Optimization Solutions for Solving Travelling Salesman Problem in Graph Theory using African Buffalo Mechanism

Yousef Methkal Abd Algani
The Faculty of Mathematics, The Arab Academic College for Education in Israel, Israel

***Abstract*—The African Buffalo Optimization (ABO), a metaheuristic optimization algorithm created from thorough study of African buffalos, a species of African cows, in African woods and savannahs, is suggested in this study. In its pursuit for food across the African continent, this animal demonstrates unusual intelligence, sophisticated organising capabilities, and remarkable navigational acumen. The African Buffalo Optimization creates a mathematical model based on this animal's behaviour and uses it to tackle several benchmark symmetrical Travel Salesman's Problem and six tough asymmetric Travelling Salesman Problem Library (TSPLIB) instances. Buffalos can ensure the effective exploitation and exploration of the problem space by frequent contact, teamwork, and a sharp mind of previous record discoveries, as well as tapping into the breed's collective exploits, according to this study. The results produced by solving these TSP problems using the ABO were compared to those obtained by utilizing other prominent methods. The results indicate that ABO gently outperformed than Lin-Kernighan and HBMO optimising solutions to the ATSP cases under investigative process, with a slightly higher accuracy of 99.5% compared to 87% for Lin-Kernighan and 80% for HBMO. The African Buffalo Optimization algorithm produces very competitive outcomes.**

Keywords*—*African buffalo optimization; solutions; travelling salesman's problem; graph theory

I. Introduction

Profit maximisation and expense reduction never been more essential in social history than they are today. As a result of this need, efficiency has become a popular research topic. As a result of this advancement, a variety of optimization methods were developed. Ant Colony Optimization, Particle Swarm Optimization, Artificial Bee Colony, Genetic Algorithm, and many other techniques are among the most prominent [1]. However, the foregoing methods have a number of disadvantages, including a premature convergent delay in getting the results, the ability to become caught in global minimum, and a complex fitness function with a lot of factors to set up[2]. The construction of the African Buffalo was motivated by a desire to offer remedies to some of these algorithms' flaws.

The ABO is a community stochastic optimization method inspired by the behaviour and attitude of African buffalos, a genus of wild cows comparable to domestic cows that traverse thousands of kilometres through African tropical rainforests and scrubland by travelling together and in large herds of up to a hundred buffalos [3]. Their travel is motivated by a desire to find abundant grazing pastures. They tend to follow the wet weather to find abundant grazing meadows. Because the seasons vary from place to place across Africa's huge expanse, buffalos are constantly on the move in search of their preferred meadows. ABO algorithm is focused to analyse how buffalos use different modes of communication to organise themselves [4]. They have uses various modes of sounds to indicate the danger zone, good and bad area of grazing fields and to encourage their herds to stay and take use of the present resources.

Over the last few decades, there has been a lot of research on the symmetric Traveling Salesman's Problem. It's astonishing how little research has been done on the asymmetrical Travelling Salesman's Problems [5]. This is perplexing because the majority of real-world uses are asymmetric. A postal service official searching for the optimum route to make deliveries to various locations within a specific area is an ideal example. For instance, a bus driver attempting to find the best way to pick up children from schools, delivering meals to residents, and other real-life scenarios illustrate its practical application. Asymmetric solutions are the most probable approach to resolving these challenges [6]. This concept explains this need for research project since it will aim to solve everyday difficulties and, as a result, will have broad applicability. TSP computations are commonly done in the literature using Eucledian or orthodromic distances. The Euclediaa distance is determined using x-y dimensions and the Pythagorean formula, and the range among x and y is clearly symmetrical. But at the other end, the orthodromic range estimates the area between the coordinates of two nodes over a portion of the globe [7]. Though more exact than Euclidean measurements, particularly over vast distances, it does provide genuine real-life measurements when nodes are connected by highways or transport systems, as do unbalanced computations that account for one-way mobility and some other construction related issues [8].

