APPLIED NONPARAMETRIC STATISTICAL TESTS TO COMPARE EVOLUTIONARY AND SWARM INTELLIGENCE APPROACHES

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Srinivas Adithya Amanchi

In Partial Fulfillment
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

March 2014

Fargo, North Dakota

North Dakota State University

Graduate School

Title

APPLIED NONPARAMETRIC STATISTICAL TESTS TO COMPARE EVOLUTIONARY AND SWARM INTELLIGENCE APPROACHES

By

Srinivas Adithya Amanchi

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Simone Ludwig

Chair

Dr. Rui Dai

Dr. Abraham Ayebo

Approved:

03/24/2014	Dr. Brian M. Slator
Date	Department Chair

ABSTRACT

Recently, in many experimental studies, the statistical analysis of nonparametric comparisons has grown in the area of computational intelligence. The research refers to application of different techniques that are used to show comparison among the algorithms in an experimental study. Pairwise statistical technique perform individual comparison between two algorithms and multiple statistical technique perform comparison between more than two algorithms. Techniques include the Sign test, Wilcoxon signed ranks test, the multiple sign test, the Friedman test, the Friedman aligned ranks test and the Quade test.

In this paper, we used these tests to analyze the results obtained in an experimental study comparing well-known algorithms and optimization functions. The analyses showed that the application of statistical tests helps to identify the algorithm that is significantly different than the remaining algorithms in a comparison. Different statistical analyses were conducted on the results of an experimental study obtained with varying dimension size.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Simone Ludwig for her mentoring, thoughtful ideas, and continuous support.

I would like to take this opportunity to thank the members of committee Dr. Rui Dai and my external mentor Dr. Abraham Ayebo for their invaluable support.

Last but not the least, special thanks to my family and friends who encouraged, and supported me in the process of completion of the degree.

TABLE OF CONTENTS

LIST OF TABLES

1. INTRODUCTION

In present days, the usage of statistical tests in computational intelligence is commonly used for improving the evaluation process. Usually, these statistical tests are employed in the process of any experimental analysis to check whether the algorithm is better than the other. Depending upon the type of data employed that is used for analyses, statistical procedures are classified as parametric and nonparametric [1].

Parametric statistics are well-known statistical methods which are based on assumptions. These tests are said to have more power with correct assumptions which provides more precise and accurate estimates. However, parametric tests can mislead in case of incorrect assumptions especially during the analyses of stochastic algorithms based on computational intelligence [2,3]. Nonparametric statistical procedures are devoid of limitation of assumptions and can grow in size to accommodate the complexity of data. Hence nonparametric tests are considered as the practical tool in single and multi-problem analysis unlike parametric tests that studies only single problem analysis.

Nonparametric procedure is categorized as pairwise and multiple comparison tests. In this paper, our interest is focused on two types of pairwise and four types of multiple comparison tests. The sign test and the Wilcoxon signed ranks tests belong to the branch of pairwise comparison and the multiple sign test, the Friedman test, the Friedman aligned ranks test and the Quade test belong to the branch of multiple comparison. The main objectives of these tests are as follows:

1. Application of nonparametric statistical tests in the area of computational intelligence. The tests used are already proposed in many papers of literature [2-5]. The properties of different tests are explained and to show how these tests can

improve the way in which practitioners and researchers can contrast the results obtained in their studies.

2. Provides a set of procedures to choose any of the statistical tests for the analysis of their results

Throughout the paper, the test problems of CEC' 2005 special session are opted to represent the real parametric optimization through illustration of tests, analysis of performances, evolutionary algorithms and swarm intelligence algorithms. This paper reiterates the efficacy of different statistical techniques and identifies the most appropriate and efficient statistical techniques among the computational intelligence algorithms.

This paper is organized well as follows. Section 2 gives some introductory background on the benchmark functions suite considered for the application of procedures, hypothesis testing and description of nonparametric tests for pair-wise and multiple comparisons. Section 3 describes the evaluation of statistical tests using MS-Excel [15] and MATLAB [16] tools. Section 4 provides statistical analysis for four data tables considered in the form of separate test cases and finally Section 5 concludes the paper.

2. LITERATURE REVIEW

This section covers the representation of benchmark functions, swarm intelligence algorithms and differential evolution algorithms along with some inferential statistics.

2.1. Bench mark functions: CEC'2005 special session on real parameter optimization

Through this paper the statistical differences between different algorithms are exhibited with non-parametric tests. (1) An experimental study relating 9 algorithms and 25 optimization functions demonstrating different statistical methodologies are used. We have chosen 25 test problems of dimension 10 used in CEC'2005 special session on real parameter optimization [6]. (2) An experimental study relating 5 strategies of differential evolution algorithm with 20 optimization functions demonstrating different statistical methods used. We have chosen 20 test problems run for dimension 10, 30 and 50.

The benchmark suite [6] is composed of 5 unimodal functions, 20 multi-modal functions.

Unimodal functions

- F1: Shifted Sphere Function.
- F2: Shifted Schwefel's Problem 1.2.
- F3: Shifted Rotated High Conditioned Elliptic Function.
- F4: Shifted Schwefel's Problem 1.2 with Noise in Fitness.
- F5: Schwefel's Problem 2.6 with Global Optimum on Bounds.

Multimodal functions

- F6: Shifted Rosenbrock's Function.
- F7: Shifted Rotated Griewank Function without Bounds.
- F8: Shifted Rotated Ackley's Function with Global Optimum on Bounds.
- F9: Shifted Rastrigin's Function.

- F10: Shifted Rotated Rastrigin’s Function.
- F11: Shifted Rotated Weierstrass Function.
- F12: Schwefel’s problem 2.13.
- F13: Expanded Extended Griewank’s plus Rosenbrock’s function (F8F2)
- F14: Shifted Rotated Expanded Scaffers F6.
- Each one (F15 to F25) has been defined through compositions of 10 out of the 14 previous functions (different in each case).

Benchmark functions: All functions are displaced in order to ensure that their optima can never be found in the center of the search space. Additionally, for the two functions, the optima cannot be found within the initialization range, and the domain of search is not limited (the optimum is out of the range of initialization).

2.2. Comparison algorithms

2.2.1. Evolutionary and swarm intelligence algorithms

Our main objective in this case study is to compare the performance of 9 continuous optimization algorithms. A brief description and the characteristics of the algorithms are described below:

- **PSO**: Particle swarm optimization (PSO) [7] is an artificial intelligence computational technique that optimizes a problem trying to improve a candidate solution for every iteration. It is used to optimize a problem by having a population of candidate solutions (particles) and moving these particles around in the search-space using mathematical formulae over the solution's position and velocity. Each particle is influenced by its local best known position and is also guided in the search space toward the best known positions that are updated as better positions. Always this method tries to move the swarm toward the best solutions. The population consists of 100 individuals and the parameters are c1 = 2.8, c2 = 1.3, and w from 0.9 to 0.4. For this characteristic, a classic PSO model for numerical optimization has been used.

- **IPOP-CMA-ES:** Evolutionary algorithms like Genetic algorithms use combination and selection method and in PSO particle shares some information with other particles that helps in next effort by the information in search-space. Unlike GA and PSO, CMA-ES finds the best solution by updating its mean and covariance matrix to displace the distribution by generating sets of search points according to the multi variation of normal distribution [26]. It is called as restart Covariant Matrix Evolutionary Strategy (CMA-ES) [8]. This CMA-ES variation begins a restart by doubling the population size, once it identifies the premature convergence. The doubling population size increases the global reach after every restart which empowers the operation mode of CMA-ES variation on multi-modal functions. We have considered using the default parameters and the initial distribution size is one third of the domain size.
- **CHC:** CHC (Cross-generational elitist selection, Heterogeneous recombination, and cataclysmic mutation) is evolved from GA that uses a highly disruptive crossover operator to generate new individuals almost completely different from their parents. Best individuals are not the only ones that participate in mating, but parents are allowed to get paired randomly in a mating pool. However, if the Hamming distance between the parents is above a certain level then recombination is applied. CHC exchanges half of the other genes using half-uniform crossover technique. Instead of

applying mutation directly, CHC uses a re-start mechanism when the population does not change after a given number of iterations [27]. CHC model was tested with real-coded chromosomes, using a real-parameter crossover operator, BLX-α (with α = 0.5), and a population size of 50 chromosomes [9, 10].

- **SSGA:** It is called as Steady state GA. It means there are no iterations (generations). Unlike generic GA the tournament selection does not replace some of the individuals in population. SSGA does not add children of selected parents into the next generation but chooses the best two individuals out of four (two parents and two children) and add them back to the population to keep its size unchanged [28]. A real-coded Steady-State Genetic algorithm is performed on high population diversity levels with BLX-α crossover operator (with α=0.5) and negative assortative mating strategy [11]. Diversity is checked by means of BGA mutation operator.
- **SS-arit & SS-BLX**: Scatter search is a population based evolutionary method that uses a reference set to combine its solutions and produce other individuals. A reference set is generated from a population of solutions. An improvement procedure is run over the solutions in the reference set that are combined to get individuals. The result may sometimes indicate an updating of the reference set and also an updating of the population of individuals. SS builds, maintains and evolves a set of solutions throughout the search [29]. SS-arit and SS-BLX are two classic scatter search models using the arithmetical combination operator and BLX-α crossover operator [12].
- **DE-Exp & DE-Bin**: DE model [13] is explained in the next section. Two classic crossover operators proposed in the literature. They are Rand/1/exp, and Rand/1/bin. The population size is 100 individuals with F=0.5 and CR=0.9

- **SaDE**: Another type of differential evolution is self-adaptive DE. Due to different characteristics and good performance on problems, two learning strategies in DE are selected as candidates. Selected strategies chosen are applied to individuals in the current population with probability proportional to their previous success rates to generate potentially new candidates. Instead of dealing with fixed values for different classes of problems two out of three parameters that is F and CR are changed adaptively and NP is set to user-defined to take care of complex problems. Adapting parameters F and CR are used with a population size of 100 individuals [14].

Mean values are tabulated in Table 1.

