Genetic Algorithms

by

Dr. Sadiq M. Sait & Dr. Habib Youssef

(special lecture for oometer group)
November 2003

Contents

Introduction

- Genetic Algorithm(s)
- Inspired by Darwinian theory, a powerful search technique
- Based on the ñTheory of Natural Selectionò
- It is an adaptive leaning heuristic
- Belongs to class of iterative non-deterministic algorithm
- GA operates on population of individuals encoded as strings
- Used to solve combinatorial optimization problems

GA Basics

- Using GAs to solve a given combinatorial optimization problem one has to come up with
  - A suitable encoding of solutions to the problem as chromosomes (generally strings, though not necessarily)
  - Translate cost function into a fitness measure
- A solution to the optimization problem and the element of the population is represented by chromosome
- One has to find an efficient representation of the solution in the form of a chromosome
- Each chromosome (individual) has a fitness value

Robust, Effective, …

- GAs are both effective and robust, independent of the choice of the initial configurations they can produce high quality solutions
- They are able to exploit favorable characteristics of previous solution attempts to construct better solutions (inheritance)
- GAs are computationally simple and easy to implement
- Their power lies in the fact that as members of the population mate and produce offsprings, they (offsprings) have a significant chance of inheriting the best characteristics of both parents

Characteristics of GA

- Work with coding of parameters
- Search from a set of points
- Only require objective function values
- Non-deterministic:
  - Non-determinism is introduced in operations (on chromosomes) and in several processes of the algorithm
- GAs are blind

