**MACHINE LEARNING**

**Practical No: 1**

**AIM: Implement feed forward neural network for a given data.**

> # Setting the work directory

> setwd("D:/College/ML/Practicals ")

> # data frame data stores the data of cereal csv file

> # read.csv:- 1st parameter is file = name of the csv file,

> # 2nd is header = if T or TRUE means csv file has name of columns in it else not.

> data = read.csv("cereal.csv", header=T)

> # Cereal data

> print(data)

name mfr type calories protein fat sodium fiber carbo sugars potass vitamins shelf weight cups rating

1 100% Bran N C 70 4 1 130 10.0 5.0 6 280 25 3 1.00 0.33 68.402973

2 100% Natural Bran Q C 120 3 5 15 2.0 8.0 8 135 0 3 1.00 1.00 33.983679

3 All-Bran K C 70 4 1 260 9.0 7.0 5 320 25 3 1.00 0.33 59.425505

4 All-Bran with Extra Fiber K C 50 4 0 140 14.0 8.0 0 330 25 3 1.00 0.50 93.704912

5 Almond Delight R C 110 2 2 200 1.0 14.0 8 -1 25 3 1.00 0.75 34.384843

6 Apple Cinnamon Cheerios G C 110 2 2 180 1.5 10.5 10 70 25 1 1.00 0.75 29.509541

7 Apple Jacks K C 110 2 0 125 1.0 11.0 14 30 25 2 1.00 1.00 33.174094

8 Basic 4 G C 130 3 2 210 2.0 18.0 8 100 25 3 1.33 0.75 37.038562

9 Bran Chex R C 90 2 1 200 4.0 15.0 6 125 25 1 1.00 0.67 49.120253

10 Bran Flakes P C 90 3 0 210 5.0 13.0 5 190 25 3 1.00 0.67 53.313813

11 Cap'n'Crunch Q C 120 1 2 220 0.0 12.0 12 35 25 2 1.00 0.75 18.042851

12 Cheerios G C 110 6 2 290 2.0 17.0 1 105 25 1 1.00 1.25 50.764999

13 Cinnamon Toast Crunch G C 120 1 3 210 0.0 13.0 9 45 25 2 1.00 0.75 19.823573

14 Clusters G C 110 3 2 140 2.0 13.0 7 105 25 3 1.00 0.50 40.400208

15 Cocoa Puffs G C 110 1 1 180 0.0 12.0 13 55 25 2 1.00 1.00 22.736446

16 Corn Chex R C 110 2 0 280 0.0 22.0 3 25 25 1 1.00 1.00 41.445019

17 Corn Flakes K C 100 2 0 290 1.0 21.0 2 35 25 1 1.00 1.00 45.863324

18 Corn Pops K C 110 1 0 90 1.0 13.0 12 20 25 2 1.00 1.00 35.782791

19 Count Chocula G C 110 1 1 180 0.0 12.0 13 65 25 2 1.00 1.00 22.396513

20 Cracklin' Oat Bran K C 110 3 3 140 4.0 10.0 7 160 25 3 1.00 0.50 40.448772

21 Cream of Wheat (Quick) N H 100 3 0 80 1.0 21.0 0 -1 0 2 1.00 1.00 64.533816

22 Crispix K C 110 2 0 220 1.0 21.0 3 30 25 3 1.00 1.00 46.895644

23 Crispy Wheat & Raisins G C 100 2 1 140 2.0 11.0 10 120 25 3 1.00 0.75 36.176196

24 Double Chex R C 100 2 0 190 1.0 18.0 5 80 25 3 1.00 0.75 44.330856

25 Froot Loops K C 110 2 1 125 1.0 11.0 13 30 25 2 1.00 1.00 32.207582

26 Frosted Flakes K C 110 1 0 200 1.0 14.0 11 25 25 1 1.00 0.75 31.435973

27 Frosted Mini-Wheats K C 100 3 0 0 3.0 14.0 7 100 25 2 1.00 0.80 58.345141

28 Fruit & Fibre Dates; Walnuts; and Oats P C 120 3 2 160 5.0 12.0 10 200 25 3 1.25 0.67 40.917047

29 Fruitful Bran K C 120 3 0 240 5.0 14.0 12 190 25 3 1.33 0.67 41.015492

30 Fruity Pebbles P C 110 1 1 135 0.0 13.0 12 25 25 2 1.00 0.75 28.025765

31 Golden Crisp P C 100 2 0 45 0.0 11.0 15 40 25 1 1.00 0.88 35.252444

32 Golden Grahams G C 110 1 1 280 0.0 15.0 9 45 25 2 1.00 0.75 23.804043

33 Grape Nuts Flakes P C 100 3 1 140 3.0 15.0 5 85 25 3 1.00 0.88 52.076897

34 Grape-Nuts P C 110 3 0 170 3.0 17.0 3 90 25 3 1.00 0.25 53.371007

35 Great Grains Pecan P C 120 3 3 75 3.0 13.0 4 100 25 3 1.00 0.33 45.811716

36 Honey Graham Ohs Q C 120 1 2 220 1.0 12.0 11 45 25 2 1.00 1.00 21.871292

37 Honey Nut Cheerios G C 110 3 1 250 1.5 11.5 10 90 25 1 1.00 0.75 31.072217

38 Honey-comb P C 110 1 0 180 0.0 14.0 11 35 25 1 1.00 1.33 28.742414

39 Just Right Crunchy Nuggets K C 110 2 1 170 1.0 17.0 6 60 100 3 1.00 1.00 36.523683

40 Just Right Fruit & Nut K C 140 3 1 170 2.0 20.0 9 95 100 3 1.30 0.75 36.471512

41 Kix G C 110 2 1 260 0.0 21.0 3 40 25 2 1.00 1.50 39.241114

42 Life Q C 100 4 2 150 2.0 12.0 6 95 25 2 1.00 0.67 45.328074

43 Lucky Charms G C 110 2 1 180 0.0 12.0 12 55 25 2 1.00 1.00 26.734515

44 Maypo A H 100 4 1 0 0.0 16.0 3 95 25 2 1.00 1.00 54.850917

45 Muesli Raisins; Dates; & Almonds R C 150 4 3 95 3.0 16.0 11 170 25 3 1.00 1.00 37.136863

46 Muesli Raisins; Peaches; & Pecans R C 150 4 3 150 3.0 16.0 11 170 25 3 1.00 1.00 34.139765

47 Mueslix Crispy Blend K C 160 3 2 150 3.0 17.0 13 160 25 3 1.50 0.67 30.313351

48 Multi-Grain Cheerios G C 100 2 1 220 2.0 15.0 6 90 25 1 1.00 1.00 40.105965

49 Nut&Honey Crunch K C 120 2 1 190 0.0 15.0 9 40 25 2 1.00 0.67 29.924285

50 Nutri-Grain Almond-Raisin K C 140 3 2 220 3.0 21.0 7 130 25 3 1.33 0.67 40.692320

51 Nutri-grain Wheat K C 90 3 0 170 3.0 18.0 2 90 25 3 1.00 1.00 59.642837

52 Oatmeal Raisin Crisp G C 130 3 2 170 1.5 13.5 10 120 25 3 1.25 0.50 30.450843

53 Post Nat. Raisin Bran P C 120 3 1 200 6.0 11.0 14 260 25 3 1.33 0.67 37.840594

54 Product 19 K C 100 3 0 320 1.0 20.0 3 45 100 3 1.00 1.00 41.503540

55 Puffed Rice Q C 50 1 0 0 0.0 13.0 0 15 0 3 0.50 1.00 60.756112

56 Puffed Wheat Q C 50 2 0 0 1.0 10.0 0 50 0 3 0.50 1.00 63.005645

57 Quaker Oat Squares Q C 100 4 1 135 2.0 14.0 6 110 25 3 1.00 0.50 49.511874

58 Quaker Oatmeal Q H 100 5 2 0 2.7 -1.0 -1 110 0 1 1.00 0.67 50.828392

59 Raisin Bran K C 120 3 1 210 5.0 14.0 12 240 25 2 1.33 0.75 39.259197

60 Raisin Nut Bran G C 100 3 2 140 2.5 10.5 8 140 25 3 1.00 0.50 39.703400

61 Raisin Squares K C 90 2 0 0 2.0 15.0 6 110 25 3 1.00 0.50 55.333142

62 Rice Chex R C 110 1 0 240 0.0 23.0 2 30 25 1 1.00 1.13 41.998933

[ reached getOption("max.print") -- omitted 15 rows ]

> # Sampling the data size 75% of total data is use for training

> # Remaining 25% of data will be used for testing.

> # nrow(data) returns the number of rows present in the data.

> samplesize = 0.75 \* nrow(data)

> # set the initial value of seed to 80 which helps in giving same output each time we run the program.

> set.seed(80)

> # using the below line we store the only numeric values in the data variable.

> # we use sapply function which takes first parameter as a vector or an object,

> # second argument is function we taken as is.numeric which will only returns numeric values.

> data = data[, sapply(data, is.numeric)]

> data

calories protein fat sodium fiber carbo sugars potass vitamins shelf weight cups rating

1 70 4 1 130 10.0 5.0 6 280 25 3 1.00 0.33 68.402973

2 120 3 5 15 2.0 8.0 8 135 0 3 1.00 1.00 33.983679

3 70 4 1 260 9.0 7.0 5 320 25 3 1.00 0.33 59.425505

4 50 4 0 140 14.0 8.0 0 330 25 3 1.00 0.50 93.704912

5 110 2 2 200 1.0 14.0 8 -1 25 3 1.00 0.75 34.384843

6 110 2 2 180 1.5 10.5 10 70 25 1 1.00 0.75 29.509541

7 110 2 0 125 1.0 11.0 14 30 25 2 1.00 1.00 33.174094

8 130 3 2 210 2.0 18.0 8 100 25 3 1.33 0.75 37.038562

9 90 2 1 200 4.0 15.0 6 125 25 1 1.00 0.67 49.120253

10 90 3 0 210 5.0 13.0 5 190 25 3 1.00 0.67 53.313813

11 120 1 2 220 0.0 12.0 12 35 25 2 1.00 0.75 18.042851

12 110 6 2 290 2.0 17.0 1 105 25 1 1.00 1.25 50.764999

13 120 1 3 210 0.0 13.0 9 45 25 2 1.00 0.75 19.823573

14 110 3 2 140 2.0 13.0 7 105 25 3 1.00 0.50 40.400208

15 110 1 1 180 0.0 12.0 13 55 25 2 1.00 1.00 22.736446

16 110 2 0 280 0.0 22.0 3 25 25 1 1.00 1.00 41.445019

17 100 2 0 290 1.0 21.0 2 35 25 1 1.00 1.00 45.863324

18 110 1 0 90 1.0 13.0 12 20 25 2 1.00 1.00 35.782791

19 110 1 1 180 0.0 12.0 13 65 25 2 1.00 1.00 22.396513

20 110 3 3 140 4.0 10.0 7 160 25 3 1.00 0.50 40.448772

21 100 3 0 80 1.0 21.0 0 -1 0 2 1.00 1.00 64.533816

22 110 2 0 220 1.0 21.0 3 30 25 3 1.00 1.00 46.895644

23 100 2 1 140 2.0 11.0 10 120 25 3 1.00 0.75 36.176196

24 100 2 0 190 1.0 18.0 5 80 25 3 1.00 0.75 44.330856

25 110 2 1 125 1.0 11.0 13 30 25 2 1.00 1.00 32.207582

26 110 1 0 200 1.0 14.0 11 25 25 1 1.00 0.75 31.435973

27 100 3 0 0 3.0 14.0 7 100 25 2 1.00 0.80 58.345141

28 120 3 2 160 5.0 12.0 10 200 25 3 1.25 0.67 40.917047

29 120 3 0 240 5.0 14.0 12 190 25 3 1.33 0.67 41.015492

30 110 1 1 135 0.0 13.0 12 25 25 2 1.00 0.75 28.025765

31 100 2 0 45 0.0 11.0 15 40 25 1 1.00 0.88 35.252444

32 110 1 1 280 0.0 15.0 9 45 25 2 1.00 0.75 23.804043

33 100 3 1 140 3.0 15.0 5 85 25 3 1.00 0.88 52.076897

34 110 3 0 170 3.0 17.0 3 90 25 3 1.00 0.25 53.371007

35 120 3 3 75 3.0 13.0 4 100 25 3 1.00 0.33 45.811716

36 120 1 2 220 1.0 12.0 11 45 25 2 1.00 1.00 21.871292

37 110 3 1 250 1.5 11.5 10 90 25 1 1.00 0.75 31.072217

38 110 1 0 180 0.0 14.0 11 35 25 1 1.00 1.33 28.742414

39 110 2 1 170 1.0 17.0 6 60 100 3 1.00 1.00 36.523683

40 140 3 1 170 2.0 20.0 9 95 100 3 1.30 0.75 36.471512

41 110 2 1 260 0.0 21.0 3 40 25 2 1.00 1.50 39.241114

42 100 4 2 150 2.0 12.0 6 95 25 2 1.00 0.67 45.328074

43 110 2 1 180 0.0 12.0 12 55 25 2 1.00 1.00 26.734515

44 100 4 1 0 0.0 16.0 3 95 25 2 1.00 1.00 54.850917

45 150 4 3 95 3.0 16.0 11 170 25 3 1.00 1.00 37.136863

46 150 4 3 150 3.0 16.0 11 170 25 3 1.00 1.00 34.139765

47 160 3 2 150 3.0 17.0 13 160 25 3 1.50 0.67 30.313351

48 100 2 1 220 2.0 15.0 6 90 25 1 1.00 1.00 40.105965

49 120 2 1 190 0.0 15.0 9 40 25 2 1.00 0.67 29.924285

50 140 3 2 220 3.0 21.0 7 130 25 3 1.33 0.67 40.692320

51 90 3 0 170 3.0 18.0 2 90 25 3 1.00 1.00 59.642837

52 130 3 2 170 1.5 13.5 10 120 25 3 1.25 0.50 30.450843

53 120 3 1 200 6.0 11.0 14 260 25 3 1.33 0.67 37.840594

54 100 3 0 320 1.0 20.0 3 45 100 3 1.00 1.00 41.503540

55 50 1 0 0 0.0 13.0 0 15 0 3 0.50 1.00 60.756112

56 50 2 0 0 1.0 10.0 0 50 0 3 0.50 1.00 63.005645

57 100 4 1 135 2.0 14.0 6 110 25 3 1.00 0.50 49.511874

58 100 5 2 0 2.7 -1.0 -1 110 0 1 1.00 0.67 50.828392

59 120 3 1 210 5.0 14.0 12 240 25 2 1.33 0.75 39.259197

60 100 3 2 140 2.5 10.5 8 140 25 3 1.00 0.50 39.703400

61 90 2 0 0 2.0 15.0 6 110 25 3 1.00 0.50 55.333142

62 110 1 0 240 0.0 23.0 2 30 25 1 1.00 1.13 41.998933

63 110 2 0 290 0.0 22.0 3 35 25 1 1.00 1.00 40.560159

64 80 2 0 0 3.0 16.0 0 95 0 1 0.83 1.00 68.235885

65 90 3 0 0 4.0 19.0 0 140 0 1 1.00 0.67 74.472949

66 90 3 0 0 3.0 20.0 0 120 0 1 1.00 0.67 72.801787

67 110 2 1 70 1.0 9.0 15 40 25 2 1.00 0.75 31.230054

68 110 6 0 230 1.0 16.0 3 55 25 1 1.00 1.00 53.131324

69 90 2 0 15 3.0 15.0 5 90 25 2 1.00 1.00 59.363993

70 110 2 1 200 0.0 21.0 3 35 100 3 1.00 1.00 38.839746

71 140 3 1 190 4.0 15.0 14 230 100 3 1.50 1.00 28.592785

72 100 3 1 200 3.0 16.0 3 110 100 3 1.00 1.00 46.658844

73 110 2 1 250 0.0 21.0 3 60 25 3 1.00 0.75 39.106174

74 110 1 1 140 0.0 13.0 12 25 25 2 1.00 1.00 27.753301

75 100 3 1 230 3.0 17.0 3 115 25 1 1.00 0.67 49.787445

76 100 3 1 200 3.0 17.0 3 110 25 1 1.00 1.00 51.592193

[ reached getOption("max.print") -- omitted 1 row ]

> # index variable stores the random indexes using sample function which choose the index randomly.

> # sample function takes number as first parameter, the seq\_len function creates the sequence,

> # second parameter is size, the number of indexes to choose from the given number.

> index = sample(seq\_len(nrow(data)), size = samplesize)

> # max variable stores the maximum values over the columns.

> # apply function takes first parameter as array or matrix,

> # second parameter is MARGIN = if the value is 1 then the it applies over the rows and if the value is 2 then it applies over the columns.

> # third parameter is function which we want to apply, here we use max function so it takes maximum value of each column.

> maxs = apply(data , 2 , max)

> # min variable stores the minimum values over the columns.

> # apply function takes first parameter as array or matrix,

> # second parameter is MARGIN = if the value is 1 then the it applies over the rows and if the value is 2 then it applies over the columns.

> # third parameter is function which we want to apply, here we use min function so it takes minimum value of each column.

> mins = apply(data, 2 , min)

> # scaled is dataframe which stores the scaled data.

> # scale function takes first parameter as a matrix, second parameter is center takes logical value or numeric vector of length equal to number of columns first argumnet matrix,

> # third parameter is scale, it takes logical or numeric vector of length equal to

> scaled = as.data.frame(scale(data, center = mins, scale = maxs - mins))

> #loading library neuralnet

> library('neuralnet')

> #prepare train and test data

> trainNN = scaled[index,]

> testNN = scaled[-index,]

> # neuralnet function is used for training the neural network.

