STUDY OF MULTI-OBJECTIVE OPTIMIZATION AND ITS IMPLEMENTATION USING NSGA-II

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
in
Electrical Engineering.

By

CHITTARANJAN SAMAL
SURYAKANT MALLICK
RAHUL KUMAR DAS

Department of Electrical Engineering

National Institute of Technology

Rourkela

2007

STUDY OF MULTI-OBJECTIVE OPTIMIZATION AND ITS IMPLEMENTATION USING NSGA-II

A THESIS SUBMITTED IN PARTIAL FULLFILMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
in
Electrical Engineering.

By
CHITTARANJAN SAMAL
SURYAKANT MALLICK
RAHUL KUMAR DAS

Under the Guidance of

Prof. (Dr.) P. K. NANDA.

Department of Electrical Engineering

National Institute of Technology

Rourkela

2007

National Institute of Technology
Rourkela

CERTIFICATE

This is to certify that the thesis entitled," Study of Multi-objective optimization and its implementation using NSGA-II" submitted by Sri Rahul Kumar Das, Sri Chittaranjan Samal and Sri Suryakant Mallick in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Electrical Engineering at the National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Date :

Place:

Prof. (Dr.) P.K Nanda.
Dept. of Electrical Engg.
National Institute of technology
Rourkela-769008

ACKNOWLEDGEMENT

We would like to articulate our deep gratitude to our project guide Prof.(Dr.) P.K. Nanda who has always been our motivation for carrying out the project.

A project of this nature could never have been attempted with our reference to and inspiration from the works of others whose details are mentioned in references section. We acknowledge our indebtedness to all of them. Last but not the least, our sincere thanks to all of our friends who have patiently extended all sorts of help for accomplishing this undertaking.

RAHUL KUMAR DAS (10302018)

CHITTARANJAN SAMAL (10302047)

SURYAKANT MALLICK (10302046)

CONTENTS

ABSTRACT

This project investigates the Multi-objective optimization strategies and their solutions using Multi-objective evolutionary algorithms (MOEAs). Multi-objective evolutionary algorithms (MOEAs) that use non-dominated sorting and sharing are criticized mainly for their; a) computational complexity, b) lack of elitism, c) need for specifying sharing parameter. In this paper the Non-Dominated Sorting Genetic Algorithm (NSGA) is studied and NSGA-II as proposed by Deb et. al. has been implemented, which alleviates the above three difficulties. In this study different objectives have been considered with
different variables and constraints. The algorithm yielded satisfactory simulation results in all the different cases. The effect of the genetic parameters on the Pareto-Optimal front in all the cases has been studied. The results show that NSGA-II find much better spread of solutions and better convergence near the true pareto optimal front compared to other elitist MOEAs.

Chapter 1

INTRODUCTION

1. INTODUCTION

Genetic Algorithms (GAs) implement optimization strategies based on simulation of natural law of evolution of species by natural selection, in order to obtain the fittest individual in the evolutionistic sense. Adopting this analogy, the optimal solution corresponds to the fittest individual. According to Darwin's Theory of Evolution: considering the population which evolves in a particular environment, only the fittest will be able to reproduce, handing down their chromosomes, while less fit ones will be doomed to extinction, due to environmental constraints. Like the evolutionary process GAs work as follows:

- Codification of variables involved in suitable strings.
- Whole function domain is created by randomly selecting a population of strings.
- At each iteration some elements of the population is chosen to reproduce through certain operators, according to their capability of adaptation to the environment.
- Iterative procedures lead the population of parameters towards the optimal solution.

1.1. MULTIOBJECTIVE OPTIMIZATION (MOP)

As the name suggests it deals with finding of optimal solutions to problems dealing with multiple objectives. Many real world search & optimization problems naturally involve multiple objectives, simply because in these problems, a user is never satisfied by finding one solution that is optimum with respect to a single criterion. Conflicting objectives introduce trade-of solutions & make the task complex. The presence of conflicting objectives give rise to a set of optimal solutions(called Pareto Optimal solutions) instead of a single optimal solution. In the absence of any priority towards any particular objectives, all pareto optimal solutions become equally important to the user.

Evolutionary Algorithms (EAs) are a natural choice for solving multi criterion optimization problems because of their population approach. Thinking along the following two lines (i) preference to non-dominated solutions in a population (ii)maintenance of diversity among non-dominated solutions as resulted in a number of successful multi criterion evolutionary algorithm, such as Hom, Nafploitis & Goldberg's niche populations.

Classical optimization methods suggest converting the multi objective optimization by emphasizing one particular Pareto-optimal solution at a time. Over the past decade a number of multi-objective evolutionary algorithms have been suggested such as Pareto Archived Evolution Strategy (PAES) & Strength Pareto Evolutionary Algorithm (SPEA), Non-Dominated Sorting GA (NSGA) etc.

Over the years, the main criticisms of the NSGA approach are,

1) High computational complexity of non-dominated sorting: The currently-used non-dominated sorting algorithm has a computational complexity of (M*N^3) where M is the number of objectives and N is the population size. This makes NSGA computationally

expensive for large population sizes. This large complexity arises because of complexity involved in non-dominated sorting procedure in each generation.

2) Lack of elitism: Elitism can speed up the performance of the GA significantly, which also prevents the loss of good solutions once they are found.

3) Need for specifying the sharing parameters : Traditional mechanisms of ensuring diversity in a population so as to get a wide variety of equivalent solutions have relied mostly on the concept of sharing. The main problem in sharing is the need to specify a sharing parameter.

In this report along with mentioning the basic concepts of MOP we have considered an improved version of NSGA known as NSGA-II. NSGA-II uses a faster sorting procedure, an elitism preserving approach and a parameter less niching operator.

Chapter 2

CONCEPTS OF MULTIOBJECTIVE OPTIMIZATION

2. CONCEPTS OF MULTIOBJECTIVE OPTIMIZATION (MOP)

2.1. MOP DEFINITION AND OVERVIEW:

Although a single objective optimization problem may have a unique solution, MOPs present a possibly unaccountable set of solutions that, when evaluated, produce vectors whose components represent trade-offs in objective space. A decision maker then implicitly chooses an acceptable solution by selecting one or more of the vectors. MOPs are mathematically defined as follows:

2.1.1. GENERAL MOP:

In general, an MOP minimizes $F(x)=(f_1(x),\ldots\ldots..,f_k(x))$ subject to $g_i(x)<=0$ where i= 0,1,2……m. An MOP solution minimizes the component of vectors F(x), where x is an n-dimensional decision variable vector ($X=x_1,x_2,\ldots..x_n$)
from some particular space.

An MOP consists of n decision variables ,m constraints and k objectives of which any or all of the objective functions may be linear or non linear. MOPs are characterized by distinct measures of performance (in)dependent and/or in commensurable. The multi-objective being optimized almost always conflict placing a partial rather than total ordering on the search space. Regardless of the implemented technique, a key concept many researchers use in determining MOP solution is that of pareto optimality.

2.2 PARETO CONCEPTS:

2.2.1.PARETO DOMINANCE:

A vector $u=(u_1,u_2,\ldots..,u_k)$ is said to dominate vector $v=(v_1,v_2,\ldots,v_k)$ if and only if $u_i< v_i$ for i=1,2,….,k.

2.2.2.PARETO OPTIMALITY:

A solution x_i belonging to a particular space R is said to be pareto optimal with respect to R if and only if there is no x_2 belonging to R for which $F(x_2)= (f_1(x_2), f_2(x_2),\ldots,f_k(x_2))$ dominates $F(x_1)=(f_1(x_1),f_2(x_1),\ldots,f_k(x_1))$. The phrase pareto optimal is taken to mean with respect to the entire decision variable space unless otherwise specified.

2.2.3.PARETO OPTIMAL SET:

For a given MOP F(x), the pareto optimal set P is defined as

$P=\{x_1$ belongs to R:| x_2 belongs to R:$F(x_2)<=F(x_1)\}$

2.2.4.PARETO FRONT:

For a given MOP F(x) and a pareto optimal set P, the pareto front (PF) is defined as

$PF= \{u=F(x)=(f_1(x), f_2(x),\ldots..,f_k(x)$:x belong to P)}

If for example there are two objectives f1, f2 to be optimized pareto optimal set form the set of all those solutions whose corresponding vectors are non dominated with respect to all other comparison vectors; we stress here that pareto optimal solution are classified as such based on

their evaluated functional values. When plotted in objective space , the non dominated vectors are called pareto front.
The following procedures can be used to find non-dominated solutions:
M= no of objectives
N = no of solutions

Step 1: begin with i=1
Step 2: for all j != I, compare solutions x(i) and x(j) using the condition of non domination.
Step 3: if for any j, x(i) is dominated by x(j), mark x(i) as dominated. Increment i by 1 and go to step 2.
Step 4: if all solutions (i.e. when i=N) are considered go to step 5, else increment i and go to step 2.
Step 5: all solutions not marked dominated are non dominated.

