Artificial Intelligence Genetic Algorithm

Zahoor Tanoli (PhD)
CUI Attock

Local Search Techniques

Introduction

Local search in the state space, evaluating and modifying one or more current states rather than systematically exploring paths from an initial state

Suitable for problems in which all that matters is the solution state, not the path cost to reach it (e.g. 8-queens problem)

In addition to finding goals, local search algorithms are useful for solving pure *optimization problems*, in which the aim is to find the best state according to an *objective function*.

Introduction

A biologically inspired model of intelligence

The principles of *biological evolution* are applied to find solutions to difficult problems

The problems are not solved by reasoning logically about them; rather populations of competing candidate solutions are spawned and then evolved to become better solutions through a process patterned after *biological evolution*

Introduction

Less worthy candidate solutions tend to die out, while those that show promise of solving a problem survive and reproduce by constructing new solutions out of their components

GA begin with a population of candidate problem solutions

Candidate solutions are evaluated according to their ability to solve problem instances: only the fittest survive and combine with each other to produce the next generation of possible solutions

Introduction

Thus increasingly powerful solutions emerge in a Darwinian universe. Learning is viewed as a competition among a population of evolving candidate problem solutions

This method is heuristic in nature and it was introduced by John Holland in 1975

Basic Algorithm

```
begin
    set time t = 0;
    initialise population P(t) = {x_1^t, x_2^t, …, x_n^t} of solutions;
    while the termination condition is not met do
        begin
            evaluate fitness of each member of P(t);
            select some members of P(t) for creating offspring;
            produce offspring by genetic operators;
            replace some members with the new offspring;
            set time t = t + 1;
        end
end
```

GENETIC ALGORITHM

- ***Main Decisions***:
  - Representation/ Encoding of Individuals/ Chromosome
  - Fitness/ Evaluation Function
  - Stochastic Operators for reproduction [Evolution]
    - Crossover
    - Mutation
  - Selection

Chromosome

Representation of Solutions: The Chromosome

Gene: A basic unit, which represents one characteristic of the individual. The value of each gene is called an *allele*

Chromosome: A string of genes; it represents an individual i.e. a possible solution of a problem. Each chromosome represents a point in the search space

Population: A collection of chromosomes

An appropriate chromosome representation is important for the efficiency and complexity of the GA

Representation of Solutions: The Chromosome

The classical representation scheme for chromosomes is binary vectors of fixed length. However, real valued chromosomes are also possible

In binary coded chromosomes, every gene has two alleles
In real coded chromosomes, a gene can be assigned any value from a domain of values

In the case of an I-dimensional search space, each chromosome consists of I variables with each variable encoded as a bit string

Representation of Solutions: The Chromosome

Example: Cookies Problem

Two parameters sugar and flour (in kgs). The range for both is 0 to 9 kgs. Therefore a chromosome will comprise of two genes called sugar and flour

5	1	Chromosome # 01
2	4	Chromosome # 02

Representation of Solutions: The Chromosome

Example: Expression satisfaction Problem

$$F = (\neg a \vee c) \wedge (\neg a \vee c \vee \neg e) \wedge (\neg b \vee c \vee d \vee \neg e) \wedge (a \vee \neg b \vee c) \wedge (\neg e \vee f)$$

Chromosome: Six binary genes a b c d e f e.g. 100111

Fitness Function

Fitness Function

Also called "evaluation function"

It is used to determine the fitness of a chromosome

Creating a good fitness function is one of the challenging tasks of using GA

GENETIC ALGORITHM

Fitness Function

Example: Cookies Problem

Two parameters sugar and flour (in kgs). The range for both is 0 to 9 kgs. Therefore a chromosome will comprise of two genes called sugar and flour

5	1

2	4

The fitness function for a chromosome is the taste of the resulting cookies; values can be from 1 to 100

Fitness Function

Example: Expression satisfaction Problem

$$F = (\neg a \vee c) \wedge (\neg a \vee c \vee \neg e) \wedge (\neg b \vee c \vee d \vee \neg e) \wedge (a \vee \neg b \vee c) \wedge (\neg e \vee f)$$

Chromosome: Six binary genes a b c d e f e.g. 100111

Fitness function: Number of clauses having truth value of 1
e.g. 010010 has fitness 2

Reproduction Operators

Reproduction Operators

Genetic operators are applied to chromosomes that are selected to be parents, to create offspring

Basically of two types: Crossover and Mutation

Crossover operator create offspring by recombining the chromosomes of selected parents

Mutation is used to make small random changes to a chromosome in an effort to add diversity to the population

Reproduction Operators: Crossover

Crossover operation takes two candidate solutions and divides them, swapping components to produce two new candidates

Reproduction Operators: Crossover

Figure illustrates crossover on bit string patterns of length 8

The operator splits them and forms two children whose initial segment comes from one parent and whose tail comes from the other

Input Bit Strings

1 1 # 0 | 1 0 1 #   # 1 1 0 | # 0 # 1

Resulting Strings

1 1 # 0 # 0 # 1   # 1 1 0 1 0 1 #

Reproduction Operators: Crossover

Two genes sugar and flour (in kgs)
Crossover operation on chromosomes

5	1	→	5	4
2	4		2	1

Reproduction Operators: Crossover

The place of split in the candidate solution is an arbitrary choice. This split may be at any point in the solution

This splitting point may be randomly chosen or changed systematically during the solution process

Crossover can unite an individual that is doing well in one dimension with another individual that is doing well in the other dimension

Reproduction Operators: Crossover (Important Similarities)

Suppose we have these chromosomes and their fitness values

A)	01101	169
B)	11000	576
C)	01000	64
D)	10011	361

If we analyze the strings which have high fitness value, we have observe that they both have a 1 in the 1st column, whereas the low fitness strings have a 0. In the second column we have a 1 in B and 0 in D, and 1 in the low fitness strings. We wonder that if 1 in the 2nd column is associated with low fitness, so we may improve the score of B by putting a 0 in place of 1 in the 2nd column