UNIT 13 THE SCHEMA THEORAM

Structure **Page No**

13.1 INTRODUCTION

Current research on evolutionary algorithms aims at developing sound theories that describe and predict the behavior of genetic algorithms. Traditional genetic algorithm theory is based on the schema theorem and the building block hypothesis. The schema theorem describes the behavior of a genetic algorithm by the way a genetic algorithm processes schemata. A schema S is a generalized form of a bit string consisting of 0s, 1s, and don't cares '*'. In a particular schema, e.g., "*011***", each '*' can be substituted by a 0 or 1. The schema is introduced in Sec. 13.2. The building block hypothesis states that a genetic algorithm combines those schemata that have above average fitness as well as a short defining length, i.e., the maximal distance between 0's and 1's, which would be 3 in the above example. It is believed that by combination building blocks, a genetic algorithm constructs a solution as a combination of sub solutions (low order schemata).

In Section 13.3, schema theorem is discussed. For the case of relative fitness proportional selection and one point crossover alone, the schema theorem states that the expected number is applied to find the expected number of instances of schema S. The Convergence of elitism model of GA is discussed in Sec. 13.4.

Objectives

After studying this unit you will be able to:

- define a schema;
- find the length and order of a schema;
- apply single point crossover for a schema;
- apply notation;
- describe the schema theorem;
- describe the convergence of GA.

13.2 INTRODUCTION TO SCHEMA

We have seen that Genetic Algorithms are one of the most successful engineering tools ever developed. They are easy to implement. A simple GA and GAs are surprisingly robust against lack of skill. GAs tend to be used ***when nothing else works***. But this does raise interesting questions:

– Why do genetic algorithms work?

– When do they work?

We need to reason rigorously about the operation of GAs. We will look at the most important theoretical work in GAs: the ***Schema Theorem.*** Reasoning about complicated GAs is very hard. Fundamental analysis is largely based on examination of simple Gas i.e. Bit string representation, Proportional selection and single-point cross-over.

First we try to understand what a schema is?

A schema is a pattern comprising 0s, 1s, and *s could be either 0 or 1. Schemas capture particular subsets of strings. * stands for "I don't care" or, if you like, "a 0 or a 1". Some examples of schemas are

1. S_1 = 000000 (every bit is defined, only one string)
2. S_2 = 111111 (every bit is defined, only one string)
3. S_3 = 101010 (every bit is defined, only one string)
4. S_4 = 0****0 (only beginning and final bits are defined, the rest middle 4 positions may be filled by 16 $z^4 =$possible strings)

We talk about the bit string *x* being *an instance of a schema* S if x is in the set defined by s. Thus 010101 is an instance of 010101. 010101 is an instance of 01****. 010101 is an instance of 01**01. 010101 is an instance of ******. 010101 is **not** an instance of 11****. A string x can be an instance of many schemata.

Of how many schemas is a particular bit string an instance? 0 is an instance of S = 0 and also an instance of s = *. Similarly 1 is an instance of s = 1 and also an instance of s = *. So for each bit value in a string there are two possible element values for a containing schema. Therefore the number of schemas containing bit string x is: 2^L where L is the length of the string. Schema can be defined population of binary strings created at random. Also a schema is a subset of the space of all possible individuals for which all the genes match the template for schema S.

Example 1: Let us consider the alphabet of gene alleles and denote it by A. Here AU* is the schema alphabet, where * is the 'wild card' symbol, which matches any allele value. For instance the binary alphabet $A \in \{0,1,*\}$ where $* \in \{0,1\}$.

Length and order of a Schema

The ***defining length d(S)*** of a schema ***S*** is the distance between the first defined bit and the last defined bit i.e. it is the distance between first and last non * gene is schema S.

Example 2: d(101010) = 5;
d(**10**) = 1;
d(**1*1*) = 2.

Thus length of a schema is position (last defined) – position (first defined).

The ***order o(s)*** of a schema ***s*** is the number of *defined* bits it contains, i.e. order is the number of non * genes in schema S.

Example 3: o(101010) = 6;
o(**10**) = 2;
o(**1*1*) = 2; o(**1*1*) = 2.

