



A HYBRID OPTIMIZATION APPROACH FOR COMPLEX NONLINEAR OBJECTIVE FUNCTIONS

Samuel O. Obadan, Zenghui Wang

College of Science, Engineering and Technology, University of South Africa,
Florida 1710, South Africa
obadansam@gmail.com, wangzengh@gmail.com

Paper history:

Received 15 February 2018
Received in revised form 09 May 2018
Accepted 25 May 2018
Available online 30 May 2018

Keywords:

Simulated Annealing;
Particle Swarm Optimization;
Genetic Algorithm;
Cuckoos-Search Algorithm;
Hybridization.

Abstract: With respect to the ‘no free lunch’ theorem, no single algorithm has a better performance when tested against a completely stochastic algorithm on all objective functions. Consequently, choosing the best algorithm for a particular problem is often more of an art than science. The complexity of an objective function can be determined by certain features such as the modality, the basins, the valleys, the separability, and the dimensionality of the objective function. While the separability and modality contribute to the complexity of the function, the dimensionality and domain range increases the function’s search space exponentially. In this paper, the authors analyze the algorithmic constructs of Simulated Annealing (SA), Cuckoo-search (CK), Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) along with two hybrid paradigms. In addition, an extensive comparative study was conducted using 30 standard benchmark functions to demonstrate how an ingenious hybrid algorithm could significantly shorten the amount of function calls (generations) needed to attain the optimal or rather near optimal solution for almost any complex objective function. Results from empirical analysis unveil the precision, robustness and success of the hybrid algorithm (without compromising run-time complexity) over its counterparts.

*Copyright © Research Institute for Intelligent Computer Systems, 2018.
All rights reserved.*

1. INTRODUCTION

Optimization is a branch of science that involves the search for the best parametric values to a solution of a specific problem [1]. The objective is to find the solution to a predefined objective function via an iterative process, towards an optimal value. In optimization problems, a mathematical representation of the objective function is clearly defined along with its constraints.

Optimization problems are usually classified under various paradigms such as linear programming, quadratic programming, combinatorial optimizations and meta-heuristics. While classical algorithms use methods based on the hessian matrix [2] and gradient decent (especially for objective functions with computable derivatives), the meta-heuristic algorithms on the other hand are deployed on non-differentiable and non-linear objective functions. Such functions are either

impracticable or very difficult to solve using classical methods. However, solutions provided by meta-heuristic algorithms are usually referred to as suboptimal solutions.

Some of the raciest meta-heuristic algorithms include: Genetic algorithm (GA), Particle Swarm Optimization (PSO), Simulated Annealing (SA), Differential Evolution Algorithm (DE), Cuckoo Search Algorithm (CK) etc. These algorithms leverage on a model matrix which evolves random solutions to the predefined objective function. In addition, some of these meta-heuristic algorithms use the variant of the basic genetic algorithm schema (selection, Mutation, and crossover) while evolving solutions. These variants can be summarized into two basic strategies namely, exploration and exploitation. While exploitation targets the best local solution within the search space, exploration attempts to leverage diversification in an attempt to

incur the best solution which in most cases lies around one of the local solutions.

A good meta-heuristic algorithm can be characterized by the rate at which it finds the global optimal solution to the objective function. In this paper, the authors propose a frame work for hybridization of multiple meta-heuristic algorithms. This frame work leverages on the strengths of each meta-heuristic algorithm in order to rapidly converge a search process to an optimal or suboptimal solution with minimal computational complexity.

Objective functions (Table 1) could be qualified or grouped into categories such that they are either continuous functions or discontinuous functions, linear functions or polynomial, differentiable or non-differentiable, uni-modal or multi-modal, separable or non-separable. In this paper, some artificial multifarious problems also referred to as test functions has been chosen to evaluate the robustness of our proposed algorithm. Artificial problems has the advantage of ease in modification and manipulation of the test algorithm in diverse scenarios. In addition, objective functions can be sorted or grouped by their modality, basins, valleys, separability and dimensionality [3].

Modality: This represents the number of peaks in the function's topology. When an algorithm come across such peaks during a search cycle, there exist the likelihood of the algorithm to asymptote at a local optima or minima depending on the predefined search criteria.

Basins: unlike peaks, these are steep decline around a large area. The presence of basins could have a significant impact on the success of an algorithm due to insufficient information to guide the algorithm towards the global minima.

Valleys: These occur when narrow domains of minimal difference is surrounded by multiple basins. The floor of a valley could have a significant impact on the success of a search algorithm.

Separable: This measures the difficulty of a function. It is easier for a search algorithm to transverse a separable function than a non-separable function. When the variables of a function are independent of each other the derivative of the function can be decomposed into sub functions. This separable feature makes it easier for an algorithm to solve. On the other hand, if the variables are dependent on each other, the function becomes non-separable thus making it more difficult for an algorithm to solve.

Dimensionality: the magnitude of the parametric variables defines the dimensionality of the objective function. Every one step increase in the number of parameters has an exponential overhead in the amount of computational search space. Almost every

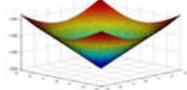
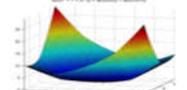
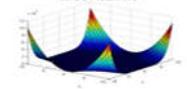
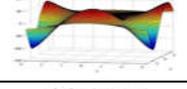
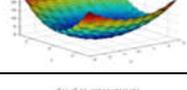
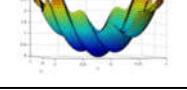
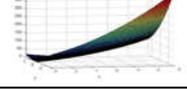
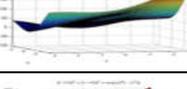
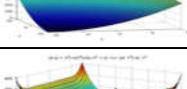
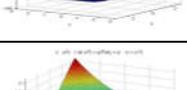
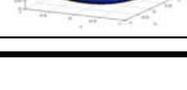
meta-heuristic algorithm has dimensionality as a major bottleneck.

