SGALAB

Evolution with the world

SxLAB Family Member

What is Genetic Algorithms ?

→→→ Genetic Algorithms(GAs) Brief
& GAs Toolboxs for Matlab introduction

Dr Leo Chen

leo.chen@ieee.org

22/Oct/2020

https://github.com/SGALAB/What-is-Genetic-Algorithms

Contents

Contents

What is Genetic Algorithms ?

→ Genetic Algorithms(*GAs*) was formally introduced in the United States in the 1970s by John Holland at University of Michigan.

(John Holland, from the University of Michigan began his work on genetic algorithms at the beginning of the 60s. A first achievement was the publication of Adaptation in Natural and Artificial System in 1975.)

"Genetic Algorithms are based on a biological metaphor: They view learning as a competition among a population of evolving candidate problem solutions. A 'fitness' function evaluates each solution to decide whether it will contribute to the next generation of solutions. Then, through operations analogous to gene transfer in sexual reproduction, the algorithm creates a new population of candidate solutions."

--From *Artificial Intelligence, Structures and Strategies for Complex Problem Solving*, Fourth Edition, at page 471. Luger, George F. 2002. Harlow, England: Addison-Wesley.

LSA UNIVERSITY OF MICHIGAN

DEPARTMENT OF Psychology

CONTACT US | MAP | WELCOME

HOME | GRADUATE PROGRAM | UNDERGRADUATE PROGRAM | PROGRAM AREAS | PEOPLE

HOME : PEOPLE : DIRECTORY : FACULTY PROFILE — JOHN HOLLAND

DIRECTORY
FACULTY RESEARCH INTERESTS
LABS
FACULTY POSITIONS
JOB OPENINGS
FACULTY MEDIA CONTACTS
ROOM SCHEDULES

QUICKLINKS

faculty in the news
alumni & friends

FACULTY PROFILE — JOHN HOLLAND

http://www.lsa.umich.edu/psych/people/directory/profiles/faculty/?uniquename=jholland

Professor of Psychology and Electrical Engineering & Computer Science
Ph.D. University of Michigan
Area: Cognition & Perception

Contact Information

Office: 1255 East Hall
Phone: 763-3648
Email: jholland@umich.edu

Research and Teaching Interests

Study of cognitive processes and complex adaptive systems in general, using mathematical models and computer simulation.

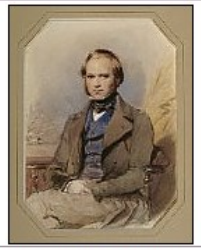
Related Links

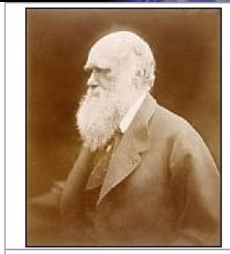
- Dept. of Electrical Engineering & Computer Science
- EECS Faculty Profile

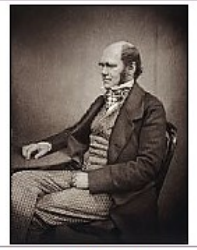
http://www.lsa.umich.edu/psych/people/directory/profiles/faculty/?uniquename=jholland

What is Genetic Algorithms?

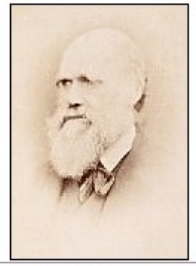
Genetic Algorithms is inspired by the theory of Charles Darwin (1809 - 1882) about Evolution.


Richmond - Charles Darwin J980057

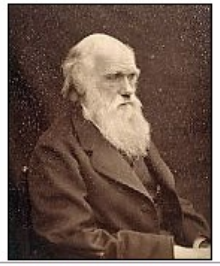

Charles Darwin K980352

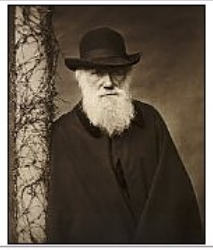

Charles Darwin K970247


Charles Darwin on the verandah at Down House K970226

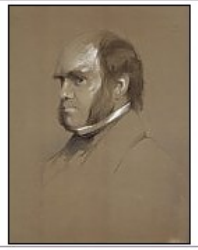

Charles Darwin K970234

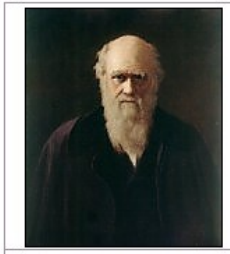

Charles Darwin K970239


Charles Darwin K970215


Charles Darwin K980123


Charles Darwin on horseback K970217

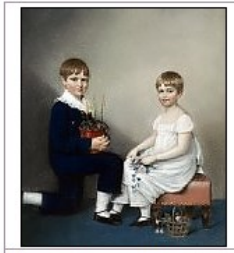

Laurence - Charles Darwin J970202

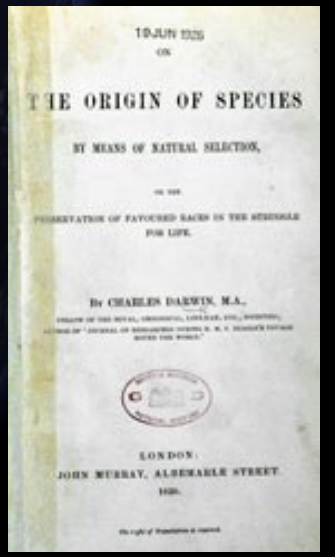

Reilly - Charles Darwin J970164


Evstafieff - Darwin in his Study J970178


Charles Darwin and his son N990002


Charles Darwin K970235


Sharples - Charles Darwin (aged six) and Catherine K971925



www.bbc.co.uk/history/historic_figures/darwin_charles.shtml
www.bbc.co.uk/darwin/
www.darwin200.org/
www.amnh.org/exhibitions/darwin/
www.englishheritageimages.com/

Charles Darwin

- 01 - Charles Darwin A Short Biography
- 02 - The Truth About Charles Darwin _ Earth Unplugged
- 03 - BBC News-Theory of Evolution How did Darwin come up with it
- 04 - Richard Dawkins - The Genius of Charles Darwin - Part 1_ Life, Darwin & Everything
- 05 - What is Natural Selection

Contents

History of Genetic Algorithms

As early as 1962, John Holland's work on adaptive systems laid the foundation for later developments; most notably, Holland was also the first to explicitly propose crossover and other recombination operators. However, the seminal work in the field of genetic algorithms came in 1975, with the publication of the book *Adaptation in Natural and Artificial Systems*. Building on earlier research and papers both by Holland himself and by colleagues at the University of Michigan, this book was the first to systematically and rigorously present the concept of adaptive digital systems using mutation, selection and crossover, simulating processes of biological evolution, as a problem-solving strategy. The book also attempted to put genetic algorithms on a firm theoretical footing by introducing the notion of schemata (Mitchell 1996, p.3; Haupt and Haupt 1998, p.147). That same year, Kenneth De Jong's important dissertation established the potential of GAs by showing that they could perform well on a wide variety of test functions, including noisy, discontinuous, and multimodal search landscapes (Goldberg 1989, p.107).

Evolutionary Computation

→ Idea of evolutionary computing was introduced in the 1960s by I. Rechenberg in his work "Evolution Strategies" (ES). His idea was then developed by other researchers.

→ Genetic Algorithms (GAs) were invented by John Holland and developed by him and his students and colleagues. This lead to Holland's book "Adaption in Natural and Artificial Systems" published in 1975.

→ In 1992 John Koza has used genetic algorithm to evolve programs to perform certain tasks. He called his method "Genetic programming" (GP). LISP programs were used, because programs in this language can expressed in the form of a "parse tree", which is the object the GA works on.

→ Evolutionary Programming" (EP), originally conceived by Lawrence J. Fogel in 1960, is a stochastic OPTIMIZATION strategy similar to Genetic Algorithms , but instead places emphasis on the behavioral linkage between PARENTs and their OFFSPRING, rather than seeking to emulate specific GENETIC OPERATORs as observed in nature. Evolutionary programming is similar to EVOLUTION STRATEGIEs, although the two approaches developed independently

- P.s. Briefly, Computational Intelligence = Genetic Algorithms + Artificial Neural Network + Fuzzy Logic sysem

Contents

Genetic Algorithms Basic workflow

Max : $F(x_i) = \{ f_1(x_i), f_2(x_i), \cdots, f_k(x_i) \}$

Subject to : $g_j(x_i) \le 0$

$i = 1,2,\cdots,m$

$j = 1,2,\cdots,n$

(1) Object design based on engineering application

Start

(2) Variable/Fitness function design

initial solution

x(1) x(2) x(n)

population

(10.4554)(74545.4)...(5.4345)

(11.0230)(75690.3)...(5.2344)

(9.99234)(74545.9)...(5.2322)

(10.4544)(74544.3)⋯(5.123)

Fitness/Cost Evaluation

No

En-coding

x(1) coding_length(1) x(2) coding_length(2) x(n) coding_length(n)

population

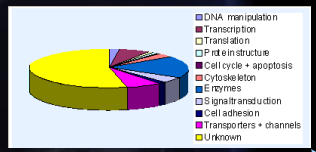
1000101010101010100...0101

0101000001010010 1...1101

0100010100010101 0...1110

0010100010101010 1...1000

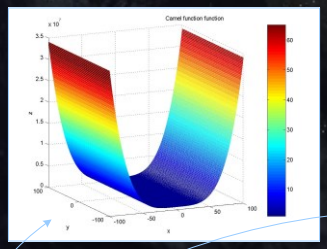
Selection

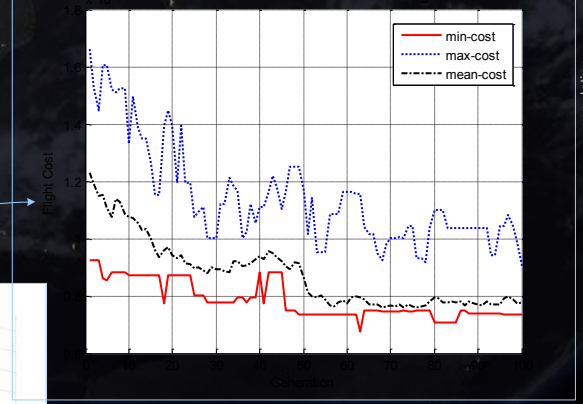


Crossover

Yes

De-coding





x(1) coding_length(1) x(2) coding_length(2) x(n) coding_length(n)

population

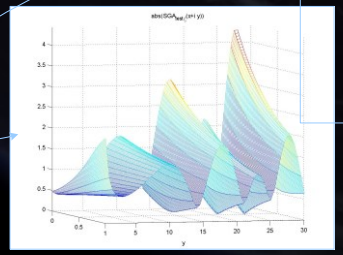
1000101010100**00101...1101**

01010000010110100...0101

0010100010**101010...1110**

01000101000010101...1000

Mutation



Stop

Result & Analysis

x(1) coding_length(1) x(2) coding_length(2) x(n) coding_length(n)

population

1000101011100**00101...1101**

01011000100100100...0101

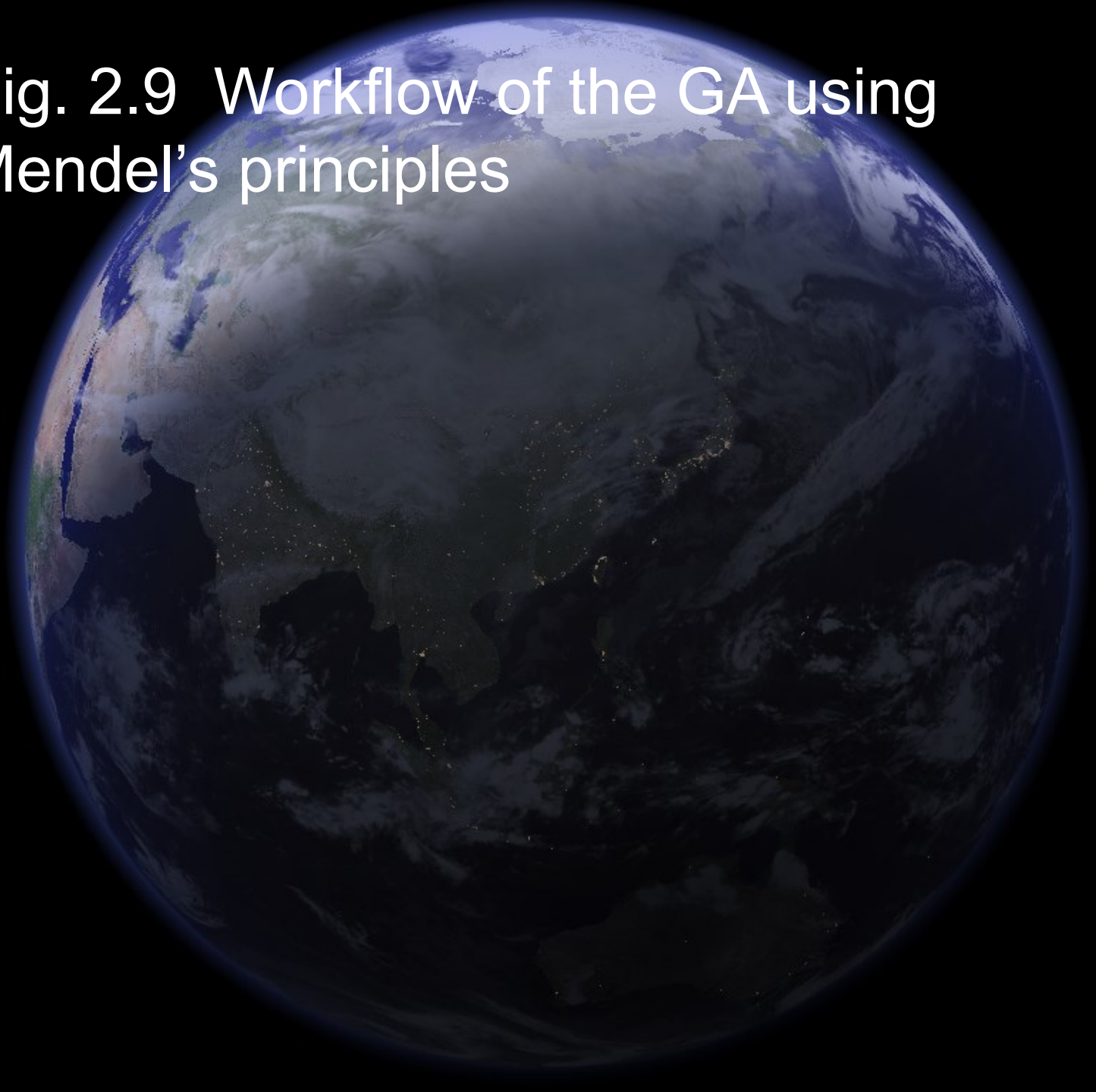
0010100001**101010...110**0

01000101010100100**1**...1000

Procedure genetic algorithm

```
Begin (1)
        t = 0 ;
        Initialize P(t);
        Evaluate P(t);
        While ( Not termination-condition) do
          Begin (2)
              { t = t+1;
                Select P(t) from P(t-1)
                Crossover P(t);
                Mutation P(t);
                Evaluate P(t); }
          End (2)
End (1)
```

- Fig. 2.9 Workflow of the GA using Mendel's principles



Contents

Genetic Algorithms tools for Matlab (I)-- GAOT

The *Genetic Algorithm Optimization Toolbox* (GAOT) for Matlab

GAOT implements simulated evolution in the Matlab environment using both binary and real representations. Ordered base representation has also been added to the toolbox. This implementation is very flexible in the genetic operators, selection functions, termination functions as well as the evaluation functions that can be used. The implementation is described in a technical paper. The paper can be referenced as follows:

College of Engineering,
North Carolina State University,
Raleigh, NC 27695

Where can we find GOAT：

http://www.ie.ncsu.edu/mirage/GAToolBox/gaot/

The Genetic Algorithm Optimization Toolbox (GAOT) for Matlab 5

GAOT implements simulated evolution in the Matlab environment using both binary and real representations. Ordered base representation has also been added to the toolbox. This implementation is very flexible in the genetic operators, selection functions, termination functions as well as the evaluation functions that can be used. The implementation is described in a technical paper. The paper can be referenced as follows

"*A Genetic Algorithm for Function Optimization: A Matlab Implementation*" by Chris Houck, Jeff Joines, and Mike Kay, NCSU-IE TR 95-09, 1995.

The entire toolbox can be download either as a compressed tar archive (GAOT.tar.gz) or a ZIP file (GAOT.zip). This includes the postscript and dvi versions of the companion paper.

The GA toolbox can be also obtained via anonymous ftp from the following directory: ftp://ftp.eos.ncsu.edu/pub/simul/GAOT as well as other GA related papers.

Contents

Genetic Algorithms tools for Matlab (II)--GATBX

Evolutionary Computation Research Group: Genetic Algorithm Toolbox

The ***Genetic Algorithm Toolbox*** for MATLAB ® was developed at the Department of Automatic Control and Systems Engineering of The University of Sheffield, UK, in order to make GA's accessible to the control engineer within the framework of a existing computer-aided control system design package. The toolbox was written with the support of a UK SERC grant, and the final version (v1.2) was completed in 1994.

The Toolbox was originally developed for MATLAB v4.2 but has also been successfully used with subsequent versions up to and including MATLAB 7.

For a more detailed introduction to the capabilities and use of the GA Toolbox, please refer to the introductory papers and user's guide detailed below and available for download opposite.

Where can find GATBX:
http://www.shef.ac.uk/acse/research/ecrg/getgat.html

The University Of Sheffield.

Automatic Control and Systems Engineering

You are here: Home / Departments / ACSE / Research / Evolutionary Computation Research Group

ACSE Home
Prospective Undergraduates
Prospective Postgraduates
Research
- Themes and Centres
- Publications
- Contracts
- Seminars

Current Students
Staff
Schools Outreach
About the Department
Contact Us

Obtaining the toolbox

The Genetic Algorithm Toolbox for MATLAB ® was developed at the Department of Automatic Control and Systems Engineering of The University of Sheffield, UK. It was originally developed for MATLAB v4.2, but has also been successfully used with subsequent versions up to and including MATLAB 7.

