Observations on Convergence in Leaders and Followers

Antonio Bolufé-Röhler
Mathematical and Computational Sciences
University of Prince Edward Island
Charlottetown, Canada
ORCID 0000-0002-3181-1864

Stephen Chen
Information Technology
York University
Toronto, Canada
ORCID 0000-0001-8657-6200

***Abstract*—Failed Exploration often involves the comparison of an exploratory search solution that has poor relative fitness with a stored reference solution that has superlative relative fitness (e.g. a local optimum). The selection mechanism of Leaders and Followers is designed to reduce the effects of Failed Exploration by attempting to compare search solutions and reference solutions only when they are likely to have similar relative fitness. Search solutions exist in a population of followers and reference solutions exist in a population of leaders, and members from different populations are only compared when the two populations have a similar median fitness. An unexpected state for this condition to occur is for all of the leaders and all of the followers to reach the same local optimum. Future modifications to Leaders and Followers will need to address this mode of failure/convergence.**

***Index Terms*—selection, leaders and followers, exploration, exploitation**

I. Introduction

Failed Exploration occurs when an exploratory search solution from a fitter/more promising region of the search space is rejected due to its comparison with a fitter reference solution from a less fit region of the search space [1]. If an exploratory search solution and a stored reference solution have the same relative fitness with respect to their nearby local optima (e.g. they are both local optima), then the risk of Failed Exploration can be reduced. The design goal for the metaheuristic of Leaders and Followers (LaF) is to use two separate populations – one for reference solutions (leaders) and one for search solutions (followers) – that would ideally be compared only when similar levels of relative fitness might be present [2].

Leaders and Followers is a promising new metaheuristic because its original format performs better than established techniques such as Particle Swarm Optimization (PSO) [3], [4] and Differential Evolution (DE) [5]. However, despite the development of LaF being based on extensive analysis of search space properties, the actual performance of LaF was only analyzed via performance on the CEC2013 benchmark function set [6]. The continuous improvement of its best overall solution [2] suggested that its design was successful in making it less prone to premature convergence than PSO and DE. The new experiments and analysis in this paper will show that LaF can also be prone to premature convergence.

The design factors of LaF and a brief description of the versions of DE and PSO used in this paper are presented in Section II. New experiments which observe these convergent behaviours are then presented in Section III. Insights from these observations are discussed in Section IV before potential remedies are presented in Section V.

II. Background

A. Leaders and Followers

The signature feature of Leaders and Followers is the use of two populations: the leaders and the followers. The population of leaders represent the references solutions which guide the search process, and the population of followers is where new search is performed. However, followers need an opportunity to become leaders in order for the search process to advance. A key design decision in LaF is thus to decide when and how to compare followers with leaders.

The explicit goal in LaF is to minimize the risk of Failed Exploration during the comparison of followers with leaders. Specifically, the goal is to make the comparisons when the "relative fitness" (i.e. the difference in fitness between a solution and the local optimum it could reach by following a monotonic path) of the followers is similar to that of the leaders. The first attempt to identify this comparison point is to use the median fitness of each population. Original LaF then uses two-tournament selection in a merged population of leaders and followers to select the new leaders.

Another important design decision in LaF is to determine how the leaders will influence the followers. In original LaF, a follower moves towards a randomly chosen leader. This convergent motion makes the assumption that highly fit regions of the search space are in the same general area – i.e. that the search space is structured or globally convex. Through these design decisions, the intention was for the population of leaders and the population of followers to both represent a large number of attraction basins, and that fair comparisons could be made between the new basins represented by the followers with the existing basins represented by the leaders.

The procedure to create a new trial solution uses Equation (1) for every dimension d of the new solution. In this formula, ϵ_d is a uniform random number in $(0, 1)$ sampled independently for every dimension. Results for LaF correspond to the original implementation described in [2], with a population size of $p = 50$.

$$trial_d = follower_d + 2 \times \epsilon_d \times (leader_i - follower_i) \quad (1)$$

B. Differential Evolution

The experiments in this paper are based on a DE/rand/1/bin version of DE with typical parameters of population size *p* = 50, crossover *Cr* = 0.9, and scale factor *F* = 0.8 [5], [7]. In each iteration, each population member x_i is considered as a *target* for replacement by a candidate solution that is constructed in two steps: creation of an intermediate solution and crossover with x_i. During the creation of an intermediate solution y_i from three distinct random solutions r_1, r_2, and r_3 in Equation 2, the scale factor *F* affects the "step size" from r_1 taken in the direction of the "difference vector" created with r_2 and r_3.