2.2.5.PARETO NOTATION:

During multi objective evolutionary algorithm a " current " set of pareto optimal solution (with respect to current generation) is determined at each generation's end and termed $P_{current}(t)$, where t represents generation no. Many algorithms also use secondary population storing non dominated solutions found through generations. Because a solution's classification as pareto optimal depends upon the context within which it is evaluated, corresponding vectors of this secondary population must be tested and those solutions whose associated vectors are dominated removed. We term this as secondary population $P_{known}(t)$. Different secondary population strategies exist. The simplest is when $P_{current}(t)$ is added at each generation (i.e.$P_{current}(t)$ U $P_{current}(t-1)$). At any given stage , $P_{known}(t)$ is thus the set of pareto optimal set denoted as $P_{true}(t)$.

2.2.6.LOCAL PARETO OPTIMAL SET:

If for every member x in a set P there exist no solution y satisfying mod(y-x) <= p where p is a small positive number (y is obtained by perturbing x in a small neighborhood) dominating any member in the set P, then solution belonging to set P constitute a local pareto optimal front.

2.2.7.GLOBAL PARETO OPTIMAL SET:

If there is no solution in the search space that dominate any member in the set P, then the solutions belonging set P constitute a global pareto optimal set.

1. Guide the search towards the global pareto optimal region.
2. Maintain population diversity(in the function space, parameter space, or both) in the current non dominated front.

2.3. A SPECIAL TWO OBJECTIVE OPTIMIZATION PROBLEM:

Consider a simple two objective optimization problem having two variables x1 and x2.

Minimize $f_1(x_1,x_2) = x_1$.

Minimize $f_2(x_1,x_2) = g(x_2)/x_1$.

Where $g(x_2)$ is a function of x_2 only. Thus the first objective function of both x_1 and x_2 in the function space (with (f_1,f_2)values), the above two functions obey the following relationships.

$f_1(x_1,x_2)*f_2(x_1,x_2) = g(x_2)$

for a fixed value of $g(x_2) = c$, a f_1 vs f_2 plot becomes a hyperbola($f_1*f_2 = c$). There exists a no of intuitive properties of the above two objective problem.

Lemma 1:
If for any two solution, the second variable x_2(or more specifically $g(x_2)$) are the same, both solutions are not dominated by each other.
The proof follows from $f_1 * f_2 = c$.

Lemma 2:
If for any two solutions, the first variable x_1 are the same, the solution corresponding in the minimum $g(x_2)$ dominates the other solutions.

Proof:
Since $x_1(1) = x_1(2)$ the first objective function values are the same, the solution having smaller $g(x_2)$ (meaning better f_2) dominates the other solutions.

Lemma 3:
For any two arbitrary solution x(1) and x(2) where $x_i(1) = x_i(2)$ for i= 1,2, and $g(x_2(1)) < g(x_2(2))$, there exist a solution $x(3) = (x_1(2), x_2(1))$ which dominate solution x(2).

Proof:
Since the solution x(3) and x(2) have the same x_1 values and $g(x(1)) < g(x(2))$, x(3) dominates x(2) according to lemma 2.

Corollary:
The solutions x(1) and x(3) have the same x_2 value and hence they are non dominated to each other according to lemma 1.

Based on the above discussion, we can present the following theorem.

Theorem:
The two objective problem described has local or global pareto optimal solutions(x_1,x_2), x_2 is the locally or globally minimum solution of $g(x_2)$ respectively and x_1 can take the value.

Proof:
Since solutions with a minimum g(x2) have the smallest possible g(x2) (in the neighborhood sense in case of local optimum and in the whole search space in case of local minimum), according to lemma 2, all such solutions dominate any other solution in the respective context. Since these solution are also non dominated to each other, they are pareto optimal solutions in the respective sense.

Lemma 4:
Although some members in a non dominated set are members of the pareto optimal front, not all members are necessarily members of the parteto optimal front.

Proof:

Say there are two distinct members in a set of which x(1) is a member of pareto optimal front and x(2) is not. The solution is chosen in such a way that $x1(2) < f1(x(1))$. Since $g2(x2(2)) > g2(x2(1))$ it follows that $f2(x(2)) > f2(x(1))$ thus x(1) and x(2) are non dominated solutions.

The pareto optimal front formed by finding the best non dominated set of solutions, it is important to realize that all solutions in the best non dominated set obtained by an optimizer may not be necessarily the members of pareto optimal set. However in absence of any better approach, a method of seeking the best set of non dominated solution is reasonable approach.

Chapter 3

NON-DOMINATED SORTING GENETIC ALGORITHM(NSGA-II)

3.NON DOMINATED SORTING GENETIC ALGORITHM (NSGA-II):

3.1.FAST NON-DOMONATED SORTING APPROACH:

For the sake of clarity, let us first describe a slow procedure of sorting a population into different non domination levels. Thereafter, we describe a faster approach.

In this slow approach, in order to identify solutions of the first non dominated front in a population of size N, each solution can be compared with every other solution in the population to find if it is dominated. This requires **M*N** comparisons for each solution, where M is the number of objectives. When this process is continued to find all members of the first non dominated level in the population, the total complexity is **(M*N^2).** In order to find the individuals in the next non dominated front the solutions of the first front are discounted temporarily
and the above procedure is repeated. In the worst case, the task of finding the second front also requires**(M*N^2)** computations, particularly when N number of solutions belong to the second and higher non dominated levels. This argument is true for finding third and higher levels of non domination. Thus, the worst case is when there are fronts and there exists only one solution in each front. This requires an overall**(M*N^3)** computations

Let us focus on the faster approach.
First, for each solution **p** we calculate two entities:

1) **domination count n_p** , the number of solutions which dominate the solution , and
2) **s_p**-a set of solutions that the solution p dominates.

All solutions in the first non dominated front will have their n_p domination count as zero. Then , we visit each member q of its set s_p and reduce its domination
count by one. In doing so, if for any member q the domination count becomes zero, we put it in a separate list Q , and the third front is identified. This process continues until all fronts are identified.

For each solution p in the second or higher level of non domination, the domination count n_p can be at most **N-1**. Thus, each solution p in the second will be visited at most N-1 times before its domination count becomes zero. At this point, the solution is assigned a non domination level and will never be visited again. Since there are at most N-1 such solutions, the total complexity is **N^2**. and the total complexity of the procedure is **M*N^2**

3.2.FAST NON-DOMINATED SORT (P):

For each p Є P
Sp = Φ
$n_p = 0$
For each q Є P
    if (p<q) then

Sp = Sp U {q}
else if (q<p) then
$n_p = n_p + 1$
if $n_p = 0$ then
$p_{rank} = 1$
$F_1 = F_1 \cup \{p\}$
i =1
while F_i =! Φ
Q = Φ
for each p Є F_i
for each q Є Sp
$n_q = n_q - 1$
if $n_q = 0$ then
$q_{rank} = i + 1$
Q = Q U {q}
i = i + 1
F_i = Q

3.3.DIVERSITY PRESERVATION:

Along with convergence to the pareto optimal set, it is desired that an evolutionary algorithm should maintain a good spread of solutions. The original NSGA used the well known sharing function approach, in which a sharing parameter, which sets the extent of sharing desired. However with this method there was great dependence on the value of sharing parameter chosen and the computation was quite complex. In proposed NSGA-II the crowded comparison approach is used which eliminates the above difficulties to some extent.

3.3.1.DENSITY ESTIMATION:

To get an estimate of density of solution surrounding a particular solution in the population, the average distance of two pints on either side of this point along each objectives are calculated. This crowding distance computation (l distance) requires sorting the population according to each objective function value in ascending order of magnitude. Then for each objective function, boundary solution are assigned a distance value equal to absolute normalized difference of two adjacent solutions. The individual distance value are summed to get the overall crowding distance value(objective function is to be normalized before calculating).

Crowding-distance-assignment(I)
l = $|I|$
for each I, set $I[i]_{distance} = 0$
for each objective in
I = sort (I,m)
$I[l]_{distance} = I[l]_{distance}$ = infinite
for i = 2 to (l-1)
$I[i]_{distance} = I[l]_{distance} + (I[i+1].m - I[i-1].m)/(f_{max} - f_{min})$
This algorithm gives the crowding distance computation procedure of all solution in non-dominated set [I]. m-refers to mth objective value of ith individual in set I. f_{max} and f_{min} are

maximum and minimum value of mth objective function. Since M independent sorting of at most N solution are involved it has O(MN log N) computational complexity(when all population member are in one front). After all population members in set I are assigned distance metric, a solution having a smaller value is considered to be more crowded by other solutions.

3.3.2.CROWDED COMPARISON OPERATOR:

The crowded comparison operator($<_n$) guides the selection process at various stages of selection process towards a uniformly spread out pareto optimal front. Assume that every has two attributes:
1-non-dominated rank (i_{rank})
2-crowding distance ($i_{distance}$)
A partial order $<_n$ is defined as
$i <_n j$ if ($i_{rank} < j_{rank}$)
or ($i_{rank} = j_{rank}$)
and ($i_{distance} < j_{distanc}$)
The solution with lower rank is preferred or if both belong to the same front, the solutions located in lesser crowded region is preferred.

