



Desarrollo de Algoritmos Dirigido por Retos

Alejandro Marrero Díaz

Universidad de La Laguna

10 de julio de 2017

Escuela Superior de Ingeniería y Tecnología
Universidad de La Laguna

- 1 Introducción
- 2 Conceptos Previos
- 3 Algoritmos Desarrollados
 - Opposition-Based Learning
 - OBL Competitive Particle Swarm Optimization
 - Covariance Matrix Adaptation Evolutionary Strategy
 - Hybrid Simulated Annealing with Global Search
- 4 Evaluación Experimental
 - Funciones Propuestas por GenOpt
 - Estudio de la Parametrización y Rendimiento
 - Comparativa de Rendimiento entre Algoritmos
 - Clasificación en el Concurso GenOpt
- 5 Conclusions and Future Work
- 6 Bibliografía

Motivación y Objetivos

- Iniciación en el mundo de la investigación.
- Investigación en el campo de las meta-heurísticas.
- Optimización Global Continua.
- Guiar el desarrollo de todo el TFG a través de la participación en una competición de optimización global continua.

Competiciones

- Congress on Evolutionary Computation (CEC)
- Genetic and Evolutionary Computation Conference (GECCO)
- Global Trajectory Optimisation Competition (GTOC)
- **Generalization-Based Contest in Global Optimization (GenOpt)**



Conceptos Previos

Definición Formal

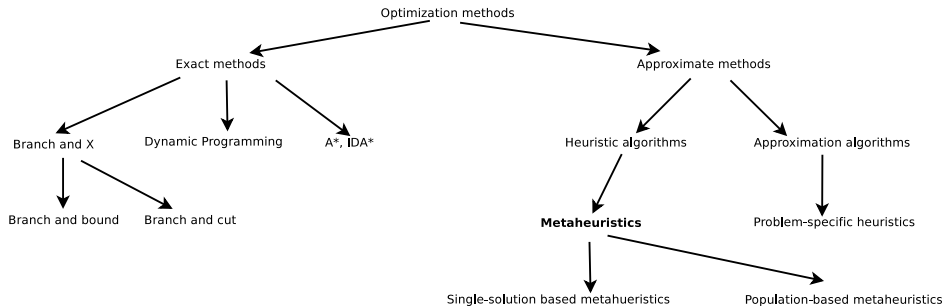
El objetivo de la optimización global, considerando un problema de minimización, es encontrar un vector $X^* \in \Omega$ tal que $f(X^*) \leq f(X)$ para todo $X \in \Omega$.

Espacio de Búsqueda

El espacio de búsqueda Ω está definido por un límite inferior (a_i) y superior (b_i) para cada una de las variables de decisión de la función, es decir:

$\Omega = \prod_{i=1}^D [a_i, b_i]$, siendo D el número de variables de decisión del problema a optimizar (Segredo et al., 2017)

Meta-heurísticas



Categorías

- **Búsquedas Locales:** Greedy Randomized Adaptive Search Procedure (GRASP) Díaz et al. (2017), Variable Neighborhood Search (VNS) Hansen et al. (2010).
- **Heurísticas Voraces:** Simulated Annealing (SA) Gerber and Bornn (2017a).
- **Algoritmos Evolutivos:** Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) Hajebi et al. (2016), Differential Evolution (DE) Zheng et al. (2017); Fu et al. (2017); Tian et al. (2017), Coevolutionary Algorithms (CEA) Atashpendar et al. (2016); Hajikolaei et al. (2016); Glorieux et al. (2016).

Criterios de Diseño

- Representación.
 - Permutaciones.
 - Cadena binaria.
 - Vector de valores naturales.
 - **Vector de números reales.**
- Condición de parada.
 - Iteraciones.
 - Factor de error.
 - **Evaluaciones:** 10^6 evaluaciones establecidas por GenOpt.