Here it can be noted that for cardinality, k, there are $(k+1)\ell$ schema in string of length l.

Now try the following exercises.

E1) Write the schema for the gene sequence $\{0111000\}$ and $\{1110011\}$.

E2) Write at least 4 chromosome set, which are identified by schema $S = (01*1*)$.

E3) Find the length and order of the following schema.

i) $S_1 = (0**11*0**)$

ii) $S_2 = (*11*0**)$

iii) $S_3 = (***0***)$

After discussing schema, we are in a position to incrementally introduce the effect of the various selection and search operators associated with the above definition.

Selection of operators-fitness proportional selection

In this let us first find the probability of an individual x_k, samples schema S i.e. $p(x_k)$. For this we assume that

i) The probability of x_k is proportional to the number of instances of schema S in the population, and

ii) The probability of x_k is proportional to the average fitness of schema S relative to the average fitness of all individuals in the population.

Let us consider the number of instances of S at time t, which is denoted by N(S,t) and the expected number of instances of S at time t is denoted by E(N(S,t)). The observed average fitness of S at time t (i.e. the average fitness of instances of S in the population) is denoted by u(S,t). Then the probability of selecting x_k under proportional selection is

$$p_k = F(x_k) \Big/ \sum_{j=1}^{N} F(x_j)$$

where $f(x_k)$ is the fitness of string s_k.

Now, the expected number of instances of schema S at time t is given by

$$E(N(S,t)) = N \times p_k = N \times \frac{F(x_k)}{\sum_{j=1}^{N} F(x_j)}$$

$$= F(x_k) \times \frac{N}{\sum_{j=1}^{N} F(x_j)}$$

$$= \frac{F(x_k)}{F(t)}$$

where $\overline{F}(t)$ is the fitness of all strings in population at time t. This E(N (s, t)) implies that the schemas with fitness greater (lower) than the average population fitness are likely to account for proportionally more (less) of the population at the next generation. Also it can be concluded that for the accurate estimates of expectation and probability, the population size should be infinite.

Crossover

The fitness of individuals in a population cannot be improved by reproduction, therefore let us apply single point crossover to modify the distribution of schema in the population.

Now consider crossover. Let p_c be the probability that single point cross over will be applied to a string. Instance of schema S is picked as a parent. Schema S ***survives*** if one of the offspring is also an instance of schema S.

Lower bound on probability $S_c(s)$ of survival of schema S is given by
$S_c(s)$ = pr(S Survive)
= pr(Survive|no cross) pr(no cross)+ pr(Survive|cross)pr(cross)
= 1x(1-p_c) + pr(Survive|cross)x(p_c)

We need to determine the pr(Survive|cross)
Let L = length of genome = 7
L-1 = number of possible cross over points = 6
And consider S = (* 0 * * 0 * *)
D(s) = defining length = 3

Any of the (3) points between the first and last ***defined*** bits is ***potentially*** disruptive. So the probability of randomly choosing a ***potentially*** destructive crossing point is d(s)/(L – 1) = 3/6 = 1/2. The probability of ***actually*** causing disruption must be ***less than*** this. And so the probability of ***surviving*** crossover ***given that you do crossover*** must be ***greater than***

$$\left(1-\frac{d(s)}{L-1}\right)$$

And so the pr(Survive|cross)pr(cross) must be greater than or equal to

$$\left(1-\frac{d(s)}{L-1}\right)p_c = p_c\left(1-\frac{d(s)}{L-1}\right)$$

And so pr(Survive) = $S_c(s)$ must satisfy

$$S_c(s) \geq (1-p_c)+p_c\left(1-\frac{d(s)}{L-1}\right)$$
$$S_c(s) \geq \left(1-p_c\left(\frac{d(s)}{L-1}\right)\right)$$

Mutation

In the special worst case lower bound $P_c = 1$

Now we consider the disruptive effects of ***mutation***. Mutation is applied gene by gene. Let p_m be the probability of flipping any bit in a string. A schema s is disrupted when any ***defined*** bit is flipped. There are o(s) bits defined, and so the probability they all non * genes given by survive is

$$S_m(s) = (1-p_m)^{o(s)}$$

The probability of applying the mutation operator is very very less, therefore $(1-\text{pm})^{\circ(S)} \approx 1-\circ(S)\,\text{pm}.$

Note: If a bit is in a position corresponding to a * in the schema template it does matter whether or not it is flipped.
Now let us discuss schema theorem, which is also known as fundamental theorem of Genetic Algorithm.