A research gap in the field of optimization solutions for solving the Traveling Salesman Problem (TSP) in Graph Theory is the development of hybrid or integrated approaches that combine multiple optimization techniques. While numerous algorithms have been proposed individually, there is a lack of comprehensive studies that explore the synergistic potential of integrating different methods. By combining the strengths of various algorithms such as metaheuristics, exact algorithms, and machine learning, it may be possible to achieve improved solution quality, faster convergence, and enhanced scalability for solving TSP. Investigating the effectiveness of hybrid approaches and identifying the optimal combinations of techniques for different TSP instances would be a valuable research direction to bridge this gap and advance the field of TSP optimization. The research on the African Buffalo Optimization (ABO) algorithm is of significant importance as it introduces a novel metaheuristic approach inspired by the behavior of African buffalos. By studying and emulating their intelligence, organizational capabilities, and navigational acumen, the ABO algorithm provides a fresh perspective on solving optimization problems. The algorithm's ability to effectively exploit and explore the problem space through teamwork, communication, and knowledge sharing among individuals mirrors the collective intelligence of the buffalo breed. The study demonstrates the algorithm's competitive outcomes by solving benchmark symmetrical Traveling Salesman's Problems and challenging asymmetric TSPLIB instances. By showcasing the potential of bio-inspired optimization, this research contributes to the development of innovative algorithms and expands the range of effective tools available for solving complex optimization problems.

The African Buffalo Optimization and the Traveling Salesman's Problem are introduced in Section II of this work. A Proposed ABO algorithm is presented in Section III. The study's results and discussion are summarised in Section IV.

II. Related Works

The African Buffalo Optimization (ABO), Ant colony optimization (ACO), LinKernighan algorithm and the hybrid Honey Bee Mating Optimization (HBMO-TSP) are the subjects of this research. The truth that all these techniques have some of the greatest outcomes and compare the effects in the research piques our interest in them.

Buffalo optimization approached with various alarms, [7] In its hunting tasks, the newly built African Buffalo Optimization imitate the warning and alarm sounds of African buffalos. The waaa sound alerts buffalos to the existence of attackers or a paucity of pastures, prompting the herds to move on to safety or more profitable regions of the grassland. Once this cry is issued, the animals are instructed to remain vigilant and find a secure or better grazed field. The maaa sounds, on the other side, are being used to advise buffalos to relax because there are plenty of grazing pastures nearby and the environment is suitable to grazing. The herds are able to utilize their seek for food sources using these signals. This ABO is being the solution of various optimised problems for both Symmetric and Asymmetric Travelling Salesman Problems and the numerical benchmark functions. ABO seems that it is old algorithm and weak in process, issues in delay and inefficiency, this optimisation is only helpful for African buffalos.

Ant Colony Optimisation is developed by [9]. This is among the most often used optimization techniques. After some preliminary work on Dorigo and Gambardella's Ant Colony System and Marc Dorigo's Ant System, Marco Dorigo and Di Caro developed the Ant Colony Optimization method in 1999. The spontaneous wandering of ants in looking for food prompted the Ant Colony Optimization. Once a food supply has been found, the ant that identified it transports a particle of it back to the nest; likely take a shorter path and continuously depositing pheromones as a way of reminding additional ants of its progress. Neighborhood ants will probably join the succeeding ant in tracking the food resources if they detect the pheromone's fragrance. Once such ants reach the food resources, they take some food bits back to the home and deposit pheromones to improve the original ant's route. This mechanism attracts other ants by increasing the pheromone content on the preferred shortest route. The swarm of ants is on the most efficient route into and out of the protein source in a short amount of time. The main defects of ACO algorithm is fuel cost minimization, less voltage profile, loss in transmission reduction and this computational algorithm is reduced to find the original paths in graphs.

The Lin-Kernighan algorithm is derived by [7] a common search method that finds solutions by carrying out extensive local searches]. This method searches employing transfers (or moves) that turn one route into another in order to shorten the route duration until the greatest (cheapest) trip is found. The basic rules of the iconic Lin-Kernighan algorithm are as follows: k Only successive transactions with huge progress are allowed; (ii) 'Shuttered' tours are permitted except when k = 2; (iii) A earlier cracked link should not be provided, and a precipitate link should not be cracked; and (iv) Journeys are only registered through the five closest neighbours; (v) If k = 4, no connection, xi, on the route should be destroyed whether it is a common factor of a tiny proportion of remedy routes; (vi) If future attempts do not provide a good outcome, the research for enhancements is ended. Principles (iii) and (iv) are heuristics principles that may decrease the time but it may produce in a degraded approach, as shown. The improved LinKernighan method has enhanced most of the core rules as a result of this: k Choose a different nodes l; (ii) Choose another network l so that k-l is an applicant side and á (k-l) = 0 and k-l relates to the actual best circuit; otherwise, choose l so that k-l is a determined based or choose l from endpoints not yet chosen. Proceed to stage ii if any locations will not be addressed; (iv) Whenever the network size picked in step iii >1, the sequence of the networks chosen includes the initial route. This algorithm provides results randomly because the algorithm begins with random partition.