Algoritmos Desarrollados

Opposition-Based Learning

Definición formal

Sea $x \in \mathbb{R}$ un número real definido dentro de un cierto intervalo: $x \in [a, b]$. El número opuesto de x denotado como \bar{x} se define de la siguiente forma:

$$\bar{x} = a + b - x \quad (1)$$

Función D-Dimensional

Sea $P(x_1, x_2, \dots, x_D)$ un punto dentro de un sistema de coordenadas D - *dimensional* con $x_1, \dots, x_D \in \mathbb{R}$ y además $x_i \in [a_i, b_i]$. El opuesto del punto P se define como las coordenadas $\bar{x}_1, \dots, \bar{x}_D$ donde:

$$\bar{x}_i = a_i + b_i - x_i \quad i = 1, \dots, D \quad (2)$$

Algorithm 1 Búsqueda global()

```
1: NumIndividuos =  $|S|$ ;
2: OrdenarPoblacion(S);
3: MarcarNoExplorados(S);
4: Centroide = CalcularCentroide();
5: NumeroMejora = 0;
6: NumeroExplorado = 0;
7: while NumeroMejora > 0 y
   NumeroExplorado <  $|S|$  do
8:   k = 0;
9:   while  $S[k]$  = explorado y
      NumeroExplorado <  $|S|$  do
10:    k = rand(0,  $|S|$ ); (1)
11:   end while
12:    $S[k]$  = explorado;
13:   NumeroExplorado = NumeroEx-
     plorado + 1;
14:   Mejora = true;
15:   while Mejora = true do
16:     while  $|a_1| + |a_2| + |a_3| \neq 1$  do
17:       GenerarRand(a1, a2, a3); (2)
18:     end while
19:     while  $r_1 < k$  do
20:        $r_1 = \text{rand}(0, |S|)$ ; (2)
21:     end while
22:     NuevoInd = ModificarIndivi-
       duo(k, a1, a2, a3, Centroide,  $r_1$ );
23:     if NuevoInd <  $S[k]$  then
24:       Mejora = true;
25:        $S = S \cap \text{NuevoInd}$ ;
26:       NumeroMejora = Numero-
         Mejora + 1;
27:     else
28:       Mejora = false;
29:     end if
30:   end while
31: end while
32: OrdenarPoblacion(S);
33:  $S = \text{ObtenerMejores}(0, \text{NumIndivi-}$ 
    $\text{duos}, S)$ ;
34: return  $|S|$  mejores individuos encon-
     trados
```

OBL Competitive Particle Swarm Optimization (OBL-CPSO)

Algorithm 2 Particle Swarm Optimization()

```
1: while Condición de parada no satisfecha do
2:   for all  $p_i$  en  $S$  do
3:     Evaluar  $p_i$ ;
4:     Actualizar mejor posición  $pb_i$ ;
5:     Actualizar mejor global  $gb$ ;
6:   end for
7:   for all  $p_i$  en  $S$  do
8:     for all  $d_i$  en  $D$  do
9:        $v_{i,d} = v_{i,d} + C_1 * Rnd(0, 1) * [pb_{i,d} - x_{i,d}] + C_2 + Rnd(0, 1) * [gb_d - x_{i,d}];$   

        $Rnd(0,1)$  devuelve un número generado aleatoriamente en el rango  $[0, 1]$   

        $x_{i,d} = x_{i,d} + v_{i,d};$ 
10:    end for
11:  end for
12: end while
13: return Mejor solución obtenida
```

Diseño

El algoritmo Opposition-based Learning Competitive Particle Swarm Optimization (OBL-CPSO) Zhou et al. (2016) incluye dos modificaciones:

- Opposition-based Learning.
- Procedimiento de Competición.

Competición

Escogemos, aleatoriamente, tres partículas dentro del enjambre y las hacemos competir entre ellas mediante su valor de función objetivo. Para un enjambre de tamaño N , se realizarán un total de $N/3$ competencias Zhou et al. (2016).

- Ganadora (w).
- Neutra (n).
- Perdedora (l).

