Optimizing fuzzy multi-objective problems using fuzzy genetic algorithms, FZDT test functions

Vikash kumar[1], D. Chakraborty[1]
[1]Department of Mathematics
Indian Institute of Technology, Kharagpur (W.B) 721 302, India.

Abstract:
The following work outlines a robust method for accounting the fuzziness of the objective space while dealing with the real world optimization problems. Use of mean/approximated value of input parameters doesn't account for the variability in the optimized solution inherited due to variability in the input parameters which is very crucial, especially in real world problems. The following work describes and evaluates a unique solution strategy for optimizing fuzzy multi-objective problems by integrating genetic algorithms with concepts of fuzzy logic. The unique way of problem formulation required no tweaking in genetic operators of mutation and crossover but the concept of ranking has been carefully extended to fuzzy domain. The standard benchmark test function, ZDT [4], have been extrapolated to fuzzy domain as FZDT and proposed to be benchmark test function for fuzzy optimization algorithms. The results have been successfully verified with FZDT test functions and were found coherent with ZDT test functions under classical assumptions.

Keywords:
Fuzzy optimization, Fuzzy multi-objective Optimization, Fuzzy Genetic Algorithms, Evolutionary Algorithms, Fuzzy test functions (FZDT test functions).

Introduction:

Multi objective optimization problem is the process of simultaneously optimizing two or more conflicting objectives subject to certain constraints. Such problems can be found in various fields: product and process design, finance, aircraft design, the oil and gas industry, automobile design, or wherever optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover and is the most commonly used search techniques in computing to find exact or approximate solutions to such optimization and search problems.

In real world problems, parameters of a process are never precisely fixed to a definite value. Transients, noise, measurement errors, Instrument's least count etc makes it even more difficult to know their exact value at any time stamp. Even if externally regulated, parameters have some variability in their values. This variability has been continuously ignored by using mean/approximated/fixed value of the parameters thus losing the precious information about the variability in the final optimized solution.

For example, in an isothermal process, temperature is externally controlled at a certain fixed level. In general, for calculations or optimizations, temperature is taken constant at that specified level. But, there is always variability or fuzziness about the fixed value in such controlled parameters which needs to be preserved and reflected in the final results.

1 Problem Formulation

A multi-objective fuzzy constrained optimization problem can be represented as:

Minimize Objective Functions: $f_i(\mathbf{A},\mathbf{X})$ i = 1, 2,….., I (1)

Subject to Constraints: $g_j(\mathbf{A},\mathbf{X})$ j = 1, 2,….., J (2)

Where $\mathbf{A} = (x_m : m = 1,2,\ldots,M)$ is a M-tuple vector of fuzzy variables,

$\mathbf{X} = (a_n : n = 1,2,\ldots,N)$ is a N-tuple vector of decision variables,

In real world problems, the input parameters of an optimization problem inherit variability due to external factors. Variability of such parameters is being accounted by taking them fuzzy in nature

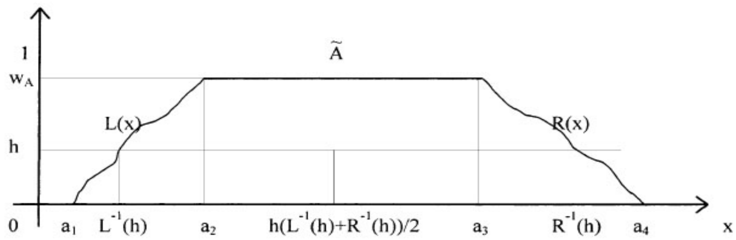
Email: iit.vkash@gmail.com, debjani@maths.iitkgp.ernet.in

about the mean/fixed value. The membership function of such parameters can easily be determined by repetitive observations and by analyzing extensive data set. Decision variable however are assumed to be crisp real numbers. Thus our result accounts and reflects the variability of the plug-in parameters while fixing the decision variable at exact values. Since the optimization problem aims to determine the optimized value of decision variables (with no inherent variability) based on the information concealed within the input parameters (with inherent variability) and the objective space. This completely justifies our assumption of taking decision variables to be crisp real numbers and input parameters as fuzzy numbers.

2 Fuzzy representation

A is a generalized fuzzy number as shown in Fig-1. It is described as any fuzzy subset of the real line R, whose membership function μ_A satisfies the following conditions:

1. $\mu_{\tilde{A}}(x)$ is a continuous mapping from R to the closed interval [0, 1],
2. $\mu_{\tilde{A}}(x) = 0$, -inf<x <a1,
3. $\mu_{\tilde{A}}(x)$ =L(x) is strictly increasing on [a1, a2]
4. $\mu_{\tilde{A}}(x) = w_A$, a2<x <a3,
5. $\mu_{\tilde{A}}(x)$ =R(x) is strictly decreasing on [a3, a4]
6. $\mu_{\tilde{A}}(x) = 0$, a4 <x <inf,

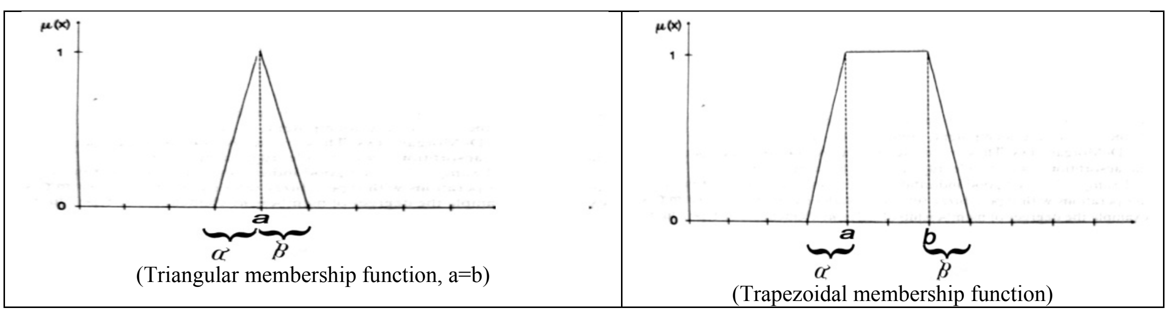


The graded mean *h*-level value of generalized fuzzy number $\tilde{A} = (a_1, a_2, a_3, a_4, w_A)_{LR}$.

Where 0 <w_A<1, and a1, a2, a3, and a4 are real numbers. Also this type of generalized fuzzy number be denoted as A = (a1, a2, a3, a4; $w_A)_{LR}$. When w_A = 1, it can be simplified as A = (a1, a2, a3, $a4)_{LR}$.

We have taken into account two types of fuzzy numbers in our study- one with triangular membership function and other with trapezoidal membership function.

Where, a: the leftmost point where $\mu_{\tilde{A}}(x)$ takes the value 1.
b: the rightmost point where $\mu_{\tilde{A}}(x)$ takes the value 1.
α: the spread of the fuzzy number to the left of a.
β: the spread of the fuzzy number to the right of b.



