International Journal of Innovative Computing, Information and Control
Volume **8**, Number **1(A)**, January **2012**

ICIC International ©2012 ISSN 1349-4198
pp. **269–284**

MULTI-OBJECTIVE OPTIMIZATION FOR PID CONTROLLER TUNING USING THE GLOBAL RANKING GENETIC ALGORITHM

Mohd Rahairi Rani, Hazlina Selamat, Hairi Zamzuri and Zuwairie Ibrahim

Centre for Artificial Intelligence and Robotics
Faculty of Electrical Engineering
Universiti Teknologi Malaysia
81310 UTM Skudai, Johor, Malaysia
rahairi@yahoo.com; hazlina@fke.utm.my

Received August 2010; revised December 2010

Abstract. *Tuning of PID controller parameters for an optimized control performance is a multi-objective optimization problem. The problem becomes particularly difficult if the plant to be controlled is an unstable, nonlinear and under actuated plant. This paper proposes a modified genetic algorithm for the multi-objective optimization of PID controller parameters, called the Global Ranking Genetic Algorithm (GRGA). It combines two types of fitness assignment methods in the algorithm – the 'global ranking fitness assignment' method proposed in this paper and the dominance rank from the classical pareto fitness assignment method. The former is employed in the selection of parents and the latter is used in the elitism mechanism. In order to investigate the performance of the proposed algorithm, it is compared with the state of the art, Non-dominated Sorting Genetic Algorithm 2 (NSGA-II) using five ZDT series test functions. From the test problems analysis, the GRGA is observed to have better convergence property than the NSGA-II although it tends to lose its diversity of solutions in the earlier part of generation before recovering back when approaching the true pareto front. Then, the GRGA is applied to a highly difficult PID controller tuning problem, balancing a rotary inverted pendulum system. Results show that the GRGA has the capability to optimally tune the PID controllers based on the nonlinear model of the pendulum.*

Keywords: PID controllers, Controller auto-tuning, Genetic algorithm, Evolutionary algorithm, Multi-objective optimization, Inverted pendulum

1. Introduction.

Proportional-Integral-Derivative (PID) controller is one of the most popular controllers applied in industries. Simple in structure, reliable in operation and robust in performance, result in its popularity among control engineers [1]. However, a research reported that 80% of the PID-type controllers in industries are poorer or less optimally tuned, that 30% of the PID loops operate in manual mode and that 25% of the PID loops actually under default factory setting [2]. Over the years, many methods have been proposed for the tuning of the PID controllers, both in the deterministic or stochastic frameworks [2,3]. Tuning of the PID controllers is not a straightforward problem especially when the plants to be controlled are nonlinear and unstable. A rotary inverted pendulum, for example, has an unstable, nonlinear and chaotic motion of its arm and pendulum, and a PID controller can be very difficult to tune using the conventional deterministic PID tuning methods.

Tuning the PID controller coefficients can be considered as a parameter optimization process to achieve a good system response. In time domain specification, a good system response is a response that has a minimum rise time, settling time, overshoot and steady state error. Therefore, the tuning process of the controller has multiple objectives to be achieved, which, in most cases, are conflicting with each other. The tradeoff between several objectives makes multi-objective optimization have not only a single optimal solution, but rather a set of optimal solutions known as the 'pareto optimal front'.

Generally, in the manual tuning of a PID controller, the optimization process is not performed simultaneously with the tuning process. In order to get the optimally tuned parameters, the parameters are slightly changed based on designer's intuition or trial and error method after the main tuning process is done [2]. One of the most popular methods to tune the PID controllers is the Ziegler and Nichols method [2]. This is an empirical method that tries to find the ultimate gain and period in order to obtain the PID gain values. Ziegler and Nichols approach is a practical method for a single output and stable plants, but the tuning process will be extremely complicated when dealing with multiple output and unstable plants. More systematic ways to optimize the PID control parameters have been proposed in, for instance, [4,5] using the deterministic optimization methods based on the integrated absolute error (IAE) criterion. The main drawback of these methods is that the tuned parameters are only optimum in certain operational zones and have an unsatisfactory design robustness property.

Moreover, the above tuning methods are limited to linear or linearized plant models and single-input single-output (SISO) systems as well as not directly incorporated with time specifications in their tuning optimization. Thus, stochastic optimization approaches may be needed in dealing with the more complicated multiple-input multiple output (MIMO) and open loop unstable plants. Genetic Algorithm (GA), for instance, is a powerful search algorithm used by researchers to optimize PID controllers but most available methods focus on a single-objective optimization while the PID tuning problem is clearly a multi-objective optimization problem. In the single-objective tuning optimization, IAE is usually used as an objective function with the hope of getting a good time specifications in system responses. On the other hand, in the multi-objective optimization, the time specifications can be directly formulated as objective functions in the optimization process. Hence, the tuning objectives and parameter variations can be directly incorporated to find the optimum PID controller parameter values.

Recently, the number of multi-objective evolutionary algorithms (MOEA) increases drastically due to their popularity and capability of successfully solving multi-objective optimization problems. During the World Congress of Computational Intelligence (WCCI) in Vancouver 2006, MOEA has been evaluated as one of the three fastest growing fields of research and application among all computational intelligence topics [6].

This paper proposes a new and systematic PID tuning algorithm based on the MOEA approach. A rotary inverted pendulum will be used as the plant to be controlled using the PID controller to test the capability of the proposed approach in dealing with highly complex plants. The nonlinear dynamical model of the rotary inverted pendulum is derived from its equations of motion and implemented in Simulink and Matlab.

In the remainder of the paper, we briefly mention the basic of MOEA and state a number of famous MOEAs in Section 2. Thereafter, in Section 3, we describe the proposed GRGA algorithm in detail. Section 4 presents the comparison of simulation results between GRGA and NSGA-II in ZDT test problems. In Section 5, we present the application of GRGA in PID controller optimization. Finally, we present the conclusions of this paper.

2. Multi-objective Evolutionary Algorithms (MOEAs).

Almost all MOEAs are constructed based on a single objective evolutionary algorithm (EA) before they are extended to handle multi-objective problems. Similarly, the approach presented in this paper begins with a single objective genetic algorithm (GA) as the basis of the algorithm.

GA consists of five main stages: initialization, fitness evaluation, selection, genetic operation such as mutation and generation of new population. The algorithm is initialized with a randomly generated population of individuals, which are then directed towards convergence at the global optimum using the selection and genetic operators. In each generation, the fitness of every individual in the population is evaluated. Then, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Normally, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. The search power of GA lies on its population of solutions, which enhance the search capability in many points simultaneously. In this paper, the term 'solution' and 'individual' are interchangingly used and represent one or more decision variable values that cascaded to form a vector.

Generally, the critical part in MOEAs that make one algorithm different from the other lies on the selection mechanism on how the individuals that are non-dominated are preferred to be selected over those that are dominated. There are three main classifications among MOEA techniques in order to achieve such selection mechanism: aggregating function, population based function and pareto based function [7].

The aggregation function is the simplest method for handling multi-objective optimization problem. It aggregates all the objective functions into one single objective function [8]. Each objective has its assigned weight and the total of all the weights must equal to one. The aggregating function is defined as follows:

$$\sum_{k=1}^{M} w_k f_k(x) \tag{1}$$

and

$$\sum_{k=1}^{M} w_k = 1 \tag{2}$$

where M is the number of objectives, w_k is the weight for kth objective and f_k is the fitness value for kth objective.

