

Improved Lion Optimization Algorithm: a new metaheuristic algorithm based on LOA

Deo FETALVERO

April 24, 2019

1 Introduction

An optimization algorithm is a way to maximize or minimize results that would involve one or multiple parameters. For an optimization algorithm to work, it would need a fitness function. A fitness function is a way to define an optimization problem. It takes in the currently worked on solution and describes how good, efficient or optimal that solution is to the problem. After describing those solutions, the algorithm also picks out new solutions and then repeats the process for those solutions until it stops on condition.

Take for an example scheduling jobs for people of different skills. It can be said that different people excels in a job more than others because of their skill and the better they are, the more efficient they work. So, it is imperative that a manager recognize those skills and assign the jobs correctly according to those skills. An optimization algorithm with the right parameters can assign jobs to everyone such that the different skills of people can be best utilized. Being an algorithm not only for Job Scheduling, optimization algorithm can also be used to solve various problems such as data clustering, pattern recognition, tuning of neural networks and so much more.

There are two ways to solve an optimization problem. One way is through using the fitness function's derivative. Upon finding the parameters that make the function's derivative equal to zero, the maximized or minimized solution is found. The first way to solve an optimization problem is derivative-based optimization. Second, another way to solve an optimization problem is described by not knowing what the derivative of the fitness function is. This is done by testing each random solution and informedly picking new ones, which could be called as the way of derivative-free optimization (or black-box optimization).

1.1 Lion Optimization Algorithm

The optimization algorithm that is tackled on in this paper is called the 'Lion Optimization Algorithm'. It is an algorithm inspired by the lifestyle of lions including prides and nomads. While this is also done by the Lion's Algorithm and the Lion Pride Optimizer, the algorithm brings all of the techniques found in the mentioned algorithms all together. The main feature of the Lion Optimization Algorithm is its degree of adaptability tightly coupled on the parameters used. Depending on the fitness function and parameters used, it could find a sweet spot in between to have better performance than other algorithms. An improvement is made to extend the functionality of the Optimization Algorithm. The improvement utilizes more information that is generated within the algorithm to further improve on its functionality.

1.2 No Free Lunch Theorems

Since there are already so many optimization algorithms, what's the point of making a new one? The No Free Lunch Theorems were introduced in 1997 by Wolpert and Macready to address the need for newer optimization algorithms. The first of their theorems states that for any pair of algorithms a_1 and a_2 , iterated m times,

$$\sum_f P(d_m^y|f, m, a_1) = \sum_f P(d_m^y|f, m, a_2),$$

where d_m^y denotes the set of size m of the cost values $y \in Y$ associated to the input values $x \in X$, $f : X \rightarrow Y$ is the function being optimized and $P(d_m^y|f, m, a)$ is the conditional probability of obtaining a given sequence of cost values from algorithm a run m times on function f . Essentially, this says that when all functions f are equally likely to be used, the probability of observing an arbitrary sequence of cost values over the course of optimization is independent of the algorithm.

This theorem indicates that if an algorithm performs better than another algorithm on some class of problems, then it must perform worse on the remaining problems. Consequently, this implies that there is no general-purpose optimization algorithm that is universally superior than the rest. Hence, the development of newer optimization algorithms is still needed.

While the algorithm is an extension to Lion Optimization Algorithm, the new algorithm may work better in some functions than the original all the while performing worse on others. The new algorithm can also be adjusted in such a way that the algorithm works like the original Lion Optimization Algorithm.

2 Improvements

This is a research of a multi-part improvement to the Lion Optimization Algorithm. The improvement is done across multiple sections of the algorithm to improve the overall performance of every run. Most of this improvements are also modeled heuristically after lions, their influence to another, their nature and broadly, evolution in general. While the Lion Optimization Algorithm includes ways to slightly modify its behavior, there are still rooms for improvement for the algorithm which will be talked about in the next sections.

2.1 Group Direction in Prides

The best position in the pride influences where lions in a pride would roam. When doing roaming, female and male lions would have a bit of influence from the direction and distance to the best position in the pride.

This group influence is seen among lions as peers tend to swarm with each other and most lions that stray away from the pride will get attracted to where the most of his peers are located. [3]

Similar to a group best in the Particle Swarm Optimizer (PSO), the group best in a Lion Pride will be a reference to where the lions in the pride are influenced to go to. [4] Depending on a percentage variable, the influence of this global best in the path of roaming lions will vary.

A new variable will be introduced, %I which will determine how much the direction of the lion will be influenced by the direction to the best position. The new modified roaming equation would be:

$$\text{Lion}' = \text{Lion} + 2D \cdot \text{rand}(0, 1)(R1 \cdot (1 - \%I) + R3 \cdot \%I) + U(-1, 1) \cdot \tan(\theta) \times D \times R2$$

where $R1 \cdot R2 = 0, ||R2|| = 1$

where Lion and Lion' is the previous and next position of the lion, respectively, and D is the distance between the lion's position and the selected point chosen by tournament selection in the pride's territory. The following figure shows the range of possible next positions of the lion.

A newly included variable $R3$ will represent the direction vector from the Lion's position to the best position in the pride. $R3$ can be represented by:

$$R3 = \frac{(\text{GBest} - \text{Lion})}{||\text{GBest} - \text{Lion}||}$$

where GBest is the best position in the pride.

2.2 Fitness Weighted Mating

Averaging between males in preparation for mating can be improved by weighing fitness values so that a better gene could be created to be used in mating with a female. The best male lion among suitors in mating will have more influence on the traits of the offsprings.

To produce better offsprings, nature has always arranged the better fit organisms to survive. In order to find better offsprings, female organisms would look for better fit organisms among the crowd to mate. To better model this trait in mating between multiple male lions to a female lion. The gene of the best male lion should better influence the gene of the offsprings.