> # the function takes first parameter as formula, it is a symbolic description of the model which we are going to fit,

> # second parameter is the data for training the neural network,

> # third parameter is hidden which is a number of hidden neurons we want

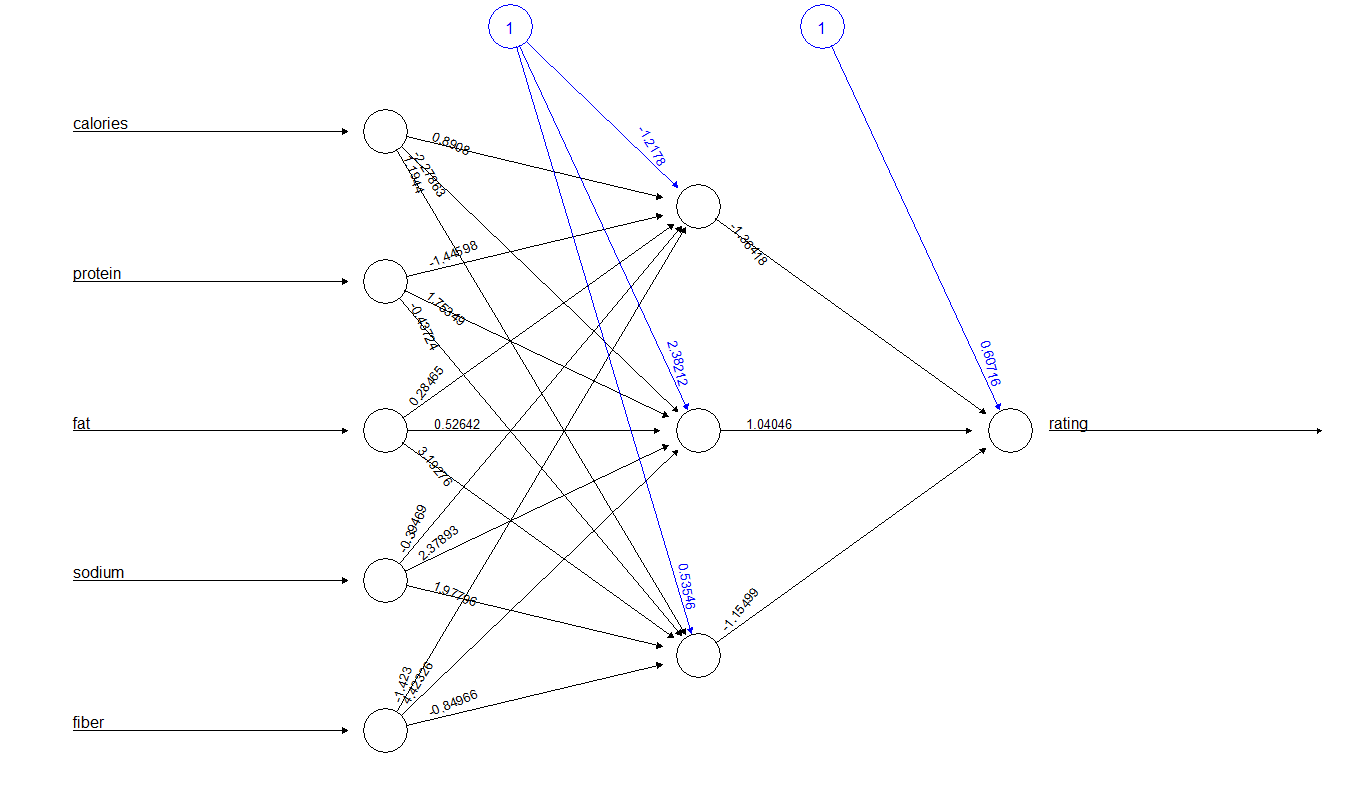
> # fourth parameter is linear.output it takes logical values TRUE or False if differentiable function shoud not be applied to the output node then give TRUE as linear.output value otherwise FALSE.

> NN = neuralnet(rating ~ calories + protein + fat + sodium + fiber, trainNN, hidden = 3 ,

+ linear.output = T )

> # ploting Neural Network

> plot(NN)



**Name: Sanesh Sagvekar Roll No: 170122**

**Practical No.: 2**

**AIM: Implement Self Organizing Map neural network.**

> # loading kohonen

> library("kohonen")

>

> # loading iris dataset

> data(iris)

>

> # storing all rows and first four columns from dataset to data variable.

> data = (iris[, 1:4])

> print(data)

Sepal.Length Sepal.Width Petal.Length Petal.Width

1 5.1 3.5 1.4 0.2

2 4.9 3.0 1.4 0.2

3 4.7 3.2 1.3 0.2

4 4.6 3.1 1.5 0.2

5 5.0 3.6 1.4 0.2

6 5.4 3.9 1.7 0.4

7 4.6 3.4 1.4 0.3

8 5.0 3.4 1.5 0.2

9 4.4 2.9 1.4 0.2

10 4.9 3.1 1.5 0.1

11 5.4 3.7 1.5 0.2

12 4.8 3.4 1.6 0.2

13 4.8 3.0 1.4 0.1

14 4.3 3.0 1.1 0.1

15 5.8 4.0 1.2 0.2

16 5.7 4.4 1.5 0.4

17 5.4 3.9 1.3 0.4

18 5.1 3.5 1.4 0.3

19 5.7 3.8 1.7 0.3

20 5.1 3.8 1.5 0.3

21 5.4 3.4 1.7 0.2

22 5.1 3.7 1.5 0.4

23 4.6 3.6 1.0 0.2

24 5.1 3.3 1.7 0.5

25 4.8 3.4 1.9 0.2

26 5.0 3.0 1.6 0.2

27 5.0 3.4 1.6 0.4

28 5.2 3.5 1.5 0.2

29 5.2 3.4 1.4 0.2

30 4.7 3.2 1.6 0.2

31 4.8 3.1 1.6 0.2

32 5.4 3.4 1.5 0.4

33 5.2 4.1 1.5 0.1

34 5.5 4.2 1.4 0.2

35 4.9 3.1 1.5 0.2

36 5.0 3.2 1.2 0.2

37 5.5 3.5 1.3 0.2

38 4.9 3.6 1.4 0.1

39 4.4 3.0 1.3 0.2

40 5.1 3.4 1.5 0.2

41 5.0 3.5 1.3 0.3

42 4.5 2.3 1.3 0.3

43 4.4 3.2 1.3 0.2

44 5.0 3.5 1.6 0.6

45 5.1 3.8 1.9 0.4

46 4.8 3.0 1.4 0.3

47 5.1 3.8 1.6 0.2

48 4.6 3.2 1.4 0.2

49 5.3 3.7 1.5 0.2

50 5.0 3.3 1.4 0.2

51 7.0 3.2 4.7 1.4

52 6.4 3.2 4.5 1.5

53 6.9 3.1 4.9 1.5

54 5.5 2.3 4.0 1.3

55 6.5 2.8 4.6 1.5

56 5.7 2.8 4.5 1.3

57 6.3 3.3 4.7 1.6

58 4.9 2.4 3.3 1.0

59 6.6 2.9 4.6 1.3

60 5.2 2.7 3.9 1.4

61 5.0 2.0 3.5 1.0

62 5.9 3.0 4.2 1.5

63 6.0 2.2 4.0 1.0

64 6.1 2.9 4.7 1.4

65 5.6 2.9 3.6 1.3

66 6.7 3.1 4.4 1.4

67 5.6 3.0 4.5 1.5

68 5.8 2.7 4.1 1.0

69 6.2 2.2 4.5 1.5

70 5.6 2.5 3.9 1.1

71 5.9 3.2 4.8 1.8

72 6.1 2.8 4.0 1.3

73 6.3 2.5 4.9 1.5

74 6.1 2.8 4.7 1.2

75 6.4 2.9 4.3 1.3

76 6.6 3.0 4.4 1.4

77 6.8 2.8 4.8 1.4

78 6.7 3.0 5.0 1.7

79 6.0 2.9 4.5 1.5

80 5.7 2.6 3.5 1.0

81 5.5 2.4 3.8 1.1

82 5.5 2.4 3.7 1.0

83 5.8 2.7 3.9 1.2

84 6.0 2.7 5.1 1.6

85 5.4 3.0 4.5 1.5

86 6.0 3.4 4.5 1.6

87 6.7 3.1 4.7 1.5

88 6.3 2.3 4.4 1.3

89 5.6 3.0 4.1 1.3

90 5.5 2.5 4.0 1.3

91 5.5 2.6 4.4 1.2

92 6.1 3.0 4.6 1.4

93 5.8 2.6 4.0 1.2

94 5.0 2.3 3.3 1.0

95 5.6 2.7 4.2 1.3

96 5.7 3.0 4.2 1.2

97 5.7 2.9 4.2 1.3

98 6.2 2.9 4.3 1.3

99 5.1 2.5 3.0 1.1

100 5.7 2.8 4.1 1.3

101 6.3 3.3 6.0 2.5

102 5.8 2.7 5.1 1.9

103 7.1 3.0 5.9 2.1

104 6.3 2.9 5.6 1.8

105 6.5 3.0 5.8 2.2

106 7.6 3.0 6.6 2.1

107 4.9 2.5 4.5 1.7

108 7.3 2.9 6.3 1.8

109 6.7 2.5 5.8 1.8

110 7.2 3.6 6.1 2.5

111 6.5 3.2 5.1 2.0

112 6.4 2.7 5.3 1.9

113 6.8 3.0 5.5 2.1

114 5.7 2.5 5.0 2.0

115 5.8 2.8 5.1 2.4

116 6.4 3.2 5.3 2.3

117 6.5 3.0 5.5 1.8

118 7.7 3.8 6.7 2.2

119 7.7 2.6 6.9 2.3

120 6.0 2.2 5.0 1.5

121 6.9 3.2 5.7 2.3

122 5.6 2.8 4.9 2.0

123 7.7 2.8 6.7 2.0

124 6.3 2.7 4.9 1.8

125 6.7 3.3 5.7 2.1

126 7.2 3.2 6.0 1.8

127 6.2 2.8 4.8 1.8

128 6.1 3.0 4.9 1.8

129 6.4 2.8 5.6 2.1

130 7.2 3.0 5.8 1.6

131 7.4 2.8 6.1 1.9

132 7.9 3.8 6.4 2.0

133 6.4 2.8 5.6 2.2

134 6.3 2.8 5.1 1.5

135 6.1 2.6 5.6 1.4

136 7.7 3.0 6.1 2.3

137 6.3 3.4 5.6 2.4

138 6.4 3.1 5.5 1.8

139 6.0 3.0 4.8 1.8

140 6.9 3.1 5.4 2.1

141 6.7 3.1 5.6 2.4

142 6.9 3.1 5.1 2.3

143 5.8 2.7 5.1 1.9

144 6.8 3.2 5.9 2.3

145 6.7 3.3 5.7 2.5

146 6.7 3.0 5.2 2.3

147 6.3 2.5 5.0 1.9

148 6.5 3.0 5.2 2.0

149 6.2 3.4 5.4 2.3

150 5.9 3.0 5.1 1.8

> # scaled the data

> # scale function is use for scaling the data, parameter it has taken is a numeric matrix.

> scaled = scale(data)

> print(scaled)

Sepal.Length Sepal.Width Petal.Length Petal.Width

[1,] -0.89767387920 1.0156019907 -1.33575163424 -1.311052148205

[2,] -1.13920048346 -0.1315388121 -1.33575163424 -1.311052148205

[3,] -1.38072708773 0.3273175091 -1.39239928624 -1.311052148205

[4,] -1.50149038987 0.0978893485 -1.27910398224 -1.311052148205

[5,] -1.01843718133 1.2450301513 -1.33575163424 -1.311052148205

[6,] -0.53538397279 1.9333146329 -1.16580867823 -1.048666794995

[7,] -1.50149038987 0.7861738302 -1.33575163424 -1.179859471600

[8,] -1.01843718133 0.7861738302 -1.27910398224 -1.311052148205

[9,] -1.74301699414 -0.3609669726 -1.33575163424 -1.311052148205

[10,] -1.13920048346 0.0978893485 -1.27910398224 -1.442244824810

[11,] -0.53538397279 1.4744583118 -1.27910398224 -1.311052148205

[12,] -1.25996378560 0.7861738302 -1.22245633023 -1.311052148205

[13,] -1.25996378560 -0.1315388121 -1.33575163424 -1.442244824810

[14,] -1.86378029627 -0.1315388121 -1.50569459025 -1.442244824810

[15,] -0.05233076426 2.1627427935 -1.44904693825 -1.311052148205

[16,] -0.17309406639 3.0804554357 -1.27910398224 -1.048666794995

[17,] -0.53538397279 1.9333146329 -1.39239928624 -1.048666794995

[18,] -0.89767387920 1.0156019907 -1.33575163424 -1.179859471600

[19,] -0.17309406639 1.7038864724 -1.16580867823 -1.179859471600

[20,] -0.89767387920 1.7038864724 -1.27910398224 -1.179859471600

[21,] -0.53538397279 0.7861738302 -1.16580867823 -1.311052148205

[22,] -0.89767387920 1.4744583118 -1.27910398224 -1.048666794995

[23,] -1.50149038987 1.2450301513 -1.56234224226 -1.311052148205

[24,] -0.89767387920 0.5567456696 -1.16580867823 -0.917474118390

[25,] -1.25996378560 0.7861738302 -1.05251337422 -1.311052148205

[26,] -1.01843718133 -0.1315388121 -1.22245633023 -1.311052148205

[27,] -1.01843718133 0.7861738302 -1.22245633023 -1.048666794995

[28,] -0.77691057706 1.0156019907 -1.27910398224 -1.311052148205

[29,] -0.77691057706 0.7861738302 -1.33575163424 -1.311052148205

[30,] -1.38072708773 0.3273175091 -1.22245633023 -1.311052148205

[31,] -1.25996378560 0.0978893485 -1.22245633023 -1.311052148205

[32,] -0.53538397279 0.7861738302 -1.27910398224 -1.048666794995

[33,] -0.77691057706 2.3921709540 -1.27910398224 -1.442244824810

[34,] -0.41462067066 2.6215991146 -1.33575163424 -1.311052148205

[35,] -1.13920048346 0.0978893485 -1.27910398224 -1.311052148205

[36,] -1.01843718133 0.3273175091 -1.44904693825 -1.311052148205

[37,] -0.41462067066 1.0156019907 -1.39239928624 -1.311052148205

[38,] -1.13920048346 1.2450301513 -1.33575163424 -1.442244824810

[39,] -1.74301699414 -0.1315388121 -1.39239928624 -1.311052148205

[40,] -0.89767387920 0.7861738302 -1.27910398224 -1.311052148205

[41,] -1.01843718133 1.0156019907 -1.39239928624 -1.179859471600

[42,] -1.62225369200 -1.7375359359 -1.39239928624 -1.179859471600

[43,] -1.74301699414 0.3273175091 -1.39239928624 -1.311052148205

[44,] -1.01843718133 1.0156019907 -1.22245633023 -0.786281441785

[45,] -0.89767387920 1.7038864724 -1.05251337422 -1.048666794995

[46,] -1.25996378560 -0.1315388121 -1.33575163424 -1.179859471600

[47,] -0.89767387920 1.7038864724 -1.22245633023 -1.311052148205

[48,] -1.50149038987 0.3273175091 -1.33575163424 -1.311052148205

[49,] -0.65614727493 1.4744583118 -1.27910398224 -1.311052148205

[50,] -1.01843718133 0.5567456696 -1.33575163424 -1.311052148205

[51,] 1.39682886135 0.3273175091 0.53362088187 0.263259971054

[52,] 0.67224904855 0.3273175091 0.42032557787 0.394452647659

[53,] 1.27606555922 0.0978893485 0.64691618588 0.394452647659

[54,] -0.41462067066 -1.7375359359 0.13708731785 0.132067294449

[55,] 0.79301235068 -0.5903951332 0.47697322987 0.394452647659

[56,] -0.17309406639 -0.5903951332 0.42032557787 0.132067294449

[57,] 0.55148574641 0.5567456696 0.53362088187 0.525645324264

[58,] -1.13920048346 -1.5081077754 -0.25944624618 -0.261510735366

[59,] 0.91377565281 -0.3609669726 0.47697322987 0.132067294449

[60,] -0.77691057706 -0.8198232937 0.08043966584 0.263259971054

[61,] -1.01843718133 -2.4258204176 -0.14615094217 -0.261510735366

[62,] 0.06843253788 -0.1315388121 0.25038262186 0.394452647659

[63,] 0.18919584001 -1.9669640965 0.13708731785 -0.261510735366

[64,] 0.30995914214 -0.3609669726 0.53362088187 0.263259971054

[65,] -0.29385736853 -0.3609669726 -0.08950329017 0.132067294449

[66,] 1.03453895495 0.0978893485 0.36367792586 0.263259971054

[67,] -0.29385736853 -0.1315388121 0.42032557787 0.394452647659

[68,] -0.05233076426 -0.8198232937 0.19373496985 -0.261510735366

[69,] 0.43072244428 -1.9669640965 0.42032557787 0.394452647659

[70,] -0.29385736853 -1.2786796148 0.08043966584 -0.130318058761

[71,] 0.06843253788 0.3273175091 0.59026853388 0.788030677474

[72,] 0.30995914214 -0.5903951332 0.13708731785 0.132067294449

[73,] 0.55148574641 -1.2786796148 0.64691618588 0.394452647659

[74,] 0.30995914214 -0.5903951332 0.53362088187 0.000874617844

[75,] 0.67224904855 -0.3609669726 0.30703027386 0.132067294449

[76,] 0.91377565281 -0.1315388121 0.36367792586 0.263259971054

[77,] 1.15530225708 -0.5903951332 0.59026853388 0.263259971054

[78,] 1.03453895495 -0.1315388121 0.70356383788 0.656838000869

[79,] 0.18919584001 -0.3609669726 0.42032557787 0.394452647659

[80,] -0.17309406639 -1.0492514543 -0.14615094217 -0.261510735366

[81,] -0.41462067066 -1.5081077754 0.02379201384 -0.130318058761

[82,] -0.41462067066 -1.5081077754 -0.03285563816 -0.261510735366

[83,] -0.05233076426 -0.8198232937 0.08043966584 0.000874617844

[84,] 0.18919584001 -0.8198232937 0.76021148989 0.525645324264

[85,] -0.53538397279 -0.1315388121 0.42032557787 0.394452647659

[86,] 0.18919584001 0.7861738302 0.42032557787 0.525645324264

[87,] 1.03453895495 0.0978893485 0.53362088187 0.394452647659

[88,] 0.55148574641 -1.7375359359 0.36367792586 0.132067294449

[89,] -0.29385736853 -0.1315388121 0.19373496985 0.132067294449

[90,] -0.41462067066 -1.2786796148 0.13708731785 0.132067294449

[91,] -0.41462067066 -1.0492514543 0.36367792586 0.000874617844

[92,] 0.30995914214 -0.1315388121 0.47697322987 0.263259971054

[93,] -0.05233076426 -1.0492514543 0.13708731785 0.000874617844

[94,] -1.01843718133 -1.7375359359 -0.25944624618 -0.261510735366

[95,] -0.29385736853 -0.8198232937 0.25038262186 0.132067294449

[96,] -0.17309406639 -0.1315388121 0.25038262186 0.000874617844

[97,] -0.17309406639 -0.3609669726 0.25038262186 0.132067294449

[98,] 0.43072244428 -0.3609669726 0.30703027386 0.132067294449

[99,] -0.89767387920 -1.2786796148 -0.42938920219 -0.130318058761

[100,] -0.17309406639 -0.5903951332 0.19373496985 0.132067294449

[101,] 0.55148574641 0.5567456696 1.27004035792 1.706379413708

[102,] -0.05233076426 -0.8198232937 0.76021148989 0.919223354078

[103,] 1.51759216349 -0.1315388121 1.21339270591 1.181608707288

[104,] 0.55148574641 -0.3609669726 1.04344974990 0.788030677474

[105,] 0.79301235068 -0.1315388121 1.15674505391 1.312801383893

[106,] 2.12140867416 -0.1315388121 1.60992626994 1.181608707288

[107,] -1.13920048346 -1.2786796148 0.42032557787 0.656838000869

[108,] 1.75911876775 -0.3609669726 1.43998331393 0.788030677474

[109,] 1.03453895495 -1.2786796148 1.15674505391 0.788030677474

[110,] 1.63835546562 1.2450301513 1.32668800992 1.706379413708

[111,] 0.79301235068 0.3273175091 0.76021148989 1.050416030683

[112,] 0.67224904855 -0.8198232937 0.87350679389 0.919223354078

[113,] 1.15530225708 -0.1315388121 0.98680209790 1.181608707288

[114,] -0.17309406639 -1.2786796148 0.70356383788 1.050416030683

[115,] -0.05233076426 -0.5903951332 0.76021148989 1.575186737103

[116,] 0.67224904855 0.3273175091 0.87350679389 1.443994060498

[117,] 0.79301235068 -0.1315388121 0.98680209790 0.788030677474

[118,] 2.24217197629 1.7038864724 1.66657392194 1.312801383893

[119,] 2.24217197629 -1.0492514543 1.77986922595 1.443994060498

[120,] 0.18919584001 -1.9669640965 0.70356383788 0.394452647659

[121,] 1.27606555922 0.3273175091 1.10009740191 1.443994060498

[122,] -0.29385736853 -0.5903951332 0.64691618588 1.050416030683

[123,] 2.24217197629 -0.5903951332 1.66657392194 1.050416030683

[124,] 0.55148574641 -0.8198232937 0.64691618588 0.788030677474

[125,] 1.03453895495 0.5567456696 1.10009740191 1.181608707288

[126,] 1.63835546562 0.3273175091 1.27004035792 0.788030677474

[127,] 0.43072244428 -0.5903951332 0.59026853388 0.788030677474

[128,] 0.30995914214 -0.1315388121 0.64691618588 0.788030677474

[129,] 0.67224904855 -0.5903951332 1.04344974990 1.181608707288

[130,] 1.63835546562 -0.1315388121 1.15674505391 0.525645324264

[131,] 1.87988206989 -0.5903951332 1.32668800992 0.919223354078

[132,] 2.48369858056 1.7038864724 1.49663096593 1.050416030683

[133,] 0.67224904855 -0.5903951332 1.04344974990 1.312801383893

[134,] 0.55148574641 -0.5903951332 0.76021148989 0.394452647659

[135,] 0.30995914214 -1.0492514543 1.04344974990 0.263259971054

[136,] 2.24217197629 -0.1315388121 1.32668800992 1.443994060498

[137,] 0.55148574641 0.7861738302 1.04344974990 1.575186737103

[138,] 0.67224904855 0.0978893485 0.98680209790 0.788030677474

[139,] 0.18919584001 -0.1315388121 0.59026853388 0.788030677474

[140,] 1.27606555922 0.0978893485 0.93015444590 1.181608707288

[141,] 1.03453895495 0.0978893485 1.04344974990 1.575186737103

[142,] 1.27606555922 0.0978893485 0.76021148989 1.443994060498

[143,] -0.05233076426 -0.8198232937 0.76021148989 0.919223354078

[144,] 1.15530225708 0.3273175091 1.21339270591 1.443994060498

[145,] 1.03453895495 0.5567456696 1.10009740191 1.706379413708

[146,] 1.03453895495 -0.1315388121 0.81685914189 1.443994060498

[147,] 0.55148574641 -1.2786796148 0.70356383788 0.919223354078

[148,] 0.79301235068 -0.1315388121 0.81685914189 1.050416030683

[149,] 0.43072244428 0.7861738302 0.93015444590 1.443994060498

[150,] 0.06843253788 -0.1315388121 0.76021148989 0.788030677474

attr(,"scaled:center")

Sepal.Length Sepal.Width Petal.Length Petal.Width

5.843333333 3.057333333 3.758000000 1.199333333

attr(,"scaled:scale")

Sepal.Length Sepal.Width Petal.Length Petal.Width

0.8280661280 0.4358662849 1.7652982333 0.7622376690

> # somgrid function use for plotting functions for self organizing map.

