OPTEC LAB REPORT

Optimisation Algorithms

Professor Fouad Bennis

Presented By

Arijit Mallick

Sivanand Yadav Katamani

May 2016

Dichotomous Search Optimisation algorithm

1. Theory

The Algorithm for dichotomous search can be explained as follows:

A function f is given along with $Xmin,Xmax$ and ε.

a. We define L= $Xmin$-$Xmax$, and $m= (Xmin+Xmax)/2$, while $k=m-\varepsilon$ and $l=m+\varepsilon$.

b. We compute $f(k)$ and $f(l)$

if $f(k)>f(l)$ then $d2=l;$

else if $f(k)<f(l)$ then $d1=l;$

else if $(f(k) = f(l))$ then $d1 = k; d2 = k$.

c. Go to step *a* until L>2ε.

d. We take the optimum value as the $xopt=min(d1,d2)$ and $fopt=f(xopt)$.

2. MATLAB File and function definition

• For single variable dichotomous Search:

dicho_while.m : This contains the relevant program implementing the aforesaid algorithm for a given function.

Following function has been used for simulation purpose:

$$y=(x-3)^2+5$$

In the following section, we will analyse the result.

RESULTS

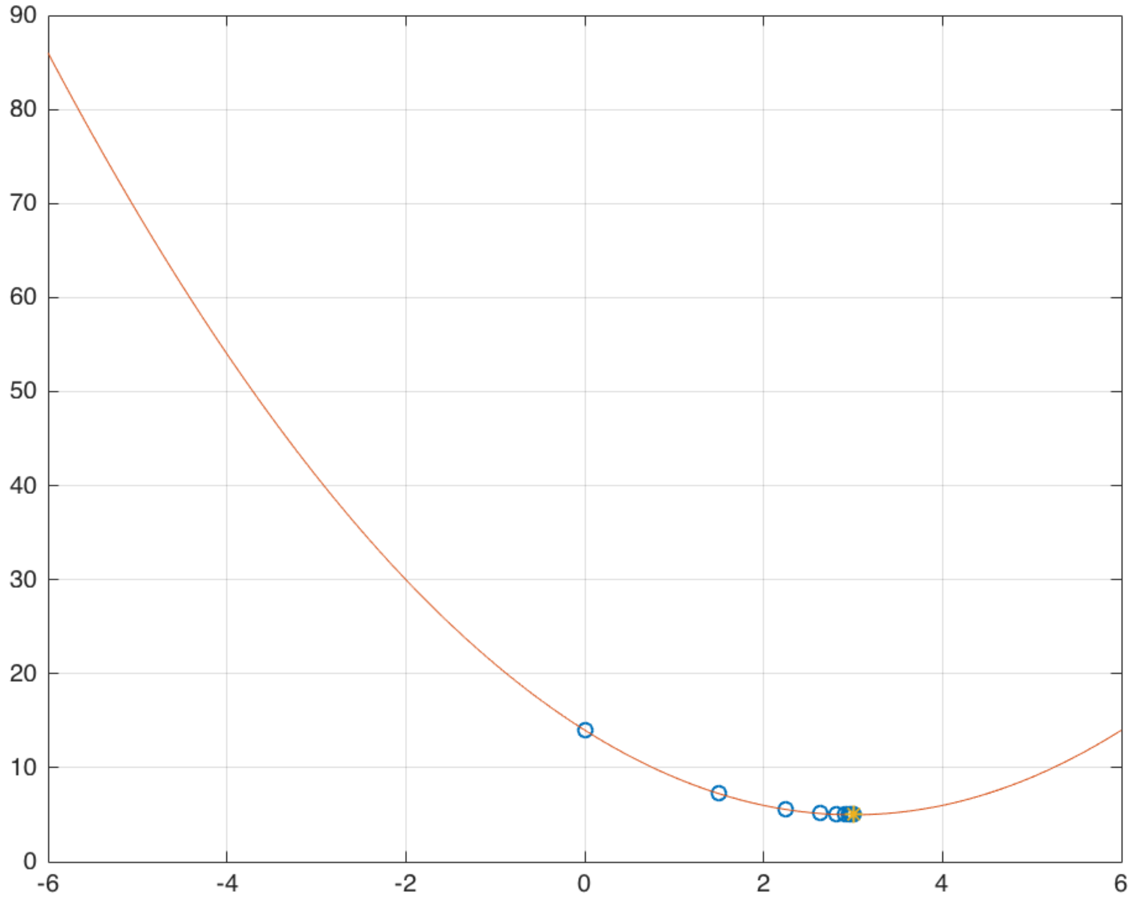


figure 1: for the given function , we get solution w=3.006 as the minima point, ad foot is given by 5.000. Moreover, we can see how the solution converges to the local minima, from the starting points as blue circles and the end point as a golden star at approximately 3.