To simulate this effect, an equation similar to inverse distance weighting [1] is created that instead uses fitness difference as basis to create a position that has a weighted average that relies more on better fit positions from multiple males. [6]

$$\text{Lion Average} = \frac{\sum_{i=1}^n \left(\frac{\text{Lion}_i}{f_{\text{fem}} - f(\text{Lion}_i)} \right)}{\sum_{i=1}^n \left(\frac{1}{f_{\text{fem}} - f(\text{Lion}_i)} \right)}$$

where n is the number of male lions, Lion_i is the male lion's best position, f_{fem} is the mating female lion's current fitness and Lion Average is the weighted average of the positions between the male lions based on their fitness.

2.3 Simulated Annealing in Nomads

A simple mechanism would be added to not accept lions fit below a lower bound in nomad lions. The lower bound would be updated regularly such that lions that are bound to become nomad would be removed from the set of solutions if they do not meet the current lower bound of the nomads.

Nomad lions have to rely on themselves. They don't have territories, they are always moving and they need to adapt to changes to the environment otherwise they die from it. This is a defining process of evolution. [2]

To simulate this trait in the algorithm, a lower bound defined by the current least fit nomad is made. Similar to simulated annealing, in every iteration this lower bound is updated and lions that are to be added to nomads are checked if they're fitness is greater than the lower bound fitness (meaning they are less fit). [5] If they are less fit, they are removed from the whole set of solutions. The procedure is as follows:

$$\begin{cases} \text{Remove Lion} & \text{if } f(\text{Lion}) > f_{LB} \\ \text{Add Lion to Nomads} & \text{otherwise} \end{cases}$$

where f_{LB} is the current lower bound fitness of the nomad group.

2.4 Improved Ranked Selection Randomization

In selecting lions in a group sorted by fitness, with the best fit at the least index, the random selection of lions listed in a ranked order can be improved. By raising the random function that selects a number between 0 to 1 (inclusive) to a power k there will be higher chances for lower indexes in the list which represents better fit lions to be selected.

The new algorithm should adopt a new random selection function that allows to select more higher fit lions than other lesser lions. The new $rand(0, 1)$ function will be modified for a new function

$$rand(0, 1, K) = U(0, 1)^K$$

where $U(0, 1)$ is a random number between 0 to 1 with uniform distribution.

The new equation creates a random number between 0 to 1 with **exponential distribution** such that lower values will tend to have higher spawn rate than higher values.

A new variable K will be introduced to the algorithm. This will represent the degree to which the random selection curve will tend to lean. When $K > 1$, the random selection function will produce more lower indices that has better fit lions. When $K = 1$, the function will be back to its normal setting. When $K < 1$, the function will produce more higher indices.

2.5 Center Based Per Axis Randomization

Generating new points can be improved by randomizing based on a central point in the search space. Points far from the center point should be generated at lesser quantities and points near the center should be generated at greater quantities. **This section focuses on a new center based randomization method that improves finding more points that are either near or far a “hotspot” or a central point [3].**

In the algorithm, nomads either stay in their positions doing nothing or reset to a new position in every iteration. This can be improved by introducing a new randomization method that can randomize a new position based on the previous position. In this improvement, Nomad lions either roam around a place to “stay” or find a new place.

The nomad lion roaming can be improved by changing the nomad lion roaming equation to

$$\text{Lion}'_{ij} = \begin{cases} \text{RANDC}(\text{Lion}_{ij}) & \text{if } \text{rand}(0, 1) > pr_i \\ \text{RAND}_j & \text{otherwise} \end{cases}$$

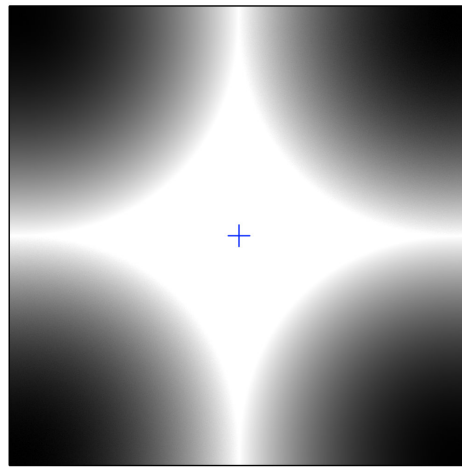
such that RANDC is a new function to be added to the algorithm such that:

$$\begin{aligned} \text{RANDC}(\text{Lion}_{ij}) = & \text{Lion}_{ij} - (\text{Lion}_{ij} - \text{LB}) \cdot |\min(0, U_{-1 \text{ to } 1})|^{\text{deg}} \\ & + (\text{UB} - \text{Lion}_{ij}) \cdot \max(0, U_{-1 \text{ to } 1})^{\text{deg}} \end{aligned}$$

where LB is the lower bound or the “minimum” of the search space, UB is the upper bound or the “maximum” of the search space, deg is the degree of nearness of the generated points to the center and $U_{-1 \text{ to } 1}$ is a random number between -1 to 1 with uniform random distribution.

The points that are near to the axis of the center point are more likely to be generated than those that are far from the center point when the ‘deg’ is greater than 1. When ‘deg’ is less than 1, points far from the center point are generated more while when ‘deg’ is 1, it is generating a uniform random number only.

The following figure shows generating random points using the mentioned technique with different centers and degree of nearness.



