DGAA*: A Dynamically Repaired Double-Strand Break Genetic Algorithm with Pheromone Guided A*

J. Ridley[2] . Tanveer[1], RC. Voicu[1], A. Charles[1], R. Ahmed[1], and O. Irfan[1]

[1] Department of Robotics and Mechatronics Engineering, Kennesaw State University, Marietta, Ga 30060, USA
[2] Department of Computer Science, Kennesaw State University, Marietta, Ga 30060, USA

Abstract. Path planning and obstacle avoidance are key components in automating robots, which have two conventional means of traversal: wheels and legs. Wheeled robots have the advantage of stability, yet their performance drops when an incline is introduced. Legged robots are less stable but are better at exploring environments with rougher terrain. This journal introduces DGAA*: a hybrid path-finding algorithm combining A-star (A*), Ant Colony Optimization (ACO), and Genetic Algorithm (GA) that will serve as the basis for an autonomous quadrupedal robot system.

Keywords: Path-Planning · Automation · Robotics · Algorithm · Simulation.

1 Introduction

1.1 Current State of Robotics

As the use of robotics expands throughout industries and applications such as agriculture, construction, and exploration of hazardous environments, ground mobile robotics become increasingly important in aiding human endeavors. Unmanned Aerial Vehicles (UAVs) are good for applications concerning environmental analysis (surveillance, reconnaissance, etc) yet lack stability due to interaction with objects and organisms native to said environments. This is where ground mobile robots excel: terrain maneuverability allows for equipment of more technologies like increased amounts of sensors and robotic limbs, since there is less concern in making said robotic unit more air resistant, and heavier payloads since the unit in question is being supported by the earth and/or structures protruding from Earth. This expands the ability of the robotic unit due to there being more to work with.

The two main methods of terrain traversal are wheels and legs [10][16] . Wheeled robots have the advantage of stability yet suffer when exploring complex terrains, whereas legged robots are faulted in the realm of stability yet move more easily within said difficult terrains as compared to their wheeled counterparts.

Legged robots are commonly implemented with two, four, or six legs. Two legs grant more speed but are the least stable of the three versions, while six leg robots lack in speed but have the utmost stability from the aforementioned. Though quadrupedal robots are a compromise between the two templates, giving them both moderate speed and stability, they are still slower than their wheeled counterparts on smooth surfaces and are prone to slippage, which subtracts from the time it takes for them to reach their goal.

1.2 The Nature of A*, Ant Colony Optimization, and Genetic Algorithm

A* is a heuristic algorithm that can be used for path planning. To reach its goal, it finds the sum between the cost to move to the next node and the cost to move from said node to the target destination. This process continues for the rest of the nodes of the current node's immediate surroundings, after which the neighbor node with the smallest sum is chosen. This process repeats until the target is reached. Though it is one of the most efficient path-planning algorithms in both speed and results, it doesn't always find the shortest path despite the chosen heuristic method and can get stuck on local optima, which Ant Colony Optimization (ACO) is good at avoiding. ACO uses the premise of ants looking for the best route to a food source: the more pheromones on the trail, the more appealing it is to each ant. An ant's desire to go to a new node is the weight of the new node divided by the sum of weights of each node on which it's standing. Once each node's desire is found, a number from zero to one, which is also the sum of each node's desire, is randomly generated, and the range of desire in which that number lands determines which node the ant chooses. As time goes on, the pheromones left on the trail evaporate, leaving the only viable trails being those the ants go to most frequently. ACO has a high global exploration factor because of the aspect of randomness, preventing it from falling into the trap of local optima. The Genetic Algorithm (GA) reaches a target by spawning a population in which each member finds its way to the target node by taking random steps. Those with the shortest paths move into the next generation where 'children' are formed by merging the paths of two 'parents' of the previous generation. Those children are then subject to mutations that change an aspect of their path. The population is then sorted again according to those with the least path steps, and the same process is repeated, eventually resulting in the single best path the algorithm can produce.

1.3 Double-Strand Break and Genetic Algorithm

Double-strand breaks (DSBs) are cuts along a strand of DNA and are generally looked down upon due to their ability to cause Down Syndrome and cancer since the cell ends up lacking the proper amount of chromosomes. Organisms have ways to amend this brokenness by process of break-induced repair (BIR), which uses one of the two ends of the broken DNA to invade a homologous chromosome to use as a template to synthesize a new DNA sequence for itself. Another technique, synthesis-dependent strand annealing (SDSA), works in nearly the same way, but instead of using the DNA sequence of another to complete itself, it instead uses the template to construct a bridge between the two broken ends. BIR and SDSA are both examples of homologous recombination methods- methods that involve using homologous chromosomes as templates in whatever measure for extending a DNA sequence. Liu and Malkova reviewed the intricacies and instabilities of BIR [12]. Non-homologous recombination only entails non-homologous end joining, the ligation of two endpoints on separate segments of DNA.