(Triangular membership function, a=b)

(Trapezoidal membership function)

Fig-2.

3 Solution strategy: Fuzzy genetic algorithm

3.1 Genetic Algorithms (GA)

Genetic Algorithms are probabilistic search algorithms which simulate natural evolution. Genetic algorithms find the individual from the search space with the best "genetic material". The search space (called decision space) of a problem is randomly initialized with a collection of individuals.

Email: iit.vkash@gmail.com, debjani@maths.iitkgp.ernet.in

These individuals are represented by characters, strings or matrices which are often referred to as chromosomes. The quality of an individual is measured with evaluation functions called objective functions. Now, in every iteration, the parent population is subjected to genetic operators of Crossover and Mutation to produce the child population. The combined parent and child population is subjected to rank based population sizing to produce the next generation. The process iterates until the converging criteria is met.

3.2 Fuzzy genetic algorithm (FGA)

FGA follows the basic framework of a genetic algorithm.

BEGINFGA
Generation_count =1
Random initialization of parent population.
WHILE converging criteria = false
    BEGIN
    Select parents from the population.
    Produce children from the selected parents using genetic operators.
    Extend the population adding the child population to parent population.
    Rank the extended population.
    Size the extended population to obtain the next generation.
    Generation_count= Generation_count+1
    END
Output the final population.
END FGA

Fuzzy adaptation of the individual GA steps and other important FGA terminologies has been discussed below.

3.2.1 Fuzzy objective space/ Fussy decision space:

Objective Functions: $f_i(\mathbf{A},\mathbf{X})$ $i = 1, 2,\ldots, I$ (1)

Constraints: $g_j(\mathbf{A},\mathbf{X})$ $j = 1, 2,\ldots, J$ (2)

Where $\mathbf{A} = (x_m : m = 1,2,\ldots,M)$ is a M-tuple vector of fuzzy variables,

$\mathbf{X} = (a_n : n = 1,2,\ldots,N)$ is a N-tuple vector of decision variables,

As per the problem formulation, decision space is a fuzzy space and the fi's and gi's are fuzzy numbers.

3.2.2 Initialization:

First generation of individuals is randomly initialized. Individuals are represented as strings/arrays/matrices of variables called chromosomes. Since decision variables of FGA are crisp real number, the initialization step of FGA is same as that of a normal GA.

3.2.3 Genetic operators – Mutation & Crossover:

Mutation and Crossover operators are used to mix genetic materials, called schema, to build new schema/genetic material and hence drive the evolution. Genetic operators tweak the chromosome sequence of selected parents to produce child chromosomes. Since chromosomes are strings of crisp decision variables, crossover operations(like one point Cx, multi point Cx, hereditary Cx etc) and mutations operations(like …..) in FGA works in the same way as in normal GA.

3.2.4 Fuzzy Ranking:

Rank is a measure of goodness of a solution. The combined population is ranked, following either the Goldberg [21] or the Fonseca [22] approach. In the former, the non-dominated members of the population are taken out of the population as rank 1. The truncated population is again checked for dominance and the non-dominated set is once again removed out of it, this time as rank 2 members,

Email: iit.vkash@gmail.com, debjani@maths.iitkgp.ernet.in

and the procedure continues. In the Fonseca [22] strategy the entire population is checked for dominance and the ranks to the individuals are assigned using the formula:

$$Ri = 1 + Nd$$

Where R_i is the rank of the individual i, and N_d is the number of individuals that dominate it. In this study The Fonseca approach was preferred as it reduces the computational burden.

A objective vector $(\Omega) = (f_i : i = 1, 2, \ldots, I)$

Then the condition for dominance between any two objective vectors can be taken as:

$$(\Omega_l \prec \Omega_m) \Leftrightarrow (\forall_i)(f_{il} \leq f_{im}) \wedge (\exists_m)(f_{il} < f_{im})$$

In other words, if one particular solution is at least as good, or better in terms of all the objective functions, when compared to another solution, and definitely better in terms of at least one objective function, it is considered to be a weakly dominating solution. Where fuzzy comparisons are made using Graded Mean Integration (GMI)

3.2.5 Graded Mean Integration

GMI is a method of comparing two fuzzy numbers. We compare the numbers based on their defuzzified values. The number with higher defuzzified value is larger. The general formula for Graded Mean Integration is given by:

$$P(A) = \left(\int_0^{w_A} h\left(\frac{L^{-1}(h) + R^{-1}(h)}{2}\right)dh \right) / \int_0^{w_A} h\,dh.$$

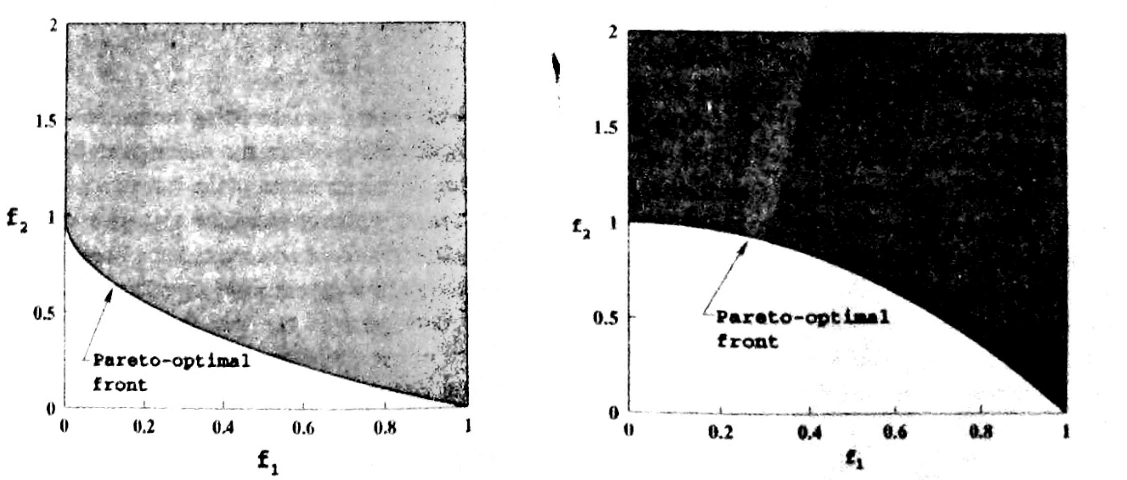
Where L(h) and R(h) are the left and right shape functions, respectively and w_A is the maximum value attained by L(h) and R(h) whereas the minimum value is zero. For a trapezoidal fuzzy number $\tilde{A} = (a,b,\alpha,\beta)$ it reduces to $P(\tilde{A}) = (3a + 3b + \beta - \alpha)/6$

3.2.6 Advanced concepts:

Due to the inherent beauty of problem formulation, the advances concepts of a GA remain valid for a FGA as well. Concepts pertaining to the variable space like Nadir point initialization, normalized random population etc can be directly borrowed from GA. The concepts pertaining to decision space/objective space like Crowding, rank based sizing etc also remains valid after proper fuzzy adaptation.

