World Applied Sciences Journal 34 (7): 911-916, 2016
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2016.34.7.329

Solving Traveling Salesman's Problem Using African Buffalo Optimization, Honey Bee Mating Optimization & Lin-Kerninghan Algorithms

[1]Julius Beneoluchi Odili, [1]Mohammad Nizam Kahar and [1, 2]A. Noraziah

[1]Faculty of Computer Systems & Software Engineering,
Universiti Malaysia Pahang, Kuantan 26300, Malaysia
[2]IBM Centre of Excellence, Universiti Malaysia Pahang, Kuantan 26300, Malaysia

Abstract: This paper compares the performances of the African Buffalo Optimization (ABO), hybrid Honey Bee Mating Optimization (HBMO) and the Lin-Kernighan (LKH) algorithms for solving the problems of the Symmetric Travelling Salesman's Problems. The three techniques have been applied successfully to solve the popular problem of an anonymous travelling salesman who is searching for the most optimized route to visiting all his customers in different locations of a large city or in a number of cities. This study focusses on these three methods with the aim of ascertaining the most efficient and effective. Results obtained from using these algorithms to solve the benchmark dataset on TSP available in TSPLIB95 serve as the comparative data. The outcome of this experiment shows that the newly-developed African Buffalo Optimization has very encouraging performance in terms of capacity to obtain optimal or near-optimal results consistently and in the most cost-effective manner.

Key words: African Buffalo Optimization · Honey Bee Mating Optimization · Lin-Kernighan algorithm · Travelling salesman's problem

INTRODUCTION

Over the years, human beings have drawn inspiration from nature to solve complex problems in Science, Engineering, Technology and Mathematics. This has particularly been very potent in research studies concerned with obtaining minimization or maximization of operational costs in order to obtain better yields. Nature has inspired several heuristic algorithms that have yielded very incredible results. Some of the algorithms derived from nature are: Genetic Algorithm [1], Particle Swarm Optimization [2], hybrid Honey Bee Mating Optimization [3] Lin-Kernighan algorithm [4] African Buffalo Optimization [5, 6] , amongst many others.

Thrilled by the exceptional performance of these three algorithms that hold some of the best results in literature in solving the Travelling Salesman's Problem, this study specifically compares their performances in terms of their capacity to arrive at optimal or near-optimal solutions, speed and consistency in achieving results.

The organization of this paper is as follows: the first section highlights the motivation of this study and introduces the Travelling Salesman's Problem; the second section discusses the three algorithms, namely, the African Buffalo Optimization, Honey Bee Mating Optimization and the Lin-Kernighan algorithm. This is followed by the experiment and discussion of results. Finally, the fourth section is concerned with the conclusions, the fifth with acknowledgement of support for the study and then the references.

Travelling Salesman's Problem: The history of the Travelling Salesman's Problem dates back to 1930. It is believed to be one of the most studied combinatorial optimization problems [7]. The Travelling Salesman's Problem is a minimization problem of a particular salesman who has customers scattered in different locations within a large city or in a set of cities. His assignment is to visit each of his customers' locations, using the shortest possible route and return to the starting city/node or

Corresponding Author: Julius Beneoluchi Odili, Faculty of Computer Systems & Software Engineering, Universiti Malaysia Pahang, Kuantan 26300, Malaysia.

location. This problem can be represented by a complete weighted graph where G= V, E. Here V is a set of n nodes/vertices/customer locations and E represents the fully interconnected edges (routes) to the different nodes in the graph G. Each edge is weighted by a given weight d_{bc}, that is, distance between nodes 'b' and 'c.' The routes of the Travelling Salesman could be symmetric (in which case the distances between city 'b' to be city 'c' is the same as 'c' to 'b') in all edges or asymmetric where distances between city 'b' to 'c' is not the same as city 'c' to 'b' in at least in one edge. This could be as a result of one-way routes or other civil engineering, commercial or administrative considerations. However, the scope of this paper is on symmetric routes. Mathematically, the minimization equation for the Travelling Salesman's Problem is: given n cities and their x-y coordinates, find an integer permutation $\pi = (C1, C2, C3,.....Cn)$ with Cn being the city 'n', our task is to minimize the sum of the cities [8].

$$f(\pi) = \sum_{i=1}^{n-1} d(ci+1) + d(Cn, C1) \quad (1)$$

where $d(Ci, Ci+1)$ = the distance between city 'i' and city i + 1; $d(Cn, C1)$ = the distance of city n' and city '1'.

