International Journal of Software Engineering and Computer Systems (IJSECS)
ISSN: 2289-8522, Volume 4 Issue 2, pp. 49-61, August 2018
©Universiti Malaysia Pahang
DOI: https://doi.org/10.15282/ijsecs.4.2.2018.4.0048

THE DAWN OF METAHEURISTIC ALGORITHMS

Odili, J.B
Department of Mathematical Sciences, Faculty of Natural and Applied Sciences, Anchor University, Lagos, Nigeria

Abstract: Optimization has become such a favored area of research in recent times necessitating the need for technical papers and tutorials that will properly analyze and explain the basics of the field. At the heart of efficiency and effectiveness of optimization of engineering, business and industrial processes is metaheuristics, hence the need for proper explanations of the basics of optimization algorithms since the optimization algorithms are the engine room of successful optimization enterprise. This paper presents a foundational discussion on metaheuristic algorithms as a necessary ingredient in successful optimization endeavors and concludes, after analysis of some metaheuristic algorithms that a good metaheuristic algorithm should consist of four components, namely global search, local search, randomization and identification of the best solution at each iteration.

Keywords: Algorithms, Efficiency, Effectiveness, Metaheuristics, Optimization

1. INTRODUCTION

To say that optimization is at the center of many industrial and technological breakthroughs, the world over is not an overstatement. Optimization, fundamentally, is concerned with the search for greater efficiency and effectiveness in industrial, business, engineering, decision-making and manufacturing concerns (Ahmadizar & Soltanpanah, 2011) through the identification and choice of the most cost-effective procedure. Optimization which has been defined as the economics of computer science is concerned with the efficient management of systems and resources in order to achieve a desired end (Julius Beneoluchi Odili, Mohd Nizam Mohmad Kahar, A. Noraziah, Zarina, & Haq, 2017). Optimization has to do with the search for the optimum means of achieving an end in the midst of several means (Faludi, 2013; Odili & Noraziah, 2018).

Basically, optimization involves the maximization or minimization of a function by systematically choosing some input values within an allowable set of input values in order to compute the value of the function with the aim of determining the best values of the objective function (Odili, 2018). The overall aim of optimization is to ensure greater efficiency through the use of less resource to achieve the most-desirable outcome. This most desirable outcome could be the minimization of input needed to achieve an end or the maximization of profits. A computer program, for instance could be optimized to use less memory, execute faster or utilize less resources. Similarly, industrial systems, procedures and processes could be optimized to yield maximum output from as little as possible resources. Therefore, it may be safe to assert that optimization has relevance in situations where there exist a need to maximize output cum profits while minimizing the cost/ input (Odili & Kahar, 2016). One can hardly imagine situations in any industrial, engineering cum business concern where such objective is nonexistent.

In as much as the overall aim of any optimization procedure is to ensure optimal use of available resources, it should also be noted that, in most cases, this ideal comes at a cost. For instance, a computer program may execute faster and obtain more effectiveness, probably due to its use of more computer memory and vice-versa. Overall, therefore, there is need to design some algorithms that will ensure better trade-off between the different constraints in an optimization procedure. Over time, it has been established that metaheuristic algorithms has proven to be very effective and efficient in optimization procedures. This is evident in several successful applications of metaheuristic algorithms. Some of the successful applications include the travelling salesman’s problem, global optimization, tuning of PID parameters of Automatic Voltage Regulators (Odili, Kahar, & Noraziah, 2017), job scheduling, examination time-tabling scheduling (McCollum et al., 2010) etc. In the light of the above, this paper, which is a technical position paper, aims to bring to fore the essentials of a good optimization algorithm such that novel researchers can easily identify a possibly good metaheuristic algorithm that can obtain competitive results in a given optimization problem

The rest of this paper is structured as follows: section two discusses optimization algorithms; section three examines the concept of heuristics and metaheuristics; section four is concerned with the essential characteristics of metaheuristic algorithms and section five draws conclusions on the study.

2. OPTIMIZATION ALGORITHMS

In view of the enormous contributions of optimization algorithms to enhancing industrial, business, decision-making and engineering processes, a few exact algorithms, popularly called deterministic or traditional algorithms have been developed. Examples of such exact algorithms include finite volume methods (Said & Wegman, 2009), Linear Programming (LP) (Kuhn, 2014), Newton-Raphson (Wooldridge, 2010) Dynamic Programming (Sniedovich, 2010), finite elements (Hughes, 2012) etc.