Table 1: Error table obtained for each 25 benchmark functions and 9 algorithms with dimension=10 [31]

Fun	PSO	IPOP	CHC	SSGA	SS-BLX	SS-Arit	DE-Bin	DE-Exp	SaDE
F1	1.23E-04	0	2.464	8.42E-09	3.40E+01	1.06E+00	7.72E-09	8.26E-09	8.42E-09
F2	2.60E-02	0	1.18E+02	8.72E-05	1.73E+00	5.28E+00	8.34E-09	8.18E-09	8.21E-09
F3	5.17E+04	0	2.70E+05	7.95E+04	1.84E+05	2.54E+05	4.23E+01	9.94E+01	6.56E+03
F4	2.488	2.93E+03	9.19E+01	2.59E-03	6.23E+00	5.76E+00	7.69E-09	8.35E-09	8.09E-09
F5	4.10E+02	8.10E-10	2.64E+02	1.34E+02	2.19E+00	1.44E+01	8.61E-09	8.51E-09	8.64E-09
F6	7.31E+02	0	1.42E+06	6.17E+00	1.15E+02	4.95E+02	7.96E-09	8.39E-09	1.61E-02
F7	26.78	1.27E+03	1.27E+03	1.27E+03	1.97E+03	1.91E+03	1.27E+03	1.27E+03	1.26E+03
F8	20.43	2.00E+01	2.03E+01	2.04E+01	2.04E+00	2.04E+01	2.03E+01	2.04E+01	20.32
F9	14.38	2.84E+01	5.89E+00	7.29E-09	4.20E+00	5.96E+00	4.55E+00	8.15E-09	8.33E-09
F10	14.04	2.33E+01	7.12E+00	1.71E+01	1.24E+01	2.18E+01	1.23E+01	1.12E+01	15.48
F11	5.59	1.34E+00	1.60E+00	3.26E+00	2.93E+00	2.86E+00	2.43E+00	2.07E+00	6.796
F12	6.36E+02	2.13E+02	7.06E+02	2.79E+02	1.51E+02	2.41E+02	1.06E+02	6.31E+01	56.34
F13	1.503	1.13E+00	8.30E+01	6.71E+01	3.25E+01	5.48E+01	1.57E+00	6.40E+01	70.7
F14	3.304	3.78E+00	2.07E+00	2.26E+00	2.80E+00	2.97E+00	3.07E+00	3.16E+00	3.415
F15	3.40E+02	1.93E+02	2.75E+02	2.92E+02	1.14E+02	1.29E+02	3.72E+02	2.94E+02	84.23
F16	1.33E+02	1.17E+02	9.73E+01	1.05E+02	1.04E+02	1.13E+02	1.12E+02	1.13E+02	1.23E+02
F17	1.50E+02	3.39E+02	1.05E+02	1.19E+02	1.18E+02	1.28E+02	1.42E+02	1.31E+02	1.39E+02
F18	8.51E+02	5.57E+02	8.80E+02	8.06E+02	7.67E+02	6.58E+02	5.10E+02	4.48E+02	5.32E+02
F19	8.50E+02	5.29E+02	8.80E+02	8.90E+02	7.56E+02	7.01E+02	5.01E+02	4.34E+02	5.20E+02
F20	8.51E+02	5.26E+02	8.96E+02	8.89E+02	7.46E+02	6.41E+02	4.93E+02	4.19E+02	4.77E+02
F21	9.14E+02	4.42E+02	8.16E+02	8.52E+02	4.85E+02	5.01E+02	5.24E+02	5.42E+02	5.14E+02
F22	8.07E+02	7.65E+02	7.74E+02	7.52E+02	6.83E+02	6.94E+02	7.72E+02	7.72E+02	7.66E+02
F23	1.03E+03	8.54E+02	1.08E+03	1.00E+03	5.74E+02	5.83E+02	6.34E+02	5.82E+02	6.51E+02
F24	4.12E+02	6.10E+02	2.96E+02	2.36E+02	2.51E+02	2.01E+02	2.06E+02	2.02E+02	2.00E+02
F25	5.10E+02	1.82E+03	1.76E+03	1.75E+03	1.79E+03	1.80E+03	1.74E+03	1.74E+03	1.74E+03

All the algorithms considered have been run 50 times for each test function.

2.2.2. Differential evolution algorithm with different strategies

Another study in this paper consists of comparison of performance between 5 strategies of differential evolution algorithm on 20 problems. DE [30] optimizes a problem by maintaining a population of candidate solutions and creating new candidate solutions by combining existing ones according to its simple formulae, and then keeping whichever candidate solution has the best score or fitness on the optimization problem at hand. Strategies of DE are best explained by the type of mutation scheme we consider. They are described below from [30]:

- DE/rand/1: The mutation scheme uses a randomly selected vector and one weighted difference and hence the name DE/rand/1. The equation for DE/rand/1 is $V_{i,G} = X_{3,G} + F(X_{r_{1,G}} - X_{r_{2,G}})$
- DE/rand/2: It uses two weighted differences and hence the name. The equation is $V_{i,G} = X_{3,G} + F(X_{r_{1,G}} - X_{r_{2,G}}) + F(X_{r_{4,G}} - X_{r_{5,G}})$
- DE/best/1: The mutation scheme uses a best vector and one weighted difference and hence the name. The equation is $V_{i,G} = X_{best,G} + F(X_{r_{1,G}} - X_{r_{2,G}})$
- DE/best/2: The mutation scheme uses a best vector and two weighted differences and hence the name. Equation is $V_{i,G} = X_{best,G} + F(X_{r_{1,G}} - X_{r_{2,G}}) + F(X_{r_{4,G}} - X_{r_{5,G}})$
- DE/target-to-best/1: The mutation scheme used is $V_{i,G} = X_{i,G} + F(X_{best,G} - X_{i,G}) + F(X_{r_{1,G}} - X_{r_{2,G}})$

All the algorithms in the study considered have been run 25 times for each test function and averages are tabulated in Table 2 for dimension=10, Table 3 for dimension=30, and Table 4 for dimension=50.

Table 2: Error obtained for each 20 benchmark functions and 5 DE strategies with dimension=10

	Best1	Best2	Rand1	Rand2	TTB
F1	3.23E-01	0	0.00E+000	0.00E+000	0.00E+000
F2	1.62E+04	0.00E+000	0.00E+000	7.46E-012	6.18E+002
F3	7.66E-03	0	0.00E+000	3.14E-009	1.02E-002
F4	1.11E-02	0	0.00E+000	0.00E+000	0.00E+000
F5	2.05E+01	9.09E-015	0.00E+000	0.00E+000	1.90E+001
F6	4.31E+00	0	0.00E+000	0.00E+000	5.97E-002
F7	1.01E+01	9.75E-012	6.74E-012	6.29E-011	1.56E-003
F8	2.05E+01	2.04E+001	2.04E+001	2.04E+001	2.04E+001
F9	2.38E+00	4.40E-002	9.53E-001	2.53E+000	4.12E-001
F10	2.16E+00	4.05E-001	9.43E-002	4.36E-001	6.30E-002
F11	1.78E+01	1.76E+000	6.54E+000	2.02E+001	1.67E+000
F12	1.57E+01	2.01E+001	1.60E+001	2.66E+001	7.72E+000
F13	3.21E+01	2.28E+001	1.82E+001	2.61E+001	1.14E+001
F14	5.05E+02	9.18E+002	4.11E+002	8.69E+002	2.71E+002
F15	4.98E+02	9.80E+002	2.11E+002	3.22E+002	5.20E+002
F16	6.82E-01	8.48E-001	8.49E-001	9.17E-001	6.33E-001
F17	1.62E+01	2.09E+001	1.58E+001	3.01E+001	1.27E+001
F18	2.70E+01	2.86E+001	3.00E+001	2.59E+001	2.03E+001
F19	9.11E-01	5.74E-001	4.49E-001	1.15E+000	5.51E-001
F20	2.71E+00	2.23E+000	1.85E+000	3.20E+000	2.11E+000

Table 3 tabulates the averages of algorithm/problem pair for dimension = 30.

Table 3: Error obtained for each 20 benchmark functions and 5 DE strategies with dimension=30

	Best1	Best2	Rand1	Rand2	TTB
F1	3.52E+03	2.36E-13	0.00E+00	1.39E+00	6.72E+02
F2	2.07E+07	8.23E+04	5.98E+05	4.57E+07	6.01E+06
F3	4.64E+10	2.03E+00	1.64E-01	3.86E+08	3.16E+10
F4	3.05E-07	4.23E+00	2.08E+03	5.27E+04	3.72E-01
F5	1.37E+03	1.73E-13	1.09E-13	1.57E+00	2.22E+02
F6	5.58E+02	1.65E-01	2.72E+00	5.09E+01	2.73E+02
F7	1.00E+02	7.77E+00	1.10E-10	4.81E+01	4.52E+01
F8	2.09E+01	2.09E+01	2.10E+01	2.09E+01	2.10E+01
F9	2.07E+01	1.81E+01	2.64E+01	3.80E+01	9.85E+00
F10	8.23E+02	1.61E-02	2.96E-04	5.32E+01	2.51E+02
F11	2.18E+02	1.87E+02	1.55E+02	2.20E+02	8.97E+01
F12	2.10E+02	1.98E+02	1.79E+02	2.34E+02	8.70E+01
F13	2.88E+02	2.05E+02	1.79E+02	2.33E+02	1.58E+02
F14	3.25E+03	6.63E+03	5.80E+03	6.39E+03	5.95E+03
F15	3.67E+03	6.93E+03	6.80E+03	6.87E+03	6.68E+03
F16	2.14E+00	2.34E+00	2.38E+00	2.35E+00	2.31E+00
F17	3.01E+02	2.15E+02	1.96E+02	2.83E+02	2.13E+02
F18	3.25E+02	2.34E+02	2.12E+02	2.95E+02	2.33E+02
F19	6.65E+02	1.99E+00	1.37E+00	1.79E+01	6.25E+01
F20	1.50E+01	1.25E+01	1.27E+01	1.29E+01	1.31E+01

Table 4 tabulates the averages of algorithm/problem pair for dimension = 50.

Table 4: Error obtained for each 20 benchmark functions and 5 DE strategies with dimension=50

	Best1	Best2	Rand1	Rand2	TTB
F1	2.58E+04	4.18E-13	2.09E-13	2.88E+03	1.60E+04
F2	2.75E+08	3.19E+06	2.23E+07	3.10E+08	4.50E+07
F3	3.03E+14	4.63E+01	5.59E+00	4.57E+10	1.74E+11
F4	2.80E+01	6.79E+04	7.05E+04	1.50E+05	2.96E+01
F5	5.79E+03	4.14E-13	9.74E-10	1.42E+02	2.53E+03
F6	2.96E+03	4.11E+01	5.89E+01	6.65E+02	1.23E+03
F7	9.63E+04	3.20E+01	1.50E+00	1.35E+02	1.29E+02
F8	2.11E+01	2.11E+01	2.11E+01	2.11E+01	2.11E+01
F9	4.38E+01	6.81E+01	6.40E+01	7.22E+01	2.52E+01
F10	3.99E+03	2.77E-02	1.32E-02	1.37E+03	1.56E+03
F11	6.75E+02	4.21E+02	3.35E+02	4.90E+02	3.28E+02
F12	7.20E+02	4.44E+02	3.64E+02	5.43E+02	3.41E+02
F13	8.76E+02	4.37E+02	3.72E+02	5.21E+02	4.93E+02
F14	6.81E+03	1.33E+04	1.08E+04	1.32E+04	1.26E+04
F15	6.96E+03	1.39E+04	1.34E+04	1.35E+04	1.30E+04
F16	3.23E+00	3.39E+00	3.36E+00	3.28E+00	3.39E+00
F17	8.23E+02	4.63E+02	3.96E+02	7.05E+02	5.13E+02
F18	1.09E+03	4.84E+02	4.16E+02	7.72E+02	6.35E+02
F19	2.19E+04	5.50E+00	2.03E+00	3.94E+02	5.46E+03
F20	2.50E+01	2.43E+01	2.24E+01	2.36E+01	2.50E+01

2.3. Concepts of inferential statistics

In the computational intelligence community, single problem analysis and multi-problem analyses would drastically differ. The single problem analysis provides results of running the algorithms multiple times, while the multi-problem analysis demonstrates the result per algorithm and problem pair.