13.3 THE SCHEMA THEOREM

The completed schema theorem can now be stated.

The Schema Theorem

The expected number of schema S at generation t + 1 with proportional selection, single point crossover and genewise mutation is given by

$$E(N(s,t+1)) \geq \frac{\hat{u}(s,t)}{\overline{F}(t)} N(s,t)\left[1-p_c\left(\frac{d(s)}{L-1}\right)\right](1-p_m)^{o(s)}$$

Conservatism

Note that the schema theorem is conservative.

1. Only deals with ***destructive*** effects.
2. Evolution may also be ***constructive*** effects.
3. Non-s schema instances may combine to give s-schema instances.
4. Similar properties are valid for mutation.

Implications of Schema theorem

Schemas, like families, need nourishment and encouragement and careful protective management

- The more bits in your building block family the more likely one is to go off the rails (causing great frustration and heartache).
- Genes living far apart are prone to breaking up
- Conducting a constructive relationship at a distance is hard.
- Best results achieved by the family unit huddled together in consecutive positions.

Applications of Schema Theorem

The schema theorem is more applicable at the early stages of a search rather than at the end. Schema theorem indicates that fitter than average schemas are rewarded. The fitter the schema the more it is rewarded.

- Reward is immediate: you see it in the next generation.
- Should alert us to one danger immediately.
- Premature converge of the population.

Now try the following exercise.

E4) What implications does the schema theorem have for your choice of mutation rates?

So far we have discussed the schema theorem. In the following section, we shall discuss the convergence of the elitism model of genetic algorithms.

13.4 CONVERGENCE OF GA

Goldberg (1989) explained the convergence criteria of a GA from the viewpoint of schema. As we have discussed that a schema represents a subset of strings with similarities at certain positions. In general, any specific string is simultaneously an instance of 2^n schemata, where n is the length of the string. We can always associate a fitness value with a schema, i.e. the average fitness of the schema because schema is a representation of a robust of strings. Hence, a schema's average fitness value is different for each population from one generation to another.

GAs research can be visualized for the optimal strings as a simultaneous competition among schemata increases the number of their instances in the population.

The winners of individual schema competitions may form the optimal string, such schemata are called **building blocks.** A number of such building blocks in various parts along the string get emphasized by applying the genetic operators on a population of string. But there is every danger that the crossover operator may destroy good schemata. So, selection of good appropriate crossover operator plays a vital role here. If the cross site cuts the well defined position in the schema, this may not be preserved in the next generation unless otherwise. Second parent also will have the same schema in that position. In case of single-point crossover, falling of cross-sites within the defined positions has less probability while in the case of uniform crossover, disturbance of good schema takes place with a higher probability. As far as GA to engineering field is concerned single- and two-point crossover are common. The meaning of search in the genetic space is the development of building blocks. Building blocks are combined together due to combined action of genetic operators to form bigger and better building blocks and finally converge to the optimal solution.

If we see the processing of individual schema, it is found that the worst schema is not able to survive. So it depends totally on the selection of the genetic operators. For example if a cross site cuts a particular well defined position in a schema, it may be a reason for not preserving it in the next generation, unless the second parent has same schema at that position. In genetic spaces the search is only for the development of building blocks. To form bigger and better building blocks which finally converge to the optimal solution, the building blocks are combined together due to combined action of genetic operators. The corresponding GA cycle is shown in Fig 1.

