

Reporte práctica 11

Frentes de Pareto

Introducción

Para la práctica actual, analizaremos la optimización multicriterio, donde a un conjunto de variables se le asignan valores para lograr la optimización de funciones objetivo.

En este caso, se optó por la implementación de un generador de polinomios aleatorios, que servirán como restricciones y funciones objetivo. Para seleccionar las mejores soluciones, se usa la dominancia de Pareto, la cual se basa en que una solución domina a otra si esta misma no empeora ninguno de los objetivos y mejora al menos una. A todo el conjunto de soluciones que no fueron dominadas se le denomina frente de Pareto.

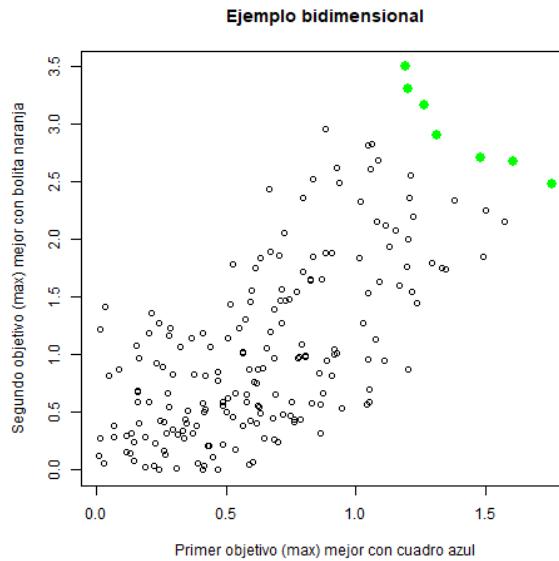


Figura 1. Ejemplo de frente de Pareto para dos funciones objetivo.

Objetivos

Paralelizar el cálculo donde convenga y graficar de manera porcentual las soluciones de Pareto como función del número de funciones objetivo.

Simulación y Resultados

Para poder implementar el uso de clúster y realizar los procedimientos de modo paralelo, se acude a la librería *doParallel*, la cual ya ha sido utilizada anteriormente y por ende no es necesaria una explicación extensa sobre la misma.

Como parámetros para la comparación en los tiempos de ejecución, se utilizaron valores secuenciales en 10 para el número de funciones objetivo k desde 10 hasta 50. Así como también un número de repeticiones i , para lograr la formación de los diagramas caja-bigote.

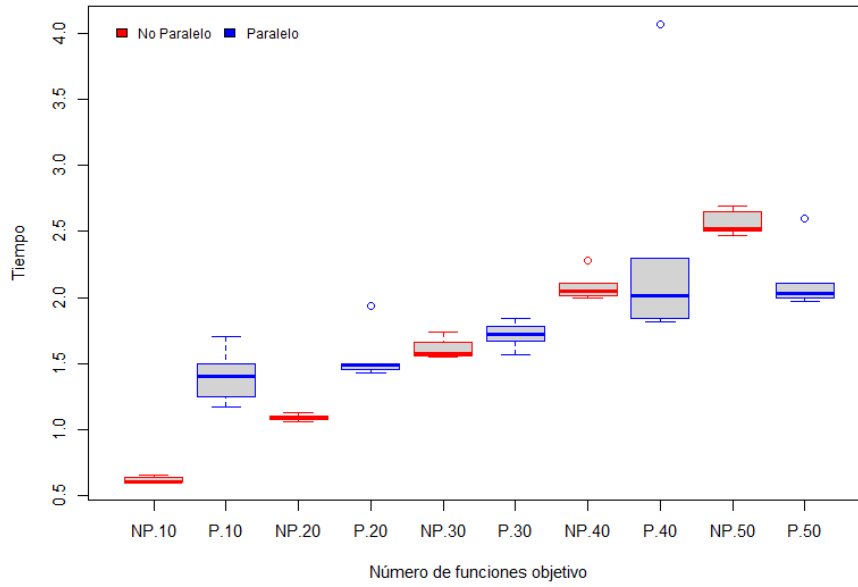


Figura 2. Efecto en el tiempo de ejecución al incrementar k .

Como se muestra en la figura 2, existe un cambio notable de los tiempos de ejecución de nuestro código sin paralelizar, creando una tendencia relativamente lineal en base al aumento del número de funciones objetivo, iniciando cerca del medio segundo para el caso más simple y consiguiendo un tiempo mucho mayor cerca de los 2.5 segundos por corrida para su caso con el mayor número de funciones objetivo. Caso contrario al ocurrido en el código con el uso de paralelismo, que no consigue un aumento tan significativo, fluctuando desde el caso con menos funciones objetivos alrededor de los 1.5 segundos y en el caso de mayor uso de funciones objetivo hasta aproximadamente 2 segundos.

Cabe mencionar que como se puede apreciar en la figura 2, los tiempos de ejecución son más rápidos con el código original si no sobrepasamos las 30

funciones objetivos. Mientras que si utilizamos un número mayor a este, para mayor eficiencia, es mejor optar por el código en paralelo.

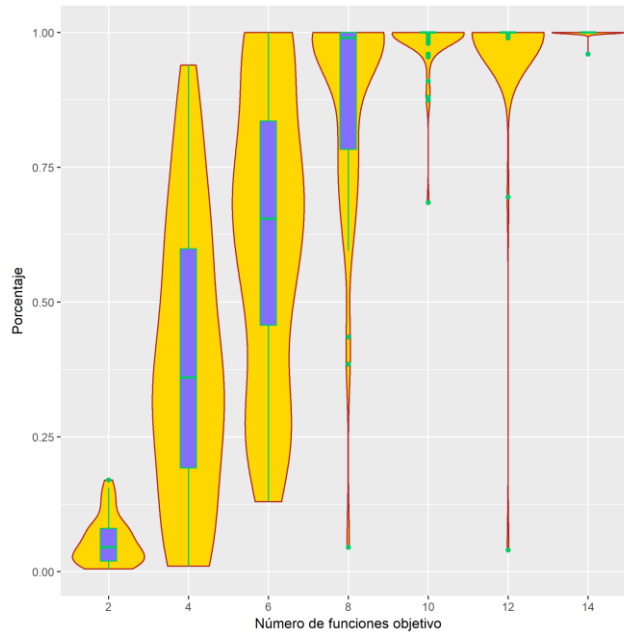


Figura 3. Porcentaje de soluciones de Pareto.

En la figura 3, se aprecia como con el uso de un número mayor de funciones repercute directamente en el porcentaje de soluciones de Pareto para dichas funciones. Como se logra ver, al usar solamente 2 funciones objetivo, el porcentaje de soluciones es relativamente bajo llegando a un porcentaje de máximo 0.20, mientras que si aumentamos a 4 el número de funciones objetivo, esta alcanza un rango de porcentaje muy extenso hasta casi alcanzar un 0.90 y así sucesivamente con los siguientes casos, donde entre más número de funciones objetivos se utilizan, mayor será el porcentaje.

Conclusiones

Para finalizar se puede decir que se logró modificar el código de forma que se consiguió paralelizar procedimientos para que disminuyan considerablemente el tiempo de ejecución, esto sin olvidar que será efectivo mientras se lleguen a utilizar más de 30 funciones objetivo, en caso contrario será mejor optar por no utilizar paralelismo.

Mientras que si hablamos del porcentaje en relación con el número de funciones objetivo, se pudo deducir que entre mayor sea el número funciones objetivo se conseguirá un porcentaje más cercano a la unidad.