Evolutionary Algorithm Sandbox

A Web-Based Graphical User Interface for Evolutionary Algorithms

Brent G. Gardner and Dan Simon
Cleveland State University

Supported by NSF
CMMI Grant 0826124

IEEE Conference on Systems, Man, and Cybernetics
San Antonio, Texas
October 12, 2009

Outline

- Introduction
- Software Design
  - Interactive GUI
  - Simulation
- Usage
- Extendibility
- Conclusion

Introduction

- The Evolutionary Algorithm (EA) Sandbox is a graphical user interface (GUI) for EAs
  - Web-based for easy accessibility
  - Available for download
  - Meant for *quick parameter testing* and *tradeoffs*
  - Extendible through Adobe® Flex®
- Goal
  - Provide an easily accessible learning tool that inspires users (especially students) to learn more about EAs
  - Ability to see an EA simulation in action

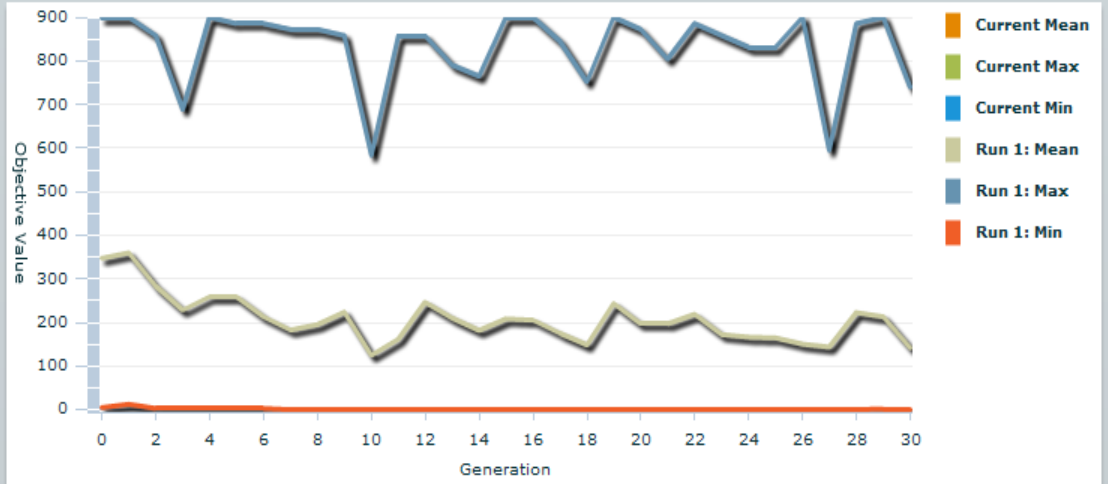
Introduction

- Previous EA GUIs:
  - Focused on one algorithm
  - Written in archaic languages
  - Required third-party simulation software (Matlab)

- EA Sandbox:
  - Adobe Flex
    - MXML/ActionScript (similar to HTML/JavaScript)
  - Advanced graphics components
  - Requires web browser and Adobe Flash Player

Evolutionary Algorithm Sandbox

Statistics Graph



Clear Hidden | Clear All | Export Shown | Import

Current Population

Genotype	Objective Value
1111010	1.9775390625
1111001	2.691650390625
1111001	2.691650390625
1111000	3.515625
1110111	4.449462890625
1110110	5.4931640625
1110101	6.646728515625
1110100	7.91015625
1110011	9.283447265625
1101111	15.875244140625
1101100	21.97265625
1101010	26.5869140625
1101001	29.058837890625
1100101	40.045166015625

Control Panel

Stop Conditions: Generation Limit: 30 | Objective Value Limit: Mean | Below | -1000 | Mean | Objective Value Change Less Than: 0

Population Properties: Number of Islands: 30 | SIVs per Island: 1 | Bits per SIV: 8 | Bits per Island: 8

Problem Parameters: Objective function: Axis parallel hyper-ellipsoid ? | Phenotype Range: -30 to 30 | for SIV #: 1 | ☑ Apply to All

Common Options: Mutation Probability: 0.003 per bit | ☑ Clear Duplicates | ☑ Use Gray Coding | ☐ Use Elitism

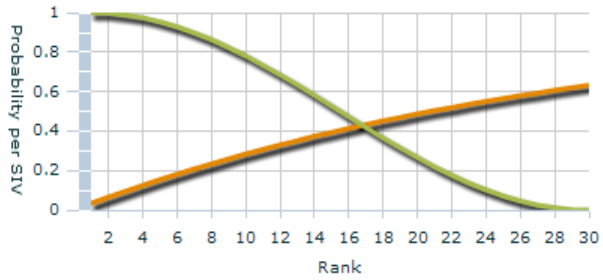
Basic GA | BBO

Immigration Probability Function:

Exponential

Emigration Probability Function:

Sine [-pi/2,pi/2]

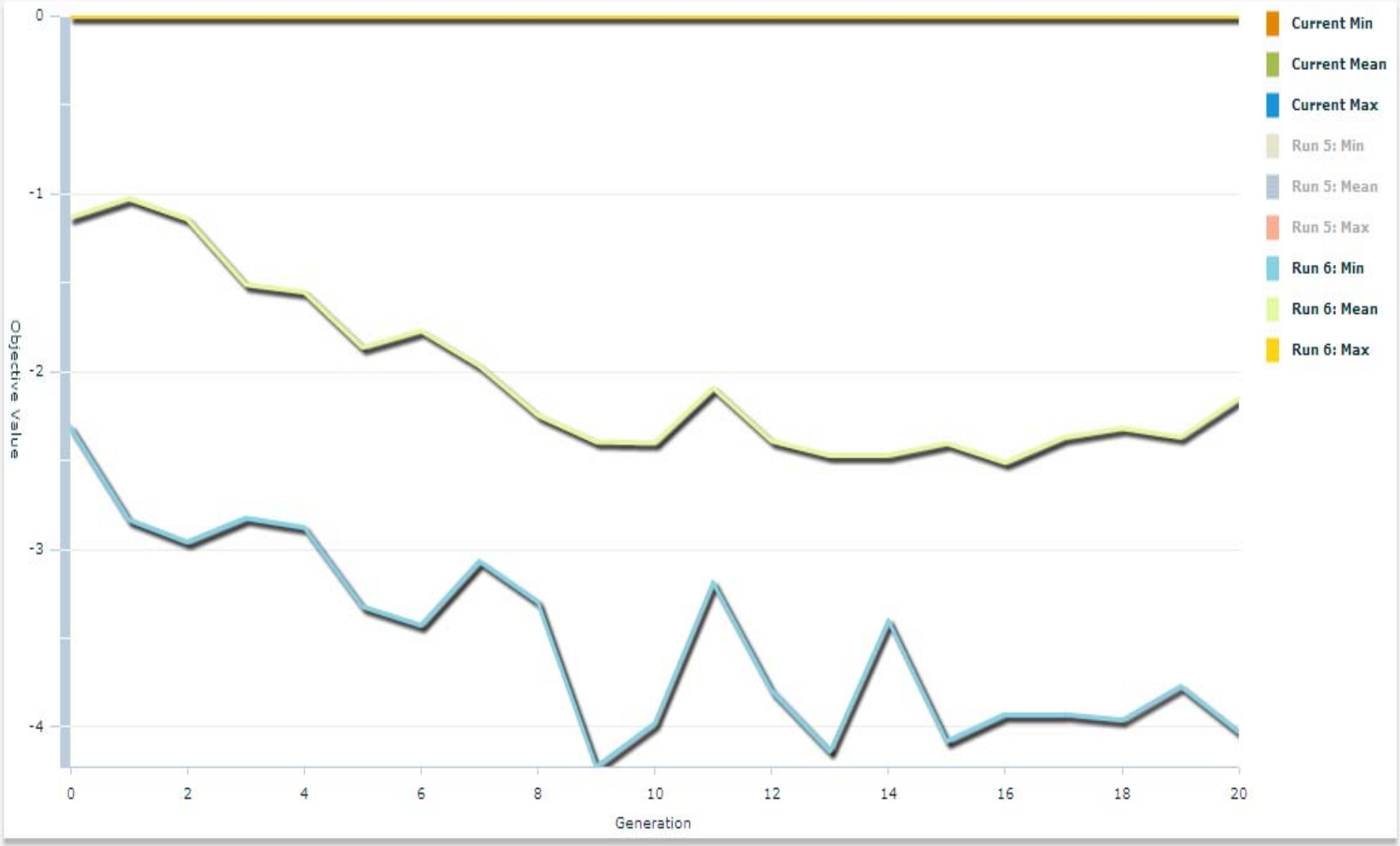


Start | ☑ Run Continuously | Step Delay: | Step | Stop | Generation 30 of 30 | Save | Load

