Proceedings of the 2001 IEEE International
Conference on Control Applications
September 5–7, 2001 • México City, México

MULTIOBJECTIVE OPTIMIZATION DESIGN VIA GENETIC ALGORITHM

Haiming Lu, *Student Member, IEEE*, Gary G. Yen, *Senior Member, IEEE*
lhaimin@okstate.edu gyen@okstate.edu
School of Electrical and Computer Engineering
Oklahoma State University, Stillwater, OK, 74078

Abstract–**Many real-world problems involve multiple objectives that need to be optimized simultaneously. However, in most cases, a suitable optimal solution meeting all the objectives can hardly be found since these objectives are generally conflicting. Compared to conventional optimization techniques, Genetic Algorithms (GA's) are well suited to solve Multiobjective Optimization (MO) problems since a family of "acceptable" solutions—a so called Pareto set—can be identified by different individuals through the evolution process. However, most of the existing Multiobjective Optimization Genetic Algorithms (MOGAs) have difficulty dealing with the trade-off between uniformly distributing the computational resources and avoiding the "genetic drift" phenomenon. This paper proposes a new evolutionary approach to MO problems—the Rank-Density based Genetic Algorithm (RDGA), which can be characterized as a) simplifying the problem domain by converting high-dimensional multiple objectives into two objectives to minimize the individual rank value and population density value, b) searching for and keeping better-approximated Pareto points by diffusion and elitism schemes, and c) preventing harmful individuals by introducing a "forbidden region" concept. From the result of the simulation study, RDGA clearly outperforms two representative MOGAs on three benchmark testing problems in terms of keeping the diversity of the individuals along trade-off surface, tending to extend the Pareto front to new areas, and finding a well-approximated Pareto optimal set.**

I. INTRODUCTION

In many scientific and engineering disciplines, it is not uncommon to experience a design difficulty when there are several design objectives to be met simultaneously. If the objectives are conflicting, then the problem becomes one of finding the best possible design that satisfies the conflicting objectives under different trade-off scenarios. With these multiple objectives and constraints taken into consideration, an optimum design problem can then be formulated. This type of problem is known as a *multiobjective*, *multicriteria*, or *vector optimization* problem [1].

Multiobjective Optimization (MO) is a very important research topic because most real world problems have not only a multiobjective nature but also many open issues to be answered qualitatively and quantitatively. In fact, the solution to a MO problem is generally not a single point. It consists of a family of non-dominated points, a so called Pareto front, which describes the trade-off among contradicted objectives [2]. The Pareto front yields many candidate solutions, from which we can choose the desired one under different trade-off conditions. In most cases, the Pareto front is on the boundary of the feasible range as shown in Figure 1.

In their early development, Genetic Algorithms (GA's), a class of population-based optimization approaches, have been recognized to be well suited for multiobjective optimization. In GA's, multiple individuals can search for multiple solutions in parallel, advantageously producing a family of possible solutions to the problem. The ability to handle complex problems involving features such as discontinuities, multimodality, and disjoint feasible spaces, reinforces the potential effectiveness of GA's in multiobjective search and optimization [3].

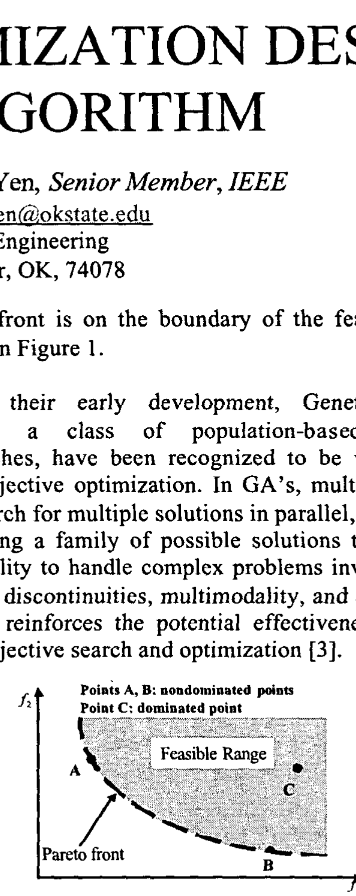


***Figure 1** Graphical illustration of the Pareto optimality*

Since the 1980's, several Multiobjective Genetic Algorithms (MOGAs) have been proposed and applied in MO problems [1]. These algorithms all have almost the same purpose—searching for a uniformly distributed and near-optimal Pareto front for a given MO problem. However, this ultimate goal is far from accomplished by the existing MOGAs described in literature. In one respect, most of the MO problems are very complicated and require the computational resources to be homogenously distributed in a high dimensional search space. On the other hand, those better-fit individuals generally have strong tendencies to restrict searching efforts within certain areas due to the "genetic drift" phenomenon. This is yet another trade-off decision pertaining to the *efficiency* and *efficacy* dilemma [3].