The GA Toolbox is copyright the original authors and The University of Sheffield, and is published here under the GNU General Public License.

We would be interested to hear of your experiences with, criticisms of, and enhancements to the GA Toolbox. Please direct all such correspondence to ga-toolbox@acse.sheffield.ac.uk.

The GA Toolbox v1.2 is available in two packages: one suitable for DOS/Windows systems, and one suitable for Unix systems.

To download a copy of the Toolbox use the form below

Genetic Algorithm TOOLBOX

For Use with MATLAB®

Andrew Chipperfield
Peter Fleming
Hartmut Pohlheim
Carlos Fonseca

Department of
AUTOMATIC CONTROL and SYSTEMS ENGINEERING
UNIVERSITY of SHEFFIELD

Version 1.2

User's Guide

Contents

Genetic Algorithms tools for Matlab(III) – GADS (Matlab 7.xx)

The ***Genetic Algorithm and Direct Search*** Toolbox is a collection of functions that extend the capabilities of the Optimization Toolbox and the MATLAB® numeric computing environment. The Genetic Algorithm and Direct Search Toolbox includes routines for solving optimization problems using:

Genetic algorithm
Direct search

These algorithms enable you to solve a variety of optimization problems that lie outside the scope of the standard Optimization Toolbox.

All the toolbox functions are MATLAB M-files, made up of MATLAB statements that implement specialized optimization algorithms. You can view the MATLAB code for these functions using the statement

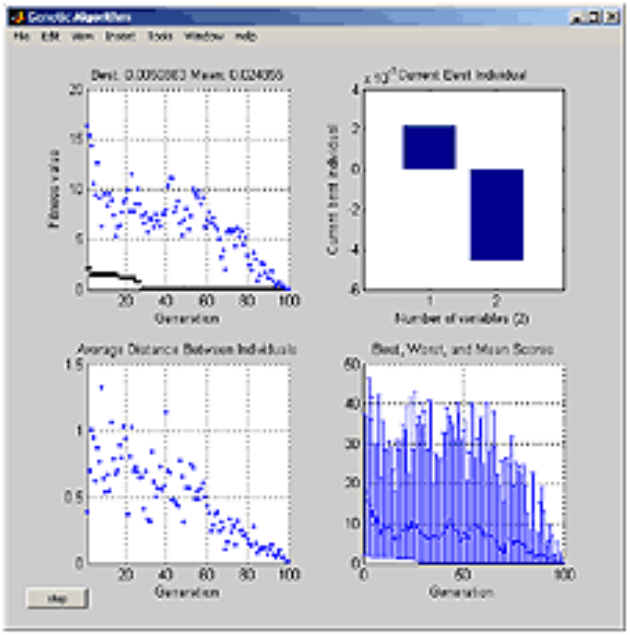
You can extend the capabilities of the Genetic Algorithm and Direct Search Toolbox by writing your own M-files, or by using the toolbox in combination with other toolboxes, or with MATLAB or Simulink®.

Keyin “gatool “ in command window

http://www.mathworks.com/products/gads/
http://www.mathworks.com/access/helpdesk/help/toolbox/gads/

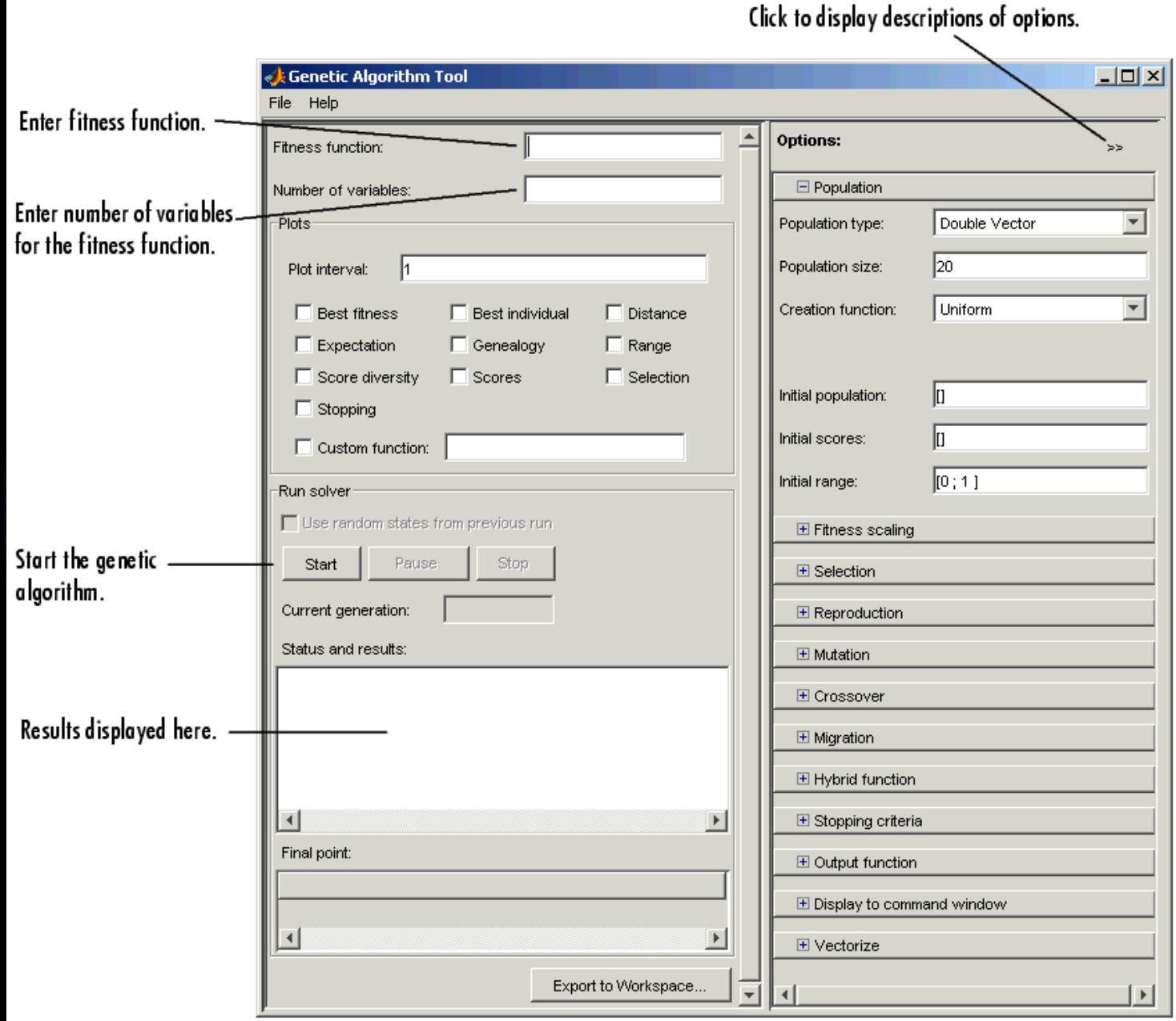
Genetic Algorithm and Direct Search Toolbox 1.0.3

Solve optimization problems using genetic and direct search algorithms



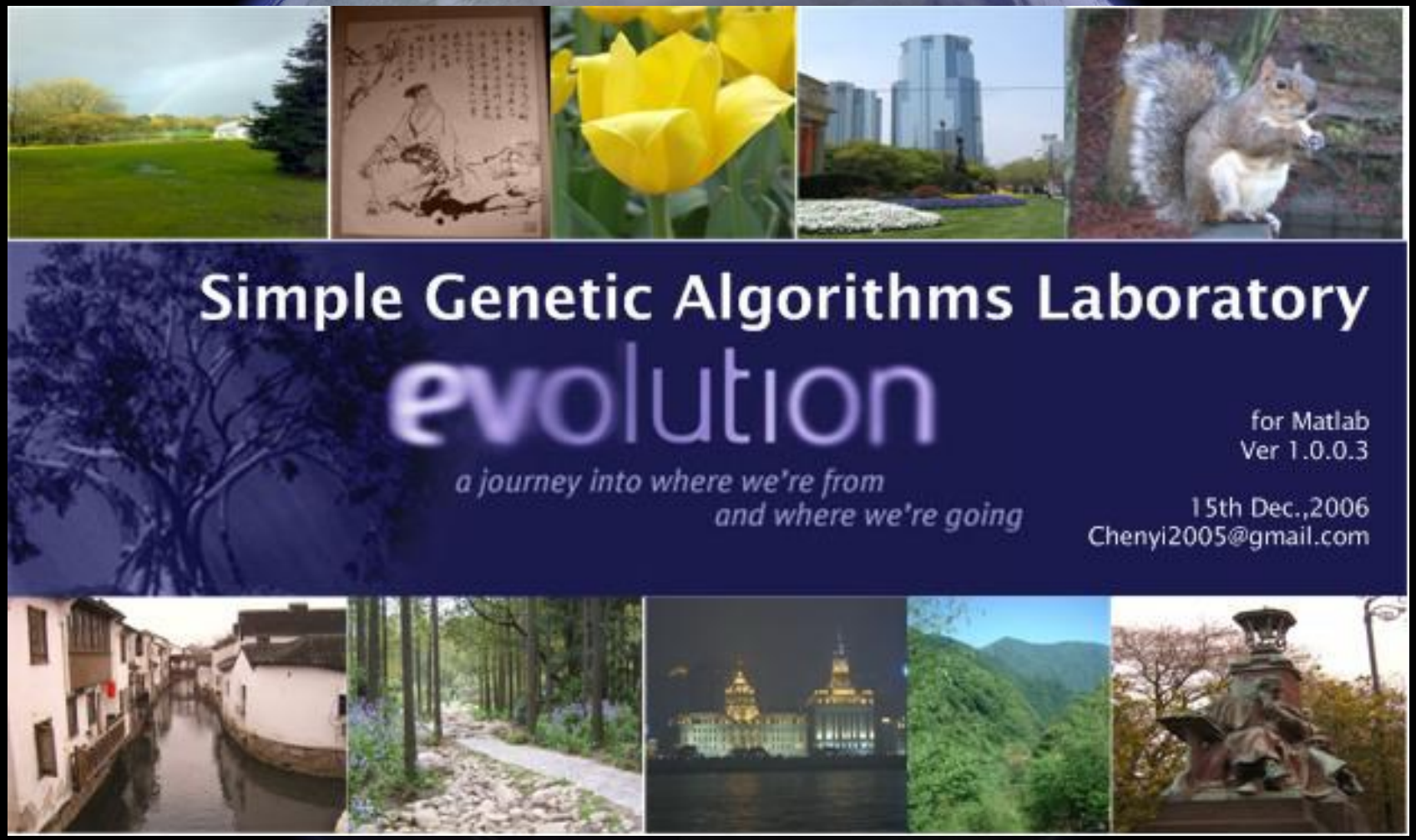
The Genetic Algorithm and Direct Search Toolbox extends the optimization capabilities in MATLAB and the Optimization Toolbox with tools for using the genetic and direct search algorithms. You can use these algorithms for problems that are difficult to solve with traditional optimization techniques, including problems that are not well defined or are difficult to model mathematically. You can also use them when computation of the objective function is discontinuous, highly nonlinear, stochastic, or has unreliable or undefined derivatives.

- Overview and Key Features
- Graphical Interface and Command-Line Functions
- Genetic Algorithm Tools
- Direct Search Algorithm Tools
- Using Other Functions and Solvers
- Displaying, Monitoring, and Ouputting Results

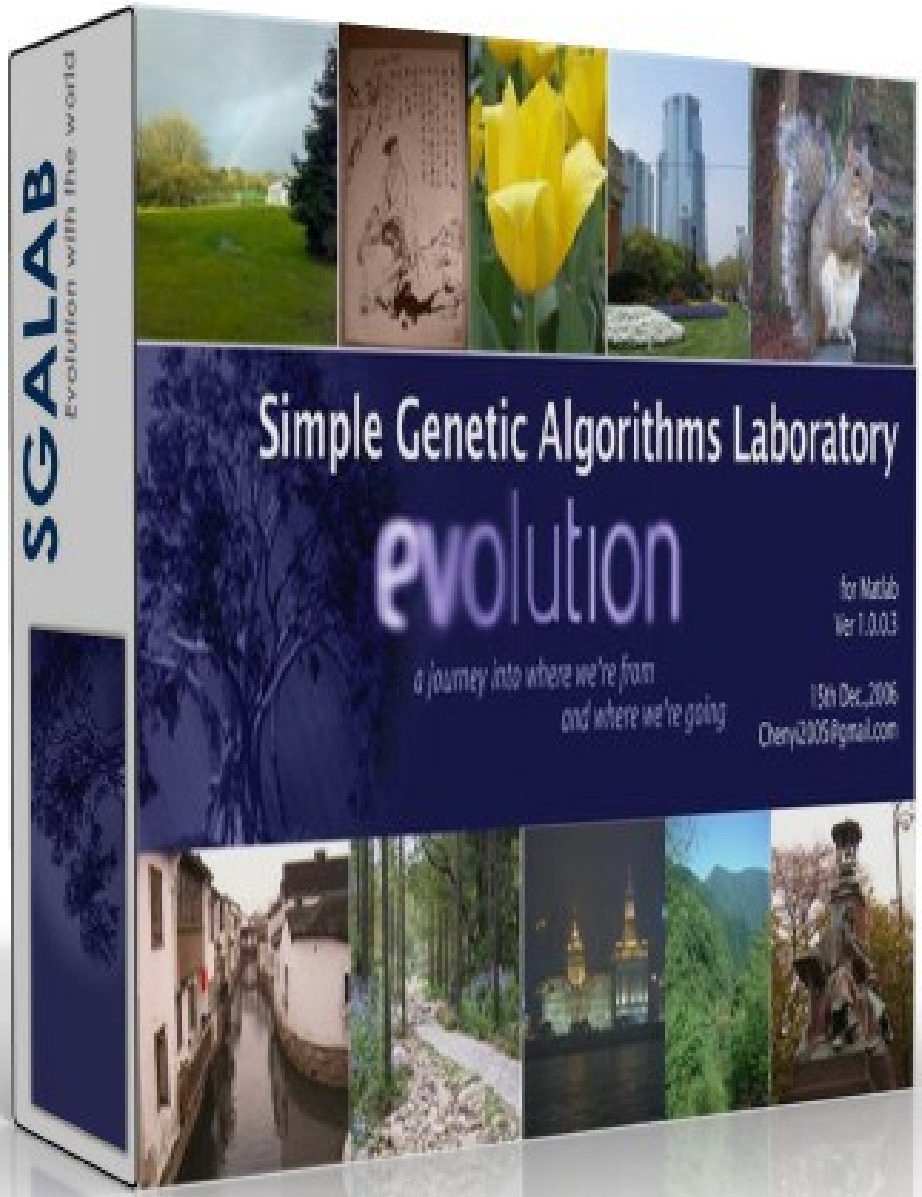

Click to display descriptions of options.
Enter fitness function.
Enter number of variables for the fitness function.
Start the genetic algorithm.
Results displayed here.
Genetic Algorithm Tool
File Help
Fitness function:
Number of variables:
Plots
Plot interval: 1
Best fitness
Best individual
Distance
Expectation
Genealogy
Range
Score diversity
Scores
Selection
Stopping
Custom function:
Run solver
Use random states from previous run
Start
Pause
Stop
Current generation:
Status and results:
Final point:
Export to Workspace...
Options:
>>
Population
Population type: Double Vector
Population size: 20
Creation function: Uniform
Initial population: []
Initial scores: []
Initial range: [0 ; 1]
Fitness scaling
Selection
Reproduction
Mutation
Crossover
Migration
Hybrid function
Stopping criteria
Output function
Display to command window
Vectorize

Contents

Genetic Algorithms tools for Matlab(IV) – SGALAB



http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=5882


SGALAB
evolution with the world
Simple Genetic Algorithms Laboratory
evolution
for Matlab
Ver 1.0.0.3
a journey into where we're from
and where we're going
15th Dec, 2006
Cheryl2005@gmail.com

How to get this toolbox :

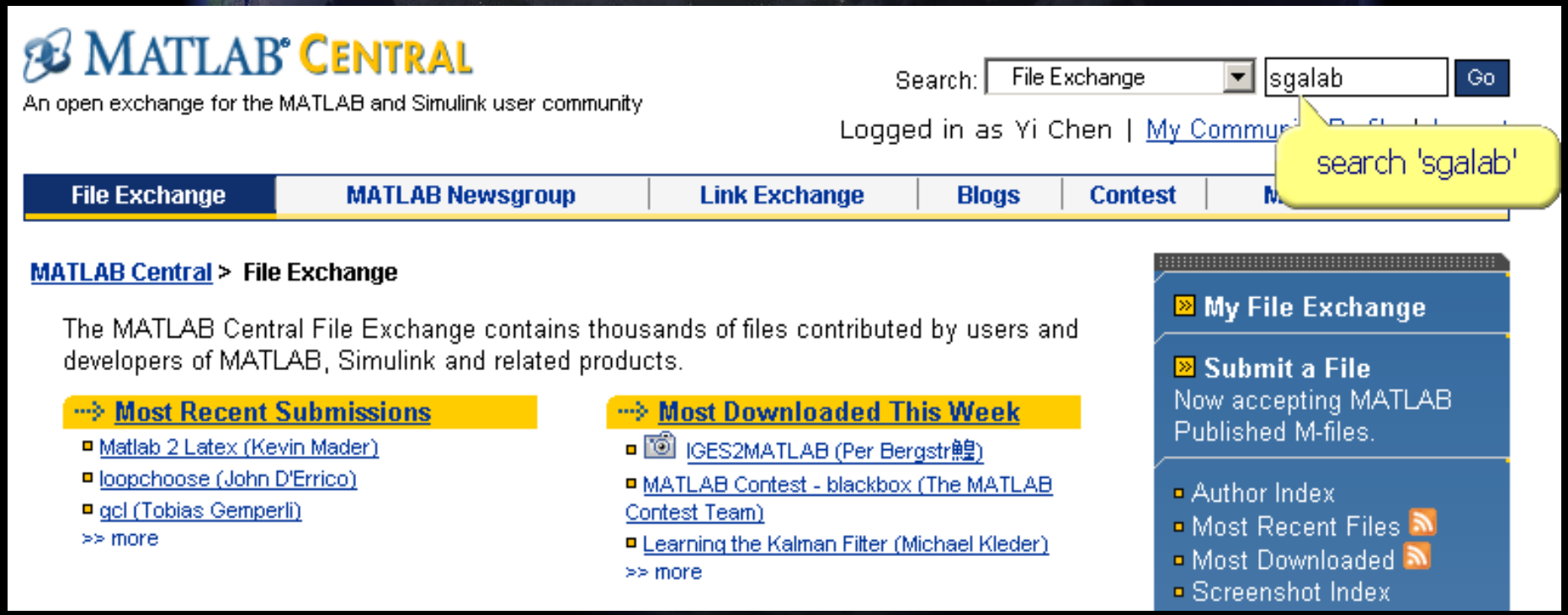
Step 1:

http://www.mathworks.com/

Step 2:
Get into “File Exchange”

Genetic Algorithms tools for Matlab(IV) – SGALAB

Step 3:
Search ‘SGALAB’



MATLAB CENTRAL

An open exchange for the MATLAB and Simulink user community

Search: File Exchange | Go

Logged in as Yi Chen | My Community Profile | Logout

File Exchange | MATLAB Newsgroup | Link Exchange | Blogs | Contest | MathWorks.com

MATLAB Central > File Exchange > Search Results

Search Results for "sgalab"

Click on column heading to sort by column

Downloads (last 30 days)	Title	Submitted	Rating (5=best)
135	SGALAB1003 Beta5 (Inc. NSGA-II) Genetic Algorithms Toolbox Author: Yi Chen Category: Medical	2004-09-15	4.45 27 reviews
80	Semi-active Control of Skyhook for 1/4 Suspension System Semi-active control of skyhook Author: Yi Chen Category: Automotive	2006-05-19	4.67 3 reviews

My File Exchange

Submit a File
Now accepting MATLAB Published M-files.

