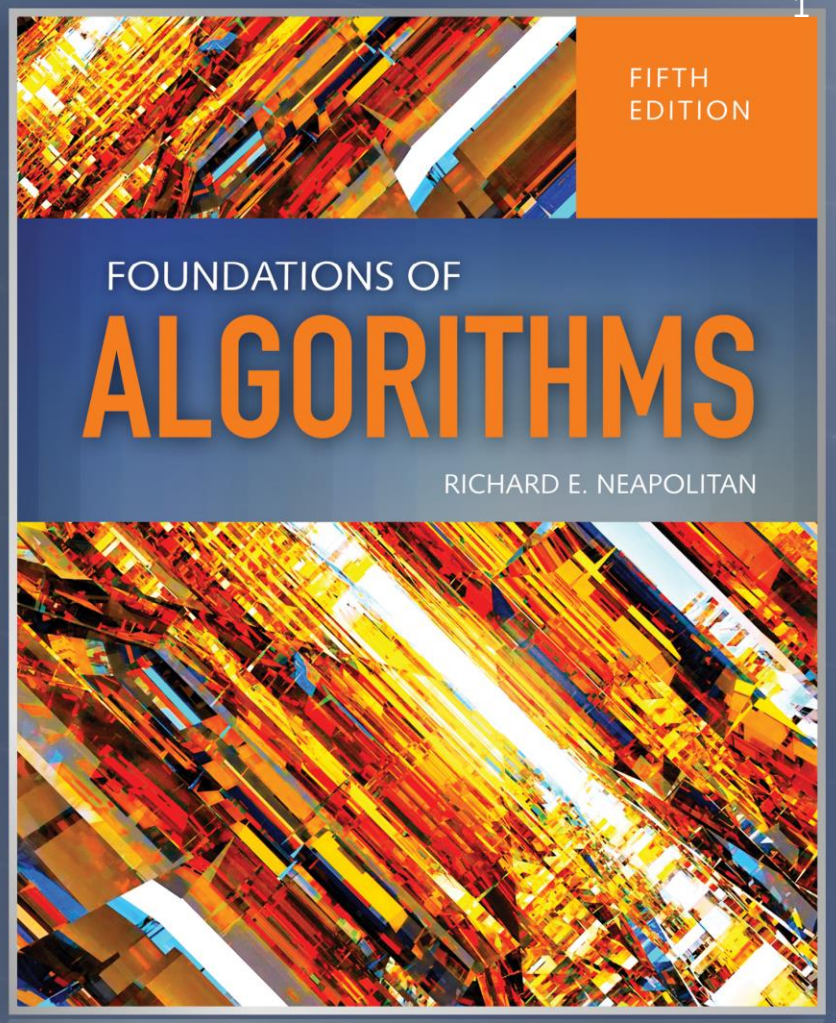
Genetic Algorithms

Chapter 10



Objectives

- Define terms associated with genetics.
- Describe the general approach to genetic algorithms.
- Describe how to select individuals based on fitness.
- Define steps to genetic algorithm development.
- Define a genetic algorithm to solve the traveling salesperson problem.
- Distinguish between genetic algorithms and genetic programming.
- Define the steps to develop a genetic program.

Genetics Review - Definitions

- Organism
- Chromosomes
- Genome
- Haploid cell
- Diploid cell
- Homologous pair
- Somatic Cell
- Haploid organism
- Gamete
- Zygote
- Binary Fission
- Fusion
- Transient Diploid Meiocyte
- DNA
- Nucleotides
- Purines
- Pyrimidines

Definitions Continued

- Canonical Base pair
- Gene
- Genotype
- Phenotype
- Allele
- Meiosis
- Chromatids
- Mutations
- Substitute mutation
- Insertion mutation
- Deletion mutation
- Evolution
- Natural Selection

Genetic Algorithms

- Use fusion in haploid organisms as a model
- Candidate solutions to a problem are represented by haploid individuals in a population
- Each individual has one chromosome
- Alphabet for the chromosome consists of characters representing solutions

Genetic Algorithms

- Generation: a certain number of fit individuals are allowed to reproduce
- Individuals representing better solutions are more fit
- Chromosomes from 2 fit individuals then line up and exchange genetic material
- Mutations possibly occur
- Results in next generation
- Process repeated until terminal condition

```
void generate_populations()
{
t = 0;
initialize population P0;
repeat
evaluate fitness of each individual in Pi;
        Select individuals for reproduction
based on fitness;
        Perform crossover and mutation on
the selected
                Individual;
        t++;
until terminal condition is met;


}
```

