Review and Reformulation of the Sea Lion Optimization Metaheuristic

Conor O'Brien
Computer Science
William & Mary
Williamsburg, United States
csobrien@wm.edu

***Abstract*—The Sea Lion Optimization Algorithm (SLnO) is a recent metaheuristic optimization algorithm suitable for optimizing given functions over given search spaces. Its biphasic behavior helps find minima in many circumstances, including unimodal and multimodal distributions. We present the most concise and best articulated formulation of SLnO to date, identify shortcomings in the original formulation, and summarize the strengths of SLnO.**

***Index Terms*—optimization; metaheuristic optimization algorithms; Sea Lion Optimization Algorithm (SLnO)**

I. Introduction

Optimization problems are critical to many domains, including machine learning, usually framed as finding local and global optima for a given mathematical function [1]. Unlike traditional optimization algorithms such as gradient descent, which often impose specific assumptions on the problem (e.g., requiring the function optimization be differentiable), metaheuristic optimization algorithms generally make fewer assumptions [8].

A prototypical formulation of a metaheuristic optimization algorithm is Particle Swarm Optimization (PSO) [3]. PSO operates by generating a swarm of particles (a set of points in the search space), assigning each particle a random initial velocity. Then, PSO iteratively updates each particle based on factors like its velocity, random noise, the best position that particle encountered, and the best position encountered by the swarm overall.

Metaheuristic optimization algorithms, often inspired by systems in nature, generally fall into one of three categories: Evolution-based, physics-based, and swarm-based [5]. This field of metaheuristics is fast-paced and receiving constant innovations [8], with competitive algorithms numbering in the hundreds [7].

In section 2, we reformulate the algorithm as set forth Sea Lion Optimization Algorithm (SLnO) paper, informed by existing implementations [7]. In section 3, we summarize the strengths of the algorithm, as well as document the shortcomings of the original formulation. In section 4, we conclude with general thoughts about the field.

A. Notation

- $[a, b]$ is the range of real numbers between a and b inclusive.
- $x \sim U$ means x is sampled from a distribution U, assumed to be uniform unless otherwise specified.
- $x \leftarrow E$ means update the value of x to be that of the expression E.

II. Sea Lion Optimization Algorithm

Sea lions are uniquely situated in the animal kingdom with immense intelligent and sensitive whiskers. The Sea Lions' whiskers allow them to sense prey and make immediate decisions about prey's "size, shape, and position" based solely on the wakes and waves resultant from the prey's movement [5].

By leveraging both their individual sensing prowess and their hierarchical social structure, Sea Lions engage in group hunting as an optimal method of securing prey [5]. They follow this hunting pattern:

1) Identify and follow prey individually;
2) Call upon other nearby Sea Lions to surround their target; and
3) Converge on the prey to attack it.

These facts of nature inspire the mathematical model behind the Sea Lion Optimization Algorithm (SLnO). Although the authors separate the algorithm's description into distinct phases [5], we deem these phases to be mostly metaphorical, and instead prefer the following, more mathematical definition.

Consider a function $f : \mathbb{R}^n \to \mathbb{R}$. We wish to find a suitable vector $\mathbf{x}$ such that $f(\mathbf{x})$ is minimal over some search space S, usually a range $[b_l, b_u]$, which controls the minimal and maximal values for each component $x_j \in \mathbf{x}$. As a swarm-based metaheuristic, SLnO starts with p initial particles (sea lions). Each sea lion $\mathbf{SL}_k$ is initially set to a uniformly random position in S. We evaluate our function $f(\mathbf{SL}_k)$ for each sea lion and keep track of the sea lion which best minimizes our function, called $\mathbf{SL}_{best}$ [5].

The SLnO algorithm runs over N iterations. In each iteration, we perform one of two updating routines, depending on a condition the authors give as $SP_{leader} < 0.25$ [5]. For the effect of the algorithm, this can be treated as a black-box stochastic process, and, according to our analysis in the next section, seems to be able to be substituted with other stochastic processes. Depending on the choice made, at any iteration, the algorithm either hunts or circles.

A. Hunt

The hunt process involves two phases, controlled for by a linearly decreasing constant C. C ranges from 2 in the first iteration down to 0 (or close to 0) in the last iteration, and controls the target $\mathbf{T}$ the algorithm nudges each particle towards[1]. We have

$$\mathbf{T} = \begin{cases} \mathbf{SL}_{best} & \text{if } C < 1 \\ \mathbf{SL}_{rand} & \text{if } C \geq 1 \end{cases},$$

where $\mathbf{SL}_{rand}$ is a sea lion randomly sampled from the population. Then, depending on $\mathbf{T}$ and a random variable (either scalar or vector) $b \sim [0, 1]$, the hunt process updates the position of each sea lion as

$$\mathbf{SL}'_k \leftarrow \mathbf{T} - C|2b\mathbf{T} - \mathbf{SL}_k|.$$

B. Circle

The circle process distributes sea lions around the current best target in an n-sphere. For some random variable (either scalar or vector) $m \sim [-1, 1]$, the circle process updates the position of each sea lion as

$$\mathbf{SL}'_k \leftarrow \mathbf{SL}_{best} + \cos(2\pi m)|\mathbf{SL}_{best} - \mathbf{SL}_k|.$$

After either the hunt process or the circle process updates the positions of each sea lion, the algorithm must perform a normalization step. Each component $x_j \in \mathbf{SL}_k$ must be constrained to the search space S by replacing the affected $x_j \leftarrow x'_j \sim S$.

