SGALAB 1003

Genetic Algorithms Basic and it's Engineering Applications Using SGALAB Toolbox

à Simple Genetic Algorithms Laboratory Toolbox For Matlab 7.x User's Guide

Leo Chen(Yi)
Chenyi2005@gmail.com
8th Mar., 2005

Revision History

Doc Rev.#	Date of change	Name of person making change	Description of Change	SGALAB Version
1.0	7th Mar., 2005	Chen Yi	Initial version	1003

Preface

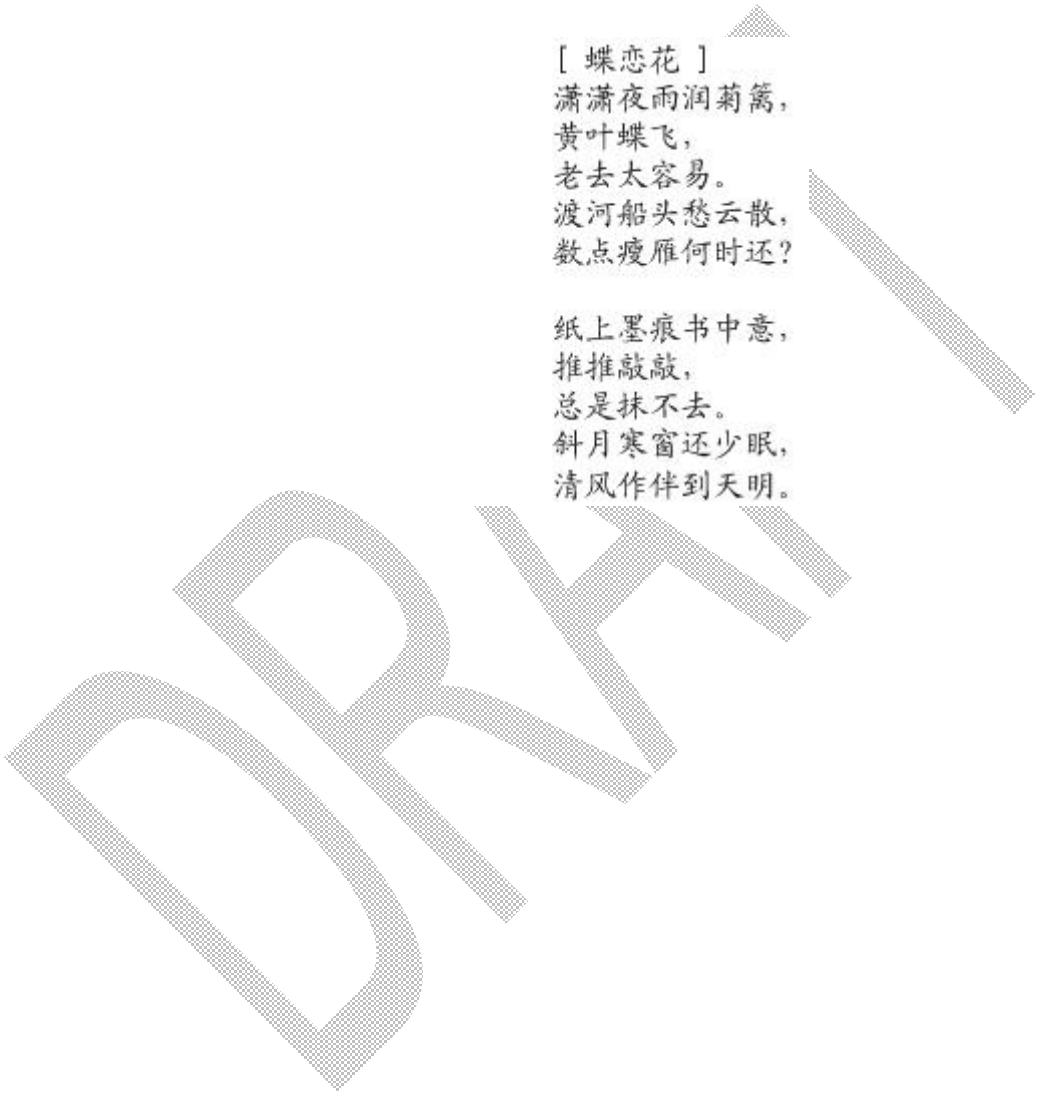
Genetic Algorithms(*GAs or GA*) was formally introduced in the United States in the 1970s by John Holland at University of Michigan. It is inspired by Darwin's theory about evolution. Solution to a problem solved by GAs is evolved. GAs has been widely studied, experimented and applied in many fields in engineering worlds. Not only does GAs provide a group of alternative methods to solve problems, but also it consistently outperforms other traditional methods in most of the problems link. Many of the real world problems involved finding optimal parameters, which might prove difficult for traditional methods but ideal for GAs. However, because of its outstanding performance in optimization, GAs has been wrongly regarded as a family of function optimizers. In particular, genetic algorithms works very well on mixed (continuous and discrete), combinatorial problems. GAs is started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are selected to form new solutions (offspring) are selected according to their fitness - the more suitable they are the more chances they have to produce. This is repeated until some condition (for example number of populations or improvement of the best solution) is satisfied. GAs is less susceptible to getting 'stuck' at local optima than gradient search methods. But it tends to be computationally expensive.

