Journal of Theoretical and Applied Information Technology
31st May 2016. Vol.87. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AFRICAN BUFFALO OPTIMIZATION AND THE RANDOMIZED INSERTION ALGORITHM FOR THE ASYMMETRIC TRAVELLING SALESMAN'S PROBLEMS

JULIUS BENEOLUCHI ODILI[1]* MOHD NIZAM MOHMAD KAHAR[1], A. NORAZIAH[1 3], ESTHER ABIODUN ODILI[2]

[1*] Research Scholar, Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, Kuantan, Malaysia

Senior [1]Lecturer, Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, Kuantan, Malaysia

[1, 3]Assoc. Prof Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, Kuantan, Malaysia

[1, 3]IBM Centre of Excellence, Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, Kuantan, Malaysia

[2]Researcher, Department Of Public Administration, Lagos State University, Ojo, Lagos, Nigeria

E-mail: [1*]odili_julest@yahoo.com [1]mnizam@ump.edu.my [1,3]noraziah@ump.edu.my [2]estherodili@yahoo.com

ABSTRACT

This paper presents a comparative study of the African Buffalo Optimization algorithm and the Randomized Insertion Algorithm to solving the asymmetric Travelling Salesman's Problem with the overall objective of determining a better method to solving the asymmetric Travelling Salesman's Problem instances. Our interest in the asymmetric Travelling Salesman's Problem (ATSP) is borne out of the fact that most practical daily-life problems are asymmetric rather than symmetric. The choice of the Random Insertion Algorithm as a comparative algorithm was informed by our desire to investigate the general belief among the scientific community that Heuristics being mostly problem-dependent algorithms are more efficient that metaheuristics that are usually general-purpose algorithms. Moreover, both the metaheuristic, the African Buffalo Optimization and the Heuristic, Randomized Insertion Algorithms hold some of the best results in literature in solving the ATSP. Similarly, both methods employ different search techniques in attempting solutions to the ATSP: while the African Buffalo Optimization uses the modified Karp-Steele technique, the Randomized Insertion employs random insertion mechanism. After investigating all the 19 benchmark ATSP datasets available in TSPLIB, it was discovered that the Randomized Insertion Algorithm achieves slightly better result to the problems but the African Buffalo Optimization is much faster.

Keywords: *Heuristics, Metaheuristics, Asymmetric Travelling Salesman's Problem, Randomized Insertion Algorithm, African Buffalo Optimization.*

1. INTRODUCTION

The search for better ways of doing things has led to several scientific investigations and the development of several deterministic, heuristic and metaheuristic algorithms, especially, in attempts to solve combinatorial problems [1]. Some of the deterministic algorithms include Raphson-Newton [2], Nelder-Mead [3] and Hooke-Jeeves [4] etc. The deterministic algorithms display exceptional capacity in identifying optimal solutions, only that they get weaker, slower and less efficient as the problem space enlarges [5]. The need for the development of more efficient search optimization techniques led to the design of heuristics and metaheuristic algorithms. Some of the popular and extremely efficient heuristic algorithms include the Lin-Kernighan algorithm [6], Randomized Insertion Algorithm [7], Branch-and-Bound heuristics [8], Divide-and-conquer algorithm [9], Dynamic programming [10], Greedy algorithm [11] etc. Similarly, the popular metaheuristics algorithms that have enjoyed wide applications include the Genetic Algorithm (GA) [12] Ant Colony Optimization (ACO) [13], Artificial Bee Colony (ABC) [14], Particle Swarm Optimization (PSO) [15] etc.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The major difference between the heuristics and metaheuristics algorithms being that heuristics are generally educated guesses which exploit readily available information towards solving a particular problem [16] while metaheuristics are rather general-purpose algorithms designed to solve different kinds of problems with proper tuning of the metaheuristic algorithm parameters. Moreover, metaheuristics are simply intelligent high level heuristics that operate by deliberately controlling and tuning lower level heuristic algorithms [17]. To this end, many experts claim that heuristics are more efficient than their metaheuristics counterpart [18]. This is one of the motivations for this comparative investigation since these two algorithms: African Buffalo Optimization (ABO), a metaheuristic technique and the heuristic, Randomized Insertion Algorithm (RAI) hold some of the best results in literature with respect to providing solutions to the asymmetric Travelling Salesman's Problems (ATSP).

