JOURNAL OF SCIECNCE TECHNOLOGY AND EDUCATION 7(3), SEPTEMBER, 2019
ISSN: 2277-0011; JOURNAL HOMEPAGE: WWW.ATBUFTEJOSTE.COM

An Investigative Parameter Analysis of Pastoralist Optimization Algorithm (Poa): A Novel Metaheuristic Optimization Algorithm

*Abdullahi, I. M., [1]Mu'azu M. B., [2]Olaniyi O. M. and [3]Agajo J.
[*,2,3]*Department of Computer Engineering, Federal University of Technology, Minna, Nigeria*
[1]*Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria*

ABSTRACT

In this paper, one of the most important parameters that affects the performance of a novel population based nature-inspired metaheuristic optimization algorithm called the Pastoralist Optimization Algorithm (POA) inspired by the pastoralists herding strategies was investigated. This is to determine the suitable value or range of values that should be used when applying the algorithm to solve optimization problems. The parameter that was investigated is the number of pastoralist (nP), that is the number of search agents or population size. Eight different pastoralist size (10, 20, 30, 40, 50, 60, 70 and 80) were investigated by testing the parameter on three standard test functions; unimodal Sphere function, multimodal Dejong and Shubert functions. Each test is simulated ten times, and the average optimal value and the average convergence time(s) obtained for each function and each parameter value was recorded. The experimental results obtained show that a pastoralist population size of 20 is desirable for convergence accuracy and speed.

ARTICLE INFO
Article History
Received: May, 2019
Received in revised form: July, 2019
Accepted: August, 2019
Published online: September, 2019

KEYWORDS
Algorithms; Benchmark test functions; Nature inspired metaheuristic algorithms; Pastoralist Optimization Algorithm (POA); Dejong Function; Sphere Function; Shubert Function

INTRODUCTION

Population-based Metaheuristic Optimization Algorithms (MOA) which are mostly inspired by nature have been successful in solving complex optimization problems (Yang, 2013). They possess some characteristics that makes them suitable for solving such problems. One of the most important characteristic of MOA is their ability to use multiple search agents instead of single search agents as in the case of trajectory based optimization algorithms. The number of search agent is an important parameter that allows communication between search agents which is a key factor that gives MOA the ability to balance between exploitation (local search) and exploration (global search). Balance between exploration and exploitation by any algorithm determines how successful the algorithm will be in solving optimization problems.

MOA are inspired by some nature characteristics like animal behaviors, evolution, ecology, culture and so on, controlled by high level strategies (Bronlee, 2011). The inspirations that have been used to develop MOA are generally classified as Biology-Based Algorithm (BBA), Physics-Based Algorithm (PBA), Chemistry-Based Algorithm (CBA) and Mathematics-Based Algorithm (MBA) (Xing & Gao, 2013). (Siddique & Adeli, 2015), classified BBA as either Bio-Inspired Algorithms (BIA), Swarm Intelligence Algorithms (SIA), and Evolutionary Algorithms (EA). BIA are algorithms whose behavior are inspired by the characteristics of some organisms and animal species which include movement, communication and coordination. Algorithms in this category, include Bacteria Foraging Algorithm (BFA), Biogeography-Based Optimization (BBO), Lion Optimization Algorithm (LOA), Ant-Lion Optimizer (ALO) and so on.

SIA is inspired by the behaviours of group of insects like bees, ants, termites and wasp each living in separate colonies. Algorithms in this category include Ant Colony Optimization (ACO), Bee Algorithm (BA), Bat Algorithm, Artificial Bee Colony (ABC) and so on. Lastly, the EA which are inspired by the theory of Evolution proposed by Charles Darwin. The most famous amongst them is the Genetic Algorithm (GA), Evolutionary Strategy and Genetic Programming (Siddique & Adeli, 2015). (Mirjalili & Lewis, 2016) classified metaheuristic algorithms as either swarm-based, human-based, evolutionary-based or physics-based MHA. The human based algorithms are MHA that are inspired by human behaviours. Examples are the Teaching Learning Based Optimization (TLBO), Harmony Search (HS), Imperialist Competitive Algorithm (ICA), Group Search Optimizer (GSO) and so on as shown in Figure 1.

