

Stepping ahead based hybridization of meta-heuristic model for solving Global Optimization Problems

Ravneil Nand * ‡, Kaylash Chaudhary * §, Bibhya Sharma * ¶

* School of Computing, Information and Mathematical Sciences
The University of the South Pacific, Suva, Fiji. <http://scims.fste.usp.ac.fj/>

‡ ravneil.nand@usp.ac.fj, § kaylash.chaudhary@usp.ac.fj, ¶ bibhya.sharma@usp.ac.fj

Abstract—Intelligent optimization algorithms based on swarm principles have been widely researched in recent times. The Firefly Algorithm (FA) is an intelligent swarm algorithm for global optimization problems. In literature, FA has been seen as one of the efficient and robust optimization algorithm. However, the solution search space used in FA is insufficient, and the strategy for generating candidate solutions results in good exploration ability but poor exploitation performance. Although, there are a lot of modifications and hybridizations of FA with other optimizing algorithms, there is still a room for improvement. Therefore, in this paper, we first propose modification of FA by introducing a stepping ahead parameter. Second, we design a hybrid of modified FA with Covariance Matrix Adaptation Evolution Strategy (CMAES) to improve the exploitation while containing good exploration. Traditionally, hybridization meant to combine two algorithms together in terms of structure only, and preference was not taken into account. To solve this issue, preference in terms of user and problem (time complexity) is taken where CMAES is used within FA's loop to avoid extra computation time. This way, the structure of algorithm together with the strength of the individual solution are used. In this paper, FA is modified first and later combined with CMAES to solve selected global optimization benchmark problems. The effectiveness of the new hybridization is shown with the performance analysis.

Index Terms—Firefly Algorithm, covariance matrix adaptation evolution strategy, optimization

I. INTRODUCTION

The Digital era has witnessed an avalanche of research conducted by educationist's dealing with designs and architecture [1], [2], frameworks and policies, pilots and interventions, forecasts and predictions [3], theories and pedagogies [4], [5], tools and technologies, diagnostics, however, an area that provides improvement for all is optimization. Optimization problems involve finding values from set of variables which will give an optimal functional value of the cost or objective function. The literature has witnessed a plethora of optimization techniques to solve problems integrated to human livelihood and endeavors including education, economics, engineering, robotics, and smart cities [6]–[12]. A sub-area of optimization is global optimization which involves finding a global minimum or maximum on the domain of an objective function. There are a lot of optimization algo-

rithms in literature which either minimizes or maximizes the objective function [13], [14]. This research firstly modifies Firefly Algorithm (FA) and later does the hybridization of modified FA with Covariance Matrix Adaptation Evolution Strategy (CMAES) algorithm. The reason for hybridization is that the two algorithms separately sometimes do not perform well in terms of finding the optimality of a particular function and also it may take longer time to find best solutions [15]–[17]. In 20 dimension problem, the median algorithm solves about 30% of the functions while the best algorithm can solve about 75% [18]. In 2008, Yang proposed FA scheme, a swarm intelligence algorithm that is based on the grouping behaviour of fireflies [19]. The algorithm belongs to stochastic algorithm and is classified as a meta-heuristic technique. The technique is widely used in different application fields for their stochastic search property, similar to Particle Swarm Optimization (PSO) [20] method. CMAES was proposed by Hansen et al. in [21] and later enhanced in [22], [23]. The modification of FA will be done in terms of exploration while later hybridization would be done to improve the exploitation. The hybridization of FA with CMAES would allow to gradually change the search exploration into exploitation which is searching for best results within the current search space.

We would start by considering some of the results from literature on modification of FA and their hybridization with other algorithms. FA was modified in the past to solve problems from different industries, sectors and domain. In the computing domain, Sharma et al. in (2020) added adaptive inertia weight for position update of search agents to overcome trap in local optima and slow convergence of original FA [24]. Aggarwal and Kumar in (2019) enhanced the exploitation capability of FA using randomness factor and the modified algorithm was applied to vehicle routing problem [25]. Yelghi et al. in (2018) addressed the problem of the original FA being trapped at the local minimum, hence finding the global minimum [26]. The authors modified FA by applying Tidal Force formula (exploitation) where the modified FA keeps a balance between exploration and exploitation. Teshome et al. in (2017) modified the FA to improve the performance of the maximum power point tracking by reducing the number of computation and

the convergence time to the global maxima [27]. The modified FA was able to suppress voltage fluctuations. He and Huang used the modified FA for multilevel color image thresholding segmentation [28] where three different objective functions were evaluated. Chou and Ngo in (2017) proposed the modified FA for multidimensional structural design optimization problem by incorporating logistic for population, Gauss for attractiveness parameter tuning, adaptive inertia weight for local exploitation and global exploration and Levy flight for exploitation [29]. While Wang et al. in (2016) proposed the modification based on variation trend of the light intensity difference where it decreases premature convergence [30]. Fister et al. use quaternions which extend complex numbers to enhance the performance of the FA in computing domain [31]. Verma et al. in (2016) modified the FA at the initialization of candidate solution step using opposition-based methodology and update of position of each firefly in different dimensions using dimensional-based approach [32] to achieve optimal solution in minimum time. In the electricity industry, Kavousi-Fard et al. in (2014) proposed a modified FA to determine the parameters of the Support Vector Regression algorithm, which in turn will predict the short term electrical load [33]. Every modification improves the original FA either in terms of exploration or exploitation. According to [34], usually this balance is insured by favoring exploration at the beginning of the search and exploitation at the end. Therefore, modification in terms of exploration is done by searching further in the search space through stepping ahead in this research.