> # somgrid function takes 3 parameter first parameter is xdim which takes number of dimensions of grid,

> # second parameter is ydim, it also takes number of dimensions of the grid,

> # third parameter is topo which takes type of topology we want.

> grid = somgrid(xdim = 5, ydim = 5, topo = "hexagonal")

> # som function four parameter as an input, first parameter is data we passed the scaled which is a scaled data,

> # second parameter is grid which takes object of the class somgrid,

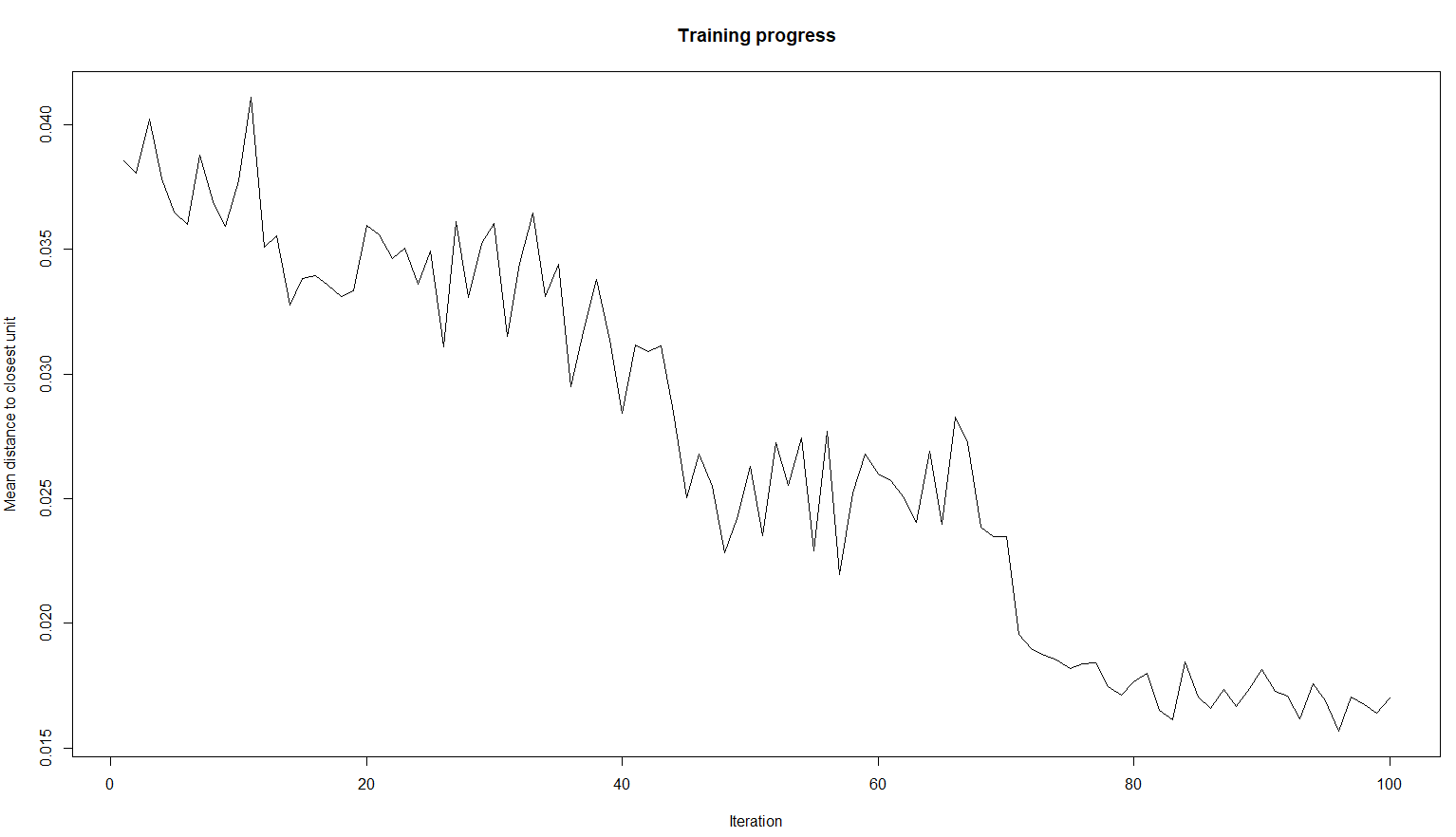
> # third parameter is rlen which takes the number of times the complete dataset we want to present to the network,

> # fourth parameter is alpha which is a learning rate of the network, two numbers indicating the amount of change.

> som = som(scaled, grid = grid, rlen = 100, alpha = c(0.05, 0.01))

> # it shows the progress over time.

> plot(som, type = "changes")

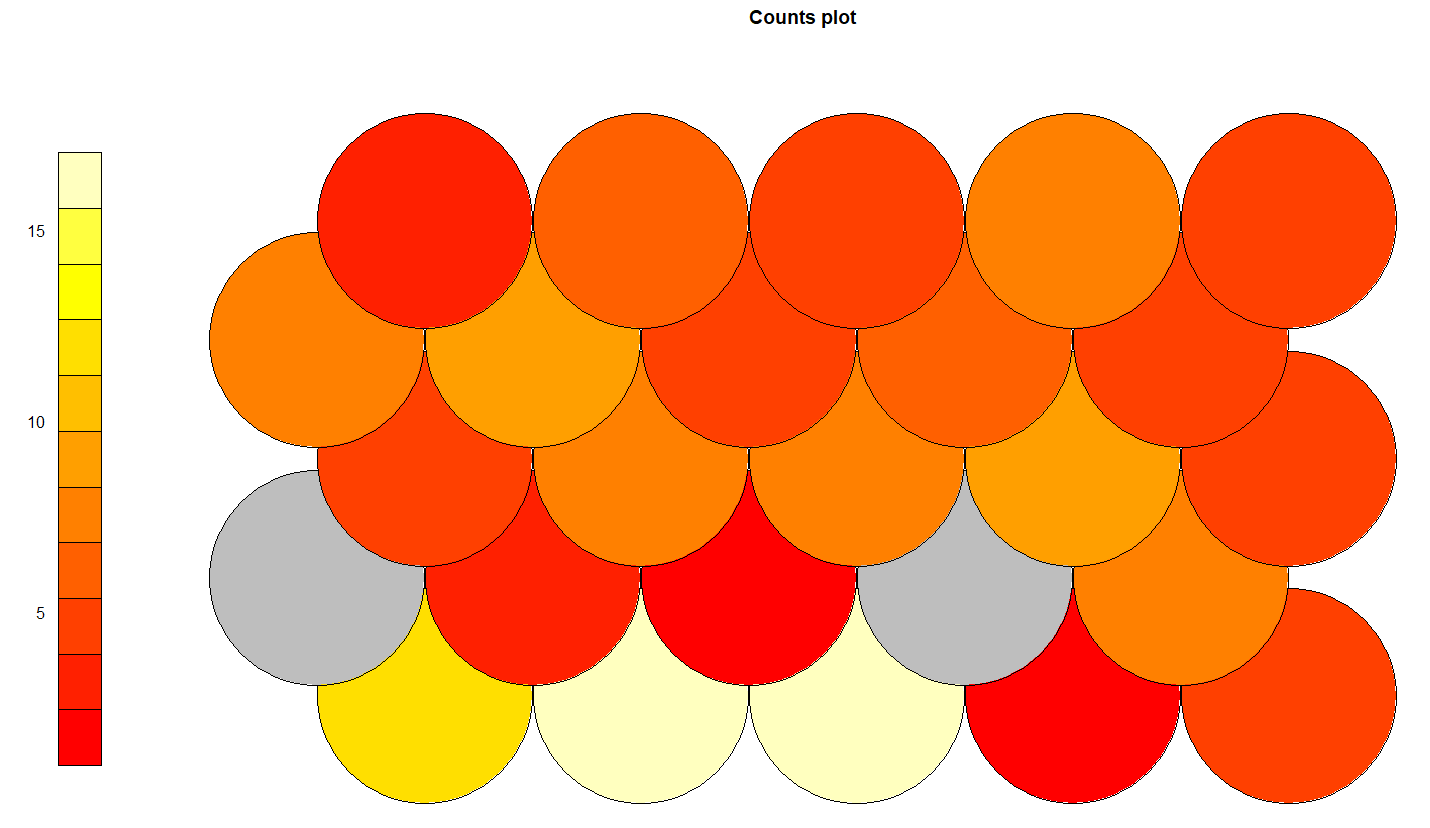


>

> # it shows how many samples are mapped to each node on the map, large values in the map areas would be more beneficial.

> plot(som, type = "count")

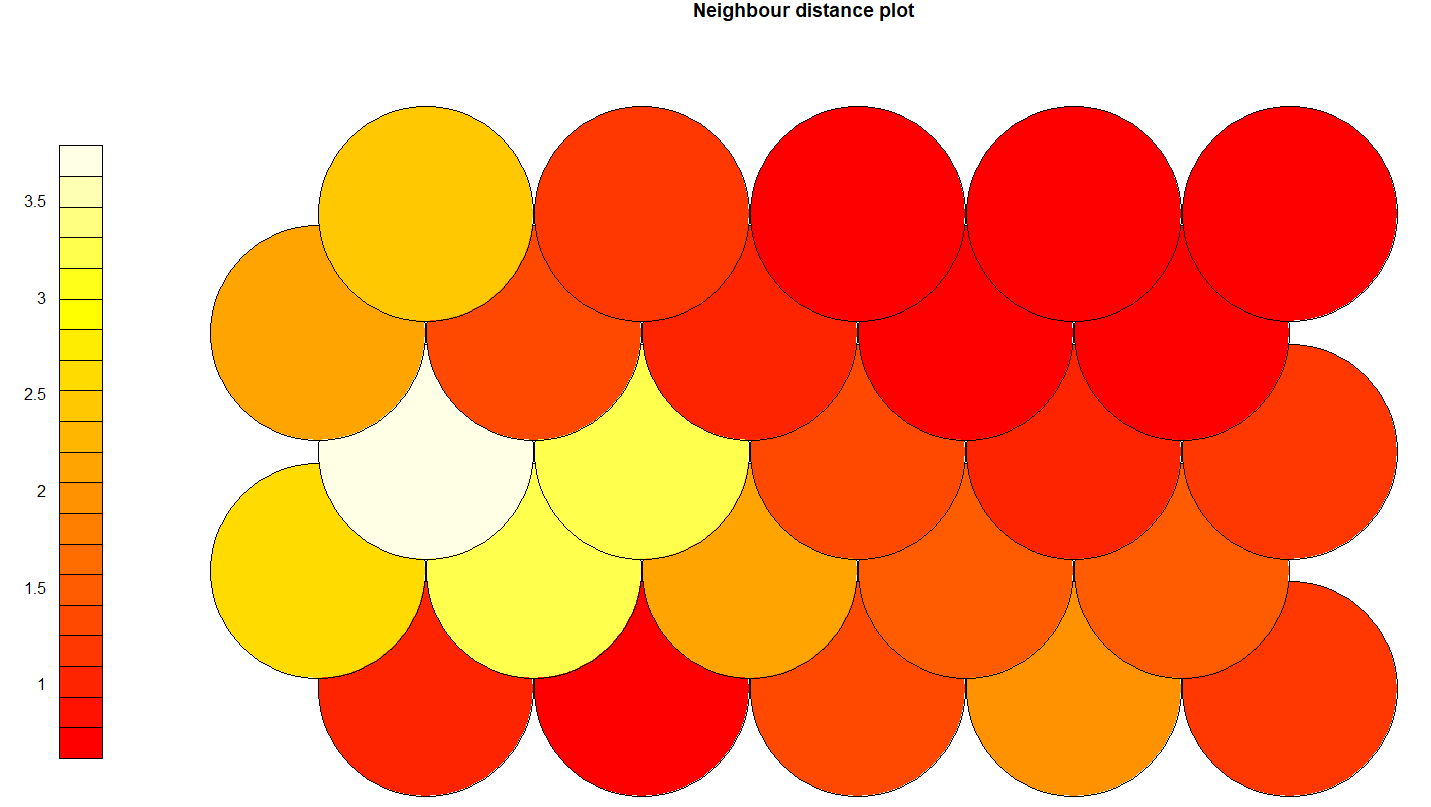
>



> # it shows the distance between each node and its neighbours.

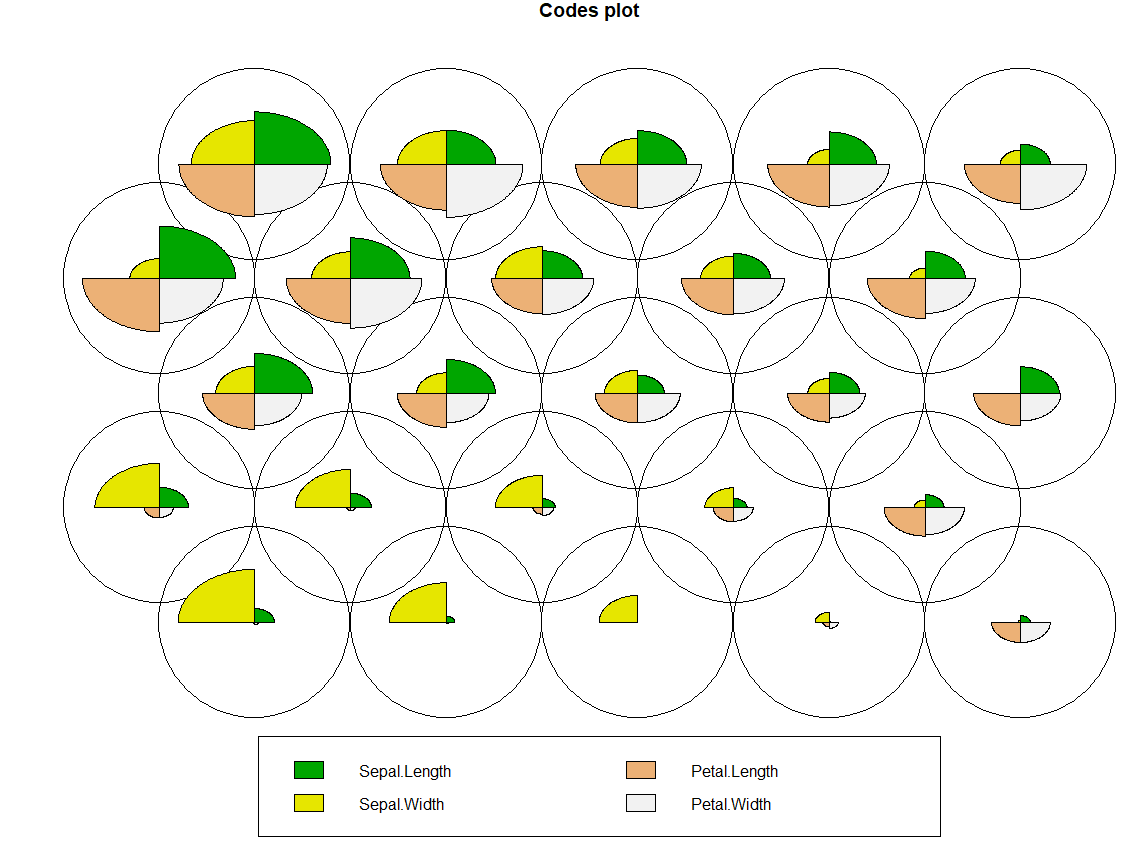
> plot(som, type="dist.neighbours")

>



> # it shows the each node's weight vector, by default it show fan diagram where individual fan representations of the magnitude of each variable in the weight vector is shown for each node.

> plot(som, type="codes")



**Name: Sanesh Sagvekar Roll No: 170122**

**Machine Learning**

**Practical No.: 3**

**AIM: Implement Radial Basis Function neural network with gradient descent.**

> #install.packages('Rcpp')

> # loading RSNNS

> library('RSNNS')

> # creating input data using seq function which takes three inputs.

> # first parameter is from, starting value, second parameter is to, end value,

> # third parameter is by how many steps we want to increase the the number.

> # we use as.matrix function to make a all values as matrix.

> inputs = as.matrix(seq(0, 10, 0.1))

> # for output each row it calculates sin value and runif generate random numbers, these two values are added.

> outputs = as.matrix(sin(inputs) + runif(inputs \* 0.2))

> # rbf function is seven parameters as input rbf function is used for creating radial basis function network.

> # first parameter is x which is a matrix of training inputs for the network,

> # second parameter is y which is a target values, third parameter is a size which is a number of hidden layers,

> # fourth parameter is maxit it is a maximum number of iterations to learn, fifth parameter is initFuncParams which takes parameters for initialization function,

> # sixth parameter is learnFuncParam which takes parameters for the learning function, seventh parameter is linOut it takes logical values TRUE or False to set activation function of the output units to linear or logistic.

