

Lista de Algoritmos Heurísticos

Carlos Antonio Macías Duarte

20 de marzo de 2019

1. Recocido Simulado

Algoritmo 1: Recocido Simulado (SA)

Entrada:

n : Tamaño de vecindad

c : Control de temperatura

t : Número de iteraciones

Salida:

g : Mejor solución

1 $i \leftarrow 1$: Contador de iteraciones;

2 $g \leftarrow generarSolucionInicial()$: Solución inicial;

3 **mientras** $i \leq t$ **hacer**

4 $p \leftarrow generarVecindad(n, g)$: Generar nueva vecindad;

5 $j \leftarrow 1$: Contador de la vecindad;

6 **mientras** $j \leq n$ **hacer**

7 **si** p_j mejor que g **entonces**

8 $g \leftarrow p_j$;

9 **si no, si** $\exp(\frac{f(g) - f(p_j)}{c}) > rand(0, 1)$ **entonces**

10 $g \leftarrow p_j$;

11 $i \leftarrow i + 1$;

12 $c \leftarrow generarNuevoTamanoVecindad()$: Genera el nuevo tamaño de la vecindad siguiente;

13 $n \leftarrow generarNuevaTemperatura()$: Genera el nuevo valor de control de temperatura;

14 **devolver** g

2. Algoritmo de Selección Clonal

Algoritmo 2: Algoritmo de Selección Clonal (CLONALG)

Entrada: N : Tamaño de población n : Número de células a clonar t : Número de generaciones**Salida:** g : Mejor célula

```
1  $i \leftarrow 1$  : Contador de generaciones;
2  $p \leftarrow generarPoblacionInicial(N)$  : Población inicial;
3 mientras  $i \leq t$  hacer
4    $q \leftarrow seleccionarMejoresCelulas(p, n)$  : Mejores células de la
     generación;
5    $c \leftarrow clonar(q)$  : Generar clones;
6    $m \leftarrow mutar(c)$  : Hipermutación;
7    $p \leftarrow generarNuevaPoblacion(p, m, N)$  : Genera la nueva población;
8    $actualizarParametros()$  : Actualiza los parámetros de clonación y
     mutación;
9    $g \leftarrow seleccionarMejorCelula(p)$  : Mejor célula de la generación;
10   $i \leftarrow i + 1$ ;
11 devolver  $g$ 
```

3. Algoritmo Genético

Algoritmo 3: Algoritmo Genético (GA)

Entrada: n : Tamaño de la población t : Número de generaciones**Salida:** g : Mejor individuo

```
1  $p \leftarrow inicializarPoblacion(n)$  : Inicialización la primera generación;
2  $i \leftarrow 1$  : Contador de iteraciones;
3  $g \leftarrow seleccionarMejorIndividuo(p)$  : Mejor mejor individuo de la
     generación;
4 mientras  $i \leq t$  hacer
5    $q \leftarrow seleccionarParejas(p)$  : Selección de las parejas;
6    $p \leftarrow cruzar(q)$  : Proceso de cruce de parejas;
7    $p \leftarrow mutar(p)$  : Proceso de Mutación de individuos;
8    $g \leftarrow seleccionarMejorIndividuo(p)$  : Obtener el mejor individuo;
9    $i \leftarrow i + 1$ ;
10 devolver  $g$ 
```

4. Optimización por Enjambre de Partículas

Algoritmo 4: Optimización por Enjambre de Partículas (PSO)

Entrada:

n : Tamaño del enjambre

t : Número de iteraciones

Salida:

g : Partícula con la mejor posición

1 $p \leftarrow \text{inicializarEnjambre}(n)$: Inicialización del enjambre;

2 $i \leftarrow 1$: Contador de iteraciones;

3 $g \leftarrow \text{seleccionarMejorParticula}(p)$: Mejor partícula;

4 **mientras** $i \leq t$ **hacer**

5 $\text{generarNuevaVelocidad}(p, g)$: Actualiza la velocidad de las partículas;

6 $\text{generarNuevaPosicion}(p)$: Calcular nuevas posiciones;

7 $g \leftarrow \text{seleccionarMejorParticula}(p)$: Obtener la Mejor partícula;

8 $i \leftarrow i + 1$;

9 **devolver** g