Exact algorithms, otherwise called deterministic algorithms operate without the use of stochastic (probabilistic or random) elements (Motwani & Raghavan, 2010). This implies that with a given set of input data, these algorithms will produce the same output values. Similarly, given the same input data the computer’s back-end will likely use the same sequence of states (Kornblum, 2006).

On the other hand, stochastic (probabilistic) algorithms make use of some inbuilt randomness. This use of randomness means that given the same set of input data and initial conditions probabilistic algorithms may generate different outputs in each iteration until they home in at a final solution (Gentle, 2013; Machairas, Tsangrassoulis, & Axarli, 2014). In spite this, it is surprising, however, that stochastic models have proven to be very successful in large problems with several input parameters and operating conditions. As a result, many of newly-developed metaheuristics that draw their inspiration from the consistent, harmonious and self-organized systems in nature have also been developed with probabilistic components. These modern set of algorithms are classified as Natural Computing (Odili, Kahar, & Noraziah, 2018; Păun, 2012).

Natural Computing refers to such algorithms that simply use computers to extract relatively-common but complex ideas from nature to develop computational systems or use natural materials such as molecules to perform computation. From this explanation, it is clear that natural computing can be drawing direct inspiration from nature, sometimes called Nature-Inspired Computing (NIC) or simply computing with natural materials (CWN) (Dodig-Crnkovic, 2012). Computing with natural materials is one of the most recent innovations in computing approaches. Here algorithm developers, in place of silicon, make use of natural elements as software and hardware computational tools. (Zang, Zhang, & Hapeshi, 2010).

It is pertinent to note that the unprecedented popularity of NIC algorithms in and engineering, industrial and scientific researches all over the world in the past few decades has attracted the attention of many scientists. The primary reason given for this popularity is that these algorithms are developed to simulate the most successful natural dynamics in chemical, biological and physical processes in nature (Rozenberg, Bck, & Kok, 2011). This increasing popularity cum demand for NIC algorithms throws up the issue of choice of algorithm (since we now have so many of them) whenever a researcher has an optimization problem to solve. The eventual choice of a particular algorithm in solving a particular is, therefore, dependent on the capacity of that particular technique to solve the given problem. This scenario can be said to be have given impetus to the No free-lunch theorems for optimization ((Yang, 2011).

2.1 Format of Metaheuristic Algorithms

In general, most metaheuristic optimization algorithms have this format:

$$Minimize\ \ f_i(x)\ (i\ =\ 1,2,3,\ldots,M),\qquad x\ \in\ \Re n \tag{1}$$

$$subject\ to\ \ h_a(x) =\ 0,\qquad (a\ =\ 1,2,3\ \ldots,N), \tag{2}$$

$$g_b(x) \leq\ 0,\qquad (b\ =\ 1,2,3,..,K) \tag{3}$$

$where\ f_i(x), h_a(x)\ and\ \ g_b(x)$ are functions of the design vectors.

$$xiL\ \leq\ \mathrm{xi}\ \leq\ \mathrm{xiU}\ \ \mathrm{i}\ =\ 1,\mathrm{N} \tag{4}$$

From the above equations, the function $f_i(x)$ where $i\ =\ 1,2,\ldots,M$ is called the cost (goal or objective) function. The cost function could be designed as a minimization or maximization problem depending on the interest of the designer. Please note that an optimization problem could have the objective of either minimizing a function or maximizing it. If the objective is to increase the profit margin of an organization, then the easier thing to do is to formulate the cost function as a maximization problem. If the reverse (that is, to reduce the input values needed to achieve a particular goal), then it is easier formulated as a minimization problem.

Furthermore, in a situation where $M = 1$, then it is an instance of single objective function; in a situation where $M\ \geq 2$, that is a multi-objective problem. Moreover, the variable $x(i)$ of x is called the design or decision variable and it could be continuous, discrete or a mixture of both, otherwise called mixed decision variable (Feist & Palsson, 2010). The space covered by the decision variables is called the search space $\in\ \Re n$. Similarly, the space covered by the objective function is called the solution space. In the same vein, $h_a\ and\ g_b$ are the equality and inequality constraints respectively.

It should be observed that, as the name implies, equality constraints take the form of $= 0$, while the inequality constraints could be in the form of $\geq\ 0$ when it is a maximization problem or $\leq\ 0$ in which case, it is a minimization problem. Also, please note that the searchable design space usually contains by the lower and upper bounds, xiL and xiU , of the design or decision variables, otherwise referred to as the side constraints.

In general, objective (goal or cost) function can be formulated to be nonlinear or linear, explicit or implicit. Optimization problems that have some all or of the decision variables to be integer or discrete values are called integer or discrete optimization problems. Even though both the deterministic and the stochastic optimization techniques employ similar format, most times, traditional optimization techniques encounter a lot of difficulties solving discrete or integer optimization problems. This is usually the area of strength of the stochastic algorithms (Venter, 2010).

