A PROJECT REPORT

ON

EXPECTATION ALGORITHM FOR UNCONSTRAINED OPTIMIZATION

REPORT SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY IN

(Mechanical Engineering)

SUBMITTED BY

Meet Patel

Amit Sehgal

Aishwary Jagetia

Under the Guidance of

Dr. Anand J Kulkarni

॥वसुधैव कुटुम्बकम्॥

SYMBIOSIS INSTITUTE OF TECHNOLOGY

(A CONSTITUENT OF SYMBIOSIS INTERNATIONAL UNIVERSITY)

Pune- 412115

YEAR 2017

A PROJECT REPORT

ON

EXPECTATION ALGORITHM FOR UNCONSTRAINED OPTIMIZATION

REPORT SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY IN

(Mechanical Engineering)

SUBMITTED BY

Meet Patel

Amit Sehgal

Aishwary Jagetia

Under the Guidance of

Dr. Anand J Kulkarni

SYMBIOSIS INSTITUTE OF TECHNOLOGY

(A CONSTITUENT OF SYMBIOSIS INTERNATIONAL UNIVERSITY)

Pune- 412115

YEAR 2017

CERTIFICATE

This is to certify that the project work entitled "**Expectation Algorithm for Unconstrained Optimization**" submitted to the Symbiosis Institute of Technology, Pune for the final year project in the VIII semester of mechanical engineering is based on our original work carried out under the guidance of Dr. Anand J Kulkarni and Prof. Apoorva Shastri. The report has not been submitted elsewhere for award of any degree.

The material borrowed from other source and incorporated in the report has been duly acknowledged and/or referenced.

We understand that we would be held responsible and accountable for plagiarism, if any, detected later on.

Date: 19-04-2017

Place: Symbiosis Institute of Technology

Meet Patel

Amit Sehgal

Aishwary Jagetia

(STUDENTS)

Prof. Apoorva Shastri Associate Project Guide Department of Electronics and Telecommunication	Dr. Anand J Kulkarni Project Guide and H.O.D. Department of Mechanical Engineering	Dr. T. P. Singh Director Symbiosis Institute of Technology

ACKNOWLEDGEMENTS

We would like to thank our guide, **Dr. Anand J Kulkarni** for his encouragement and associate guide **Prof. Apoorva Shastri** for her valuable support throughout the project without which this project would not have been possible We would also like to thank **Prof. Mandar Sadanand Sapre** for his guidance and contribution during the project work. We are thankful to **Dr. Tai Kang** from Nanyang Technological University and to all the faculty members of mechanical engineering department of Symbiosis Institute of Technology, for their expertise knowledge and guidance throughout our study, which has provided us with valuable insight in number of areas. Finally, we thank all the people who provided us with the facilities being required and conductive conditions for our project.

Meet Patel

Amit Sehgal

Aishwary Jagetia

Department of Mechanical Engineering,

Symbiosis Institute of Technology,

Pune-412115

Date: 19th April 2017

ABSTRACT

This is a new socio-inspired technique referred to as Expectation Algorithm (ExA). It is inspired by the self-interested and competitive behavior of individuals which makes them improve. Expectation Algorithm demonstrated superior performance as compared to some other well-known techniques in solving unconstrained test problems. Wilcoxon signed-rank test is applied to verify the performance of Expectation Algorithm in solving optimization problems. The results are compared with eight well-known and some recently proposed optimization algorithms (PSO, CLPSO, CMAES, ABC, JDE, SADE, BSA and IA). A total of 48 unconstrained benchmark problems are used to test the performance of Expectation Algorithm up to 30 dimensions. The results from this study highlighted that the Expectation Algorithm outperforms the other algorithms in terms of number function evaluations and computational time.

Contents

Figures

Tables

Chapter-1

INTRODUCTION

1.1 What is Optimization?

The extreme states (i.e. the maximum and the minimum states) of many quantities arising from mathematical models, natural/ physical phenomena or human activities/artifacts are usually of interest. Some abstract mathematical problems can be analyzed/ solved by minimizing a specific quantity derived from the problem.

In the nature, for example, the stable equilibrium states of many physical systems are usually the states at which total potential energy is a minimum. For almost all human activities and creations, there is a desire to do or be the best in some sense. To set record in a race, for example, the aim is to be the fastest (shortest time); in the conduct of a retail business, the desire may be to maximize profits; in the construction of a building, the desire may be to minimize costs; in the planning of a project schedule, the aim may be to minimize project time; in the design of a power generator turbine, the objective may be to maximize efficiency.

Hence the concept of minimization and maximization has great significance in both human affairs and the laws of nature. Optimization therefore refers to a positive and intrinsically human concept of minimization or maximization to achieve the best or most favorable outcome from a given situation.

A fundamental approach is adopted to bring out the mathematical/geometrical basis behind the optimization problems and methods, to facilitate understanding and a wider application to a diverse range of problem types.

1.2 Types of optimization

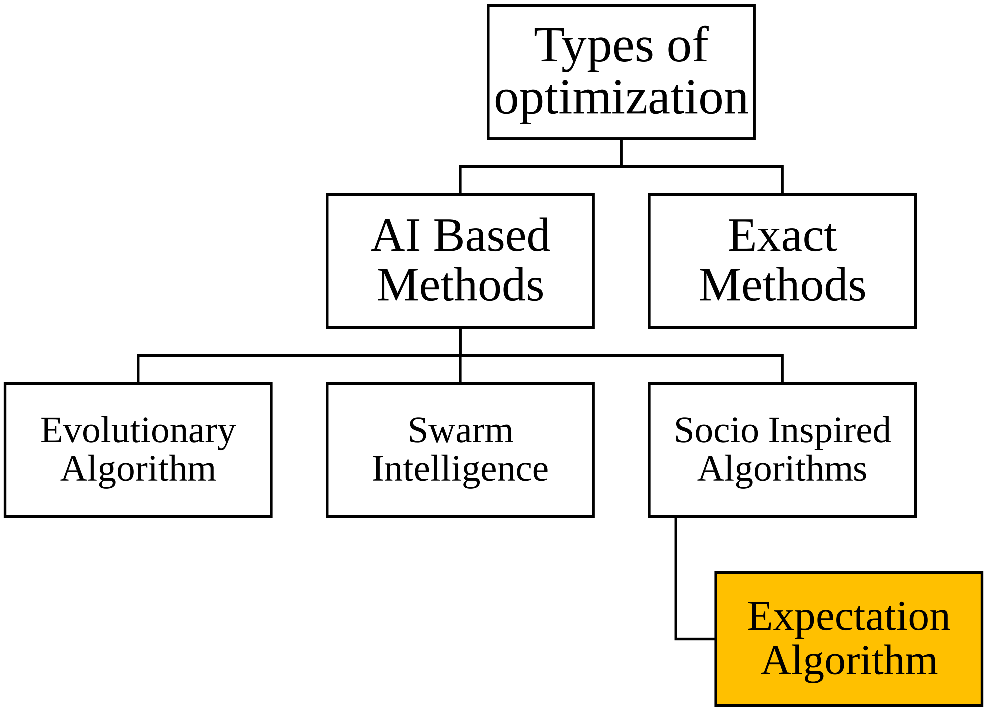

Figure 1Where exactly Expectation Algorithm lies

Exact optimization methods that guarantee finding an optimal solution and AI based optimization methods where we have no guarantee that an optimal solution is found.

1.2.1 AI Based Method

a. Evolutionary Algorithm

In artificial intelligence, an **evolutionary algorithm** (**EA**) is a subset of evolutionary computation, a generic population-based metaheuristic optimization algorithm. An EA uses mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection. Candidate solutions to the optimization problem play the role of individuals in a population, and the fitness function determines the quality of the solutions (see also loss function). Evolution of the population then takes place after the repeated application of the above operators.

Evolutionary algorithms often perform well approximating solutions to all types of problems because they ideally do not make any assumption about the underlying fitness landscape; this generality is shown by successes in fields as diverse as engineering, art, biology, economics, marketing, genetics, operations research, robotics, social sciences, physics, politics and chemistry.

Techniques from evolutionary algorithms applied to the modelling of biological evolution are generally limited to explorations of micro-evolutionary processes and planning models based upon cellular processes.

In most real applications of EAs, computational complexity is a prohibiting factor. In fact, this computational complexity is due to fitness function evaluation. Fitness approximation is one of the solutions to overcome this difficulty. However, seemingly simple EA can solve often complex problems; therefore, there may be no direct link between algorithm complexity and problem complexity.

b. Swarm Intelligence

Swarm intelligence (**SI**) is the collective behaviour of decentralized, self-organized systems, natural or artificial. The concept is employed in work on artificial intelligence. The expression was introduced by Gerardo Beni and Jing Wang in 1989, in the context of cellular robotic systems. SI systems consist typically of a population of simple agents or boids interacting locally with one another and with their environment. The inspiration often comes from nature, especially biological systems. The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, local, and to a certain degree random, interactions between such agents lead to the emergence of "intelligent" global behaviour, unknown to the individual agents. Examples in natural systems of SI include ant colonies, bird flocking, animal herding, bacterial growth, fish schooling and microbial intelligence.

The application of swarm principles to robots is called swarm robotics, while 'swarm intelligence' refers to the more general set of algorithms. 'Swarm prediction' has been used in the context of forecasting problems.

c. Socio Inspired Algorithms

Over the last two decades, metaheuristic optimization techniques have become increasingly popular and essential in applied mathematics. Optimization algorithms are functioning as to find the best values for system variables under various conditions. Some well-known metaheuristics such as particle swarm optimization (PSO) , genetic algorithm (GA) , ant colony optimization (ACO) are fairly well known, and they are applied in various fields. With regard to some drawbacks of classical optimization strategies as well as to achieve simplicity, flexibility and derivation-free mechanism, several metaheuristics have been designed.

1.2.2 Exact Methods

a. Nonlinear Optimization

1. Single variable unconstrained optimization, which can be evaluated by
   - i) Initial Bracketing of Minimum Point
     Further this can be done by two methods
     1. Equal Interval Search
     2. Variable Interval Search
   - ii) Approximation of Minimum Point
     Further it can be solved by
     1. Golden Section Method
     2. Polynomial approximation
2. Multi Variable Unconstrained Optimization
   Further evaluation methods,
   - i) Steepest Descent Method
   - ii) Conjugate Gradient Method
   - iii) Hooke and Jeeves Method
3. Multi Variable Constrained Optimization
   Further evaluation methods,
   - i) Exterior Penalty Function Method
   - ii) Generalized Reduced Gradient Method

However, we will focus only on Multi variable unconstrained optimization performed by steepest descent method only.