Selecting Individuals Based on Fitness

- Exploit knowledge already obtained by concentrating on regions that look good
- Exploration: look for new regions without regard for how good they currently appear
- Explore – choose random individual with probability ε
- Exploit by choosing a fit individual with probability 1- ε

Steps to Develop Algorithm

1. Decide on an alphabet to represent solutions to the problem.
2. Decide on how many individuals make up a population.
3. Decide how to initialize the population.
4. Decide how to evaluate fitness.
5. Decide on which individuals to select for reproduction.
6. Determine how to perform crossovers and mutations.
7. Decide when to terminate.

Traveling Salesperson Problem (TSP)

- NP-hard problem
- n cities. Sales person wants to start at a given city, visit every city once such that the length of the tour is minimum.
- TSP represented by a weighted directed graph: vertices represent cities and weights on the edges represent road length

Genetic Algorithms to Solve TSP

- Order Crossover
- Nearest Neighbor Crossover
- Nearest Neighbor Crossover (NNX)
- Greedy Edge Crossover

Evaluation

- Genetic algorithms do not have provably correct properties.
- Evaluation is done by investigating their performance on a number instances of the problem.

Table 10.5

Algorithm	Mutation	Init. Pop.	Dev	#Gen	Time
	No M	R	3.10	45.39	0.38
		H	4.82	33.52	2.95
NNX	M1	R	1.67	40.21	0.62
		H	1.57	36.09	3.55
	M2	R	0.55	53.37	5.52
		H	0.55	43.53	8.11
	No M	R	12.54	17.35	48.23
		H	7.19	16.37	54.27
GEX	M1	R	4.36	60.44	208.70
		H	3.67	48.44	178.65
	M2	R	3.30	26.30	82.79
		H	3.01	25.83	90.58
	No M	R	8.15	42.50	73.25
		H	5.53	38.47	75.67
50% NNX	M1	R	1.92	66.04	113.81
50 % GEX		H	1.68	61.21	112.77
	M2	R	1.76	19.25	26.40
		H	1.61	20.68	34.19
	No M	R	7.23	41.16	13.39
		H	5.19	34.93	14.95
90% NNX	M1	R	1.84	55.60	19.14
10% GEX		H	1.67	46.93	20.16
	M2	R	0.51	37.13	19.26
		H	0.48	37.24	21.95
	no M	R	6,69	41.23	6.74
		H	5.06	33.04	8.93
95% NNX	M1	R	1.77	52.62	10.03
5% GEX		H	1.41	44.33	11.30
	M2	R	0.49	37.15	11.58
		H	0.44	36.19	14.88

