Flexible Code Foundation Development For Further Study In Gender-based GA with Recombination Hotspots

Presenter: Siu Kei, Muk (u5721042)
Supervisor: Tom Gedeon
Acknowledgement

• I would like to thank Prof. Tom Gedeon for valuable comments and support on the project.

• I would also like to show my gratitude to Prof. Bob Mckay for sharing his knowledge and insights on genetic mechanisms in biology.
Background

• The natural has evolved many amazing biological systems
• Designs are self-sustaining and highly optimized
• Bio-inspired computing aims to replicate properties of such systems:
  • Evolvability
  • Self-sustainability
  • Etc
• Examples:
  • Neural Networks, Genetic Algorithms
What is Genetic Algorithm

• Genetic algorithm is a family of algorithms that solves optimization problem by mimicking natural selection.
• Candidate solutions are encoded as chromosomes
• New candidate solutions are generated by:
  • Reproduction (Recombination/crossover + mating)
  • Mutation, etc
• Solutions with better performance has higher chance to participate in reproduction process (selection)
Motivation

• Traditional genetic algorithm is not scalable  
  • One needs a more effective way to obtain good solutions
• The biological evolution is more effective in several ways:
  • Traits heritability  
  • Adaptability  
  • Maintenance of diversity
• Need extremely flexible platform to explore
Motivation

• Recent research - recombination hotspots
• Human : Chimpanzee
  • 99% similar in genetic material
  • Completely different recombination rate distributions
• Male : Female – Different hotspots
• Current GA assumes recombination rate independent of location
Overview

• The implementation is divided into two types of modules
  • Main (Core) module
  • Extension module
• The main module provides a foundation for general GA
• Extension modules provide special features
Main Module

- Hierarchical Architecture
- Main loop is to be defined in the client code
- Frame – administrative use
- State – state management
- Population – pools of individuals
State

• A state consists of:
  • Population
  • Genetic Operators (Recombination, Mutation)
  • Selection Operator

• The “State” class governs the current state and triggers reproduction and mutation
Frame

• A frame consists of a state, and several administrative operations

• It defines a single generation evolution step
Extension Module (Gender-based & hotspots)

- This module provides implementation for gender-based GA with recombination hotspots
- Some classes are re-defined to specifically match the requirement:
  - Components – Chromosome, Hotspots
  - Collections – Population, Statistics
  - Operations – Hotspot mutation, elitism, reproduction, selection
  - Frame – Frame, state
Hotspot

• Provides a recombination rate for each gene location.
• Comprises two parts:
  • Encoding (for discrete probability level)
  • Actual rate
• Mutation is done by specialized operator
Features

• Coupleable Interface: Ensures a chromosome has gender and hotspot
• Population:
  • Forces individuals to be coupleable
  • Maintains a fixed proportion of male and female
• Statistics: Contains separated records for different gender
• State, Frame: Takes care of hotspot mutation
• **Selector: Selects exactly one individual from each gender
• Reproduction Operator: Recombination guided by hotspots
Illustrations
Implementation Map

Initializer creates Population

Fitness Function
Mutator
Reproducer
Selector
Hotspot Mutator

Dynamic Handler
Prior Operator
Post Operator
Statistics

Frame

Main Loop
Flexibility of Model

• Easily extendable

• Restriction is minimal

• Expression Map extends Dominance Map

• Operations are interchangeable on run-time
Future Works

• Main module modification

• Extension module development

• Exploration of possible effective biological features to be included in GA
Questions
Individual Components

- Gene – Container of a single value
- Material – Container of genes
- Chromosome – Container of
  - Genotype – a set/list of materials
  - Phenotype – a material
  - Expression Map – a map from genotype to phenotype
- Individual – Comparable wrapper of a chromosome
Population

• A population consists of several pools of individuals

• Responsible for holding individuals for
  • Current generation
  • Upcoming generation
Operations

- Operations are split into several categories
  - Administrative operations
  - Genetic operations
  - Others

- Each operation is abstracted as an interface for flexible usage

- An object of the operation interface is provided at the time the process requires
Administrative Operations

• Prior Operation:
  • Performed before reproduction
  • Example: Elitism operation
  • **Optional

• Post Operation:
  • Performed after reproduction
  • Example: Filling operation

• Statistics
  • Keeps record of the evolution states
  • Example: saves the best performing individual in each generation
Genetic Operations

• Operations that creates/modifies chromosomes

• Reproduction
  • Generates offspring from given parents by recombination

• Mutation
  • Modifies gene values randomly
Other Operations

• Fitness Function
  • Evaluates the objective function value, given a phenotype

• Selection Operator
  • Selects parents according to their fitness

• Expression Map
  • Maps a genotype to its corresponding phenotype

• Initializer
  • Initializes a population
  • **Optional