

MATLAB library implementation

The MATLAB library implementation has been designed on the similar lines of the already available GA toolbox of MATLAB. The example usage is given in the file *sitodriver.m*. It contains two functions namely:

1. *function options = sitoOptimset(varargin)*

The function *sitoOptimset* should be called as

options=sitoOptimset('param1',value1,'param2',value2,...).

A structure called *options* is created with the value of 'param1' to value1, 'param2' to value2, and so on. The unspecified parameters are set to their default values. The parameter names are case insensitive. The parameters and their description have been listed in Table 1.

TABLE 1: Description of *sitoOptimset* parameters

| Parameters | Description |
|-------------------|---|
| PopulationType | The type of Population being entered There are two options in the current release: 'doubleVector', 'bitstring'. The default value is 'bitstring'. |
| PopInitRange | Initial range of values of a population. The default value is [0:1]. For variant 'CODO', the value can be [Xmin, Xmax]. |
| SocietySize | Positive scalar indicating the Society Size. The user should input a positive scalar number indicating row or column of square topology. e.g. SocietySize of 7 indicates a society of 49 individuals in 7 x 7 matrix. |
| NeighbourhoodSize | It is the size of Moore neighbourhood. The value can be any positive integer and should be less than SocietySize. |
| DiversityFactor | It is the probability k with which an individual may change its attitude. The value may range between [0:1]. For variant 'CODO', the value can be any real no. |
| InitialPopulation | It is a matrix representing the initial attitudes of the society. This matrix could be a 3D matrix of size [SocietySize x SocietySize x length of 'bitstring'] or a 2D matrix of size [SocietySize2 x length of 'bitstring']. If not supplied, a random 3D matrix is generated. |
| CreationFcn | Handle to a user defined function for the generation of initial population. |
| Display | If set 'on', plots of iteration vs. fitness values (i.e. min, max and average) and society's strength colour map is displayed. |
| MaxIteration | Positive scalar indicating the maximum iterations desired. The default value is 100. |
| Variant | Various binary variants of SITO included in current release are "Osito/SsitoSum/SsitoMean/Gsito/CODO" |
| Group | If variant used is "Gsito" then the user can specify the Group. The average neighborhood size of individuals is 5. |

2. *function [x, fVal] = Sito(funcnt, nvars, options)*
function [x, fVal] = Sito(funcnt, nvars)

Sito function is called and it takes the parameters values as specified in options structure. If the options structure is unspecified than solver takes **default values**.

INPUT : *funcnt*: A function handle to the fitness function.

nvars: Positive integer which represents the number of variables in the problem.

OUTPUT: *x*: Best individual's attitudes obtained after iterations completion.

fVal: the value of the fitness function at *x*.

2.1 Example of a binary optimization function (Hamming function):

y = Hamming(x)

INPUT : *x* accepts binary vector of an individual's attitudes.

Dimensions of *x* (No_of_Features x 1)

OUTPUT: *y* is the scalar (fitness value) evaluated at *x*.

2.2 Example of a real valued optimization function :

y = rastriginfn(x)

INPUT : *x* accepts binary vector of an individual's attitudes

OUTPUT: *y* is the scalar (fitness value) evaluated at *x*.

SITO is a binary optimizer which optimizes the binary optimization problems. For a real valued optimization problem, the individual's attitudes (No_of_Features x 1) are converted to codewords using *codeWordfn*. Like in Rastriginfn if No_of_Features = 34 then codewords = 2 with each codeword size =17. These codewords (2) are encoded to real values (2) using *bin2real* function. After this scalar (fitness value) is evaluated at real values (2).

A sample implementation has been provided in the sitodriver.m file.

Anonymous Objective Function:

In SITO, there are seven different benchmark functions included. If a user wants to use some different objective function than user must first write a MATLAB file for an anonymous function that you want to optimize. The function should accept a vector whose length is the number of features representing individual's attitudes, and should return a scalar.