• For Multivariable dichotomous search at a given direction:

multi_optec_final.m : Here we use almost same version of the given algorithm, but using an additional direction parameter.

We have used a direction parameter and error parameter whose values have been taken as:

d=[1 1]

ε=0.001

We will analyse the result in the next section.

RESULTS

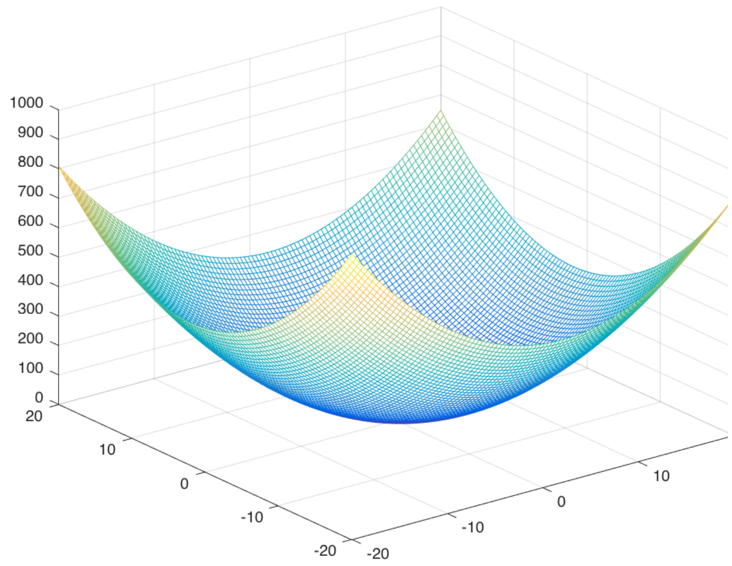
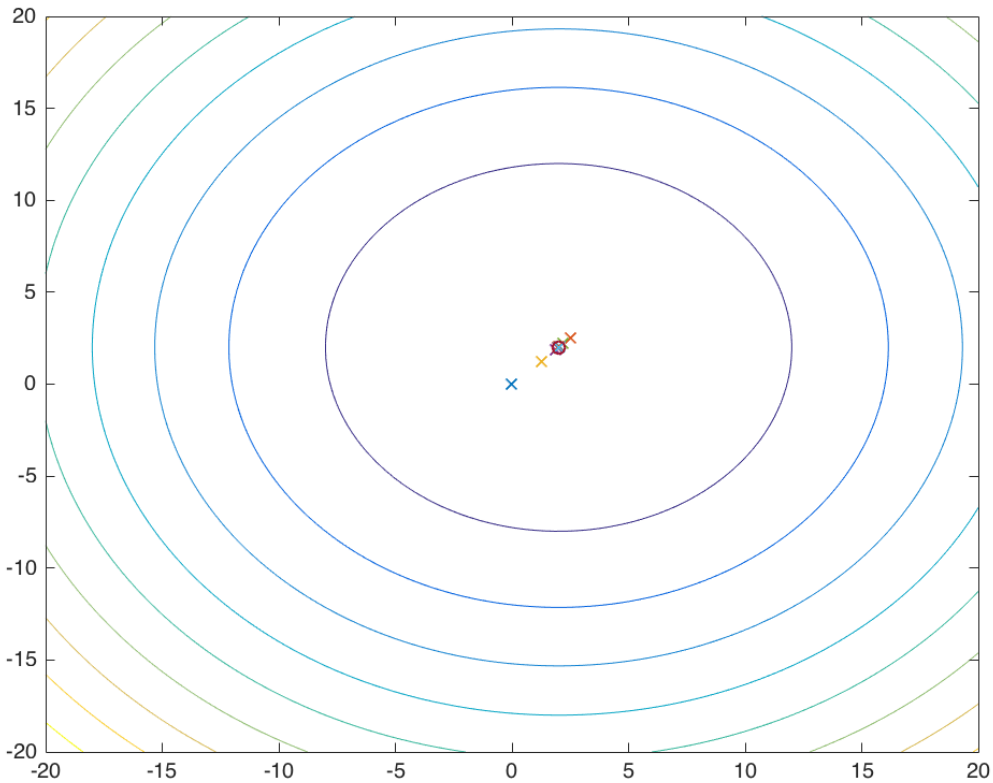


figure 2(a)



<u>*figure 2(a) and (b)*</u>*:* For given parameters as above and starting point as [0 0], it can be clearly how the convergence of minima takes place. The cross points are convergence points while the red circle at centre denotes the solution. The minima point is given by *w= [1.9986 1.9986]* for the given function.