These aggregating approaches have been largely underestimated by MOEA researcher due to its high possibility of converging at the local optimum and the knowledge for every objective is required in order to assign the appropriate weight for each objective. However, due to its simplicity, this weighted-sum aggregation approach appears to still be in widespread use.

In contrary, the population-based function aims to diversify the search algorithm based on the fitness information in each objective of the individuals. The most popular algorithm in this class of MOEAs is the Vector Evaluated Genetic Algorithm (VEGA), proposed in [9]. VEGA is actually a simple genetic algorithm (GA) with a modified selection mechanism. VEGA uses a switching objectives mechanism to generate a population of offspring by having N/M of offspring for each objective, where N is the population size and M is the number of objectives. The weakness of this approach is that the concept of pareto dominance is not incorporated directly in the selection process.

According to [7], the pareto-based approach is perhaps the most successful approach to guide the search space to the true pareto front. Goldberg in [10] suggested that every MOEA algorithm should have a tendency to select the individuals that are non-dominated with respect to the current population. Goldberg also suggested 'fitness sharing', the crowding information of an individual and 'niche parameter', the distance of an individual to its neighbor to preserve the diversity in solutions. Fosenca and Fleming [11] proposed a fitness assignment strategy in Multi-objective Genetic Algorithm (MOGA), slightly different from Goldberg's suggestion that takes into account how many other individuals dominate the respective individual as fitness values. Thus, all the non-dominated individuals will have the same fitness values and have the same probability to be selected.

Srinivas and Deb realized Goldberg's suggestion most directly in the Non-dominated Sorting Genetic Algorithm (NSGA) [12] where every non-dominated front in the current population is evaluated and the individuals in the same fronts will have the same fitness value. To maintain the diversity in the pareto solutions, NSGA introduced 'niche count', a measure of individual's density respect to other individuals in the objective space. The offshoot of this approach, known as NSGA-II [13] uses the same fitness assignment strategy but has elitism mechanism and crowded comparison operator, a normalized distance between the two neighbors to preserve the diversity. NSGA-II has always been considered as the state-of-the-art algorithm by MOEA researchers [7]. However, the non-dominated sorting in NSGA is computationally expensive although Deb et al. [13] suggested the fast version of non-dominated sorting in NSGA-II. This paper investigates the performance of a new fitness assignment that has a lower computational complexity compared with NSGA-II. The proposed algorithm is known as the global ranking GA (GRGA).

3. Global Ranking Genetic Algorithm (GRGA).

In this section, the main components in GRGA, which are the proposed fitness assignment, the dominance rank [11] and the crowding distance [13] procedures, are described before the complete algorithm is presented.

3.1. Fitness assignment.

The proposed fitness assignment method, called the 'global fitness assignment' in this paper, uses the rank of individuals in each objective to promote the 'global' rank values. The rank of a solution X_i is given by the vector $R(X_i) = \left[\begin{array}{cccc} r_1(X_i) & r_2(X_i) & \cdots & r_M(X_i) \end{array}\right]^T$, where $r_m(X_i)$ is the sub-rank of X_i for the mth objective. Once the vector $R(X_i)$ has been calculated for each solution X_i, its global rank, G_i, is found using:

$$G_i = \sum_{m=1}^{M} r_m(X_i) \tag{3}$$

where M is the total number of objectives and G_i is the global rank for X_i.

Table 1 shows an example of the global ranking fitness assignment of 5 individuals, $(X_1 - X_5)$ with the objective values for three objectives $(f_1 - f_3)$ respectively, where the minimization of all objectives is assumed. In the example, say for the first objective, f_1, X_2 have the smallest objective value, followed, in ascending order, by X_3, X_4, X_1 and X_5. Therefore, X_2 is sub-ranked '1' in X_2, giving $r_1(X_2) = 1$, followed by $r_1(X_3) = 2$, $r_1(X_4) = 3$, $r_1(X_1) = 4$ and $r_1(X_5) = 5$. Similarly, the sub-ranks are assigned in the same manner to all the solutions, for r_2 and r_3. Finally, the global rank value, G, for each solution is calculated by summing all its sub-rank values.

Table 1. An example of the proposed fitness assignment

Solution/ Individual	**Objective**			**Sub-rank**			**Global rank, G**
	f_1	f_2	f_3	r_1	r_2	r_3	
X_1	1.1566	0.56	0.20	4	5	1	10
X_2	0.9656	0.50	0.56	1	3	5	9
X_3	0.9823	0.48	0.86	2	1	3	6
X_4	1.1456	0.51	0.75	3	4	2	9
X_5	1.2566	0.49	0.89	5	2	4	11

3.2. **Dominance rank.** In this algorithm, the dominance rank fitness assignment as introduce in [11] is used where the fitness value of an individual is the number of other individuals that dominate it. The fitness for the individual X_i, $f(X_i)_i$ is given by Equation (4).

$$f(X_i)_i = 1 + \text{ no. of superior individuals} \tag{4}$$

Figure 1 shows an example of dominance rank fitness assignment where 10 individuals are visualized in their 2-objective space, f_1 and f_2. With the assumption of maximization of both functions, the vertical and horizontal lines from X_i to the x- and y-axis respectively form the boundaries of the dominance area of an individual. The value in the bracket placed near each individual is the fitness value obtained from dominance rank method.

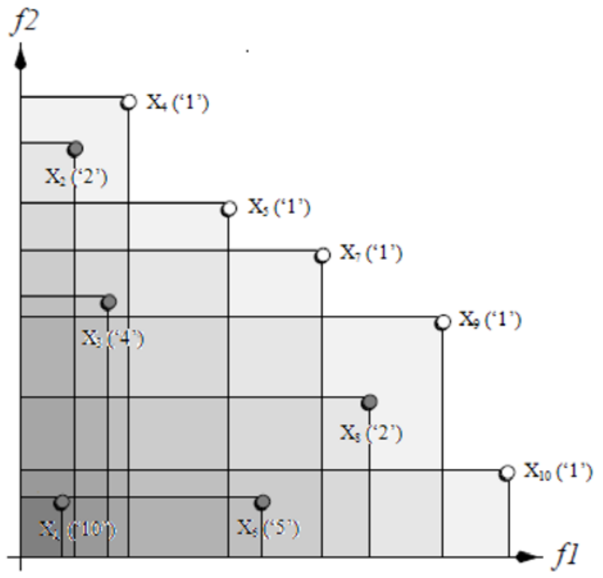


Figure 1. Dominance rank for 10 examples of solutions

Taking two individuals, X_4 and X_5, in Figure 1 as examples, it can be observed that X_4 and X_5 have the fitness values of 1 and 5 respectively. For X_4, since it is not dominated by any other individual, the fitness value is equal to '1'. On the other hand, X_5 lies in the domination region of 4 other individuals (superiors), hence its fitness value is $1 + 4 = 5$.

The dominance rank fitness assignment is employed in elitism mechanism, where the current population and offspring produced from the genetic operation of the parents are combined to form a combined population, CP, with size of $2N$, where N is the population size. In order to select N individuals from CP to form a new generation, all the individuals in CP will be ranked based on the dominance rank method and the best N individuals are chosen. However, at some points, the number of individuals with the fitness of '1' (non-dominated individuals) may exceed N. Thus, a second sorting procedure is required and this time the sorting procedure will be based on the crowdedness among the non-dominated individuals. This is also known as the diversity preservation technique.