**Corresponding author: Abdullahi, I. M. amibrahim@futminna.edu.ng Department of Computer Engineering, Federal University of Technology, Minna.*

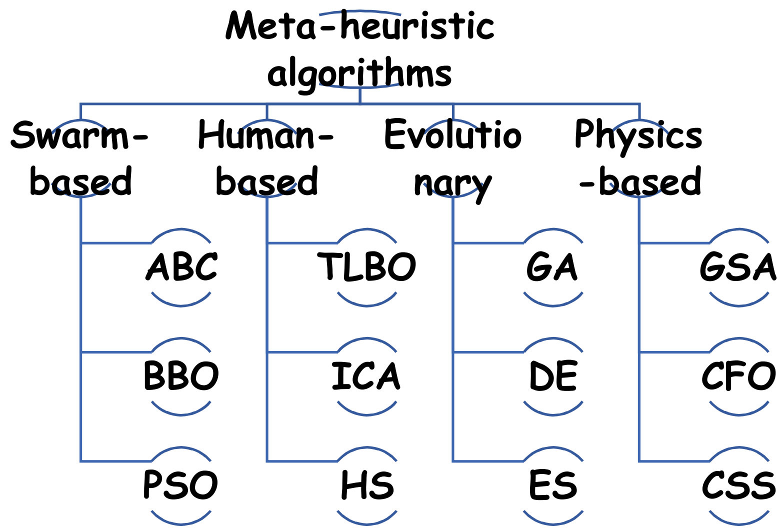

Figure 1: Classification of Metaheuristic Algorithms (Mirjalili & Lewis, 2016)

PASTORALIST OPTIMIZATION ALGORITHM

POA is a novel population-based metaheuristic optimization algorithm inspired by nomadic pastoralist herding strategies proposed by (Abdullahi, Mu'azu, Olaniyi, & Agajo, 2018). Nomadic Pastoralism which is a system of producing livestock characterized by movement of animals in search of water and quality pasture while maintaining environmental equilibrium (Msuya, 2015; Rota & Sperandini, 2008). The strategies adopted by the nomadic pastoralist include: Scouting for exploration or search of suitable camp site (Gerald & Dorothy, 2013), camp selection and camping for temporary settlements for daily exploitation (Liao, Clark, DeGloria, Mude, & Barret, 2016), herding, which include splitting or herd dispersal for risk minimization and trap avoidance (Gert, 2011), finally, merging for camp fitness evaluation and the search for a new camp depending on the quality assessment (Adriansen, 2008).

These strategies were modeled mathematically and used to develop the POA shown in Figure 2. In POA, a set of pastoralists were randomly generated to form the initial population of the search space. 25% of pastoralists are selected as scout pastoralists from the initial pastoralist population. The scout pastoralists search for the best location for camping where herding takes place. During herding, pastoralist split themselves to minimize risk of getting stuck in local optima and this is followed by merging where the fitness of each pastoralist is evaluated and the decision for a new camp search is taken until the stopping criteria is reached. The algorithm has shown promising results when tested on a set of unimodal and multimodal benchmark functions. Just like any other MOA, POA was developed using some parameters. They include; Number of Pastoralist (number of search agents), the Scouting Rate and Splitting Rate. The number of search agents is the parameter that is common to all MOA, hence, it was investigated in this paper.

EXPERIMENTATION

In this section, the experiments that were performed in order to investigate the effect of the number of pastoralist on the performance of POA are presented. Two groups of test functions were selected to benchmark POA performance. They are; unimodal and multimodal test functions shown in Table 1 (Pohlheim, 2005). Unimodal functions have only a single global optimal solution, and they are used to evaluate the algorithm exploitation capability while multimodal test functions are used to evaluate algorithms exploration capability because they contain many local optima whose optimal number increase exponentially with increase in the number of variables (Mirjalili & Lewis, 2016). Figure 3, Figure 4 and Figure 5 shows the plots of the Sphere, Dejong N5 and Shubert functions respectively. The experiments were carried out by testing POA with eight different pastoralist sizes {nP ϵ (10, 20, 30…, 80)} on the three benchmark test functions. Each test was simulated 10 times and the average convergence accuracy and time in seconds was recorded

*Corresponding author: Abdullahi, I. M. amibrahim@futminna.edu.ng Department of Computer Engineering, Federal University of Technology, Minna.

JOURNAL OF SCIECNCE TECHNOLOGY AND EDUCATION 7(3), SEPTEMBER, 2019
ISSN: 2277-0011; JOURNAL HOMEPAGE: WWW.ATBUFTEJOSTE.COM

Algorithm 1: Pastoralist Optimization Algorithm
Initialize all POA parameters
Select 25% of Pastoralist as Scout pastoralist randomly
Initialize scout location randomly and evaluate the fitness of each scout
    While i < maximum scouting rate
        Update scouts' locations and evaluate their fitness's
        Select best scout location S_{best} as camping location
Initialize pastoralist location within camp
Evaluate fitness of each pastoralist and determine best pastoralist P_{best}
Split pastoralist to different locations within camp and evaluate fitness of each pastoralist
    While j < maximum splitting rate
        Divide the current camp radius by the number of pastoralist
        Split pastoralist within the remaining camp area and evaluate their fitness's
        Update best camp pastoralist Pc_{best}
For (k = 1: maximum iteration)
    Update scout location
    Update the global best pastoralist Gc_{best}.
Return the global best-found pastoralist G_{best}

Figure 2: POA Algorithm (Abdullahi, Mu'azu, Olaniyi, & Agajo, 2018)

.