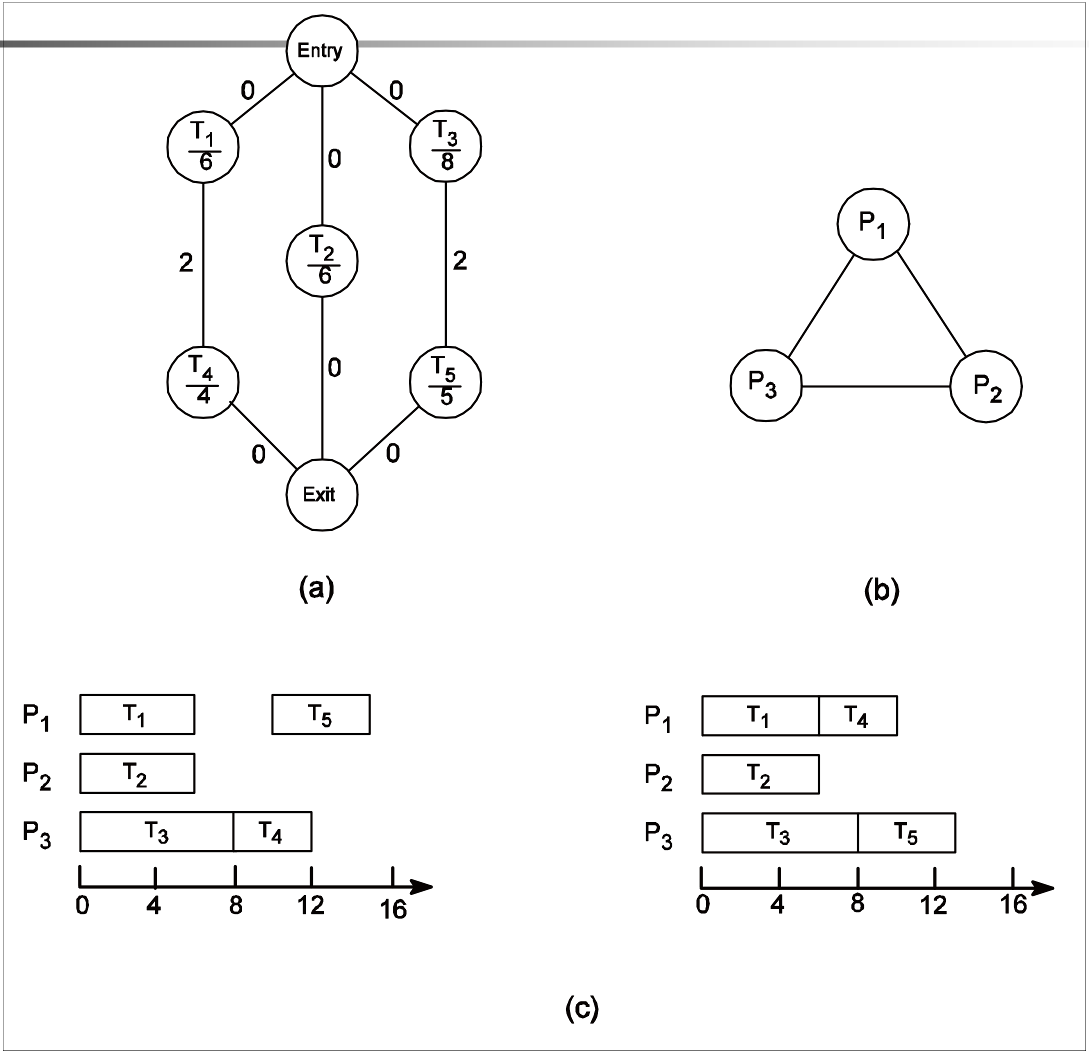
GA Terminology

- How the organism is constructed or the solution represented is called as chromosome (basically an encoding)
- Complete set of chromosomes is called a genotype and resulting organism is called as phenotype
- Genes: Symbols that make up a chromosome
- Alleles: Different values taken by a gene
- Fitness: It is always a positive number, it is a measure of goodness (for optimization problems it is a function of the cost of the solution)
- Initial Population:
  - An initial population constructor is required to generate a certain predefined number of solutions
  - Quality of the final solution produced by genetic algorithm depends upon the size of the population and how the initial population is constructed
  - Initial population may comprise random solutions (sometimes seeding is used)

Examples of Chromosomes

- Linear assignment problem (a permutation problem)
  - 13482765 (the index is the position and the number is the node/block for example)
- Bi-partitioning problem
  - An example of a possible chromosome is 01001101
- Task assignment problem
  - Say we have 8 tasks (numbered) and 3 processors 33123122
- There can be several different chromosomes for the same problem. A chromosome does not have to be a linear string (2-D chromosomes have been proposed)

Task Graph


Entry
0
0
0
T_1 6
T_2 6
T_3 8
2
2
T_4 4
T_5 5
0
0
0
Exit
(a)
P_1
P_2
P_3
(b)
P_1 T_1 T_5
P_2 T_2
P_3 T_3 T_4
0 4 8 12 16
P_1 T_1 T_4
P_2 T_2
P_3 T_3 T_5
0 4 8 12 16
(c)

Parents, Genetic Operators

- Chromosomes or pairs of chromosomes produces new solutions called as offsprings
- Genetic operators:
  - Crossover
  - Mutation
  - Inversion
- Crossover operator is applied to pairs of chromosomes
- Two individuals selected for crossover are called parents
- Mutation is a genetic operator that is applied to a single chromosome (maybe to a gene or pairs of genes)
- Resulting individuals produced when genetic operators are applied on the parents is called as offsprings

Choice Of Parents

- Choice of parents is probabilistic
- Higher fitness individuals are more likely to mate than the weaker ones
- Select parents with a probability that is directly proportional to fitness values
- Larger the fitness of chromosome greater is its chance of being selected for crossover
- The Roulette-wheel method is generally employed. It is a wheel/disk in which each member of the population is given a sector whose size is proportional to its fitness
- Selection for crossover: wheel is spun and whichever individual comes up gets selected as the parent

Example: **Roulette-wheel method**

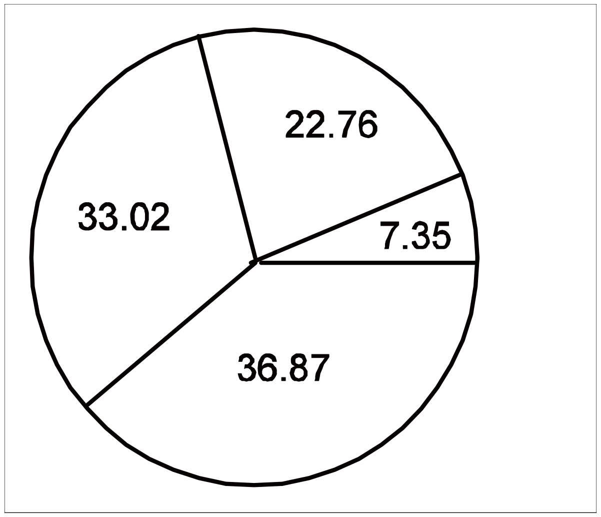
- Fitness values and their percentages :

  s1 = [0 1 1 0 0 1] = 625 = 7.35%

  s2 = [1 0 1 1 0 0] = 1936 = 22.76%

  s3 = [1 1 0 1 0 1] = 2809 = 33.02%

  s4 = [1 1 1 0 0 0] = 3136 = 36.87%



Crossover (χ)