DSBs are positively offset within meiosis since they are needed to initiate crossovers that result in genetic diversity between parents and their offspring, amongst other functions.

Saito and Colaiácovo classified crossover types and how they operate in *Caenorhabditis elegans* [1], and Pazhayam et al. did a more extensive observation of said crossover types and how they interplay [9]. A few regulatory meiotic methods include crossover assurance in which at least one segment of genetic information is swapped; crossover interference, which spreads the occurrence of crossovers out along the chromosome in question; crossover homeostasis, which keeps a constant number of crossovers per chromosome; crossover invariance (known as Spatial Cluster Model in Pazhayam et al's work) which acts as the negation of crossover invariance due to interference being reduced close to a non-factor; the centromere effect, which prevents crossovers from occurring at the point two sister chromatids are connected (the centromere) and their respective ends (telomere) to prevent detrimental health effects. Alongside the mentioned crossover regulation methods are overarching crossover classes: class 1, which follows the enabling of crossover interference, and class 2, in which crossovers are formed independently from one another.

Damia et al.[2] provided a Probabilistic Adaptive Genetic Algorithm that dynamically adapts the mutation, crossover, and selection mechanisms per iteration. They use a uniform crossover, rank-based chromosome selection, and a mutation expressing minimal changes in a slight proportion of the population. Utami et al. [14] approached their crossovers by using the roulette wheel method to determine where they should be placed among three sections and used arithmetic to enact them. Their selection entails using the better between parent and child for mutations that are randomly chosen using the roulette wheel and are applied using arithmetic. In discovering which crossover operators increase the lifetime of wireless sensor networks, Jaradat et al.[4] used a one-point crossover, two-point crossover, uniform crossover, and flat crossover each individually to which they found minuscule differences between each application. Rikatsih and Mahmudy[11]use a reciprocal exchange for their mutation process, keeping the range of possible mutations bounded by the organism in question and utilizing both an extended intermediate and one-point crossover methods to produce an offspring from two parents. A cross-average crossover operator is proposed by Orong et al. [8] in which the average value between like indexes in different chromosomes is found and used to create the child, using rank-based selection to choose the progenitors. Ning et al.. [7] made use of a chaotic algorithm to optimize the conventional GA to produce an efficient population count. GA was used to maximize traffic management by Tiberio et al. [13]. Each traffic light's fitness value was stored in a vector, the combination providing the least value being the optimal solution for the organization of traffic.

Guo et al. [3] fused A* and ACO to produce an algorithm able to combine global static planning and local dynamic planning such that single target path-finding can be avoided. This saves time from both having to plan a path from the current spot to the closest target at each point until back at the starting position and from having to go through the repetition of returning home after finding a single target. In a related paper, Li et al. [5] developed a priority-based path interference resolution method that has their amphibious robots either stop and wait for the other to the path or locally re-plan the path of the lower-ranked robot to prevent collisions.

Lastly, Zhu et al. [17] made way for an anchor-point-based path-planning algorithm that minds the edges of their robot when creating an optimal path.

2 Methodology

2.1 A*-ACO Pathfinding

The foundation of DGAA* is GA due to its evolutionary nature. The generation size and population size are each based on the number of ants a within each population, which equates to

$$log_2(m) - 1 \tag{1}$$

rounded down, where m are the map dimensions. The generation size is

$$\left[\frac{(a-1)}{2} * 100\right] - 100 \tag{2}$$

where the quotient is rounded down, and the population size is

$$a * 2 - 2. \tag{3}$$

The amount of ants in the population is the ant base, each ant searching the map using the principles of A*'s heuristic. The g(n) is treated the same as in the original A* algorithm, yet to provide a bias toward either exploration or exploration, the h(n) value has been modified to

$$h(n) = [c + u] * d \tag{4}$$

$$c = t_r * t_p \tag{5}$$

$$u = \frac{g_{max} - g_{curr}}{c + t_f * rand(0,1)} \tag{6}$$

$$d = |1 - p| \tag{7}$$

where t_r is the quotient between the number of found targets and the maximum amount of targets, t_p is the h(n) calculation from the original A* method, g_{max}

is the maximum amount of generations, g_{curr} is the current generation, c is map complexity calculated by the product of quotient between the number of barriers and the map's dimensions and the number of targets. In the event that no barriers are within the map, then the map complexity has a value of one. t_f represents how many targets have been found, *rand(0,1)* is a random number between 0 and 1, and p is the pheromone for a particular path spot. An ant becomes more exploratory the bigger the improvised h(n) value, but more exploitative the more minuscule it is.

$$t_r * t_p \tag{8}$$

represents the balance of confidence c for the next node chosen: the more targets the ant found and the closer it is to its current target, the smaller the resulting product, helping it become more exploitative. The product is bigger the fewer targets are found and the larger the distance between the ants, promoting an emphasis on the search's breadth.

