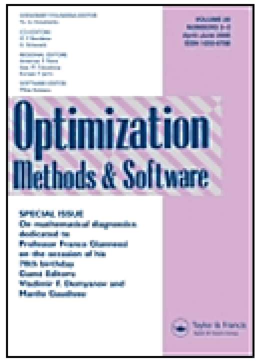


A Comparative Study of Some Real Coded Genetic Algorithms for Unconstrained Global Optimization

Journal:	*Optimization Methods and Software*
Manuscript ID:	Draft
Manuscript Type:	Original Article
Date Submitted by the Author:	n/a
Complete List of Authors:	Sawyerr, Babatunde; University of Lagos, Department of Computer Sciences Ali, M. Montaz; University of the Witwatersrand, Computational and Applied Mathematics; University of the Witwatersrand, Computational and Applied Mathematics Adewumi, Aderemi; University of KwaZulu-Natal, School of Computer Sciences
Keywords:	Global Optimization, Genetic Algorithms, Pattern Search, Projection

A Comparative Study of Some Real Coded Genetic Algorithms for Unconstrained Global Optimization

B. A. Sawyerr[1,*], M. M. Ali[2] and A. O. Adewumi[3]

[1] Department of Computer Sciences, University of Lagos, Akoka, Yaba, Lagos, Nigeria.

[2] School of Computational and Applied Mathematics, University of the Witwatersrand, Private Bag 3, Wits 2050, Johannesburg, South Africa.

[3] School of Computer Science, University of KwaZulu-Natal, Durban, South Africa

Abstract

In this paper, a set of new Real-Coded Genetic Algorithms (RCGAs) with local and global exploratory search capabilities are proposed. The search capabilities are based on the inclusion of a modified crossover procedure and a new global exploratory method in RCGA. The global exploratory method is based on vector projection while the modified crossover procedure is based on a limited version of the pattern search (PS) method. These modifications are introduced to increase the efficiency and robustness of RCGAs through better local and global exploration of the search region. An experimental study of the new algorithms was carried out using a set of 57 test problems. Statistical analyses and comparisons of the new algorithms with standard real coded genetic algorithm (SRCGA) and some recent global optimization algorithms were carried out. Results obtained show that the modifications remarkably improve the performance of RCGAs across the test problems.

Keywords: Global Optimization, Genetic Algorithms, Pattern Search, Projection.

1 Introduction

Unconstrained global optimization problems can be represented as:
Given $f : S \rightarrow \mathbb{R}$ where $S \subset \mathbb{R}^n$, find $x^* \in S$ for which,

$$f(x^*) \leq f(x), \quad \forall x \in S. \tag{1}$$

The variable x^* is called the global minimizer of f and $f(x^*)$ is called the global minimum value of f. Global optimization problems are frequently found in many practical applications in engineering, physics, economics, systems biology and other scientific applications. They are known as nonlinear programming problems and are generally very difficult problems [1, 2, 3, 4, 5].

Over the last four decades, efficient algorithms have been developed for solving global optimization problems within a reasonable time frame. These algorithms include Evolutionary Programming (EPs) [6], Evolution Strategies (ESs) [7], Genetic Algorithms (GAs) [8] and Differential Evolution (DE) [9]. They are popularly known as evolutionary algorithms (EAs). EAs are a class of direct and probabilistic algorithms based on the Darwinian notion of natural selection. They are robust, simple to implement and have recorded great successes in search and optimization because of their ability to exploit information accumulated from an initially unknown search space [6, 10].

*Corresponding author's email: tundesawyerr@yahoo.com

In recent years, RCGAs have received a lot of attention within the EA research community where many researchers have solved continuous optimization problems with RCGAs [1, 6, 10, 11, 12, 13, 14]. RCGAs offer several advantages over the binary coded genetic algorithms (BCGAs) because they are better adapted to numerical optimization of continuous problems. They can also be easily hybridized with other search methods. Despite their advantages over BCGAs, RCGAs are prone to premature convergence (partly due to lack of population diversity and high selection pressure) and slow convergence as GAs do not exploit the local basins of solutions in the population [11].

The inability of GAs to effectively exploit the local basins of the search region led to their hybridization with other algorithms, especially with local optimization algorithms. Hybridization has become an important approach in overcoming the shortcomings of GAs. GAs have been hybridized with other optimization methods to improve their search capability, for example GA with Nelder-Mead simplex method [15, 16], GA with particle swarm optimization [17], GA with self organizing migrating algorithm [18], and GA with quadratic approximation [19]. Hybrid GAs have been used to solve real-life problems and they have recorded impressive improvements especially in maintaining population diversity throughout the search process. Population diversity ensures that a rich variety of solution points are maintained in a GA population set thus leading to the exploration of a wider scope of the solution landscape. Population diversity also prevents the algorithm from converging prematurely to local optima.

However, the possibilities of developing more efficient and robust GAs still abound. Therefore the main objective of this paper is to introduce new ways of improving the performance of RCGAs by incorporating a feature from another algorithm that adds complementary strength to RCGAs, and by introducing a new exploratory feature to RCGAs.

In this paper, three new algorithms are proposed. They consist of either a modified crossover procedure using 'limited pattern search' and /or vector projection-based exploratory search. These algorithms are labeled as RCGA-PS (real coded genetic algorithm with pattern search), RCGA-P (real coded genetic algorithm with projection), and RCGA-PS-P (real coded genetic algorithm with pattern search and projection).

The remaining part of the paper is organized as follows: Section 2 provides a brief introduction to genetic algorithms. Section 3 presents the new real coded genetic algorithms with full descriptions of the proposed modifications. A brief introduction to pattern search method and vector projection-based search method is also provided. Section 4 provides the experimental settings and parameter selection while Section 5 presents the results, comparisons and discussion. Finally, Section 6 concludes with some remarks.

2 Genetic Algorithms (GAs)

The genetic algorithm was developed by John Holland in 1975 (then called adaptive or reproductive plans) [8]. GAs are defined as search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest strategy among string structures with ordered yet randomized information exchange to form a robust global exploration algorithm. At every generation t, a new set of solution points, $P_t = \{x_{1,t}, x_{2,t}, \dots, x_{N,t}\}$, is created using bits and pieces of the fittest parent solutions and an occasional new part is sampled. While randomized, GAs ingeniously exploit historical information to consider new search points with expected superior performance [20]. Several variations to the original GA [8] have been developed, using different representation schemes, selection, crossover, mutation and elitism operators [3, 10]. RCGA is a GA that uses floating point representation for holding values of the solution points. A typical solution or chromosome is a vector of floating point numbers [10, 15].

At each generation t, the standard real coded genetic algorithm (SRCGA) performs selection, crossover, mutation and elitism to update the current population set P_t. In this study, the linear ranked selection is used as suggested by James Baker [21, 22] to create the mating pool, $\hat{P}_t = \{x_{1,t}, x_{2,t}, \dots, x_{m,t}\}$, $m \leq N$ [20], which in turn is transformed by crossover, mutation and elitism into the population, P_{t+1}, for the next generation. Linear ranked selection consists of two parts: (1) determination of $x_{i,t}$'s expected value, $Ev(x_{i,t})$, and (2) conversion of the expected value to discrete numbers of offspring. The expected value, $Ev(x_{i,t})$, of $x_{i,t}$ is a real number indicating the average number of offspring that $x_{i,t}$ should receive, where $\sum_{i=1}^{N} Ev(x_{i,t}) = N$ [22].

Linear selection is implemented by sorting the solutions in the population according to their fitness and each solution is assigned a rank in the sorted population. The best solution gets the first position, i.e. ranked number 1 while the worst solution receives the last rank, N, in the population. The rank of $x_{i,t}$ is used to calculate $Ev(x_{i,t})$ by a linear function,

$$Ev(x_{i,t}) = Max - \frac{2 \times (Max - 1.0) \times (i-1)}{N-1}, \quad (1.0 \leq Max \leq 2.0), \tag{2}$$

where Max is the expected value distribution for the linear function. Max = 1.1 (as recommended in [21, 22]) and $m = N$ (as recommended in [20]) are used. Then stochastic universal sampling algorithm (SUS) is used to convert the expected number, $Ev(x_{i,t})$, of $x_{i,t}$ to discrete numbers of $x_{i,t}$. These discrete numbers are used by SUS to select the corresponding numbers of points, $x_{i,t}$, that would be placed in $\hat{P}_t$ [22].

Crossover is applied pairwise with probability, p_c, to all members of $\hat{P}_t$. If the probability of crossover is successful then arithmetic crossover (AC) is carried out on a pair $(x_{i,t}, x_{i+1,t})$ as follows

$$\begin{aligned} y_{i,t}^j &= \alpha^j x_{i,t}^j + (1-\alpha^j) x_{i+1,t}^j, \\ y_{i+1,t}^j &= \alpha^j x_{i+1,t}^j + (1-\alpha^j) x_{i,t}^j, \end{aligned} \tag{3}$$

where α^j is uniform in [-0.5,1.5], i.e. $\alpha^j \sim Unif([-0.5, 1.5])$ for each j, $j = 1, 2, \ldots, n$. The new pair $(y_{i,t}, y_{i+1,t})$ is then copied to the set C_t. If on the other hand, crossover probability is unsuccessful, then the pair $(x_{i,t}, x_{i+1,t})$ is copied to C_t. Without loss of generality, we write C_t as

$$C_t = \{y_{1,t}, y_{2,t}, \ldots, y_{m,t}\}. \tag{4}$$

We then apply mutation to the components of each member of C_t with probability, p_μ. If the probability of mutation is successful at a component $y_{i,t}^j$ of $y_{i,t} \in C_t$, then random mutation [10] is carried out as follows

$$z_{i,t}^j = y_{i,t}^j + \beta^j (u^j - l^j), \tag{5}$$

where $\beta^j \sim Unif([-0.01, 0.01])$ for each j, $j = 1, 2, \ldots, n$, u^j and l^j are the upper and lower boundaries of $x \in S$, respectively. On the other hand, if the mutation probability is unsuccessful at component $y_{i,t}^j$ of $y_{i,t} \in C_t$, then $y_{i,t}^j$ is retained. The resultant $z_{i,t} \in M_t$ consists of both mutated and non mutated components. We denote M_t by

$$M_t = \{z_{1,t}, z_{2,t}, \ldots, z_{m,t}\}, \tag{6}$$

where

$$z_{i,t}^j = \begin{cases} y_{i,t}^j + \beta^j (u^j - x_{i,t}^j), & \text{if} \quad y_{i,t}^j \text{ is mutated} \\ y_{i,t}^j, & \text{otherwise}. \end{cases} \tag{7}$$

If $m < N$ then m points in M_t replace m worst points in P_t to create P_{t+1}. In this paper, we use $m = N$. After the creation of P_{t+1}, elitism is applied to preserve the elite solutions. The elitism is applied when the best point in P_t is better than the best point in P_{t+1}. The elitism is carried out to replace the worst point in in P_{t+1} with the best point in P_t [23]. We now present a step by step description of the standard real coded genetic algorithm:

Algorithm 1: The SRCGA Algorithm

1. *Randomly initialize and evaluate N uniformly distributed solution points in population set $P_t = \{x_{1,t}, x_{2,t}, \ldots, x_{N,t}\}$ from search space S. Set the generation counter $t = 0$.*
2. *While necessary stopping condition is not met, do steps 3 - 7.*
3. *Select $m \leq N$ solutions from P_t as parents using linear-ranked selection to form a mating pool $\hat{P}_t$.*
4. *Select pairs of parents sequentially from $\hat{P}_t$ and use arithmetic crossover with probability p_c to create offspring solutions. Save the offspring in C_t.*
5. *Using random mutation, perform mutation on each component of $y_{i,t} \in C_t$ with a low probability, p_μ, to create M_t.*
6. *Update P_t by replacing m solutions in P_t with the solutions in M_t to create P_{t+1}.*
7. *If $m = N$, then elitism is applied.*

3 New Genetic Algorithms

Three new RCGAs are presented in this section. The motivation for designing these algorithms is based on the quest for developing practical global optimization algorithms that can solve a greater number of global optimization problems in reasonable time. These new algorithms are specifically designed to address the shortcomings of SRCGA. RCGA-PS incorporates a limited version of pattern search in the crossover procedure while RCGA-P applies the projection based (P-based) exploratory search on the set of solutions after crossover and mutation operators have been used. RCGA-PS-P incorporates both PS-based crossover and P-based exploratory search.