3.3.3.MAIN LOOP:

Initially a random parent population P_0 is created basing on non domination. Each solution is assigned a fitness equal to its non domination level. The minimization of fitness is assumed. The offspring population Q_0 of size N is created using binary tournament selection, recombination and mutation operators. Elitism is introduced by comparing the current population with the previously found best non dominated solution. The algorithm for lth generation is described below.
$R_t = P_t \cup Q_t$
F= fast non dominate sort(R_t)
$P_{t+1} = \Phi$ and i= 1
Until $|Pt+1| + |Fi| <= N$
    Crowding distance assignment(F_i)
    $P_{t+1} = P_{t+1} \cup F_i$
    i = i + 1
Sort($F_i, <_n$)
$P_{t+1} = P_{t+1} \cup F_i[1:(N-|Pt+1|)]$
Q_{t-1} = make new pop(P_{t+1})
t = t+1

First a combined population Rt = Pt U Qt is formed. The population Rt is of size 2N. Rt is sorted according to non domination. The solution belonging to best non dominated set F1 must be emphasized more than in any other solution in combined population. If size of F1 is smaller than N, then all member of set F1 are chosen for new population Pt+1. Remaining members of Pt+1 are chosen from subsequent non dominated fronts in order to their ranking. Thus solutions from set F2 are chosen are next followed by F3 and so on. To choose exactly N population member the solution of last front are sorted using crowed comparison operator <n in descending order and choose the best solution needed to fill population slots.

Chapter 4

SIMULATION RESULTS

4. SIMULATION RESULTS:

In this section we describe the test problem used to find the performance of NSGA-II. We have considered the following combination for optimization in our simulation.

1. f1(x) = x^2.
   f2(x) = (x-2)^2.
   One variable x is taken lying within [-10^3, 10^3]

2. f1(x) = 1-exp[-(x1- $1/\sqrt{3}$)^2 – (x2 -$1/\sqrt{3}$)^2]
   f2(x) = 1-exp[-(x1 + $1/\sqrt{3}$)^2 – (x2 + $1/\sqrt{3}$)^2]
   Two variables are taken lying within [-4,4]

3. f1(x) = x1
   f2(x) = cstr1(x) * [1-$\sqrt{x1/g(x)}$]
   cstr1(x) = 1+9*x2
   Two variables are taken lying within [0,1] in addition one constraint cstr1(x) is also taken.

4. f1(x) = (x1 - 2) ^{2} +(x2 – 1)^2 + 2
   f2(x) = 9 * x1 – (x2 – 1)^2
   cstr1(x) = x1^2 + x2^2 < 225
   cstr2(x) = x1 – 3*x2 <= -10
   Two variables are taken lying within [-20,20], in addition two constraints are also taken.

The following are the important parameter considered and the limits taken in the program.

4.1.PARAMETER:

1. Probability of mutation(Pm)
2. Probability of crossover(Pcr)
3. Population size
4. No of generations(gen)
5. Chromosome lenth(chromelen)

Variables are binary coded. Both cases of simple and uniform crossover have been considered.

4.2.RESULTS:

The optimal solution set is found. The pareto front found(as shown in the figures) is almost continuous.

4.3.DISCUSSION:

Various plots are studied for the objectives obtained after varying some of the parameters of the given objectives.

Fig 1.1-1.6- These figures show the pareto optimal front of the first set of functions. It is seen that the pareto optimal front is convex (as expected). Although the pareto optimal front is convex for all the simulation results the convexity, range and density of the front varies along with the variation in different parameters such as i.e. probability of mutation(Pm), probability of crossover(Pcr), population size, number of generations (gen) etc.

Fig 2.1-2.2-These figures show that the pareto optimal front of the second set of functions is non convex in nature. The optimal front changes with the change in parameters as in above.

Fig 3-This figure shows that the pareto optimal front of the third set of functions is convex but disconnected.

Fig 4.1-4.5-These figures show that the pareto optimal front of the fourth set of functions is non convex in nature. Here along with the two functions two constraints are also taken which also contribute to the nature of the pareto front. It is also seen that the range and density of the front varies along with the variation in different parameters such as i.e. probability of mutation(Pm), probability of crossover(Pcr), population size, number of generations (gen) etc.

SIMULATION 1.1:

```
GA PARAMETERS
-----------------------------------------------------------
Population Size ->200
Chromosome Length ->8
No. of generations ->200
No. of Functions ->2
No. of Constraints ->0
No. of binary-coded variables ->1
No. of real-coded variables ->1
Selection Strategy is Tournament Selection
Binary-coded variable No.-> 0
No. of bits assigned to it ->8
Lower limits on 0th variable-> -100.000000
Upper limits on 0th variable ->100.000000
Real-coded variable No.-> 0
Lower limits on 0th variable-> -100.000000
Upper limits on 0th variable ->100.000000
Variable bounds are rigid
X-over on binary strings is UNIFORM X-OVER
Crossover parameter in the SBX operator is 20.000000
Cross-over Probability ->0.800000
Mutation Probability for binary strings -> 0.000000
Mutation Probability for real-coded vectors -> 0.020000
Random Seed ->0.000000
```

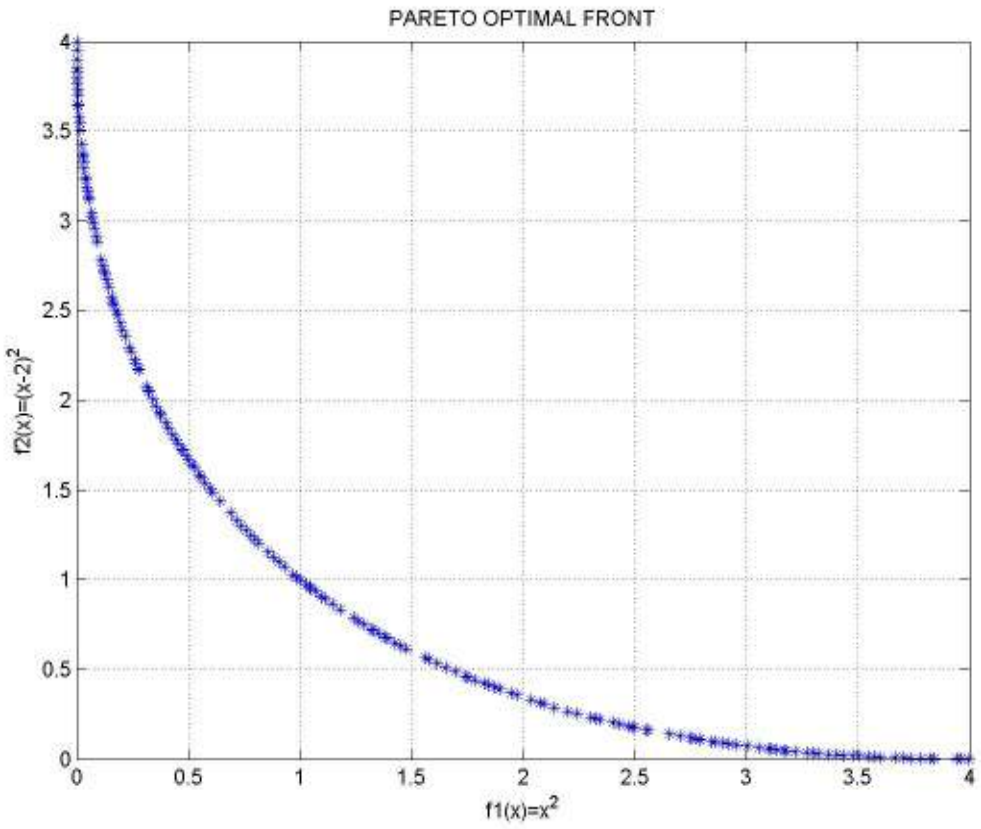

Fig 1.1

SIMULATION 1.2:

```
GA PARAMETERS
-----------------------------------------------------------
Population Size ->300
Chromosome Length ->8
No. of generations ->300
No. of Functions ->2
No. of Constraints ->0
No. of binary-coded variables ->1
No. of real-coded variables ->1
Selection Strategy is Tournament Selection
Binary-coded variable No.-> 0
No. of bits assigned to it ->8
Lower limits on 0th variable-> -100.000000
Upper limits on 0th variable ->100.000000
Real-coded variable No.-> 0
Lower limits on 0th variable-> -100.000000
Upper limits on 0th variable ->100.000000
Variable bounds are rigid
X-over on binary strings is UNIFORM X-OVER
Crossover parameter in the SBX operator is 30.000000
Cross-over Probability ->0.700000
Mutation Probability for binary strings -> 0.080000
Mutation Probability for real-coded vectors -> 0.300000
Random Seed ->0.000000
```