So far, there has been no hybridization of FA and CMAES as per our knowledge. One of the reasons for hybridization of different algorithms together is to get to the optimal solutions of any given problem. Apart from FA, other popular optimization algorithms have been hybridized. For instance, Particle Swarm Optimization (PSO) [20] and Genetic Algorithm (GA) [35]. Marinakis and Marinaki in [36] combined genetic algorithm and Particle Swarm Optimization (PSO) to solve vehicle routing problem. while, DziwiÅski et al. presented a hybrid of PSO and evolutionary algorithm for faster convergence presented by PSO and global search by evolutionary algorithm [37]. As for FA, Wang et al. combined the fireworks algorithm neighborhood search operator with FA to solve the problem of getting trapped in a local minimum [38]. CMA-ES has also seen hybridization with various algorithms. Yu et al. combined Brain storm optimization algorithm (BSO) and CMA-ES to achieve optimal solution by utilising the search capability of BSO and search efficiency of CMA-ES [39]. Likewise, other hybridization of optimization algorithms include [40]–[42].

There are a number of meta-heuristic algorithms, some commonly used ones are PSO [20], Genetic Algorithm (GA) [35], FA [19] and Ant Colony Optimization (ACO) [43]. There has been many modifications and hybridizations of these algorithms but there is still a chance of better algorithms to tackle optimization problems. This has given the motivation to look at FA in much detail and modify in terms of the structure and later combine it with another algorithm architecture that would provide a better outcome. The reason to choose FA

is due to its simplicity and success in different application domains as it is widely used to solve the continuous problem same as PSO. The main aim involves the modification of the FA method and its hybridization with CMAES to solve some test and real-world optimization problems. The implementation needs to combine FA with CMAES where the other method is only applied within the execution stage of the FA method. Authors in [44], [45] have shown that evolutionary algorithms can benefit a lot from CMAES, which is used a framework to improve the performance. In this research, a hybrid of modified FA and CMAES is discussed with application to real parameter optimization. The algorithm is tested against 25 benchmarks and 2 real world test problems and later compared with literature.

The rest of the paper is organized as follows. Section II highlights on the modification of FA. Section III displays the proposed hybrid algorithm and section IV discusses the experimental setup and results. Section V is on the discussion of the results while section VI concludes the paper with future overviews.

II. MODIFICATION OF FIREFLY ALGORITHM

Firefly algorithm is a bio-inspired meta-heuristic algorithm for optimization problems. It was introduced in 2008 by Yang [19]. The algorithm is inspired by the flashing behavior of fireflies at night. One of the three rules used to construct the algorithm is that all fireflies are unisex, which means any firefly can be attracted to any other brighter one. The second rule is that the brightness of a firefly is determined from the encoded objective function. The last rule is that attractiveness is directly proportional to brightness but decreases with distance, and a firefly will move towards the brighter one, and if there is no brighter one, it will move randomly [19]. The assumption of fireflies to be unisex can be removed in the proposed method as the search is moving further away from the attracted firefly, but it needs to be tested.

The original FA has been modified and is stated in Algorithm 1.

Steps 1 to 3 are original steps of the FA. Step 1, the population of fireflies is initialized, while in steps 2 and 3, the variables are defined and the iteration starts. Step 4 was introduced to allow the algorithm to avoid local optimum. In step 4, new solution is found in relation to the firefly j where the movement of the firefly is further away than usual movement of firefly i closer to best solution as in standard FA. The term which we have given to this approach is *Stepping Ahead*. It is further explained in the following subsection.

A. Stepping Ahead

The notion behind stepping ahead is to allow the firefly to journey ahead to look for better solutions. This allows for a better exploration of the wider search space. The distance is calculated beside on current position of firefly (FF_i) and firefly (FF_j). The firefly (FF_i) is moved in front of firefly (FF_j) to search for a better position, and if a better position is found, the light intensity of firefly (FF_i) increases to attract

Algorithm 1: modified Firefly Algorithm

```
Step 1: Initialize population of fireflies
Random partition  $x_i (i = 1, 2, \dots, n)$ 
Step 2: Formulate light intensity  $I$  so that it is associated with  $f(x)$ 
Step 3: Define absorption coefficient  $\gamma$ 
while  $t < MaxGeneration$  do
  foreach  $i = 1: n$  (all  $n$  fireflies) do
    foreach  $j = 1: i$  (n fireflies) do
      if  $I_j \leq I_i$  then
        Vary attractiveness with distance  $r$  via  $\exp(-\gamma r)$ ;
        Step 4: Stepping ahead
        move firefly  $i$  further than  $j$  that is  $j$  distance plus
        distance difference  $j$  and  $i$ ;
        Evaluate new solutions and update light intensity;
        if new solution not better then
          move firefly  $i$  in relation to  $j$  that is  $i$ 
          distance plus distance difference  $j$  and  $i$ ;
          Evaluate new solutions and update light
          intensity;
        end
      else
        nothing
      end
    end
  end
end
Rank fireflies and find the current best;
end
Post-processing the results and visualization;
```

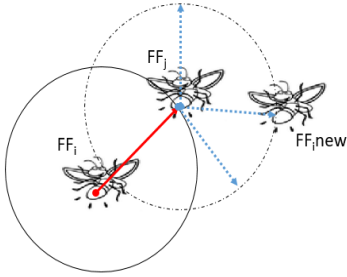


Fig. 1. Firefly stepping ahead

more fireflies to that position. If firefly (FF_i) fails to search for a better solution another chance is given where search is closer to firefly (FF_j) is in standard FA algorithm. This gives a chance of any algorithm to move out from local optimum as quiet often it was seen that local optimum hinders best algorithms to find global optimum. The stepping ahead feature for the firefly can be seen in Figure 1. The figure shows how a firefly travels from the current best position to new best position in terms of the best firefly position. It is normal for an algorithm to attract to the best solution, for exploration there need to be a smart technique that can produce even better results in terms of optimal solutions when the problem in hand is difficult to solve.