Chapter-2

STEEPEST DESCENT

2.1 The Method

Gradient descent is a first-order iterative optimization algorithm. To find a local minimum of a function using gradient descent, one takes steps proportional to the *negative* of the gradient (or of the approximate gradient) of the function at the current point. If instead one takes steps proportional to the *positive* of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent.

Gradient descent is also known as steepest descent, or the method of steepest descent. Gradient descent should not be confused with the method of steepest descent for approximating integrals.

Gradient descent is based on the observation that if the function $\boldsymbol{F}(\boldsymbol{x})$ is defined and differentiable in a neighborhood of a point $\boldsymbol{a}$, then $\boldsymbol{F}(\boldsymbol{x})$ decreases *fastest* if one goes from $\boldsymbol{a}$ in the direction of the negative gradient of $\boldsymbol{F}$ at $\boldsymbol{a}$, $-\boldsymbol{\nabla F}(\boldsymbol{a})$. It follows that if, $\boldsymbol{a}^{\boldsymbol{n}+\mathbf{1}} = \boldsymbol{a}^{\boldsymbol{n}} - \boldsymbol{\gamma}\boldsymbol{\nabla F}(\boldsymbol{a}^{\boldsymbol{n}})$, for γ small enough then $F(a^n) \geq F(a^{n+1})$. In other words, the term $\boldsymbol{\gamma}\boldsymbol{\nabla F}(\boldsymbol{a})$ is subtracted from $\boldsymbol{a}$ because we want to move against the gradient, namely down toward the minimum.

With this observation in mind, one starts with a guess $\boldsymbol{x}_{\mathbf{0}}$ for a local minimum of $\boldsymbol{F}$ and considers the sequence $\boldsymbol{x}_{\mathbf{0}}, \boldsymbol{x}_{\mathbf{1}}, \boldsymbol{x}_{\mathbf{2}}$ such that $\boldsymbol{x}_{\boldsymbol{n}+\mathbf{1}} = \boldsymbol{x}_{\boldsymbol{n}} - \boldsymbol{\gamma}_{\boldsymbol{n}}\boldsymbol{\nabla F}(\boldsymbol{x}_{\boldsymbol{n}})\,, \boldsymbol{n} \geq \mathbf{0}$.

We have,

$\boldsymbol{F}(\boldsymbol{x}_{\mathbf{0}}) \geq \boldsymbol{F}(\boldsymbol{x}_{\mathbf{1}}) \geq \boldsymbol{F}(\boldsymbol{x}_{\mathbf{2}}) \geq . . . ,$

So, the sequence $(\boldsymbol{x}_{\boldsymbol{n}})$ converges to the desired local minimum.

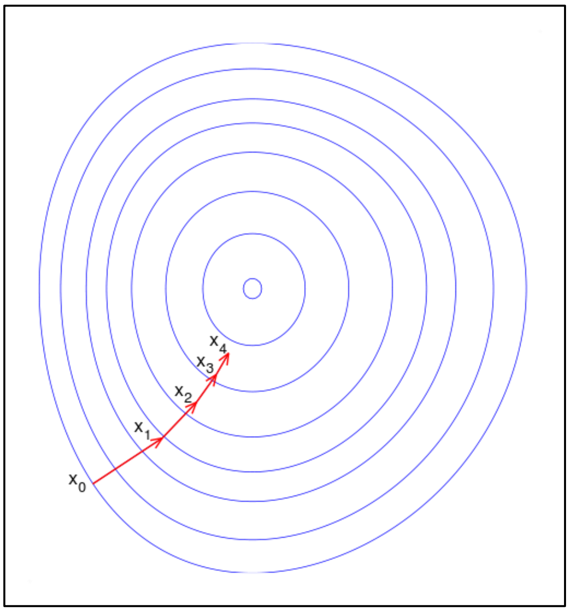

Figure 2 Steepest Descent

Here ***F*** is assumed to be defined on the plane, and that its graph has a bowl shape. The blue curves are the contour lines, that is, the regions on which the value of ***F*** is constant. A red arrow originating at a point shows the direction of the negative gradient at that point. Note that the (negative) gradient at a point is orthogonal to the contour line going through that point. We see that gradient *descent* leads us to the bottom of the bowl, that is, to the point where the value of the function ***F*** is minimal.

2.2 Limitations

Gradient descent has problems with pathological functions such as the Rosenbrock function.

$$f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^{\,2})^2$$

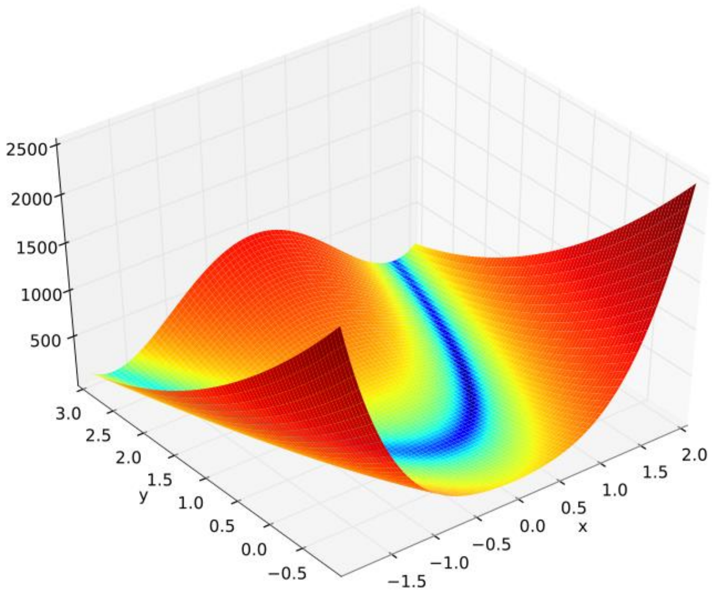

Figure 3 Rosenbrock function

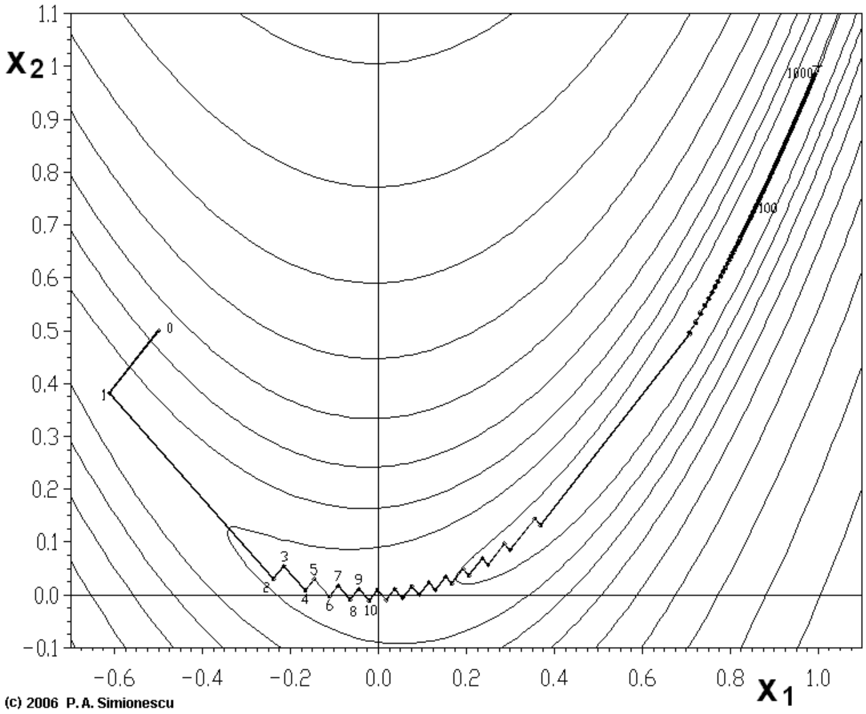

Figure 4 Using Steepest Descent to solve Rosenbrock Function

Looking at the above plot of Steepest Descent trying to achieve minima in Rosenbrock Function, we can observe that the Rosenbrock function has a narrow-curved valley which contains the minimum. Because of the curved flat valley, the optimization is zig-zagging slowly with small step sizes towards the minimum but is not able to achieve that minima because of the fact that in Rosenbrock function, the bottom of the valley is very flat.

From the above example we can conclude that,

- Gradient descent is relatively slow close to the minimum: technically, its asymptotic rate of convergence is inferior to many other methods. For poorly conditioned convex problems, gradient descent increasingly 'zigzags' as the gradients point nearly orthogonally to the shortest direction to a minimum point.
- For non-differentiable functions, gradient methods are ill-defined.

Chapter-3

MOTIVATION

3.1 Inspiration for Expectation Algorithm

- The motivation for this algorithm is based on "no free lunch" concept which states that no algorithm is best suited for solving problems, there are solutions which are best for some problems or most of the problems but for some, the solution might not be satisfactory. To overcome this situation, new algorithms are introduced so that a set of problems are solved by one or more than one algorithms.

- Socio Inspired: It is the ability of the algorithm to learn from other candidates which are present in the same global interval providing solution better than rest of the candidates, which allows every candidate to reach out to the global optimum.

The motivation of introducing a new socially inspired algorithm referred to as Expectation algorithm was to overcome “no free lunch” concept. In our day to day life we as people/candidates are required to perform certain set of actions which we can improve upon by having experience or by learning from other people/candidates around us. However, the experience takes a longer duration to attain certain level of excellence and therefore we tend to learn faster from the people/candidates around us by expecting or predicting from their ongoing actions.

Chapter-4

EXPECTATION ALGORITHM

4.1 Introduction

In the society, every person has some expectation about the behavior or performance of another person. The degree to which something expected is to be true is shown using Expectation algorithm. In the proposed Expectation algorithm (ExA) every variable expects a value of every other variable. By applying steepest descent algorithm on these expected values, every variable tries to move towards the most effective degree of expectation. It is the ability of the algorithm that variable expects value of every other variable which are present in the same global interval providing solution better than rest of the variables, which allows every variable to reach out to the global optimum. Then it shrinks the sampling interval in the close neighborhood of the optimum (converged) value.

The expected algorithm treats every variable as one candidate and behavior /opinion of each candidate is considered independently of others. Firstly, the variables are defined for a specific interval which are known as bounds (i.e. upper bound and lower bound). The variables defined have their own upper and lower bounds which are a subset of the global interval. Each variable will try to move towards the function minimum value by steepest descent method (directional behavior).

In this process, there will be conflict of directional behaviors, but now the expectation of other variables comes into play. Every variable will look at other variables and will try to learn from them, given a condition where there are 3 variables, one or more out of these variables will follow the other variable which gives the least function value. As mentioned above the variables have their own sub bounds, these sub bounds will shift according to the function minimum value, this is so, because if we don’t shift the bounds, every variable will have to repeat the process and might never get to the function minimum.