Hypothesis testing [17] can be used in the field of inferential statistics to draw inferences about one or more populations from the given samples. To do so we have defined both null hypothesis H0 and the alternative hypothesis H1. For testing the hypothesis in this paper we consider the null hypothesis if there is no difference between algorithms and the alternative hypothesis if there is a difference. Significance level α is used to determine at which level the hypothesis may be rejected when applied to a statistical procedure. The p-value helps us to estimate how significant the results are. If the test value does not fall in the critical region, the decision is not to reject the null hypothesis else the decision is to reject the null hypothesis because our sample mean is far away from indicating the difference.

Sometimes parametric tests are used in analysis in cases like finding the difference between the results of two algorithms in non-random paired t-test. This test checks whether the average difference is significant (not equal to 0). For comparing the multiple algorithms, ANOVA [18] tests are commonly used statistical methods used for finding the differences. In one way ANOVA, we find one independent variable with three levels, but in two ways ANOVA we concentrate on the severity of two independent variables.

Nonparametric tests apart from ordinal data can also be extended to continuous data by ranking based transformations and making modifications to the input test data. The nonparametric tests can yield pair wise comparisons and multiple comparisons. Pairwise comparisons are used to compare two individual algorithms using an independent p-value. Thus, to compare more than two algorithms, multiple comparison tests are ideal. In the comparisons the best performing algorithm is highlighted with the application of the test. Statistical procedures used in the paper are collected in Table 5 with appropriate section numbers that describes the tests.

Table 5: Nonparametric statistical procedures performed on algorithms [31]

Type of Comparison	Procedures	Section
Pairwise comparisons	Sign test	2.4.1
	Wilcoxon test	2.4.2
Multiple comparisons	Multiple sign test	2.5.1
	Friedman test	2.5.2
	Friedman Aligned	2.5.3
	Quade test	2.5.4

For analyzing the data of the results obtained in evolutionary algorithms:

n refers to number of problems; i refers to its associated index

k refers to number of algorithms used for comparison; j refers to its associated index

d refers the difference of performance between the algorithms used.

2.4. Pairwise comparisons

The Pairwise comparisons are the simplest kind and are used to compare the performances of two algorithms when applied to a set of problems. Two tests used for pairwise comparisons are the sign test and Wilcoxon signed rank test. This section characterizes the behavior of each algorithm with every other algorithm (1x1comparison).

2.4.1. A simple procedure: Sign test

The sign test is comparing the overall performances of algorithms and counting the number of cases on which an algorithm is the overall winner. In the inferential statistics, two tailed binomial test is known as sign test [19]. If the algorithm is compared as null hypothesis out of n problems each should win $n/2$ problems. The number of wins is attributed to binomial distributions; greater the number of cases the wins is under null hypothesis distributed as $n(n/2, \sqrt{n}/2)$, it allows for the use of z-test. If the number of wins is at least $n/2 + 1.96.\sqrt{n}/2$ then the algorithm is better with $p < 0.05$.

Table 6 shows the number of wins needed to achieve α=0.05 and α=0.1 levels of significance. Tied matches should be counted by splitting evenly between the two algorithms. For odd number of ties one must be ignored.

Table 6: Critical number of wins needed at α=0.05 and α=0.1 for Sign test [31]

#Cases	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
α = 0.05	5	6	7	7	8	9	9	10	10	11	12	12	13	13	14	15	15	16	17	18	18
α = 0.1	5	6	6	7	7	8	9	9	10	10	11	12	12	13	13	14	14	15	16	16	17

2.4.2. Wilcoxon signed ranks test

Wilcoxon signed ranks test is used for answering if there are two samples from two different populations. This is a nonparametric test used in hypothesis testing situations, which has two sample designs and analogous to the paired t-test;

Let d_i as the difference between the performances of the two algorithms on i^{th} out of n problems. Differences are ranked based on absolute values; in case of any ties, take the average of ranks with same differences and assign [20].

Wilcoxon's test [31] is calculated as R^+ be the sum of ranks for the problems where first algorithm outperformed the second and R^- is the sum of ranks for the opposite. When d_i=0, ranks are separated evenly among the sums; the odd number among them is ignored.

$$R^+ = \sum_{d_{i>0}} rank(\, d_i) + \frac{1}{2} \sum_{d_{i=0}} rank(\, d_i)$$

$$R^- = \sum_{d_{i<0}} rank(\, d_i) + \frac{1}{2} \sum_{d_{i=0}} rank(\, d_i)$$

If T is less than or equal to the value of the distribution for n degrees of freedom then the null hypothesis is rejected. The Wilcoxon signed rank test assumes that the greater differences are only counted but ignores the absolute magnitudes. This test is safer as it does not assume normal distributions; this test has difference d_i. It should not be rounded to one or two decimals, since it would decrease the power of test in case of more number of such differences.

2.5. Multiple comparisons

In most situations statistical procedures are frequently requested in the joint analysis of results achieved by various algorithms. In this method each block represents the results obtained over a particular problem. One block here represents three or more subjects or results. In the analysis of pairwise comparison, an accumulated error is obtained from the conclusion that involves more than one pairwise comparison and its combination. The family wise error rate

(FWER) [31], defined as probability of making one or more false discoveries among the entire hypothesis when performing multiple pairwise tests. Therefore, a pairwise comparison test such as Wilcoxon test should not be used to conduct various comparisons with a set of algorithms, because FWER is not controlled.

First we get to learn about the sign test for multiple comparisons. The multiple sign method is not very effective for finding the differences between the algorithms. Next the well know procedures that help in testing more than two samples is the Friedman test and its advanced versions Friedman aligned ranks test and the Quade test [31].

2.5.1. Multiple sign test

This test allows us to highlight the difference in the performances of algorithms of statistical difference when compared to the control algorithm. The procedure proposed in [21, 22] is as follows:

1. Let $X_{i,1}$and $X_{i,j}$ are the performances of the control and the j^{th} algorithm in the i^{th} problem.
2. Computing the difference with the following equation d_{ij}=$X_{i,j}$ -$X_{i,1}$.
3. Let r_j equal the number of differences, d_{ij} that has less frequently occurring sign.
4. Let $M1$ be the median response of a sample of results of control algorithm and Mj be the median sample of i^{th} algorithm.
5. For testing $\boldsymbol{Ho}$: $\boldsymbol{Mj} \; >= \; \boldsymbol{M1}$ against $\boldsymbol{H1}$: $\boldsymbol{Mj} \; < \; \boldsymbol{M1}$, reject Ho if the number of minus signs is less than or equal to the critical value of R_j appearing in [33] for $k-1, n$.
6. For testing $\boldsymbol{Ho}$: $\boldsymbol{Mj} \; <= \; \boldsymbol{M1}$ against $\boldsymbol{H1}$: $\boldsymbol{Mj} \; > \boldsymbol{M1}$, reject Ho if the number of plus signs is less than or equal to the critical value of R_j appearing in [33] for $k-1, n$.

In fact, it is a possible argue that if the number of methods (algorithms) is reduced in the comparison, R_j value may be changed (increased) and we would detect significant differences of a few algorithms as expected with the control algorithm. It means that the rejection of pairwise hypothesis with the control depends on the rest of the algorithms in the comparison.

2.5.2. Friedman test

Friedman Test [23, 24], the two-way analysis of variance of ranks is nonparametric analog of parametric two way analysis of variance. It answers the questions of a set of k samples (k>=2). Samples represent the population. The Friedman test is analogous to the repeated measures ANOVA in nonparametric procedure. Therefore, it aims at finding the significant differences between two or more algorithms.

The null hypothesis states the equality of medians between the populations and the alternative hypothesis states the negation of the null hypothesis. Before calculating the test statistic, first the results are converted to ranks and are computed as follows:

1. Gather observed results for each algorithm/problem pair.
2. For each problem i, rank values from 1 for good result and k for poor result. Denote these ranks as r_i^j $(1 <= j <= k)$.
3. For each algorithm j, average the ranks obtained in all problems to get the final rank that is $R_j = \frac{1}{n} \sum_i r_i^j$

Therefore, it ranks algorithms for each problem individually; the algorithm with good performance is ranked as 1 and the next best is ranked as 2. If in case there is a tie we need to compute average ranks. The Friedman statistic is computed as $F_f = \frac{12n}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right]$ that is distributed according to a χ^2 distribution with $k-1$ degrees of freedom.

The Friedman test allows for intra-set comparisons only. When number of algorithms is small for comparison, this may pose a disadvantage since inter-set comparisons may be meaningless. In such cases, comparability among problems is desirable.

2.5.3. Friedman aligned ranks method

For Friedman aligned ranks test [25], a value of location is computed as the average performance by all algorithms in each problem. The difference between the performance and the value of location is obtained. This is repeated in each combination of algorithms and problems. The results of the aligned observations keep their identities with respect to problem and combination of algorithms which are ranked from 1 to k times of n. The ranks assigned to the aligned observations are called aligned ranks.

The test statistic of Friedman aligned ranks test can be defined as

$F_{AR}=\frac{(k-1)\left[\sum_{j=1}^{k} R_j^2-\left(kn^2/4\right)(kn+1)^2\right]}{[kn(kn+1)(2kn+1)/6]-(1/k)\sum_{i=1}^{n} R_i^2}$ where R_i is equal to the rank total of i^{th} problem and R_j is the rank total of j^{th} algorithm. The test statistic F_{AR} is compared with χ^2 distribution with $k-1$ degrees of freedom.

2.5.4. Quade test

The last test in performing multiple comparisons is Quade test [32]. The procedure for the Quade test is as follows:

1. Finding the ranks r_i^j as the same way done in the Friedman test.
2. Let X_i^j be the original values of performance of algorithms.
3. Ranks are assigned to the problems according to the size of the sample range in each problem. Sample range for the problem i is the difference between two extreme observations within that problem. Where Sample Range $= max\, X_i^j - min\, X_i^j$. There

are n number of ranks for n number of problems. Assign best rank to the smallest range and so on such that the highest range will get a rank of n. Assign average of ranks for any ties.

4. Let Q_i denotes the ranks for sample ranges for problems $1,2,3,4 \ldots \ldots n$ respectively.
5. Let S_i^j be the statistic that represents the relative size of each observation within the problem. The equation is S_i^j=$Q_i[r_i^j - \frac{k+1}{2}]$

Also, S_j is sum ofS_i^j's for each algorithm where $j = 1,2,3, \ldots. k$.

6. To establish a relationship with Friedman test, rankings without average adjusting is used with W_i^j=$Q_i[r_i^j]$

The average ranking for the jth algorithm, T_j is given as $\frac{W_j}{n(n+1)/2}$ and where $W_j$is the sum of W_i^j's for each algorithm where $j = 1,2,3, \ldots. k$.

