Parallel Approach to the Firefly Algorithm

Gabriel Linhares
Computer Science Dep.
Centro Universitário FEI
São Bernardo do Campo, Brazil
elinhares@outlook.com.br

Guilherme Costa
Computer Science Dep.
Centro Universitário FEI
São Bernardo do Campo, Brazil
gui.costa200897@gmail.com

Leonardo Jacobson
Computer Science Dep.
Centro Universitário FEI
São Bernardo do Campo, Brazil
unifIteixeira@fei.edu.br

Lucio Rodrigues
Computer Science Dep.
Centro Universitário FEI
São Bernardo do Campo, Brazil
lucior.jr@gmail.com

Vitor Ballabenute
Computer Science Dep.
Centro Universitário FEI
São Bernardo do Campo, Brazil
vitor.ballabenute@gmail.com

Paulo Sergio Rodrigues
Computer Science Dep.
Centro Universitário FEI
São Bernardo do Campo, Brazil
psergio@fei.edu.br

Guilherme Alberto Wachs-Lopes
Computer Science Dep.
Centro Universitário FEI
São Bernardo do Campo, Brazil
gwachs@fei.edu.br

***Abstract*—Distributed computing is a form of computation in which many calculations are made at the same time, operating under the understanding that big problems can sometimes be divided into little ones, that them are solved at the same time. This paper has the objective of introducing the distributed computing concept to an optimized version of the Firefly Algorithm (FA) and based on the results discuss if the proposed distributed version shows itself to be superior or inferior to the regular existing algorithm.**

***Index Terms*—Optimization, Distributed Computing, Bio-Inspired Algorithm, Firefly**

I. Introduction

Distributed's computing, also known as Grid computing main purpose is to compute faster than what a single computer can by using multiple computer resources working concurrently. The constant need for faster solutions and even solving larger problems has had a huge influence on almost every activity computing related. These problems can cover a wide variety of applications including modeling, weather prediction, fluid dynamics and the simulation of a large system like a Swarm Intelligence tool, in this case, the Firefly Algorithm. The purpose of this paper is to analyze the real speed of a serial firefly algorithm adapted for MLOTP purposes and conclude if distributing it is a viable choice, and them, analyze the performance of the distributed algorithm. It will be construed some crucial points of the algorithm that have big influence to its performance, as the number of fireflies and number of thresholds used. This analysis will be done in both the serial and distributed algorithms and the Berkeley's Database (Session III) will be used as the image base to the analysis.

II. Firefly Algorithm

A. Swarm Intelligence

The term Swarm Intelligence comes from the sense of cellular robotic systems consisting of simple agents that organize themselves through neighborhood interactions [1].

Swarm Intelligence (SI) is part of the artificial intelligence (AI) area, inspired by the collective behavior of social swarms like ants, bees, fish, worms; such as, when to reproduce, where to live, where to forage, directions to take, or how to divide the necessary tasks among each other. Those type of decisions are made based on local information from its individuals that are gathered from their interactions with the environments intermediating [1].

The coordinated behavior of this swarms directs them to their desired goals, and this coordinated behavior within self-organized and decentralized systems is the concern of the Swarm Intelligence research field [2].

B. Overview

Firefly is one the Swarm Intelligence algorithms proposed by Xin-She Yang in [3] and is a stochastic, nature-inspired, meta-heuristic method that has been applied to solve a lot of the hard optimization problems. Belongs to the stochastic algorithms, meaning that it uses a randomization while searching for a set of solutions. Heuristic comes from the fact that it tries to find solutions by trial and error. And meta-heuristic is the method to solve in a generic form the optimization problems (also NP-hard problems). Its inspiration comes from the lights of fireflies in nature and their behavior based on it.