• <u>For Multivariable dichotomous search with variable gradient:</u>

gradient_multiOptec.m : Here we use almost same version of the algorithm given in next section, but using an additional direction parameter.

function_mult_dich.m : Here the dichotomous algorithm function is written to be used continuously as the gradient changes.

inline_gradient.m : Calculation of the gradient for the given function.

Here, we will revisit the algorithm as we have made significant changes. Here, *f(x)* be the given function, and say a given initial point *x*.

```
d=-inline_gradient(f,x);

x=function_mult_dich(x(1),x(2),d,ep,f);
```

The algorithm can be sufficed as follows:

a. Let us enter an initial point and let this be the optimum point.
b. We obtain the negative gradient *d* as shown in above program snapshot, at that point and we find the functional output, *f* and we say that this is the optimum output.
c. As per last dichotomous algorithm, we chose a suitable x vector (2 points).
d. We start a loop.
e. A dichotomous search is done.
f. We reset *x* vector.
g. In the next iteration, we use the optimal parameters for re evaluation.
h. The loop is functioned until the value of gradient is zero.

In the next section, we will discuss our results of the obtained simulation.

<u>RESULTS</u>

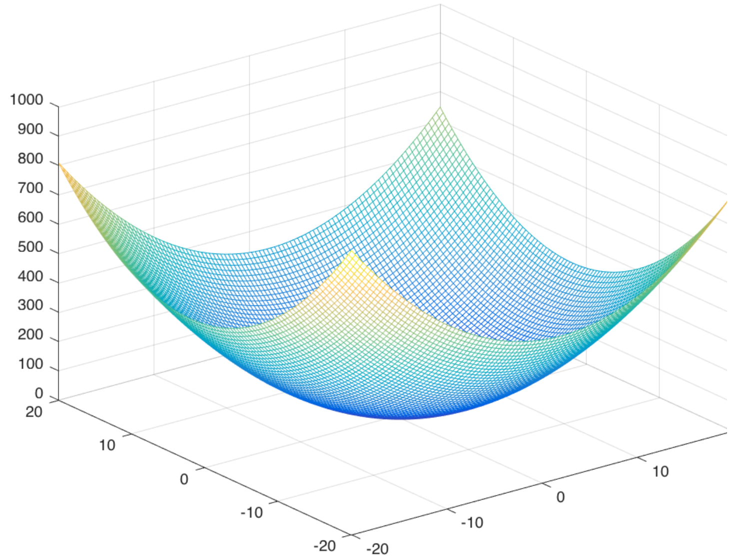


figure 3(a)