- Author Index
- Most Recent Files
- Most Downloaded
- Screenshot Index
- Most Recent Comments
- Files By Product
- Metrics and Reports

E-mail this page to a colleague

SGALAB1003 Beta5 (Inc. NSGA-II)

Download Now: .zip

Rating: ★★★★½ 27 reviews **Review this file**

Author: Yi Chen

Summary: Genetic Algorithms Toolbox

MATLAB Release: R14SP2

Required Products: Communications Toolbox

View fullsize image

Description: what's new for SGALAB1003Beta5

1) NSGA-II (SGALAB_demo_MO_NSGAII.m)

2) Pareto front plot & show all Multi-Object Methods in SGALAB
(SGALAB_demo_MO_showall.m)

3) documents updated

4) plot bugs fixed for VEGA,MOGA,NSGA,NPGA

Run 'SGALAB_demo_*.m' to see the demos, and to modify the 'SGALAB_demo_*.m' as your main function for your applications

%%%%%%%%%%%%%%%%%%%%

SGALAB 1003 Beta 4&4.0.0.1

My File Exchange

Submit a File
Now accepting MATLAB Published M-files.

- Author Index
- Most Recent Files
- Most Downloaded
- Screenshot Index
- Most Recent Comments
- Files By Product
- Metrics and Reports

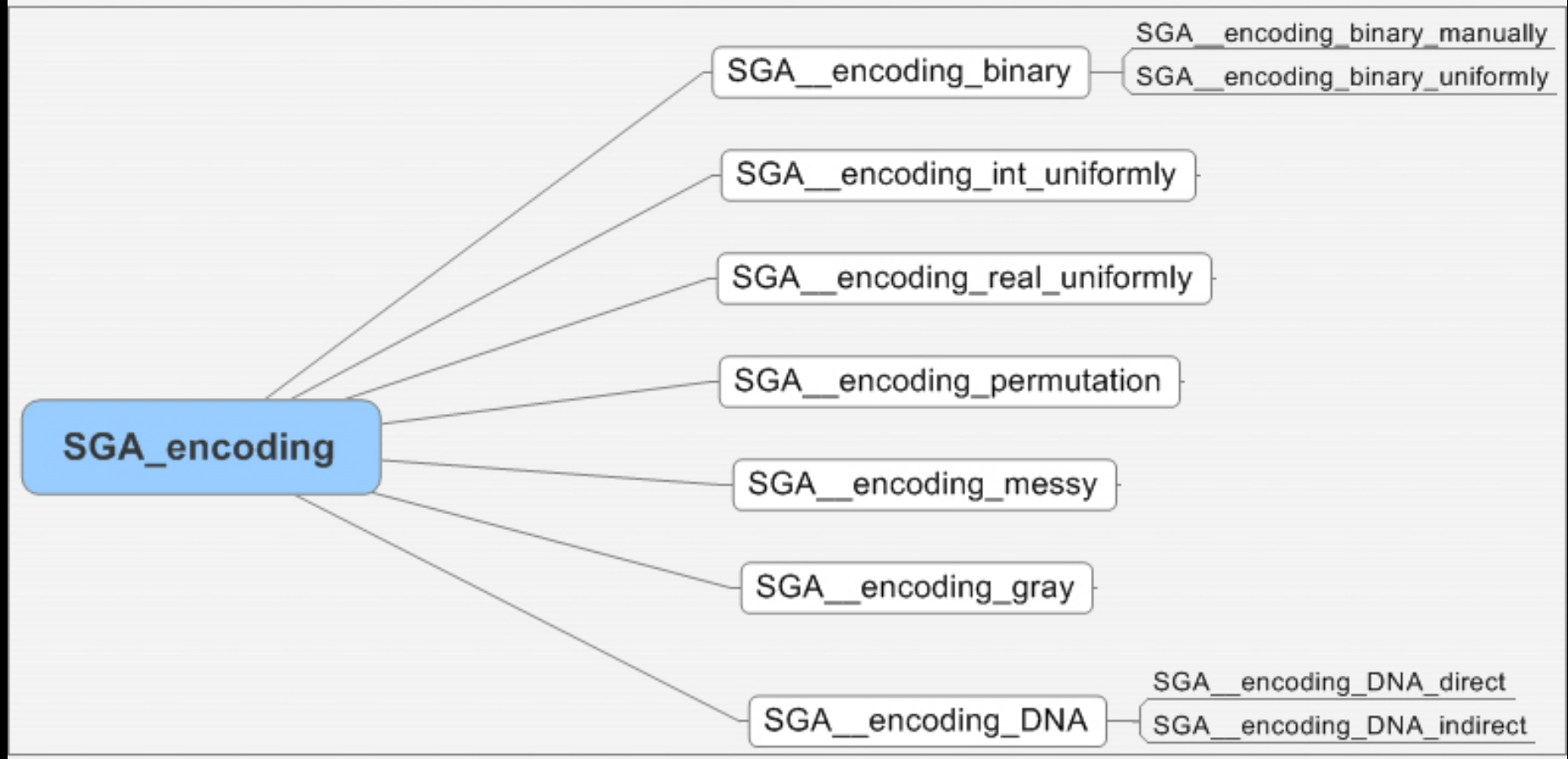
- Download this File
- Update this File
- Remove this File

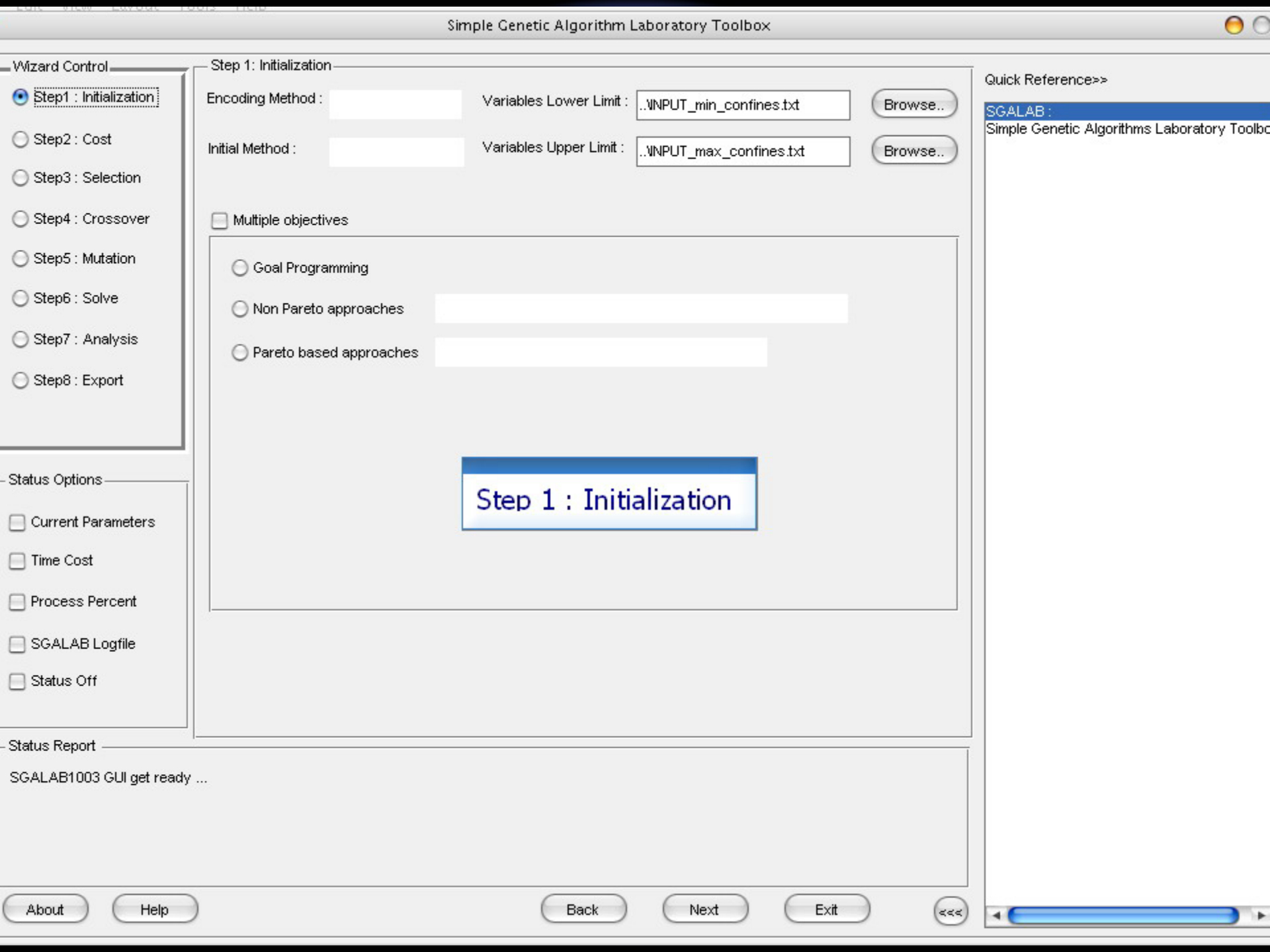
Notify me when changes are made to this file

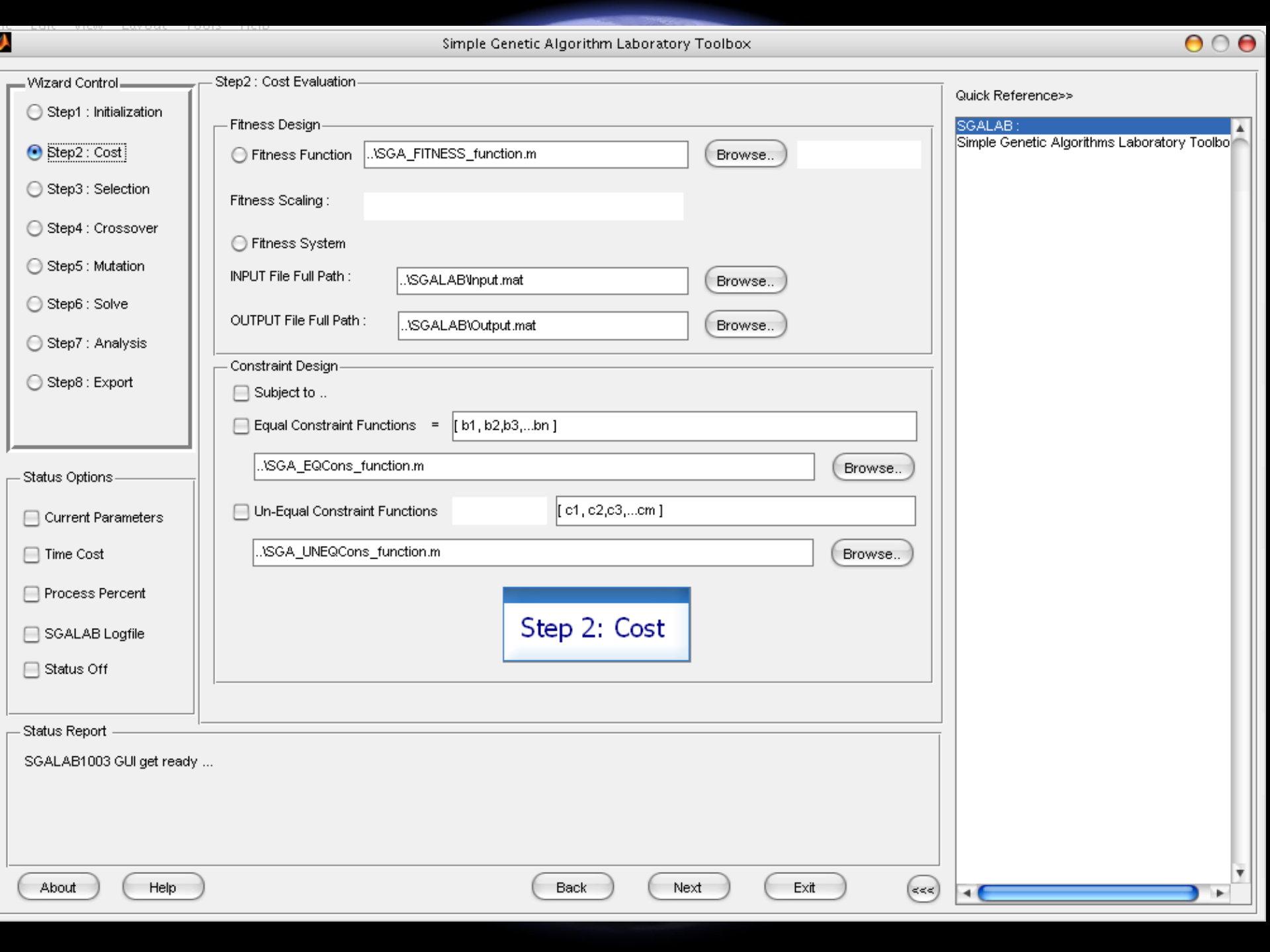
E-mail: Notify Me

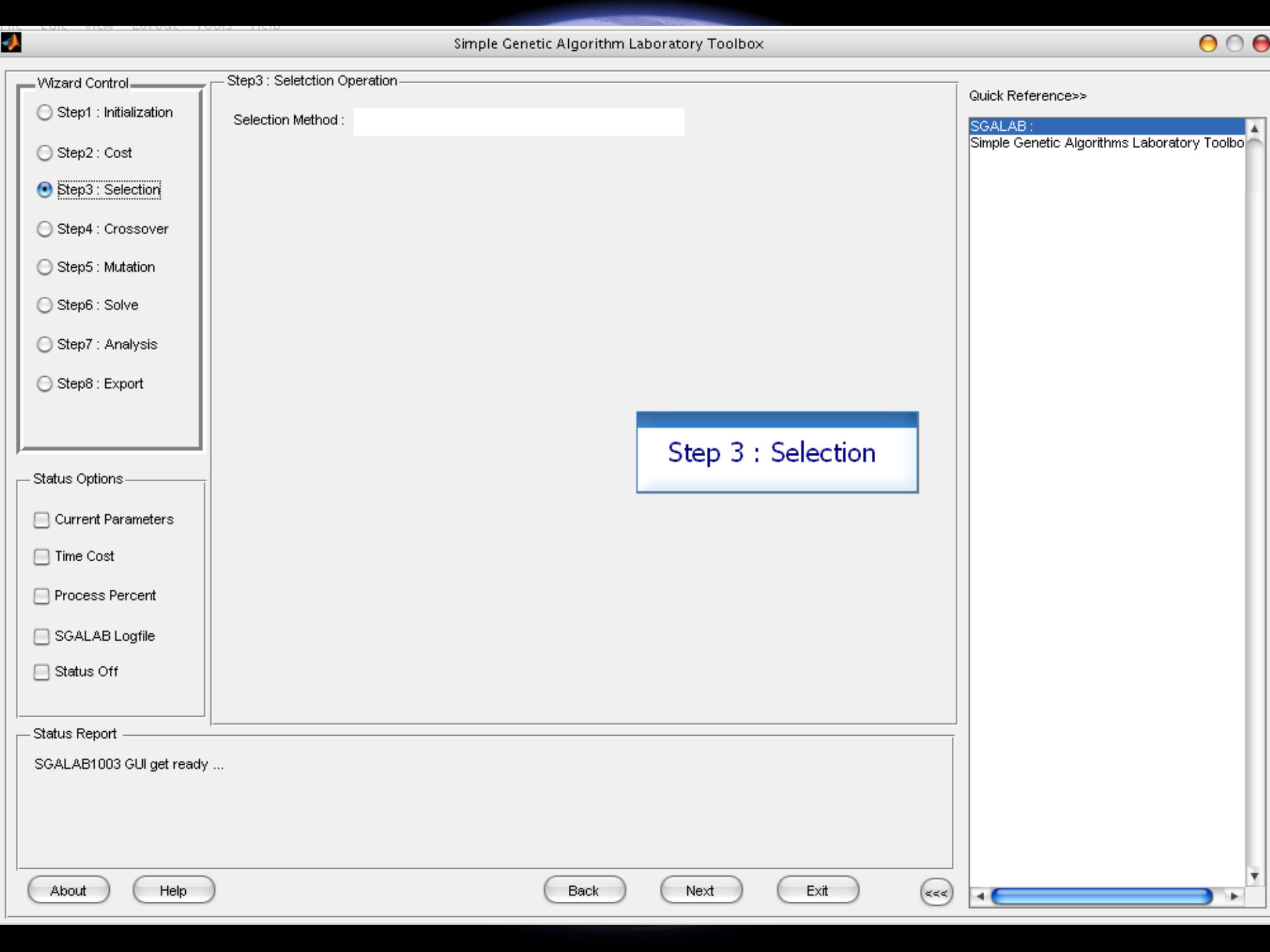
How do I unsubscribe?