The Three Algorithms: In this study, our concern is the African Buffalo Optimization (ABO), the hybrid Honey Bee Mating Optimization (HBMO-TSP) and the Lin-Kernighan algorithm. Our interest in these algorithms is borne out of the fact that they have some of the best results in literature.

African Buffalo Optimization: The newly-designed African Buffalo Optimization (A.B.O) [5] simulates the alert ('maaa') and alarm ('waaa') calls of African buffalos in its foraging assignments. The waaa sound is used to warn the buffalos of the presence predators or the lack of pastures and, therefore, urging the herd to move on to safer or more rewarding areas of the search (exploration). Whenever this call is made, the animals are asked to be alert and to seek a safer or better grazing field. The maaa calls, on the other hand, is used to encourage the buffalos to be relaxed as there are good grazing fields around and the atmosphere is conducive for grazing (exploitation). With these sounds, the buffalos are able to optimize their search for food source. The ABO is a population-based algorithm in which individual buffalos work together to solve a given problem. The parameter setting of ABO are population = 50; lp1 = 0.6; lp2 = 0.5 and ?? = 01-1.

1. Initialization: randomly place buffalos to nodes at the solution space;
2. Update the buffalos exploitation using Equation (2)
3.

$$m'_k = m_k + lp1(bg - w_k) + lp2(bp.k - w_k) \quad (2)$$

where m_k and w_k represents the exploitation and exploration moves respectively of the k[th] buffalo (k=1,2,..........N) ; $lp1$ and $lp2$ are learning factors;

bg is the herd's best fitness and bp , the individual buffalo's best location

4. Update the location of buffalos using (3):

$$w_k' = \frac{(w_k + m_k)}{\lambda} \quad (3)$$

Fig. 1: ABO algorithm

Using the waaa (move on) signal or the maaa (hang around) signal, the animals are able to obtain amazing solutions in their exploration and exploitation of the search space. The ABO algorithm is shown in Figure 1.

In Figure 1, the $w._k$ represents the waaa (explore) signals of the buffalos with particular reference to buffalo k; $m._k$ is the maaa (stay to exploit); $m._k$+1 represents requests for further exploitation; lpr1 & lpr2 are learning factors; $w._k$ +1 is the request for exploration. The ABO models the democratic nature of the buffalos (refer to Eq.2) highlighting their exceptional memory capacity (m. k + 1: this being an indication that the buffalos are aware they have left the previous location, m. k); the buffalo's famed cooperative abilities represented by their constant communication (lp1(bgmax–w.k) and their intelligence (lp2(bpmax,k–w.k), denoted by their awareness of their individual best exploit since the search began).

So far, the African Buffalo Optimization has been used to solve a number of optimization problems such as the benchmark numerical functions [9], symmetric Travelling Salesman Problems [10], asymmetric Travelling Salesman's Problems [11, 12].

Hybrid Honey Bee Mating Optimization: The Honey Bees Mating Optimization algorithm [3], which is a population-based metaheuristic, is the simulation of the mating behavior of the queen of the hive. The queen of the bees, based on her energy and speed, mates when in flight with the drones accompanying her (the queen).

After mating, the queen stores the genotype of the drone in her spermathecal and the brood is made when the mating flight is concluded. The genotype of the drones represents part of the solution. To create the broods, a crossover operator is used. If a brood is fitter than the queen, that brood becomes the queen. The bees are divided into three: queen, workers and drones. The workers maintain the brood feeding them with royal jelly; the drones mates with the queen and the queen is the fittest bee. This is the exploitation part of this algorithm.

The hybrid the Honey Bees Mating Optimization (HBMOTSP) is simply a combination of Honey Bee Mating Optimization with the Expanding Neighborhood Search Technique (ENST) [13] and the Multi-Phase Neighboring Search-Greedy Randomized Adaptive Search Procedure (MPNS-GRASP) [14, 15]. This merger of different procedures helps HBMO-TSP to reduce the computational time and ensure efficiency in solving the Travelling Salesman's Problems. This algorithm which is a modification of the classical Honey Bee Mating Optimization algorithm has proved very effective in solving combinatorial optimization problems [16]. The HBMO-TSP pseudo-code is presented in Figure 2.

