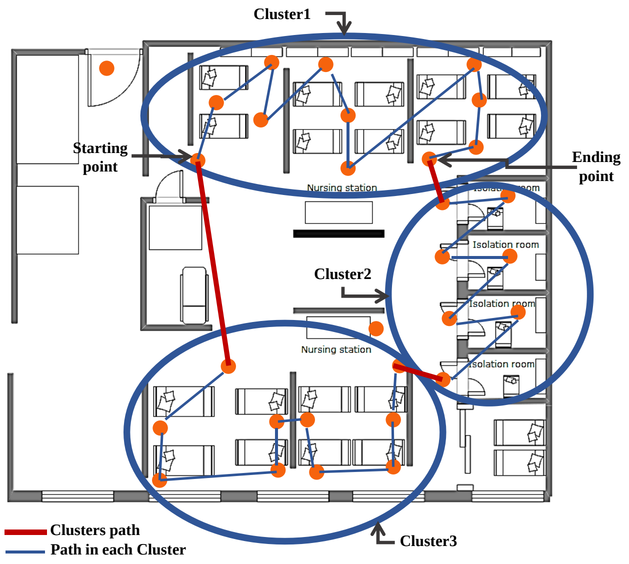

Figure 1. Illustrates how inter-cluster routes are determined for single linkage clustering using start and endpoints

Note that in the algorithm, we also refer to points as clusters. One may think of each point as a single-element cluster. In steps 1 and 2, we look for the minimum distance to merge the closest clusters into a new cluster. In step 3, we represent the distance between two clusters as the distance between their closest points. This is referred to as a single linkage. Other linkages are listed below. Step 4 then tells us to continue grouping until we obtain N clusters:

Average – The average of all distances between points of two clusters.

Centroid – The distance between computed centroids of two clusters.

Complete – The furthest distance between points of two clusters; the opposite of single.

Median – The center value of all distances between points of two clusters.
Ward – Inner squared distance between points of two clusters.
Weighted – Weighted average of the distance between points of two clusters.

2.2 K-Means

K-Means is a popular and relatively simple way of clustering data. Opposite of Hierarchical Clustering, it can produce clusters of varying size and location. We will not go in-depth with K-Means here as many resources cover the subject. The algorithm is as follows.

1. *Randomly Initialize K centroids.*
2. *Associate each data point with a centroid based on its proximity to the centroid, creating K clusters.*
3. *Update each centroid by moving it closer to the mean of its cluster.*
4. *Repeat steps 2 and 3, pending the centroids no longer move.*

2.3 Fuzzy inference system

A fuzzy system is widely used in applications when soft transition without complex nonlinear control analysis is desirable. Soft navigation of the assistive robot needs a fuzzy inference system to achieve smooth traveling among high obstacles density environment. In this paper, we use this feature of FIS to be part of the system. The FIS usually has four major parts: fuzzifier, knowledge base, inference engine, and defuzzifier. The fuzzifier translates a real crisp input into a fuzzy membership value identifying the input degree of membership. Defuzzification translates fuzzy control values to crisp numbers. The controller's decision-making logic runs by the inference engine. It uses fuzzy inference procedures and fuzzy implications to infer the fuzzy control travels.

2.4 Genetic Algorithms

Genetic Algorithms (GA) are commonly used to find a solution to complex problems such as path planning algorithms. They model the natural process of evolution to solve challenging optimization problems. Within the algorithm is a group of members referred to as the population. Each member of a population represents a candidate solution to the problem. They find an optimal solution by changing members of a population over many iterations of the algorithm or generations. Each of these members is modeled after a chromosome. Like the natural world, the chromosomes can be crossed over and mutated. The aim is to change the chromosomes of a population for the better so that they provide a more optimal solution to the problem. These two operations and how we implemented them will be covered in more in-depth later. Below we show the standard structure of GA.

1. *We encode the problem. A bit pattern, string, or combination of numbers can represent each chromosome.*
2. *Initialize the population randomly.*
3. *Compute the fitness of each member of the population.*
4. *Based on each individual's fitness, select a percentage of the population using a selection operator.*
5. *Cross over a certain percentage of the population and obtain the next generation.*
6. *Mutate a certain percentage of the new population.*
7. *Repeat steps 3 through 6 until we reach some terminating condition.*

The number of pages for the manuscript must be no more than ten, including all the sections. Please make sure that the whole text ends on an even page. Please do not insert page numbers. Please do not use the Headers or the Footers because they are reserved for technical editing by editors.