3.1 Pattern Search based Local Exploration

Pattern search is a class of direct local search methods that explore sample points around the current point, say $x_{i,t} \in \hat{P}_t$ [24, 25]. It consists of two major components namely, the SEARCH step and the POLL step [24, 25, 26]. For a detailed description of the PS method see [24, 26]. The version of PS used in this study consists of only a modified POLL step (MPS). It has been used to augment simulated annealing in [26]. It is simple, effective and different from the one used in [27] because only a unit coordinate vector d_k is selected randomly from the direction coordinate matrix $D = \{d_1, d_2, \ldots d_{2n}\} = \{a_1, a_2, \ldots, a_n, -a_1, -a_2, \ldots, -a_n\}$, where a_k is the k^{th} unit coordinate vector in $\mathbb{R}^n$. The selected d_k is used to search for a better solution point within the neighborhood of $x_{i,t}$. MPS starts by generating a trial point $y_{i,t}$ around the current solution $x_{i,t}$ by randomly selecting a poll coordinate vector $d_k \in D$ from uniform distribution, i.e. $d_k \sim Unif(D)$, and using it to generate an intermediate point:

$$\hat{y}_{i,t} = x_{i,t} + \Delta_t d_k, \tag{8}$$

where Δ_t is a step size parameter. The operation in equation (8) is the main step in PS [24, 26]. We call this step as single pattern search step and denote it by SPS. MPS does not calculate the function value at $\hat{y}_{i,t}$, instead it calculates trial point $y_{i,t}$ using:

$$y_{i,t} = \hat{y}_{i,t} + rU, \tag{9}$$

where $\hat{y}_{i,t}$ is from equation (8), $r = \eta\Delta_t$, is a step size and $U = (U_1, U_2, \ldots, U_n)^T$ is a directional cosines with random components

$$U_j = R_j/(R_1^2 + \ldots + R_n^2)^{\frac{1}{2}}, \quad j = 1, 2, \ldots, n. \tag{10}$$

$R_j \sim Unif([-1,1])$ and η is a step factor. There are cases when the components of the trial point $y_{i,t} = (y^1_{i,t}, y^2_{i,t}, \ldots, y^n_{i,t})$ generated by equation (9) fall outside the search space S during the search. In these cases, the components of $y_{i,t}$ are regenerated using,

$$y^j_{i,t} = \begin{cases} x^j_{i,t} + \lambda(u^j - x^j_{i,t}), & \text{if} \quad y^j_{i,t} > u^j \\ x^j_{i,t} + \lambda(x^j_{i,t} - l^j), & \text{if} \quad y^j_{i,t} < l^j. \end{cases} \tag{11}$$

where $\lambda \sim Unif([0,1])$ and $x^j_{i,t}$ is the corresponding component of the current solution $x_{i,t} \in \hat{P}_t$. The operations used to create $y_{i,t}$ within the neighborhood of the current point in equations (8) and (9) are presented in Figure 1.

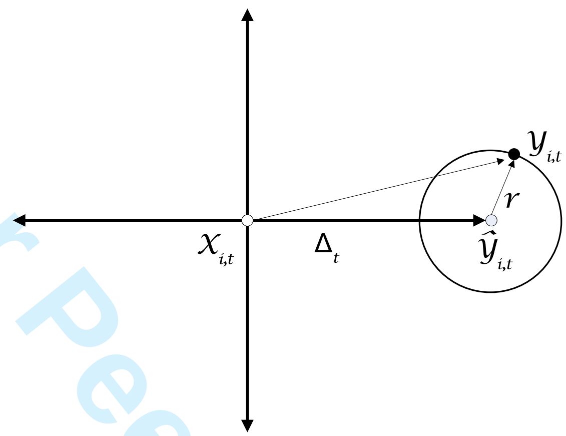


Figure 1: The generation of a trial point, $y_{i,t}$, by MPS

In Figure 1, the current point $x_{i,t}$ is treated as the POLL position. Then the intermediate point $\hat{y}_{i,t}$ is found by equation (8) and then the trial point $y_{i,t}$ is found by equation (9).

MPS has a parameter Δ_t which is initialized at the beginning of RCGA. At $t = 0$, Δ_t, is initialized by:

$$\Delta_0 = \tau \times \max\{u^j - l^j| \quad j = 1, 2, \ldots, n\}, \tag{12}$$

where $\tau \in [0,1]$. For this study, the value chosen for τ is 0.2 because it provides an initial step size that is not too large or too small. The step size[1], Δ_0, is initially set to a fraction of the length of the search region. The idea of using equation (12) to generate the initial step length is to accelerate the search by starting with a suitably large step size to quickly traverse the search space and as the search progresses the step size is adaptively adjusted.

Note that MPS used in this study is not applied iteratively on the point $x_{i,t} \in \hat{P}_t$ but it is applied in a single iteration to $x_{i,t}$ to produce $y_{i,t}$ via equations (8) and (9) within the modified crossover operator. Hence, the iteration counter of MPS is not required.

The MPS-based crossover operation introduced here is applied to each member $x_{i,t}$ of $\hat{P}_t$ to produce the corresponding $y_{i,t} \in C_t$. At the end of each generation, t, of RCGA, the parameter Δ_t is updated as follows.

A set of q distinct points, $\Omega = \{x_1, x_2, \ldots, x_q\} \subset P_t$ are randomly selected and the mean $\bar{x} = \frac{1}{q}\sum_{i=1}^{q} x_i$ of the points in Ω is calculated. Then the distances $d(\bar{x}, x_i)$ between $\bar{x}$ and each $x_i \in \Omega$ are also calculated. A comparison of all the distances $d(\bar{x}, x_i)$ is made and K nearest solutions to $\bar{x}$ are selected. Without loss of generality, the set of K nearest distances to $\bar{x}$ is denoted by $\{\gamma^1, \gamma^2, \ldots, \gamma^K\}$, and Δ_{t+1} is updated by,

$$\Delta_{t+1} = \frac{1}{K}\sum_{i=1}^{K} \gamma^i. \tag{13}$$

[1] Δ_0 can be considered component wise for problems where $(u^j - l^j)$ varies significantly from component j to component k. However, this is not needed for the problems considered in this paper.

The method (13) uses population statistics to determine the rate of change of the step size Δ_t (i.e. the solutions in P_t at time t are used to determine the step size for time $t+1$). This is a new idea that is introduced to utilize the population statistics of RCGA. A study of the behavior of this method is underway. During the implementation of MPS for adjusting Δ_t, the median of the set $\{\gamma^1, \gamma^2, \cdots, \gamma^K\}$ was also tried but the overall results were inferior.

3.2 The RCGA-PS Algorithm

This section describes the RCGA-PS algorithm. RCGA-PS is a real coded genetic algorithm with pattern search incorporated into its crossover procedure. The resulting crossover operator is called *modified crossover* (MC). An iteration of RCGA-PS consists of selection, crossover, mutation and elitism.

At generation t, RCGA-PS creates a mating pool $\hat{P}_t$ by using equation (2) in Section 2 as in SRCGA. After the mating pool, $\hat{P}_t$, is created, MC is applied to each member of $\hat{P}_t$. Hence, there is no need to implement the crossover probability, p_c, as in SRCGA. MC uses a probability distribution over a set of crossover operators, e.g. over the set {MPS, PSAC}, where PSAC is a crossover operator that combines SPS defined by equation (8) with AC in equation (3). MC uses MPS with a probability ρ and PSAC with probability $1-\rho$. If MC uses MPS at $x_{i,t} \in \hat{P}_t$, then the corresponding $y_{i,t}$ is created using equations (8) and (9). On the other hand, if MC uses PSAC at $x_{i,t} \in \hat{P}_t$, then another point $x_{j,t} \in \hat{P}_t$, is selected at random and SPS is applied to both $x_{i,t}$ and $x_{j,t}$ and two corresponding points $\hat{y}_{i,t}$ and $\hat{y}_{j,t}$ are created using equation (8). Then AC is applied to $\hat{y}_{i,t}$ and $\hat{y}_{j,t}$ resulting in $y_{1,t}$ and $y_{2,t}$. The best point $y_{i,t} = \arg\min\{f(y_{1,t}), f(y_{2,t})\}$ is considered as the crossover point corresponding to $x_{i,t} \in \hat{P}_t$. Whether $y_{i,t}$ is created via MPS or PSAC, it then competes with the corresponding $x_{i,t} \in \hat{P}_t$. If $f(y_{i,t}) < f(x_{i,t})$ then $y_{i,t}$ is copied to C_t else $x_{i,t}$ is copied to C_t.

Mutation is applied to the components of each $x_{i,t} \in C_t$ with probability, p_μ, see equation (5). This results in M_t. M_t replaces P_t to create P_{t+1} and elitism is applied as in SRCGA.

RCGA-PS combines the complementary strengths of GAs and PS to form a strong, efficient and robust algorithm. It has the ability to explore both the global and local regions of the search space. It performs local search through MPS and global search through its crossover and mutation operators. With these properties RCGA-PS becomes more robust and superior to SRCGA. A step by step description of the RCGA-PS algorithm is presented below:

Algorithm 2: The RCGA-PS Algorithm

---

1. *Randomly initialize and evaluate N uniformly distributed solution points in population set $P_t = \{x_{1,t}, x_{2,t}, \ldots, x_{N,t}\}$ from search space S. Set the generation counter $t = 0$.*
2. *While necessary stopping condition is not met, do steps 3 - 8.*
3. *Select $m \leq N$ solutions from P_t as parents using linear-ranked selection to form a mating pool $\hat{P}_t$.*
4. **Perform crossover using modified crossover (MC)**
   (a) *For each $x_{i,t} \in \hat{P}_t, i = 1, 2, \ldots, m$ do*
   Select MPS with probability ρ and PSAC with probability $1-\rho$.
   If MC = MPS then
   - *Use the POLL step in equation (8) to generate $\hat{y}_{i,t}$ from $x_{i,t}$*
   - *Perturb the coordinate direction using equations (9) and (10) to create $y_{i,t}$ from $\hat{y}_{i,t}$*

   else if MC = PSAC then
   - *Randomly select a partner $x_{j,t}$ for $x_{i,t}$ from the set $\hat{P}_t$*
   - *Use the POLL step in equation (8) on parents $x_{j,t}$ and $x_{i,t}$ to produce $\hat{y}_{j,t}$ and $\hat{y}_{i,t}$ respectively*

- *Perform AC on $\hat{y}_{j,t}$ and $\hat{y}_{i,t}$ to create $\{y_{1,t}, y_{2,t}\}$*
- *The best point $y_{i,t} = \arg\min\{f(y_{1,t}), f(y_{2,t})\}$ is chosen*

(b) *If $f(y_{i,t}) < f(x_{i,t})$ then $y_{i,t}$ is copied to C_t else $x_{i,t}$ is copied to C_t*

5. *Using random mutation, perform mutation on each component of $x_{i,t} \in C_t$ with a low probability, p_μ, to create $M_t = \{x_{1,t}, x_{2,t}, \ldots, x_{m,t}\}$.*
6. *Update P_t by replacing m solutions in P_t with M_t to create P_{t+1}.*
7. *If $m = N$, then elitism is applied.*
8. *Adjust Δ_t adaptively using equation (13)*

Remarks

1. Algorithm 2 is similar to Algorithm 1 except step 4 which is highlighted in bold.
2. Step 4 replaces AC with MC. MC probabilistically uses MPS and PSAC. This step distinguishes RCGA-PS from SRCGA which simply uses AC.

3.3 Projection-based Exploration

The projection-based exploration search method is based on the concept of orthogonal projection of vector x on vector y. This concept is not new in linear algebra, but its application to evolutionary computation especially in genetic algorithm is new. To the best of our knowledge, there has been no application of this concept in global optimization.

Vector projection is used in this work to enhance the exploration of points in the search space by using two solution points to locate a better point in the solution landscape. It is a two-parent operator that produces only one offspring. Suppose two solutions x and y are randomly selected from P_t and are evaluated, if $f(x)$ is better than $f(y)$ then we project y on x otherwise project x on y.

For any two n dimensional vectors, the projection of x on y generates a vector $\hat{y}$ defined by:

$$\hat{y} = \frac{x^T y}{y^T y} y = \frac{x^T y}{\|y\|^2} y = \left(\frac{\|x\| \cos(\theta)}{\|y\|} y \right). \tag{14}$$

Note that the projected vector $\hat{y}$ (the offspring) will be in the same direction as y unless $\frac{\pi}{2} < \theta < \frac{3\pi}{2}$ in which case the angle, θ, between the two vectors is such that $\cos(\theta) < 0$. As a result, the projected vector is in the opposite direction (the reflection of y about the origin).