Algorithm 3 OBL Competitive Particle Swarm Optimization()

```
1: Inicializar();
2: while Condición de parada no satisfecha do
3:   Agitar();
4:   for all  $k = 1 : N/3$  do
5:      $r_1 = S(k)$ ;
6:      $r_2 = S(k + N/3)$ ;
7:      $r_3 = S(k + 2N/3)$ ;
8:      $(w, n, l) = \text{competir}(r_1, r_2, r_3)$ ;
9:     Actualizar  $X_{ld}^k(t)$ ;
10:    Actualizar  $X_{nd}^k(t)$ ;
11:    Actualizar los valores de fitness para N y L;
12:   end for
13:   BusquedaGlobal();
14: end while
15: return Mejor solución obtenida
```

Parámetros

- λ : tamaño de la población.
- C : la matriz de covarianza C de dimensión C_{λ}^D .
- σ : índice de variación entre generaciones.
- μ : individuos seleccionados en la selección elitista.
- $m^g \in \mathbb{R}^D$: Valor medio de la distribución en la generación g

Algorithm 4 Covariance Matrix Adaptation Evolutionary Strategy ()**Require:**

$$m \in R^n, \sigma \in R_+, \lambda$$

1: Inicialización

$$C = I, p_c = 0, p_\sigma = 0$$

$$c_c \approx 4/n, c_\sigma \approx 4/n, c_1 \approx 2/n^2, c_\mu \approx \mu_w/n^2, c_1 + c_\mu \leq 1$$

$$d_\sigma \approx 1 + \sqrt{\frac{\mu_w}{n}}, w_i = 1 \dots \lambda \quad \text{tal que} \quad \mu_w = \frac{1}{\sum_{i=1}^{\mu} w_i^2} \approx 0,3\lambda$$

2: **while** Condición de parada no satisfecha **do**

3: Muestreo

4: Actualizar el valor medio

5: Actualizar C

6: Actualizar σ

7: BúsquedaGlobal();

8: **if** Reinicio necesario **then**

9: Reiniciar();

10: **end if**11: **end while**12: **return** Mejor solución obtenida

Hybrid Simulated Annealing with Global Search (HSAGS)

Simulated Annealing (SA)

Simulated Annealing (SA) Robini et al. (2017); Karagiannis et al. (2017); Gerber and Bornn (2017b) es una meta-heurística que está inspirada por el proceso de recocido en la metalurgia Du and Swamy (2016).

- Iniciamos con un valor T_0 muy elevado.
- Un único individuo en la población.

Pasos del Algoritmo

- Perturbación.
- Evaluación.
- Actualizar Temperatura.
- Búsqueda global para HSAGS.

Algorithm 5 Hybrid Simulated Annealing with Global Search()

```
1: S = GenerarSolucionAleatoria();
2: T = InicializarTemperatura();
3: while Condición de parada no satisfecha do
4:   S' = AplicarPerturbacionAleatoria(S);
5:   Dif = EvaluarDiferencia(S, S');
6:   S = ActualizarSolucion(S, S', Dif);
7:   T = ActualizarTemperatura(T);
8:   S = BusquedaGlobalSA();
9: end while
10: return Mejor solución obtenida
```



Evaluación Experimental

Funciones propuestas por GenOpt

Características

GenOpt ha propuesto un total de **18 funciones** de dimensiones $D = 10, 30$ a optimizar, realizando **cient ejecuciónes independientes** para cada una de ellas.

Familias de Funciones

- Funciones GKLS.
- Funciones Clásicas Transformadas.
 - Rastrigin.
 - Rosenbrock.
 - Zakharov.
- Funciones Compuestas.
 - Goldstein-Price.
 - Hartmann.
 - Sphere.

Estudio de la Parametrización

Objetivos del Estudio

Determinar los mejores valores para los parámetros de cada algoritmo buscando obtener el máximo rendimiento de cada uno de ellos.