The Honey Bees Mating Optimization method is a community metaheuristic, simulates the beehive queen's mating rituals. The princess of the bee’s couples with the swarms joining her in flying, dependent on her strength and ability of the queen retains the vehicle's genetics in her spermathecal after pairing, and the swarm is formed when the breeding flight is completed. Part of the answer is the genetics of the drones. A transformation function is utilised to construct the nests. When a progeny is stronger than the queen, this becomes the empress. The leader, labourers, and flyers are the three kinds of bees. The endeavour golden jelly to the offspring, while the flies pair with the monarch and the emperor is the healthiest bee. This is the HBMO algorithm analysed by [10]. Honey Bee Breeding Optimization to Broadening Nearby Search Algorithm (ENST) and Multi-

Phase Nearby Lookup Randomly assigned Flexible Described (HBMOTSP) is a hybrid that combines HBMO with the Broadening Community Search Method (ENST) and the Multi-Phase Nearby Lookup Randomly assigned Dynamic Mapped (MPNS-GRASP). This combination of processes aids HBMO-TSP in reducing calculation time and ensuring effectiveness when addressing Traveling Salesman's Issues. This method, which is based on the Honey Bee Mating Optimizer, has proven to be particularly useful to overcome multimodal optimization issues. This optimisation is mainly based on the matting strategy of bees and the problem solving is very competitive with other state.

III. Proposed ABO Algorithm

ABO was made in an attempt to address problems with certain existing techniques, such as stimulated annealing, Particle Swarm Optimization, Genetic Algorithm and Ant Colony Optimization, such as medium speed, rate of convergence and use of different factors, complex fitness features. The ABO is essentially a replica of the warning 'maaa' contact calls that the herds use to coordinate troops to stay put and utilize a specific grazing field, as well as the warning 'waaa' contact calls that organize the entire species to demand a new or secure grazing position. From this alarm sounds Herds were able to optimise their searches in order to reach food-rich areas [11]. Mathematical representations for these various alarms are shown in Eq. (1) and Eq. (2).

$$M_{x+1} = N_x + St1(pq_{max} - U_x) + St2(pq_{max.x} - U_x) \quad (1)$$

Where, M_x is the "maaa" alarm of buffalo with reference x. $(x = 1,2, \dots \dots .. x)$, pq_{max} is the better buffalo in group. $pq_{max.x}$ is the good place found in buffalo herds. $St1, St2$ Are the parameters for learning. M_{x+1} denotes the movement of buffalo from their current position mk to a new location, and it indicates the migration's enormous storage size, Lifestyle. The real behaviour of the animals can be adjusted be accomplished in accordance with Eq. (2).

$$U_{x+1} = \frac{(U_x + N_x)}{\theta} \quad (2)$$

Where, U_{x+1} represents new parameters, U_x is the alert for "waaa" sound and θ is time intervals with respect to the moments of buffalo.

The African buffalos have used these vocalisations to organise together as they navigate through the African woods in pursuit of rich grassy fields to feed their enormous appetites. Each organism's position indicates a solution in the search area in this method. Fig. 1 shows the implementation chart of ABO algorithm. The ABO algorithms are maintaining this structure on every step, it monitors each buffalo's dynamical position as it approaches the pq_{max}. Based on how far the emphasis is applied at a given iteration, x and pq_{max} are used. The training settings have an impact on every animal's efficiency.

A. *Working Procedure of ABO*

The ABO begins by allocating herds to subnetworks in the solution space. The mammals can visit any unpopulated node that is nearest and cheaper to them based on probability. The price of the movement, as calculated by the accessible heuristics in the previous move, influences this decision[12]. The cost heuristics for such manoeuvres, the personal profit of the transfer to the animal as judged by its past experience, and the actual benefit of the specific move to the grassy field all impact further movements [13]. This is expressed by the equation by in a subsequent section. The buffalo's fitness is then updated by the program. In this method, the method finds the best buffalo's (pq_{max}) location in the herds in respect to the best option. The personal record (pq_{max}) of each individual is also measured. Its parameters are remembered by the buffalos. If a buffalo's strength and conditioning value is higher than$(pq_{\max})$, the method records it as the animal's ideal position, bgmax. Likewise, if a buffalo's present performance is greater than any other in its memories, the system records it as the animal's greatest (pq_{max}).If the pq_{max} matches the exit condition at this point, the algorithm ends and returns the greatest buffalo's position matrix as the perfect result [8]. Or else, it moves on to the next cycle and resumes the procedure till the exiting requirements are met.

