

Práctica 11 : Frentes de Pareto

I E G

9 de noviembre de 2021

Resumen

En esta práctica se utiliza la optimización multicriterio [2] donde a un mismo conjunto de variables se le asignan valores de forma que optimicen dos o más funciones objetivo sin que una mejora empeore a la otra figura 1. El objetivo de esta práctica es observar el porcentaje de soluciones que pertenecen al frente al ir incrementando la cantidad de funciones objetivo para $k \in \{2, 3, \dots, 9\}$ graficando con diagramas de violín que sean combinados con diagramas de caja bigote, verificando las diferencias observadas estadísticamente significativas.

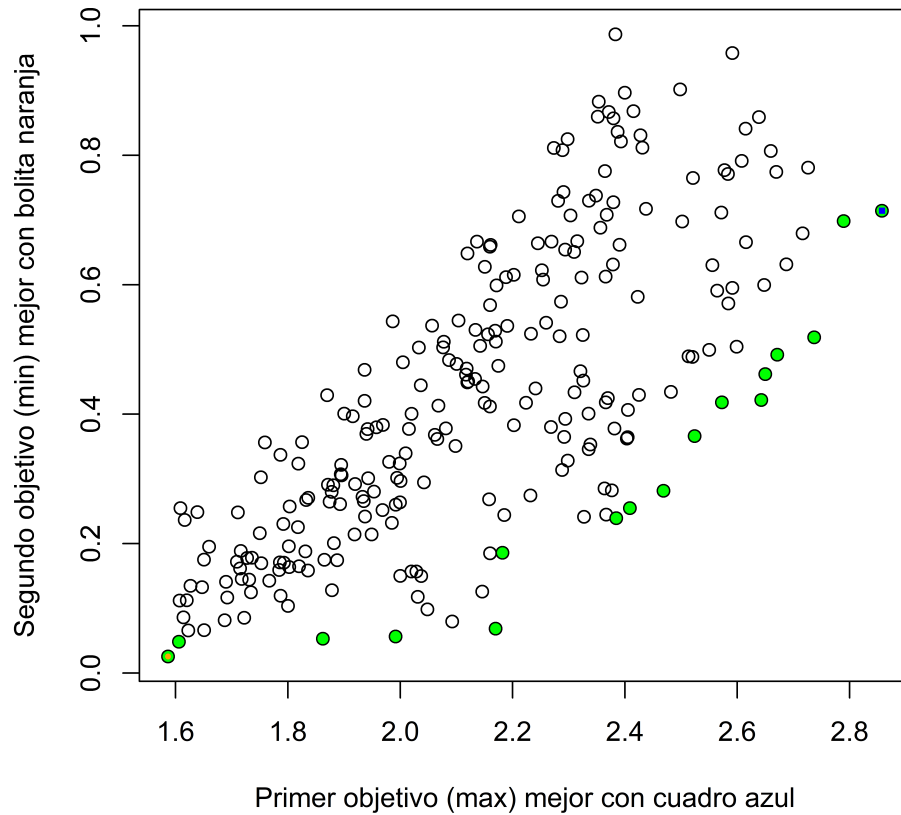


Figura 1: Ejemplo bidimensional de frente de Pareto.

1. Desarrollo

Se utiliza el lenguaje de programación Python 3.9.6 para la generación del código previamente reportado en [1, 2] primero se generan polinomios pseudo aleatorios y se determina si se va a minimizar o maximizar. Se determinan todas aquellas soluciones que dominan y a dicho conjunto de soluciones se le denomina frente de Pareto. Al código fuente de módica agregando dos ciclos for para las replicas que son 30 y para variar las funciones objetivo. El número de soluciones aleatorias es de 280.

```
iteracion = 30 # cuantas iteraciones
porcentajes = []
for k in range(1, 10): # cuantas funciones objetivo

    for it in range(iteracion):
        vc = 4
        md = 3
        tc = 5
        obj = [poli(md, vc, tc) for i in range(k)]
        minim = np.random.rand(k) > 0.5
        n = 280 # cuantas soluciones aleatorias
        sol = np.random.rand(n, vc)
        val = np.zeros((n, k))
        for i in range(n): # evaluamos las soluciones
            for j in range(k):
                val[i, j] = evaluate(obj[j], sol[i])
        sign = [1 + -2 * m for m in minim]

        no_dom = []
        for i in range(n):
            d = [domin.by(sign * val[i], sign * val[j]) for j in range(n)]
            no_dom.append(not np.any(d)) # si es cierto que ninguno es verdadero
        frente = val[no_dom, :]
        porcentaje = (len(frente)/n)*100
        porcentajes.append(porcentaje)
```