Once the algorithm executes all N iterations, SLnO emits $\mathbf{SL}_{best}$ as the final solution. This may optionally be augmented with a pocket algorithm to save potentially better intermediate optimizations, although, for many cases and parameter combinations, the algorithm terminates with the best solution it found.

III. Analysis

A. Performance

The SLnO authors tested the algorithm on 23 mathematical function optimization problems of varying complexity and modes. As compared to 5 other metaheuristics, SLnO consistently matched or outperformed the competition, performing the best in 16 cases across the board [5].

B. Underspecification

Unfortunately, the original paper by Masadesh et al. is underspecified, leaving implicit things perhaps clear to the authors, but unclear to those reviewing the paper. In crafting this review, we identified sources of confusion in the original statement of the algorithm [5], which filters down into subsequent papers [2] [4]. Although these subsequent papers which utilize SLnO, do so with reported success, their restatements of SLnO are just as imprecise and unhelpful as the original paper; the lack of accompanying code renders their theory injurious to understanding.

[1]The metaphorical motivation is when $C \geq 1$, the sea lions engage in search patterns, and when $C < 1$, the sea lions engage in "dwindling encircling techniques" [5].

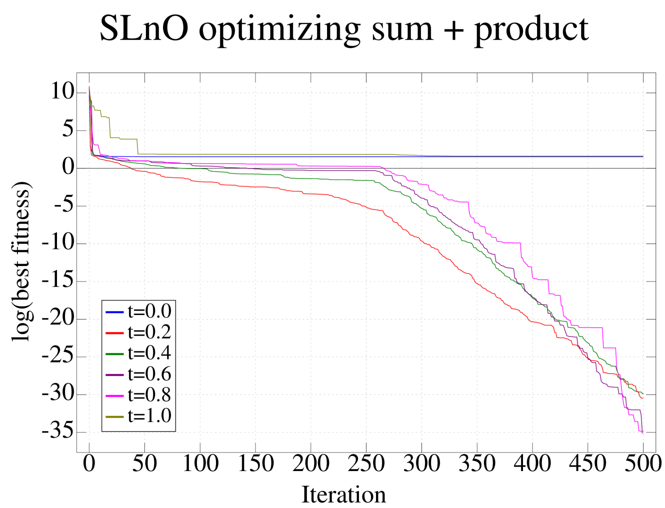

Fig. 1. The optimization process varying various threshold parameters $x < t$ tested in place of $SP_{leader} < 0.25$ for $x \sim [0, 1]$. Optimizing the function $F_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$ from [5] with a population $p = 300$, over $N = 500$ iterations, $n = 30$, and search space $S = [-10, 10]$.

To rebut this confusion and imprecision, we identify explicitly the following shortcomings in the original paper.

1) Calculating SP_{leader}: While the authors give $SP_{leader} = |V_1(1 + V_2)/V_2|$, defining V_1 and V_2 as the speed of sound in water and air, respectively, and compute SP_{leader} as their branch condition [5], they fail to give meaningful definitions of these terms beyond the metaphorical motivation. SLnO implementations replace this computation by various stochastic approximations, either computing $V_1 = \sin 2\pi r$ and $V_2 = \sin 2\pi(1 - r)$ with $r \sim [0, 1]$ [7] or forgoing the original equations for a simple threshold comparison (e.g. $t \geq 0.6$ for $t \sim [0, 1]$ [6]). We produce our own comparison of varying thresholds (see Figure 1), and find that thresholds $t \in [0.1, 0.9]$ seem to work about as well as each other. We expect, however, choice of threshold could be impacted by the function to be optimized, and anticipate future investigation on this front.

2) Normalization: The aforementioned normalization step to ensure the position of all sea lions is noticeably absent from the original paper. Without this step, the positions of sea lions overwhelmingly tend to escape the search space, rendering the resultant optimization useless. This normalization step is thanks to the work of the `mealpy` implementation of SLnO [7].

IV. Conclusion

The Sea Lion Optimization Algorithm provides a competitive metaheuristic which is able to optimize many mathematical functions over high-dimension vector spaces. We provided a more precise formulation, and identified shortcomings in the original formulation.

The fast pace of research in the field of metaheuristic optimization algorithms, both in developing novel algorithms, and iterating and utilizing existing algorithms, seems to leave the field in a precarious position of balancing progress with intelligibility. Papers documenting algorithms ought be self-sufficient; and if not self-sufficient, they ought provide access to their functional code, which can speak for itself.

In light of that, all code used in this paper is available at this GitHub link: https://github.com/ConorOBrien-Foxx/Sea-Lion-Optimizer-Summary.

REFERENCES

[1] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[2] Nidhal El-Omari. Sea lion optimization algorithm for solving the maximum flow problem. 11 2021.

[3] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995.

[4] Kamal Kumar Gola, Nishant Chaurasia, Bhumika Gupta, and Deepak Singh Niranjan. Sea lion optimization algorithm based node deployment strategy in underwater acoustic sensor network. *International Journal of Communication Systems*, 34(5):e4723, 2021. e4723 IJCS-20-1176.R1.

[5] Raja Masadeh, Basel Mahafzah, and Ahmad Sharieh. Sea lion optimization algorithm. *International Journal of Advanced Computer Science and Applications*, 10:388–395, 05 2019.

[6] pfnet-research Team. batch-metaheuristics. https://github.com/pfnet-research/batch-metaheuristics, 2021.

[7] Nguyen Van Thieu. mealpy. https://github.com/thieu1995/mealpy, 2023.

[8] Xin-She Yang. Metaheuristic optimization: Algorithm analysis and open problems. In Panos M. Pardalos and Steffen Rebennack, editors, *Experimental Algorithms*, pages 21–32, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.