2.2 Traditional algorithms

Traditional Optimization techniques such as the Newton-Raphson and Simplex Method use the gradient-based approach and are usually deterministic (Davoodi, Hagh, & Zadeh, 2014). They have proven to be very good in solving smooth mono-modal problems because they make use function values and their derivatives in their solution process. Nevertheless, in instances where there exists a noise in the objective function, these techniques encounter some challenges. In such cases, derivatives-free (non-gradient) methods such as Nelder-Mead downhill simplex and Hooke-Jeeves pattern search since they only make use of function values (Haftka & Gürdal, 2012).

Again, the traditional (deterministic) techniques are very effective and efficient in solving problems with large number of decision variables. Moreover, traditional techniques hardly require problem-specific tuning of parameters, so they are usually good at obtaining the optimal solutions in mono-modal optimization problems. However, in addition to their being rather tedious optimization techniques, especially to non-professional users, they encounter lots of challenges in multimodal optimization problems. Also, their efficiency is usually in continuous optimization environments. In most cases, their inefficiency in solving discrete optimization problems coupled with their weak handling of optimization situations with numerical noise is of concern to researchers (Toga, Clark, Thompson, Shattuck, & Van Horn, 2012). These observed weaknesses gave rise to the development of stochastic algorithms.

2.3 Stochastic algorithms

Stochastic (probabilistic) algorithms make extensive use of randomness in their search for optimization solutions are, generally, of two types, namely, Nature-inspired Computing (NIC) and Computing with Nature (CWN) (Dodig-Crnkovic, 2012).

2.3.1 Nature-inspired computing

Nature Inspired Computing (NIC) suite of algorithms draw their inspiration from the close observation of the complex problem-solving techniques coupled with the harmonious co-existence of different elements in natural environments (Kefi, Rokbani, Krömer, & Alimi, 2015). Scientific investigations inspired by NIC includes cellular automata (Codd, 2014), artificial immune systems (Hemamalini & Simon, 2011), neural networks (Mäkisara, Simula, Kangas, & Kohonen, 2014), evolutionary computation (Thiele, Miettinen, Korhonen, & Molina, 2009) and swarm intelligence (Ducatelle, Di Caro, & Gambardella, 2010) etc. Besides, active researchers in robotics have drawn tremendous inspiration from nature in developing mechanical artificial intelligence disciplines leading to the manufacture of self-configuring robots, water strider robot, mechanical cockroaches, robotic salamander, and so on (Dewangan, Naik, & Agrawal, 2014).

Another subset of NIC is the Biologically-inspired Algorithms (BIA) which are primarily concerned with harnessing the collective intelligence cum interaction of a group of biological agents leading to the incredible solutions to complex optimization problems (Pandiri & Singh, 2015). NIC has been developed with inspiration from biology, chemistry, physics and other engineering platforms. Typically, NIC techniques simulate the interaction, harmonious self-organization, interdependence and competition among natural elements in the ecosystem. Broadly speaking, NIC has proven to obtain good solutions to problems with the aid of heuristics or meta-heuristic information and this has enabled them to be very flexible, adaptable and robust to such extent that they are applicable to a wide range of optimization applications with very good outcomes (Fister Jr, Yang, Fister, Brest, & Fister, 2013).

2.3.2 Computing with nature (CWN)

CWN refers to the computing paradigm which is one of the latest innovations that is revolutionizing computing through its focus on using natural materials such as molecules (e.g. RNA and DNA) and quantum (quantum computing) for executing computational processing rather than silicon. Other forms of CWN include bio-chemical computing, molecular computing otherwise called bio-computing, bio-molecular computing or DNA computing that represents data as bio-molecules (such as DNA strands) and uses tools from molecular biology in processing data to perform arithmetic, logical or other computing operations (Rozenberg et al., 2011). Since its development, molecular computing has successfully been applied to solve 7-vertice TSP problems by merely manipulating DNA strands in a test-tube, cryptography, 20-variable 3SAT problems, splicing systems, sticker systems and the design applications for smart drugs (de Castro, 2007).

