Comparative Analysis of GWO and YKSP in Traffic Routing

Abstract - This research paper explains the comparative differences between the GWO algorithm and the Yen-K shortest path algorithm, notably comparing the methodologies of each algorithm. This study analyzes the algorithms computational complexity, memory consumption, time difference, and efficiency as GPS traffic routers. Our results concluded that GWO outperformed Yen–K shortest path in efficiency and in finding the optimal path, however its large memory consumption leads it to being unsuitable for lightweight applications.

I. INTRODUCTION

In modern intelligent transportation systems, the most crucial aspects of their systems are efficient pathfinding/planning and the inclusion of traffic routing algorithms so that travel time can be optimized, congestion is minimized, and the overall navigation experience is enhanced [18]. As urban areas grow and become more incredibly complex in their road designs and traffic situations, the traditional routing methods are being reevaluated in favor of more adaptive and dynamic routing solutions [16], [19]. This paper covers the study done between two prevalent algorithms in GPS-based traffic routing: Grey Wolf Optimizer (GWO) and Yen K's Shortest Path (YKSP) algorithm. GWO is a nature-based metaheuristic algorithm that falls under the swarm intelligence group due to it taking inspiration from the hunting behaviors and dynamics of grey wolves [1], [2]. This novel approach offers a new dynamic way to create efficient routes and path planning through population-based exploration. YKSP, on the other hand, is a classical graph-based pathfinding algorithm extension of Dijkstra's algorithm [12], [13]. YKSP operates by systematically computing the multiple shortest loopless paths between two points in a graph [12]. This paper investigates the computational efficiency of GWO and YKSP algorithms by comparing their results through time and space complexity of their applications when used in traffic routing situations. Through systematic testing and analysis, we aim to showcase the strengths and weaknesses of GWO and YKSP algorithms. Through these highlights, we can then show potential applications of this approach and offer any insights into which method is better suited for future optimization and navigation systems.

II. RELATED WORKS

Smart City Traffic Data Analysis and Prediction Based on Weighted K-means Clustering (2024)
Significant progress has been made in optimizing traffic routing, as shown by research of Lei Li [9]. This research analyzed traffic patterns in urban cities such as Chengdu, by understanding the traffic in 5 parts of the city. In this study, it improved the conventional K-means method by creating a new traffic flow prediction algorithm that can more accurately predict city traffic flow. Taking the traditional K-means algorithm, they changed its value weights, plus combined it with Holt's exponential smoothing algorithm. As a result, the weighted K-means method was able to accurately identify the patterns of the traffic congestions in Chengdu's 5 urban regions, and with the help of Holt's smoothing algorithm, made successful prediction performance

Vehicle Routing Optimization System with Smart Geopositioning Updates (2021) -This research investigated geopositioning updates and their impact on the efficiency of optimization algorithms, by updating the distance matrix [10]. These measures are critical for optimizing the VRP (Vehicle Routing Problem). This research was conducted by developing an optimization system in which updates are carried out in integration with both an open-source routing machine and GPS tracking services. As it comes to the dynamically changing list of destinations, continuous updates were required. The first step of the research was to generate temporary values of the distance matrix based on the correction of the quasi-Euclidean distance, followed by investigating the update progress on the proposed optimization algorithms. Based on these results, it was compared to manually obtained results. The research concluded that updating data should start from the smallest values in the distance matrix.

Real-time Deep Reinforcement Learning based Vehicle (2020)-Routing and Navigation This research paper introduced a proposed Deep Reinforcement Learning (DRL) method to build a real-time intelligent vehicle routing and navigation system by formulating the task as a sequence of decisions [11]. Additionally, an integrated framework was developed to facilitate intelligent vehicle navigation research by embedding smart agents into the SUMO simulator. This research was conducted by simulating 9 realistic traffic scenarios in order to test the proposed navigation method. The results concluded efficient convergence of the vehicle navigation agents and their effectiveness in making optimal decisions under volatile traffic conditions. In addition, the results also demonstrated a better navigation solution compared to the benchmark routing optimization algorithms

III. METHODOLOGY

Methodology will include multiple parts such as the algorithm section where we discuss and introduce both Grey Wolf Optimizer and Yen K’s. Then we enter the implementation section of how we utilize these algorithms in our code. And then the final section will be centered around the explanation of how our code draws and stores paths and routes for the GPS.