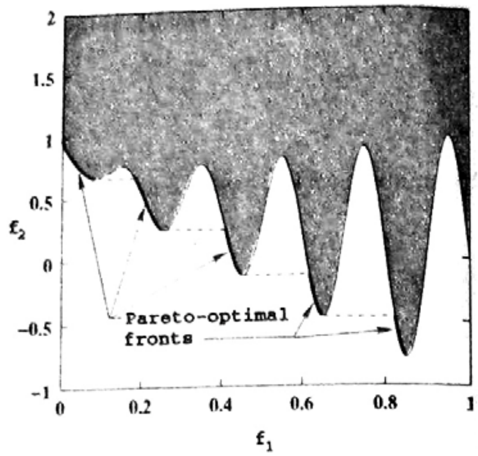
4 Test functions

4.1 ZDT:

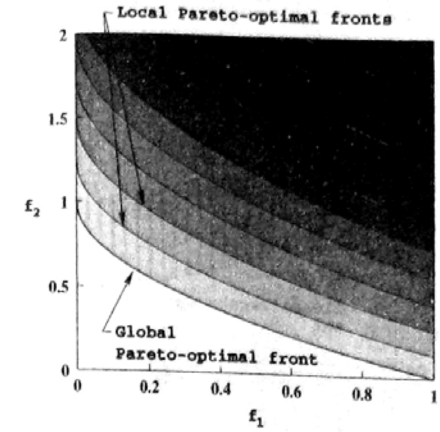


The search space near the Pareto-optimal region for ZD   The search space near the Pareto-optimal region for ZI

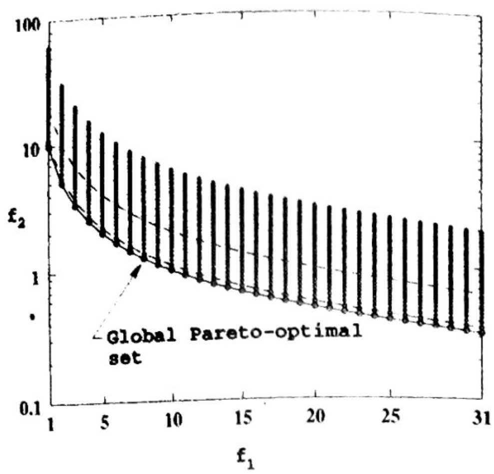
Email: iit.vkash@gmail.com, debjani@maths.iitkgp.ernet.in



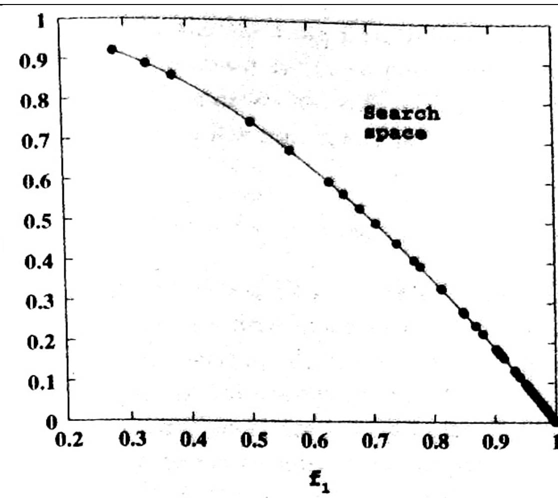
The search space near the Pareto-optimal region for ZI



The partial search space near the Pareto-optimal region for ZDT4



The search space and the Pareto-optimal region for Z



The Pareto-optimal region for ZDT6.

Five functions belonging to the ZDT series of test functions [4] are well known benchmark functions for testing GA. These functions checks the convergence and scalability of a GA on parameters like: Convex/Non convex optimal fonts, Continuous/Discontinuous fonts, Sparsely/Densely populated fonts etc.

4.2 FZDT

FZDT test functions are fuzzy adaptations of ZDT test functions [4] that provide a check to the convergence and scalability of a FGA for above mentioned parameters.

These functions have two objectives that need to be minimized:
Minimize $f_1(X)$,
Minimize $f_2(X) = g(X) * h(f_1(X),g(X))$

Underlined 5 problems have their Pareto-optimal font when g(X) reaches unity. Although f_1 is a single variable function, the difficulty of the functions can be enhanced by using a multivariate f1 function.

Table #1: FZDT test functions.

Function	Decision Space	Objective Function	Optimal solution

Email: iit.vkash@gmail.com, debjani@maths.iitkgp.ernet.in

FZDT1	$X \in [0,1]^{30}$ n=30	$f_1(X) = \tilde{1}*x_1$ $h(f_1,g) = \tilde{1} - \sqrt{(f_1(X) / g(X))}$ $g(X) = \tilde{1} + 9 *(\sum_{i=2}^{n} \tilde{1} *x_i)/(n-1)$	$0 \le x_1^* \le 1$ and $x_i^* = 0$ for $i = 2,3,\dots,30$ This is a problem having a convex Pareto-optimal set.
FZDT2	$X \in [0,1]^{30}$ n=30	$f_1(X) = \tilde{1}*x_1$ $h(f_1,g) = \tilde{1} - (f_1(X) / g(X))^2$ $g(X) = \tilde{1} + 9 *(\sum_{i=2}^{n} \tilde{1} *x_i)/(n-1)$	$0 \le x_1^* \le 1$ and $x_i^* = 0$ for $i = 2,3,\dots,30$ This is a problem having a non-convex Pareto-optimal set.
FZDT3	$X \in [0,1]^{30}$ n=30	$f_1(X) = \tilde{1}*x_1$ $h(f_1,g) = \tilde{1} - \sqrt{(f_1(X) / g(X))} -$ $\sin(10*\pi*f_1)* f_1(X) /g(X)$ $g(X) = \tilde{1} + 9 * (\sum_{i=2}^{n} \tilde{1}*x_i)/(n-1)$	$0 \le x_1^* \le 1$ and $x_i^* = 0$ for $i = 2,3,\dots,30$ This is a 30 variable problem and has a number of disconnected Pareto-optimal fronts.
FZDT4	$X \in [0,1] \times [-5,5]^9$ n=10	$f_1(X) = \tilde{1}*x_1$ $h(f_1,g) = \tilde{1} - \sqrt{(f_1(X) / g(X))}$ $g(X) = \tilde{1} + \tilde{10}*(n-1) + \{ \sum_{i=2}^{n} (\tilde{1}*x_i^2 -$ $10*\cos(\tilde{4}*\pi*x_i)) \}$	$0 \le x_1^* \le 1$ and $x_i^* = 0$ for $i = 2,3,\dots,10$ This is a 10 variable problem with a convex Pareto-optimal set.
FZDT6	$X \in [0,1]^{10}$ n=10	$f_1(X) = \tilde{1}*x_1 - \exp(-4*x_1)*\sin^6(\tilde{6}*\pi*x1)$ $h(f_1,g) = \tilde{1} - (f_1(X) / g(X))^2$ $g(x) = \tilde{1} + 9 *\{(\sum_{i=2}^{n} \tilde{1}*x_i) /(n-1) \}^{0.25}$	$0 \le x_1^* \le 1$ and $x_i^* = 0$ for $i = 2,3,\dots,10$ This is a 10-variable problem having a non-convex Pareto-optimal set. Here the density of solutions across the Pareto-optimal region is non-uniform and the density towards the Pareto-optimal front is also thin.