Further, once after 1st iteration or the first trial, all the variables will come close to the function optimum. This process will get repeated again and again till all the variables attain the function minimum value

<<<<<<<<<<< Procedure and Illustration – Confidential >>>>>>>>>

4.2 Flow Chart

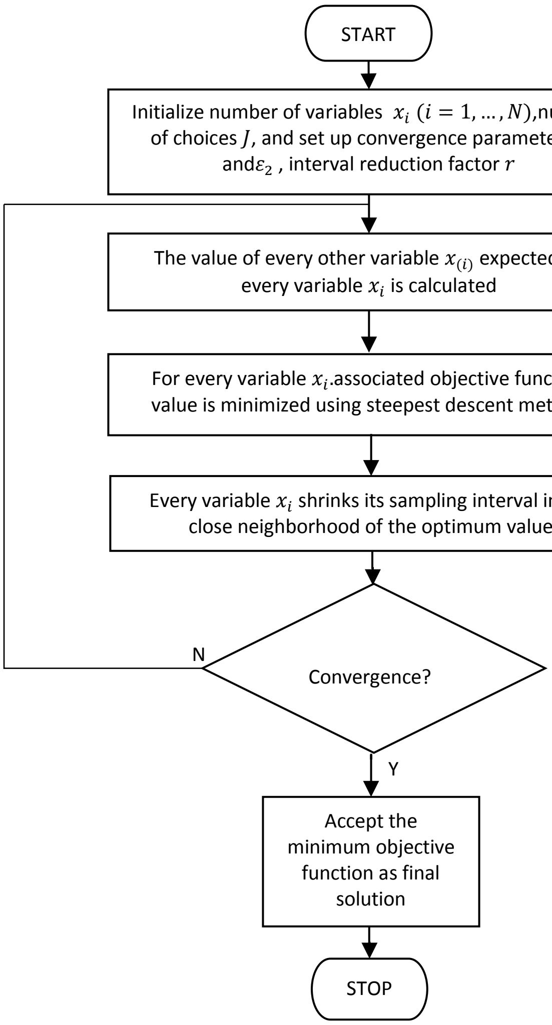

Figure 5 Expectation Algorithm - Flow chart

4.3 Plot for Rastrigin function

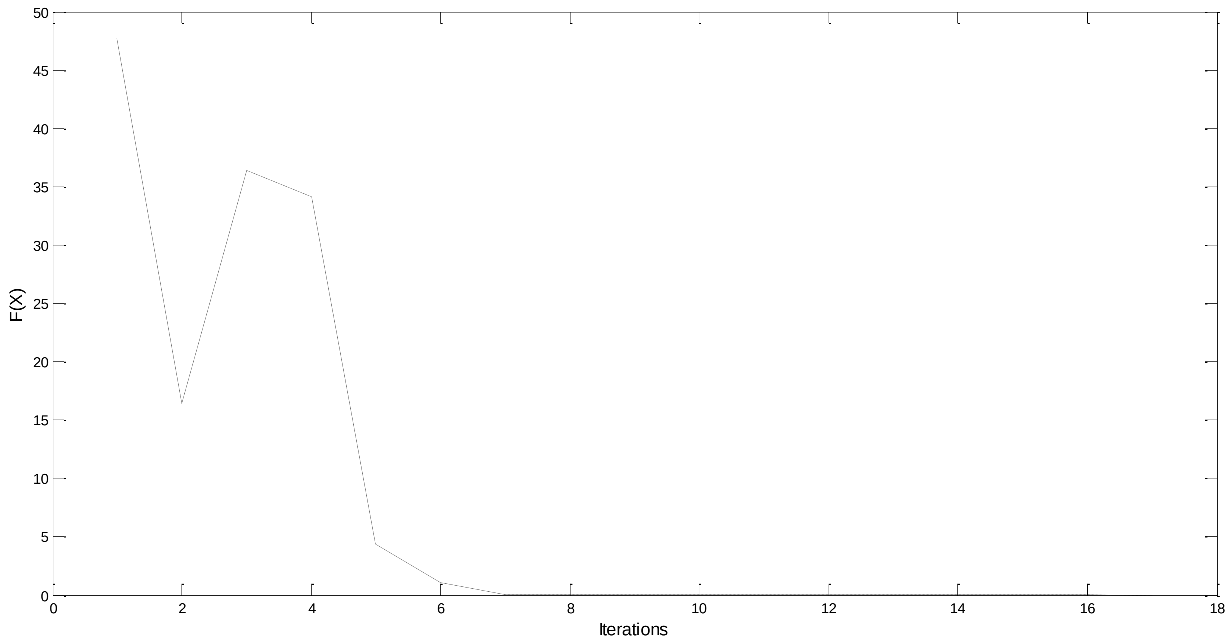

Figure 6 Expectation Algorithm solving Rastrigin Function

This plot depicts the variation of F(X) value of Rastrigin function with number of iterations for 3 variables. For every variable x_i (i $= 1,2,3$) , randomly generates 5 number of choices within the given interval $[-5.12, 5.12]$ for optimization of Rastrigin function. The optimum value of variables from this iteration will be initial value for succeeding iteration. The F(X) value of Rastrigin is calculated after each iteration by substituting the optimum value of variables from the iteration. The algorithm is assumed to have converged if all the stopping criterions are satisfied for successive considerable number of iterations and accept any of the solution as final solution $F(\boldsymbol{X})$,else the optimum value of each variables is used as initial value for succeeding iteration.

$$\left\| min\left(\left(F(\boldsymbol{X})^i\right)^n\right) - min\left(\left(F(\boldsymbol{X})^i\right)^{n-1}\right) \right\| \leq 1e-16$$

The program gets terminated after 16th iteration as shown in plot as the above condition is satisfied.

Chapter-5

STATISTICAL SOLUTIONS

5.1 Performance Analysis

The Performance Analysis Results shows the Mean solutions, Standard Deviation of mean values, Best value, and average of Runtime in seconds after running each test functions for unconstrained optimization 30 times. The Statistical Results consists of optimum solution of test function by PSO, CMAES, ABC, CLPSO, SADE, BSA IA and proposed ExA as shown below.

The Standard Deviation of mean values represents the Deviation in optimum answer obtained when it ran 30 times and it signifies the accuracy of an algorithm i.e. tendency to get same answer as many times the programmed algorithm is ran, lower the Standard Deviation higher is the accuracy of the algorithm. The Best Value shows the most optimum value obtained by that algorithm.

The runtime mentioned in the below table is average runtime of time taken by algorithm to obtain the optimum value in MatlabR2013a when ran for 30 times.

Table 1 The benchmark problems used in Test 1 (Dim dimension, low and up limitations of search space, U unimodal, M multimodal, S separable, N nonseparable)

Problem	Name	Type	Low	Up	Dimensio
F1	Foxholes	MS	−65.536	65.536	2
F2	Goldstein-Price	MN	−2	2	2
F3	Penalized	MN	−50	50	30
F4	Penalized2	MN	−50	50	30
F5	Ackley	MN	−32	32	30
F6	Beale	UN	−4.5	4.5	5
F7	Bohachevsky1	MS	−100	100	2
F8	Bohachevsky2	MN	−100	100	2
F9	Bohachevsky3	MN	−100	100	2
F10	Booth	MS	−10	10	2
F11	Branin	MS	−5	10	2
F12	Colville	UN	−10	10	4
F13	Dixon-Price	UN	−10	10	30
F14	Easom	UN	−100	100	2
F15	Fletcher	MN	−3.1416	3.1416	2
F16	Fletcher	MN	−3.1416	3.1416	5
F17	Fletcher	MN	−3.1416	3.1416	10
F18	Griewank	MN	−600	600	30
F19	Hartman3	MN	0	1	3
F20	Hartman6	MN	0	1	6
F21	Kowalik	MN	−5	5	4
F22	Langermann2	MN	0	10	2
F23	Langermann5	MN	0	10	5
F24	Langermann10	MN	0	10	10
F25	Matyas	UN	−10	10	2
F26	Michalewics2	MS	0	3.1416	2
F27	Michalewics5	MS	0	3.1416	5
F28	Michalewics10	MS	0	3.1416	10
F29	Perm	MN	−4	4	4
F30	Powell	UN	−4	5	24
F31	Powersum	MN	0	4	4
F32	Quartic	US	−1.28	1.28	30
F33	Rastrigin	MS	−5.12	5.12	30
F34	Rosenbrock	UN	−30	30	30
F35	Schaffer	MN	−100	100	2
F36	Schwefel	MS	−500	500	30
F37	Schwefel_1_2	UN	−100	100	30
F38	Schwefel_2_22	UN	−10	10	30
F39	Shekel10	MN	0	10	4
F40	Shekel5	MN	0	10	4
F41	Shekel7	MN	0	10	4
F42	Shubert	MN	−10	10	2
F43	Six-hump camelback	MN	−5	5	2
F44	Sphere2	US	−100	100	30
F45	Step2	US	−100	100	30
F46	Stepint	US	−5.12	5.12	5
F47	Sumsquares	US	−10	10	30
F48	Trid6	UN	−36	36	6
F49	Trid10	UN	−100	100	10
F50	Zakharov	UN	−5	10	10

Table 2 Statistical solutions obtained by PSO, CMAES, ABC, CLPSO, SADE, BSA, IA and proposed ExA in Test 1 (Mean = Mean solution; Std. Dev. = Standard-deviation of mean solution; Best = Best solution; Runtime = Mean runtime in seconds)