> model = rbf(inputs, outputs, size = 40, maxit = 1000, initFuncParams = c(0, 1, 0, 0.01, 0.01), learnFuncParams=c(1e-8, 0, 1e-8, 0.1, 0.8), linOut = TRUE)

> # par function is used to set or query graphical parameters.

> # par function takes one parameter as mfrow, it takes nr and nc from which plot is drawn in an nr by nc array on device by rows.

> par(mfrow = c(2, 1))

> # plotIterativeError function is used for ploting iterative and test error of the network of rsnns object.

> plotIterativeError(model)

> # ploting graph

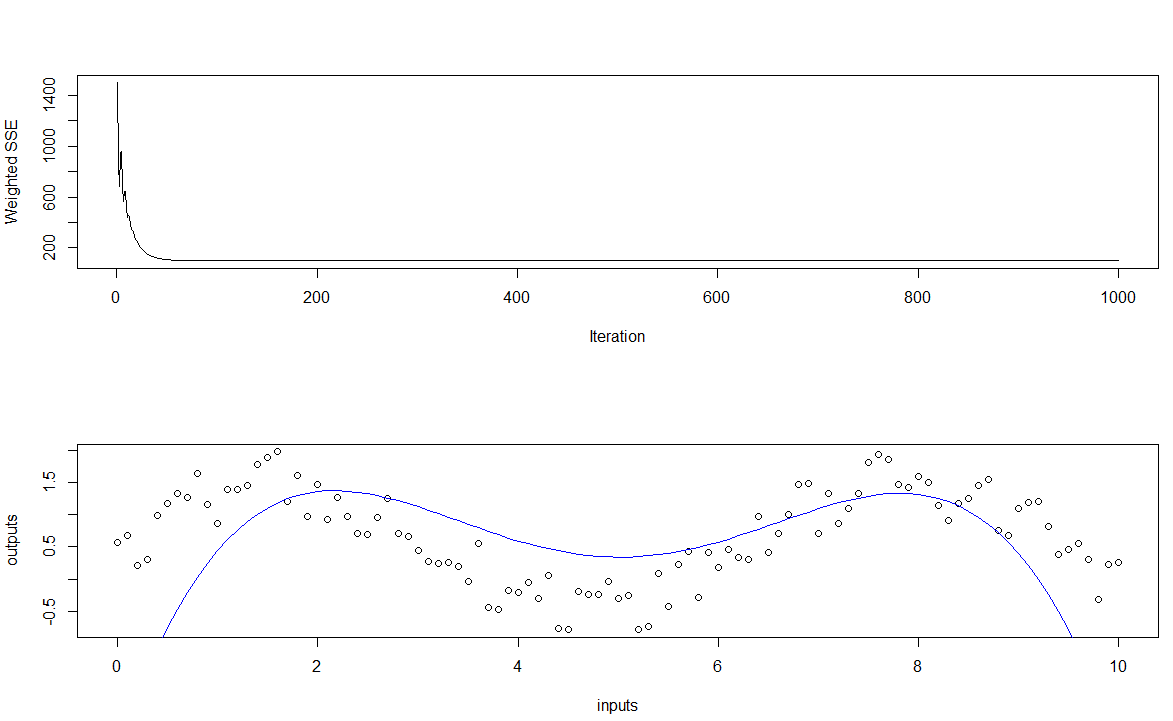
> plot(inputs, outputs)

> # lines function is used for joining the corresponding points with line segments.

> # lines function takes three parameter first two parameters are coordinate points to join,

> # third parameter is col which takes the name of color for the line

> lines(inputs, fitted(model), col = "blue")



**Name: - Sanesh Sagvekar Roll No: 170122**

**Practical No.: 4**

**AIM: Implement a basic genetic algorithm with selection, mutation and crossover as genetic operators.**

> #install.packages('GA')

> # loading GA library

> library(GA)

> ## Example01

> # f is function for which want to find the solution.

> f = function(x) (x^2 + x) \* cos(x)

> # lbound is the lower bound of the search space.

> lbound = -10

> # ubound is the upper bound of the search space.

> ubound = 10

> # curve function is used to draw a curve corresponding to function over the interval.

> # function takes four parameters, first parameter is the function for which we want to draw the curve,

> # second parameter is from it takes starting value, third parameter is to it takes ending value,

> # fourth parameter is n it takes the number of values at which to evaluate.

> curve(f, from = lbound, to = ubound, n = 1000)

> # ga function takes four parameters, first parameter is type it takes type of genetic algorithm to be run,

> # second parameter is fitness it takes the fitness function, third parameter is lower it takes the lower bound integer of the search space,

> # fourth parameter is upper it takes upper bound integer of the search space.

> GA = ga(type = "real-valued", fitness = f, lower = lbound, upper = ubound)

GA | iter = 1 | Mean = -4.145560361 | Best = 47.653749138

GA | iter = 2 | Mean = 9.395556588 | Best = 47.653749138

GA | iter = 3 | Mean = 23.00024845 | Best = 47.65374914

GA | iter = 4 | Mean = 25.74356654 | Best = 47.65374914

GA | iter = 5 | Mean = 42.51078012 | Best = 47.65374914

GA | iter = 6 | Mean = 40.37580439 | Best = 47.69881883

GA | iter = 7 | Mean = 40.44368496 | Best = 47.69881883

GA | iter = 8 | Mean = 43.18279033 | Best = 47.69881883

GA | iter = 9 | Mean = 39.00670767 | Best = 47.69881883

GA | iter = 10 | Mean = 34.95812714 | Best = 47.70562008

GA | iter = 11 | Mean = 40.24280872 | Best = 47.70562008

GA | iter = 12 | Mean = 39.77962726 | Best = 47.70562008

GA | iter = 13 | Mean = 43.34747477 | Best = 47.70562008

GA | iter = 14 | Mean = 35.60586167 | Best = 47.70562008

GA | iter = 15 | Mean = 36.34040117 | Best = 47.70562008

GA | iter = 16 | Mean = 41.62988813 | Best = 47.70562008

GA | iter = 17 | Mean = 43.65325314 | Best = 47.70562008

GA | iter = 18 | Mean = 42.40463482 | Best = 47.70562008

GA | iter = 19 | Mean = 45.61074301 | Best = 47.70562008

GA | iter = 20 | Mean = 45.53713556 | Best = 47.70562008

GA | iter = 21 | Mean = 40.34106572 | Best = 47.70562008

GA | iter = 22 | Mean = 39.71184654 | Best = 47.70562008

GA | iter = 23 | Mean = 39.29873838 | Best = 47.70562008

GA | iter = 24 | Mean = 38.08329901 | Best = 47.70562008

GA | iter = 25 | Mean = 32.76572916 | Best = 47.70562008

GA | iter = 26 | Mean = 22.87284143 | Best = 47.70562008

GA | iter = 27 | Mean = 27.55574687 | Best = 47.70562008

GA | iter = 28 | Mean = 30.11729261 | Best = 47.70562008

GA | iter = 29 | Mean = 36.53805714 | Best = 47.70562008

GA | iter = 30 | Mean = 39.52442715 | Best = 47.70562008

GA | iter = 31 | Mean = 32.13345991 | Best = 47.70562008

GA | iter = 32 | Mean = 35.13790636 | Best = 47.70562008

GA | iter = 33 | Mean = 38.52270446 | Best = 47.70562008

GA | iter = 34 | Mean = 29.19004582 | Best = 47.70562008

GA | iter = 35 | Mean = 39.35060159 | Best = 47.70562008

GA | iter = 36 | Mean = 39.11416809 | Best = 47.70562008

GA | iter = 37 | Mean = 38.86346129 | Best = 47.70562008

GA | iter = 38 | Mean = 35.50033781 | Best = 47.70562008

GA | iter = 39 | Mean = 43.46066753 | Best = 47.70562008

GA | iter = 40 | Mean = 43.06415782 | Best = 47.70562008

GA | iter = 41 | Mean = 38.63476036 | Best = 47.70562008

GA | iter = 42 | Mean = 39.15177369 | Best = 47.70562008

GA | iter = 43 | Mean = 38.50308870 | Best = 47.70562008

GA | iter = 44 | Mean = 36.17145552 | Best = 47.70562008

GA | iter = 45 | Mean = 38.16553237 | Best = 47.70562008

GA | iter = 46 | Mean = 33.11539038 | Best = 47.70562008

GA | iter = 47 | Mean = 38.74054644 | Best = 47.70562008

GA | iter = 48 | Mean = 38.49016248 | Best = 47.70562008

GA | iter = 49 | Mean = 41.60030722 | Best = 47.70562008

GA | iter = 50 | Mean = 40.61978809 | Best = 47.70562008

GA | iter = 51 | Mean = 34.49646052 | Best = 47.70562008

GA | iter = 52 | Mean = 31.85552707 | Best = 47.70562008

GA | iter = 53 | Mean = 36.68909960 | Best = 47.70562008

GA | iter = 54 | Mean = 38.76139186 | Best = 47.70562008

GA | iter = 55 | Mean = 32.42910148 | Best = 47.70562008

GA | iter = 56 | Mean = 33.35790875 | Best = 47.70562008

GA | iter = 57 | Mean = 29.49085771 | Best = 47.70562008

GA | iter = 58 | Mean = 37.73686425 | Best = 47.70562008

GA | iter = 59 | Mean = 31.68202935 | Best = 47.70562008

GA | iter = 60 | Mean = 30.75012860 | Best = 47.70562008

GA | iter = 61 | Mean = 39.27538580 | Best = 47.70562008

GA | iter = 62 | Mean = 36.69170315 | Best = 47.70562008

GA | iter = 63 | Mean = 40.07791669 | Best = 47.70562008

GA | iter = 64 | Mean = 33.72835123 | Best = 47.70562008

GA | iter = 65 | Mean = 33.35391980 | Best = 47.70562102

GA | iter = 66 | Mean = 35.47356230 | Best = 47.70562102

GA | iter = 67 | Mean = 39.46314129 | Best = 47.70562102

GA | iter = 68 | Mean = 41.09789117 | Best = 47.70562102

GA | iter = 69 | Mean = 39.39544596 | Best = 47.70562104

GA | iter = 70 | Mean = 38.08045737 | Best = 47.70562104

GA | iter = 71 | Mean = 36.76040330 | Best = 47.70562104

GA | iter = 72 | Mean = 38.14011012 | Best = 47.70562104

GA | iter = 73 | Mean = 42.38844363 | Best = 47.70562104

GA | iter = 74 | Mean = 38.30285897 | Best = 47.70562104

GA | iter = 75 | Mean = 45.66385434 | Best = 47.70562104

GA | iter = 76 | Mean = 46.05548984 | Best = 47.70562104

GA | iter = 77 | Mean = 43.38997327 | Best = 47.70562104

GA | iter = 78 | Mean = 40.97291165 | Best = 47.70562104

GA | iter = 79 | Mean = 42.32073677 | Best = 47.70562104

GA | iter = 80 | Mean = 45.86342382 | Best = 47.70562104

GA | iter = 81 | Mean = 45.27921312 | Best = 47.70562104

GA | iter = 82 | Mean = 39.93153642 | Best = 47.70562104

GA | iter = 83 | Mean = 42.82876540 | Best = 47.70562104

GA | iter = 84 | Mean = 43.76491797 | Best = 47.70562104

GA | iter = 85 | Mean = 41.13371054 | Best = 47.70562104

GA | iter = 86 | Mean = 39.92632955 | Best = 47.70562104

GA | iter = 87 | Mean = 37.28409710 | Best = 47.70562104

GA | iter = 88 | Mean = 39.75966028 | Best = 47.70562104

GA | iter = 89 | Mean = 37.51619208 | Best = 47.70562104

GA | iter = 90 | Mean = 32.87743999 | Best = 47.70562104

GA | iter = 91 | Mean = 33.13549420 | Best = 47.70562104

GA | iter = 92 | Mean = 37.84747342 | Best = 47.70562104

GA | iter = 93 | Mean = 36.83248859 | Best = 47.70562104

GA | iter = 94 | Mean = 36.53374959 | Best = 47.70562104

GA | iter = 95 | Mean = 38.55497005 | Best = 47.70562104

GA | iter = 96 | Mean = 41.45835690 | Best = 47.70562104

GA | iter = 97 | Mean = 40.82087369 | Best = 47.70562104

GA | iter = 98 | Mean = 38.95339849 | Best = 47.70562104

GA | iter = 99 | Mean = 36.45711453 | Best = 47.70562104

GA | iter = 100 | Mean = 35.57146560 | Best = 47.70562104

> # summary function is used to produce result summaries of the results of the model.

> summary (GA)

-- **Genetic Algorithm** -------------------

GA settings:

Type = real-valued

Population size = 50

Number of generations = 100

Elitism = 2

Crossover probability = 0.8

Mutation probability = 0.1

Search domain = x1

lower -10

upper 10

GA results:

Iterations = 100

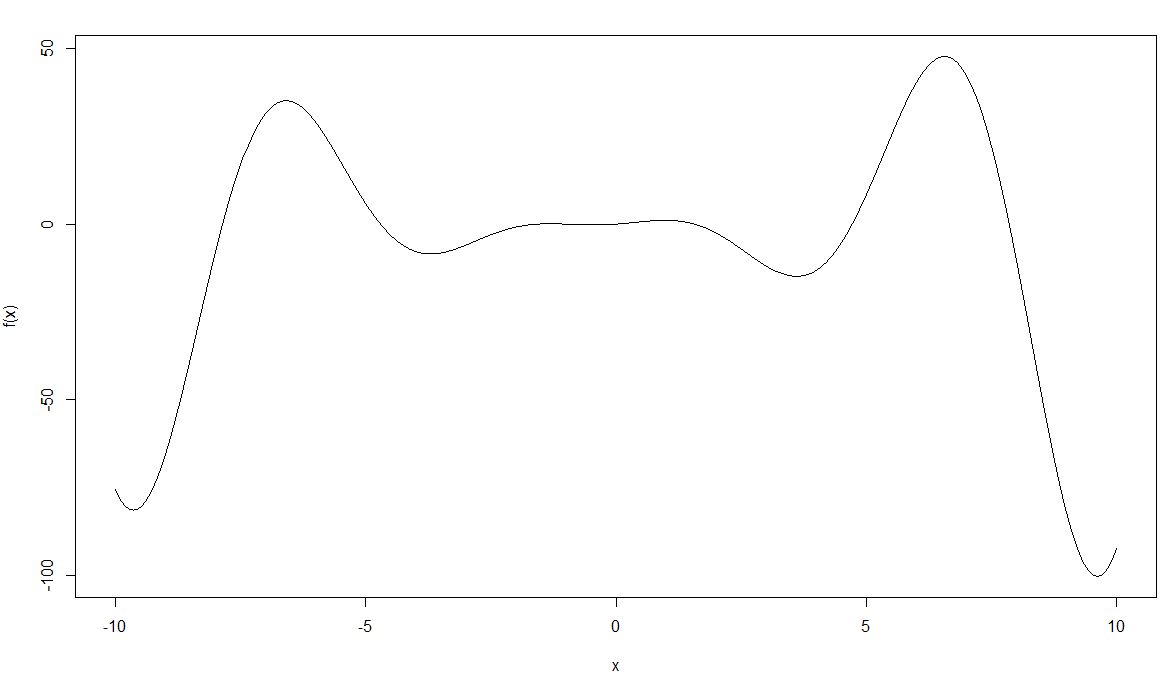
Fitness function value = 47.70562104

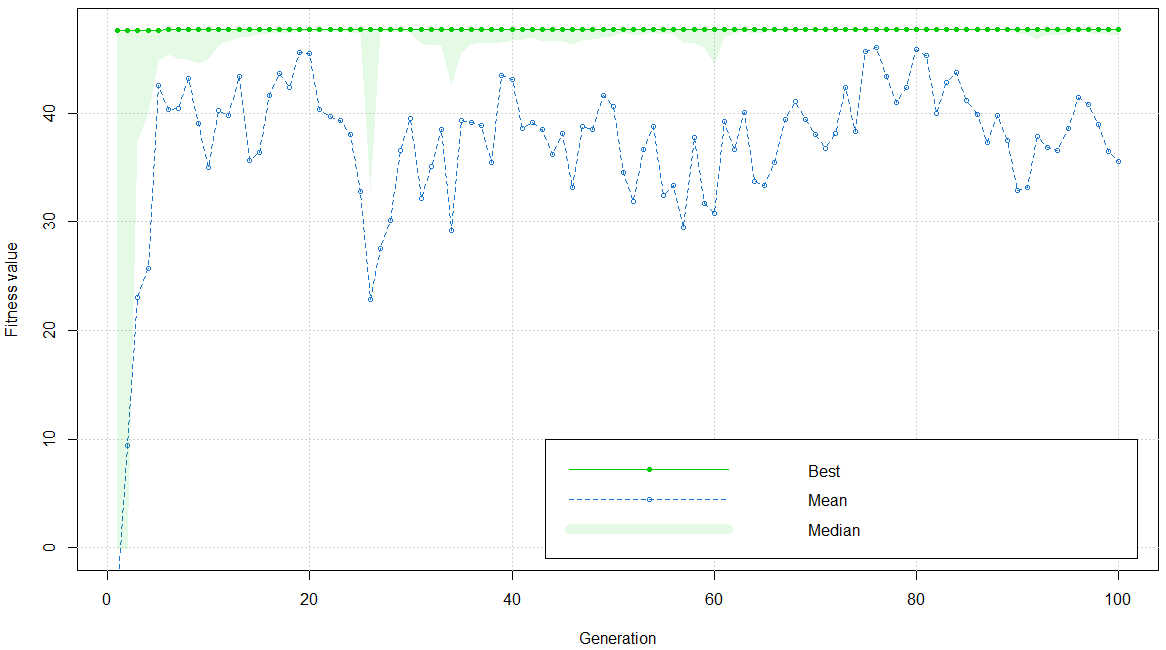
Solution = x1

[1,] 6.560527123

> # plotting GA graph

> plot(GA)





**Name: Sanesh Sagvekar Roll No: 170122**

**Practical No.: 5**

**AIM: Implement evolution strategy algorithm.**

> # loading DEoptim library

> library("DEoptim")

>

> # rosebrock store the function which has two variables x1 and x2.

> Rosenbrock = function(x)

+ {

+ x1 = x[1]

+ x2 = x[2]

+ 100 \* (x2 - x1 \* x1)^2 + (1 - x1)^2

+ }

>

> # set the initial value of seed to 100 which helps in giving same output each time we run the program.

> set.seed(100)

>

> # lbound is the lower bound of the search space.

>

>

> # ubound is the upper bound of the search space.

> ubound = c(20, 20)

> # ubound is the upper bound of the search space.

> ubound = c(20, 20)

> # Using strategy 1

> # DEoptim function is use to performs evolutionary global optimization using differential evolution algorithm.

> # the function takes four parameters first parameter takes the function for which we want to perform optimization,

> # second parameter takes lower bound for each variable in the function, third parameter upper takes upper bound value for each variable in function,

> # fourth parameter DEoptim.control function which 5 input parameters first parameter takes number of population members,

> # second parameter is strategy which integer corresponding with different differential evolution strategies,

> # third parameter is itermax which takes maximum number of iterations allowed, forth parameter is F it takes differential weigth from interval 0 to 2,

> # fifth parameter is CR takes crossover probability between 0 and 2.