Minimizing Fitness Function:

SITO minimize the objective or fitness function. If user wants to use an anonymous objective function than write function file to compute function. The problem is of type minimize $f(x)$. If problem is to maximize $f(x)$ than minimize $-f(x)$ is used because the point at which the minimum of $-f(x)$ occurs is the same as that of $f(x)$.

Fitness Functions:

The benchmark functions are designed as:

- $function\ y = fitnessfcn(x)$

INPUT : x accepts binary vector of an individual's attitudes. Dimensions of x (No_of_Features x 1)

OUTPUT: y is the scalar (fitness value) evaluated at x.

The algorithms developed in SITO LIB are employed and tested on seven benchmark problems.

- **OneMax**

The OneMax or one maximum function counts the number of 1s in the string (formed by combining the binary value of each dimension of single individual). The function locally searches for the individual with maximum number of one's.

- **Hamming**

Hamming function computes the hamming distance between individual's attitudes with the default string specified. The function tries to minimize the hamming distance.

- **Order3Deceptive**

Goldberg's Order3Deceptive problem computes number of ones in 3-bit substring. Fitness is computed using mapping table which maps substring to value as below:

| String | Mapped value |
|--------|--------------|
| 000 | 28 |
| 001 | 26 |
| 010 | 22 |
| 011 | 0 |
| 100 | 14 |
| 101 | 0 |
| 110 | 0 |
| 111 | 30 |

Usage Description:

The user should provide number of features as multiple of three as the problem is of order 3. Otherwise the error message is displayed on screen. So, *sizeCodeword* is set to 3 and should not be altered by user.

- **Bipolar**

In multimodal deceptive problem of order 6, the fitness is calculated as the sum of the partial fitness functions. In this problem, only number of ones is computed in the binary substring. The function f_6 is applied to the disjoint six-bit substrings of the string to compute the fitness of an individual.

Formula used for computing fitness is:

$$f_6(u) = f_3(|3 - u|)$$

$$\text{where } f_3(0,1,2,3) = \{.9, .8, 0, 1\}$$

u is the number of ones in six-bit substring.
 f_3 is the Order3Deceptive function.

Usage Description:

The Bipolar is order-6 deceptive problem so the user should set number of features as multiple of six. Otherwise the error message is displayed on screen. So, *sizeCodeword* is set to 6 and should not be altered by user.

- **Ecc**

In Error correcting code (Ecc) design problem, the transmitting message is partitioned into codewords of fixed length. Further, each codeword is assigned to the alphabet which minimizes the minimal hamming distance of the codewords. The objective function to be minimized is:

$$f = \frac{1}{\sum_{i=1}^n \sum_{j=1}^n \frac{1}{\delta^2}}$$

n is the number of codewords in message.
 δ is the hamming distance between i^{th} and j^{th} codeword.

Usage Description:

In *Ecc*, n codewords are formed according to the size specified in *sizeCodeword*. In this paper, *sizeCodeword* is set to 12. The user can change *sizeCodeword* as per requirement. The number of features should be multiple of specified size of codeword otherwise error message is displayed.

- **MaxCut**

In Maximum Cut graph problem, cut operation on the undirected, weighted graph leads to partition of vertices of graph into two disjoint subsets of vertices such that sum of weighted edges having endpoints in different subsets is maximized. The formula for computing fitness is given by:

$$f = \sum_{i=1}^{n-1} \sum_{j=i+1}^n w_{ij} \cdot [x_i(1 - x_j) + x_j(1 - x_i)]$$

w_{ij} is the weight of edge with endpoints $\{i, j\}$. w_{ij} participate in summation only if i and j belong to different subsets.

- **Rastrigin**

Rastrigin problem is scalable and highly multimodal. This problem has large number of local minima. Rastrigin function is defined as:

$$f = 10 \cdot n + \sum_{i=1}^n x_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot x_i)$$

where $-5.12 \leq x_i \leq 5.12$

n is the number of features
x is the real value