All the coefficients of the decision variables can be seen as normalized fuzzy input parameters taking part in the optimization process. Proper care must be taken about the lower and the upper bound of terns like $(\tilde{1}*x_i)$. The entire spread of such parameters must lie within the search space.

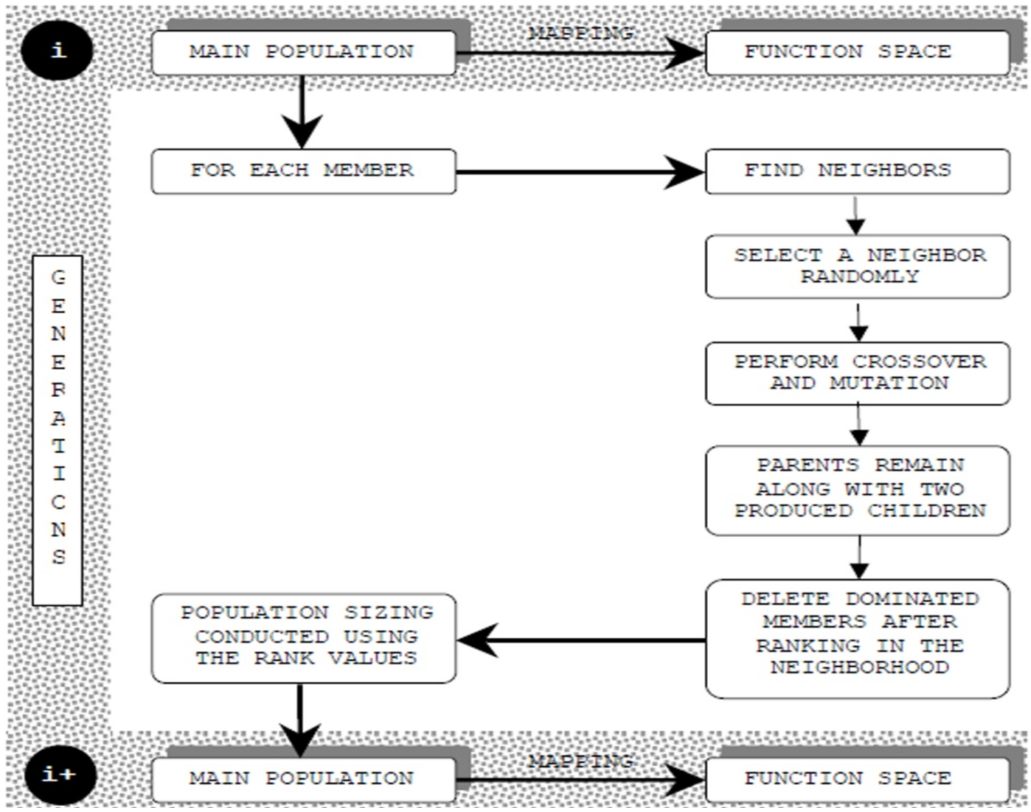
For example: In FZDT1 the entire spread of $(\tilde{1}*x_1)$ must lie within [0,1].

5 FNSGA & FNMGA

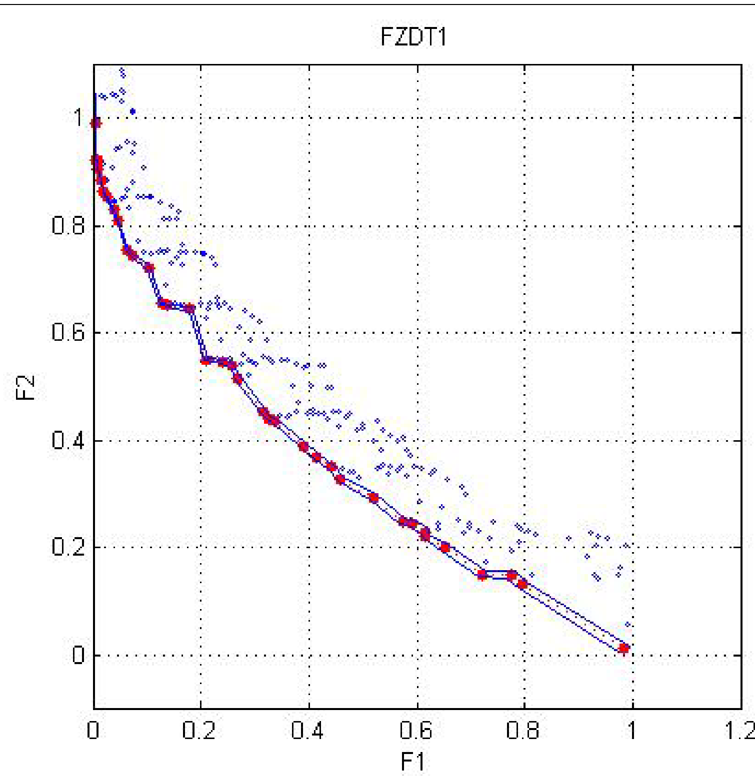
Based on the FGA framework discussed above, two real coded algorithms are developed: FNSGA and FNMGA. These algorithms are fuzzy adaptations of two very effective GAs: NSGA [ ] & NMGA [ ].

5.1 **FNMGA**

It works on a neighborhood concept in the functional space, utilizes the ideas on weak-dominance and ranking and uses its own procedures for population sizing.