On the other hand, quantum computing regards data as quantum bits and then engages them mechanically through entanglements and super-positioning to perform computations. A quantum bit (otherwise called qubit) holds either a '1', '0' or a quantum superposition of either a '1' or '0'. Through the use of logic gates, the quantum computing performs computational operations on the qubits with the aid of either Shor's polynomial algorithm for factoring the integers and/or the Grover's algorithm for quantum database query (Hirvensalo, 2013). Quantum algorithm has been successfully applied to quantum teleportation quantum cryptography, pattern identification and classification, nuclear magnetic resonance imaging and so on (Hirvensalo, 2013). In spite of its initial success, quantum computing is still at its early stage of development. It is, therefore, too early to fully appreciate its merits and demerits because its potentials are still being investigated. In any case, a common characteristic of NIC is that they employ either heuristic or metaheuristic in their effort at arriving at acceptable solution.

One important to feature of deterministic algorithms according to a recent study (Yang, 2018), is that solutions usually is determined by an iterative procedure that starts from an initial point. In other words, the only randomness in a deterministic algorithm is the search starting point which usually is either initialized randomly or a mere educated guess. Similarly, with regards to the algorithm structure, the main exception in a deterministic algorithm is the stochastic gradient method that utilizes the approximation to the true gradient using with some randomness. For most other analytical approaches and deterministic search algorithms, there is virtually no exact randomness component.

On the contrary, randomness is the crux of metaheuristic algorithms whether they are evolutionary or other swarm intelligence techniques. In this class of algorithms, randomness in algorithms development is the first rule of the game. Continuing, Yang (2018), asserts that the capacity of the algorithm to properly exploit and manage its randomness component is a major determinant of its effectiveness. This proper deployment of the randomness component is crucial since metaheuristic algorithms since metaheuristic approaches use a trial-by-eliminating-errors mechanism in finding solutions to difficult optimization search problems.

In their contribution, Bottou *et al* (2018) asserts that since large-scale machine learning is a component beneficiary of stochastic-gradient methods, another name for metaheuristic techniques, perhaps, it is necessary that optimization methods should diminish noise that sometimes are present in stochastic techniques. It must be emphasized that nature-inspired algorithms, commonly called metaheuristics are developed with inspiration from the in-depth observation and analysis of natural phenomena. This body of algorithms learns from the efficiency, eff ectiveness cum beauty of nature. Their main selling point, it must be emphasized is their capacity to harness and properly manage a balanced deployment of randomness cum a proper combination with certain deterministic components is in fact the essence of making such algorithms so powerful and eff ective. Yang (2018) concludes that if the randomness component in a search algorithm is too high, the solutions obtained by the search algorithm may not easily converge since the algorithm may continue rather endlessly search the search space for a solution. Conversely, not having a random component reduces the stochastic algorithm to a deterministic one A good metaheuristic algorithm, therefore is one that is able to achieve a balanced tradeoff is needed.

In principle, metaheuristic algorithms use the regular social behavior of the search agent(s). To achieve this, metaheuristic algorithms deploy the real-number randomness coupled with some form of normal social communication and interactions among the search agents. This class of algorithms is easy to implement since there exists no encoding or decoding of the algorithms' parameters into strictly binary strings as is common in evolutionary algorithms such as Genetic Algorithm and Genetic Programming. As such metaheuristic algorithms are usually very efficient and flexible (Odili *et al*, 2017b). These characteristics, perhaps, is a pointer to their wide acceptability and popularity among researchers.

In summary, suffice to say that metaheuristic algorithms are designed from inspiration from nature and deliberately mimic some successful characteristics of chemical, biological or physical systems in nature. Today, among many other algorithms, the metaheuristic algorithms dominate the optimization search landscape (Yang, 2018). The reasons for this are:

(i) Most metaheuristic algorithms deploy multiple agents in their search which is akin to what operates among swarms in natural environments.
(ii) Most metaheuristic approaches permit the use of parallelization and vectorization implementations. As such they allow straight-forward implementation
(iii) Most metaheuristic algorithms are so flexible that they find easy applications to diverse kinds of optimization problems
(iv) Most metaheuristic algorithms are very efficient and effective in arriving at solutions
(v) Most metaheuristic algorithms are able to steer away from falling into local optima.

3. HEURISTICS AND METAHEURISTICS

As stated earlier, NIC makes use of heuristics and metaheuristics in its quest for solutions. Heuristic techniques simply exploit some information about a problem being solved to obtain solutions to such problems. Exploiting the heuristic information about the problem enables heuristic algorithms to obtain competitive solutions to difficult optimization problems within an acceptable time (Safari, 2015). However, heuristics are near-exact algorithms. In other words, heuristic algorithm does not lay claim to being able to obtain the exact optimal solutions. On its part, metaheuristics, simply means 'beyond heuristics' and are deemed to perform better than heuristics since they incorporate intelligent memory, experiential and other biases to direct the search process (Prakasam & Savarimuthu, 2015).