E-mail this page to a colleague


SGA_encoding
SGA__encoding_binary
SGA__encoding_binary_manually
SGA__encoding_binary_uniformly
SGA__encoding_int_uniformly
SGA__encoding_real_uniformly
SGA__encoding_permutation
SGA__encoding_messy
SGA__encoding_gray
SGA__encoding_DNA
SGA__encoding_DNA_direct
SGA__encoding_DNA_indirect


Simple Genetic Algorithm Laboratory Toolbox
Wizard Control
Step1 : Initialization
Step2 : Cost
Step3 : Selection
Step4 : Crossover
Step5 : Mutation
Step6 : Solve
Step7 : Analysis
Step8 : Export
Status Options
Current Parameters
Time Cost
Process Percent
SGALAB Logfile
Status Off
Step 1: Initialization
Encoding Method :
Initial Method :
Variables Lower Limit :
..\INPUT_min_confines.txt
Browse..
Variables Upper Limit :
..\INPUT_max_confines.txt
Browse..
Multiple objectives
Goal Programming
Non Pareto approaches
Pareto based approaches
Step 1 : Initialization
Quick Reference>>
SGALAB :
Simple Genetic Algorithms Laboratory Toolbo
Status Report
SGALAB1003 GUI get ready ...
About
Help
Back
Next
Exit
<<<


Simple Genetic Algorithm Laboratory Toolbox
Wizard Control
Step1 : Initialization
Step2 : Cost
Step3 : Selection
Step4 : Crossover
Step5 : Mutation
Step6 : Solve
Step7 : Analysis
Step8 : Export
Status Options
Current Parameters
Time Cost
Process Percent
SGALAB Logfile
Status Off
Step2 : Cost Evaluation
Fitness Design
Fitness Function
..\SGA_FITNESS_function.m
Browse..
Fitness Scaling :
Fitness System
INPUT File Full Path :
..\SGALAB\Input.mat
Browse..
OUTPUT File Full Path :
..\SGALAB\Output.mat
Browse..
Constraint Design
Subject to ..
Equal Constraint Functions =
[b1, b2,b3,...bn]
..\SGA_EQCons_function.m
Browse..
Un-Equal Constraint Functions
[c1, c2,c3,...cm]
..\SGA_UNEQCons_function.m
Browse..
Step 2: Cost
Quick Reference>>
SGALAB :
Simple Genetic Algorithms Laboratory Toolbo
Status Report
SGALAB1003 GUI get ready ...
About
Help
Back
Next
Exit
<<<


Simple Genetic Algorithm Laboratory Toolbox
Wizard Control
Step1 : Initialization
Step2 : Cost
Step3 : Selection
Step4 : Crossover
Step5 : Mutation
Step6 : Solve
Step7 : Analysis
Step8 : Export
Step3 : Seletction Operation
Selection Method :
Step 3 : Selection
Quick Reference>>
SGALAB :
Simple Genetic Algorithms Laboratory Toolbo
Status Options
Current Parameters
Time Cost
Process Percent
SGALAB Logfile
Status Off
Status Report
SGALAB1003 GUI get ready ...
About
Help
Back
Next
Exit
<<<

Simple Genetic Algorithm Laboratory Toolbox

Wizard Control

- Step1 : Initialization
- Step2 : Cost
- Step3 : Selection
- Step4 : Crossover
- Step5 : Mutation
- Step6 : Solve
- Step7 : Analysis
- Step8 : Export

Status Options

- Current Parameters
- Time Cost
- Process Percent
- SGALAB Logfile
- Status Off

Step4 : Crossover Operation

Crossover Method :	
Crossove Probability :	0.9
Crossove Probability Steps :	0.01
Max Crossove Probability :	0.99
Elite Count :	2

Step 4 : Crossover

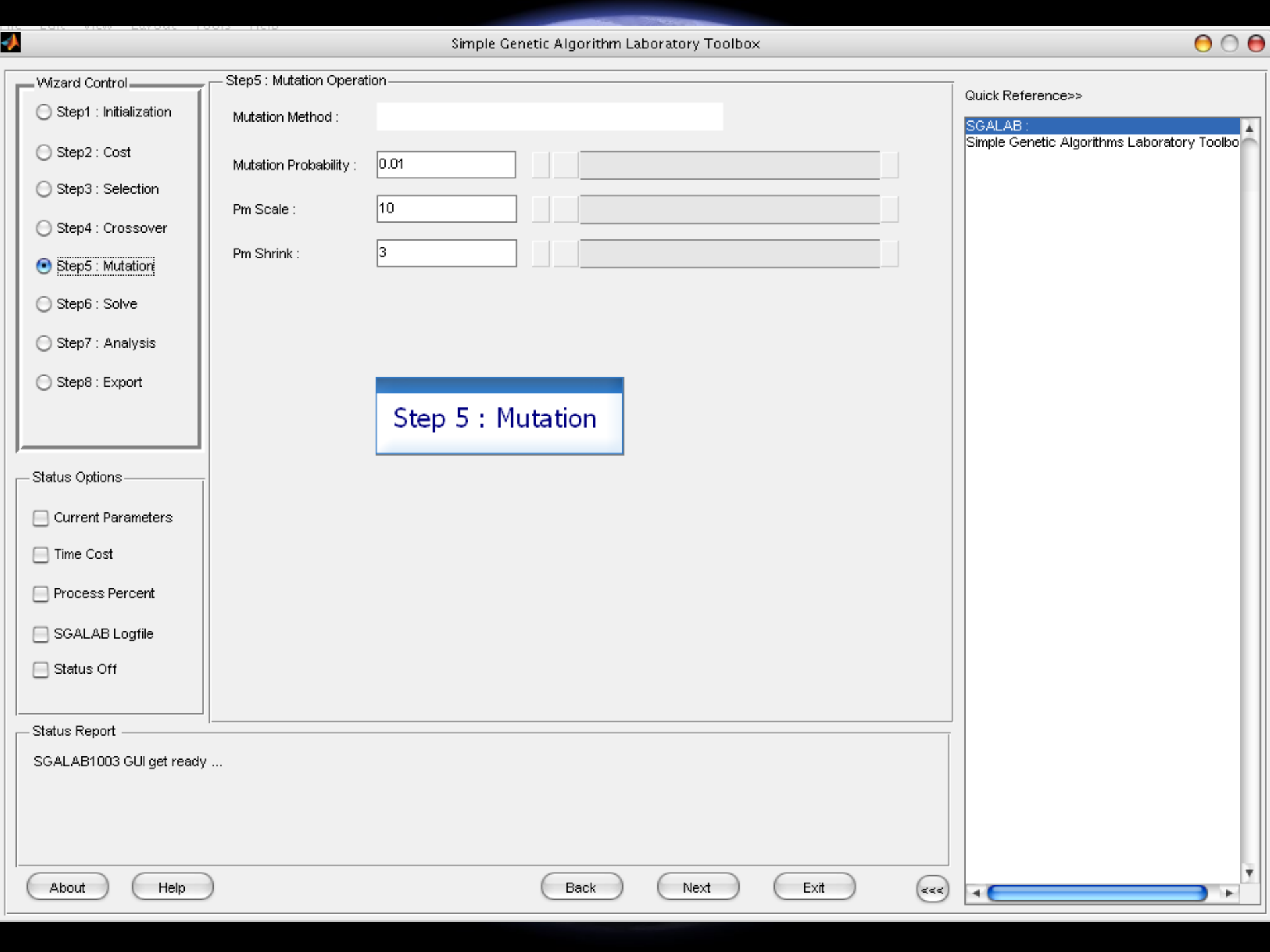
Quick Reference>>

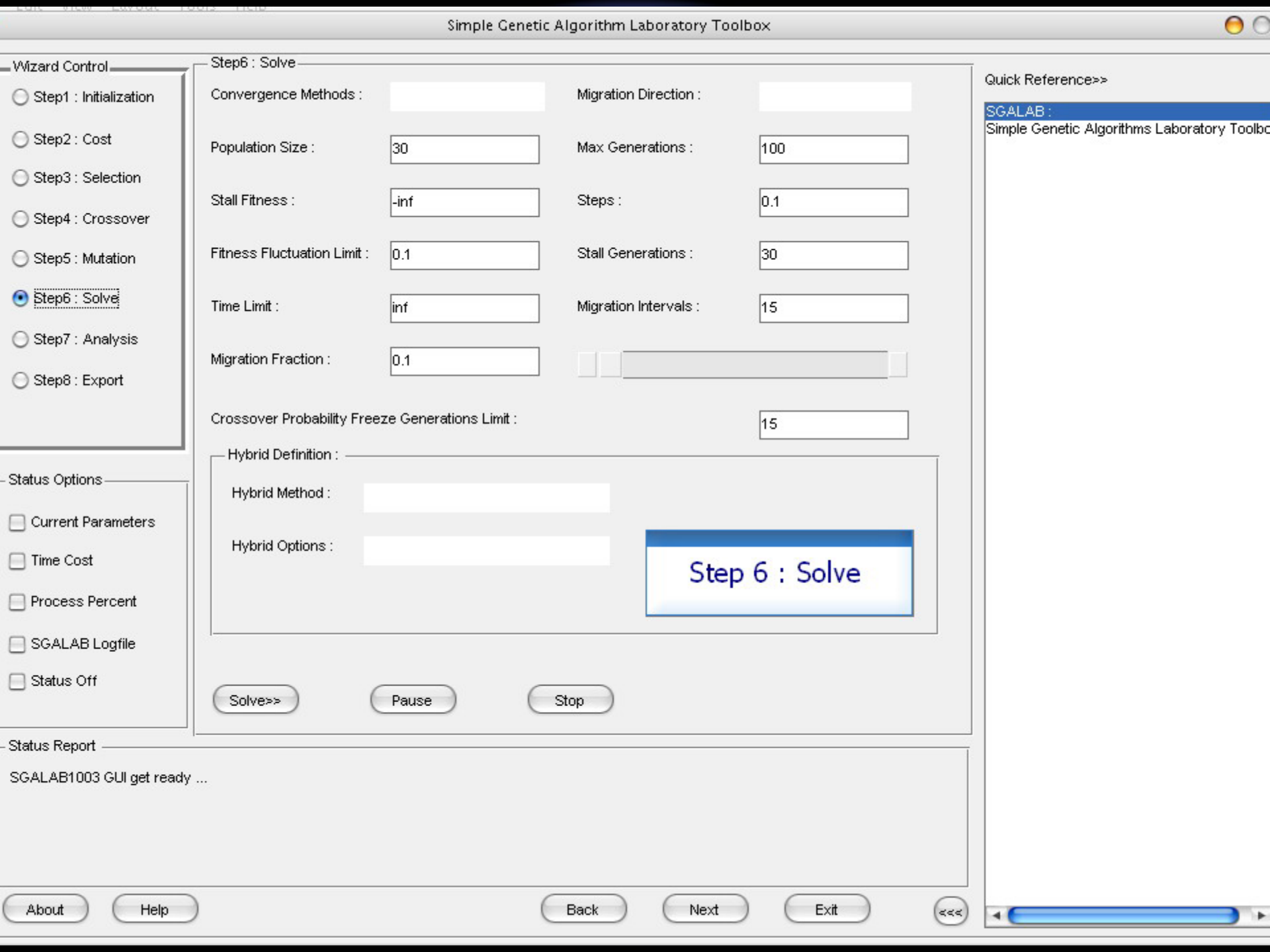
SGALAB :
Simple Genetic Algorithms Laboratory Toolbo

Status Report

SGALAB1003 GUI get ready ...

About | Help | Back | Next | Exit | <<<


Simple Genetic Algorithm Laboratory Toolbox
Wizard Control
Step1 : Initialization
Step2 : Cost
Step3 : Selection
Step4 : Crossover
Step5 : Mutation
Step6 : Solve
Step7 : Analysis
Step8 : Export
Status Options
Current Parameters
Time Cost
Process Percent
SGALAB Logfile
Status Off
Step5 : Mutation Operation
Mutation Method :
Mutation Probability :
0.01
Pm Scale :
10
Pm Shrink :
3
Step 5 : Mutation
Quick Reference>>
SGALAB :
Simple Genetic Algorithms Laboratory Toolbo
Status Report
SGALAB1003 GUI get ready ...
About
Help
Back
Next
Exit
<<<


Simple Genetic Algorithm Laboratory Toolbox
Wizard Control
Step1 : Initialization
Step2 : Cost
Step3 : Selection
Step4 : Crossover
Step5 : Mutation
Step6 : Solve
Step7 : Analysis
Step8 : Export
Status Options
Current Parameters
Time Cost
Process Percent
SGALAB Logfile
Status Off
Step6 : Solve
Convergence Methods :
Migration Direction :
Population Size :
30
Max Generations :
100
Stall Fitness :
-inf
Steps :
0.1
Fitness Fluctuation Limit :
0.1
Stall Generations :
30
Time Limit :
inf
Migration Intervals :
15
Migration Fraction :
0.1
Crossover Probability Freeze Generations Limit :
15
Hybrid Definition :
Hybrid Method :
Hybrid Options :
Step 6 : Solve
Solve>>
Pause
Stop
Status Report
SGALAB1003 GUI get ready ...
Quick Reference>>
SGALAB :
Simple Genetic Algorithms Laboratory Toolbo
About
Help
Back
Next
Exit
<<<

Simple Genetic Algorithm Laboratory Toolbox

Wizard Control

- Step1 : Initialization
- Step2 : Cost
- Step3 : Selection
- Step4 : Crossover
- Step5 : Mutation
- Step6 : Solve
- Step7 : Analysis
- Step8 : Export

Status Options

- Current Parameters
- Time Cost
- Process Percent
- SGALAB Logfile
- Status Off

Step7 : Data Analysis & Report

Analysis

Encoding Type:

Schema order | Defining Length | Walsh Ave. data

Hamming | Markov Chain Analysis | List>>

Report Generation

File Type: | Export>>

..\SGALAB_Subsys.mdl | Browse..

Step 7 : Analysis

Quick Reference>>

SGALAB :
Simple Genetic Algorithms Laboratory Toolbo

Status Report

SGALAB1003 GUI get ready ...

About | Help | Back | Next | Exit | <<<

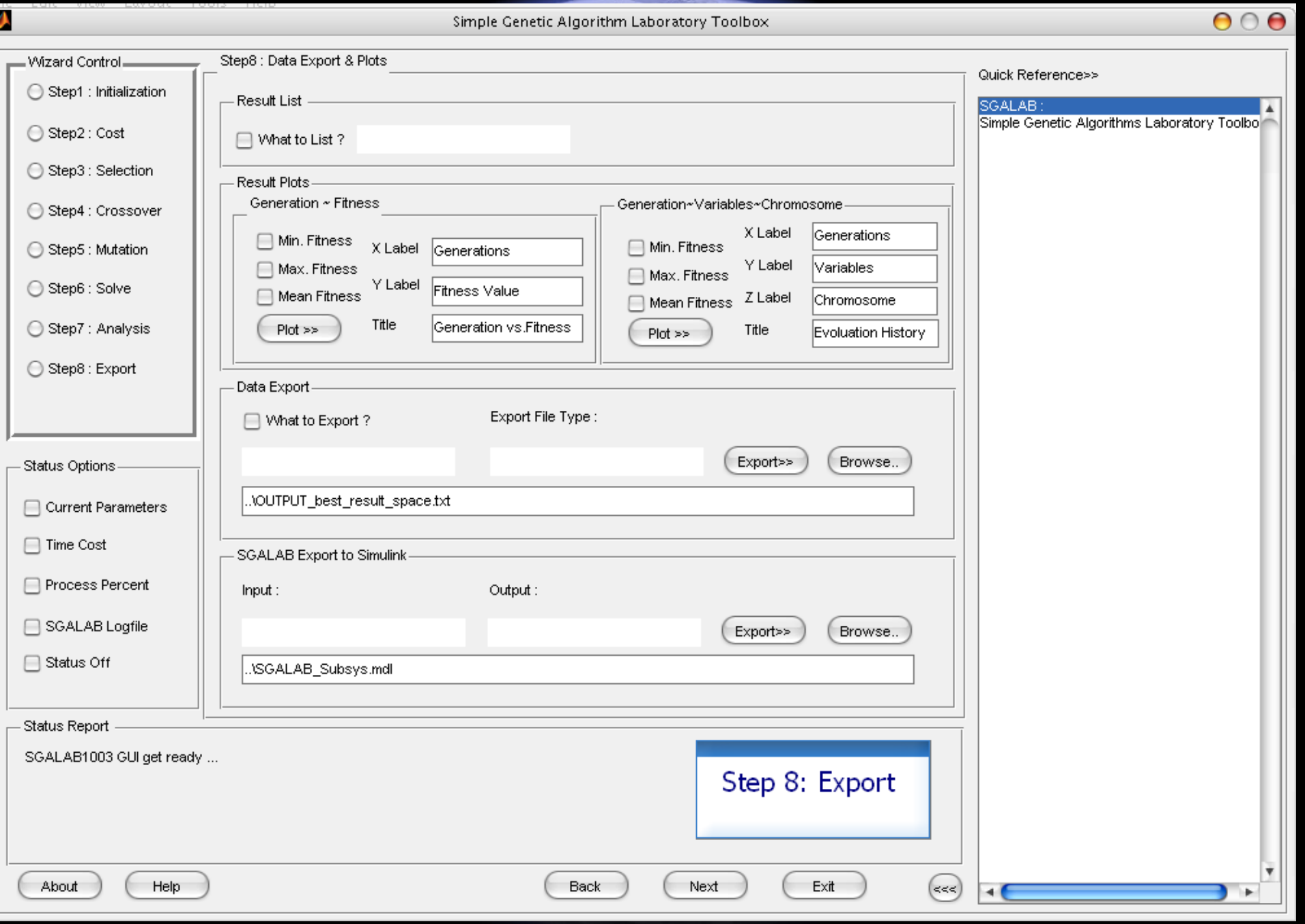

Simple Genetic Algorithm Laboratory Toolbox
Wizard Control
Step1 : Initialization
Step2 : Cost
Step3 : Selection
Step4 : Crossover
Step5 : Mutation
Step6 : Solve
Step7 : Analysis
Step8 : Export
Status Options
Current Parameters
Time Cost
Process Percent
SGALAB Logfile
Status Off
Step8 : Data Export & Plots
Result List
What to List ?
Result Plots
Generation ~ Fitness
Min. Fitness
Max. Fitness
Mean Fitness
Plot >>
X Label
Generations
Y Label
Fitness Value
Title
Generation vs.Fitness
Generation~Variables~Chromosome
Min. Fitness
Max. Fitness
Mean Fitness
Plot >>
X Label
Generations
Y Label
Variables
Z Label
Chromosome
Title
Evoluation History
Data Export
What to Export ?
Export File Type :
Export>>
Browse..
..\OUTPUT_best_result_space.txt
SGALAB Export to Simulink
Input :
Output :
Export>>
Browse..
..\SGALAB_Subsys.mdl
Quick Reference>>
SGALAB :
Simple Genetic Algorithms Laboratory Toolbo
Status Report
SGALAB1003 GUI get ready ...
Step 8: Export
About
Help
Back
Next
Exit
<<<