Center Based Randomization
Heatmap

+ Center Point

Figure 1: Heat map of a 2D area of points that are more likely to less likely to be generated from light to dark based on a center point

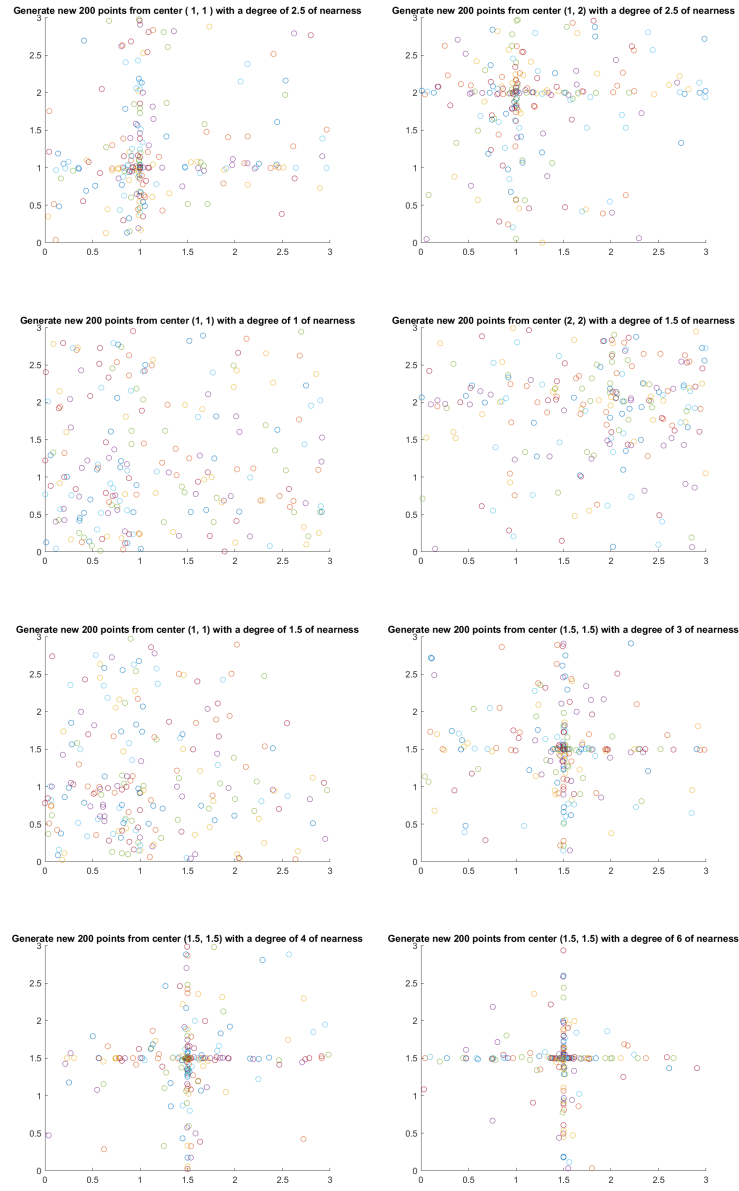


Figure 2: Center Based Random Points with Varying Centers and Nearness

3 Modified Program Flow for the New Algorithm

Combining together every improvement to the Lion Optimization Algorithm, the new flow of the new algorithm, that can be now called as the Improved Lion Optimization Algorithm (iLOA), is as follows:

Improved Lion Optimization Algorithm pseudo code

- a. Generate random sample of Lions N_{pop} (N_{pop} is number of initial population).
- b. Initiate prides and nomad lions
 - Randomly select %N (Percent of lions that are nomad) of initial population as nomad lion. Partition remained lions into P (P is number of prides) prides randomly, and formed each prides territory.
 - In each pride %S (Sex rate) of entire population are known as females and the rest as males. This rate for each sex in nomad lions is used vice-versa.
- c. For each pride do
 - Some randomly selected female lion go hunting.
 - Each of remained female lion in pride go toward one place in the territory selected randomly but with bias to the better places (Ranked Selection Random). The direction of each female would be influenced by the direction to the best place in the territory by %I (Group Influence)
 - In pride, for each resident male; %R (Roaming percent) of territory randomly are selected with bias to the better places (Ranked Selection Random) and then checked with the influence of the best place in the territory by %I (Group Influence).
 - %Ma (Mating probability) of females with bias to the best females (Ranked Selection Random) in pride mate with one or several resident male. The stronger the male lions gene, the better it influences the offspring (Fitness Weighted Mating). New cubs become mature.
 - Weakest male drive out from pride and checked whether it is within the lower bound fitness of nomads.
 - Joins nomads, otherwise, killed (Simulated Annealing)
- d. For Nomad do
 - Nomad lion (both male and female) moving randomly in search space. Every random is centered on the best nomad lion in the group and biased near it (Center Based Randomization).
 - %Ma (Mating probability) of nomad Female mate with one of the best nomad male. New cubs become mature.
 - Prides randomly attacked by nomad male.
- e. For each pride do
 - Some female with I rate ((Immigrate rate)) immigrate from pride and become nomad.
- f. Do
 - First, based on their fitness value each gender of the nomad lions are sorted. After that, the best females among them are selected and distributed to prides filling empty places of migrated females.
 - With respect to the maximum permitted number of each gender, nomad lions with the least fitness value will be removed.

If termination criterion is not satisfied, then go to step c

4 Testing against LOA

Both algorithms are tested head to head through various benchmarking functions which includes: Griewank 1D and 2D, Rastrigin 1D and 2D, Rosenbrock 2D, Parabola and Paraboloid. Each function use different dimensional space but all run for 50 iterations with 50 population and the following parameters:

Number of Prides = 4 Nomad Percentage (%N) = 0.2 Roaming Percentage (%R) = 0.2 Sex Percentage (%S) = 0.8 Mating Rate (%Ma) = 0.3 Mutation Probability = 0.2 Immigration Rate = 0.4 – More parameters for iLOA – Percent Group Influence = 0.4 Annealing = True Ranked Selection Pressure = 2 Near to Best Random Pressure = 2

4.1 Griewank 1D

The Griewank function is a function that is typically used for testing optimization. The function is defined by:

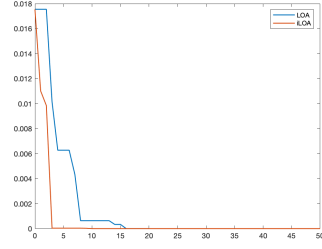
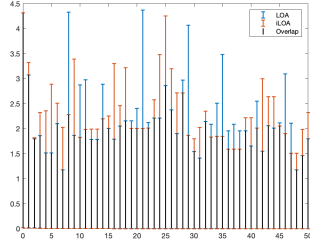
$$f_1(x) = 1 + (1/4000) \cdot x_1^2 - \cos(x_1)$$

It has multiple maxima and minima but its global minima is at $x = 0$.