3.3. **Crowding distance.** The crowding distance operator has been introduced in [13], which is a simple yet effective method to estimate the crowdedness density. Crowding distance calculation requires the sorting of the population according to ascending order of each objective. The smallest and largest values (boundaries) will be assigned as an infinite distance value. For other intermediate individuals, the distance of each objective,

d_i, is calculated based on Equation (5).

$$d_i = \sum_{m=1}^{M} \frac{f_{(i+1)_m} - f_{(i-1)_m}}{f_{\max_m} - f_{\min_m}} \tag{5}$$

where M is the number of objective, $f_{\max}$ and $f_{\min}$ are the values of maximum and minimum objective values respectively. The larger the value of the crowding distance, the smaller (better) its crowdedness property.

3.4. **The main loop of GRGA.** Firstly, a random population of N size is created. After evaluating the objective values for all the individuals in the population, the fitness values are calculated based on the proposed global fitness assignment. By using the binary tournament of selection, the parents are selected to fill the mating pool based on the global rank values where, in our case, the size of the mating pool is chosen to be a half of the population size.

The parents in the mating pool will be going through recombination (crossover) and mutation [15] operation to generate another N-size population. This new population, along with the previous population, will be combined and sorted according to the dominance rank and the crowding distance in elitism. Note that the crowding distance sorting procedure will only be triggered if the number of non-dominated solutions, whose fitness is '1' in the dominance rank, exceeds N. Using these two sorting procedure, a population of size N is extracted as a new generation. With this new population replacing the initial population, the above procedures will be repeated until the stopping criterion is met. Figure 2 shows the complete flow chart for the GRGA.

The uniqueness of the GRGA lies in the combination of two different types of fitness assignment, which are the proposed global ranking (population-based function) and the dominance rank (pareto-based function). Typically, most MOEAs only employ the same fitness assignment in selection and elitism mechanism. Here, we aim to show that it is not necessary for MOEA to use the pareto-based fitness assignment in the selection mechanism in order to produce optimum solutions in the final results.

4. **Test Problems and Performance Measures.** In this section, GRGA is compared with one of the famous MOEAs, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) using the ZDT series test functions [14]. The ZDT test functions are described in Table 2. This test series consists of various properties of problems, such as convex, non-convex, disconnected and non-uniformly spaced problems. All the objective functions in the test problems in Table 2 are subjected to be minimization.

In order to analytically evaluate the performances of the GRGA and NSGA-II, two performance metrics are employed, namely, the convergence metric, γ [13] and the diversity metric, Δ [16]. The first metric, γ, is used to measure the convergence of the final obtained solutions with respect to the true pareto front. Firstly, 500 of uniformly spaced solutions are placed in the pareto optimal front in the objective space as the reference. Then, for each final obtained solutions, the nearest Euclidian distance from one of the reference solutions is measured. The average of these distances for all the final solutions is calculated as γ value. The smaller the metric, the better convergence of the solutions towards the true pareto front.

Although the convergence metric itself can provide information on the diversity in solutions, this paper uses a different metric to measure the diversity in the solutions. The diversity metric, Δ measures the diversity among the obtained non-dominated solutions (P*) with respect to a reference set (RS). For the test problems in this paper, the values of f_1 will be the reference set, which discretized in a number of grids with the range of

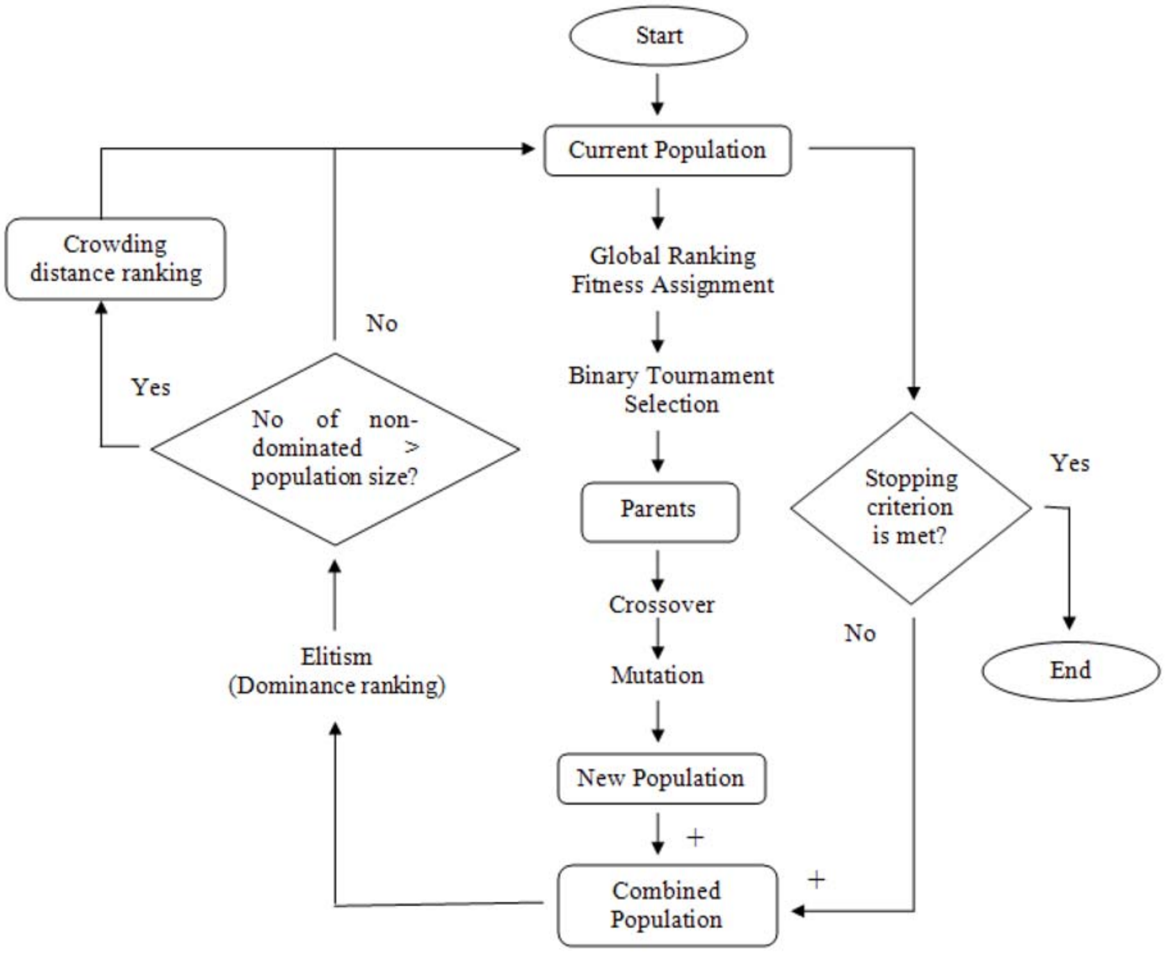

Start
Current Population
Global Ranking Fitness Assignment
Binary Tournament Selection
Parents
Crossover
Mutation
New Population
+
Combined Population
+
Elitism (Dominance ranking)
No of non-dominated > population size?
Yes
No
Crowding distance ranking
Stopping criterion is met?
Yes
No
End