Poblaciones

- $popsiz$ = 20, 50, 75, 100 para OBL-CPSO y CMA-ES.
- $popsiz$ = 1 para HSAGS.

Rendimiento de OBL-CPSO

	OBL-CPSO-20			OBL-CPSO-50		
	μ	\tilde{x}	σ	μ	\tilde{x}	σ
f_0	$4,361e-01$	$2,041e-01$	$4,492e-01$	$3,937e-01$	$1,255e-01$	$4,193e-01$
f_1	$1,776e+00$	$1,681e+00$	$4,699e-01$	$1,464e+00$	$1,401e+00$	$2,359e-01$
f_2	$7,417e-01$	$1,002e+00$	$4,173e-01$	$7,657e-01$	$1,002e+00$	$3,618e-01$
f_3	$1,779e+00$	$1,699e+00$	$4,513e-01$	$1,455e+00$	$1,402e+00$	$2,441e-01$
f_4	$7,822e-01$	$1,003e+00$	$4,026e-01$	$7,764e-01$	$1,003e+00$	$3,803e-01$
f_5	$1,790e+00$	$1,698e+00$	$5,522e-01$	$1,451e+00$	$1,406e+00$	$2,479e-01$
f_6	$9,041e-01$	$3,609e-01$	$1,297e+00$	$5,852e-01$	$3,130e-01$	$7,651e-01$
f_7	$1,057e+01$	$1,014e+01$	$4,274e+00$	$8,992e+00$	$8,667e+00$	$3,394e+00$
f_8	$5,837e-01$	$3,580e-01$	$6,216e-01$	$3,908e-01$	$2,501e-01$	$4,752e-01$
f_9	$5,574e+00$	$5,209e+00$	$2,245e+00$	$4,881e+00$	$4,840e+00$	$1,817e+00$
f_{10}	$6,175e-03$	$4,830e-03$	$5,254e-03$	$4,564e-03$	$3,010e-03$	$3,711e-03$
f_{11}	$4,735e-02$	$4,723e-02$	$1,483e-02$	$4,155e-02$	$3,961e-02$	$1,488e-02$
f_{12}	$4,103e-02$	$2,806e-02$	$4,074e-02$	$2,463e-02$	$1,685e-02$	$2,296e-02$
f_{13}	$8,880e-02$	$8,295e-02$	$3,394e-02$	$8,708e-02$	$8,431e-02$	$2,912e-02$
f_{14}	$1,026e-01$	$3,233e-02$	$1,907e-01$	$3,723e-02$	$2,042e-02$	$4,753e-02$
f_{15}	$1,317e-01$	$1,180e-01$	$5,010e-02$	$1,198e-01$	$1,121e-01$	$3,953e-02$
f_{16}	$1,516e-02$	$1,252e-02$	$1,090e-02$	$1,380e-02$	$1,075e-02$	$1,142e-02$
f_{17}	$2,666e-01$	$2,286e-01$	$1,587e-01$	$2,049e-01$	$1,744e-01$	$1,091e-01$