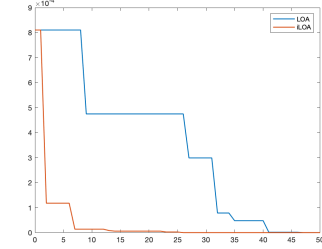
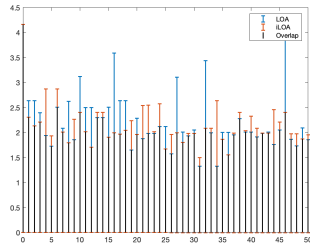
Both functions are tested five times with the Griewank function with the same starting random population and a dimensional space of $[-100, 100]$.

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
LOA End Fitness	1.5846E-07	1.2658E-07	7.4251E-08	3.5575E-06	5.4992E-08
LOA Evaluations	3499	3453	3465	3560	3468
iLOA End Fitness	9.3769E-10	3.1541E-13	4.5715E-11	1.8532E-09	2.1836E-12
iLOA Evaluations	3042	2803	2710	2718	2857

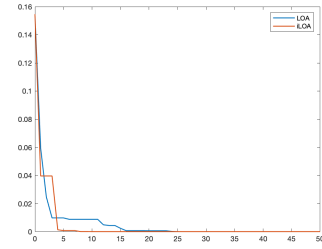
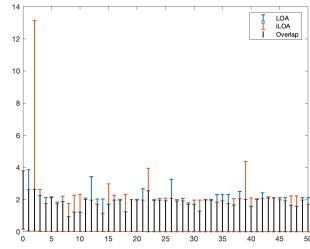
Table 1: LOA vs. iLOA: Griewank 1D (f_1)



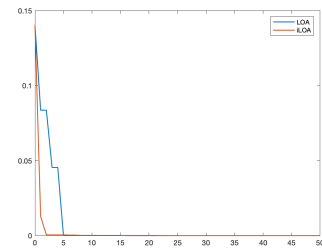
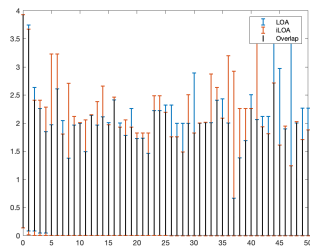
(a) Trial 1: Fitness Range (y) over Iterations (x) (b) Trial 1: Minimum Fitness (y) over Iterations (x)



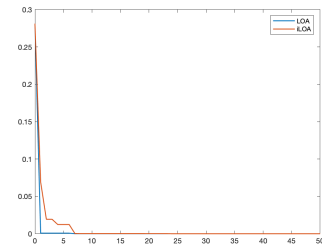
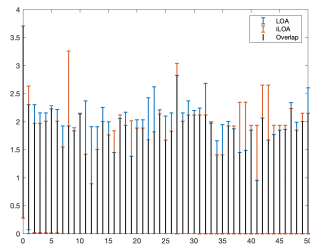
(c) Trial 2: Fitness Range (y) over Iterations (x) (d) Trial 2: Minimum Fitness (y) over Iterations (x)



(e) Trial 3: Fitness Range (y) over Iterations (x) (f) Trial 3: Minimum Fitness (y) over Iterations (x)



(g) Trial 4: Fitness Range (y) over Iterations (x) (h) Trial 4: Minimum Fitness (y) over Iterations (x)



(i) Trial 5: Fitness Range (y) over Iterations (x) (j) Trial 5: Minimum Fitness (y) over Iterations (x)

4.2 Griewank 2D

The 2 dimensional form of the Griewank is the Griewank 2D function. The function is defined by:

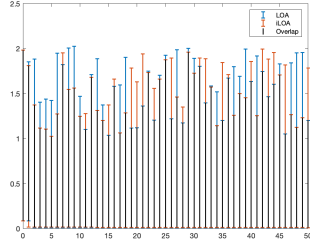
$$f_2(x) = 1 + \frac{1}{4000}x_1^2 + \frac{1}{4000}x_2^2 - \cos(x_1) \cos\left(\frac{1}{2}x_2\sqrt{2}\right)$$

It also has multiple maxima and minima and its global minima is also at $x = 0$.

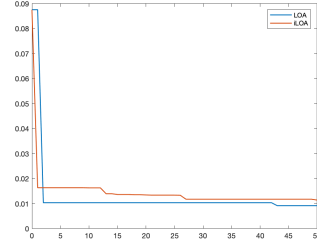
Both functions are tested five times with the Griewank 2D function with the same starting random population and a dimensional space of $[-100, 100]$.

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
LOA End Fitness	0.0091998	0.010329	0.0069254	0.0010238	0.01839
LOA Evaluations	3440	3468	3508	3525	3476
iLOA End Fitness	0.011425	0.00038845	0.00004408	0.00011175	0.011947
iLOA Evaluations	2609	2588	2705	2554	2649

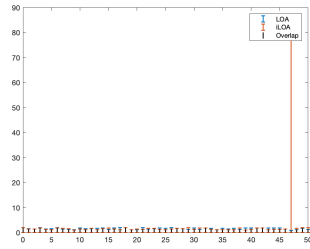
Table 2: LOA vs. iLOA: Griewank 2D (f_2)



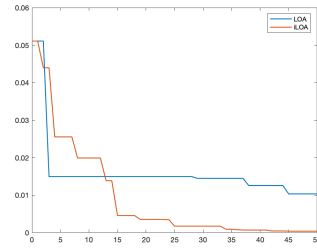
(a) Trial 1: Fitness Range (y) over Iterations (x)