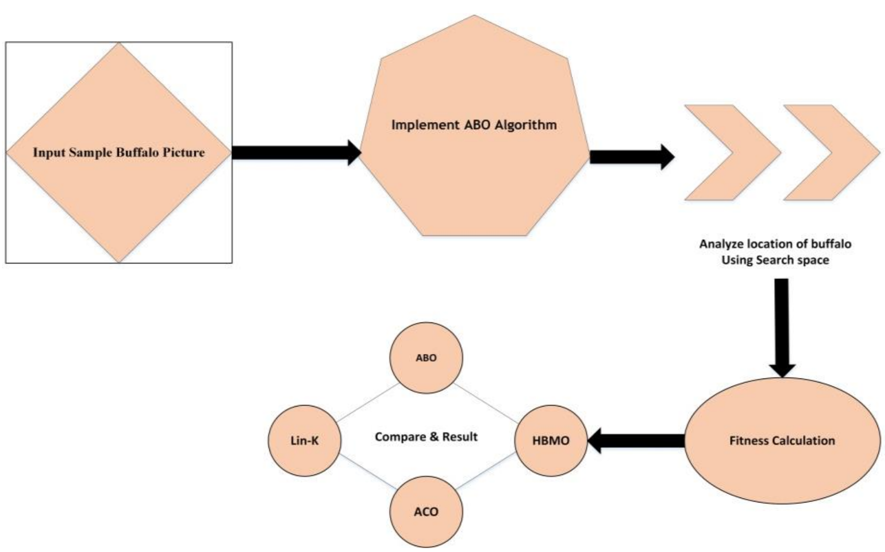

Fig. 1. Implementation of ABO.

Step by step working procedure of African Buffalo (ABO) algorithm is given below:

Step 1-Initialization: Define the population size, maximum number of iterations, and other algorithmic parameters. Generate an initial population of buffalo individuals with random positions.

Step 2-Fitness Evaluation: Evaluate the fitness of each buffalo individual by calculating the objective function value for the given optimization problem. Assign a fitness score to each buffalo based on their objective function value.

Step 3-Selection: Select buffalo individuals from the population based on their fitness scores. The selection process can be based on various strategies like roulette wheel selection, tournament selection, or rank-based selection.

Step 4-Movement and Communication: Simulate the movement and communication behavior of African buffalos. Determine the new positions of the buffalo individuals by considering their current positions, velocities, and movement rules.

Step 5- Local Search: Apply a local search operator to explore the neighborhood of the buffalo individuals' positions. This step helps improve the quality of solutions by making small adjustments to the positions.

Step 6-Update Personal and Global Best: Update the personal best positions and fitness scores for each buffalo based on the newly obtained solutions. Keep track of the global best position and fitness score achieved by any buffalo individual in the population.

Step 7-Termination Condition: Check if a termination condition is met, such as reaching the maximum number of iterations or achieving a satisfactory solution. If the termination condition is met, proceed to the next step. Otherwise, go back to step 3.

Step 8-Output: Once the termination condition is met, output the best solution found, which corresponds to the global best position obtained during the algorithm's execution. The solution represents the optimized solution for the given optimization problem.

ABO Algorithm

Function $P(n) = (P1, P2, \dots \dots .Pn)^k$

Place buffaloes in solution path

Input fitness values used Eq. (1)

$N_{.x} + 1 = N_{.x} + Stx_1(pq_{max.x} - U_x) + Stx_2(pq_{max.x} - U_x))$

Where, $N_{.x}$ and U_x is exploitation and exploration of x buffalo $(x = 1,2, \dots \dots \dots ..)$, $pq_{max.x}$ is the herds fitness.

Location Update $pq_{max.x}$ and $pg_{max.x}$

$N_{x+1} = \theta(M_{.x} + N_{.x})$, where θ *is unit time*

If $pq_{max.x}$ provides Yes, then proceed.

If $pq_{max.x}$ provides No, then back with initial step

Stopping procedure not performs, and then starts with fitness step

Get Output.

B. *Steps to Solve TSP using ABO*

The ABO has the benefit of solving difficult optimization issues like the TSP with very basic stages. The basic steps for solving the problem are as continues to follow:

- Initial step to migrate buffalos shown in Eq. (3)

$$f_{xy} = \frac{U^{st1}xy^{st2}xy}{\sum_{j=1}^{k} u^{st1}xyz^{st2}xy}, \text{ Here } xy = \pm 0.10 \quad (3)$$

- Use Eq. (1) and Eq. (2) and update fitness
- Determine $pq_{max.x}$ and max.
- Using value of heuristic and add non- visited cities of buffalos
- If pq_{max} Updated, Yes then proceed
- If pq_{max} Updated, No then back to initial stage
- If reach with exit parameters, Yes then go for output
- If reach with exit parameters, No then back to fitness
- Get best results (output).