Rendimiento de OBL-CPSO

OBL-CPSO-75			OBL-CPSO-100		
μ	\tilde{x}	σ	μ	\tilde{x}	σ
4,095e - 01	1,364e - 01	4,224e - 01	1,228e + 00	1,218e + 00	2,404e - 01
1,403e + 00	1,351e + 00	2,047e - 01	3,706e + 00	3,572e + 00	7,136e - 01
7,968e - 01	1,004e + 00	3,407e - 01	1,250e + 00	1,227e + 00	1,530e - 01
1,373e + 00	1,350e + 00	1,722e - 01	3,708e + 00	3,572e + 00	7,161e - 01
8,439e - 01	1,004e + 00	3,219e - 01	1,248e + 00	1,216e + 00	1,645e - 01
1,423e + 00	1,357e + 00	2,284e - 01	3,707e + 00	3,572e + 00	7,162e - 01
4,691e - 01	2,570e - 01	4,897e - 01	3,368e + 00	2,636e + 00	2,445e + 00
9,134e + 00	8,482e + 00	3,717e + 00	2,545e + 01	2,466e + 01	6,987e + 00
3,449e - 01	2,537e - 01	3,115e - 01	2,202e + 00	1,999e + 00	1,283e + 00
4,720e + 00	4,588e + 00	1,643e + 00	1,276e + 01	1,279e + 01	3,019e + 00
3,874e - 03	2,688e - 03	3,495e - 03	1,661e - 02	1,440e - 02	1,151e - 02
4,151e - 02	3,904e - 02	1,285e - 02	8,903e - 02	8,608e - 02	2,672e - 02
3,093e - 02	2,092e - 02	3,662e - 02	4,394e - 01	3,230e - 01	3,904e - 01
8,200e - 02	8,032e - 02	2,643e - 02	2,611e - 01	2,452e - 01	1,296e - 01
3,049e - 02	1,543e - 02	4,289e - 02	1,381e + 00	1,147e + 00	1,006e + 00
1,196e - 01	1,115e - 01	4,406e - 02	4,437e - 01	3,095e - 01	4,113e - 01
1,201e - 02	1,052e - 02	8,260e - 03	4,439e - 02	3,137e - 02	4,109e - 02
1,845e - 01	1,687e - 01	8,040e - 02	3,330e + 00	2,985e + 00	1,709e + 00

Rendimiento de CMA-ES

	CMA-ES-2 - 50			CMA-ES-0,3 - 50			CMA-ES-0,8 - 50		
	μ	\tilde{x}	σ	μ	\tilde{x}	σ	μ	\tilde{x}	σ
f_0	8,503e - 01	1,000e + 00	2,677e - 01	6,503e + 00	1,089e + 00	2,677e - 01	8,503e - 01	1,000e + 00	2,677e - 01
f_1	1,145e + 00	1,121e + 00	9,401e - 02	1,145e + 00	1,331e + 00	9,401e - 02	1,145e + 00	1,121e + 00	9,401e - 02
f_2	9,650e - 01	1,000e + 00	1,102e - 01	6,890e - 01	1,120e + 00	1,102e - 01	9,650e - 01	1,000e + 00	1,102e - 01
f_3	1,145e + 00	1,127e + 00	9,699e - 02	1,175e + 00	1,197e + 00	9,699e - 02	1,145e + 00	1,127e + 00	9,699e - 02
f_4	9,568e - 01	1,000e + 00	1,375e - 01	7,544e - 01	1,000e + 00	1,375e - 01	9,568e - 01	1,000e + 00	1,375e - 01
f_5	1,139e + 00	1,107e + 00	9,068e - 02	2,177e + 00	1,107e + 00	9,068e - 02	1,139e + 00	1,107e + 00	9,068e - 02
f_6	7,098e - 03	2,675e - 03	1,102e - 02	6,068e - 03	2,895e - 03	1,102e - 02	7,098e - 03	2,675e - 03	1,102e - 02
f_7	3,485e + 00	3,206e + 00	1,737e + 00	4,785e + 00	3,206e + 00	1,737e + 00	3,485e + 00	3,206e + 00	1,737e + 00
f_8	1,114e - 02	3,920e - 03	2,502e - 02	1,184e - 02	3,500e - 03	2,502e - 02	1,114e - 02	3,920e - 03	2,502e - 02
f_9	1,801e + 00	1,713e + 00	8,352e - 01	2,881e + 00	1,503e + 00	8,352e - 01	1,801e + 00	1,713e + 00	8,352e - 01
f_{10}	2,323e - 02	2,214e - 02	2,532e - 02	1,393e - 02	2,124e - 02	2,532e - 02	2,323e - 02	2,214e - 02	2,532e - 02
f_{11}	5,283e - 02	5,061e - 02	1,399e - 02	3,883e - 02	5,031e - 02	1,399e - 02	5,283e - 02	5,061e - 02	1,399e - 02
f_{12}	2,242e - 04	1,147e - 04	5,236e - 04	4,442e - 04	1,147e - 04	5,236e - 04	2,242e - 04	1,147e - 04	5,236e - 04
f_{13}	3,018e - 02	2,998e - 02	7,994e - 03	2,018e - 02	2,998e - 02	7,994e - 03	3,018e - 02	2,998e - 02	7,994e - 03
f_{14}	1,443e - 04	2,946e - 05	3,504e - 04	1,443e - 04	2,946e - 05	3,504e - 04	1,443e - 04	2,946e - 05	3,504e - 04
f_{15}	4,870e - 02	4,838e - 02	1,111e - 02	4,870e - 02	4,838e - 02	1,111e - 02	4,870e - 02	4,838e - 02	1,111e - 02
f_{16}	9,820e - 04	2,520e - 04	5,633e - 03	9,820e - 04	2,520e - 04	5,633e - 03	9,820e - 04	2,520e - 04	5,633e - 03
f_{17}	3,485e - 02	3,375e - 02	9,540e - 03	3,485e - 02	3,375e - 02	9,540e - 03	3,485e - 02	3,375e - 02	9,540e - 03

