

---

**Algorithm 1:** Algoritmo Evolutivo

---

```
1 población ← crearPoblaciónInicial()
2 evaluarPoblación (población)
3 while la condición de termino no ha sido alcanzada do
4   | poblaciónSeleccionada ← selección (población)
5   | poblaciónDecendiente ← cruzamiento (poblacionSeleccionada)
6   | poblaciónDecendiente ← mutación (poblaciónDecendiente)
7   | poblaciónDecendiente ← evaluarPoblación (poblaciónDecendiente)
8   | población ← remplazar (población, poblaciónDecendiente)
9 end
```

---

Figura 1: Pseudocodigo algoritmo evolutivo

---

**Algorithm 2:** Función de remplazo para el algoritmo NSGA-II

---

```
1 Function remplazar(población, poblaciónDescendiente)
2   unionPoblacion  $\leftarrow$  población  $\cup$  poblaciónDescendiente
3   /*  $F = (F_1, F_2, \dots)$  */
4    $F \leftarrow$  ordenarPorFrentesNoDominados (unionPoblacion)
5   nuevaPoblacion  $\leftarrow \emptyset$ 
6    $i = 1$ 
7   /* Hasta que nuevaPoblacion este lleno */
8   while ( $|nuevaPoblacion| + |F_i| \leq N$ ) do
9     /* Calcular y asignar la densidad a cada solución
       del frente  $F_i$  */
10    asignarDensidad ( $F_i$ )
11    /* Añadir a nuevaPoblacion las soluciones del frente
        $F_i$  */
12    nuevaPoblacion  $\leftarrow$  nuevaPoblacion  $\cup F_i$ 
13     $i = i + 1$ 
14  end
15  /* Ordenar el frente  $F_i$  usando el comparador de
     densidad */
16  ordenar ( $F_i, \prec_n$ )
17  /* Elegir los primeros  $N - |nuevaPoblacion|$  */
18  nuevaPoblacion  $\leftarrow$  nuevaPoblacion  $\cup F_i[1 : N - |nuevaPoblacion|]$ 
19  retornar nuevaPoblacion
20 fin
```

---

Figura 2: Pseudocódigo de la función de remplazo utilizada en el algoritmo NSGA-II [?]

---

**Algorithm 3:** Función de ordenación en frentes no dominados

---

```
1 Function ordenarPorFrentesNoDominados (población)
2    $F \leftarrow \emptyset$ 
3   foreach  $p \in \text{población}$  do
4     /* Conjunto de soluciones dominadas por  $p$  */
5      $S_p = \emptyset$ 
6     /* Numeros de soluciones que dominan a  $p$  */
7      $n_p = 0$ 
8     foreach  $q \in \text{población}$  do
9       if  $p \prec q$  then /*  $p$  domina a  $q$  */
10         $S_p = S_p \cup \{q\}$ 
11       else if  $q \prec p$  then /*  $q$  domina a  $p$  */
12         $n_p = n_p + 1$ 
13       end
14       if  $n_p = 0$  then
15          $prango = 1$ 
16          $F_1 = F_1 \cup \{p\}$ 
17       end
18     end
19   end
20    $i = 1$ 
21   while  $F_i \neq \emptyset$  do
22      $Q = \emptyset$ 
23     foreach  $p \in F_i$  do
24       foreach  $q \in S_p$  do
25          $n_q = n_q - 1$ 
26         if  $n_q = 0$  then
27            $qrango = i + 1$ 
28            $Q = Q \cup \{q\}$ 
29         end
30       end
31     end
32      $i = i + 1$ 
33      $F_i = Q$ 
34   end
35 fin
```

---

Figura 3: Pseudocódigo de la función de ordenamiento utilizada en NSGA-II [?]

---

**Algorithm 4:** Función de calculo de densidades

---

```
1 Function asignarDensidad( $I$ ) /*  $I$  : frente de soluciones no
   dominadas */
2    $l = |I|$  /* Obtiene el tamaño del frente */
3
4   /* Inicializa la distancia para cada solución */
5   foreach  $i \leftarrow 1$  to  $l$  do
6      $I[i].distancia \leftarrow 0$ 
7   end
8   foreach  $m \leftarrow 1$  to numero de objetivos do
9      $I \leftarrow \text{sort}(I, m)$  /* Ordena por el objetivo  $m$  */
10    /* Asignar a la primera y ultima solucion el valor  $\infty$  */
11     $I[1].distancia \leftarrow I[l].distancia \leftarrow \infty$ 
12    for  $i \leftarrow 2$  to  $l - 1$  do
13      /* Asigna la distancia a las soluciones restantes */
14       $I[i].distancia \leftarrow$ 
15         $I[i].distancia + (I[i + 1].m - I[i - 1].m) / (f_m^{max} - f_m^{min})$ 
16    end
17 end
18 fin
```

---

Figura 4: Pseudocódigo de la función de asignación de densidad [?]