INTELLIGENT AUTOMATION & SOFT COMPUTING, 2017
https://doi.org/10.1080/10798587.2017.1334370

Performance Analyses of Nature-inspired Algorithms on the Traveling Salesman's Problems for Strategic Management

Julius Beneoluchi Odili[a], Mohd Nizam Mohmad Kahar[a], A. Noraziah[a,b], M. Zarina[c] and Riaz Ul Haq[a]

[a]Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, Kuantan, Malaysia; [b]IBM Centre of Excellence, Universiti Malaysia Pahang, Kuantan, Malaysia; [c]Faculty of Informatic and Computing, Universiti Sultan Zainal Abidin, Gong Badak, Malaysia

ABSTRACT
This paper carries out a performance analysis of major Nature-inspired Algorithms in solving the benchmark symmetric and asymmetric Traveling Salesman's Problems (TSP). Knowledge of the workings of the TSP is very useful in strategic management as it provides useful guidance to planners. After critical assessments of the performances of eleven algorithms consisting of two heuristics (Randomized Insertion Algorithm and the Honey Bee Mating Optimization for the Travelling Salesman's Problem), two trajectory algorithms (Simulated Annealing and Evolutionary Simulated Annealing) and seven population-based optimization algorithms (Genetic Algorithm, Artificial Bee Colony, African Buffalo Optimization, Bat Algorithm, Particle Swarm Optimization, Ant Colony Optimization and Firefly Algorithm) in solving the 60 popular and complex benchmark symmetric Travelling Salesman's optimization problems out of the total 118 as well as all the 18 asymmetric Travelling Salesman's Problems test cases available in TSPLIB91. The study reveals that the African Buffalo Optimization and the Ant Colony Optimization are the best in solving the symmetric TSP, which is similar to intelligence gathering channel in the strategic management of big organizations, while the Randomized Insertion Algorithm holds the best promise in asymmetric TSP instances akin to strategic information exchange channels in strategic management.

KEYWORDS
Decision-making; logistics management; nature-inspired algorithms; performance analysis; strategic planning; traveling salesman's problems

1. Introduction

Information, Communication & Telecommunications (ICT) has proven to be very useful in decision-making in diverse spheres of human endeavor ranging from Pharmacy to Medicine, Engineering to business establishments, corporate to diplomatic circles etc. (Shrivastava, Ivanaj, & Ivanaj, 2016). The contributions of ICT to human development could be in hardware or in software. ICT has been beneficial to strategic management in the development of strategic management tools, organizational strategies and marketing, intelligence gathering, strategic information exchange, management service delivery, design of management methodologies etc. (Hénard, Diamond, & Roseveare, 2012).

Strategic management is a vital tool in organizational effectiveness and efficiency. Basically, strategic management is concerned with the initialization, development and implementation of an organization's objectives based primarily on the available resources and the constraints. Strategic management has been the subject of a number of research studies, because of its importance in the success of any organization and since organizational resources and constraints are dynamic, it follows; therefore that strategic management should also be dynamic. This reason necessitates the continuous search for organizational tools and concepts in order to enhance strategic management. The guiding principles of TSP become useful in strategic information exchange and strategic management service delivery (David & David, 2016; Freeman, 2010).

One of the most studied software concepts, but least applied in ICT is the Travelling Salesman's Problem (Applegate, Bixby, Chvatal, & Cook, 2011; Odili, 2013). The Travelling Salesman's Problem (TSP) has been one of the most popular combinatorial optimization problems since its design in the early 20th century. It has attracted so much attention of researchers in the past five decades to the extent that, consciously or unconsciously, it is fast becoming a testbed for any novel optimization algorithms (Odili & Kahar, 2015a). Basically, the TSP is a problem of a particular salesman who has customers in diverse geographical locations in a large city or in different locations in a given geographical area, possibly a State, Province, nation, continent or even a number of continents. The job of the Salesman is to visit all his customers and return to the starting location using the cheapest, shortest or fastest route. To assist the Salesman in his assignment, the travelling cost of each route in terms or time, length or other costs consideration are indicated with a weight value (Karagul, Aydemir, & Tokat, 2016).

The TSP could be formulated to be symmetric or asymmetric. In symmetric TSP, the cost is the same in either direction of the trip. That is to say that the weight value is the same whether the Salesman is travelling from location x to y or y to x. However, in the asymmetric TSP, there exists at least an edge in the travelling graph where the travelling weight value is not the same. This means that the cost in terms of time, distance etc. could be different depending on the Salesman's direction of travel (Osaba, Yang, Diaz, Lopez-Garcia, & Carballedo, 2016). The asymmetric TSP finds relevant applications in real life situations arising from one-way traffic, variations in traffic tax, or other civil engineering or commercial situations (Odili & Mohmad Kahar, 2016b). In a situation where a few locations/cities are involved, the problem appears cheap, but in other

CONTACT Julius Beneoluchi Odili odili_julest@yahoo.com

situations where the locations/nodes run into several hundreds or thousands, it is a complex combinatorial optimization problem. The TSP finds application in product assembly plants as it provides a direction as to the most optimized procedural path to complete the assembly process in the most effective and efficient manner. Other application areas of the TSP include postal mail delivery, packet delivery in networking environments, routing of vehicles in a transport firm, air-traffic routing etc. In all of these problems, effectiveness and efficiency derive from taking the appropriate and well-informed decisions. Several software have proven handy to strategic planners in the past few decades, hence the need to investigate the application of the TSP (Barry & Edgman-Levitan, 2012; Ma, Wang, Zhu, & Zhou, 2015; Tremblay & Dossou-Yovo, 2015).

Among several algorithms that have attempted the TSP are the Ant Colony Optimization (Odili & Kahar, 2016), Artificial Bee Colony (Uz, Kiran, & Özceylan, 2015), Particle Swarm Optimization (Baridam & Nnamani, 2016), Genetic Algorithm (Chang, 2015), Simulated Annealing (Zeb et al., 2016) etc. A closer look at the performances of these algorithms, however, indicate that there exist a gulf in time, use of computer resources, ability to obtain optimal or near-optimal results etc. in their solution to the same benchmark TSP problem (Osaba et al., 2016). Our interest in the performance analysis of these algorithms is as a result of the observed differences in efficiency and effectiveness. It is hoped that the outcome of this study will assist researchers confronted with the TSP or any other similar strategic/logistic planning problem make informed choices of combinatorial optimization algorithms in solving real life problems.

The rest of this paper is structured as follows: Section Two discusses the comparative Nature-inspired Algorithms (NAs); Section Three deals with NAs diagnoses and performs a comparative analysis of different NAs, while Section Four draws conclusions on the study.

2. Nature-inspired Algorithms

Metaheuristic algorithms have been classified in several ways in literature. One of such ways is to classify them as either population-based or trajectory-based. Population-based optimization techniques solve problems using a population of solutions (otherwise called, decision vectors) at a time (Wong & Moin, 2015). Usually the initial solution is obtained randomly and then improved upon iteratively. Population-based algorithms are sometimes called exploration-based methods since they are exceptionally good in the diversification of the search space. An example of this is the GA, which uses a set of strings. Also is the Particle Swarm Optimization that uses a number of agents or particles (Kennedy, 2010). On the other hand, Simulated Annealing, Great Deluge and Hill Climbing etc. are trajectory-based and use a single agent that moves through the search space in a zigzag fashion as the iterations continue (Kennedy, 2010; Mirjalili, Mirjalili, & Lewis, 2014). The difference between the Population-based and the Trajectory-based metaheuristics is primarily the number of temporal solutions applied in an iteration of the search. While the population-based methods use a number of agents and, therefore, solutions, the Trajectory-based methods use a single search agent and, as such, generate a single solution per iteration.