To use a genetic algorithms, you must represent a solution to your problem as a genome (or chromosome). The genetic algorithms then creates a population of solutions and applies genetic operators such as mutation and crossover to evolve the solutions in order to find the best one(s). This presentation outlines some of the basics of genetic algorithms. The three most important aspects of using a genetic algorithms are: (1)definition of the objective function (Fitness Function); (2)definition and implementation of the genetic representation(Encoding Method); (3)definition and implementation of the genetic operators. Once these three have been defined, the generic genetic algorithm should work fairly well. Beyond that you can try many different variations to improve performance, find multiple optima (species - if they exist), or parallelize the algorithms.

Simple Genetic Algorithms Laboratory (SGALAB)Toolbox for Matlab is a generic toolbox of genetic algorithms. The first version of SGALAB was made in April, 2002, when the author was still a post-graduate student in the State Key Laboratory of Mechanical Transmission of Chongqing University. His interesting on the Genetic Algorithms ,Fuzzy Logic Control, Artificial Neural Network or some related things

was and is pushing him to update the SGALAB toolbox, even now. Yes, it is a hard work need much efforts and it also thanks to the SGALAB to give the author a clue to insist on his study.

During the studying and growing period, the author wrote a poem as a motto for himself to encourage him to keep the evolutionary and adaptive attitude like GAs in his everyday life. Now, the author gives this poem to those people whose interesting is also on genetic algorithms, See the following:



Author's Info

Chen Yi
chenyi2005@gmail.com

Chen Yi is a versatile guy from Chonqqing, China. Leo is his English name he gave to himself. He graduated from the State Key Laboratory of Mechanical Transmission, Chongqing University in the July 2004, with the major Automotive Engineering.

After that, he attended UGS on July 5th, 2004. And now he's working as a Software Engineer in PLM Product Development Center of UGS Company, Shanghai.

[临江仙]
陈昔往事益归去，
古道西风依旧。
人生长河水长东，
春去遇秋来，天际寻鹤鸿。

日月星移话沧桑，
是非功过何在？
一路风雨兼程行，
唯清风明月，独在情怀中。

If you are interested in Genetic Algorithms or SGALAB Toolbox, we can have a discussion about it, the best place to find leo is in the following forum:

bbs.matwav.com ID: see_moonlight
www.dytrol.com ID: gas_boy
www.simwe ID: cdey

and you can also contact the author directly in these way :