When looking for an object for the first time in an unfamiliar environment, there is a factor of uncertainty, u, that is leveraged with the size of the search party and the confidence paradigm previously explained. The search party's initial uncertainty decreases if it frequently returns to the said environment and functionally reduces to 0 eventually. This concept is expressed in the second part of the improvised h(n) equation:

$$\frac{(g_{max} - g_{curr})}{(c + t_f * rand(0,1))} \tag{9}$$

When the search procedure begins for the first time, the number of targets found is, naturally, zero, creating an exploitative bias. Equation 6 is the overarching scale for the global search factor and influences the effect of the target paradigm. The ant is tailored to a global probe, the larger the quotient, yet capitalizes on the best choice to make when the quotient is smaller. The ant is trained to narrow its search effort as the weight of the remaining generations decreases, as seen in the difference between the g_{max} and g_{curr}. Narrowing is further encouraged with the biased map complexity factor. When zero targets have been found, the divisor is only the map complexity, driving the overall uncertainty to be high and the focus to be on global analysis. The uncertainty is compounded with familiarity with a random scale upon the advent of targets found, as seen in

$$c + t_f * rand(0,1) \tag{10}$$

suiting the resulting sum to get larger such that the ant becomes more exploitative in nature.

$$|1 - p| \tag{11}$$

expresses displeasure d for the neighboring path spot. The greater p is, the less displeasure from the ant, and the lesser p is, the more displeasure from the ant.

The sum of confidence and uncertainty may be low for the neighboring spot, but if the pheromones for said step are low, then it is discouraged to choose it.

The explanation of these concepts grants the visual simplification of equation 4

$$(c + u) * d. \tag{12}$$

After the conclusion of each ant's search, their respective paths are graded by length. The path with the least steps is added to the population count, and the pheromones within the map are reset for the next population to prevent restricting ants from other populations from following the same trail as those prior. This multi-colony effort is repeated until the number of paths reaches the population limit.

2.2 Selection

After a generation's population limit has been reached, the best 10 percent of the current population is preserved for the next generation if there are at least 2 chromosomes in that percentage: if not then every path in the current generation gets sent into the next generation due to a lack of best paths within the percentage range.

2.3 Crossover

Two parent paths are used in the crossover, one from the pair being randomly selected to be the dominant parent and the other the recessive parent. The dominant parent is used as the child's path basis for whenever there is not a crossover. The number of crossovers x is determined by

$$\frac{|p_{avg} - p_i|}{10} \tag{13}$$

for any quotient that is more than 0 where p_{avg} is the average path length and $p_i$is the path length of the dominant parent. If the quotient is less than zero then only one crossover occurs.

The number of Double-Strand Breaks (DSBs) is the product of x and 10. The location of the crossovers are found by breaking up the path of the dominant parent into segments with each end being a target point. The length of the segment is divided by the length of the overall path, the sum of each segment adding to one. DSBs are randomly generated from 0 to 1 and land in the range of a segment. To enforce crossover interference using the beam-film model method [12][9], the neighbors of a DSB are reduced by a random number between 0 and 1. If all segments have the same value, then this process repeats until a bias is formed. If a DSB lands on the centromere, the path section with the least length, then it will be randomly applied to one of the neighbors belonging to the centromere up to three times to loosely mimic the centromere effect. Children with affected centromeres will be subject to mutations. Once all DSBs have been made, the segments with the greatest values, up to the number of crossovers dictated, will be chosen to be swapped with that of the second parent.

If the path length of the dominant parent is longer than the average path length then the beam-film model is abandoned to allow crossovers to conceptually occur anywhere along the path of the dominant parent as an effort to make the length of the child shorter than said parent, changing the crossover type from class 1 to class 2. The obligate crossover is always enforced since there is at least one crossover between chromosomes.

2.4 Mutation

Break-Induced Repair (BIR) is used for the mutation process: a DSB will be spawned along some random point of the path. The original path up to that point is preserved, but each target after the point is shuffled before being found again. This increases the risk of an inefficient path yet also introduces the possibility of a better one.