$$y_i = r_1 + F(r_2 - r_3) \quad (2)$$

In Equation 3, this intermediate solution is then crossed term-by-term in each dimension *d* of the search space with the *target* solution x_i to produce the *new* candidate solution x'_i.

$$x'_{i,d} = \begin{cases} y_{i,d} & u_d \leq Cr \\ x_{i,d} & u_d > Cr \end{cases} \quad (3)$$

C. Particle Swarm Optimization

The experiments in this paper use a version of standard particle swarm optimization [4] with a ring topology. The key parameters specified from this standardization are χ = 0.72984, and c_1 = c_2 = 2.05 for the velocity updates given in Equation 4. Additional implementation details are the use of *p* = 50 particles [4], zero initial velocities [8], and "Reflect-Z" for particles that exceed the boundaries of the search space (i.e. reflecting the position back into the search space and setting the velocity to zero) [9]. The source code for this implementation is available online [10].

$$\begin{aligned} v_{i+1,d} = \chi\big\{ & v_{i,d} + c_1\epsilon_1(pbest_{i,d} - x_{i,d}) \\ & + c_2\epsilon_2(lbest_{i,d} - x_{i,d})\big\} \end{aligned} \quad (4)$$

III. New Results

The experiments presented in this section are performed with the Rastrigin function (5) on $d = 20$ dimensions and a maximum budget of $FEs = 10.000 \times d = 200.000$ function evaluations (FEs). Rastrigin has the useful property that every solution with integer values for each dimension is a local optimum, and all other solutions are in the attraction basin of the optimum found by rounding each of that solution's coordinate values to their nearest integer value. This property makes it possible to determine the attraction basin in which solutions are located and how close they are to the local optimum of their attraction basin (i.e. their relative fitness).

$$f(x) = 10d + \sum_{i=1}^{d} \left(x_i^2 - 10\cos(2\pi x_i)\right) \quad (5)$$

Figure 1 shows the average performance of PSO, DE and LaF for 30 independent runs. It can be noticed that PSO converges very quickly, while DE and LaF have slower convergence. However, LaF performs better than DE and also shows a continuous improvement until the very end of the optimization process. Both DE and LaF show consistent improvements until around 50% of available FEs are used, but LaF keeps improving very slightly after this point while the improvements in DE stop almost completely. A similar plateau in performance is observed in PSO, but the end of improvements occurs after much fewer FEs.

A plot of performance can be a useful visualization tool for analyzing the convergence of a metaheuristic. However, a overall performance doesn't differentiate between an improvement that is due to the discovery of a new attraction basin (i.e. exploration) and an improvement that is due to movement within an attraction basin towards its local optimum (i.e. exploitation). The effects of exploitation can be observed by plotting the "relative fitness" (i.e. the difference between the fitness of a solution and the fitness of the local optimum of its current attraction basin).

It is noted that Figure 2 is based on the relative fitness of the best solution in the population, and not all of the solutions in the population. It is further noted that the plateau in performance in Figure 1 is coincident to the best solution reaching a local optima in Figure 2. One of the goals of a population-based metaheuristic is to use the diversity of the population to avoid (premature) convergence. However, this is not observed in PSO or DE – once the best population member reaches a local optimum, it is exceptionally rare for another population member to find a better (locally optimal) solution. The potential for two followers to enter a selection tournament in which one of them will be selected to be a future leader allows for occasional bursts of divergence among the leaders that can lead to the detection of better attraction basins.

To better understand this dynamic, Figure 3 shows the number of different attraction basins represented by the population of leaders. It is typical to present the results of experiments as the average of a statistically relevant set of runs. However, many insights into the operation of metaheuristics can be lost through the process of averaging their behaviour. Thus, the results presented in Figure 3 correspond to a single run of the algorithm LaF; this run does not correspond to a best or worse case scenario, but it instead represents a typical execution of LaF.