Table 6.1 Summary of Research on MO Optimization

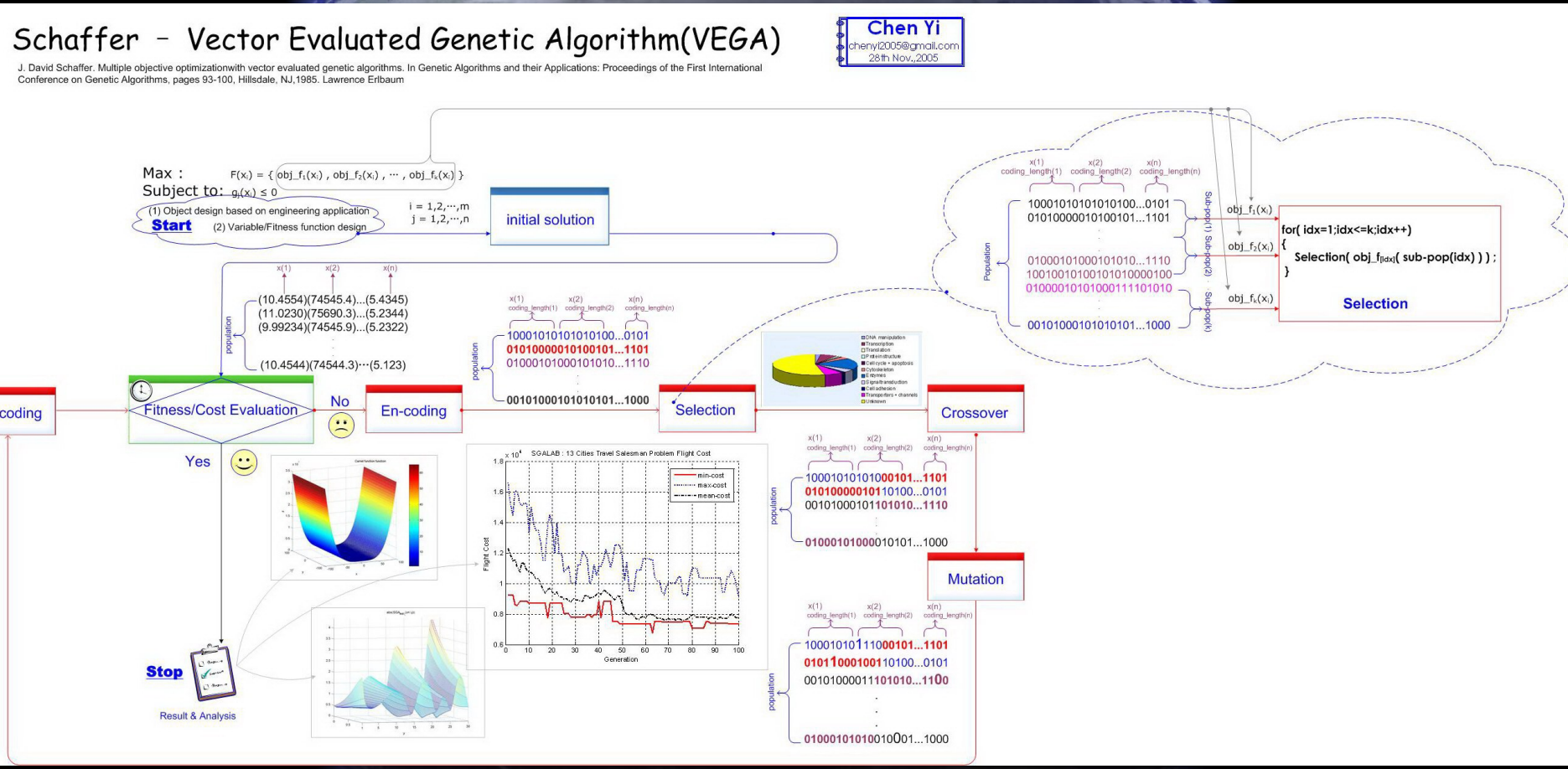


Fig. 6.2 Vector Evaluated Genetic Algorithm [Schaffer (1984)PhD, Schaffer (1985)]

Multiobjective Genetic Algorithm (MOGA)

Chen Yi
chenyi2005@gmail.com
28th Nov.,2005

Fonseca, C. M. & P. J. Fleming: "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization", *Proc. of the 5th Inter. Conf. on GAs*, pp. 416-423, 1993

Multiobjective Genetic Algorithm (MOGA:Fonseca & Fleming 1993)

1. Fitness **Pareto ranking** -- for each chromosome *ci* calculate the number of chromosomes that dominate it (*ni*). Let rank *ri*=1+*ni* ;
2. **Fitness assignment** -- Assign fitness to *ci* using linear interpolation so that *ci* with lowest rank has maximum fitness;
3. Fitness **Average** fitnesses of chromosomes with same rank and assign as the new fitness;
4. Fitness **Sharing** -- Use niching methods to distribute chromosomes over the Pareto-optimal region.

The rest of the algorithm is a standard GA.

Max : $F(x_i) = \{ obj\_f_1(x_i) , obj\_f_2(x_i) , \cdots , obj\_f_k(x_i) \}$

Subject to: $g_j(x_i) \le 0$

(1) Object design based on engineering application

Start (2) Variable/Fitness function design

$i = 1,2,\cdots,m$

$j = 1,2,\cdots,n$

initial solution

x(1) x(2) x(n)

population

(10.4554)(74545.4)...(5.4345)
(11.0230)(75690.3)...(5.2344)
(9.99234)(74545.9)...(5.2322)
(10.4544)(74544.3)⋯(5.123)

De-coding

Fitness/Cost Evaluation

No

En-coding

x(1) x(2) x(n)
coding_length(1) coding_length(2) coding_length(n)

population

1000101010101010100...0101
0101000001010010 1...1101
0100010100010101 0...1110
00101000101010101...1000

MOGA Block

Selection

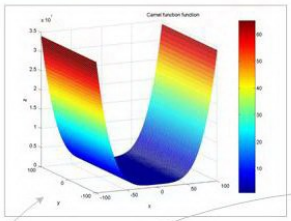
Crossover

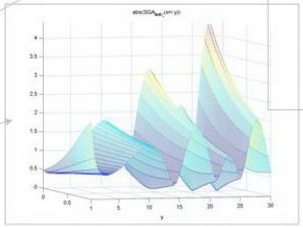
Mutation

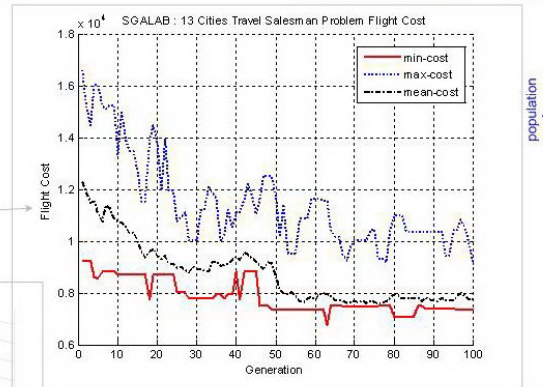
Yes

Stop

Result & Analysis







population

1000101010100010 1...1101
0101000001011010 0...0101
0010100010110101 0...1110
01000101000010101...1000

population

1000101011100010 1...1101
0101100010011010 0...0101
0010100001110101 0...1100
01000101010010001...1000

Niched Pareto Genetic Algorithm (NPGA)

Horn, J. and N. Nafpliotis (1993). Multiobjective optimization using the niched pareto genetic algorithm. IlliGAL Report 93005, Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana, Champaign.

Chen Yi
chenyi2005@gmail.com
28th Nov.,2005

Max : $F(x_i) = \{ obj\_f_1(x_i), obj\_f_2(x_i), \cdots, obj\_f_k(x_i) \}$

Subject to: $g_j(x_i) \le 0$

(1) Object design based on engineering application

Start (2) Variable/Fitness function design

$i = 1,2,\cdots,m$

$j = 1,2,\cdots,n$

initial solution

population

x(1) x(2) x(n)

(10.4554)(74545.4)...(5.4345)
(11.0230)(75690.3)...(5.2344)
(9.99234)(74545.9)...(5.2322)
(10.4544)(74544.3)⋯(5.123)

Niche count $m_i = \sum_{j=1}^{population} SH[d(i,j)]$

j = 1 ~ population

De-coding

Fitness/Cost Evaluation

No

En-coding

Yes

x(1) coding_length(1) x(2) coding_length(2) x(n) coding_length(n)

10001010101010100...0101
01010000010100101...1101
01000101000101010...1110
00101000101010101...1000

This technique might also be called **flat** fitness sharing, since the same effect is produced in normal (single-attribute) fitness sharing when two individuals have the exact **same fitness**. When fi = fj (i.e. the fitness function is flat for individuals i and j), then their shared fitnesses, fi/mi and fj/mj respectively, are ordered solely by their niche counts mi and mj. This situation would arise on a plateau of constant Fitness, where the radius of the plateau was much larger than sigma-share. Although we found no studies that analyzed or reported the behaviour of fitness sharing in such regions, we speculate that the population would spread out across the plateau. We should expect to find individuals sigma-share units apart from one Another, assuming a sufficiently large population to cover the plateau, and assuming crossover and mutation do not remove excessive numbers of individuals from the plateau each generation.

Stop

Result & Analysis

Selection (tournament)

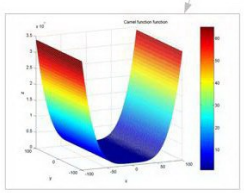
Flat fitness sharing

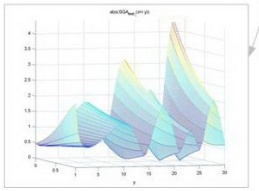
Crossover

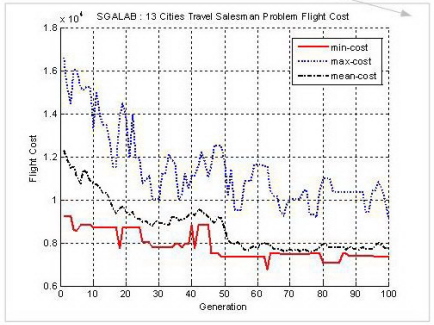
10001010101000101...1101
01010000010110100...0101
00101000101101010...1110
01000101000010101...1000

Mutation

10001010111000101...1101
01011000100110100...0101
00101000011101010...1100
01000101010010001...1000







Non-dominated Sorting Genetic Algorithm(NSGA)

Srinivas, N. and K. Deb (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* 2(3),221–248.

Chen Yi
chenyi2005@gmail.com
28th Nov.,2005

Max : $F(x_i) = \{ obj\_f_1(x_i) , obj\_f_2(x_i) , \cdots , obj\_f_k(x_i) \}$

Subject to: $g_j(x_i) \le 0$

(1) Object design based on engineering application

Start (2) Variable/Fitness function design

$i = 1,2,\cdots,m$

$j = 1,2,\cdots,n$

initial solution

population

x(1) x(2) x(n)

(10.4554)(74545.4)...(5.4345)
(11.0230)(75690.3)...(5.2344)
(9.99234)(74545.9)...(5.2322)
.
(10.4544)(74544.3)⋯(5.123)

Goldberg' s method
Input : chromosomes x(i), I = 1,2,⋯,population;
Output : rank of fitness

Step 1 : rank 1 to the non-dominated chromosomes;
Step 2 : removing ranked chromosomes from input chromosomes;
Step 3: go on next round, rank 2 will be assigned to nondominated fitness among rest chromosomes;
Step 4: until all the fitness are ranked

MOGA method(MOGA:Fonseca and Fleming, 1993)
Input : chromosomes x(i), I = 1,2,⋯,population;
Output : rank of fitness

step 1: rank 1 is given to the nondominated chromosomes ;
step 2: removing them from contention;
step 3: finding the next non-dominated chromosomes, removing them from contention, rank equally to the number of its dominating individuals plus one ;
step 4: process continues until the entire population is ranked.

De-coding

Fitness/Cost Evaluation

No

En-coding

Yes

x(1) coding_length(1) x(2) coding_length(2) x(n) coding_length(n)

population

1000101010101010100...0101
0101000001010010 1...1101
0100010100010101...1110
.
00101000101010101...1000

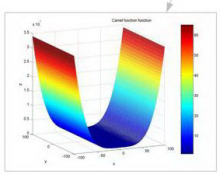
Pareto-ranking

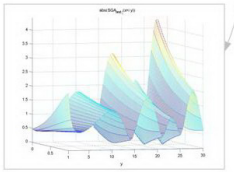
Non-dominated Sorting GA (NSGA: Deb, 1995)
Input : chromosomes x(i), I = 1,2,⋯,population;
Output : fitness

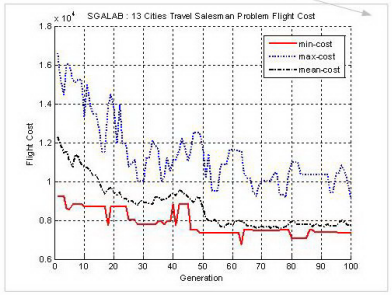
step 1: The non-dominated individuals are first identified and assigned a large/small **dummy** fitness value;
step 2: To maintain diversity in the population, these individuals are shared with their dummy fitness values;
step 3: After **sharing**, these non-dominated individuals are ignored temporarily;
step 4: The second non-dominated **front** in the rest of the population is identified and assigned a dummy fitness value that is kept smaller than the minimum shared dummy fitness of the previous front;
step 5: This process is continued until the entire population is classified into several fronts.