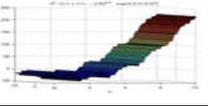
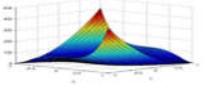
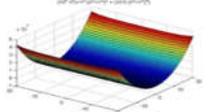
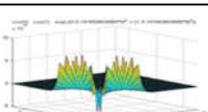
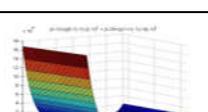
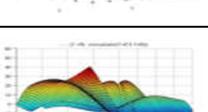
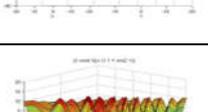
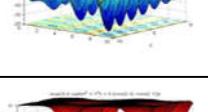
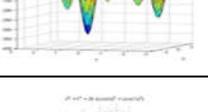
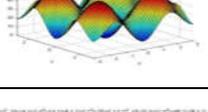
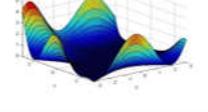
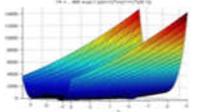
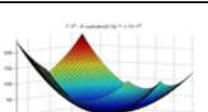
The complexity of the function can determined by two major factors: modality and separability as depicted in table 1(a) below.

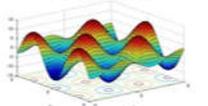
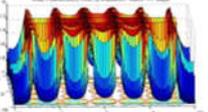
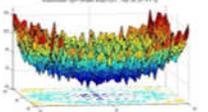
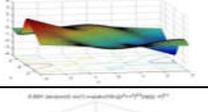
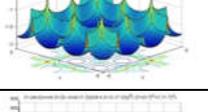
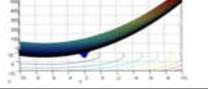
Table 1(a) Complexity variants

Objective function	Complexity level
Single peaks (unimodal), separable	LOW (L)
(Unimodal, non-separable) OR (multi-modal, separable)	MEDIUM (M)
Multiple peaks (multi-modal), non-separable	HIGH (H)

Table 1(b) (30 bench mark functions)

Label	Function name/Complexity	Function Plot	F(x)*
F1	Ackley2 Function [4] (M)		-200
F2	Bartels Conn Function [3] (H)		1
F3	Beale Function [3] (M)		0
F4	Bird Function [3] (H)		-106.76
F5	Bohachevsky Function [5] (M)		0
F6	Bohachevsky Function [5] (H)		0
F7	Booth Function [3] (M)		0
F8	Branin RCOS-2 Function [6] (H)		-39.195
F9	Brent Function [7] (M)		0
F10	Camel Function - Six Hump [7] (H)		-1.0316
F11	Camel Function - Three Hump [7] (H)		0

F12	Chichinadze Function [3] (M)		-42.9444
F13	Cube Function [8] (M)		0
F14	Deckkers-Aarts Function [9] (H)		-24776.5
F15	Easom Function [10] (M)		-1
F16	Freudenstein Roth Function [11] (H)		0
F17	Haupt Function 16 [2] (M)		-25.2305
F18	Haupt Function 07 [2] (H)		-18.5547
F19	Haupt Function 15 [2] (M)		-345.36
F20	Egg Crate Function [3] (M)		0
F21	Goldstein Price Function [12] (H)		3
F22	Rosenbrock Modified Function [13] (H)		34.0412
F23	Rotated Ellipse Function [3] (M)		0
F24	Scahffer-1 Function [14] (M)		0

F25	Test-tube Holder Function [15] (M)		-10.872
F26	Pen-Holder Function [15] (H)		-0.9635
F27	Trefethen Function [16] (H)		-3.388
F28	Adjiman Function [17] (H)		-2.02181
F29	Cross-in-Tray Function [15] (H)		-2.06261
F30	Damavandi Function [18] (H)		0

2. LITERATURE REVIEW

Evolution computation is a combination of genetic programming and genetic algorithm which incorporates models such as selection and mutation which forms the core of the entire evolution computation algorithm.

Genetic algorithm (GA) is a classical optimization meta-heuristic based on the biological model of natural selection. The algorithm involves a clever manipulation of an objective function, a vector or matrix of objective variables, definition of variable constraints, selection, crossover and mutation (Fig. 1). The total number of iterations (epochs) usually depends on the chosen termination criteria which could either be a predetermined number of epochs or the convergence of the algorithm. The GA is said to have converged when little or no significant improvement is observed in the population.

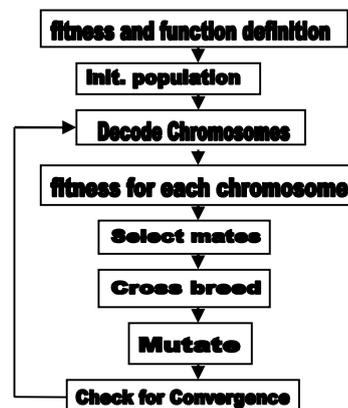


Figure 1 – Flowchart for genetic algorithm.

Two key process guides the GA towards an optimal solution: the selection and the crossover [19]. The roulette wheel selection model is a typical and intuitive probabilistic approach which favors the best pair of objective variables (variables with high fitness scores) for participating in the mating process.

The mating process is implemented by the crossover model. A typical procedure is binary fission of binary encoded objective variables. A random split point is chosen for both selected string of binary coded bits, thereafter an exchange (crossover) is conducted.

A subtle but very efficient model is the mutation process. Given a predetermined probabilistic mutation rate, the resulting string from the crossover process may undergo an alteration in one or more of its bits. This procedure helps the algorithm's converging process progress towards the global optima.

2.1. SIMULATED ANNEALING (SA)

The simulated annealing algorithm as a meta-heuristic optimizer was the 'brain child' of Kirkpatrick et. al., in the year 1983 [20]. The algorithm mimics the process of a crystal-like lattice via a quick heating and slow cooling process. Like other standard procedures, the algorithm begins with generating a random number of objective variables that are modified via some parametric turning before assigning them to a fitness function which then outputs a fitness score for each pair/vector of variables.

$$v(new) = d \cdot v(old) \tag{1}$$

where,

- $v(new)$ = new variables,
- $v(old)$ = old variables,
- d = control variable.