III. ALGORITHM OVERVIEW: HYBRIDIZATION OF MODIFIED FIREFLY ALGORITHM WITH CMAES

In this section, modified FA is combined with CMAES. To avoid increasing the time complexity of the algorithm, the implementation of the CMAES is done within the second FOR loop of the firefly algorithm. The original algorithm

is incorporated with CMAES to allow for better search and exploitation.

The proposed algorithm is shown in Algorithm 2.

Algorithm 2: Proposed hybrid Firefly algorithm

```
Step 1: Initialize population of fireflies and initialize state variables
Random partition  $x_i (i = 1, 2, \dots, n)$ 
Step 2: Formulate light intensity  $I$  so that it is associated with  $f(x)$ 
Step 3: Define absorption coefficient  $\gamma$  and update  $M$ 
while  $t < MaxGeneration$  do
  foreach  $i = 1: n$  (all  $n$  fireflies) do
    foreach  $j = 1: i$  (n fireflies) do
      if  $I_j \leq I_i$  then
        Vary attractiveness with distance  $r$  via  $\exp(-\gamma r)$ ;
        Step 4: Stepping ahead
        move firefly  $i$  further than  $j$  that is  $j$  distance plus
        distance difference  $j$  and  $i$ ;
        Evaluate new solutions and update light intensity;
        if new solution not better then
          move firefly  $i$  in relation to  $j$  that is  $i$ 
          distance plus distance difference  $j$  and  $i$ ;
          Evaluate new solutions and update light
          intensity;
        end
        Step 5: CMAES
        sample new solutions in terms of CMAES and
        evaluate them;
        sample multivariate normal and fitness;
      else
        nothing
      end
    end
  end
end
Rank fireflies and find the current best; (sort solutions)
move mean to better solutions;
update isotropic evolution path;
update anisotropic evolution path;
update covariance matrix;
update step-size using isotropic path length;
end
Post-processing the results and visualization;
```

Steps 1 to 4 are same as in the previous section. The stepping ahead modification is the Step 4. The new introduction is Step 5. In Step 5, the algorithm CMAES is integrated with FA. Here the CMAES algorithm is implemented in the same cycle to avoid extra time complexity. Later, the entire population is ranked and the current best is found. The variables of CMAES are updated. This is done until the maximum iteration is reached. Finally, post-processing of the results is carried out and visualization are stored.

IV. EXPERIMENTS AND RESULTS

This section shows the experimental setup and results analysis of the proposed hybrid algorithm. The experimental setup highlights the parameters used in each of the algorithm and the objective function. Results section shows the mean, median and best and worst results from the experiment. The hybrid model is referred to as FACmaes in this section.

A. Experimental Setup

To keep a controlled environment, it is assumed that the functions are a blackbox. The bounds are not modified and kept at -10 and 10 for all the functions as in the literature. There are only a few parameters that need to be set up for

TABLE I
PARAMETER SETTING

Parameter	FACmaes
Initial Population size	25
Maximum iterations	1000
No. of Runs	30
Lambda	25
Damping Ratio	0.98
Light Absorption Coefficient	1
Attraction Coefficient Base Value	2
Mutation Coefficient	0.2

TABLE II
BENCHMARK FUNCTIONS

No.	Name	No.	Name
F01	ackleyfcn	F14	schwefel222fcn
F02	alpinen1fcn	F15	schwefel223fcn
F03	alpinen2fcn	F16	shubert3fcn
F04	exponentialfcn	F17	shubert4fcn
F05	griewankfcn	F18	shubertfcn
F06	periodicfcn	F19	styblinski tankfcn
F07	powellsumfcn	F20	sumsquaresfcn
F08	quarticfcn	F21	xinsheyangn1fcn
F09	rastriginfcn	F22	xinsheyangn2fcn
F10	rosenbrockfcn	F23	xinsheyangn3fcn
F11	salomonfcn	F24	xinsheyangn4fcn
F12	schwefel220fcn	F25	zakharovfcn
F13	schwefel221fcn		

FACmaes and these have been described in Table I. The experiments are conducted on the 25 functions taken from [46] including two real world functions taken from CEC 2011 [47]. The subsection gives details about the functions.

Benchmark and Real world Functions

Table II shows 25 benchmark functions used while Table III shows 2 real world problems that have been taken out from CEC 2011 to show the performance of the proposed algorithm on real world problems.

The 25 benchmark functions (Table II) are same as the ones used in [48]. The classification done in the research was based on four group types, which gave the motivation to use these 25 functions. The groups were based on non-convex, convex, continuous, differentiable, non-differentiable, separable, non-separable, multimodal and uni-modal attributes. A complete formulation and optimal values of functions can be accessed from [46]. The details about real world functions are shown in Table III.

B. Results

This subsection reports on analysis of the algorithms. Tables IV and V show the results obtained on benchmark functions

TABLE III
CEC 2011 REAL WORLD OPTIMIZATION PROBLEM

No.	Name
CEC4	Optimal control of a non-linear stirred tank reactor
CEC7	Spread Spectrum Radar Polly phase Code Design

while Table VI shows results obtained on CEC functions to validate the effectiveness of the algorithm.

In Table IV, the performance of standalone methods together with the proposed method is shown on 25 benchmark problems. The best, median, and the worst of the 30 runs are shown in this table. The best results were seen for FACmaes model where from 25 functions, it was able to give the best results in 19 functions. That is, the minimum of the functions. The two standalone methods, FA and CMAES, were able to give the best results in 8 and 7 functions, respectively. Similar results can be seen for median and worst values obtained for each of the algorithms.