Stop

Result & Analysis

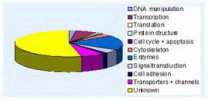






NSGA Block

Selection
(stochastic remainder proportionate selection)



Crossover

x(1) coding_length(1) x(2) coding_length(2) x(n) coding_length(n)

population

1000101010100 00101...1101
0101000001011 0100...0101
0010100010110 1010...1110
.
0100010100 0010101...1000

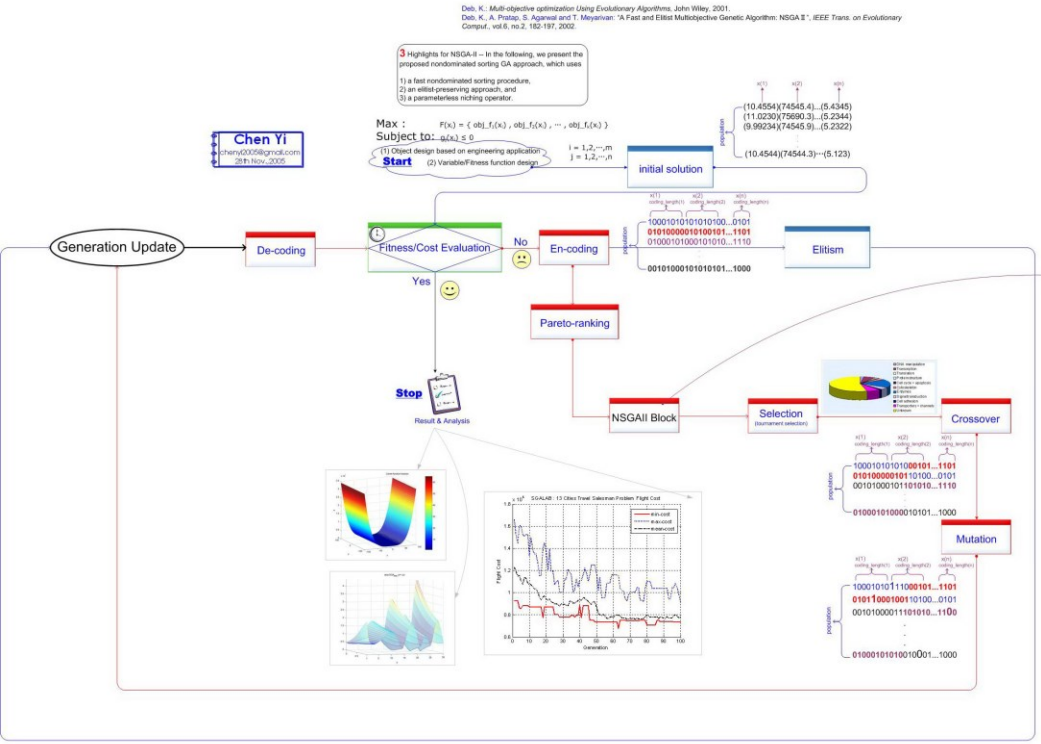
Mutation

x(1) coding_length(1) x(2) coding_length(2) x(n) coding_length(n)

population

1000101011110 00101...1101
0101100010011 0100...0101
0010100001110 1010...1100
.
.
.
0100010101001 0001...1000

Non-dominated Sorting Genetic Algorithm-II(NSGA-II)




```
Fast non-dominated sort ()
{
Step 1) pareto rank
Step 2) get fronts
Step 3) sort population into different groups according to fronts
Step 4) crowding distance assignment
}
```

NSGAII goes as following steps:

Step 1: Fast non-dominated sort

Step 2: crowding distance assignment

Step 3 :crowded-comparison operator

```
Crowding_distance_assignment(I)
/* in current front , there are m fitness functions , n = 1,2,…,m*/
{
(1) Set initialization distance to 0 ,
     Dist(2:n-1) = 0,
     Dist(1)     = Dist(n) = inf

(2) MAX & MIN sort objective value ,namely,
    MAX & MIN sort( fitness )

(3) get each distance by following function:

    for  idx = 2:1:(n-1)
    {
    Dist[idx] = Dist[idx]|last + (fitness[idx+1]|k - fitness[idx-1]|k) / (fitness|k^MAX - fitness|k^MIN)
    }

}
```

```
crowded-comparison operator( ranks, crowded_distances )
{                    i < j
  if ( rank( i ) < rank( j ) )
    or
    ( rank( i ) == rank( j ) )&& distance( i ) > distance( j )

       get j;
  else

       get i;
}
```

2) Crowded-Comparison Operator: The crowded-comparison operator ($\prec_n$) guides the selection process at the various stages of the algorithm toward a uniformly spread-out Pareto-optimal front. Assume that every individual i in the population has two attributes:

1) nondomination rank (i_{rank}).
2) crowding distance (i_{distance}).

We now define a partial order $\prec_n$ as

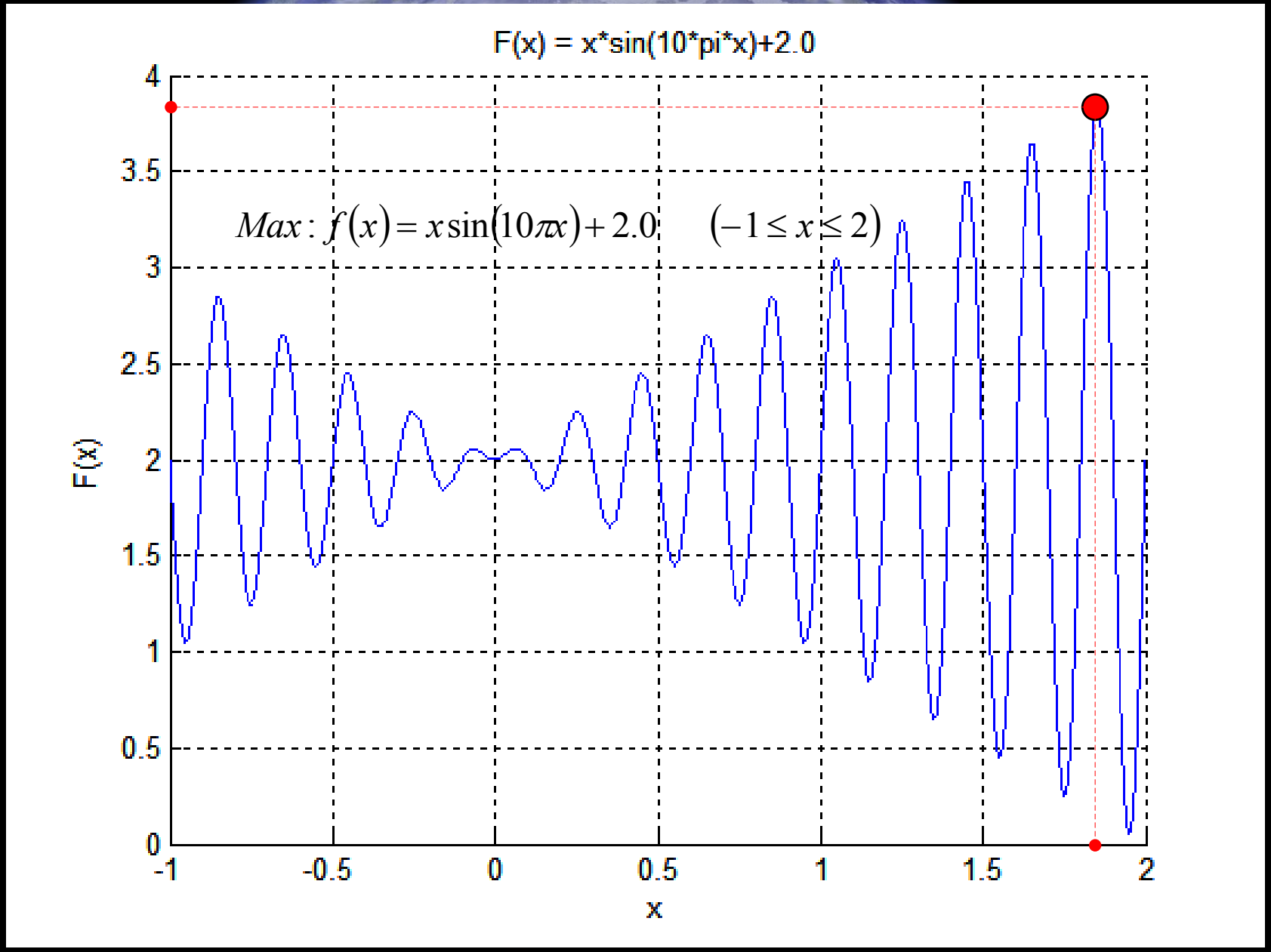
$$i \prec_n j \quad \text{if } (i_{\text{rank}} < j_{\text{rank}}) \text{ or } ((i_{\text{rank}} = j_{\text{rank}}) \text{ and } (i_{\text{distance}} > j_{\text{distance}})).$$

That is, between two solutions with differing nondomination ranks, we prefer the solution with the lower (better) rank. Otherwise, if both solutions belong to the same front, then we prefer the solution that is located in a lesser crowded region.

With these three new innovations—a fast nondominated sorting procedure, a fast crowded distance estimation procedure, and a simple crowded comparison operator, we are now ready to describe the NSGA-II algorithm.

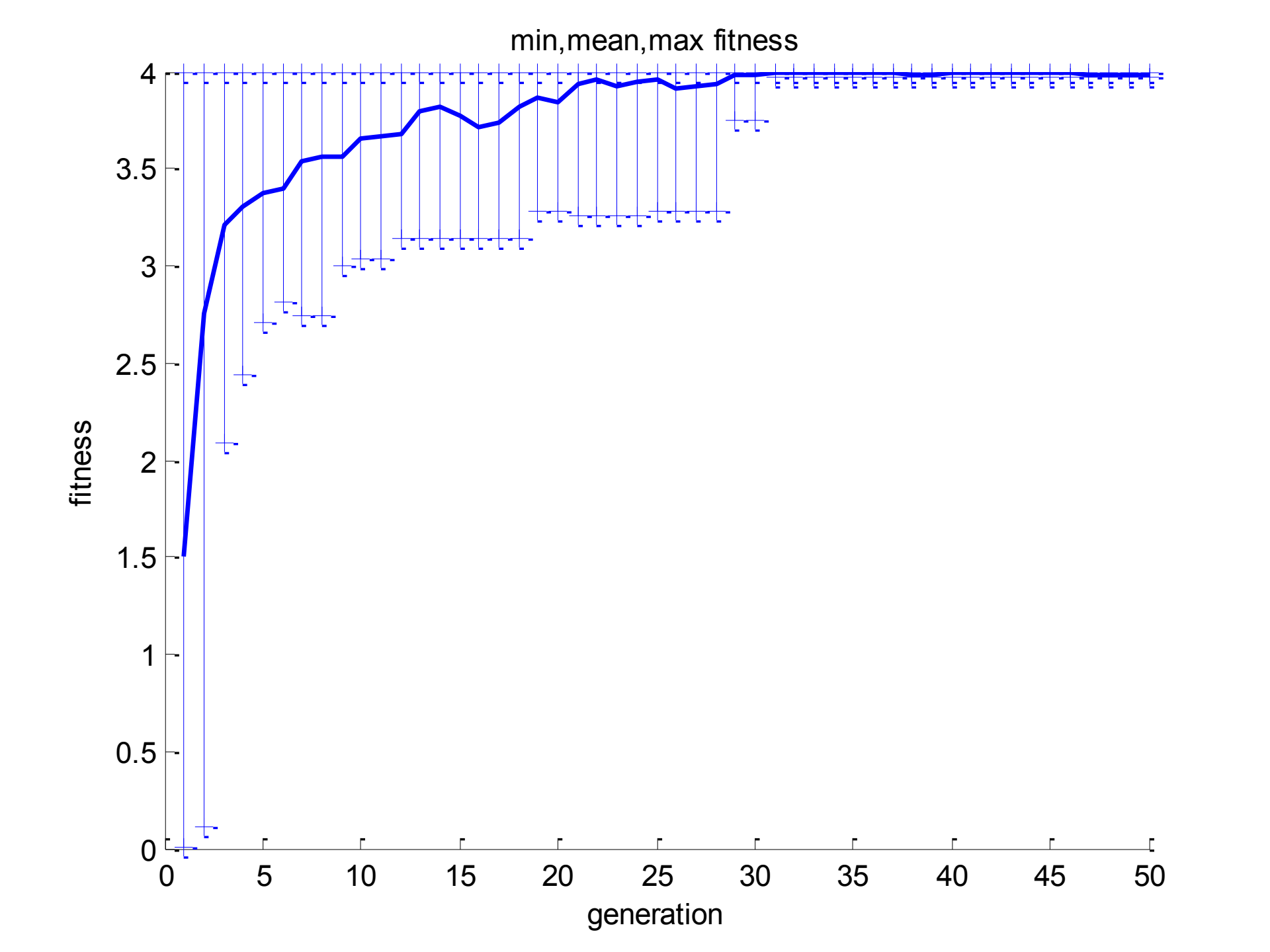
Contents

A Simple Example (using SGALAB1001)
Unconstrained Numerical Optimization (Michalewicz, 1996)



Parameters	Value	
Max Generation	100	300
Crossover Probability	0.8	0.9
Mutation Probability	0.01	0.01
Population	30	100
step	0.01	0.001
$F(x_0)\|_{max}$	3.843702	3.849781
$F(x_0)\|_{min}$	0.060569	0.071226
$F(x_0)\|_{mean}$	3.635146	3.830261
x_0	1.853229	1.851282
Cost time (sec.)	1.2350	15.1090

Contents


min,mean,max fitness
fitness
generation
0
0.5
1
1.5
2
2.5
3
3.5
4
0
5
10
15
20
25
30
35
40
45
50

TSP 13 Cities Demo for SGALAB1003

The ‘SGALAB’ will take a world wide show of its new product -SGALAB 1003 , there are 13 cities on the list of its Biz-traveling.
Give 1~13 numbers to each city as following:

Num.	City
1	*San Francisco*
2	*New York*
3	*Brasilia*
4	*Madrid*
5	*London*
6	*Stockholm*
7	*Johannesburg*
8	*Moscow*
9	*New Delhi*
10	***Chongqing***
11	*Shanghai*
12	*Tokyo*
13	*Canberra*

The start point is the city of Chongqing (10), how can SGALAB company make a plan for its world wide Biz-travel with lowest traveling cost?

Simulation Parameters Setting

Parameters	Value
Population	50
Max Generation	100
City-Number	13
Start city	10
Pc (CX)	0.9
Pm(Insertion)	0.01
TSP_cost_matrix	Appendix1
Run time	7.172000

13 Cities Traveling Salesman Problem

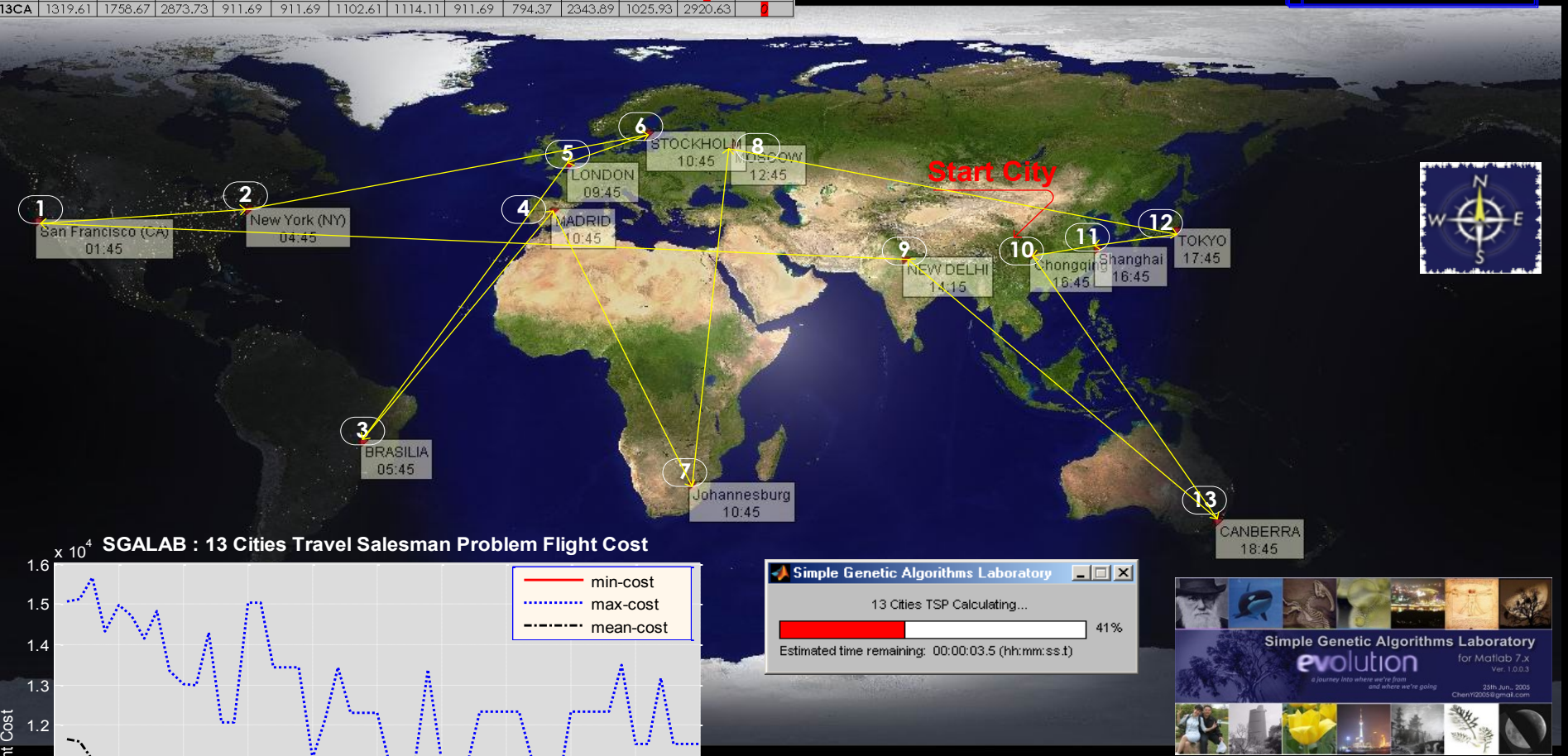
Simple Genetic Algorithms Laboratory (SGALAB)1.0.0.3 for Matlab 7.x