Where, Stx_1 and Stx_1 are the parameters values 0.5 and 0.3 respectively. xy takes alternate values on +0.10 and -0.10 iterations. Z is the reinforcement.

Fig. 2 shows the work flow of ABO Algorithm The positive reinforcement warning invitation z informs the creatures to stop and utilize the surroundings because there are ample meadows, whereas the negative reinforcement warning u urges them to continue exploring the area because the current place is not profitable [14]. The possibility St of an animal n migrating from town j to town k is based on a combination of two possible values: the perceived benefits of the transition, as calculated by certain heuristic denoting the attack's prior appeal, and the evaluation advantage of the relocation to the breed, denoting how effective have been in the sense of making that specific travel. The lowest values suggest whether or not that move is desirable [15].

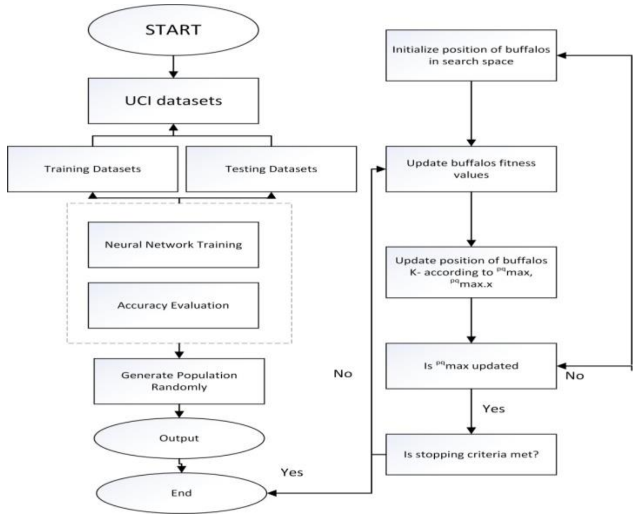

Fig. 2. Flowchart of ABO algorithm.

IV. Results and Discussion

The ABO was applied to a set of asymmetric TSP (ATSP) datasets and three sets of symmetric TSP datasets and from TSPLIB95 ranges from 50 to 14470 locations in this research. The first experimental compared ABO's effectiveness in TSP situations to data from a recent survey that included St70, Pr76, Ch150, Eil76, KroA100, Eil101, Berlin52and Tsp225. The following set of studies compared ABO's effectiveness to that of Att48, Rd400, St70, Eil76, Gil262, Pr152, Brd14051, D1291, Pr1002 and Fn14461 in a recent study [16].

The final experiment looked at how well ABO performed in asymmetric TSP scenarios. The findings of the fourth set of trials were compared to those achieved utilising some common Artificial Neural Networks approaches [17]. The following are the variables for the PSO-related phases of testing: 300 people; 2000 iterations (G_{max}); 0.95 inertia weights, $T_1: 2, T_2: 1$ rand $1(0,1)$, and rand2 $(0,1), (0,1)$. The following are the HPSACO research factors: Pheromone parameter (): 1.0; heuristic element (): 2.0; condensation rate (): 0.05; pheromone quantity: 100; people: 300; iterations (G_{max}): 2000; inertia gravity:(0.85); $T_1: 2, T_2: 1$; insects (N): 200; pheromones factor $(\propto)$:2.0; therapeutic factor (β):3.0; condensation variable (ρ): 0.05. The studies were run on a 3.40 GHz Intel Duo Core i7-3770 CPU with 6 GB RAM and MATLAB [18]. The tests on the asymmetrical Traveling Salesman's Problems were run on a desktop or laptop with a 4 Gb Of ram and Pentium Duo Core 1.80 Ghz cpu. Similarly, the ANN experiments were conducted out with MS Virtual C++ 2009 on an Intel Duo Core i6 CPU [9]. We first did research on eight TSP examples to verify the ABO algorithm's accuracy in handling the TSP. Error rate is calculated using Eq. 4 from the average value referred through fitness factor of each ABO algorithm.

$$Error\ Rate = \frac{(Average\ Value - Best\ Value)}{Best\ Value} \times 100 \quad (4)$$

In all of the tests that were conducted, ABO outscored the other methods. For example, the ABO found the best solution to Eil76 and Berlin52. No other technique came close. In addition, as comparing to other method, the ABO found the closest-optimal response to the remainder TSP instances. The ABO continues to have the optimum value of average outcomes obtained from each method. It's odd that the Hybrid Algorithm (HA), it utilizes a storage matrix comparable to the ABO, really cannot come up with a decent result [15]. Because the HA is a mixture of the ACO and the CGAS, this can be traced back to the employment of multiple factors[2].The ABO's superiority may also be demonstrated in the usage of computer capabilities in which the ABO is easily the quickest of the four methods.