> outputDEopim1 = DEoptim(Rosenbrock, lower = lbound, upper = ubound, DEoptim.control(NP = 100, strategy = 1, itermax = 400, F = 1.2, CR = 0.7))

Iteration: 1 bestvalit: 35.117743 bestmemit: -1.804997 2.736002

Iteration: 2 bestvalit: 35.117743 bestmemit: -1.804997 2.736002

Iteration: 3 bestvalit: 12.848064 bestmemit: 2.988492 8.632858

Iteration: 4 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 5 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 6 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 7 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 8 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 9 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 10 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 11 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 12 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 13 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 14 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 15 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 16 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 17 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 18 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 19 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 20 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 21 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 22 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 23 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 24 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 25 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 26 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 27 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 28 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 29 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 30 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 31 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 32 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 33 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 34 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 35 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 36 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 37 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 38 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 39 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 40 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 41 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 42 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 43 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 44 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 45 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 46 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 47 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 48 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 49 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 50 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 51 bestvalit: 0.073358 bestmemit: 1.256781 1.570884

Iteration: 52 bestvalit: 0.059256 bestmemit: 1.195982 1.415935

Iteration: 53 bestvalit: 0.059256 bestmemit: 1.195982 1.415935

Iteration: 54 bestvalit: 0.059256 bestmemit: 1.195982 1.415935

Iteration: 55 bestvalit: 0.059256 bestmemit: 1.195982 1.415935

Iteration: 56 bestvalit: 0.059256 bestmemit: 1.195982 1.415935

Iteration: 57 bestvalit: 0.059256 bestmemit: 1.195982 1.415935

Iteration: 58 bestvalit: 0.059256 bestmemit: 1.195982 1.415935

Iteration: 59 bestvalit: 0.059256 bestmemit: 1.195982 1.415935

Iteration: 60 bestvalit: 0.046727 bestmemit: 1.147887 1.301878

Iteration: 61 bestvalit: 0.046727 bestmemit: 1.147887 1.301878

Iteration: 62 bestvalit: 0.046727 bestmemit: 1.147887 1.301878

Iteration: 63 bestvalit: 0.046727 bestmemit: 1.147887 1.301878

Iteration: 64 bestvalit: 0.046727 bestmemit: 1.147887 1.301878

Iteration: 65 bestvalit: 0.046727 bestmemit: 1.147887 1.301878

Iteration: 66 bestvalit: 0.046727 bestmemit: 1.147887 1.301878

Iteration: 67 bestvalit: 0.023973 bestmemit: 0.883200 0.769878

Iteration: 68 bestvalit: 0.023973 bestmemit: 0.883200 0.769878

Iteration: 69 bestvalit: 0.023973 bestmemit: 0.883200 0.769878

Iteration: 70 bestvalit: 0.023973 bestmemit: 0.883200 0.769878

Iteration: 71 bestvalit: 0.023973 bestmemit: 0.883200 0.769878

Iteration: 72 bestvalit: 0.023973 bestmemit: 0.883200 0.769878

Iteration: 73 bestvalit: 0.023973 bestmemit: 0.883200 0.769878

Iteration: 74 bestvalit: 0.023973 bestmemit: 0.883200 0.769878

Iteration: 75 bestvalit: 0.017248 bestmemit: 1.006777 1.026716

Iteration: 76 bestvalit: 0.017248 bestmemit: 1.006777 1.026716

Iteration: 77 bestvalit: 0.017248 bestmemit: 1.006777 1.026716

Iteration: 78 bestvalit: 0.017248 bestmemit: 1.006777 1.026716

Iteration: 79 bestvalit: 0.005960 bestmemit: 0.964026 0.936176

Iteration: 80 bestvalit: 0.005960 bestmemit: 0.964026 0.936176

Iteration: 81 bestvalit: 0.005960 bestmemit: 0.964026 0.936176

Iteration: 82 bestvalit: 0.005960 bestmemit: 0.964026 0.936176

Iteration: 83 bestvalit: 0.005960 bestmemit: 0.964026 0.936176

Iteration: 84 bestvalit: 0.005960 bestmemit: 0.964026 0.936176

Iteration: 85 bestvalit: 0.005960 bestmemit: 0.964026 0.936176

Iteration: 86 bestvalit: 0.005960 bestmemit: 0.964026 0.936176

Iteration: 87 bestvalit: 0.005960 bestmemit: 0.964026 0.936176

Iteration: 88 bestvalit: 0.001741 bestmemit: 1.017352 1.031210

Iteration: 89 bestvalit: 0.001741 bestmemit: 1.017352 1.031210

Iteration: 90 bestvalit: 0.001741 bestmemit: 1.017352 1.031210

Iteration: 91 bestvalit: 0.001741 bestmemit: 1.017352 1.031210

Iteration: 92 bestvalit: 0.001741 bestmemit: 1.017352 1.031210

Iteration: 93 bestvalit: 0.001741 bestmemit: 1.017352 1.031210

Iteration: 94 bestvalit: 0.001741 bestmemit: 1.017352 1.031210

Iteration: 95 bestvalit: 0.001741 bestmemit: 1.017352 1.031210

Iteration: 96 bestvalit: 0.001741 bestmemit: 1.017352 1.031210

Iteration: 97 bestvalit: 0.001741 bestmemit: 1.017352 1.031210

Iteration: 98 bestvalit: 0.001741 bestmemit: 1.017352 1.031210

Iteration: 99 bestvalit: 0.001741 bestmemit: 1.017352 1.031210

Iteration: 100 bestvalit: 0.001741 bestmemit: 1.017352 1.031210

Iteration: 101 bestvalit: 0.001741 bestmemit: 1.017352 1.031210

Iteration: 102 bestvalit: 0.001633 bestmemit: 1.030603 1.064782

Iteration: 103 bestvalit: 0.001143 bestmemit: 0.981969 0.967123

Iteration: 104 bestvalit: 0.000780 bestmemit: 0.999498 0.996204

Iteration: 105 bestvalit: 0.000780 bestmemit: 0.999498 0.996204

Iteration: 106 bestvalit: 0.000045 bestmemit: 0.993415 0.986733

Iteration: 107 bestvalit: 0.000045 bestmemit: 0.993415 0.986733

Iteration: 108 bestvalit: 0.000045 bestmemit: 0.993415 0.986733

Iteration: 109 bestvalit: 0.000045 bestmemit: 0.993415 0.986733

Iteration: 110 bestvalit: 0.000045 bestmemit: 0.993415 0.986733

Iteration: 111 bestvalit: 0.000045 bestmemit: 0.993415 0.986733

Iteration: 112 bestvalit: 0.000045 bestmemit: 0.993415 0.986733

Iteration: 113 bestvalit: 0.000045 bestmemit: 0.993415 0.986733

Iteration: 114 bestvalit: 0.000045 bestmemit: 0.993415 0.986733

Iteration: 115 bestvalit: 0.000045 bestmemit: 0.993415 0.986733

Iteration: 116 bestvalit: 0.000045 bestmemit: 0.993415 0.986733

Iteration: 117 bestvalit: 0.000045 bestmemit: 0.993415 0.986733

Iteration: 118 bestvalit: 0.000034 bestmemit: 1.005334 1.010462

Iteration: 119 bestvalit: 0.000034 bestmemit: 1.005334 1.010462

Iteration: 120 bestvalit: 0.000034 bestmemit: 1.005334 1.010462

Iteration: 121 bestvalit: 0.000034 bestmemit: 1.005334 1.010462

Iteration: 122 bestvalit: 0.000034 bestmemit: 1.005334 1.010462

Iteration: 123 bestvalit: 0.000034 bestmemit: 1.005334 1.010462

Iteration: 124 bestvalit: 0.000034 bestmemit: 1.005334 1.010462

Iteration: 125 bestvalit: 0.000034 bestmemit: 1.005334 1.010462

Iteration: 126 bestvalit: 0.000034 bestmemit: 1.005334 1.010462

Iteration: 127 bestvalit: 0.000020 bestmemit: 0.996814 0.993331

Iteration: 128 bestvalit: 0.000000 bestmemit: 0.999697 0.999348

Iteration: 129 bestvalit: 0.000000 bestmemit: 0.999697 0.999348

Iteration: 130 bestvalit: 0.000000 bestmemit: 0.999697 0.999348

Iteration: 131 bestvalit: 0.000000 bestmemit: 0.999697 0.999348

Iteration: 132 bestvalit: 0.000000 bestmemit: 0.999697 0.999348

Iteration: 133 bestvalit: 0.000000 bestmemit: 0.999697 0.999348

Iteration: 134 bestvalit: 0.000000 bestmemit: 0.999697 0.999348

Iteration: 135 bestvalit: 0.000000 bestmemit: 0.999697 0.999348

Iteration: 136 bestvalit: 0.000000 bestmemit: 0.999697 0.999348

Iteration: 137 bestvalit: 0.000000 bestmemit: 0.999697 0.999348

Iteration: 138 bestvalit: 0.000000 bestmemit: 0.999697 0.999348

Iteration: 139 bestvalit: 0.000000 bestmemit: 0.999697 0.999348

Iteration: 140 bestvalit: 0.000000 bestmemit: 0.999697 0.999348

Iteration: 141 bestvalit: 0.000000 bestmemit: 0.999697 0.999348

Iteration: 142 bestvalit: 0.000000 bestmemit: 0.999697 0.999348

Iteration: 143 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 144 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 145 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 146 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 147 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 148 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 149 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 150 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 151 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 152 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 153 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 154 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 155 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 156 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 157 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 158 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 159 bestvalit: 0.000000 bestmemit: 0.999932 0.999818

Iteration: 160 bestvalit: 0.000000 bestmemit: 1.000017 1.000071

Iteration: 161 bestvalit: 0.000000 bestmemit: 1.000017 1.000071

Iteration: 162 bestvalit: 0.000000 bestmemit: 1.000017 1.000071

Iteration: 163 bestvalit: 0.000000 bestmemit: 1.000017 1.000071

Iteration: 164 bestvalit: 0.000000 bestmemit: 1.000017 1.000071

Iteration: 165 bestvalit: 0.000000 bestmemit: 0.999983 0.999974

Iteration: 166 bestvalit: 0.000000 bestmemit: 0.999983 0.999974

Iteration: 167 bestvalit: 0.000000 bestmemit: 1.000070 1.000138

Iteration: 168 bestvalit: 0.000000 bestmemit: 1.000070 1.000138

Iteration: 169 bestvalit: 0.000000 bestmemit: 1.000070 1.000138

Iteration: 170 bestvalit: 0.000000 bestmemit: 1.000070 1.000138

Iteration: 171 bestvalit: 0.000000 bestmemit: 1.000070 1.000138

Iteration: 172 bestvalit: 0.000000 bestmemit: 0.999987 0.999969

Iteration: 173 bestvalit: 0.000000 bestmemit: 0.999987 0.999969

Iteration: 174 bestvalit: 0.000000 bestmemit: 0.999987 0.999969

Iteration: 175 bestvalit: 0.000000 bestmemit: 0.999987 0.999969

Iteration: 176 bestvalit: 0.000000 bestmemit: 0.999987 0.999969

Iteration: 177 bestvalit: 0.000000 bestmemit: 0.999987 0.999969

Iteration: 178 bestvalit: 0.000000 bestmemit: 1.000031 1.000058

Iteration: 179 bestvalit: 0.000000 bestmemit: 0.999985 0.999974

Iteration: 180 bestvalit: 0.000000 bestmemit: 0.999985 0.999974

Iteration: 181 bestvalit: 0.000000 bestmemit: 0.999985 0.999974

Iteration: 182 bestvalit: 0.000000 bestmemit: 0.999985 0.999974

Iteration: 183 bestvalit: 0.000000 bestmemit: 0.999985 0.999974

Iteration: 184 bestvalit: 0.000000 bestmemit: 0.999985 0.999974

Iteration: 185 bestvalit: 0.000000 bestmemit: 0.999985 0.999974

Iteration: 186 bestvalit: 0.000000 bestmemit: 0.999985 0.999974

Iteration: 187 bestvalit: 0.000000 bestmemit: 0.999985 0.999974

Iteration: 188 bestvalit: 0.000000 bestmemit: 0.999985 0.999974

Iteration: 189 bestvalit: 0.000000 bestmemit: 0.999987 0.999973

Iteration: 190 bestvalit: 0.000000 bestmemit: 0.999987 0.999973

Iteration: 191 bestvalit: 0.000000 bestmemit: 0.999987 0.999973

Iteration: 192 bestvalit: 0.000000 bestmemit: 0.999987 0.999973

Iteration: 193 bestvalit: 0.000000 bestmemit: 0.999987 0.999973

Iteration: 194 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 195 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 196 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 197 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 198 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 199 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 200 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 201 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 202 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 203 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 204 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 205 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 206 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 207 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 208 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 209 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 210 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 211 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 212 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 213 bestvalit: 0.000000 bestmemit: 1.000006 1.000011

Iteration: 214 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 215 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 216 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 217 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 218 bestvalit: 0.000000 bestmemit: 1.000001 1.000003

Iteration: 219 bestvalit: 0.000000 bestmemit: 1.000001 1.000003

Iteration: 220 bestvalit: 0.000000 bestmemit: 1.000001 1.000003

Iteration: 221 bestvalit: 0.000000 bestmemit: 1.000001 1.000003

Iteration: 222 bestvalit: 0.000000 bestmemit: 1.000001 1.000003

Iteration: 223 bestvalit: 0.000000 bestmemit: 1.000001 1.000003

Iteration: 224 bestvalit: 0.000000 bestmemit: 1.000001 1.000003

Iteration: 225 bestvalit: 0.000000 bestmemit: 1.000001 1.000003

Iteration: 226 bestvalit: 0.000000 bestmemit: 1.000001 1.000003

Iteration: 227 bestvalit: 0.000000 bestmemit: 1.000001 1.000003

Iteration: 228 bestvalit: 0.000000 bestmemit: 0.999999 0.999998

Iteration: 229 bestvalit: 0.000000 bestmemit: 0.999999 0.999998

Iteration: 230 bestvalit: 0.000000 bestmemit: 0.999999 0.999998

Iteration: 231 bestvalit: 0.000000 bestmemit: 0.999999 0.999998

Iteration: 232 bestvalit: 0.000000 bestmemit: 1.000001 1.000001

Iteration: 233 bestvalit: 0.000000 bestmemit: 1.000001 1.000001

Iteration: 234 bestvalit: 0.000000 bestmemit: 1.000001 1.000001

Iteration: 235 bestvalit: 0.000000 bestmemit: 1.000001 1.000001

Iteration: 236 bestvalit: 0.000000 bestmemit: 1.000001 1.000001

Iteration: 237 bestvalit: 0.000000 bestmemit: 1.000001 1.000001

Iteration: 238 bestvalit: 0.000000 bestmemit: 1.000001 1.000001

Iteration: 239 bestvalit: 0.000000 bestmemit: 1.000001 1.000001

Iteration: 240 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 241 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 242 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 243 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 244 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 245 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 246 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 247 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 248 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 249 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 250 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 251 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 252 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 253 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 254 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 255 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 256 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 257 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 258 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 259 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 260 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 261 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 262 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 263 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 264 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 265 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 266 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 267 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 268 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 269 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 270 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 271 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 272 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 273 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 274 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 275 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 276 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 277 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 278 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 279 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 280 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 281 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 282 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 283 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 284 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 285 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 286 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 287 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 288 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 289 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 290 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 291 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 292 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 293 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 294 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 295 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 296 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 297 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 298 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 299 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 300 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 301 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 302 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 303 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 304 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 305 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 306 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 307 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 308 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 309 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 310 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 311 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 312 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 313 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 314 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 315 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 316 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 317 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 318 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 319 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 320 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 321 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 322 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 323 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 324 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 325 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 326 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 327 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 328 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 329 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 330 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 331 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 332 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 333 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 334 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 335 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 336 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 337 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 338 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 339 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 340 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 341 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 342 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 343 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 344 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 345 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 346 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 347 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 348 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 349 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 350 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 351 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 352 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 353 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 354 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 355 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 356 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 357 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 358 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 359 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 360 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 361 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 362 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 363 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 364 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 365 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 366 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 367 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 368 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 369 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 370 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 371 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 372 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 373 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 374 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 375 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 376 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 377 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 378 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 379 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 380 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 381 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 382 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 383 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 384 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 385 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 386 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 387 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 388 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 389 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 390 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 391 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 392 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 393 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 394 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 395 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 396 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 397 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 398 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

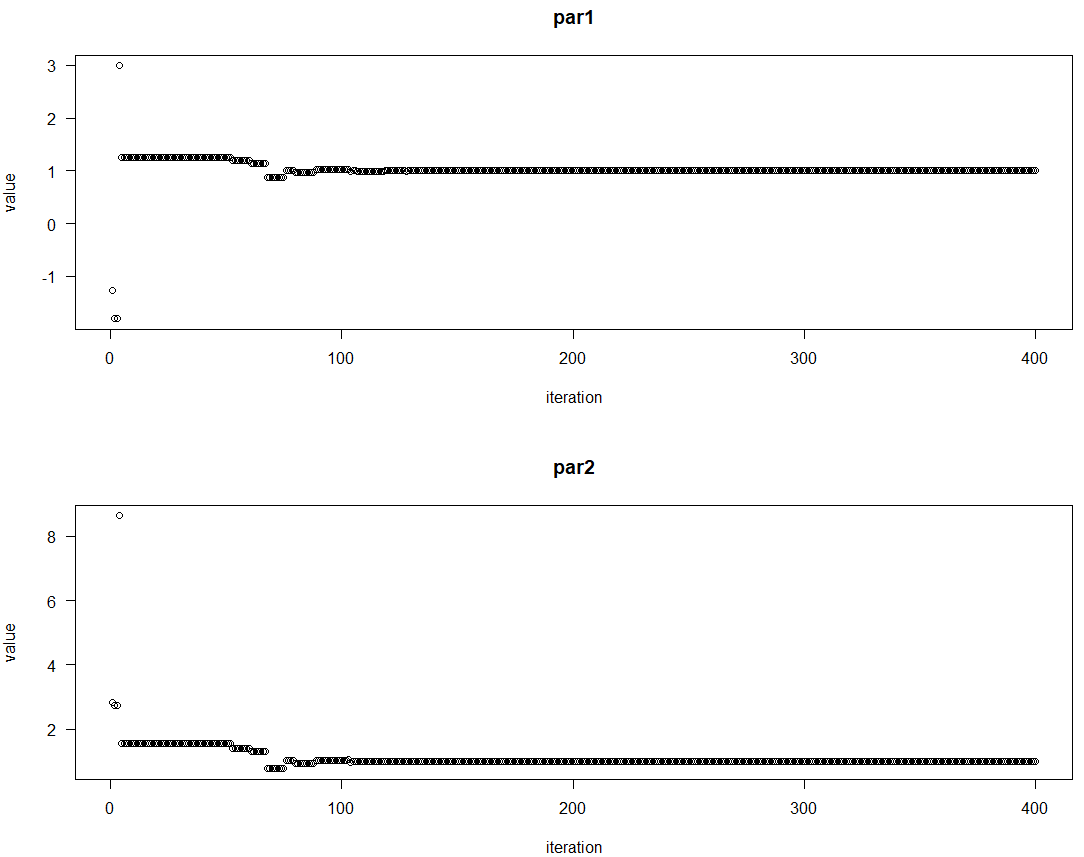
Iteration: 399 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 400 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

>

> # ploting the output

> plot(outputDEopim1)



> # plotting the output

> plot(outputDEopim1)

> # Using strategy 2

> # DEoptim function is use to performs evolutionary global optimization using differential evolution algorithm.