Looking at Table IV, a lot of information can be gathered regarding the three algorithms. Any of the algorithms did not reach the optimal value of F1, however, CMAES and FACmaes were closer to the optimal compared to FA method. For F2, the optimal value was reached by FACmaes method only. As for F3 function, none of the methods are closer to the optimal result, the closer result was found by FA method. All the methods were able to find the optimal value for the F4 function. F5 optimal value was possible by CMAES method in best, median and worst results.

Moving on, FA and FACmaes had close results to F6 function's optimal when compared to CMAES method. For function F7, FACmaes was the only algorithm to obtain the optimal result of 0. For F8, the proposed method got the closed result whereby FA got the second best. For function F9, all the three methods were able to find the optimal value of 0. FACmaes got the closest result in function F10. For function F11, all the three methods got the same result of 0.099. For functions F12 to F14, again it was FACmaes method that was able to get the best results compared to FA and CMAES methods. This clearly shows the strength of combining two methods together. The stepping ahead strategy has worked well as it provides a better search path.

For function 15, again FACmaes was able to find the optimal value together with CMAES method. In this case, FA was not able to find the optimal. As for the functions F16 to F19, none of the methods are able to find the optimal values. In F16, FA had a close result, in F17, FACmaes method had close result as for F18 and F19, again FA method had better results. From function F20 to F25, FACmaes and CMAES were two methods that performed well where FACmaes was able to find optimal values for function F20, F21, F23 and F25. Similar results can be seen on the same Table IV for median and worst results, where in 17 out of 25 functions and 14 out of 25 functions, the proposed method FACmaes has outperformed in median and worst results, respectively.

Table V shows the mean results obtained by the algorithms for the benchmark functions. In most cases the proposed method has better solution than the other two methods. An interesting note is that, in just 1 out of 25 functions, all the three methods were able to produce same results.

In Table VI, the results are compared with selective algorithms from literature based on best results obtained for FACmaes. It can be seen that, the proposed algorithm was

TABLE IV
BEST, MEDIAN, WORST RESULTS ON BENCHMARK FUNCTIONS

Fun.	Optimal	Best			Median			Worst		
		FA	CMAES	FACmaes	FA	CMAES	FACmaes	FA	CMAES	FACmaes
F01	0	7.56E-11	-8.88E-16	-8.88E-16	1.63E-10	-8.88E-16	-8.88E-16	2.36E-10	-8.88E-16	-8.88E-16
F02	0	9.87E-12	1.19E-21	0.00E+00	1.71E-11	1.23E-19	4.18E-321	2.51E-11	1.89E-19	1.22E-15
F03	-6.1295	-7.85E+01	-1.80E+03	-2.48E+02	-5.35E+01	-1.80E+03	-1.87E+02	-3.23E+01	-5.15E+02	-1.62E+02
F04	-1	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00
F05	0	7.40E-03	0.00E+00	9.86E-03	2.71E-02	0.00E+00	0.00E+00	3.20E-02	5.17E-02	0.00E+00
F06	0.9	1.00E+00	1.33E+00	1.00E+00	1.00E+00	1.90E+00	1.00E+00	1.00E+00	2.28E+00	1.00E+00
F07	0	5.28E-26	2.14E-12	0.00E+00	4.92E-25	1.34E-10	0.00E+00	5.44E-24	4.57E-10	1.34E-294
F08	0	2.51E-05	1.51E-04	2.06E-05	1.07E-04	1.41E-03	1.04E-04	5.72E-04	4.32E-03	2.76E-04
F09	0	0.00E+00	0.00E+00	0.00E+00	9.95E-01	3.84E+00	2.98E+00	3.98E+00	2.91E+01	9.95E+00
F10	0	2.01E-15	9.89E-03	3.18E-27	6.16E-07	2.34E-02	4.18E-10	3.21E-05	4.05E-02	1.15E-05
F11	0	9.99E-02	9.99E-02	9.99E-02	9.99E-02	9.99E-02	9.99E-02	9.99E-02	1.15E-01	9.99E-02
F12	0	5.90E-11	7.99E-47	3.95E-240	1.67E-10	2.66E-46	1.88E-236	2.05E-10	1.06E-45	5.77E-235
F13	0	3.88E-11	3.09E-44	2.15E-176	6.03E-11	1.19E-43	2.86E-173	8.35E-11	2.80E-43	1.22E-169
F14	0	9.25E-11	1.53E-46	5.63E-240	1.63E-10	4.11E-46	4.88E-237	2.52E-10	2.38E-45	1.96E-235
F15	0	2.13E-104	0.00E+00	0.00E+00	2.40E-102	0.00E+00	0.00E+00	7.64E-101	0.00E+00	0.00E+00
F16	-29.6733337	-7.42E+01	-8.29E+01	-8.90E+01	-7.42E+01	-7.34E+01	-8.90E+01	-6.28E+01	-6.51E+01	-6.11E+01
F17	-25.740858	-1.91E+01	-2.26E+01	-2.30E+01	-1.91E+01	-1.95E+01	-2.00E+01	-1.91E+01	-1.72E+01	-1.64E+01
F18	-186.7309	-5.70E+05	-5.29E+09	-8.27E+06	-5.70E+05	-7.57E+08	-6.51E+06	-4.49E+05	-1.99E+08	-1.12E+06
F19	-39.16599	-1.96E+02	-3.92E+02	-2.35E+02	-1.96E+02	-3.92E+02	-2.35E+02	-1.82E+02	-3.78E+02	-2.21E+02
F20	0	4.66E-21	4.20E-97	0.00E+00	2.18E-20	2.13E-96	0.00E+00	5.05E-20	1.26E-95	0.00E+00
F21	0	1.10E-14	2.32E-07	0.00E+00	3.77E-14	2.14E-06	0.00E+00	6.80E-08	2.51E-05	1.72E-73
F22	0	4.18E-02	5.66E-04	1.85E-02	4.18E-02	5.66E-04	1.85E-02	4.18E-02	7.52E-04	3.74E-02
F23	-1	9.17E-01	0.00E+00	-1.00E+00	9.17E-01	0.00E+00	9.01E-01	9.17E-01	0.00E+00	9.01E-01
F24	-1	7.45E-24	1.02E-35	-6.04E-29	6.98E-23	5.67E-35	-6.04E-29	1.42E-22	2.10E-04	4.19E-16
F25	0	2.53E-21	8.04E-01	0.00E+00	1.34E-20	3.83E+00	0.00E+00	2.64E-20	2.05E+01	1.47E-315