Problem	Statistics	PSO2011	CMAES	ABC	JDE	CLPSO	SADE	BSA	IA	ExA
F1	Mean	1.3316029264876300	10.0748846367972000	0.9980038377944500	1.0641405484285200	1.8209961275956800	0.9980038377944500	0.9980038377944500	0.9980038690000000	0.9980947437164300
	Std. Dev.	0.9455237994690700	8.0277365400340800	0.0000000000000001	0.3622456829347420	1.6979175079427900	0.0000000000000000	0.0000000000000000	0.0000000000000035	0.0000000000000000
	Best	0.9980038377944500	0.9980038377944500	0.9980038377944500	0.9980038377944500	0.9980038377944500	0.9980038377944500	0.9980038377944500	0.9980038685998520	0.9980947437164300
	Runtime	72.527	44.788	64.976	51.101	61.650	66.633	38.125	43.535	48.779
F2	Mean	2.9999999999999200	21.8999999999995000	3.0000000465423000	2.9999999999999200	3.0000000000000700	2.9999999999999200	2.9999999999999200	3.0240147900000000	3.0000394229402100
	Std. Dev.	0.0000000000000013	32.6088098948516000	0.0000002350442161	0.0000000000000013	0.0000000000007941	0.0000000000000020	0.0000000000000011	0.0787814840000000	0.0001247078365269
	Best	2.9999999999999200	2.9999999999999200	2.9999999999999200	2.9999999999999200	2.9999999999999200	2.9999999999999200	2.9999999999999200	3.0029461118668700	3.0000220549616100
	Runtime	17.892	24.361	16.624	7.224	24.784	28.699	7.692	41.343	44.151
F3	Mean	0.1278728062391630	0.0241892995662904	0.0000000000000004	0.0034556340083499	0.0000000000000000	0.0034556340083499	0.0000000000000000	0.3536752140000000	
	Std. Dev.	0.2772792346028400	0.0802240262581864	0.0000000000000001	0.0189272869685522	0.0000000000000000	0.0189272869685522	0.0000000000000000	1.4205454130000000	
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000003	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0014898619035614	
	Runtime	139.555	5.851	84.416	9.492	38.484	15.992	18.922	34.494	
F4	Mean	0.0043949463343535	0.0003662455278628	0.0000000000000004	0.0007324910557256	0.0000000000000000	0.0440448539086004	0.0000000000000000	0.0179485820000000	
	Std. Dev.	0.0054747064090174	0.0020060093719584	0.0000000000000001	0.0027875840585535	0.0000000000000000	0.2227372747439610	0.0000000000000000	0.0526650620000000	
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000003	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000165491	
	Runtime	126.507	6.158	113.937	14.367	48.667	33.019	24.309	322.808	
F5	Mean	1.5214322973725000	11.7040011684582000	0.0000000000000340	0.0811017056422860	0.1863456353861950	0.7915368220335460	0.0000000000000105	0.0000000000000009	0.0000000000056986
	Std. Dev.	0.6617570384662600	9.7201961540865200	0.0000000000000035	0.3176012689149320	0.4389839299322230	0.7561593402959740	0.0000000000000034	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000080	0.0000000000000080	0.0000000000000293	0.0000000000000044	0.0000000000000080	0.0000000000000044	0.0000000000000080	0.0000000000000009	0.0000000000056986
	Runtime	63.039	3.144	23.293	11.016	45.734	40.914	14.396	49.458	190.382
F6	Mean	0.0000000041922968	0.2540232169641050	0.0000000000000028	0.0000000000000000	0.0000444354499943	0.0000000000000000	0.0000000000000000	0.0082236060000000	0.0000000119623730
	Std. Dev.	0.0000000139615552	0.3653844307786430	0.0000000000000030	0.0000000000000000	0.0001015919507724	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000496002626
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000005	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0082236059357692	0.0000000001024682
	Runtime	32.409	4.455	22.367	1.279	125.839	4.544	0.962	50.246	61.688
F7	Mean	0.0000000000000000	0.0622354533647150	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std. Dev.	0.0000000000000000	0.1345061339146580	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Runtime	16.956	6.845	1.832	1.141	2.926	4.409	0.825	38.506	1.241
F8	Mean	0.0000000000000000	0.0072771062590204	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000001819

	Std. Dev.	0.0000000000000000	0.0398583525142753	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000001819
	Runtime	17.039	2.174	1.804	1.139	2.891	4.417	0.824	39.023	1.092
F9	Mean	0.0000000000000000	0.0001048363065820	0.0000000000000006	0.0000000000000000	0.0000193464326398	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000001819
	Std. Dev.	0.0000000000000000	0.0005742120996051	0.0000000000000003	0.0000000000000000	0.0000846531630676	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000001
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000001	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000001815
	Runtime	17.136	2.127	21.713	1.129	33.307	4.303	0.829	40.896	0.920
F10	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0006005122443674	0.0000000000000000	0.0000000000000000	0.8346587090000000	0.0000000011373191
	Std. Dev.	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0029861918862801	0.0000000000000000	0.0000000000000000	0.0000000000000005	0.0000000065277288
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.8346587086917530	0.0000000000008471
	Runtime	17.072	1.375	22.395	1.099	28.508	4.371	0.790	39.978	331.676
F11	Mean	0.3978873577297380	0.6372170283279430	0.3978873577297380	0.3978873577297380	0.3978873577297390	0.3978873577297380	0.3978873577297380	0.4156431270000000	0.3978874405750270
	Std. Dev.	0.0000000000000000	0.7302632173480510	0.0000000000000000	0.0000000000000000	0.0000000000000049	0.0000000000000000	0.0000000000000000	0.0406451050000000	0.0000007990241356
	Best	0.3978873577297380	0.3978873577297380	0.3978873577297380	0.3978873577297380	0.3978873577297380	0.3978873577297380	0.3978873577297380	0.4012748152492080	0.3978873669283590
	Runtime	17.049	24.643	10.941	6.814	17.283	27.981	5.450	40.099	30.026
F12	Mean	0.0000000000000000	0.0000000000000000	0.0715675060725970	0.0000000000000000	0.1593872502094070	0.0000000000000000	0.0000000000000000	0.0014898620000000	0.0000000000000000
	Std. Dev.	0.0000000000000000	0.0000000000000000	0.0579425013417103	0.0000000000000000	0.6678482786713720	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0013425253994745	0.0000000000000000	0.0000094069599934	0.0000000000000000	0.0000000000000000	0.0082029783984983	0.0000000000000000
	Runtime	44.065	1.548	21.487	1.251	166.965	4.405	2.460	48.067	133.191
F13	Mean	0.6666666666666750	0.6666666666666670	0.0000000000000038	0.6666666666666670	0.0023282133668190	0.6666666666666670	0.6444444444444440	0.2528116640000000	0.2707841542431270
	Std. Dev.	0.0000000000000022	0.0000000000000000	0.0000000000000012	0.0000000000000002	0.0051792840882291	0.0000000000000000	0.1217161238900370	0.0000000006509080	0.0374228734316504
	Best	0.6666666666666720	0.6666666666666670	0.0000000000000021	0.6666666666666670	0.0000120708732167	0.6666666666666670	0.0000000000000000	0.2528116633611470	0.1429291028181160
	Runtime	167.094	3.719	37.604	18.689	216.261	47.833	21.192	67.463	37.071
F14	Mean	-1.0000000000000000	-0.1000000000000000	-1.0000000000000000	-1.0000000000000000	-1.0000000000000000	-1.0000000000000000	-1.0000000000000000	-0.9997989620000000	-0.0000000002894566
	Std. Dev.	0.0000000000000000	0.3051285766293650	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000167151	0.0000000000000000
	Best	-1.0000000000000000	-1.0000000000000000	-1.0000000000000000	-1.0000000000000000	-1.0000000000000000	-1.0000000000000000	-1.0000000000000000	-0.9997989624626810	-0.0000000002894566
	Runtime	16.633	3.606	13.629	6.918	16.910	28.739	5.451	39.685	0.050
F15	Mean	0.0000000000000000	1028.3930784026900000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std. Dev.	0.0000000000000000	1298.1521820113500000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Runtime	27.859	15.541	40.030	2.852	4.030	6.020	2.067	38.867	3.691
F16	Mean	48.7465164446927000	1680.3460230073400000	0.0218688498331872	0.9443728655432830	81.7751618148164000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std. Dev.	88.8658510972991000	2447.7484859066000000	0.0418409568792831	2.8815514827061600	379.9241117377270000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000016	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Runtime	95.352	11.947	44.572	4.719	162.941	5.763	7.781	48.262	10.691

F17	Mean	918.9518492782850000	12340.2283326398000000	11.0681496253548000	713.7226974626920000	0.8530843976878610	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std. Dev.	1652.4810858411400000	22367.1698875802000000	9.8810950146557100	1710.071307430120000	2.9208253191698800	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.3274654777056860	0.0000000000000000	0.0016957837829822	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Runtime	271.222	7.631	43.329	16.105	268.894	168.310	33.044	69.060	17.691
F18	Mean	0.0068943694819713	0.0011498935321349	0.0000000000000000	0.0048193578543185	0.0000000000000000	0.0226359326967139	0.0004930693556077	0.0000000000000000	0.0000000000000000
	Std. Dev.	0.0080565201649587	0.0036449413521107	0.0000000000000001	0.0133238235582874	0.0000000000000000	0.0283874287215679	0.0018764355751644	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Runtime	73.895	2.647	19.073	6.914	14.864	25.858	5.753	2.717	4049.784
F19	Mean	-3.8627821478207500	-3.7243887744664700	-3.8627821478207500	-3.8627821478207500	-3.8627821478207500	-3.8627821478207500	-3.8627821478207500	-3.8596352620000000	-3.8627821386763200
	Std. Dev.	0.0000000000000027	0.5407823545193820	0.0000000000000024	0.0000000000000027	0.0000000000000027	0.0000000000000027	0.0000000000000027	0.0033967610000000	0.0000000099856225
	Best	-3.8627821478207600	-3.8627821478207600	-3.8627821478207600	-3.8627821478207600	-3.8627821478207600	-3.8627821478207600	-3.8627821478207600	-3.8613076574052300	-3.8627821454352600
	Runtime	19.280	21.881	12.613	7.509	17.504	24.804	6.009	46.167	132.605
F20	Mean	-3.3180320675402500	-3.2942534432762600	-3.3219951715842400	-3.2982165473202600	-3.3219951715842400	-3.3140689634962500	-3.3219951715842400	-2.5710247593206100	-3.3223439238258300
	Std. Dev.	0.0217068148263721	0.0511458075926848	0.0000000000000014	0.0483702518391572	0.0000000000000013	0.0301641516823498	0.0000000000000013	0.0000000000000009	0.0266647307938119
	Best	-3.3219951715842400	-3.3219951715842400	-3.3219951715842400	-3.3219951715842400	-3.3219951715842400	-3.3219951715842400	-3.3219951715842400	-2.5710247593206100	-3.3223608131963600
	Runtime	26.209	7.333	13.562	8.008	20.099	33.719	6.822	59.083	512.307
F21	Mean	0.0003074859878056	0.0064830287538208	0.0004414866359626	0.0003685318137604	0.0003100479704151	0.0003074859878056	0.0003074859878056	0.0016993410000000	0.0003443694407639
	Std. Dev.	0.0000000000000000	0.0148565973286009	0.0000568392289725	0.0002323173367683	0.0000059843325073	0.0000000000000000	0.0000000000000000	0.0000013058400000	0.0000021998179893
	Best	0.0003074859878056	0.0003074859878056	0.0003230956007045	0.0003074859878056	0.0003074859941292	0.0003074859878056	0.0003074859878056	0.0016989914552560	0.0003345696732740
	Runtime	84.471	13.864	20.255	7.806	156.095	45.443	11.722	48.920	37.422
F22	Mean	-1.0809384421344400	-0.7323679641701760	-1.0809384421344400	-1.0764280762657400	-1.0202940450426400	-1.0809384421344400	-1.0809384421344400	-1.4315374190000000	-2.0000000000000000
	Std. Dev.	0.0000000000000006	0.4136688304155380	0.0000000000000008	0.0247042912888477	0.1190811583120530	0.0000000000000005	0.0000000000000005	0.0000000000000009	0.0000000000000000
	Best	-1.0809384421344400	-1.0809384421344400	-1.0809384421344400	-1.0809384421344400	-1.0809384421344400	-1.0809384421344400	-1.0809384421344400	-1.4315374193830000	-2.0000000000000000
	Runtime	27.372	32.311	27.546	19.673	52.853	36.659	21.421	34.714	0.594
F23	Mean	-1.3891992200744600	-0.5235864386288060	-1.4999990070800800	-1.3431399432579700	-1.4765972735526500	-1.4999992233525000	-1.4821658762555300	-1.5000000000000000	-1.5000000000000000
	Std. Dev.	0.2257194403158630	0.2585330714077300	0.0000008440502079	0.2680292304904580	0.1281777579497830	0.0000000000000009	0.0976772648082733	0.0000000000000000	0.0000000000000000
	Best	-1.4999992233524900	-0.7977041047646610	-1.4999992233524900	-1.4999992233524900	-1.4999992233524900	-1.4999992233524900	-1.4999992233524900	-1.5000000000000000	-1.5000000000000000
	Runtime	33.809	17.940	37.986	20.333	42.488	36.037	18.930	41.848	10.181
F24	Mean	-0.9166206788680230	-0.3105071678265780	-0.8406348096500680	-0.8827152798835760	-0.9431432797743700	-1.2765515661973800	-1.3127183561646500	-1.5000000000000000	-1.5000000000000000
	Std. Dev.	0.3917752367440500	0.2080317241440800	0.2000966365984320	0.3882445165494030	0.3184175870987750	0.3599594108130040	0.3158807699946290	0.0000000000000000	0.0000000000000000
	Best	-1.5000000000003800	-0.7976938356122860	-1.4999926800631400	-1.5000000000003800	-1.5000000000003800	-1.5000000000003800	-1.5000000000003800	-1.5000000000000000	-1.5000000000000000
	Runtime	110.798	8.835	38.470	21.599	124.609	47.171	35.358	54.651	28.124
F25	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000004	0.0000000000000000	0.0000041787372626	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std. Dev.	0.0000000000000000	0.0000000000000000	0.0000000000000003	0.0000000000000000	0.0000161643637543	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000023
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000001	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Runtime	25.358	1.340	19.689	1.142	31.632	4.090	0.813	35.662	107.658