1.1 The Travelling Salesman's Problem

The travelling Salesman's Problem (TSP) which is said to be one of the most studied problems among combinatorial optimizations problems since its development [19, 20], basically, refers to the problem of a particular salesman who travels round a set of locations in a large community or a given set of customers locations in a number of villages/town/cities visiting his customers and returning to the starting location at the end of the tour using the shortest/cheapest possible route. A constraint in this problem is that the travelling salesman is to ensure that as much as possible, he should visit a particular location/village/town/city exactly once. The starting city which is also the ending location is the only one allowed to have double visitation. In a way, therefore, the TSP is a graph theory challenge with the cities represented as vertices and the connecting links/roads are viewed as arcs of the vertices. To make the TSP as practical as possible, the arcs are given a weighted cost which represents the travel expenses, time or distances between points *j* to *k*. Associating an arc with a cost, helps in determining the route with the cheaper cost in the graph [21].

There are, primarily, two types of TSP: asymmetric TSP and the symmetric TSP. In the symmetric TSP, the cost/distance between arcs *j* and *k* is the same as that between *k* and *j* in the whole graph. Whereas, for the asymmetric TSP, there exist, at least, an instance where the distance between arcs *j* and *k* is not exactly the same cost/distance as that between *k* and *j*. The asymmetric TSP can be represented mathematically as:

$d_{jk} \neq d_{kj}$, $jk \in n$ in at least one edge in the graph. (1)

Also the symmetric TSP can be represented as

$$d_{jk} = d_{kj} + \varepsilon.$$
(2)

Since the development of the TSP in the early 1930s, there has been several scholarly investigations on the symmetric TSP. However, the same cannot be said of the asymmetric TSP. This is rather puzzling because the asymmetric TSP finds regular applications in most practical daily-life experiences. For instance, the problem of a haulage driver on a long haul is very likely asymmetric. So also is the case of an inspectorate officer on routine inspection to a number of company locations; a post office official delivering mails to different addresses within a given geographical locations; a school bus driver picking up school children and returning them at the end of a school day; a welfare official delivering food to home-bound people; an itinerant teacher on routine tutorial visits to his students' locations etc. [22] The most likely route-selection in solving these problems would be asymmetric. The asymmetric measurement of distances has a closer relationship to real-life applications as it takes cognizance of one-way traffic, special cost factors and other civil engineering considerations [21].

Specifically, this paper is a contribution to the ongoing investigation on determining the more efficient cum effective search system between the heuristics and metaheuristics algorithms [23, 24] by investigating the search efficiency and speed of the metaheuristic, African Buffalo Optimization algorithm and the heuristic Randomized Insertion Algorithm.

This paper is organized in this way: the first part discusses the major classes of optimization algorithms, introduces the Travelling Salesman's Problem as well as highlights the need for the asymmetric TSP; section two examines the African Buffalo Optimization (ABO) detailing the algorithms basic flow. The third section concentrates on the Randomized Insertion Algorithm (RAI) with emphasis on its procedures to obtain solutions to the target problem. The fourth session focusses on the experiments performed and the discussion of the results obtained. This is followed by the con-

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

clusion, acknowledgement of support for the study and references

2. AFRICAN BUFFALO OPTIMIZATION

The African Buffalo Optimization (ABO) is a recently developed lean metaheuristic optimization algorithm [25] which was designed primarily to provide solutions to issues of delay in obtaining solutions, stagnation, the use of several parameters etc. in the existing algorithms like the Genetic Algorithm, Simulated Annealing, Ant Colony Optimization and Particle Swarm Optimizations, to mention a few.