Chen Yi
Email : chenyi2005@gmail.com
Leo.chen@ugs.com
MSN : cdey_china@hotmail.com
QQ : 2376635
Skype : see_moonlight

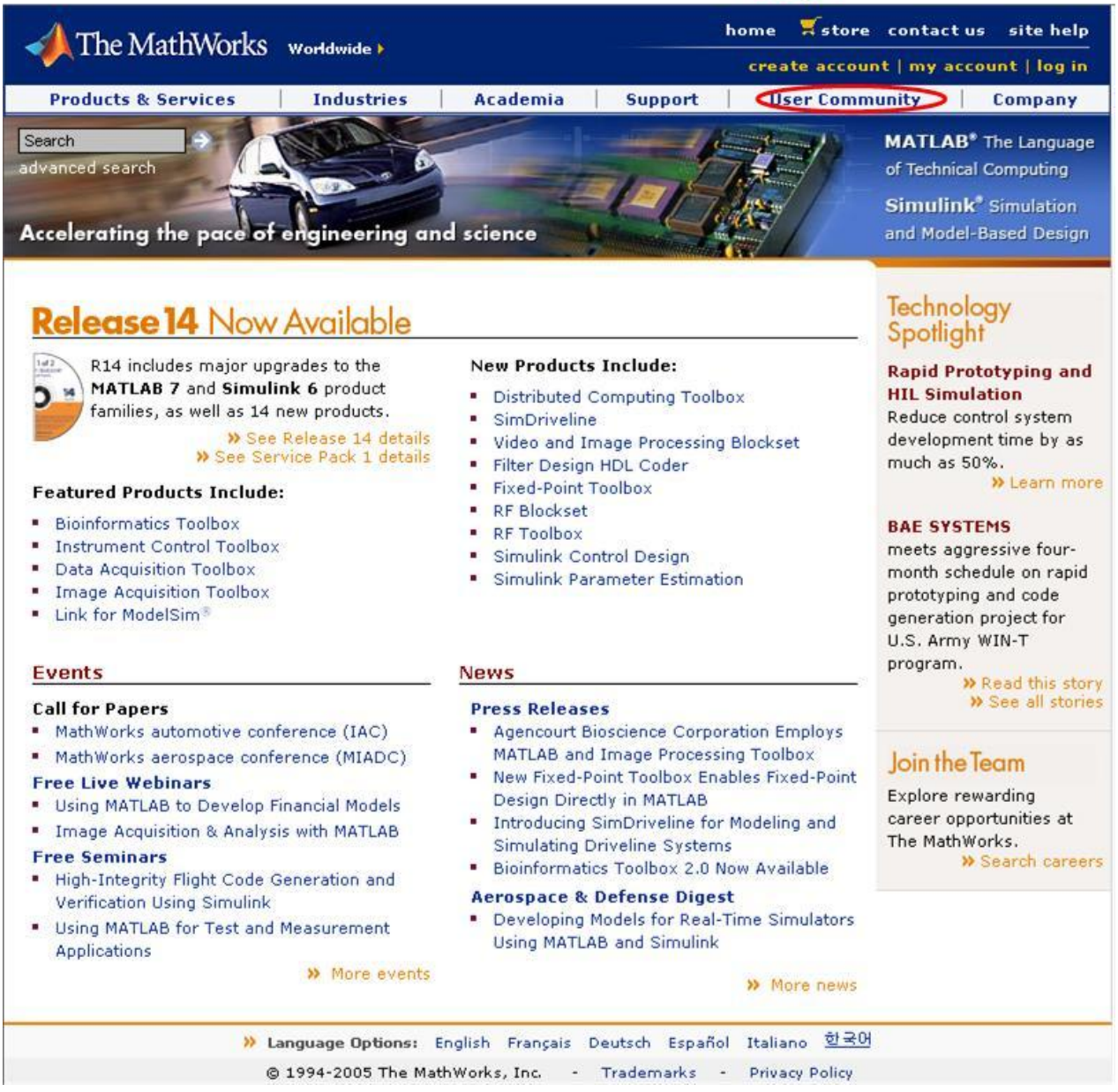
How to get SGALAB Toolbox

[1] From www.Mathworks.com

Step 1: get into www.mathworks.com, then into "User community"