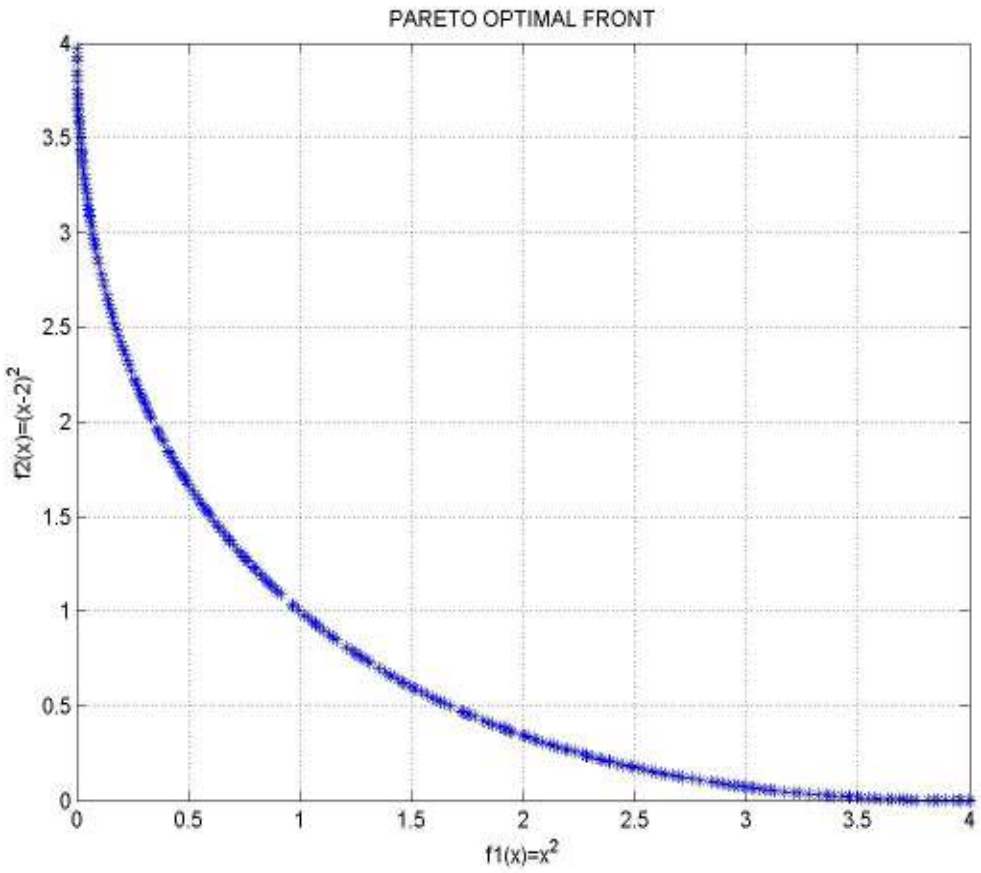

Fig 1.2

SIMULATION 1.3:

```
GA PARAMETERS
-----------------------------------------------------------
Population Size ->400
Chromosome Length ->8
No. of generations ->300
No. of Functions ->2
No. of Constraints ->0
No. of binary-coded variables ->1
No. of real-coded variables ->1
Selection Strategy is Tournament Selection
Binary-coded variable No.-> 0
No. of bits assigned to it ->8
Lower limits on 0th variable-> -100.000000
Upper limits on 0th variable ->100.000000
Real-coded variable No.-> 0
Lower limits on 0th variable-> -100.000000
Upper limits on 0th variable ->100.000000
Variable bounds are rigid
X-over on binary strings is UNIFORM X-OVER
Crossover parameter in the SBX operator is 20.000000
Cross-over Probability ->0.800000
Mutation Probability for binary strings -> 0.900000
Mutation Probability for real-coded vectors -> 0.600000
Random Seed ->0.000000
```

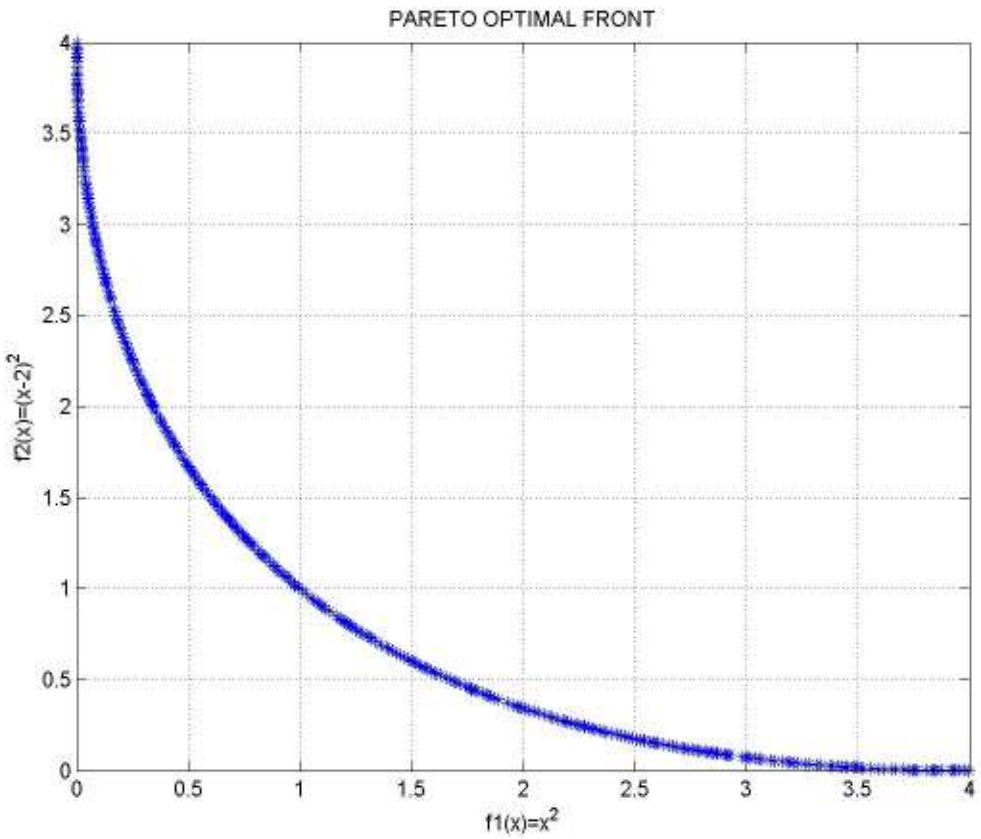

Fig1.3

SIMULATION 1.4:

```
GA PARAMETERS
-----------------------------------------------------------
Population Size ->400
Chromosome Length ->8
No. of generations ->400
No. of Functions ->2
No. of Constraints ->0
No. of binary-coded variables ->1
No. of real-coded variables ->1
Selection Strategy is Tournament Selection
Binary-coded variable No.-> 0
No. of bits assigned to it ->8
Lower limits on 0th variable-> -100.000000
Upper limits on 0th variable ->100.000000
Real-coded variable No.-> 0
Lower limits on 0th variable-> -100.000000
Upper limits on 0th variable ->100.000000
Variable bounds are rigid
X-over on binary strings is UNIFORM X-OVER
Crossover parameter in the SBX operator is 15.000000
Cross-over Probability ->0.800000
Mutation Probability for binary strings -> 0.100000
Mutation Probability for real-coded vectors -> 0.500000
Random Seed ->0.000000
```