The ABO basically simulates the two basic vocalizations of the African buffalos in their migrant lifestyle through the African vast forests and savannahs in search of lush pastures. These are the '*maaa*' vocalizations with which the buffalos summon themselves to graze at a particular location because it is safe, favorable and has sufficient pastures as well as the alarm *'waaa'* communication calls with which they organize themselves to explore safer or more fruitful grazing locations [26, 27].

2.1. The Basic flow of ABO

The ABO starts by randomly initializing the buffalos, placing them to nodes/locations within the search space (in this case, the ATSP graph). Next, the animals, probabilistically, choose any closest and/or cheapest unvisited node for them to visit. This choice is influenced by the cost of the move determined solely by the available heuristic in the first move. Subsequent movements are influenced by the cost heuristic of such moves, the personal benefit of the move to the buffalo as determined by its previous experience and the overall benefit of the particular move to the entire buffalo population. Mathematically, this is represented by the democratic Equation (3) in Figure 1.

1. Initialization: randomly place buffalos at vertices at the solution space;
2. Assess the buffalos fitness using Equation (3)

$$m.k+1 = m.k + lp1(bgmax - w.k) + lp2(bpmax.k - w.k \quad (3)$$

where $w.k$ and $m.k$ are the exploration and exploitation moves respectively of buffalo k (k=1,2,………..N) ; $lp1$ and $lp2$ are learning factors; $bgmax$ is the herd's best fitness and $bpmax.k$, the individual buffalo's best fitness

3. Update the location of buffalo k ($bgmax$ and $bpmax.k$) using (4)

$$w.k+1 = \frac{(w.k+m.k)}{\pm 0.5} \quad (4)$$

4. Is $bgmax$ updating? Yes, go to 5. No, go to 2
5. Is the stopping criteria met? Yes, go to step 6, else step 3
6. Output best solution.

Figure 1: ABO algorithm

Moreover, the algorithm updates the buffalos' fitness. In this way, the algorithm determines the location of best buffalo ($bgmax$) in the herd in relation to the optimal solution. Also, each animal's personal best ($bpmax.k$) is determined. The buffalos keep a memory of their coordinates. If the current fitness value of a particular buffalo is superior to the prevailing $bgmax$, the algorithm saves it as the herd's best location $bgmax$. Similarly, if the current fitness of a particular buffalo is better than any in its memory, the algorithm saves it as that animal's best ($bpmax.k$). At this juncture, if the $bgmax$ meets the exit criteria, the algorithm terminates and provides the best buffalo's location vector as the optimal solution. If not, it goes to another iteration and repeats the process until it meets the exit criteria.

2.2. African Buffalo Optimization Solution mechanism for the ATSP

The basic solution steps of the ABO in solving the ATSP are:

A. Choose, according to the available heuristics, an initial city for each of the buffalos and randomly locate them in those cities.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$Pab = \frac{w^{lp1}ab\, m^{lp2}ab}{\sum_{i=1}^{n} w^{lp1}ab\, m^{lp2}ab} \quad (5)$$

$$ab = \pm 0.5$$

B. Update buffalo fitness using Equations 3 and 4, respectively.
C. Determine $bpmax.k$ & $bgmax$.
D. Using Equation 5 and available heuristic values, probabilistically construct a buffalo tour by adding cities that the buffalos have not visited.
E. Confirm that the $bgmax$ is updating? Yes, go to (F.). No, go to (A.)
F. Is the exit criteria reached? Yes, go to (G.). No, return to (B.)
G. Output the best result.