Grey wolf Optimizer

Grey Wolf Optimizer (GWO) is a metaheuristic optimization algorithm and was created by Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis in 2014 [1]. Specifically, GWO is a swarm intelligence-based algorithm that is structured from the hunting behaviors of the Grey Wolves. GWO is a population –based algorithm that simulates the leadership hierarchy of wolves into four groups (Alpha, Beta, Delta, and Omega) and sorts them accordingly based on effectiveness and updates them accordingly to any change or problem that may occur. The Alpha is considered the dominant wolf, followed by Beta as subordinate wolf, then delta and the end Omega as the scapegoat of the pack, also the least important wolf [2], [7]. The main phases of the grey wolf hunting mechanism are tracking, encircling, and attacking.

Encircling the prey

The t represents the current iteration, A and C are coefficients vectors, X is the position vector of the prey, and X indicates the position vector of a grey wolf. In this part of the process for n-dimensional space, two points (wolves) are considered and the position of the first is updated based on the position of the second.

$$\mathbf{X(t+1) = X(t)-A.D}$$

X(t+1) is the next location of the wolf, and the X(t) is the current location of the wolf. A is a coefficient matrix and D is a vector that depends on the location of the prey X(p). It is calculated as such.

$$\mathbf{D = |C.X_p (t)-X(t)|}$$

C is the coefficient matrix that is then calculated by this equation:

$$\mathbf{C=2 . r2}$$

R2 is a random generation of vector ∈ [0,1]. Both solutions can relocate around another solution. Since these equations use vectors, they can be applied to any dimension. Up to this point everything has represented the steps and speed of the wolves; to define their values we use this equation:

$$\mathbf{A=2a.r1-a}$$

a*a*’s vector value linearly decreased from 2 to 0. R1*r1* is a randomly generated vector from the interval [0,1]. Now the parameter must be updated:

$$\mathbf{a=2-t\ (2T)}$$

t is the current iteration, and *T* is the maximum number of iterations.[7]

Hunting

Each iteration updates their position in accordance with the position s alpha, beta, delta, and omega.

$$\mathbf{X(t+1) = 1/3(X1 + X2 + X3)}$$

X1, X2, and X3 are the wolves that are being calculated with the following equation.

$$\mathbf{X1 = X\propto (t)-A1 \cdot D\alpha}$$
$$\mathbf{X2 = X\propto (t)-A1 \cdot D\alpha}$$
$$\mathbf{X3 = X\propto (t)-A1 \cdot D\alpha}$$

Next is to calculate the distance which is done with following equation [1].

$$\mathbf{D\alpha = |C1 \cdot X\alpha - X|}$$
$$\mathbf{D\beta = |C2 \cdot X\beta - X|}$$
$$\mathbf{D\beta = |C3 \cdot X\delta - X|}$$

Exploitation

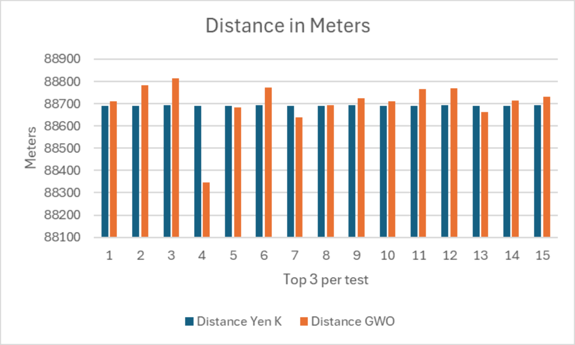
Once the prey finally surrenders the grey wolf attacks the prey, this translates to a mathematical model that decreases the value of a to A as a random value in the interval [-2a, 2a], which decreases from 2 to 0 over the course of iterations [1].