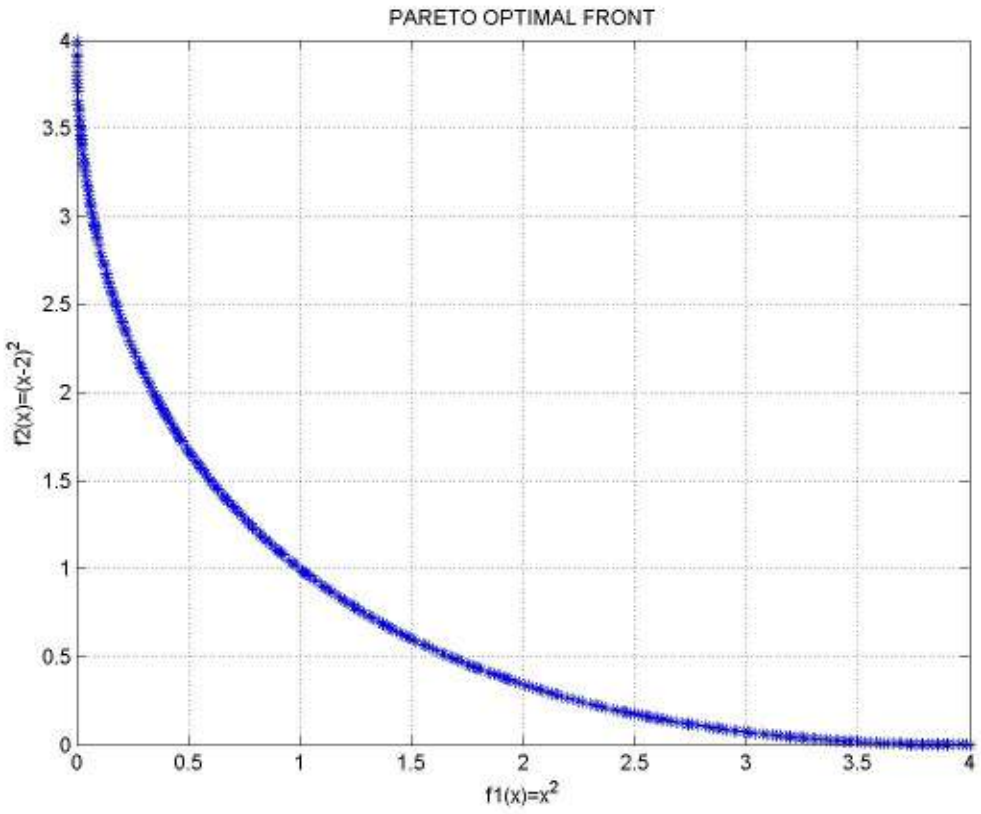

Fig 1.4

SIMULATION 1.5:

```
GA PARAMETERS
-----------------------------------------------------------
Population Size ->500
Chromosome Length ->8
No. of generations ->500
No. of Functions ->2
No. of Constraints ->0
No. of binary-coded variables ->1
No. of real-coded variables ->2
Selection Strategy is Tournament Selection
Binary-coded variable No.-> 0
No. of bits assigned to it ->8
Lower limits on 0th variable-> -100.000000
Upper limits on 0th variable ->100.000000
Real-coded variable No.-> 0
Lower limits on 0th variable-> -100.000000
Upper limits on 0th variable ->100.000000
Variable bounds are rigid
Real-coded variable No.-> 1
Lower limits on 1th variable-> -100.000000
Upper limits on 1th variable ->100.000000
Variable bounds are rigid
X-over on binary string is SINGLE POINT X-OVER
Crossover parameter in the SBX operator is 30.000000
Cross-over Probability ->0.600000
Mutation Probability for binary strings -> 0.100000
Mutation Probability for real-coded vectors -> 0.500000
Random Seed ->0.000000
```

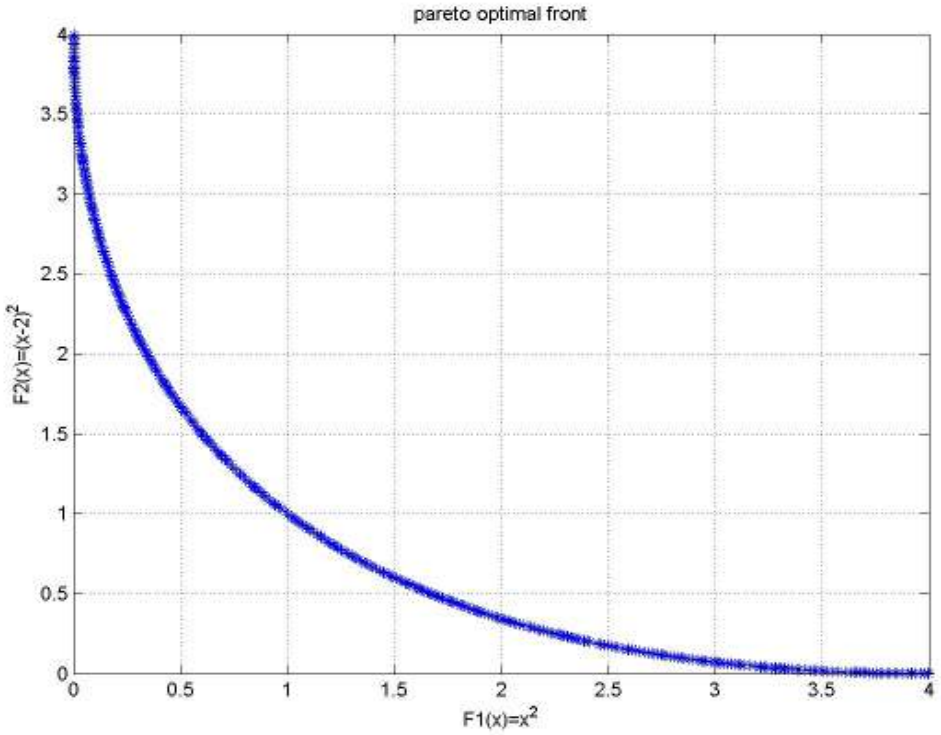

Fig 1.5

SIMULATION 1.6:

```
GA PARAMETERS
-----------------------------------------------------------
Population Size ->500
Chromosome Length ->8
No. of generations ->500
No. of Functions ->2
No. of Constraints ->0
No. of binary-coded variables ->1
No. of real-coded variables ->1
Selection Strategy is Tournament Selection
Binary-coded variable No.-> 0
No. of bits assigned to it ->8
Lower limits on 0th variable-> -100.000000
Upper limits on 0th variable ->100.000000
Real-coded variable No.-> 0
Lower limits on 0th variable-> -100.000000
Upper limits on 0th variable ->100.000000
Variable bounds are rigid
X-over on binary string is SINGLE POINT X-OVER
Crossover parameter in the SBX operator is 10.000000
Cross-over Probability ->0.900000
Mutation Probability for binary strings -> 0.050000
Mutation Probability for real-coded vectors -> 0.150000
Random Seed -0.000000
```

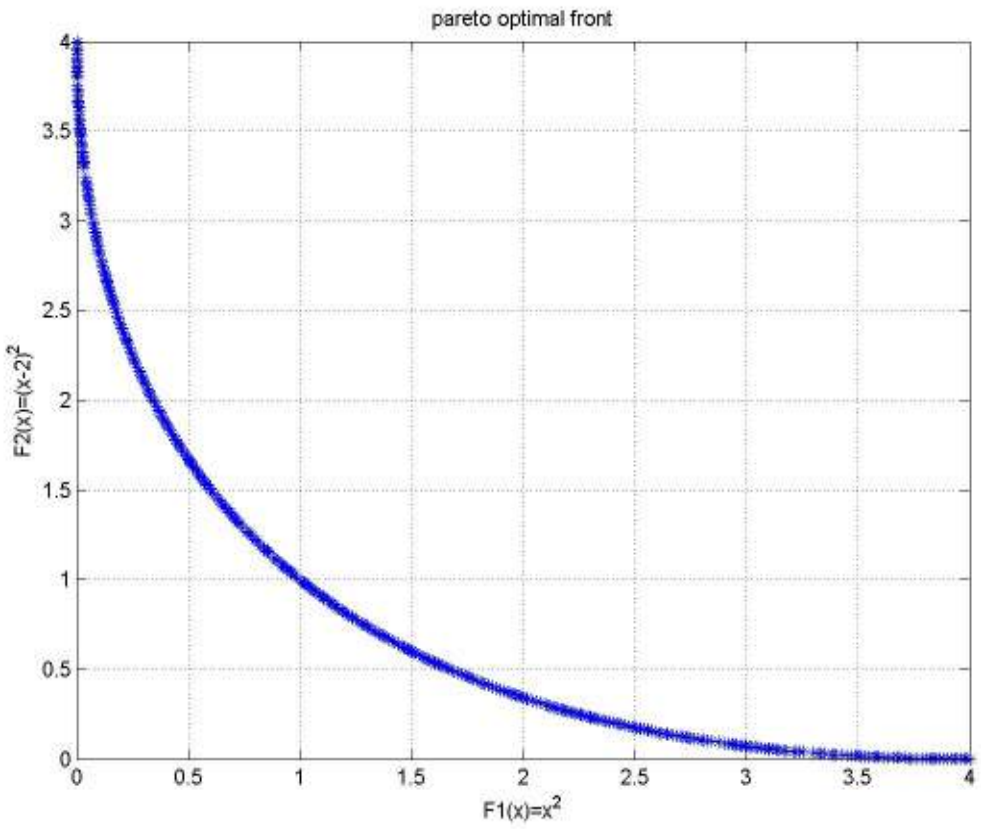

Fig 1.6

SIMULATION 2.1:

```
GA PARAMETERS
-----------------------------------------------------------
Population Size ->500
Chromosome Length ->16
No. of generations ->500
No. of Functions ->2
No. of Constraints ->0
No. of binary-coded variables ->2
No. of real-coded variables ->2
Selection Strategy is Tournament Selection
Binary-coded variable No.-> 0
No. of bits assigned to it ->8
Lower limits on 0th variable-> -4.000000
Upper limits on 0th variable ->4.000000
Binary-coded variable No.-> 1
No. of bits assigned to it ->8
Lower limits on 1th variable-> -4.000000
Upper limits on 1th variable ->4.000000
Real-coded variable No.-> 0
Lower limits on 0th variable-> -4.000000
Upper limits on 0th variable ->4.000000
Variable bounds are rigid
Real-coded variable No.-> 1
Lower limits on 1th variable-> -4.000000
Upper limits on 1th variable ->4.000000
Variable bounds are rigid
X-over on binary string is SINGLE POINT X-OVER
Crossover parameter in the SBX operator is 1.000000
Cross-over Probability ->0.800000
Mutation Probability for binary strings -> 0.020000
Mutation Probability for real-coded vectors -> 0.010000
Random Seed ->1.000000
```