Lin-Kernighan Algorithm: The Lin-Kernighan algorithm is one of the most popular search algorithms that uses extensive local search to obtain solutions [4]. This algorithm conducts its search using exchanges (or moves) that converts one tour into another with a view to reducing the tour length until the best (cheapest) tour is reached. The basic rules of the classical Lin-Kernighan algorithm are: (i) Only sequential exchanges with positive gains are allowed; (ii) 'Closed' tours can be allowed except in cases where *i = 2*; (iii) A previously broken link must not be added and a previously added link must not be broken, (iv) Tours are only entered via the five nearest neighbor; (v) In a case where *i = 4*, no link, *xi*, on the tour must be broken if it is a common link of a small number (ii-iv) of solution tours; (vi) The search for improvements is stopped if subsequent efforts don't yield a better result.

As can be observed, rules (iv) and (v) are heuristic rules that could save time but may result in a compromised solution. Based on this, the modified Lin-Kernighan algorithm has refined some of the basic rules thus: (i) Select a random node j; (ii) Then select another node k such that j-k is a candidate edge and á (j-k) = 0 and j-k belongs to the current best tour; else select k so that j-k is a candidate solution or select k among nodes not yet selected (iii) Let j=k. If some nodes have not been covered, return to step ii; (iv) When nodes chosen in step ii >1, the order of chosen nodes makes up the initial tour.

```
Initialization
Create the initial population of the bees using MPNS-GRASP
Select the fittest bee as the queen.
Select the maximum number of mating flights (N)
Main Phase
do while i ≤ N
Initialize queen's spermatheca, energy and speed.
Select α
 While energy > a given threshold and spermatheca is not full, do

(i)Select a drone
(ii)if the drone passes the probabilistic condition then add sperm of the drone in the spermatheca
Endif
Speed(t + 1) = α * Speed(t)
energy(t + 1) = α* energy(t)
End do.
do j = 1, Size of Spermatheca
Select a sperm from the spermatheca
Generate a brood by applying a crossover operator between the queen, the selected drones and the adaptive memory
Choose  a worker randomly,
Use the selected worker to improve the brood's fitness (ENS strategy).
If the brood's fitness is greater  than the queen's fitness then replace the queen with the brood
else
if the brood's fitness is better than one of the drone's fitness then
Replace the drone with the brood
end if
end if
end do
end do
return The Queen which is the Best Solution Found
```

Fig. 2: HMBO-TSP pseudo-code

i. Generate a random initial tour T
ii. Let i=1, choose t_1
iii. Choose $x_1 = (t_1, t_2) \in T$
iv. Choose $y_1 = (t_2, t_3)$ ⊚ T such that $G_1>0$. If this is not possible, go to step 12
v. Let i = i+1
vi. Choose $x_i = (t_{2i-1}, t_{2i})$ such that
  (a) if t_{2i} is joined to t_i, the resulting configuration is a tour T' and
  (b) (b) $x_i = y_s$ for all s < i . If T' is a better tour than T, let T = T' and go to step 2
vii. Choose $y_i = (t_{2i}, t_{2i+1}) \in T$
  such that (a) $G_1>0$
  (b) $y_i \neq x_s$ for all s < i, and
  (c) x_{i+1} exists. If such y_i exists, go to step 5.
viii. If there is an untried alternative for y_2, let i=2 and go to step 7
ix. If there is an untried alternative for x_2, let i=2 and go to step 6
x. If there is an untried alternative for y_1, let i=1 and go to step 4
xi. If there is an untried alternative for x_1, let i=1 and go to step 3
xii. If there is an untried alternative for t_1, and go to step 2
xiii. Stop (or go to step 1)

Fig. 3: Lin-Kernighan algorithm

RESULTS AND DISCUSSIONS

The comparison of the performance of these algorithms is based on their ability to obtain consistent results over time (Success Rate), deviation from the optimal solution (Relative Error) and use of computational resources as measured by the time taken to obtain best results (Computational time). The choice of comparative data was informed by the fact that the chosen instances from the TSPLIB95 datasets are among the most difficult for any algorithm [17]. HBMO-TSP data were obtained from [3], modified Lin-Kernighan's from [4] and ABO from direct experiment. The experiments were carried out using a desktop computer running windows 7, 64-bit, Intel Core [18], i7-3770 CPU@ 3.4GHz, 3.4GHz, 4 GB RAM. The program is coded in MATLAB programming language and ran on MATLAB 2012b Compiler. The experimental parameters for the ABO are population: 40; initial m.k:1.0; initial bgmax/bpmax:0.6; lp1/lp2:0.5; w.k:1.0. The Lin-Kernighan uses k=0; $\Pi^0 = 0$; w =-?; t ≙ size of node; length= n/2. For the HBMO, the parameters are Queen:1; Drones:200; Spermatecal:50; Mating drones:50; Brood:50; α :0.9; Mating flights:1000; w1:3; w2:4; □: $10^{-10.}$ The comparative results are presented in Table 1.

