

Genetic Programming for Julia:

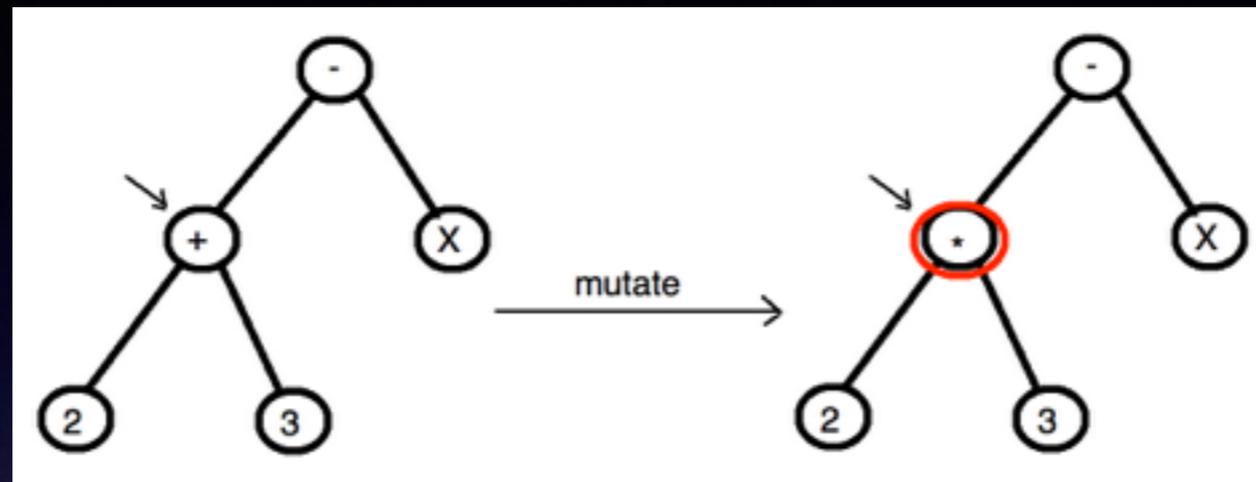
fast performance and island model implementation

Morgan R. Frank

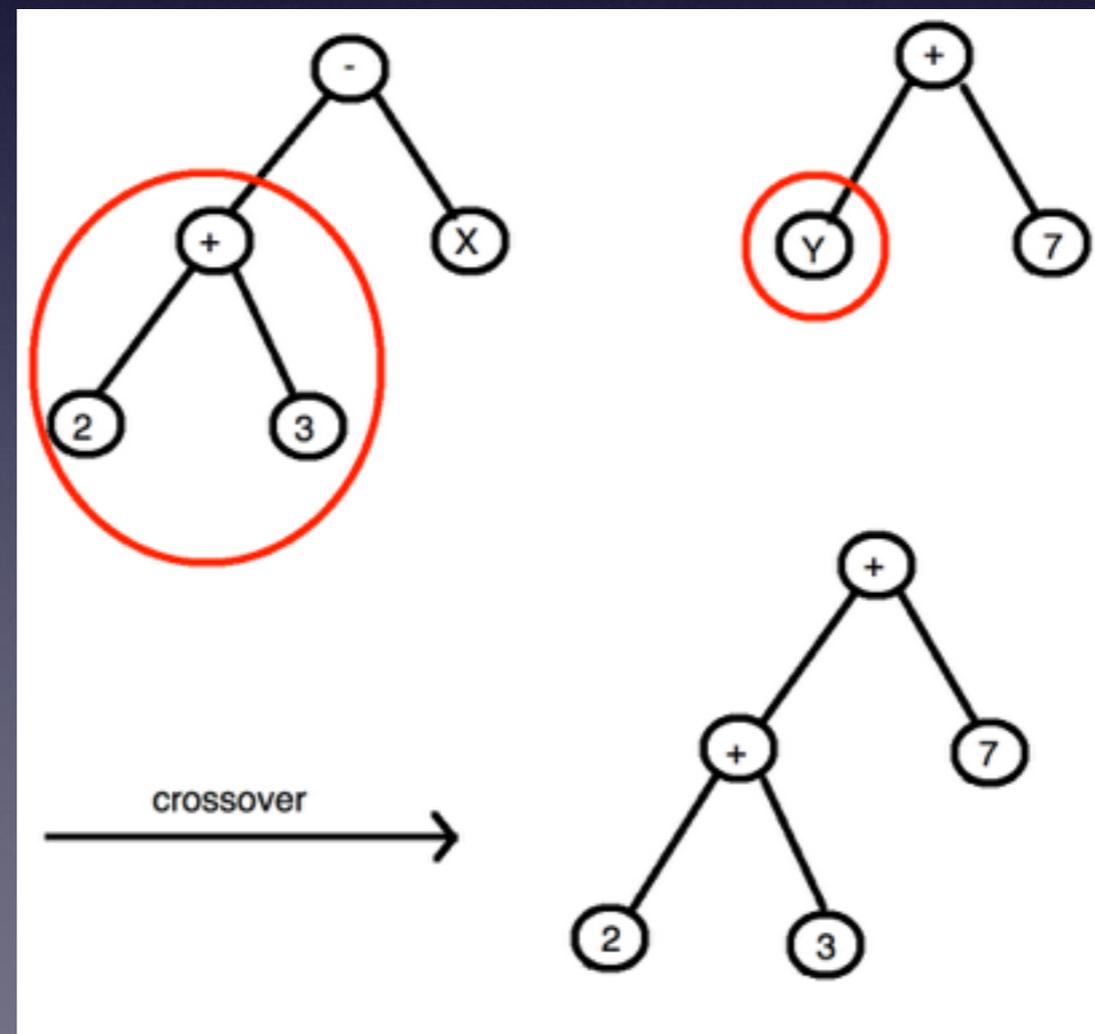
What is Genetic Programming?