http://www.mathworks.com/

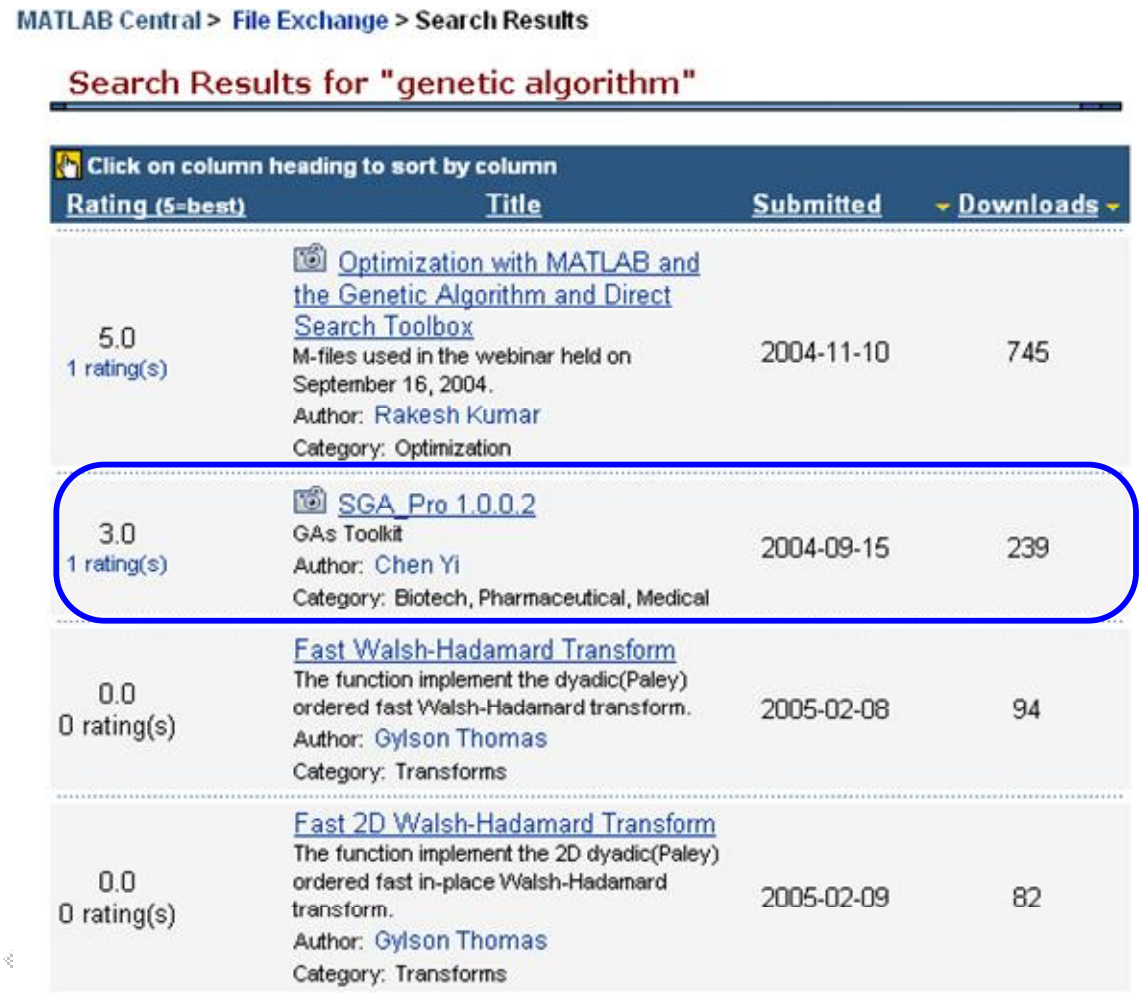
Step 1



Step 2: get into ‘‘File Exchange’’