Table shows clearly the superior performance of the ABO over the other algorithms. First, in comparing the rate of successful runs, ABO obtained 100% success in all test instances. This cannot be said of the Lin-Kernighan algorithm that scored as low as 10% success in Fl400 and 40% in VM1748 and RL1889. Similarly, Lin-Kernighan could only obtain 80 and 88% success in RL1304 and FL1304 respectively. Data on the success rate of HBMO are not available in literature. The relative error was obtained with the formula

$$\text{Re}l.\ Error \frac{Best - Opt\ Values}{Opt\ Values} X10 \quad (3)$$

In terms of capacity to obtain optimal or near-optimal results, the three algorithms performed creditably well. The cumulative relative error of the ABO is 2.79% to Lin-Kernighan's 0.739% for the 16 out of the 17 test cases under investigation and the HBMO-TSP's 0.51% for the 11 test instances it investigated. Using the 10 test instances that all the algorithms jointly investigated as a measure, the ABO's cumulative relative error is 0.69%, to Lin-Kernighan's 0.739% and HBMO-TSP's 0.41%. This shows that the HBMO-TSP has a slightly better capacity to obtain optimal or near-optimal results than the other two algorithms. In terms of the use of computer resources, the worst performer is the HBMO-TSP. For just 11 out of the 17 TSP instances under investigation, the HBMO-TSP used 2, 744.500 seconds. In solving the same set of problems, the ABO used 2.849 seconds and the Lin-Kernighan algorithm used 44.500 seconds. It should be observed that the Lin-Kernighan algorithm did not attempt the time-consuming RL1189. From this analysis, it is obvious the fastest algorithm here is the ABO. In the light of the standard metrics for measuring the performance of optimization algorithms [19, 20], the ABO is the best of the three algorithms since it has the best performance in terms of consistency in obtaining solutions (Success Rate) as well as speed (CPU time). Similarly, it did very well in obtaining the optimal solution.

Table 1: Comparative Results

		ABO			LIN-KERNIGHAN			HBMO-TSP	
TSP instance	Optimum	Relative Error	Success Rate (%)	CPU Time (secs)	Relative Error	Success Rate (%)	CPU Time (secs)	Relative Error	CPU Time (secs)
HK48	11461	0.1	100		0.0	100	0.0	-	-
SWISS48	1273	1.5	100	0.07	0.0	100	0.0	-	-
GR96	55209	0.0	100	0.041	0.0	100	0.0	-	-
GR120	6942	0.26	100	0.039	0.0	100	0.0	-	-
GR137	69853	0.0	100	0.038	0.0	100	0.0	-	-
PR152	73682	0.065	100	0.048	0.0	100	0.1	0.0	2.21
BRG180	1950	0.21	100	0.140	0.0	100	0.0	-	-
FL417	11861	0.01	100	0.03	0.43	88	1.2	0.0	24.67
RL1304	252948	0.06	100	0.06	0.019	80	1.1	0.0	103.29
FL1400	20127	0.035	100	0.062	0.20	10	13.1	0.011	198.67
U1432	152970	0.0	100	0.087	0.0	100	0.8	0.016	200.01
D1655	62128	0.35	100	0.094	0.0	100	10.5	0.122	241.67
VM1748	336556	0.03	100	0.03	0.05	40	7.3	0.19	257.81
RL1889	316536	0.03	100	0.079	0.04	40	1.2	0.017	291.57
U2319	234256	0.05	100	0.128	0.0	100	0.5	0.028	391.08
PR2392	378032	0.06	100	0.133	0.0	100	8.7	0.026	401.28
RL11849	923288	0.03	100	1.958	-	-	-	0.10	890.05

CONCLUSION

This study examines the capacity of the ABO, HBMO-TSP & the Lin-Kernighan algorithms in solving some of the most difficult Symmetric Travelling Salesman's Problems. After close examination of the performances of the algorithms in terms of their Consistency, Speed and Efficiency [19, 20], the study concludes that the ABO performed better than the two other algorithms. The HBMO-TSP has an edge in obtaining optimal or near-optimal results that the other algorithms but it is an extremely slow algorithm. The performance of the Lin-Kernighan algorithm is better than that of the HBMO because of cost considerations (time taken to arrive at solutions). Overall, the best performer is the ABO. We, therefore, recommend more studies on the ABO to verify her capacity in solving other optimization problems.

