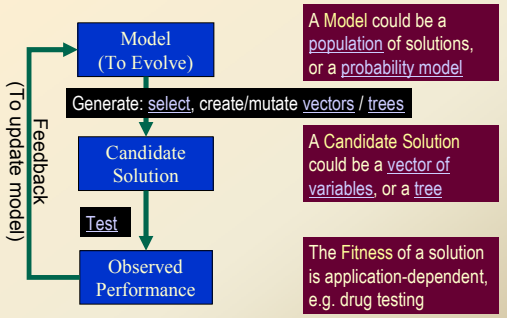
Evolutionary Computation: Model-based Generate & Test



Motivation

- Idea from natural selection
- To contain *combinatorial explosion*
  - E.g. the travelling salesman problem
- To evolve quality solutions
  - E.g. to find hardware configurations

Terminology in GA

String with Building blocks with values

1	0	0	0	1	1	0

Chromosome with ***Genes*** with ***alleles***

evaluation ↓

fitness

- Example of a candidate solution
- in binary representation (vs ***real coding***)
- A ***population*** is maintained

Edward Tsang (all rights reserved)

Example Problem For GA

maximize f(x) = 100 + 28x x^2

optimal solution: x = 14 (f(x) = 296)

Use 5 bits representation

e.g. binary 01101 = decimal 13

f(x) = 295

Note: representation issue can be tricky

Choice of representation is crucial

10 November, 2009 Copyright Edward Tsang 5

Example Initial Population

To maximize f(x) = 100 + 28x $x2$

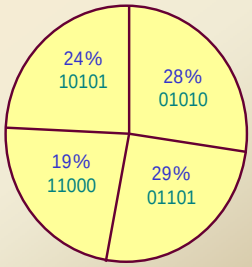
No.	String	Decim.	f(x)	weight	Accum
1	01010	10	280	28	28
2	01101	13	295	29	57
3	11000	24	196	19	76
4	10101	21	247	24	100
		Accum	1,018		
		averg	254.5		

10 November, 2009 Copyright Edward Tsang 6

Selection in Evolutionary Computation

Roulette wheel method

The fitter a string, the more chance it has to be selected



10 November, 2009 Copyright Edward Tsang 7

Claimed advantages of GA

- Simple to program
- General -- wide range of applicability
- Efficient -- finding solutions fast
- Effective -- finding good solutions
- Robust -- finding solutions consistently

➢ Are these claims justified?

➢ Different people have different opinions

Epistasis
(interaction between Genes)

Constraints

Penalties, Repair

Guided Genetic Algorithm

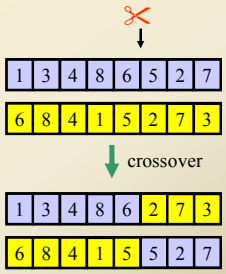
Constraints

- Task: find optimal solution in constrained satisfaction
- Difficulties:
  - Before one can attempt to optimization, one needs to find legal solutions
  - Crossover may generate illegal solutions
  - Sometimes satisfying constraints alone is hard
    - When problem is tightly constrained

Edward Tsang (all rights reserved)

Epistasis, Example in TSP

- Travelling Salesman Problem
- After crossover, offspring may not be legal tours
  - some cities may be visited twice, others missing



Penalty Method

- If a string violates certain constraints
- Then a penalty is deducted from the fitness
  - E.g. in TSP, if penalties are high, GA may attempt to satisfy constraints before finding short tours
- Problem with tightly constrained problems:
  - most strings are illegal

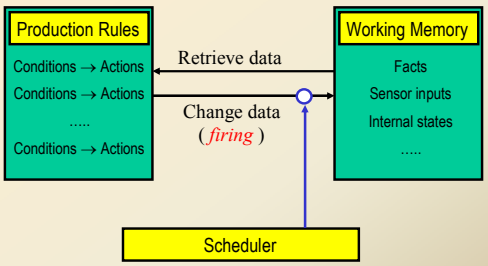
Repair Method

- If a string violates certain constraints
- Then attempt to repair it
  - E.g. in TSP, replace duplicated cities with missing cities
  - possibly with complete search or heuristics
- Make sure that a population only contains legal candidate solutions
  - One can then focus on optimization