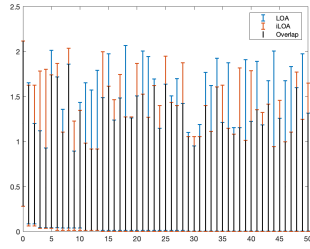
(b) Trial 1: Minimum Fitness (y) over Iterations (x)



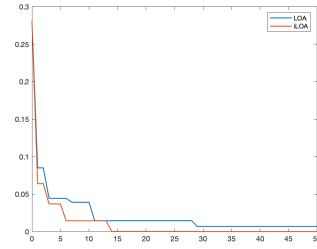
(c) Trial 2: Fitness Range (y) over Iterations (x)



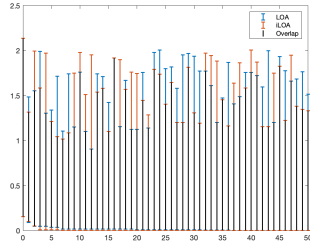
(d) Trial 2: Minimum Fitness (y) over Iterations (x)



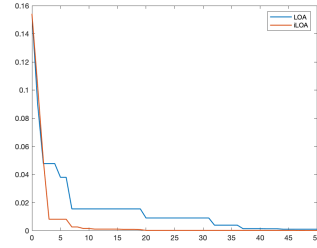
(e) Trial 3: Fitness Range (y) over Iterations (x)



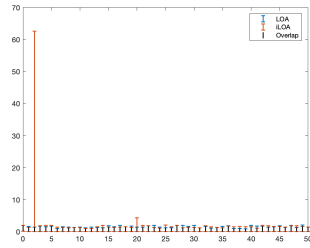
(f) Trial 3: Minimum Fitness (y) over Iterations (x)



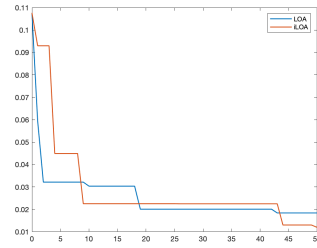
(g) Trial 4: Fitness Range (y) over Iterations (x)



(h) Trial 4: Minimum Fitness (y) over Iterations (x)



(i) Trial 5: Fitness Range (y) over Iterations (x)



(j) Trial 5: Minimum Fitness (y) over Iterations (x)

4.3 Rastrigin 1D

Rastrigin is a non-convex function first proposed by Rastrigin as a 2-dimensional function and then generalized later on to multiple dimensions. The one-dimensional version of the function is defined by:

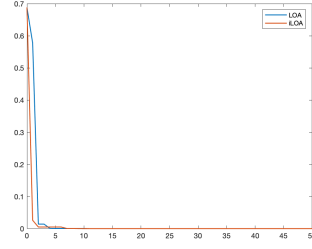
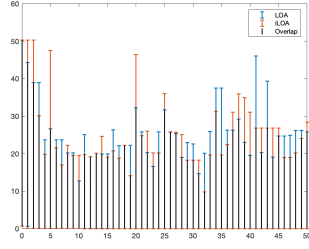
$$f_3(x) = 10 + x_1^2 - 10 \cos(2\pi x_1)$$

It has multiple maxima and minima and its global minima is also at $x = 0$.

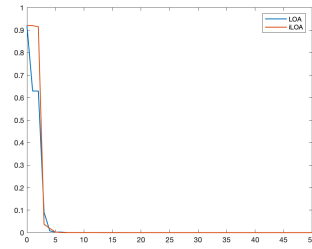
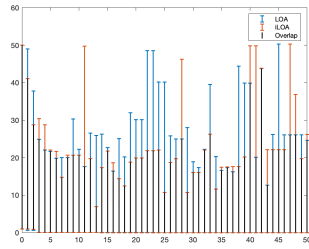
Both functions are tested five times with the Rastrigin 1D function with the same starting random population and a dimensional space of $[-2\pi, 2\pi]$.

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
LOA End Fitness	3.2384E-09	2.4366E-09	2.7632E-08	3.3077E-08	2.8756E-08
LOA Evaluations	3401	3296	3442	3423	3379
iLOA End Fitness	1.7764E-15	3.5527E-15	3.6447E-09	2.2848E-10	2.8727E-11
iLOA Evaluations	2559	2631	2664	2694	2919

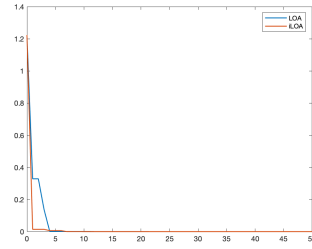
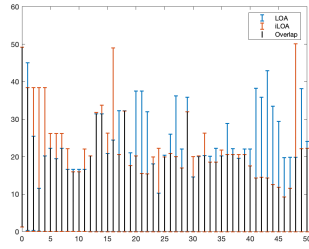
Table 3: LOA vs. iLOA: Rastrigin 1D (f_3)



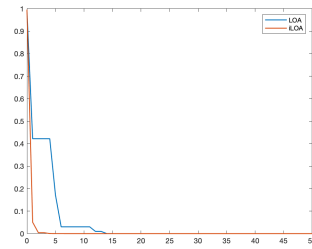
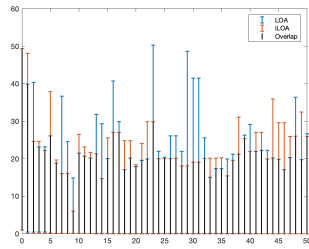
(a) Trial 1: Fitness Range (y) over Iterations (x) (b) Trial 1: Minimum Fitness (y) over Iterations (x)



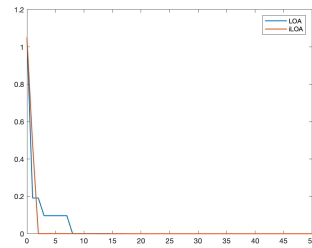
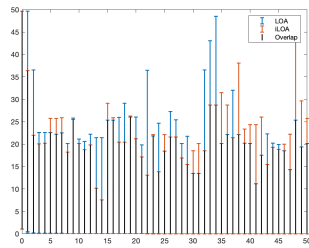
(c) Trial 2: Fitness Range (y) over Iterations (x) (d) Trial 2: Minimum Fitness (y) over Iterations (x)