2.3. Internal Workings of the ABO in Arriving at a Solution

The ABO employs the Modified Karp Steele solution technique in its solution of the Asymmetric Travelling Salesman's Problem [28]. This solution technique follows a simple solution steps of, firstly, constructing a cycle factor *F* of the cheapest weight in the *K* graph. Secondly, select a pair of arcs taken from different cycles of the *K* graph and patch in a way that will result in a minimum weight. Patching is basically removing the selected arcs in the two cycle factors and then replacing them with cheaper arcs and in this way forming a bigger cycle factor, thereby reducing the number of cycle factors in graph *K* by one.

Thirdly, the second step is repeated until we arrive at a single cycle factor in the entire graph *K*. The African Buffalo Optimization handles the problem of delay in arriving at solutions through the use of two primary parameters to ensure speed, namely lp1 and lp2. Similarly, the algorithm closely monitoring the route of the $bgmax$ as well as ascertaining the $bpmax.k$ in each construction step helps solve the problem of stagnation and ensure efficiency. In the event that the $bgmax$ fails to update in a number of iterations, the ABO re-initializes the entire herd.

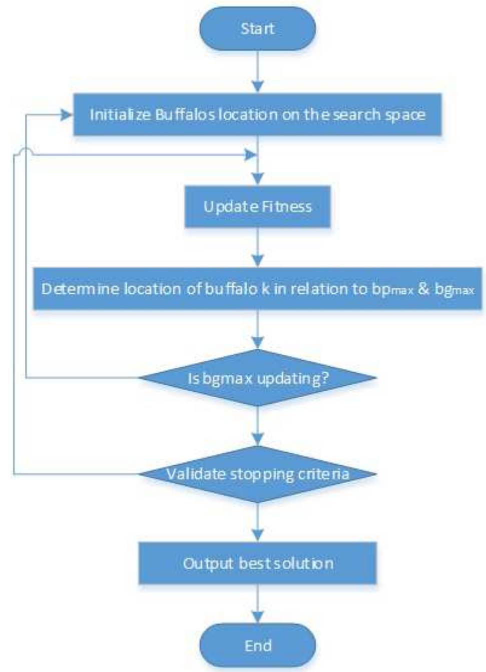

Figure 2: ABO flowchart

3. THE RANDOMIZED INSERTION ALGORITHM (RAI)

The Randomized Insertion Algorithm which uses the arbitrary insertion algorithm is a relaxation of the Cheapest Insertion Algorithm and was developed to provide solution to the Asymmetric Travelling Salesman's Problems through the use of approximate algorithm that will obtain optimal results within the shortest possible time [29]. The RAI has one of the best results in literature in attempting solutions to the ATSP and this informs its choice in this comparative investigation.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.1 The RAI Algorithm

The RAI algorithm is presented in Figure 3 below:

1. Initiate tour construction with a randomly selected node and self–loop.
2. Randomly select an arc that is not in the initial tour.
3. Add this arc between neighboring arcs on the tour in the cheapest way possible. If the tour is still incomplete, return to step 2.
4. Store this tour solution as, say, Tour S.

Repeat twice in each construction step, steps 5 through 9.

5. Randomly choose i and j (i, j Є N = {1… n}, 1 ≤ i ≤ j ≤ n).
6. From the circuit with all arcs, remove a path beginning with arc i through arc j, and connect arc $i - 1$ with arc $j + 1$.
7. Randomly choose an arc from the removed path.
8. Insert this arc between two neighboring arcs on the tour in the cheapest possible way. If the tour is still not complete, go to step 7.
9. Compare present solution with the solution S. Keep the better one.

Figure 3: The RAI algorithm

This algorithm works by generating an initial tour (See algorithm steps 1-4) and through subsequent systematic removal and addition of arcs in the cheapest possible generates a good near-optimal or optimal solution.

3.2. The Internal Solution Strategy of RAI

The basic solution steps of the RAI is, firstly, randomly select two initial nodes i and j, then form a Cycle ij within the graph. Next in each subsequent construction step, randomly choose a node which is not a part of the current Cycle and insert into the Cycle in the cheapest possible way. In this way the cost of the Cycle increases minimally. This procedure is repeated until all the nodes are added into the tour. Then the algorithm calculates the path discovered and outputs a solution.