7. Definitions required for computing the test statistic F_Q are

$A = \frac{n(n+1)(2n+1)k(k+1)(k-1)}{72}$ and $B = \frac{1}{n}\sum_{j=1}^{k} S_j^2$

The test statistic F_Q is $F_Q = \frac{(n-1)B}{A-B}$, which is distributed according to the F-distribution with $k-1$ and $(k-1)(n-1)$ degrees of freedom. Note that when computing the statistic, if A=B then p-value is $(1/k!)^{n-1}$.

3. DESCRIPTION OF TESTS

Now-a-days there are many tools available to evaluate complex statistical and engineering procedures. Tools use the appropriate statistical and engineering macro functions and then display the results in the output table. Some tools generate charts in addition to output tables.

In this paper the tools used for the evaluation of nonparametric pairwise and multiple comparisons is MS-Excel [15] and MATLAB [16]. Both these tools provide many statistical, financial, engineering functions some are built in and others become available through customization.

Statistical procedures analyzed in MS-Excel are:

3.1. The Sign test

3.2. Multiple Sign test

Procedures that are analyzed using MATLAB are:

3.3. Wilcoxon Sign test

3.4. Friedman test

3.5. Friedman aligned ranks test

3.6. Quade test

3.1. Description of Sign test

In this pairwise comparison, the performances of every algorithm are compared with the performances of every other algorithm.

1. Imported data to Excel worksheet in the form of nxk. where n= number of rows (performances of the algorithms) and k = number of columns (number of algorithms)

2. Chosen one algorithm (Column*1*) and compared its performances with the performances of the other algorithms (Column*2*, Column*3*…….. Column*k*) separately. Algorithm with the best performance (lesser value) is given a score of 1 and using the count function the number of wins for both algorithms is tabulated.
3. Step 2 is repeated for all other algorithms similar to Column*1*.
4. If both algorithms compared are assumed to be under null hypothesis, each should win *n/2* out of *n*.
5. To reject null hypothesis, the number of wins must be greater than or equal to:
   - 18 at 0.05 significance level for n=25
   - 17 at 0.1 significance level for n=25
   - 15 at 0.05 significance level for n=20
   - 14 at 0.1 significance level for n=20

3.2. Description of Wilcoxon sign test (1x1 Comparison)

Tool used for Wilcoxon test is MATLAB.

1. Data is supplied to MATLAB from Excel worksheet and saved as Matrix m x n form
2. $[p, h, stats] = signrank(MatrixName(Column\#), MatrixName(Column\#))$ is a prebuilt function used to calculate the p-value, h value and statistic value. Where p-value describes the significant differences between comparison algorithms, h=0 describes the equality of two algorithms that is supporting null hypothesis, and h=1 describes the rejection of null hypothesis. Signrank is the function that uses the required columns from the matrix for comparison.

3.3. Description of Multiple sign test (1xn Comparison)

Used MS-Excel to perform multiple comparison Sign test.

1. Imported data to Excel worksheet in the form of *n* x *k*. where n= number of rows (performances of the algorithms) and *k* = number of columns (number of algorithms)
2. Chosen one algorithm (Column) as a control and subtracted its performances from the performances of every other algorithm to form a new matrix that is *n* x *k-1*.
3. For every column in the new matrix that is *n* x *(k-1)* number of minus and plus signs are tabulated.
4. To test for null hypothesis, critical value R_j is considered from [33] for a given *n* and *k-1* values.
   - For *n*=25 and *k-1*=8 are

   R_j=5, for 0.05 level of significance.

   R_j=6, for 0.1 level of significance

   - For *n*=20 and *k-1*=4 are

   R_j=4, for 0.05 level of significance.

   R_j=5, for 0.1 level of significance.
5. Algorithms with number of minus or plus signs less than or equal to the critical value is considered as significantly different.

3.4. Description of Friedman Test

Used MATLAB to perform statistical analysis

1. Read data from Excel to MATLAB.
2. $FriedmanTest$ function is called in the command window that results in Friedman statistic value and Ranks of the Algorithms.
3. p-value is calculated from the Statistic calculator.
4. Methods and variables involved in Friedman aligned Ranks test are:

- $RawData$: Data read from Excel worksheet is stored.
- $[n, k]$: Size of data that is n x k.
- $RankOfTheProblems$: used to get ranks for the performances in a problem from$RawData$. Equal ranks are controlled.
- $MeanRj$: It is the average of all ranks in an algorithm. Repeated for all other algorithms.
- $FStats$: used to get the statistic value for Friedman test.

5. Hypothesis testing: If $F > \chi^2_{0.05}$ at a significance level of 0.05 or $F > \chi^2_{0.1}$ at a significance level of 0.1 for $k - 1$ degrees of freedom then reject null hypothesis, else support null hypothesis.

3.5. Description of Friedman aligned ranks test

Used MATLAB to perform the statistical analysis.

1. Read data from Excel to MATLAB.
2. $FriedmanAllignedTest$ function is called in the command window that results in Friedman statistic value and ranks of the algorithms.
3. p-value is calculated from the Statistic calculator.
4. Methods and variables involved in Friedman aligned Ranks test:
   - $RawData$: Data is read from the excel worksheet.
   - $[n, k]$: Size of data that is n x k.
   - $AvgOfProblems$: used to get the average of problems.
   - $DiffData(i, j)$: Difference of each cell of the problem and average of that problem and in the form of n x k.

- $RankOfTheProblems()$: Ranking $DiffData$ (whole table) with rank 1 for the least value and rank $n * k$ for the highest value. Same ranks are controlled in this method.
- $FARStats$: used to get the statistic value for Friedman aligned ranks test
- $MeanRanks()$: used to get the mean rank of algorithm for all problems.

5. Hypothesis testing: If $F_{AR} > \chi^2_{0.05}$ at a significance level of 0.05 or $F_{AR} > \chi^2_{0.1}$ at a significance level of 0.1 for $k - 1$ degrees of freedom then reject null hypothesis, else support null hypothesis.

3.6. Description of Quade test

Used MATLAB to perform statistical analysis.

1. Read data from Excel to MATLAB.
2. $QuadeTest$ Function is called in the command window that results in Quade statistic value, and ranks of the algorithms.
3. p-value is calculated from the Statistic calculator.
4. Methods and variables involved in Friedman aligned ranks test:
    - $RawData$: Data is read from the excel worksheet.
    - $[n, k]$: Size of data that is n x k.
    - $MinValueRow$: Gives the minimum value for every problem (row).
    - $MaxValueRow$: Gives the maximum value for every problem (row).
    - $DiffMaxMinValue$: Difference of maximum and minimum values for every problem.
    - $RankOfDiff$: used to get the ranks of $DiffMaxMinValue$.

- $RankOfTheProblems$: used to get ranks for the performances in a problem. Similarly to all problems. Equal ranks are controlled.
- $StatsSij()$: Ranking of $RankOfTheProblems$ with average adjusting.
- $SumOfStatsSj$: Sum of the ranks of problems from $StatsSij()$ for every algorithm.
- $StatsWij()$: Ranking of $RankOfTheProblems$ without average adjusting.
- $SumOfStatsWj$: Sum of the ranks of problems from $StatsWij()$ for every algorithm.
- $StatsTj$: Adjusted ranks of $SumOfStatsWj$. $StatsTj$ ranks decide the best algorithm out of all.
- $FQStats$: used to get the statistic value for $Quade$ test.

5. Hypothesis testing: If $F_Q > F_{0.05}$ at a significance level of 0.05 or $F_{AR} > \chi^2_{0.1}$ at a significance level of 0.1 for $k-1$ and $(k-1)(n-1)$ degrees of freedom then reject null hypothesis else support null hypothesis.

4. RESULTS AND DISCUSSIONS

4.1. Test case 1: Table 1 is considered for the statistical analysis

Data considered for statistical analysis is shown in Table 1.

Number of problems *(n)* = 25.

Number of algorithms *(k)* = 9.

Dimension = 10.

4.1.1. Application of Sign test

In this experimental study, performing a sign test to compare the results of an algorithm is simple. It only requires the number of wins achieved by an algorithm with the comparison algorithms. Table 7 summarizes the winning algorithms count with comparison algorithms.

Table 7: Wins of an algorithm over rest of the algorithms for Sign test on Table 1

	PSO	IPOP	CHC	SSGA	SS-BLX	SS-Arit	DE-Bin	DE-Exp	SaDE
PSO	-	8	13	7	7	8	4	3	5
α=									
IPOP	17	-	17	16	12	14	11	11	11
α=	0.1		0.1						
CHC	12	8	-	10	8	9	6	7	5
α=									
SSGA	18	9	15	-	10	12	6	6	7
α=	0.05								
SS-BLx	18	13	17	15	-	17	9	9	10
α=	0.05		0.1			0.1			
SS-Arit	17	11	16	13	8	-	7	7	8
α=	0.1								
DE-Bin	21	14	19	19	16	18	-	10	14
α=	0.05		0.05	0.05		0.05			
De-Exp	22	14	18	19	16	18	15	-	16
α=	0.05		0.05	0.05		0.05			
SaDE	20	14	20	18	15	17	11	9	-
α=	0.05		0.05	0.05		0.1			

IPOP wins over PSO and CHC with detected difference of 0.1 when compared with the remaining 8 algorithms. SSGA wins over PSO with a difference of 0.05, SS-BLX over PSO with a difference of 0.05 and CHC, SS-Arit with a difference of 0.1, SS-Arit over PSO with a difference of 0.1, DE-Bin and DE-Exp over PSO, CHC, SSGA, SS-Arit with a difference of 0.05 and SaDE algorithm over PSO, CHC, SSGA with 0.05 and SS-Arit with 0.1.

4.1.2. Application of Wilcoxon test

When using Wilcoxon test in our study, Table 8 shows that R+, R- and p-values computed for all the pairwise comparisons concerning PSO. As the table states, PSO shows a significant improvement over DE-Bin, DE-Exp and SaDE with a level of significance $\alpha=0.05$.

Table 8: Ranks and p-value of PSO over other algorithms for Table 1

PSO versus	R+	R-	p-value
IPOP	209	116	0.2109
CHC	121	204	0.2641
SSGA	203	122	0.2758
SS-BLX	225	100	1.6817
SS-Arit	221	104	0.1155
DE-Bin	263	62	0.0068
DE-Exp	265	60	0.0058
SaDE	251	74	0.0173

As Table 9 states, IPOP shows a significant improvement over CHC, DE-Bin, DE-Exp and SaDE with a level of significance $\alpha=0.1$.

Table 9: Ranks and p-value of IPOP over other algorithms for Table 1

IPOP versus	R+	R-	p-value
PSO	116	209	0.2109
CHC	92	233	0.0578
SSGA	119	206	0.2418
SS-BLX	164	161	0.9678
SS-Arit	143	182	0.5998
DE-Bin	228	97	0.078
DE-Exp	229	96	0.0736
SaDE	226	99	0.0875

As Table 10 states, CHC shows a significant improvement over SSGA, SS-BLX, SS-Arit, DE-Bin, DE-Exp and SaDE with a level of significance α=0.05 and over IPOP with α=0.1.