ACKNOWLEDGEMENT

The authors acknowledge the immense contributions of the anonymous reviewers and the Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, Kuantan, for providing support for this research. Grant GRS 1403118. Also, our appreciation to the Ministry of Higher Education, Malaysia for funding this study under Exploratory Research Grant 120608.

REFERENCES

1. Gen, M. and R. Cheng, 2000. "Genetic algorithm and engineering optimization, " John Wily and Sons, New York.
2. Kennedy, J., 2010. "Particle swarm optimization, " in Encyclopedia of Machine Learning, ed: Springer, 2010, pp: 760-766.
3. Marinakis, Y., M. Marinaki and G. Dounias, 2011. "Honey bees mating optimization algorithm for the Euclidean traveling salesman problem, " Information Sciences, 181: 4684-4698.
4. Helsgaun, K., 2000. "An effective implementation of the Lin–Kernighan traveling salesman heuristic, " European Journal of Operational Research, 126: 106-130.
5. Odili, J.B. and M.N.M. Kahar, "African Buffalo Optimization (ABO): a New Meta-Heuristic Algorithm."
6. Odili, J.B., M.N.M. Kahar and S. Anwar, 2015. "African Buffalo Optimization: A Swarm-Intelligence Technique," Procedia Computer Science.
7. Applegate, D.L., R.E. Bixby, V. Chvatal and W.J. Cook, 2011. The Traveling Salesman Problem: A Computational Study: A Computational Study: Princeton University Press.
8. A.N. Julius Beneoluchi Odili Mohd Nizam Mohmad Kahar, Esther Abiodun Odili, "African Buffalo Optimization and the Randomized Insertion Algorithm for the Asymmetric Travelling Salesman's Problems" Journal of Theoretical and Applied Information Technology, 87: 356-364.
9. Odili, J.B. and M.N.M. Kahar, "Numerical Function Optimization Solutions Using the African Buffalo Optimization Algorithm (ABO)."
10. Odili, J.B. and M.N.M. Kahar, "Solving the Traveling Salesman's Problem using the African Buffalo Optimization."
11. J.B.O.A.M.N.M. Kahar, 2016. "African Buffalo Optimization," International Journal of Software Engineering and Computer Systems, 2: 28-50.
12. Odili, J.B., M.N.M. Kahar, S. Anwar and M.A.K. Azrag, 2015. "A comparative study of African Buffalo Optimization and Randomized Insertion Algorithm for asymmetric Travelling Salesman's Problem, " in Software Engineering and Computer Systems (ICSECS), 2015 4th International Conference on, pp: 90-95.
13. Quimper, C.G. and L.M. Rousseau, 2010. "A large neighbourhood search approach to the multi-activity shift scheduling problem," Journal of Heuristics, 16: 373-392.
14. Hemmelmayr, V.C., J.F. Cordeau and T.G. Crainic, 2012. "An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics," Computers & Operations Research, 39: 3215-3228.
15. Marinakis, Y. and M. Marinaki, 2008. "A bilevel genetic algorithm for a real life location routing problem," International Journal of Logistics: Research and Applications, 11: 49-65.
16. Odili, J.B., 2013. "Application of Ant Colony Optimization to Solving the Traveling Salesman's Problem," Science Journal of Electrical & Electronic Engineering, pp: 175-177.

17. Krishnamoorthy, M., A.T. Ernst and Y.M. Sharaiha, 2001. "Comparison of algorithms for the degree constrained minimum spanning tree," Journal of Heuristics, 7: 587-611.
18. Farmer, W.M., J.D. Guttman and F.J. Thayer, 1993. "IMPS: An interactive mathematical proof system," Journal of Automated Reasoning, 11: 213-248.
19. Baritompa, B. and E.M. Hendrix, 2005. "On the investigation of stochastic global optimization algorithms," Journal of Global Optimization, 31: 567-578.
20. Khompatraporn, C., J.D. Pintér and Z.B. Zabinsky, 2005. "Comparative assessment of algorithms and software for global optimization," Journal of Global Optimization, 31: 613-633.