4. EXPERIMENTS AND DISCUSSION OF RESULTS

In applying these two algorithms, the African Buffalo Optimization and the Randomized Insertion Algorithm to solve Asymmetric Travelling Salesman's Problems, a number of experiments were performed on all the 19 ATSP instances as listed in the TSPLIB95. The experiments were done using a desktop computer running the Windows 7 O.S, Intel Core, i7-3770 CPU@ 3.4GHz, 3.4GHz, 4 GB RAM. The ATSP is coded in MATLAB programming language and ran on MATLAB 2012b Compiler. The results obtained from the experiment using African Buffalo Optimization were compared with the results from the Randomized Insertion Algorithm [30] and presented in Table 1.

In Table 1, the first column lists the ATSP instances as available in TSPLIB [31], the second column indicates the number of cities represented by the ATSP instances. This is followed by the Opt Values, that is, the optimal results as listed in TSPLIB. The next two sets of columns calculates the value obtained from applying the ABO and the RAI respectively with Best = the best values obtained by the algorithm; Average = Average values obtained after 50 runs; Rel. Error% = percentage relative error from the optimum; and Time (secs) = the best time in seconds that the algorithm used to obtain result.

The relative error was obtained by:

$$Relative\ Error = \frac{Best - Opt\ Values}{Opt\ Values}\ x\ 100 \quad (6)$$

As can be seen from the Table, the ABO obtained very good results in all instances, obtaining over 96.7% in all 19 ATSP instances under investigation. This is a noble feat. The RAI performed excellently well too, as can be seen in Table 1, obtaining over 98.2% cumulative accuracy in all cases.

Moreover, the ABO obtained the optimal solution in five instances to RAI's 13 accurate results. The difference in performance here can be traceable to their use of two different search techniques in obtaining results. While the RAI uses the random insertion method, ABO employs the modified Karp- Steele method. Whatever, it is a competitive performance.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Comparative Experimental Results

ATSP Instances	No of Cities	Opt Value	ABO				RAI			
			Best	Average	Rel. Error%	Time (secs)	Best	Average	Rel. Error%	Time (secs)
Br17	17	39	39	39.98	0	0.028	39	39	0	0.027
Ry48p	48	14422	14440	14455	0.12	0.037	14422	14543.20	0	1.598
Ft70	70	38673	38753	38870.5	0.21	0.05	38855	39187.75	0.47	7.068
Ftv33	34	1286	1287	1288.4	0.08	0.029	1286	1288.16	0	0.393
Ftv35	36	1473	1474	1475.8	0.07	0.030	1473	1484.48	0	0.508
Ftv38	39	1530	1530	1536.4	0	0.026	1530	1543.12	0	0.674
Ftv44	45	1613	1614	1647.25	0.06	0.032	1613	1643.6	0	1.198
Ftv47	48	1776	1777	1783	0.06	0.029	1776	1782	0	1.536
Ft53	53	6905	6905	6920.25	0	0.028	6905	6951	0	2.398
Ftv55	56	1608	1610	1618.2	0.12	0.029	1608	1628.74	0	2.878
Ftv64	65	1839	1839	1938	0	0.041	1839	1861	0	5.241
Ftv70	71	1950	1955	1958.5	0.26	0.09	1950	1968.44	0	7.376
Ftv170	171	2755	2795	2840.5	1.45	0.65	2764	2832.74	0.33	276.1
Kro124p	100	36230	36275	36713	0.12	0.08	36241	36594.23	0.04	30.34
P43	43	5620	5645	5698	0.44	0.065	5620	5620.65	0	0.997
Rbg323	323	1326	1326	1417.75	0	2.050	1335	1348	0.68	3874
Rbg358	358	1163	1187	1299.2	0.18	3.040	1166	1170.85	0.26	6825
Rbg403	403	2465	2467	2475	0.08	4.741	2465	2466	0	11137
Rbg443	443	2720	2723	2724	0.11	10.377	2720	2720	0	17126
TOTAL					3.24	21.452			1.78	39300.332

