

## Heuristic Search

### 3.1 Simulated Annealing

Recall that hill climbing could be an approximate search or optimization. It can sometimes be improved by adding a dash of randomness. This idea has been carried to a method known as *simulated annealing*. This is a search based on brownian motion of a physical system settling down: like a mixture of random walk and hill climbing.

Assume we are trying to find the global minimum. The analogy is a physical “cooling” of the system: think of a ping-pong ball bouncing around a bumpy terrain, eventually settling in the lowest spot (or at least a local minimum). We assume there is an **objective function** which is to be minimized.

#### SIMULATED ANNEALING

- choose initial position and temperature  $T$
- repeat until happy
  - find random neighbor  $N$  of current position
  - let  $\Delta$  be change in objective function
  - if  $\Delta < 0$  (meaning  $N$  is better) then go to  $N$
  - else go to  $N$  with probability  $e^{-\Delta/T}$
  - lower  $T$

To improve performance, one may store good intermediate results, and every now and again use randomness or one of these to revive the process. One major application is in problems where there is a mixture of discrete and continuous variables. Plain search does the former well and calculus the latter, but not many approaches work for such hybrids. Simulated annealing has had success on the *traveling salesman problem*.

### 3.2 Genetic Algorithms

Genetic algorithms are based on the concept of evolution. The idea is to breed the best solution using survival of the fittest.

In general, one starts with an initial population of candidates. Then in each round (called generation) one does the following:

## GENETIC ALGORITHM

- repeat until happy
  - evaluate individuals by fitness function
  - select some to survive
  - select some to crossover: combine pair into another pair
  - select some to mutate: randomly change slightly
  - update population

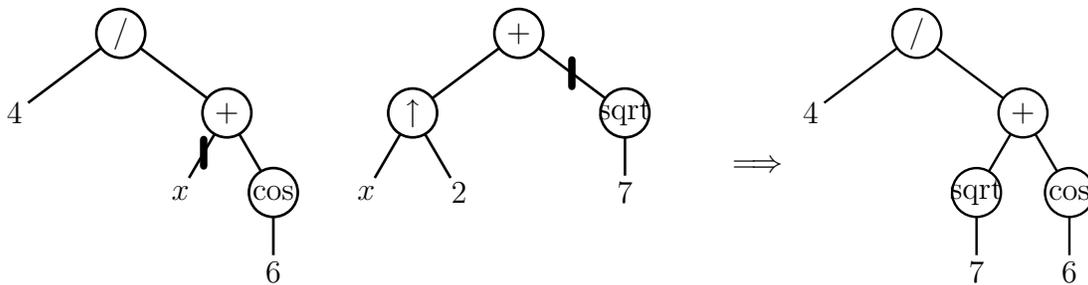
Naturally, the fitter the individual the more chance it has of breeding. In some implementations, all individuals die and each generation is a renewal.

The simplest representation of the data is a bit-string. Then

- a crossover is achieved by cutting the two strings randomly in pieces, and gluing the left piece of the one to the right piece of the other.
- A mutation is simply flipping a random bit(s).

There are many parameters to play with in any particular implementation: so one should try several settings.

One classic application of genetic algorithms is to develop a function that fits a given set of data (a scatter plot  $x_i, y_i$ ). The idea is to store the candidate functions as **expression trees**. For evaluation, compare the function value and the given data (and look at maximum difference or sum of squares of differences). For the crossover, choose an edge of each tree: chop the trees at that edge and swap branches. Mutation is the random change of operation or operand. Here is crossover:



One positive about this approach is that the crossover allows one to use the work done getting two candidates partly right: normally, search rejects a solution even though it is partly right. Genetic algorithms are best when the problem is loosely coupled (and so small local changes to a candidate do not radically alter its behavior).

### Exercises

- 3.1. Consider using a genetic algorithm to solve the 8 queens problem. Discuss the details of the implementation.

- 3.2. Use *simulated annealing* to solve the *Gross queens problem*. The task is to place 144 queens, a dozen of each of a dozen colors, on separate squares of the  $12 \times 12$  board, such that no two queens of the same color attack each other.
- 3.3. Write a genetic algorithm to produce word squares. The idea is an arrangement in a  $4 \times 4$  square such that each row and column is a word. Example:

B	U	D	S
E	P	E	E
T	O	N	E
A	N	T	S

- 3.4. For the *Wandering Professor problem*, an academic has to give lectures at all 972 state universities in the country, subject only to the constraint that the one-word names of consecutive universities may not have a letter in common: he may not go from Clemson to Florida but he may go from Clemson to Hawaii.
- (a) Discuss the details of implementing a genetic algorithm for this.
- (b) Discuss the details of implementing simulated annealing for this.