3. IMPLEMENTATION

Geno-fuzzy path planning algorithm is designed to achieve safe and fast access of the assistive robot to assist service request points as much as possible. The service request points number depends mainly on the real word applications the AR is serving. We designed the Geno-fuzzy path planning algorithm for many service request point applications in hospitals, hotels, airports, and universities. The designed algorithm structure is illustrated in Figure 2. The algorithm starts the global path planning and then feeds the clusters to the local path planning algorithm. The local path planning algorithm uses the output of FIS for dynamic and static obstacles severity with the points in its cluster to provide optimum path selection and safe travel of the AS while implementing its scheduled duties. The designed algorithm implementation is discussed in this section based on Figure 2.

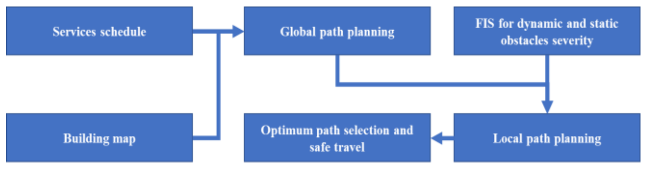
Services schedule
Global path planning
FIS for dynamic and static obstacles severity
Building map
Optimum path selection and safe travel
Local path planning

Figure 2. Geno-fuzzy path planning algorithm structure

3.1 The global path planning algorithm

The designed approach starts by running the global path planning algorithm, either Hierarchical Clustering (HC) or K-Means on the service requested points combined with the building map where the AR is working. This produces K clusters. We then define the entry point and exit points for each cluster. When using HC, these points should already be defined. When using K-Means, they must be found by iterating over every possible pair of points between two clusters to find the minimum distance between the clusters. Once we know the path cost between clusters, we can find the optimum possible intra-cluster path as shown in Figure 1.

3.2 FIS for dynamic and static obstacles severity

AR needs an obstacle avoidance strategy to achieve safe travel while implementing tasks in an environment with people working around and furniture fixed on it. Dynamic (people) and static (furniture) obstacles must be considered in calculating the fitness for each path in GA to detect their severity and avoid any harmful possibility. This motivates the implementation of FIS to detect dynamic and static obstacles severity and include it in the cost function of GA. To FIS design FIS, we used Dataset [24] to generate severity indicators to feed the GA cost function and provide traveling suggestions for safe obstacle avoidance. Five zones represent the severity of an obstacle. For every zone of speed and distance, seven points are incorporated that begin from 0° to 90° degrees located on the right side of the robot and 0° to 90°

degrees to the left side. For every zone, ten readings are recorded along the radial path from obstacle to AR. For zone 1, the velocity=10 cm/sec, distance = 60 cm, and the covered angles are -90,-75,-60,-45,-30,-15,0,15, 30,45, 60, 75, and 90. The total samples are 2080. The designed FIS has three inputs: relative velocity, relative distance, and the relative angle between the AR and the obstacle; outputs are the AR'se velocity and steering angle. The outputs are fed to the cost function of the local path planning algorithm GA to represent the severity of the obstacle to the AR.

3.3 The local path planning algorithm

After applying the global path planning algorithm, next, each cluster is passed through a Genetic Algorithm (GA) to obtain the shortest intra-cluster path for that cluster. The genetic algorithm is as follows [25]:

```
1. Calculate the path cost for every point pair
2. Initialize the population with different paths
3. Calculate the cost function for each path using
Cost function= Steering Angle /(velocity*Distance$_{ij}$ )
Where Distance$_{ij}$ represents the distance between point i and
   point j where point i and point j ϵ Cluster$_k$
4. Conduct Selection Operator
5. Conduct Crossover Operator to find Child X and Child Y
   with the Forward & Backward Methods
6. Conduct Inversion Mutation Operator
7. Repeat 3-6 for Many Iterations
8. Return the optimum path obtained
```

