

Reporte Tarea 5 Algoritmo Evolutivos.

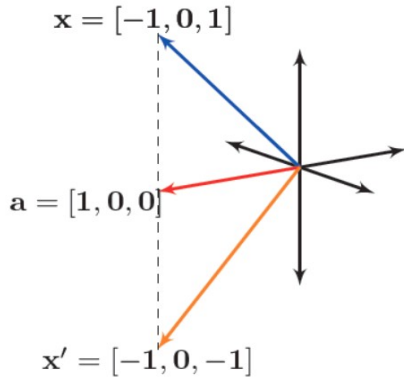
An evolutionary algorithm using spherical inversions.

Introducción.

En esta práctica se implementa el algoritmo evolutivo propuesto en el artículo “An evolutionary algorithm using spherical inversions” y se analiza su desempeño contra un conjunto de funciones de prueba conocido.

El algoritmo propuesto está basado en el enfoque geométrico de los mecanismos de reproducción del algoritmo PSO (Particle Swarm Optimization), y del algoritmo de Evolución Diferencial (DE). Básicamente consiste en la aplicación de dos operadores geométricos de búsqueda, a saber, ISO (Inversion Search Operator) y RSO (Reflection Search Operator). La estructura básica del Spherical Evolutionary Algorithm (SEA) se describe en el Algoritmo 1:

Aquí la población inicial se asigna de manera aleatoria. El operador de reflexión realiza una transformación sobre un individuo proyectándolo con respecto a una base canónica, como se ilustra en la siguiente figura:



Reflections of the vector \mathbf{x} with respect to the vector \mathbf{a}

Esta reflexión se logra multiplicando el individuo \mathbf{x} por una matriz \mathbf{A} que se compone de la siguiente manera:

$$\mathbf{A} = \begin{pmatrix} \alpha_1 & 2a_1a_2 & \cdots & 2a_1a_{n-1} & 2a_1a_n \\ 2a_2a_1 & \alpha_2 & \cdots & 2a_2a_{n-1} & 2a_2a_n \\ 2a_3a_1 & 2a_3a_2 & \ddots & 2a_3a_{n-1} & 2a_3a_n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 2a_{n-1}a_1 & 2a_{n-1}a_2 & \cdots & \alpha_{n-1} & 2a_{n-1}a_n \\ 2a_na_1 & 2a_na_2 & \cdots & 2a_na_{n-1} & \alpha_n \end{pmatrix}$$

$$\alpha_j = a_j^2 - \sum_{i \neq j}^n a_i^2.$$

Algorithm 1 Pseudo-code of the SEA.

```

procedure SEA
    /*  $d$ : dimension of the problem
        $ps$ : population size
        $\eta$ : Number of hyperspheres for inverse points */
    /*1. Initialize a population */
     $Pop \leftarrow \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(ps)}\}$ 
    where the individual  $\mathbf{x}^{(i)} \in \mathbb{R}^d$ 
     $t \leftarrow 0$ 
    while not stopping condition do
         $\mathcal{P} \leftarrow$  Select the best  $\eta$  individuals.
        for  $i \leftarrow 1$  to  $ps$  do
            /*2. Select a center of the hypersphere*/
            repeat
                 $\mathbf{c} \leftarrow$  select randomly one from  $\mathcal{P}$ 
            until  $\mathbf{c} \neq \mathbf{x}^{(i)}$ 
            if  $U(0, 1) \geq 0.5$  then
                /*3. Inversion Search Operator*/
                 $\mathbf{y}^{(i)} \leftarrow \text{ISO}(\mathbf{c}, \mathbf{x}^{(i)})$ 
            else
                /*4. Reflection Search Operator*/
                 $\mathbf{y}^{(i)} \leftarrow \text{RSO}(\mathbf{c}, \mathbf{x}^{(i)})$ 
            end if
        end for
        /*5. Replace the individuals */
        for  $i \leftarrow 1$  to  $ps$  do
            if  $\text{fitness}(\mathbf{x}^{(i)}) > \text{fitness}(\mathbf{y}^{(i)})$  then
                 $\mathbf{x}^{(i)} \leftarrow \mathbf{y}^{(i)}$ 
            end if
        end for
         $t \leftarrow t + 1$ 
    end while
end procedure

```

El operador de Inversión tiene el propósito de proyectar hacia afuera los puntos que quedan dentro de las hiper-esferas que se generan al seleccionar los centros, y viceversa. La hiper-esfera determina la región del espacio donde potencialmente se puede encontrar un mínimo.

