

- Algoritmo de escalada y enfriamiento simulado+ Idea principal (escalada)

Partiendo de un estado inicial, generar un vecino (puede ser de manera aleatoria o pseudoraleatoria), comprobar su valor y tomar el mejor entre actual - vecino. Tras esto, iterar hasta cumplir criterio de parada.

Una posible mejora es que tras un empeoramiento del valor vecino, el siguiente se tome de forma aleatoria, permitiendo escapar mínimos o mínimos locales.

+ Ejemplo pseudocódigo

1º Variables y funciones de inicio:

- Generar_inicio (x_i aleatorio)
- $F(x)$: calcula valor x_i
- $F_{\text{vecino}}(x_i)$: genera un vecino a partir de x_i

2º Bucle while (criterio de parada): $F(x_i) \geq \alpha \leq$ valor buscado o realizar n iteraciones

2.1 ~~Generar~~ $F_{\text{vecino}}(\text{Actual})$, si es la primera iteración sería Generar_inicio

2.2 $F(F_{\text{vecino}}(\text{Actual}))$

2.3 Si 2.2 es mejor que $F(x_i)$, Actual se vuelve $F_{\text{vecino}}(\text{Actual})$, sino se tomará un vecino para la siguiente iteración de forma aleatoria

3º Devuelve el mejor resultado

+ Idea principal (enfriamiento)

Paralelo al algoritmo anterior permitiendo explorar estados peores, mediante la función:

$$p(\Delta E, T) = e^{-\frac{\Delta E}{K \cdot T}}$$

Para la aceptación de un estado peor, usamos la adaptación:

$$p(\Delta F, T) = e^{-\frac{\Delta F}{T}}$$

ΔF : Incremento de la función de valoración

T : Valor inicial, este varía de la forma $T = \alpha \cdot T$ (con cada iteración)

La función T decremente con cada iteración para que la posibilidad de cambiar disminuya a medida que el algoritmo se itera (si $T \rightarrow \infty$ $e^{-\frac{\Delta F}{T}} = 1$, se tomará el ~~peor~~ ^{estado} aunque empeore el valor $F(x)$)

- Algoritmos genéticos

+ Conceptos básicos

- * Genes: Valores que puede tomar (binario, rango, lista de valores, ...)
- * Cromosomas: Conjunto de genes (Definir longitud de la cadena de genes)
- * Población: conjunto de cromosomas a estudiar
- * Fitness: Valoración del cromosoma para el problema
- * Decodificar (x): Interpretación del valor de los genes en el cromosoma (ej: 0110 significa que se toman los objetos 1 y 2)
- * Generación (evaluación): nueva población tras valoración, variación y selección

+ Tipos de cruce

- * En un punto:

1	2	3	4	5
5	4	3	2	1

 \Rightarrow

1	2	3	2	1
5	4	3	4	5
- * Multipunto: Como el anterior pero con varias secciones
- * Cruce uniforme: Cada índice del cromosoma se decide el cruce aleatoriamente
- * Permutaciones: Los genes cambian de lugar en el cromosoma

+ Mutaciones

Se cambia el valor de un gen de forma aleatoria (puede necesitar de penalizaciones en el algoritmo para calcular la función Fitness (x))

+ Tipos de permutaciones

- * Por intercambio:

1	2	3	4	5	6
6	5	4	3	2	1

 \Rightarrow

6	2	1	3	5	1
1	5	3	4	2	6

+ Tipos de permutaciones

- * Por intercambio:

1	2	3	4	5	6
---	---	---	---	---	---

 \Rightarrow

1	5	3	4	2	6
---	---	---	---	---	---
- * Por inserción:

1	2	3	4	5	6
---	---	---	---	---	---

 \Rightarrow

1	2	5	3	4	6
---	---	---	---	---	---
- * Por mezcla:

1	2	3	4	5	6
---	---	---	---	---	---

 \Rightarrow

1	4	2	3	5	6
---	---	---	---	---	---
- * Por orden:

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

 \Rightarrow

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

Nos fijamos en los valores del otro padre fuera de rango y miramos si está dentro del nuestro

* Por ciclos:

- | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
- \Rightarrow
- | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 9 |
| 9 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 |

Mantenemos en los padres los valores que forman el ciclo e intercambiamos los demás

+ Mecanismo de selección

* Proporcional a la valoración: Selección de maneras aleatorias con una probabilidad proporcional a su valoración

$$p(i) = \frac{F(i)}{\sum_{j=1}^n F(j)}$$

Este método solo se puede usar para maximizar (para min hacer $-F$)

Una alternativa sería asignar la proporción por ranking

ej (ruleta):

$$F(i_1) = 2$$

$$F(i_2) = 3$$

$$F(i_3) = 1$$

$$F(i_4) = 5$$

Rangos para tomar elemento i : $i_1 = [0, 1]$

$$i_2 = [2, \dots, 4]$$

$$i_3 = [5]$$

$$i_4 = [6, \dots, 10]$$

* Por torneo

Se divide la población en K individuos y se seleccionan el mejor por grupos, mientras mayor sea K , mayor será la precisión evolutiva

* Elitista

Parecido a por torneo, se selecciona un porcentaje de las mejores valoradas y se complementa con individuos aleatorios

+ Ejemplo pseudocódigo

1° Variables y funciones de inicio

- c : contador
- Generar - población $P(t)$
- Evalua - población $P(t)$

2° Bucle while (criterio de parada)

- 2.1 generan dos poblaciones a partir de $P(t)$ mediante torneo (dividen en K grupos, toman los mejores) o elitista (toman los mejores y rellenan con aleatorios)
- 2.2 Unifican las poblaciones del punto 2.1
- 2.3 Mutan la población del 2.2
- 2.4 Iteran

3° Devuelve el mejor valor de la población final $P(t)$

- PSO

+ Idea principal

Determinar el óptimo en base a los movimientos de una población hacia este

+ Conceptos básicos

* Determinar la siguiente posición de las partículas:

$$\vec{X}_i^{t+1} = \vec{X}_i^t + \vec{V}_i^{t+1}$$

* Determinar la siguiente velocidad (\vec{V}_i^{t+1}):

$$\vec{V}_i^{t+1} = W \cdot \vec{V}_i^t + C_1 \cdot r_1 \cdot (\vec{m}_i^t - \vec{X}_i^t) + C_2 \cdot r_2 \cdot (\vec{best}^t - \vec{X}_i^t)$$

W = Inercia

C_1 = Influencia individual

C_2 = Influencia social

r_x = Valor aleatorio

\vec{m}_i^t = mejor posición del individuo

\vec{best}^t = mejor posición general