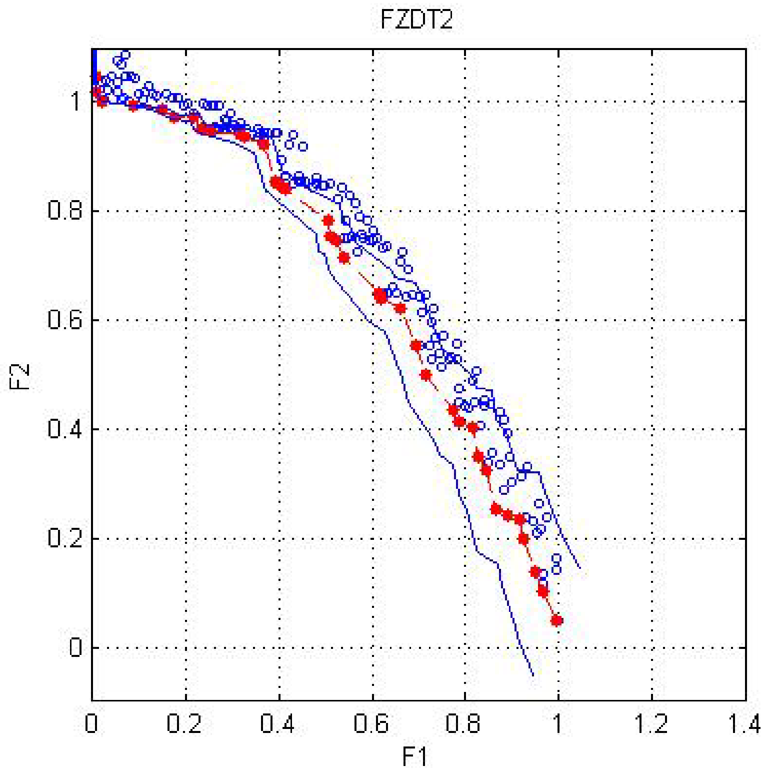


5.1.1 Results

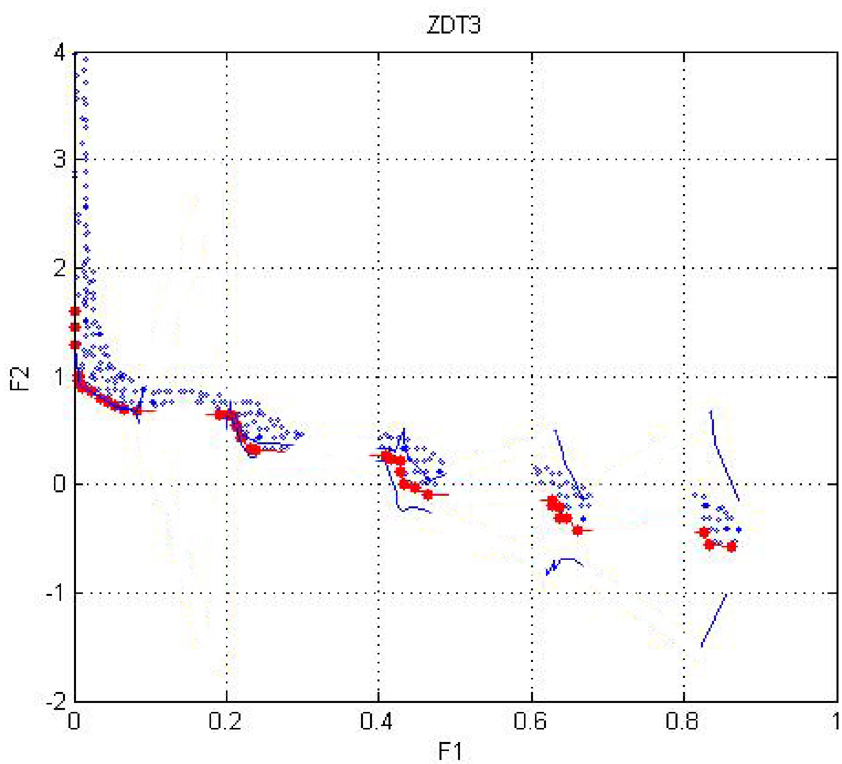


No. of generations = 300
Population size = 500
Input parameters Spread = 5%

Email: iit.vkash@gmail.com, debjani@maths.iitkgp.ernet.in

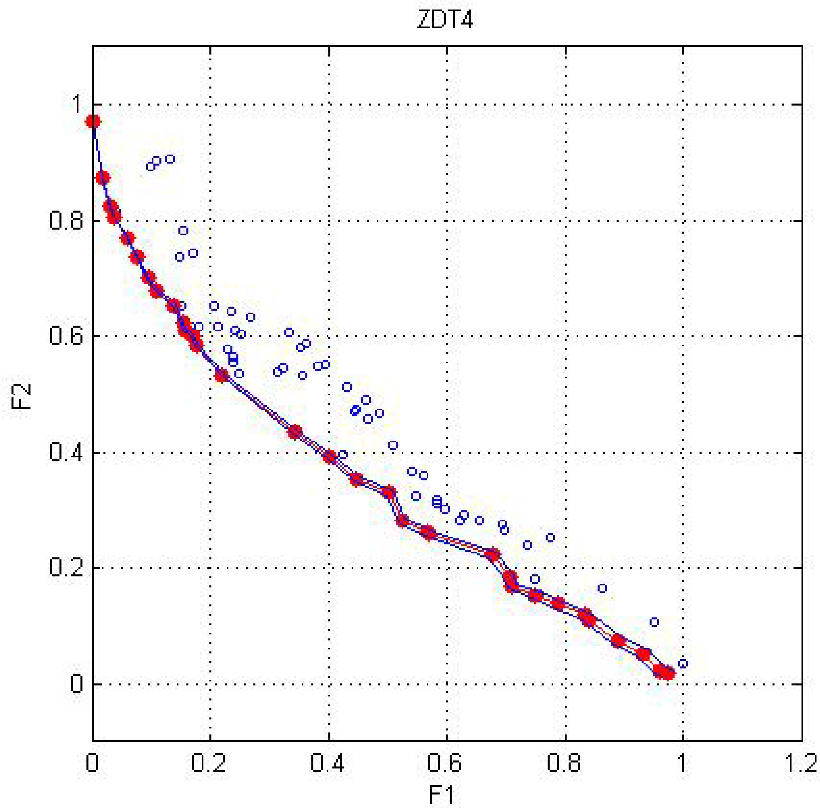
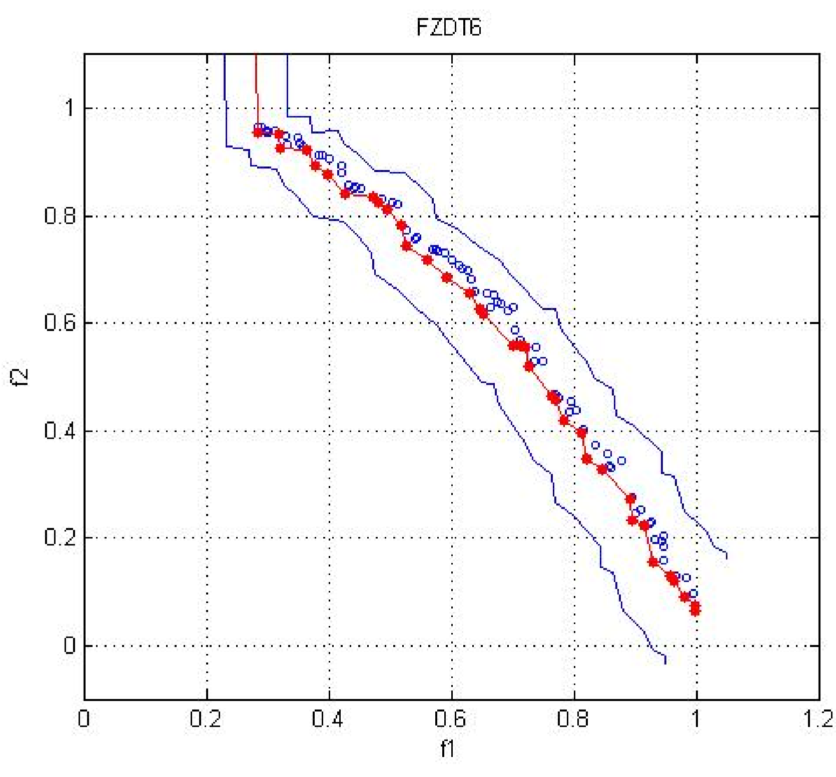