FIGURE 2. The complete flow chart of GRGA

TABLE 2. Test problems used to evaluate the performance of GRGA and NSGA-II

Test Problem	n	Variable bounds	Objective function	Optimal solutions	Comments
ZDT1	30	$[0, 1]$	$f_1 = x_1$ $f_2 = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}}\right]$ $g(x) = 1 + 9\frac{(\sum_{i=2}^{n} x_i)}{n-1}$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \cdots, n$	Convex
ZDT3	30	$[0, 1]$	$f_1(x) = x_1$ $f_2 = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)}\sin(10\pi x_1)\right]$ $g(x) = 1 + 9\frac{(\sum_{i=2}^{n} x_i)}{n-1}$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \cdots, n$	Convex, disconnected
ZDT2	30	$[0, 1]$	$f_1(x) = x_1$ $f_2 = g(x)\left[1 - \left(\frac{x_1}{g(x)}\right)^2\right]$ $g(x) = 1 + 9\frac{(\sum_{i=2}^{n} x_i)}{n-1}$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \cdots, n$	Non-convex
ZDT4	10	$x_1 \in [0, 1]$ $x_i \in [-5, 5]$ $i = 2, \cdots, n$	$f_1 = x_1$ $f_2 = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}}\right]$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}\left[x_i^2 - 10\cos(4\pi x_i)\right]$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \cdots, n$	Non-convex
ZDT6	10	$[0, 1]$	$f_1 = 1 - \exp(-4x_1)\sin^6(6\pi x_i)$ $f_2 = g(x)\left[1 - \left(\frac{x_1}{g(x)}\right)^2\right]$ $g(x) = 1 + 9\left[\frac{(\sum_{i=2}^{n} x_i)}{n-1}\right]^{0.25}$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \cdots, n$	Non-convex, Non-uniformly, spaced

$^*n$ = number of decision variables

0.01. The more grids that contain a point of RS and also contain a point of P*, the higher the metric value. This metric defines two arrays, $H(i)$ and $h(i)$ which are computed as in Equations (6) and (7).

$$H(i) = \begin{cases} 1, & \text{the grid has a representative point in RS} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$$h(i) = \begin{cases} 1, & \text{if } H(i) = 1 \text{ and the grid has a representative point in P}^* \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Then, the array, $m(h(i))$ is calculated based on the neighboring scheme as in Table 3.

Table 3. Neighboring scheme for $m(h(i))$

$h(i-1)$	$h(i)$	$h(i+1)$	$m(h(i))$
0	0	0	0
0	0	1	0.5
1	0	0	0.5
0	1	1	0.67
1	1	0	0.67
1	0	1	0.75
0	1	0	0.75
1	1	1	1

Finally, the diversity metric, Δ is computed using

$$\Delta = \frac{\sum_{i|h(i)\neq 0} m(h(i))}{\sum_{i|H(i)\neq 0} m(H(i))} \tag{8}$$

For diversity metric, with the assumption that the final non-dominated solutions obtained approach the global pareto front, the higher the metric becomes, the better the diversity properties in solutions.

In order to ensure consistency in the results, each algorithm will be run 10 times for each test problem. The parameters used in both the GRGA and NSGA-II in all the test functions are summarized in Table 4.

Table 4. GRGA and NSGA-II parameters used in the tests

Parameters setting	Values
Number of generation	250
Population Size	100
Probability of Crossover	0.8
Probability of Mutation	$1/V$
Distribution index in SBX	20
Distribution index in polynomial mutation	20

*V = number of decision variables

Table 5 shows the mean and the variance of the convergence and the diversity metrics for the GRGA and NSGA-II in every ZDT test problems.

It can be seen from Table 5 that GRGA has better convergence performance for all the test problems compared with NSGA-II. The critical ranking procedure in global ranking fitness assignment has improved the convergence properties in approximating the true pareto front. For the diversity metric, NSGA-II has better diversity in ZDT1, ZDT3 and ZDT6 while GRGA has better diversity in ZDT2 and ZDT4. This is because, the same

Table 5. Performance metric

Test Problem	Algorithm	Convergence		Diversity	
		Mean	Variance	Mean	Variance
ZDT1	GRGA	**0.001101**	1.5638×10^{-8}	0.764401	6.9623×10^{-4}
	NSGA-II	0.001119	1.2551×10^{-8}	**0.766419**	4.0807×10^{-4}
ZDT2	GRGA	$\mathbf{7.7960 \times 10^{-4}}$	1.8840×10^{-9}	**0.786110**	8.0106×10^{-4}
	NSGA-II	7.8993×10^{-4}	1.5270×10^{-9}	0.769790	$7.6968x \times 10^{-4}$
ZDT3	GRGA	**0.004002**	4.9444×10^{-8}	0.564398	7.6650×10^{-4}
	NSGA-II	0.004201	1.3778×10^{-7}	**0.576890**	6.0154×10^{-4}
ZDT4	GRGA	**0.008501**	1.1825×10^{-5}	**0.722523**	0.003501
	NSGA-II	0.010118	1.2867×10^{-5}	0.693457	0.005209
ZDT6	GRGA	**0.003001**	3.3778×10^{-8}	0.684756	3.1095×10^{-4}
	NSGA-II	0.003012	2.6222×10^{-8}	**0.686165**	1.9182×10^{-4}

diversity preservation method has been used in both the GRGA and NSGA-II and so the difference between the diversity metric of the two algorithms are not very significant.