We are using the Unitree Go1 dog to hopefully implement automation for agricultural purposes such as the work done by X. Yang, et al. [15] As we have yet to automate our Unitree Go1 Dog (Go1), we have not drafted an Error Detection and Patching (EDP) system to work alongside DGAA*.

We use a Spectral HD webcam attached to the ceiling and pointed towards the ground to analyze the environment encompassing our Go1, which is done by the combined efforts of YOLO such as the work done by S. Liu et al. [6], Canny Edge Detection, and MiDas Depth Estimation, each denoting obstructions inside the environment. Due to a few inaccuracies each individual method has in realizing obstructions, whether by false classification or negligence, the average obstruction value in each method is found to get a better picture of the Go1's environment. A map constructed upon DGAA* will be employed, guided by coordinates from OptiTrack's Flex 3 motion capture cameras that grant us the ability to use IR nodes mounted on the Go1's back to track its movement. This serves as an analogy for a UGV being guided by GPS coordinates when making its way to a target destination. The Go1 itself will use LiDAR and its native cameras to scope out its environment in a 2D plane to navigate around any forthcoming obstacles. The fusion of LiDAR and cameras allows the Go1 to get a better picture of its environment so that it can make better decisions than would be available if it were only equipped with either or.

3 Results

A simulation has been built to compare A*, ACO, GA, and DGAA*. Since ACO uses multiple ants, a multi-threaded (T) and non-multi-threaded (NT) version of both ACO (ACO-T and ACO-NT, respectively) and DGAA* (DGAA*-T and DGAA*-NT, respectively). This allows the comparing of the performance of ants that search one after another against the performance of ants that search simultaneously. GA uses the one-point crossover and ACO includes pheromone evaporation. Below are conceptual images of how the Unitree Go1 Dogs would traverse their environment given a certain method.

