

Desarrollo de Algoritmos Dirigido por Retos

Alejandro Marrero Díaz

Eduardo Manuel Segredo González, Carlos Segura González

Universidad de La Laguna

3 de julio de 2017

Escuela Superior de Ingeniería y Tecnología
Universidad de La Laguna

- 1 Introducción
- 2 Conceptos Previos
- 3 Algoritmos Desarrollados
- 4 Evaluación Experimental
- 5 Conclusiones y Líneas Futuras
- 6 Conclusions and Future Work
- 7 Bibliografía

Motivación y Objetivos

- Iniciación en el mundo de la investigación.
- Investigación en el campo de las meta-heurísticas.
- 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

- Optimización Global.
- Meta-heurísticas.

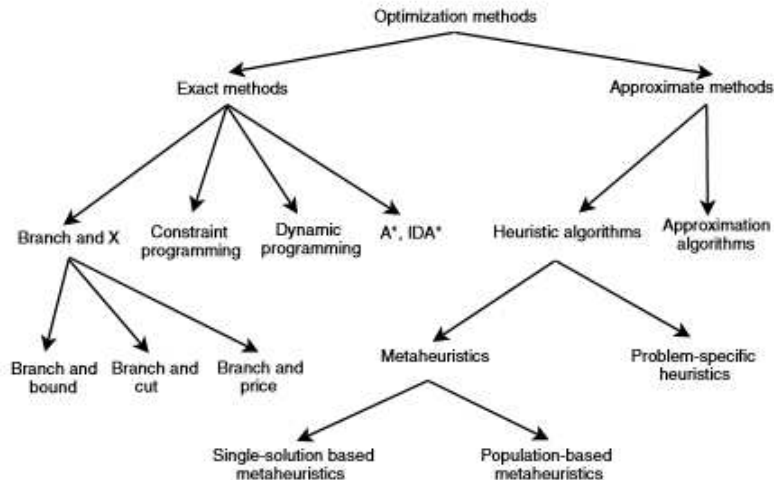
Definición

Optimizar una función f dentro de un intervalo especificado.

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 $X \in \Omega$, donde Ω es el espacio de búsqueda delimitado por un límite inferior lb y un límite superior ub ?.

Meta-heurísticas



Características

- Soluciones factibles en tiempo aceptable.
- Eficiencia y eficacia.

Categorías

- **Búsquedas Locales:** Greedy Randomized Adaptive Search Procedure (GRASP) ?, Variable Neighborhood Search (VNS) ?.
- **Heurísticas Voraces:** Simulated Annealing (SA) ?.
- **Algoritmos Evolutivos:** Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) ?, Differential Evolution (DE) ???, Coevolutionary Algorithms (CEA) ???.

Criterios de Diseño

- Intensificación.
 - Diversificación.
-
- Representación.
 - Condición de parada.

Representación.

- **Cadena binaria.**

1	0	1	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---

- **Vector de valores naturales.**

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

Representación.

- **Vector de números reales.**

0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3
-----	-----	-----	-----	-----	-----	-----	-----	-----

- **Permutaciones.**

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

 →

1	5	3	4	2	6	7	8	9
---	---	---	---	---	---	---	---	---

Condición de Parada

- **Iteraciones.**
- **Evaluaciones.**
- **Factor de error.**

En nuestro trabajo, el criterio de parada utilizado por todos los algoritmos desarrollados es 10^6 **evaluaciones**, criterio prefijado por la organización del concurso GenOpt.

Algoritmos Desarrollados

- Opposition-Based Learning (OBL)
 - Global Search (GS)
-
- Opposition-Based Learning Competitive Particle Swarm Optimization (OBL-CPSO)
 - Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES)
 - Hybrid Simulated Annealing with Global Search (HSAGS)

Opposition-Based Learning

Opposition-Based Learning es un concepto en computación que ha demostrado gran efectividad a la hora de mejorar diversas técnicas de optimización.

Al evaluar una solución candidata X perteneciente al conjunto S , simultáneamente calcularemos y evaluaremos la solución opuesta \bar{X} .

Variantes ?

- Quasi-opposition-based Learning (QOBL)
- Quasi-reflected Opposition-based Learning (QROBL)

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:

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

Búsqueda Global

En ciencias de la computación, una búsqueda global (Global Search) es un método heurístico para resolver problemas complejos de optimización ???.

- Explorar nuevas regiones del espacio de búsqueda Ω .
- Equilibrio entre intensificación y diversificación.