The above algorithm depicts the flow of our GA. Three things, in particular, must be explained: encoding and selection, crossover, and mutation operators. The members of our population are encoded as a possible route through a cluster. A possible route would be represented as (1, 5, 4, 7, 6, 2, 3). Point 1 is the entry point to the cluster, and Point 3 is the exit point of the cluster. Everything between point 1 and point 3 is a possible route through the cluster, and a particular route's fitness is its length. It is important to note that neither the entrance nor exit point is modified during the GA. The selection operator is accomplished by sorting the members in ascending order, and a certain percentage is chosen, always keeping the number 1 solution. Using the crossover operator, two parents are randomly picked to generate two children, child x and child y. The crossover operator generates child x & y using the forward and backward method. The forward method is as follows.

```
1. Pick a point at random, point x, and add it to an empty
   vector, v.
2. Find the minimum path cost between point x and the next
   point in parent 1 and parent 2, pointmin.
3. Append pointmin to v.
4. Repeat steps 2 & 3 for pointmin as point x until all points
   are covered.
```

The algorithm above finds Child x. To find Child y, we use the backward method. The backward method is the same as the forward method, except the next point is the point behind point x, not the point in front of point x.

The mutation operator is then applied to a percentage of the population. The mutation operator used is an inversion [25]. We randomly select two points within a member and reverse the order of points between those two points. After the GA is repeated for every cluster generated by HC, we should theoretically have an optimized path.

4. EXPERIMENTS, RESULTS, AND DISCUSSION

To test the Geno-Fuzzy path planning algorithm, we use two challenging experimental set-ups: the first one is a large scale where we test the global algorithm behavior while facing a large number of points to visit, and in the second scenario, we test the local path planning with obstacles avoiding ability.

4.1 Large-scale test

In this test, the algorithm is analyzed with datasets provided by Universität Heidelberg [26], then implemented on one of Iraq's territory challenges represented by Bashiqa Hospital. The Universität Heidelberg datasets come in many forms. Some sets represent cities spread out geographically, some have certain patterns, and others are seemingly randomly distributed points. The experiments utilize two of the larger geographic datasets and two of the smaller geographic datasets. These datasets are named usa13509, d15112, att532, and rat783 respectively. For the clustering step, we compared K-means and some implementations of agglomerative hierarchical clustering with different numbers of clusters. We also tried to find the optimal number of clusters using the dendrogram manually, but after increasing the number of experiments, we decided not to run multiple clusters and pick the best result. Hence, we generated five random values of K between 1 and 10 and ran three different clustering algorithms with those values of K, keeping the best answer among all runs for each algorithm. We ran the global path planning algorithm on the clusters to find the best route between clusters and then ran the local path planning algorithm on the cities as covered points inside each cluster to find the best intra-cluster path. Clusters are connected in a minimum path. The closest points between every two clusters were found. These closest two points are then defined as their respective cluster's start or endpoints. After running each clustering algorithm multiple times with multiple combinations of parameters on each file, the results are shown in Table 1, where we compared the best computation and length of the path between different configurations. Table 1 also shows the minimum length found by the authors [27], and based on that, we computed the error value as the difference between our best result and the best result of previous works.

Table 1. Results as compared with optimum path of every Dataset

# Records	K	Cluster Name	Best Length	File Length	Best Error	% Error
rat783	1	K-means	8835.19	8806	29.1948	0.33%
rat783	1	single	8836.68	8806	30.681	0.35%
rat783	1	complete	8836.26	8806	30.2636	0.34%
rat783	4	K-means	9249.77	8806	443.772	5.04%
a280	4	K-means	5094.05	4230	864.0462	20.43%
a280	4	single	4506.05	4230	276.0504	6.53%
a280	4	complete	4673.52	4230	443.5247	10.49%