31st Nov., 2005

Chen Yi

chenyi2005@gmail.com

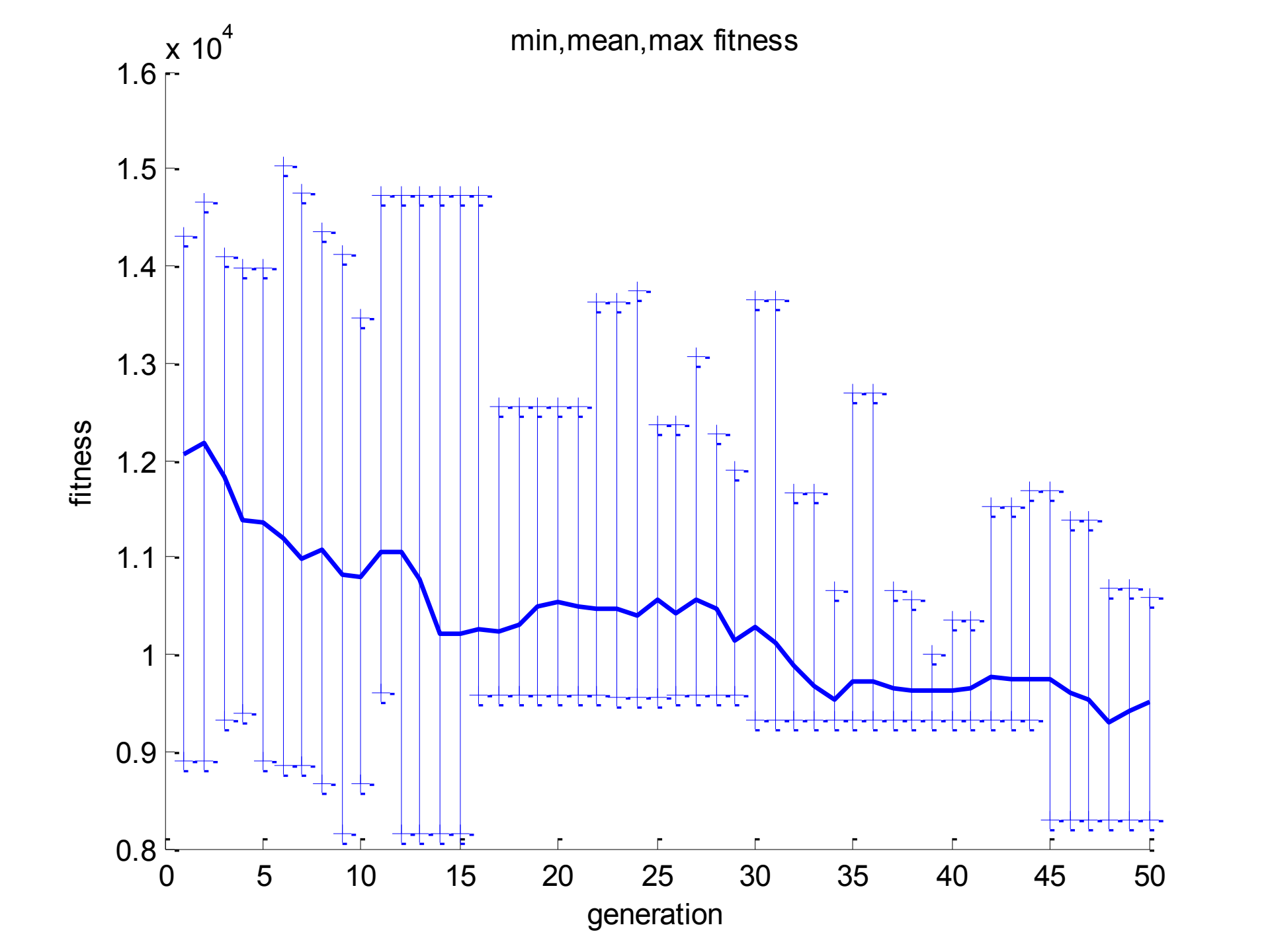
Cost	1SA	2NY	3BR	4MA	5LO	6ST	7JO	8MO	9ND	10CQ	11SH	12TO	13CA
1SA	0	106.05	1345	845	845	845	1295	845	607	656.69	525	1019	735.34
2NY	106.05	0	916	575	297	381	989	685	604	775.69	644	359	933.34
3BR	1109	999	0	784	919	1068	1279	1059	1643	2145	1971	1266	1954.68
4MA	382.95	273.71	882.89	0	44.18	79.78	352.26	239.34	1042.07	942.65	365.76	870.23	2163.71
5LO	564.38	496.88	632.54	47.49	0	18.26	465.82	155.27	405.54	886.89	548.03	652.16	1296.52
6ST	1374.72	260.9	877.05	54.18	18.06	0	702.44	124.43	434.84	936.59	634.87	600.09	2864.24
7JO	948.65	816.11	2054.74	610.59	580.81	647.82	0	610.59	707.39	1496.76	896.53	787.81	2696.92
8MO	509	315	810	291	170	145	433	0	307	774	525	700	1286.8
9ND	597.23	568.52	1519.73	384.98	372.12	385.9	551.29	385.9	0	546.69	470.89	493.86	1804.95
10CQ	703.19	836.1	2510.72	927.92	927.92	927.92	1496.76	1074.72	712.86	0	180.02	918.26	1767.14
11SH	555.79	459.13	1555	490.29	398.71	473.63	146.31	451.88	567.87	180.02	0	459.13	1753.85
12TO	1647.04	2229.67	3952.34	2352.91	2352.91	3323.96	3006.5	2711.45	2117.62	1266.09	427.26	0	2582.71
13CA	1319.61	1758.67	2873.73	911.69	911.69	1102.61	1114.11	911.69	794.37	2343.89	1025.93	2920.63	0



Best TSP Path :

10 →13→9→1→2→6→5→3→4→7→8→12→11→10

Lowest Flight Cost : 7.3505e+003

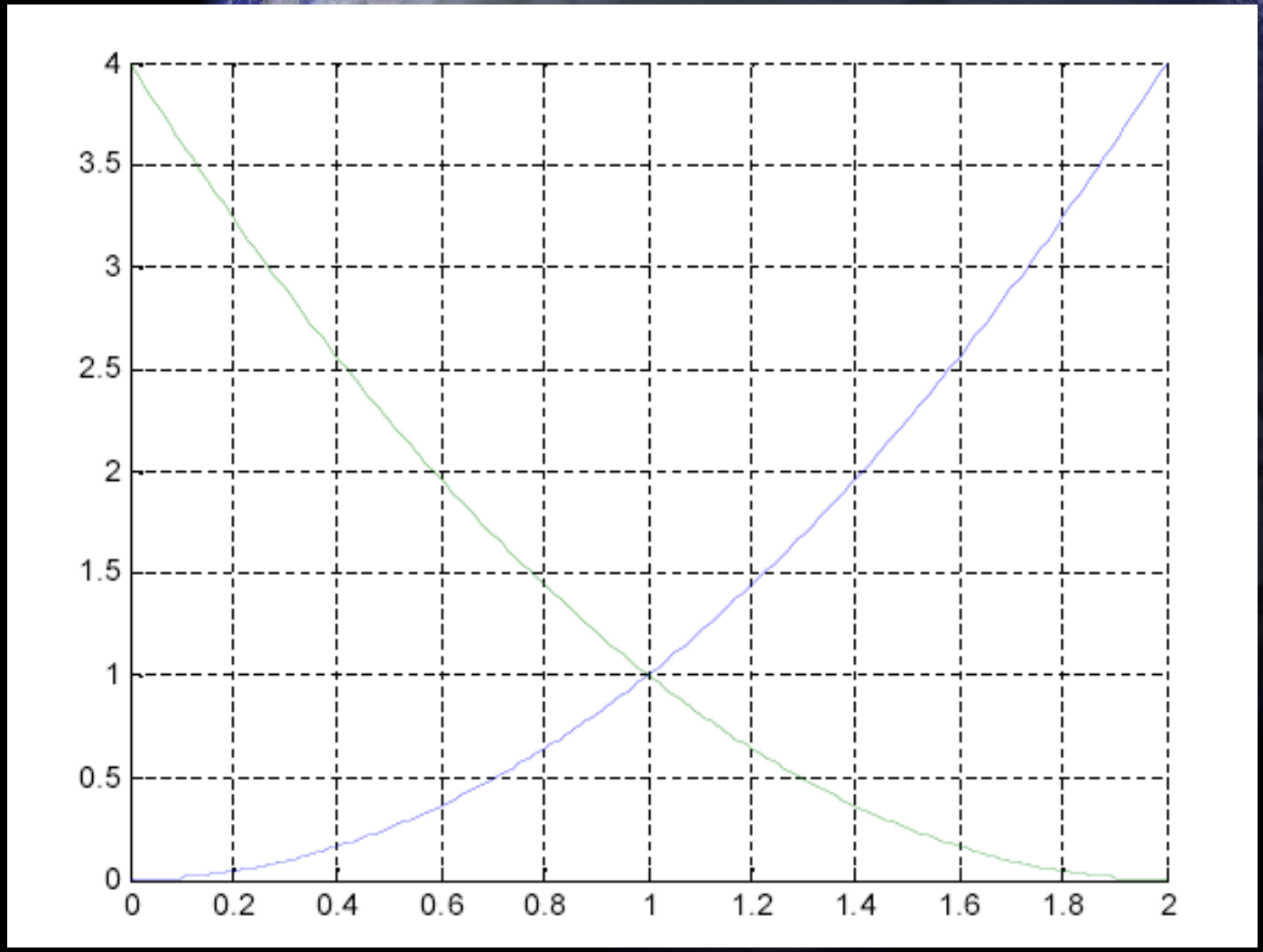

min,mean,max fitness
x 10^4
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
fitness
0
5
10
15
20
25
30
35
40
45
50
generation

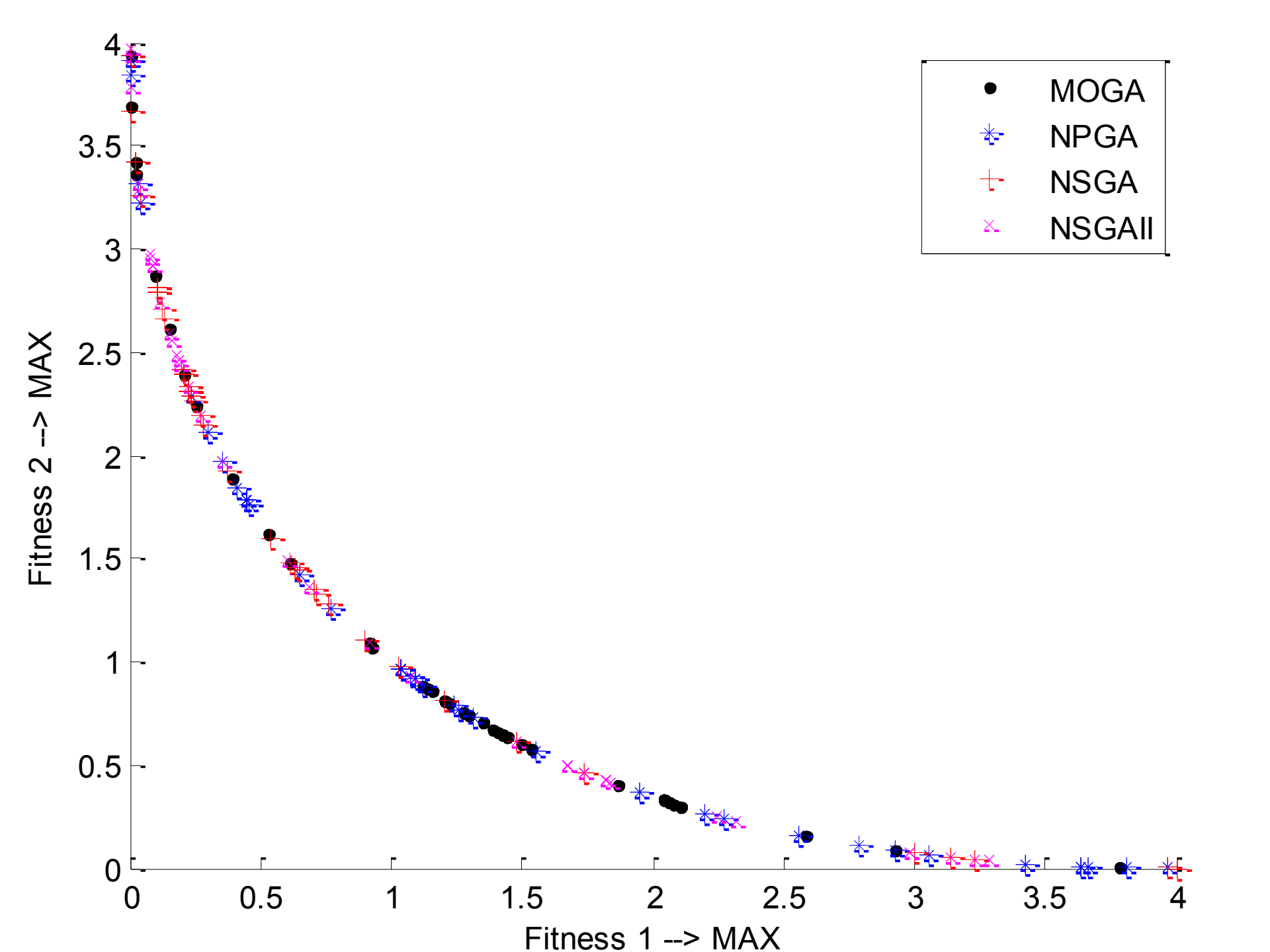
Contents

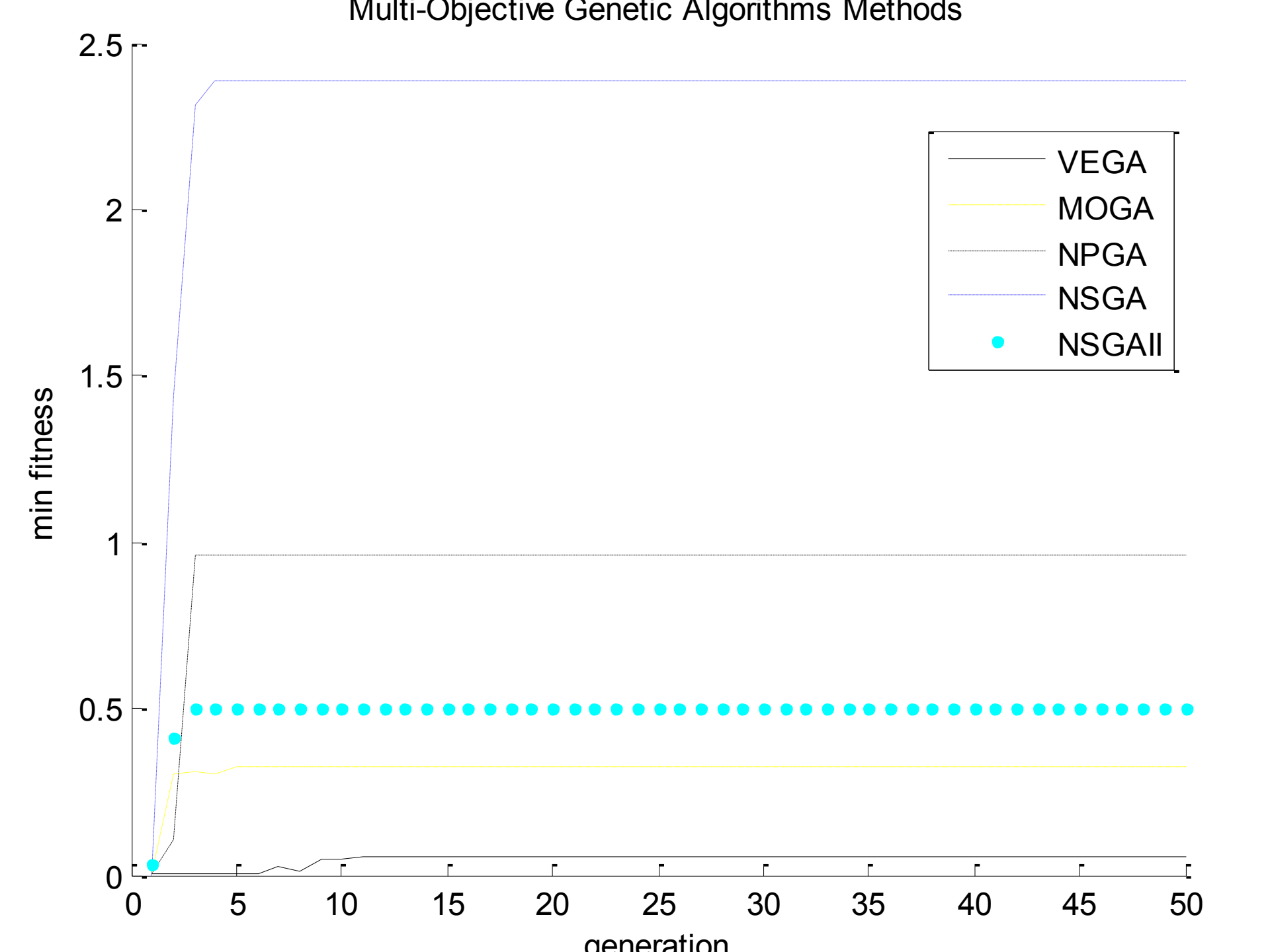
Demo3 - A multi-objective problem

Assume the following 2 fitness ,

$$\begin{cases} f_1(x) = x^2 \\ f_2(x) = (x-2)^2 \end{cases} \qquad x \in [0,2]$$, so f1 = f2 = 1 should be the best results




MOGA
NPGA
NSGA
NSGAII
Fitness 2 --> MAX
Fitness 1 --> MAX
0
0.5
1
1.5
2
2.5
3
3.5
4


Multi-Objective Genetic Algorithms Methods
min fitness
generation
VEGA
MOGA
NPGA
NSGA
NSGAII
0
0.5
1
1.5
2
2.5
0
5
10
15
20
25
30
35
40
45
50

Contents

GA Videos

- 01-MATLAB - What is a Genetic Algorithm
- 02-How algorithms evolve (Genetic Algorithms)

Some Links where have GAs discussion :

(1) http://www.mathtools.net/MATLAB/Genetic_algorithms/

Mathtools.net >

Link Exchange for the Technical Computing Community

Hosted by The MathWorks
www.mathworks.com

Search MATLAB Go

Add Link | Update a Link | Recent Links | Newsletter | Email this page | My Profile

Home > MATLAB > Genetic algorithms

Top-Rated Links
Most-Visited Links

Genetic algorithms

Related Categories

- C,C++/Genetic algorithms (9)
- Excel/Genetic algorithms (3)
- Fortran/Genetic algorithms (4)
- Java/Genetic algorithms (27)

MathWorks - Genetic Algorithm and Direct Search Toolbox Update Link / Bad Link?

The Genetic Algorithm and Direct Search Toolbox extends the optimization capabilities in MATLAB and the Optimization Toolbox with tools for using the genetic and direct search algorithms.

Submitted Jan 23, 2004

Rating: N/A Rate this link Total Visits: 1

Free Online Seminar: Advanced Optimization Using MATLAB Update Link / Bad Link?

Optimization is an important and expanding area that helps engineers and scientists find improved solutions to their problems. MATLAB and the Optimization Toolbox provide powerful capabilities for optimization with robust algorithms for general and large-scale optimization, linear programming, quadratic programming, nonlinear equations, and more.

Submitted Mar 12, 2003
Updated Feb 06, 2004

Rating: N/A Rate this link Total Visits: 1

GA Toolbox Update Link / Bad Link?