For the purpose of this study, the comparative population-based Nature-inspired Algorithms are the Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Genetic Algorithm (GA), Artificial Bee Colony (ABC), Firefly Algorithm (FFA), African Buffalo Optimization (ABO) and the Bat Algorithm (BA). Similarly, the trajectory algorithms under investigation are the Simulated Annealing, and the Hill Evolutionary Simulated Annealing. Also, the heuristics algorithms under investigation are the Honey Bee Mating Optimization for the Travelling (HBMO-TSP) (Marinakis, Marinaki, & Dounias, 2011) and the Randomized Insertion Algorithm (RAI) (Brest & Zerovnik, 2005). The choice of the comparative is a result of their exceptionally good results coupled with their popularity among researchers. Let us attempt a brief description of the comparative algorithms.

2.1. Simulated Annealing

Simulated Annealing (SA) is a simulation of the heating and cooling processes of metals in metallurgical engineering. It was developed by Kirkpatrick, Gelatt and Vecchi (Kirkpatrick, Gelatt, & Vecchi, 1983). The temperature is slowly lowered to ensure that the cooling process minimizes the system energy, thus making the metals very strong. During the cooling phase, SA starts with a random search at high temperature and gradually degenerates into greedy descent until the temperature gets to zero. SA is able to avoid premature convergence, which is a major weakness of Greedy Descent through the use of randomness. At each move, the algorithm assigns random values to random variables, and then it evaluates the acceptance probability to determine whether the move provides an improvement to the objective function without increasing the conflict. Sometimes, however, the SA accepts points that raise the objective function and in this way avoids being trapped in local minima. Application areas of the SA includes, Job Shop Scheduling (van Laarhoven, Aarts, & Lenstra, 1992), N-Queens problem (Tambouratzis, 1997), Artificial Neural Networks Training (Ledesma, Aviña, & Sanchez, 2008), the Quadratic Assignment Problem (Bilbao & Alba, 2009), Travelling Salesman's Problems (Malek, Guruswamy, Pandya, & Owens, 1989) etc.

A critical assessment of the SA indicates its strength in escaping being trapped in a local minimum through its careful manipulation of the cooling temperature. Nonetheless, SA is ineffective in smooth energy landscape as well as in instances with few local minima. Also, SA's speed in solving optimization problems is suspect (Kumbharana & Pandey, 2013). This reason for SA's slow speed could be traceable to several evaluations of the cost function in each iteration.

2.2. Evolutionary Simulated Annealing

The Evolutionary Simulated Annealing (ESA) is an attempt to solve the problems of being trapped in local minima coupled with delay in obtaining reasonable solution often associated with the classical Simulated Annealing. ESA is a hybridization of the SA with evolutionary principles of Evolutionary algorithms such as the GA, GP and ES. There have been some other efforts aimed at similar objectives. Examples include the hybrid approaches combining the Simulated Annealing (SA) with hill climbing or Genetic Algorithms (Kolonko, 1999; Zeb et al., 2016).

In ESA, the SA is adopted as an evolutionary operator together with other evolutionary operators, which do not contain any reproduction operators (crossover or mutation). After initialization and parameters setting, the algorithm repeats the following steps: (i) choses one individual defined by the running selection rule, (ii) operates on that individual with the SA operator, and (iii) decides with certain probability whether to put it back into the population or not through a stated replacement principle. This process is repeated over and over again until a stopping criterion is reached (Aydin & Fogarty, 2004).

2.3. Randomized Insertion Algorithm

The Randomized Insertion Algorithm (RAI) is a heuristic algorithm, which was developed to provide solution to the asymmetric Travelling Salesman's Problem. The RAI uses the arbitrary insertion algorithm, which is a relaxation of the cheapest insertion algorithm in its search for solutions. RAI has proven to be capable of obtaining solutions through the use of approximation mechanism. Till date, the RAI has one of the best results in solving ATSP instances available in the TSPLIB (Odili, Kahar, Anwar, & Azrag, 2015).

2.4. Honey Bee Mating Optimization for the Travelling Salesman's Problem

Honey Bee Mating Optimization for the Travelling Salesman's Problem (HBMO-TSP), is a combination of the Honey Bees Mating Optimization (HBMO) algorithm, the Expanding Neighborhood Search Strategy (ENS) and the Multiple Phase Neighborhood Search-Greedy Randomized Adaptive Search Procedure (MPNS-GRASP) algorithm. In the HBMO-TSP, the MPNS-GRASP is used to calculate the initial population of bees and their initial queen, the ENS randomly picks effective workers that will be used for the improvement of its solution during the local search phase. In order to have fitter brood, the Adaptive Memory Procedure is used to select crossover operators that combine the genotype of the queen and those of more than one drone to produce a brood. This hybridization has proven quite successful in solving the ATSP (Marinakis et al., 2011).

2.5. Particle Swarm Optimization

Particle swarm optimization (PSO) was proposed by Kennedy and Eberhart as a population-based, stochastic, global optimization technique inspired by schooling of fishes and the flocking of birds (Kefi, Rokbani, Krömer, & Alimi, 2015). PSO models the velocity and positions of fishes and birds in its search for solutions. PSO calls these simple search agents particles and they represent solutions to the search enterprise. Using a simple rule, a particle updates its velocity and position with each iteration as the algorithm progresses until it reaches termination condition. (Kefi et al., 2015).

PSO has been used successfully to address several optimization problems such as task assignment (Shi & Eberhart, 1999), complex nonlinear function optimization, antenna design, biomedical and pharmaceutical applications, communications and control applications, combinatorial optimization problems, fault diagnosis etc. (Poli, 2007).

A critical examination indicates that PSO has a more efficient memory capability than algorithms like the GA. Moreover, PSO achieves better search diversity since all the particles use the information obtained by the best particles in each iteration to improve their locations and speed. However, PSO uses several parameters and this affects the speed and efficiency of the algorithm (Tanweer, Suresh, & Sundararajan, 2015).

2.6. Genetic Algorithm

The Genetic Algorithm (GA) was developed by John Holland in the early 1970s as a model of biological evolutions, which uses such operators as selection, crossover, mutation, and inversion. It uses a population of individuals as x-y chromosomes in the form of character strings. The selected individuals in a population are made to evolve through crossover and mutations to form the next generation from where the best generations are selected to form the next generation. This process is continued until the algorithm arrives at the optimal or near optimal solution to a particular problem. Real-life applications of Genetic Algorithms include automotive design, robotics, Airline revenue management, control engineering, water resource management, software testing etc. (Dasgupta & Michalewicz, 2013; Srivastava & Kim, 2009).

GA has been used to solved several problems ranging from the Travelling Salesman's Problems, numerical function optimization, proportional, integral and derivative parameters tuning of Automatic Voltage Regulators (Mustafa, Al Gizi, & Alsaedi, 2012) etc.

The GA was originally invented to optimize problems that can be represented as a vector of binary values. In such problems, the GA has proven to be very effective. The efficiency of the GA is however in doubt, partly because it employs a complicated calculation of fitness function in addition to its use of several parameters such as population size, crossover, mutation rate, recombination, selection operator etc. (Man, Tang, & Kwong, 2012).