GA for Machine Learning

Classifiers
Bucket Brigade

Production System Architecture



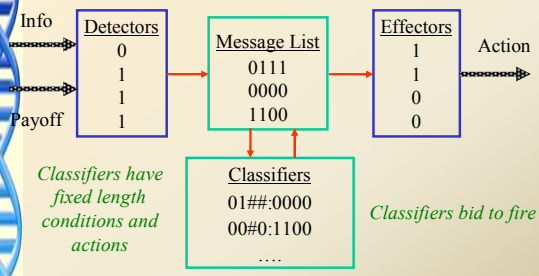
10 November, 2009 Copyright Edward Tsang 31

Classifier System Components

- *Classifiers*: Condition-Action rules
  - Special type of *production system*
- A Credit System
  - Allocating rewards to fired rules
- Genetic Algorithm
  - for evolving classifiers

10 November, 2009 Copyright Edward Tsang 32

Classifier System Example



Classifiers have fixed length conditions and actions

Classifiers bid to fire

Apportionment of Credit

- The set of classifiers work as one system
  - They react to the environment
  - The system as a whole gets feedback
- How much to credit each classifier?
  - E.g. *Bucket Brigade method*
- Each classifier *i* has a strength S_i
  - which form the basis for credit apportionment
  - as well as fitness for evolution

Bucket Brigade, Basics

- A classifier may make a bid to fire

  $B_i = S_i * C_{bid}$

  - where C_{bid} is the bid coefficient
- Effective bid: $EB_i = S_i * C_{bid} + N(\sigma_{bid})$
  - where $N(\sigma_{bid})$ is a noise function
  - with standard deviation σ_{bid}
- Winner of an auction pays the bid value to source of the activating message

Edward Tsang (all rights reserved)

Classifiers In Action

Classifiers	S_0	Msg	B_0	S_1	Msg	B_1	S_2	Msg	B_2	S_3	Msg	B_3	S_4	Msg	B_4	R_4	S_5
01##:0000	200		20	180	0000		220			220			220				220
00#0:1100	200			200		20	180	1100		218			218				218
11##:1000	200			200			200		20	180	1000		196				196
##00:0001	200			200		20	180	0001	18	162	0001	16	146	0001		*50*	196
Environment	0	0111		20			20			20			20				20

$C_{bid} = 0.1$

$C_{tax} = 0$

More on Bucket Brigade

- Each classifier is "taxed"
- $S_i(t+1) = S_i(t) - C_{bid}S_i(t) - C_{tax}S_i(t) + R_i(t)$
  - where $S_i(t)$ is strength of classifier *i* at time *t*
  - $C_{bid}S_i(t)$ is the bid accepted
  - C_{tax} is *tax*
  - $R_i(t)$ is payoff
- For stability, the *bid value* should be comparable to *receipts* from environment

Classifiers: Genetic Algorithms

- T_{ga} = # of steps between GA calls
  - GA called once every T_{ga} cycles; or
  - GA called with probability reflecting T_{ga}
- A proportion of the population is replaced
- Selection: roulette wheel
  - weighted by strength S_i
- Mutation: $0 \rightarrow \{1,\#\}$, $1 \rightarrow \{0,\#\}$, $\# \rightarrow \{0,1\}$

Edward Tsang (all rights reserved)

Genetic Programming

Building decision trees
GP for Machine Learning

Genetic Programming, Overview

- Young field
  - Koza: Genetic Programming, 1992
  - Langdon & Poli: Foundations of GP, 2001
- Diverse definitions
  - Must use trees? May use lists?
- Must one evolve programs?
  - Suitable for LISP
- Machine learning: evolving trees
  - dynamic data structure