Table 10: Ranks and p-value of CHC over other algorithms for Table 1

CHC versus	R+	R-	p-value
PSO	204	121	0.2641
IPOP	233	92	0.0578
SSGA	248	77	0.0214
SS-BLX	267	58	0.0049
SS-Arit	261	64	0.008
DE-Bin	277	48	0.0021
DE-Exp	280	45	0.0016
SaDE	290	35	6.02E-04

As Table 11 states, SSGA shows a significant improvement over CHC, DE-Bin, DE-Exp and SaDE with a level of significance α=0.05.

Table 11: Ranks and p-value of SSGA over other algorithms for Table 1

SSGA versus	R+	R-	p-value
PSO	122	203	0.2758
IPOP	206	119	0.2418
CHC	77	248	0.0214
SS-BLX	204	121	0.2641
SS-Arit	187	138	0.5098
DE-Bin	256	69	0.0119
DE-Exp	265	60	0.0058
SaDE	258	67	0.0102

As Table 12 states, SS-BLX shows a significant improvement over CHC with a level of significance α=0.05 and over SaDE with a level of significance α=0.1.

Table 12: Ranks and p-value of SS-BLX over other algorithms for Table 1

SS-BLX versus	R+	R-	p-value
PSO	100	225	1.6817
IPOP	161	164	0.9678
CHC	58	267	0.0049
SSGA	121	204	0.2641
SS-Arit	202	123	0.2879
DE-Bin	222	103	0.1094
DE-Exp	220	105	0.1218
SaDE	228	97	0.078

As Table 13 states, SS-Arit shows a significant improvement over CHC, SS-BLX, DE-Bin, DE-Exp and SaDE with a level of significance α=0.05.

Table 13: Ranks and p-value of SS-Arit over other algorithms for Table 1

SS-Arit versus	R+	R-	p-value
PSO	104	221	0.1155
IPOP	182	143	0.5998
CHC	64	261	0.008
SSGA	138	187	0.5098
SS-BLX	202	123	0.2879
DE-Bin	239	86	0.0396
DE-Exp	247	78	0.023
SaDE	236	89	0.048

As Table 14 states, DE-Bin shows a significant improvement over PSO, CHC, SSGA, SS-Arit, with a level of significance α=0.05 and over IPOP with a level of significance α=0.1.

Table 14: Ranks and p-value of DE-Bin over other algorithms for Table 1

DE-Bin versus	R+	R-	p-value
PSO	62	263	0.0068
IPOP	97	228	0.078
CHC	48	277	0.0021
SSGA	69	256	0.0119
SS-BLX	222	103	0.1094
SS-Arit	86	239	0.0396
DE-Exp	222	103	0.1094
SaDE	154	171	0.8191

As Table 15 states, DE-Exp shows a significant improvement over PSO, CHC, SSGA, SS-Arit with a level of significance α=0.05 and over IPOP with a level of significance α=0.1.

Table 15: Ranks and p-value of DE-Exp over other algorithms for Table 1

DE-Exp versus	R+	R-	p-value
PSO	60	265	0.0058
IPOP	96	229	0.0736
CHC	45	280	0.0016
SSGA	60	265	0.0058
SS-BLX	220	105	0.1218
SS-Arit	78	247	0.023
DE-Bin	103	222	0.1094
SaDE	115	210	0.2012

As Table 16 states, SaDE shows a significant improvement over PSO, CHC, SSGA, SS-Arit with a level of significance α=0.05 and over IPOP,SS-BLX with a level of significance α=0.1.

Table 16: Ranks and p-value of SaDE over other algorithms for Table 1

SaDE versus	R+	R-	p-value
PSO	74	251	0.0173
IPOP	99	226	0.0875
CHC	35	290	6.02E-04
SSGA	67	258	0.0102
SS-BLX	228	97	0.078
SS-Arit	89	236	0.048
DE-Bin	171	154	0.8191
DE-Exp	210	115	0.2012

4.1.3. Application of Multiple sign test

Critical values are taken from [33] where $\boldsymbol{R_j}$=5 for α=0.05 and $\boldsymbol{R_j}$= 6 for α=0.1. Table 17 shows the number of wins and losses of control algorithm with the rest of the algorithms.

Table 17: Number of wins and losses by control algorithm over rest of them using Multiple sign test for Table 1

	CONTROL ALGORITHM																	
	PSO		**IPOP**		**CHC**		**SSGA**		**SS-BLX**		**SS-Arit**		**DE-Bin**		**DE-Exp**		**SaDE**	
	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-
PSO	0	0	17	8	12	13	18	7	18	7	17	8	21	4	22	3	19	6
IPOP	8	17	0	0	8	17	9	16	13	12	11	14	14	11	14	11	14	11
CHC	13	12	17	8	0	0	15	10	17	8	16	9	19	6	18	7	20	5
SSGA	7	18	16	9	10	15	0	0	15	10	13	12	19	6	18	7	18	7
SS-BLX	7	18	12	13	8	17	10	15	0	0	8	17	16	9	16	9	15	10
SS-Arit	8	17	14	11	9	16	12	13	17	8	0	0	18	7	17	8	17	8
DE-Bin	4	21	11	14	6	19	6	19	9	16	7	18	0	0	15	10	12	13
DE-Exp	3	22	11	14	7	18	7	18	9	16	8	17	10	15	0	0	9	16
SaDE	6	19	11	14	5	20	7	18	10	15	8	17	13	12	16	9	0	0

1. Labeling PSO as a control algorithm, we may reuse the results of Table 17 for applying multiple sign test. Considering Ho: Mj <= M1 against H1: Mj > M1 hypothesis testing, the algorithms with number of plus signs less than or equal to a

critical value of 5 is DE-Bin and DE-Exp and less than or equal to a critical value of 6 is SaDE. We may conclude that PSO is significantly different than these two.

2. Labeling IPOP as a control algorithm, results in Table 17 supports null hypothesis when compared with all other algorithms. It does not fall in to the critical region and hence IPOP is not significantly different than all other algorithms.
3. Labeling CHC as a control algorithm, we may reuse the results of Table 17 for applying multiple sign test. Considering Ho: Mj <= M1 against H1: Mj > M1 hypothesis testing, the algorithms with number of plus signs less than or equal to a critical value of 5 is DE-Bin and DE-Exp and less than or equal to a critical value of 6 is SaDE. We may conclude that PSO has better performance than these three.
4. Labeling SSGA as a control algorithm, we may reuse the results of Table 17 for applying multiple sign test. Considering Ho: Mj <= M1 against H1: Mj > M1 hypothesis testing, the algorithms with number of plus signs less than or equal to a critical value of 6 is DE-Bin. We may conclude that SSGA has better performance than DE-Bin.
5. Labeling SS-BLX and SS-Arit as a control algorithm, results in Table 17 support null hypothesis when compared with all other algorithms. They does not fall in to the critical region with a critical value less than or equal to 6. Hence SS-BLX and SS-Arit are not significantly different than all other algorithms.
6. Labeling DE-Bin as a control algorithm, we may reuse the results of Table 17 for applying multiple sign test. Considering Ho: Mj >= M1 against H1: Mj < M1 hypothesis testing, the algorithms with number of minus signs less than or equal to a critical value of 5 is PSO and less than or equal to a critical value of 6 is CHC and

SSGA. We may conclude that DE-Bin is significantly different than PSO, CHC and SSGA.

7. Labeling DE-Exp as a control algorithm, we may reuse the results of Table 17 for applying multiple sign test. Considering Ho: Mj >= M1 against H1: Mj < M1 hypothesis testing, the algorithms with number of minus signs less than or equal to a critical value of 5 is PSO. We may conclude that DE-Exp is significantly different than PSO.
8. Similarly, labeling SaDE as a control algorithm, the algorithms with number of minus signs less than or equal to a critical value of 5 is CHC and less than or equal to a critical value of 6 is PSO. We may conclude that SaDE is significantly different than PSO, CHC.

4.1.4. Application of Friedman, Friedman aligned ranks and Quade tests

Continuing with our experimental study, the ranks of the Friedman, Friedman aligned, and Quade tests are computed for all the algorithms considered. Table 18 shows that DE-Exp as the best performing algorithm of the comparison, with a rank of 3.56, 85.74, and 3.169 for the Friedman, Friedman aligned, and Quade tests, respectively.

Table 18: Ranks, statistic value and p-value of algorithms using Friedman, Friedman aligned ranks and Quade test on Table 1

Algorithms	Friedman	Friedman Aligned	Quade
PSO	6.76	135.28	6.42154
IPOP	4.64	112.4	4.52615
CHC	6.40	159.32	7.37538
SSGA	5.64	131.72	5.96923
SS-BLX	4.68	108.2	5.13846
SS-Arit	5.48	108.76	5.60308
DE-Bin	3.80	86.72	3.48615
DE-Exp	3.56	85.48	3.16923
SaDE	4.04	87.12	3.31077
statistic	34.55	15.735625	6.99483
p-value	3.20E-05	0.0463247	4.00E-08

4.2. Test case 2: Table 2 is considered for the statistical analysis

Data considered for statistical analysis is given in Table 2.

Number of problems *(n)* = 20.

Number of algorithms *(k)* = 5

Dimension = 10.

4.2.1. Application of Sign test

Table 19 summarizes the winning count of algorithms with comparison algorithms. Best2 wins over Best1 and Rand2 with detected difference of 0.1 when compared with the remaining 4 algorithms. Rand1 wins over Best1 and Rand2 with a difference of 0.05, and TargetToBest over Best1 with a difference of 0.05.

Table 19: Wins of an algorithm over rest of the algorithms for Sign test on Table 2

Sign Table	Best1	Best2	Rand1	Rand2	TargetToBest
Best1 wins	-	6	3	8	2
α=					
Best2 wins	14	-	7	14	7
α=	0.1			0.1	
Rand1 wins	17	12	-	17	9
α=	0.05			0.05	
Rand2 wins	12	5	3	-	7
α=					
TargetToBest	18	13	11	13	-
α=	0.05				

4.2.2. Application of Wilcoxon test

When using Wilcoxon test in our study, Table 20 shows that R+, R- and p-values computed for all the pairwise comparisons concerning the Best1 algorithm. As the table states, Best1 is significantly different than Rand1 and TargetToBest with a level of significance α=0.05.

Table 20: Ranks and p-value of Best1 over other algorithms for Table 2

Best1 versus	R+	R-	p-value
Best2	136	74	0.2471
Rand1	189	21	0.0017
Rand2	126	84	0.433
TargetToBest	191	19	0.0013

As Table 21 states, Best2 supports null hypothesis and is not significantly different than any of the comparison algorithms as p-value is greater than 0.1.

