Randomised Optimization Algorithms With Examples

NAGUBANDI MAHESH M.Tech, IEEE Member, Lecturer, SECE, HIOT- Hawassa University,Ethiopia
DULAM SRIKANTH M.Tech(NIT Kurukshetra),India.

The main objective of this Paper is the "OPTIMIZATION OF UNCONSTRAINED FUNCTIONS". Here we will optimize some of the functions using the optimization tool box which is one of the tool boxes of the MATLAB and we will find the minima of the function. After finding the minima the result is examined such that it satisfies the lemma 11.1 of Vidyasagar.M. With help of this software we can gather the required data, and it is easy to define the models and also to know the results. These algorithms will solve constrained, unconstrained, continuous and un-continuous problems.

The methodology is, considered an Unconstrained Non-linear function and calculated the minimum value of the given function by using the Optimization tool box of the MATLAB. And written MATLAB code for the considered functions and by using optimization tool box, run the solver and view the minimum value of the given function and compared with the Vidyasagar lemma 11.1. the results had shown that the functions satisfies all the conditions of Vidyasagar theorem.

INTRODUCTION

"Optimizationcan be related with Engineering, because Engineering is the process of taking all the discoveries from science and implementing them as a practical devices and making them better and better which is nothing but Optimization".

For finding a best solution to a problem is non-trivial task in all the fields of Engineering and Science even though the meaning of best solution is not the accurate solution. To simplify given problem in terms of mathematical representation is given by F for some non-linear functions with n parameters. The problem of finding the best solution is nothing but finding the maxima or minima for the given test function. There are many ways for doing optimization completely but there exists a large class of problems whose performance of the systems cannot be determined.Since there are different class of test problems there exists different methods and algorithms for solving them and each having with its own advantages and disadvantages.

When considering different class of test problems the most difficult class of test problems would be the class of test problems which are having more than one local minima or maxima. Therefore these problems require a method of search which is global in nature. There are also numerous test functions which are continuous in nature,

1. Unimodal, convex, multidimensional.
2. Multimodal, two-dimensional.
3. Multi modal, multi-dimensional.

Class 1 test functions contain good functions and also they have nasty cases causing slow convergence to single global minimum or maximum. Class 2 type of test functions are of two types one is the test function which are having small number of local minima or maxima and the other is with large number of local minima or maxima. This type of test functions is used to test the quality standard optimization procedure in difficult environment. Class 3 type test functions and class 2 type test functions which are having large number of local minima or maxima are use to test the quality of optimization methods. Usually these test problems are considered as difficult test problems.

Optimization problem can be expressed as given a test function which is $f: X \to R$(whereR denotes real number set) that is X is some set belongs to real number setR. Assumingxand $x'$are the elements Xof such that $f(x') \geq f(x)$ for all x in

R this is the condition for finding the maxima and $f(x') \leq f(x)$ is the condition for finding the minima. The function f has different names like objective function, cost function, energy function. The best solution that maximises or minimises the objective function is called optimal solution.

As there are several algorithms for solving different types of test problems. There is an algorithm which can solve the test problem accurately most of the time is"randomised algorithm". This is newly attaining popularity algorithm among other algorithms.The theory of randomised algorithm is proposed by M.Vidyasagar. For a given function it is not easy to find the maximaor minima. In the field of control systems Vidyasagar has a tremendous work and his work plays a crucial role in the development of randomized approach. The main quality of this approach is described by constructing algorithms which make use of independent sampling in order to find the approximate maxima for the givenunconstrained function.

Assume a test function fwhich is unconstrained function such that f : X->R that is the function belongs to the real number set. x And x'are the variables of the set X such that x and x' belongs toR. Assuming all these conditions according to the lemma 11.1 of Vidyasagar and the algorithms he proposed in the lemma can be used forfinding maxima of an unconstrained function. Main intention is to find the X_{max} which is shown in the below figure.

APPROXIMATE OPTIMIZERS

Consider an optimization problem, Assuming f:X->R is a random variable with $X \subseteq R$and, $f(X) = \max_{1 \leq i \leq m} f(x_i)$

Let $\{x_1 ....x_m\}$ be the independently and identically distributed samples generated uniformly.Assuming a test function fwhich is unconstrained function such that f: -> R that is the function belongs to the real number set. x And x' are the variables of the set X such that x and x' belongs to R.