This paper proposes a multiobjective genetic algorithm named the Rank-Density based Genetic Algorithm (RDGA). Applied in three test functions, RDGA outperforms the other algorithms under consideration by providing a near-optimal and near-uniformly distributed Pareto front. The remainder of this paper is organized as follows: Section 2 reviews two existing typical MOGAs. Section 3 proposes a new Rank-Density based Genetic Algorithm that is designed to deal with high dimensional objective functions, explore the optimality of the candidate Pareto points, and maintain the diversity of the final Pareto front. In Section 4, we apply RDGA with the other two MOGAs to three test functions. The resulting Pareto sets are examined to compare the performances of RDGA and two well-regarded algorithms. Finally, Section 5 provides some concluding remarks along with pertinent observations.

II. GENETIC ALGORITHMS IN SOLVING MO PROBLEMS

Generally, MOGAs can be categorized by various fitness assignment strategies. In particular, vector evaluated genetic algorithm and niched Pareto genetic algorithm are two known representatives of population based MOGAs and have been exploited in various applications [3].

A. *Vector Evaluated Genetic Algorithm—VEGA*

VEGA is the most typical population based non-Pareto MOGA [4]. In VEGA, the entire population is divided into n subpopulations with equal size; subpopulation i is filled with individuals that are randomly chosen from the current population according to an objective i. Afterwards, the entire population is shuffled, and crossover and mutation are then performed as usual. However, VEGA has difficulty in finding a uniformly distributed Pareto front because it does not incorporate any diversity-keeping scheme. Furthermore, VEGA can be shown to perform an implicitly-weighted sum of the objectives. This leads to the difficulty found in conventional multiobjective optimization approaches to search for a Pareto front when the problem has a concave trade-off surface.

B. *Niched Pareto Genetic Algorithm—NPGA*

NPGA is a typical Pareto-based multiobjective genetic algorithm [5]. It combines tournament selection and the concept of Pareto dominance. To compare two individuals, a number of other individuals are randomly selected to help determine dominance. When both competitors are either dominated or non-dominated, the result of the tournament is decided through *fitness sharing*—the individual with the fewest individuals in its niche is selected for reproduction. Based on fitness sharing technique, the more individuals a niche contains, the more its members' fitness values degrade. Since NPGA only applies Pareto selection to a portion of the entire population in each generation, it is very fast compared to the other Pareto-based approaches. In addition, it can produce good non-dominated solutions that can be kept for a large number of generations. The weakness of NPGA is that it requires heuristic choices of the sharing factors and the size of the tournament, which makes the process relatively complex in practice. Moreover, as the sharing technique degrades the fitness value, "harmful" individuals may be generated which cause the entire population evolve in a direction opposite to the Pareto front [6].

III. RANK-DENSITY BASED GENETIC ALGORITHM

From the literature review, the main difficulty in existing MOGA approaches is designing a suitable fitness assignment strategy in order to find a near-complete and near-optimal approximated Pareto front for the given optimization problem. Unfortunately, these two objectives are contradictory. In one respect, the "genetic drift" character of GA needs to be encouraged to converge the solution to a nearly optimal point. On the other hand, the "genetic drift" phenomenon must be avoided in order to sketch a uniformly sampled trade-off surface for the final Pareto front. Based on these considerations, a new Rank-Density based Genetic Algorithm (RDGA), which converts a high dimensional MO problem into a bi-objective optimization problem to minimize fitness rank values and cell densities, is proposed. Five crucial procedures of RDGA will be discussed as follows.