Furthermore, the cumulative relative error of the ABO (calculated by summing up the values of the relative errors) is 3.24% to RAI's 1.78%. This is also a commendable performance by the ABO in view of the fact that the RAI is a heuristic algorithm and has one of the best results in literature. It is also interesting to note that the while the RAI had the biggest problem in Rbg323, the ABO obtained the optimal result there. Conversely, the ABO biggest challenge was in Ftv170 where it recorded the highest relative error of 1.45%. Here the RAI had a relative error of just 0.33%. This, again, could be attributable to their use of different search techniques.

Nonetheless, the uncommon strength of the ABO comes to play when calculating the execution costs. The execution cost calculates the relative use of CPU resources in achieving results. In an instance-by-instance execution speed assessment, the RAI slightly outperformed the ABO in just one instance, namely, Br17 where ABO executed in 0.028 seconds to RAI's 0.027 seconds. In all the remaining 18 instances, the algorithm of choice when speed is a factor is the ABO. In the second ATSP instance, it took ABO 0.037 seconds to obtain optimal result to RAI's 1.598 seconds: ABO here was over 43 times faster. This trend continues in all other instances. For example, in Ftv64 while RAI spent 5.241seconds to execute, the ABO used as just 0.041 seconds. Here ABO is about 127.83 times faster. Also, in the difficult ATSP instance Ftv170, it took ABO 0.65 seconds to RAI's 276.1 seconds. This translates to ABO being over 424.76 times faster.

Similarly, the ATSP instances involving larger number of cities such as rbg323, rbg358, rbg403 and rbg403 are where the speed of ABO is even more glaring. For instance it took ABO 2.050 seconds to obtain optimal solution to RAI's 3874 seconds [ABO being 1889.75 faster] in rbg323; in solving rbg358, ABO used 3.040 seconds to RAI's 6825 seconds [ABO being 2245.07 times faster]; for rbg403, ABO used 4.741 seconds to RAI's 11137 seconds [ABO being 2349.08 times faster] and finally in rbg443, ABO spent 10.377 seconds to RAI's 17126 seconds [ABO, again, being 1650.38 times faster].

In all, while the ABO used 21.452 seconds to solve all the 19 ATSP instances under investigation, the RAI spent 39300.332 seconds.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

So cumulatively, the ABO was 1,832.012 times faster than the RAI.

Someone may rightly say that speed is a function of the CPU speed, RAM's capacity, the programmer's expertise, the programming language of implementation and a few other factors but it should also be observed that the algorithm that uses such few parameters and has a straight-forward fitness function like the ABO will be difficult to match whenever speed is a factor [32, 29, 33, 34]. And since speed is one of the hallmarks of a good algorithm, it could be safe to say that the ABO clearly is a faster algorithm and outperforms the RAI in executing the benchmark ATSP instances in TSPLIB.

5. CONCLUSION

From the foregoing analysis and discussion of results, it is obvious that both ABO and RAI are competitive in obtaining optimal solutions to Asymmetric Travelling Salesman's Problem instances. The results show that RAI slightly outperformed the ABO in attaining optimal solutions to the ATSP cases under investigation with 98.22% accuracy to ABO's 96.76%. Also, RAI obtained the optimal solution in 13 out of the 19 instances, with the ABO obtaining optimal solution in five instances and very near optimal solution in the remaining cases. However, in terms of speed required to obtain results, the ABO is the dominant algorithm of choice. The ABO cumulatively was 1,832.012 times faster than the RAI. Similarly, in an instance-by-instance speed assessment, the ABO clearly outperformed the RAI in 18 out of the 19 ATSP instances under investigation. The RAI was slightly faster than the ABO only in Br17 where it executed at 0.027 seconds to ABO's 0.028 seconds.