Then the set of approximate domain optimizer is H(α),

$H(\alpha)=\{ x \in X : \lambda \{x' \in X : f(x') > f(x) \} \leq \alpha\lambda(X)\}$

For a given test function the above condition is used to find the number of x values lies within the setH(α). After obtaining the x values that lies inside the set it is observed that the region of x will be less than the size of the set X and α which is a given number which lies between [0, 1]. The region of x is called the feasible region, according to Vidyasagar the maxima lies within that feasible region.

For a given test function need to check the condition $f(x') > f(x)$ and take out the x values which satisfies the condition and then check for the condition,

$\lambda\{x' \in X : f(x') > f(x) \} \leq \alpha\lambda(X)$

By this condition the number of x values lies within the setH(α) are obtained.Now according to the lemmaindependently and identically distributed samples are generated uniformly $\{x_1 ....x_m\}$ at random. Then for eachα>0, it can be stated with confidence that the approximate domain optimizer set is greater than or equals to $1-(1-\alpha)^m$.

Therefore then check for the condition, $H(\alpha) \geq 1-(1-\alpha)^m$

Where 'm' is number of samples generated uniformly.

By generating the samples uniformly in between the range ofx and checking the condition then according to Vidyasagar approximately near maxima for the given test function can be found.

ALGORITHM AND VIDYASAGAR RESULT

Algorithm 1: (to find thenumber of elements of the set x which are inside the set H(α))

1. Considering approximate domain optimizer set,

   $H(\alpha)=\{ x \in X : \lambda\{x' \in X : f(x') > f(x) \} \leq \alpha\lambda(X)\}$

2. Assuming x and x' are the elements of X and belongs to the real number set R.
3. Assuming f is the test function whose maxima have to be found.
4. Substitute x and x^' values in the given test function and check for the conditionf(x')>f(x).
5. For a given range of x and x' number of elements of x which satisfy the above condition can be obtained.
6. Check the condition that total number of elements of x which are obtained from the above condition is less than or equals to the product of size of the set X and α which is a given number and lies between 0 and 1.
7. If all the above condition satisfies, then for a given range of x how many number of elements of x are inside the set H(α) can be obtained.
8. Plot the set H(α) for the given test function and givenα.

In general for a given test function plot for the set H(α) can be,

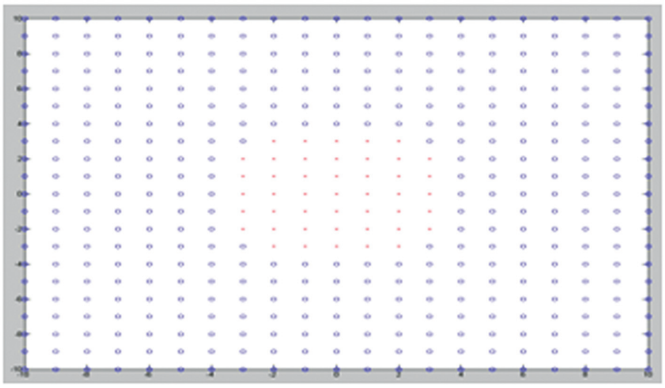

Figure 1 Response to be obtained for algorithm 1.

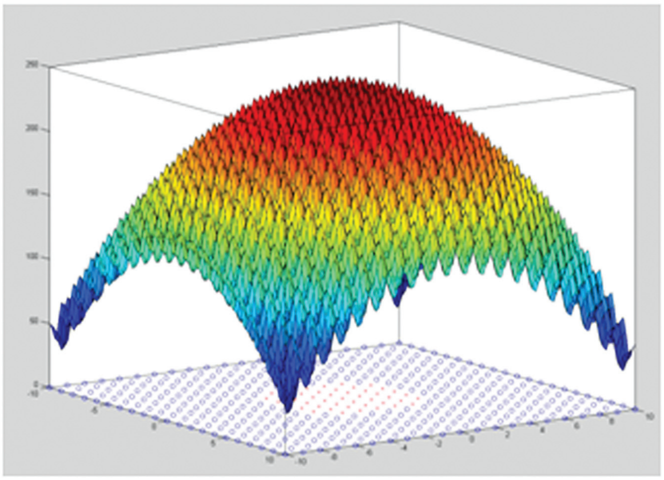

Figure 2 Response of feasible region within the function area.