Evolutionary Algorithm Sandbox

Status: Simulation Complete

Control Panel



Run 1: Mean ▾ | Add | Show All | Hide All | Clear Hidden | Clear All

Auto-add: ☑ Min ☑ Mean ☑ Max | ☑ Autoscale | Scale to Fit | Export | Import

Evolutionary Algorithm Sandbox

Control Panel

Genotype	Objective Value
01100101100010001100110110101000101110010100101000010110010100101010100011001 1	-4.0279251966298855
01100101110110011100001011101000111100111111100100000001010011111010100001000 0	-3.9671613783038024
01110010100000011110110111111111111110011101110111	500322
00100000100000101100110110011000111100110100100	0262918
01100101100000100100101011111000101110001110110	9006627
01100111100000001100001000101000111100111111000	016326
11010011111111010110000100011100100010010010010	083957
00101000100100101100110111111100101100001101000	435827
01010001100010001100111010110010000001010110110	129397
01110110100000010111110100101000101101000000111	135203
10010010100010001110011000111000101110011110000	3129193
00001000111010101100110111001101000101100100101010010110011011111110011011000 1	-1.6984447536441973
01111111100100101100110101111010000101101110000000000101111011011101011001100 1	-1.6940855988845502
10010010100010000100111111011000110011010100101010010110111001010010100100000 1	-1.6267322580100285
01111110111110111100100111111110011110011010110000000101110000001110011010010 1	-1.5683025928349026

Individual Genotype Breakdown

SIV #:

1

Genotype:

01100101

Phenotype:

0.6796875

Evolutionary Algorithm Sandbox

Control Panel

Stop Conditions: Generation Limit: 10 | Objective Value Limit: Mean | Below | -1000 | Mean | Objective Value Change Less Than: 0

Population Properties: Number of Islands: 10 | SIVs per Island: 1 | Bits per SIV: 8 | Bits per Island: 8

Problem Parameters: Objective function: De Jong 1 ? | Phenotype Range: 0 to 1 for SIV #: 1 | Apply to All

Common Options: Mutation Probability: 0.003 per bit | Replace Duplicates | Gray Coding | Elitism | Selection Strategy: μ,λ

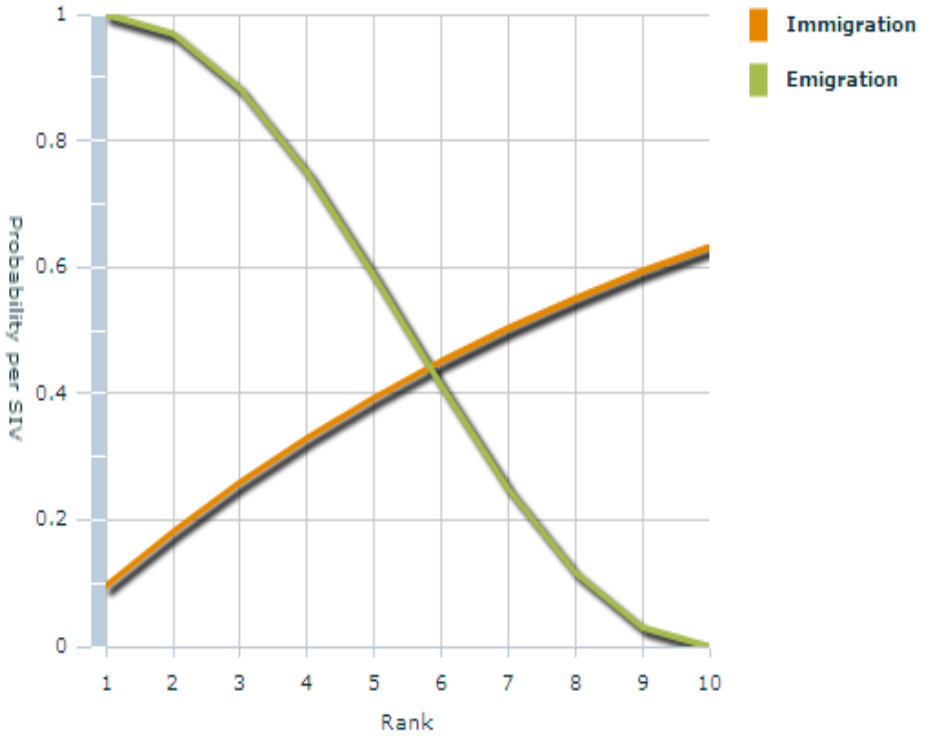
μ,λ
μ+λ

Basic Genetic Algorithm | Biogeography-Based Optimization | Opposition-Based Learning

Immigration Probability Function:

Sine [-pi/2,pi/2]