Results show that our algorithm primarily generated better paths with shorter lengths for smaller data sets and smaller values of K. Especially by keeping the value of K equal to 1, the local path planning algorithm always works great for any size of the algorithm, and our results are always better or comparable to the references. However, increasing the number of clusters, as expected, increases the length of the best path, and this result is entirely dependent on the type of clustering algorithm we use. For example, the complete and average algorithms always generated the best results, while the median and weighted results were often the worst due to the structural similarities of the geographic datasets. For K-means, we got different results because of the dependency on the initialization- although we ran K-means multiple times and got the best result out of that. The main reason for this situation is the property of the k-means, which always tries to find compact circular-shaped clusters. We ran the agglomerative clustering algorithm with multiple parameters. Besides comparing the path length, we also compared the computational times for each clustering algorithm and the combination of parameters. As expected, in most cases, K-means was the fastest clustering method, but in the whole process 'single' algorithm always generated the results in a better time. We found some cases where k-means generated clusters with very few points inside them, while the 'single' algorithm for the same case made the cluster sizes more equal (not all cases). We could not compare different configurations due to the massive number of configurations we used; instead, we always kept the best results among all those configurations.

Due to the lack of ground truth to compare our computational time with other approaches in the literature, we compared the time between different clustering techniques and configurations. It can be concluded that increasing the number of clusters significantly improved computational time, especially in large datasets. Table 2 shows the computational time for some values of K, and as expected, the algorithm operates faster on the small clusters, and the result is generated quicker. After testing the algorithm's performance with well-known datasets, the algorithm performs reasonably well in most cases.

Table 2. Results for clustering algorithms

Algorithm	K	Avg. Time	Error %
K-Means	1	441.91	0.33%
Single	1	503.90	0.35%
Complete	1	520.32	0.34%
K-Means	4	383.78	20.43%
Single	4	364.85	6.53%
Complete	4	358.28	10.49%
K-Means	64	547.72	25.14%
Single	64	443.48	15.22%
Complete	64	497.31	18.40%

Next, the algorithm is implemented on the Bashiqa Hospital ground floor. Bashiqa Hospital is one of a series of 100 beds hospitals in Nineveh governorate, and 100-bed hospitals share the same map. The map illustrated in Figure 3 contains points in orange color that can be visited based on the possible service schedule provided to the AR. The entrance points depicted in Figure 3 in pink color are the waiting spot of the AR. The total number of points is 126 including the entrances and doors, and doors are also considered service points. Distances (traveling cost) between every pair of points are calculated based on the position in the map and the possibility of connection to generate the Dataset of the map.

After implementing the Geno-fuzzy path planning algorithm to travel from point A to point B in Figure 3, the results show the selected path cost 146 m while the optimum path is 135m with an %8 error and an average computation cost of 256 seconds. The experiment is implemented on a Windows 10, 64-bit intel machine with a 2.50 GHz core i5 CPU and 8 GB RAM. This implementation is obstacles free test and the K=2 clusters. Hence, the FIS did not contribute.

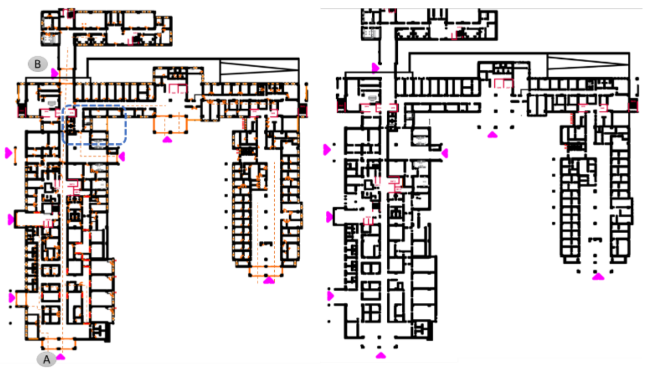

Figure 3. 100 BED ground floor map; (Left) with points in orange color, and (Right) without the points

4.2 Local Geno-fuzzy test

The previous test for the whole map is challenging to be implemented in the Webots environment to simulate the Global part of the algorithm. Instead, we simulated the global and compared the results with known optimum paths in the previous section. The algorithm is tested on a complex environment, where static and dynamic obstacles are presented. We select part of the map, the region surrounded by the dotted blue line in Figure 3, to place a variety of obstacles and test it using Webots. The designed simulation environment is a hospital scenario, as shown in Figure 4. The selected AR for simulation is TIAGo Iron, designed by PAL Robotics. TIAGo Iron is a two-wheeled human-like robot. The simulation started with two dynamic obstacles, a human model programmed to walk randomly in different directions. Next, we keep adding two human models to the simulation and compute the time duration to reach the target for a scheduled service plan and the path distance. The scheduled service plan for this test is depicted in orange spots in Figure 4. The implementation results on Webots with two, four, six, and eight dynamic obstacles are summarized in Table 3, and the Webots implementation is shown in Figure 5.