La inversión de un punto con respecto a una hiper-esfera en notación euclidiana esta definida por la siguiente expresión:

$$\hat{\mathbf{x}} = \frac{r^2}{\|\mathbf{c} - \mathbf{x}\|} \frac{(\mathbf{c} - \mathbf{x})}{\|\mathbf{c} - \mathbf{x}\|} + \mathbf{c}.$$

Los factores β_1 y β_2 en el algoritmo se utilizan para restringir la inversión a una región aceptable.

Los operadores de Reflexión e Inversión son descritos en los algoritmos 2 y 3 respectivamente.

Algorithm 2 Pseudo-code of the Reflection Search Operator

```

procedure RSO(c,x)
  y ← c - x
  a ← Select randomly one canonical component
        according to the problem dimensionality.
  Create the matrix A with the vector a.
  return x' ← Ay
end procedure

```

Algorithm 3 Pseudo-code of the Inverse Search Operator.

```

procedure ISO(c, x)
  /*1. Mutation for the center*/
  if U(0,1) ≥ 0.5 then
    k ← Select randomly one component of the vector
          which corresponds to the center of the
          hypersphere.
    c_k ← c_k + N(0,1) //Add a normal random number.
  end if
  /*2. Calculate the radius of the hypersphere*/
  rand ← U(0,2) //Obtain an Uniform random variable
            in the interval (0,2).
  r ← (rand) (|| c - x ||2)
  if || c - x || > 0 then
    /*3. Compute the inverse point*/
    x̂ ← ( (r2 / ||c-x||) ) ( (c-x) / ||c-x|| ) + c
    /*4. Calculate the acceptable regions and identify
        when inverse individual is inside or outside
        of the acceptable region*/
    β1 ← r (e-1/r); β2 ← r (e1/r)
    if || c - x̂ || < r then
      x̂ ← β1 ( (c-x̂) / ||c-x̂|| ) + c
    else if || c - x̂ || > r then
      x̂ ← β2 ( (c-x̂) / ||c-x̂|| ) + c
    end if
  else
    x̂ ← c //The center of the hypersphere is returned
          when c = x.
  end if
  return x̂
end procedure

```

Resultados.

Los resultados obtenidos fueron en base a un benchmark de funciones unimodales y multimodales:

Sphere	$f_{sph}(\vec{x}) = \sum_{i=1}^N x_i^2$	$[-100, 100]^{30}$
Elliptic	$f_{ell}(\vec{x}) = \sum_{i=1}^N (10^6)^{\left(\frac{i-1}{N-1}\right)} x_i^2$	$[-100, 100]^{30}$
Schwefel 1.2	$f_{sch}(\vec{x}) = \sum_{i=1}^N \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^{30}$
Ackley	$f_{ack}(\vec{x}) = 20 + e - 20e^{\left(-0.2\sqrt{\left(\frac{1}{N}\sum_{i=1}^N x_i^2\right)}\right)} - e^{\left(\frac{1}{N}\sum_{i=1}^N \cos(2\pi x_i)\right)}$	$[-32, 32]^{30}$
Rastrigin	$f_{ras}(\vec{x}) = 10N + \sum_{i=1}^N (x_i^2 - 10 \cos(2\pi x_i))$	$[-5.12, 5.12]^{30}$
Griewank	$f_{grw}(\vec{x}) = \sum_{i=1}^N x_i^2 / 4000 - \prod_{i=1}^N \cos(x_i / \sqrt{i}) + 1$	$[-600, 600]^{30}$
Rosenbrock	$f_{ros}(\vec{x}) = \sum_{i=1}^{N-1} (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2)$	$[-100, 100]^{30}$
Weierstrass	$f_{wrs}(\vec{x}) = \sum_{i=1}^N \left(\sum_{k=0}^{k_{max}} (a^k \cos(2\pi b^k (x_i + 0.5))) \right) - N \sum_{k=0}^{k_{max}} (a^k \cos(\pi b^k))$	$[-0.5, 0.5]^{30}$
Schaffer	$f_{scf}(\vec{x}) = \sum_{i=1}^N F(x_i, x_{i+1}); \quad x_{N+1} = x_1$ where, $F(x, y) = 0.5 + \frac{\sin^2(\sqrt{x^2 + y^2}) - 0.5}{(1 + 0.001(x^2 + y^2))^2}$	$[-0.5, 0.5]^{30}$
Salomon	$f_{sal}(\vec{x}) = 1 - \cos(2\pi \mathbf{x}) + 0.1 \mathbf{x} $ where, $ \mathbf{x} = \sqrt{\left(\sum_{i=1}^N x_i^2\right)}$	$[-100, 100]^{30}$