F26	Mean	-1.8210436836776800	-1.7829268228561700	-1.8210436836776800	-1.8210436836776800	-1.8210436836776800	-1.8210436836776800	-1.8210436836776800	-1.8203821100000000	-1.8210413449747600
	Std. Dev.	0.0000000000000009	0.1450583631808370	0.0000000000000009	0.0000000000000009	0.0000000000000009	0.0000000000000009	0.0000000000000009	0.0000000000000014	0.0000023227973188
	Best	-1.8210436836776800	-1.8210436836776800	-1.8210436836776800	-1.8210436836776800	-1.8210436836776800	-1.8210436836776800	-1.8210436836776800	-1.8203821095139300	-1.8210433994221000
	Runtime	19.154	26.249	17.228	9.663	18.091	28.453	7.472	34.891	23.653
F27	Mean	-4.6565646397053900	-4.1008953007033700	-4.6934684519571100	-4.6893456932617100	-4.6920941990586400	-4.6884965299983800	-4.6934684519571100	-3.2820108350000000	-4.6452376313815900
	Std. Dev.	0.0557021530063238	0.4951250481844850	0.0000000000000009	0.0125797149251589	0.0075270931220834	0.0272323381095561	0.0000000000000008	0.0000000000000023	0.0076720944206201
	Best	-4.6934684519571100	-4.6934684519571100	-4.6934684519571100	-4.6934684519571100	-4.6934684519571100	-4.6934684519571100	-4.6934684519571100	-3.2820108345268900	-4.6514538047881300
	Runtime	38.651	10.956	17.663	14.915	25.843	38.446	11.971	45.085	80.143
F28	Mean	-8.9717330307549300	-7.6193507368464700	-9.6601517156413500	-9.6397230986132500	-9.6400278592589600	-9.6572038232921700	-9.6601517156413500	-6.2086254390000000	-8.1736283729093200
	Std. Dev.	0.4927013165009220	0.7904830398850970	0.0000000000000008	0.0393668145094111	0.0437935551332868	0.0105890022905617	0.0000000000000007	0.0000000000000027	0.5866949323320730
	Best	-9.5777818097208200	-9.1383975057875100	-9.6601517156413500	-9.6601517156413500	-9.6601517156413500	-9.6601517156413500	-9.6601517156413500	-6.2086254392105500	-8.6571677840756600
	Runtime	144.093	6.959	27.051	20.803	32.801	46.395	22.250	71.652	404.923
F29	Mean	0.0119687224560441	0.0788734736114700	0.0838440014038032	0.0154105130055856	0.0198686590210374	0.0140272066690658	0.0007283694780796	1.3116221610000000	0.0000812300613956
	Std. Dev.	0.0385628598040034	0.1426911799629180	0.0778327303965192	0.0308963906374663	0.0613698943155661	0.0328868042987376	0.0014793717464195	0.5590904820000000	0.0002253714457363
	Best	0.0000044608370213	0.0000000000000000	0.0129834451730589	0.0000000000000000	0.0000175219764526	0.0000000000000000	0.0000000000000000	1.0960146962658900	0.0000378817770911
	Runtime	359.039	17.056	60.216	35.044	316.817	92.412	191.881	34.697	161.868
F30	Mean	0.0000130718912008	0.0000000000000000	0.0002604330013462	0.0000000000000001	0.0458769685199585	0.0000002733806735	0.0000000028443186	0.0000000000000000	0.0000000000000000
	Std. Dev.	0.0000014288348929	0.0000000000000000	0.0000394921919294	0.0000000000000002	0.0620254411839524	0.0000001788830279	0.0000000033308990	0.0000000000000000	0.0000000000000000
	Best	0.0000095067504097	0.0000000000000000	0.0001682411286088	0.0000000000000000	0.0005277712020642	0.0000000944121661	0.0000000004769768	0.0000000000000000	0.0000000000000000
	Runtime	567.704	14.535	215.722	194.117	252.779	360.380	144.784	153.221	27.447
F31	Mean	0.0001254882834238	0.0000000000000000	0.0077905311094958	0.0020185116261490	0.0002674563703837	0.0000000000000000	0.0000000111676630	0.0071082040000000	0.0008686349620636
	Std. Dev.	0.0001503556280087	0.0000000000000000	0.0062425841086448	0.0077448684015362	0.0003044909265796	0.0000000000000000	0.0000000184322163	0.0000000000000000	0.0002025576867284
	Best	0.0000000156460198	0.0000000000000000	0.0003958766023752	0.0000000000000000	0.0000023064754605	0.0000000000000000	0.0000000000000000	0.0071082039505830	0.0000097325749354
	Runtime	250.248	12.062	34.665	48.692	227.817	220.886	149.882	43.098	428.133
F32	Mean	0.0003548345513179	0.0701619169853449	0.0250163252527030	0.0013010316180679	0.0019635752485802	0.0016730768406953	0.0019955316015528	0.0002254250000000	0.0004399017508275
	Std. Dev.	0.0001410817500914	0.0288760292572957	0.0077209314806873	0.0009952078711752	0.0043423828633839	0.0007330246909835	0.0009698942217908	0.0005270410000000	0.0007201737722147
	Best	0.0001014332605364	0.0299180701536354	0.0094647580732654	0.0001787238105452	0.0004206447422138	0.0005630852254632	0.0006084880639553	0.0000023800831017	0.0000974337204994
	Runtime	290.669	2.154	34.982	82.124	103.283	171.637	48.237	218.722	1222.683
F33	Mean	25.6367602258676000	95.9799861204982000	0.0000000000000000	1.1276202647057400	0.6301407361590880	0.8622978494808570	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std. Dev.	8.2943512684216700	56.6919245985100000	0.0000000000000000	1.0688393637536800	0.8046401822326410	0.9323785263847000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	12.9344677422129000	29.8487565993415000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Runtime	76.083	2.740	4.090	7.635	18.429	23.594	5.401	2.266	6.984
F34	Mean	2.6757043114269700	0.3986623855035210	0.2856833465904130	1.0630996944802500	5.7631786582751800	1.2137377447007000	0.3986623854300930	0.0000154715000000	0.0000000000000003
	Std. Dev.	12.3490058210004000	1.2164328621946200	0.6247370987465170	1.7930895051734300	13.9484817304201000	1.8518519388285700	1.2164328622195200	0.0000022373400000	0.0000000000000002