2.7. Ant Colony Optimization

The Ant Colony Optimization (ACO) algorithm, which was designed by Marco Dorigo and Di Caro in 1999, was inspired by the random walk of ants in search of food (Anwar, Salama, & Abdelbar, 2015). Once a food source is located, the ant with the breakthrough carries a particle of the food and on its way to the nest, usually follows an optimized route depositing some pheromones as a way of announcing its success. The other ants retrieve this message and heads to the food source. Arriving at the food source, they, in turn, carry some fragments to the nest, dropping pheromones along the way as they further optimize the route of the initial ant. This process increases the pheromone concentration on the favorite 'shortest' route and attracts other ants. Conversely, when the food source gets harvested, the ants on their way back drop no pheromone leading to pheromone evaporation. This situation forces the ants to explore other areas (Wu, Xin, & Zhang, 2015).

The ACO has been widely applied to solve several problems ranging from routing problems, communication networks, network problems, Resource Constraint Project Scheduling, Machine Learning Problems, Sequential Ordering Problems, Travelling Salesman's Problems, Subset problems, (Odili et al., 2013; Stützle, López-Ibáñez, & Dorigo, 2011) etc.

The ACO has proven to be very efficient in handling computational noise (Mininno & Neri, 2010). Also ACO performs well in distributed computing environments (Kant, Sharma, Agarwal, & Chandra, 2010). However, it has the weakness of easily falling into premature convergence, because its pheromone update is according to the present best path. Again, it uses several parameters that require proper tuning. Such parameters as pheromone quantity, pheromone update rule, evaporation rate, pheromone reinforcement rate etc. (Gutjahr, 2003; Kumbharana & Pandey, 2013).

2.8. Artificial Bee Colony Algorithm

The Artificial Bee Colony (ABC), which was proposed by Karaboga and Basturk in 2009 and inspired by the behavior of honey bee swarm, categorizes bees into three; onlooker bees, scout bees and employed bees (Karaboga & Akay, 2009). A scout bee flies around the search space seeking food sources (solutions); an onlooker bee waits in the nest for the report of the scout bee and an employed bee joins in the harvesting of the food source after watching the waggle dance of the scout bee. A scout bee transforms to an employed bee once it (the same scout bee) gets involved in harvesting the food source. In the ABC, the food source represents a solution to the optimization problem. The volume of nectar in a food source represents the quality (fitness) of the solution (Nozohour-leilabady & Fazelabdolabadi, 2015).

The ABC has been enjoying wide applicability since its design. The application areas includes signal, image and video processing (Akay & Karaboga, 2015), training neural networks, wireless sensor networks, accident diagnosis, reactive power optimization problems, radial distribution, Travelling, Salesman's Problem, (Karaboga, Gorkemli, Ozturk, & Karaboga, 2014) etc.

So far the ABC is very effective in multidimensional search environments due to its capacity to get out of a local minimum with ease. Nonetheless, the ABC uses several parameters that require appropriate tuning in order to get good results. Moreover, the speed of the algorithm is suspect (Karaboga, Akay, & Ozturk, 2007; Karaboga & Basturk, 2007; Karaboga & Ozturk, 2009).

2.9. African Buffalo Optimization

The African Buffalo Optimization (ABO) is a relatively new population-based optimization algorithm designed using the lean metaheuristic concept. The lean design concept from production engineering attempts to achieve the same outcome using as few inputs as possible (Sörensen & Glover, 2013). The ABO draws its inspiration from the movement of buffalos in the African landscapes in search of lush grazing areas. Using two basic vocalization; the /maaa/ and the /waaa/ coupled with their democratic dispositions, the buffalos are able to meander their ways out of starving locations (Odili, Kahar, & Anwar, 2015).

The ABO has been successfully applied to solve the asymmetric and the symmetric Travelling Salesman's Problems (Odili & Mohmad Kahar, 2016b), PID tuning of Automatic Voltage Regulators (Odili & Mohmad Kahar, 2016a), numerical function optimization (Odili & Kahar, 2015b) etc. Though the applications of the ABO are not yet widespread, it has performed well in the problems areas it has been applied to solve with its use of relatively few parameters and straight-forward fitness function.

2.10. Firefly Algorithm

This population-based algorithm, which was designed by Xin-She Yang, drew its inspiration from the flashing attitude of fireflies (Yeomans & Yang, 2014). In the Firefly Algorithm (FFA), a number of fireflies work together to solve a problem through bioluminescent glowing that enables them to efficiently search solutions to optimization problems. The solutions to problems are modelled as a firefly whose flashes are proportional to their quality of solutions; a brighter firefly attracts others colleagues and this aids further exploration of the search space.

So far, the FFA has been applied to Industrial Optimization, Image processing, Business optimization, Civil engineering, Robotics, Semantic web, Chemistry, Meteorology, Antenna design, Wireless sensor networks, Travelling Salesman's Problem etc. with amazing outcomes (Fister, Yang, & Brest, 2013).

A critical look at the FFA reveals that the algorithm is believed to have least error percentage compared to many other metaheuristic algorithms such as the GA and the PSO. Also, it is relatively simple to implement in addition to performing well in multi-modal search environments (Yang, 2012). However, it employs a complicated fitness function and its effectiveness is dependent on correct parameter setting of its many parameters in order to produce good results.

2.11. Bat Algorithm

The Bat Algorithm (BA) was developed by Xin-She Yang. The BA drew its inspiration from the echolocation characteristics of real bats, which use dynamic pulse rates in tracking the food sources. In BA, micro-bats fly randomly over a search space from a given position, deploying a given frequency, loudness and at a specified velocity. These four parameters are judiciously manipulated to arrive at good optimization solutions. In each iteration, the artificial bats locate its prey and adjust their speed, position, frequency and loudness in hunting down the prey. The position of the most successful bat is recorded as the optimum solution from iteration to iteration until the termination condition is reached.

BA has enjoyed fairly wide applications since its development. Some areas of its application include; parameter estimation and classifications, combinatorial optimization problems, data mining, image processing etc. (Ramesh, Mohan, & Reddy, 2013).

The effectiveness of the BA is traceable to its straightforward implementation strategy, simplicity, capacity for quick convergence as well the algorithm's flexibility in switching from exploration to exploitation at the initial stages of the search (Fister, Rauter, Yang, & Ljubič, 2015). In any case, the same switching flexibility is the algorithm's undoing allowing BA to switch to exploitation stage too early results in premature convergence. Moreover, the use of several parameters is another challenge of the BA (Yang & He, 2013).

3. Diagnosis of NAs

NAs have made fantastic contributions to the scientific and engineering communities by helping to solve hither-to-thought-to-be impossible optimization problems in diverse fields of human endeavor ranging from pharmacy to medicine, information technology to industrial design and development, communication to power engineering etc. NAs have been successfully applied to tune proportional, integral and derivative parameters of Automatic Voltage Regulators and DC motors (Rajasekhar, Jatoth, & Abraham, 2014), inventory routing (Wong & Moin, 2015), urban transportation problems (Baridam & Nnamani, 2016), job scheduling (Bakhtar, Jazayeriy, & Valinataj, 2015) etc. However, a critical look at the performance of NAs reveals that they are fraught with several weaknesses ranging from delay in obtaining solutions, ineffectiveness (Kumbharana & Pandey, 2013), use of several

Table 1. Algorithm Diagnosis.

Weaknesses	GA	PSO	ACO	ABC	RAI	FFA	SA	ESA	BA	HBMO	ABC
Low Speed	2	1	2	3	2	2	2	2	2	2	2
Premature convergence	2	1	2	1	2	2	1	2	2	3	1
Several Evaluations	2	1	2	2	1	2	2	3	2	1	2
Ineffectiveness	1	2	1	1	1	2	1	1	1	3	2
Several parameters	3	3	3	3	1	3	1	3	3	1	2
Complicated calculation of fitness	3	1	3	3	3	2	2	1	3	1	2
Complex implementation	3	2	3	3	3	2	3	2	3	1	3

parameters, complicated fitness functions, complex implementation strategies, premature convergence, several evaluations per iteration (Chiong, 2009) etc.