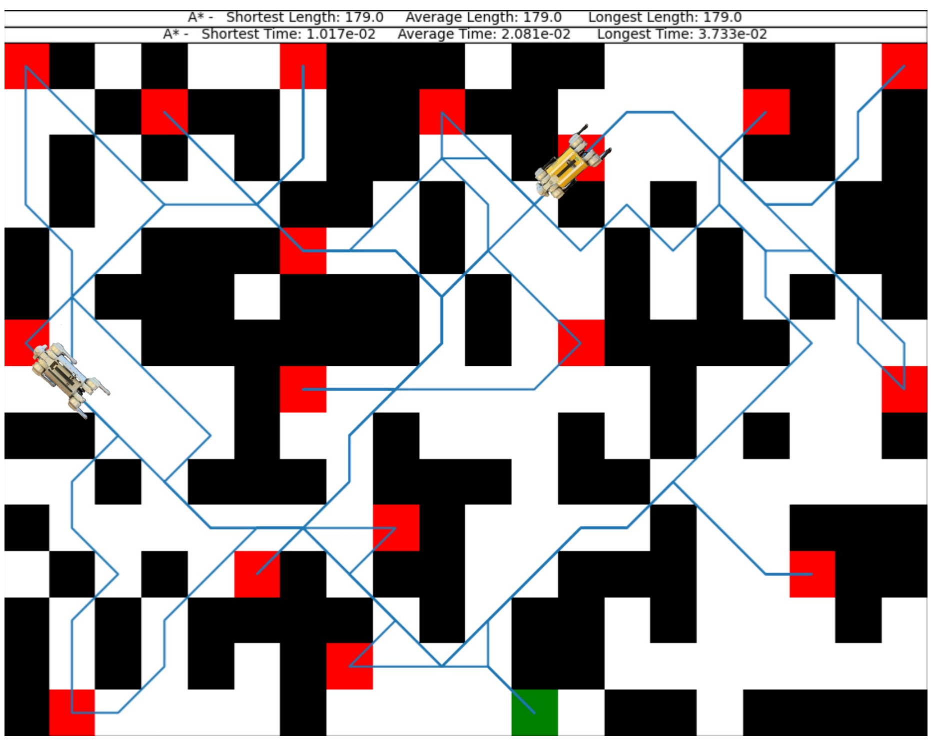

Fig. 1. A* Performance

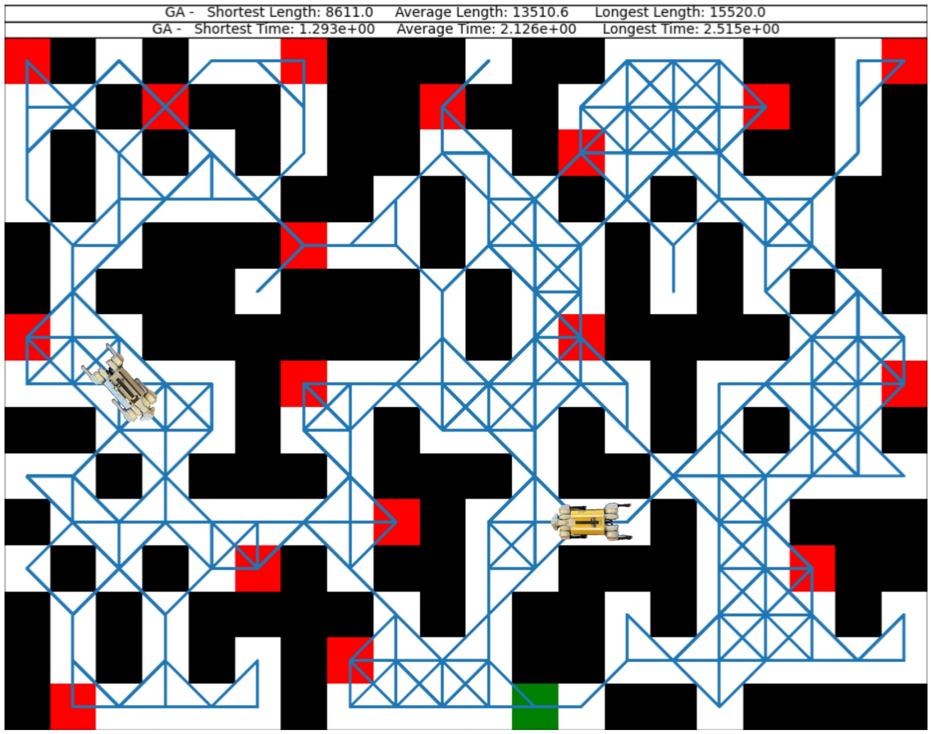

Fig. 2. GA Performance

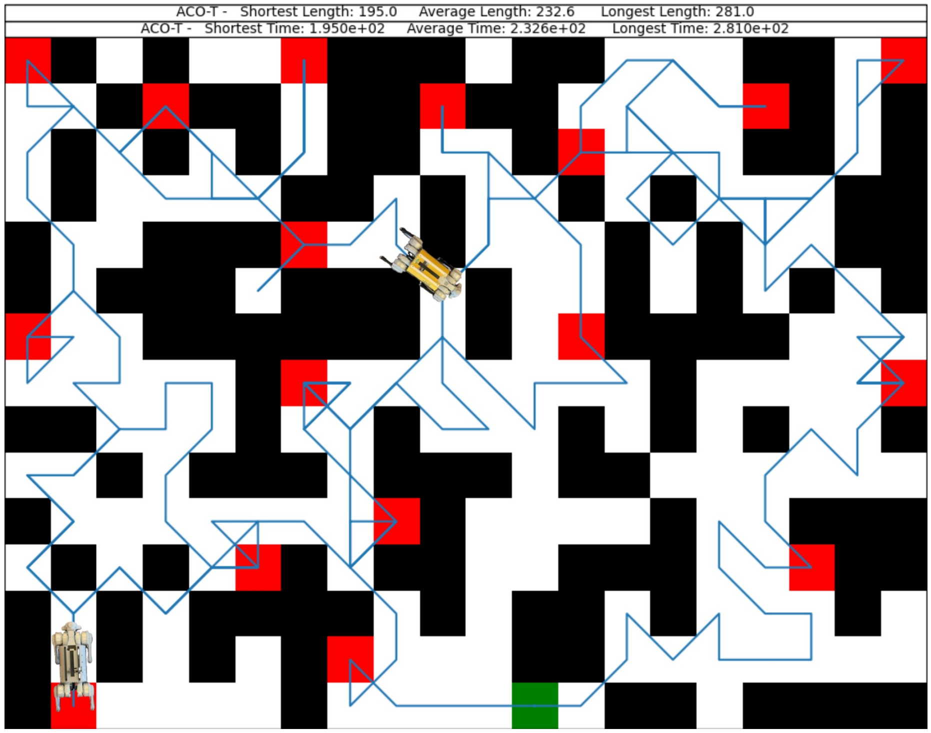

Fig. 3. ACO-T Performance

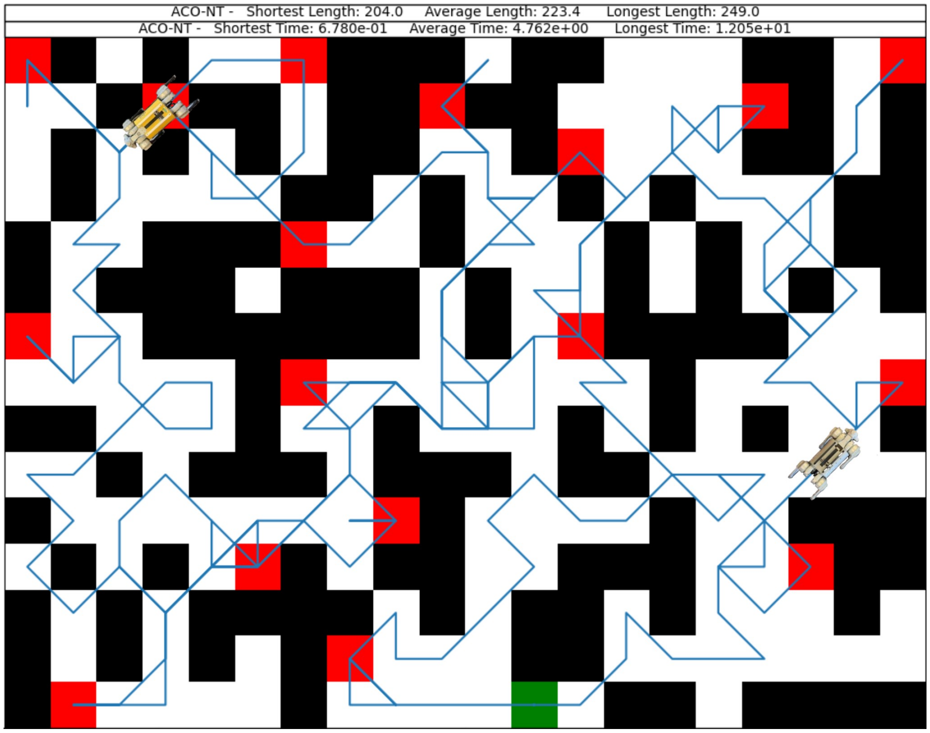

Fig. 4. ACO-NT Performance

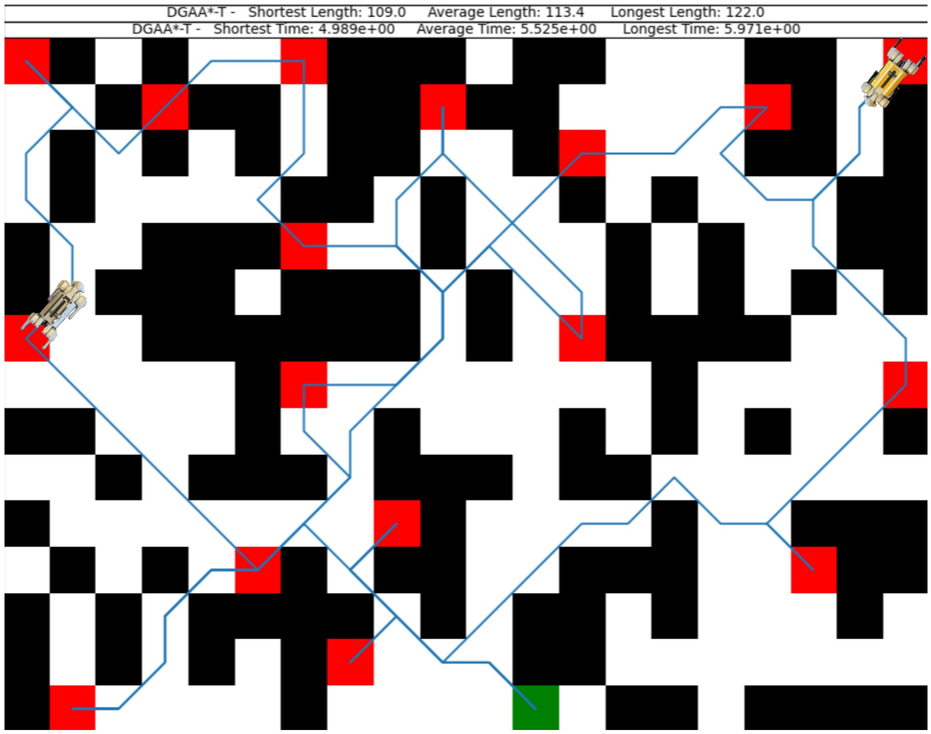

Fig. 5. DGAA*-T Performance

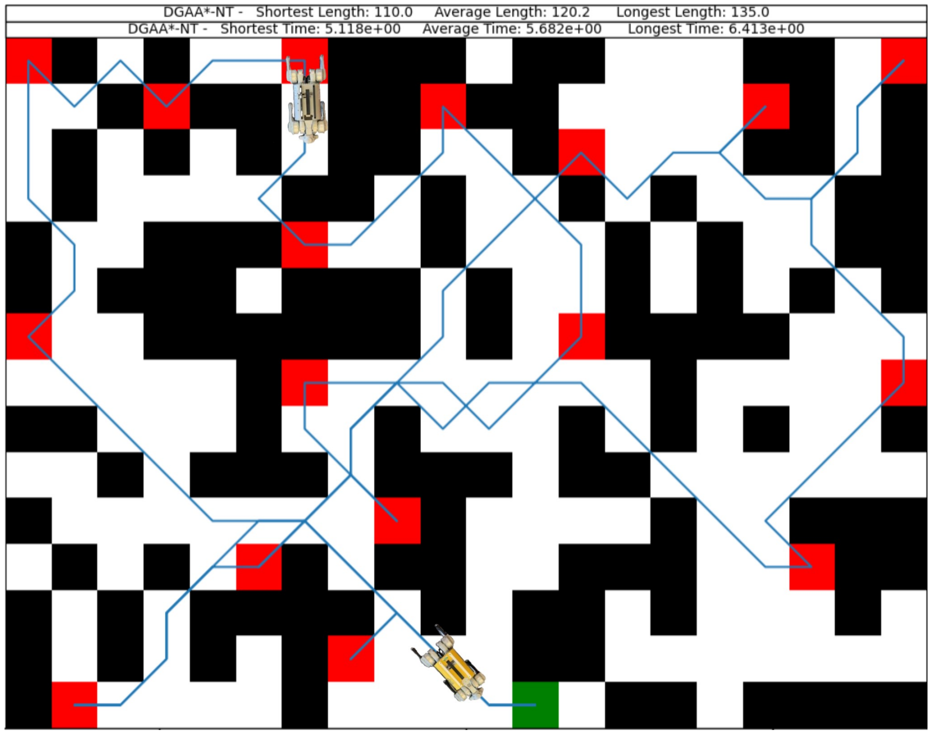

Fig. 6. DGAA*-NT Performance

Below are maps that underwent fifty trials each using each of the aforementioned methods

3.1 Future Studies and Implementations

The overarching objective of this research is to develop a novel multilayered path planning and obstacle avoidance algorithm, DGAA*, that can be used in static and dynamic environments for a Unitree Go1 Dog. This framework integrates the strengths of A*, Ant Colony Optimization (ACO), and Genetic Algorithms (GA) to enhance efficiency while mitigating their respective limitations.