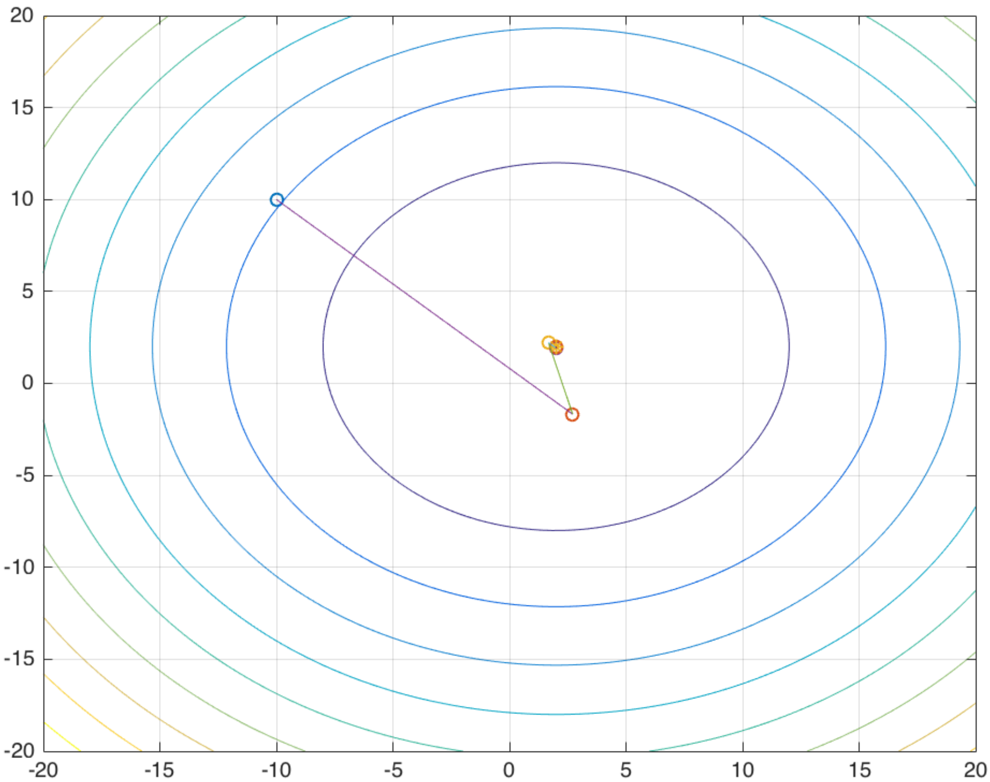


figure 3 (b)

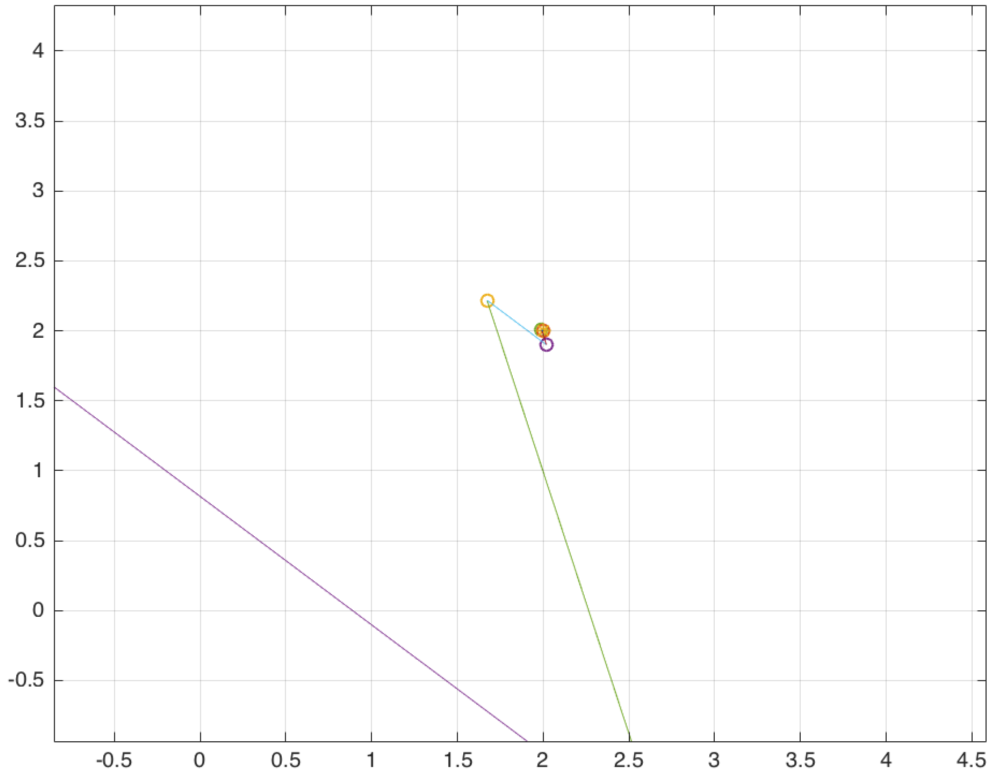


figure 3(c)

figure 3: In the figure 3 (b) we can clearly see the way convergence takes place. we took the initial point as [-10,10]. The points converges finally to exactly [2.000 2.000]. Moreover, it can be seen the way gradient changes, as we have shown it by joining the points through connecting lines. Figure 3(c) shows a magnified version to show the solution, as a golden star point at the optimal point.