It's important to note that most literature refer to a set of objective variables as a chromosome. In the Simulated annealing iterative algorithm, a new set of objective values replaces the old ones if there was an improvement in their corresponding fitness scores, however some of the less fit pairs may proceed to the next generation even if their fitness scores worsened as long as they satisfy the following conditions:

$$r \leq e^{[f(old)-f(new)]/T} \tag{2}$$

Otherwise, they are rejected.

where,

- r = uniform stochastic variable,

T = temperature.

The algorithm slowly reduces the T value until it gets close to zero before terminating. During this cooling process, the algorithm does a percentage wise reduction of the d value in an attempt to enhance the fitness scores of the population.

2.2. PARTICLE SWAM OPTIMIZATION (PSO)

The PSO algorithm is a relatively less involved algorithm with few parameters to manipulate, first proposed by Kennedy and Eberhart [21]. Like the Genetic algorithm, the PSO meta-heuristic mimics a biological model however, it exempts the crossover and mutation procedures of the GA.

The PSO algorithm iteratively updates the objective variables via a velocity vector. For brevity, the algorithm modifies each set of objective values updating their vector velocities via a clever manipulation of the global best and local best solutions. While the global best is indicative of the best fitness score thus far in the iterative process, the local best is indicative of the best fitness score within the current run. The equation below show the simplicity of this elegant algorithm:

$$vel_{new} = vel_{old} + \gamma \cdot r1 \cdot (P_{local\ best} - P_{old}) + \gamma \cdot r2 \cdot (P_{global\ best} - P_{old}) \tag{3}$$

$$P_{new} = P_{old} + vel_{new} \tag{4}$$

where,

- vel = velocity of each particle,
- P = particle variables,
- $P_{local\ best}$ = best local fitness for each particle,
- $P_{global\ best}$ = global local fitness for each particle,
- γ = learning rate (constant),
- $r1, r2$ = stochastic variables.

The ease of implementation is another significant advantage of this algorithm.

2.3. CUCKOOS SEARCH ALGORITHM (CS)

The cuckoos search meta-heuristic algorithm was first proposed by Deb and yang [22]. The algorithm was inspired the interesting reproductive characteristics of the cuckoos' bird. The cuckoos' bird is an opportunistic bird that lays its eggs among other eggs in a host nest. On return of the host bird, the host bird may or may not detect the presence of the cuckoos' egg. If the cuckoo egg is undetected, all eggs are hatched otherwise, the nest is completely abandoned or ruined.

The CS meta-heuristic combines the behavior of the host bird and the cuckoos' bird. Intuitively, each nest is a representation of a set objective variables.

The algorithm first generates an N – population pair or vector of objective variables usually referred to as candidate solutions in literature. Next, the cuckoos’ egg is laid in a randomly chosen nest using a typical random walk ‘levy flight’ approach:

$$xk' = xk + rand . (levy\ flight)xk \quad (5)$$

$$yk' = yk + rand . (levy\ flight)yk \quad (6)$$

Next, the fitness of the nest with the cuckoos’ egg is compared with the host nest. The host nest is replaced if it has a worse fitness score when compared with the cuckoo’s nest. However, if the host bird notices the presence of the of the cuckoos egg, the nest is discarded usually with a probability $p < 0.25$ consequently creating a new nest.

2.4. HYBRID OPTIMIZATION

A typical hybrid algorithm blends the strengths of genetic algorithms along with the converging speed of any local optimizer. A couple of authors such as Kazarlis et al. [23] implemented a scaled down genetic algorithm with a minute population size as a local optimizing strategy. The rationale behind hybridization is often to combine the power of GAs (exploration) with the swiftness at which a local optimizer asymptotes (exploitation). When the GA seem to gradually asymptotes, its assumed to at least be in the domain of the global solution, thereafter the local optimizing algorithm seizes the search process in an attempt to obtain optimal solution. Hybridization could be in any of the forms below:

1. Beginning with a GA until it decelerates before seeding a local optimizer
2. Start the GA with some local minima obtained from random starting points in the population
3. After a predefined number of iterations, seed a local optimizer on a selected elite population using elitism and incorporate the resulting chromosome into the population. Haupt [2] demonstrated finding the global optima by combining a continuous GA with Nelder-Mead downhill simplex algorithm.

3. HYBRID METHODOLOGY

Preliminary analysis of the reviewed meta-heuristic algorithms revealed the strengths of each algorithm on the 30 different benchmark objective functions. The classical GA employs a moderately balanced explorative and exploitative strategy while the cuckoos search algorithm is highly explorative. This quality of the cuckoo’s algorithm gives it an edge over the classical GA when deployed on

complex objective functions. The polygamy induced GA provides a highly exploitative strategy. These diverse capabilities informed our choice of algorithms for the creation of the hybridized model.

The proposed hybrid model leverages on any three different meta-heuristics combinations for solving the optimization problem. The three meta-heuristic of choice are the GA, CK, and POLY (i.e. GA-with polygamy). Using 3 meta-heuristics avails us with 3 factorial (3!) possible unique combinations of the meta-heuristics. For example [CK, GA, POLY] with [CK², GA⁴, POLY⁵] where the superscripts represents the duration of sub epoch assigned to each meta-heuristic. The algorithm (Table 2.) begins by first sampling a random population of 50 (M) chromosomes with a scalable dimension size of 2 (x_1, x_2) continuous variables. Simultaneously, a sub population ($p \leq M$) of random combinations of meta-heuristics are spurned to evolve or optimize the matrix of chromosomes towards an optimal solution to the given objective function. It is important to note that each meta-heuristic randomly obtains a duration ($f_i^{k(i)}$) of range [0, 5]. Thus the maximum number of sub epoch for each $p \leq N(\max)$ where $N(\max) = 15$.