(e) Trial 3: Fitness Range (y) over Iterations (x) (f) Trial 3: Minimum Fitness (y) over Iterations (x)



(g) Trial 4: Fitness Range (y) over Iterations (x) (h) Trial 4: Minimum Fitness (y) over Iterations (x)



(i) Trial 5: Fitness Range (y) over Iterations (x) (j) Trial 5: Minimum Fitness (y) over Iterations (x)

4.4 Rastrigin 2D

Rastrigin is a non-convex function first proposed by Rastrigin as a 2-dimensional function and then generalized later on to multiple dimensions. The function is defined by:

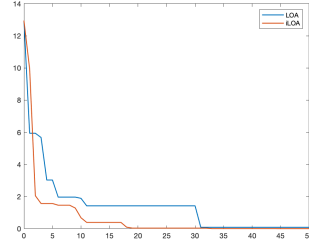
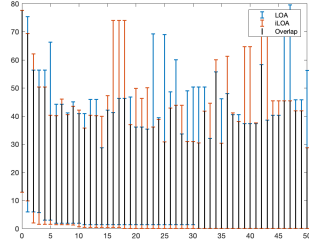
$$f_4(x) = 20 + x_1^2 - 10 \cos(2\pi x_1) + x_2^2 - 10 \cos(2\pi x_2)$$

It has multiple maxima and minima and its global minima is also at $x = 0$.

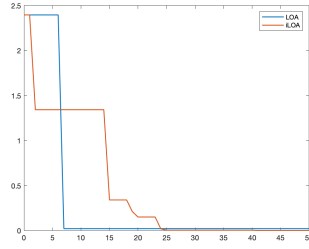
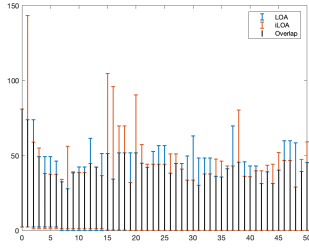
Both functions are tested five times with the Rastrigin 2D function with the same starting random population and a dimensional space of $[-2\pi, 2\pi]$.

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
LOA End Fitness	0.083296	0.022199	0.001943	0.0015496	0.30548
LOA Evaluations	3314	3513	3443	3536	3496
iLOA End Fitness	0.00022399	0.0015992	0.0012708	0.40278	3.1202E-05
iLOA Evaluations	2811	2767	2711	2743	2894

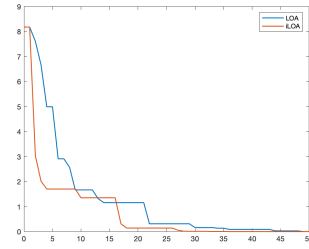
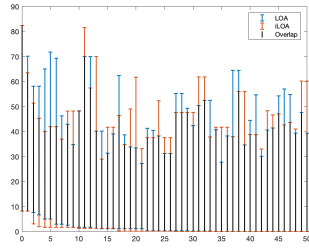
Table 4: LOA vs. iLOA: Rastrigin 2D (f_4)



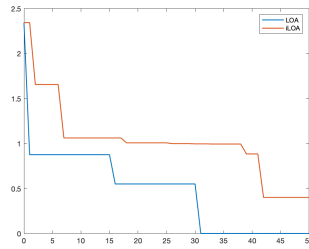
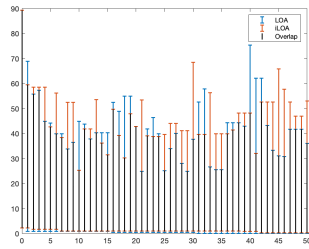
(a) Trial 1: Fitness Range (y) over Iterations (x) (b) Trial 1: Minimum Fitness (y) over Iterations (x)



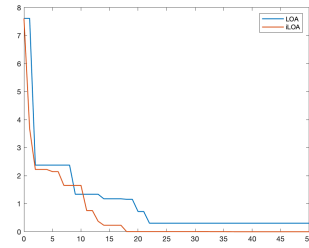
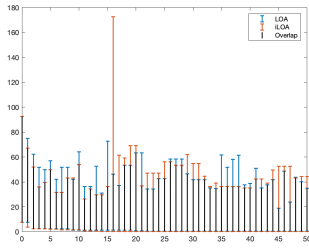
(c) Trial 2: Fitness Range (y) over Iterations (x) (d) Trial 2: Minimum Fitness (y) over Iterations (x)



(e) Trial 3: Fitness Range (y) over Iterations (x) (f) Trial 3: Minimum Fitness (y) over Iterations (x)



(g) Trial 4: Fitness Range (y) over Iterations (x) (h) Trial 4: Minimum Fitness (y) over Iterations (x)



(i) Trial 5: Fitness Range (y) over Iterations (x) (j) Trial 5: Minimum Fitness (y) over Iterations (x)

4.5 Rosenbrock 2D

Rastrigin is a non-convex function proposed by Howard H. Rosenbrock as a 2-dimensional function for testing optimization and then generalized later on to multiple dimensions. The function is defined by:

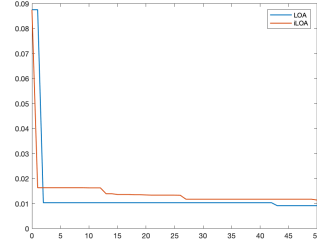
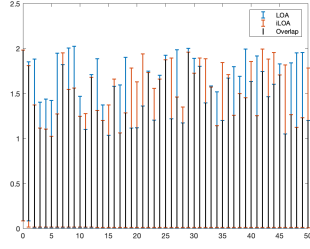
$$f_5(x) = (a - x_1)^2 + b(x_2 - x_1^2)^2$$

where $a = 1$ and $b = 100$. It has multiple maxima and minima and its global minima is at $x = \langle a, a^2 \rangle$ or $\langle 1, 1 \rangle$.