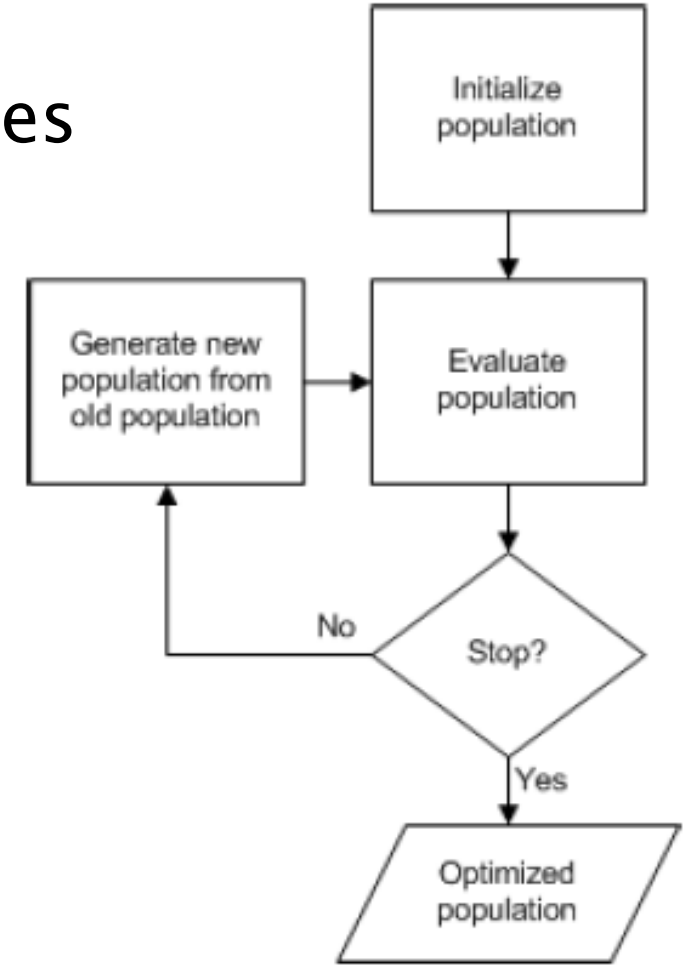
- Relative Cost/Fitness
- Exponential
- Sine [-pi/2,pi/2]
- Hyperbolic Tangent [-pi,pi]
- Step at N/2



Start | Run Continuously | Step Delay: | Step | Stop | Generation 0 of 10 | Settings: Save Load

Software Design

- Uses bit strings for genotypes
- Converts to decimal to calculate objective values
- Populations are randomly initialized
- Parameters and algorithms can be changed between generations



Usage

- Meant to be easy and self-explanatory
- Mouse-over tool tips for each option

1. Select simulation options
2. Start
3. Step or run continuously
4. Stop (manual or when conditions are met)

Extendibility

Written in a loose implementation of the Model-View-Controller (MVC) framework

- Model
  - Global variables and common routines
- View
  - Graphical interfaces (“panes,” algorithm options)
- Controller
  - Simulation routines

Extendibility

- Example:
  Add a new objective function

1. Add ActionScript code entry to Controller
2. Add MXML code entry to drop-down list in View

```
private function evalPop(pop:ArrayCollection):void
{
    for (var i:uint = 0; i < pop.length; i++)
    {
        var j:uint;
        var k:uint;
        var sum1:Number = 0;
        var sum2:Number = 0;
        switch(objFunc.selectedItem)
        {
            case "De Jong 1":
                for (j = 0; j < numChroms.value; j++)
                {
                    sum1 += Math.pow(pop[i].chroms[j].phenotype,2);
                }
                pop[i].objValue = sum1;
                maxObjFunc = false;
                break;
            case "Rotated hyper-ellipsoid":
                for (j = 0; j < numChroms.value; j++)
                {
                    sum1 = 0;
                    for (k = 0; k <= j; k++)
                    {
                        sum1 += pop[i].chroms[k].phenotype;
                    }
                    sum2 += Math.pow(sum1, 2);
                }
                pop[i].objValue = sum2;
                maxObjFunc = false;
                break;
                ...
}
```

Extendibility

- Example: Add a new algorithm
  - Create a new View for the algorithm control panel interface
  - Add *evolve* routine to new View which contains the logic to transform an old population to a new one
  - Add new View to Control Panel interface:

```
<mx:TabNavigator id="algSel" width="100%" height="100%">
    <algs:BasicGA label="Basic Genetic Algorithm" />
    <algs:BBO label="Biogeography-Based Optimization" />
    <algs:OBL label="Opposition-Based Learning" />
</mx:TabNavigator>
```

Extendibility

- Future Extensions
  - More options, objective functions, and algorithms
  - Quaternary, etc., genotype strings
  - Continuous genotypes
  - Other graphing methods
    - Comparing simulations
  - Remote Control of Simulation
    - Use the EA Sandbox interface to control and monitor a simulation via the web
    - Simulation would run on better suited tool for intense computation (rather than browser as in EA Sandbox)

Conclusion

- The EA Sandbox was created as a visual learning tool for evolutionary algorithms.
- A basic set of parameters, problem functions, and algorithms are included.
- Simulations can be run, the viewed in real time, and compared to one another.
- Users can quickly test different options to explore their effects, as well as add their own options.
- Application, source code, and documentation is available at http://embeddedlab.csuohio.edu/BBO