Table 1: Unimodal Test Functions (Pohlheim, 2005)

Function ID	Function Name	Equation	Features	Dimension	Range	Global minimum
F1	Sphere	$f(x) = \sum_{i-1}^{d} x_i^2$	Unimodal, Variable dimension	5	[-5.12, 5.12]	0
F2	Dejong N.5	$f(x) = \left(0.002 + \sum_{i-1}^{2} \frac{1}{i + (x_1 - a_{1i})^6 + (x_2 - a_{2i})^6}\right)^{-1}$	Multimodal, separable, non-scalable	2	[-65.536, 65.536]	0.998
F3	Shubert	$f(x) = \left(\sum_{i-1}^{5} ic\quad((i+1)x_1 + i)\right)\left(\sum_{i-1}^{5} ic\quad((i+1)x_2 + i)\right)$	Multimodal, non-separable, scalable	2	[-10, 10]	-186.7309

*Corresponding author: Abdullahi, I. M. ✉ amibrahim@futminna.edu.ng ✉ Department of Computer Engineering, Federal University of Technology, Minna.

JOURNAL OF SCIECNCE TECHNOLOGY AND EDUCATION 7(3), SEPTEMBER, 2019
ISSN: 2277-0011; JOURNAL HOMEPAGE: WWW.ATBUFTEJOSTE.COM

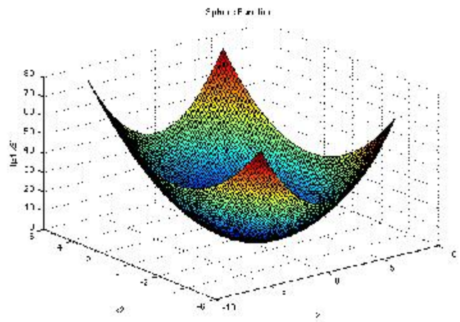

Figure 3: Sphere Function Plot

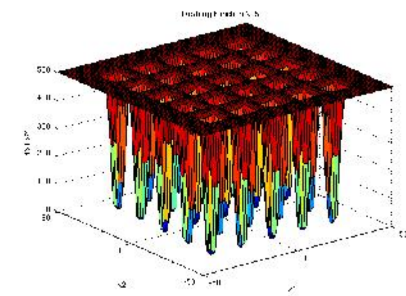

Figure 4: Dejong N5 Function Plot

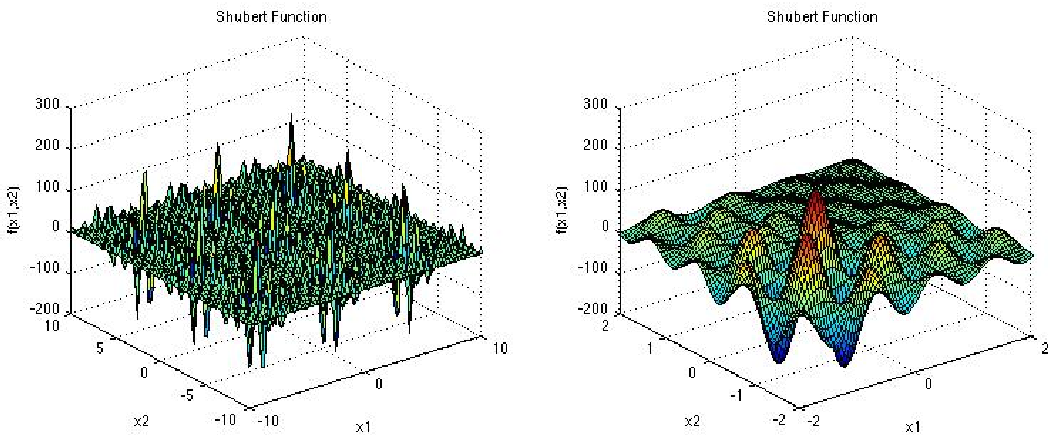

Figure 5: Shubert Function Plot

RESULTS AND DISCUSSION

The results obtained by testing the POA on Sphere, Dejong and Shubert functions are shown in Table 2. The table contains the average fitness values (Fvalue) and average time (Time (s)) for the test benchmark functions and eight different parameter values with intervals (10, 20, 30, …, 80).