- Evolutionary computation refers to a collection of algorithms that solve problems using biological evolution
 - pick a problem and model potential answers as genotype, and their ability to solve the problem is the phenotype
 - implement genetic mutation, genetic crossover, and selection pressure to find good solutions
 - see GeneticAlgorithm.jl
- Genetic Programming is one such algorithm that represents answers as syntax trees

What is Genetic Programming?

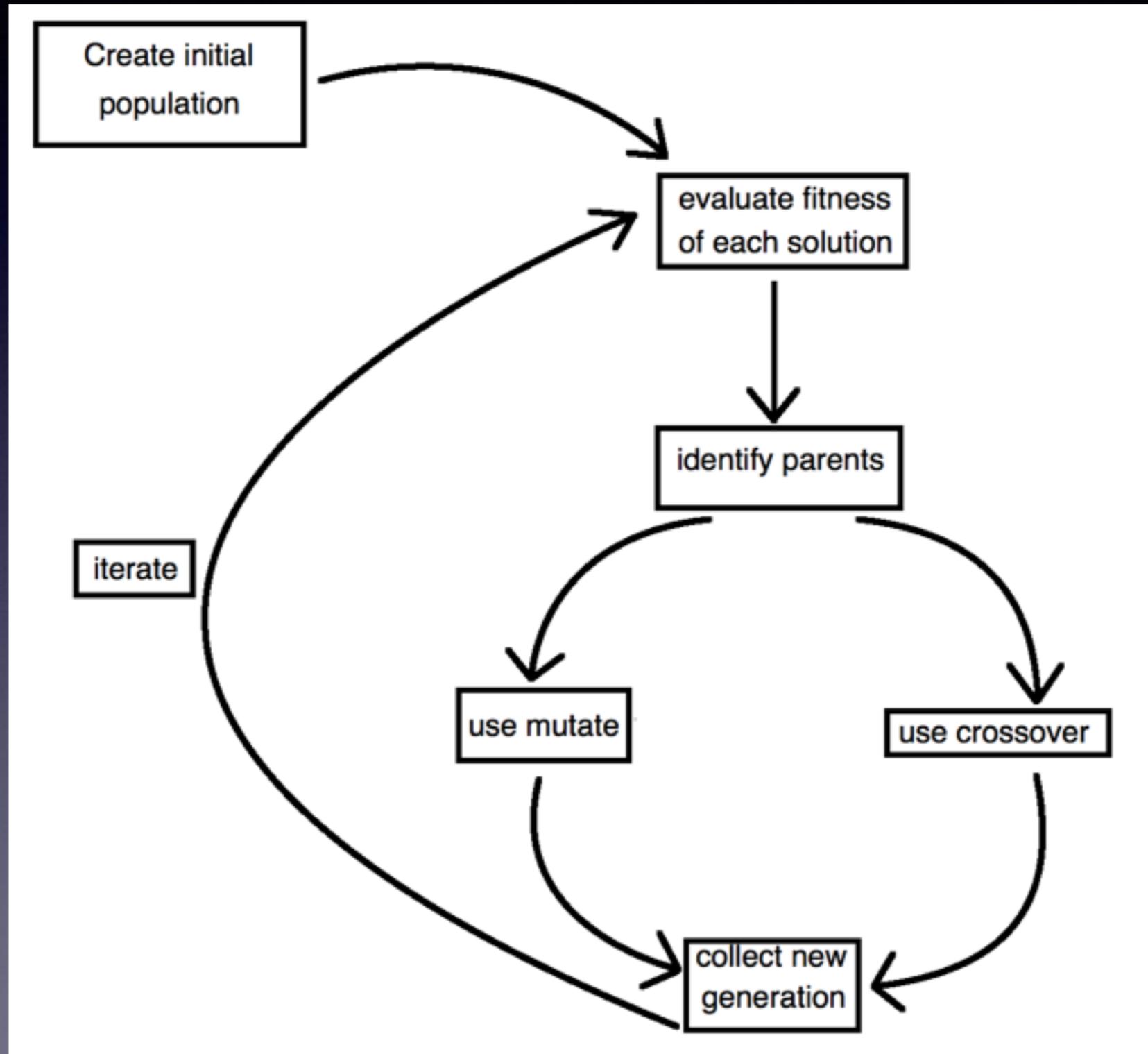


We restrict ourselves to symbolic regression



GP: Algorithmic Flow

in serial



GP: Parallelization with Island Model

- pieces of serial GP can be implemented with embarrassingly parallel methods
 - requires frequent communication between processes
- serial GP can converge prematurely
- want efficiency and to maintain exploration