Figure 3 shows how all the leaders are located in different attraction basins for the initial 800 iterations of LaF, out of 4.000 total iterations. At this point, leaders start to converge towards the same attraction basin(s), and less than 500 iterations later all the leaders have converged into the same attraction basin. This can be considered as a premature convergence, as it happens before 40% of the total budget of FEs has been consumed. Figure 3 also shows with red dots the iterations where none of the followers are in different attraction basin(s) than the leaders. It can be noticed that this happens for the first time shortly after all the leaders have converged to the

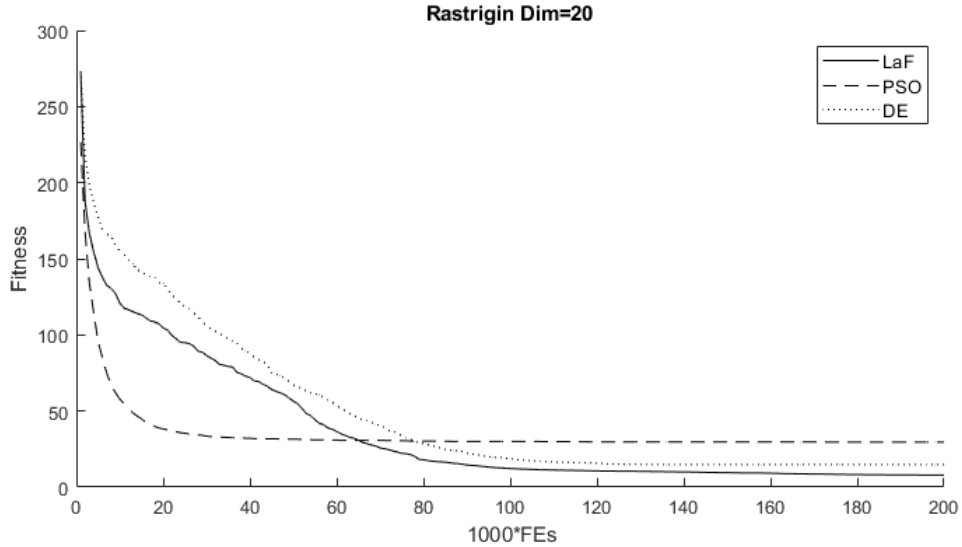

Fig. 1. Performance for LaF, PSO and DE (average of 30 independent runs). Both PSO (quite quickly) and DE (more slowly) converge, while LaF has a small amount of continuing improvement.

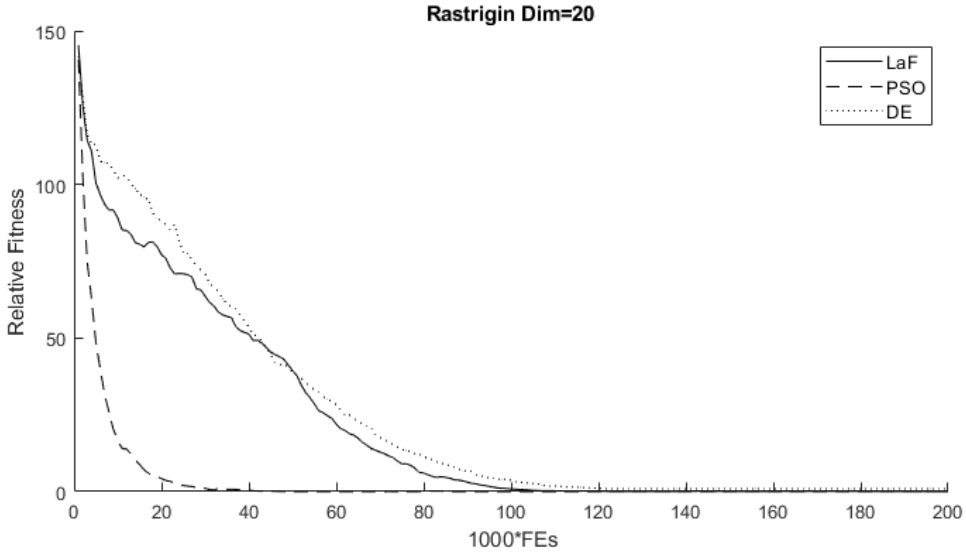

Fig. 2. Relative fitness of best solution for LaF, PSO and DE (average of 30 independent runs). The iteration at which the best solution reaches a local optimum is coincident with the plateau of performance shown in Figure 1.

same attraction basin.

The run presented in Figure 3 shows another distinctive behaviour of LaF – it's ability to occasionally diverge from a converged state. It can be noticed that after reaching 2.000 iterations, the number of different attraction basins represented by the leaders briefly peaks back up to 6 different basins. This is an unusual behavior for metaheuristics after convergence and it happens because of LaF's selection scheme which allows new attraction basins to be explored and kept in the population of followers. This will occasionally lead to the introduction of new attraction basins into the population of leaders after a restart. This is also the reason why the average performance in Figure 1 shows a very small but steady increase in fitness after half of the FEs budget has been reached.