In conclusion, therefore, since accuracy (trustworthiness) and efficiency are two of the four major criteria for determining a better algorithm, the other two being general applicability and ease of use [29, 35], the ABO can be adjudged a better algorithm than RAI since it performed creditably well in obtaining optimal or near-optimal results in all the test cases under investigation. Even though the RAI slightly outperformed the ABO here by 1.76%. ABO, being a metaheuristic algorithm should have wider applicability than the RAI which is a heuristic. Since speed is one of the measures of efficiency, it is safe to conclude that the ABO is a better algorithm in solving the benchmark ATSP instances and this result further validates the view that metaheuristics are more efficient algorithms than heuristics [36, 24]. The authors, therefore recommend the comparison of the performance of ABO with other state-of-the-art algorithms in solving other optimization problems like PID tuning of AVR parameters, knapsack problem, job scheduling and vehicle routing.

6. ACKNOWLEDGEMENT

The authors appreciate the anonymous reviewers for the critical review of this paper. Similarly, the authors acknowledge the support of the Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, Kuantan 26300, Malaysia for providing the conducive environment for this study under Grant 1403118. Also, our appreciation to the Ministry of Higher Education, Malaysia for funding this study under Exploratory Research Grant 120608.

REFERENCES

[1] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys (CSUR),* vol. 35, pp. 268-308, 2003.

[2] S. Bisterzo, R. Gallino, O. Straniero, S. Cristallo, and F. Käppeler, "The s-process in low-metallicity stars–III. Individual analysis of CEMP-s and CEMP-s/r with asymptotic giant branch models," *Monthly Notices of the Royal Astronomical Society,* vol. 422, pp. 849-884, 2012.

[3] R. Barati, "Parameter estimation of nonlinear Muskingum models using Nelder-Mead simplex algorithm," *Journal of Hydrologic Engineering,* 2011.

[4] A. Rios-Coelho, W. Sacco, and N. Henderson, "A Metropolis algorithm combined with Hooke–Jeeves local search method applied to global optimization," *Applied Mathematics and Computation,* vol. 217, pp. 843-853, 2010.

[5] A. W. Toga, K. A. Clark, P. M. Thompson, D. W. Shattuck, and J. D. Van Horn, "Mapping the human connectome," *Neurosurgery,* vol. 71, p. 1, 2012.

[6] D. Karapetyan and G. Gutin, "Lin–Kernighan heuristic adaptations for the generalized traveling salesman problem," *European Journal of Operational Research,* vol. 208, pp. 221-232, 2011.

[7] R. Motwani and P. Raghavan, *Randomized algorithms*: Chapman & Hall/CRC, 2010.

[8] C. Gatu and E. J. Kontoghiorghes, "Branch-and-bound algorithms for computing the best-subset regression

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

models," *Journal of Computational and Graphical Statistics,* 2012.

[9] X. He and K. M. Merz Jr, "Divide and conquer Hartree− Fock calculations on proteins," *Journal of chemical theory and computation,* vol. 6, pp. 405-411, 2010.

[10] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*: John Wiley & Sons, 2014.

[11] Y. Wang, G. Cong, G. Song, and K. Xie, "Community-based greedy algorithm for mining top-k influential nodes in mobile social networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 1039-1048.

[12] M. Moradi and M. Abedini, "A combination of genetic algorithm and particle swarm optimization for optimal DG location and sizing in distribution systems," *International Journal of Electrical Power & Energy Systems,* vol. 34, pp. 66-74, 2012.

[13] M. Dorigo and M. Birattari, "Ant colony optimization," in *Encyclopedia of machine learning*, ed: Springer, 2010, pp. 36-39.

[14] B. Akay and D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization," *Information Sciences,* vol. 192, pp. 120-142, 2012.

