International Journal of Scientific & Engineering Research, Volume 5, Issue 5, May-2014
ISSN 2229-5518 

# Solve Four Shapes Problem using Evolution Algorithm

Lubna zaghlul bashir

**Abstract-** Genetic algorithms evaluate the target function to be optimized at some randomly selected points of the definition domain. Taking this information into account, a new set of points (a new population) is generated. Gradually the points in the population approach local maxima and minima of the function. Genetic algorithms can be used when no information is available about the gradient of the function at the evaluated points. The function itself does not need to be continuous or differentiable. Genetic algorithms can still achieve good results even in cases in which the function has several local minima or maxima.

In this work we use Genetic Algorithm s to determine the best location of four shapes,

The problem as follows: there are four shapes, we want to find a way to place them on a board divided into 16 grid units so that no two shapes attach each others.

Experimentally results shows that the genetic algorithm have the ability to find optimal solution or find solutions nearby optimal solutions. And The Genetic Algorithm is well suited to and has been extensively applied to solve complex design optimization problems because it can handle both discrete and continuous variables, and nonlinear objective and constrain functions without requiring gradient information.

| | | | |
|---|---|---|---|
| | | | ♦ |
| ♠ | | | |
| | | ♥ | |
| | ♣ | | |

**Index terms-** genetic algorithm, evolutionary algorithms, optimization, population.



## 1 GENETIC ALGORITHMS

Genetic algorithms are stochastic search techniques that guide a population of solutions towards an optimum using the principles of evolution and natural genetics. In recent years, genetic algorithms have become a popular optimization tool for many areas of research, including the field of system control, control design, science and engineering. Significant research exists concerning genetic algorithms for control design and off-line controller analysis. [1]

Genetic algorithms are inspired by the evolution of populations. In a particular environment, individuals which better fit the environment will be able to survive and hand down their chromosomes to their descendants, while less fit individuals will become extinct. The aim of genetic algorithms is to use simple representations to encode complex structures and simple operations to improve these structures. Genetic algorithms therefore are characterized by their representation and operators. In the original genetic algorithm an individual chromosome is represented by a binary string. The bits of each string are called genes and their varying values alleles. A group of individual chromosomes are called a population. Basic genetic operators include reproduction, crossover and mutation. Genetic algorithms are especially capable of handling problems in which the objective function is discontinuous or non differentiable, non convex, multimodal or noisy. Since the algorithms operate on a population instead of a single point in the search space, they climb many peaks in parallel and therefore reduce the probability of finding local minima.[1]

## 2 BIOLOGICAL CONCEPTS IN GENETIC ALGORITHM

**Gene:** a part of chromosome ,a gene contains a part of solutions. it determines the solution .e.g.16743 is chromosome and 1,6,7,4,3 are its genes.
**Chromosome:** a set of genes, a chromosome contains the solution in form of genes.
**Individual:** same as chromosome.
**Population:** number of individuals. Present with same length of chromosome.
**Fitness:** the value assigned to an individual based on how far or close a individual is form the solution greater the fitness value better the solution it contains.
**Fitness function:** a function that assigns fitness value to the individual. it is problem specific.

**Breeding:** taking two fitness individuals and then intermingling there chromosome to create new two individuals.
**Mutation**: changing a random gene in an individual.

**Selection:** selecting individuals for creating the next generation.[2,3]

## 3 GA Operators

The simplest form of genetic algorithm involves three types of operators: selection, crossover (single point), and mutation.

**Selection** This operator selects chromosomes in the population for reproduction. The fitter the chromosome, the more times it is likely to be selected to reproduce.

**Crossover** This operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring. For example, the strings 10000100 and 11111111 could be crossed over after the third locus in each to produce the two offspring 10011111 and 11100100. The crossover operator roughly mimics biological recombination between two single−chromosome (haploid) organisms.