A. *Automatic ranking*

First introduced by Goldberg [7], population-ranking schemes have been widely applied in MOGAs. In this paper, we propose an automatic ranking strategy. Suppose we want to minimize two objectives, f_1 and f_2, and GA generates five individuals as shown in Figure 2. To describe the dominated relationship of these individuals, we define the comparison function as:

$c_{i,j,k} = 1$, if individual j has better fitness than individual k in terms of the i_{th} objective. Otherwise, $c_{i,j,k} = 0$. Then we can calculate the rank value $r_{i,j}$ for each individual j in terms of the i_{th} objective

$$r_{i,j} = \sum_{k=1}^{m} c_{i,j,k}, \quad j = 1,\dots,m, \quad i = 1,\dots,n, \tag{1}$$

where m is the total number of individuals, and n is the number of objectives to be optimized. The final rank value of individual j is assigned as

$$r_j = \max_j(\max_i(r_{i,j})) - \max_i(r_{i,j}) + 1, \quad j = 1,\dots,m, \quad i = 1,\dots,n \tag{2}$$

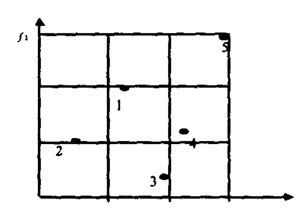


Figure 2 Dominated relationships among five individuals

From the MO problem showed in Figure 2, the automatic rank value of each individual can be derived as shown in Figure 3. The individuals with the smallest rank values are located on the current Pareto front. From this scheme, the resulting rank values represent the dominant relationships among different individuals and can be derived in any dimensional MO problems.

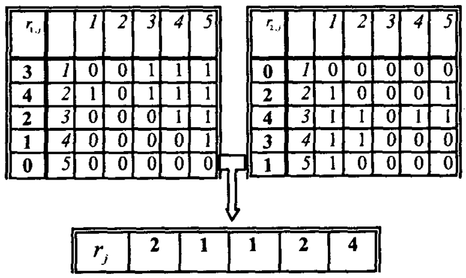


Figure 3 Resulting rank values for the individuals in Figure 2

B. Adaptive density value calculation

To maintain the diversity of the obtained Pareto front, a near-homogeneous search space is absolutely necessary. In this paper, we present an adaptive cell density evaluation scheme as shown in Figure 2. The cell width in each objective dimension can be formed as

$$d_i = \frac{\max_{x \in X} f_i(x) - \min_{x \in X} f_i(x)}{K_i}, i = 1,\ldots, n. \tag{3}$$

where d_i is the width of the cell in the i_{th} dimension, K_i denotes the number of cells designated for the i_{th} dimension (i.e., in Figure 2, $K_i = 3$). As the maximum and minimum fitness values will change with different generations, the cell size will vary from generation to generation to maintain the accuracy of the density calculation. The density value of an individual is defined as the number of the individuals located in the same cell.

C. Rank and density based fitness assignment

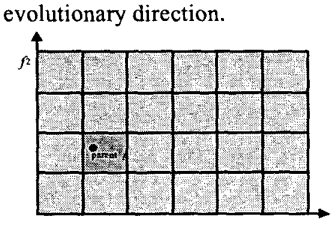
Because rank and density values represent both fitness and population diversity, respectively, we assigned them as two important attributes to each individual. Therefore, any multiobjective optimization problem can be converted into a bi-objective optimization problem. On the other hand, since we need to minimize rank value together with density value, some further modifications need to be made to the original notation.

First, instead of minimizing the density value of an individual, we minimize the density value of the entire population. Based upon the definition of the cell density, an individual located in a crowded cell must have a relatively high density value, which contributes much more to the population density value than the sparse area does. For example, a cell containing 10 individuals will contribute $10 \times 10 = 100$ to the population density value, whereas a cell containing only one individual will contribute 1 to the population density value.

Second, after the rank and density values of each individual have been extracted, a modified VEGA is applied to fulfill fitness assignment. As discussed in Section 2, VEGA has two deficiencies: 1) it does not have a scheme to maintain the diversity of the evolved Pareto front, and 2) it has difficulty in dealing with the problems with concave trade-off surface. As mentioned above, the goal of RDGA is to find the non-dominated individuals with rank value equal to 1 and reduce the population density value to obtain a uniformly distributed trade-off surface. In this setting, there is no concern about keeping the population diversity in the rank-density domain. Furthermore, whether the "Pareto front" in the rank-density domain is concave is not an issue, since it is not a real Pareto front for the MO problem under consideration. Therefore, a simple VEGA is effective enough to fulfill fitness assignment after the original optimization problem has been transformed into the rank-density domain.