Figure 1describes the range of x that lies inside the approximate domain optimizer set. Blue region in the graphs is the region which does not lies inside the set H(α)and red region that lies inside the setH(α).

Figure 2 describes where will be the feasible region lies for a given test function can be known. Red region is the feasible region and the blue region which is out of the set for the given test function.

Algorithm 2: (to find the maxima of the given test function)

1. Check for the condition P{H(α) }≥1-(1-α)^m where m is number of samples generated for a given range ofx.
2. In order to check above condition random samples for x should be generated uniformly.
3. After generating the samples then extract a sample which is best suited for the given test function and name it has $x_{max.}$
4. Check whether the obtained x_{max} is inside or outside the setH(α).
5. Repeat the above process for many times and for each time check the obtained x_{max} is inside or outside the set H(α).
6. Then check for the condition that number of x_{max} inside the set H(α) divided by total times of repetition should be greater than (1-α)^m that is H(α) / (total number of repetition) ≥1-(1-α)^m.
7. Plot the graph for the above condition.

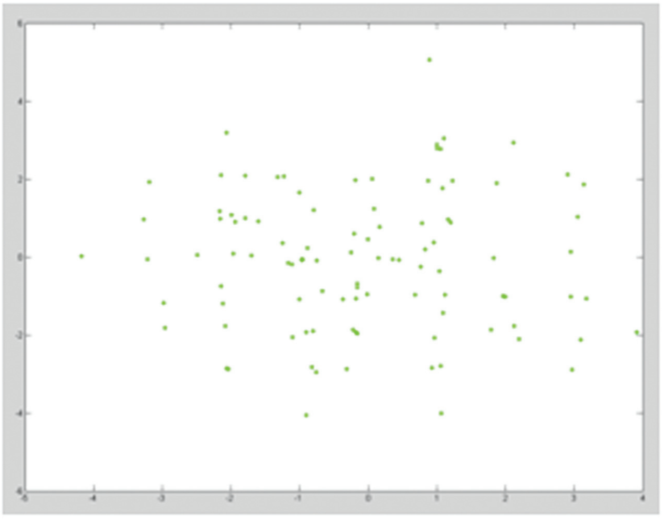

Figure 3 Response obtained for algorithm 2.

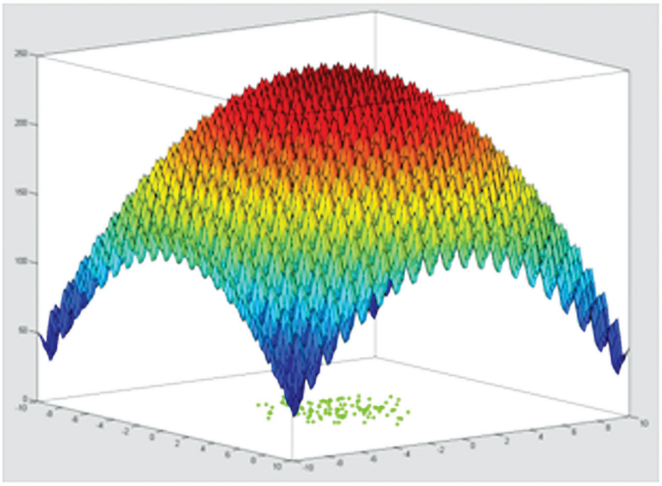

Figure 4 Location of maximum for a given test function.

In general for a given test function plot for the set H(α) / (totalnumberofrepetition) ≥1-(1-α)^m can be shown in the figures 3 &4.

Form figure 3 the green coloured "*" in this figure are the maximum values of x obtained at each time by the evaluation of the second code and it has done for many times. At each time obtained maximum value of x is inside the feasible region which is obtained from the first code. This proves the lemma 11.1 of Vidyasagar for single variable function.

