Genetic Algorithm Toolbox

Test Functions

Table of Contents

1. Introduction

This document describes a number of test functions implemented for use with the Genetic Algorithm Toolbox for MATLAB. These functions are drawn from the literature on genetic algorithms, evolutionary strategies and global optimization. The first Section describes a set of common parametric test problems implemented as MATLAB m-files. The second Section presents a number of dynamic optimization problems, implemented in SIMULINK, as s-files and m-files as appropriate.

No.	function name	description
1	`objfun1`	De Jong's function 1
2	`objfun1a`	axis parallel hyper-ellipsoid
3	`objfun1b`	rotated hyper-ellipsoid
4	`objfun2`	Rosenbrock's valley (banana function)
5	`objfun6`	Rastrigins's function
6	`objfun7`	Schwefel's function
7	`objfun8`	Griewangk's function
8	`objfun9`	sum of different power
9	`objdopi`	double integrator
10	`objharv`	harvest problem
11	`objlinq`	linear-quadratic problem (discret function)
12	`objlinq2`	linear-quadratic problem (continious function)
13	`objpush`	push-cart problem

Table 1: Set of available test functions

All of the test function implementations are scalable, i.e. the dimension of the test functions are adjustable via a single parameter value inside the function.

2. Parametric Optimization

2.1. De Jong's function 1

The simplest test function is De Jong's function 1. It is continuos, convex and unimodal.

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \qquad -5{,}12 \le x_i \le 5{,}12$$

global minimum: $x_i = 0 \qquad f(x) = 0$

This function is implemented in the m-file `objfun1.m`.

2.2. Axis parallel hyper-ellipsoid

The axis parallel hyper-ellipsoid is similar to De Jong's function 1. It is also known as the weighted sphere model. Again, it is continuos, convex and unimodal.

$$f_{1a}(x) = \sum_{i=1}^{n} i \cdot x_i^2 \qquad -5{,}12 \le x_i \le 5{,}12$$

global minimum: $x_i = 0 \qquad f(x) = 0$

This function is implemented in the m-file `objfun1a.m`.

2.3. Rotated hyper-ellipsoid

An extension of the axis parallel hyper-ellipsoid is Schwefel's function1.2. With respect to the coordinate axes, this function produces rotated hyper-ellipsoids. It is continuos, convex and unimodal.

$$f_{1b}(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2 \qquad -65{,}536 \le x_i \le 65{,}536$$

global minimum: $x_i = 0 \qquad f(x) = 0$

This function is implemented in the m-file `objfun1b.m`.

2.4. Rosenbrock's valley (De Jong's function 2)

Rosenbrock's valley is a classic optimization problem. The global optimum is inside a long, narrow, parabolic shaped flat valley. To find the valley is trivial, however convergence to the global optimum is difficult and hence this problem has been repeatedly used in assess the performance of optimization algorithms.

$$f_2(x) = \sum_{i=1}^{n-1} 100 \cdot \left(x_{i+1} - x_i^2\right)^2 + \left(1 - x_i\right)^2 \qquad -2,048 \le x_i \le 2,048$$

global minimum: $x_i = 1$ $f(x) = 0$

This function is implemented in the m-file `objfun2.m`.

2.5. Rastrigin's function 6

This function is based on function 1 with the addition of cosine modulation to produce many local minima. Thus, the test function is highly multimodal. However, the location of the minima are regularly distributed.

$$f_6(x) = 10 \cdot n + \sum_{i=1}^{n} \left(x_i^2 - 10 \cdot \cos\left(2 \cdot \pi \cdot x_i\right)\right) \qquad -5,12 \le x_i \le 5,12$$

global minimum: $x_i = 0$ $f(x) = 0$

This function is implemented in the m-file `objfun6.m`.

2.6. Schwefel's function 7

Schwefel's function is deceptive in that the global minimum is geometricaly distant, over the parameter space, from the next best local minima. Therefore, the search algorithms are potentially prone to convergence in the wrong direction.

$$f_7(x) = \sum_{i=1}^{n} -x_i \cdot \sin\left(\sqrt{|x_i|}\right) \qquad -500 \le x_i \le 500$$

global minimum: $x_i = 420,9687$ $f(x) = n \cdot 418,9829$

This function is implemented in the m-file `objfun7.m`.