> # the function takes four parameters first parameter takes the function for which we want to perform optimization,

> # second parameter takes lower bound for each variable in the function, third parameter upper takes upper bound value for each variable in function,

> # fourth parameter DEoptim.control function which 5 input parameters first parameter takes number of population members,

> # second parameter is strategy which integer corresponding with different differential evolution strategies,

> # third parameter is itermax which takes maximum number of iterations allowed, forth parameter is F it takes differential weigth from interval 0 to 2,

> # fifth parameter is CR takes crossover probability between 0 and 2.

> outputDEopim2 = DEoptim(Rosenbrock, lower = lbound, upper = ubound, DEoptim.control(NP = 100, strategy = 2, itermax = 400, F = 1.2, CR = 0.7))

Iteration: 1 bestvalit: 7.639673 bestmemit: 3.675078 13.436655

Iteration: 2 bestvalit: 7.639673 bestmemit: 3.675078 13.436655

Iteration: 3 bestvalit: 7.639673 bestmemit: 3.675078 13.436655

Iteration: 4 bestvalit: 7.639673 bestmemit: 3.675078 13.436655

Iteration: 5 bestvalit: 0.731779 bestmemit: 1.850472 3.415041

Iteration: 6 bestvalit: 0.731779 bestmemit: 1.850472 3.415041

Iteration: 7 bestvalit: 0.731779 bestmemit: 1.850472 3.415041

Iteration: 8 bestvalit: 0.731779 bestmemit: 1.850472 3.415041

Iteration: 9 bestvalit: 0.731779 bestmemit: 1.850472 3.415041

Iteration: 10 bestvalit: 0.731779 bestmemit: 1.850472 3.415041

Iteration: 11 bestvalit: 0.731779 bestmemit: 1.850472 3.415041

Iteration: 12 bestvalit: 0.561605 bestmemit: 0.949563 0.976441

Iteration: 13 bestvalit: 0.561605 bestmemit: 0.949563 0.976441

Iteration: 14 bestvalit: 0.369112 bestmemit: 1.499090 2.212628

Iteration: 15 bestvalit: 0.369112 bestmemit: 1.499090 2.212628

Iteration: 16 bestvalit: 0.369112 bestmemit: 1.499090 2.212628

Iteration: 17 bestvalit: 0.369112 bestmemit: 1.499090 2.212628

Iteration: 18 bestvalit: 0.369112 bestmemit: 1.499090 2.212628

Iteration: 19 bestvalit: 0.369112 bestmemit: 1.499090 2.212628

Iteration: 20 bestvalit: 0.369112 bestmemit: 1.499090 2.212628

Iteration: 21 bestvalit: 0.361868 bestmemit: 1.585781 2.528387

Iteration: 22 bestvalit: 0.361868 bestmemit: 1.585781 2.528387

Iteration: 23 bestvalit: 0.361868 bestmemit: 1.585781 2.528387

Iteration: 24 bestvalit: 0.361868 bestmemit: 1.585781 2.528387

Iteration: 25 bestvalit: 0.132642 bestmemit: 1.009103 1.054698

Iteration: 26 bestvalit: 0.132642 bestmemit: 1.009103 1.054698

Iteration: 27 bestvalit: 0.132642 bestmemit: 1.009103 1.054698

Iteration: 28 bestvalit: 0.132642 bestmemit: 1.009103 1.054698

Iteration: 29 bestvalit: 0.132642 bestmemit: 1.009103 1.054698

Iteration: 30 bestvalit: 0.103344 bestmemit: 0.957545 0.948758

Iteration: 31 bestvalit: 0.103344 bestmemit: 0.957545 0.948758

Iteration: 32 bestvalit: 0.103344 bestmemit: 0.957545 0.948758

Iteration: 33 bestvalit: 0.103344 bestmemit: 0.957545 0.948758

Iteration: 34 bestvalit: 0.103344 bestmemit: 0.957545 0.948758

Iteration: 35 bestvalit: 0.103344 bestmemit: 0.957545 0.948758

Iteration: 36 bestvalit: 0.081335 bestmemit: 1.254464 1.560803

Iteration: 37 bestvalit: 0.081335 bestmemit: 1.254464 1.560803

Iteration: 38 bestvalit: 0.081335 bestmemit: 1.254464 1.560803

Iteration: 39 bestvalit: 0.081335 bestmemit: 1.254464 1.560803

Iteration: 40 bestvalit: 0.057624 bestmemit: 0.973297 0.971163

Iteration: 41 bestvalit: 0.057624 bestmemit: 0.973297 0.971163

Iteration: 42 bestvalit: 0.057624 bestmemit: 0.973297 0.971163

Iteration: 43 bestvalit: 0.057624 bestmemit: 0.973297 0.971163

Iteration: 44 bestvalit: 0.057624 bestmemit: 0.973297 0.971163

Iteration: 45 bestvalit: 0.012070 bestmemit: 0.943441 0.880662

Iteration: 46 bestvalit: 0.012070 bestmemit: 0.943441 0.880662

Iteration: 47 bestvalit: 0.012070 bestmemit: 0.943441 0.880662

Iteration: 48 bestvalit: 0.006848 bestmemit: 0.987505 0.966986

Iteration: 49 bestvalit: 0.006848 bestmemit: 0.987505 0.966986

Iteration: 50 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 51 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 52 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 53 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 54 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 55 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 56 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 57 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 58 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 59 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 60 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 61 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 62 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 63 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 64 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 65 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 66 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 67 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 68 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 69 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 70 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 71 bestvalit: 0.002492 bestmemit: 1.048441 1.100436

Iteration: 72 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 73 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 74 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 75 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 76 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 77 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 78 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 79 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 80 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 81 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 82 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 83 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 84 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 85 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 86 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 87 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 88 bestvalit: 0.000573 bestmemit: 1.019081 1.037081

Iteration: 89 bestvalit: 0.000514 bestmemit: 0.998300 0.994342

Iteration: 90 bestvalit: 0.000514 bestmemit: 0.998300 0.994342

Iteration: 91 bestvalit: 0.000514 bestmemit: 0.998300 0.994342

Iteration: 92 bestvalit: 0.000514 bestmemit: 0.998300 0.994342

Iteration: 93 bestvalit: 0.000307 bestmemit: 0.988524 0.975857

Iteration: 94 bestvalit: 0.000307 bestmemit: 0.988524 0.975857

Iteration: 95 bestvalit: 0.000040 bestmemit: 1.005242 1.010862

Iteration: 96 bestvalit: 0.000040 bestmemit: 1.005242 1.010862

Iteration: 97 bestvalit: 0.000040 bestmemit: 1.005242 1.010862

Iteration: 98 bestvalit: 0.000040 bestmemit: 1.005242 1.010862

Iteration: 99 bestvalit: 0.000040 bestmemit: 1.005242 1.010862

Iteration: 100 bestvalit: 0.000040 bestmemit: 1.005242 1.010862

Iteration: 101 bestvalit: 0.000022 bestmemit: 0.996268 0.992840

Iteration: 102 bestvalit: 0.000022 bestmemit: 0.996268 0.992840

Iteration: 103 bestvalit: 0.000022 bestmemit: 0.996268 0.992840

Iteration: 104 bestvalit: 0.000022 bestmemit: 0.996268 0.992840

Iteration: 105 bestvalit: 0.000022 bestmemit: 0.996268 0.992840

Iteration: 106 bestvalit: 0.000022 bestmemit: 0.996268 0.992840

Iteration: 107 bestvalit: 0.000022 bestmemit: 0.996268 0.992840

Iteration: 108 bestvalit: 0.000022 bestmemit: 0.996268 0.992840

Iteration: 109 bestvalit: 0.000022 bestmemit: 0.996268 0.992840

Iteration: 110 bestvalit: 0.000022 bestmemit: 0.996268 0.992840

Iteration: 111 bestvalit: 0.000022 bestmemit: 0.996268 0.992840

Iteration: 112 bestvalit: 0.000022 bestmemit: 0.996268 0.992840

Iteration: 113 bestvalit: 0.000022 bestmemit: 0.996268 0.992840

Iteration: 114 bestvalit: 0.000022 bestmemit: 0.996268 0.992840

Iteration: 115 bestvalit: 0.000012 bestmemit: 1.002994 1.006178

Iteration: 116 bestvalit: 0.000012 bestmemit: 1.002994 1.006178

Iteration: 117 bestvalit: 0.000012 bestmemit: 1.002994 1.006178

Iteration: 118 bestvalit: 0.000003 bestmemit: 0.998759 0.997625

Iteration: 119 bestvalit: 0.000003 bestmemit: 0.998759 0.997625

Iteration: 120 bestvalit: 0.000003 bestmemit: 0.998759 0.997625

Iteration: 121 bestvalit: 0.000003 bestmemit: 0.998759 0.997625

Iteration: 122 bestvalit: 0.000003 bestmemit: 0.998759 0.997625

Iteration: 123 bestvalit: 0.000001 bestmemit: 1.000528 1.001126

Iteration: 124 bestvalit: 0.000001 bestmemit: 1.000528 1.001126

Iteration: 125 bestvalit: 0.000001 bestmemit: 1.000528 1.001126

Iteration: 126 bestvalit: 0.000001 bestmemit: 1.000528 1.001126

Iteration: 127 bestvalit: 0.000001 bestmemit: 1.000528 1.001126

Iteration: 128 bestvalit: 0.000001 bestmemit: 1.000528 1.001126

Iteration: 129 bestvalit: 0.000001 bestmemit: 1.000528 1.001126

Iteration: 130 bestvalit: 0.000001 bestmemit: 1.000528 1.001126

Iteration: 131 bestvalit: 0.000001 bestmemit: 1.000528 1.001126

Iteration: 132 bestvalit: 0.000001 bestmemit: 1.000528 1.001126

Iteration: 133 bestvalit: 0.000001 bestmemit: 1.000528 1.001126

Iteration: 134 bestvalit: 0.000001 bestmemit: 1.000528 1.001126

Iteration: 135 bestvalit: 0.000001 bestmemit: 1.000528 1.001126

Iteration: 136 bestvalit: 0.000000 bestmemit: 0.999584 0.999121

Iteration: 137 bestvalit: 0.000000 bestmemit: 1.000066 1.000155

Iteration: 138 bestvalit: 0.000000 bestmemit: 1.000066 1.000155

Iteration: 139 bestvalit: 0.000000 bestmemit: 1.000066 1.000155

Iteration: 140 bestvalit: 0.000000 bestmemit: 1.000066 1.000155

Iteration: 141 bestvalit: 0.000000 bestmemit: 1.000066 1.000155

Iteration: 142 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 143 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 144 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 145 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 146 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 147 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 148 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 149 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 150 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 151 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 152 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 153 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 154 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 155 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 156 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 157 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 158 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 159 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 160 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 161 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 162 bestvalit: 0.000000 bestmemit: 0.999898 0.999796

Iteration: 163 bestvalit: 0.000000 bestmemit: 1.000062 1.000132

Iteration: 164 bestvalit: 0.000000 bestmemit: 1.000062 1.000132

Iteration: 165 bestvalit: 0.000000 bestmemit: 1.000062 1.000132

Iteration: 166 bestvalit: 0.000000 bestmemit: 1.000062 1.000132

Iteration: 167 bestvalit: 0.000000 bestmemit: 1.000061 1.000124

Iteration: 168 bestvalit: 0.000000 bestmemit: 1.000061 1.000124

Iteration: 169 bestvalit: 0.000000 bestmemit: 1.000061 1.000124

Iteration: 170 bestvalit: 0.000000 bestmemit: 1.000061 1.000124

Iteration: 171 bestvalit: 0.000000 bestmemit: 1.000038 1.000074

Iteration: 172 bestvalit: 0.000000 bestmemit: 1.000038 1.000074

Iteration: 173 bestvalit: 0.000000 bestmemit: 1.000030 1.000059

Iteration: 174 bestvalit: 0.000000 bestmemit: 1.000030 1.000059

Iteration: 175 bestvalit: 0.000000 bestmemit: 1.000030 1.000059

Iteration: 176 bestvalit: 0.000000 bestmemit: 0.999991 0.999984

Iteration: 177 bestvalit: 0.000000 bestmemit: 0.999991 0.999984

Iteration: 178 bestvalit: 0.000000 bestmemit: 0.999991 0.999984

Iteration: 179 bestvalit: 0.000000 bestmemit: 0.999991 0.999984

Iteration: 180 bestvalit: 0.000000 bestmemit: 1.000013 1.000026

Iteration: 181 bestvalit: 0.000000 bestmemit: 1.000013 1.000026

Iteration: 182 bestvalit: 0.000000 bestmemit: 1.000013 1.000026

Iteration: 183 bestvalit: 0.000000 bestmemit: 1.000013 1.000026

Iteration: 184 bestvalit: 0.000000 bestmemit: 1.000013 1.000026

Iteration: 185 bestvalit: 0.000000 bestmemit: 1.000013 1.000026

Iteration: 186 bestvalit: 0.000000 bestmemit: 1.000013 1.000026

Iteration: 187 bestvalit: 0.000000 bestmemit: 1.000013 1.000026

Iteration: 188 bestvalit: 0.000000 bestmemit: 1.000013 1.000026

Iteration: 189 bestvalit: 0.000000 bestmemit: 0.999994 0.999988

Iteration: 190 bestvalit: 0.000000 bestmemit: 0.999994 0.999988

Iteration: 191 bestvalit: 0.000000 bestmemit: 0.999994 0.999988

Iteration: 192 bestvalit: 0.000000 bestmemit: 0.999994 0.999988

Iteration: 193 bestvalit: 0.000000 bestmemit: 0.999994 0.999988

Iteration: 194 bestvalit: 0.000000 bestmemit: 0.999994 0.999988

Iteration: 195 bestvalit: 0.000000 bestmemit: 0.999994 0.999988

Iteration: 196 bestvalit: 0.000000 bestmemit: 0.999994 0.999988

Iteration: 197 bestvalit: 0.000000 bestmemit: 0.999994 0.999988

Iteration: 198 bestvalit: 0.000000 bestmemit: 0.999994 0.999988

Iteration: 199 bestvalit: 0.000000 bestmemit: 0.999994 0.999988

Iteration: 200 bestvalit: 0.000000 bestmemit: 0.999994 0.999988

Iteration: 201 bestvalit: 0.000000 bestmemit: 0.999998 0.999997

Iteration: 202 bestvalit: 0.000000 bestmemit: 0.999998 0.999997

Iteration: 203 bestvalit: 0.000000 bestmemit: 0.999998 0.999997

Iteration: 204 bestvalit: 0.000000 bestmemit: 0.999998 0.999997

Iteration: 205 bestvalit: 0.000000 bestmemit: 0.999998 0.999997

Iteration: 206 bestvalit: 0.000000 bestmemit: 0.999997 0.999994

Iteration: 207 bestvalit: 0.000000 bestmemit: 0.999997 0.999994

Iteration: 208 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 209 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 210 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 211 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 212 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 213 bestvalit: 0.000000 bestmemit: 0.999999 0.999999

Iteration: 214 bestvalit: 0.000000 bestmemit: 0.999999 0.999999

Iteration: 215 bestvalit: 0.000000 bestmemit: 0.999999 0.999999

Iteration: 216 bestvalit: 0.000000 bestmemit: 0.999999 0.999999

Iteration: 217 bestvalit: 0.000000 bestmemit: 0.999999 0.999999

Iteration: 218 bestvalit: 0.000000 bestmemit: 0.999999 0.999999

Iteration: 219 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 220 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 221 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 222 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 223 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 224 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 225 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 226 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 227 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 228 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 229 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 230 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 231 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 232 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 233 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 234 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 235 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 236 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 237 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 238 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 239 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 240 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 241 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 242 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 243 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 244 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 245 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 246 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 247 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 248 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 249 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 250 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 251 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 252 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 253 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 254 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 255 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 256 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 257 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 258 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 259 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 260 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 261 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 262 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 263 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 264 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 265 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 266 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 267 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 268 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 269 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 270 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 271 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 272 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 273 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 274 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 275 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 276 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 277 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 278 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 279 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 280 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 281 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 282 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 283 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 284 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 285 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 286 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 287 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 288 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 289 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 290 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 291 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 292 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 293 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 294 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 295 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 296 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 297 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 298 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 299 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 300 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 301 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 302 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 303 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 304 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 305 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 306 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 307 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 308 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 309 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 310 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 311 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 312 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 313 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 314 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 315 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 316 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 317 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 318 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 319 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 320 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 321 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 322 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 323 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 324 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 325 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 326 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 327 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 328 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 329 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 330 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 331 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 332 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 333 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 334 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 335 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 336 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 337 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 338 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 339 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 340 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 341 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 342 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 343 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 344 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 345 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 346 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 347 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 348 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 349 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 350 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 351 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 352 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 353 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 354 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 355 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 356 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 357 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 358 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 359 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 360 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 361 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 362 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 363 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 364 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 365 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 366 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 367 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 368 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 369 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 370 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 371 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 372 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 373 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 374 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 375 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 376 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 377 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 378 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 379 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 380 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 381 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 382 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 383 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 384 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 385 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 386 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 387 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 388 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 389 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 390 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 391 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 392 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 393 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 394 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 395 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 396 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 397 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

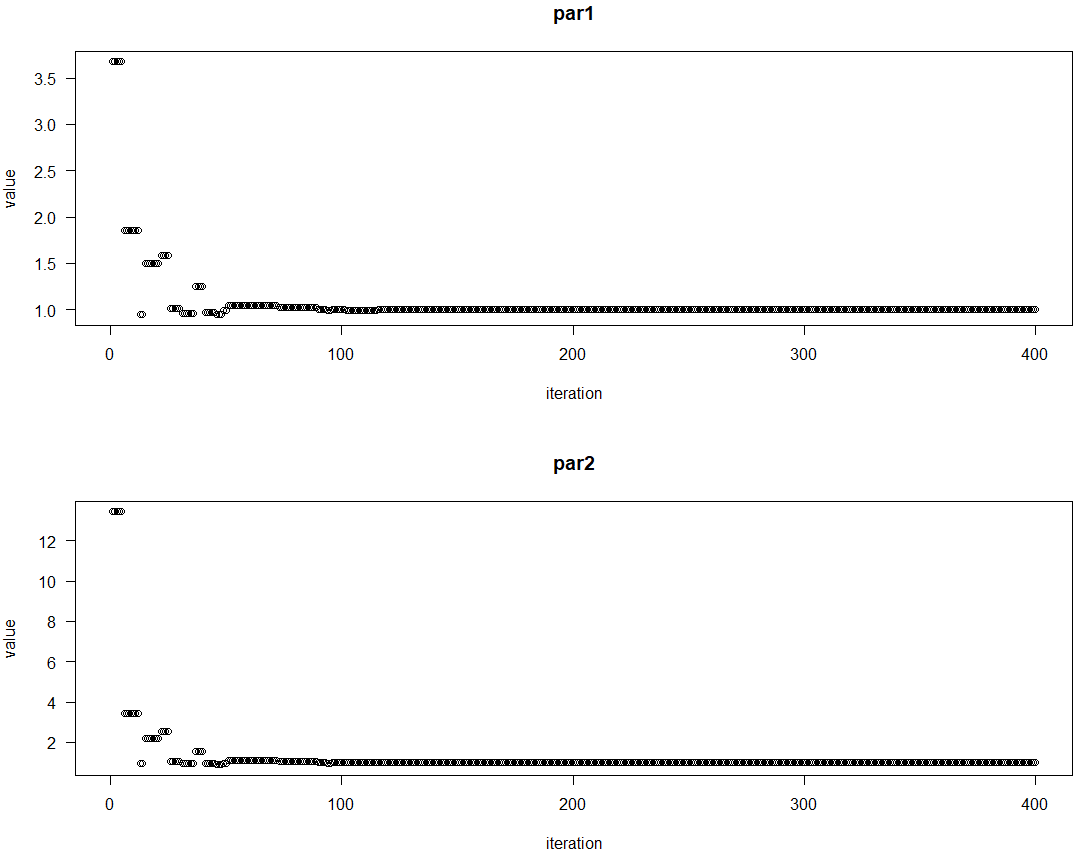
Iteration: 398 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 399 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 400 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

> # plotting the output

> plot(outputDEopim2)



> # plotting the output

> plot(outputDEopim2)

> # Using strategy 3

> # DEoptim function is use to performs evolutionary global optimization using differential evolution algorithm.