C. Problem Solving

Methods of Swarm Intelligence are recently being used in optimization, some examples of swarm intelligence opti-

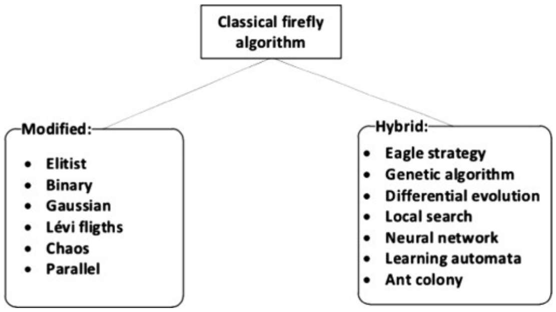

Fig. 1. Some improvements made by community to the Firefly Algorithm [6]

mization methods are: ant colony optimization (ACO), particle swarm optimization (PSO), artificial bee colony (ABC), bacterial colony optimization (BCO), cuckoo search, the bat algorithm, to name a few. Some of those optimization methods are used nowadays in fields like robot controlling, routing and load balancing in new-generation mobile telecommunication networks, which demands robustness and flexibility [4]. The Firefly Algorithm among others like algorithms for clustering have emerged over time and shown its efficiency and reliability in multiple cases. The Firefly Algorithm (FA) often guarantees that the obtained results will meet the expectation and can be applied to every problem domain.

D. MLOTP Adapted Firefly Algorithm

Knowing that FA is based on bio-inspiration algorithms and how it's functionality is based, the algorithm can be used in the following application: to find the best cut in a multi-thresholding scenario that will divide the curve provided by a histogram of pixel values in scale of 0 (black) to 255 (white) from an image to have a partition of it in multiple regions (disjoint pixels). One can think that an image could have a focused foreground plane and a background plane, and would like to divide or segment them, but consider also that, this image does not have only one or two plans, it could be a multi-plane image.

The purpose of Firefly in this study is to segment these images according to each plane of each picture, doing this through calculations and assessments leading by trial and error results (normal behavior of Firefly algorithm), but doing this in a different way, parallelizing special functions of Firefly to reduce the cost of time searching the best solution. Part of our vision is focused on using the following points to consider an improvement of the time reduction cost. Some points are listed in the following image, but some strengths points that we are inspired in this study and that we are looking for are Parallel Processing, Neural Network, Genetic Algorithm and Ant colony [5].

We can claim that a brute-force algorithm with the same purpose can find almost exact value for the best cut, however, it is an algorithm that carries out in a long cost of time in relation to Firefly algorithm, thus, may not be feasible when it comes to performance. The amount of "points of interest" in an image are defined by how many thresholds are going to be searched.

The operation of Firefly algorithm occurs as follows: random threshold values are associated with each firefly in a d-dimensional vector f (each d-dimension refers to a value of threshold), after this step is done, the algorithm goes into a loop with limited number of interactions (max generation) that will be defined by the one using the algorithm, - it is notable that the effects are, primarily, the cost of time and accuracy - so it does the Tsallis entropy calculation (whose chosen method for this type of calculation maximizes the quality and accuracy of the result) to be able to rank and evaluate the brightness of each firefly [7].

Having the first ranking vector (in descending order) after the evaluation of all the fireflies, and the algorithm have made one of its cycles in the interaction, we can rely on the fireflies that are in the top of the ranking. All the fireflies around (considering their positions in a Cartesian plane) will be pointing to the firefly with the best evaluation (more brilliant), so the next step is to copy the values of thresholds from the most brilliant firefly to all others and generate some perturbations or estimating sums or subtractions additions in these same values in a way that the value will change but no too far from the previous one and them reevaluate these new values, thus seeking with every perturbation getting closer and closer to the best possible result. Doing this, the next most brilliant firefly will be the one who have the best arrangement of values which satisfy the search of the best "cut".

The number of fireflies and the number of how many interactions the algorithm will do is defined at the beginning, how we can see in this, the amount of fireflies to perform this evaluation by definition will be 50 individuals (not necessarily the best amount) and the max generation number can be 100 interactions, enough for the algorithm to process and get the expected outcome, resulting in the very approached value that is considered acceptable, costing less time in relation to the brute-force algorithm.

Knowing the basic concept of the Firefly algorithm and thinking about how one can optimize the operation of this algorithm, we will now use concepts of parallelization from some FA functions to split the processing flow in a dedicated machine cluster (Enceladus from FEI University Center), studying in which cases the parallelization can become advantageous with the purpose of testing or even obtaining the best fireflies quantifying parameters in order to maximize the result accuracy or decrease the time cost.