From Table 3, it is clear that increasing the number of dynamic obstacles causes an increase in traveling path distance and time cost. Moreover, human models hit the AR almost exponentially with their number.

For safety, the proposed algorithm should provide collision-free traveling. However, we may accept this result since the human model is programmed to walk randomly without the real-world human sense of collision.

Table 3. Summarized results of implementing a Geno-fuzzy path planning algorithm on Webots with two, four, six, and eight dynamic obstacles

Number of dynamic obstacles	Distance (m)	Traveling time (sec)	# Collision
Two dynamic obstacles	17.5	228	0
Four dynamic obstacles	18.3	403	1
Six dynamic obstacles	20.1	830	4
Eight dynamic obstacles	25.4	1567	9

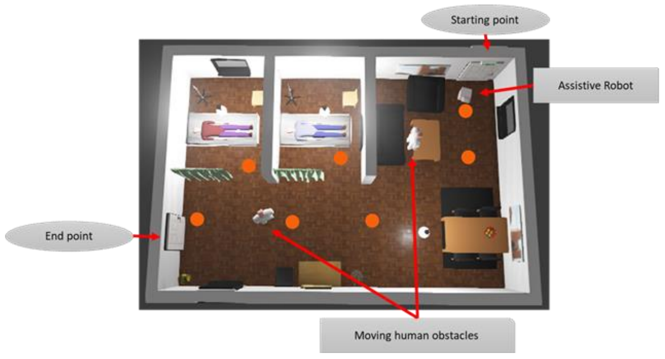

Figure 4. Webots designed simulation environment with hospital scenario

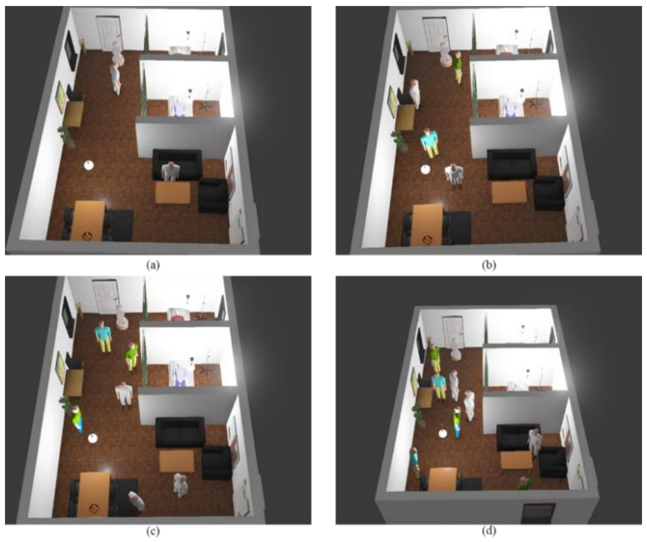

Figure 5. Experiments of the hospital scenario on Webots with (a) Two, (b) Four, (c) Six, (d) Eight dynamic obstacles

5. CONCLUSIONS

The proposed Geno-Fuzzy path planning algorithm is designed to generally support broad types of assistive robots in complex environments, mixed with many obstacles during daily working activity. The Geno-Fuzzy path planning algorithm provides an acceptable impact on Iraq's territory when implemented in the simulated environment of 100 BED hospitals. The test on a large scale without obstacles shows the ability of the algorithm to deal with more than 300 service points successfully. The local experiment on Webots proved the algorithm's performance to overcome dynamic obstacles and achieve safe traveling. For future works, the whole 100-bed hospital can be included in the Webots environment and improve the human model, adding the human sense of collision during walking and running to test the algorithm fairly.