TABLE V
MEAN RESULTS ON BENCHMARK FUNCTIONS

Fun.	Mean		
	FA	CMAES	FACmaes
F01	1.64E-10	-8.88E-16	-8.88E-16
F02	1.69E-11	1.20E-19	2.28E-16
F03	-5.41E+01	-8.15E+02	-1.89E+02
F04	-1.00E+00	-1.00E+00	-1.00E+00
F05	2.53E-02	0.00E+00	3.08E-02
F06	1.00E+00	1.89E+00	1.00E+00
F07	9.09E-25	1.53E-10	1.34E-295
F08	1.30E-04	1.72E-03	1.06E-04
F09	1.29E+00	9.34E+00	3.38E+00
F10	4.04E-06	2.54E-02	3.25E-06
F11	9.99E-02	1.01E-01	9.99E-02
F12	1.56E-10	3.13E-46	1.09E-235
F13	5.98E-11	1.41E-43	1.25E-170
F14	1.64E-10	5.40E-46	3.66E-236
F15	7.10E-102	0.00E+00	0.00E+00
F16	-7.38E+01	-7.31E+01	-8.30E+01
F17	-1.91E+01	-1.95E+01	-2.03E+01
F18	-5.50E+05	9.97E+08	-5.75E+06
F19	-1.93E+02	-3.90E+02	-2.31E+02
F20	2.26E-20	3.65E-96	0.00E+00
F21	2.37E-09	3.55E-06	1.72E-74
F22	4.18E-02	5.80E-04	2.30E-02
F23	9.17E-01	0.00E+00	7.11E-01
F24	7.18E-23	7.01E-06	4.19E-17
F25	1.42E-20	5.84E+00	1.48E-316

able to outperform in all the five functions. The five functions are the common functions between this paper and [49]. The functions are F01, F05, F13, F14 and F25.

Table VII displays results on CEC real world problems. It can be seen that FACmaes have best results in terms of best, median, worst and mean results in CEC04 and also

TABLE VII
CEC REAL WORLD PROBLEM

		FA	CMAES	FACmaes
		CEC04	Best	1.43E+01
	Median	1.76E+01	2.11E+01	1.43E+01
	Worst	2.10E+01	2.18E+01	1.48E+01
	Mean	1.76E+01	2.11E+01	1.42E+01
CEC07	Best	5.52E-01	1.62E+00	5.00E-01
	Median	9.90E-01	1.95E+00	7.95E-01
	Worst	1.31E+00	2.29E+00	1.11E+00
	Mean	9.83E-01	1.95E+00	7.92E-01

CEC07 functions. FA method has second best in terms of best, median, worst and mean results in both. The stepping ahead method was able to outperform both FA and CMAES methods. The best result in CEC04 of 13.77 and in CEC07 of 0.5 by FACmaes is the known optimal of the functions.

In Table VIII, the comparison of two CEC 2011 function results of FACmaes are done with methods from one previous research. It can be seen that FACmaes algorithm has best results in both CEC04 and CEC07. While Moth search algorithm comes second best compared to other five algorithms. It can also be seen that FACmaes has outperformed all the methods it was compared to in terms of best and mean.

The Figs. 2-5 display how the solution for functions converges with number of iterations with proposed method and the counterparts. Fig. 2 shows how fast the solution is reached on F11 for FACmaes method while Figs. 3-5 shows convergence on function F20 for the, FA, CMAES and FACmaes method respectively.

TABLE VI
COMPARISON OF THE BEST FUNCTION RESULTS ON BENCHMARK FUNCTIONS WITH LITERATURE [49]

	Best Results						
	ABC	BBO	DE	MS	PSO	SGA	FACmaes
F01	13.35	2.51	16.48	2.1E-8	17.05	2.51	-8.90E-16
F05	30.93	1.79	10.96	1	34.86	1.37	9.86E-03
F13	55	22	37.39	9.4E-8	33.69	13	2.15E-176
F14	20.12	2.2E-16	14.93	1.3E-7	28.14	1	5.63E-240
F25	137.57	39	151.77	7.6E-16	92.98	68.31	0

TABLE VIII
COMPARISON OF BEST AND MEAN OF FACMAES WITH LITERATURE [49] ON CEC REAL WORLD PROBLEM

	Best							Mean						
	ABC	BBO	DE	MS	PSO	SGA	FACmaes	ABC	BBO	DE	MS	PSO	SGA	FACmaes
CEC04	19.81	23.93	20.93	13.79	14.39	23.93	13.77	21.43	23.93	22.22	18.22	20.47	23.93	14.21
CEC07	1.62	1.73	1.88	1.45	1.95	1.62	0.50	2.34	2.13	2.44	2.06	2.42	2.17	0.79