D. Crossover and mutation

For crossover, the parent selection and replacement schemes are borrowed from Cellular GA [8] to explore the new search area by "diffusion." For each subpopulation, a fixed number of M parents are randomly selected for crossover. Then, each selected parent performs crossover with the best individual (the one with the lowest rank value) within the same cell and neighboring cells. If one offspring produces better fitness (a lower rank value or a lower population density value) than its corresponding parent, it replaces its parent. The replacement scheme of the mutation operation is analogous. To prevent "harmful" offspring surviving and affecting the evolutionary direction and speed [8], a *forbidden region* concept is proposed herein as the replacement scheme for the density subpopulation, thereby preventing the "backward" effect [6]. The *forbidden region* includes all the cells dominated by the selected parent. The offspring located in the forbidden region will not survive in the next generation, and thus the selected parent will not be replaced. As shown in Figure 4, suppose our goal is to minimize objectives f_1 and f_2, and a resulting offspring of the selected parent p is located in the forbidden region. By RDGA, this offspring will be eliminated even if it reduces the population density because this kind of offspring has the tendency to push the entire population away from the desired evolutionary direction.



The cell where the selected parent p locates in
Valid range where parent p's offspring can locate in
Forbidden region where parent p's offspring cannot locate in

Figure 4 Illustration of the valid range and the forbidden region

E. Archiving the candidate Pareto points

The elitism scheme [1] is also applied in RDGA for storing the Pareto points obtained in each generation. These points are compared to achieve the final Pareto front after the evolution process has stopped.

IV. SIMULATION STUDY

To proof-of-the-concept, we apply the proposed RDGA to three numeric MOGA test functions in comparison with VEGA and NPGA.

Problem I: Schaffer's F2 [4]

Minimize $f_1(x)$ and $f_2(x)$, where

$$f_1(x) = x^2,\quad f_2(x) = (x-2)^2, \tag{4}$$

with $0 \le x \le 20$.

Problem II: Modified Deb's MO [9]

Minimize $f_1(x,y)$ and $f_2(x,y)$, where

$$f_1(x,y) = x,\quad f_2(x,y) = (1+y)\times(1-\frac{x}{1+y})^2 - \frac{x}{1+y}\times\sin(10\pi x) \tag{5}$$

with $0 \le x, y \le 1$.

Problem III: Modified Tanaka's MO [10]

Minimize $f_1(x,y)$ and $f_2(x,y)$, where

$$f_1(x,y) = x,\quad f_2(x,y) = y \tag{6}$$

with $0 \le x, y \le \pi$,

$$(x-0.5)^2 - 5\times(y-0.5)^2 < 0,$$

$$-(x^2+y^2) + 1 + 0.1\times\cos(16\arctan(\frac{x}{y})) \le 0.$$

Here, for problem I, we use a population size 30 and run VEGA, NPGA, and the proposed RDGA for 1,000 generations. For problem II and III, population size and the maximum generations are selected to be 100 and 5,000 respectively. In addition, the population rank values and density values of VEGA and NPGA were also recorded in each generation for comparison.

Figure 5 (a) shows the feasible range and ideal final Pareto front for Problem I. Figure 5 (b), (c) and (d) show the final Pareto fronts evolved by VEGA, NPGA and RDGA respectively. Figure 5 (e) and (f) show the evolutionary trajectories of population rank and density, respectively. Likewise, Figure 6 (a)-(f) and Figure 7(a)-(f) display these results for Problem II and III respectively. From these figures, we can observe that:

1. Population density value can be a good indicator to show whether the genetic drift phenomenon has been prevented, and RDGA performs much better than VEGA and NPGA in maintaining the population density value within a low upper bound (see Figure 5(d), 5(f), 6(f) and 7(f)).
2. Minimizing the population rank value is another important strategy in RDGA. In fact, an individual's rank value implies its dominated status. By directly minimizing the rank value of each individual, RDGA provide a straightforward way to drag all the members of the given population to be nondominated solutions. This explains why RDGA produces smoother evolutionary trajectory and faster convergent speed in rank domain (see Figure 5(e), 6(e) and 7(e)).
3. RDGA also provides competitive non-dominated points on the final Pareto front. Most of them are better than those produced by VEGA and NPGA. (See Figure 5(b)-(d), Figure 6(b)-(d) and Figure 7(b)-(d)). This improvement is due to the elitism scheme and the "diffusion" effect provided by CGA.
4. For the MO problems with discontinuous or concave Pareto fronts (i.e., problems II and III), RDGA provides more uniformly distributed Pareto points than VEGA and NPGA. This is a beneficial result of the rank-density fitness scheme. In particular, the introduction of the "forbidden region" restrains the individuals from going "backward." Thus, most of the individuals who want to reduce their density values have to move themselves along the current Pareto front.

Although RDGA improves the *efficacy* of the searching process (*i.e.*, finding good Pareto front), it also sacrifices the *efficiency* of the process (computationally speaking) comparing to NPGA and VEGA. For each generation, the computational complexity is about 1.8:1.2:1 for RDGA, NPGA and VEGA. The extra time in RDGA is exerted on the ranking and density calculation and exploring the new searching area (*i.e.*, diffusion scheme). However, the degrading of efficiency is still acceptable because off-line design problems are not time-imminent.

V. CONCLUSION

From the results presented above, RDGA has shown its potential in successfully finding nearly optimal and nearly complete Pareto fronts for three given benchmark problems. Two other MOGAs are also applied for comparison. RDGA has the merits of simplifying the problem domain, keeping the diversity of the individuals along the current trade-off surface, extending the Pareto front to new areas, and finding a well-approximated non-dominated set. For the given benchmark problems, RDGA significantly improves searching efficacy in comparison with VEGA and NPGA. The trade-off of searching efficiency is not significant and is acceptable. Further research is necessary in applying RDGA to solve complicated real-world multiobjective optimization problems.

REFERENCES

[1] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, pp. 257-271, 1999.

[2] C. L. Hwang and A. S. M. Masud, *Multiple Objective Decision Making—Methods and Applications*, Vol. 164 of *Lecture Notes in Economics and Mathematical Systems*. Berlin, Germany: Springer-Verlag, 1979.

[3] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evol, Comput.*, vol. 3, pp. 1-16, 1995.

[4] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. 1st Int. Conf. Genetic Algorithms*, pp. 93-100, 1985.

[5] J, Horn, N. Nafpliotis and D. E. Goldberg, "A niched pareto genetic algorithm for multiobjetcive optimization," in *Proc. 1st IEEE Cong. Evolutionary Computation*, vol 1, pp. 82-87, 1994.

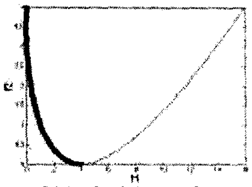
[6] G. W. Greenwood, X. S. Hu, and J. G. D'Ambrosio, "Fitness functions for multiple objective optimization problems: combining preferences with Pareto rankings," in *Proc. Foundations of Genetic Algorithms 4*, pp. 437-455, 1996.

[7] D. E. Goldberg, *Genetic algorithms in Search, Optimization and Machine Learning. Reading*, MA: Addison-Wesley, 1989.

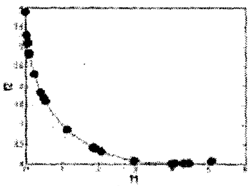
[8] Y. Davidor, "A naturally occurring niche & species phenomenon: the model and first results," in *Proc. 4th Int. Conf. Genetic Algorithms*, pp. 257-262, 1991.

[9] D. Kalyanmoy, "Multiobjective genetic algorithms: problem difficulties and construction of test problems," in Technical report TR CI-49/98, University of Dortmund, Germany, 1998.

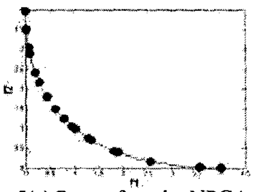
[10] Tanaka, Masahiro, et al., "GA-based decision support system for multicriteria optimization," in *Proc. Int. Conf. Systems, Man, and Cybernetics*, pp. 1556-1561, 1995.