Simulated Annealing Optimisation algorithm

1. Theory

The algorithm ca be explained as follows :

a. Let us choose a random initial point, say *x*, having function value as *f(x)*.And let *xopt* denote the optimal point while its corresponding functional point be *fopt(xopt)*.

b. We set the iteration value, say *n*.

c. As per as the theory suggests, we set an initial temperature, *T*.

d. We set the variable $a<1$ (here we will chose 0.9), a factor which continuously decrements *T*.

e. A final temperature *Tf* is set to confine the value of *T*. *here, we take T f as ep.*

f. Continious checking of $Tf<T$.

g. We select a random point y on the nearby region of x.

h. Now, we assign a difference $\Delta = f(y)-f(x)$.

i. Now, if $\Delta<0$, then we assign $x=y$. and if $f(x)<fopt$, then $xopt=x$, and $fopt=f(xopt)$.

j. Else choose a value for *p*, as given in acceptance probability function.

k. if $p>rand()$, then $x=y$., we increment the number of iteration as well.

l. We repeat step f and $T=aT$.

m. Again we go to step e.

2. MATLAB file and function definition

• For single variable results, we have provided the following files:

single_simulated.m: This is the main function implementing the aforesaid algorithm.

acceptance_probability.m: Here the acceptance probability function is defined which defines the acceptance probability formula of step J.

For simulation purpose in single variable, we have selected the following function for analysis:

$$y=(x-2)^2+(x-4)^2+(x-5)^2$$

Results are in the next section.

RESULTS

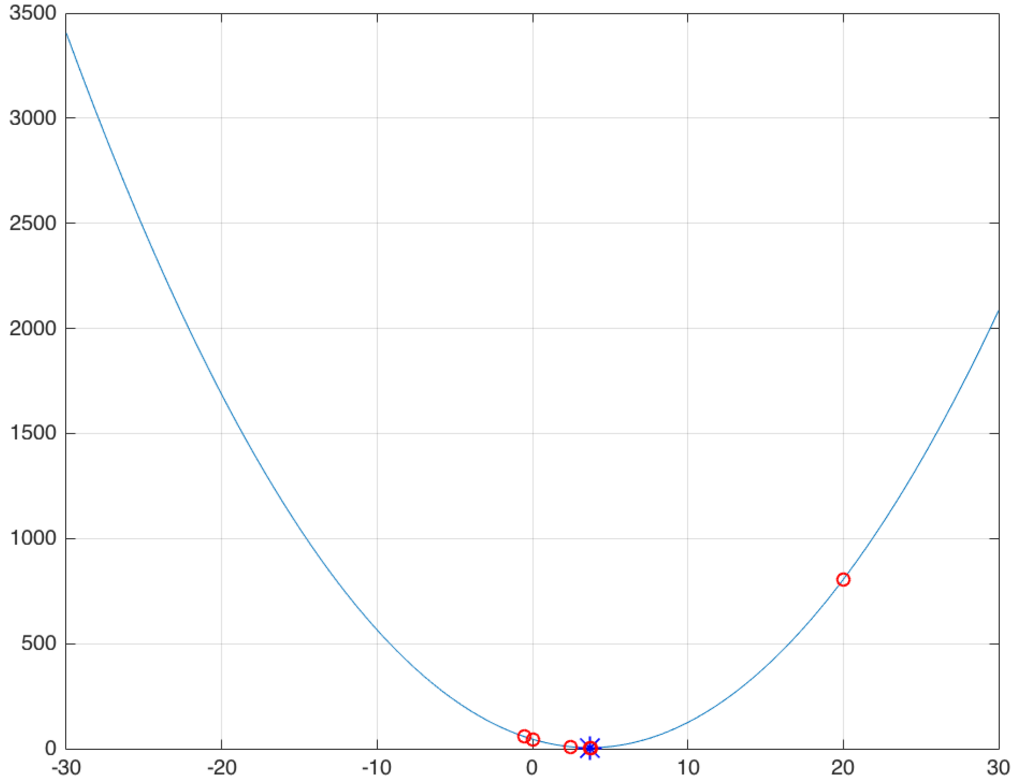


figure 4: Results show that the red dots are the points which are being evaluated, and it slightly converges to the optimum point, shown in blu star. The initial point being *(x=20)*. The function returns *opt=3.6644* and *fopt=4.6667,* which are indeed the optimal point.

• For multi variable results, we have provided the following files:

multi_simulated.m: This is the main function implementing the aforesaid algorithm for multiple variable optimisation problem statement.

acceptance_probability.m: Here the acceptance probability function is defined which defines the acceptance probability formula of step J.

For simulation purpose in multi variable, we have selected the following function for analysis:

$$y=(x_1-2)^2+(x_2-2)^2$$

Results are in the next section.