3.4 The RCGA-P Algorithm

In this section, a projection-based RCGA (RCGA-P) is presented. Structurally, SRCGA and RCGA-P are similar, except that RCGA-P incorporates the projection based exploration mechanism at the end of each generation, t, of SRCGA. After RCGA-P creates $\hat{P}_t$, the crossover operation in equation (3) is used to create C_t. Then M_t is created from C_t by mutation using equation (5).

Now the projection-based operation is used to transform M_t to Φ_t. For each $z_{i,t} \in M_t, (i = 1, 2, \ldots, m)$, a pair of points, $(z_{i,t}, z_{j,t})$, is selected at random from M_t and a projected point $s_{i,t} \in \Phi_t$ is created. Hence $\Phi_t = \{s_{1,t}, s_{2,t}, \ldots, s_{m,t}\}$.

Sometimes the components $s^j_{i,t}$ of the trial point $s_{i,t}$ may fall outside the search space S. In such cases, the corresponding component $s^j_{i,t}$ is regenerated using equation (11). After the projected vector is generated, its fitness value $f(s_{i,t})$ is determined. A new population, P_{t+1}, is created with $x_{i,t}$, where,

$$x_{i,t} = \begin{cases} s_{i,t} & \text{if } f(s_{i,t}) < f(z_{i,t}), z_{i,t} \in M_t \\ z_{i,t} & \text{otherwise.} \end{cases} \tag{15}$$

Elitism is applied as in SRCGA.

This exploratory search is found to be highly efficient by making RCGA take big jumps in the search space. Below is a step by step description of the RCGA-P algorithm.

Algorithm 3: The RCGA-P Algorithm

1. *Randomly initialize and evaluate* N *uniformly distributed solution points in population set* $P_t = \{x_{1,t}, x_{2,t}, \ldots, x_{N,t}\}$ *from search space* S. *Set the generation counter* $t = 0$.
2. *While necessary stopping condition is not met, do steps 3 - 8.*
3. *Select* $m \leq N$ *solutions from* P_t *as parents using linear-ranked selection to form a mating pool* $\hat{P}_t$.
4. *Select pairs of parents sequentially from* $\hat{P}_t$ *and use arithmetic crossover (AC) with probability* p_c *to create offspring solutions. Save the offspring in* C_t.
5. *Using random mutation, perform mutation on each component of* $y_{i,t} \in C_t$ *with a low probability,* p_μ *to create* M_t.
6. ***For each*** $z_{i,t} \in M_t (i = 1, 2, \ldots, m)$ ***generate the pair*** $(z_{i,t}, z_{j,t}), z_{i,t}, z_{j,t} \sim Unif(M_t)$***, create*** $s_{i,t} (i = 1, 2, \ldots, m)$ ***by projection and generate*** $\Phi_t = \{s_{1,t}, s_{2,t}, \ldots, s_{m,t}\}$.
7. *Update* P_t *by replacing* m *solutions in* P_t *with the solutions in* Φ_t *to create* P_{t+1}.
8. *If* $m = N$*, then elitism is applied.*

Remarks

3. Algorithm 3 is structurally similar to algorithm 1 except for step 6 which is highlighted in bold.

3.5 The RCGA-PS-P Algorithm

This section presents the RCGA-PS-P algorithm. RCGA-PS-P is very similar to RCGA-PS except for the P-based exploration incorporated at the end of each iteration of RCGA-PS-P in the same way RCGA-P incorporates projection.

At generation t, RCGA-PS-P creates a mating pool, $\hat{P}_t$, by using equation (2). After the mating pool, $\hat{P}_t$, is created, MC is applied to each member of $\hat{P}_t$ with a probability distribution over the set of crossover operators {MPS, PSAC} to create $y_{i,t}$. $y_{i,t}$ then competes with the corresponding $x_{i,t} \in \hat{P}_t$. If $f(y_{i,t}) < f(x_{i,t})$ then $y_{i,t}$ is copied to C_t else $x_{i,t}$ is copied to C_t. Mutation is applied to the components of each $x_{i,t} \in C_t$ with probability, p_μ, to create M_t. These operations are the same as in RCGA-PS. In addition, the set, Φ_t, is created using the projection-based operation. For each $z_{i,t} \in M_t$, a pair of points, $(z_{i,t}, z_{j,t})$, is selected at random from M_t and a projected point $s_{i,t} \in \Phi_t$ is created, where $\Phi_t = \{s_{1,t}, s_{2,t}, \ldots, s_{m,t}\}$.

Equation (11) is used to adjust components of $s_{i,t}$ that fall outside the search space. After the projected vector is generated, its fitness value $f(s_{i,t})$ is determined and a new population, P_{t+1}, created using equation (15). Elitism is applied as in SRCGA.

The motivation for this modification is based on the good performances of RCGA-PS and RCGA-P. The incorporation of MPS and projection-based exploratory mechanism in RCGA produces a robust algorithm that combines the complementary properties of GAs, PS and projection. Below is a step by step description of RCGA-PS-P:

Algorithm 4: The RCGA-PS-P Algorithm

1. *Randomly initialize and evaluate* N *uniformly distributed solution points in population set* $P_t = \{x_{1,t}, x_{2,t}, \ldots, x_{N,t}\}$ *from search space* S. *Set the generation counter* $t = 0$.
2. *While necessary stopping condition is not met, do steps 3 - 9.*

3. *Select $m \leq N$ solutions from P_t as parents using linear-ranked selection to form a mating pool $\hat{P}_t$.*

4. **Perform crossover using modified crossover (MC)**

   (a) *For each $x_{i,t} \in \hat{P}_t, i = 1, 2, \ldots, m$ do*

   Select MPS with probability ρ and PSAC with probability $1 - \rho$.
   If MC = MPS then

   - *Use the POLL step in equation (8) to generate $\hat{y}_{i,t}$ from $x_{i,t}$*
   - *Perturb the coordinate direction using equations (9) and (10) to create $y_{i,t}$ from $\hat{y}_{i,t}$*

   else if MC = PSAC then

   - *Randomly select a partner $x_{j,t}$ for $x_{i,t}$ from the set $\hat{P}_t$*
   - *Use the POLL step in equation (8) on parents $x_{j,t}$ and $x_{i,t}$ to produce $\hat{y}_{j,t}$ and $\hat{y}_{i,t}$ respectively*
   - *Perform AC on $\hat{y}_{j,t}$ and $\hat{y}_{i,t}$ to create $\{y_{1,t}, y_{2,t}\}$*
   - *The best point $y_{i,t} = \arg\min\{f(y_{1,t}), f(y_{2,t})\}$ is chosen*

   (b) *If $f(y_{i,t}) < f(x_{i,t})$ then $y_{i,t}$ is copied to C_t else $x_{i,t}$ is copied to C_t*

5. *Using random mutation, perform mutation on each component of $x_{i,t} \in C_t$ with a low probability, p_μ to create $M_t = \{x_{1,t}, x_{2,t}, \ldots, x_{m,t}\}$.*

6. ***For each** $z_{i,t} \in M_t(i = 1, 2, \ldots, m)$ **generate the pair** $(z_{i,t}, z_{j,t}), z_{i,t}, z_{j,t} \sim Unif(M_t)$**, create** $s_{i,t}(i = 1, 2, \ldots, m)$ **by projection and generate** $\Phi_t = \{s_{1,t}, s_{2,t}, \ldots, s_{m,t}\}$.*

7. *Update P_t by replacing m solutions in P_t with the solutions in Φ_t to create P_{t+1}.*

8. *If $m = N$, then elitism is applied.*

9. *Adjust Δ_t adaptively using equation (13)*

Remark

4. RCGA-PS-P is structurally similar to RCGA-PS except for step 6, which makes it different from RCGA-PS.

4 Experimental Settings

To compare the performance of the proposed algorithms with SRCGA and some recent GAs, a set of 57 benchmark problems from [5] was used. All the problems are minimization problems of continuous variables with their dimensions ranging from 2 to 20 with different degrees of difficulty. There are many local optima and/or saddles in the solution spaces of these test problems. Detailed description of the problems can be found in [5].

All algorithms were implemented in Microsoft Visual Studio 2005 integrated development environment using C#.NET programming language on Windows Vista business operating system running on an Intel core 2 CPU at 1.66GHz with 1GB of RAM.

4.1 Parameter Selection

All the RCGAs use the same basic parameter values which are supplied by the user through the graphical user interface. These parameters include; the size N of the population set P_t, the maximum number of generation T, mutation probability p_μ, crossover probability p_c, number of elitist solutions E, the linear-ranked selection expected value distribution Max and the probability ρ used by MPS in RCGA-PS and RCGA-PS-P. Table 1 below shows the parameter settings.

All the values shown in Table 1 are standard parameter settings that have been successfully used in literature [6, 11]. Other parameters used in RCGA-PS and RCGA-PS-P are: (i) Δ_t,

Table 1: Parameter Settings for the Experiment

Sno.	Parameter	Value
1	Population size (N)	$10 \times n$, (n is the problem dimension)
2	Maximum number of generation (T)	$10,000$
3	Mutation probability (p_μ)	0.001
4	Crossover probability (p_c)	0.6
5	Probability of selecting MPS (ρ)	0.4
6	Elitism (E)	1
7	Linear selection's expected value distribution (Max)	1.1

initialized by equation (12) and adaptively adjusted by equation (13), (ii) τ, the constant used in calculating Δ_0 in equation (12), (iii) q the size of Ω is set to 15 and (iv) K the number of the nearest neighbors to the mean $\bar{x}$ of points in Ω is set to 10 (see Section 3.1).

5 Experimental Results

The results obtained from the experiments performed on the new RCGAs and SRCGA using 57 test problems are presented. These results are the performance measurements. They are used to determine the efficiency and effectiveness of the new algorithms over SRCGA. Performance evaluation of stochastic algorithms can be done using three major criteria. These criteria are the test for: ***convergence***, ***speed*** and ***robustness*** of the algorithm [30]. The convergence measure provides a scientific means of determining how effective an algorithm converges to the desired solution. Equation (16) is used to test for convergence on all four algorithms. In the experiment, a run is terminated when the algorithm converges to a good solution i.e. if the best function value $f_{\min}$ found so far satisfies

$$|f_{\min} - f(x^*)| \leq \epsilon, \quad \epsilon = 10^{-4}, \tag{16}$$

or when the maximum number of generation T is reached.

The speed of an algorithm is determined by counting the number of function evaluations of the algorithm. The number of function evaluations is chosen as a measure of speed since it is independent of the type of machine used. The number of function evaluations can be used to compare algorithms irrespective of the machine used for implementation. For each algorithm, the mean number of function evaluations (MFE) for solving a test problem is obtained by averaging the number of function evaluations over the total number of trial runs.

The MFE is used for comparing all the proposed algorithms with SRCGA and the comparison of the best performing RCGA with recent GAs from the literature. The MFE of an algorithm is used for comparison if and only if the algorithm's success rate (SR) is at least 1. That is if an algorithm is run 100 times on a problem and SR $\geq$ 1, then MFE is obtained using 100 runs. If for a problem, SR = 0 for all independent runs then this is denoted by '- ', see Table 3.

Lastly, the robustness of an algorithm (the third criterion) is determined by comparing SR of the algorithms. SR of an algorithm is the number of successful runs (i.e. number of times that a problem is solved) of the algorithm out of a predefined number of runs, e.g 100 runs. It can also be described as the number of times an algorithm finds the optimal solution. A run is counted successful if the best solution, $f_{\min}$, found in a run satisfies

$$|f(x^*) - f_{\min}| \leq 0.009. \tag{17}$$

Any algorithm that solves a wide range of problems is considered a robust algorithm, i.e. the algorithm is not specific to some problems. This measure is an important criterion for GAs since GAs are reputed to be robust. Each algorithm was run independently for 100 trials on each of the 57 benchmark problems to determine its success rate. There are 57 problems, hence there are 5,700 runs in total for each algorithm. The algorithms collectively solved 54 problems except for Epistatic Michalewicz, Odd Square and Price's Transistor Modeling problems whose global minima could not be located by all the algorithms within the maximum number of iterations. Therefore results for the three problems are not presented.

The best fitness values (Min), mean best fitness values (MBF), mean function evaluations (MFE), success rate (SR), standard deviations (STD), p-values from analysis of variance (ANOVA) test, box-plots and multiple comparison (MCx) graphs are used in the comparison of the algorithms.

In addition, we used the Success Performance (SP) criterion introduced in [28] to reconfirm our results by estimating the expected number of function evaluations for successful runs. SP is defined by:

$$SP = mean(FEs) \times \left(\frac{TR}{SR}\right) \tag{18}$$

where FEs = function evaluations of successful runs, TR = number of total runs and SR = number of successful runs [29]. The normalized SP of an algorithm is calculated by dividing the algorithm's SP by the SP of the best algorithm (SP_{best}). Experimental results from recent literature [30, 31] are also used for comparison.