From Fig. 2, the COMa represents the stochastic combination or order of meta-heuristic implementation. A typical initial order for COMa = 0, could be [CK², GA⁴,POLY⁵], for [INTa, INTb, INTc] respectively, with the supper scripts representing the number of sub epochs (duration) each algorithm is permitted to run.

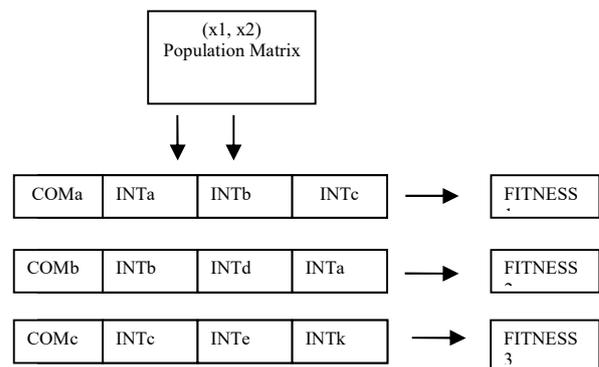


Figure 2 – High level abstraction of the hybrid framework

At the end of the first main epoch, a typical GA style algorithm (without the crossover operator) is used to select and mutate the p chromosomes. Exempting the crossover operator reduces the computational complexity and helps the hybrid algorithm evolve faster.

Table 2. Algorithm for Hybridized Model

$X^* \rightarrow 0$, $M \rightarrow$ population size, $N \rightarrow$ duration of sub epochs $k = duration(randint[0\ 5])$ $p = \{f_1^{k(1)}, f_2^{k(2)}, \dots \dots f_n^{k(n)}\} \leq M$ $f_i^{k(i)}$ $= metaheuristic(i) \text{ with } duration(k(i))$ $N = \sum_{i=0}^n f_i^{k(i)} \forall f_i \in P, s.t \ n \leq N(\max)$ <ol style="list-style-type: none"> 1. Sample New Random population (p) of size M 2. While (M) 3. For each ($p(i) < N$) 4. Evaluate $p(i) = \{f_1^{k(1)}, f_2^{k(2)}, \dots \dots f_n^{k(n)}\}$ 5. $X^* \leftarrow$ update best local optima for each $p(i)$ 6. End (for loop) 7. Preserve elite schema 8. Evaluate mutation condition = (TRUE) 9. Mutate ($f_i^{k(i)}$) order of combinations, mutate ($k(i)$) duration 10. Return best $p(i)$ schema order, X^{best} \rightarrow parameters, $X^{best} \rightarrow$ global optima

Elitism is used to preserve the best 5 performing chromosomes to the next generation. Fig. 3 shows the best performing combinations for the overall objective benchmark functions used in this research. The mutation process alters the combination or order of meta-heuristics along with their respective superscript durations.

Leveraging tournament selection, as a selection strategy, the fittest chromosomes are passed to the next generation. X^* (global minima) is updated at the end of each main epoch. For the research, the termination criteria was set at 50 epochs (MAX) or when the optimal solution has been found.

3.1. Polygamy as an Exploitative Strategy for GAs

The concept of diversity and exploitation are two paradigms that has contributed immensely to the success of the GA. While diversity attempts to prevent the algorithm from stagnating within a local optima, exploitation on the other hand attempts to achieve faster convergence of the algorithm.

One approach towards maintaining diversity within a population is by replacing existing identical solution strains with newly formed strains especially

in cases where they exist multiple similar strains in the population. [24]

Other methods include a mechanism for favoring dissimilar strains while similar ones are discouraged leading to convergence on multiple peaks [25]. Another related approach is to restrict mating among similar strains while encouraging mating among dissimilar ones thereby increasing diversity [26]. Similarly, a tag stamping mechanism has been used to indicate strains that are eligible to mate as they pass from one generation to another [27].

Polygamy one the other hand attempts to explore the power of exploitation. Using this approach, every strain within the population is forced to mate with the fittest strain within each generation. A clever implementation of this strategy helped the algorithm converge faster to the global optima. Polygamy behavior was allowed when only little improvement was observed within five consecutive generations' thus fine tuning population towards the global optima.

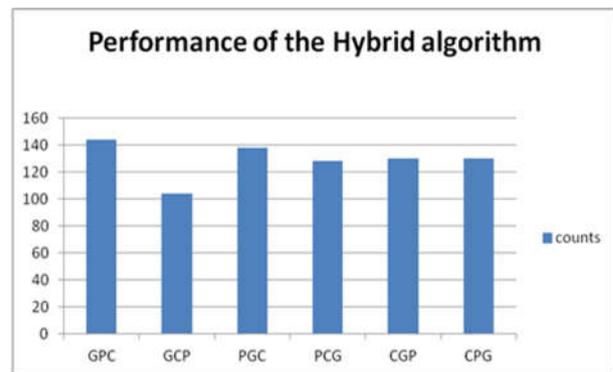


Figure 3 – Hybrid frame work performance chart for all objective functions evaluations. Height indicative of the number of times each combination was responsible for the optimal solution.

4. EXPERIMENTS

On the basis of ANOVA, we reject the NULL Hypothesis (H_0) on the premise of a significant difference between the mean optimal values of our tested meta-heuristics. Consequently, we proceed with our analysis to discover the pairs of meta-heuristics that differ significantly. For this purpose, we use the SHEFFE's test. The test states that "the critical difference (CD) for each pair of meta-heuristic can be obtained using the equation below:"

$$CD_{ith,jth} = \sqrt{MS_{within} \left(\frac{1}{n_i} + \frac{1}{n_j} \right) * (k - 1) * f_{k-1, n-k} * \alpha * n} \quad (7)$$

where, α = critical value at 5% significance.
 k = number of meta-heuristics

n = number of test bench mark functions
 MS_{within} = ANOVA mean square within samples

In this paper, we use 30 bench mark functions to test the success of the CK, PSO, SA, POLY (a proposed exploitative strategy), and HYB (our proposed hybrid frame work). Table 3 shows the best optimal values for each meta-heuristic on the bench mark functions compared against standard expected optimal solutions $f(x^*)$. A fixed dimension size of 2 was implemented for all bench mark functions. The population size was preset at 50 while the maximum number of function evaluations for each iteration was set to 500. For the purpose of cohesion, the global minimal values below 10^{-15} were considered as zero (0) in all experiments.