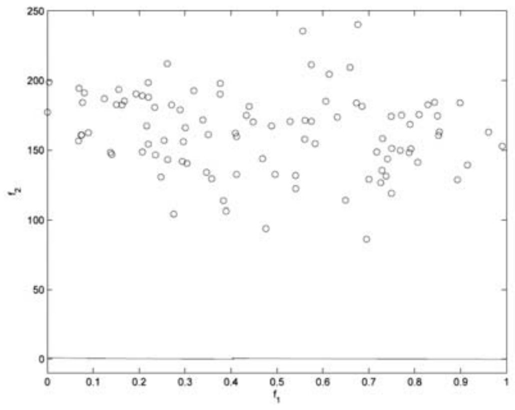
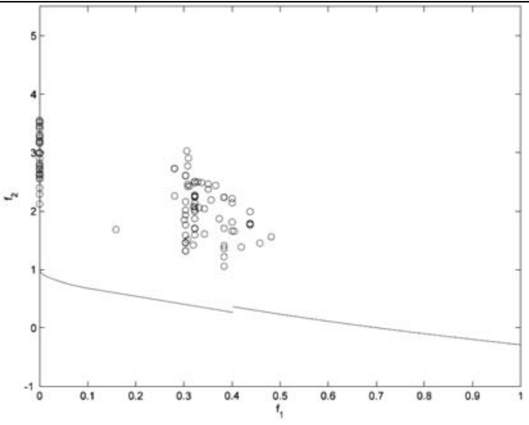
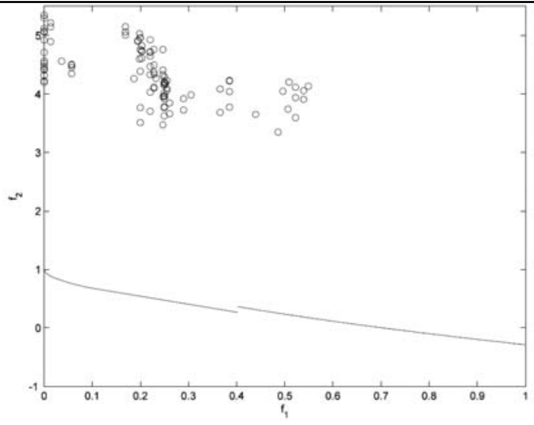
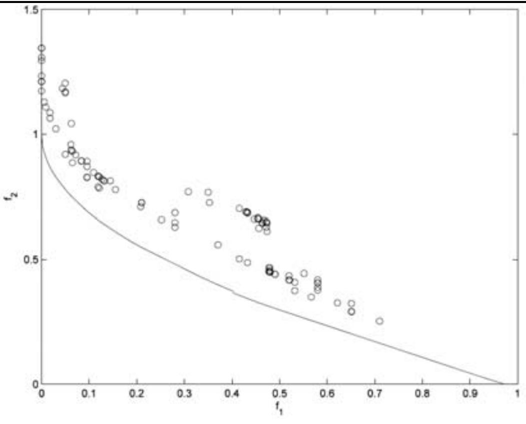
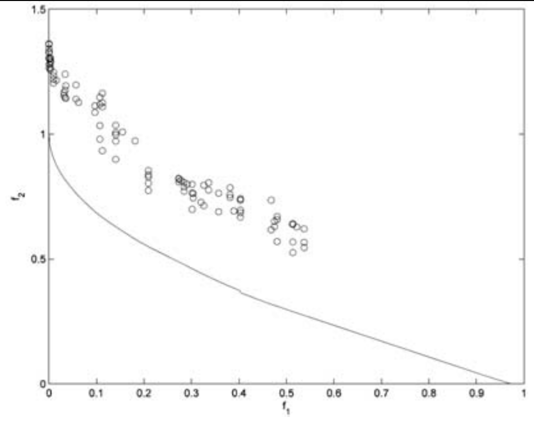
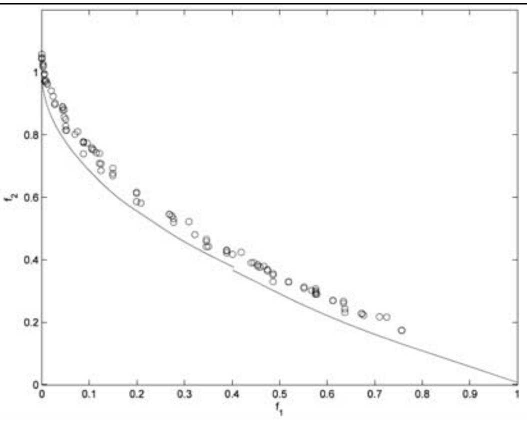
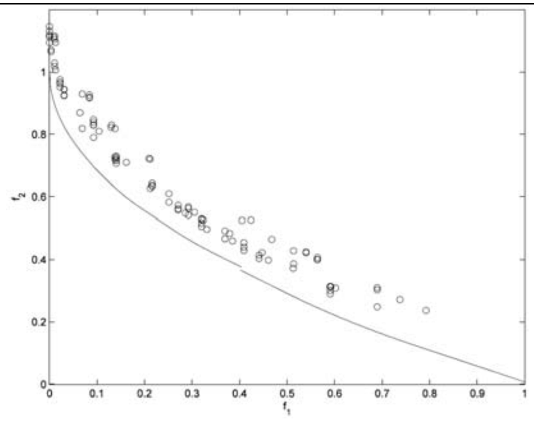
In order to further investigate the convergence and diversity properties, we took a closer look at the non-convex and challenging ZDT4 test function. ZDT4 problem has 21^9 local pareto fronts and this property causes difficulties for both algorithms to converge globally. Here, the performances of GRGA and NSGA-II from the initial population until the 250th generation were observed. The investigation started with the same initial population, and used the same parameters as shown in Table 4. Figures in Table 6 show the transitions of the solutions at 1, 50, 100, 150, 200 and 250 of generation for GRGA and NSGA-II. The figures visualize the distribution of the solution in two objective space, f_1 and f_2.

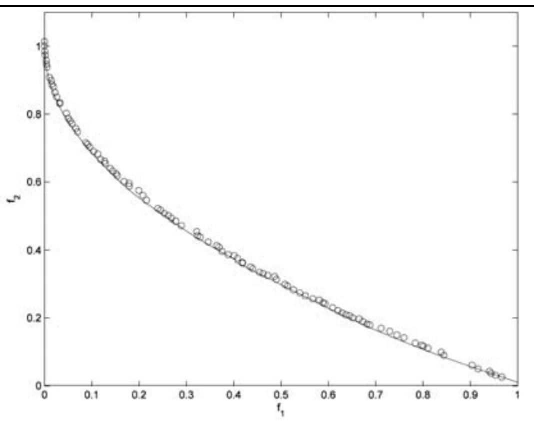
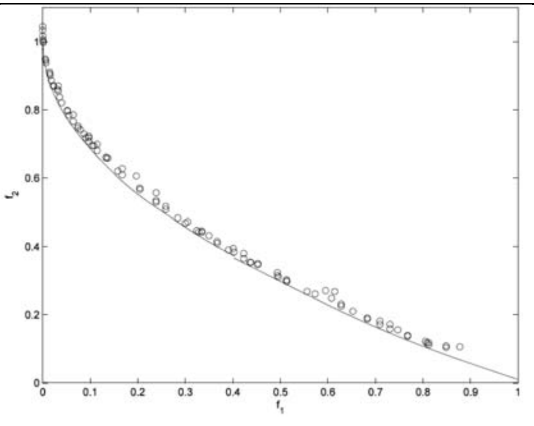
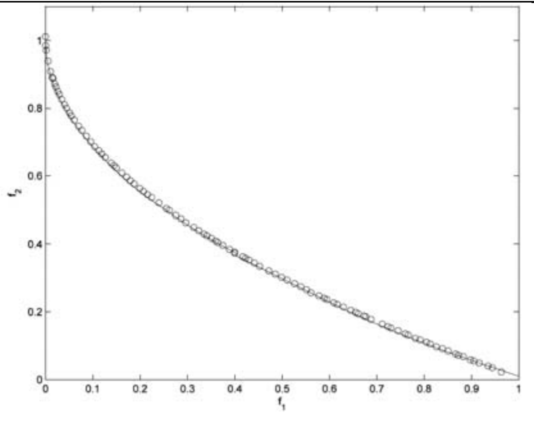
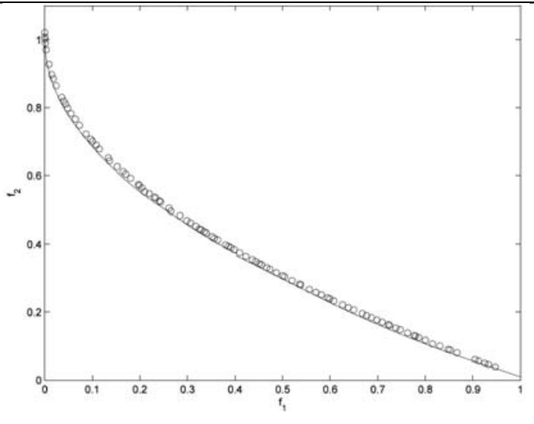
From Table 6, the continuous lines in all the plots represent the true pareto front in ZDT4. At the 50th generation, it can be observed that GRGA has shown a better convergence property with the solutions scattered in range of $f_1 = [1 - 3.5]$ compared with NSGA-II, whose solutions in the region given by $f_1 = [3 - 5.5]$. However, the solutions of NSGA-II have better distribution whereas the solutions in GRGA are slightly crowded in a few places. At the 100th generation, the solutions in GRGA started to form the pareto front, but the solutions in NSGA-II were still struggling to form the pareto front as well as maintain their diversity. At the 150th generation, the solutions from GRGA almost covered all the points in the pareto front, but for NSGA-II, there are still some dominated solutions which are quite far from the pareto front. Then, at the 200th generation, almost all the solutions in GRGA become non-dominated solutions as they cover almost all the pareto front compared with those of NSGA-II. Finally, at the 250th generation, both algorithms converge and cover approximately all the points in the pareto front. It is interesting to note that the convergence of GRGA not only surpasses NSGA-II when approaching the true pareto as in the results of convergence metrics in Table 5, but this property actually has started at the earlier generations. However, at some point, GRGA tends to lose its diversity in the earlier stage of the generation (as shown by $K = 50$ in Table 6) before recovering back when approaching the pareto front. This inconsistency of diversity may be improved by implementing a more complex diversity preservation technique as the second ranking procedure in elitism mechanism such as clustering technique [17] or hyper-volume measure [18] to replace the crowding distance.