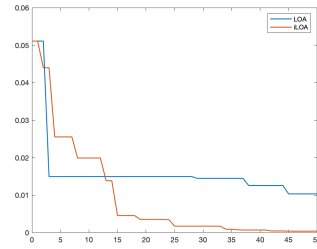
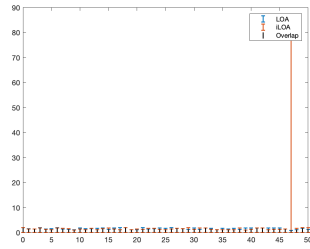
Both functions are tested five times with the Rosenbrock 2D function with the same starting random population and a dimensional space of $[-5, 5]$.

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
LOA End Fitness	0.0049932	0.00015138	0.002005	0.0018016	0.000050672
LOA Evaluations	3540	3277	3263	3260	3284
iLOA End Fitness	0.015473	0.00066979	0.00023044	0.000033068	5.8865E-06
iLOA Evaluations	2688	2873	2786	2694	2858

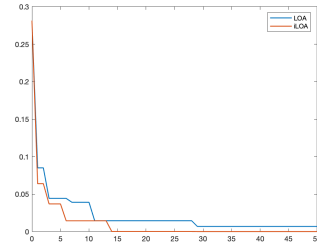
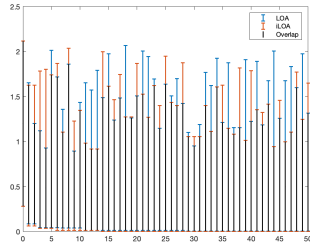
Table 5: LOA vs. iLOA: Rosenbrock 2D (f_5)



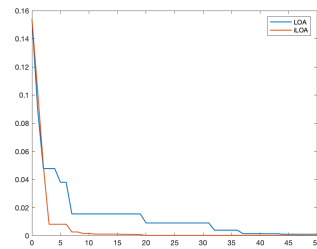
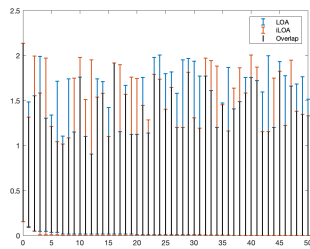
(a) Trial 1: Fitness Range (y) over Iterations (x) (b) Trial 1: Minimum Fitness (y) over Iterations (x)



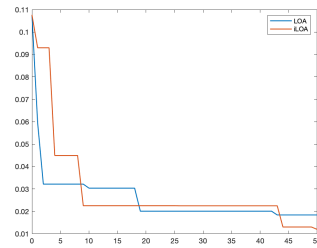
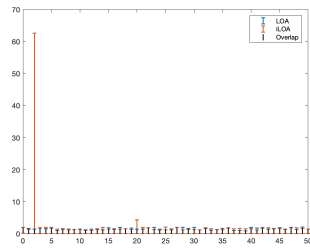
(c) Trial 2: Fitness Range (y) over Iterations (x) (d) Trial 2: Minimum Fitness (y) over Iterations (x)



(e) Trial 3: Fitness Range (y) over Iterations (x) (f) Trial 3: Minimum Fitness (y) over Iterations (x)



(g) Trial 4: Fitness Range (y) over Iterations (x) (h) Trial 4: Minimum Fitness (y) over Iterations (x)



(i) Trial 5: Fitness Range (y) over Iterations (x) (j) Trial 5: Minimum Fitness (y) over Iterations (x)

5 Parameter Testing for iLOA

Parameter tests are done for each of the aforementioned functions by performing an exhaustive search on the following parameter points:

Nomad Percentage (%N) = {0.2, 0.4, 0.6, 0.8}
Roaming Percentage (%R) = {0.2, 0.4, 0.6, 0.8}
Sex Percentage (%S) = {0.2, 0.4, 0.6, 0.8}
Mating Rate (%Ma) = {0.3, 0.6}
Mutation Probability = {0.2, 0.4, 0.6, 0.8}
Immigration Rate = {0.2, 0.4, 0.6, 0.8}
Percent Group Influence = {0.2, 0.4, 0.6, 0.8}
Ranked Selection Pressure = {2, 3}
Near to Best Random Pressure = {2, 3}

Each of the functions is run using every possible set of parameters using the above parameter points and their outputs recorded so that the best parameter set is found.

Each function is tested against iLOA with every parameter set running 5 times each for redundancy. The iterations are limited to 30, includes 4 prides and a population of 50 for the parameters.

5.1 Test Limitations

The platform used for both the algorithm code and parameter tester, MATLAB, has its own limitations of number representation when representing float values of numbers with decimal places with more than 16 binary digits (values smaller than 2^{-16}). Now, all of one-dimensional Griewank, one-dimensional Rastrigin and two-dimensional Rastrigin has encountered this limitations such that some of their fitness data represented results being zero upon data collection.

Instead of using the mentioned functions, the three-dimensional Griewank, three-dimensional Rastrigin and four-dimensional Rastrigin functions are used. These functions not only are similar to their replaced counterparts but have larger search space than their counterparts.