Rendimiento de HSAGS

	HSAGS-500		HSAGS-1000		HSAGS-10000
f_0	$8,060e - 01$	\leftrightarrow	9.338e-01	\leftrightarrow	$7,384e - 01$
f_1	0,000e + 00		$9,732e - 01$	\leftrightarrow	$9,732e - 01$
f_2	$9,241e - 01$	\leftrightarrow	$9,270e - 01$	\leftrightarrow	9.711e-01
f_3	0,000e + 00		0,000e + 00		0,000e + 00
f_4	$7,369e - 01$	\leftrightarrow	8.174e-01	\leftrightarrow	$5,917e - 01$
f_5	$9,481e - 01$	\leftrightarrow	$9,460e - 01$	\leftrightarrow	9.978e-01
f_6	$9,961e - 01$	\leftrightarrow	$9,971e - 01$	\leftrightarrow	9.990e-01
f_7	9.990e-01	\leftrightarrow	$9,971e - 01$	\leftrightarrow	$9,981e - 01$
f_8	0,000e + 00		$1,000e + 00$	\leftrightarrow	$1,000e + 00$
f_9	$1,000e + 00$	\leftrightarrow	$1,000e + 00$	\leftrightarrow	0,000e + 00
f_{10}	9.990e-01	\leftrightarrow	$9,971e - 01$	\leftrightarrow	$9,981e - 01$
f_{11}	$9,942e - 01$	\leftrightarrow	$9,961e - 01$	\leftrightarrow	9.990e-01
f_{12}	1.000e+00	\leftrightarrow	$1,000e + 00$	\leftrightarrow	$1,000e + 00$
f_{13}	9.990e-01	\leftrightarrow	$9,990e - 01$	\leftrightarrow	$1,000e + 00$
f_{14}	1.000e+00	\leftrightarrow	$1,000e + 00$	\leftrightarrow	$1,000e + 00$
f_{15}	1.000e+00	\leftrightarrow	$1,000e + 00$	\leftrightarrow	$1,000e + 00$
f_{16}	$1,000e + 00$	\leftrightarrow	9.990e-01	\leftrightarrow	$9,990e - 01$
f_{17}	1.000e+00	\leftrightarrow	$1,000e + 00$	\leftrightarrow	$1,000e + 00$

Comparativa de Rendimiento entre Algoritmos

Objetivos del Estudio

Comparar el rendimiento de cada uno de los algoritmos a la hora de optimizar las funciones propuestas por GenOpt.