> # the function takes four parameters first parameter takes the function for which we want to perform optimization,

> # second parameter takes lower bound for each variable in the function, third parameter upper takes upper bound value for each variable in function,

> # fourth parameter DEoptim.control function which 5 input parameters first parameter takes number of population members,

> # second parameter is strategy which integer corresponding with different differential evolution strategies,

> # third parameter is itermax which takes maximum number of iterations allowed, forth parameter is F it takes differential weigth from interval 0 to 2,

> # fifth parameter is CR takes crossover probability between 0 and 2.

> outputDEopim3 = DEoptim(Rosenbrock, lower = lbound, upper = ubound, DEoptim.control(NP = 100, strategy = 3, itermax = 400, F = 1.2, CR = 0.7))

Iteration: 1 bestvalit: 16.261350 bestmemit: 3.044339 9.615592

Iteration: 2 bestvalit: 16.261350 bestmemit: 3.044339 9.615592

Iteration: 3 bestvalit: 1.726643 bestmemit: -0.077314 0.081213

Iteration: 4 bestvalit: 1.726643 bestmemit: -0.077314 0.081213

Iteration: 5 bestvalit: 0.367650 bestmemit: 0.433908 0.210000

Iteration: 6 bestvalit: 0.367650 bestmemit: 0.433908 0.210000

Iteration: 7 bestvalit: 0.367650 bestmemit: 0.433908 0.210000

Iteration: 8 bestvalit: 0.367650 bestmemit: 0.433908 0.210000

Iteration: 9 bestvalit: 0.367650 bestmemit: 0.433908 0.210000

Iteration: 10 bestvalit: 0.367650 bestmemit: 0.433908 0.210000

Iteration: 11 bestvalit: 0.367375 bestmemit: 0.433923 0.209953

Iteration: 12 bestvalit: 0.367375 bestmemit: 0.433923 0.209953

Iteration: 13 bestvalit: 0.367375 bestmemit: 0.433923 0.209953

Iteration: 14 bestvalit: 0.367375 bestmemit: 0.433923 0.209953

Iteration: 15 bestvalit: 0.135248 bestmemit: 0.645794 0.407157

Iteration: 16 bestvalit: 0.135248 bestmemit: 0.645794 0.407157

Iteration: 17 bestvalit: 0.135248 bestmemit: 0.645794 0.407157

Iteration: 18 bestvalit: 0.064037 bestmemit: 0.748101 0.562072

Iteration: 19 bestvalit: 0.064037 bestmemit: 0.748101 0.562072

Iteration: 20 bestvalit: 0.064037 bestmemit: 0.748101 0.562072

Iteration: 21 bestvalit: 0.063886 bestmemit: 0.748107 0.561752

Iteration: 22 bestvalit: 0.063886 bestmemit: 0.748107 0.561752

Iteration: 23 bestvalit: 0.059388 bestmemit: 1.183569 1.384806

Iteration: 24 bestvalit: 0.059388 bestmemit: 1.183569 1.384806

Iteration: 25 bestvalit: 0.059388 bestmemit: 1.183569 1.384806

Iteration: 26 bestvalit: 0.059388 bestmemit: 1.183569 1.384806

Iteration: 27 bestvalit: 0.059388 bestmemit: 1.183569 1.384806

Iteration: 28 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 29 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 30 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 31 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 32 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 33 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 34 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 35 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 36 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 37 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 38 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 39 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 40 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 41 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 42 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 43 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 44 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 45 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 46 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 47 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 48 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 49 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 50 bestvalit: 0.000998 bestmemit: 1.007971 1.019063

Iteration: 51 bestvalit: 0.000719 bestmemit: 0.990877 0.984360

Iteration: 52 bestvalit: 0.000719 bestmemit: 0.990877 0.984360

Iteration: 53 bestvalit: 0.000719 bestmemit: 0.990877 0.984360

Iteration: 54 bestvalit: 0.000719 bestmemit: 0.990877 0.984360

Iteration: 55 bestvalit: 0.000719 bestmemit: 0.990877 0.984360

Iteration: 56 bestvalit: 0.000546 bestmemit: 0.982972 0.967833

Iteration: 57 bestvalit: 0.000546 bestmemit: 0.982972 0.967833

Iteration: 58 bestvalit: 0.000462 bestmemit: 1.020775 1.042531

Iteration: 59 bestvalit: 0.000222 bestmemit: 1.009749 1.020719

Iteration: 60 bestvalit: 0.000222 bestmemit: 1.009749 1.020719

Iteration: 61 bestvalit: 0.000139 bestmemit: 0.988781 0.977326

Iteration: 62 bestvalit: 0.000139 bestmemit: 0.988781 0.977326

Iteration: 63 bestvalit: 0.000010 bestmemit: 0.999287 0.998259

Iteration: 64 bestvalit: 0.000010 bestmemit: 0.999287 0.998259

Iteration: 65 bestvalit: 0.000010 bestmemit: 0.999287 0.998259

Iteration: 66 bestvalit: 0.000010 bestmemit: 0.999287 0.998259

Iteration: 67 bestvalit: 0.000010 bestmemit: 0.999287 0.998259

Iteration: 68 bestvalit: 0.000010 bestmemit: 0.999287 0.998259

Iteration: 69 bestvalit: 0.000010 bestmemit: 0.999287 0.998259

Iteration: 70 bestvalit: 0.000010 bestmemit: 0.999287 0.998259

Iteration: 71 bestvalit: 0.000010 bestmemit: 0.999287 0.998259

Iteration: 72 bestvalit: 0.000010 bestmemit: 0.999287 0.998259

Iteration: 73 bestvalit: 0.000010 bestmemit: 0.999287 0.998259

Iteration: 74 bestvalit: 0.000010 bestmemit: 0.999287 0.998259

Iteration: 75 bestvalit: 0.000010 bestmemit: 0.999287 0.998259

Iteration: 76 bestvalit: 0.000010 bestmemit: 0.999287 0.998259

Iteration: 77 bestvalit: 0.000001 bestmemit: 1.000689 1.001430

Iteration: 78 bestvalit: 0.000001 bestmemit: 1.000689 1.001430

Iteration: 79 bestvalit: 0.000001 bestmemit: 1.000689 1.001430

Iteration: 80 bestvalit: 0.000001 bestmemit: 1.000689 1.001430

Iteration: 81 bestvalit: 0.000001 bestmemit: 1.000689 1.001430

Iteration: 82 bestvalit: 0.000001 bestmemit: 1.000689 1.001430

Iteration: 83 bestvalit: 0.000001 bestmemit: 1.000689 1.001430

Iteration: 84 bestvalit: 0.000001 bestmemit: 1.000689 1.001430

Iteration: 85 bestvalit: 0.000001 bestmemit: 1.000689 1.001430

Iteration: 86 bestvalit: 0.000000 bestmemit: 0.999878 0.999687

Iteration: 87 bestvalit: 0.000000 bestmemit: 0.999878 0.999687

Iteration: 88 bestvalit: 0.000000 bestmemit: 0.999878 0.999687

Iteration: 89 bestvalit: 0.000000 bestmemit: 0.999878 0.999687

Iteration: 90 bestvalit: 0.000000 bestmemit: 0.999878 0.999687

Iteration: 91 bestvalit: 0.000000 bestmemit: 0.999878 0.999687

Iteration: 92 bestvalit: 0.000000 bestmemit: 0.999878 0.999687

Iteration: 93 bestvalit: 0.000000 bestmemit: 1.000027 1.000057

Iteration: 94 bestvalit: 0.000000 bestmemit: 1.000027 1.000057

Iteration: 95 bestvalit: 0.000000 bestmemit: 1.000027 1.000057

Iteration: 96 bestvalit: 0.000000 bestmemit: 1.000027 1.000057

Iteration: 97 bestvalit: 0.000000 bestmemit: 1.000027 1.000057

Iteration: 98 bestvalit: 0.000000 bestmemit: 1.000027 1.000057

Iteration: 99 bestvalit: 0.000000 bestmemit: 1.000027 1.000057

Iteration: 100 bestvalit: 0.000000 bestmemit: 1.000027 1.000057

Iteration: 101 bestvalit: 0.000000 bestmemit: 1.000027 1.000057

Iteration: 102 bestvalit: 0.000000 bestmemit: 1.000027 1.000057

Iteration: 103 bestvalit: 0.000000 bestmemit: 1.000027 1.000057

Iteration: 104 bestvalit: 0.000000 bestmemit: 1.000027 1.000057

Iteration: 105 bestvalit: 0.000000 bestmemit: 1.000027 1.000057

Iteration: 106 bestvalit: 0.000000 bestmemit: 1.000027 1.000057

Iteration: 107 bestvalit: 0.000000 bestmemit: 1.000033 1.000067

Iteration: 108 bestvalit: 0.000000 bestmemit: 1.000033 1.000067

Iteration: 109 bestvalit: 0.000000 bestmemit: 1.000033 1.000067

Iteration: 110 bestvalit: 0.000000 bestmemit: 1.000033 1.000067

Iteration: 111 bestvalit: 0.000000 bestmemit: 1.000009 1.000019

Iteration: 112 bestvalit: 0.000000 bestmemit: 1.000009 1.000019

Iteration: 113 bestvalit: 0.000000 bestmemit: 1.000009 1.000019

Iteration: 114 bestvalit: 0.000000 bestmemit: 1.000009 1.000019

Iteration: 115 bestvalit: 0.000000 bestmemit: 1.000009 1.000019

Iteration: 116 bestvalit: 0.000000 bestmemit: 1.000009 1.000019

Iteration: 117 bestvalit: 0.000000 bestmemit: 1.000009 1.000019

Iteration: 118 bestvalit: 0.000000 bestmemit: 1.000009 1.000019

Iteration: 119 bestvalit: 0.000000 bestmemit: 1.000009 1.000019

Iteration: 120 bestvalit: 0.000000 bestmemit: 1.000009 1.000019

Iteration: 121 bestvalit: 0.000000 bestmemit: 1.000000 0.999999

Iteration: 122 bestvalit: 0.000000 bestmemit: 1.000000 0.999999

Iteration: 123 bestvalit: 0.000000 bestmemit: 1.000000 0.999999

Iteration: 124 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 125 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 126 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 127 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 128 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 129 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 130 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 131 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 132 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 133 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 134 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 135 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 136 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 137 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 138 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 139 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 140 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 141 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 142 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 143 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 144 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 145 bestvalit: 0.000000 bestmemit: 1.000002 1.000004

Iteration: 146 bestvalit: 0.000000 bestmemit: 1.000001 1.000002

Iteration: 147 bestvalit: 0.000000 bestmemit: 1.000001 1.000002

Iteration: 148 bestvalit: 0.000000 bestmemit: 1.000001 1.000002

Iteration: 149 bestvalit: 0.000000 bestmemit: 0.999999 0.999999

Iteration: 150 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 151 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 152 bestvalit: 0.000000 bestmemit: 1.000000 1.000001

Iteration: 153 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 154 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 155 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 156 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 157 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 158 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 159 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 160 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 161 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 162 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 163 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 164 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 165 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 166 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 167 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 168 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 169 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 170 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 171 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 172 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 173 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 174 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 175 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 176 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 177 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 178 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 179 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 180 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 181 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 182 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 183 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 184 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 185 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 186 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 187 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 188 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 189 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 190 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 191 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 192 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 193 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 194 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 195 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 196 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 197 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 198 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 199 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 200 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 201 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 202 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 203 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 204 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 205 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 206 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 207 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 208 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 209 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 210 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 211 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 212 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 213 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 214 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 215 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 216 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 217 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 218 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 219 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 220 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 221 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 222 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 223 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 224 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 225 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 226 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 227 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 228 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 229 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 230 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 231 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 232 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 233 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 234 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 235 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 236 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 237 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 238 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 239 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 240 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 241 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 242 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 243 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 244 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 245 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 246 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 247 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 248 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 249 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 250 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 251 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 252 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 253 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 254 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 255 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 256 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 257 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 258 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 259 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 260 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 261 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 262 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 263 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 264 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 265 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 266 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 267 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 268 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 269 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 270 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 271 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 272 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 273 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 274 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 275 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 276 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 277 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 278 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 279 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 280 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 281 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 282 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 283 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 284 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 285 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 286 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 287 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 288 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 289 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 290 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 291 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 292 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 293 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 294 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 295 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 296 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 297 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 298 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 299 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 300 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 301 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 302 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 303 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 304 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 305 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 306 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 307 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 308 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 309 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 310 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 311 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 312 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 313 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 314 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 315 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 316 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 317 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 318 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 319 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 320 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 321 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 322 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 323 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 324 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 325 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 326 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 327 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 328 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 329 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 330 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 331 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 332 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 333 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 334 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 335 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 336 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 337 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 338 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 339 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 340 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 341 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 342 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 343 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 344 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 345 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 346 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 347 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 348 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 349 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 350 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 351 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 352 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 353 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 354 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 355 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 356 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 357 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 358 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 359 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 360 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 361 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 362 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 363 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 364 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 365 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 366 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 367 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 368 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 369 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 370 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 371 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 372 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 373 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 374 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 375 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 376 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 377 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 378 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 379 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 380 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 381 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 382 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 383 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 384 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 385 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 386 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 387 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 388 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 389 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 390 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 391 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 392 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 393 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 394 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 395 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 396 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 397 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

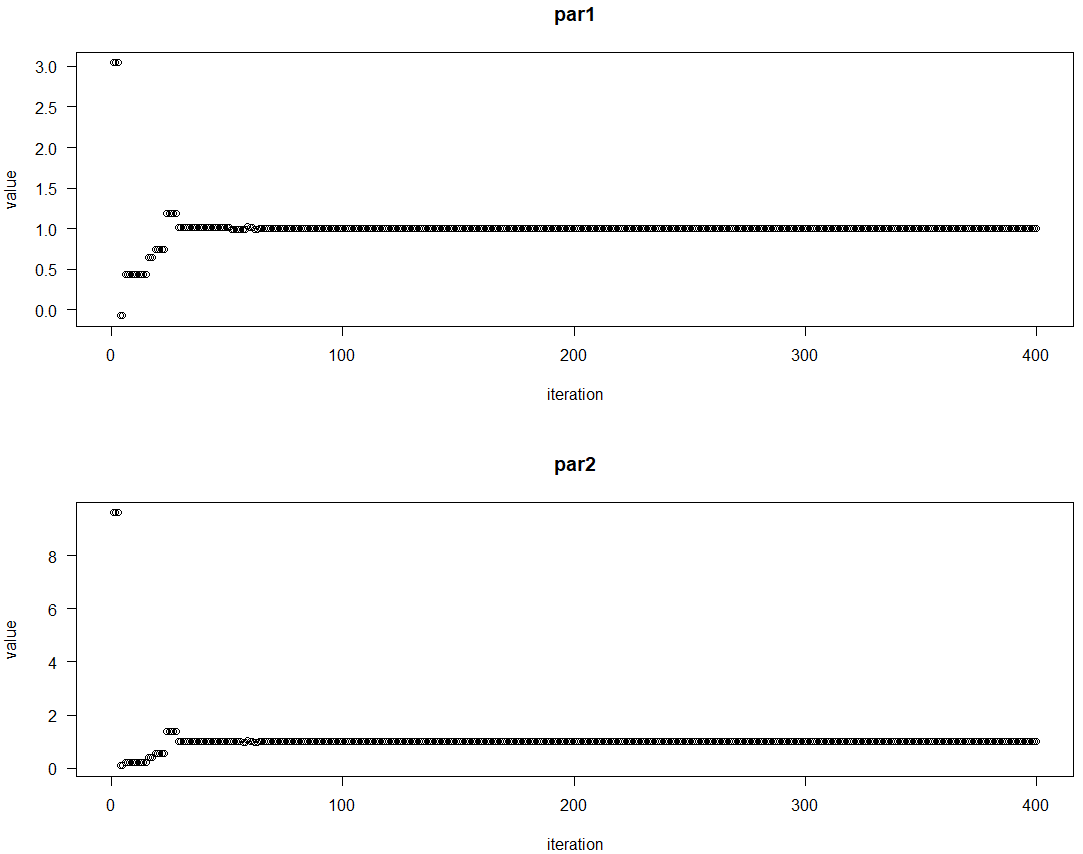
Iteration: 398 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 399 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

Iteration: 400 bestvalit: 0.000000 bestmemit: 1.000000 1.000000

> # plotting the output

> plot(outputDEopim3)



**Name: Sanesh Sagvekar Roll No: 170122**

**Practical No.: 6**

**AIM: Implement general differential evolution algorithm.**

> # loading DEoptim library

> library("DEoptim")

> # rosebrock store the function which has two variables x1 and x2.

> Rosenbrock = function(x)

+ {

+ x1 = x[1]

+ x2 = x[2]

+ 50 \* (x2 - x1 \* x1)^2 + (1 - x1)^2

+ }

> # set the initial value of seed to 100 which helps in giving same output each time we run the program.

> set.seed(123)

>

> # lbound is the lower bound of the search space.

> lbound = c(-10, -10)

> # ubound is the upper bound of the search space.

> ubound = c(10, 10)

> # Using strategy DE / current-to-p-best / 1 (6)

> # DEoptim function is use to performs evolutionary global optimization using differential evolution algorithm.

> # the function takes four parameters first parameter takes the function for which we want to perform optimization,

> # second parameter takes lower bound for each variable in the function, third parameter upper takes upper bound value for each variable in function,

> # fourth parameter DEoptim.control function which 5 input parameters first parameter takes number of population members,

> # second parameter is strategy which integer corresponding with different differential evolution strategies,

> # third parameter is itermax which takes maximum number of iterations allowed, forth parameter is F it takes differential weigth from interval 0 to 2,

> # fifth parameter is CR takes crossover probability between 0 and 2.