In this paper, global optima (minima (X^{best})) of each bench mark function was evaluated 20 times using random initial population at every instance. The mean performance of optimal solutions and durations (number of function calls) during each phase of the experiments were recorded for further analysis.

Also, two ANOVA tests were conducted for multiple comparisons of the performance of each meta-heuristic algorithm.

HYPOTHESIS 1 (H1)

The Null hypothesis for the first ANOVA test is stated as follows: ***“There is no difference in the speed of convergence to the global optima among test meta-heuristic algorithms”*** (Table 4)

HYPOTHESIS 2 (H2)

The Null hypothesis for the second ANOVA test is stated as follows: ***“There is no difference in the mean global solutions between the tested meta-heuristics”*** (Table 5)

Both hypothesis were tested with 95% confidence ($\alpha = 0.05$).

Considering the relative complexity of the Hybrid algorithm, the algorithm was allowed to run for $1/10^{th}$ of the max allowed epoch of 500. The rational was to give all algorithms a level playing ground as each one ran for approximately equal CPU time.

Table 3. Minimum optimal values for each meta-heuristic $f(x^*)$ = cross validating global optimal Values.

	CK	SA	GA	PSO	POLY	HYB	$f(x^*)$
F1	-199.879	-199.931	-200	-197.815	-200	-200	-200
F2	1.69358	2.75259	1	53.5344	1	1	1
F3	0	0.000191	0	0.0696428	0	0	0
F4	-106.76	-106.736	-106.765	-105.159	-106.765	-106.765	-106.765
F5	0.0113735	0.08879	0	2.17693	0	0	0
F6	0.0138544	0.2025	0	2.8217	0.000000115	0	0
F7	0.0002679	0.000888	0	0.002881	0.000000017	0.00000023	0
F8	-39.185	-34.613	-37.1968	-34.88	-35.032	-39.1956	-39.1956
F9	0	0	1.40746	0	1.99512	0	0
F10	-1.03115	-1.03158	-1.03164	-0.997461	-1.03163	-1.03163	-1.03163
F11	0.0000506	0.000115	0	0.0007521	0	0	0
F12	-42.9638	-42.9362	-42.9444	-42.4961	-42.9444	-42.9444	-42.9444
F13	0.000774	0.005506	0.000366	0.05672	0.001606	0.00000065	0
F14	-24761.9	-24770.2	-24776.5	-24634.7	-24776.5	-24776.5	-24776.5
F15	-0.994082	-0.97298	-1	-0.0003	-1	-1	-1
F16	0.0093124	0.05106	0	0.06918	0.00016211	0.00000086	0
F17	-25.2292	-25.2269	-25.2305	-25.2133	-25.2305	-25.2305	-25.2305
F18	-18.5458	-18.23	-18.5547	-17.7819	-18.5547	-18.5547	-18.5547
F19	-345.242	-343.827	-345.36	-325.002	-345.36	-345.36	-345.36
F20	0.0007833	0.000198	0	0.02121	0	0	0
F21	3.01455	3.00531	3	3.00764	3.00041	3	3
F22	34.0975	34.0825	34.5025	74.0162	53.6992	34.0412	34.0412
F23	5.43332	1.87413	0	67.995	0.003607	0.000000003	0
F24	1.9E-07	3.8E-07	0	0.121312	0	0	0
F25	-10.8698	-10.8718	-10.8723	-10.8391	-10.8723	-10.8723	-10.8723
F26	-0.963513	-0.96354	-0.96352	-0.936979	-0.963535	-0.963535	-0.963535
F27	-3.29763	-3.25321	-3.38008	-2.69568	-3.38282	-3.38814	-3.38814
F28	-2.02181	-2.02181	-1.91849	-1.79863	-1.99297	-2.02181	-2.02181
F29	-2.06248	-2.06261	-2.06261	-2.0489	-2.06261	-2.06261	-2.06261
F30	0.065897	2.06078	2	2.00756	2	0.00311832	0

Table 4. ANOVA analysis of mean function calls of each meta-heuristic for all objective functions. (* indicates “no significant difference”)

The significantly different algorithms based on Mean Function calls (500 EPOCHS)