In differentiating between metaheuristics and heuristics, heuristics make elaborate use of local search components, but metaheuristic techniques deploy some local search (exploitation) coupled with global exploration (exploration) as well as randomizations. The use of randomizations enables metaheuristics to steer away from being ensnared in a local optimum, drive them to a more global search as well as assist the search by obtaining different results in any of the iterations until the algorithm homes in at a solution. The ultimate objective of metaheuristics is to obtain the best possible result via the use of internal mechanisms to achieve adequate exploration cum exploitation of the search space (Blum & Roli, 2003). In general, metaheuristics, unlike heuristics enjoy wider applications to diverse problems ranging from economics telecommunications, bioinformatics to manufacturing etc. (Osman & Kelly, 2012) .

Broadly speaking, metaheuristics is classified either trajectory-based or population-based (Beheshti & Shamsuddin, 2013). Some researchers regard John Holland as the father of population-based metaheuristic techniques because his works used a combination of automata methodology and theoretical genetics to good effect in 1962. Since the publication of his successful experimentation of the above, many researchers followed this trend of applying diversification and variation techniques to a population to obtain results within a given search space. Some of the earlier techniques that followed John Holland trajectory include Dorigo and Di Caro's Ant Colony Optimization ACO (Di Caro & Dorigo, 1998), Schaffer's Vector-Evaluated Genetic Algorithm (VEGA) (Pierre, Zakaria, & Pal, 2011); Farmer, Packard and Pearson's Artificial Immune Systems (Farmer, Packard, & Perelson, 1986); Holland's and Rosenberg's Evolutionary Strategies (Cuomo et al., 2012), and so on.

4. ESSENTIAL COMPONENTS OF METAHEURISTICS

Four important features distinguishes a good metaheuristic algorithms and these include the use of randomness, global search mechanism (otherwise called diversification or exploration), local search (intensification or exploitation) mechanism and the mechanism that identifies the best outcome per iteration in course of a search (Osaba, Yang, Diaz, Lopez-Garcia, & Carballedo, 2016).

4.1 Exploration and exploitation

The exploration component of metaheuristics ensures that the algorithm covers as much space as it can of the search space within a reasonable search time. In Particle Swarm Optimization (PSO), the exploration component is the part represented by the Pg which is the velocity tracked by the best particle (see Equation 5) and in the African Buffalo Optimization, the bg trail of the best buffalo (see Equation 6).

$$v_i\,(t+1) = \omega v_i\,(t) + c_1\,r_1\,(p_i\,(t) - X_i\,(t)) + c_2\,r_2\,(Pg_i\,(t) - X_i\,(t)) \qquad (5)$$

$$\mathrm{m_k}' = \mathrm{m_k} + \mathrm{lp1}(bg - \mathrm{w_k}) + \mathrm{lp2}(\mathrm{bp_k} - \mathrm{w_k}) \qquad (6)$$

In Equation 5, $c_2\,r_2\,(Pg_i\,(t) - X_i\,(t))$ calculates the velocity of individual particles in relation to the global best particle and this is the exploration component of the PSO, while the $c_1\,r_1\,(p_i\,(t) - X_i\,(t)$ traces the path of each particle in vis-a-vis the local best particle and this represents the local search component of the algorithm. Similarly, in Equation 6, the exploration part of the ABO is calculated for each buffalo in relation to the path followed by the best buffalo represented by $\mathrm{lp1}(bg - \mathrm{w_k})$.

On its part, the exploitation component constrains the metaheuristic to concentrate its search around the locations with promising results. In some algorithms, such as the ACO, besides concentrating the search around the areas of good solutions, the exploitation helps in selecting the decision vectors (search agents) that has the best outcomes in a particular iteration. In PSO and the ABO, the exploitation component is represented by $c_1\, r_1\, (p_i\ (t)$ and $\mathrm{lp1}(bg - \mathrm{w_k})$ respectively

Please note that a good metaheuristic, therefore, is one that is able to achieve the best tradeoff between exploitation and exploration utilizing the randomness component of the algorithm while at the same time identifying the best outcome in any given iteration (Fang, Lee, & Schilling, 2010). However, when a metaheuristic embarks on too elaborate exploitation, it may be ensnared in a local minimum thus being unable to locate the global optimum. Conversely, embarking on too elaborate exploration with a little exploitation may result in the system experiencing delay in convergence. Conversely, too detailed exploitation cum exploration may lead to system delay at a great cost of computer resources and users' time and. Moreover, too little exploration and exploitation may result in the degradation of the algorithm's effectiveness and efficiency (Aydoğdu, Akın, & Saka, 2016).