Table 10.6

Algorithm	Mutation	Dev	Time
NNX	No M	5.40	18.4
	M1	1.44	26.4
	M2	0.35	26.2

Table 10.7

Algorithm	Mutation	Dev	Time
NNX	No M	7.61	25.3
	M1	4.94	65.0
	M2	4.70	1063.0

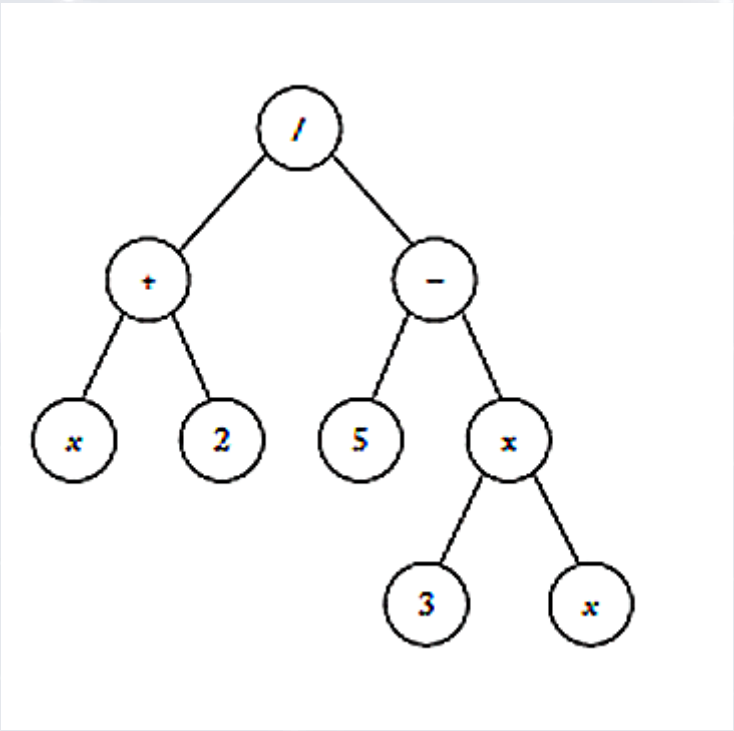
Genetic Programming

- Genetic algorithms: “chromosome” or “individual” represents a solution to a problem
- Genetic Programming: the individual represents a program that solves a problem
- Fitness function for the individual measures how well the program solves the problem.

Individuals in a Genetic Program

- Represented by trees
- Each node
  - Terminal symbol
  - Function symbol
- Function symbol: arguments are its children

Figure 10.7



Illustrative Example

- Points generated from $y=x^2/2$

Table 10.9

x	y
0	0
.1	.005
.2	.020
.3	.045
.4	.080
.5	.125
.6	.180
.7	.245
.8	.320
.9	.405

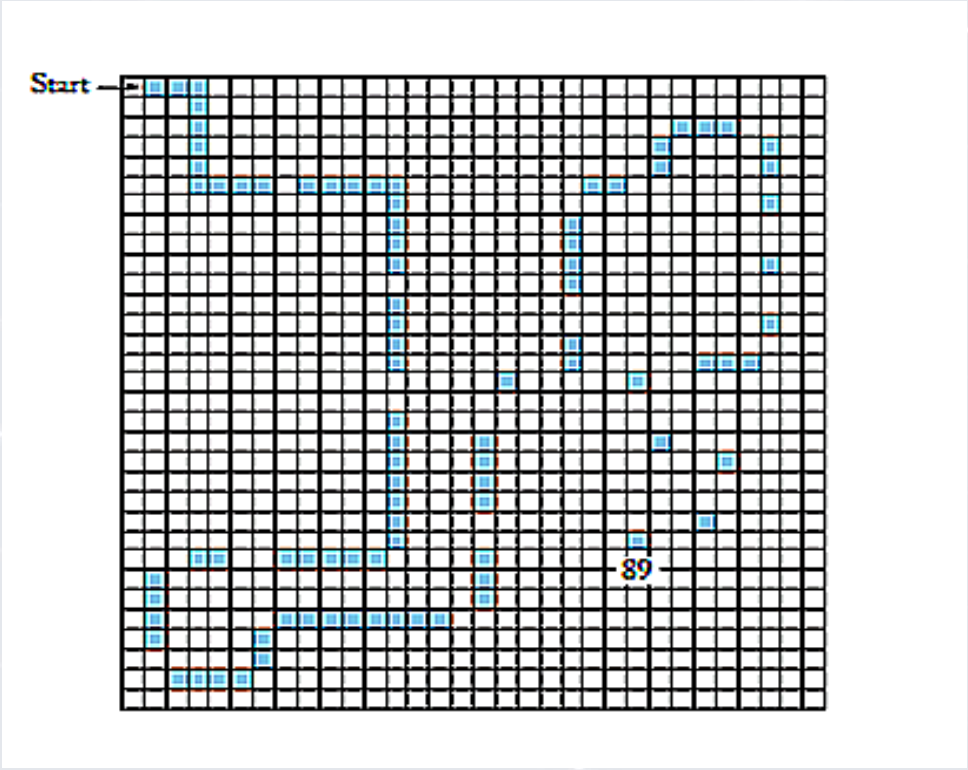
Steps for developing a genetic program for the discovery problem

1. Decide on the terminal set T.
2. Decide on the function set F.
3. Decide on how many individuals make up a population.
4. Decide how to initialize the population.
5. Decide on a fitness function.
6. Decide on which individuals to select for reproduction.
7. Decide on how to perform crossovers and mutations.
8. Decide when to terminate.

Artificial Ant

- Sante Fe Trail – black squares represent one pellet of food (89 such pellets)
- Ant starts at square labeled start facing right
- Goal: arrive at square labeled 89 after visiting all 89 black squares – eating all of the food on the trail – in as few steps as possible
- The problem with a time limit represents a challenging planning problem

Figure 10.9



Application to Financial Trading

- On a given day, decide to buy, sell, or hold
- Develop genetic program for the trading system