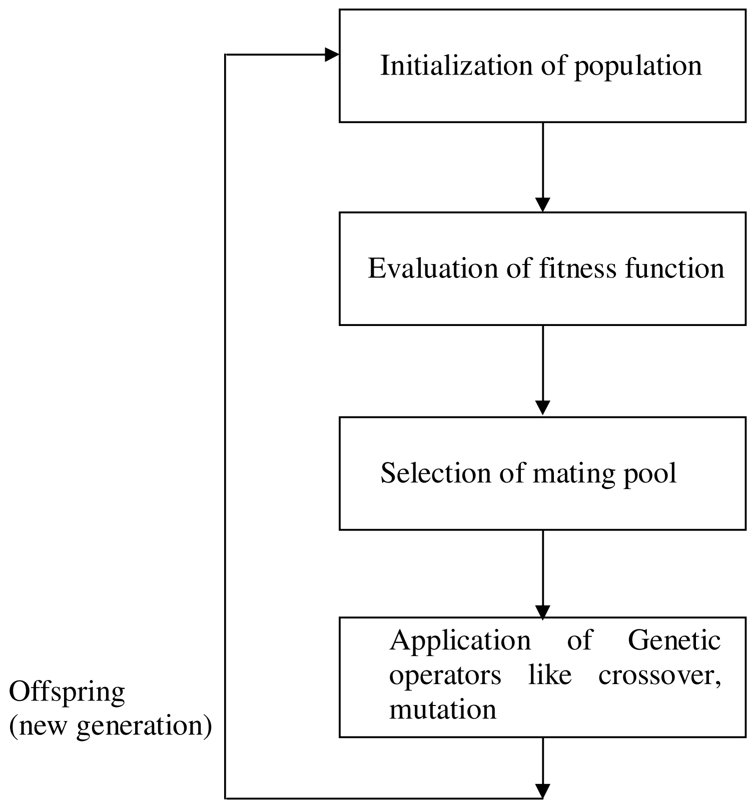
Initialization of population
Evaluation of fitness function
Selection of mating pool
Application of Genetic operators like crossover, mutation
Offspring (new generation)

Fig. 1: The GA cycle

The situation of good strings in a population set and random information exchange among good strings are simple and straightforward. No mathematical proof has been derived yet for convergence of GA. Rajeev and Krishnamoorthy (1992), discussed one criterion for convergence. When a fixed percentage of columns and rows in population matrix becomes the same, it can be assumed that convergence is attained. The fixed percentage may be 80% or 85%.

In genetic algorithms, as we proceed with more generations, there may not be much improvement in the population fitness. As the generation progresses, the population gets filled with more fit individuals with only slight deviation from the fitness of best individuals so far found, and the average fitness comes very close to the fitness of the best individuals. We can specify some fixed number of generations after getting the optimum point to confirm that there is no change in the optimum in the subsequent generations.

Now let us summarise the unit.

13.5 SUMMARY

1. A schema is a subset of the space of all possible individuals for which all the genes match the template for schema.
2. Schema order, o(S), is the number of non '*' genes in schema S.
3. Schema Defining Length, d(S), is the distance between first and last non '*' gene in schema S.

4. Schemas with fitness greater (lower) than the average population fitness are likely to account for proportionally more (less) of the population at the next generation.
5. Reproduction does nothing to improve the fitness of individuals in a population. (Single point Crossover was the first of two search operators introduced to modify the distribution of schema in the population).
6. Mutation is applied gene by gene. In order for schema S to survive, all non * genes in the schema much remain unchanged.
7. The expected number of schema S at generation t + 1 when using a canonical GA with proportional selection, single point crossover and gene wise mutation (where the latter are applied at rates p_c and p_m) is defined in schema theorem.

13.6 SOLUTIONS/ANSWERS

E1) $S = (*11*0**)$

E2) The chromosome sets are (01010), (01011), (01110) and (01111).

E3) i) $d(S_1) = 7 - 1 = 6$

$0(S_1) = 4$

ii) $d(S_2) = 5 - 2 = 3$

$0(S_2) = 3$

iii) $d(S_3) = 4 - 4 = 0$

$0(S_3) = 1$

E4) Mutation is used to promote intensification (where local optimisation is required) and of course, to promote diversity.

13.7 REFERENCES

1. S. Rajasekaran and G.A. Vijayalakshmi Pai, (2010), *Neural Netwoks: Fuzzy Logic, and Genetic Algorithms*.
2. D.K. Pratihar, (2008), *Soft Computing*.
3. J. –S. R. Jang, C. –T. Sun and E. Mizutani, (2004), *Neuro-Fuzzy and Soft Computing*.
4. David Goldberg, (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*.