A module that contains software routines for implementing genetic algorithms (GAs) and other evolutionary computing techniques.

Submitted Jul 02, 1999

Rating: N/A Rate this link Total Visits: 1

Genetic Algorithm Optimization Toolbox Update Link / Bad Link?

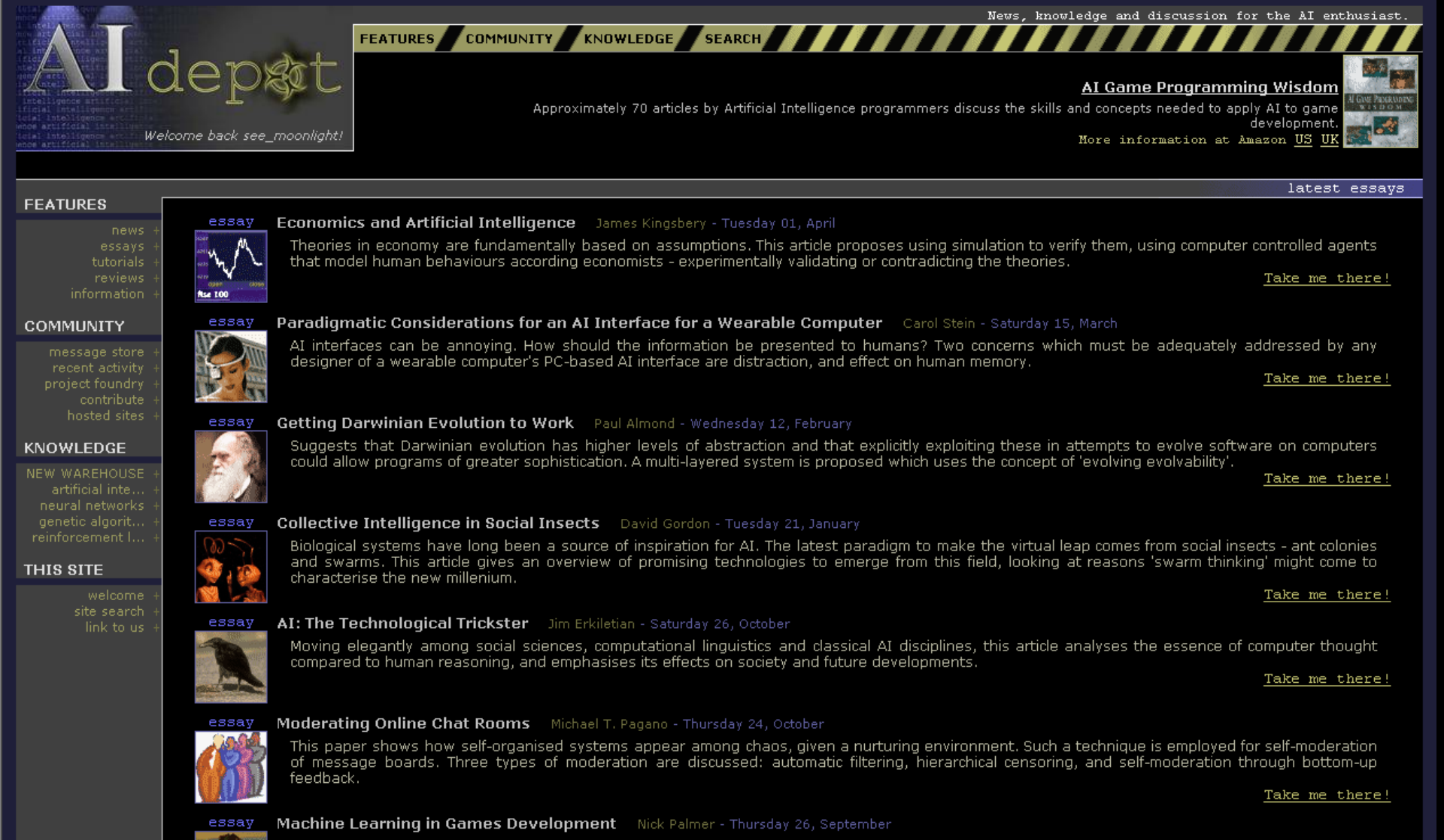
GAOT implements simulated evolution in the MATLAB environment using both binary and real representations. Ordered base representation has also been added to the toolbox. This implementation is very flexible in the genetic operators, selection functions, termination functions as well as the evaluation functions that can be used. The implementation is described in a technical paper.

Submitted Jul 10, 1999
Updated Jan 23, 2004

Rating: N/A Rate this link Total Visits: 1

Some Links where have GAs discussion :

(2) http://geneticalgorithms.ai-depot.com/



Some Links where have GAs discussion :

(3) http://www.aic.nrl.navy.mil/galist/

** Privacy and Security Notice **

GA Archives

The Genetic Algorithms Archive

Please note: Many of the links in the GA Archive connect to sites outside of this NCARAI web server. We are not responsible for the content of these other sites.

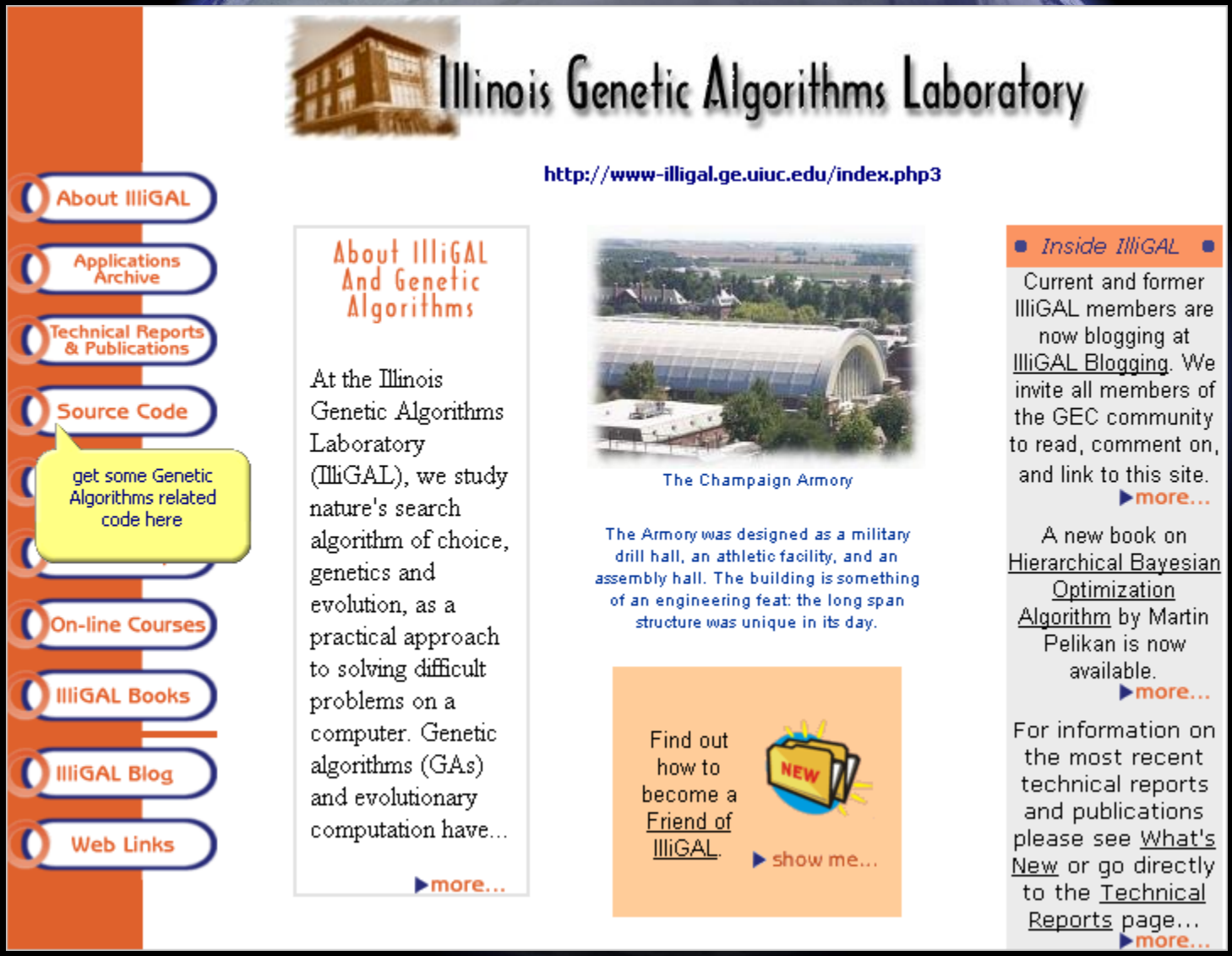
The Genetic Algorithms Archive is a repository for information related to research in genetic algorithms and other forms of evolutionary computation. Available from this site are past issues of the GA-List (now called EC Digest,) source code for many EA implementations, and announcements about EC-related conferences. Also, links are given to many interesting sites around the World with material related to evolutionary computation. This archive is maintained at The Navy Center for Applied Research in Artificial Intelligence.

Contents:

- Calendar of EC-related events
- GA-List Archive: back issues, source code, information
- Links to other GA-related information
- Links to EC research groups' home pages
- NEW Dissertations on Evolutionary Computation
- Other related societies and sites

Some Links where have GAs discussion :

(4) http://www-illigal.ge.uiuc.edu/index.php3



Options	Source Code
FTP	XCS (+ tournament selection) classifier system implementation in C, version 1.2 (for IlliGAL Report 2003023)
FTP	Implementation of anticipatory classifier system (ACS2) in C++ (for IlliGAL Report 2001026)
FTP	Ordering Messy GA (OmeGA) in C++ (for IlliGAL Report 2000034)
FTP	Bayesian Optimization Algorithm with Decision Graphs in C++, version 1.1 (for IlliGAL Report 2000025)
FTP	XCS classifier system implementation in C, version 1.1
FTP	XCS implementation in Java, version 1.0 (for IlliGAL Report 2000017)
FTP	Bayesian Optimization Algorithm with Decision Graphs in C++, version 1.0 (for IlliGAL Report 2000025)
FTP	XCS classifier system implementation in C, version 1.0 (for IlliGAL Report 99021.)
FTP	Extended Compact Genetic Algorithm in C++ (for *IlliGAL Report 99016*.)
FTP	Bayesian Optimization Algorithm (BOA) code in C++ (for *IlliGAL Report 99011*.)
FTP	Linkage Learning GA code in C++ (for *IlliGAL Report 98006*.)
FTP	Messy GA code in C (for *IlliGAL Report 91008*.)
FTP	Messy GA code in Lisp (for *TCGA Report 90004*.)
FTP	Simple GA code (Pascal code from Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*.)
FTP	Simple GA code (C translation of the code from Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*.)
FTP	Simple Classifier System code (Pascal code from Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*.)

SGALAB

Evolution with the world

SxLAB Family Member

What is Genetic Algorithms ?

Thanks.

Dr Leo Chen

leo.chen@ieee.org

22/Oct/2020

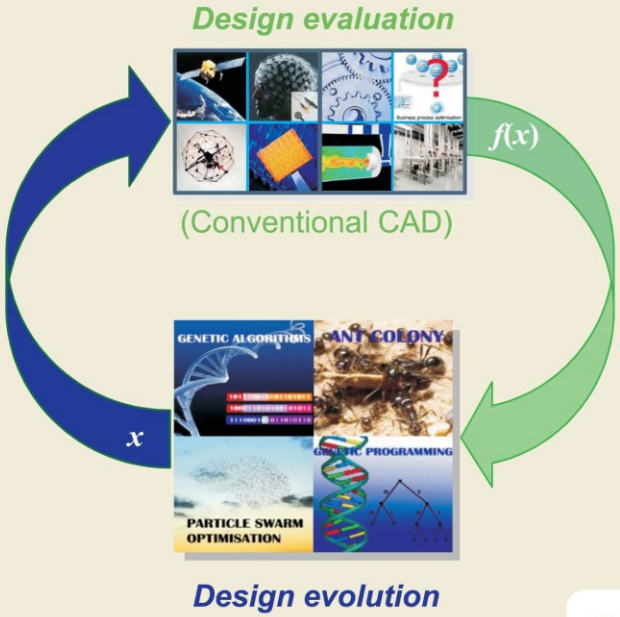
This is the last page

https://github.com/SGALAB/What-is-Genetic-Algorithms

Computational Intelligence Assisted Design

In Industrial Revolution 4.0

Yi Chen and Yun Li



CRC Press
Taylor & Francis Group
A SCIENCE PUBLISHERS BOOK

CPD CERTIFIED
The CPD Certification Service

For more information visit:
www.crcpress.com/9781498760669

i4AI Lab
Industry 4.0 Artificial Intelligence Laboratory
工业4.0人工智能实验室

東莞理工學院
DONGGUAN UNIVERSITY OF TECHNOLOGY

工业 4.0 时代的计算智能辅助设计

CRC 出版社，ISBN 9781498760669

内容概要

- 计算智能及工业 4.0 时代的系统设计自动化
- 工程和社会科学的多学科应用案例分析，包含如何利用计算智能研究工业产品和系统设计的自动化
- 表格、插图、图形、实例、习题、算法实施的 MATLAB 源代码，完整的 CI 实现、实验框架

目录

作者

陈益　李耘

东莞理工学院高层次骨干人才　东莞理工学院海外特聘杰出人才教授
工业4.0人工智能实验室产业转化副主任　工业 4.0 人工智能实验室主任
IEEE高级会员　英国格拉斯哥大学新加坡分校创校校长
英国皇家特许工程师，英国皇家特许工程师认证官　英国思克莱德大学设计、制造与工程管理专业教授

- 目录及第 1、5 两章赠阅：CIAD_ToC_and_Chapters_1&5
- 全书直接从出版社购买：Taylor & Francis
- 指导教师获取免费教材：Request Inspection Copy here
- 图书馆中心获取产品资料：Available on Taylor & Francis eBooks

ISBN 9781498760669
Published July 23, 2018 by CRC Press
504 Pages 13 Color & 297 B/W Illustrations

一张世界上智慧最集中的照片

1.彼得．德拜　美国物理化学家。1884年出生于荷兰。1901年进入德国亚琛工业大学学习电气工程， 1905年获电子工程师学位，
因他通过偶极矩研究及x射线衍射研究对分子结构学科所作贡献而于1936年获诺贝尔化学奖金。1966年逝世。

2.威廉．亨利．布喇格（w.h.bragg，1862 - 1942）是现代固体物理学的奠基人之一，他早年在剑桥三一学院学习数学，曾任利兹大学、伦敦大学教授，1940年出任皇家学会会长。由于在使用x射线衍射研究晶体原子和分子结构方面所作出的开创性贡献，他与儿子w.l.布喇格分享了1915年诺贝尔物理学奖。父子两代同获一个诺贝尔奖，这在历史上恐怕是绝无仅有的。同时，他还作为一名杰出的社会活动家，在二三十年代是英国公共事务中的风云人物。

3. 爱因斯坦是20世纪最伟大的科学家，被公认为人类历史上最具有创造性才智的人物之一。
他的名字与相对论密不可分，其实，相对论包括两种理论：其一是他1905年提出声狭义相对论；
其二是他1915年提出的广义相对论。后者，我们最好称之为爱因斯坦引力论

4.埃伦费斯特（p. ehrenfest, 1880 - 1933）¯荷兰物理学家

5.1930年，英国物理学家保罗．狄拉克（paul adrien maurice dirac，1902～1984）
用数学方法描述电子运动规律时，发现电子的电荷可以是负电荷、也可以是正电荷的。狄拉克猜想，
在自然界中可能存在一种“反常的”带正电荷的电子

6.薛定谔（erwin schrodinger，1887-1961）奥地利理论物理学家，与爱因斯坦、玻尔、玻恩、海森伯等一起20世纪20年代后期，发展了量子力学。因建立描述电子和其他亚原子粒子的运动的波动方程，
获得1933年诺贝尔物理奖。

7.1922¯1923年间，康普敦（a.h.compton l892¯1962）研究了x射线经金属或石墨等物质散射后的光谱。

8.美籍奥地利科学家沃尔夫冈．泡利（wolfgang e.pauli，1900～1958），是迎着20世纪一同来到世界的，父亲是维也纳大学的物理化学教授，教父是奥地利的物理学家兼哲学家。

9.海森伯，w.k.（werner karl heisenberg 1907～1976）德国理论物理学家，量子力学第一种有效形式（矩阵力学）的创建者。

10.玻恩，m.(max born 1882～1970)德国理论物理学家，量子力学的奠基人之一。

11.尼尔斯．玻尔(bohr,niels)1885年10月7日生于丹麦首都哥本哈根，父亲是哥本哈根大学的生理学教授．从小受到良好的家庭教育．1903年进入哥本哈根大学学习物理，1909年获科学硕士学位，1911年获博士学位大学二年级时研究水的表面张力问题，自制实验器材，通过实验取得了精确的数据，
并在理论方面改进了物理学家瑞利的理论，研究论文获得丹麦科学院的金奖章．

12.普朗克，m.（max planck 1858～1947）近代伟大的德国物理学家，量子论的奠基人

13.居里夫人（1867 - 1934〕是最著名的女物理学家。她曾两次获诺贝尔奖，1903年的物理奖，1911年的化学奖。她受教育较晚，于1893 年获物理学位，1894年获数学学位，1903年获博士学位。局里夫人以放射性作为论文题目，她研究了很多物质，发现钍及其化合物的特性与铀相同。研究沥青铀矿时，她发现了镭和仆。1910年她成功的分离了纯镭。居里夫人对巴黎的局里实验室的建立作出很大贡献。

14.洛仑兹（hendrik antoon lorentz　1853～1928）与塞曼（pietr zeeman 1865～1943）因研究磁场对辐射现象的影响、发现塞曼效应，分享了1902年度诺贝尔物理学奖。

15.朗之万:1872年1月23日生于巴黎，法国著名的物理学家 。