The above listed weaknesses pose such a huge challenge to researchers in their quest for an effective and efficient optimization algorithm to solve a pressing commercial, engineering, industrial or technological optimization at hand. A good knowledge of the strengths and weaknesses of different NAs, therefore, would be an asset to such experts when in need.

A tabular representation of the weaknesses of some of the popular algorithms is presented in Table 1. Low speed is in reference to computational time required by an algorithm in solving optimization problems; premature convergence is a situation where an algorithm gets stagnated at a particular local minimum and thus unable to search out the global minimum. Similarly, several evaluations are a situation where an algorithm embarks on several evaluations of the objective function per iteration; several parameters are a reference to the number of tools deployed by an algorithm in its search for solutions. Also, complicated fitness function is a measure of how simple or otherwise an algorithm determines the fitness of its search candidates/agents. Finally, complex implementation is a measure of how easy or difficult it is to program an algorithm.

In Table 1, the value 3 represents very severe weakness; 2 a moderate weakness and 1 represents a very minor weakness (Chiong, 2009; Kumbharana & Pandey, 2013; von Stryk & Bulirsch, 1992; Yang, 2010).

A brief analysis of the above Table reveals that the PSO remains one of the best optimization algorithms in spite of it being developed over two decades ago. One may, however, trace its robustness, efficiency and effectiveness to the several modifications of the algorithm through the additions of more parameters such as the constriction factor, inertia weight etc. In spite of its success, however, the PSO's present use of several parameters makes the algorithm much less user-friendly in addition to making its implementation more complicated. These are not indications of a good algorithm (Khompatraporn, Pintér, & Zabinsky, 2005).

3.1. Experiments and Discussion of Results

The experiments were performed using MATLAB on Intel Duo Core ™ i7-3770 CPU, 3.40 GHz with 4 GB RAM. The experiments on the asymmetric Travelling Salesman's Problems were executed on a Pentium (R), Duo Core, 1.80 GHz processor and 2 GB RAM desktop computer. The experimental data are available in TSPLIB91 (Reinelt, 1991). The city-coordinates data are available in Reference (Georg, 2008).

For the experiments in this study, Table 2 contains the details of other parameters.

For the sets of experiments involving PSO in this study, the parameters are population size: 200; iteration (T_{max}): 1000; inertia weight: 0.85; C_1:2; C_2: 2 rand1 (0, 1) rand2 (0, 1). For the RAI, a spanning tree is chosen randomly from graph G, and a sub-tour is selected from a tree T. The tree T∈T is taken from a distribution μ defined over T in the graph G. Next, RAI chooses a non-negative λ such that for every edge e ∈ E and the tree T∈T is sampled from μ. This process is repeated until termination condition (Gharan, Saberi, & Singh, 2011).

3.2. Comparative Performance Evaluation

Overall, the findings of this study after a comprehensive review of different optimization algorithms in solving the symmetric Travelling Salesman's Problems, our comparative findings of the performances of major algorithms like the Ant Colony Optimization (ACO) (Deng, Chen, & Li, 2014; Gündüz, Kiran, & Özceylan, 2015; Jia, 2015), Artificial Bee Colony (ABC) (Gündüz et al., 2015), Particle Swarm Optimization (PSO) (Jia, 2015; Yan et al., 2012), African Buffalo Optimization (ABO) (Odili & Kahar, 2015a), Honey Bee Mating Optimization (HBMO) (Marinakis et al., 2011), Genetic Algorithm (GA) (Deng et al., 2014; Yan et al., 2012), Bat Algorithm (BA) (Osaba et al., 2016), discrete Firefly Algorithm (DFA) (Osaba et al., 2016), and the Simulated Annealing (SA) (Arram, Ayob, & Zakree, 2014; Dorigo & Gambardella, 1997) are presented in Table 3.

Table 3 clearly shows that the ACO and the ABO are very competitive in obtaining the optimal or near optimal solutions to the benchmark symmetric TSP test cases under investigation. In spite of their versatility as evidenced by their attempting virtually all the test instances, the ACO and the ABO usually has the best results among the lot. Aside the first two, the other algorithms did pretty well in this order: HBMO, GA, SA, PSO, DFA, BA and ABC respectively.

3.3. Comparative Performance Analysis of NAs on ATSP

Finally, in summarizing the discussion on the asymmetric TSP, it is pertinent to examine the performance of the algorithms in solving all the 18 asymmetric TSP instances in TSPLIB. The comparative algorithms here are the African Buffalo Optimization (ABO) (Odili et al., 2015), the Randomized Insertion Algorithm (RAI) (Brest & Zerovnik, 2005), Bat Algorithm (BA), Genetic Algorithm (GA), discrete Firefly Algorithm (DFA), and the evolutionary Simulated Annealing (ESA) (Osaba et al., 2016). The analyses are presented in Table 4.

Table 4 indicates that the RAI and the ABO have superior performance in solving asymmetric TSP instances. The RAI obtained the best result in 12 instances and the ABO in the remaining 6. RAI's excellent performance could be due to the fact that it is a pure heuristic algorithm developed only to provide solution to asymmetric TSP instances quite unlike the ABO, which is a metaheuristic designed to solve different kinds of optimization problems. It is also worthy to note that the ABO was the second best algorithm in 9 other instances. The next good performers are the DFA, GA, ESA and SA respectively.

Table 2. Experimental Parameters.

ABO		ACO		ABC		BA		FA		SA		ESA		HBMO		GA	
Para	*Val*	*Para*	*Val*	Para	*Val*	Para	Val	Para	Val	To	100	Pop	50	Para	Val	Para	Value
N	40	*ants*	*D**	Pop	D*	N	20	N	D*	*M*	*1*	To	100	Queen	1	Gen	100
rand	*(0, 1)*	*β*	*5.0*	ϕij	*rand(−1,1)*	*A*	*0.9*	*Q*	*200*	*B*	*2*	*α*	*0.95*	*Drones*	*200*	*β*	2.0
lp1	0.6	*ρ*	*0.65*	*ωij*	rand(0, 1.5)	*Rand*	(0,1)	*α*	*0.5*	*α*	*0.9*	*D*	*0.01*	*Sper-mathecal*	*50*	*ρ*	0.1
/lp2	0.5	*α*	*1.0*	*SN*	NP/2	*α*	*3*	*MG*	*1000*	*MIt*	*100*	*MIt*	*100*	*Mating drones*	*50*	Ro	0.33
–	–	*Q*	*200*	*Limit*	D*SN	*γ*	*3*	*γ*	*2*			*M*	*10000*	*Brood*	*50*	Crosso-ver rate	1.0
–	–	*qo*	0.9	*Max Cycle no*	500	*nl*	*2500*	*qo*	*0.9*					*α*	0.9	*qo*	0.9
–	–	–	–	Colony	*50*									*Mating flights*	1000	ϕr	0.3
–	–	–	–	–	–											ϕ *ρ*	0.2
–	–	–	–	–	–									*w1*	3	τ_{min}	$\tau_{max}/20$
–	–	–	–	–	–									w2	*4*	τ_{max}	$1-(1-\rho)$
–	–	–	–	–	–									ϕ	10^{-10}	–	–
–	–	–	–	–	–									–	–	–	–
Run	50		50		50										50		50

D* represents dimension of the optimization problem which, in this case is the number of nodes; Q is the pheromone amount; N is the size of population; *qo* is the exploitation ratio. MG = Maximum Generation; MIt is the Maximum Iteration; M is the Moves; Pop is the Population; Val is the Values; Gen is Generation.