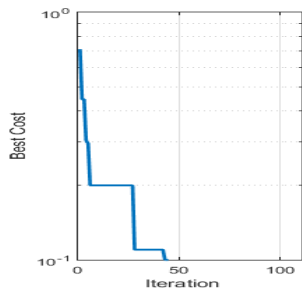


Fig. 2. Typical convergence on F11 for FACmaes method

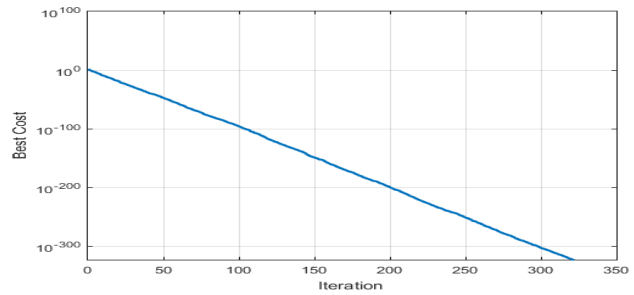


Fig. 5. Typical convergence on F20 for FACmaes method

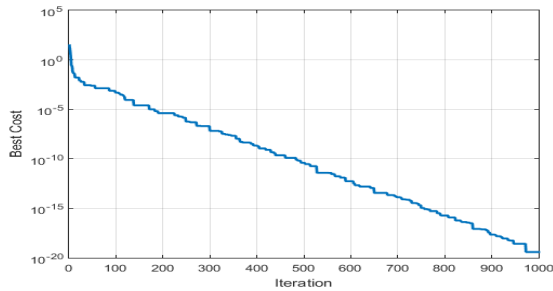


Fig. 3. Typical convergence on F20 for FA method

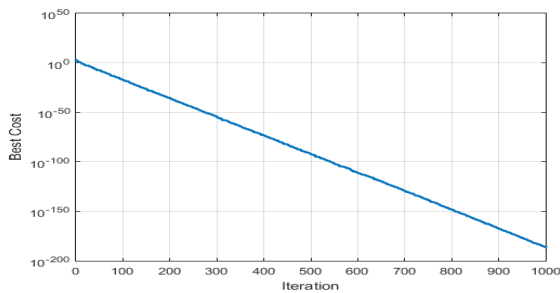


Fig. 4. Typical convergence on F20 for CMAES method

V. DISCUSSION

It was seen that in 19 out of 25 benchmark functions, the newly proposed algorithm in this research paper FACmaes model performed better in comparison to the two standalone methods. For all the benchmark functions, the performance of the algorithms converged towards the known optimal values. This shows that the proposed method is capable enough to search at the optimal search space of any function. The newly proposed method firstly modifies the FA by introducing a new stepping ahead strategy, which provides better solution hunt, i.e. exploration. The wider search space is looked at in this case. Then the hybrid of the method with CMAES takes care of exploitation. The original FA is good at exploitation but sometimes fails to perform in global search as it gets stuck into some local optima [50].

Table V, shows mean results of the three algorithms on the benchmark functions discussed earlier. Again, the proposed method has performed better than the other two methods that is in 15 out of 25 functions. The mean value suggests that out of the 30 runs still the newly proposed stepping ahead FACmaes method outperformed its counterparts. In three cases, the mean obtained for the proposed method is same as the optimal value of that function.

In addition, five of the function results obtained for FACmaes are compared with a research from literature in Table

VI. The author in [49], compares ones result on five other algorithms. In our research, we are comparing our proposed method's result with those algorithms together with the one proposed in research from [49]. It is evident from the table VI that the proposed method FACmaes has outperformed in all the functions compared to the six algorithms. This indicates that the newly proposed algorithm is a better solution to optimization problems.

Table VII shows the result of the proposed method and the other two methods on two real world functions. FACmaes is the only method that has performed well as these functions are hard to solve. FA is the second best in both functions while the stepping ahead proposed method is better in both CEC04 and CEC07. The best results for FACmaes are the known optimal results of the two functions which was done with higher iteration count than what we have used.

The results of FACmaes on CEC04 and CEC07 were compared to other methods in literature in Table VIII. Again, it was seen that the proposed method did better in both cases in terms of the best value and mean.

By looking at Figs. 2-5, further analysis can be done on the performance of individual methods. It can be seen that for function F11 the proposed method converges faster that is in 50 iterations while for F20, it is a constant drop to optimal for the proposed method while FA and CMAES try to look for optimal till the last iteration. Especially for F11, it can be seen that in just 50 iterations, the method was able to converge. The notion to modify the FA before the integration with CMAES method has proved statistically sound. This is evident from the fact, where individual methods were not able to find solutions, the proposed method was able to. Fig. 3 shows that FA method struggles to find the solution and the graph is not smooth while Fig. 4 shows a smooth graph which converges to the optimal. As for the Fig. 5 it has constant convergence, which is only possible if the method is able to find the best fitness in first instance.

The results are carefully reconsidered, in three cases, the worst case values are optimal results as well. These results indicate, that irrespective of running the algorithm many times, the method is able to converge close to optimal value. The results are proof of concept at the preliminary level and more in-depth analysis needs to be carried out to allow the best algorithm to solve such problems faster with better at most optimal results.

The results are compared to the existing literature as benchmark functions are available, and more importantly, the source code of the existing literature is available as well. However, comparison to more algorithms needs to be carried out, which would give more insight into algorithm structure together with problem type.