Table I shows the comparative results of various optimisation technique involves in TSP. The ABO, for example, is 58,885 times better than the ACO, 1,435 times higher than the ABC, and 30,412 programs are to improve than the Hybrid Mechanism in Berlin52 (HA). This pattern can be found in all of the instances under inquiry [19]. Lin-Kernighan algorithm took 0.283 seconds to complete all of the TSP issues above, compared to ACO's 7356 seconds, ABC's 44.11 seconds, and HA's 3288.27 seconds. But since ABO uses the route methodology, as opposed to the delayed path fabrication used by the ACO, the ABO's speed can be traced back to good storage management strategies. Fig. 3 shows the Error rate graph, which is typically shows how the error rate changes as the problem size increases. It helps in evaluating the efficiency and effectiveness of different algorithms in solving TSP instances of varying complexities. A lower error rate indicates that the algorithm is capable of finding solutions closer to the optimal solution, while a higher error rate suggests larger deviations from optimality. The graph can compare multiple algorithms or optimization techniques, allowing for a comparative analysis of their performance. It can highlight which algorithms perform better in terms of minimizing the error rate and provide insights into their scalability and suitability for different TSP instances. By analyzing the error rate graph, researchers and practitioners can make informed decisions about the choice of algorithm and the potential trade-offs between solution quality and computational efficiency when solving TSP optimization problems.

TABLE I. PERFORMANCE COMPARISON OF ABO, LIN-KERNIGHAN AND HBMO

TSP Samples	Higher Value	African Buffalo Optimization			Lin-Kernighan Algorithm			Honeybee Mating Optimization		
		Error rate	Plus rate (%)	Processor Time (secs)	Error rate	Plus rate (%)	Processor Time (secs)	Error rate	Plus rate (%)	Processor Time (secs)
RL11849	11554	0.1	99	0.05	1.2	97	1.5	1.15	98	2.25
HK48	87534	0.02	98	0.77	0.5	92	2.4	2.5	94	3.5
BRG180	3412	0.5	99	0.03	1.3	94	2.7	1.22	91	1.85
FL1400	1654	0.09	94	0.56	1.5	91	1.45	1.45	88	4.5
VM1748	78632	0.66	100	0.98	0.78	93	2.5	2.5	84	3
GR120	54392	0.432	99	1.2	0.67	98	2.5	1.15	92	3.5
RL1889	23895	0.225	100	1.5	1.3	95	2.75	0.45	89	4.5
U1432	65782	0.09	96	0.876	1.15	91	1.85	1.34	95	2.75
PR152	47651	0.12	98	0.34	0.75	94	2.5	0.5	92	3
FL417	89675	0.5	100	1.4	1.25	97	2.25	1.5	98	3.5

Fig. 4 plus rate for solving the Traveling Salesman Problem (TSP) in optimization problems represents the percentage of instances or problem instances for which a specific algorithm or optimization technique achieves a solution that is better than or equal to a certain threshold. It provides insights into the algorithm's success rate in finding solutions that meet a desired level of optimality. The plus rate graph allows for a comparative analysis of different algorithms or optimization techniques. It shows how well each algorithm performs in terms of meeting a certain level of optimality for various TSP instances. A higher plus rate indicates that the algorithm consistently achieves solutions that are equal to or better than the specified threshold. By examining the plus rate graph, researchers and practitioners can assess the algorithm's effectiveness in providing high-quality solutions for TSP optimization problems. It helps in evaluating the algorithm's reliability, robustness, and suitability for different problem instances. Fig. 5 shows the CPU rate graph for solving the Traveling Salesman Problem (TSP) in optimization problems represents the computational efficiency or runtime performance of different algorithms or optimization techniques. The CPU rate graph shows how the computational time changes as the problem size increases. It helps in evaluating the efficiency and scalability of different algorithms in solving TSP instances of varying complexities. A lower CPU rate indicates faster computation and better efficiency, while a higher CPU rate suggests longer computational time.

The difference in speed between the ABC and the ABO is related to the CGAS utilisation of multiple parameters. The mix of path creation and path enhancement approaches, as well as the utilisation of numerous factors, could have had an impact on the HA's performance [20].