Figure 4 describes the location of the feasible region within the function area and this can be indicated by green coloured "*" symbols which can be seen.

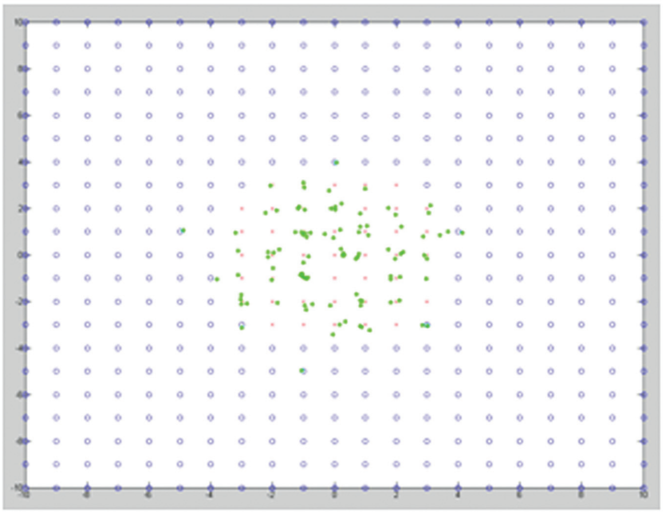

Figure 5 Response for verifying Vidyasagar's theorem

According to Vidyasagar the result of this theorem states that after the completion of the first algorithm a feasible region for x is known where maxima for the given test function lies and after second algorithm location of maxima can be found. In this according to Vidyasagar the maxima for a given test function can be known.

EXAMPLES

MATLAB CODE DESCRIPTION

General Matlab code description is for two variable unconstrained functions, changes implicated for single variable function are notified at relevant location.

In general for finding the maxima of a function according to Vidyasagar two m-files need to be created because there will be two codes for each function.

First code (for Algorithm1)

It is mainly implemented to find a feasible range of data where expectancy for lying of maxima is maximum.

- In first code select the range of (x,y)and(x',y') as -10 to 10.
- Select the range of α which should be positive and lies in between 0 and 1.
- Checking whether *f*(x',y') is greater than *f*(x,y) or not that is *f*(x',y')>*f*(x,y).
- Assuming 'o' as a variable and if the above condition satisfies then 'o' will be incremented by 1.
- Then checking the condition that number of (x,y) values which satisfies the condition are less than or equal to the product of size of the set X andα.
- The sample value of data which satisfies the above condition accumulates to form feasible range of data which is needed for maxima.
- Plotting the setH(α).
- By doing this a feasible region for x and y is obtained where the maxima of the function lies.

Second code (for Algorithm2):

In this code the maxima will be derived from the feasible range taken from the above code.

- In second code for the selected range of x and y generate samples uniformly in between the range of x andy.
- After generating the samples checking which sample will best describes the given function.
- After that checking whether the best sample is inside the feasible region or not.
- A graph is plotted with the obtained best samples.
- Checking the condition H(α)/(totalnumberofrepetition) ≥1-(1-α)^m.
- From this the maxima for the given function can be observed.

Example 1

Test Function taken for computation: $f(x)= -x^2$ is called De Jong's first function. Which is one of the simplest test function and it is continuous, convex and unimodal. The function taken is single variable. Select the range of x and x' as -10 to 10., Step size as 0.01.

Results obtained after evaluating the code for algorithm1, the feasible region of x lies in between -1.25 to 1.25.

Example 2

Function taken for computation: $f(x,y)= -(x^2+ 2y^2)$, is called as weighted sphere model it is also continuous, convex and unimodel. The function taken here has two variables. The range for the variables x,y,x' and y'are taken as -10 to 10. Step size as 1.

Results obtained after evaluating the second code, the feasible region of x lies in between -4 to 4 and y lies in between -3 to 3.

The feasible region can be clearly referred in figure9

ISSN No: 2348-4845

Example 3

Function taken for computation: f(x,y)= -(|x|²+ |y|³). [sum of different powers] which is commonly used unimodel function. All the assumptions are same as the above example 2. The range of xfor this test function is from -4 to 4 and y is from -2 to 2.

The response graphs for this test function can be seen in the figures 16 &17.