- Provides a mechanism for the offspring to inherit the characteristics of both the parents
- It operates on two parents (P1 and P2) to generate offspring(s)
- Simple crossover:
  - Performs the ñcut-catenateòoperation
  - A random cut point is chosen to divide the chromosome into two
  - The offspring is generated by catenating the segment of one parent to the left of the cut point with the segment of the second parent to the right of the cut point

Example of Crossover

- Example: For the following 2 parent chromosomes s2 = [1 0 | 1 1 0 0] and s4 = [1 1 | 1 0 0 0]
  - If the crossover point is chosen after the 2nd gene, as shown above, the offspring will contain genes from the left of crossover point of parent P1 and genes from the right of cut point of parent P2
  - Offspring chromosome is [1 1 1 1 0 0]
  - What about the fitness of the above chromosome, and does it always represent a valid solution?

Permutation & other Crossovers

- Partially Mapped Crossover (PMX)
  - dbcae | fghi
  - hfbed | icga
  - Result   dbcaeighf and hfbedigca
- Order Crossover (OX)
  - Result for the above chromosomes and cut points are dbcaehfig and hfbedcagi
- Cyclic Crossover (A cycle contains a common subset of alleles in the two parents that occupy a common subset of positions)
  - dbcae | fghi
  - hfbec | idga (three genes d,h,g have the same set of positions in both the parents and so form a cycle, similarly, e,f,c,b,i,a form another cycle. There can be more than two cycles)
  - Result dxxxxxghx + xfbecixxa = dfbcigha
- Two point PMX and 2-point simple crossovers
- And othersé

Multi-point crossovers & other variations

- Generate two offsprings by treating the chromosome for P2 and P1 and vice versa
- Example: The two parent chromosomes P 1 = [1 | 0 1 1 | 0 1] and P 2 = [1 | 1 1 0 | 1 0], if the two cut points are chosen after the first and fourth positions, then the offsprings generated for the two parents are
  - O1 = [1 | 1 1 0 | 0 1] and
  - O2 = [1 | 0 1 1 | 1 0]
- With the two-point crossover the chances of offsprings inheriting the goodness of the schemata are higher

Mutation, Generation and Selection

- Produces incremental random changes (with very low probability) in the offsprings by changing allele values of some genes
- Mutation perturbs a chromosome in order to introduce new characteristics not present in any element of the population
- Example: Swap two alleles, toggle one or two (in case of binary chromosomes), etc
- A generation is an iteration of GA where individuals in the current population are selected for crossover and offsprings are created
- Addition of offsprings increases size of population
- Number of members in a population kept is fixed (preferably)
- A constant number of individuals are selected from the individuals of the initial population, and the generated offsprings
- If M is the size of the initial population and No is the number of offsprings created in each generation then, M new parents from M+No individuals are selected
- A greedy selection mechanism may be used (there are several other ways to select too)

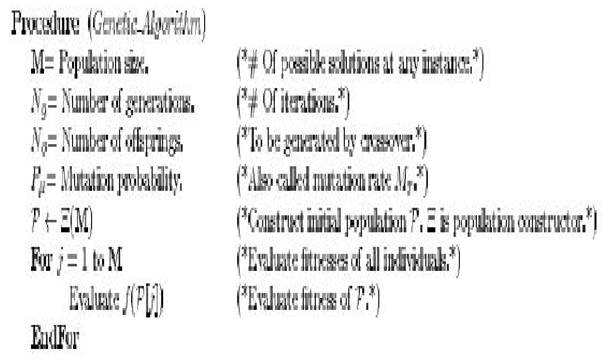
Selection for new generation

- A new generation is formed by selecting a fixed number of individuals from the population of parents and their offspring. Strategies include:
  - Greedy
  - Elitist
  - Roulette wheel
  - Random
  - A combination of some/all of the above
- The fitness of the best individual, will be the same or better than the fitness of the best individual of the previous generation (if greedy, elitist strategy)
- The average fitness of the population will be same or higher than the average fitness of the previous generations
- The fitness of the entire population and the fitness of the best individual increase in each generation