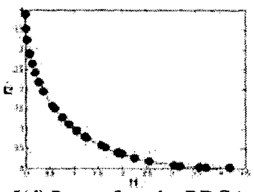
5(a) Ideal Pareto front



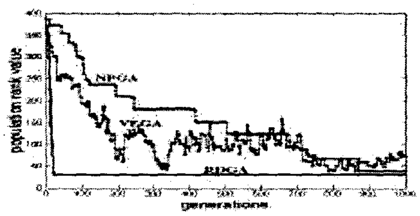
5(b) Pareto front by VEGA



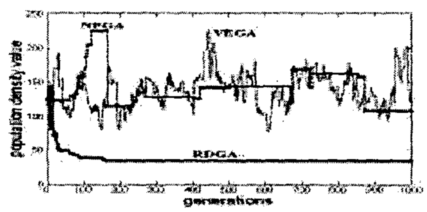
5(c) Pareto front by NPGA



5(d) Pareto front by RDGA

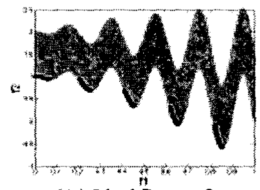


5(e) Population rank value evolving trajectory

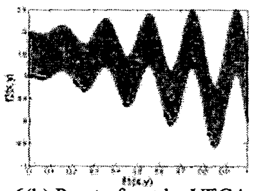


5(e) Population density value evolving trajectory

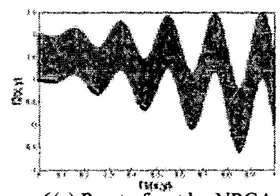
Figure 5 Simulation results for problem I



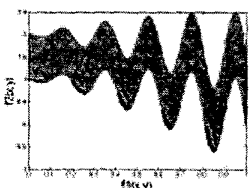
6(a) Ideal Pareto front



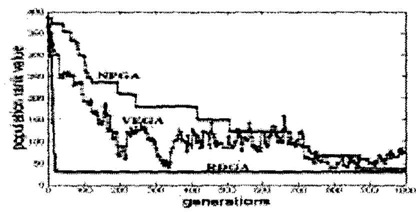
6(b) Pareto front by VEGA



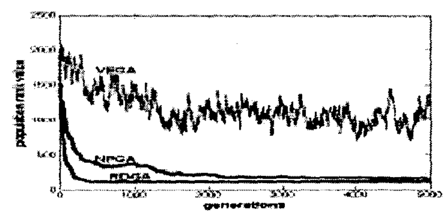
6(c) Pareto front by NPGA



6(d) Pareto front by RDGA

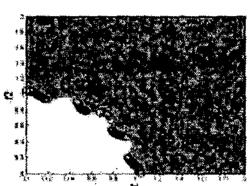


6(e) Population rank value evolving trajectory

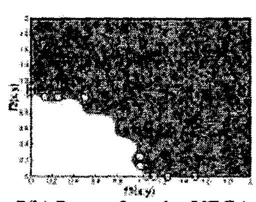


6(e) Population density value evolving trajectory

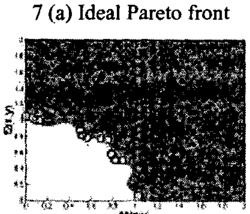
Figure 6 Simulation results for problem II



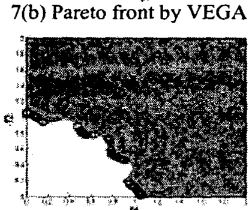
7 (a) Ideal Pareto front



7(b) Pareto front by VEGA

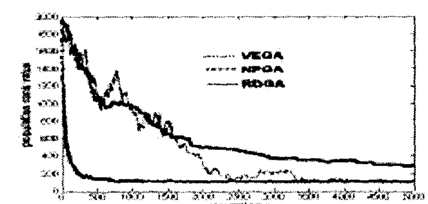


7(c) Pareto front by NPGA

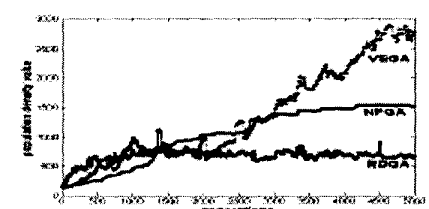


7(d) Pareto front by RDGA

Figure 7 Simulation results for problem III



7(e) Population rank value evolving trajectory



7(f) Population density value evolving trajectory