Table 21: Ranks and p-value of Best2 over other algorithms for Table 2

Best2 versus	R+	R-	p-value
Best1	74	136	0.2471
Rand1	84	36	0.1876
Rand2	45	108	0.1359
TargetToBest	121	50	0.1221

As Table 22 states, Rand1 is significantly different than Best1 and Rand2 with a level of significance α=0.05 as p-value is less than 0.05.

Table 22: Ranks and p-value of Rand1 over other algorithms for Table 2

Rand1 versus	R+	R-	p-value
Best1	21	189	0.0017
Best2	36	84	0.1876
Rand2	10	126	0.0027
TargetToBest	99	72	0.5566

As Table 23 states, Rand2 is significantly different than Rand1 with a level of significance α=0.05.

Table 23: Ranks and p-value of Rand2 over other algorithms for Table 2

Rand2 versus	R+	R-	p-value
Best1	84	126	0.433
Best2	108	45	0.1359
Rand1	126	10	0.0027
TargetToBest	115	56	0.1989

As Table 24 states, TargetToBest is significantly different than Best1 with a level of significance α=0.05.

Table 24: Ranks and p-value of TargetToBest over other algorithms for Table 2

Target to Best versus	R+	R-	p-value
Best1	19	191	0.0013
Best2	50	121	0.1221
Rand1	72	99	0.5566
Rand2	56	115	0.1989

4.2.3. Application of Multiple sign test

Critical values are taken from [33] where $\boldsymbol{R_j}$=4 for α=0.05 and $\boldsymbol{R_j}$= 5 for α=0.1. Table 25 tabulates the number of wins and losses by control algorithm on rest of the algorithms.

Table 25: Number of wins and losses by control algorithm over rest of them using Multiple sign test for Table 2

	Control Algorithm									
	Best1		**Best2**		**Rand1**		**Rand2**		**TargetToBest**	
	+	-	+	-	+	-	+	-	+	-
Best1	0	0	14	6	17	3	12	8	18	2
Best2	6	14	0	0	10	5	4	13	12	6
Rand1	3	17	5	10	0	0	1	15	10	8
Rand2	8	12	13	4	15	1	0	0	12	6
TargetToBest	2	18	6	12	8	10	6	12	0	0

1. Labeling Best1 as a control algorithm, we may reuse the results of Table 25 for applying multiple sign test. Considering **Ho: Mj <= M1 against H1: Mj > M1** hypothesis testing, the algorithms with number of plus signs less than or equal to a critical value of 4 is Rand1 and TargetToBest at a level of 0.05. We may conclude that Best1 is significantly different than these two.
2. Labeling Best2 as a control algorithm, we may reuse the results of Table 25 for applying multiple sign test. Considering **Ho: Mj >= M1 against H1: Mj < M1** hypothesis testing, the algorithms with number of minus signs less than or equal to a critical value of 4 is Rand2 at a level of 0.05. We may conclude that Best2 is significantly different than Rand2.

3. Labeling Rand1 as a control algorithm, we may reuse the results of Table 25 for applying multiple sign test. Considering **Ho: Mj >= M1 against H1: Mj < M1** hypothesis testing, the algorithms with number of minus signs less than or equal to a critical value of 4 is Best1 and Rand2 at a level of 0.05 and minus signs less than or equal to 5 is Best2 at a level of 0.1. We may conclude that Rand1 is significantly different than these three.
4. Labeling Rand2 as a control algorithm, we may reuse the results of Table 25 for applying multiple sign test. Considering **Ho: Mj <= M1 against H1: Mj > M1** hypothesis testing, the algorithms with number of plus signs less than or equal to a critical value of 4 is Best2 and Rand1 at a level of 0.05. We may conclude that Rand2 is significantly different than these two.
5. Labeling TargetToBest as a control algorithm, we may reuse the results of Table 25 for applying multiple sign test. Considering **Ho: Mj >= M1 against H1: Mj < M1** hypothesis testing, the algorithms with number of minus signs less than or equal to a critical value of 4 is Best1 at a level of 0.05. We may conclude that TargetToBest is significantly different than Best1.

4.2.4. Application of Friedman, Friedman aligned ranks and Quade tests

Continuing with our experimental study, the ranks of the Friedman, Friedman aligned, and Quade tests can be computed for all the algorithms considered.

Following are the guidelines exposed and the results are tabulated in the below table. Table 26 shows that Rand1 algorithm as the best performing algorithm of the comparison, with a rank of 2.2, 35.975, and 2.1 for the Friedman, Friedman aligned, and Quade tests, respectively.

Table 26: Ranks, statistic value and p-value of algorithms using Friedman, Friedman aligned ranks and Quade test on Table 2

Algorithms	Friedman	Friedman Aligned	Quade
Best1	4.05	66.3	3.97
Best2	2.825	53.75	3.09
Rand1	2.2	35.975	2.1
Rand2	3.6	57.975	3.54
TargetToBest	2.225	38.5	2.27
statistic	17.07	14.49947	4.71482
p-value	0.00187334	0.00586	0.001879

The p-values computed through the statistics of each of the tests considered (0.00187334, 0.00586, and 0.001879) strongly suggest the existence of significant differences among the algorithms considered. These values also suggest that at which probability level the null hypothesis can be rejected.

4.3. Test case 3: Table 3 is considered for the statistical analysis

Data considered for statistical analysis is shown in Table 3.

Number of problems *(n)* = 20.

Number of algorithms *(k)* = 5.

Dimension = 30.

4.3.1. Application of Sign test

Table 27 summarizes the winning algorithms with comparison algorithms. Best2 wins over Best1 and Rand2 with detected difference of 0.05 when compared with the remaining 4 algorithms.

Rand1 wins over Best1 with a difference of 0.1 and Rand2 with a difference of 0.05, and TargetToBest over Best1 with a difference of 0.05. Hence, we could reject null hypothesis and conclude that there is a significant difference between the compared algorithms.

Table 27: Wins of an algorithm over rest of the algorithms for Sign test on Table 3

	Best1	Best2	Rand1	Rand2	TargetToBest
Best1 wins	-	5	6	8	5
α=					
Best2 wins	15	-	7	17	10
α=	0.05			0.05	
Rand1 wins	14	13	-	18	12
α=	0.1			0.05	
Rand2 wins	12	3	2	-	8
α=					
TargetToBest	15	10	8	12	-
α=	0.05				

4.3.2. Application of Wilcoxon test

When using Wilcoxon test in our study, the rank values that is R+ and R- and p-values are noted down in the below tables.

Table 28 shows the ranks and p-value of pairwise comparisons concerning Best1 algorithm when the Wilcoxon signed ranks test is used for the statistical analysis. As the table states, Best1 is significantly different than Best2 and TargetToBest with a level of significance α=0.05 and Rand1 with a level of 0.1.

Table 28: Ranks and p-value of Best1 over other algorithms for Table 3

Best1 versus	R+	R-	p-value
Best2	169	41	0.0169
Rand1	155	55	0.062
Rand2	124	86	0.4781
TargetToBest	170	40	0.0152

As Table 29 states, Best2 is significantly different than Best1 and Rand2 with a level of significance α=0.05.

Table 29: Ranks and p-value of Best2 over other algorithms for Table 3

Best2 versus	R+	R-	p-value
Best1	41	169	0.0169
Rand1	137	73	0.2322
Rand2	32	178	0.0064
TargetToBest	88	122	0.5257

As Table 30 states, Rand1 is significantly different than Best1 with level of significance α=0.1 and Rand2 with a level of significance α=0.05 as p-value is less than 0.05.

Table 30: Ranks and p-value of Rand1 over other algorithms for Table 3

Rand1 versus	R+	R-	p-value
Best1	55	155	0.062
Best2	73	137	0.2322
Rand2	3	207	0.000140
TargetToBest	65	145	0.1354

As Table 31 states, Rand2 is significantly different than Rand1 with a level of significance α=0.05.

Table 31: Ranks and p-value of Rand2 over other algorithms for Table 3

Rand2 versus	R+	R-	p-value
Best1	86	124	0.4781
Best2	178	32	0.0064
Rand1	207	3	0.000140
TargetToBest	121	89	0.5503

As Table 32 states, TargetToBest is significantly different than Best1 with a level of significance α=0.05.

Table 32: Ranks and p-value of TargetToBest over other algorithms for Table 3

Target to Best versus	R+	R-	p-value
Best1	40	170	0.0152
Best2	122	88	0.5257
Rand1	145	65	0.1354
Rand2	89	121	0.5503

4.3.3. Applying Multiple sign test

Critical values are taken from [33] where $\boldsymbol{R_j}$=4 for α=0.05 and $\boldsymbol{R_j}$= 5 for α=0.1. Table 33 tabulates the number of wins and losses of control algorithm with the rest of the algorithms.

Table 33: Number of wins and losses by control algorithm over rest of them using Multiple sign test for Table 3

	Control Algorithm									
	Best1		Best2		Rand1		Rand2		TargetToBest	
	+	-	+	-	+	-	+	-	+	-
Best1	0	0	15	5	14	5	12	8	15	5
Best2	5	15	0	0	13	7	3	17	10	10
Rand1	6	14	7	13	0	0	2	18	8	12
Rand2	8	12	17	3	18	2	0	0	12	8
TargetToBest	5	15	10	10	12	8	8	12	0	0

1. Labeling Best1 as a control algorithm, we may reuse the results of Table 33 for applying multiple sign test. Considering **Ho: Mj <= M1 against H1: Mj > M1** hypothesis testing, the algorithms with number of plus signs less than or equal to a critical value of 5 is Best2 and TargetToBest at a level of 0.1. We may conclude that Best1 is significantly different than these two.
2. Labeling Best2 as a control algorithm, we may reuse the results of Table 33 for applying multiple sign test. Considering **Ho: Mj >= M1 against H1: Mj < M1** hypothesis testing, the algorithms with number of minus signs less than or equal to a critical value of 4 is Rand2 at a level of 0.05 and minus signs less than or equal to a critical value of 5 is Best1 at a level of 0.1. We may conclude that Best2 is significantly different than Rand2 and Best1.
3. Labeling Rand1 as a control algorithm, we may reuse the results of Table 33 for applying multiple sign test. Considering **Ho: Mj >= M1 against H1: Mj < M1** hypothesis testing, the algorithms with number of minus signs less than or equal to a critical value of 4 is Rand2 at a level of 0.05. We may conclude that Rand1 is significantly different than Rand2.
4. Labeling Rand2 as a control algorithm, we may reuse the results of Table 33 for applying multiple sign test. Considering **Ho: Mj <= M1 against H1: Mj > M1**

hypothesis testing, the algorithms with number of plus signs less than or equal to a critical value of 4 is Best2 and Rand1 at a level of 0.05. We may conclude that Rand2 is significantly different than these two.

5. Labeling TargetToBest as a control algorithm, we may reuse the results of Table 33 for applying multiple sign test. Considering **Ho: Mj >= M1 against H1: Mj < M1** hypothesis testing, the algorithms with number of minus signs less than or equal to a critical value of 5 is Best1 at a level of 0.1. We may conclude that TargetToBest is significantly different than Best1.