Table 3. Algorithms' Performance Assessment.

TSP instances	Opt	ABO	ACO	ABC	HBMO	PSO	GA	BA	FA	SA	PERFORMANCE ANALYSES
bayg29	1610	1617	1634	–	–	2959	1620	–	–	–	ABO, GA, ACO, PSO
bays29	2020	2121	2020		–	3889	2034	–	–	–	ACO, ABO, GA, PSO
oliver30	420	425	452	–	–	–	423	420	420	424	BA, FA, GA, SA ABO, ACO
att48	33522	33524	33649	–	–	33734	33818	–	–	–	ABO, ACO, PSO, GA
eil51	426	426	426	–	–	427	477	430	426	443	ABO, ACO, FA, PSO, BA, SA, GA
berlin52	7542	7542	7549	9479	7542	7542	7542	7542	7542	–	ABO, ACO, GA, HBMO, PSO, FA, BA, ABC
st70	675	676	696	1162	–	717	710	675	675	–	BA, DFA, ABO, ACO, GA, PSO, ABC
eil76	538	538	559	877	538	546	641	538	543	580	ABO, BA, GA, FA, PSO, ACO, ABC
pr76	108159	108167	110917	195198	–	118028	115329	–	–	–	ABO, ACO, GA, PSO, ABC
rat99	1211	1211	1236	–		1278	1269	–	–	–	ABO, ACO, GA, PSO,
rd100	7910	7935	–	–	7910	–	8506	–	–	–	HBMO, ABO, GA
kroA100	21282	21554	22455	49519	–	21310	21350	2 1292	21282	–	DFA, BA, PSO, GA, ABO, ACO, ABC
kroB100	22141	22160	23106	–	–	133081	22176	22373	22183	–	ABO, GA, DFA, GA, ACO, PSO
kroC100	20749	20755	21569	–	20749	–	20861	20802	20756	–	HBMO, ABO, DFA, BA, GA, ACO
kroD100	21294	21347	22837	–	21294	–	21492	21727	21408	–	HBMO, ABO, DFA, GA, BA, ACO
kroE100	22068	22088	23529		22068	–	22150	22323	22079	–	HBMO, DFA, ABO, GA, BA, ACO
Eil101	629	640	678	1237	629	–	655	640	643	–	HBMO, ABO, BA, DFA, GA, ABC
lin105	14379	14419	14426	–	–	–	–	–	–	–	ABO. ACO
pr107	44303	44407	44354	–	–	44436	44417	44618	44303	–	DFA, ACO, ABO, GA, PSO, BA
pr124	59030	59058	9113	–	–	59283	59247	59030	59030	–	BA, DFA, ABO, ACO, GA, PSO
ch130	6110	6111	6141	6648	–	6181	6158	–	–	–	ABO, ACO, GA, PSO, ABC
pr136	96772	97684	96785	–	–	–	–	100485	97716	–	ACO, ABO, DFA, 100485
pr144	58537	58587	58822	–	–	–	–	58588	58546	–	DFA, ABO, BA, ACO
pr152	73682	73730	73835	–	–	73898	73872	74172	74033	–	ABO, ACO, GA, PSO, DFA, BA

U159	42080	42107	42080	–	–	–	42834	–	–	–	ACO, ABO, GA
kroA200	29368	29370	29511	–	–	29968	29868	–	–	–	ABO, ACO, GA, PSO
kroB200	29437	29487	32073	–	–	–	–	–	–	–	ABO, ACO
tsp225	3916	3917	4112	16998		–	–	–	–	–	ABO, ACO, ABC
gil262	2378	2378	2647	–	–	2513	4569	–	–	–	ABO, PSO, ACO, GA
pr299	48191	48211	48400	–	–	48540	48739	49142	48579	–	ABO, ACO, PSO, DFA, GA, BA
lin318	42029	42101	46563	–	–	–	–	–	–	–	ABO, ACO
a280	2579	2579	3092	–	–	–	3218	–	–	–	ABO, ACO, GA
rd400	15281	15301	15461	–	–	16964	15764	–	–	–	ABO, ACO, GA, PSO
pr439	107217	107340	11618	–	–	1674258	113576	–	115538	–	ABO, GA, DFA, ACO, PSO
pcb442	50778	50799	58407	–	–	700718	305671	–	–	–	ABO, ACO, PSO, GA
ali535	202310	202420	–		–	–	242310	231120	–	–	ABO, BA, GA
rat575	6773	6777	6876	–	–	6910	6897	–	–	–	ABO, ACO, GA, PSO
fl1577	22249	22249	25589	–	–	1276106	25856	–	–	–	ABO, ACO, GA, PSO
d657	48912	49055	–	–	–	50612	50912	–	–	–	ABO, PSO, GA
rat783	8806	8811	8998	–	–	8905	8965	–	–	10019	ABO, PSO, GA, ACO, SA
dsj1000	18659688	186807714	–	–	18660556	–	–	–	–	–	HBMO, ABO
pr1002	259045	259132	264141	–	–	278923	265669	–	–	277344	ABO, ACO, GA, DFA, SA, PSO
u1060	224094	224481	273776	–	–	375284	–	–	–	248891	ABO, SA, ACO, GA
vm1084	232929	232931	–	–	–	268808	279094	–	–	–	ABO, PSO, GA
d1291	50801	50839	52942	–	–	53912	–	–	–	–	ABO, ACO, PSO
rl1323	270199	270480	307021	–	–	313524	–	–	–	–	ABO, ACO, GA
nrw1379	56638	56653	67578	–	–	68521	–	–	–	–	ABO, ACO, GA
fl1400	20127	20134	–	–	–	–	24251	–	–	22490	ABO, SA, GA
f1417	11861	11862	–	–	–	–	118756	–	–	–	ABO, GA
u1432	152970	152970	–	–	62149	177890	182780	–	–	–	ABO, PSO, GA
d1655	62128	62346	–	–		–	–	–	–	–	HBMO, ABO
d2103	80450	80456	85265	–	80450	3075609	85514	–	–	–	HBMO, ABO, ACO, GA, PSO
pcb3038	137694	137700	141095	–		–	143341	–	–	140130	ABO, SA, ACO, GA
fl3795	28772	28772	–	–	28783	–	–	–	–	31124	ABO, HBMO, SA
rl5915	565530	565800	–	–	565530	–	–	–	–	–	HBMO, ABO
rl5934	556045	556078	–	–	556080	–	–	–	–	–	ABO, HBMO
pla7379	23260728	23268269	–	–	23261400	–	–	–	–	–	HBMO, ABO
usa13509	19982859	19993952	–	–	19982859	–	–	–	–	–	HBMO, ABO
brd14051	469385	469835	476949	–	–	477346	477304	–	–	–	ABO, ACO, GA, PSO
fn14461	182566	182745	192964	–	–	199314	–		–	–	ABO, ACO, PSO

TSP instances = benchmark TSP cases available in TSPLIB; Opt = optimal solutions; the other columns indicate the best results of the comparative algorithms; Performance Analyses = ranks the performances according their optimal results

Table 4. Comparative Performance Analyses.