RESULTS

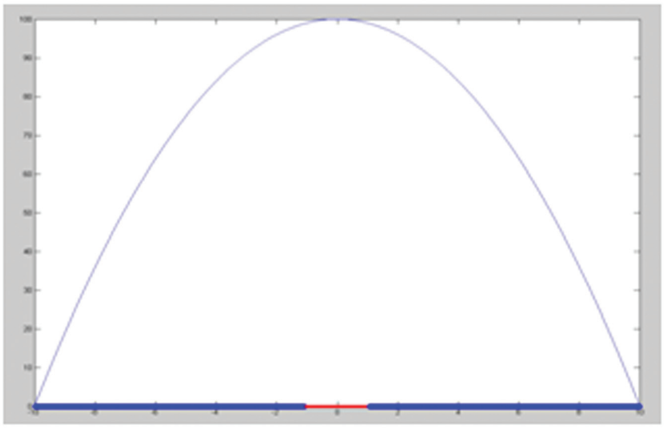

Fig.6

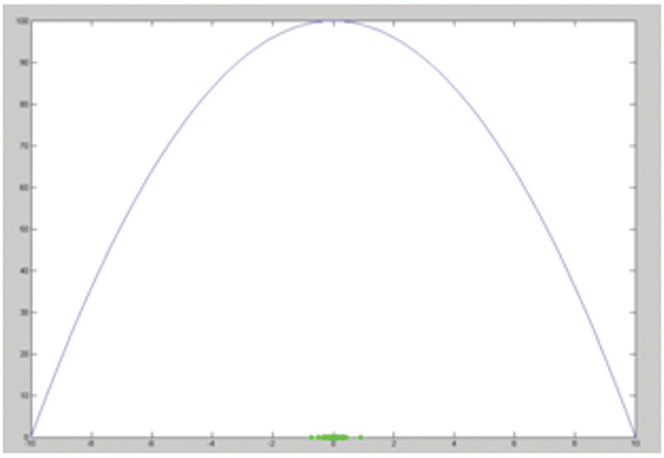

Fig.7

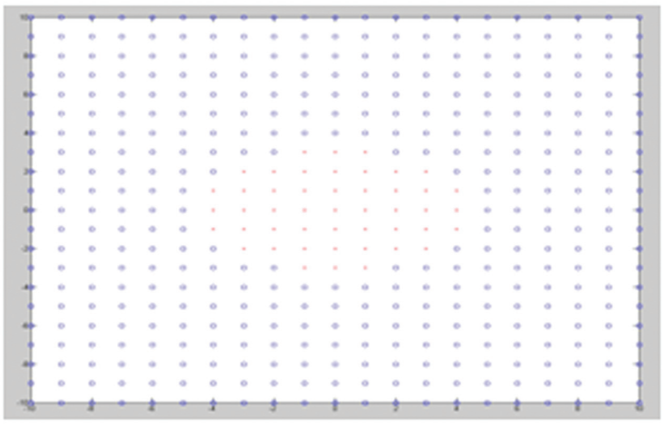

Fig.8

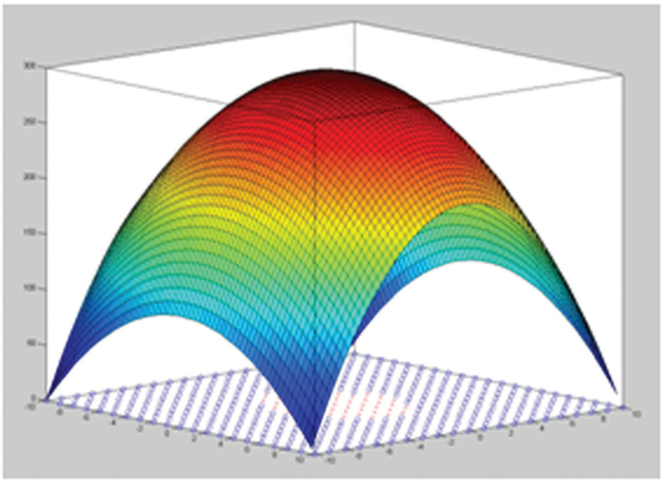

Fig.9

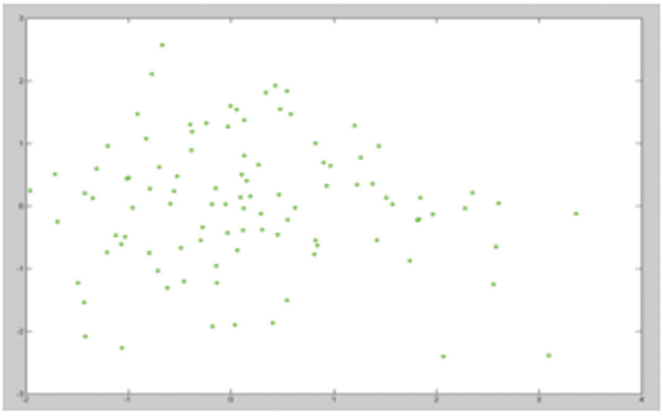

Fig.10

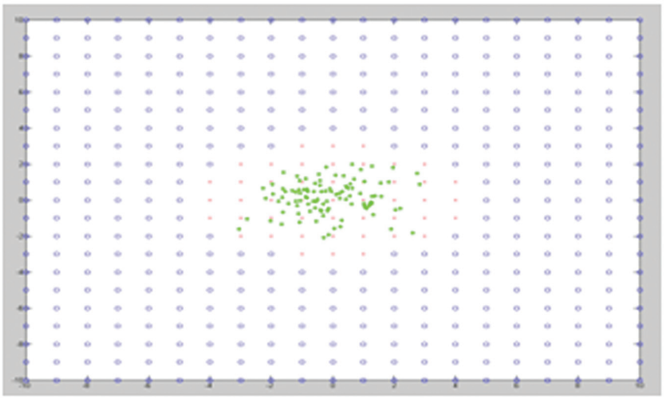

Fig.11