4.2 Best solution identification

Another key feature of a good metaheuristic is the ability to identify the best solution in an iteration and possibly the best design vector associated with such best solution. This is generally called 'The survival of the fittest' criterion. One way of achieving this is to keep updating the current best found so far (Yang, 2011). In Cuckoo Search, this identification is carried out by these two lines of code

If ($f_i\ (X_{ij}(\mathrm{t}\ +\ 1)) > f_k(X_{ij}(\mathrm{t}\,))$ **then**
Replace k by the new solution (7)

In ABO, the identification of the best buffalo is done by:

$$\mathrm{w_k}' = \frac{(\mathrm{w_k} + \mathrm{m_k})}{\lambda} \tag{8}$$

Similarly, the PSO executes this by:

$$X_i(t+1) = X_i(t) + v_i(t+1)$$
If fitness≤ 0,
Print $G_i\ \ best$ of each particle (9)

4.3 Randomization in metaheuristics

Having established that the four main essentials of a good metaheuristic are randomness, exploitation, exploration as well as the identification of the best performer, the technique employed by each algorithm to achieve these distinguishes an algorithm from the rest (Li, Chu, Langford, & Schapire, 2010). In general, it is safe to claim that algorithms attain these noble objectives through the use of randomization in coupled with a deterministic process via exploration and exploitation. A common mechanism to achieve randomization is to determine the upper and lower boundaries in a uniform distribution between 0 and 1. Algorithms such as Firefly Algorithm and Particle Swarm Optimization employ this method. Other metaheuristics such as Cuckoo Search use the Lévy flight (Senthilnath, Das, Omkar, & Mani, 2012). A Lévy flight is a random movement (walk) or random process that is characterized by step-jumps which is akin to the uncoordinated movement of dust particles in a fluid or the movement of housefly in search of food. The lambda (λ) component of Equation 8 above in

ABO is used to achieve randomness in the algorithm just as r_1 and r_2 components of Equation 5 above performs the same function in PSO.

In summary, some metaheuristics like the Evolutionary Programming, Genetic Programming and Genetic Algorithm make use of mutation and crossover to achieve the exploration effects. Mutation ensures that new solutions are different from the initial populations (parents) while crossover places a limit on exploitation (Rani, Jain, Srivastava, & Perumal, 2012). These kinds of algorithms, such as the Genetic Algorithm (GA), embark on exploitation through generating new solutions around a promising (superior) solution. This could be achieved by employing a random walk as the move in Simulated Annealing (SA) (Kirkpatrick, Gelatt, & Vecchi, 1983) and pitch adjustment in Harmony Search (HS) algorithm (Mahdavi, Fesanghary, & Damangir, 2007) represented by:

$$Xnew = Xold + s\,w \qquad (10)$$

Here s represents the step size and w is drawn from a Gaussian distribution with zero mean. Care is taken to ensure that the step size is neither too narrow nor too wide. Too wide a step size will lead to the algorithm becoming inefficient in exploitation in favor of exploration. In the same way, when the step size is too narrow, too much exploitation that may lead to falling into local optima/minima results. It is rather suggested that algorithms employ random walks such as Levy flights where the step size is drawn from a Levy distribution with acceptable step sizes (Kennedy, 2010).

5. CONCLUSION

This paper examined the concept of optimization in science, engineering and industrial applications generally before homing in on optimization algorithms with special reference to metaheuristic algorithms in particular. Since metaheuristics have assumed a very prominent place in the optimization of engineering, scientific and industrial processes leading to mass interest in the field by many experienced and budding scientists, there is the need to analyze the main components of a metaheuristic algorithms so as to enhance understanding, engender user-friendliness, assist in better choice of a particular metaheuristic in problem-solving and ensure wider applicability, hence the need for this research.

After some analysis, this paper opines that a good metaheuristic algorithm should contain and properly balance four critical elements namely: identification of best solution per iteration, exploration, exploitation and randomness mechanisms. A metaheuristic that lacks any of the above or that is unable to balance any of the four mechanisms may not be the best choice in solving any optimization problems. Again, this paper observes that different algorithms achieve those four essential requirements using different techniques as highlighted in course of the discussions above. A good knowledge of how each of the algorithms achieves any of the four essential components is helpful in algorithm analysis and choice in solving any optimization problem since no particular algorithm has been proven to be the best in solving all kinds of optimization problems, hence the No free Lunch theorem of optimization algorithms (Wolpert & Macready, 1997).