REFERENCES

[1] Siciliano, B., Khatib, O. (2016). Springer Handbook of Robotics, Springer Handbooks. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-32552-1

[2] Galceran, E., Carreras, M. (2013). A survey on coverage path planning for robotics. Robotics and Autonomous Systems, 61: 1258-1276. https://doi.org/10.1016/j.robot.2013.09.004

[3] Hu, J., Lennox, B., Arvin, F. (2021). Collaborative coverage for a network of vacuum cleaner robots. In Fox, C., Gao, J., Ghalamzan Esfahani, A., Saaj, M., Hanheide, M., Parsons, S. (Eds.), Towards Autonomous Robotic Systems, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 112–115. https://doi.org/10.1007/978-3-030-89177-0_11

[4] Ruan, K., Wu, Z., Xu, Q. (2021). Smart cleaner: A new autonomous indoor disinfection robot for combating the COVID-19 pandemic. Robotics, 10: 87. https://doi.org/10.3390/robotics10030087

[5] Mai, X., Li, D., Ouyang, J., Luo, Y. (2021). An improved dynamic window approach for local trajectory planning in the environment with dense objects. Journal of Physics: Conference Series, 1884: 012003. https://doi.org/10.1088/1742-6596/1884/1/012003

[6] Le, A.V., Kyaw, P.T., Mohan, R.E., Swe, S.H.M., Rajendran, A., Boopathi, K., Nhan, N.H.K. (2021). Autonomous floor and staircase cleaning framework by reconfigurable sTetro robot with perception sensors. Journal of Intelligent & Robotic Systems, 101: 17. https://doi.org/10.1007/s10846-020-01281-2

[7] Demetriou, G.A. (2009). Robotic wheelchairs. In 2009 9th International Conference on Information Technology and Applications in Biomedicine. Presented at the 2009 9th International Conference on Information Technology and Applications in Biomedicine (ITAB 2009), IEEE, Larnaka, Cyprus, pp. 1-4. https://doi.org/10.1109/ITAB.2009.5394345

[8] Gillham, M., Howells, G., Spurgeon, S., Kelly, S., Pepper, M. (2013). Real-time doorway detection and alignment determination for improved trajectory generation in assistive mobile robotic wheelchairs. In 2013 Fourth International Conference on Emerging Security Technologies, IEEE, Cambridge, United Kingdom, pp. 62-65. https://doi.org/10.1109/EST.2013.18

[9] Rao, R.S., Conn, K., Jung, S.H., Katupitiya, J., Kientz, T., Kumar, V., Ostrowski, J., Patel, S., Taylor, C.J. (2002). Human robot interaction: Application to smart wheelchairs. In Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), IEEE, Washington, DC, USA, pp. 3583-3588. https://doi.org/10.1109/ROBOT.2002.1014265

[10] Perrin, X., Colas, F., Pradalier, C., Siegwart, R., Chavarriaga, R., del R. Millan, J. (2011). Learning user habits for semi-autonomous navigation using low throughput interfaces. In 2011 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, Anchorage, AK, USA, pp. 1-6. https://doi.org/10.1109/ICSMC.2011.6083633

[11] Song, K.T., Lin, C.Y. (2009). A new compliant motion control design of a walking-help robot based on motor current and speed measurement. In 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, IEEE, pp. 4493-4498. https://doi.org/10.1109/IROS.2009.5354293

[12] Morris, A., Donamukkala, R., Kapuria, A., Steinfeld, A., Matthews, J.T., Dunbar-Jacob, J., Thrun, S. (2003). A robotic walker that provides guidance. In 2003 IEEE

International Conference on Robotics and Automation (Cat. No.03CH37422), IEEE, Taipei, Taiwan, pp. 25-30. https://doi.org/10.1109/ROBOT.2003.1241568

[13] Di, P., Huang, J., Sekiyama, K., He, S., Nakagawa, S., Chen, F., Fukuda, T. (2012). Optimal posture control for stability of intelligent cane robot. In 2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication, IEEE, Paris, France, pp. 725-730. https://doi.org/10.1109/ROMAN.2012.6343837

[14] Dubowsky, S., Genot, F., Godding, S., Kozono, H., Skwersky, A., Yu, H., Yu, L.S. (2000). PAMM - a robotic aid to the elderly for mobility assistance and monitoring: A "helping-hand" for the elderly, in: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), IEEE, San Francisco, CA, USA, pp. 570-576. https://doi.org/10.1109/ROBOT.2000.844114