ATSPCases	Opt	ABO	RAI	BA	GA	FA	ESA	PERFORMANCE ANALYSIS
Br17	39	39	39	39	39	39	39	ABO RAI, BA, GA, DFA, ESA
Ry48p	14422	14440	14422	14790	14545	14453	14485	RAI, ABO, DFA, ESA, GA, BA
Ftv33	1286	1287	1286	1348	1290	1286	1286	RAI, DFA, BA, ABO, GA, BA
Ftv35	1473	1474	1473	1490	1490	1498	1473	RAI, ESA, ABO, BA, GA, DFA
Ftv38	1530	1530	1530	1530	1565	1560	1530	ABO, RAI, BA, ESA, DFA, GA
Ftv44	1613	1614	1613	1735	1649	1690	1645	RAI, ABO, ESA, GA, DFA, BA
Ftv47	1776	1777	1776	1842	1820	1795	1795	RAI, ABO, DFA, ESA, GA, BA
Ft53	6905	6905	6905	7105	7270	6993	6990	ABO, RAI, ESA, DFA, BA, GA
Ftv55	1608	1610	1608	1686	1700	1628	1725	RAI, ABO, DFA, BA, GA, ESA
Ftv64	1839	1839	1839	2068	2014	1903	1955	ABO, RAI, DFA, ESA, GA, BA
P43	5620	5645	5620	5632	5620	5620	5620	RAI, GA, DFA, ESA, BA, ABO
Rbg323	1326	1326	1335	1713	1514	1599	1620	ABO, RAI, GA, DFA, ESA, BA
Ftv70	1950	1955	1950	2238	2184	2173	2200	RAI, ABO, DFA, GA, ESA, BA
Ft70	38673	38673	38855	2238	39407	39668	39650	ABO, RAI, GA, ESA, DFA, BA
Kro124p	36230	36275	36241	40070	39325	39438	40019	RAI, ABO, GA, DFA, ESA, BA
Rbg358	1163	1187	1166	–	–	–	–	RAI, ABO - - - -
Rbg403	2465	2467	2465	–	–	–	–	RAI, ABO - - - -
Rbg443	2720	2723	2720	–	–	–	–	RAI, ABO - - - -

TSP instances = benchmark TSP cases available in TSPLIB; Opt = optimal solutions; the other columns indicate the best results of the comparative algorithms; Performance Analyses = ranks the performances according their optimal results

Nonetheless, the BA's performance in this analysis is suspect; the algorithm performed extremely poorly in 9 out of the 15 instances it attempted. In fact, the two worst performers are the ABC and the BA respectively. This could be traceable to both algorithms' use of several parameters and complicated fitness function, which is consistent with the Frankenstein phenomena (Sorensen & Glover, 2013). Frankenstein phenomena refer to a situation where optimization search algorithms use several parameters to the extent that the it is difficult to pinpoint the exact contribution of individual parameters.

4. Conclusions

Having established that the knowledge of TSP is useful in decision-making in different spheres of human endeavor, especially in product and personnel routing, resources scheduling problems and management decision making etc., this study carried out an analysis of several NAs with the overall aim of ascertaining a more effective NA for solving the benchmark symmetric TSP as well as the asymmetric TSP in TSPLIB95 as it relates to strategic management. In all, experimental performances of 11 NAs in solving a total of 60 benchmark symmetric TSP and all of the 18 asymmetric and asymmetric test cases were examined. After exhaustive analysis, it was discovered that the ABO and the ACO were very effective in searching out the optimal or near-optimal solutions to all the solving the symmetric TSP, which is similar to intelligence gathering channel in the strategic management of big organizations. The other good performers are the HBMO, GA, SA, PSO, DFA, BA and ABC respectively.

Also, the investigation on the asymmetric TSP reveals that the RAI holds the best promise in asymmetric TSP instances akin to strategic information exchange channels in strategic management. The RAI's good performance is closely followed by the ABO. The third best performer is the discrete FFA, then the GA and the ESA. The least performer is the BA. The RAI's performance could be due to the fact that it is a heuristic designed primarily to solve the asymmetric TSP. The ABO has provided interesting outcome in these two benchmark problems, probably due to its use of just three main parameters coupled with a straight-forward calculation of fitness function in its quest for solutions.

The performance of the FFA is also very interesting in spite of its use of several parameters. The good performance of FFA is in tandem line with previous findings that the algorithm performs very well in multidimensional search environments and has one of the least error performances in its search efforts (Yang, 2012). This is a mark of a good algorithm (Khompatraporn, Pintér, & Zabinsky, 2005). The GA has always been very competitive in diverse search spaces, thus making the algorithm very popular among researchers in spite of it being several decades old. Similarly, the BA was not very competitive here, possibly due to the reasons adduced above. It is hoped that further researches be made on making the BA more effective.

In all, having investigated the potentials of the different methods, we recommend the copious use of the best performing algorithms to researchers and practitioners involved in technological decision management in various engineering, corporate and scientific establishments whenever they are confronted with procedural optimization problems. Also, we recommend the practical applications of the algorithms investigated in this study to solving other technology management related problems.

Acknowledgement

The authors appreciate the Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, Kuantan for funding this research under Grant GRS 1403118. Also, much thanks to the Ministry of Higher Education Malaysia for additional funding under the project FRGS-2-2013-ICT07-UniSZA-0201 and FRGS, RDU140101. Moreover, our gratitude to Dr. Shafiq Kamarulzaman and Research Management, Innovation and Commercialization Center (RMCC, UniSZA) for their support.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by Universiti Malaysia Pahang, Kuantan [grant number GRS 1403118]; Ministry of Higher Education Malaysia [project number FRGS-2-2013-ICT07-UniSZA-0201, FRGS [project number RDU140101].

Notes on contributors

Julius Beneoluchi Odili was born in Ogume in Delta State, Nigeria. He holds a Bachelor of Arts degree in English Language and Education from the Bendel State University, Ekpoma, Nigeria, Postgraduate Diploma in Computer Science of the Delta State University, Abraka, Nigeria, a Master's of Education in Educational Administration of the University of Lagos, Nigeria, a Master's of Science degree in Computer Science of the University of Lagos, Nigeria and a Ph.D. in Computer Science of the Universiti Malaysia Pahang, Kuantan, Malaysia. His research interests are artificial intelligence, software development, metaheuristics and green computing.

Mohd Nizam Mohmad Kahar received a Bachelor of Engineering degree in Computer Engineering and a Master's of Science degree in Mathematics from the Universiti Teknologi Malaysia. He holds a Ph.D. in Computer Science of the University of Nottingham, UK. He is presently a Senior Lecturer in Computer Science in the Universiti Malaysia, Pahang, Kuantan, Malaysia. His research interests include optimization algorithms and computer networking.

A. Noraziah is an associate professor in the Faculty of Computer Systems and Software Engineering Dept., Universiti Malaysia Pahang, Kuantan, Malaysia. She holds a BSc from the Universiti Putra Malaysia; Certificate in Teaching and Learning from The Northern Illinois University, USA and a PhD from the Universiti Malaysia Terengganu. Her research interests are algorithms, distributed database, data grid, distributed systems, cloud computing and big data.

M. Zarina is a senior lecturer in the Faculty of Informatics and Computing Dept., Universiti Sultan Zainal Abidin, Gong Badak, Terengganu, Malaysia. She holds a BSc from the Universiti Putra Malaysia and PhD from the Universiti Tun Hussien Onn Malaysia. Her research interests are distributed database, data grid, distributed systems, cloud computing and big data

Riaz Ul Haq was born in Khyber Pakhtunkhwa State, Pakistan. He received a Bachelor of Computer Science from Karakoram International University, Northern Areas, Pakistan. He did his Master's in Computer Science from Universiti Malaysia Pahang, Malaysia. He is currently a Ph.D. scholar in the same faculty. His research interests are in wireless ADHOC networking, artificial intelligence, and E-learning.