Table 2: Parameter Investigation with Sphere Function

Population size (*nP*)	Sphere Function		Dejong N5 Function		Shubert Function	
	Fvalue	**Time(s)**	**Fvalue**	**Time(s)**	**Fvalue**	**Time(s)**
10	2.11E-75	47.3681	0.9980	31.1786	-185.917	22.5731
20	7.71E-76	50.5751	0.9980	59.5136	-186.731	41.9328
30	4.35E-76	75.0557	0.9980	87.6398	-186.731	71.4133
40	1.92E-75	93.0362	0.9980	115.9359	-186.731	84.1309
50	8.04E-77	114.7081	0.9980	146.4384	-186.731	105.1111
60	6.02E-76	138.7337	0.9980	180.0950	-186.731	131.5985
70	2.69E-76	170.4469	0.9980	211.5607	-186.731	146.2042
80	3.32E-76	196.9875	0.9980	250.1643	-186.731	159.8675

*Corresponding author: Abdullahi, I. M. amibrahim@futminna.edu.ng Department of Computer Engineering, Federal University of Technology, Minna.

JOURNAL OF SCIECNCE TECHNOLOGY AND EDUCATION 7(3), SEPTEMBER, 2019
ISSN: 2277-0011; JOURNAL HOMEPAGE: WWW.ATBUFTEJOSTE.COM

The results obtained show that when the pastoralist population is 10, the fitness value obtained for Sphere, Dejong and Shubert functions are 2.11E-75, 0.998 and 185.917 respectively. Only for Dejong function that the global minimum value was obtained. At population of 20 to 80, the global minimum was obtained for Dejong and Shubert function while for Sphere function, the best value was obtained at a population size of 50 with a value of 8.04E-77 compared to the second best value of 7.71E-76 obtained at population size of 20. Also, at population size of twenty, all the functions took less than one minute to converge compared to population size of 30 and above as shown in Figure 6.

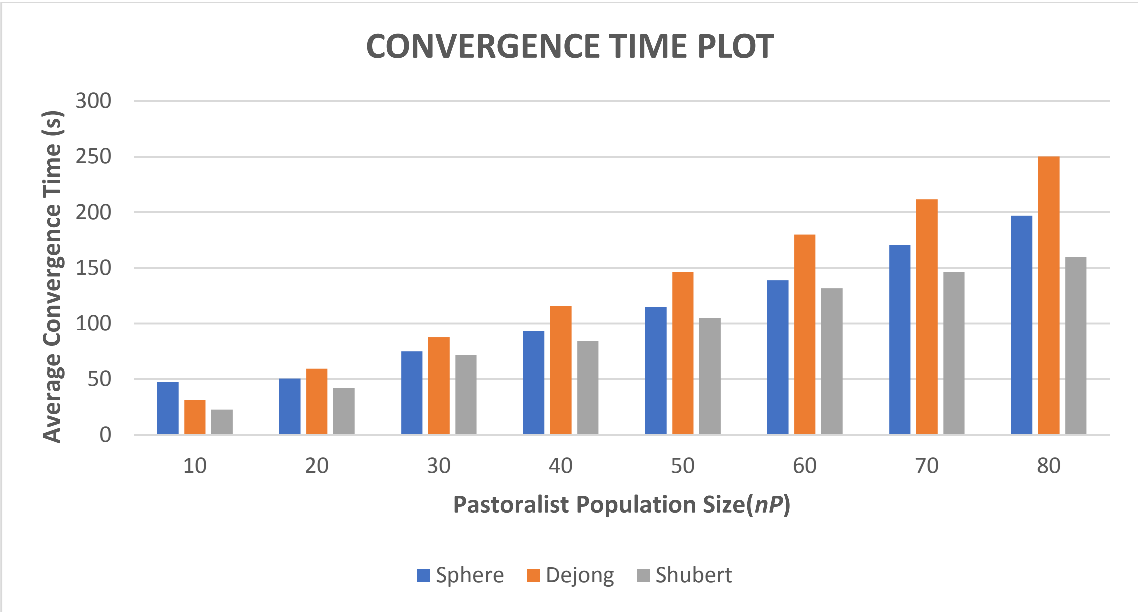

Figure 6: Convergence Time Plot for Sphere, Dejong and Shubert Functions

From Figure 6, the average time taken to converge per 1000 iterations and 10 independent runs show that as the population size increases, the time also increases. For example, for population size of less than 20, the maximum average convergence time per 1000 iterations is less than 60 seconds while at a population size of 80, it takes an average of 250 seconds for Dejong function, above 159.9 seconds for Shubert function and 197 seconds for sphere function.