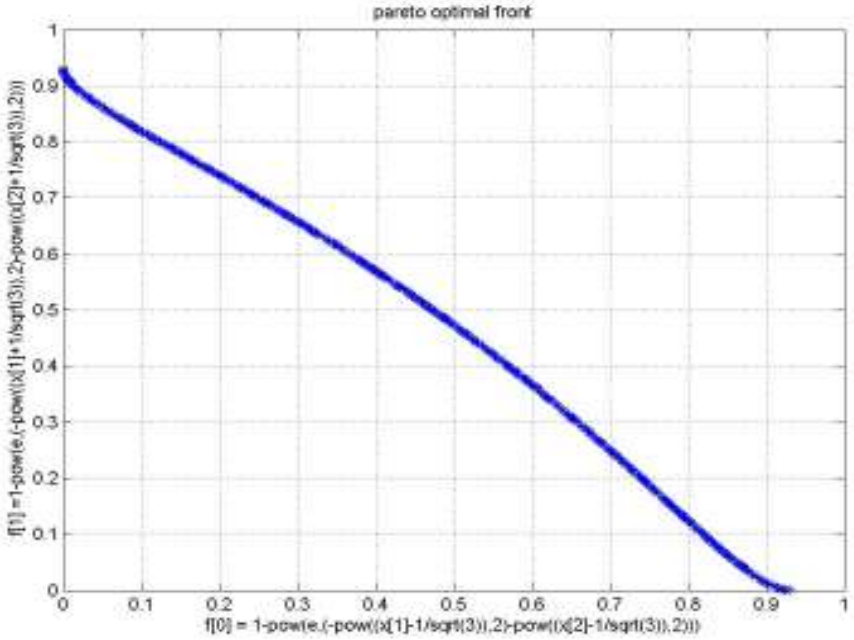

Fig 2.1

SIMULATION 2.2:

```
GA PARAMETERS
-----------------------------------------------------------
Population Size ->500
Chromosome Length ->16
No. of generations ->200
No. of Functions ->2
No. of Constraints ->0
No. of binary-coded variables ->2
No. of real-coded variables ->2
Selection Strategy is Tournament Selection
Binary-coded variable No.-> 0
No. of bits assigned to it ->8
Lower limits on 0th variable-> -4.000000
Upper limits on 0th variable ->4.000000
Binary-coded variable No.-> 1
No. of bits assigned to it ->8
Lower limits on 1th variable-> -4.000000
Upper limits on 1th variable ->4.000000
Real-coded variable No.-> 0
Lower limits on 0th variable-> -4.000000
Upper limits on 0th variable ->4.000000
Variable bounds are rigid
Real-coded variable No.-> 1
Lower limits on 1th variable-> -4.000000
Upper limits on 1th variable ->4.000000
Variable bounds are rigid
X-over on binary string is SINGLE POINT X-OVER
Crossover parameter in the SBX operator is 5.000000
Cross-over Probability ->0.850000
Mutation Probability for binary strings -> 0.012500
Mutation Probability for real-coded vectors -> 0.150000
Random Seed ->1.000000
```

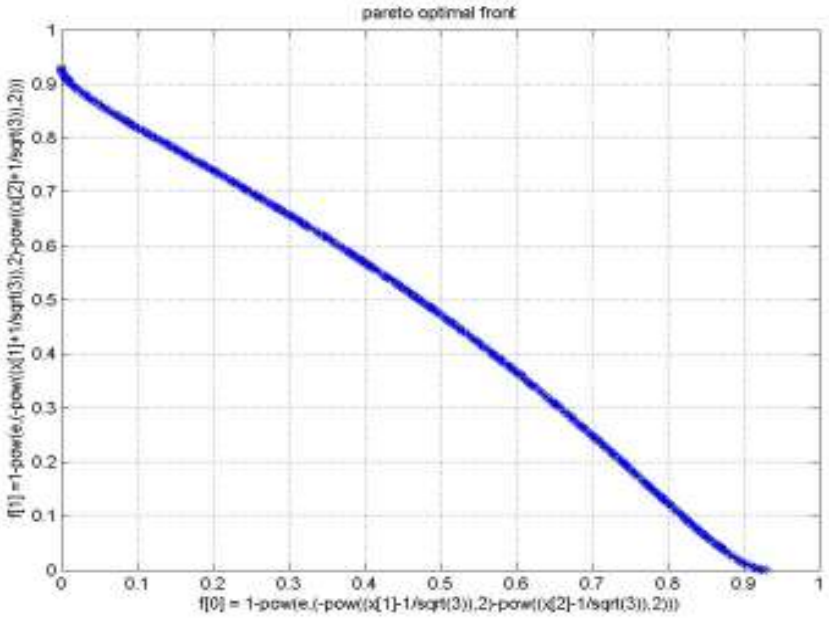

Fig 2.2

SIMULATION 3:

```
GA PARAMETERS
-----------------------------------------------------------
Population Size ->500
Chromosome Length ->16
No. of generations ->500
No. of Functions ->2
No. of Constraints ->1
No. of binary-coded variables ->2
No. of real-coded variables ->2
Selection Strategy is Tournament Selection
Binary-coded variable No.-> 0
No. of bits assigned to it ->8
Lower limits on 0th variable-> 0.000000
Upper limits on 0th variable ->1.000000
Binary-coded variable No.-> 1
No. of bits assigned to it ->8
Lower limits on 1th variable-> 0.000000
Upper limits on 1th variable ->1.000000
Real-coded variable No.-> 0
Lower limits on 0th variable-> 0.000000
Upper limits on 0th variable ->1.000000
Variable bounds are rigid
Real-coded variable No.-> 1
Lower limits on 1th variable-> 0.000000
Upper limits on 1th variable ->1.000000
Variable bounds are rigid
X-over on binary string is SINGLE POINT X-OVER
Crossover parameter in the SBX operator is 20.000000
Cross-over Probability ->0.800000
Mutation Probability for binary strings -> 0.010000
Mutation Probability for real-coded vectors -> 0.100000
Random Seed ->1.000000
```

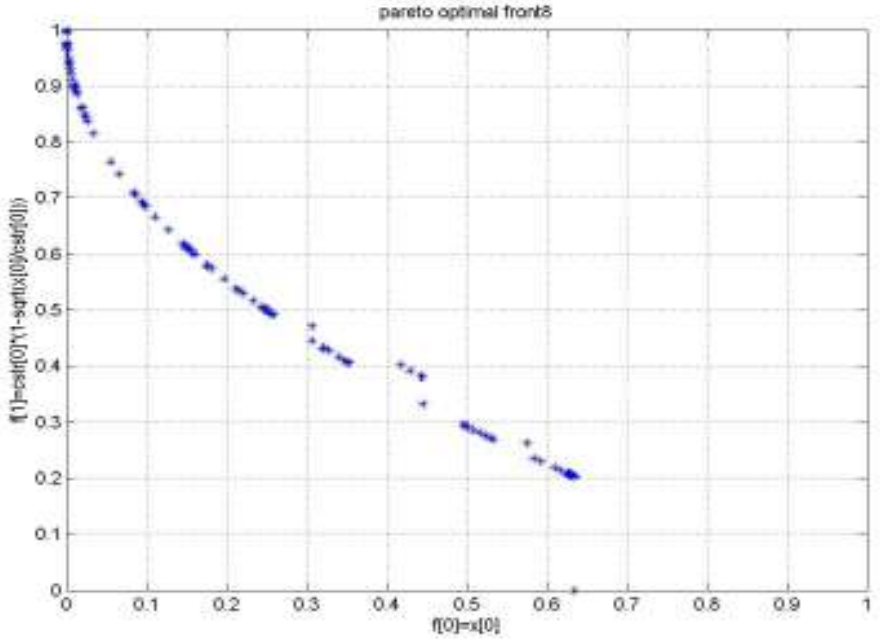

Fig 3

SIMULATION 4.1:

```
GA PARAMETERS
-----------------------------------------------------------
Population Size ->200
Chromosome Length ->16
No. of generations ->200
No. of Functions ->2
No. of Constraints ->2
No. of binary-coded variables ->2
No. of real-coded variables ->2
Selection Strategy is Tournament Selection
Binary-coded variable No.-> 0
No. of bits assigned to it ->8
Lower limits on 0th variable-> -20.000000
Upper limits on 0th variable ->20.000000
Binary-coded variable No.-> 1
No. of bits assigned to it ->8
Lower limits on 1th variable-> -20.000000
Upper limits on 1th variable ->20.000000
Real-coded variable No.-> 0
Lower limits on 0th variable-> -20.000000
Upper limits on 0th variable ->20.000000
Variable bounds are rigid
Real-coded variable No.-> 1
Lower limits on 1th variable-> -20.000000
Upper limits on 1th variable ->20.000000
Variable bounds are rigid
X-over on binary strings is UNIFORM X-OVER
Crossover parameter in the SBX operator is 30.000000
Cross-over Probability ->0.700000
Mutation Probability for binary strings -> 0.010000
Mutation Probability for real-coded vectors -> 0.200000
Random Seed ->0.000000
```