2.7. Griewangk's function 8

Griewangk's function is similar to Rastrigin's function. It has many widespread local minima. However, the location of the minima are regularly distributed.

$$f_8(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \qquad -600 \le x_i \le 600$$

global minimum: $x_i = 0 \qquad f(x) = 0$

This function is implemented in the m-file `objfun8.m`.

2.8. Sum of different Powers (function 9)

The sum of different powers is a commonly used unimodal test function.

$$f_9(x) = \sum_{i=1}^{n} |x_i|^{(i+1)} \qquad -1 \le x_i \le 1$$

global minimum: $x_i = 0 \qquad f(x) = 0$

This function is implemented in the m-file `objfun9.m`.

3. Dynamic optimization

Dynamic control problems are complex and difficult to solve. The use of dynamic-optimization specific methods, such as Hamiltonian, is complicated and problematic. The application of specific methods requires a large amount of mathematical support even for systems of moderate size, and only the most trivial systems can be solved analytically.

In the following example problems, each individual in the evolutionary algorithm corresponds to a (discrete) control vector. Each variable in an individual is associated with the control input at a time step of the dynamic optimization problem. In this section, x is the state vector and u the control vector of a system.

3.1. Double integrator

The double integrator problem is described by the following state equations.

$$\dot{x}_1 = u$$
$$\dot{x}_2 = x_1$$
$$y = x_2$$

The value of x_2 has to be changed during a time period with as little control effort as possible and the final conditions must be met, such that the following criteria are satisfied.

- time: $0 \le t \le 1$
- initial conditions: $x_1(0) = 0$ $x_2(0) = -1$
- final conditions: $x_1(1) = 0$ $x_2(1) = 0$

The objective function to be minimized is:

- $f(u) = \int_{t=0}^{1} u(t)^2 \, dt$.

For these conditions an analytical solution is found to be:

- $u = 6 - 12 \cdot t$, Minimum $= 12$.

Figure 1 shows the optimal control vector and states for the continuous system:

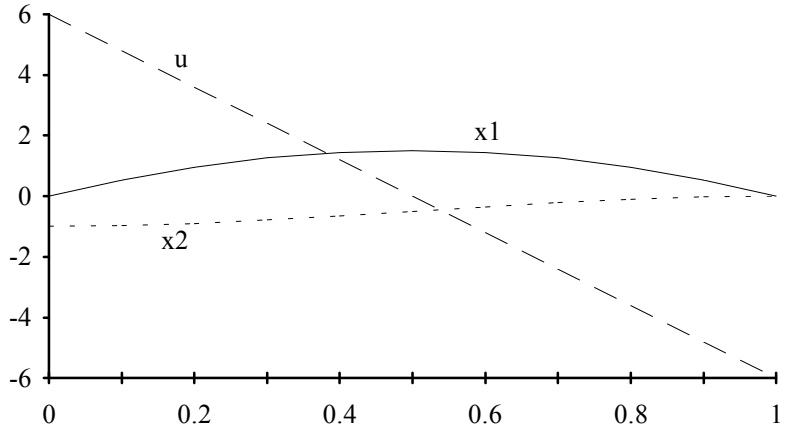

Fig. 1: Input and states of double integrator for optimal solution

The double integrator is implemented in the m-file `objdopi.m` using a SIMULINK model, an s-function and *Control System Toolbox* routines. The implementation used by the objective function is chosen through an optional parameter inside the function, the default is the SIMULINK model.

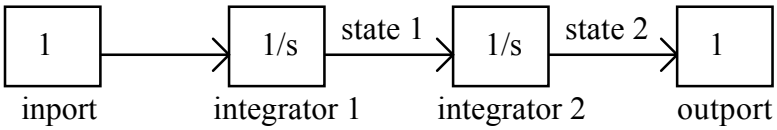

Fig. 2: Block diagram of double integrator

The SIMULINK method uses the model in figure 2 (`simdopi1.m`). The s-function solves the linear equations (`simdopi2.m`) given above and the third method (*Control System Toolbox*) uses the transfer function:

$$G(s) = \frac{1}{s^2}.$$

3.2. Linear-quadratic system

The linear-quadratic system is one-dimensional:

$$x(k+1) = a \cdot x(k) + b \cdot u(k) \qquad k = 1, 2, \ldots, \mathrm{N}.$$

The objective function for minimization is defined as:

$$f(x,u) = q \cdot x(N+1)^2 + \sum_{k=1}^{N} \left(s \cdot x(k)^2 + r \cdot u(k)^2 \right).$$

set	N	x(0)	s	r	q	a	b	exact solution
1	45	100	1	1	1	1	1	16180.3399
2	45	100	10	1	1	1	1	109160.7978
3	45	100	1000	1	1	1	1	10009990.0200
4	45	100	1	10	1	1	1	37015.6212
5	45	100	1	1000	1	1	1	287569.3725
6	45	100	1	1	0	1	1	16180.3399
7	45	100	1	1	1000	1	1	16180.3399
8	45	100	1	1	1	0.01	1	10000.5000
9	45	100	1	1	1	1	0.01	431004.0987
10	45	100	1	1	1	1	100	10000.9999

Table 2: Parameter sets for linear-quadratic syste

The ten test cases described by Michalewicz [Mic92] are all implemented in the m-file `objlinq.m`. The values of the parameter sets are shown in table 2. The solution obtained for the first test case is shown in figure 3.

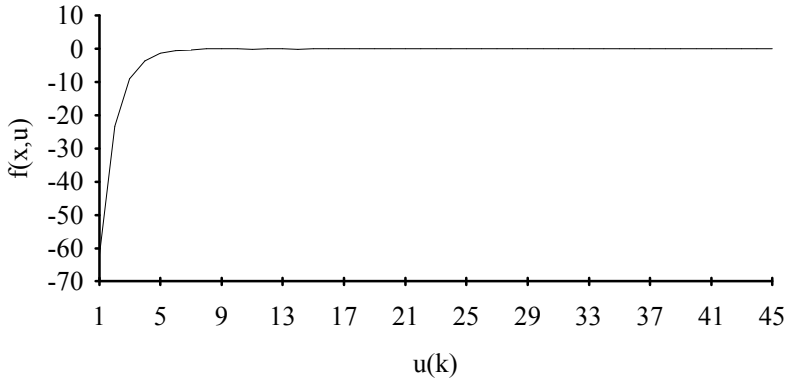

Fig. 3: Optimal control vector for the linear-quadratic system set 1

The linear-quadratic system is identical to a single integrator with positive feedback. A continuous version of this problem using a SIMULINK model, an s-function and *Control System Toolbox* routines is implemented in the m-file `objlinq2.m`. The implementation used by the objective function is chosen through an optional parameter inside the function, the default is the SIMULINK model.

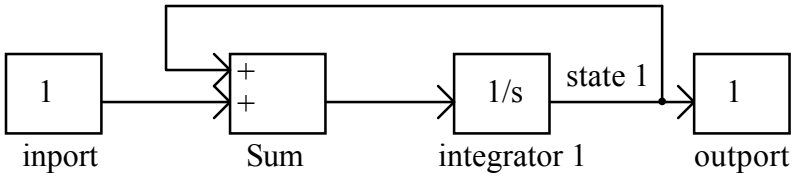

Fig. 4: Block diagram of linear-quadratic system

The SIMULINK method uses the model in figure 4 (`simlinq1.m`). The s-function solves the linear equations (`simlinq2.m`):

$$\dot{x}_1 = x_1 + u$$
$$y = x_1 .$$

and the third method (*Control System Toolbox*) uses the transfer function:

$$G(s) = \frac{1}{s-1}.$$

3.3. Harvest system

The harvest system is a one-dimensional equation of growth with one constraint:

$$x(k+1) = a \cdot x(k) - u(k) \qquad k = 1,2,\ldots,\mathrm{N},$$

such that $x(0) = x(N)$.

The objective function for minimization is therefore defined as:

$$f(u) = -\sum_{k=1}^{N} \sqrt{u(k)}\,.$$

The exact solution can be analytically found by:

$$\text{Minimum} = -\sqrt{\frac{x(0)\cdot\left(a^N - 1\right)^2}{a^{N-1}\cdot(a-1)}}\,.$$

Figure 5 shows the control vector for the harvest system with N=20.