4.3.4. Applying Friedman, Friedman aligned ranks and Quade tests

Continuing with our experimental study, the ranks of the Friedman, Friedman aligned, and Quade tests can be computed for all the algorithms considered. Following are the guidelines exposed. Table 34 shows that Rand1 as the best performing algorithm of the comparison, with a rank of 2.15, 37.5, and 1.88 for the Friedman, Friedman aligned, and Quade tests, respectively.

Table 34: Ranks, statistic value and p-value of algorithms using Friedman, Friedman aligned ranks and Quade test on Table 3

Algorithms	Friedman	Friedman Aligned	Quade
Best1	3.8	67.5	3.8
Best2	2.55	40.45	2.58
Rand1	2.15	37.5	1.88
Rand2	3.75	58	3.84
TargetToBest	2.75	48.6	2.91
statistic	17.52	14.7075257	5.1914
p-value	0.001531	0.005347	0.00094445

The p-values computed through the statistics of each of the tests considered (0.001531, 0.005347, and 0.00094445) strongly suggest the existence of significant differences among the algorithms considered.

4.4. Test case 4: Table 4 is considered for the statistical analysis

Data considered for statistical analysis is given in Table 4.

Number of problems *(n)* = 20.

Number of algorithms *(k)* = 5.

Dimension = 50.

4.4.1. Application of Sign test

Table 35 summarizes the winning algorithms with comparison algorithms. Best2 wins over Best1 and Rand2 with detected difference of 0.05 when compared with rest 4 algorithms. Rand1 wins over Best1, Best2, Rand2 and TargetToBest with a difference of 0.05, and TargetToBest over Best1 with a difference of 0.05.

Table 35: Wins of an algorithm over rest of the algorithms for Sign test on Table 4

	Best1	**Best2**	**Rand1**	**Rand2**	**TargetToBest**
Best1 wins	-	5	5	6	4
α=					
Best2 wins	15	-	5	15	13
α=	0.05			0.05	
Rand1 wins	15	15	-	18	15
α=	0.05	0.05		0.05	0.05
Rand2 wins	14	5	2	-	9
α=	0.1				
TargetToBest	15	7	5	11	-
α=	0.05				

4.4.2. Applying Wilcoxon test

When using Wilcoxon test in our study, Table 36 shows that R+, R- and p-values computed for all the pairwise comparisons concerning Best1 Algorithm. As the table states, Best1 is significantly different than Best2, Rand1 and TargetToBest with a difference of α=0.05.

Table 36: Ranks and p-value of Best1 over other algorithms for Table 4

Best1 versus	R+	R-	p-value
Best2	160	50	0.04
Rand1	161	49	0.0366
Rand2	140	70	0.1913
TargetToBest	158	32	0.0112

As Table 37 states, Best2 is significantly different than Best1 and Rand2 with a level of significance α=0.05 and Rand1 at a level of 0.1.

Table 37: Ranks and p-value of Best2 over other algorithms for Table 4

Best2 versus	R+	R-	p-value
Best1	50	160	0.04
Rand1	157	53	0.0522
Rand2	26	184	0.0032
TargetToBest	63	147	0.1169

As Table 38 states, Rand1 is significantly different than Best2 with level of significance α=0.1 and Best1, Rand2, TargetToBest with a level of significance α=0.05 as p-value is less than 0.05.

Table 38: Ranks and p-value of Rand1 over other algorithms for Table 4

Rand1 versus	R+	R-	p-value
Best1	49	161	0.0366
Best2	53	157	0.0522
Rand2	3	207	0.000140
TargetToBest	44	166	0.0228

As Table 39 states, Rand2 is significantly different than Best2 and Rand1 with a level of significance α=0.05.

Table 39: Ranks and p-value of Rand2 over other algorithms for Table 4

Rand2 versus	R+	R-	p-value
Best1	70	140	0.1913
Best2	184	26	0.0032
Rand1	207	3	0.000140
TargetToBest	113	97	0.7652

As Table 40 states, TargetToBest is significantly different than Best1 and Rand1 with a level of significance α=0.05.

Table 40: Ranks and p-value of TargetToBest over other algorithms for Table 4

Target to Best versus	R+	R-	p-value
Best1	32	158	0.0112
Best2	147	63	0.1169
Rand1	166	44	0.0228
Rand2	97	113	0.7652

4.4.3. Application of Multiple sign test

Critical values are taken from [33] where $\boldsymbol{R_j}$=4 for α=0.05 and $\boldsymbol{R_j}$= 5 for α=0.1. Table 41 tabulates the wins and losses of control algorithm over rest of the other algorithms.

Table 41: Number of wins and losses by control algorithm over rest of them using Multiple sign test for Table 4

	Control Algorithm									
	Best1		Best2		Rand1		Rand2		TargetToBest	
	+	-	+	-	+	-	+	-	+	-
Best1	0	0	15	5	15	5	14	6	15	4
Best2	5	15	0	0	15	5	5	15	7	13
Rand1	5	15	6	14	0	0	2	18	6	14
Rand2	6	14	15	5	18	2	0	0	11	9
TargetToBest	4	15	13	7	15	5	9	11	0	0

1. Labeling Best1 as a control algorithm, we may reuse the results of Table 41 for applying multiple sign test. Considering **Ho: Mj <= M1 against H1: Mj > M1** hypothesis testing, the algorithms with number of plus signs less than or equal to a critical value of 4 is TargetToBest at a level of 0.05 and plus signs less than or equal to a critical value of 5 is Best2 and Rand1 at a level of 0.1. We may conclude that Best1 is significantly different than these three.
2. Labeling Best2 as a control algorithm, we may reuse the results for applying multiple sign test. Considering **Ho: Mj >= M1 against H1: Mj < M1** hypothesis testing, the algorithms with number of minus signs less than or equal to a critical value of 5 is

Best1 and Rand2 at a level of 0.1. We may conclude that Best2 is significantly different than Rand2 and Best1.

3. Labeling Rand1 as a control algorithm, we may reuse the results for applying multiple sign test. Considering **Ho: Mj >= M1 against H1: Mj < M1** hypothesis testing, the algorithms with number of minus signs less than or equal to a critical value of 4 is Rand2 at a level of 0.05 and minus signs less than 5 is Best1, Best2, TargetToBest at a level of 0.1. We may conclude that Rand1 is significantly different than these four.
4. Labeling Rand2 as a control algorithm, we may reuse the results for applying multiple sign test. Considering **Ho: Mj <= M1 against H1: Mj > M1** hypothesis testing, the algorithms with number of plus signs less than or equal to a critical value of 4 is Rand1 and plus signs less than or equal to a critical value of 5 is Best2 at a level of 0.1. We may conclude that Rand2 is significantly different than these two.
5. Labeling TargetToBest as a control algorithm, we may reuse the results for applying multiple sign test. Considering **Ho: Mj >= M1 against H1: Mj < M1** hypothesis testing, the algorithms with number of minus signs less than or equal to a critical value of 4 is Best1 at a level of 0.05. We may conclude that TargetToBest is significantly different than Best1.

4.4.4. Application of Friedman, Friedman aligned ranks and Quade tests

Continuing with our experimental study, the ranks of the Friedman, Friedman aligned, and Quade tests can be computed for all the algorithms considered, following the guidelines exposed, above table shows that Rand1 as the best performing algorithm of the comparison with a rank of 1.85, 36.85, and 1.77 for the Friedman, Friedman aligned, and Quade tests, respectively.

Table 42: Ranks, statistic value and p-value of algorithms using Friedman, Friedman aligned ranks and Quade test on Table 4

Algorithms	Friedman	Friedman Aligned	Quade
Best1	3.975	70.375	3.95
Best2	2.6	41.95	2.41
Rand1	1.85	36.85	1.77
Rand2	3.5	54	3.73
TargetToBest	3.075	49.275	3.14
statistic	21.51	14.8511977	6.531626
p-value	0.000251	0.00502	0.00014253

The p-values computed through the statistics of each of the tests considered (0.000251, 0.00502, and 0.00014253) strongly suggest the existence of significant differences among the algorithms considered.

5. CONCLUSION

In statistical analyses, parametric procedures are most commonly used that are based on assumptions. Due to the fact that assumptions are violated while performing analyses on stochastic algorithms in computational intelligence, nonparametric statistical procedures are used that are more effective, especially in multi-problem analysis. We have used wide range of tests in nonparametric statistical analysis starting from basic techniques like sign tester to complex techniques like the Friedman aligned ranks and the Quade tests.

In this paper, we used all the tests and applied on the results obtained for evolutionary swarm intelligence algorithms to find the algorithm that is significantly different than remaining algorithms in a comparison. Analysis reveals that the algorithm which is significantly different and better than remaining algorithms is same in every statistical test.

Also, to present the efficacy of the different procedures, we have implemented comprehensive case study analysis on the results with varied dimension. Application of the tests reveals that the significantly different algorithm became more powerful and tries to act as the best algorithm when results with increased dimensions are analyzed.

In the future, these tests can be applied to other engineering and research areas and leaves a choice to pick the most suitable test for their analysis.

6. REFERENCES

[1] Higgins, James J. *An introduction to modern nonparametric statistics*. Pacific Grove, CA: Brooks/Cole, 2004.

[2] García, Salvador, et al. "A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability."*Soft Computing* 13.10 (2009): 959-977.

[3] García, Salvador, et al. "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization." *Journal of Heuristics* 15.6 (2009): 617-644.

[4] Demšar, Janez. "Statistical comparisons of classifiers over multiple data sets."*The Journal of Machine Learning Research* 7 (2006): 1-30.

[5] Garcia, Salvador, and Francisco Herrera. "An Extension on" Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons." *Journal of Machine Learning Research* 9.12 (2008).

[6] Suganthan, Ponnuthurai N., et al. "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization." *KanGAL Report*2005005 (2005).

[7] Kennedy, James, and Russell Eberhart. "Particle swarm optimization."*Proceedings of IEEE international conference on neural networks*. Vol. 4. No. 2. 1995.

[8] Auger, Anne, and Nikolaus Hansen. "A restart CMA evolution strategy with increasing population size." *Evolutionary Computation, 2005. The 2005 IEEE Congress on*. Vol. 2. IEEE, 2005.

[9] Eshelman, Larry J. "The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination." *Foundations of genetic algorithms* (1990): 265-283.

[10] Eshelman, Larry J. "chapter Real-Coded Genetic Algorithms and Interval-Schemata." *Foundations of genetic algorithms* 2 (1993): 187-202.

[11] Fernandes, Carlos, and Agostinho Rosa. "A study on non-random mating and varying population size in genetic algorithms using a royal road function."*Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*. Vol. 1. IEEE, 2001.

[12] Herrera, Francisco, Manuel Lozano, and Daniel Molina. "Continuous scatter search: an analysis of the integration of some combination methods and improvement strategies." *European Journal of Operational Research* 169.2 (2006): 450-476.