ACKNOWLEDGEMENT

The author appreciates the support of the Department of Mathematical Sciences, Faculty of Natural and Applied Sciences, Anchor University Lagos in funding this research work.

REFERENCES

Ahmadizar, F., & Soltanpanah, H. (2011). Reliability optimization of a series system with multiple-choice and budget constraints using an efficient ant colony approach. *Expert systems with Applications, 38*(4), 3640-3646.

Aydoğdu, İ., Akın, A., & Saka, M. (2016). Design optimization of real world steel space frames using artificial bee colony algorithm with Levy flight distribution. *Advances in engineering software, 92,* 1-14.

Beheshti, Z., & Shamsuddin, S. M. H. (2013). A review of population-based meta-heuristic algorithms. *Int. J. Adv. Soft Comput. Appl, 5*(1), 1-35.

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR), 35*(3), 268-308.

Codd, E. F. (2014). *Cellular automata*: Academic Press.

Cuomo, C. A., Desjardins, C. A., Bakowski, M. A., Goldberg, J., Ma, A. T., Becnel, J. J., . . . Levin, J. Z. (2012). Microsporidian genome analysis reveals evolutionary strategies for obligate intracellular growth. *Genome research, 22*(12), 2478-2488.

Davoodi, E., Hagh, M. T., & Zadeh, S. G. (2014). A hybrid Improved Quantum-behaved Particle Swarm Optimization–Simplex method (IQPSOS) to solve power system load flow problems. *Applied Soft Computing, 21*, 171-179.

de Castro, L. N. (2007). Fundamentals of natural computing: an overview. *Physics of Life Reviews, 4*(1), 1-36.

Dewangan, S., Naik, A., & Agrawal, A. (2014). Study of Nature Inspired Computing.

Di Caro, G., & Dorigo, M. (1998). AntNet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research,* 317-365.

Dodig-Crnkovic, G. (2012). Info-computationalism and morphological computing of informational structure *Integral Biomathics* (pp. 97-104): Springer.

Ducatelle, F., Di Caro, G. A., & Gambardella, L. M. (2010). Principles and applications of swarm intelligence for adaptive routing in telecommunications networks. *Swarm Intelligence, 4*(3), 173-198.

Faludi, A. (2013). *A reader in planning theory*: Elsevier.

Fang, C., Lee, J., & Schilling, M. A. (2010). Balancing exploration and exploitation through structural design: The isolation of subgroups and organizational learning. *Organization Science, 21*(3), 625-642.

Farmer, J. D., Packard, N. H., & Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena, 22*(1), 187-204.

Feist, A. M., & Palsson, B. O. (2010). The biomass objective function. *Current opinion in microbiology, 13*(3), 344-349.

Fister Jr, I., Yang, X.-S., Fister, I., Brest, J., & Fister, D. (2013). A brief review of nature-inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186.*

Gentle, J. E. (2013). *Random number generation and Monte Carlo methods*: Springer Science & Business Media.

Haftka, R. T., & Gürdal, Z. (2012). *Elements of structural optimization* (Vol. 11): Springer Science & Business Media.

Hemamalini, S., & Simon, S. P. (2011). Dynamic economic dispatch using artificial immune system for units with valve-point effect. *International Journal of Electrical Power & Energy Systems, 33*(4), 868-874.

Hirvensalo, M. (2013). *Quantum computing*: Springer.

Hughes, T. J. (2012). *The finite element method: linear static and dynamic finite element analysis*: Courier Corporation.

Julius Beneoluchi Odili, Mohd Nizam Mohmad Kahar, A. Noraziah, Zarina, M., & Haq, R. U. (2017). Performance Analyses of Nature-inspired Algorithms on the Traveling Salesman's Problems for Strategic Management. *Intelligent Automation & Soft Computing*, 1-11.

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, 2015.

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

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science, 220*(4598), 671-680.

Kornblum, J. (2006). Identifying almost identical files using context triggered piecewise hashing. *Digital investigation, 3*, 91-97.

Kuhn, H. W. (2014). Nonlinear programming: a historical view *Traces and Emergence of Nonlinear Programming* (pp. 393-414): Springer.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal (2018). Optimization Methods for Large-Scale Machine LearningSIAM Review 2018 60:2, 223-311

Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010). *A contextual-bandit approach to personalized news article recommendation.* Paper presented at the Proceedings of the 19th international conference on World wide web.

Machairas, V., Tsangrassoulis, A., & Axarli, K. (2014). Algorithms for optimization of building design: A review. *Renewable and Sustainable Energy Reviews, 31*, 101-112.

Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied mathematics and computation, 188*(2), 1567-1579.