Genetic Algorithm

- An initial population constructor is required to generate a certain predefined number of solutions
- The quality of final solution depends upon the size of the population and the initial population is constructed
- A mechanism to generate offsprings from parent solutions
- Each generation has a set of offsprings that are produced by the application of the crossover operator
- New alleles are introduced by applying mutation

Genetic algorithm



```
Procedure (Genetic_Algorithm)
  M= Population size.              (*# Of possible solutions at any instance.*)
  $N_g$= Number of generations.    (*# Of iterations.*)
  $N_o$= Number of offsprings.     (*To be generated by crossover.*)
  $P_\mu$= Mutation probability.   (*Also called mutation rate $M_r$.*)
  $P \leftarrow \Xi(M)$            (*Construct initial population $P$. $\Xi$ is population constructor.*)
  For $j = 1$ to M                 (*Evaluate fitnesses of all individuals.*)
      Evaluate $f(P[j])$           (*Evaluate fitness of $P$.*)
  EndFor
```

Genetic Algorithm

```
For $i = 1$ to $N_g$
    For $j = 1$ to $N_g$
        $(x, y) \leftarrow \phi(P)$            (*Select two parents $x$ and $y$ from current population.*)
        offspring[$j$] $\leftarrow \chi(x, y)$   (*Generate offsprings by crossover of parents $x$ and $y$.*)
        Evaluate $f$(offspring[$j$])     (*Evaluate fitness of each offsprings.*)
    EndFor

    For $j = 1$ to $N_g$                 (*With probability $P_\mu$ apply mutation.*)
        mutated[$j$] $\leftarrow \mu(y)$
        Evaluate $f$(mutated[j])
    EndFor
    $P \leftarrow Select(P$, offsprings)       (*Select best M solutions from parents & offsprings.*)
EndFor
Return highest scoring configuration in $P$.
End
```

Mutation

- It produces incremental random changes in the offspring generated by the crossover
- Mutation is important because crossover alone will not guarantee to obtain a good solution
- Crossover is only an inheritance mechanism
- The mutation operator generates ónewô characteristics assuring that crossover, the recombination operator, will have the complete range of all possible allele values to explore
- Mutation increases the variability in the population

GA parameters & strategies …

- Size of the Initial population M
  - typical values between 10 and 50
  - depend on available memory
  - convergence rate
  - solution quality
  - larger M may mean a more informed search
- Probabilities of Crossover and Mutation
- Populations constructors:
  - Initial population is constructed randomly
  - Initial population may comprise solutions of some well known constructive heuristics. This method is called seeding and gives best solutions and faster

GA parameters & strategies

- Generation Gap (G) controls the percentage of the population to be replaced during each generation. In each generation M*G offsprings are generated. G=1.0 Means entire generation replaced
  - Steady state GA, Incremental: GA in which only one crossover operation is performed per generation
  - Termination with prejudice: each offspring replaces a randomly selected parent from those which currently have a below-average fitness
  - Elitist strategy: the current best solution is forced to survive and included in the population for the next generation
- A rule of thumb, the computational requirements for both, the genetic operations, and the fitness calculation, must be low (estimates are used)

GA Applications

- Classical optimization problems discussed in our book:
  - The knapsack problem
  - TSP
  - N-queens problem, and
  - the Steiner tree problem
- Engineering problems:
  - Graph partitioning
  - Job shop and multiprocessor scheduling
  - discovery of maximal distance codes for data communications
  - test sequence generation for digital system testing
  - VLSI cell placement, floor planning
  - pattern matching
  - CAD of digital systems: Technology mapping, PCB assembly planning, and High-Level Synthesis of Digital Systems
- Others:
  - Optimization of pipelines systems, Medical imaging to applications, Robot trajectory generation, Parametric design of aircraft

Other issues

- Schema Theorem and Implicit parallelism
- Convergence issues
- Parallelization issues
  - Population is partitioned into subpopulations and they evolve independently using sequential GA
    - Interaction among communities allowed occasionally
    - It represents explicit parallelism
    - It converge faster to desirable solution
    - It is more realistic
  - Parallelization strategies:
    - Island Model
    - Stepping stone Model
    - Neighborhood Model or cellular Model
- Research problems for oometer group