Function	P-value	Algorithms					
		CK	SA	GA	PSO	POLY	HYB
F1	1.82898E-17	*	*	CK,SA, POLY, PSO	*	*	CK, SA, PSO, POLY
F2	1.07148E-10	*	*	CK	*	*	CK, POLY
F3	2.08378E-43	*	*	*	*	*	CK, SA, PSO, POLY, GA
F4	2.49674E-29	*	*	CK	*	CK	CK, SA, PSO, POLY, GA
F5	2.20353E-16	*	*	CK, POLY	*	CK	CK, POLY
F6	1.73217E-14	*	*	*	*	*	CK, SA, PSO, POLY, GA
F7	3.20673E-09	*	*	*	*	*	CK, SA, PSO, POLY, GA
F8	1.23486E-29	*	*	*	*	*	CK, SA, PSO, POLY, GA
F9	2.75464E-98	GA, PSO, POLY	*	*	*	*	GA, PSO, POLY
F10	9.70E-27	*	*	CK, SA, PSO	*	*	CK, SA, PSO, POLY, GA
F11	1.6558E-07	*	*	CK, SA, PSO, POLY	*	*	CK, SA, PSO
F12	1.0131E-27	*	*	CK, SA, PSO, POLY	*	*	CK, SA, PSO, POLY, GA
F13	-----	*	*	*	*	*	*
F14	9.59387E-26	*	*	CK, SA, PSO	*	*	CK, SA, PSO, POLY, GA
F15	1.59751E-16	*	*	CK, SA, PSO	*	*	CK, SA, PSO, POLY, GA
F16	-----	*	*	*	*	*	*
F17	2.50691E-48	*	*	*	*	*	CK, SA, PSO, POLY, GA
F18	9.84919E-28	*	*	*	*	*	CK, SA, PSO, POLY, GA
F19	7.07297E-21	*	*	CK, SA, PSO	*	*	CK, SA, PSO, POLY
F20	5.20026E-08	*	*	CK, SA, PSO, POLY	*	*	*
F21	1.48091E-11	*	*	*	*	*	CK, SA, PSO, POLY, GA
F22	-----	*	*	*	*	*	*
F23	-----	*	*	*	*	*	*
F24	1.8132E-15	*	*	CK, SA, PSO, POLY	*	*	CK, SA, PSO, POLY
F25	4.61192E-27	*	*	CK, SA, PSO	*	*	CK, SA, PSO, POLY, GA
F26	1.65387E-32	*	*	*	*	*	CK, SA, PSO, POLY, GA
F27	-----	*	*	*	*	*	*
F28	1.34422E-45	GA, PSO, POLY	*	*	*	*	CK, SA, PSO, POLY, GA
F29	1.52471E-39	*	*	CK, SA, PSO, POLY	*	*	CK, SA, PSO, POLY
F30	-----	*	*	*	*	*	*

Table 5. ANOVA analysis of mean global optimal values of each meta-heuristic for all objective functions (* indicates “no significant difference”)

The significantly different algorithms based on minimum optimal values (X*) @ 500 EPOCHS

Function	P-value	Algorithms					
		CK	SA	GA	PSO	POLY	HYB
F1	4.42439E-45	PSO	PSO	SA, PSO	*	PSO	SA, PSO
F2	5.73609E-20	PSO	PSO	PSO	*	PSO	PSO
F3	4.2247E-13	PSO	PSO	PSO	*	PSO	PSO
F4	5.09685E-19	PSO	PSO	PSO	*	PSO	PSO
F5	9.08241E-22	PSO	PSO	PSO	*	PSO	PSO
F6	7.49381E-37	PSO	PSO	PSO	*	PSO	PSO
F7	3.49218E-15	PSO	PSO	PSO	*	PSO	PSO
F8	3.2879E-33	GA, SA, POLY, PSO	POLY, PSO	POLY, PSO	*	PSO	GA, SA, POLY, PSO
F9	1.10236E-14	POLY, PSO	POLY, PSO	SA, CK	*	PSO	GA, POLY, PSO
F10	1.51277E-22	PSO	PSO	PSO	*	PSO	PSO
F11	9.35908E-16	PSO	PSO	PSO	*	PSO	PSO
F12	1.68892E-14	PSO	PSO	PSO	*	PSO	PSO
F13	1.62595E-19	PSO	PSO	PSO	*	PSO	PSO
F14	9.59387E-26	PSO	PSO	PSO	*	PSO	PSO
F15	1.39766E-26	SA, PSO	PSO	SA, PSO	*	PSO, SA	CK, SA, PSO
F16	4.45976E-18	PSO	PSO	PSO	*	PSO	PSO
F17	1.26316E-19	PSO	PSO	PSO	*	PSO	PSO
F18	9.84919E-28	SA, PSO	PSO	SA, PSO	*	PSO, SA	SA, PSO
F19	3.62808E-37	PSO	PSO	PSO	*	PSO	PSO
F20	1.55505E-10	PSO	PSO	PSO	*	PSO	PSO
F21	2.68003E-11	PSO	PSO	PSO	*	PSO	PSO
F22	1.30234E-57	GA, POLY, PSO	GA, POLY, PSO	PSO	*	*	GA, POLY, PSO
F23	2.8735E-16	PSO	PSO	PSO	*	PSO	PSO
F24	4.33902E-60	SA, PSO	PSO	SA, PSO	*	SA, PSO	SA, PSO
F25	1.47518E-34	PSO	PSO	PSO	*	PSO	PSO
F26	3.7071E-35	PSO	PSO	PSO	*	PSO	PSO
F27	3.70578E-41	PSO	PSO	CK, SA, PSO	*	CK, SA, PSO	CK, SA, POLY, PSO
F28	2.02948E-40	GA, POLY, PSO	GA, POLY, PSO	PSO	*	PSO	GA, POLY, PSO
F29	6.21067E-30	PSO	PSO	PSO	*	PSO	PSO
F30	1.0275E-27	SA, PSO	PSO	SA, PSO	*	SA, PSO	GA, CK, SA, POLY, PSO

5. SUMMARY OF RESULTS

HYPOTHESIS 1 (H1)

In this paper, the success of the proposed Hybrid model, CK, GA, SA, and a modified GA (polygamy induced) has been statistically compared. When compared over hypothesis 1 which is based on the mean function calls (speed of convergence), the

proposed hybrid algorithm displayed superiority over all other models. Next to the Hybridized model was the GA algorithm, thereafter the CK search algorithm. It was observed that the CK model outperformed the GA (with a statistically significant difference) in just 2 out of the 30 bench mark functions (Table 1(b)) (the F9 – ‘brent function’ and the F28- ‘Adjiman Function’) which are relatively complex functions in terms of their differentiability, modality, separability and modality. The PSO owes its performance to its stability problem and also, the amount of permitted epochs for evaluation used in most PSO implementations is usually high (approx. 2 million) [28,29,30] for each bench mark function as against 500 used in this research. There exists no significant difference in the speed of convergence (in cases where convergence occurred) between the SA, PSO and the polygamy induced GA. However, the polygamy induced GA performed better than the CK in two of the bench mark functions (F4-bird Function, F5- Bohachevsky 1 Function).