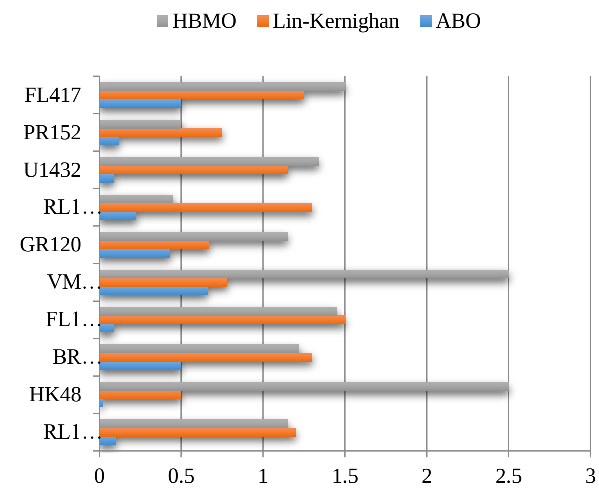

Fig. 3. Error rate comparison of ABO, lin-kernighan and HBMO.

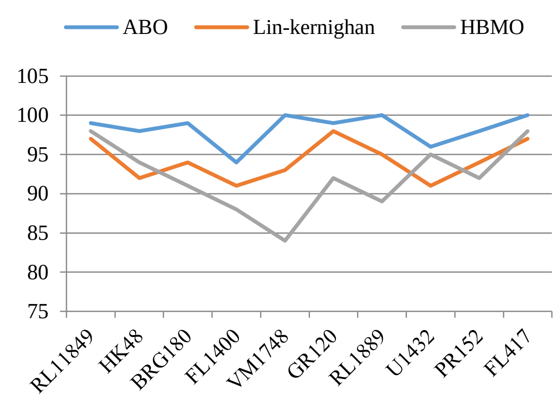

Fig. 4. Plus rate (%) comparison of ABO, lin-kernighan and HBMO.

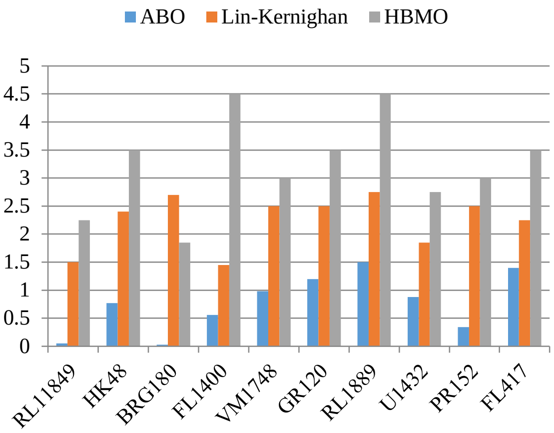

Fig. 5. CPU rate (Secs) comparison of ABO, lin-kernighan and HBMO.

V. Conclusion

Decision has made from the following comparison of ABO, Lin-Kernighan and HBMO algorithm, ABO provides better results on both error and plus rate even CPU rate also. Asymmetric Traveling Salesman's Problems cases have optimum solution. The results indicate that ABO gently outperformed than Lin-Kernighan and HBMO optimising solutions to the ATSP cases under investigative process, with a slightly higher accuracy of 99.5 percent compared to 87 percent for Lin-Kernighan and 80 percent for HBMO. However, HBMO was able to find the best solution in 12 of the 20 examples studied, while the ABO was able to find the best option in all cases and came close in the others. Unfortunately, when it comes to the quickness with which findings must be obtained, the ABO is the best algorithm. Because reliability and efficiency are two of the three main methods for evaluating a superior algorithm the ABO can be recognised a better algorithm than other optimization techniques.

In future work, the optimization solutions for solving the Traveling Salesman Problem (TSP) in Graph Theory using the African Buffalo mechanism can be extended in several directions. There is a scope for refining and enhancing the African Buffalo Optimization (ABO) algorithm by exploring different variations and incorporating additional features inspired by the behavior of African buffalos. The scalability of the ABO algorithm can be investigated to handle larger TSP instances with thousands or even millions of cities. The ABO algorithm can be extended to address variations of the TSP, such as the dynamic TSP where the cities and their distances change over time. Adapting the ABO algorithm to handle dynamic scenarios would enable its application in dynamic routing problems where the optimal route needs to be continuously updated. Analyzing the algorithm's performance guarantees, complexity bounds, and robustness to problem variations will contribute to establishing a solid theoretical foundation for the ABO algorithm.

References

[1] J. B. Odili and M. M. Kahar, “African buffalo optimization approach to the design of PID controller in automatic voltage regulator system,” in National Conference for Postgraduate Research, Universiti Malaysia Pahang, 2016, pp. 641–648.

[2] M. Azrag, T. A. Kadir, J. Odili, and M. Essam, “A global african buffalo optimization,” Int J Softw Eng Comput Syst, vol. 3, no. 3, pp. 138–145, 2017.