Finalmente se gráfica los porcentajes de cada función en grafica violín figura fig2.

2. Experimento

Para $k \in \{2, 3, \dots, 9\}$ se grafica con diagramas de violín que sean combinados con diagramas de caja bigote, donde se observa en la figura 2 que el porcentaje de las soluciones pertenecen al frente de Pareto para cada función se observa además con 6 es el 50 de porcentaje y para las funciones 7 o más el porcentaje es casi la totalidad del porcentaje como en el cuadro 1.

3. Conclusiones

En conclusión, el porcentaje de soluciones de frente de Pareto no dominantes aumenta conforme se incrementa el número de funciones objetivo, para trabajo futuro y un estudio más profundo la hipótesis seria: mientras más funciones se agrega al sistema se puede llegar a una solución más perfecta.

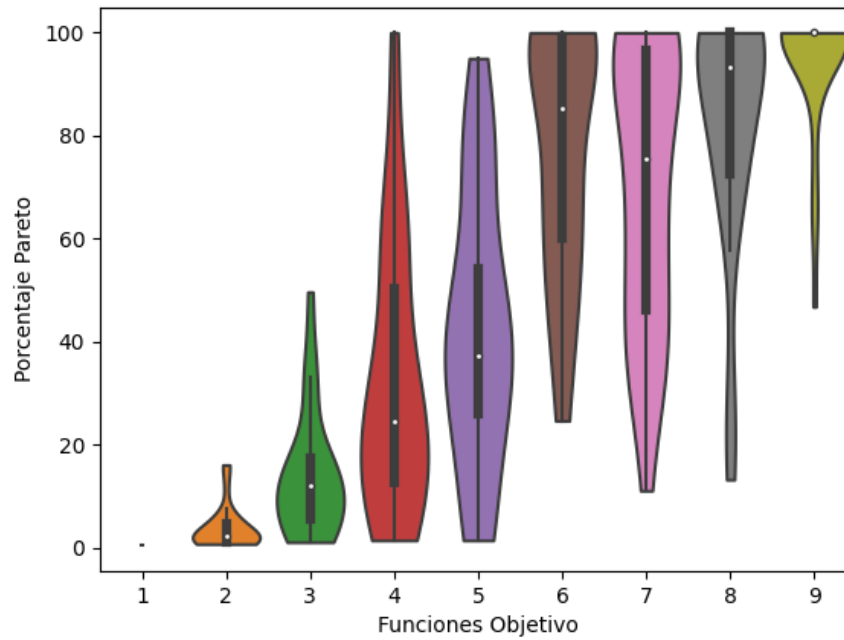


Figura 2: Gráficos de violín del porcentajes frente de Pareto.

Cuadro 1: Porcentaje de los datos.

Funciones	Porcentaje
2	8.0
3	24.0
4	1.0
5	1.0
6	57.0
7	99.0
8	99.5
9	100.0

Referencias

- [1] D. Leyva. Práctica 11: frentes de pareto, mayo 2021. URL <https://github.com/Denisse251/Simulation/blob/main/Tarea.11/Practica11.py>.
- [2] E. Schaeffer. Práctica 11: frentes de pareto, noviembre 2021. URL <https://elisa.dyndns-web.com/teaching/comp/par/p11.html>.