References

Akay, B., & Karaboga, D. (2015). A survey on the applications of artificial bee colony in signal, image, and video processing. *Signal, Image and Video Processing, 9*, 967–990.

Anwar, I.M., Salama, K.M., & Abdelbar, A.M. (2015). Instance selection with ant colony optimization. *Procedia Computer Science, 53*, 248–256.

Applegate, D.L., Bixby, R.E., Chvatal, V., & Cook, W.J. (2011). *The traveling salesman problem: A computational study.* Princeton university press.

Arram, A., Ayob, M., & Zakree, M. (2014). Comparative study of meta-heuristic approaches for solving traveling salesman problems. *Asian Journal of Applied Sciences, 7*, 662–670.

Aydin, M.E., & Fogarty, T.C. (2004). A distributed evolutionary simulated annealing algorithm for combinatorial optimization problems. *Journal of Heuristics, 10*, 269–292.

Bakhtar, S., Jazayeriy, H., & Valinataj, M. (2015). A multi-start pathrelinking algorithm for the flexible job-shop scheduling problem. Paper presented at the 7th Conference on Information and Knowledge Technology.

Baridam, B.B., & Nnamani, C.L. (2016). Application of particle swarm optimization to solve transportation problem. *International Journal of Computer Science and Information Security, 14* (1), 31–45.

Barry, M.J., & Edgman-Levitan, S. (2012). Shared decision making — The pinnacle of patient-centered care. *New England Journal of Medicine, 366*, 780–781.

Bilbao, M., & Alba, E. (2009). *Simulated annealing for optimization of wind farm annual profit.* Paper presented at the Logistics and Industrial Informatics, 2009. LINDI 2009. 2nd International Logistics and Industrial.

Brest, J., & Zerovnik, J. (2005). A heuristic for the asymmetric traveling salesman problem. Paper presented at the 6th Metaheuristics International Conference, Vienna Austria, 145–150.

Chang, C.-C. (2015). A 2-Opt with mutation operator to the traveling salesman problem. *International Journal of Advancement in Engineering, Technology and Computer Sciences, 2*, 16–21.

Chiong, R. (2009). *Nature-inspired algorithms for optimization*, 193, 1–50. Springer.

Dasgupta, D., & Michalewicz, Z. (2013). *Evolutionary algorithms in engineering applications.* Springer Science & Business Media. Retrieved from www.books.google.com.ng.

David, F., & David, F.R. (2016). Strategic management: A competitive advantage approach, concepts and cases www.dl.icdst.org.

Deng, W., Chen, H., & Li, H. (2014). A novel hybrid intelligence algorithm for solving combinatorial optimization problems. *JCSE, 8*, 199–206.

Dorigo, M., & Gambardella, L.M. (1997). Ant colonies for the travelling salesman problem. *Biosystems, 43*, 73–81.

Fister, I., Rauter, S., Yang, X.-S., & Ljubič, K. (2015). Planning the sports training sessions with the bat algorithm. *Neurocomputing, 149*, 993–1002.

Fister, I., Yang, X.-S., & Brest, J. (2013). A comprehensive review of firefly algorithms. *Swarm and Evolutionary Computation, 13*, 34–46.

Freeman, R. E. (2010). *Strategic management.* Cambridge: Cambridge University Press.

Georg, S. (2008). MP-TESTDATA-The TSPLIB symmetric traveling salesman problem instances. Retrieved from www.elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/.

Gharan, S.O., Saberi, A., & Singh, M. (2011). *A randomized rounding approach to the traveling salesman problem.* Paper presented at the IEEE 52nd Annual Symposium on Foundations of Computer Science.

Gündüz, M., Kiran, M.S., & Özceylan, E. (2015). A hierarchic approach based on swarm intelligence to solve the traveling salesman problem. *Turkish Journal of Electrical Engineering & Computer Sciences, 23*, 103–117.

Gutjahr, W.J. (2003). A converging ACO algorithm for stochastic combinatorial optimization. In *Stochastic algorithms: Foundations and applications* (pp. 10-25). Springer.

Hénard, F., Diamond, L., & Roseveare, D. (2012). Approaches to internationalization and their implications for strategic management and institutional practice. *IMHE Institutional Management in Higher Education, 11*(12), 2013. Retrieved from http://www.oecd.org/edu/imhe/Approaches%20to%20internationalisation%20-%20final%20-%20web.pdf

Jia, H. (2015). A novel hybrid optimization algorithm and its application in solving complex problem. *International Journal of Hybrid Information Technology, 8*(2), 1–10.

Kant, A., Sharma, A., Agarwal, S., & Chandra, S. (2010). An ACO approach to job scheduling in grid environment. In *Swarm, evolutionary, and memetic computing* (pp. 286–295). Springer.

Karaboga, D., & Akay, B. (2009). A survey: Algorithms simulating bee swarm intelligence. *Artificial Intelligence Review, 31*, 61–85.

Karaboga, D., Akay, B., & Ozturk, C. (2007). Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks. In *Modeling decisions for artificial intelligence* (pp. 318–329). Springer.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*, 459–471.

Karaboga, D., Gorkemli, B., Ozturk, C., & Karaboga, N. (2014). A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review, 42*, 21–57.

Karaboga, D., & Ozturk, C. (2009). Neural networks training by artificial bee colony algorithm on pattern classification. *Neural Network World, 19* (3), 279–292.

Karagul, K., Aydemir, E., & Tokat, S. (2016). Using 2-Opt based evolution strategy for travelling salesman problem. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA), 6*, 103–113.

Kefi, S., Rokbani, N., Krömer, P., & Alimi, A.M. (2015). *A new ant supervised-PSO variant applied to traveling salesman problem*. Paper presented at the Hybrid Intelligent Systems: 15th International Conference HIS 2015 on Hybrid Intelligent Systems, Seoul, South Korea, November 16–18.

Kennedy, J. (2010). Particle swarm optimization. In *Encyclopedia of Machine Learning* (pp. 760–766). Springer.

Khompatraporn, C., Pintér, J.D., & Zabinsky, Z.B. (2005). Comparative assessment of algorithms and software for global optimization. *Journal of Global Optimization, 31*, 613–633.

Kirkpatrick, S., Gelatt, C.D., & Vecchi, M.P. (1983). Optimization by simulated annealing. *Science, 220*, 671–680.

Kolonko, M. (1999). Some new results on simulated annealing applied to the job shop scheduling problem. *European Journal of Operational Research, 113*, 123–136.

Kumbharana, N., & Pandey, G.M. (2013). A comparative study of ACO, GA and SA for solving travelling salesman problem. *International Journal of Societal Applications of Computer Science, 2*, 224–228.

Ledesma, S., Aviña, G., & Sanchez, R. (2008). Practical considerations for simulated annealing implementation. *Simulated Annealing, 20*, 401–420.

Ma, J., Wang, X., Zhu, D., & Zhou, X. (2015). Analysis on patent collaborative patterns for emerging technologies: A case study of nano-enabled drug delivery. *International Journal of Technology Management, 69*, 210–228.

Malek, M., Guruswamy, M., Pandya, M., & Owens, H. (1989). Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem. *Annals of Operations Research, 21*, 59–84.