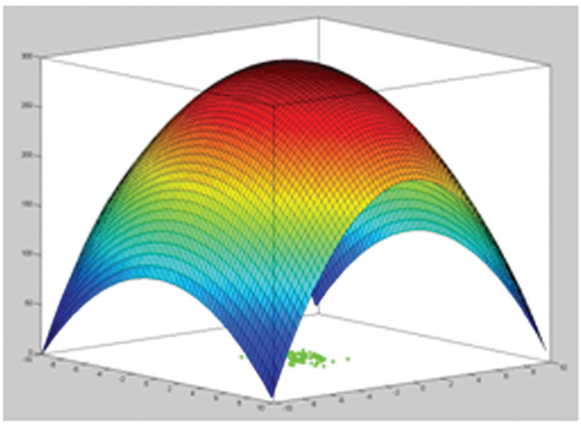

Fig.12

Volume No: 1(2014), Issue No: 4 (April)
ISSN No: 2348-4845

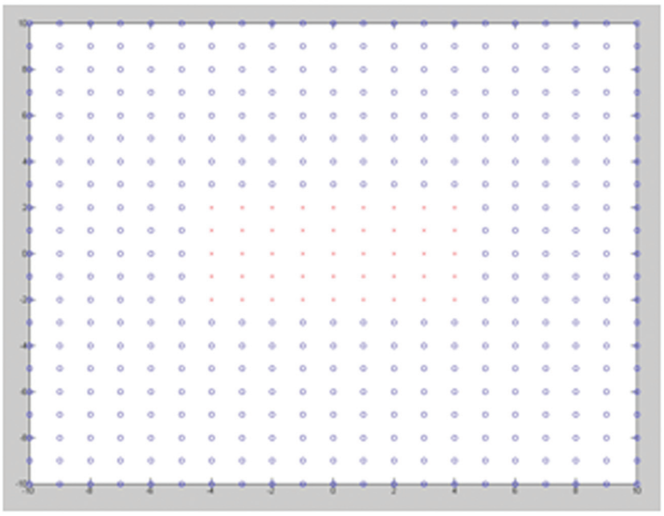
Fig.13

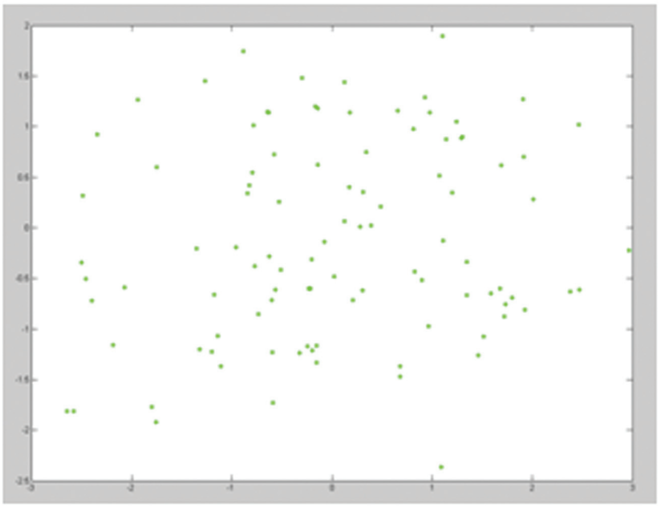
Fig.14

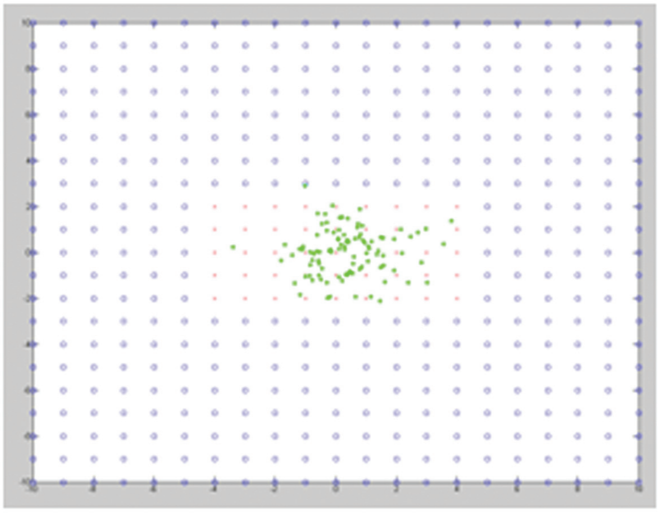
Fig.15

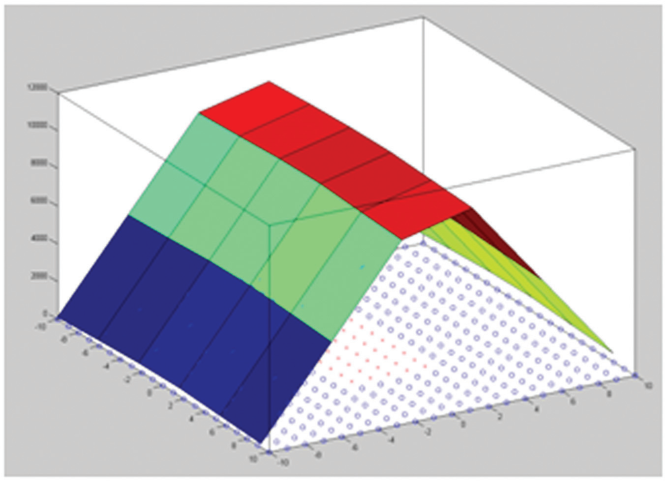
Fig.16

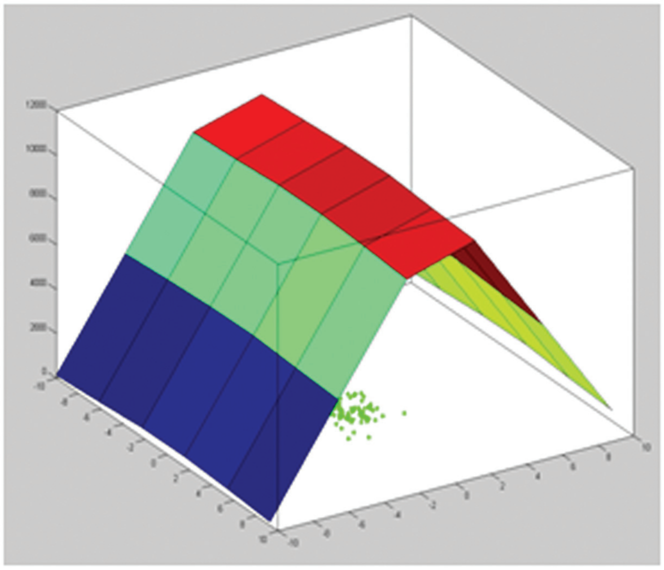
Fig.17

The figures 6 ,8 &13,shows the the feasible region (red colour)where the maxima of the function lies and the region with blue colour is that it is outside the approximate optimizer domain set that is the x values of the blue region does not lie in the set H(α)for the examples 1,2 & 3 respectively..