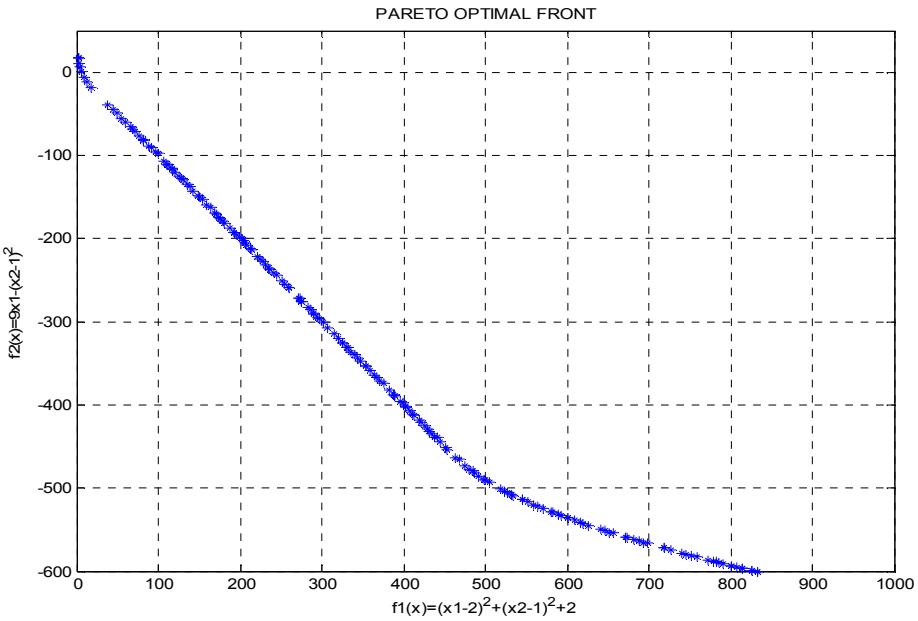

Fig 4.1

SIMULATION 4.2:

```
GA PARAMETERS
-----------------------------------------------------------
Population Size ->250
Chromosome Length ->16
No. of generations ->250
No. of Functions ->2
No. of Constraints ->2
No. of binary-coded variables ->2
No. of real-coded variables ->2
Selection Strategy is Tournament Selection
Binary-coded variable No.-> 0
No. of bits assigned to it ->8
Lower limits on 0th variable-> -20.000000
Upper limits on 0th variable ->20.000000
Binary-coded variable No.-> 1
No. of bits assigned to it ->8
Lower limits on 1th variable-> -20.000000
Upper limits on 1th variable ->20.000000
Real-coded variable No.-> 0
Lower limits on 0th variable-> -20.000000
Upper limits on 0th variable ->20.000000
Variable bounds are rigid
Real-coded variable No.-> 1
Lower limits on 1th variable-> -20.000000
Upper limits on 1th variable ->20.000000
Variable bounds are rigid
X-over on binary strings is UNIFORM X-OVER
Crossover parameter in the SBX operator is 20.000000
Cross-over Probability ->0.600000
Mutation Probability for binary strings -> 0.010000
Mutation Probability for real-coded vectors -> 0.200000
Random Seed ->0.000000
```

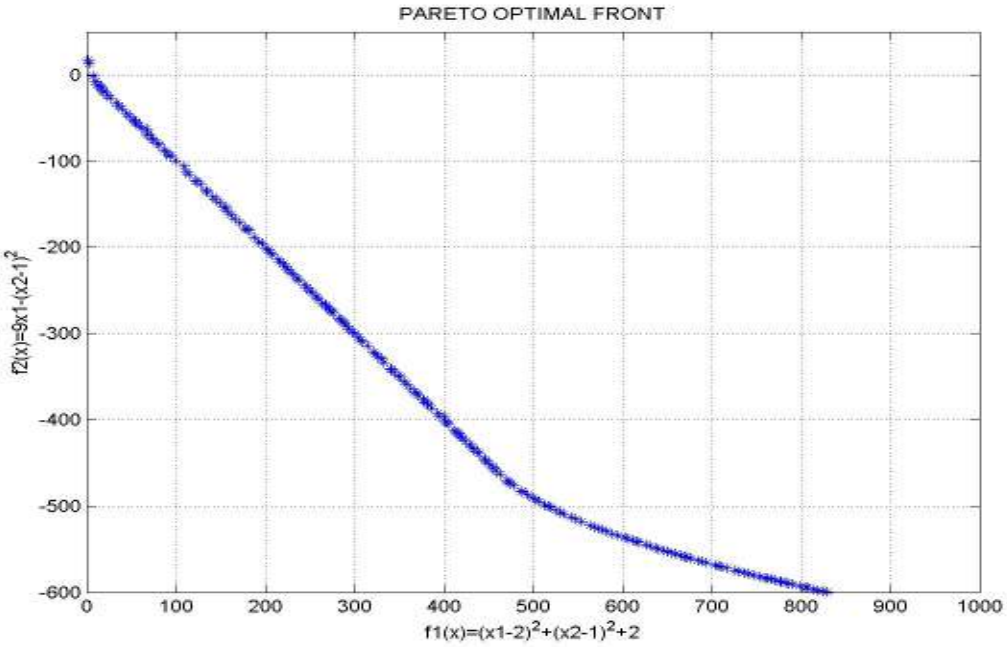

Fig 4.2

SIMULATION 4.3:

```
GA PARAMETERS
-----------------------------------------------------------
Population Size ->400
Chromosome Length ->16
No. of generations ->400
No. of Functions ->2
No. of Constraints ->2
No. of binary-coded variables ->2
No. of real-coded variables ->2
Selection Strategy is Tournament Selection
Binary-coded variable No.-> 0
No. of bits assigned to it ->8
Lower limits on 0th variable-> -20.000000
Upper limits on 0th variable ->20.000000
Binary-coded variable No.-> 1
No. of bits assigned to it ->8
Lower limits on 1th variable-> -20.000000
Upper limits on 1th variable ->20.000000
Real-coded variable No.-> 0
Lower limits on 0th variable-> -20.000000
Upper limits on 0th variable ->20.000000
Variable bounds are rigid
Real-coded variable No.-> 1
Lower limits on 1th variable-> -20.000000
Upper limits on 1th variable ->20.000000
Variable bounds are rigid
X-over on binary string is SINGLE POINT X-OVER
Crossover parameter in the SBX operator is 25.000000
Cross-over Probability ->0.600000
Mutation Probability for binary strings -> 0.010000
Mutation Probability for real-coded vectors -> 0.100000
Random Seed ->0.000000
```