Therefore, we defined some highlighted lines to show how our algorithm have been adapted for the purpose and modified to run with parallel functions on Enceladus cluster. A dataset for run and use a valid input of images to feed our test will be quoted below:

III. BERKLEY DATABASE

This paper will be using 300 images from the Berkeley University. Those images are a diversity of natural scenes, in

Algorithm 1 Generic Firefly Algorithm for MLOTP (from [7])

Input: n: number of fireflies in each generation; d: dimension; γ: absorption coefficient; $alpha$: step of motion; β: attactivity factor; Γ: number of max generation;
Output: an acceptable thresholding set $f_i^* = \{x_1^i, x_2^i, \ldots, x_d^i\}$
Set the initial values: $t = 0$, $\alpha_0 = 1.0$
Generate a randomly initial population $\{f_1, f_2, \ldots, f_n\}$ where $f_i = \{x_1^i, x_2^i, \ldots, x_d^i\}$ is the i-th d-dimensional firefly solution
For each firefly f_i compute the $Z(f_i)$ evolution function as its brightness
while $t \leq \Gamma$ **do**
    for $i = 1 \to n$ **do**
        for $j = 1 \to n$ **do**
            Compute the distance $r_{i,j}$ between bright $Z(f_i)$ and $Z(f_j)$
            for $i = 1 \to n$ **do**
                Bright evaluation $Z(f_i)$
                for $j = 1 \to n$ **do**
                    if $Z(f_i) < Z(f_j)$ **then**
                        Move firefly f_i towards firefly f_j, according to the following update rule:
                        Generate new random solution $\mu_t = \{x_1, x_2, \ldots, x_d\}$
                        $\alpha_t = \alpha\alpha_t$
                        $\beta_0 = \beta exp(-\gamma r_{i,j}^2)$
                        for $k = 1 \to d$ **do**
                            $x_k^i = (1 - \beta_0)x_k^i + \beta_0 x_k^j + \alpha_t \mu_t$
                        end for
                        end if
                    For each firefly f_i, compute the new evaluation function $Z(f_i)$ as its brightness
                    Rank all fireflies according to their individual brightness $Z(.)$
                    Set the brighter firefly $f_i^* as the current result$
                    $t = t + 1$
                    return f_i^* =0

which each of them were manually segmented. The results of the segmentations are available in the same database [8].

The image segmentation in cognitive regions task still is an open discussion. It is possible to list two big reasons to not consider it an easy task:

- A good segmentation depends of its context, as well as a point of view; and
- It is rare in Computer Science and related areas to find a database for comparison of formal results. Usually, researchers show their results with some images and point what they believe is correct. It is never clear that the same technique will be functional with other images of the same class.

Fig. 2. Some Berkeley figures and human segmentation

However, the question that persists is: ”What is a right segmentation?”. On the absence of a precise answer for this question, it is necessary at least a ”guiding light” to show the way, and that the comparison of various techniques under the same database can be possible. In this approach, Berkeley's University database can be considered an attempt to find this ”guiding light”.

One of the steps of this project is to establish a ”golden-pattern” for the segmentation of the images from Berkeley's database. Therefore, it is needed that an algorithm capable of finding optimal values of thresholding. Therefore, the proposed cluster (Enceladus) will be also used as a tool to establish the thresholding values that can get closer to the human segmentation through a brute force sweep. These values will compose the ”golden-pattern” that will be compared with the thresholds found by the parallel segmentation Firefly algorithm.

It is substantial to highlight that the use of the cluster is fun-

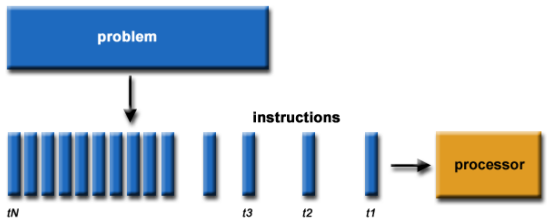

Fig. 3. Serial representation

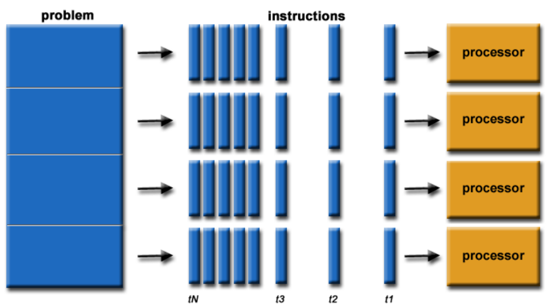

Fig. 4. Parallel representation

damental for this search process, once that the computational order to find thresholds is $O(256^L)$ for 256 distinct values in shades of gray.