No. of generations = 200
Population size = 400
Input parameters Spread = 5%



No. of generations = 200
Population size = 500
Input parameters Spread = 1%

Email: iit.vkash@gmail.com, debjani@maths.iitkgp.ernet.in

	No. of generations = 300 Population size = 500 Input parameters Spread = 5%
FZDT5 applicable only in the case of binary coded fuzzy genetic algorithm	
	No. of generations = 300 Population size = 500 Input parameters Spread = 5%

Email: iit.vkash@gmail.com, debjani@maths.iitkgp.ernet.in

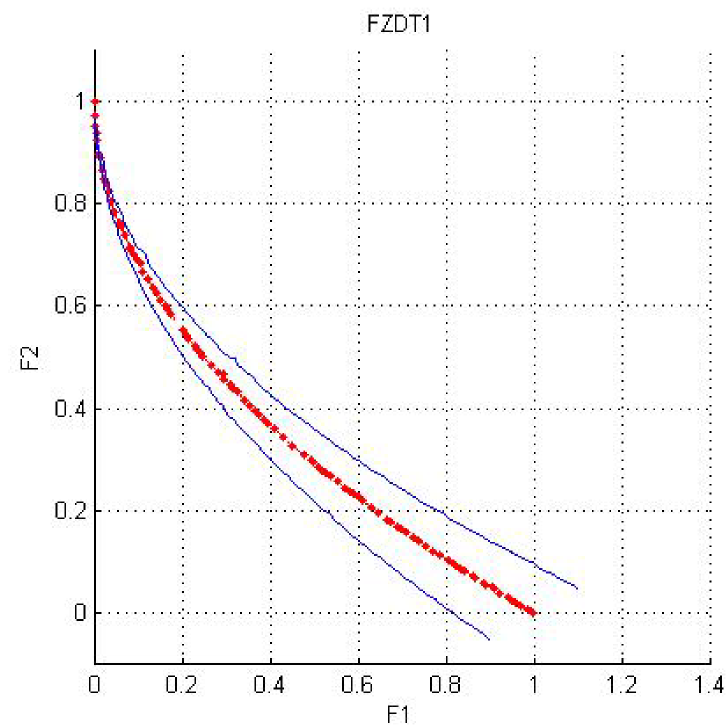
Matlab (7.6.0 R2008a) code was developed and above algorithm was tested on Windows XP platform. FNMGA was found to be consistent with the FZDT functions. Area between solid blue lines denotes the entire spread of the Pareto font. The defuzzified value of the Pareto front (red line) was found to resemble the characteristics of the Pareto font of the corresponding ZDT functions.

5.2 FNSGA

FNSGA II is a fuzzy adaptation of NSGA II algorithm. The initialized population is sorted based on non-domination into each front. The first front being completely non-dominant set in the current population and the second front being dominated by the individuals in the first front only and the front goes so on. Each individual in the each front are assigned rank (fitness) values or based on front in which they belong to. Individuals in first front are given a fitness value of 1 and individuals in second are assigned fitness value as 2 and so on. In addition to fitness value a new parameter called crowding distance is calculated for each individual. The crowding distance is a measure of how close an individual is to its neighbors. Large average crowding distance will result in better diversity in the population. Parents are selected from the population by using binary tournament selection based on the rank and crowding distance. An individual is selected in the rank is lesser than the other or if crowding distance is greater than the other 1. The selected population generates offspring from crossover and mutation operators. The population with the current population and current offspring is sorted again based on non-domination and only the best N individuals are selected, where N is the population size. The selection is based on rank and the on crowding distance on the last front.

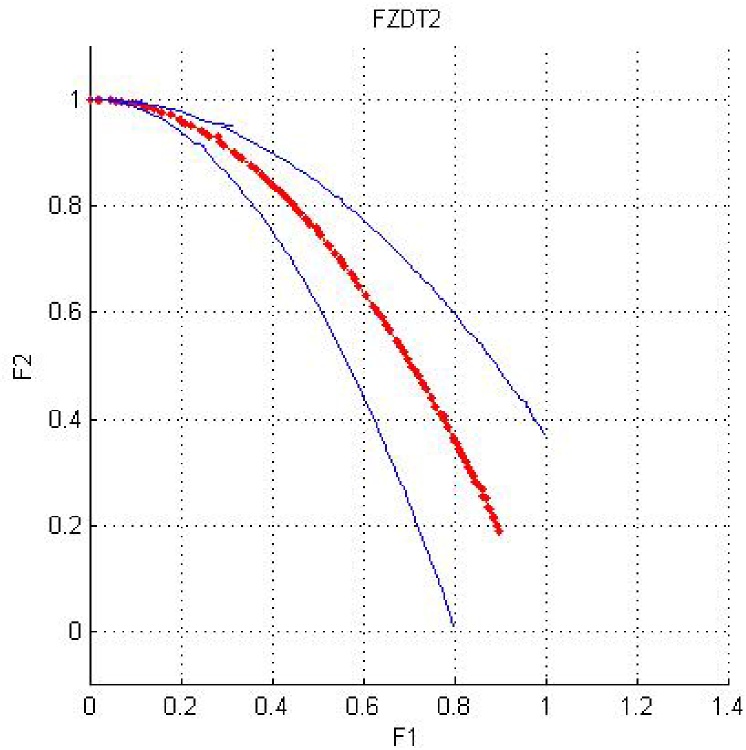
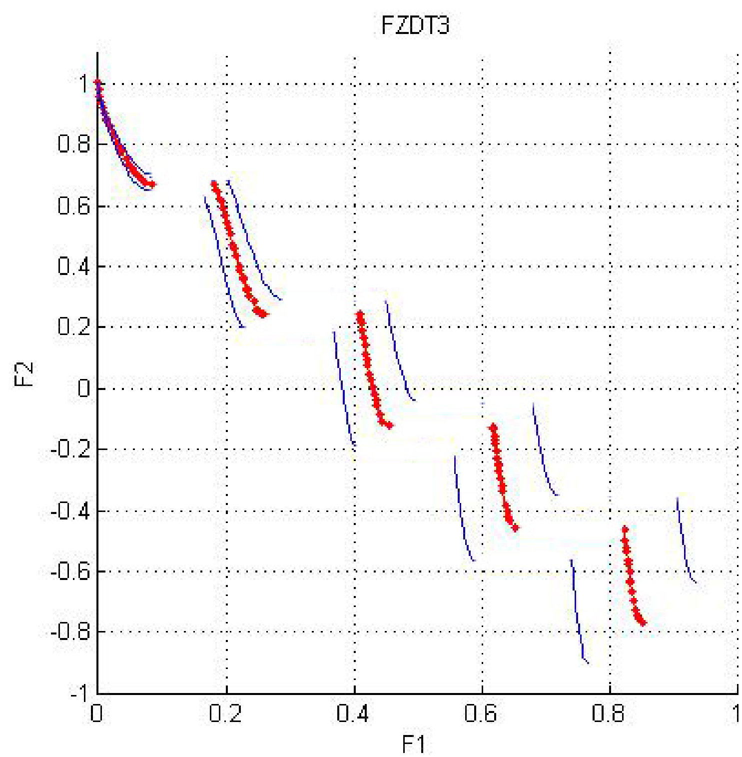
5.2.1 Results