[15] Lyu, S.-R., You, W.-T., Chen, Y.-S., Chiang, H.-H., Chen, Y.-L. (2014). Development of robotic walking-aid system with mobility assistance and remote monitoring, in: 2014 IEEE International Conference on Automation Science and Engineering (CASE), IEEE, Taipei, pp. 830-835. https://doi.org/10.1109/CoASE.2014.6899422

[16] Song, K.T., Jiang, S.Y. (2011). Force-cooperative guidance design of an omni-directional walking assistive robot. In 2011 IEEE International Conference on Mechatronics and Automation, IEEE, Beijing, China, pp. 1258-1263. https://doi.org/10.1109/ICMA.2011.5985842

[17] Threatt, A.L., Merino, J., Green, K.E., Walker, I.D., Brooks, J.O., Ficht, S., Kriener, R., Mossey, M., Mutlu, A., Salvi, D., Schafer, G., Srikanth, P., Xu, P., Manganelli, J., Yanik, P. (2012). A vision of the patient room as an architectural-robotic ecosystem. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Vilamoura-Algarve, Portugal, pp. 3322-3323. https://doi.org/10.1109/IROS.2012.6386261

[18] Chang, W.-L., Šabanović, S. (2014). Exploring Taiwanese nursing homes as product ecologies for assistive robots, in: 2014 IEEE International Workshop on Advanced Robotics and Its Social Impacts. Presented at the 2014 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), IEEE, Evanston, IL, USA, pp. 32-37. https://doi.org/10.1109/ARSO.2014.7020976

[19] Sugano, S., Shirai, Y. (2006). Robot design and environment design - waseda robot-house project. In 2006 SICE-ICASE International Joint Conference, Busan, Korea. https://doi.org/10.1109/SICE.2006.314981

[20] Green, K.E., Wakjer, I.D., Brooks, J.O., Mohktar, T., Smolentzov, L. (2009). ComforTABLE: A robotic environment for aging in place. In Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction, ACM, La Jolla California USA, pp. 223-224. https://doi.org/10.1145/1514095.1514147

[21] Lauretti, C., Cordella, F., Guglielmelli, E., Zollo, L. (2017). Learning by demonstration for planning activities of daily living in rehabilitation and assistive robotics. IEEE Robotics and Automation Letters, 2: 1375-1382. https://doi.org/10.1109/LRA.2017.2669369

[22] Mustapha, B., Zayegh, A., Begg, R.K. (2013). Wireless obstacle detection system for the elderly and visually impaired people. In 2013 IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), IEEE, Kuala Lumpur, Malaysia, pp. 1-5. https://doi.org/10.1109/ICSIMA.2013.6717949

[23] Sharma, A., Sharma, N. （2013). Sensor based technologies for visually impaired: A comparative study. International Journal of Computer Applications, 975: 8887.

[24] Yosif, Z.M., Mahmood, B.S., Saeed, S.Z. (2022). Artificial techniques based on neural network and fuzzy logic combination approach for avoiding dynamic obstacles. Journal Européen des Systèmes Automatisés, 55(3): 339-348. https://doi.org/10.18280/jesa.550306

[25] Yu, Y., Chen, Y., Li, T. (2011). A new design of genetic algorithm for solving TSP. In 2011 Fourth International Joint Conference on Computational Sciences and Optimization, IEEE, Kunming and Lijiang City, China, pp. 309-313. https://doi.org/10.1109/CSO.2011.46

[26] Chen, S.M., Chien, C.Y. (2011). Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques. Expert Systems with Applications, 38: 14439-14450. https://doi.org/10.1016/j.eswa.2011.04.163

[27] Jebari, K., El moujahid, A., Bouroumi, A., Ettouhami, A. (2012). Unsupervised fuzzy clustering-based genetic algorithms to traveling salesman problem. In 2012 International Conference on Multimedia Computing and Systems, IEEE, Tangiers, Morocco, pp. 1013-1015. https://doi.org/10.1109/ICMCS.2012.6320145