	Best	0.0042535368984501	0.0000000000000000	0.0004266049929880	0.0000000000000000	0.0268003205820685	0.0001448955835246	0.0000000000000000	0.0000118803557196	0.0000000000000000
	Runtime	559.966	9.462	35.865	23.278	187.894	268.449	34.681	7.250	3752.161
F35	Mean	0.0000000000000000	0.4651202457398910	0.0000000000000000	0.0038863639514140	0.0019431819755029	0.0006477273251676	0.0000000000000000	0.0000000000000000	0.0000000000000007
	Std. Dev.	0.0000000000000000	0.0933685176073728	0.0000000000000000	0.0048411743884718	0.0039528023354469	0.0024650053428137	0.0000000000000000	0.0000000000000000	0.0000000000000001
	Best	0.0000000000000000	0.0097159098775144	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000006
	Runtime	18.163	24.021	7.861	4.216	8.304	5.902	1.779	33.155	47.882
F36	Mean	-7684.6104757783800000	-6835.1836730901400000	-12569.4866181730000	-12304.9743375341000	-12210.8815698372000	-12549.746895737300000	-12569.486618173000000	-12569.3622100000000000	0.0000431175670315
	Std. Dev.	745.3954005014180000	750.7338055436110000	0.0000000000022659	221.4322514436480000	205.9313376284770000	44.8939348779747000	0.0000000000024122	0.0000000273871000	0.0001654617933147
	Best	-8912.8855854978200000	-8340.0386911070600000	-12569.4866181730000	-12569.4866181730000	-12569.4866181730000	-12569.486618173000000	-12569.486618173000000	-12569.3622054081000000	0.0000254551338230
	Runtime	307.427	3.174	19.225	10.315	31.499	34.383	11.069	2.306	16.132
F37	Mean	0.0000000000000000	0.0000000000000000	14.5668734126948000	0.0000000000000000	6.4655746330439100	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std. Dev.	0.0000000000000000	0.0000000000000000	8.7128443012950300	0.0000000000000000	8.2188901353055800	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	4.0427699323673400	0.0000000000000000	0.1816624029553790	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Runtime	543.180	3.370	111.841	19.307	179.083	109.551	57.294	100.947	54.725
F38	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000005	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std. Dev.	0.0000000000000000	0.0000000000000000	0.0000000000000001	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000003	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Runtime	163.188	2.558	20.588	1.494	12.563	5.627	3.208	47.009	100.737
F39	Mean	-10.1061873621653000	-5.2607563471326400	-10.5364098166920000	-10.3130437162426000	-10.3130437162026000	-10.5364098166921000	-10.5364098166921000	-10.5063235800000000	-10.5363772084116000
	Std. Dev.	1.6679113661236400	3.6145751818694000	0.0000000000000023	1.2234265179812200	1.2234265179736500	0.0000000000000016	0.0000000000000018	0.0000000025211900	0.0000098784966141
	Best	-10.5364098166921000	-10.5364098166921000	-10.5364098166920000	-10.5364098166921000	-10.5364098166920000	-10.5364098166921000	-10.5364098166920000	-10.5063235792920000	-10.5363963624224000
	Runtime	31.018	11.024	16.015	8.345	37.275	28.031	7.045	55.666	4.065
F40	Mean	-9.5373938082045500	-5.7308569926624600	-10.1531996790582000	-9.5656135761215700	-10.1531996790582000	-9.9847854277673500	-10.1531996790582000	-10.1529842600000000	-10.1531985956925000
	Std. Dev.	1.9062127067994200	3.5141202468383400	0.0000000000000055	1.8315977756329900	0.0000000000000076	0.9224428443735560	0.0000000000000072	0.0000000000542921	0.0000003801065968
	Best	-10.1531996790582000	-10.1531996790582000	-10.1531996790582000	-10.1531996790582000	-10.1531996790582000	-10.1531996790582000	-10.1531996790582000	-10.1529842649756000	-10.1531987237359000
	Runtime	25.237	11.177	11.958	7.947	30.885	25.569	6.864	51.507	2.670
F41	Mean	-10.4029405668187000	-6.8674070870953700	-10.4029405668187000	-9.1615813354737300	-10.4029405668187000	-10.4029405668187000	-10.4029405668187000	-10.3988303400000000	-10.4029086366619000
	Std. Dev.	0.0000000000000018	3.6437803702691000	0.0000000000000006	2.8277336448396200	0.0000000000000010	0.0000000000000018	0.0000000000000017	0.0000000001978980	0.0002103856046932
	Best	-10.4029405668187000	-10.4029405668187000	-10.4029405668187000	-10.4029405668187000	-10.4029405668187000	-10.4029405668187000	-10.4029405668187000	-10.3988303385534000	-10.4029228929926000
	Runtime	21.237	11.482	14.911	8.547	31.207	27.064	8.208	53.190	3.559
F42	Mean	-186.7309073569880000	-81.5609772893002000	-186.730908831024000	-186.730908831024000	-186.730908831024000	-186.7309088310240000	-186.7309088310240000	-186.2926481000000000	-185.3137861632100000
	Std. Dev.	0.0000046401472660	66.4508342743478000	0.0000000000000236	0.0000000000000388	0.0000000000000279	0.0000000000000377	0.0000000000000224	0.0000000000000578	0.0000000000000583
	Best	-186.7309088310240000	-186.7309088310240000	-186.730908831024000	-186.730908831024000	-186.730908831024000	-186.7309088310240000	-186.7309088310240000	-186.2926480689880000	-185.3137861632100000
	Runtime	19.770	25.225	13.342	8.213	20.344	27.109	9.002	31.766	1.241

F43	Mean	-1.0316284534898800	-1.0044229658530100	-1.0316284534898800	-1.0316284534898800	-1.0316284534898800	-1.0316284534898800	-1.0316284534898800	-1.0304357800000000	-1.0316039304478900
	Std. Dev.	0.0000000000000005	0.1490105926664260	0.0000000000000005	0.0000000000000005	0.0000000000000005	0.0000000000000005	0.0000000000000005	0.0014911900000000	0.0000844585095952
	Best	-1.0316284534898800	-1.0316284534898800	-1.0316284534898800	-1.0316284534898800	-1.0316284534898800	-1.0316284534898800	-1.0316284534898800	-1.0314500753985900	-1.0316039304478900
	Runtime	16.754	24.798	11.309	7.147	18.564	27.650	5.691	39.897	74.093
F44	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000004	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std. Dev.	0.0000000000000000	0.0000000000000000	0.0000000000000001	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000003	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Runtime	159.904	2.321	21.924	1.424	14.389	5.920	3.302	174.577	0.239
F45	Mean	2.3000000000000000	0.0666666666666667	0.0000000000000000	0.9000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000538870000000	0.0000000000000000
	Std. Dev.	1.8597367258983700	0.2537081317024630	0.0000000000000000	3.0211895350832500	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000005399890	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000538860819891	0.0000000000000000
	Runtime	57.276	1.477	1.782	2.919	3.042	4.307	0.883	2.215	39.570
F46	Mean	0.1333333333333330	0.2666666666666670	0.0000000000000000	0.0000000000000000	0.2000000000000000	0.0000000000000000	0.0000000000000000	-0.0153463301609662	0.0000000000186260
	Std. Dev.	0.3457459036417600	0.9444331755018490	0.0000000000000000	0.0000000000000000	0.4068381021724860	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	-0.0153463301609662	0.0000000000186260
	Runtime	20.381	2.442	1.700	1.074	6.142	4.319	0.764	31.068	1.979
F47	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000005	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000094641
	Std. Dev.	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000135300152
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000003	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000001
	Runtime	564.178	2.565	24.172	1.870	15.948	6.383	4.309	31.296	95.295
F48	Mean	-50.0000000000002000	-50.0000000000002000	-49.9999999999997000	-50.0000000000002000	-49.4789234062579000	-50.0000000000002000	-50.0000000000002000	-44.7416748700000000	-42.8605934032616000
	Std. Dev.	0.0000000000000361	0.0000000000000268	0.0000000000001408	0.0000000000000354	1.3150773145311700	0.0000000000000268	0.0000000000000361	0.0000000000000217	2.8540566374217100
	Best	-50.0000000000002000	-50.0000000000002000	-50.0000000000001000	-50.0000000000002000	-49.9999994167392000	-50.0000000000002000	-50.0000000000002000	-44.7416748706606000	-45.8421227762905000
	Runtime	24.627	8.337	22.480	8.623	142.106	36.804	7.747	52.486	83.631
F49	Mean	-210.0000000000010000	-210.0000000000030000	-209.999999999947000	-210.000000000003000	-199.592588547503000	-210.0000000000030000	-210.0000000000030000	-150.5540859185450000	-113.9232087177520000
	Std. Dev.	0.0000000000009434	0.0000000000003702	0.0000000000138503	0.0000000000008251	9.6415263953591700	0.0000000000004625	0.0000000000003950	0.0000000000000000	6.0948469712421200
	Best	-210.0000000000030000	-210.0000000000030000	-209.999999999969000	-210.000000000004000	-209.985867409029000	-210.0000000000040000	-210.0000000000040000	-150.5540859185450000	-126.2351567732370000
	Runtime	48.580	5.988	36.639	11.319	187.787	54.421	11.158	70.887	226.062
F50	Mean	0.0000000000000000	0.0000000000000000	0.0000000402380424	0.0000000000000000	0.0000000001597805	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std. Dev.	0.0000000000000000	0.0000000000000000	0.0000002203520334	0.0000000000000000	0.0000000006266641	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000210	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Runtime	86.369	1.868	86.449	1.412	157.838	4.930	5.702	33.573	12.910

5.2 Outstanding Results

Expectation algorithm often perform well in optimising an unconstrained test function to all types of problems, However Expectation algorithm gives best solution for some functions among the proposed algorithms.

Expectation algorithm solves the Rosenbrock function with 30 variables which other proposed algorithm failed to do with precision up to 16 decimals.

Expectation algorithm gives the most satisfy optimum value of Schwefel function i.e. 0.0000254551338230 which is nearest to the best possible optimum value of the function being $f(x^*) = 0$ for $x = (1,\ldots,1)$ whereas other algorithm gives the solution which are not feasible.

Expectation algorithm gives the optimum value with least runtime among other algorithm for Langermann2, Langermann5, Powell, Shekel10, Shekel5, Shekel7, Shubert and Sphere function.

5.3 Wilcoxon Signed Rank Test

The Wilcoxon signed-rank test is a non-parametric statistical hypothesis test used when comparing two related samples, matched samples, or repeated measurements on a single sample to assess whether their population mean ranks differ (i.e. it is a paired difference test). It can be used as an alternative to the paired Student's t-test, t-test for matched pairs, or the t-test for dependent samples when the population cannot be assumed to be normally distributed.

An analysis for the benchmark functions of test 1 (50 in total) was performed to compare which among the presently available algorithms and Expected Algorithm (ExA) performs better.

A '+' sign indicates cases in which the null hypothesis is rejected and Expected Algorithm displays a statistically superior performance. The '-' sign indicates cases in which the null hypothesis was rejected and Expected Algorithm displayed an inferior performance; '=' indicates cases in which there was no statistical difference between the two algorithms' success in solving the problems.