Leaders and Followers uses a selection scheme that was designed to limit the negative effect that selection has on exploration. The introduction of a two level selection process aims to reduce the biased comparison that can happen in other metaheuristics – when solutions with exceptional relative fitness are compared against new solutions with poor relative fitness. The design partially achieves this goal as evidenced by its better overall performance and the ability to make progress from a "converged" state. However, the early arrival of the converged state shows the power of selection to overwhelm other aspects of the design.

In order to shed some light about when this converged state is reached, Figure 4 shows a histogram for 100 independent runs. The histogram indicates the moments, measured as a

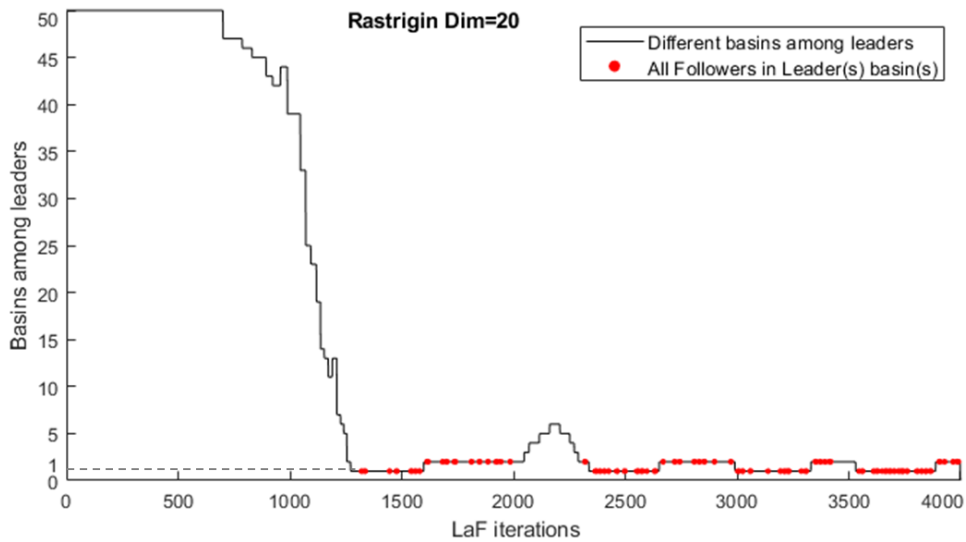

Fig. 3. Number of different basins among the leaders in a single run of LaF with a population size of 50. The red dots indicate the iterations at which all the followers are in the same attraction basin(s) as the leaders.

percent of the total budget of FEs, when all leaders converge into the same attraction basin for the first time. It can be noticed that most runs converge in an interval around 30% to 40% of the total amount function evaluations. (The selected run for Figure 3 also converges during this interval.) It is interesting to notice that there can be a large variance in when this moment is reached, ranging from as early as 20% and up to 60% of the allocated FEs. However, the converged state is reached in every execution of LaF and is always reached before the 60% of FEs has been consumed.

IV. Discussion

Metaheuristics that use fitness-based selection to compare against a reference solution which has the opportunity to experience local optimization towards a local optimum are prone to the failure mechanism of Failed Exploration. In original LaF, the movement of followers towards leaders encourages the creation of new search solutions near the existing leaders – i.e. that exploitation is performed in the attraction basins already represented by the leaders. As the leaders improve towards their local optima, the similar median fitness criterion means that followers must also approach local optima, and the local optimum they are most likely to approach is that of the leader. Thus, all the leaders and all the followers tend to converge to the same (local) optimum.

Less convergent search trajectories for the followers could reduce their ability to perform exploitation. However, if a follower solution is ever a (local) optimum (e.g. even random search can occasionally find optima), then fitness-based selection would likely cause it to become a leader which could then cause future Failed Exploration. The possibility of Failed Exploration will exist for any selection process based solely on the fitness of two solutions.

In general, the combination of fitness-based selection and elistism encourages reference solutions to approach and become local optima. The operation of metaheuristics when they have local optima reference solutions can be very different from their initial behaviour. In particular, the performance curves of PSO, DE, and LaF all stall at around the same moment that their best overall solution first reaches a local optimum. The design of metaheuristics needs to be cognizant of this moment, and potential courses of action include delaying the arrival at local optima or introducing alternate search strategies after a local optimum is reached.

V. Future Work

The presented failure mechanism is related to fitness-based selection, so a natural remedy is an alternate basis for selection. For example, a diversity measure could be introduced into the selection of new leaders from the merged population of leaders and followers. This diversity measure would ideally ensure that all of the new leaders are from a minimum number of different basins. A simple distance measurement between the new candidate leader and all of the previously selected leaders could be added to the existing fitness-based selection mechanism.