Man, K.-F., Tang, K.S., & Kwong, S. (2012). Genetic algorithms: Concepts and designs. Springer Science & Business Media. Retrieved from www.books.google.com

Marinakis, Y., Marinaki, M., & Dounias, G. (2011). Honey bees mating optimization algorithm for the Euclidean traveling salesman problem. *Information Sciences, 181*, 4684–4698.

Mininno, E., & Neri, F. (2010). A memetic differential evolution approach in noisy optimization. *Memetic Computing, 2*, 111–135.

Mirjalili, S., Mirjalili, S.M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61.

Mustafa, M., Al Gizi, A.J., & Alsaedi, M.A. (2012). Adaptive PID controller based on real base function network identification, and genetic algorithm in automatic voltage regulator system. *International Journal of Science of Research Publications, 2*(11), 1–11.

Nozohour-leilabady, B., & Fazelabdolabadi, B. (2015). On the application of Artificial Bee Colony (ABC) algorithm for optimization of well placements in fractured reservoirs; efficiency comparison with the Particle Swarm Optimization (PSO) methodology. *Petroleum*.

Odili, J. B. (2013). Application of ant colony optimization to solving the traveling salesman's problem. *Science Journal of Electrical & Electronic Engineering, 2013*, 175–177.

Odili, J.B., Deng, H., Yang, C., Sun, Y., Santo Ramos, M.C.d.E., Ramos, M.C.G., … Loh, P.K. (2013). Application of ant colony optimization to solving the traveling salesman's problem. *Science Journal of Electrical & Electronic Engineering.*

Odili, J.B., & Kahar, M.N.M. (2015a). African buffalo optimization (ABO): A new meta-heuristic algorithm. *Journal of Advanced & Applied Sciences, 3*, 101–106.

Odili, J.B., & Kahar, M.N.M. (2015b). Numerical function optimization solutions using the african buffalo optimization algorithm. *British Journal of Mathematics & Computer Science, 10*(1), 1–12.

Odili, J.B., & Kahar, M.N.M. (2016). African buffalo optimization. *International Journal of Software Engineering & Computer Systems, 2*, 28–50. doi:10.15282/ijsecs.2.2016.1.0014

Odili, J.B., Kahar, M.N.M., & Anwar, S. (2015). African buffalo optimization: A swarm-intelligence technique. *Procedia Computer Science* (IEEE-IRIS 2015).

Odili, J.B., Kahar, M.N.M., Anwar, S., & Azrag, M.A.K. (2015). *A comparative study of African buffalo optimization and randomized insertion algorithm for asymmetric travelling salesman's problem.* Paper presented at the Software Engineering and Computer Systems (ICSECS), 2015 4th International Conference on.

Odili, J.B., & Mohmad Kahar, M.N. (2016a). African buffalo optimization approach to the design of PID controller in automatic voltage regulator system. National Conference for Postgraduate Research Universiti Malaysia Pahang.

Odili, J. B. & Mohmad Kahar, M. N. (2016b). Solving the traveling salesman's problem using the African buffalo optimization. *Computational Intelligence and Neuroscience, 2016*, 1–12.

Osaba, E., Yang, X.-S., Diaz, F., Lopez-Garcia, P., & Carballedo, R. (2016). An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. *Engineering Applications of Artificial Intelligence, 48*, 59–71.

Poli, R. (2007). *An analysis of publications on particle swarm optimization applications*. Essex, UK: Department of Computer Science, University of Essex.

Rajasekhar, A., Jatoth, R.K., & Abraham, A. (2014). Design of intelligent PID/PI λ D μ speed controller for chopper fed DC motor drive using opposition based artificial bee colony algorithm. *Engineering Applications of Artificial Intelligence, 29*, 13–32.

Ramesh, B., Mohan, V.C.J., & Reddy, V.V. (2013). Application of bat algorithm for combined economic load and emission dispatch. *International Journal of Electrical Engineering and Telecommunications, 2*(1), 1–9.

Reinelt, G. (1991). TSPLIB—A traveling salesman problem library. *ORSA Journal on Computing, 3*, 376–384.

Shi, Y., & Eberhart, R.C. (1999). Empirical study of particle swarm optimization. Paper presented at the Congress on Evolutionary Computation.

Shrivastava, P., Ivanaj, S., & Ivanaj, V. (2016). Strategic technological innovation for sustainable development. *International Journal of Technology Management, 70*, 76–107.

Sörensen, K. & Glover, F. W. (2013). Metaheuristics. *Encyclopedia of Operations Research and Management Science, 62*, 960–970.

Srivastava, P.R., & Kim, T.-H. (2009). Application of genetic algorithm in software testing. *International Journal of software Engineering and its Applications, 3*, 87–96.

Stützle, T., López-Ibáñez, M., & Dorigo, M. (2011). A concise overview of applications of ant colony optimization. Wiley Encyclopedia of Operations Research and Management Science. Retrieved from www.onlinelibrary.wiley.com

Tambouratzis, T. (1997). A simulated annealing artificial neural network implementation of the n-queens problem. *International Journal of Intelligent Systems, 12*, 739–751.

Tanweer, M., Suresh, S., & Sundararajan, N. (2015). Improved SRPSO algorithm for solving CEC 2015 computationally expensive numerical optimization problems. Paper presented at the IEEE Congress oo Evolutionary Computation.

Tremblay, D.-G., & Dossou-Yovo, A. (2015). Territory, innovation processes in SMEs, and intermediary actors: The case of the ICT sector in the greater montreal area. *International Journal of Technology Management, 69*(1), 1–19.

Uz, M.G., Kiran, M.S., & Özceylan, E. (2015). A hierarchic approach based on swarm intelligence to solve the traveling salesman problem. *Turkish Journal of Electrical Engineering & Computer Sciences, 23*, 103–117.

van Laarhoven, P.J., Aarts, E.H., & Lenstra, J.K. (1992). Job shop scheduling by simulated annealing. *Operations Research, 40*, 113–125.

von Stryk, O., & Bulirsch, R. (1992). Direct and indirect methods for trajectory optimization. *Annals of Operations Research, 37*, 357–373.

Wong, L., & Moin, N.H. (2015). *Enhanced ant colony optimization for inventory routing problem.* Paper presented at the the 22nd National Symposium on Mathematical Sciences (SKSM22): Strengthening Research and Collaboration of Mathematical Sciences in Malaysia.

Wu, Y., Xin, Y., & Zhang, Y. (2015). Application of ACO to vehicle routing problems using three strategies. International Conference on Logistics Engineering, Management and Computer Science.

Yan, X., Zhang, C., Luo, W., Li, W., Chen, W., & Liu, H. (2012). Solve traveling salesman problem using particle swarm optimization algorithm. *International Journal of Computer Science, 9*, 264–271.

Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization. Retrieved from link.springer.com/chapter/10.1007%2F978-3-642-12538-6_6

Yang, X.-S. (2012). Nature-inspired mateheuristic algorithms: Success and new challenges. arXiv. *preprint, 1*(1), 1–3.

Yang, X.-S., & He, X. (2013). Bat algorithm: Literature review and applications. *International Journal of Bio-Inspired Computation, 5*, 141–149.

Yeomans, J.S., & Yang, X.-S. (2014). Municipal waste management optimization using a firefly algorithm-driven simulation-optimization approach. *International Journal of Process Management and Benchmarking, 4*, 363–375.

Zeb, A., Khan, M., Khan, N., Tariq, A., Ali, L., Azam, F., & Jaffery, S.H.I. (2016). Hybridization of simulated annealing with genetic algorithm for cell formation problem. *The International Journal of Advanced Manufacturing Technology, 86*(5), 2243–2254.