6 Rastrigin Function

The Rastrigin function, as defined in the earlier sections, is a non-convex function with a multi-dimensional form:

$$f(x) = An + \sum_{i=1}^n [x_i^2 - A \cos(2\pi x_i)]$$

6.1 Three-Dimensional Rastrigin Function

The parameter set found to have the best average fitness for the three-dimensional Rastrigin is the parameters

Nomad Percentage (%N) = 0.2
Roaming Percentage (%R) = 0.8
Sex Percentage (%S) = 0.2
Mating Rate (%Ma) = 0.6
Mutation Probability = 0.8
Immigration Rate = 0.8
Percent Group Influence = 0.6
Ranked Selection Pressure = 2
Near to Best Random Pressure = 3
Parameter Set Average Fitness: 2.8400e-15

6.2 Four-Dimensional Rastrigin Function

The parameter set found to have the best average fitness for the four-dimensional Rastrigin is the parameters

Nomad Percentage (%N) = 0.4
Roaming Percentage (%R) = 0.8
Sex Percentage (%S) = 0.8
Mating Rate (%Ma) = 0.6
Mutation Probability = 0.6
Immigration Rate = 0.6
Percent Group Influence = 0.8
Ranked Selection Pressure = 2
Near to Best Random Pressure = 3
Parameter Set Average Fitness: 0.1499

While the three-dimensional function parameters has a low sex percentage (Percent females in prides, Percent males in nomads), the four-dimensional function paramaters does the opposite with a high sex percentage while balancing it with an increased nomad percentage. The four-dimensional function parameters also tries to decrease its mutation probability to focus on not losing the current solutions in the larger search space and improve those solutions for better ones. Also since the Rastrigin function has many optimas, the “Near to Best” Random Pressure is increased such that the solutions generated are near

each other such that those new solutions likely would be generated next to an optima.

7 Griewank Function

7.1 Two-Dimensional Griewank

The parameter set found to have the best average fitness for the Two-Dimensional Griewank are the parameters

Nomad Percentage (%N) = 0.2
Roaming Percentage (%R) = 0.8
Sex Percentage (%S) = 0.2
Mating Rate (%Ma) = 0.6
Mutation Probability = 0.8
Immigration Rate = 0.2
Percent Group Influence = 0.8
Ranked Selection Pressure = 2
Near to Best Random Pressure = 2
Parameter Set Average Fitness: 1.1100e-16

7.2 Three-Dimensional Griewank

The parameter set found to have the best average fitness for the Three-Dimensional Griewank are the parameters

Nomad Percentage (%N) = 0.2
Roaming Percentage (%R) = 0.6
Sex Percentage (%S) = 0.2
Mating Rate (%Ma) = 0.3
Mutation Probability = 0.4
Immigration Rate = 0.8
Percent Group Influence = 0.2
Ranked Selection Pressure = 3
Near to Best Random Pressure = 3
Parameter Set Average Fitness: 0.0064

Notice that even using the same function design, the two-dimensional and the three-dimensional Griewank still doesn't have alike parameter sets. This is likely due to the search space of the three-dimensional Griewank being larger than that of the the two-dimensional Griewank. Due to the search space being larger, the three-dimensional function used a less arbitrary point generating parameter set for iLOA so that it could improve only the best solutions when found.

8 Rosenbrock Function

The parameter set found to have the best average fitness for the Rosenbrock Function are the parameters

Nomad Percentage (%N) = 0.2
Roaming Percentage (%R) = 0.8
Sex Percentage (%S) = 0.6
Mating Rate (%Ma) = 0.6
Mutation Probability = 0.2
Immigration Rate = 0.8
Percent Group Influence = 0.8
Ranked Selection Pressure = 2
Near to Best Random Pressure = 3
Parameter Set Average Fitness: 3.3754e-14

The above parameter set focuses on more information sharing between points and less random generation of solutions. While there is a high roaming percentage, the nomads for this parameter set are minimal. The high immigration rate and percent group influence increases the chances for prides to exchange information within and to other prides and the minimal mutation probability decreases the randomness of solutions.

Since the Rosenbrock function's surface has a ridge where the fitness tends to get better, this parameter set gives the algorithm a chance to focus on not losing more solutions and focus on improving the solutions that get to the narrow window of high fitness that is found on the ridge of the Rosenbrock function's surface while also influencing other solutions to reach the area.

8.1 Parameter Testing Conclusion

There are different parameter sets that perform better for each function. This only implies that the No Free Lunch Theorem is at work. Having different parameter sets makes each instance of iLOA algorithm different from other instances with a different parameter set. Each function works best with a certain instance of iLOA with a certain parameter set.

The previous test proves that smaller search spaces tends to have a higher randomness in its parameter set since smaller spaces are easy to scan through for the best points and that larger search spaces decrease its randomness since larger spaces are harder to scan through such that improving on the best solutions so far is more convenient.

The parameter set also depends on how many optima that can be found from the functions. High count optima functions like Rastrigin tend to favor randomness more while low count optima functions like Rosenbrock tend to favor selection improvement more over randomness.

Taking the most common best parameter points that were used by the functions, the most common parameter setup for iLOA is:

Nomad Percentage (%N) = 0.2 (ave 0.24)
Roaming Percentage (%R) = 0.8 (ave 0.76)
Sex Percentage (%S) = 0.4 (ave 0.4)
Mating Rate (%Ma) = 0.5 (ave 0.54)
Mutation Probability = 0.6 (ave 0.56)
Immigration Rate = 0.6 (ave 0.64)
Percent Group Influence = 0.6 (ave 0.64)
Ranked Selection Pressure = 2 (ave 2.3)
Near to Best Random Pressure = 3 (ave 2.8)

Since most of the functions used in the tests is of type high count optima, the above parameter set is biased to high count optima functions.

References

- [1] NSF National Center for Geographic Information and Analysis interpolation: Inverse distance weighting. <http://www.ncgia.ucsb.edu/pubs/spherekit/inverse.html>. Accessed: 2019-02-05.
- [2] Charles Darwin. On the origin of species by means of natural selection or the preservation of favoured races in the struggle for life. page 72, 1969.
- [3] R.; C. Packer Heinsohn. Complex cooperative strategies in group-territorial african lions. pages 1260–1262, 1995.
- [4] R. Kennedy, J.; Eberhart. Particle swarm optimization. 4:1942–1948, 1995.
- [5] C. D.; Vecchi M. P Kirkpatrick, S.; Gelatt Jr. Optimization by simulated annealing. pages 671–680, 1983.
- [6] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. pages 517–524, 1995.