From the results, the best population size (number of pastoralist or search agents) that will be appropriate for the novel POA is 20 both in terms of the convergence accuracy and the convergence time. If a value lower than 20 is selected, it might affect the accuracy of convergence, that is, the ability to obtain the global optimal solutions while a higher value of *nP* above 20 will increase the time of convergence despite obtaining the global best solution.

CONCLUSION

The effect of the population of search agents (number of pastoralist (nP)) on the accuracy and time of convergence of the newly developed POA was investigated in this paper. The algorithm was first applied to solve three numerical optimization problem using the Sphere, Dejong and Shubert functions. The ability of the algorithm to obtain the global optimal (near optimal) solutions and time of convergence per 1000 iterations were measured. The results indicate that a population size of 20 is most appropriate because most of the accuracy were solved at the lowest possible time.

RECOMMENDATION

In this paper, three functions were tested, more functions can also be tested to further observe the effect of the population size. Also, lower population size in multiples of fives can be investigated as well. Other POA parameters like scouting rate and splitting rate that are likely to affect the performance of the algorithm can be investigated as well.

REFERENCES

Abdullahi, I. M., Mu'azu, M. B., Olaniyi, O. M., & Agajo, J. (2018). Pastoralist Optimization Algorithm (POA): A Novel Nature-Inspired

*Corresponding author: Abdullahi, I. M. amibrahim@futminna.edu.ng Department of Computer Engineering, Federal University of Technology, Minna.

Metaheuristic Optimization Algorithm. *International Conference on Global and Emerging Trends (2018)* (pp. 101-105). Abuja: Global Trends Academy.

Adriansen, H. K. (2008). Understanding pastoral mobility: the case of Senegalese Fulani. *The Geographical Journal , 174*(3), 207-222.

Bronlee, J. (2011). Clever algorithms: Nature-inspired programming recipes. Retrieved 12 2, 2016, from http://www.cleveralgorithms.com

Gerald, J., & Dorothy, R. (2013). *Pastoralism in Practice: Monitoring Livestock Mobility in Contemporary Sudan.* Kenya: United Nations Environment Programme.

Gert, d. G. (2011). *Taking Indegenious Knowledge Serious: The case of Pastoral Strategies Among Turkana Nomads of Northwest Kenya.* Eldoret, Kenya: ILEIA Newsletter. Retrieved 10 31, 2016, from www.agriculturesnetwork.org/magazines/global

Hesse, C., & Cavanna, S. (2010). *(2010). Modern and Mobile: The Future of Livestock Production in Africa's Drylands.* UK: International Instutute for Environment and Develpment (IIED) and SOS Sahel International UK.

Liang, J. J., suganthan, P. N., & Deb, K. (2005). Novel Composition Test Functions for Numerical Global Optimization. *Proceedings of the 2005 swarm intelligence symposium, SIS* (pp. 68-75). IEEE, xplore.

Liao, C., Clark, P. E., DeGloria, S. D., Mude, A., & Barret, C. B. (2016). Pastoral mobility and policy recommendations for livestock herding in the Borana pastoral system in Southern Ethiopia. *Solutions Journal, 7*(5), 17-20.

Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 51-67.

Msuya, D. G. (2015). Pastoralism Beyond Ranching: A Farming System in Severe Stress in Semi-Arid Tropics Especially in Africa. *Journal of Agriculture and Ecology Research International, 4*(3), 128-139.

Pohlheim, H. (2005). GEATbx Examples: Examples of Objective Functions. Retrieved August 2017, from http://www.geatbx.com/download/GEATbx_ObjFunExpl_v37.pdf.

Rota, A., & Sperandini, S. (2008). *Livestock and Pastoralist,.* International Fund for Agricultural Development.

Siddique, N., & Adeli, H. (2015). Nature Inspired Computing: An Overview and Some Future Directions. *Cogn Compu, 7*, 706–714.

Sutton, M. Q., & Anderson, E. N. (2010). *Introduction to Cultural Ecology.* Plymouth, UK: AltaMira Press.

Xing, B., & Gao, W.-J. (2013). *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms* (Vol. 62). New York: Intelligent Systems Reference Library.

Yang, X. S. (2013). Nature-Inspired Mateheuristic Algorithms: Success and New Challenges. *J Comput. Eng. Inf. Technol., 1*(1), 1-3.

**Corresponding author: Abdullahi, I. M. amibrahim@futminna.edu.ng Department of Computer Engineering, Federal University of Technology, Minna.*