> outputDEopim = DEoptim(Rosenbrock, lower = lbound, upper = ubound, DEoptim.control(NP = 100, strategy = 6, itermax = 400, F = 1.2, CR = 0.7))

Iteration: 1 bestvalit: 11.587446 bestmemit: -1.029673 0.673759

Iteration: 2 bestvalit: 1.511849 bestmemit: 1.882840 3.424056

Iteration: 3 bestvalit: 1.511849 bestmemit: 1.882840 3.424056

Iteration: 4 bestvalit: 1.511849 bestmemit: 1.882840 3.424056

Iteration: 5 bestvalit: 1.511849 bestmemit: 1.882840 3.424056

Iteration: 6 bestvalit: 1.511849 bestmemit: 1.882840 3.424056

Iteration: 7 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 8 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 9 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 10 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 11 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 12 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 13 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 14 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 15 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 16 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 17 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 18 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 19 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 20 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 21 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 22 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 23 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 24 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 25 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 26 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 27 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 28 bestvalit: 0.389559 bestmemit: 1.082412 1.084121

Iteration: 29 bestvalit: 0.357109 bestmemit: 0.435839 0.217824

Iteration: 30 bestvalit: 0.357109 bestmemit: 0.435839 0.217824

Iteration: 31 bestvalit: 0.357109 bestmemit: 0.435839 0.217824

Iteration: 32 bestvalit: 0.357109 bestmemit: 0.435839 0.217824

Iteration: 33 bestvalit: 0.357109 bestmemit: 0.435839 0.217824

Iteration: 34 bestvalit: 0.357109 bestmemit: 0.435839 0.217824

Iteration: 35 bestvalit: 0.357109 bestmemit: 0.435839 0.217824

Iteration: 36 bestvalit: 0.357109 bestmemit: 0.435839 0.217824

Iteration: 37 bestvalit: 0.357109 bestmemit: 0.435839 0.217824

Iteration: 38 bestvalit: 0.357109 bestmemit: 0.435839 0.217824

Iteration: 39 bestvalit: 0.357109 bestmemit: 0.435839 0.217824

Iteration: 40 bestvalit: 0.357109 bestmemit: 0.435839 0.217824

Iteration: 41 bestvalit: 0.357109 bestmemit: 0.435839 0.217824

Iteration: 42 bestvalit: 0.357109 bestmemit: 0.435839 0.217824

Iteration: 43 bestvalit: 0.233604 bestmemit: 0.537726 0.309103

Iteration: 44 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 45 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 46 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 47 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 48 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 49 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 50 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 51 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 52 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 53 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 54 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 55 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 56 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 57 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 58 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 59 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 60 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 61 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 62 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 63 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 64 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 65 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 66 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 67 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 68 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 69 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 70 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 71 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 72 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 73 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 74 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 75 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 76 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 77 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 78 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 79 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 80 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 81 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 82 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 83 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 84 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 85 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 86 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 87 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 88 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 89 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 90 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 91 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 92 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 93 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 94 bestvalit: 0.045264 bestmemit: 1.157507 1.360048

Iteration: 95 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 96 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 97 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 98 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 99 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 100 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 101 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 102 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 103 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 104 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 105 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 106 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 107 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 108 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 109 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 110 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 111 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 112 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 113 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 114 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 115 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 116 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 117 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 118 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 119 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 120 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 121 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 122 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 123 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 124 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 125 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 126 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 127 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 128 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 129 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 130 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 131 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 132 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 133 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 134 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 135 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 136 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 137 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 138 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 139 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 140 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 141 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 142 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 143 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 144 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 145 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 146 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 147 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 148 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 149 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 150 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 151 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 152 bestvalit: 0.015782 bestmemit: 0.905428 0.808106

Iteration: 153 bestvalit: 0.006982 bestmemit: 1.013169 1.038180

Iteration: 154 bestvalit: 0.006982 bestmemit: 1.013169 1.038180

Iteration: 155 bestvalit: 0.006982 bestmemit: 1.013169 1.038180

Iteration: 156 bestvalit: 0.006982 bestmemit: 1.013169 1.038180

Iteration: 157 bestvalit: 0.006982 bestmemit: 1.013169 1.038180

Iteration: 158 bestvalit: 0.006982 bestmemit: 1.013169 1.038180

Iteration: 159 bestvalit: 0.006982 bestmemit: 1.013169 1.038180

Iteration: 160 bestvalit: 0.006982 bestmemit: 1.013169 1.038180

Iteration: 161 bestvalit: 0.005151 bestmemit: 1.012513 1.035177

Iteration: 162 bestvalit: 0.005151 bestmemit: 1.012513 1.035177

Iteration: 163 bestvalit: 0.005151 bestmemit: 1.012513 1.035177

Iteration: 164 bestvalit: 0.005151 bestmemit: 1.012513 1.035177

Iteration: 165 bestvalit: 0.005151 bestmemit: 1.012513 1.035177

Iteration: 166 bestvalit: 0.005151 bestmemit: 1.012513 1.035177

Iteration: 167 bestvalit: 0.005151 bestmemit: 1.012513 1.035177

Iteration: 168 bestvalit: 0.005151 bestmemit: 1.012513 1.035177

Iteration: 169 bestvalit: 0.005151 bestmemit: 1.012513 1.035177

Iteration: 170 bestvalit: 0.005151 bestmemit: 1.012513 1.035177

Iteration: 171 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 172 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 173 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 174 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 175 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 176 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 177 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 178 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 179 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 180 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 181 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 182 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 183 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 184 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 185 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 186 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 187 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 188 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 189 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 190 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 191 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 192 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 193 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 194 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 195 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 196 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 197 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 198 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 199 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 200 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 201 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 202 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 203 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 204 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 205 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 206 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 207 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 208 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 209 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 210 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 211 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 212 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 213 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 214 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 215 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 216 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 217 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 218 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 219 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 220 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 221 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 222 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 223 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 224 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 225 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 226 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 227 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 228 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 229 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 230 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 231 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 232 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 233 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 234 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 235 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 236 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 237 bestvalit: 0.001662 bestmemit: 1.029984 1.056960

Iteration: 238 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 239 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 240 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 241 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 242 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 243 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 244 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 245 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 246 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 247 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 248 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 249 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 250 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 251 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 252 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 253 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 254 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 255 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 256 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 257 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 258 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 259 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 260 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 261 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 262 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 263 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 264 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 265 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 266 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 267 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 268 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 269 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 270 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 271 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 272 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 273 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 274 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 275 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 276 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 277 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 278 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 279 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 280 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 281 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 282 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 283 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 284 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 285 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 286 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 287 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 288 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 289 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 290 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 291 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 292 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 293 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 294 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 295 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 296 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 297 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 298 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 299 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 300 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 301 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 302 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 303 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 304 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 305 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 306 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 307 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 308 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 309 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 310 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 311 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 312 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 313 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 314 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 315 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 316 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 317 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 318 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 319 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 320 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 321 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 322 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 323 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 324 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 325 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 326 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 327 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 328 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 329 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 330 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 331 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 332 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 333 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 334 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 335 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 336 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 337 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 338 bestvalit: 0.000264 bestmemit: 0.983822 0.967678

Iteration: 339 bestvalit: 0.000229 bestmemit: 1.015117 1.030435

Iteration: 340 bestvalit: 0.000229 bestmemit: 1.015117 1.030435

Iteration: 341 bestvalit: 0.000229 bestmemit: 1.015117 1.030435

Iteration: 342 bestvalit: 0.000229 bestmemit: 1.015117 1.030435

Iteration: 343 bestvalit: 0.000229 bestmemit: 1.015117 1.030435

Iteration: 344 bestvalit: 0.000229 bestmemit: 1.015117 1.030435

Iteration: 345 bestvalit: 0.000229 bestmemit: 1.015117 1.030435

Iteration: 346 bestvalit: 0.000229 bestmemit: 1.015117 1.030435

Iteration: 347 bestvalit: 0.000229 bestmemit: 1.015117 1.030435

Iteration: 348 bestvalit: 0.000229 bestmemit: 1.015117 1.030435

Iteration: 349 bestvalit: 0.000229 bestmemit: 1.015117 1.030435

Iteration: 350 bestvalit: 0.000229 bestmemit: 1.015117 1.030435

Iteration: 351 bestvalit: 0.000229 bestmemit: 1.015117 1.030435

Iteration: 352 bestvalit: 0.000229 bestmemit: 1.015117 1.030435

Iteration: 353 bestvalit: 0.000104 bestmemit: 1.009890 1.020222

Iteration: 354 bestvalit: 0.000104 bestmemit: 1.009890 1.020222

Iteration: 355 bestvalit: 0.000104 bestmemit: 1.009890 1.020222

Iteration: 356 bestvalit: 0.000104 bestmemit: 1.009890 1.020222

Iteration: 357 bestvalit: 0.000104 bestmemit: 1.009890 1.020222

Iteration: 358 bestvalit: 0.000104 bestmemit: 1.009890 1.020222

Iteration: 359 bestvalit: 0.000104 bestmemit: 1.009890 1.020222

Iteration: 360 bestvalit: 0.000104 bestmemit: 1.009890 1.020222

Iteration: 361 bestvalit: 0.000104 bestmemit: 1.009890 1.020222

Iteration: 362 bestvalit: 0.000048 bestmemit: 0.993457 0.986649

Iteration: 363 bestvalit: 0.000048 bestmemit: 0.993457 0.986649

Iteration: 364 bestvalit: 0.000048 bestmemit: 0.993457 0.986649

Iteration: 365 bestvalit: 0.000048 bestmemit: 0.993457 0.986649

Iteration: 366 bestvalit: 0.000048 bestmemit: 0.993457 0.986649

Iteration: 367 bestvalit: 0.000048 bestmemit: 0.993457 0.986649

Iteration: 368 bestvalit: 0.000048 bestmemit: 0.993457 0.986649

Iteration: 369 bestvalit: 0.000048 bestmemit: 0.993457 0.986649

Iteration: 370 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 371 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 372 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 373 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 374 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 375 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 376 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 377 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 378 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 379 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 380 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 381 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 382 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 383 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 384 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 385 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 386 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 387 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 388 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 389 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 390 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 391 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 392 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 393 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 394 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 395 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 396 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 397 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

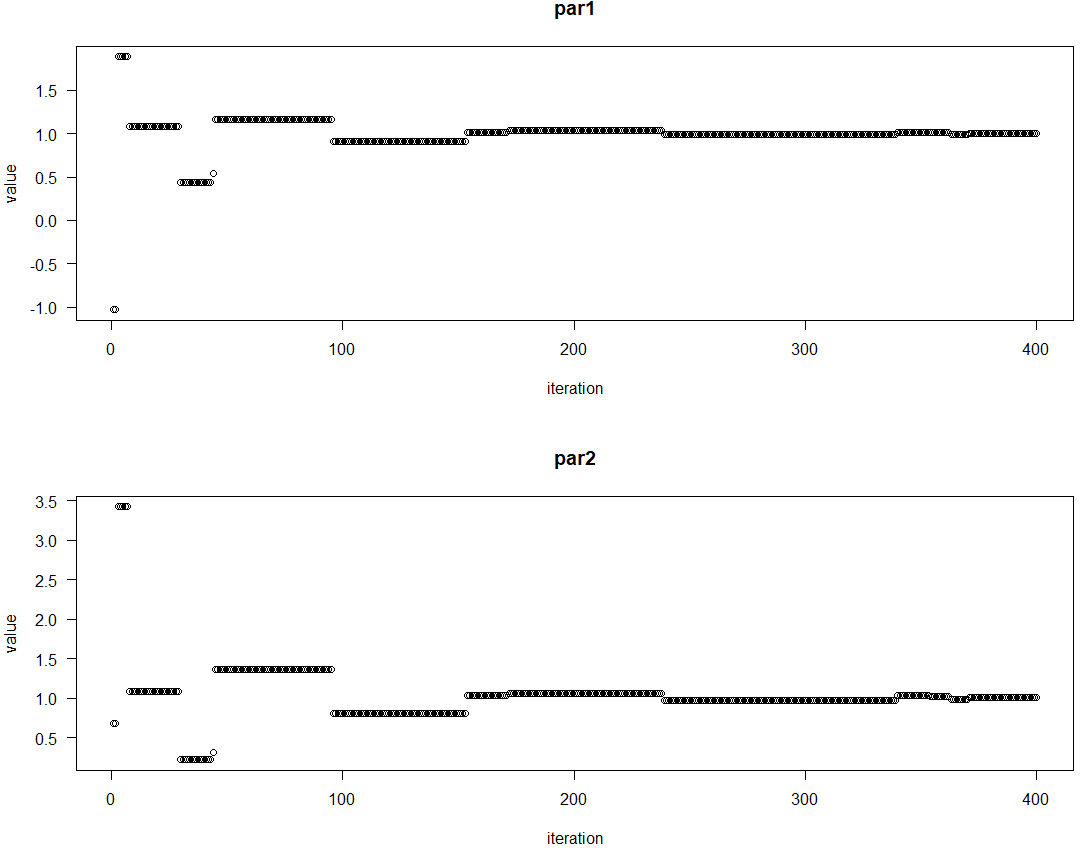
Iteration: 398 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 399 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

Iteration: 400 bestvalit: 0.000009 bestmemit: 1.002164 1.004053

> # ploting the output

> plot(outputDEopim)



**Name: Sanesh Sagvekar Roll No: 170122**

**Practical No.: 7**

**AIM: Implement gbest and lbest of PSO.**

> #install.packages("pso")

> library("pso")

> # f stores function to be optimize with one variable.

> f = function(x) 20 + sum(x^2 - 10 \* cos(2 \* pi \* x))

> # lower bound on the variable.

> lbound = -5

> # upper bound on the variable.

> ubound = 5

> # local exploration constant.

> # psoptim function is used to implement particle swarm optimization.

> # function takes five parameter first parameter is par which takes vector with length defining the dimensionality of the function,

> # second parameter is fn it takes the function to be optimized, third parameter is lower it is a lower bound of the variable,

> # fourth parameter is upper which takes upper bound of the variable, fifth parameter takes list of control parameters.

> psoptim(par = rep(NA, 2), fn = f, lower = lbound, upper = ubound, control = list(abstol = 1e-8, c.p = 0.5 + log(2)))

$`par`

[1] -5.693785e-06 -1.267103e-06

$value

[1] 6.750241e-09

$counts

function iteration restarts

1440 120 0

$convergence

[1] 0

$message

[1] "Converged"

>

> # global exploration constant.

> # psoptim function is used to implement particle swarm optimization.

> # function takes five parameter first parameter is par which takes vector with length defining the dimensionality of the function,

> # second parameter is fn it takes the function to be optimized, third parameter is lower it is a lower bound of the variable,

> # fourth parameter is upper which takes upper bound of the variable, fifth parameter takes list of control parameters.

> psoptim(par = rep(NA, 2), fn = f, lower = lbound, upper = ubound, control = list(abstol = 1e-8, c.g = 0.5 + log(2)))

$`par`

[1] -1.104314e-06 8.780133e-07

$value

[1] 3.948841e-10

$counts

function iteration restarts

1704 142 0

$convergence

[1] 0

$message

[1] "Converged"

**Name: Sanesh Sagvekar Roll No: 170122**

**Practical No.: 8**

**AIM: Implement simple Ant colony optimization algorithm.**

> #install.packages("ShortForm")

> # loading ShortForm library

> library("ShortForm")

>

> # antcolony.lavaan function is used for implementing any colony optimization algorithm which can produce short forms of scales that are optimized with respect to characterisitcs.

> # the function takes fourteen parameters as a input, first parameter is data on which we want to apply ant colony optimization here we are using holzinger and swineford dataset consist of mental ability test scores of seventh and eighth grade studnets,

> # second parameter is ants which takes a number of ants to send per iteration, third parameter is evaporation which takes percentage of the pheremone that is retained after evaporation between steps between algorithm,

> # fourth parameter is antModel which takes lavaan formatted model, fifth parameter is list.items it takes a list of containing one or more character vectors of item name of each factor,

> # sixth parameter is full which takes a numeric value indicating the total number of unique items in the test, seventh parameter is fit.indices which takes the indices from the lavaan model for model optimization,

> # eigth parameter is i.per.f which takes vector with number of items per factor in the same order of factor and list.item, ninth parameter is factors which takes name of the factors in the same order of i.per.f and list.item,

> # tenth parameter is steps which sets the stopping rule, eleventh parameter is fit.statistics.test it takes a character vector of the logical test for model optimization, twelfth parameter is summary it takes file name to generate summary,

> # thirteenth parameter is feedbackfile it takes file name to generate feedback file, fourteenth parameter is max.run takes the maximum number of times ants run in algorithm.

> abilityShortForm1 = antcolony.lavaan(data = lavaan::HolzingerSwineford1939,

+ ants = 1, evaporation = 0.7,

+ antModel = ' visual =~ x1 + x2 + x3

+ textual =~ x4 + x5 + x6

+ speed =~ x7 + x8 + x9',

+ list.items = list(c('x1', 'x2', 'x3'),

+ c('x4', 'x5', 'x6'),

+ c('x7', 'x8', 'x9')), full = 9, i.per.f =

+ c(3,3,3), factors = c('visual', 'textual', 'speed'), steps = 1, fit.indices =

+ c('cfi'), fit.statistics.test = "(cfi > 0.6)", summaryfile = NULL, feedbackfile = NULL, max.run = 2)

Run number 2. [1] "Compiling results."

Warning message:

In antcolony.lavaan(data = lavaan::HolzingerSwineford1939, ants = 1, :

Max runs reached! Problems converging onto a solution.

>

> # exampleAntModel is a character vector containing the model syntax for the one factor.

> data(exampleAntModel)

> characterVectorAntModel = exampleAntModel

>

> # listItems stores the list of one or more character vectors of item name for each factor.

> listItems = list(c('Item1','Item2','Item3','Item4','Item5',

+ 'Item6','Item7','Item8','Item9','Item10',

+ 'Item11','Item12','Item13','Item14','Item15',

+ 'Item16','Item17','Item18','Item19','Item20',

+ 'Item21','Item22','Item23','Item24','Item25',

+ 'Item26','Item27','Item28','Item29','Item30',

+ 'Item31','Item32','Item33','Item34','Item35',

+ 'Item36','Item37','Item38','Item39','Item40',

+ 'Item41','Item42','Item43','Item44','Item45',

+ 'Item46','Item47','Item48','Item49','Item50',

+ 'Item51','Item52','Item53','Item54','Item55','Item56'))

>

> # simulated\_test\_data is dataset based on a standardized test it has simulated response patterns, abilities and outcomes based on a unidimensional state issued stanadardized test.

> data(simulated\_test\_data)

> dataframe = simulated\_test\_data

>

> # antcolony.lavaan function is used for implementing any colony optimization algorithm which can produce short forms of scales that are optimized with respect to characterisitcs.

> # the function takes fourteen parameters as a input, first parameter is data on which we want to apply ant colony optimization here we are using holzinger and swineford dataset consist of mental ability test scores of seventh and eighth grade studnets,

> # second parameter is ants which takes a number of ants to send per iteration, third parameter is evaporation which takes percentage of the pheremone that is retained after evaporation between steps between algorithm,

> # fourth parameter is antModel which takes lavaan formatted model, fifth parameter is list.items it takes a list of containing one or more character vectors of item name of each factor,

> # sixth parameter is full which takes a numeric value indicating the total number of unique items in the test, seventh parameter is fit.indices which takes the indices from the lavaan model for model optimization,

> # eigth parameter is i.per.f which takes vector with number of items per factor in the same order of factor and list.item, ninth parameter is factors which takes name of the factors in the same order of i.per.f and list.item,

> # tenth parameter is steps which sets the stopping rule, eleventh parameter is fit.statistics.test it takes a character vector of the logical test for model optimization, twelfth parameter is summary it takes file name to generate summary,

> # thirteenth parameter is feedbackfile it takes file name to generate feedback file, fourteenth parameter is max.run takes the maximum number of times ants run in algorithm.

> abilityShortForm1 = antcolony.lavaan(data = dataframe,

+ ants = 5, evaporation = 0.7,

+ antModel = characterVectorAntModel,

+ list.items = listItems, full = 56, i.per.f =

+ 20, factors = 'Ability', steps = 3, fit.indices =

+ c('cfi', 'rmsea'), fit.statistics.test = "(cfi > 0.95)&(rmsea < 0.05)", summaryfile = 'summary.txt', feedbackfile = 'iteration.html', max.run = 500)

Run number 31. [1] "Compiling results."