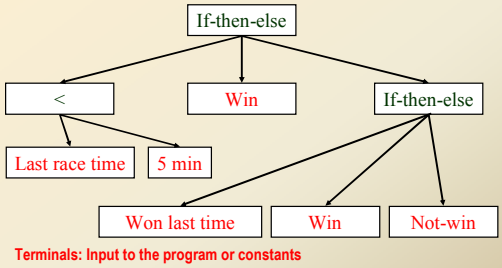
Terminals and Functions

- Terminal set:
  - Inputs to the program
  - Constants
- Function set:
  - Statements, e.g. IF-THEN-ELSE, WHILE-DO
  - Functions, e.g. AND, OR, +, :=
  - Arity sensitive

Edward Tsang (all rights reserved)

GP: Example Tree (1)

Functions: Statements (e.g. IF-THEN-ELSE) or functions (e.g. AND, OR, +, –)



Terminals: Input to the program or constants

10 November, 2009 Copyright Edward Tsang 42

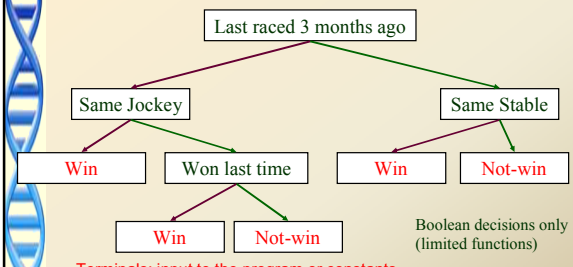
GP Application

- A tree can be anything, e.g.:
  - a program
  - a decision tree
- Choice of terminals and functions is crucial
  - Domain knowledge helps
  - Larger grammar ⇒
    larger search space ⇒
    harder to search

10 November, 2009 Copyright Edward Tsang 43

GP: Example Tree (2)

Functions: Statements (e.g. IF-THEN-ELSE) or functions (e.g. AND, OR, +, –)

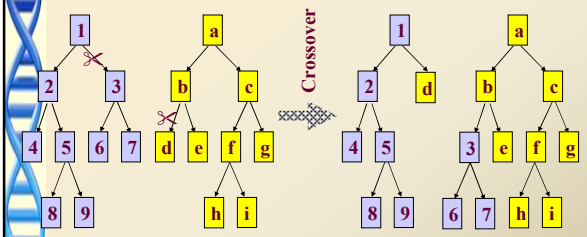


Boolean decisions only (limited functions)

Terminals: input to the program or constants

10 November, 2009 Copyright Edward Tsang 44

GP Operators



Mutation: change a branch

Fitness in GP

- Generating Programs
  - How well does the program meet the specification?
- Machine Learning:
  - How well can the tree predict the outcome?
- Function Fitting:
  - How great/small the error is

Generational GP Algorithm

- **Initialize population**
- **Evaluate individuals**
- **Repeat**
  - **Repeat**
    - **Select parents, crossover, mutation**
  - **Until enough offspring have been generated**
- **Until termination condition fulfilled**

Steady-state GP Algorithm

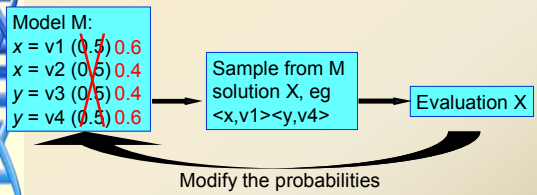
- Initialize population P
- Repeat
  - Pick random subset of P for tournament
  - Select winners in tournament
  - Crossover on winners, mutation
  - Replace loser(s) with new offspring in P
- Until termination condition fulfilled

Estimation of Distribution Algorithms (EDAs)

Population-based Incremental Learning (PBIL)

Building Bayesian networks

Population-based Incremental Learning (PBIL)



- Statistical approach
- Related to ant-colonies, GA