La siguiente tabla muestra los resultados obtenidos para las diferentes funciones del benchmark, el tamaño de la población se fijó a 129 y el número de hiper-esferas en 9, tal como se indica en el artículo:

SEA							
Function	Dim	Fmin	Fmax	Favg	Fmedian	FstdDev	S_ratio
Sphere	30	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	100
Sphere	50	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	100
Elliptic	30	0.00E+000	1.20E+007	7.85E+005	1.97E+004	2.04E+006	32
Elliptic	50	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	100
Schwefel	30	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	100
Schwefel	50	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	100
Ackley	30	4.44E-016	4.44E-016	4.44E-016	4.44E-016	0.00E+000	100
Ackley	50	4.44E-016	4.44E-016	4.44E-016	4.44E-016	0.00E+000	100
Rastrigin	30	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	100
Rastrigin	50	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	100
Griewank	30	0.00E+000	4.26E+000	1.00E+000	9.60E-001	9.85E-001	22
Griewank	50	0.00E+000	1.05E+001	8.96E-001	1.03E-001	1.71E+000	42
Rosenbrock	30	2.74E+001	2.82E+001	2.79E+001	2.80E+001	1.86E-001	0
Rosenbrock	50	4.72E+001	4.81E+001	4.77E+001	4.77E+001	1.73E-001	0
Weierstrass	30	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	100
Weierstrass	50	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	100
Schaffer	30	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	100
Schaffer	50	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	100
Salomon	30	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	100
Salomon	50	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	100

Conclusiones.

Al comparar los resultados obtenidos con los que se publicaron artículos se concluye que en casi todas las instancias que se presentan en el artículo el resultado publicado coincide con el resultado que se obtuvo en esta práctica, con excepción de la función de Rosenbrock, donde, como se aprecia, se tiene un 0% de éxito en esta implementación, aunque se tienen dudas de que la función que aquí se utilizó sea la misma que se propone en el artículo, ya que los mínimos globales son distintos. Por último, en la siguiente tabla se compara el algoritmo esférico con una versión mejorada de Evolución Diferencial y con los PSO implementados en la práctica pasada:

Function	Dim	SEA		SaNSDE		PSO Complete		PSO Ring		PSO Dynamic	
		Favg	S_ratio	Favg	S_ratio	Favg	S_ratio	Favg	S_ratio	Favg	S_ratio
Sphere	30	0.00E+000	100	9.21E-009	100	4.25E-155	100	2.78E-076	100	1.12E-151	100
Sphere	50	0.00E+000	100	1.21E+000	0	6.79E-089	100	7.20E-073	100	9.82E-152	100
Elliptic	30	7.85E+005	32	1.57E+002	0	1.03E-139	100	1.72E-072	100	2.59E-147	100
Elliptic	50	0.00E+000	100	9.77E+004	0	8.23E-082	100	8.35E-070	100	7.20E-145	100
Schwefel	30	0.00E+000	100	8.85E-009	100	5.76E-023	100	1.07E-008	62	5.56E-036	100
Schwefel	50	0.00E+000	100	5.19E+000	0	2.44E-009	96	5.87E+000	0	4.66E-037	100
Ackley	30	4.44E-016	100	3.10E-002	96.66	2.22E+000	10	7.48E-015	100	6.63E-015	100
Ackley	50	4.44E-016	100	5.24E+000	0	6.66E+000	0	4.11E-002	96	4.44E-016	100
Rastrigin	30	0.00E+000	100	2.28E+001	0	7.56E+001	0	7.14E+001	0	0.00E+000	100
Rastrigin	50	0.00E+000	100	5.56E+001	0	1.79E+002	0	1.74E+002	0	0.00E+000	100
Griewank	30	1.00E+000	22	5.25E-003	60	3.89E-002	26	1.63E-003	84	0.00E+000	100
Griewank	50	8.96E-001	42	5.38E-001	0	3.58E-001	2	5.42E-004	94	0.00E+000	100
Rosenbrock	30	2.79E+001	0	2.74E+001	0	3.95E+001	0	4.36E+001	0	4.36E+001	0
Rosenbrock	50	4.77E+001	0	8.80E+002	0	1.70E+001	0	1.28E+002	0	1.28E+002	0
Weierstrass	30	0.00E+000	100	5.60E-001	0	6.51E+000	0	8.42E+000	0	0.00E+000	100
Weierstrass	50	0.00E+000	100	7.46E+000	0	2.13E+001	0	2.61E+001	0	0.00E+000	100
Schaffer	30	0.00E+000	100	9.00E-009	100	1.17E-003	96	0.00E+000	100	0.00E+000	100
Schaffer	50	0.00E+000	100	6.28E-005	0	7.77E-004	96	0.00E+000	100	0.00E+000	100
Salomon	30	0.00E+000	100	1.57E-001	0	6.68E-001	0	2.46E-001	0	7.33E-003	70
Salomon	50	0.00E+000	100	7.03E-001	0	1.85E+000	0	3.60E-001	0	2.87E-003	90

Compilación/Ejecución.

El programa implementado incluye un makefile para compilarse, que soporta los comandos *make*, *make run* y *make clean*, así como un script de ejecución en el cluster para correr todas las instancias, los resultados se generan en un archivo para cada ejecución.

El programa reciben cuatro parámetros:

1. Tamaño de la población (aquí se utilizó $N=129$).
2. Dimensión de las variables (se utilizó 30 y 50).
3. Numero de hiper-esferas (aquí se utilizó $\eta=9$).
4. Referencia a la función objetivo a evaluar:
 - [0] = Sphere;
 - [1] = Elliptic;
 - [2] = Schwefel;
 - [3] = Ackley;
 - [4] = Rastrigin;
 - [5] = Griewank;
 - [6] = Rosenbrock;
 - [7] = Weierstrass;
 - [8] = Schaffer;
 - [9] = Salomon;

Compilación del script:

```
almeida@almeida-X501A1: ~/SEA
almeida@almeida-X501A1:~$ cd /home/almeida/SEA
almeida@almeida-X501A1:~/SEA$ g++ -Wall -o script script.cpp
almeida@almeida-X501A1:~/SEA$
```

Compilación del programa:

```
almeida@almeida-X501A1: ~/SEA
almeida@almeida-X501A1:~$ cd /home/almeida/SEA
almeida@almeida-X501A1:~/SEA$ make
g++ -std=c++11 -fopenmp -c src/functions.cpp -o obj/functions.o
g++ -std=c++11 -fopenmp -c src/main.cpp -o obj/main.o
g++ -std=c++11 -fopenmp -c src/memo.cpp -o obj/memo.o
g++ -std=c++11 -fopenmp -c src/sea.cpp -o obj/sea.o
g++ -std=c++11 -fopenmp -c src/vec.cpp -o obj/vec.o
g++ -std=c++11 -o bin/ejecutable obj/functions.o obj/main.o obj/memo.o obj/sea.o obj/vec.o -lgomp
almeida@almeida-X501A1:~/SEA$
```

Ejecución del programa:

```
almeida@almeida-X501A1: ~/SEA
almeida@almeida-X501A1:~$ cd /home/almeida/SEA
almeida@almeida-X501A1:~/SEA$ g++ -Wall -o script script.cpp
almeida@almeida-X501A1:~/SEA$ make run args="129 30 9 0 >output.dat"
```