Table 2: Comparison of MFEs and SR of SRCGA and the New RCGAs for Problems with Dimensions 2 - 4

Pno.	Problem name	n	Mean Function Evaluations (MFE)				Success Rate (SR)			
			SRCGA	RCGA-PS	RCGA-PS-P	RCGA-P	SRCGA	RCGA-PS	RCGA-PS-P	RCGA-P
1	Aluffi-Pentini	2	1,449	988	911	969	100	100	100	100
2	Becker and Lago	2	1,602	5,373	6,413	1,761	100	100	100	100
3	Bohachevsky 1	2	2,927	1,567	819	630	100	100	100	100
4	Bohachevsky 2	2	20,598	1,606	816	558	91	100	100	100
5	Branin	2	4,358	6,517	8,478	1,374	100	100	100	100
6	Camel Back-3	2	1,204	1,041	513	275	100	100	100	100
7	Camel Back-6	2	1,253	2,535	3,713	1,700	100	100	100	100
8	Cosine Mixture	2	697	849	346	248	100	100	100	100
9	Dekkers and Aarts	2	10,270	4,765	7,111	222,544	100	100	100	85
10	Easom	2	2,136	1,202	1,197	1,718	100	100	100	100
11	Goldstein and Price	2	11,688	1,576	1,704	106,406	97	100	100	92
12	Hosaki	2	1,015	756	740	848	100	100	100	100
13	McCormick	2	1,446	1,063	1,176	2,996	100	100	100	100
14	Modified Rosenbrock	2	104,570	211,046	78,495	81,904	81	100	100	100
15	Multi-Gaussian	2	95,028	64,111	2,040	802	53	89	100	100
16	Periodic	2	71,366	59,389	1,014	374	65	90	100	100
17	Schaffer 1	2	200020	404,963	10,256	745	7	28	99	100
18	Schaffer 2	2	200020	554,458	722,392	376,127	16	83	100	100
19	Shubert	2	7,738	80,796	48,590	48,381	99	99	100	89
20	Gulf Research	3	252,118	232,275	418,314	400,839	100	100	100	99
21	Hartman 3	3	3,115	2,353	1,832	24,471	100	100	100	100
22	Helical Valley	3	3,903	3,393	2,738	111,348	100	100	100	100
23	Levy and Montalvo 1	3	2,473	2,593	2,391	6,823	100	100	100	100
24	Meyer and Roth	3	177,738	4,897	11,798	304,354	100	100	100	100
25	Cosine Mixture	4	3,420	3,250	428	256	100	100	100	100
26	Kowalik	4	550	919	670	421	100	100	100	100
27	Miele and Cantrell	4	19,650	1,980	3,189	87,350	100	100	100	100
28	Neumaier 2	4	400,040	923,869	1,267,658	800,040	41	100	100	35
29	Powell's Quadratic	4	88,887	7,338	1,300	435	100	100	100	100
30	Shekel 5	4	187,988	108,896	6,604	5,263	54	91	100	100
31	Shekel 7	4	136,878	40,962	20,184	44,325	67	97	99	95
32	Shekel 10	4	82,353	7,053	20,274	68,124	81	100	99	92
33	Wood	4	400,040	178,309	230,927	758,378	7	100	100	35
	Total :		**2,498,538**	**2,922,688**	**2,885,031**	**3,462,787**	**2,759**	**3,177**	**3,297**	**3,122**

5.1 Comparisons of new RCGAs with SRCGA

In this section, experimental results and their statistical analyses are summarized in Tables 2, 3, and appendices I, II and III. Table 2 Table 2 provides MFE SR for all algorithms for problems with $n = 2$ to 4 while Table 3 provides MFE and SR for problems with $n = 5$ to 20. The test problems are numbered serially in column 1 of the two tables. The problem names are also listed in column 2, MFE in columns 3 to 6 and their corresponding SR in the last four columns of Tables

Table 3: Comparison of MFEs and SR of SRCGA and the New RCGAs for Problems with Dimensions 5 - 20

Pno.	Problem name	n	Mean Function Evaluations (MFE)				Success Rate (SR)			
			SRCGA	RCGA-PS	RCGA-PS-P	RCGA-P	SRCGA	RCGA-PS	RCGA-PS-P	RCGA-P
34	Levy and Montalvo 2	5	18,036	4,786	4,917	630,803	98	100	100	38
35	Salomon	5	-	1,387,240	1,874	450	0	1	100	100
36	Shekel's Foxholes 5	5	466,254	1,150,774	1,881,333	-	7	18	1	0
37	Hartman 6	6	92,323	27,783	4,813	12,421	86	99	100	100
38	Storn's Tchebychev 9	9	180	342	433	270	100	100	100	100
39	Ackley	10	1,000,100	49,618	1,988	1,276	100	100	100	100
40	Exponential	10	16,506	12,105	484	308	100	100	100	100
41	Griewank	10	1,000,100	2,374,582	1,455	878	2	52	100	100
42	Levy and Montalvo 2	10	70,913	14,524	204,220	1,887,732	96	100	95	6
43	Modified Langerman	10	-	2,181,129	3,537,542	-	0	23	7	0
44	Neumaier 3	10	999,510	120,170	153,944	326,904	100	100	100	100
45	Paviani	10	132,825	655,504	1,226,774	445,250	100	100	100	100
46	Rastrigin	10	-	2,634,645	1,239	758	0	6	100	100
47	Rosenbrock	10	1,000,100	2,779,964	3,799,916	-	1	81	35	0
48	Salomon	10	-	-	1,804	1,136	0	0	100	100
49	Schwefel	10	-	1,668,610	2,600,115	-	0	41	32	0
50	Shekel's Foxholes 10	10	-	2,717,235	-	-	0	3	0	0
51	Sinusoidal 10	10	39,602	15,646	13,162	349,758	100	100	100	87
52	Spherical	10	30,913	15,074	714	446	100	100	100	100
53	Storn's Tchebychev 17	17	340	644	817	510	100	100	100	100
54	Sinusoidal 20	20	1,396,882	53,946	50,008	3,889,840	100	100	100	3
	Total :		**6,264,584**	**17,864,321**	**13,487,552**	**7,548,740**	**1,190**	**1,424**	**1,670**	**1,334**

2 and 3. The total MFE, SR and SR ratio (in round brackets) for the algorithms are presented in Table 4. SR ratio is calculated by dividing SR by the total number of runs, i.e. 100. Tables 2 and 3 show that each of RCGA-PS and RCGA-PS-P solved 53 problems. RCGA-P and SRCGA solved 49 and 48 problems, respectively.

Total SR in Table 4 shows that RCGA-PS-P is superior to RCGA-PS, RCGA-P and SRCGA by 366, 511 and 1,018 successes, respectively. RCGA-PS is superior to RCGA-P and SRCGA by 145 and 652 successes, respectively. RCGA-P is superior to SRCGA by 507 successes. Clearly, RCGA-PS-P is the best performer and RCGA-PS is the runner up followed by RCGA-P with respect to SR. This is reflected in the success ratio which shows that RCGA-PS-P is the best algorithm with 92% success, followed by RCGA-PS, RCGA-P and SRCGA with $85\%, 83\%$ and 73% successes, respectively.

Table 4: Total results of MFE and SR from Tables 2 and 3

Measure	SRCGA	RCGA-PS	RCGA-PS-P	RCGA-P
MFE	8,763,122	20,787,009	16,372,583	11,011,527
SR	3949 (0.73)	4601 (0.85)	4967 (0.92)	4456 (0.83)

Total MFE of SRCGA seems superior to the total MFE of all the new algorithms in Table 4, but taking a closer look at columns 3 to 6, of Tables 2 and 3, one would observe that SRCGA has not solve 6 problems that RCGA-PS and RCGA-PS-P are able to solve. By excluding MFE of all the algorithms for problems that at least one algorithm is not able to solve, a different picture emerges. These problems are; Modified Langerman, Rastrigin, Rosenbrock, Salomon 5, Salomon 10, Schwefel, Shekel's foxholes 5 and Shekel's foxholes 10. After excluding MFE of these 8 problems, the remaining 46 problems are used to determine a new set of results for comparing the performances of the algorithms. The new results are summarized in Table 5.

Table 5: Comparison of SRCGA and the New RCGAs using total results for 46 test problems

Measure	SRCGA	RCGA-PS	RCGA-PS-P	RCGA-P
MFE	7,296,768	6,267,412	4,548,760	11,009,183
SR	3941 (0.86)	4428 (0.96)	4592 (0.99)	4156 (0.90)

In Table 5, RCGA-PS-P emerged as the best algorithm with a total of 4592 SR out of 4600 runs, followed by RCGA-PS, RCGA-P and SRCGA with 4428, 4156, 3941 respectively. In terms of MFE, RCGA-PS-P is also the best with the smallest MFE. RCGA-P has the highest MFE because it performed badly on problems 24 and 51. If we compare the total results by excluding these two problems then RCGA-P becomes superior to SRCGA by 597,362 MFE.

The normalized SP of the four algorithms are presented in Appendix I. SP of the best algorithm (SP_{best}) is listed in column 2, while normalized SP for all algorithms are listed in columns 3 - 6. The symbol '- ' is used to indicate nil. The normalized SP's printed in bold from Appendix I indicate the best performing algorithm(s) for the corresponding problem. From Appendix I, RCGA-PS-P emerged the best with the least normalized SP, after excluding the SP for problems where at least one algorithm has no SP. The runner-up is RCGA-P, followed by RCGA-PS and SRCGA respectively.

In appendix II, 50 out of 52 ANOVA p-values in the last column indicate that there are significant differences between MFE of the algorithms. ANOVA test was not carried out for problems 52 and 53 because the algorithms have the same MBF values of 0.0. The symbol '- ' is used to indicate nil. This statistical information provides a clear evidence that the algorithms differ in their performances. These differences on some selected problems can be seen graphically in Figures 2(a-c) and Figure 3(a-c) in appendix IV and V. These figures are discussed later on in this section.

The best fitness values (Min) and the worst fitness values (WF) for each algorithm for all the test problems are presented in appendix III. Algorithms whose Min values are printed in bold outperformed the others. A careful study of appendix III shows that RCGA-PS-P is the most superior and robust algorithm because it is able to accurately locate 45 out of 46 global minima. RCGA-PS, RCGA-P and SRCGA are able to locate, accurately, 43, 40 and 36 global minima respectively.

Next, the algorithms are compared graphically using the box plots and multiple comparisons from the ANOVA test on some representative problems. Figures 2(a-c) show the box plots for the algorithms on Ackley, Griewank and Modified Langerman. Figures 2(a-b) show that the MBF of RCGA-PS-P and RCGA-P are closest to the global minimum with very small solution spread around their means. The worst algorithm on Ackley and Griewank problems is SRCGA with a wide dispersion of solutions. On the other hand Figure 2(c) shows that RCGA-PS is the best algorithm on Modified Langerman (a problem with multi modal and non symmetrical properties) but with a wide spread of solutions. It is followed by RCGA-PS-P.

Figures 3(a-c) show the multiple comparisons between the algorithms on the same problems. They show that the algorithms are significantly different. From the multiple comparison plots in Figures 3(a-b), we see that RCGA-PS-P and RCGA-P performed better than the other two algorithms. Figure 3(c) shows that RCGA-PS outperformed the other algorithms.

Finally, we now rank the algorithms using total MFE and SR from Table 5 and SP from Appendix I. The rankings are presented in Table 6 which clearly shows the superiority of RCGA-PS-P.

Table 6: Rank order of SRCGA and the New RCGAs

Rank	**1st**	**2nd**	**3rd**	**4th**
MFE	RCGA-PS-P	RCGA-PS	SRCGA	RCGA-P
SR	RCGA-PS-P	RCGA-PS	RCGA-P	SRCGA
SP	RCGA-PS-P	RCGA-P	RCGA-PS	SRCGA

Finally, we study the performance of the algorithms presented in this papers on a number of scalable functions [2] e.g. Ackley, Rastrigin, Rosenbrock and Schewfel. We have tested our algorithms on these problem for $n = 10, 20$ and 30 and the summarized results are presented in Tables 7 and 8. Tables 7 and 8 show that Ackley, Rastrigin and Spherical problems were consistently solved by all the new algorithm proposed. The algorithms however failed for Rosenbrock and Schwefel problems. Table 8 also shows that MFE increases with n. However, this increase in MFE is not quite significant in the new algorithms proposed. This clearly shows that the best performing algorithm has a role to play in global optimization.