5. PID Controller Tuning with GRGA.

In this section, the GRGA has been applied to the PID controller tuning problem of balancing a rotary inverted pendulum (RIP). A rotary inverted pendulum shown in Figure 3 was used as the plant of controlled by the

Table 6. The transition of solution at Kth generation for GRGA and NSGA-II in ZDT4 problem

K	**GRGA**	**NSGA-II**
1		
50		
100		
150		

200		
250		

PID controller whose parameters are optimized using GRGA. RIP consists of a driven arm which rotates in the horizontal plane and a pendulum that is attached to the arm that are free to rotate in vertical plane. The plant also consists of a DC servo motor that acts as an actuator to provide a torque to the pendulum arm. The interesting aspect about the RIP is its under-actuated property, where the number of degree of freedom is more than the number of actuators. This property causes the PID tuning mechanism to be rather challenging.

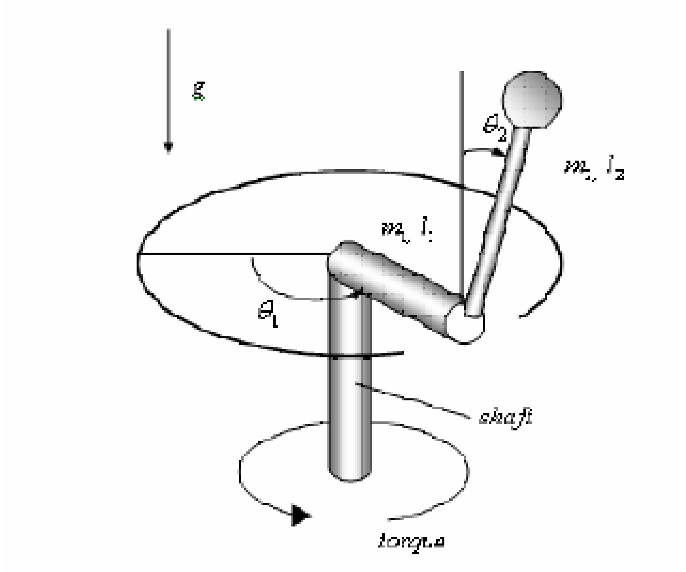


FIGURE 3. Rotary inverted pendulum

With the under-actuated system, two PID controllers are used: one for controlling θ_1 (PID 1) and the other one for controlling θ_2 (PID 2). Figure 4 shows the block diagram of the rotary inverted pendulum control system.

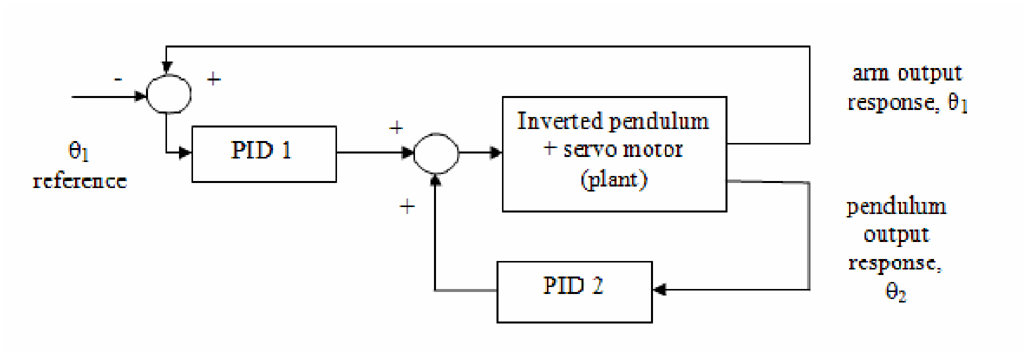

-
+
θ1 reference
PID 1
+
+
Inverted pendulum + servo motor (plant)
PID 2
arm output response, θ1
pendulum output response, θ2

FIGURE 4. PID control strategy and plant

The optimization process was carried out to determine the optimal controller gains, K_p, K_i and K_d for both PID controllers. Assuming that the initial angular position for the arm, θ_1, is zero and the angular position between the pendulum and a vertical line, θ_2, is 0.087 radian, the control objectives were to turn the arm, θ_1 and the pendulum, θ_2 to zero radian with the gains provided by GRGA. From the simulation response, the three objectives were evaluated – the settling time, overshoot percentage and mean square steady-state error (MSE). Both the nonlinear and linear models were used in the optimization process. The objective values were evaluated from the nonlinear model while the closed-loop stability is derived from the linearized model. From the equations of motion, the nonlinear models for the rotary inverted pendulum are given by Equations (9) and (10).

$$\begin{aligned}\ddot{\theta}_1 = {}& \frac{P_4}{\Delta}\left(P_8 e - P_7\dot{\theta}_1 - \frac{1}{2}P_5\dot{\theta}_2\dot{\theta}_1\sin^2(2\theta_2) + P_3\dot{\theta}_2{}^2\sin\theta_2 - \frac{1}{2}P_2\dot{\theta}_1\dot{\theta}_2\sin(2\theta_2)\right)\\ & - \frac{P_8\cos\theta_2}{\Delta}\left(P_6\sin\theta_2 - \frac{1}{2}m_2c_2\dot{\theta}_1{}^2\cos(2\theta_2)\right)\end{aligned} \tag{9}$$

$$\begin{aligned}\ddot{\theta}_2 = {}& \frac{-P_8\cos\theta_2}{\Delta}\left(P_8 e - P_7\dot{\theta}_1 - \frac{1}{2}P_5\dot{\theta}_2\dot{\theta}_1\sin^2(2\theta_2) + P_3\dot{\theta}_2{}^2\sin\theta_2\right.\\ & \left. - \frac{1}{2}P_2\dot{\theta}_1\dot{\theta}_2\sin(2\theta_2)\right) + \frac{P_1 + P_2\sin^2\theta_2}{\Delta}\left(P_6\sin\theta_2 - \frac{1}{2}m_2c_2\dot{\theta}_1{}^2\cos(2\theta_2)\right)\end{aligned} \tag{10}$$

where $\Delta = P_4P_1 + P_4P_2\sin^2\theta_2 - P_3^2\cos^2\theta_2$, $P_1 = J_1 + m_2l_1^2$, $P_2 = m_2c_2^2$, $P_3 = m_2l_1c_2$, $P_4 = J_2 + m_2c_2^2$, $P_5 = m_2l_1^2$, $P_7 = \dfrac{k_b^2}{R_m}$, $P_6 = m_2c_2g$, $P_8 = \dfrac{k_b}{R_m}$.