[13] Price, Kenneth, Rainer M. Storn, and Jouni A. Lampinen. *Differential evolution: a practical approach to global optimization*. Springer, 2006.

[14] Qin, A. Kai, and Ponnuthurai N. Suganthan. "Self-adaptive differential evolution algorithm for numerical optimization." *Evolutionary Computation, 2005. The 2005 IEEE Congress on*. Vol. 2. IEEE, 2005.

[15] Levine, David M., Patricia P. Ramsey, and Robert K. Smidt. *Applied statistics for engineers and scientists: using Microsoft Excel and Minitab*. Prentice Hall, 2001..

[16] Etter, Delores M., and David C. Kuncicky. *Introduction to MATLAB*. Prentice Hall, 2011.

[17] Conover, William Jay, and W. J. Conover. "Practical nonparametric statistics." (1980).

[18] Fisher, Ronald A. "Statistical methods and scientific inference." (1956).

[19] Sheskin, David J. *Handbook of parametric and nonparametric statistical procedures*. crc Press, 2003.

[20] Gibbons, Jean Dickinson, and Subhabrata Chakraborti. *Nonparametric statistical inference*. Springer Berlin Heidelberg, 2011.

[21] Rhyne, A. L., and R. G. D. Steel. "Tables for a treatments versus control multiple comparisons sign test." *Technometrics* 7.3 (1965): 293-306.

[22] Steel, Robert GD. "A multiple comparison sign test: treatments versus control."*Journal of the American Statistical Association* (1959): 767-775.

[23] Friedman, Milton. "The use of ranks to avoid the assumption of normality implicit in the analysis of variance." *Journal of the American Statistical Association* 32.200 (1937): 675-701.

[24] Friedman, Milton. "A Comparison of Alternative Tests of Significance for the Problem of $ m $ Rankings." *The Annals of Mathematical Statistics* 11.1 (1940): 86-92.

[25] Hodges, J. L., and Erich L. Lehmann. "Rank methods for combination of independent experiments in analysis of variance." *The Annals of Mathematical Statistics* 33.2 (1962): 482-497.

[26] Hoshimura, Keiichirou. *Covariance Matrix Adaptation Evolution Strategy for Constrained Optimization Problem*. Diss. Kyoto University, 2007.

[27] Alba, Enrique, Gabriel Luque, and Lourdes Araujo. "Natural language tagging with genetic algorithms." *Information Processing Letters* 100.5 (2006): 173-182.

[28] Syswerda, Gilbert. "A study of reproduction in generational and steady-state genetic algorithms." *Foundation of {G} enetic {A} lgorithms* (1991): 94-101.

[29] Herrera, Francisco, Manuel Lozano, and Daniel Molina. "Continuous scatter search: an analysis of the integration of some combination methods and improvement strategies." *European Journal of Operational Research* 169.2 (2006): 450-476.

[30] Das, Swagatam, and Ponnuthurai Nagaratnam Suganthan. "Differential evolution: A survey of the state-of-the-art." *Evolutionary Computation, IEEE Transactions on* 15.1 (2011): 4-31.

[31] Derrac, Joaquín, et al. "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms." *Swarm and Evolutionary Computation* 1.1 (2011): 3-18.

[32] Quade, Dana. "Using weighted rankings in the analysis of complete blocks with additive block effects." *Journal of the American Statistical Association* 74.367 (1979): 680-683.

[33] Rhyne, A. L., and R. G. D. Steel. "Tables for a treatments versus control multiple comparisons sign test." *Technometrics* 7.3 (1965): 293-306.

APPENDIX

A.1. MATLAB code for Friedman test

```
function FriedmanTest
% the functions calculates the Friedman statistic and
% mean rank values
%
% input:
% Update the input file name in the code
% Example:
% FreidmanTest
%
%
% Output:
% * Friedman stats value
% * Mean ranks
%
%
% Author: Srinivas Adithya Amanchi
% Data: 12.02.2014
clear all
clc
% impoting the given data into a variable called RawData
RawData = xlsread('RawDataFriedman.xlsx');
```

```
% n = NoOfRows
% k = NoOfColumns
[n, k] = size(RawData);
%% Finding Rank of the problems
for i = 1:n
    RankOfTheProblems(i,:) = tiedrank(RawData(i,:));
end
%% Taking average of the rank of the problems
AvgOfRankOfProblems = mean(RankOfTheProblems);
SquareOfTheAvgs = AvgOfRankOfProblems  .* AvgOfRankOfProblems;
SumOfTheSquares = sum(SquareOfTheAvgs);
FfStats = (12*n/(k*(k+1))) * (SumOfTheSquares - ((k*(k+1)^2)/4));
%% Display the results
formatSpec = 'Friedman statistic  is %4.2f and \n ';
fprintf(formatSpec,FfStats);
disp('Average of the ranks obtained in all problems');
disp(AvgOfRankOfProblems)
```

A.2. MATLAB code for Friedman aligned test

```
function FriedmanAllignedTest
% the functions calculates the Friedman statistic and
% mean rank values
%
% input:
```

```
% Update the input file name in the code
% Example:
% FreidmanAllignedTest
%
%
% Output:
% * Friedman stats value
% * Mean ranks
%
%
% Author: Srinivas Adithya Amanchi
% Data: 05.02.2014
clear all
clc
% impoting the given data into a variable called RawData
RawData = xlsread('RawDataFriedmanAlligned.xlsx');
% n = NoOfRows
% k = NoOfColumns
[n, k] = size(RawData);
% Taking the average of all the problems
AvgOfProblems = mean(RawData')';
% calculating the difference of each and every variable with respect to their
% respective mean value and created a new data file that has all the
```

```
% differences
for i = 1:n;
   for j = 1:k;
      DiffData(i,j) = RawData(i,j)-AvgOfProblems(i);
   end
end
clear i j AvgOfProblems
%% finding the Rank (rather ORDER) of each and every number in the difference matrix
[~, ~, RankTemp] = unique(DiffData);
% Finding values with equal rank and turn them into average ranks
UniqueRanks = unique(RankTemp);
EqualRanks=UniqueRanks(histc(RankTemp,UniqueRanks)>1);
for i=1:length(EqualRanks)
TempMatrix{i} = find(RankTemp==EqualRanks(i));
NoTemp = numel(TempMatrix{i});
for j = 2:NoTemp
ix = TempMatrix{i}(j);
if DiffData(ix) >0
DiffData(ix) = DiffData(ix)+(0.00000000001*j);
else
DiffData(ix) = DiffData(ix)-(0.00000000001*j);
end
end
```

```
end
[~, ~, RankTemp] = unique(DiffData);
ix = length(RankTemp)/k;
j =1;
for i = 1:ix:length(RankTemp);
if j<=k
RankOfTheProblems(:,j) = RankTemp(i:i+ix-1);
j = j+1;
end
end
clear NoTemp UniqueRanks i ix j EqualRanks
for i = 1:length(TempMatrix)
RankOfTheProblems(TempMatrix{i}) = mean(RankTemp(TempMatrix{i}));
end
clear RankTemp TempMatrix i
%% Information on ranks - ROW's wise
SumOfEachRanksRows = sum(RankOfTheProblems,2);%% Ri
%SumOfRanksRows = sum(SumOfEachRanksRows);
SquareOfSumOfRanksRows = SumOfEachRanksRows .* SumOfEachRanksRows;  %%
Ri^2
SumOfSquaresOfRanksRows = sum(SquareOfSumOfRanksRows);   %% sum(Ri^2)
%% Information on ranks - COLUMN's wise
SumOfEachRanksColumns = sum(RankOfTheProblems,1);   %% Rj
```

```
%SumOfRanksColumns = sum(SumOfEachRanksColumns);
SquareOfSumOfRanksColumns==SumOfEachRanksColumns*SumOfEachRanksColum
ns;
%% Rj^2
SumOfSquaresOfRanksColumns = sum(SquareOfSumOfRanksColumns);%% sum(Rj^2)
clear DiffData
%% Friedman statistic
FARStats = ((k-1) * [(SumOfSquaresOfRanksColumns) - (((k*(n*n))/4)*(k*n +1)^2)])/
    ((((k*n)*(k*n+1)*(2*k*n+1))/6) - (1/k)*(SumOfSquaresOfRanksRows));
MeanRanks = (SumOfEachRanksColumns)/n;
Sigma = std(MeanRanks);
%% Display the results
formatSpec = 'Friedman Alligned statistic is %4.2f and  \n';
fprintf(formatSpec,FARStats);
disp('Mean Ranks');
disp(MeanRanks)
```

A.3. MATLAB code for the Quade test

```
function QuadeTest
% the functions calculates the Quade statistic and
% mean rank values
%
% input:
% Update the input file name in the code
```

```
% Example:
% QuadeTest
%
%
% Output:
% * Quade stats value
% * Mean ranks
%
%
% Author: Srinivas Adithya Amanchi
% Data: 06.02.2014
clear all
clc
% impoting the given data into a variable called RawData
RawData = xlsread('RawDataQuade.xlsx');
% n = NoOfRows
% k = NoOfColumns
[n, k] = size(RawData);
%%
MinValueRow = min(RawData')';
MaxValueRow = max(RawData')';
DiffMaxMinValue = MaxValueRow - MinValueRow;
RankOfDiff = tiedrank(DiffMaxMinValue);
```

```
for i = 1:n
RankOfTheProblems(i,:) = tiedrank(RawData(i,:));
end
clear MinValueRow MaxValueRow DiffMaxMinValue RawData
%% rankings without average adjusting statistic that represents the relative size of each observation within the problem,  Sj
for i = 1:k
StatsSij(:,i) = RankOfDiff .* ( RankOfTheProblems(:,i) - ((k+1)/2));
end
SumOfStatSj = sum(StatsSij);
SquareOfSumOfStatsSj = SumOfStatSj .* SumOfStatSj;
SumOfSquareSj = sum(SquareOfSumOfStatsSj);
clear SquareOfSumOfStatsSj SumOfStatSj StatsSij
%% rankings without average adjusting
for i = 1:k
StatsWij(:,i) = RankOfDiff .* ( RankOfTheProblems(:,i));
end
SumOfStatWj = sum(StatsWij);
clear StatsWij
%% the average ranking for the jth algorithm, Tj
StatsTj = SumOfStatWj/(n*(n+1)/2);
clear SumOfStatWj
%% Remaining statistics
```

```
% A = n(n + 1)(2n + 1)k(k + 1)(k ? 1)/72

StatsAValue = (n*(n+1)*(2*n+1)*k*(k+1)*(k-1))/72;

% B = mean(kSj^2)

StatsBValue = (SumOfSquareSj)/n;

%% Quade Stats Value

FQStats = ((n-1)*StatsBValue)/(StatsAValue - StatsBValue);

%% Display the results

formatSpec = 'Quade statistic  is %4.2f\n A value is %4.2f and \n  B Value is %4.2f\n';

fprintf(formatSpec,FQStats,StatsAValue,StatsBValue);

disp('The average ranking for the jth algorithm, Tj');

disp(StatsTj)
```