Future work entails the resolution of all errors in the base DGAA* algorithm, the development of the Error Detection and Patching (EDP) mechanism to optimize the gait sequencing of the Unitree Go1 Dog, and a multi-robot interface with a priority system and anchor point system that allows for collaborative robotics environment with minimal conflict. Additionally, we aim to extend this research by integrating Unmanned Aerial Vehicles (UAVs) to assist in guiding a Husky A300 for agricultural applications.

Furthermore, we expect the results from this research to contribute towards multiple domains such as agriculture, search and rescue, security purposes, swarm application, and possible guide dog implementation on the Unitree Go1.

4 Conclusion

In this study, we introduced the foundation of a multilayered path-planning algorithm in DGAA*: A Dynamically Repaired Double-Strand Break Genetic Algorithm with Pheromone Guided A*. By leveraging the strengths of **A***, **Ant Colony Optimization (ACO)**, and **Genetic Algorithms (GA)**, the proposed framework improves path optimization while addressing the limitations of each approach. Additionally, integrating a **redefined gait sequencing technique** reduces mobilization effort and enhances movement stability through error correction.

Our preliminary implementation of a **partially hybridized model** combining **A*** and **ACO** has demonstrated promising results in improving navigation efficiency. These findings suggest that hybrid approaches can significantly enhance path-planning performance in dynamic environments.

Moving forward, further development of the **Error Detection and Patching (EDP) mechanism** will be prioritized to refine gait sequencing for robotic locomotion, particularly in systems such as the **Unitree Go1 Dog**. Additionally, expanding this research to incorporate **Unmanned Aerial Vehicles (UAVs)** for guiding ground-based robots, such as the **Husky A300**, will enable broader applications in autonomous navigation, particularly in agricultural and real-world operational settings.