HYPOTHESIS 2 (H2)

The success of the proposed Hybrid model can also be seen from the mean global optimal value as show from the ANOVA analysis (Table 5). In addition to the convergence speed advantage, the proposed hybrid model has a significant better performance when compared with the other meta-heuristics with an impressive advantage over the traditional GA in (F8- ‘Branin RCOS-2 Function’, F9-‘Brent function’, F22- ‘Rosenbrock Modified Function’ F28-‘Adjiman Function’ and F30-‘Damavandi Function’). These functions are recognized for their complexity in terms of their differentiability, separability and modality. We can deduce from statistical results that the speed of convergence is a major advantage of our proposed model while its ability to consistently converge at the global optima within a minimal number of epochs is an added advantage. Figs. 4-8 show a box plot representation of the mean difference between test algorithms on the frequency with which the optimal solutions is found within the predefined number of epochs.

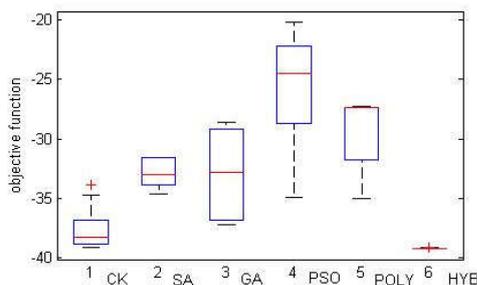


Figure 4 – ‘Branin RCOS-2 Function (F8)’ with global minima = -39.1956

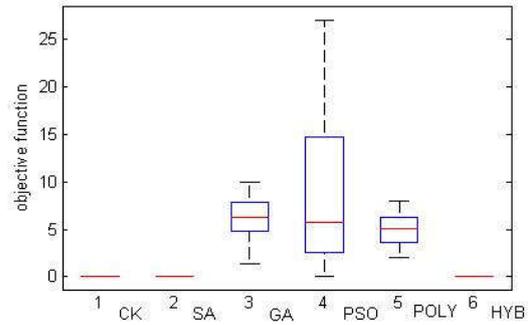


Figure 5 – ‘Brent function (F9)’ with global minima = 0

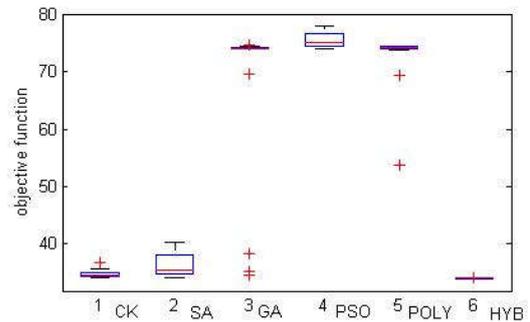


Figure 6 – ‘Rosenbrock Modified Function (F22)’ with global minima = 34.0412

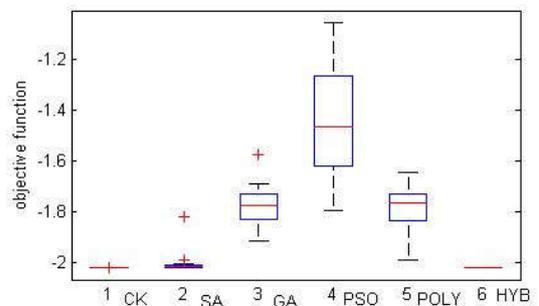


Figure 7 – ‘Adjiman Function (28)’ with global minima = -2.02181

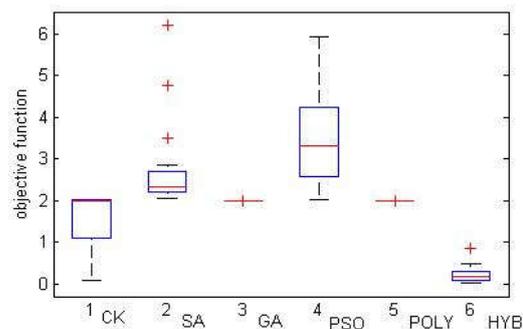


Figure 8 – ‘Damavandi Function (F30)’ with global minima = 0

6. CONCLUSION

This paper empirically compares the success of a hybrid algorithm with the CK, SA, GA, PSO and a modified GA in solving optimization problems. The framework systematically combines strengths of multiple meta-heuristics leveraging on the traditional GA mutation strategy. Empirical analysis revealed faster convergence to a global optima with minimal computational complexity.

The algorithm provides a platform for future research work on the scalability (max number of meta-heuristics) and also the efficiency of the algorithm when combined with other meta-heuristics apart from those used in this research such as ABC (artificial bee colony), DE (Differential Evolution), Ant Colony, etc. In addition, the success of such framework could be analyzed on multi-objective optimization problems.

ACKNOWLEDGMENTS

Research supported in part by South African National Research Foundation Grants (No. 112108 & 112142), South African National Research Foundation Incentive Grant (Grant No. 114911) and Tertiary Education Support Programme (TESP) of South African ESKOM.

CONFLICTS OF INTEREST

The authors declare that there are no conflicts of interest regarding the publication of this paper.

