

NSGA-II: Algoritmo Genético de Clasificación No Dominada

Una Introducción a la Optimización Multiobjetivo

AAB

Circe

16 de octubre de 2025

¿Qué es la Optimización Multiobjetivo?

- A diferencia de la optimización con un solo objetivo, aquí buscamos encontrar la mejor solución para **múltiples objetivos en conflicto** simultáneamente.
- Mejorar un objetivo a menudo significa empeorar otro.
- El resultado no es una única solución, sino un conjunto de soluciones de compromiso.

Dominancia de Pareto

- Una solución p **domina** a otra solución q si:
 - ① p es al menos tan buena como q en todos los objetivos.
 - ② p es estrictamente mejor que q en al menos un objetivo.
- Las soluciones que no son dominadas por ninguna otra en el conjunto se llaman **no dominadas**.

El Frente de Pareto

- Es el conjunto de todas las soluciones **no dominadas** de un problema.
- Cada solución en el frente de Pareto representa un equilibrio diferente entre los objetivos.
- El objetivo de los algoritmos de optimización multiobjetivo es encontrar una aproximación lo más cercana y diversa posible a este frente.

¿Qué es NSGA-II?

- Significa **Non-dominated Sorting Genetic Algorithm II**.
- Es un algoritmo genético de optimización multiobjetivo muy popular y eficaz.
- Es una mejora de su predecesor, NSGA, que resuelve problemas de complejidad computacional y falta de diversidad.

Dos Criterios de Selección Clave

NSGA-II selecciona las mejores soluciones basándose en dos criterios principales:

- 1 **Rango de Pareto (Rank):** Prioriza las soluciones en frentes de Pareto de menor orden.
- 2 **Distancia de Crowding (Crowding Distance):** Prioriza las soluciones en áreas menos densas del frente, fomentando la diversidad.

Flujo del Algoritmo (1/3)

- 1 **Inicialización:** Se crea una población inicial de N individuos de forma aleatoria.
- 2 **Clasificación No Dominada:** Se clasifica la población en frentes de Pareto (F_1 , F_2 , etc.).
- 3 **Cruce y Mutación:** Se generan N descendientes a través de estos operadores.

Flujo del Algoritmo (2/3)

- 4 **Combinación de Poblaciones:** Se une la población de padres (P_t) y la de descendientes (Q_t) en una nueva población R_t de tamaño $2N$.
- 5 **Nueva Clasificación:** Se realiza una nueva clasificación no dominada para la población combinada R_t .

6 Formación de la Siguiete Generación:

- Se copian los frentes F_1, F_2, \dots en la nueva población P_{t+1} hasta que el tamaño sea menor que N .
- El último frente que no cupo se ordena por **distancia de crowding** y se toman los mejores individuos hasta completar el tamaño N .

7 El ciclo se repite.

Rango de Pareto (Rank)

- A un individuo p se le asigna un rango ($rank$) igual a la cantidad de individuos que lo dominan.
- Las soluciones con $rank = 0$ (en el código se usa 'rank=1') forman el primer frente no dominado.

Fórmula

$$rank(p) = |\{q \in \text{Población} \mid q \text{ domina a } p\}|$$

Distancia de Crowding

- Mide la densidad de soluciones alrededor de un individuo.
- Es la suma de las distancias euclidianas a sus vecinos más cercanos en el frente de Pareto.
- El algoritmo favorece individuos con una **mayor distancia de crowding**, lo que ayuda a mantener una buena distribución del frente.

Ecuación de la Distancia de Crowding

Para un individuo i en un frente de Pareto ordenado, su distancia de crowding (d_i) es:

Fórmula de la Distancia

$$d_i = \sum_{m=1}^{\text{num_objetivos}} \frac{|f_m(i+1) - f_m(i-1)|}{f_m^{\text{máx}} - f_m^{\text{mín}}}$$

- Los extremos del frente ($f_m(1)$ y $f_m^{\text{máx}}$) se les asigna una distancia infinita.

- **Cruce (Crossover):** Se utiliza el cruce binario. Dos individuos (x_1, x_2) se combinan para crear dos hijos (y_1, y_2) .

Fórmula de Cruce

$$y_1 = \alpha \cdot x_1 + (1 - \alpha) \cdot x_2$$

$$y_2 = \alpha \cdot x_2 + (1 - \alpha) \cdot x_1$$

Donde α es un vector aleatorio en $[0, 1]$.

- **Mutación:** Se elige aleatoriamente un gen (variable) del individuo y se le asigna un nuevo valor aleatorio dentro de los límites del problema.

Fórmula de Mutación

$$x_j^{\text{mutado}} = \text{aleatorio}(\text{límites de } x_j)$$

El código incorpora métodos de penalización para problemas con restricciones.

- **Penalización Estática:** Se aplica una penalización fija a las soluciones que violan las restricciones.
- **Penalización Dinámica:** El factor de penalización aumenta con cada iteración, forzando al algoritmo a encontrar soluciones viables a medida que avanza.

El código permite dos tipos de selección:

- **Selección por Torneo Binario:**

- Se eligen dos individuos aleatoriamente.
- El ganador es el de menor rango de Pareto. En caso de empate, el de mayor distancia de crowding.

- **Selección por Ruleta de Pareto (NRGA):**

- Se asigna una probabilidad de selección inversamente proporcional al rango.
- Eligen un frente y luego, por ruleta, un individuo dentro de ese frente.

Ventajas de NSGA-II

- **Eficiencia:** Rápida convergencia a un frente de Pareto.
- **Diversidad:** El mecanismo de distancia de crowding asegura una buena distribución de las soluciones.
- **Simplicidad:** Su lógica es relativamente simple de implementar.
- Es una de las metaheurísticas más utilizadas para optimización multiobjetivo.

Desventajas de NSGA-II

- No garantiza la convergencia a un óptimo global.
- El rendimiento puede depender de la correcta sintonización de los parámetros del algoritmo.
- El cálculo de la distancia de crowding puede ser costoso para un gran número de objetivos.

Conclusiones

- NSGA-II es un algoritmo robusto y eficiente para resolver problemas de optimización multiobjetivo.
- Su éxito radica en el equilibrio entre la **calidad** de la solución (mediante la clasificación no dominada) y la **diversidad** (a través de la distancia de crowding).
- Es una herramienta esencial en la caja de herramientas de la optimización evolutiva.

¡Gracias por su atención!

¿Alguna pregunta?