Centroide

El centroide de un conjunto de k elementos, tal que $k = |S|$, se define como:

$$C = \frac{x_1 + x_2 + \dots + x_k}{k} \quad (3)$$

Cada elemento x_i representa una solución factible a nuestro problema con D variables.

Algorithm 1 Cálculo del centroide

```
1: for  $i \leftarrow 0$  hasta  $D$  do  
2:    $Suma = 0$ ;  
3:   for  $j \leftarrow 0$  hasta  $|S|$  do  
4:      $Suma = Suma + S[i][j]$ ;  
5:   end for  
6:    $Centroide[i] = \frac{Suma}{|S|}$ ;  
7: end for  
8: return  $Centroide$ 
```

Búsqueda Global

Algorithm 2 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  $S[k] = \text{explorado}$  y
      NumeroExplorado <  $|S|$  do
10:    k = rand(0,  $|S|$ ); (1)
11:   end while
12:    $S[k] = \text{explorado}$ ;
13:   NumeroExplorado = NumeroExplorado + 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 = ModificarIndividuo(k, a1, a2, a3, Centroide,  $r_1$ );
23:     if NuevoInd <  $S[k]$  then
24:       Mejora = true;
25:        $S = S \cap \text{NuevoInd}$ ;
26:       NumeroMejora = NumeroMejora + 1;
27:     else
28:       Mejora = false;
29:     end if
30:   end while
31: end while
32: OrdenarPoblacion(S);
33: S = ObtenerMejores(0, NumIndividuos, S);
34: return  $|S|$  mejores individuos encontrados
```

Particle Swarm Optimization

Particle Swarm Optimization (PSO) ???? es una estrategia de optimización que ha demostrado ser muy eficiente en problemas de optimización global continua.

Características

- Conjunto de soluciones S como un enjambre de partículas.
- Las partículas se mueven dentro de Ω .
- Cada partícula posee una posición x_i y una velocidad \vec{v} .
- Recuerdan su mejor posición alcanzada pb_i ?.
- Mejor partícula global gb .

Algorithm 3 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
```

El algoritmo Opposition-based Learning Competitive Particle Swarm Optimization (OBL-CPSO) ? 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 $N/3$ competiciones ?.

- Ganadora - Winner (w): pasa directamente a la siguiente iteración.
- Neutra (n): utilizamos OBL.
- Perdedora - Loser (l): aprende de la partícula ganadora.

$$V_{ld}^k(t+1) = R_{1d}^k(t) * V_{ld}^k(t) + R_{2d}^k(t) * (X_{wd}^k(t) - X_{ld}^k(t)) + \varphi * R_{3d}^k(t) * (\bar{X}_{ld}^k(t+1)) \quad (4)$$

$$X_{ld}^k(t+1) = X_{ld}^k(t) + V_{ld}^k(t+1) \quad (5)$$

$$X_{nd}^k(t+1) = ub_d + lb_d - X_{nd}^k(t) + R_{4d}^k(t) * X_{nd}^k(t) \quad (6)$$

Donde $X_{wd}^k(t)$, $X_{ld}^k(t)$ y $X_{nd}^k(t)$ son las posiciones d-ésimas de las w, l y n en la k-ésima ronda de competición dentro de la iteración t.

V_{ld}^k es la velocidad de la partícula l en la dimensión d-ésima en la k-ésima ronda de competición de la iteración t ?.

$R_{1d}^k, R_{2d}^k(t), R_{3d}^k(t)$ y $R_{4d}^k(t) \in [0, 1]$

φ es un parámetro fijado manualmente.

$\bar{X}_{ld}^k(t)$ representa el valor medio de las posiciones de las partículas dentro del enjambre.

ub_d y lb_d son las cotas superiores e inferiores del espacio de búsqueda en la dimensión d-ésima ?.

Problemas en la actualización

Si una variable x_i toma un valor fuera del rango $[a, b]$:

- $X_{id} > b$: $X_{id} = b$.
- $X_{id} < a$: $X_{id} = a$.

Algorithm 4 OBL Competitive Particle Swarm Optimization()

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

Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES)

Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) ???
es un algoritmo evolutivo diseñado para problemas de optimización continua no lineales.

Características

- Muestreo mediante Distribución Normal Multivariante.
- No es necesario un gran estudio de los parámetros.
- Gran abánico de posibilidades en el diseño. Reinicios y criterios de finalización.

Bibliografía