[2] Griewank is also a scalable function but this problem becomes easier as its dimension increases, see ([32]). Our

Table 7: Comparison of Min and SR of SRCGA and the New RCGAs on scalable problems

Pno.	$f(x^*)$	Min				SR			
		SRCGA	RCGA-PS	RCGA-PS-P	RCGA-P	SRCGA	RCGA-PS	RCGA-PS-P	RCGA-P
Dimension = 10									
39	0.0000	8.08E-04	5.50E-05	**0.00E+00**	**1.00E-06**	100	100	100	100
46	0.0000	9.95E-01	2.50E-05	**0.00E+00**	**0.00E+00**	0	6	100	100
47	0.0000	7.75E-03	9.20E-05	1.12E-03	1.64E+00	1	81	35	0
49	-4189.8289	-3.62E+03	**-4.19E+03**	**-4.19E+03**	-3.42E+03	0	41	32	0
52	0.0000	4.20E-05	1.50E-05	**0.00E+00**	**0.00E+00**	100	100	100	100
Dimension = 20									
39	0.0000	1.21E-02	8.20E-05	**0.00E+00**	**0.00E+00**	0	100	100	100
46	0.0000	4.98E+00	2.98E+00	**0.00E+00**	**0.00E+00**	0	0	100	100
47	0.0000	1.12E+00	4.19E-02	3.30E-01	1.14E+01	0	0	0	0
49	-8379.6578	-6.03E+03	**-8.38E+03**	-8.26E+03	-5.56E+03	0	1	0	0
52	0.0000	5.90E-05	4.80E-05	**0.00E+00**	**0.00E+00**	100	100	100	100
Dimension = 30									
39	0.0000	2.65E-02	2.37E-04	**1.00E-06**	**0.00E+00**	0	100	100	100
46	0.0000	1.49E+01	9.95E+00	**0.00E+00**	**0.00E+00**	0	0	100	100
47	0.0000	6.73E+00	**1.28E-03**	8.12E+00	2.12E+01	0	2	0	0
49	-12569.4867	-8.56E+03	-1.21E+04	-1.23E+04	-8.13E+03	0	0	0	0
52	0.0000	7.00E-05	6.20E-05	**0.00E+00**	**0.00E+00**	100	100	100	100

Table 8: Comparison of MBF and MFE of SRCGA and the New RCGAs on scalable problems

Pno.	$f(x^*)$	MBF of successful runs				MFE of successful runs			
		SRCGA	RCGA-PS	RCGA-PS-P	RCGA-P	SRCGA	RCGA-PS	RCGA-PS-P	RCGA-P
Dimension = 10									
39	0.0000	2.57E-03	9.00E-05	3.40E-05	4.00E-05	1,000,100	49,618	1,988	1,276
46	0.0000	-	6.83E-05	2.00E-05	2.50E-05	-	42,062	1,239	758
47	0.0000	7.75E-03	4.84E-04	6.62E-03	-	1,000,100	2,775,232	3,799,851	-
49	-4189.8289	-	-4.19E+03	-4.19E+03	-	-	40,401	50,023	-
52	0.0000	8.50E-05	7.20E-05	1.70E-05	2.60E-05	30,913	15,074	714	446
Dimension = 20									
39	0.0000	-	9.60E-05	3.70E-05	3.90E-05	-	1,834,080	2,044	1,252
46	0.0000	-	-	2.50E-05	3.10E-05	-	-	1,311	739
47	0.0000	-	-	-	-	-	-	-	-
49	-8379.6578	-	-8.38E+03	-	-	-	234,214	-	-
52	0.0000	9.20E-05	8.50E-05	2.20E-05	2.40E-05	202,580	25,961	732	448
Dimension = 30									
39	0.0000	-	6.23E-04	3.60E-05	3.60E-05	-	2,799,978	1873	1126
46	0.0000	-	-	2.20E-05	2.80E-05	-	-	1212	730
47	0.0000	-	1.34E-03	-	-	-	2,800,846	-	-
49	-12569.4867	-	-	-	-	-	-	-	-
52	0.0000	9.60E-05	9.10E-05	1.50E-05	1.90E-05	762,717	41,550	700	412

5.2 Comparison of RCGA-PS-P with recent GAs

In this section, the best performing algorithm, RCGA-PS-P, is compared with some recent genetic algorithms from the literature. The results of the comparisons are summarized in Tables 9 and 10. The problem numbers used in Tables 2 and 3 are used to represent the problems used in Tables 9 and 10. We use the test problems that are common to the test problems we have used and the ones used in [30, 31]. For the purpose of fair comparison, we used the same parameters that were used in [30, 31]. These parameter settings are provided below.

The parameter settings used for solving the problems taken from [31] are N (given in Table 9), and T = 3000. GAs used in [31] are classical genetic algorithm (CGA) and a hybrid RCGA with quasi-simplex technique (RCGAQS).

Table 9: Comparison of RCGA-PS-P with two GA algorithms from [31] on common problems

Pno	N	$f(x^*)$	Min			STD		
			RCGA-PS-P	RCGAQS	CGA	RCGA-PS-P	RCGAQS	CGA
39	110	0.0000	**4.4408E-16**	3.9697E-01	3.6244E+00	0.00E+00	6.5006E-01	1.9395E-03
41	165	0.0000	**0.00E+00**	3.4927E-02	7.8300E-01	0.00E+00	1.7985E-02	6.3444E-02
52	110	0.0000	**4.5639E-241**	3.2030E-29	9.70E+00	0.00E+00	1.0128E-28	3.8623E-01

numerical results also confirmed this.

The results in Table 9 show that RCGA-PS-P outperforms RCGAQS and CGA with respect to the number of problems solved. The best solutions in Table 9 are printed in bold. The STD for RCGA-PS-P is zero for all problems. This means that all the solutions found by RCGA-PS-P are better than the ones found by RCGAQS and CGA.

Next, we compare RCGA-PS-P with GAs presented in [30]. These are standard binary coded genetic algorithm (SBGA) and enhanced binary coded genetic algorithm (EBGA). Again, for a fair comparison we use N=200 (fixed for all problems in [30]) and T = 500 to obtain our results. The comparison is presented in Table 10.

Table 10: Comparison of RCGA-PS-P with two GA algorithms from [30] on nine common test problems

Pno	$f(x^*)$	Min			MFE			SR		
		RCGA-PS-P	SBGA	EBGA	RCGA-PS-P	SBGA	EBGA	RCGA-PS-P	SBGA	EBGA
5	0.39789	**0.39789**	**0.39789**	0.39791	7,133	8,125	**2,040**	100	81	100
7	-1.03163	**-1.03163**	**-1.03163**	**-1.03163**	4,459	**1,316**	**1,316**	100	98	100
11	3.00000	**3.00000**	**3.00000**	3.00028	11,211	8,185	**4,632**	100	59	100
19	-186.73091	**-186.73091**	-186.73100	-186.72802	125,062	6,976	**2,364**	100	93	100
21	-3.86278	**-3.86270**	-3.86249	-3.86114	5,544	1,993	**1,680**	100	94	100
30	-10.15320	**-10.15316**	-10.13490	-10.14866	25,675	**7,495**	36,388	100	1	97
31	-10.40294	**-10.40293**	-10.16770	-10.38253	**22,201**	-	36,774	100	0	98
32	-10.53641	**-10.53639**	-10.40340	-10.51404	**19,239**	-	36,772	100	0	100
37	-3.32237	**-3.32236**	-3.30652	-3.31383	**11,476**	19,452	53,792	100	23	92
	Total :				**232,000**	**53,542**	**175,758**	**900**	**449**	**887**

Table 10 shows the best function values, mean function value and success rate of RCGA-PS-P, SBGA and EBGA [30]. The algorithm that performed the best has its results printed in bold. Table 10 shows that RCGA-PS-P is the best performer with respect to the quality solutions than SBGA and EBGA. In terms of MFE, EBGA performed better in 5 problems, but at the cost of sacrificing accuracy in some lower dimensional problems. RCGA-PS-P becomes superior in terms of MFE as the problem dimension increases. It performed better in problems 31, 32 and 37. Although SBGA has the lowest MFE, its overall performance is the worst among the algorithms presented in Table 10.

In terms of SR, RCGA-PS-P is superior to EBGA and SBGA by 13 and 451 success respectively. Overall comparisons show that RCGA-PS-P is superior, more reliable and more robust than the other algorithms.

5.3 Discussion

In the last two sections, we have compared the new algorithms with SRCGA. We have also compared the best performer, RCGA-PS-P, with recent genetic algorithms from the literature. The new algorithms were also compared with SRCGA using scalable problems. The comparison of SRCGA, RCGA-PS, RCGA-P and RCGA-PS-P shows that each of PS-based crossover and P-based exploratory search improved the performance of RCGA, but their combination in RCGA-PS-P is the best. The multiple comparison plots also illustrate significant differences between the new algorithms and SRCGA. We have also shown that RCGA-PS-P performed better than the recent algorithms from the literature.

We attribute the performance of the new algorithms to the inclusion of the PS-based crossover operation and the P-based exploratory search method in RCGA. The PS-based crossover operation enhances the local exploitation of the solution space. The P-based exploratory search property, on the other hand, is used to explore a wider range of the search region when the solution points in the population set are far apart. As soon as the solution points are within a local basin, it intensify the search within the local neighborhood of the solutions.

6 Conclusion

We have proposed three new versions of real coded genetic algorithm and tested their performances on a large set of test problems. Numerical comparison have shown that all new versions are better than standard real coded genetic algorithm. We have also demonstrated the superiority of the new versions over the standard real coded genetic algorithm using graphs and p-values from ANOVA.

Finally, we have compared the best performing real coded genetic algorithm with two versions of genetic algorithms from the literature. This comparison again confirmed the superiority of the new real coded genetic algorithm over its recent competitors.

We have discussed the new features introduced in the genetic algorithm and shown that the features are contributory to the superior performance. The new algorithms can be implemented at ease for solving continuous global optimization problems.

New research is underway to design an efficient real coded genetic algorithm for constrained global optimization.

References

[1] M. M. Ali, A. Törn, A population set-based global optimization algorithms: some modifications and numerical studies, *Computers & Operations Research*, 2004; 31(10), 1703-1725.

[2] F. van den Bergh, An analysis of particle swarm optimizers, *PhD Thesis*, University of Pretoria, South Africa, 2001.

[3] A. P. Engelbrecht, *Fundamental of computational swarm intelligence*. John Wiley & Sons, Ltd, West Sussex, England, 2005.

[4] M. G. H. Omran, Particle swarm optimization methods for pattern recognition and image processing, *PhD Thesis*, University of Pretoria, South Africa, 2005.

[5] M. M. Ali, C. Khompatraporn, Z. B. Zabinsky, A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems, *Journal of Global Optimization*, 2005; 31, 635-672.

[6] T. Back, *Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms*, Oxford University press, New York, 1996.

[7] I. Rechenberg, Artificial evolution and artificial intelligence, *Machine learning principle and techniques*. Forsyth, R. (Ed) Chapman and Hall, London, 1989.

[8] J. H. Holland, *Adaptation in natural and artificial systems, An introductory analysis with applications to biology, control and artificial intelligence*. MIT press/ Bradford Books edition Cambridge, Massachusetts, 1975.

[9] R. Storn, K. Price, Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997; 2, 341-359.

[10] Z. Michalewicz, *Genetic algorithms + data structures = evolution programs*, Springer-Verlag, Berlin Heidelberg, N.Y., 1996.

[11] P. Kaelo, Some population set-based methods for unconstrained global optimization, *PhD thesis*, School of Computational and Applied Mathematics, University of the Witwatersrand, South Africa, 2005.

[12] K. Deep, M. Thakur, A new crossover operator for real coded genetic algorithms, *Applied Mathematics and Computation*, 2007; 188, 895-911.

[13] K. Deep, M. Thakur, A new mutation operator for real coded genetic algorithms, *Applied Mathematics and Computation*, 2007; 193, 211-230.

[14] K. Deep, M. Thakur, A real coded multi parent genetic algorithms for function optimization, *Journal of Hybrid Computing Research*, 2008; 1(2), 67-83.

[15] J. Yen, B. Lee, A simplex genetic algorithm hybrid. Evolutionary computation, *IEEE international conference on evolutionary computation*, 1997; 175 - 180.

[16] R. Chelouah, P. Siarry, Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions, *European Journal of Operational Research*, 2003; 148(2), 335-348.