7. REFERENCES

- [1] A. Prügel-Bennett, "Benefits of a population: five mechanisms that advantage population-based algorithms," *IEEE Trans. Evol. Comput.*, Vol. 14, No. 4, pp. 500-517, 2010.
- [2] R. L. Haupt, S. E. Haupt, *Practical Genetic Algorithms*, John Wiley and Sons Inc. 2004.
- [3] M. Jamil and X.-S. Yang, "A literature survey of benchmark functions for global optimization problems," *Int. Journal of Mathematical Modelling and Numerical Optimisation*, Vol. 4, No. 2, pp. 150-194 2013.
- [4] D. H. Ackley, *A Connectionist Machine for Genetic Hill-Climbing*, Kluwer, 1987.
- [5] I. O. Bohachevsky, M. E. Johnson, M. L. Stein, "General Simulated Annealing for Function Optimization," *Technometrics*, Vol. 28, No. 3, pp. 209-217, 1986.
- [6] C. Muntenau, V. Lazarescu, "Global search using a new evolutionary framework: the adaptive reservoir genetic algorithm," *Complexity International*, vol. 5, 1998.
- [7] F. H. Branin Jr., "Widely convergent method of finding multiple solutions of simultaneous nonlinear equations," *IBM Journal of Research and Development*, vol. 16, no. 5, pp. 504-522, 1972.
- [8] A. Lavi, T. P. Vogel (eds), *Recent Advances in Optimization Techniques*, John Wiley & Sons, 1966.
- [9] M. M. Ali, C. Khompatraporn, Z. B. Zabinsky, "A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems," *Journal of Global Optimization*, vol. 31, pp. 635-672, 2005.
- [10] C. J. Chung, R. G. Reynolds, "CAEP: an evolution-based tool for real-valued function optimization using cultural algorithms," *International Journal on Artificial Intelligence Tool*, vol. 7, no. 3, pp. 239-291, 1998.
- [11] S. S. Rao, *Engineering Optimization: Theory and Practice*, John Wiley & Sons, 2009.
- [12] A. A. Goldstein, J. F. Price, "On descent from local minima," *Mathematics and Computation*, vol. 25, no. 115, pp. 569-574, 1971.
- [13] H. H. Rosenbrock, "An automatic method for finding the greatest or least value of a function," *Computer Journal*, vol. 3, no. 3, pp. 175-184, 1960.
- [14] S. K. Mishra, "Some new test functions for global optimization and performance of repulsive particle swarm method," <http://mp.ra.uni-muenchen.de/2718/>
- [15] S. K. Mishra, "Global optimization by differential evolution and particle swarm methods: evaluation on some benchmark functions," *Munich Research Papers in Economics*: <http://mp.ra.uni-muenchen.de/1005/>
- [16] E. P. Adorio, U. P. Dilman, "MVF – multivariate test function library in C for unconstrained global optimization methods," <http://www.geocities.ws/eadorio/mvf.pdf>
- [17] C. S. Adjiman, S. Sallwig, C. A. Flouda, A. Neumaier, "A global optimization method, aBB for general twice-differentiable NLPs-1, theoretical advances," *Computers Chemical Engineering*, vol. 22, no. 9, pp. 1137-1158, 1998.
- [18] N. Damavandi, S. Safavi-Naeini, "A hybrid evolutionary programming method for circuit optimization," *IEEE Transaction on Circuit and Systems I*, vol. 52, no. 5, pp. 902-910, 2005.
- [19] M. Negnevitsky, *Artificial Intelligence: A Guide to Intelligent Systems*, second edition, Pearson Education Limited, Edinburgh Gate, Harlow, 2005.
- [20] S. Kirkpatrick, C.D. Gelatt Jr., and M.P Vecchi, "Optimization by simulated annealing,"

- Science, Vol. 220, Issue 4598, pp. 671-680, 1983.
- [21] J. Kennedy and R.C. Eberhart, "Particle swarm optimization," *Proceedings of the IEEE International Conference on Neural Networks*, 1995.
- [22] X.-S. Yang, S. Deb, "Cuckoo search via Lévy flights," *Proceedings of the IEEE World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, India, 1999, pp. 210-214.
- [23] S.A. Kazarlis, et al., "Micro-genetic algorithms as generalized hill climbing operators for GA optimization," *IEEE Trans. Evol. Comp.*, Vol. 5, No. 3, pp. 204-217, 2001.
- [24] K. A. De Jong, *An Analysis of a Class of Genetic Adaptive Systems*, Ph.D. thesis, University of Michigan, 1975.
- [25] D. E. Goldberg and J. Richardson, "Genetic algorithm with sharing for multimodal function optimization," *Proceedings of the second International Conference on Genetic Algorithms and their Applications*, Cambridge, Massachusetts, USA, 1987, pp. 41-49.
- [26] I. J Eshelman and J. D. Schaffer, "Preventing premature convergence in genetic algorithms by preventing incest," In: R.Belew, L.B. Booker, (eds), *Proc. of the Fourth Int. Conf. on Genetic Algorithms*. Morgan Kaufmann, San Mateo, CA, 1991, pp. 115-122.
- [27] J. H. Holland, *Adaptation in Neural and Artificial Systems*, second edition, University of Michigan press, 1975.
- [28] P. Civicioglu, E. Besdok, "A conceptual comparison of the Cuckoo-search, particle Swarm optimization, differential evolution and artificial bee colony algorithms," *Artificial Intelligence Review*, Vol. 39, Issue 4, pp. 315-346, 2013. DOI 10.1007/s10462-011-9276-0.
- [29] N. Barton and T. Paixão, "Can quantitative and population genetics help us understand evolutionary computation?," *Proceedings of the 15th annual conference on Genetic and evolutionary computation GECCO '13*, July 6-10'2013, pp. 1573-1580.
- [30] B. Doerr, C. Doerr and F. Ebel, "From black-box complexity to designing new genetic algorithms", *Theor. Comput. Sci.*, Vol. 567, pp. 87-104, 2015.



Obadan Samuel, is a PhD student of University of South Africa (UNISA), in South Africa. In 2009 hereceived the MSc degree in Information systems at the European University Cyprus.

His research interests include: robotics, computer control systems, optimization and Machine Learning.



Zenghui Wang is currently a Professor at the University of South Africa (Unisa). After graduating with a Doctoral degree in 2007, Prof Wang worked as an Associate Professor in the Department of Automation at Shandong University of Science and Technology and a post-Doctoral Fellow at the French South African Technical Institute in Electronics at the Tshwane University of Technology (TUT). He has published more than 70 papers, including 33 in accredited journals. His research Interest are in Automatic Control: Adaptive Control and Predictive Control.