Step 3: Search keyword of "Genetic Algorithm"

Step 4: You will find "SGALAB1002"



MATLAB Central > File Exchange > Search Results

Search Results for "genetic algorithm"

Click on column heading to sort by column

Rating (5=best)	Title	Submitted	Downloads
5.0 1 rating(s)	Optimization with MATLAB and the Genetic Algorithm and Direct Search Toolbox M-files used in the webinar held on September 16, 2004. Author: Rakesh Kumar Category: Optimization	2004-11-10	745
3.0 1 rating(s)	SGA_Pro 1.0.0.2 GAs Toolkit Author: Chen Yi Category: Biotech, Pharmaceutical, Medical	2004-09-15	239
0.0 0 rating(s)	Fast Walsh-Hadamard Transform The function implement the dyadic(Paley) ordered fast Walsh-Hadamard transform. Author: Gylson Thomas Category: Transforms	2005-02-08	94
0.0 0 rating(s)	Fast 2D Walsh-Hadamard Transform The function implement the 2D dyadic(Paley) ordered fast in-place Walsh-Hadamard transform. Author: Gylson Thomas Category: Transforms	2005-02-09	82

[2] From bbs.matwav.com

Step 1: get into "Genetic Algorithms" Block

Step 2 : to look for "SGALAB Ver1002"

[3] From www.dytrol.com

Step 1: step into “Control theory and Application” Block

Step 2 : to look for “SGALAB Ver1002”

Table of Contents

Part II
Code Analysis for Simple Genetic Algorithms Laboratory Toolbox

Inversion Evolving Crossover "Hot Spots Messy Gas

4.2 Binary En-coding
4.3 Real Number En-coding
4.4 Messy En-coding
4.5 Gray En-coding
4.6 DNA En-coding& Non-DNA coidng
4.7 User Define String

Chapter 5 Selection Operators Design

Fitness–Proportionate Selection with "Roulette Wheel" and "Stochastic Universal" Sampling

Sigma Scaling
Elitism

Boltzmann Selection

Rank Selection

Tournament Selection

Steady–State Selection

Chapter 6 Crossover Operators Design

GENETIC OPERATORS
Crossover
Mutation
Other Operators and Mating Strategies

Understanding the Role of Schemas in GAs
Understanding the Role of Crossover

Chapter 7 Mutation Operators Design

7.1 Overview

Chapter 8 Fitness Function Design

8.1 Overview

Part III
Apply Genetic Algorithms into Engineering Applications

Chapter17 Genetic Algorithms for Constrained Service Laboratory visioning

Chapter18 Using a Genetic Algorithm to Determine the Optimum Two-Impulse Transfer Between Coplanar, Elliptical Orbits

Chapter19 Optimized Non-Coplanar Orbital Transfers Using Genetic Algorithms

Chapter20 Data Mining Using Genetic Algorithms

Chapter21 Space Shuttle Main Engine Condition Monitoring Using Genetic Algorithms and Radial Basis Function Neural Network

Chapter22 Tuning Bama Optimized Recurrent Neural Networks Using Genetic Algorithms

Chapter23 Gauss-Legendre Integration Using Genetic Algorithms

Chapter24 Using Genetic Operators to Distinguish Chaotic Behavior from Noise in a Time Series

Chapter25 Development of Mobile Robot Wall-following Algorithms Using Genetic Laboratorygramming

Chapter26 Hydrocyclone Model Using Genetic Laboratorygramming

Chapter27 What Can I Do with a Learning Classifier System?

Chapter28 Genetic Algorithms for Game Playing

Chapter29 Simulation of an Artificial Eco-System Using Genetic Algorithms