VI. CONCLUSIONS

This paper employed the Firefly Algorithm (FA) together with covariance matrix adaptation evolution strategy (CMAES) to solve the optimization problem. Firstly, the FA method was modified by introducing a new technique called

stepping ahead and later it was combined with CMAES in the same cycle to avoid increasing the time complexity of the new proposed algorithm. The stepping ahead hybrid algorithm was used to find the optimal solution of benchmark problems extracted from the literature. Experimental results showed that the proposed algorithm has potential to do better than standalone algorithms and some methods from literature. The 25 objective functions were treated as blackbox problems to see if the algorithm is smart enough to search in the optimal search space or not.

The proposed method using a newly designed stepping ahead feature and combination of CMAES method has proved that this algorithm can get better solutions in terms of iteration, runs and even complexity of the problem. The modification gives a chance to search further for better results for the firefly. Applying another method within the loop without iterating the second method further, allows to reduce the running time in terms of time complexity.

The results obtained are proof of concept at the preliminary stage since it is based on benchmark functions and two real-world CEC problems. Future research would focus on parameter tuning to further improve the convergence and success rates. The research can be extended to cater for other optimization functions with in-depth analysis on the use of the algorithms. The results have given scope to test the algorithm in other optimization areas such as combinatorial problems and scheduling.

REFERENCES

- [1] B. Sharma, R. Nand, M. Naseem, and E. V. Reddy, "Effectiveness of online presence in a blended higher learning environment in the pacific," *Studies in Higher Education*, pp. 1–19, 2019.
- [2] A. H. Eden and Y. Hirshfeld, "Principles in formal specification of object oriented design and architecture," in *Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*. IBM Press, 2001, p. 3.
- [3] R. Nand, "Neuron-synapse level problem decomposition method for cooperative coevolution of recurrent networks for time series prediction," in *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2016, pp. 3102–3109.
- [4] E. Reddy, P. Reddy, B. Sharma, K. Reddy, and M. Khan, "Student readiness and perception to the use of smart phones for higher education in the pacific," in *2016 3rd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)*. IEEE, 2016, pp. 258–264.
- [5] B. Sharma and P. Reddy, "Effectiveness of tablet learning in online courses at university of the south pacific," in *2015 2nd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)*. IEEE, 2015, pp. 1–9.
- [6] R. Nand, A. Sharma, and K. Reddy, "Skill-based group allocation of students for project-based learning courses using genetic algorithm: Weighted penalty model," in *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. IEEE, 2018, pp. 394–400.
- [7] R. Nand and A. Sharma, "Meta-heuristic approaches to tackle skill based group allocation of students in project based learning courses," in *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 1782–1789.
- [8] D. Leonard, N. Van Long, and V. L. Ngo, *Optimal control theory and static optimization in economics*. Cambridge University Press, 1992.
- [9] B. Babu and A. M. Gujarathil, "Multi-objective differential evolution (mode) for optimization of supply chain planning and management," in *2007 IEEE Congress on Evolutionary Computation*. IEEE, 2007, pp. 2732–2739.