A package in for FNSGA II algorithm was developed and above algorithm was tested on Linux platform (Debian). FNSGA II was found to be consistent with the FZDT functions. Area between solid blue lines denotes the entire spread of the Pareto font. The defuzzified value of the Pareto front (red line) was found to resemble the characteristics of the corresponding ZDT functions.

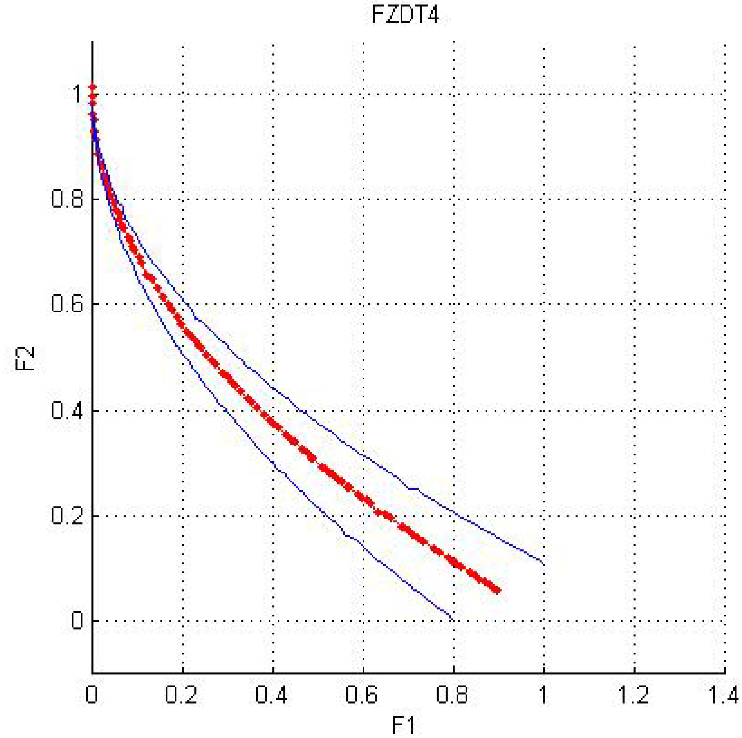
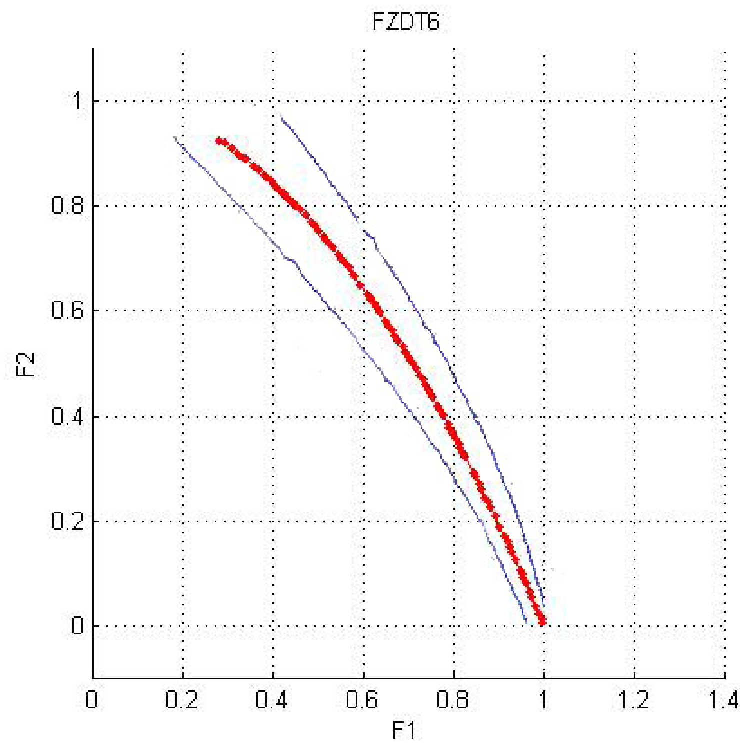


No. of generations = 300
Population size = 500
Input parameters Spread = 5%

Email: iit.vkash@gmail.com, debjani@maths.iitkgp.ernet.in

	No. of generations = 300 Population size = 500 Input parameters Spread = 5%
	No. of generations = 300 Population size = 500 Input parameters Spread = 5%