[15] S. Intelligence, "Particle Swarm Optimization," *MCCAFFREY, James.[online].[cit. 2014-05-20]. Dostupné z: http://msdn. microsoft. com/en-us/magazine/hh335067. aspx,* 2013.

[16] A. Perdomo-Ortiz, S. E. Venegas-Andraca, and A. Aspuru-Guzik, "A study of heuristic guesses for adiabatic quantum computation," *Quantum Information Processing,* vol. 10, pp. 33-52, 2011.

[17] X.-S. Yang, "Harmony search as a metaheuristic algorithm," in *Music-inspired harmony search algorithm*, ed: Springer, 2009, pp. 1-14.

[18] D. Pisinger and S. Ropke, "A general heuristic for vehicle routing problems," *Computers & Operations Research,* vol. 34, pp. 2403-2435, 2007.

[19] J. B. Odili, H. Deng, C. Yang, Y. Sun, M. C. d. E. Santo Ramos, M. C. G. Ramos*, et al.*, "Application of Ant Colony Optimization to Solving the Traveling Salesman's Problem," *Science Journal of Electrical & Electronic Engineering,* vol. 2013, 2013.

[20] A. Rodríguez and R. Ruiz, "The effect of asymmetry on traveling salesman problems," 2010.

[21] J. Brest and J. Zerovnik, "A heuristic for the asymmetric traveling salesman problem," in *The 6th Metaheuristics International Conference*, 2005, pp. 145-150.

[22] R. Gottlieb, A. Fisher, M. Dohan, L. O'Connor, and V. Parks, "Homeward Bound: Food-related transportation strategies in low income and transit dependent communities," *University of California Transportation Center,* 1996.

[23] F. Glover and M. Laguna, "Tabu search, 1997," *Cited on,* p. 83, 1997.

[24] F. Glover and M. Laguna, *Tabu Search∗*: Springer, 2013.

[25] J. B. Odili and M. N. M. Kahar, "African Buffalo Optimization (ABO): a New Meta-Heuristic Algorithm," *Journal of Advanced & Applied Sciences,* vol. 03, pp. 101-106, april 2015 2015.

[26] J. B. Odili and M. N. M. Kahar, "Numerical Function Optimization Solutions Using the African Buffalo Optimization Algorithm (ABO)," *British Journal of Mathematics & Computer Science,* vol. 10, pp. 1-12, 2015.

[27] J. B. Odili and M. N. Mohmad Kahar, "Solving the Traveling Salesman's Problem Using the African Buffalo Optimization," *Computational Intelligence and Neuroscience,* vol. 501, p. 929547, 2015.

[28] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," *BioSystems,* vol. 43, pp. 73-81, 1997.

[29] B. Baritompa and E. M. Hendrix, "On the investigation of stochastic global optimization algorithms," *Journal of global optimization,* vol. 31, pp. 567-578, 2005.

[30] J. Brest and J. Zerovnik, "An approximation algorithm for the asymmetric traveling salesman problem," *Ricerca operativa,* vol. 28, pp. 59-67, 1999.

[31] G. Reinelt, "TSPLIB—A traveling salesman problem library," *ORSA journal on computing,* vol. 3, pp. 376-384, 1991.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[32] P. J. Angeline, "Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences," in *Evolutionary Programming VII*, 1998, pp. 601-610.

[33] D. Karaboga and B. Akay, "A modified artificial bee colony (ABC) algorithm for constrained optimization problems," *Applied Soft Computing,* vol. 11, pp. 3021-3031, 2011.

[34] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary programming VII*, 1998, pp. 591-600.

[35] C. Khompatraporn, J. D. Pintér, and Z. B. Zabinsky, "Comparative assessment of algorithms and software for global optimization," *Journal of global optimization,* vol. 31, pp. 613-633, 2005.

[36] G. Gigerenzer and W. Gaissmaier, "Heuristic decision making," *Annual review of psychology,* vol. 62, pp. 451-482, 2011.