IV. Distributed Computing

A. Parallel Computing

Traditionally, software is written in a serial computation format, in which a problem is broken into a discrete set of instructions that are executed one after another in a single processor where only one instruction is executed at any moment in time [9].

Parallel computing in a simple sense is solving a computational problem with the use of simultaneous compute resources. In that way a problem is divided into smaller parts that can be solved concurrently, after that each part is further broken down into a series of instructions and them a control/coordination mechanism is employed to administrate [10].

A computational problem to be worth parallelizing should be capable of being divided into discrete pieces that are able to be solved simultaneously, execute various instructions at any moment in time and the problem must be solved in less time than it would with a single processor.

B. Clustering

Typically, parallel computers are clusters where each compute node is a multi-processor parallel computer in itself and those multiple computer nodes are networked together in a local or long-distance network. Furthermore, a cluster is a group of computers that are connect together which are used as servers or for high processing power. In this paper's approach the cluster Enceladus is being used for a high data processing, the cluster Enceladus is a cluster of computers composed by

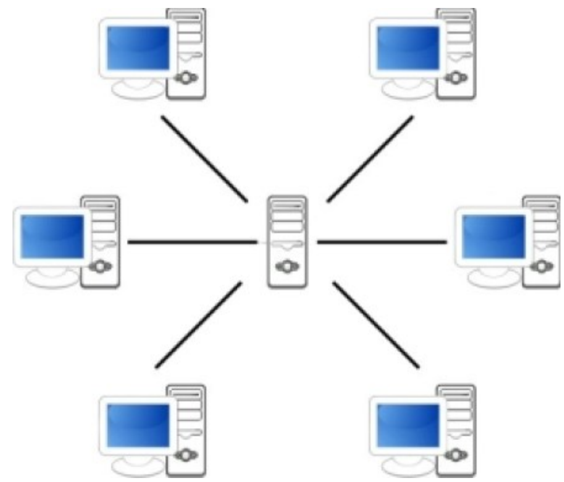

Fig. 5. Grid computing

using a C++ library called **Mpich** that makes the connection between machines while the code is running.

C. Grid Computing

Inside the concept of distributed computing, when talking about computer clusters, it becomes a new computational concept called grid computing which just like parallelization, it is capable of reaching a high processing power by dividing tasks, in the grid's case, between multiple computers, it can be from a local network or long distance, forming a virtual machine.

D. Methodology

This paper's purpose is to create a distributed Firefly Algorithm (FA) and identify if it is superior when compared to the linear algorithm. The number of fireflies will be distributed in N-subordinates whilst the master will be in charge of the information exchange during the algorithm. It will be divided in two parts, the evaluation (Line 5. from Generic Firefly Algorithm for MLOTP), where every subordinate will evaluate the fireflies in their reach. Due to the high cost of this part it is believed that the algorithm can become faster by distributing. The second part that will be changed is the movement of the fireflies, each subordinate will be in charge of moving the firefly that is in its care. To measure both there will be used the Berkeley's Database (Section III – Berkeley's Database) as our image database. Will them, be evaluated two aspects of the new algorithm, its speed and quality of image segmentation (precision). To measure the speed two evaluation parameters will be used: the number of thresholds and the number of fireflies. There will be tested 10, 20, 40, 60, 80 and 100 fireflies and 2 to 10 thresholds in all 300 images from Berkeley's Database. To conclude if the algorithm is as precise as its linear version it will be measured the Euclidian distance of thresholds using the first six images from the database between the serial algorithm and itself to find out the variance between thresholds running multiples times, and after that compare to

		Euclidian Distance					
		Im. 1	Im. 2	Im. 3	Im. 4	Im. 5	Im. 6
Number	2	5.38	5.83	1.41	13.60	8.94	1.00
of	3	33.50	26.32	45.93	22.23	26.24	17.38
Thresholds	4	26.02	12.80	45.93	41.90	15.84	38.34
	5	22.89	60.17	33.67	56.44	41.26	17.64

TABLE I

RESULT OF THE EUCLIDIAN DISTANCE BETWEEN THE SERIAL ALGORITHM VERSUS THE DISTRIBUTED VERSION APPLIED TO THE SIX FIRST IMAGES OF BERKELEY'S DATABASE.