**Mutation** This operator randomly flips some of the bits in a chromosome. For example, the string 00000100 might be mutated in its second position to yield 01000100. Mutation can occur at each bit position in a string with some probability, usually very small (e.g., 0.001).[4]

## 4 Major Elements of the Genetic Algorithm

The simple genetic algorithm (SGA) is described by Goldberg [5] and is used here to illustrate the basic components of the GA. A pseudo-code outline of the SGA is shown in Figure. 1. The population at time t is represented by the time-dependent variable P, with the initial population of random estimates being P(0). Using this outline of a GA, figure.1 describes the major elements of the GA.[6,7].Procedure GA

```
Begin
T=0;
Initialize P(t);
Evaluate P(t);
while not finished do
        begin
                t=t+1;
                select P(t) from P(t-1);
                reproduce pairs in P(t);
                evaluate p(t);
         end;
end.
```

Fig.1.a simple genetic algorithm

## 5. Flow Chart of Genetic Programming.

Figure.2 illustrate genetic programing flow chart.[2,8].

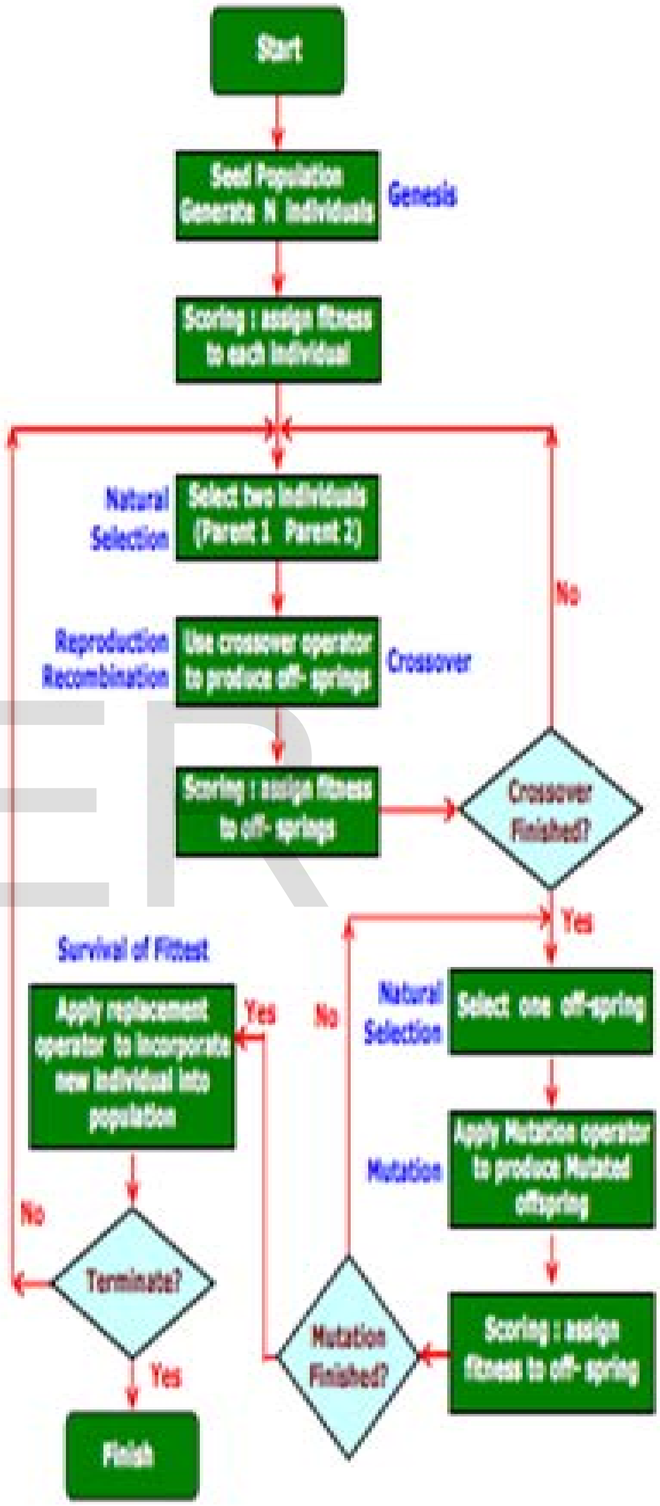
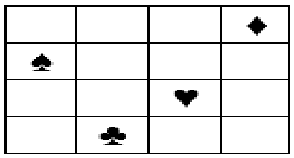