[17] M. Settles, T. Soule, Breeding Swarms: A GA/PSO hybrid, *In genetic and evolutionary computation - GECCO 2005*, Washington, D.C., July 2005.

[18] K. Deep, Dipti, A new hybrid self organizing migrating genetic algorithm for function optimization, *Proc of the IEEE Congress on Evolutionary Computations*, September 25-28, 2007; Singapore, 2796-2803.

[19] K. Deep, K. N. Das, Quadratic approximation based hybrid genetic algorithm for function optimization, *Applied Mathematics and Computation*, 2008; 203, 86-98.

[20] D.E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, Reading, Massachusetts, 1989.

[21] J. E. Baker, Reducing bias and inefficiency in the selection algorithm, In: Grefenstette JJ, editor. *Proc of the Second Int Conf on Genetic Algorithms and their Applications*. Hillsdale, MA: L. Erlbraum Associates, 1987; 14-21.

[22] J. E. Baker, An analysis of the effects of selection in genetic algorithms, *Doctoral dissertation*, Vanderbilt University, Nashville, TN, 1989.

[23] K. A. De Jong, An analysis of the behavior of a class of genetic adaptive systems, *Doctoral dissertation*, University of Michgan, Ann Arbor, MI. Dissertation abstract international, 1975; 36(10): 5140B.

[24] J. E. Dennis, V. J. Torczon, Derivative-free pattern search methods for Multidisciplinary design problems, *Proc of the AIAA/NASA/USAF/ISSMO symposium on Multidisciplinary Analysis and Optimization*, September 7-9, 1994; Panama City Beach, Florida.

[25] C. Audet, J. E. Dennis, Analysis of generalized pattern search, *SIAM journal on optimization*, 2004; 13(3), 889-903.

[26] M. N. Gabere, Simulated annealing driven pattern search algorithms for global optimization, *MSc Thesis*, School of Computational and Applied Mathematics, University of the Witwatersrand, South Africa, 2007.

[27] M. M. Ali, Z. Kajee-Bagdadi, A local exploration-based differential evolution algorithm for constrained global optimization, *Applied Mathematics and Computation*, 2009; 208, 31-48.

[28] A. Auger, N. Hansen, Performance evaluation of an advanced local search evolutionary algorithm, *Proc of the IEEE Congress on Evolutionary Computation*, CEC 2005; 1777-1784.

[29] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. Coello, K. Deb, Problem definitions and evaluation criteria for the CEC 2006, *Special session on constrained real-parameter optimization*, Technical report, 2006.

[30] J. Andre, P. Siarry, T. Dognon, An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization, *Advances in Engineering Software*, 2001; 32, 49-60.

[31] G. Zhang, H. Lu, Hybrid real coded genetic algorithm with quasi-simplex technique, *Journal of Computer Science and Network Security*, 2006; 6(10), 246-255.

[32] M. Locatelli, A Note on the Griewank Test Function, *Journal of Global Optimization*, 27, 169-174, 2003.

Appendix I: Comparison of Normalized SP of SRCGA and the New RCGAs

Pno.	SP_{best}	**SRCGA**	**RCGA-PS**	**RCGA-PS-P**	**RCGA-P**
1	911	1.5906	1.0845	**1.0000**	1.0637
2	1602	**1.0000**	3.3539	4.0031	1.0993
3	630	4.6460	2.4873	1.3000	**1.0000**
4	558	5.6186	2.8781	1.4624	**1.0000**
5	1374	3.1718	4.7431	6.1703	**1.0000**
6	275	4.3782	3.7855	1.8655	**1.0000**
7	1253	**1.0000**	2.0231	2.9633	1.3567
8	248	2.8105	3.4234	1.3952	**1.0000**
9	4765	2.1553	**1.0000**	1.4923	47.2134
10	1197	1.7845	**1.0042**	**1.0000**	1.4353
11	1576	3.8359	**1.0000**	1.0812	55.7789
12	740	1.3716	**1.0216**	**1.0000**	1.1459
13	1063	1.3603	**1.0000**	1.1063	2.8184
14	78495	1.2925	2.6887	**1.0000**	**1.0434**
15	802	4.5241	3.9816	2.5436	**1.0000**
16	374	8.6014	11.2121	2.7112	**1.0000**
17	745	80.2685	29.1323	3.6337	**1.0000**
18	376127	3.3237	1.7724	1.9206	**1.0000**
19	5854.55	**1.0000**	13.1051	8.2995	9.2852
20	232275	1.0854	**1.0000**	1.8009	1.7344
21	1832	1.7003	1.2844	**1.0000**	13.3575
22	2738	1.4255	1.2392	**1.0000**	40.6676
23	2391	**1.0343**	**1.0845**	**1.0000**	2.8536
24	4897	36.2953	**1.0000**	2.4092	62.1511
25	256	13.3594	12.6953	1.6719	**1.0000**
26	421	1.3064	2.1829	1.5914	**1.0000**
27	1980	9.9242	**1.0000**	1.6106	44.1162
28	923869	**1.0561**	**1.0000**	1.3721	2.3248
29	435	204.3379	16.8690	2.9885	**1.0000**
30	5263	2.5872	1.8539	1.2548	**1.0000**
31	4790.53	2.2622	1.6314	**1.0616**	**1.0000**
32	4869.57	1.9861	1.4484	1.0629	**1.0000**
33	178309	32.0503	**1.0000**	1.2951	10.9121
34	4786	1.7481	**1.0000**	**1.0274**	15.5866
35	450	-	25879.5556	4.1644	**1.0000**
36	85527.8	2.8824	**1.0000**	21.4761	-
37	4813	2.3357	2.3287	**1.0000**	2.5807
38	180	**1.0000**	1.9000	2.4056	1.5000
39	1276	783.7774	38.8856	1.5580	**1.0000**
40	308	53.5909	39.3019	1.5714	**1.0000**
41	878	56953.3030	4340.4372	1.6572	**1.0000**
42	14524	2.3092	**1.0000**	**1.0831**	146.0801
43	472487	-	**1.0000**	1.4857	-
44	120170	8.3175	**1.0000**	1.2811	2.7203
45	132825	**1.0000**	4.9351	9.2360	3.3522
46	758	-	924.8461	1.6346	**1.0000**
47	3426212	29.1897	**1.0000**	3.1688	-
48	1136	-	-	1.5880	**1.0000**
49	98539	-	**1.0000**	1.5864	-
50	1205067	-	**1.0000**	-	-
51	13162	3.0088	1.1887	**1.0000**	9.0084
52	446	69.3117	33.7982	1.6009	**1.0000**
53	340	**1.0000**	1.8941	2.4029	1.5000
54	50008	27.9332	**1.0787**	**1.0000**	214.3210
	Total	58,384.8516	31,414.1359	128.9949	719.0070

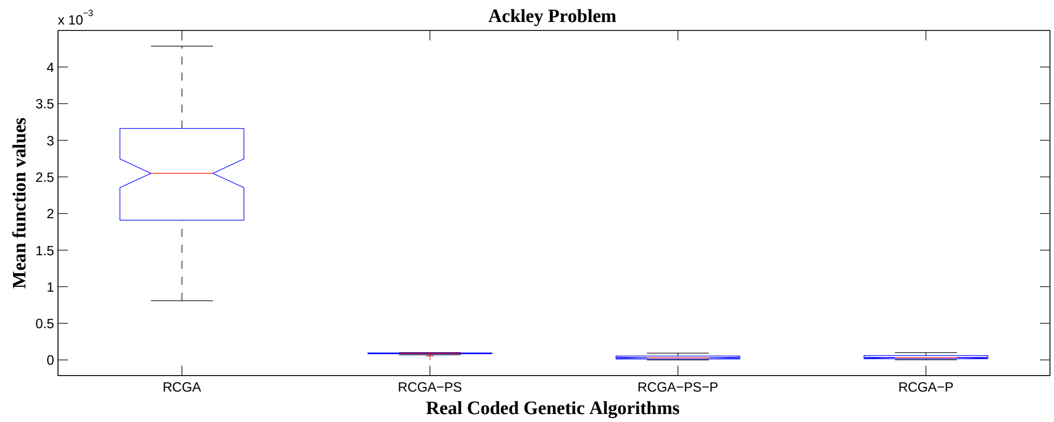
Appendix II: Comparison of the Means, Standard Deviations and p-values of SRCGA and the New RCGAs