Email: iit.vkash@gmail.com, debjani@maths.iitkgp.ernet.in

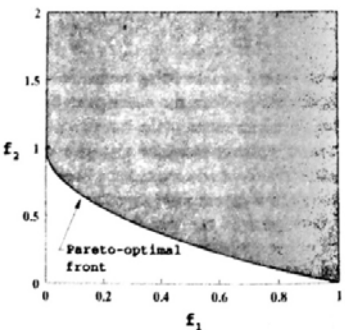
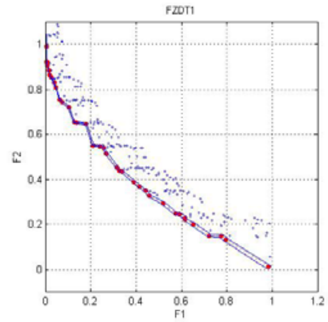
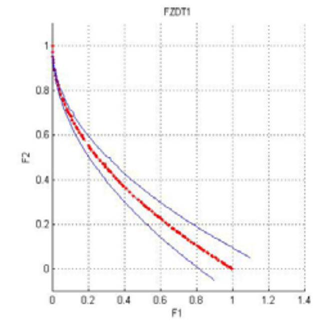
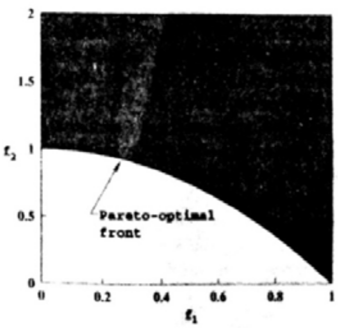
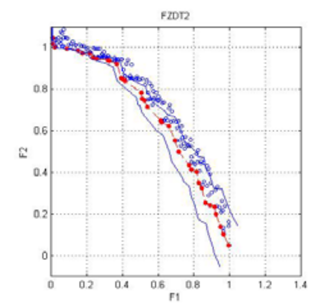
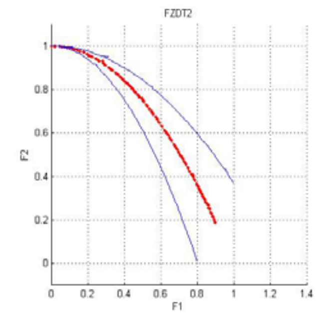
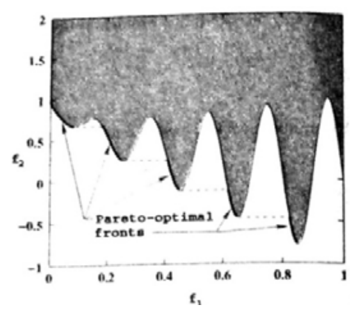
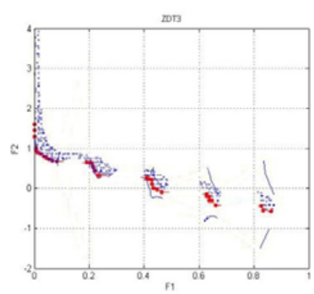
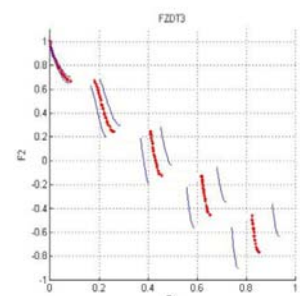
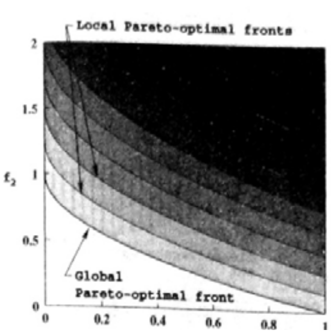
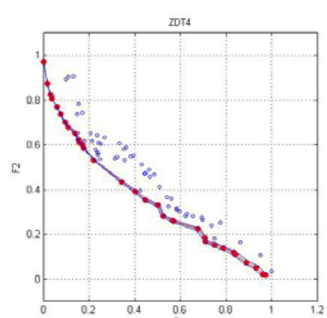
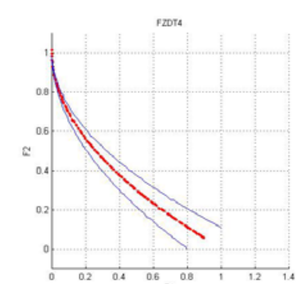
	No. of generations = 300 Population size = 500 Input parameters Spread = 5%
FZDT5 applicable only in the case of binary coded fuzzy genetic algorithm	
	No. of generations = 300 Population size = 500 Input parameters Spread = 5%

6.0 A comparative study of the two FGA: FNMGA & FNSGA II

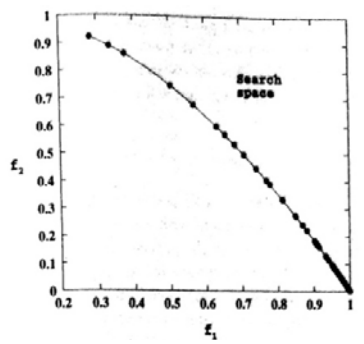
A study of the results of the two algorithms was done. For the same no. of generation and same population size FNMGA and FNSGA II show variability in convergence and distribution. FNMGA II gives better convergence but the distribution of individuals across the Pareto front is not uniform.

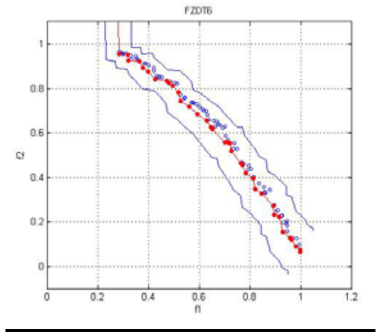
Email: iit.vkash@gmail.com, debjani@maths.iitkgp.ernet.in

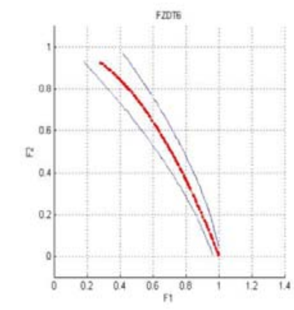
FNSGA on the other hand falls behind in convergence but display an excellence distribution of individuals across the entire frontier

ZDT Functions	FZDT results on NMGA	FZDT results on NSGA II
		
		
		
		

Email: iit.vkash@gmail.com, debjani@maths.iitkgp.ernet.in







7.0 Conclusion

The results obtained are in direct coherence with that of ZDT functions working in real environment. Fuzzy Genetic algorithm successfully maintained the shape and range of the frontier for all FZDT functions. The defuzzified values (GMI) of the elite members exactly copied the shape and range of the frontier for each of the ZDT functions for real environment. Therefore it can be concluded that the Fuzzy genetic algorithm very well captures the essence of multi objective optimization in fuzzy domain. And the proposed Fuzzy ZDT (FZDT) test functions can act as a benchmark for testing fuzzy genetic algorithms.

8.0 References

1) S.H.Chen, C.H.Hsieh, Graded Mean Integration Representation of Generalized Fuzzy Numbers, Proceeding on Sixth Conference of Fuzzy Theory and its Applications (CD ROM), *Chinese Fuzzy Systems Association, Taiwan, Republic of China (1998), 1-6.*

2) S.H.Chen, C.H.Hsieh, Ranking Generalized Fuzzy Numbers with Graded Mean Integration Representation, Proceeding of Eighth International Fuzzy Systems Association Word Congress (IFSA'99 Taiwan), *Taiwan, Republic of China, 2(1999), 551-555.*

3) K.Deb, Multi-objective Optimization using Evolutionary Algorithms, *Wiley-Interscience Series in Systems and Optimization, John Wiley & Sons, 2001.*

4) Zitzler E., Deb k., Thiele L, (2000) Comparison of multi objective evolutionary algorithms: Empirical Results, *Evolutionary computation journal 8(2), 125-148.*

5) H.J.Zimmermann, Fuzzy Set Theory and its Applications, *Allied, Delhi & Kluwer, Boston, 1991.*

6) D.Chakraborty, O.Dey, A Single-Period Inventory Problem with Resalable Returns: A Fuzzy Stochastic Approach, International Journal of Mathematical, Physical and Engineering Sciences, Volume 1, Number 1, 8-15.

Email: iit.vkash@gmail.com, debjani@maths.iitkgp.ernet.in