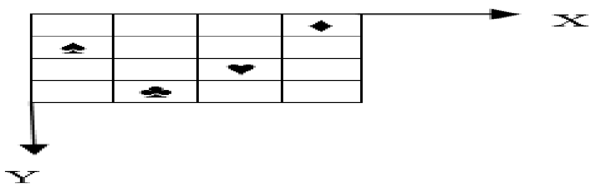

Fig.2.genetic algorithm program flow chart

## 6.The Four Shapes Problem

there are four shapes, we want to find a way to place them on a board divided into 16 grid units so that no two shapes attach each others. In this work we use GAs to determine the best location of four shapes.



### 6.1 Problem Representation (how to define a facility location)




♦: has (X=4 ,Y=1), ♠: has (X=1 ,Y=2),
♥: has(X=3 ,Y=3), ♣: has (X=2 , Y=4)

### 6.2. Chromosome Structure

The variables in the problem are the locations of four shapes. Then, the chromosome structure are as follows. Note that the Genes of a chromosome are the problem variables.

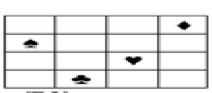
| X-♦ | Y-♦ | X-♠ | Y-♠ | X-♥ | Y-♥ | X-♣ | Y-♣ |
|---|---|---|---|---|---|---|---|
| 4 | 1 | 1 | 2 | 3 | 3 | 2 | 4 |

8 genes (values from 1 to 4)

### 6.3. Generate Population (50 to 100 is reasonable diversity & processing time)

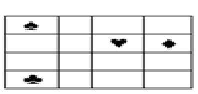
(P1)

| X-♦ | Y-♦ | X-♠ | Y-♠ | X-♥ | Y-♥ | X-♣ | Y-♣ |
|---|---|---|---|---|---|---|---|
| 4 | 1 | 1 | 2 | 3 | 3 | 2 | 4 |



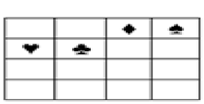
(P2)

| X-♦ | Y-♦ | X-♠ | Y-♠ | X-♥ | Y-♥ | X-♣ | Y-♣ |
|---|---|---|---|---|---|---|---|
| 4 | 2 | 1 | 1 | 3 | 2 | 1 | 4 |



(P3)

| X-♦ | Y-♦ | X-♠ | Y-♠ | X-♥ | Y-♥ | X-♣ | Y-♣ |
|---|---|---|---|---|---|---|---|
| 3 | 1 | 4 | 1 | 1 | 2 | 2 | 2 |



### 6. 4 Evaluate the Population

Objective function = Minimize site score = Minimum of Σ d . W

Score = d♦♠ . W♦♠ + d♦♥. W♦♥ + d♦♣ . W♦♣ + d♠♥ . W♠♥ + d♠♣ . W♠♣+ d♥♣. W♥♣ (1)

Let's consider the closeness weights (W) as follows:
**W two shapes= 10** (positive means two shapes are far from each other)
**W two shapes = -10** (negative means two shapes close to each other)
Let's also consider the distance (d) between two shapes as the number of horizontal and vertical blocks between them.
**P1 Score** =4x10+3x10+5x10+3x10+3x10+2x10=200
**P2 Score** = 4x10+1x-10+5x10+3x10+3x10+4x10=180
**P3 Score**= 1x-10+3x10+2x10+4x10+3x10+1x-10=100

### 6. 5 Calculate the Merits of Population Members

**Merit of P1** = (200+180+100) / 200 = 2.4=2
**Merit of P2** = (200+180+100) / 180 = 2.6=3
**Merit of P3** = (200+180+100) / 100 = 4.8=5
the sum of merits = 10
the smaller score gives higher merit because we are interested in minimization. In case of maximization, we use the inverse of the merit calculation.