Yen-k Path

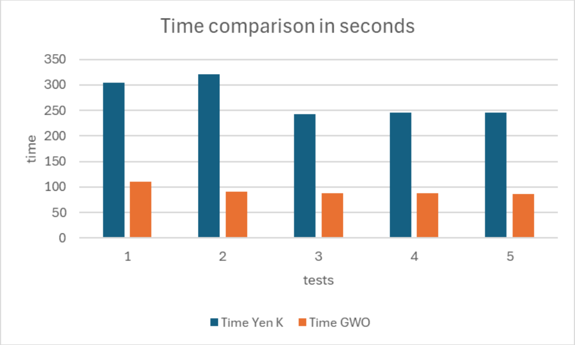
The yen-k path algorithm computes k-shortest loopless path between 2 nodes, as K is the number of shortest paths to compute [12]. This algorithm behaves similarly to Dijkstra shortest path algorithm [15], as it also builds path based of the shortest distance. This algorithm is limited to the number of nodes in the source –target paths [13], [14].

IV. RESULTS AND DISCUSSION

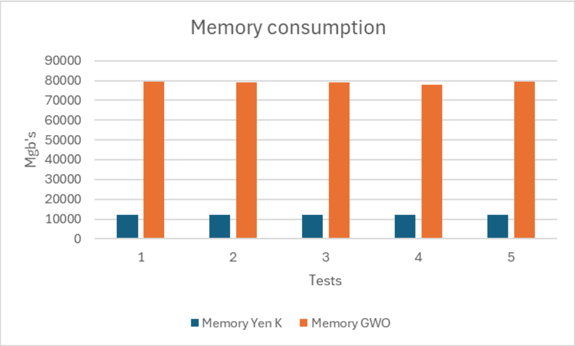
FINAL: Through fifty tests of both Yen-k's shortest path and the grey wolf optimizer, the optimality of the pathing was found to be the same, meaning that both algorithms came to find the same optimal path. The difference between the algorithms was in their speed of calculation and the amount of memory needed to complete the calculations. Grey wolf optimizer trades space for time in its approach as it takes vastly more memory than Yen-k's algorithm to calculate the optimal path. However, this makes it much faster than Yen-k [18].



Yen-k comparative to GWO algorithm consumes more time when you calculate equal number of k paths to the number of wolves and iterations, making it less efficient. The Yen –k algorithm time is approximately 3 times higher rate than GWO, the striking difference is due to GWO's ability to test multiple paths efficiently, while Yen K's is only computing a single path which requires more accuracy and time to calculate [20].



The bar graph below demonstrates the memory usage between GWO and Yen-k algorithms, displaying that GWO consumes approximately 8 times more storage than Yen-k because it uses more iterations. It's a factor of the number of wolves times the number of iterations is the reason for the high number of consumptions, while yen-k memory is determined solely number of k-paths was a high priority, yen-k has a better yield [6].



V. CONCLUSION

Looking at both algorithms overall there are pros and cons to both. While GWO may be fast, its large memory consumption leads to it being unusable for lightweight [2], [5] applications and on hardware that doesn't have the memory requirements met to run the program. This issue leads to YKSP being much more universal in its application. The longer run time of YKSP is an issue when running on sufficient hardware but the time tradeoff is much easier to accept when running the program on hardware that lacks memory [19]. So, in determining which algorithm is better overall it's not exactly black and white. Currently Dijkstra's algorithm is the most prevalent in the GPS pathfinding space [15] but because of its speed it may be worth it to companies in the industry to consider switching to GWO. YKSP in our application is built on Dijkstra's algorithm however, so implementation may be more familiar to those who have worked with these programs previously [15], [17]. Overall, both algorithms have their own strengths and weaknesses but it's worth further researching in the future to determine if these algorithms could replace our currently implemented solutions and improve upon them in one or more aspects [18].

REFERENCES

[1] S. M. M. Seyedali Mirjalili, Andrew Lewis, "Grey Wolf Optimizer, Advances in Engineering Software," *Science Direct,* vol. 69, no. Advances in Engineering Software, pp. 46-61, 2014, doi: https://doi.org/10.1016/j.advengsoft.2013.12.007.

[2] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: a review of recent variants and applications," *Neural Computing and Applications,* vol. 30, no. 2, pp. 413-435, 2018, doi: 10.1007/s00521-017-3272-5.