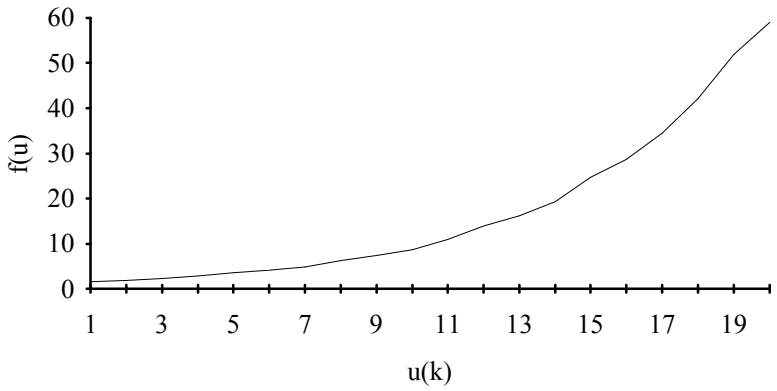

Fig. 5: Optimal control vector for the harvest system with N=20

This function is implemented in the m-file `objharv.m`.

3.4. Push-cart system

The push-cart system is a two-dimensional system described by the following equations:

$$x_1(k+1) = x_2(k) \qquad k = 1,2,\ldots,\mathrm{N},$$

$$x_2(k+1) = 2 \cdot x_2(k) - x_1(k) + \frac{1}{N^2} u(k).$$

The objective function for minimization is therefore defined as:

$$f(x,u) = -x_1(N+1) + \frac{1}{2 \cdot N} \sum_{k=1}^{N} u^2(k).$$

The exact solution can be analytically found by:

$$\text{Minimum} = -\frac{1}{3} + \frac{3 \cdot N - 1}{6 \cdot N^2} + \frac{1}{2 \cdot N^3} \sum_{k=1}^{N-1} k^2.$$

Figure 6 shows the control vector for the push-cart system with N=20.

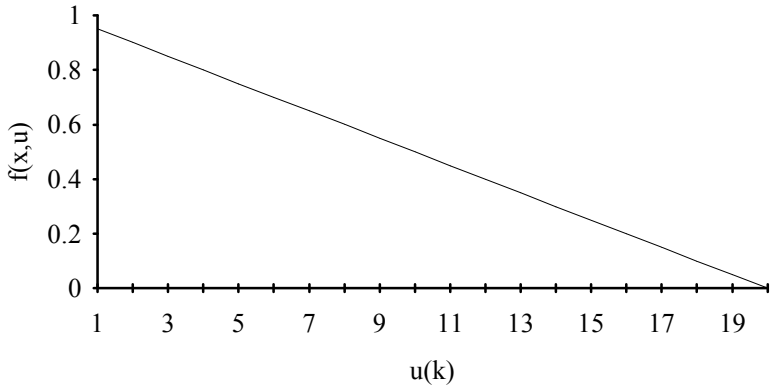

Fig. 6: Optimal control vector for the push-cart system with N=20

This function is implemented in the m-file `objpush.m`.

4. References

[Mic92] Michalewicz, Z.: "Genetic Algorithms + Data Structures = Evolution Programs"; Berlin, Heidelberg, New York: Springer, 1992

[MSB91] Mühlenbein, H., Schomisch, M. and Born, J.: "The parallel genetic algorithm as function optimizer"; Parallel Computing, 17, pp.619-632, 1991

[OGH93] Ostermeier, A., Gawelczyk, A. and Hansen, N.: "A Derandomized Approach to Self Adaptation of Evolution Strategies"; Technical Report TR-93-003, TU Berlin, 1993

[Rec73] Rechenberg, I.: "Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution"; Stuttgart: Frommann-Holzboog, 1973

[Rec94] Rechenberg, I.: "Evolutionsstrategie 94"; Stuttgart: Frommann-Holzboog, 1994

[Sch81] Schwefel, H.-P.: "Numerical optimization of computer models"; Chichester: Wiley & Sons, 1981

[SE92] Stuckmann, B. E.; Easom, E. E.: "A Comparison of Bayesian/Sampling Global Optimization Techniques"; IEEE Transactions on Systems, Man, and Cybernetics, Vol. 22, No. 5, pp.1024-1032, 1992

[SHF94] Schöneburg, E., Heinzmann, F. and Feddersen, S.: "Genetische Algorithmen und Evolutionsstrategien"; Bonn, Paris, Reading, Mass.: Addison-Wesley, 1994

[TZ89] Törn, A. and Zilinskas, A.: "Global Optimization"; volume 350 of Lecture Notes in Computer Science, Berlin, Heidelberg, New York: Springer, 1989