Comparativa de Rendimiento entre Algoritmos

	CMA-ES-2 - 50			OBL-CPSO-75			HSAGS-500		
	μ	\bar{x}	σ	μ	\bar{x}	σ	μ	\bar{x}	σ
f_0	$8,503e-01$	$1,000e+00$	$2,677e-01$	$4,095e-01$	$1,364e-01$	$4,224e-01$	$3,875e+00$	$3,731e+00$	$8,967e-01$
f_1	$1,145e+00$	$1,121e+00$	$9,401e-02$	$1,403e+00$	$1,351e+00$	$2,047e-01$	$1,065e+01$	$1,052e+01$	$1,439e+00$
f_2	$9,650e-01$	$1,000e+00$	$1,102e-01$	$7,968e-01$	$1,004e+00$	$3,407e-01$	$3,878e+00$	$3,731e+00$	$8,982e-01$
f_3	$1,145e+00$	$1,127e+00$	$9,699e-02$	$1,373e+00$	$1,350e+00$	$1,722e-01$	$1,065e+01$	$1,052e+01$	$1,439e+00$
f_4	$9,568e-01$	$1,000e+00$	$1,375e-01$	$8,439e-01$	$1,004e+00$	$3,219e-01$	$9,568e-01$	$1,000e+00$	$1,375e-01$
f_5	$1,139e+00$	$1,107e+00$	$9,068e-02$	$1,423e+00$	$1,357e+00$	$2,284e-01$	$1,064e+01$	$1,050e+01$	$1,435e+00$
f_6	$7,098e-03$	$2,675e-03$	$1,102e-02$	$4,691e-01$	$2,570e-01$	$4,897e-01$	$1,373e+01$	$1,227e+01$	$6,326e+00$
f_7	$3,485e+00$	$3,206e+00$	$1,737e+00$	$9,134e+00$	$8,482e+00$	$3,717e+00$	$4,560e+01$	$4,423e+01$	$1,053e+01$
f_8	$1,114e-02$	$3,920e-03$	$2,502e-02$	$3,449e-01$	$2,537e-01$	$3,115e-01$	$6,038e+00$	$6,183e+00$	$1,809e+00$
f_9	$1,801e+00$	$1,713e+00$	$8,352e-01$	$4,720e+00$	$4,588e+00$	$1,643e+00$	$1,903e+01$	$1,891e+01$	$3,118e+00$
f_{10}	$2,323e-02$	$2,214e-02$	$2,532e-02$	$3,874e-03$	$2,688e-03$	$3,495e-03$	$1,270e-01$	$5,255e-02$	$2,558e-01$
f_{11}	$5,283e-02$	$5,061e-02$	$1,399e-02$	$4,151e-02$	$3,904e-02$	$1,285e-02$	$4,264e+00$	$1,808e-01$	$2,459e+01$
f_{12}	$2,242e-04$	$1,147e-04$	$5,236e-04$	$3,093e-02$	$2,092e-02$	$3,662e-02$	$5,330e+00$	$5,016e+00$	$2,499e+00$
f_{13}	$3,018e-02$	$2,998e-02$	$7,994e-03$	$8,200e-02$	$8,032e-02$	$2,643e-02$	$2,689e+00$	$2,567e+00$	$1,265e+00$
f_{14}	$1,443e-04$	$2,946e-05$	$3,504e-04$	$3,049e-02$	$1,543e-02$	$4,289e-02$	$8,964e+00$	$8,338e+00$	$4,235e+00$
f_{15}	$4,870e-02$	$4,838e-02$	$1,111e-02$	$1,196e-01$	$1,115e-01$	$4,406e-02$	$1,009e+01$	$8,802e+00$	$5,327e+00$
f_{16}	$9,820e-04$	$2,520e-04$	$5,633e-03$	$1,201e-02$	$1,052e-02$	$8,260e-03$	$4,049e+00$	$3,340e+00$	$3,172e+00$
f_{17}	$3,485e-02$	$3,375e-02$	$9,540e-03$	$1,845e-01$	$1,687e-01$	$8,040e-02$	$1,603e+01$	$1,465e+01$	$5,702e+00$

Criterios de Clasificación de GenOpt

- **High Jump**: mejor valor obtenido en los puntos de control.
- **Target Shooting**: éxito a la hora de alcanzar el óptimo global de la función.
- **Biathlon Score**: media entre el High Jump y Target Shooting.