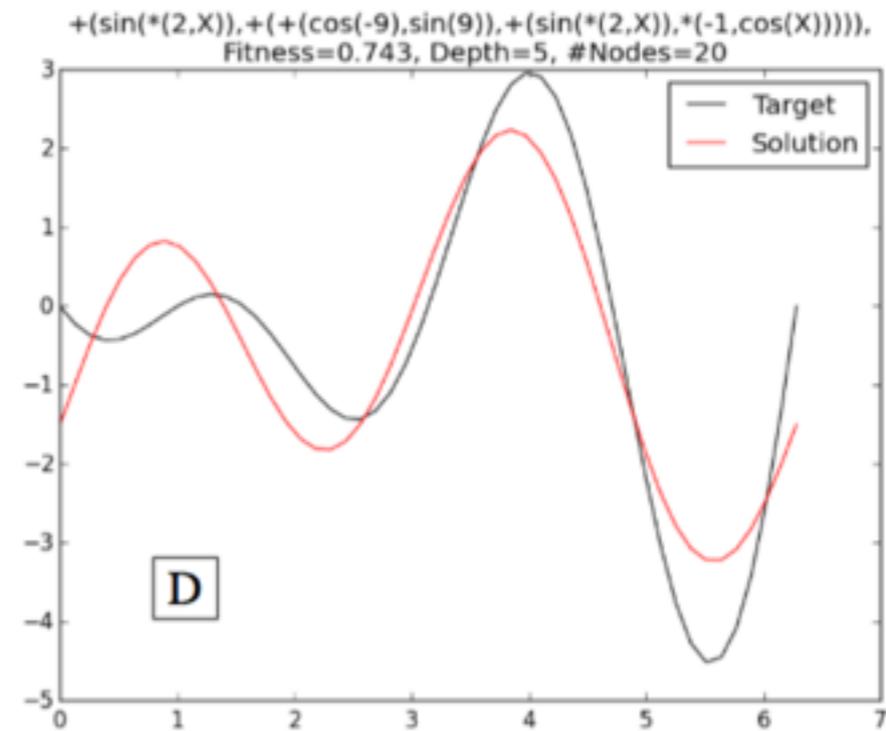
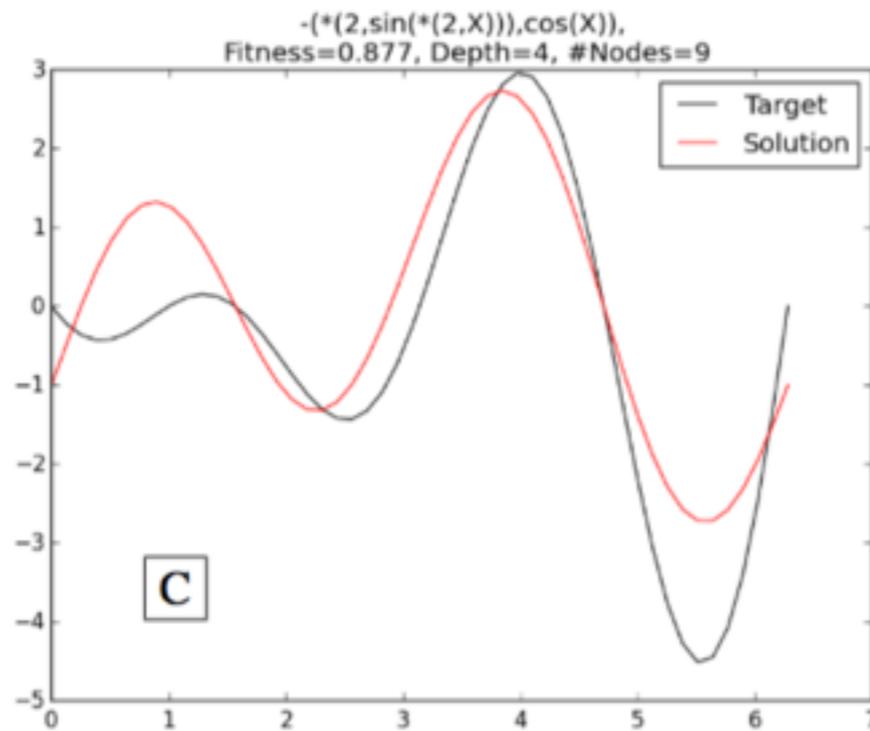
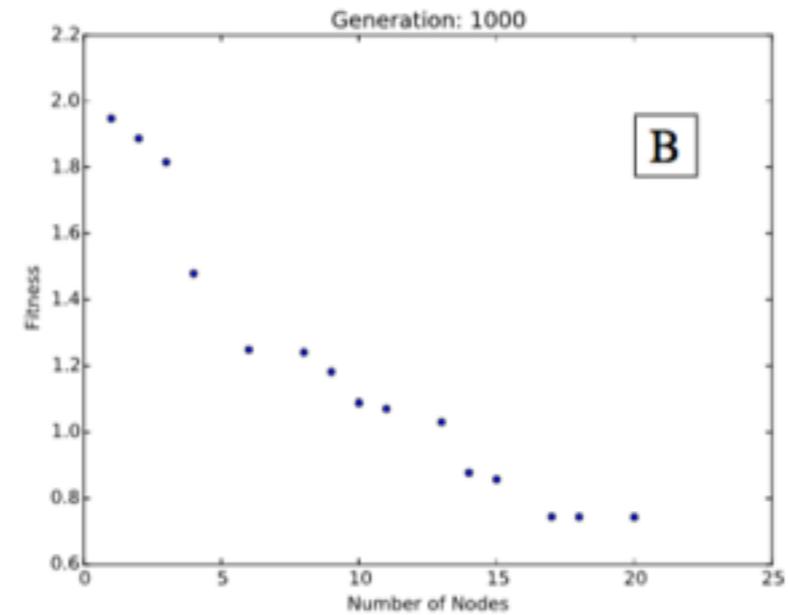
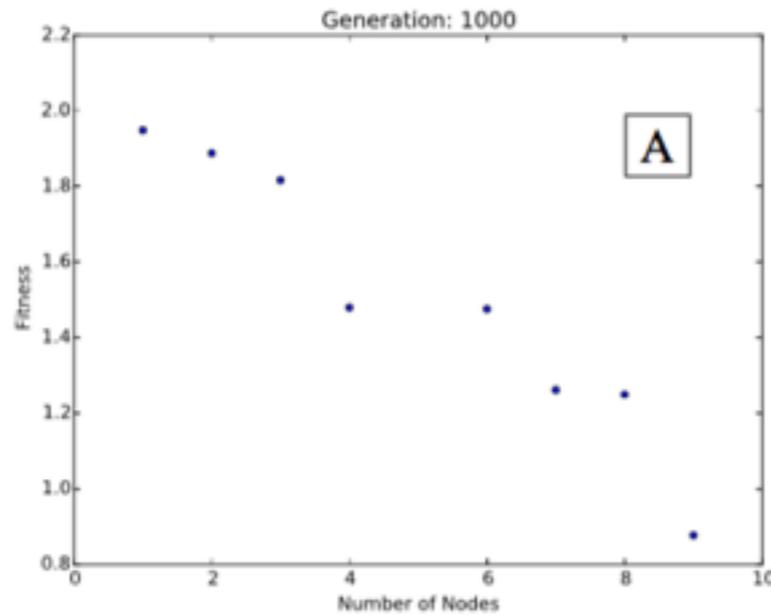
GP: Parallelization with Island Model

- each process runs an independent GP population, called an island
- infrequently, each island randomly selects a few good solutions to send to another island.
 - call this process “migration”
- almost embarrassingly parallel, but with infrequent communication of only a little data
- migration discourages individual populations from prematurely converging
 - ie. more exploration of solution space

An Example using GP

Serial

Island Model



Future Work

- parameters and parameter optimization instead of constant terminals
- combat tree bloat by occasionally condensing syntax trees
- extend GP capability in Julia to handle more than just symbolic regression
 - ex: GP for debugging, or for algorithmic generation