RESULTS

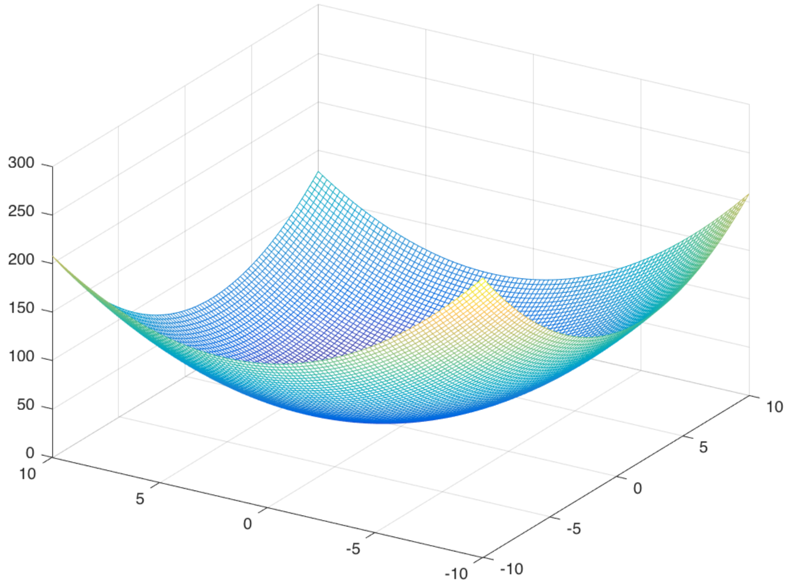


figure 5(a)

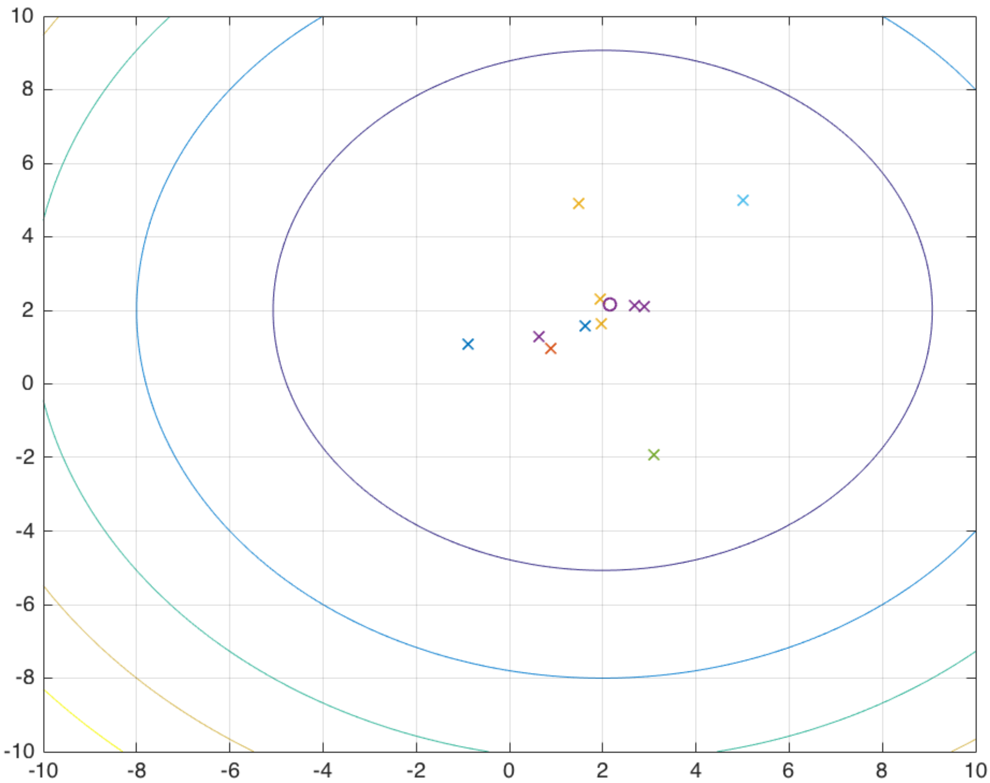


figure 5(b)

figure 5: figure (a) denotes the contour representation of the given function. The next figure shows the traversal of the points before they converge to the optimal point near [2,2]. The circle shows the optimal point, whereas the cross marks are the intermediate points. It is to be noted that the initial point is [5,5].