Form figure 7, 10& 14 the green coloured "*" in this figure are the maximum values of x (x & y for figures 10 & 14) obtained at each time by the evaluation of the code for algorithm2 and it has done for many times. At each time obtained maximum value of x (x & y for examples 2& 3)is inside the feasible region which is obtained from the code for algorithm2 which is shown clearly in figures 11 & 15 . This proves the lemma 11.1 of Vidyasagar for single and two variable functions.

The feasible region can be seen clearly in Figure s 9& 16 for the examples 2&3. Response indicating where the maxima lie within the function for the examples 2& 3 shown in the Figures 12&1 7

Matlab code for algorithms1 & 2 is written as per the Discription 4.1 and then simulated. The results obtained for the three examples considered are shown in the above figures.

CONCLUSION

Based on the simulation results obtained from the examples considered,

$f(x)= -x^2$.

$f(x,y)= -(x^2+ 2y^2)$.

$f(x,y)= -(|x|^2+ |y|^3)$.

Hencevidyasagar theorem has been provedthat the maximum of the test functions lies within the feasible region.This theorem is accurate for most of the functions.

In this project, the maxima for a given test function is found by generating random samples uniformly. This report describes the lemma 11.1 of vidyasagar and procedure for finding the maxima. All the goals kept in front before the starting of work are achieved successfully with considerable results.

7.ACKNOWLEDGMENTS: we would like to express our deepest sense of gratitude to Dr.Katta Ravindra for his valuable suggestions, support and encouragement.

REFERENCES

1. Bruce E. Stuckman (1988). A Global Search Method for Optimizing Nonlinear Systems. IEEE Transactions on systems, Man, and Cybernetics vol.18 965-978.
2. Gaviano, M., Kvasov,D.E., Lera,D., Sergeyev, Y.D.: Algorithm 829: software for generation of classes of test functions with known local and global minima for global optimization. ACM Trans. Math. Softw. 29, 469–480 (2003).
3. Hock, W., Schittkowski, K.: Test examples for nonlinear programming codes. Lecture Notes in Economics and Mathematical Systems, Vol. 187. Springer, Berlin (1981).
4. Locatelli, M.: On the multilevel structure of global optimization problems. Comput. Optim. Appl. 30,5–22 (2005).
5. Marcin Molga, Czesław Smutnicki. 3 kwietnia 2005. Test functions for optimization needs.
6. Marrison, C., & Stengel, R. (1994). The use of random search and genetic algorithms to optimize stochastic robustness functions. Proceedings of the American Control Conference, Baltimore, MD (pp. 1484-1489).
7. M. Vidyasagar. Learning and Generalization: With Application to Neural Networks. Springer-Verlag, London, second edition, 2003.
8. Vidyasagar, M. (1997a). A theory of learning and generalization: With applications to neural networks and control systems. London: Springer.
9. Vidyasagar, M. (1997b). Statistical learning theory and its applications to randomized algorithms for robust controller synthesis, In: G. Basten, & M. Gevers (Eds.), Semi-Plenary Lecture, European Control Conference, Brussels, Belgium (pp. 161-189).
10. Vidyasagar, M., & Blondel, V. (2001). Probabilistic solutions to some NP-hard matrix problems. Automatica, to appear.
11. Vidyasagar, M. (2001). Randomized algorithms for robust controller synthesis using statistical learning theory.Automatica 37 (2001) 1515-1528.
12. V. N. Vapnik. The Nature of Statistical Learning Theory. Cambridge University Press, Springer, New York, US, 1995.
13. www.mathworks.co.uk