4.1 Author Contributions

Conceptualization, MH.T. and J.R.; Data Collection, J.R., A.C., R.A., and O.I.; Equations, J.R.; Methodology, J.R., A.C., R.A and O.I.; Software development, J.R., A.C., and O.I.; Formal analysis, J.R., A.C., and MH.T.; Investigation, J.R., A.C., R.A., and O.I.; Resources, MH.T., RC.V. and J.R.; Data curation, J.Z.; Writing–original draft, J.R., and A.C.; Critical Review, MH.T., RC.V., and J.R; Project administration, J.R. and A.C. All authors have read and agreed to the published version of the manuscript.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Malkova A. ALiu L. Break-induced replication: Unraveling each step. In *2022 PubMed Journal, National Library of Medicine*, pages 752–765, 2022.

2. Amir Hossein Damia, Mehdi Esnaashari, and Mohammadreza Parvizimosaed. Adaptive genetic algorithm based on mutation and crossover and selection probabilities. *2021 7th International Conference on Web Research (ICWR)*, pages 86–90, 2021.
3. Jian Guo, Xiaojie Huo, Shuxiang Guo, and Jigang Xu. A path planning method for the spherical amphibious robot based on improved a-star algorithm. In *2021 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1274–1279, 2021.
4. Yousef Jaradat, Mohammad Masoud, Ismael Jannoud, Ahmad Manasrah, and Amer Zerek. Comparison of genetic algorithm crossover operators on wsn lifetime. In *2022 IEEE 2nd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, pages 356–360, 2022.
5. Chenqi Li, Jian Guo, Shuxiang Guo, and Qiang Fu. Study on a two-layer path planning method of spherical multi-robot. In *2023 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 2413–2418, 2023.
6. Shida Liu, Yu Bi, Honghai Ji, and Li Wang. Improved yolov5 algorithm for special people recognition in a new guide robot dog for the blind. In *2024 IEEE 13th Data Driven Control and Learning Systems Conference (DDCLS)*, pages 1585–1590, 2024.
7. Xuefeng Ning, Yuanjia Li, and Zehuai Liu. Improved genetic algorithm-based obstacle avoidance path planning method for inspection robots. In *2023 2nd International Symposium on Control Engineering and Robotics (ISCER)*, pages 346–350, 2023.
8. Markdy Y. Orong, Ariel M. Sison, and Ruji P. Medina. A new crossover mechanism for genetic algorithm with rank-based selection method. In *2018 5th International Conference on Business and Industrial Research (ICBIR)*, pages 83–88, 2018.
9. Sekelsky J. Pazhayam NM, Turcotte CA. Meiotic crossover patterning. In *Front Cell Dev Biol.*, 2021.
10. G. Chen Q. Meng and S. Guo. Research on walking stairs of wheeled-legged robot based on leg structure transformation. In *2022 China Automation Congress (CAC)*, pages 5718–5723, 2022.
11. Nindynar Rikatsih and Wayan Firdaus Mahmudy. Adaptive genetic algorithm based on crossover and mutation method for optimization of poultry feed composition. In *2018 International Conference on Sustainable Information Engineering and Technology (SIET)*, pages 110–114, 2018.
12. Colaiácovo MP. Saito TT. Regulation of crossover fequency and distribution during meiotic recombination. In *2017 Cold Spring Harb Symposia on Quattitative Biology*, 2017.
13. Janos Lance L. Tiberio, Ryan Rhay Vicerra, John Anthony Jose, and Argel Bandala. Density-based optimization approach for coordinated traffic management on a simplified traffic model using genetic algorithms. In *TENCON 2022 - 2022 IEEE Region 10 Conference (TENCON)*, pages 1–8, 2022.
14. Nia Saputri Utami, Achmad Jazidie, and Rusdhianto Effendi Abdul Kadier. Path planning for differential drive mobile robot to avoid static obstacles collision using modified crossover genetic algorithm. In *2019 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, pages 282–287, 2019.
15. J.; Paneru B.; Lin J.; Bist R.B.; Lu G.; Chai L. Yang, X.; Zhang. Precision mon itoring of dead chickens and floor eggs with a robotic machine vision method. In *AgriEngineering 2025*, pages 7–35, 2025.

16. W. Zhang and K. Wang. Wheely: An accessible platform for experimenting with wheeled-legged quadrupedal robots. In *2023 3rd International Conference on Computer, Control and Robotics (ICCCR)*, pages 220–224, 2023.
17. Jianyi Zhu, Xiaojun Wang, Xianyu Qi, Bin Lian, and Yanpeng Dong. Multi-stage local path planning algorithm based on obstacles as anchor points. In *2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE)*, pages 1440–1445, 2024.