Mäkisara, K., Simula, O., Kangas, J., & Kohonen, T. (2014). *Artificial neural networks* (Vol. 2): Elsevier.

McCollum, B., Schaerf, A., Paechter, B., McMullan, P., Lewis, R., Parkes, A. J., . . . Burke, E. K. (2010). Setting the research agenda in automated timetabling: The second international timetabling competition. *INFORMS Journal on Computing, 22*(1), 120-130.

Motwani, R., & Raghavan, P. (2010). *Randomized algorithms*: Chapman & Hall/CRC.

Odili, J. B. (2018). Implementation Analysis Of Cuckoo Search For The Benchmark Rosenbrock And Levy Test Functions. *Journal of ICT, 17*(1), 17-32.

Odili, J. B., & Kahar, M. N. M. (2016). AFRICAN BUFFALO OPTIMIZATION ico-pdf. *International Journal of Software Engineering and Computer Systems, 2*(1), 28-50.

Odili, J. B., Kahar, M. N. M., & Noraziah, A. (2017). Parameters-tuning of PID controller for automatic voltage regulators using the African buffalo optimization. *PLoS ONE, 12*(4), e0175901.

Odili, J. B., Kahar, M. N. M., & Noraziah, A. (2018). Swarm Intelligence Optimization Algorithms: A Review. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 10*(1-4), 139-142.

Odili, J. B., Kahar, M. N. M., & Noraziah, A. (2017). A Comparative Study of Neural Networks Methods and the African Buffalo Optimization for the Travelling Salesman's Problems. *Advanced Science Letters*, *23*(11), 11044-11047.

Odili, J. B., & Noraziah, A. (2018). African buffalo optimization for global optimization. *Current Science (00113891), 114*(3).

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.

Osman, I. H., & Kelly, J. P. (2012). *Meta-heuristics: theory and applications*: Springer Science & Business Media.

Pandiri, V., & Singh, A. (2015). Swarm intelligence approaches for multidepot salesmen problems with load balancing. *Applied Intelligence*, 1-13.

Păun, G. (2012). Membrane computing. *Handbook of Natural Computing*, 1355-1377.

Pierre, D. M., Zakaria, N., & Pal, A. J. (2011). *Master-slave parallel vector-evaluated genetic algorithm for unmanned aerial vehicle's path planning.* Paper presented at the Hybrid Intelligent Systems (HIS), 2011 11th International Conference on.

Prakasam, A., & Savarimuthu, N. (2015). Metaheuristic algorithms and probabilistic behaviour: a comprehensive analysis of Ant Colony Optimization and its variants. *Artificial Intelligence Review*, 1-34.

Rani, D., Jain, S. K., Srivastava, D. K., & Perumal, M. (2012). 3 Genetic Algorithms and Their Applications to Water Resources Systems. *Metaheuristics in Water, Geotechnical and Transport Engineering*, 43.

Rozenberg, G., Bck, T., & Kok, J. N. (2011). *Handbook of natural computing*: Springer Publishing Company, Incorporated.

Safari, S. A. (2015). Wildlife and Big 5. *http://sasafari.com/wildlife-and-the-big-five/, 2016*.

Said, Y., & Wegman, E. (2009). Roadmap for optimization. *Wiley Interdisciplinary Reviews: Computational Statistics, 1*(1), 3-17.

Senthilnath, J., Das, V., Omkar, S., & Mani, V. (2012). *Clustering Using Levy Flight Cuckoo Search.* Paper presented at the BIC-TA (2).

Sniedovich, M. (2010). *Dynamic programming: foundations and principles*: CRC press.

Thiele, L., Miettinen, K., Korhonen, P. J., & Molina, J. (2009). A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary Computation, 17*(3), 411-436.

Toga, A. W., Clark, K. A., Thompson, P. M., Shattuck, D. W., & Van Horn, J. D. (2012). Mapping the human connectome. *Neurosurgery, 71*(1), 1.

Venter, G. (2010). Review of optimization techniques. *Encyclopedia of aerospace engineering*.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation, 1*(1), 67-82.

Wooldridge, J. M. (2010). *Econometric analysis of cross section and panel data*: MIT press.

Yang, X.-S. (2011). Review of meta-heuristics and generalised evolutionary walk algorithm. *International Journal of Bio-Inspired Computation, 3*(2), 77-84.

Yang, X. S. (2018). *Optimization Techniques and Applications with Examples*. John Wiley & Sons.

Zang, H., Zhang, S., & Hapeshi, K. (2010). A review of nature-inspired algorithms. *Journal of Bionic Engineering, 7*, S232-S237.