Table 3 Wilcoxon signed-rank test results

Problem	PSO2011 vs ExA				CMAES vs ExA				ABC vs ExA				JDE vs ExA			
	p-value	T+	T-	winner	p-value	T+	T-	winner	p-value	T+	T-	winner	p-value	T+	T-	winner
F1	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+
F2	1.6912e-06	0	465	+	1.6912e-06	0	465	+	1.6912e-06	465	0	-	1.6912e-06	465	0	-
F3																
F4																
F5	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	465	0	-	4.3205e-08	0	465	+
F6	1.7235e-06	465	0	-	1.7116e-06	0	465	+	1.7116e-06	465	0	-	1.7116e-06	0	465	+
F7	4.3205e-08	0	465	+	4.3205e-08	0	465	+	1	0	0	=	1	0	0	=
F8	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	465	0	-	4.3205e-08	465	0	-
F9	6.0003e-07	0	465	+	6.0003e-07	0	465	+	6.0003e-07	465	0	-	6.0003e-07	0	465	+
F10	1.5029e-06	0	465	+	1.5029e-06	0	465	+	1.5029e-06	0	465	+	1.5029e-06	0	465	+
F11	1.7116e-06	0	465	+	1.7116e-06	0	465	+	1.7116e-06	465	0	-	1.7116e-06	465	0	-
F12	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+
F13	1.7224e-06	0	465	+	1.7224e-06	465	0	-	1.7224e-06	465	0	-	1.7224e-06	465	0	-
F14	4.3205e-08	465	0	-	4.3205e-08	465	0	-	4.3205e-08	465	0	-	4.3205e-08	465	0	-
F15	1.5029e-06	0	465	+	4.3205e-08	0	465	+	1	0	0	=	1	0	0	=
F16	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+
F17	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+
F18	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+
F19	1.5029e-06	0	465	+	1.5441e-06	0	465	+	1.5441e-06	465	0	-	1.5441e-06	465	0	-
F20	2.5994e-05	30	435	+	2.5994e-05	30	435	+	2.5994e-05	30	435	+	2.5994e-05	30	465	+
F21	5.8939e-07	0	465	+	5.8939e-07	0	465	+	5.8939e-07	0	465	+	5.8939e-07	0	465	+
F22	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+
F23	1.5029e-06	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+
F24	1.7116e-06	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+
F25	0.1777	167	298	+	0.1777	167	298	+	0.1777	167	298	+	1.7138e-06	0	465	+
F26	1.6955e-06	0	465	+	1.6955e-06	0	465	+	1.6955e-06	465	0	-	1.6955e-06	465	0	-
F27	1.7127e-06	465	0	-	1.7127e-06	0	465	+	1.7127e-06	465	0	-	1.7127e-06	0	465	+
F28	1.3670e-06	465	0	-	7.5808e-05	42	423	+	1.3670e-06	465	0	-	1.3670e-06	465	0	-
F29	2.6158e-07	0	465	+	2.6158e-07	0	465	+	2.6158e-07	0	465	+	2.6158e-07	0	465	+
F30	4.3205e-08	0	465	+	1	0	0	=	4.3205e-08	0	465	+	4.3205e-08	0	465	+
F31	1.9042e-06	465	0	-	1.7192e-06	465	0	-	1.7192e-06	0	465	+	1.7192e-06	0	465	+
F32	0.4652	268	197	-	1.7235e-06	0	465	+	1.7235e-06	0	465	+	3.5759e-04	59	406	+
F33	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+
F34	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+
F35	4.3205e-08	0	465	+	4.3205e-08	0	465	+	1.7062e-06	0	465	+	1.7062e-06	0	465	+
F36	1.6647e-06	465	0	-	1.6647e-06	465	0	-	1.6615e-06	465	0	-	1.6615e-06	465	0	-
F37	1	0	0	=	1.3626e-06	465	0	-	4.3205e-08	0	465	+	4.3205e-08	0	465	+
F38	1	0	0	=	1.7105e-06	465	0	-	1.7105e-06	0	465	+	1.7105e-06	465	0	-
F39	1.5851e-06	0	465	+	1.5851e-06	0	465	+	1.5851e-06	465	0	-	1.5851e-06	0	465	+
F40	4.3205e-08	0	465	+	9.0067e-07	0	465	+	9.0067e-07	0	465	+	9.0067e-07	0	465	+
F41	4.3205e-08	0	465	+	1.2919e-06	0	465	+	1.2919e-06	0	465	+	1.2919e-06	0	465	+
F42	1.5851e-06	465	0	-	4.3205e-08	0	465	+	4.3205e-08	465	0	-	4.3205e-08	465	0	-
F43	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	465	0	-	4.3205e-08	465	0	-
F44	1	0	0	=	7.3805e-05	210	0	+	1.4402e-06	0	465	+	7.3805e-05	210	0	-
F45	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+
F46	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+
F47	1.6901e-06	465	0	-	1.6901e-06	465	0	-	3.0512e-05	435	30	-	1.6901e-06	465	0	-
F48	1.6859e-06	465	0	-	1.6859e-06	465	0	-	1.6859e-06	465	0	-	1.6859e-06	465	0	-
F49	1.7094e-06	465	0	-	1.7094e-06	465	0	-	1.7094e-06	465	0	-	1.7094e-06	465	0	-
F50	4.3205e-08	0	465	+	1.5700e-06	0	465	+	1.5700e-06	0	465	+	1.5700e-06	0	465	+
+/=/-	34/3/11				38/1/9				27/2/19				30/2/16			

Problem	CLPSO vs ExA				SADE vs ExA				BSA vs ExA				IA vs ExA			
	p-value	T+	T-	winner	p-value	T+	T-	winner	p-value	T+	T-	winner	p-value	T+	T-	winner
F1	4.3205e-08	0	465	+	4.3205e-08	465	0	-	4.3205e-08	465	0	-	1.6657e-06	465	0	-
F2	1.6912e-06	465	0	-	1.6912e-06	465	0	-	1.6912e-06	465	0	-	1.7041e-06	0	465	+
F3																
F4																
F5	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	465	0	-	4.3205e-08	465	0	-
F6	1.7116e-06	0	465	+	1.7116e-06	465	0	-	1.7116e-06	465	0	-	1.7116e-06	0	465	+
F7	1	0	0	=	1	0	0	=	1	0	0	=	1	0	0	=
F8	4.3205e-08	465	0	-	4.3205e-08	465	0	-	4.3205e-08	465	0	-	4.3205e-08	465	0	-
F9	6.0003e-07	0	465	+	6.0003e-07	465	0	-	6.0003e-07	465	0	-	6.0003e-07	465	0	-
F10	1.5029e-06	0	465	+	1.5029e-06	465	0	-	1.5029e-06	465	0	-	4.3205e-08	465	0	-
F11	1.7116e-06	465	0	-	1	0	0	=	1.7116e-06	465	0	-	1.7289e-06	0	465	+
F12	4.3205e-08	0	465	+	1.7224e-06	0	465	+	1	0	0	=	5.9869e-07	0	465	+
F13	1.7224e-06	465	0	-	1.7224e-06	0	465	+	1.7224e-06	0	465	+	0.0243	465	0	-
F14	4.3205e-08	465	0	-	4.3205e-08	465	0	-	4.3205e-08	465	0	-	1.0789e-06	465	0	-
F15	1	0	0	=	1	0	0	=	1	0	0	=	1	0	0	=
F16	4.3205e-08	0	465	+	1	0	0	=	1	0	0	=	1	0	0	=
F17	4.3205e-08	0	465	+	1	0	0	=	1	0	0	=	1	0	0	=
F18	6.8564e-07	465	0	-	4.3205e-08	0	465	+	4.3205e-08	0	465	+	1	0	0	=
F19	1.5441e-06	465	0	-	1.5441e-06	465	0	-	1.5441e-06	465	0	-	1.6944e-06	0	465	+
F20	2.5994e-05	30	465	+	2.5994e-05	30	435	+	2.5994e-05	30	435	+	1.3715e-06	0	465	+
F21	5.8939e-07	465	0	-	5.8939e-07	465	0	-	5.8939e-07	465	0	-	6.9570e-07	0	465	+
F22	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+
F23	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	1	0	0	=
F24	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	1	0	0	=
F25	1.7138e-06	0	465	+	1.7138e-06	465	0	-	1.7138e-06	465	0	-	1	0	0	=
F26	1.6955e-06	465	0	-	1.6955e-06	465	0	-	1.6955e-06	465	0	-	1.6955e-06	0	465	+
F27	1.7127e-06	465	0	-	1.7127e-06	465	0	-	1.7127e-06	465	0	-	1.7127e-06	0	465	+
F28	1.3670e-06	465	0	-	1.3670e-06	465	0	-	1.3670e-06	465	0	-	1.3670e-06	0	465	+
F29	2.6158e-07	0	465	+	2.6158e-07	0	465	+	3.3093e-07	2	463	+	3.3519e-07	0	465	+
F30	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	1	0	0	=
F31	1.9042e-06	464	1	-	1.7192e-06	465	0	-	1.7192e-06	465	0	-	1.7192e-06	0	465	+
F32	1.9630e-05	25	440	+	3.5759e-04	59	406	+	9.2651e-06	17	448	+	1.7235e-06	465	0	-
F33	4.3205e-08	0	465	+	4.3205e-08	0	465	+	1	0	0	=	1	0	0	=
F34	4.3205e-08	0	465	+	4.3205e-08	0	465	+	4.3205e-08	0	465	+	1.7344e-06	0	465	+
F35	1.7148e-06	0	465	+	1.7148e-06	0	465	+	1.7235e-06	465	0	-	1	0	0	=
F36	1.6615e-06	465	0	-	1.6615e-06	465	0	-	4.3205e-08	465	0	-	4.3205e-08	465	0	-
F37	4.3205e-08	0	465	+	1.3626e-06	465	0	-	1.3626e-06	465	0	-	1	0	0	=
F38	1.7105e-06	465	0	-	1.7105e-06	465	0	-	1.7105e-06	465	0	-	1	0	0	=
F39	1.5851e-06	0	465	+	1.5851e-06	465	0	-	1.5851e-06	465	0	-	1.7322e-06	0	465	+
F40	9.0067e-07	0	465	+	9.0067e-07	0	465	+	9.0067e-07	465	0	-	1.7322e-06	0	465	+
F41	1.2919e-06	465	0	-	1.2919e-06	465	0	-	1.2919e-06	465	0	-	1.7322e-06	0	465	+
F42	4.3205e-08	465	0	-	4.3205e-08	465	0	-	4.3205e-08	465	0	-	4.3205e-08	465	0	-
F43	4.3205e-08	465	0	-	4.3205e-08	465	0	-	4.3205e-08	465	0	-	1.7322e-06	0	465	+
F44	7.3805e-05	210	0	-	7.3805e-05	210	0	-	7.3805e-05	210	0	-	1	0	0	=
F45	1	0	0	=	1	0	0	=	1	0	0	=	1.7300e-06	0	465	+
F46	4.3205e-08	0	465	+	4.3205e-08	465	0	-	4.3205e-08	465	0	-	4.3205e-08	465	0	-
F47	1.6901e-06	465	0	-	1.6901e-06	465	0	-	1.6901e-06	465	0	-	1	0	0	=
F48	1.6859e-06	465	0	-	1.6859e-06	465	0	-	1.6859e-06	465	0	-	4.3205e-08	465	0	-
F49	1.7094e-06	465	0	-	1.7094e-06	465	0	-	1.7094e-06	465	0	-	4.3205e-08	465	0	-
F50	1.5700e-06	0	465	+	4.3205e-08	465	0	-	4.3205e-08	465	0	-	1	0	0	=
+/=/-	24/3/21				15/6/27				10/7/31				19/16/13			

5.4 Comparison with other algorithms

5.4.1 PSO & CLPSO compared to ExA

In PSO, the individual particles of a swarm symbolize potential solutions. They 'fly' through the search space of the problem, trying to seek an optimal solution. The current positions of the particles are broadcasted to other neighboring particles. Previously identified 'good position' is then used as a starting point by the swarm for further search. On the other hand, the individual particles adjust their current positions and velocities. A distinct characteristic of PSO is its fast-convergent behavior and inherent adaptability, especially when compared to conventional EAs.

The drawback of the basic PSO algorithm is that it easily suffers from the partial optimism, which might lead to reduced precision in speed and the regulation of direction. PSO is unable to solve the problems of scattering and optimization, as well as the problems of non-coordinate system, such as the solution to the energy field and the moving rules of the particles in the energy field.