¹Manifiesto del concurso GenOpt: <http://www.genopt.org/genopt.pdf>.

Clasificación en el Concurso GenOpt



Conclusions and Future Work

Conclusions

- High amount of parameters increases the complexity to evaluate the performance of the algorithms.
- The task of assessing a new modification was really difficult.
- CMA-ES algorithm accomplished the **third place** in the final leaderboard of the GenOpt contest considering the High Jump criterion.

Future Work

- Improve the different tested algorithms.



¿Preguntas?
Gracias por su atención.

- Atashpendar, A., Dorronsoro, B., Danoy, G., and Bouvry, P. (2016). A parallel cooperative coevolutionary smmpso algorithm for multi-objective optimization. pages 713–720.
- Du, K.-L. and Swamy, M. N. S. (2016). *Search and Optimization by Metaheuristics*.
- Díaz, J., Luna, D., Camacho-Vallejo, J.-F., and Casas-Ramírez, M.-S. (2017). Grasp and hybrid grasp-tabu heuristics to solve a maximal covering location problem with customer preference ordering. *Expert Systems with Applications*, 82:67–76.
- Fu, C., Jiang, C., Chen, G., and Liu, Q. (2017). An adaptive differential evolution algorithm with an aging leader and challengers mechanism. *Applied Soft Computing Journal*, 57:60–73.
- Gerber, M. and Bornn, L. (2017a). Improving simulated annealing through derandomization. *Journal of Global Optimization*, 68(1):189–217.
- Gerber, M. and Bornn, L. (2017b). Improving simulated annealing through derandomization. *Journal of Global Optimization*, 68(1):189–217.
- Glorieux, E., Svensson, B., Danielsson, F., and Lennartson, B. (2016). Improved constructive cooperative coevolutionary differential evolution for large-scale optimisation. pages 1703–1710.
- Hajebi, M., Hoorfar, A., and Bou-Daher, E. (2016). Inverse profiling of inhomogenous buried cylinders with arbitrary cross sections using cma-es. pages 863–864.
- Hajikolaie, K., Cheng, G., and Wang, G. (2016). Optimization on metamodeling-supported iterative decomposition. *Journal of Mechanical Design, Transactions of the ASME*, 138(2).

Bibliografía (cont.)

- Hansen, P., Mladenović, N., and Moreno Pérez, J. A. (2010). Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175(1):367–407.
- Karagiannis, G., Konomi, B., Lin, G., and Liang, F. (2017). Parallel and interacting stochastic approximation annealing algorithms for global optimisation. *Statistics and Computing*, 27(4):927–945.
- Robini, M., Ozon, M., Frindel, C., Yang, F., and Zhu, Y. (2017). Global diffusion tractography by simulated annealing. *IEEE Transactions on Biomedical Engineering*, 64(3):649–660.
- Segredo, E., Paechter, B., Segura, C., and González-Vila, C. I. (2017). On the comparison of initialisation strategies in differential evolution for large scale optimisation. *Optimization Letters*, pages 1–14.
- Tian, M., Gao, X., and Dai, C. (2017). Differential evolution with improved individual-based parameter setting and selection strategy. *Applied Soft Computing Journal*, 56:286–297.
- Zheng, L., Zhang, S., Tang, K., and Zheng, S. (2017). Differential evolution powered by collective information. *Information Sciences*, 399:13–29.
- Zhou, J., Fang, W., Wu, X., Sun, J., and Cheng, S. (2016). An Opposition-Based Learning Competitive Particle Swarm Optimizer. pages 515–521.



Desarrollo de Algoritmos Dirigido por Retos

Alejandro Marrero Díaz

Universidad de La Laguna

10 de julio de 2017

Escuela Superior de Ingeniería y Tecnología
Universidad de La Laguna