[3] J. B. Odili, M. N. M. Kahar, and A. Noraziah, “African buffalo optimization algorithm for tuning parameters of a pid controller in automatic voltage regulators,” J. PLoS ONE, vol. 12, no. 4, 2017.

[4] J. B. Odili, M. N. MOHMAD KAHAR, A. Noraziah, and E. A. Odili, “AFRICAN BUFFALO OPTIMIZATION AND THE RANDOMIZED INSERTION ALGORITHM FOR THE ASYMMETRIC TRAVELLING SALESMAN’S PROBLEMS.,” J. Theor. Appl. Inf. Technol., vol. 87, no. 3, 2016.

[5] J. B. Odili, M. N. M. Kahar, and S. Anwar, “African buffalo optimization: a swarm-intelligence technique,” Procedia Comput. Sci., vol. 76, pp. 443–448, 2015.

[6] J. B. Odili, M. N. M. Kahar, and A. Noraziah, “Convergence analysis of the African buffalo optimization algorithm,” Int. J. Simul. Syst. Sci. Technol., vol. 17, no. 44, pp. 44–41, 2016.

[7] J. B. Odili, M. N. Kahar, and A. Noraziah, “Solving Traveling Salesman’s Problem Using African Buffalo Optimization, Honey Bee Mating Optimization & Lin-Kerninghan Algorithms,” World Appl. Sci. J., vol. 34, no. 7, pp. 911–916, 2016.

[8] J. Odili, M. N. M. Kahar, S. Anwar, and M. Ali, “Tutorials on African buffalo optimization for solving the travelling salesman problem,” Int. J. Softw. Eng. Comput. Syst., vol. 3, no. 3, pp. 120–128, 2017.

[9] N. M. Al Salami, “Ant colony optimization algorithm,” UbiCC J., vol. 4, no. 3, pp. 823–826, 2009.

[10] R. Kumar, D. Sharma, and A. Kumar, “A new hybrid multi-agent-based particle swarm optimisation technique,” Int. J. Bio-Inspired Comput., vol. 1, no. 4, pp. 259–269, 2009.

[11] J. B. Odili and M. N. Mohmad Kahar, “Solving the Traveling Salesman’s Problem Using the African Buffalo Optimization,” Comput. Intell. Neurosci., vol. 2016, p. e1510256, Jan. 2016, doi: 10.1155/2016/1510256.

[12] P. Singh, N. K. Meena, A. Slowik, and S. K. Bishnoi, “Modified african buffalo optimization for strategic integration of battery energy storage in distribution networks,” IEEE Access, vol. 8, pp. 14289–14301, 2020.

[13] N. Nedjah, L. D. M. Mourelle, and R. G. Morais, “Inspiration-wise swarm intelligence meta-heuristics for continuous optimisation: a survey - part II,” Int. J. Bio-Inspired Comput., vol. 16, no. 4, pp. 195–212, Jan. 2020, doi: 10.1504/IJBIC.2020.112340.

[14] J. B. Odili, M. N. M. Kahar, and A. Noraziah, “Convergence analysis of the African buffalo optimization algorithm,” Int. J. Simul. Syst. Sci. Technol., vol. 17, no. 44, pp. 44–41, 2016.

[15] J. B. Odili and J. O. Fatokun, “The mathematical model, implementation and the parameter-tuning of the African buffalo optimization algorithm,” in 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), IEEE, 2020, pp. 1–8.

[16] A. C. Cinar, S. Korkmaz, and M. S. Kiran, “A discrete tree-seed algorithm for solving symmetric traveling salesman problem,” Eng. Sci. Technol. Int. J., vol. 23, no. 4, pp. 879–890, Aug. 2020, doi: 10.1016/j.jestch.2019.11.005.

[17] R. Durbin and D. Willshaw, “An analogue approach to the travelling salesman problem using an elastic net method,” Nature, vol. 326, no. 6114, pp. 689–691, 1987.

[18] I. Fares, A. E. Hassanien, R. M. Rizk-Allah, R. M. Farouk, and H. M. Abo-donia, “Solving capacitated vehicle routing problem with route optimisation based on equilibrium optimiser algorithm,” Int. J. Comput. Sci. Math., vol. 17, no. 1, pp. 13–27, Jan. 2023, doi: 10.1504/IJCSM.2023.130420.

[19] D. Karaboga and others, “An idea based on honey bee swarm for numerical optimization,” Technical report-tr06, Erciyes university, engineering faculty, computer …, 2005.

[20] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” IEEE Comput. Intell. Mag., vol. 1, no. 4, pp. 28–39, 2006.