Pno.	Problem name	n	$f(x^*)$	Mean Best Fitness value (MBF)				Standard Deviations (STD)				p-value
				SRCGA	RCGA-PS	RCGA-PS-P	RCGA-P	SRCGA	RCGA-PS	RCGA-PS-P	RCGA-P	
1	Aluffi-Pentini	2	-0.3523	-3.52E-01	-3.52E-01	-3.52E-01	-3.52E-01	6.00E-05	5.50E-05	5.30E-05	5.80E-05	1.42E-03
2	Becker and Lago	2	0.0000	5.10E-05	4.80E-05	4.40E-05	4.70E-05	3.00E-05	3.30E-05	2.90E-05	2.80E-05	4.19E-01
3	Bohachevsky 1	2	0.0000	5.10E-05	4.00E-05	2.40E-05	3.30E-05	3.00E-05	2.90E-05	2.60E-05	3.00E-05	1.72E-09
4	Bohachevsky 2	2	0.0000	1.97E-02	7.02E-07	2.70E-05	2.60E-05	6.25E-02	2.80E-05	2.90E-05	2.70E-05	2.98E-06
5	Branin	2	0.3979	3.98E-01	3.98E-01	3.98E-01	3.98E-01	2.90E-05	3.00E-05	3.10E-05	2.90E-05	4.02E-03
6	Camel Back-3	2	0.0000	5.30E-05	4.10E-05	3.50E-05	2.60E-05	2.90E-05	2.90E-05	3.20E-05	2.90E-05	5.93E-09
7	Camel Back-6	2	-1.0316	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	3.80E-05	3.80E-05	4.10E-05	3.80E-05	2.69E-04
8	Cosine Mixture	2	-0.2000	-2.00E-01	-2.00E-01	-2.00E-01	-2.00E-01	3.10E-05	3.10E-05	2.80E-05	2.90E-05	5.19E-09
9	Dekkers and Aarts	2	-24776.5183	-2.48E+04	-2.48E+04	-2.48E+04	-2.48E+04	4.00E-05	2.90E-05	3.00E-05	1.12E-01	3.37E-03
10	Easom	2	-1.0000	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	3.00E-05	3.00E-05	3.00E-05	3.10E-05	2.81E-02
11	Goldstein and Price	2	3.0000	3.81E+00	3.00E+00	3.00E+00	5.16E+00	4.61E+00	3.00E-05	3.10E-05	7.32E+00	1.05E-03
12	Hosaki	2	-2.3458	-2.35E+00	-2.35E+00	-2.35E+00	-2.35E+00	3.30E-05	3.10E-05	3.10E-05	3.40E-05	1.79E-03
13	McCormick	2	-1.9133	-1.91E+00	-1.91E+00	-1.91E+00	-1.91E+00	7.00E-06	7.00E-06	6.00E-06	7.00E-06	9.20E-04
14	Modified Rosenbrock	2	0.0000	5.16E-03	2.64E-03	7.87E-04	1.53E-03	6.66E-03	3.52E-03	2.21E-03	2.95E-03	2.01E-12
15	Multi-Gaussian	2	-1.2969	-1.26E+00	-1.29E+00	-1.30E+00	-1.30E+00	4.02E-02	2.51E-02	2.90E-05	3.10E-05	0.00E+00
16	Periodic	2	0.9000	9.35E-01	9.10E-01	9.00E-01	9.00E-01	4.77E-02	3.00E-02	2.60E-05	2.80E-05	0.00E+00
17	Schaffer 1	2	0.0000	7.82E-02	7.01E-03	1.32E-04	3.40E-05	9.80E-03	4.34E-03	9.64E-04	3.00E-05	0.00E+00
18	Schaffer 2	2	0.0012	6.25E-02	4.91E-03	1.07E-03	4.17E-04	9.98E-02	5.29E-03	1.25E-03	2.41E-04	0.00E+00
19	Shubert	2	-186.7309	-1.86E+02	-1.87E+02	-1.87E+02	-1.80E+02	6.28E+00	3.18E-03	2.63E-04	1.98E+01	1.12E-06
20	Gulf Research	3	0.0000	9.73E-04	9.30E-05	9.40E-05	6.39E-04	1.14E-03	3.50E-05	4.60E-05	1.41E-03	1.43E-13
21	Hartman 3	3	-3.8628	3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	2.60E-05	2.70E-05	2.70E-05	2.70E-05	2.19E-02
22	Helical Valley	3	0.0000	6.70E-05	5.90E-05	5.40E-05	6.30E-05	2.70E-05	2.50E-05	2.90E-05	3.40E-05	7.60E-03
23	Levy and Montalvo 1	3	0.0000	6.10E-05	6.10E-05	5.40E-05	6.60E-05	2.60E-05	2.80E-05	2.70E-05	3.10E-05	1.97E-02
24	Meyer and Roth	3	0.0019	2.36E-03	1.97E-03	1.97E-03	2.35E-03	6.77E-04	2.30E-05	2.40E-05	6.98E-04	2.45E-12
25	Cosine Mixture	4	-0.4000	-4.00E-01	-4.00E-01	-4.00E-01	-4.00E-01	2.40E-05	2.50E-05	2.40E-05	2.70E-05	0.00E+00
26	Kowalik	4	0.0003	3.73E-04	3.70E-04	3.69E-04	3.75E-04	2.40E-05	2.50E-05	2.50E-05	2.30E-05	3.05E-01
27	Miele and Cantrell	4	0.0000	9.20E-05	5.30E-05	6.50E-05	9.60E-05	1.10E-05	2.90E-05	2.80E-05	1.00E-05	0.00E+00
28	Neumaier 2	4	0.0000	1.89E-02	6.37E-04	5.51E-04	1.87E-02	1.64E-02	1.06E-03	6.62E-04	1.48E-02	0.00E+00
29	Powell's Quadratic	4	0.0000	8.10E-05	6.90E-05	2.90E-05	2.30E-05	2.20E-05	2.60E-05	2.60E-05	2.80E-05	0.00E+00
30	Shekel 5	4	-10.1532	-6.78E+00	-9.70E+00	-1.02E+01	-1.02E+01	3.68E+00	1.45E+00	2.90E-05	2.40E-05	0.00E+00
31	Shekel 7	4	-10.4029	-8.11E+00	-1.02E+01	-1.03E+01	-1.01E+01	3.30E+00	9.02E-01	6.65E-01	1.46E+00	0.00E+00
32	Shekel 10	4	-10.5364	-9.25E+00	-1.05E+01	-1.05E+01	-9.96E+00	2.67E+00	2.70E-05	6.67E-01	1.97E+00	1.17E-07
33	Wood	4	0.0000	1.70E+00	9.20E-05	9.10E-05	1.19E+00	2.25E+00	1.30E-05	1.60E-05	2.12E+00	0.00E+00
34	Levy and Montalvo 2	5	0.0000	2.92E-04	6.50E-05	6.50E-05	2.87E-02	1.53E-03	2.30E-05	2.30E-05	4.23E-02	0.00E+00
35	Salomon	5	0.0000	1.06E-01	9.89E-02	4.20E-05	3.90E-05	2.37E-02	9.93E-03	3.00E-05	2.60E-05	0.00E+00
36	Shekel's Foxholes	5	-10.4056	-3.11E+00	-4.14E+00	-2.80E+00	-2.67E+00	2.04E+00	2.95E+00	7.82E-01	1.29E-01	6.31E-10
37	Hartman 6	6	-3.3224	-3.31E+00	-3.32E+00	-3.32E+00	-3.32E+00	4.13E-02	1.85E-03	2.10E-05	2.20E-05	7.53E-09
38	Storn's Tchebychev	9	0.0000	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-
39	Ackley	10	0.0000	2.57E-03	9.00E-05	3.40E-05	4.00E-05	7.49E-04	8.00E-06	2.60E-05	2.90E-05	0.00E+00
40	Exponential	10	-1.0000	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	1.20E-05	1.80E-05	1.40E-05	1.30E-05	0.00E+00
41	Griewank	10	0.0000	6.25E-02	9.58E-03	2.00E-05	2.30E-05	3.54E-02	8.65E-03	2.40E-05	2.70E-05	0.00E+00
42	Levy and Montalvo 2	10	0.0000	5.21E-04	7.50E-05	6.20E-04	1.13E-01	2.14E-03	1.70E-05	2.38E-03	1.29E-01	0.00E+00
43	Modified Langerman	10	-0.9650	-1.89E-01	-6.11E-01	-4.53E-01	-1.77E-01	1.45E-01	2.31E-01	1.84E-01	1.22E-01	0.00E+00
44	Neumaier 3	10	-210.0000	-2.10E+02	-2.10E+02	-2.10E+02	-2.10E+02	3.51E-04	7.00E-06	1.50E-05	9.00E-06	0.00E+00
45	Paviani	10	-45.7780	-4.58E+01	-4.58E+01	-4.58E+01	-4.58E+01	1.52E-04	2.08E-04	2.24E-04	1.83E-04	2.10E-06
46	Rastrigin	10	0.0000	7.06E+00	2.59E+00	2.00E-05	2.50E-05	3.30E+00	1.42E+00	2.70E-05	2.80E-05	0.00E+00
47	Rosenbrock	10	0.0000	1.24E+01	5.81E-01	9.83E-03	2.04E+00	3.34E+01	1.38E+00	2.87E-03	1.91E-01	1.37E-07
48	Salomon	10	0.0000	1.16E-01	9.99E-02	3.80E-05	4.00E-05	3.67E-02	0.00E+00	2.90E-05	2.70E-05	0.00E+00
49	Schwefel	10	-4189.8289	-2.86E+03	-4.07E+03	-4.05E+03	-2.36E+03	2.89E+02	1.28E+02	1.36E+02	3.27E+02	0.00E+00
50	Shekel's Foxholes	10	-10.2088	-1.48E+00	-1.85E+00	-1.48E+00	-1.48E+00	2.19E-02	1.51E+00	2.50E-02	0.00E+00	0.00E+00
51	Sinusoidal	10	-3.5000	-3.50E+00	-3.50E+00	-3.50E+00	-3.36E+00	1.30E-05	1.80E-05	1.80E-05	4.05E-01	0.00E+00
52	Spherical	10	0.0000	8.50E-05	7.20E-05	1.70E-05	2.60E-05	2.19E-05	1.70E-05	2.50E-05	3.00E-05	0.00E+00
53	Storn's Tchebychev	17	0.0000	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-
54	Sinusoidal	20	-3.5000	-3.50E+00	-3.50E+00	-3.50E+00	-1.07E+00	1.20E-05	1.00E-05	1.10E-05	4.26E-01	4.93E-07

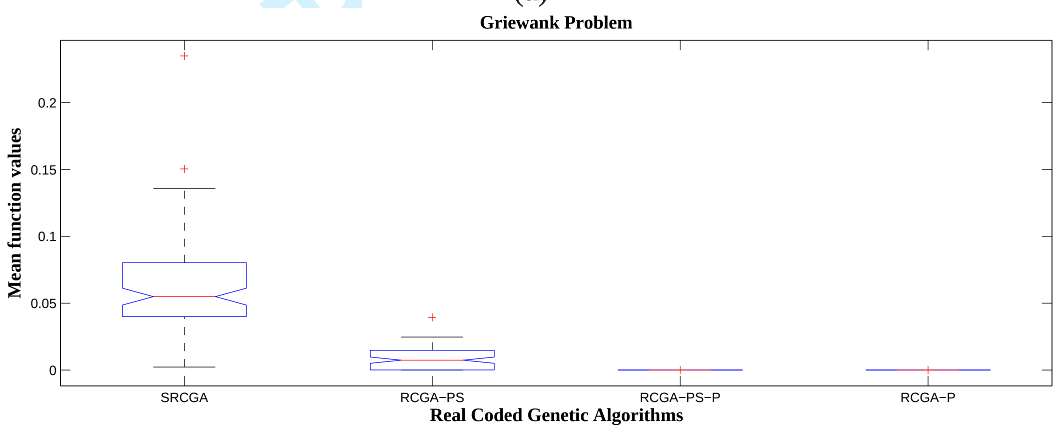
Appendix III: Comparison of the Best Min Fitness values and Worst Fitness values of SRCGA and the New RCGAs