The meanings and values of other symbols used in Equations (9) and (10) are given in the Appendix.

Linearizing the pendulum model about $\theta_2 = 0$, the linearized state space equation is given by Equation (11).

$$\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix} = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & -4.4256 & -0.0390 & 0\\ 0 & 42.6498 & 0.0334 & 0\end{bmatrix}\begin{bmatrix}\theta_1\\ \theta_2\\ \dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 2.1334\\ -1.8286\end{bmatrix}e \tag{11}$$

The optimization goal was to minimize all the control objectives subject to the stability constraints. Every combination of gains in the PID controllers should have all the closed-loop poles on the left-hand side of the s-plane. Stability violations were calculated when the solutions were not closed-loop stable [19,20]. The violation value is the summation of the total closed-loop poles on the right-hand side of s-plane. These violation values will be used in the binary tournament of selection [13]. A solution a is said to constraint-dominate a solution b, if any of the following conditions is true,

1) Solution a is valid and solution b violates the constraint.
2) Solutions a and b both violate the constraint, but solution a has less violation value.
3) Solution a and b both valid and solution a has better fitness value.

The parameter settings for GRGA are shown in Table 7.

TABLE 7. Parameters of the PID controller optimization

Parameters			**Values**
No. of generations			250
Population Size			100
Probability of Crossover			0.8
Probability of Mutation			0.1666
Distribution index in SBX			10
Distribution in polynomial mutation			20
Variable bounds	PID 1	K_{p1}	$[0, 100]$
		K_{i1}	$[0, 50]$
		K_{d1}	$[0, 50]$
	PID 2	K_{p2}	$[0, 500]$
		K_{i2}	$[0, 100]$
		K_{d2}	$[0, 100]$

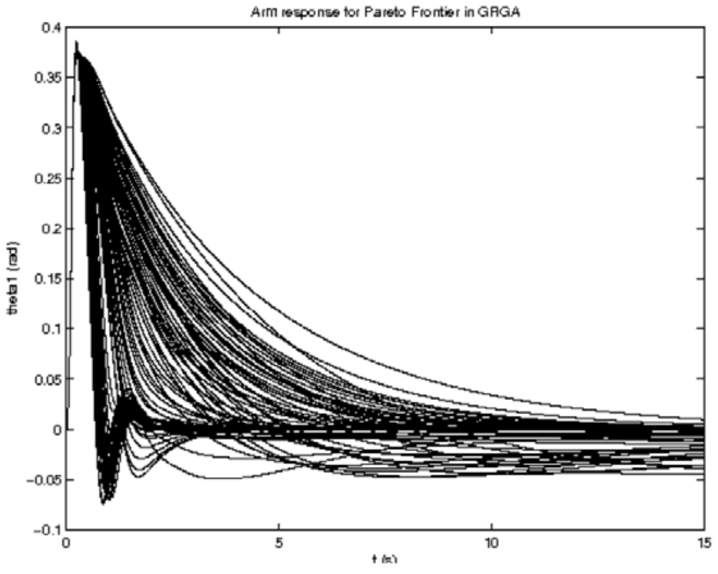


FIGURE 5. Arm responses for pareto solutions from GRGA

Unlike any conventional/manual methods of tuning the PID controller, GRGA provides a large number of solutions (pareto frontier) to the control designers, giving them more

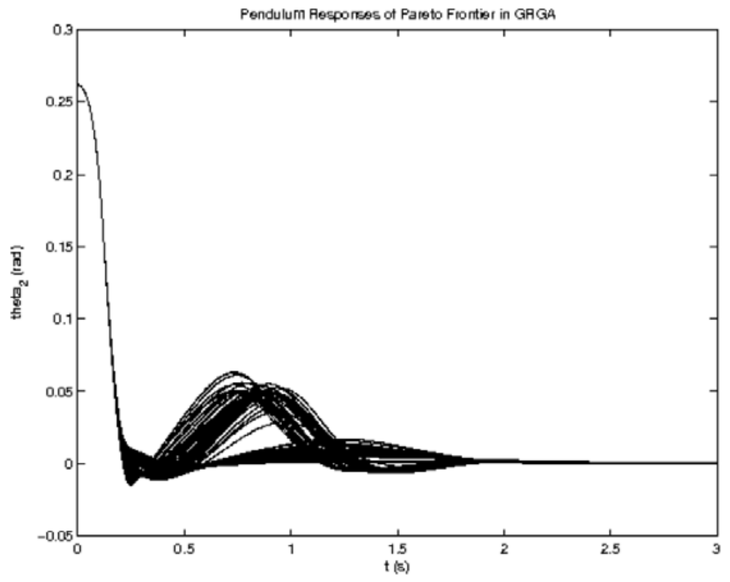


FIGURE 6. Pendulum responses for pareto solutions from GRGA

flexibility and options by considering the various design trade-off. Figures 5 and 6 show the 98 of final solutions (pareto frontier) which then are extracted to gives the arm and pendulum responses respectively.

Interestingly, all the final pareto front solutions in this experiment are closed-loop stable which gives another credit to GRGA compared with the manual tuning approach. This guarantee of stability will add reliability to the designers when choosing GRGA as the medium of tuning controller. The effect of diversity preservation can be seen in Figures 5 and 6 where the various response characteristics of the arm and pendulum are obtained. For instance, response with lower settling time has larger overshoot or vice versa.

Figures 7 and 8 show an example where the solution that give smallest settling time for both are chosen.