### 6.6 Calculate the Relative Merits of Population Members

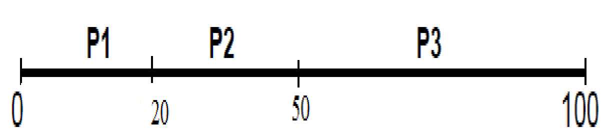
**RM of P1** = merit * 100 / Sum of merits
= 2 . 100 / 10 = 20
**RM of P2** = merit * 100 / Sum of merits
= 3 . 100 / 10 = 30
**RM of P3** = merit * 100 / Sum of merits
= 5 . 100 / 10 = 50

### 6.7 Randomly Select Operator (Crossover or Mutation)

Crossover rate = 96% (marriage is the main avenue for evolution)
Mutation rate = 4% (genius people are very rare)
To select which operator to use in current cycle, we generate a random number (from 0 to 100). If the value is between 0 to 96, then crossover, otherwise, mutation

### 6. 8 Use the Selected Operator (Assume Crossover)

Randomly select two parents according to their relative merits of Step 6. 6




Relative merits on a cumulative horizontal scale.

For first

parent, we generate a random number (0 to 100). According to its value, we pick the parent. For example, assume value is 10, then P1 is selected.
For the 2nd parent,get a random number (0 to 100) assume 30,then P2 is picked.
Lets apply crossover to generate an offsprings.

(P1)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4 | 1 | 1 | 2 | 3 | 3 | 2 | 4 |

(P2)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4 | 2 | 1 | 1 | 3 | 2 | 1 | 4 |

Offspring 1

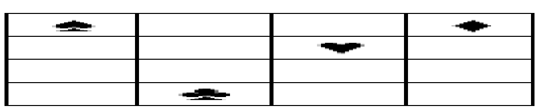
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4 | 1 | 1 | 1 | 3 | 2 | 2 | 4 |

Offspring2

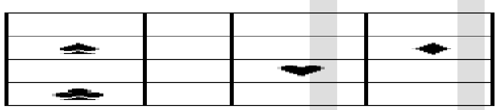
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4 | 2 | 1 | 2 | 3 | 3 | 1 | 4 |

### 6.9. Evaluate offspring

Offspring1



Offspring2



Offspring1 Score = 3x10+2x10+5x10+3x10+4x10+3x10=200
Offspring2 Score = 3x10+2x10+5x10+3x10+2x10+3x10=180

### 6.10 Compare the Offspring with the Population (Evolve the Population)

Since the offspring1 score = 200 is better than the worst population member (P3 has a score of 100), then the offspring1 survives and P3 dies (will be replaced by the offspring1).

(P3) become

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4 | 1 | 1 | 1 | 3 | 2 | 2 | 4 |

### 6.11 Experimental Result

Executing the simple genetic algorithm code (SGA) written in Pascal programming language ,the primary data structure is the population of 22 string and the following parameters :

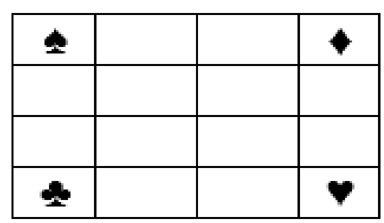
**SGA Parameters**

Population size=22
Chromosome length= 4
Maximum of generation=10
Crossover probability=0.6
Mutation probability=0.03