- [10] B. Sharma, J. Vanualailai, and S. Singh, "Motion planning and posture control of multiple n-link doubly nonholonomic manipulators," *Robotica*, vol. 35, no. 1, pp. 1–25, 2017.
- [11] A. Prasad, B. Sharma, and J. Vanualailai, "A solution to the motion planning and control problem of a car-like robot via a single-layer perceptron," *Robotica*, vol. 32, no. 6, pp. 935–952, 2014.
- [12] I. Aydin, M. Karakose, and E. Karakose, "A navigation and reservation based smart parking platform using genetic optimization for smart cities," in *2017 5th International Istanbul Smart Grid and Cities Congress and Fair (ICSG)*. IEEE, 2017, pp. 120–124.
- [13] C. A. Floudas and P. M. Pardalos, *Recent advances in global optimization*. Princeton University press, 2014.
- [14] R. V. Rao and V. Patel, "Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm," *Applied Mathematical Modelling*, vol. 37, no. 3, pp. 1147–1162, 2013.
- [15] P. Xu, W. Luo, X. Lin, Y. Qiao, and T. Zhu, "Hybrid of pso and cma-es for global optimization," in *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 27–33.
- [16] G.-G. Wang, L. Guo, H. Duan, and H. Wang, "A new improved firefly algorithm for global numerical optimization," *Journal of Computational and Theoretical Nanoscience*, vol. 11, no. 2, pp. 477–485, 2014.
- [17] S. Łukasik and S. Żak, "Firefly algorithm for continuous constrained optimization tasks," in *International conference on computational collective intelligence*. Springer, 2009, pp. 97–106.
- [18] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík, "Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009," in *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*. ACM, 2010, pp. 1689–1696.
- [19] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *International symposium on stochastic algorithms*. Springer, 2009, pp. 169–178.
- [20] D. P. Rini, S. M. Shamsuddin, and S. S. Yuhaziz, "Particle swarm optimization: technique, system and challenges," *International journal of computer applications*, vol. 14, no. 1, pp. 19–26, 2011.
- [21] N. Hansen, A. Ostermeier, and A. Gawelczyk, "On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation," in *ICGA*, 1995, pp. 57–64.
- [22] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [23] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es)," *Evolutionary computation*, vol. 11, no. 1, pp. 1–18, 2003.
- [24] P. J. Shailja Sharma and A. Saxena, "Adaptive inertia-weighted firefly algorithm," *Intelligent Computing Techniques for Smart Energy Systems*, pp. 495–503, 2020.
- [25] D. Aggarwal and V. Kumar, "Performance evaluation of distance metrics on firefly algorithm for vrp with time windows," *International Journal of Information Technology*, Nov 2019. [Online]. Available: <https://doi.org/10.1007/s41870-019-00387-7>
- [26] A. Yelghi and C. KÄse, "A modified firefly algorithm for global minimum optimization," *Applied Soft Computing*, vol. 62, pp. 29 – 44, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1568494617306373>
- [27] D. F. Teshome, C. H. Lee, Y. W. Lin, and K. L. Lian, "A modified firefly algorithm for photovoltaic maximum power point tracking control under partial shading," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 5, no. 2, pp. 661–671, June 2017.
- [28] L. He and S. Huang, "Modified firefly algorithm based multilevel thresholding for color image segmentation," *Neurocomputing*, vol. 240, pp. 152 – 174, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S09525231217303314>
- [29] J.-S. Chou and N.-T. Ngo, "Modified firefly algorithm for multidimensional optimization in structural design problems," *Structural and Multidisciplinary Optimization*, vol. 55, pp. 2013–2028, 2017. [Online]. Available: <https://doi.org/10.1007/s00158-016-1624-x>
- [30] J.-P. J. Bin Wang, Dong-Xu Li and Y.-H. Liao, "A modified firefly algorithm based on light intensity difference," *Journal of Combinatorial Optimization*, vol. 31, pp. 1045–1060, 2016. [Online]. Available: <https://doi.org/10.1007/s10878-014-9809-y>
- [31] I. Fister, X.-S. Yang, J. Brest, and I. Fister, "Modified firefly algorithm using quaternion representation," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7220 – 7230, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417413004697>
- [32] O. P. Verma, D. Aggarwal, and T. Patodi, "Opposition and dimensional based modified firefly algorithm," *Expert Systems with Applications*, vol. 44, pp. 168 – 176, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417415006119>
- [33] A. Kavousi-Fard, H. Samet, and F. Marzbani, "A new hybrid modified firefly algorithm and support vector regression model for accurate short term load forecasting," *Expert Systems with Applications*, vol. 41, no. 13, pp. 6047 – 6056, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417414001912>
- [34] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, p. 35, 2013.
- [35] L. Davis, "Handbook of genetic algorithms," 1991.
- [36] Y. Marinakis and M. Marinaki, "A hybrid genetic à particle swarm optimization algorithm for the vehicle routing problem," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1446 – 1455, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417409006460>
- [37] P. Dziwiński, Ł. Bartczuk, and P. Goetzen, "A new hybrid particle swarm optimization and evolutionary algorithm," in *Artificial Intelligence and Soft Computing*, L. Rutkowski, R. Scherer, M. Korytkowski, W. Pedrycz, R. Tadeusiewicz, and J. M. Zurada, Eds. Cham: Springer International Publishing, 2019, pp. 432–444.
- [38] X. Wang, H. Peng, C. Deng, L. Li, and L. Zheng, "An improved firefly algorithm hybrid with fireworks," in *Computational Intelligence and Intelligent Systems*, H. Peng, C. Deng, S. Wu, and Y. Liu, Eds. Singapore: Springer Singapore, 2019, pp. 27–37.
- [39] Y. Yu, L. Yang, Y. Wang, and S. Gao, *Brain Storm Algorithm Combined with Covariance Matrix Adaptation Evolution Strategy for Optimization*. Cham: Springer International Publishing, 2019, pp. 123–154. [Online]. Available: https://doi.org/10.1007/978-3-030-15070-9_6
- [40] B. K. Tripathy and A. Agrawal, "A comprehensive analysis of kernelized hybrid clustering algorithms with firefly and fuzzy firefly algorithms," in *Computational Intelligence in Data Mining*, H. S. Behera, J. Nayak, B. Naik, and D. Pelusi, Eds. Singapore: Springer Singapore, 2020, pp. 351–365.
- [41] N. A. Al-Thanoon, O. S. Qasim, and Z. Y. Algamal, "A new hybrid firefly algorithm and particle swarm optimization for tuning parameter estimation in penalized support vector machine with application in chemometrics," *Chemometrics and Intelligent Laboratory Systems*, vol. 184, pp. 142 – 152, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0169743918305446>
- [42] J. Wang, M. Zhang, H. Song, Z. Cheng, T. Chang, Y. Bi, and K. Sun, "Improvement and application of hybrid firefly algorithm," *IEEE Access*, vol. 7, pp. 165 458–165 477, 2019.
- [43] M. Dorigo and M. Birattari, *Ant colony optimization*. Springer, 2010.
- [44] Z.-Z. Liu, Y. Wang, S. Yang, and K. Tang, "An adaptive framework to tune the coordinate systems in nature-inspired optimization algorithms," *IEEE transactions on cybernetics*, vol. 49, no. 4, pp. 1403–1416, 2018.
- [45] Y. Wang, Z.-Z. Liu, J. Li, H.-X. Li, and G. G. Yen, "Utilizing cumulative population distribution information in differential evolution," *Applied Soft Computing*, vol. 48, pp. 329–346, 2016.
- [46] "BenchmarkFncs Toolbox," <http://benchmarkfncs.xyz/fncs>, accessed: 2019-07-24.
- [47] S. Das and P. N. Suganthan, "Problem definitions and evaluation criteria for cec 2011 competition on testing evolutionary algorithms on real world optimization problems," *Jadavpur University, Nanyang Technological University, Kolkata*, pp. 341–359, 2010.
- [48] J. McCard and A. Almomani, "New criteria for comparing global stochastic derivative-free optimization algorithms," 2019.
- [49] G.-G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing*, vol. 10, no. 2, pp. 151–164, 2018.
- [50] M. Sababha, M. Zohdy, and M. Kafafy, "The enhanced firefly algorithm based on modified exploitation and exploration mechanism," *Electronics*, vol. 7, no. 8, p. 132, 2018.