The proposed ExA is being compared with PSO and its variants, CLPSO. The statistical results of Test indicate that ExA is better as compared with PSO (34/3/11) and CLPSO (24/3/21).

5.4.2 CMAES compared to ExA

The CMAES algorithm stands for covariance matrix adaptation evolution strategy. It is a mathematical-based algorithm that makes use of adaptive mutation parameters through computing a covariance matrix.

One major drawback of CMAES is the cost in calculating the covariance matrix. The cost increases rapidly with increasing dimensions. Plus, sampling using a multivariate normal distribution and factorization of the covariance matrix also becomes increasingly expensive.

The proposed ExA is being compared with CMAE. The statistical results of Test indicate that ExA is better as compared with CMAE (38/1/9).

5.4.3 ABC compared to ExA

In ABC algorithm, the artificial bee colony is made up of employed bees, onlooker bees and scout bees. An onlooker bee waits on the dance area for making decision in choosing a food source. An employed bee goes to the previously visited food source to search for food. A scout bee carries out random search. An existing challenge to all stochastic optimization methods is the balance between exploration and exploitation. A poor optimization will meet the problems of premature convergence and get trapped from local minima. Meanwhile, excessively exploitative will cause the algorithm to converge very slowly. ABC is good at exploration but poor at exploitation; its convergence speed is also an issue in some cases.

The proposed ExA is being compared with ABC. The statistical results of Test indicate that ExA is better as compared with ABC (27/2/19).

5.4.4 SADE & JDE compared to ExA

DE is a population-based algorithm which uses the similar operators as GA: crossover, mutation and selection. The only difference is that GA relies on crossover where DE relies on mutation operation. DE algorithm uses mutation operation as a search mechanism and selection operation to direct the search in the search space. By creating trial vectors using the components of existing individuals in the population, the crossover operator effectively sorts information about successful combinations, enabling better solution search space.

In DE, a population of solution vectors is randomly created at the start. This population is successfully improved by applying mutation, crossover and selection operators. In DE algorithm, each new solution produced competes with a mutant vector and the better one wins the competition. In other words, the chance of succession is independent on their fitness values. Every new solution produced competes with its parent, and the better one wins the competition.

The proposed ExA is being compared with SADE and JDE. The statistical results of Test indicate that ExA is inferior to SADE (15/6/27) but superior in performance when compared with JDE (30/2/16).

5.4.5 BSA compared with ExA

In BSA, three basic genetic operators—selection, mutation and crossover—are used to generate trial individuals. A random mutation strategy is performed such that only one direction individual is used for each target individual. BSA randomly chooses the direction individual from a randomly chosen individual from previous generation. BSA uses a non-uniform crossover strategy that is more complex as compared with other GAs. The unique mutation and crossover strategies of BSA make it a powerful minimization technique and is proved in comparison that BSA performs better than ExA (10/7/31).

5.4.6 Ideology Algorithm (IA) compared with ExA

The self-interested behaviour of every individual in IA enables them to communicate with each other in order to seek for better solutions. They respond adaptively to the shape of the fitness landscape. Thus, IA is able to achieve higher convergence rate in the iterative processes. It is because the efforts of improving the best solution depend on not only the current position of the particle itself but also the position of the global best individual, local best individual and the local second best individual. This can prevent the problem of falling into local optimum in high dimensional space, which is the common problem faced by most of the EAs. The proposed ExA is being compared with IA. The statistical results of Test indicate that ExA is slightly better or almost similar to IA (19/16/13).

Table 4 : Multi-problem based statistical pair wise comparison of PSO, CMAES, ABC, JDE, CLPSO, SADE, BSA, IA and ExA

Other Algorithm vs IA	p-Value	T+	T-	Winner
PSO vs ExA	0.3644	325	455	ExA
CMAES vs ExA	9.347e-5	129	732	ExA
ABC vs ExA	0.6571	487	416	ABC
JDE vs ExA	0.2064	316	504	ExA
CLPSO vs ExA	0.2850	366	537	ExA
SADE vs ExA	0.6346	383	320	SADE
BSA vs ExA	0.2854	401	265	BSA
IA vs ExA	0.2959	222	339	ExA

Chapter-6

LIMITATIONS

This method does not work when it comes to performing derivative operation on variables which are not represented symbolically in the MATLAB program.

For example, Penalized function given below:

$$f(x) = \frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\,[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\}$$
$$+ \sum_{i=1}^{n} u(x_i, a, k, m)$$

Where:

$$y_i = 1 + \frac{(x_i+1)}{4}\ , \quad u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & if\ \ x_i > a \\ 0 & if - a \le x_i \le a \\ k(-x_i - a)^m & if\ x_i < -a \end{cases}$$

$a = 10, k = 100, m = 4$

$-50 \le x_i \le 50, i = 1,2,\ldots.,n$

$f_{\min}(X^*) = 0$

$x_i^* = -1$

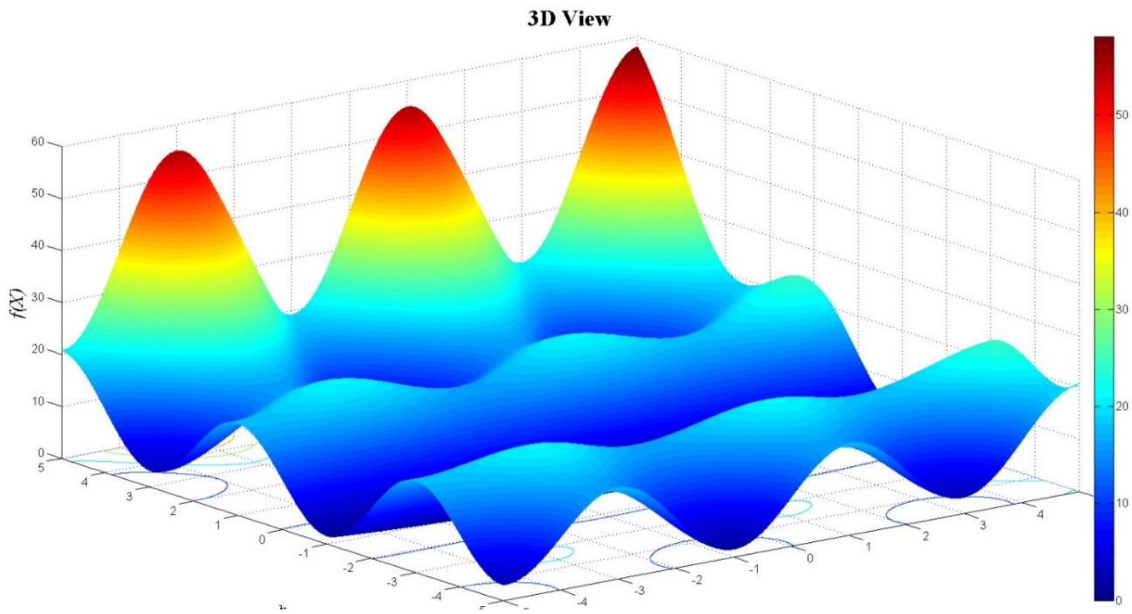

Figure 7 Penalized function

When it comes to representing the function in MATLAB, we use symbolic operator to generalize the algorithm, but in this case (i.e. for penalized function) as there is a conditional operator which checks if the value of variable is greater than 10 or not. Which means according to our algorithm, a symbolic variable is being checked or compared with a constant. Therefore, whenever there's a need for convergence (Steepest descent) the algorithm is designed such that it has to leave the main algorithm and go to a linked steepest descent program which converges the function in an interval and returns the value to main algorithm. However, when it goes to steepest decent program it gets checked for the above-mentioned condition and this results in termination of the program. Naturally, it not feasible to compare a symbolic variable to a constant. This gives us nothing but error in the first iteration itself which implies our program is not suitable for functions like these.

To overcome this problem, one can use any method other than symbolic representation of variables.

Chapter-7

APPLICATIONS

Engineering optimization helps engineers zero in on the most effective, efficient solutions to problems.

- Expectation algorithm can be useful in the field of transport where the shortest path is mandatory or of great use. For example, finding the shortest route from an origin to a destination within a road network, can be treated as finding the shortest path between two nodes in an undirected graph with non-negative costs.
- Expectation algorithm can be used for solving various mazes and puzzles of the same type.
- Since, by nature it learns from other candidates as well, this method could be well suited for designing optimised processors, as the processor requires transfer of information in the shortest route within a time frame.
- As it is known that minimum potential energy of the particle exists at equilibrium position of a particle, if by some means a large particle or shape or combination of particles need to be defined in equilibrium, this algorithm can be used for such applications.

Chapter-8

SUMMARY AND FUTURE DIRECTIONS

A socio-inspired algorithm referred to as expectation algorithm (ExA), which is mainly inspired from the society individuals following certain ideology, is proposed. Several operators were proposed and mathematically modelled for equipping the ExA with high exploration and exploitation. The performance of the proposed algorithm was benchmarked on 50 test functions. It can be concluded that the proposed algorithm benefits from high exploitation and convergence rate. The ExA is compared to eight well-known and recent algorithms: PSO, CMAES, ABC, JDE, CLPSO, SADE, BSA and IA. Wilcoxon statistical tests were also conducted when comparing the algorithms. The results showed that the proposed algorithm outperforms other algorithms in the majority of test functions. The statistical tests proved that the results were statistically significant for the ExA. Thus, it may be concluded from the results that the proposed ExA is comparable with other algorithms. Also, it can be applied as an alternative optimizer for different optimization problems.

ExA effectively searches and converges towards promising search space. Thus, the proposed algorithm can discover different regions of an optimization problem. Other remarks based on the results of this study are as follows:

• Initial random values of candidates around the space or within the interval emphasize exploration of the search space within the global interval.

• Effective in local optima avoidance since ExA employs expectation method which allows it to learn from other candidates makes it to reach global optimum.

Chapter-9

REFERENCES

- Teo Ting Huan, Anand J. Kulkarni, Jeevan Kanesan, Chuah Joon Huang, Ajith Abraham, Ideology algorithm: a socio-inspired optimization methodology
  doi: 10.1007/s00521-016-2379-4
- https://sites.google.com/site/oatresearch/home
- Jan A. Snyman (2005). *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms.* Springer Publishing. ISBN 0-387-24348-8
- Jeff Clune, Benjamin Beckmann, Charles Ofria, and Robert Pennock. "Evolving Coordinated Quadruped Gaits with the HyperNEAT Generative Encoding". *Proceedings of the IEEE Congress on Evolutionary Computing Special Section on Evolutionary Robotics*, 2009. Trondheim, Norway.
- Beni, G., Wang, J. Swarm Intelligence in Cellular Robotic Systems, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26–30 (1989)