**Population**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1. | 1 | 1 | 4 | 1 | 1 | 4 | 4 | 4 |
| 2. | 2 | 1 | 4 | 1 | 1 | 4 | 4 | 4 |
| 3. | 3 | 1 | 4 | 1 | 1 | 4 | 4 | 4 |
| 4. | 1 | 2 | 4 | 1 | 1 | 4 | 4 | 4 |
| 5. | 2 | 2 | 4 | 1 | 1 | 4 | 4 | 4 |
| 6. | 3 | 2 | 4 | 1 | 1 | 4 | 4 | 4 |
| 7. | 4 | 2 | 4 | 1 | 1 | 4 | 4 | 4 |
| 8. | 1 | 3 | 4 | 1 | 1 | 4 | 4 | 4 |
| 9. | 2 | 3 | 4 | 1 | 1 | 4 | 4 | 4 |
| 10. | 3 | 3 | 4 | 1 | 1 | 4 | 4 | 4 |
| 11. | 4 | 3 | 4 | 1 | 1 | 4 | 4 | 4 |
| 12. | 2 | 4 | 4 | 1 | 1 | 4 | 4 | 4 |
| 13. | 3 | 4 | 4 | 1 | 1 | 4 | 4 | 4 |
| 14. | 2 | 2 | 4 | 2 | 2 | 4 | 4 | 4 |
| 15. | 2 | 2 | 1 | 1 | 2 | 4 | 4 | 4 |
| 16. | 2 | 2 | 2 | 1 | 2 | 4 | 4 | 4 |
| 17. | 2 | 2 | 3 | 1 | 2 | 4 | 4 | 4 |
| 18. | 2 | 2 | 4 | 1 | 2 | 4 | 4 | 4 |
| 19. | 2 | 2 | 1 | 2 | 2 | 4 | 4 | 4 |
| 20. | 2 | 2 | 3 | 2 | 2 | 4 | 4 | 4 |
| 21. | 2 | 2 | 1 | 3 | 2 | 4 | 4 | 4 |
| 22. | 2 | 2 | 2 | 3 | 2 | 4 | 4 | 4 |

At the end of the run repeating the process thousands of times until the best solution is determined. One of the top solutions is as follows:



## 7.CONCLUTIONS

- This paper showed that Shape problem can be successfully solved using optimization function. Results shows that the genetic algorithm have the ability to find optimal solution for solving shapes problem.

- because of the stochastically behavior of the GA it is also concluded that there are no guarantees that GA will reach the optimal solution in every run. However it is shown that all results obtained with GA are close to optimal in single-objective problem.

- The GA does find near optimal results quickly after searching a small portion of the search space.

- GA represent an intelligent exploitation of a random search used to solve optimization problem.

- The GA is well suited to and has been extensively applied to solve complex design optimization problems because it can handle both discrete and continuous variables, and nonlinear objective and constrain functions without requiring gradient information.

## 8.REFRENCES

1. Ms.Dharmistha D.Vishwakarma" Genetic Algorithm based Weights Optimization ofArtificial Neural Network", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 1, Issue 3, August 2012.

2. R C chakraborty "Fandemantaels of genetic algorithms" :AI course lecture 39-40,notes ,slides,2010.

3. Olesya Peshko,"Global OptimizationGenetic Algorithms",2007

4. Mitchell Melanie," An Introduction to Genetic Algorithms ",A Bradford Book The MIT Press Cambridge, Massachusetts • London, England Fifth printing, 1999.

5. Goldberg, David E. "Genetic Algorithm in Search, Optimization, and Machine Learning", Addison Wesley Longmont, International Student Edition 1989.

6. Andrew Chipperfield, Peter Fleming, Hartmut Pohlheim, Carlos Fonseca," Genetic Algorithm TOOLBOX For Use with MATLAB",department of automatic control and system engineering,2001.

7. Saif Hasan ,Sagar Chordia ,Rahul Varshneya,"genetic algorithm",February 6, 2012.

8. ?,"genetic algorithms",2007.