An alternate approach is to ensure that the followers cannot converge to the leaders. In original LaF, followers move on a line segment towards a leader, and this line segment can be up to the entire distance which would cause the new follower position to be identical to that of the leader. A simple change could be to implement a maximum step a follower can take (which would thus create a minimum distance between a follower and its current leader). More complex search trajectories could also be considered, especially for non-globally convex search spaces in which movements towards the current leaders are less likely to lead to better solutions.

LaF selection schemes favours the comparison of solutions with similar relative fitness, thus reducing Failed Exploration.

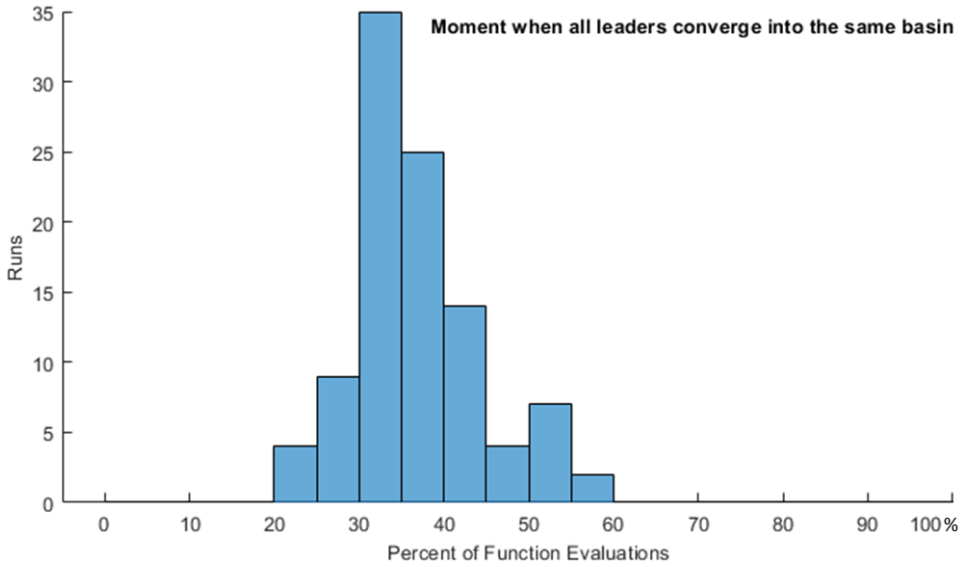

Fig. 4. Histogram showing the moment when all the leaders are in the same attraction basin for the first time.

Subsequently this also delays the moment at which a converged state is reached. However, the power of selection still makes leaders reach solutions with very high relative fitness (i.e. local optima) early in the search, causing a premature convergence of the algorithm. An alternate design that could delay the convergence state being reached in the leaders is to add a third population in between the followers and the leaders. This "middle" population would be updated by merging with the followers once the followers reach a similar mean fitness value. Similarly, leaders will merge with this middle population using the same criterion. An alternate approach to delaying the converged state would be to have a mechanism for detecting when this state is reached and "shaking" the leaders out of their local optima while keeping them in a promising region of the search space.

References

[1] S. Chen, A. Bolufé-Röhler, J. Montgomery, and T. Hendtlass, "An analysis on the effect of selection on exploration in particle swarm optimization and differential evolution," in *Evolutionary Computation (CEC), 2019 IEEE Congress on*. IEEE, 2019.

[2] Y. Gonzalez-Fernandez and S. Chen, "Leaders and followers — a new metaheuristic to avoid the bias of accumulated information," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 776–783.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4. IEEE, 1995, pp. 1942–1948.

[4] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*. IEEE, 2007, pp. 120–127.

[5] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[6] J. Liang, B. Qu, P. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*, vol. 201212, pp. 3–18, 2013.

[7] J. Montgomery and S. Chen, "An analysis of the operation of differential evolution at high and low crossover rates," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, 2010, pp. 1–8.

[8] A. Engelbrecht, "Particle swarm optimization: Velocity initialization," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pp. 1–8.

[9] S. Helwig, J. Branke, and S. Mostaghim, "Experimental analysis of bound handling techniques in particle swarm optimization," *IEEE Transactions on Evolutionary computation*, vol. 17, no. 2, pp. 259–271, 2013.

[10] "Pso source code," June 2017. [Online]. Available: https://www.researchgate.net/publication/259643342_Source_code_for_an_implementation_of_Standard_Particle_Swarm_Optimization_–_revised?ev=prf_pub