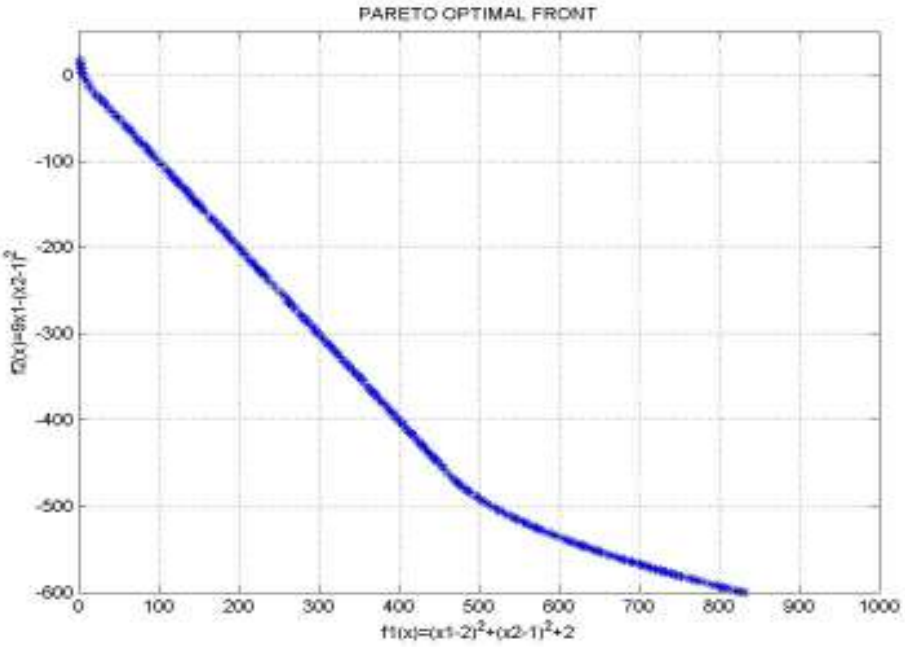

Fig 4.3

SIMULATION 4.4:

```
GA PARAMETERS
-----------------------------------------------------------
Population Size ->400
Chromosome Length ->16
No. of generations ->400
No. of Functions ->2
No. of Constraints ->2
No. of binary-coded variables ->2
No. of real-coded variables ->2
Selection Strategy is Tournament Selection
Binary-coded variable No.-> 0
No. of bits assigned to it ->8
Lower limits on 0th variable-> -20.000000
Upper limits on 0th variable ->20.000000
Binary-coded variable No.-> 1
No. of bits assigned to it ->8
Lower limits on 1th variable-> -20.000000
Upper limits on 1th variable ->20.000000
Real-coded variable No.-> 0
Lower limits on 0th variable-> -20.000000
Upper limits on 0th variable ->20.000000
Variable bounds are rigid
Real-coded variable No.-> 1
Lower limits on 1th variable-> -20.000000
Upper limits on 1th variable ->20.000000
Variable bounds are rigid
X-over on binary string is SINGLE POINT X-OVER
Crossover parameter in the SBX operator is 30.000000
Cross-over Probability ->0.700000
Mutation Probability for binary strings -> 0.020000
Mutation Probability for real-coded vectors -> 0.200000
Random Seed ->0.000000
```

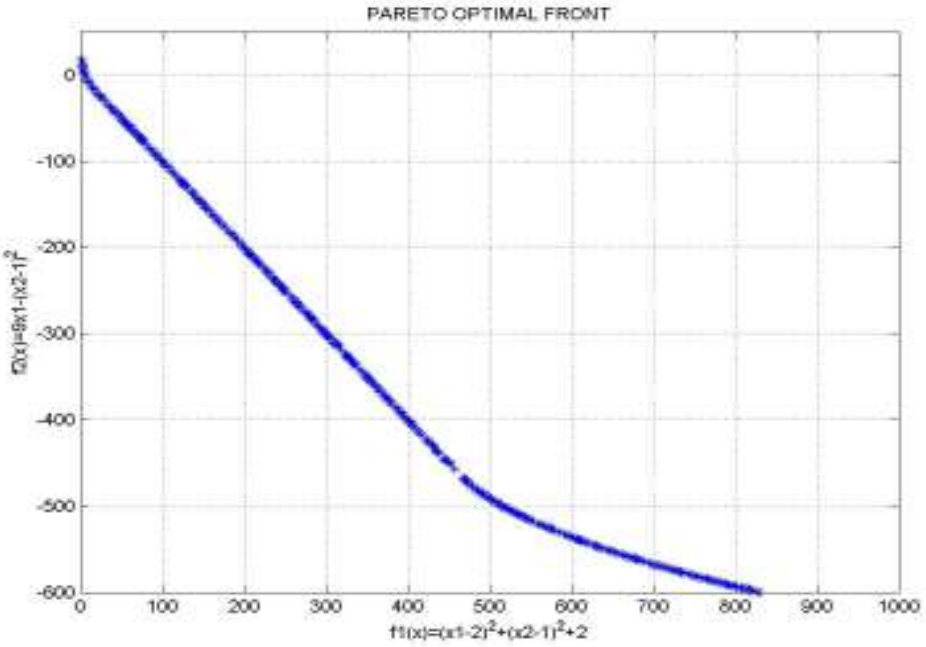

Fig 4.4

SIMULATION 4.5:

```
GA PARAMETERS
-----------------------------------------------------------
Population Size ->200
Chromosome Length ->16
No. of generations ->200
No. of Functions ->2
No. of Constraints ->2
No. of binary-coded variables ->2
No. of real-coded variables ->2
Selection Strategy is Tournament Selection
Binary-coded variable No.-> 0
No. of bits assigned to it ->8
Lower limits on 0th variable-> -20.000000
Upper limits on 0th variable ->20.000000
Binary-coded variable No.-> 1
No. of bits assigned to it ->8
Lower limits on 1th variable-> -20.000000
Upper limits on 1th variable ->20.000000
Real-coded variable No.-> 0
Lower limits on 0th variable-> -20.000000
Upper limits on 0th variable ->20.000000
Variable bounds are rigid
Real-coded variable No.-> 1
Lower limits on 1th variable-> -20.000000
Upper limits on 1th variable ->20.000000
Variable bounds are rigid
X-over on binary string is SINGLE POINT X-OVER
Crossover parameter in the SBX operator is 10.000000
Cross-over Probability ->0.900000
Mutation Probability for binary strings -> 0.010000
Mutation Probability for real-coded vectors -> 0.010000
Random Seed ->0.000000
```

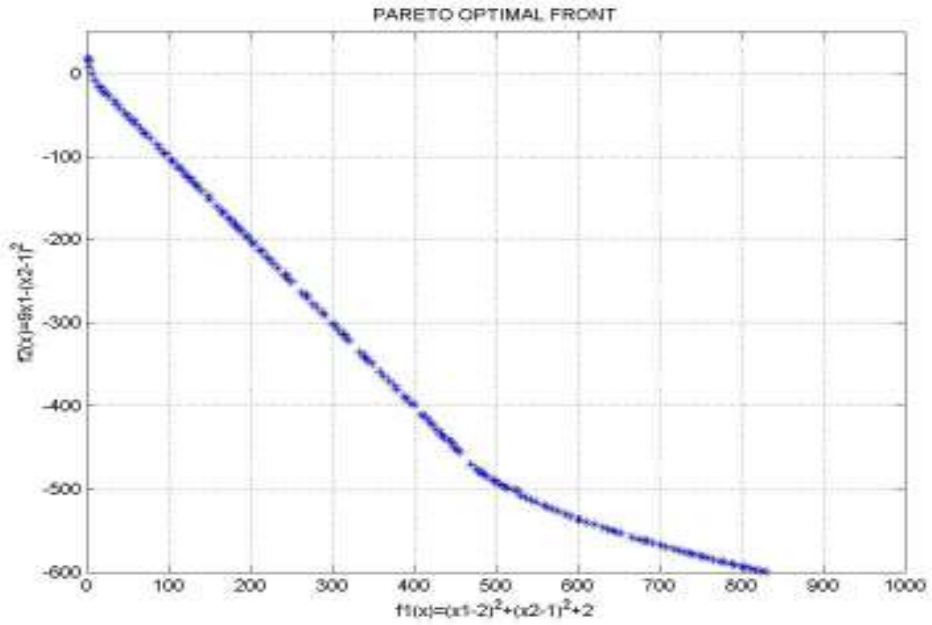

Fig 4.5

Chapter 5

CONCLUSION

5.CONCLUSION:

Multi objective evolutionary algorithms (MOEAs) that use non dominated sorting and sharing have been criticized mainly for their :

1. computational complexity (M*N^3) (where M the no of objectives and N is the no of solutions)
2. non elitism approach
3. need for specifying a sharing parameter.

So non dominated sorting based multi objective evolutionary algorithm (MOEA),called fast non dominated sorting genetic algorithm(NSGA-II), which alleviates all the above three difficulties was suggested. The specialty of this algorithm is that it employs a faster sorting approach than other available algorithms. Simulation on certain standard problems show that NSGA-II has better spread of solutions and better convergence near true pareto front as compared to other algorithms. Moreover, the definition of dominance in order to solve constraint multi objectives has been efficiently modified in this algorithm. The test results all so that NSGA-II has a better constraint handling approach than any other method.

REFERENCES:

1. Goldberg David E., Genetic Algorithm in Search, Optimization and machine learning, Addison Wesley,1989.

2. Deb Kalyanmoy, Associate Member, IEEE, Pratap Amrit, Agarwal Sameer, and. Meyarivan T, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II", IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 6, NO. 2, APRIL 2002, Page(182-197)

3. Deb. K, Multi objective Optimization Using Evolutionary algorithms. Chichester, U.K.: Wiley, 2001.

4. Deb K. and Goldberg D. E., "An investigation of niche and species formation in genetic function optimization", in Proceedings of the Third International Conference on Genetic Algorithms, J. D. Schaffer, Ed. San Mateo, CA: Morgan Kauffman, 1989, pp. 42–50

5. Deb K. and Agrawal S., "Understanding interactions among genetic algorithm parameters," in Foundations of Genetic Algorithms *V*, W. Banzhaf and C. Reeves, Eds. San Mateo, CA: Morgan Kauffman, 1998, pp. 265–286.

6. Deb K. and Agrawal R. B., "Simulated binary crossover for continuous search space," in Complex Syst., Apr. 1995, vol. 9, pp. 115–148.

7. Goldberg D. E., Deb K., Kargupta H., and Harik G., "Rapid, accurate optimization of difficult problems using fast messy genetic algorithms," in Proc. 5th Int. Conf. on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann, 1993, pp. 56–64.