Sno.	Problem name	n	$f(x^*)$	Minimum Best Fitness value (Min)				Worst Fitness Value (WF)			
				SRCGA	**RCGA-PS**	**RCGA-PS-P**	**RCGA-P**	**SRCGA**	**RCGA-PS**	**RCGA-PS-P**	**RCGA-P**
1	Aluffi-Pentini	2	-0.3523	-3.52E-01	-3.52E-01	-3.52E-01	-3.52E-01	-3.52E-01	-3.52E-01	-3.52E-01	-3.52E-01
2	Becker and Lago	2	0.0000	**2.00E-06**	**1.00E-06**	**1.00E-06**	**1.00E-06**	9.90E-05	9.90E-05	1.00E-04	9.90E-05
3	Bohachevsky 1	2	0.0000	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	9.90E-05	1.00E-04	9.80E-05	1.00E-04
4	Bohachevsky 2	2	0.0000	**0.00E+00**	**1.00E-06**	**0.00E+00**	**0.00E+00**	2.18E-01	9.90E-05	9.80E-05	1.00E-04
5	Branin	2	0.3979	**3.98E-01**	**3.98E-01**	**3.98E-01**	**3.98E-01**	3.98E-01	3.98E-01	3.98E-01	3.98E-01
6	Camel Back-3	2	0.0000	**1.00E-06**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.00E-04	9.60E-05	9.80E-05	1.00E-04
7	Camel Back-6	2	-1.0316	**-1.03E+00**	**-1.03E+00**	**-1.03E+00**	**-1.03E+00**	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
8	Cosine Mixture	2	-0.2000	**-2.00E-01**	**-2.00E-01**	**-2.00E-01**	**-2.00E-01**	-2.00E-01	-2.00E-01	-2.00E-01	-2.00E-01
9	Dekkers and Aarts	2	-24776.5183	**-2.48E+04**	**-2.48E+04**	**-2.48E+04**	**-2.48E+04**	-2.48E+04	-2.48E+04	-2.48E+04	-2.48E+04
10	Easom	2	-1.0000	**-1.00E+00**	**-1.00E+00**	**-1.00E+00**	**-1.00E+00**	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00
11	Goldstein and Price	2	3.0000	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	3.00E+01	3.00E+00	3.00E+00	3.00E+01
12	Hosaki	2	-2.3458	**-2.35E+00**	**-2.35E+00**	**-2.35E+00**	**-2.35E+00**	-2.35E+00	-2.35E+00	-2.35E+00	-2.35E+00
13	McCormick	2	-1.9133	-1.91E+00	-1.91E+00	-1.91E+00	-1.91E+00	-1.91E+00	-1.91E+00	-1.91E+00	-1.91E+00
14	Modified Rosenbrock	2	0.0000	**4.00E-06**	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.70E-02	8.27E-03	7.42E-03	7.42E-03
15	Multi-Gaussian	2	-1.2969	-1.30E+00	-1.30E+00	-1.30E+00	-1.30E+00	-1.21E+00	-1.22E+00	-1.30E+00	-1.30E+00
16	Periodic	2	0.9000	**9.00E-01**	**9.00E-01**	**9.00E-01**	**9.00E-01**	1.00E+00	1.00E+00	9.00E-01	9.00E-01
17	Schaffer 1	2	0.0000	2.00E-06	0.00E+00	**0.00E+00**	**0.00E+00**	1.14E-02	9.72E-03	9.72E-03	9.90E-05
18	Schaffer 2	2	0.0012	4.15E-04	4.14E-04	6.90E-05	0.00E+00	5.15E-01	2.41E-02	3.57E-03	1.29E-03
19	Shubert	2	-186.7309	**-1.87E+02**	**-1.87E+02**	**-1.87E+02**	**-1.87E+02**	-1.24E+02	-1.87E+02	-1.87E+02	-1.24E+02
20	Gulf Research	3	0.0000	**7.00E-06**	**1.70E-05**	**1.00E-06**	**5.00E-06**	6.39E-03	3.22E-04	3.65E-04	1.11E-02
21	Hartman 3	3	-3.8628	**-3.86E+00**	**-3.86E+00**	**-3.86E+00**	**-3.86E+00**	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00
22	Helical Valley	3	0.0000	**0.00E+00**	**2.00E-06**	**1.00E-06**	**0.00E+00**	9.90E-05	1.00E-04	1.00E-04	1.24E-04
23	Levy and Montalvo 1	3	0.0000	**0.00E+00**	**2.00E-06**	**1.00E-06**	**1.00E-06**	1.00E-04	1.00E-04	9.80E-05	1.00E-04
24	Meyer and Roth	3	0.0019	1.95E-03	**1.90E-03**	**1.91E-03**	**1.93E-03**	5.14E-03	2.00E-03	2.00E-03	4.68E-03
25	Cosine Mixture	4	-0.4000	**-4.00E-01**	**-4.00E-01**	**-4.00E-01**	**-4.00E-01**	-4.00E-01	-4.00E-01	-4.00E-01	-4.00E-01
26	Kowalik	4	0.0003	**3.21E-04**	**3.10E-04**	**3.10E-04**	**3.17E-04**	4.07E-04	4.07E-04	4.06E-04	4.07E-04
27	Miele and Cantrell	4	0.0000	**3.70E-05**	**0.00E+00**	**1.00E-06**	**2.60E-05**	1.00E-04	9.90E-05	1.00E-04	1.00E-04
28	Neumaier 2	4	0.0000	1.33E-04	6.50E-05	**2.40E-05**	5.30E-05	7.36E-02	7.27E-03	4.11E-03	7.79E-02
29	Powell's Quadratic	4	0.0000	**1.90E-05**	**4.00E-06**	**0.00E+00**	**0.00E+00**	1.00E-04	1.00E-04	9.70E-05	9.70E-05
31	Shekel 5	4	-10.1532	**-1.02E+01**	**-1.02E+01**	**-1.02E+01**	**-1.02E+01**	-2.63E+00	-5.06E+00	-1.02E+01	-1.02E+01
32	Shekel 7	4	-10.4029	**-1.04E+01**	**-1.04E+01**	**-1.04E+01**	**-1.04E+01**	-2.75E+00	-5.09E+00	-3.72E+00	-3.72E+00
30	Shekel 10	4	-10.5364	**-1.05E+01**	**-1.05E+01**	**-1.05E+01**	**-1.05E+01**	-2.87E+00	-1.05E+01	-3.84E+00	-2.42E+00
33	Wood	4	0.0000	3.95E-04	**2.10E-05**	**2.00E-06**	7.50E-05	7.87E+00	1.00E-04	1.00E-04	7.88E+00
34	Levy and Montalvo 2	5	0.0000	**1.60E-05**	**1.20E-05**	**1.20E-05**	**1.20E-05**	1.10E-02	1.00E-04	1.00E-04	2.09E-01
35	Salomon	5	0.0000	9.99E-02	7.80E-05	**0.00E+00**	**0.00E+00**	2.00E-01	9.99E-02	9.80E-05	9.90E-05
36	Shekel's Foxholes	5	-10.4056	**-1.04E+01**	**-1.04E+01**	-1.04E+01	-2.70E+00	-1.61E+00	-1.83E+00	-1.94E+00	-2.11E+00
37	Hartman 6	6	-3.3224	**-3.32E+00**	**-3.32E+00**	**-3.32E+00**	**-3.32E+00**	-3.20E+00	-3.20E+00	-3.32E+00	-3.32E+00
38	Storn's Tchebychev	9	0.0000	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	0.00E+00	0.00E+00	0.00E+00	0.00E+00
39	Ackley	10	0.0000	8.08E-04	5.50E-05	**0.00E+00**	**1.00E-06**	4.29E-03	1.00E-04	9.30E-05	1.00E-04
40	Exponential	10	-1.0000	**-1.00E+00**	**-1.00E+00**	**-1.00E+00**	**-1.00E+00**	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00
41	Griewank	10	0.0000	2.22E-03	**3.60E-05**	**0.00E+00**	**0.00E+00**	2.35E-01	3.94E-02	9.10E-05	9.90E-05
42	Levy and Montalvo 2	10	0.0000	**2.70E-05**	**2.70E-05**	**2.80E-05**	8.30E-05	1.10E-02	1.00E-04	1.10E-02	7.89E-01
43	Modified Langerman	10	-0.9650	-5.13E-01	**-9.65E-01**	-9.65E-01	-5.13E-01	-2.81E-02	-5.32E-02	-5.32E-02	-1.49E-02
44	Neumaier 3	10	-210.0000	-2.10E+02	**-2.10E+02**	**-2.10E+02**	-2.10E+02	-2.10E+02	-2.10E+02	-2.10E+02	-2.10E+02
45	Paviani	10	-45.7780	-4.58E+01	-4.58E+01	-4.58E+01	-4.58E+01	-4.58E+01	-4.58E+01	-4.58E+01	-4.58E+01
46	Rastrigin	10	0.0000	9.95E-01	2.50E-05	0.00E+00	0.00E+00	1.79E+01	6.96E+00	9.80E-05	9.90E-05
47	Rosenbrock	10	0.0000	7.75E-03	9.20E-05	1.12E-03	1.64E+00	2.53E+02	4.73E+00	1.66E-02	2.52E+00
48	Salomon	10	0.0000	9.99E-02	9.99E-02	**2.00E-06**	**0.00E+00**	2.00E-01	9.99E-02	1.00E-04	9.90E-05
49	Schwefel	10	-4189.8289	-3.62E+03	**-4.19E+03**	**-4.19E+03**	-3.42E+03	-1.96E+03	-3.72E+03	-3.60E+03	-1.62E+03
50	Shekel's Foxholes	10	-10.2088	-1.48E+00	**-1.02E+01**	-1.60E+00	-1.48E+00	-1.26E+00	-1.35E+00	-1.26E+00	-1.48E+00
51	Sinusoidal	10	-3.5000	**-3.50E+00**	**-3.50E+00**	**-3.50E+00**	**-3.50E+00**	-3.50E+00	-3.50E+00	-3.50E+00	-1.00E+00
52	Spherical	10	0.0000	**4.20E-05**	**1.50E-05**	**0.00E+00**	**0.00E+00**	1.00E-04	1.00E-04	9.30E-05	1.00E-04
53	Storn's Tchebychev	17	0.0000	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	0.00E+00	0.00E+00	0.00E+00	0.00E+00
54	Sinusoidal	20	-3.5000	**-3.50E+00**	**-3.50E+00**	**-3.50E+00**	**-3.50E+00**	-3.50E+00	-3.50E+00	-3.50E+00	-1.00E+00

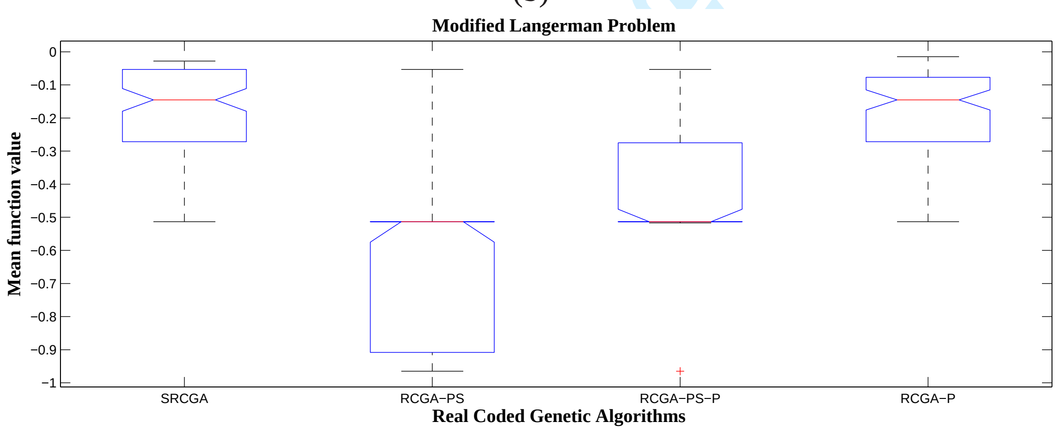
Appendix IV



(a)



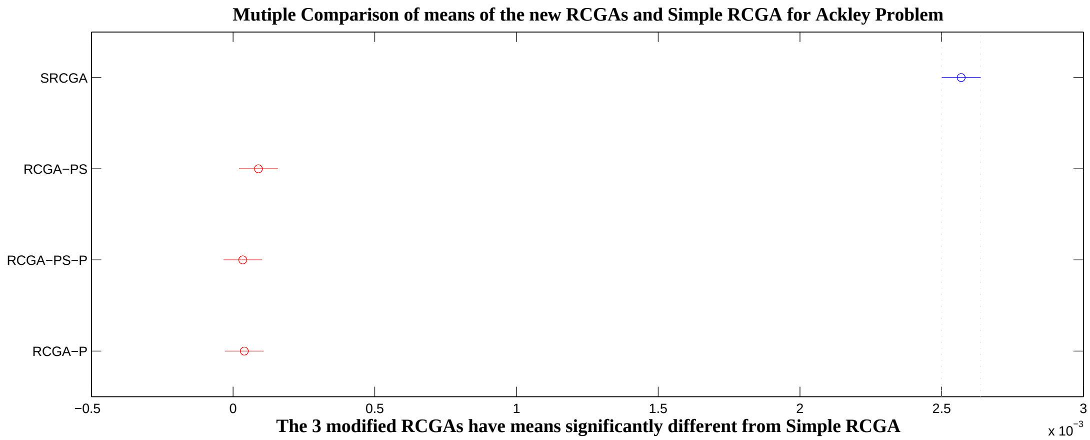
(b)



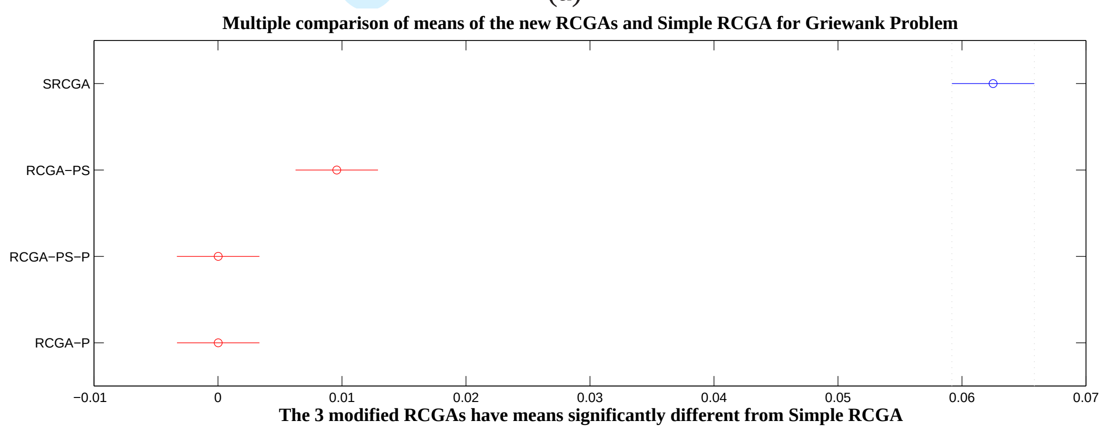
(c)

Figure 2: A set of box plots (a) to (c) showing the differences between the mean function values of SRCGA and the new RCGAs on Ackley, Griewank and Modified Langerman, respectively.

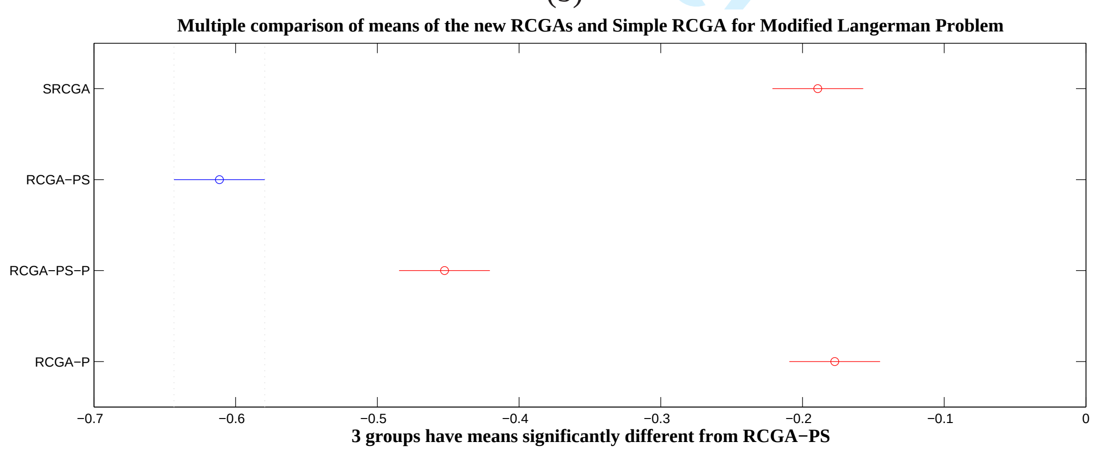
Appendix V



(a)



(b)



(c)

Figure 3: Multiple comparison graphs (a) to (c) showing the difference between the means of SRCGA and the new RCGAs on Ackley, Griewank and Modified Langerman, respectively.