		Euclidian Distance					
		Im. 1	Im. 2	Im. 3	Im. 4	Im. 5	Im. 6
Number	2	3.6	7.07	3.6	4.12	2.23	5.1
of	3	10.48	9.22	9.11	14.63	33.03	10.20
Thresholds	4	21.44	41.59	47.05	17.05	19.21	22.49
	5	82	142.46	63.8	47.95	45.4	68.78

TABLE II

RESULT OF THE EUCLIDIAN DISTANCE BETWEEN TWO SERIAL ALGORITHM APPLIED TO THE SIX FIRST IMAGES OF BERKELEY'S DATABASE.

finally conclude if it maintained the same quality of images, or if it got worse due to distributing the algorithm.

V. CONCLUSION

A. Results

As one can see in Figs. 6 and 7, the distributed algorithm was much more efficient than the serial algorithm, in a matter of time, we can say that it hears a significant improvement of the distribution process. We can also observe that the number of thresholds do not change computing time significantly. On the other hand, the number of fireflies has a highly correlation with time in both serial and parallel version of algorithm.

Regarding the precision, one can see that the comparison between the Euclidean distance of the two serial runs and between the distributed and serial runs are very similar. Considering all data from Tables I and II, we compute the t-student test in order to measure if the distribution are comparable. Results show that the $p-$value was 0.5458, which means that both distributions are comparable.

All in all, this work proposed a parallel version of the Firefly algorithm to MLTP problem with natural images. Results shows no significant difference between solutions from serial a parallel version. Moreover, the time could be reduced significantly using a set of computing nodes. As future work, we plan to use the same technique with other bio-inspired algorithms to and compare them all in a benchmark experiment.

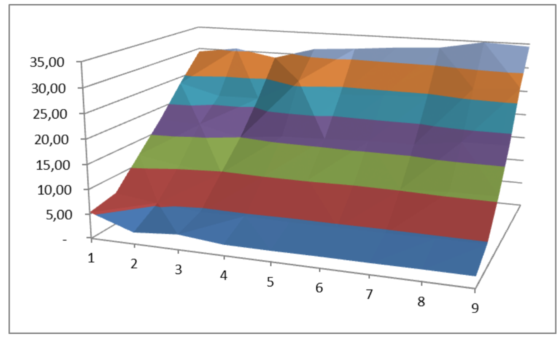

Fig. 6. Serial Algorithm. x-axis is the number of thresholds used; y-axis is the number of fireflies; z-axis represents the computing time

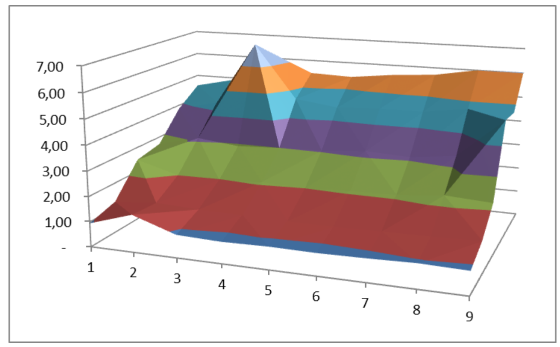

Fig. 7. Parallel Algorithm. x-axis is the number of thresholds used; y-axis is the number of fireflies; z-axis represents the computing time

REFERENCES

[1] C. Blum and D. Merkle, *Swarm Intelligence: Introduction and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[2] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[3] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2008.

[4] E. Chong and S. Zak, *An Introduction to Optimization*, ser. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2013.

[5] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications*, O. Watanabe and T. Zeugmann, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 169–178.

[6] I. Fister, I. Fister jr, X.-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms," vol. 13, p. 34–46, 12 2013.

[7] P. S. Rodrigues, G. A. Wachs-Lopes, H. R. Erdmann, M. P. Ribeiro, and G. A. Giraldi, "Improving a firefly meta-heuristic for multilevel image segmentation using tsallis entropy," *Pattern Anal. Appl.*, vol. 20, no. 1, pp. 1–20, Feb. 2017. [Online]. Available: https://doi.org/10.1007/s10044-015-0450-x

[8] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.

[9] B. Barney, "Introduction to parallel computing," 2018, (Accessed on 07/30/2018). [Online]. Available: https://computing.llnl.gov/tutorials/parallel_comp/

[10] J. JáJá, *An Introduction to Parallel Algorithms*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1992.