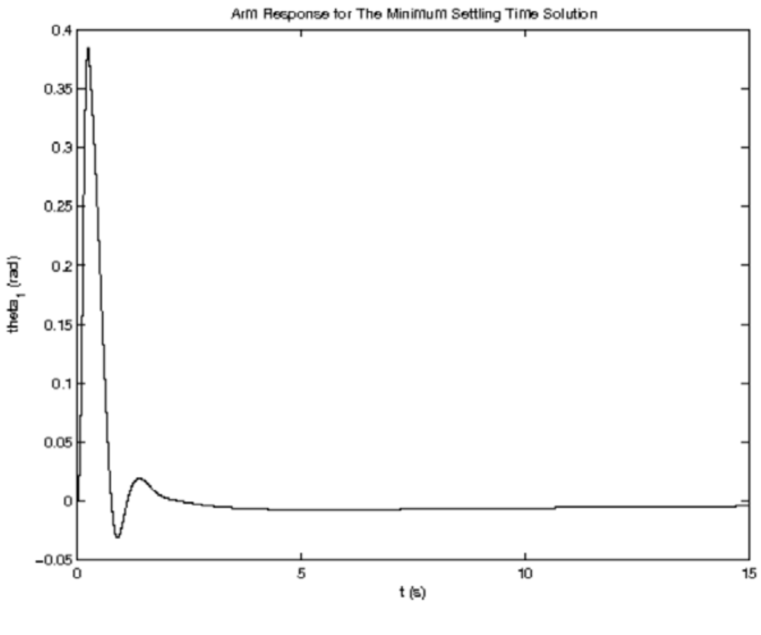


FIGURE 7. Arm response for the most minimum settling time solution

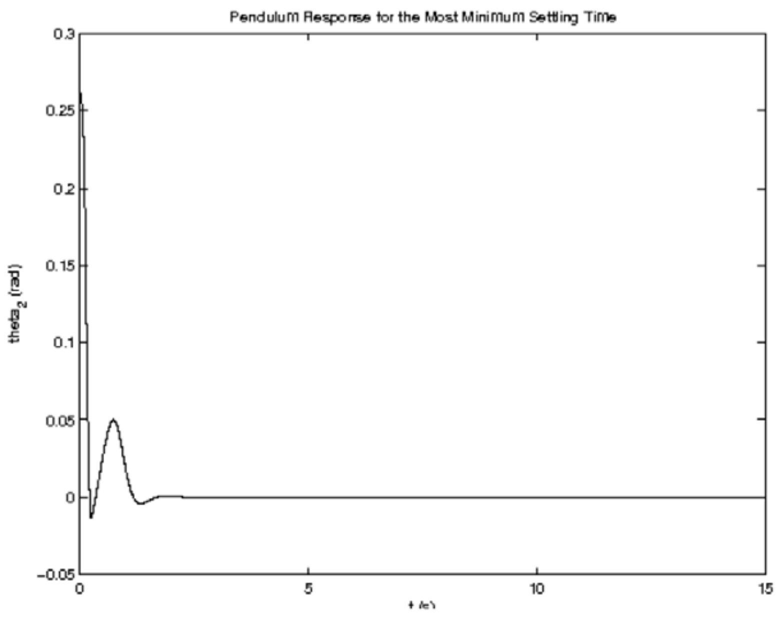


FIGURE 8. Pendulum response for the most minimum settling time solution

Analytically from this solution, the settling time is at 0.6760 s, the overshoot is 0.3849% and the IAE is 0.0479.

6. Conclusions.

A MOEA based on the combination of population and pareto based fitness assignments method with the good convergence property has been proposed in this paper. Based on the five ZDT test problems, the proposed GRGA was able to converge faster than NSGA-II. However, in the early generation, GRGA tends to lose its solution's diversity. Hence, the diversity preservation technique in GRGA should further be improved in order to overcome such a problem.

In the application of GRGA in PID controller tuning of RIP, tuning mechanism with GRGA has shown several advantages compared with the manual tuning methods. Due to the reliability, the capability in dealing with complex problems and the better convergence property, GRGA should gain increased attention and application in the near future.

Acknowledgment. The authors would like to thank Universiti Teknologi Malaysia and the Ministry of Higher Education, Malaysia for their supports.

REFERENCES

[1] K. M. Eibayomy, Z. Jiao and H. Zhang, PID controller optimization by GA and its performances on the electro-hydraulic servo control system, *Chinese Journal of Aeronautics*, vol.21, no.4, pp.378-384, 2008.
[2] P. van Overschee et al., RAPID: The end of heuristic PID tuning, *Journal A*, vol.38, no.3, pp.6-10, 1997.
[3] I. Mizumoto, H. Tanaka and Z. Iwai, Adaptive PID control for nonlinear systems with a parallel feedforward compensator, *International Journal of Innovative Computing, Information and Control*, vol.6, no.7, pp.2901-2916, 2010.
[4] R. Ciancone and T. Marlin, Tune controllers to meet plant objectives, *Control May*, pp.50-57, 1992.
[5] A. Lopez, P. Murrill and C. Smith, Tuning PI and PID digital controllers, *Instruments and Control Systems*, vol.42, pp.89-95, 1969.
[6] V. Guliashki, H. Toshev and C. Korsemov, Survey of evolutionary algorithms used in multiobjective optimization, *Problems of Engineering Cybernetics and Robotics*, vol.60, pp.42-54, 2009.

[7] C. A. Coello and G. B. Lamont, *Applications of Multi-objective Evolutionary Algorithms*, World Scientific Pub Co Inc., 2004.

[8] D. Martín, R. Toro, R. Haber and J. Dorronsoro, Optimal tuning of a networked linear controller using a multi-objective genetic algorithm and its application to one complex electromechanical process, *International Journal of Innovative Computing, Information and Control*, vol.5, no.10(B), pp.3405-3414, 2009.

[9] J. Schaffer, *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*, L. Erlbaum Associates Inc., 1985.

[10] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[11] C. M. Fonseca and P. J. Fleming, An overview of evolutionary algorithms in multiobjective optimization, *Evolutionary Computation*, vol.3, no.1, pp.1-16, 1995.

[12] N. Srinivas and K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms, *Evolutionary Computation*, vol.2, no.3, pp.221-248, 1994.

[13] K. Deb et al., A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, vol.6, no.2, pp.182-197, 2002.

[14] K. Deb and R. Agrawal, Simulated binary crossover for continuous search space, *Complex Systems*, vol.9, no.2, pp.115-148, 1995.

[15] E. Zitzler, K. Deb and L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, *Evolutionary Computation*, vol.8, no.2, pp.173-195, 2000.

[16] K. Deb and S. Jain, Running performance metrics for evolutionary multi-objective optimization, *KanGAL Report*, 2002.

[17] E. Zitzler, M. Laumanns and L. Thiele, SPEA2: Improving the strength pareto evolutionary algorithm, *Technical Report TIK-Rep 103*, Swiss Federal Institute of Technology, Lausanne, Switzerland, 2001.

[18] E. Zitzler and L. Thiele, Multiobjective optimization using evolutionary algorithms – A comparative case study, *Proc. of the 5th International Conference on Parallel Problem Solving from Nature*, pp.292-301, 1998.

[19] C. Liu, New evolutionary algorithm for multi-objective constrained optimization, *ICIC Express Letters*, vol.2, no.4, pp.339-344, 2008.

[20] A. Kozáková, Tuning decentralized PID controllers for performance and robust stability, *ICIC Express Letters*, vol.2, no.2, pp.117-122, 2008.

Appendix 1.

m_1 = mass of arm (0.056 kg).
m_2 = mass of pendulum (0.022 kg).
l_1 = length of arm (0.16 m).
l_2 = length of pendulum (0.16 m).
c_1 = distance to the center of mass of arm (0.08 m).
c_2 = distance to the center of mass of pendulum (0.08 m).
J_1 = inertia of arm ($0.00215058 \text{ kg} \cdot \text{m}^2$).
J_2 = inertia of pendulum ($0.00018773 \text{ kg} \cdot \text{m}^2$).
θ_1 = angular displacement of arm.
θ'_1 = angular velocity of arm.
θ_2 = angular displacement of pendulum.
θ'_2 = angular velocity of pendulum.
τ_1 = applied torque.
g = gravitational acceleration ($9.81 \text{ m} \cdot \text{s}^{-2}$).
k_b = back-emf constant ($0.01826 \text{ V} \cdot \text{s/rad}$).
k_t = torque constant ($0.01826 \text{ N} \cdot \text{m/A}$).
R_m = armature resistance ($2.5604\ \Omega$).
e = input voltage.