[3] W. Long, S. Cai, J. Jiao, M. Xu, and T. Wu, "A new hybrid algorithm based on grey wolf optimizer and cuckoo search for parameter extraction of solar photovoltaic models," *Energy Conversion and Management,* vol. 203, p. 112243, 2020.

[4] S. Kassaymeh, M. Alweshah, M. A. Al-Betar, A. I. Hammouri, and M. A. Al-Ma'aitah, "Software effort estimation modeling and fully connected artificial neural network optimization using soft computing techniques," *Cluster Computing,* vol. 27, no. 1, pp. 737-760, 2024.

[5] A. Saxena, R. Kumar, and S. Mirjalili, "A harmonic estimator design with evolutionary operators equipped grey wolf optimizer," *Expert Systems with Applications,* vol. 145, p. 113125, 2020.

[6] S. N. Makhadmeh *et al.*, "Recent advances in Grey Wolf Optimizer, its versions and applications," *Ieee Access,* vol. 12, pp. 22991-23028, 2023.

[7] S. M. Almufti, H. B. Ahmad, R. B. Marqas, and R. R. Asaad, "Grey wolf optimizer: Overview, modifications and applications," *International Research Journal of Science, Technology, Education, and Management,* vol. 1, no. 1, pp. 1-1, 2021.

[8] B. Tu, F. Wang, Y. Huo, and X. Wang, "A hybrid algorithm of grey wolf optimizer and harris hawks optimization for solving global optimization problems with improved convergence performance," *Scientific Reports,* vol. 13, no. 1, 2023, doi: 10.1038/s41598-023-49754-2.

[9] Li, Lei. "Smart City Traffic Data Analysis and Prediction Based on Weighted K-Means Clustering Algorithm." *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 6, 2024, pp. 162–171. www.ijacsa.thesai.org.

[10] Belka, Radosław, and Mateusz Godlewski. "Vehicle Routing Optimization System with Smart Geopositioning Updates." *Applied Sciences*, vol. 11, no. 22, 2021, p. 10933. MDPI, https://doi.org/10.3390/app112210933.

[11] Koh, Songsang, et al. "Real-Time Deep Reinforcement Learning Based Vehicle Routing and Navigation." *Applied Soft Computing*, vol. 96, 2020, p. 106694. Elsevier, https://doi.org/10.1016/j.asoc.2020.106694.

[12] J. Y. Yen, "Finding the K shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712-716, 1971. doi: 10.1287/mnsc.17.11.712

[13] D. Pascoal and J. Santos, "An efficient implementation of Yen's ranking loopless paths algorithm," *4OR*, vol. 10, no. 2, pp. 195-219, June 2012. doi: 10.1007/s10288-011-0181-5

[14] X. Qin, X. Yang, and Y. Yang, "An improved Yen's algorithm for finding K-shortest loopless paths in a network," *Procedia Computer Science*, vol. 131, pp. 1201-1207, 2018.

[15] W. Dijkstra and R. W. Floyd, "Shortest Path Algorithms," in *Handbook of Graph Theory*, 2nd ed., J. L. Gross, J. Yellen, and P. Zhang, Eds. Boca Raton, FL, USA: CRC Press, 2013, pp. 124-127.

[16] S. Chen, L. Xu, and Z. Li, "Dynamic K-shortest paths routing algorithm for urban traffic networks," *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 4, pp. 7529-7539, 2021.

[17] A. F. T. Winata, D. D. Puspita, and S. Yuwono, "K-shortest path algorithm for real-time dynamic traffic conditions," *Procedia Computer Science*, vol. 197, pp. 407-414, 2022.

[18] F. L. M. Costa, R. F. S. Teixeira, and J. F. Rodrigues, "Performance analysis of pathfinding algorithms for traffic routing in smart cities," *Sensors*, vol. 20, no. 15, 2020.

[19] M. A. O. Khalifa and S. A. Mahmoud, "Multi-criteria traffic-aware K-shortest path routing using Yen’s algorithm," *Egyptian Informatics Journal*, vol. 24, no. 2, pp. 205-215, 2023.

[20] R. Zhang and P. Wang, "Traffic incident-aware routing optimization based on modified K-shortest paths," *Transportation Research Part C: Emerging Technologies*, vol. 156, 2024.