

# Práctica 10

## Algoritmo Genético

### 29 de Abril del 2018

#### Introducción

El problema de la mochila (Knapsack problem) es un problema clásico en los problemas denominados POC (Problemas de Optimización Combinatoria) de inteligencia artificial. Este tipo de problemas es considerado NP (Por sus siglas en inglés Non- Probabilistic Problem), debido a que existe una combinación exponencial de instancias que no pueden ser resueltas en su totalidad [1].

En la realización de la práctica 10 se plantea resolver un problema de mochila, en el simuló un algoritmo genético para resolver el problema de la mochila.

Para la simulación se parte del siguiente problema, se tienen una mochila con capacidad de carga  $P$ , en su interior contiene  $n$  objetos cada uno de ellos con un peso  $p_i$ , un valor asociado  $v_i$ . Se desea conocer cuales objetos llevar en el interior de la mochila sin exceder su capacidad de tal manera de maximizar el beneficio.

#### 2.- Especificaciones Computacionales

Para el desarrollo de esta práctica se utilizó un equipo de cómputo tipo Notebook marca HP Pavilion x360, Intel Core i5 7th Gen, 8GB de memoria, 1TB Disco duro, 4 núcleos.

#### 3.- Objetivos

Paralelizar el algoritmo genético y estudiar el efecto en su tiempo de ejecución.

Cambiar la selección de padres para tener una reproducción usando selecciones de ruleta, estudiar el cambio para comprobar el uso de selecciones de ruleta produce una mejora significativa en la calidad de la solución.

## Simulación y resultados.

Se realizó una simulación para demostrar la eficiencia de la implementación paralela para su desarrollo de hizo uso de la librería *parallel*.

Se genera una población inicial con un tamaño *init* con valores de 20, 50, 100, 200, 350 y 400 con la finalidad de comparar los tiempos de ejecución.

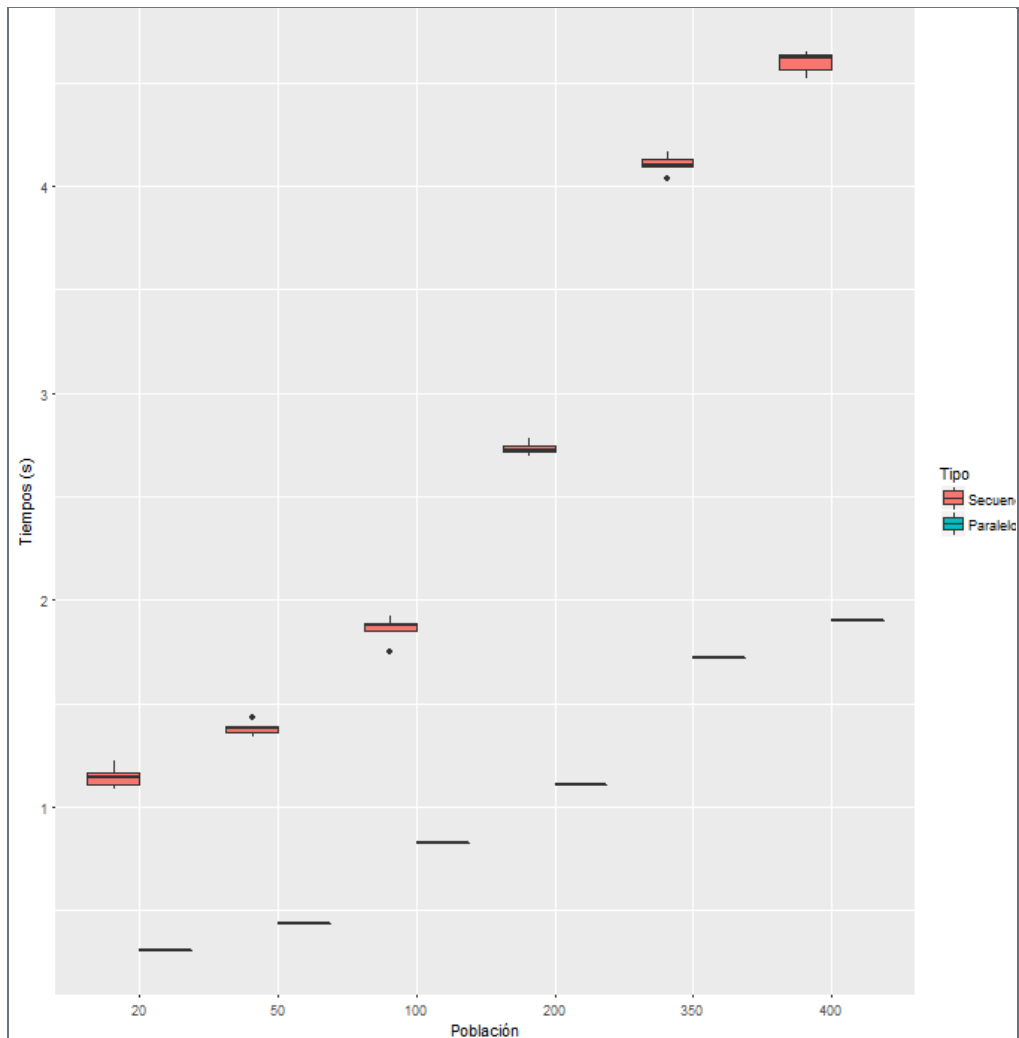


Figura 1 Tiempos de ejecución de la simulación

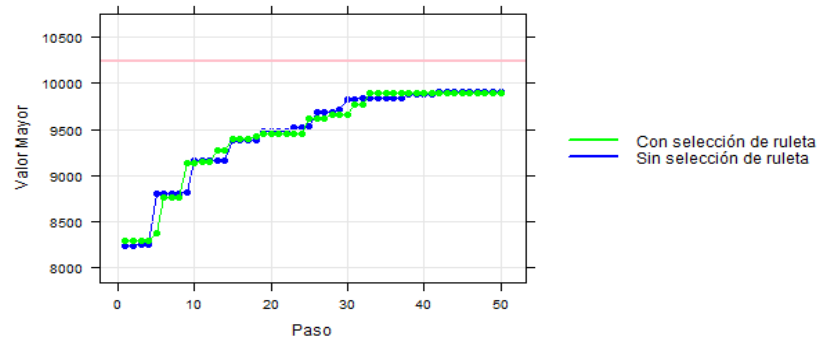
Los resultados obtenidos en la simulación son muy notorios, comparando desde nuestro valor inicial de población más pequeño (20) tardó, 13 segundos de ejecución (paralelizado) contra poco más de 1 minuto en la versión sin paralelizado.

En todas las muestras se apreció un comportamiento similar, demostrando de esta manera la eficiencia al paralelizar la simulación.

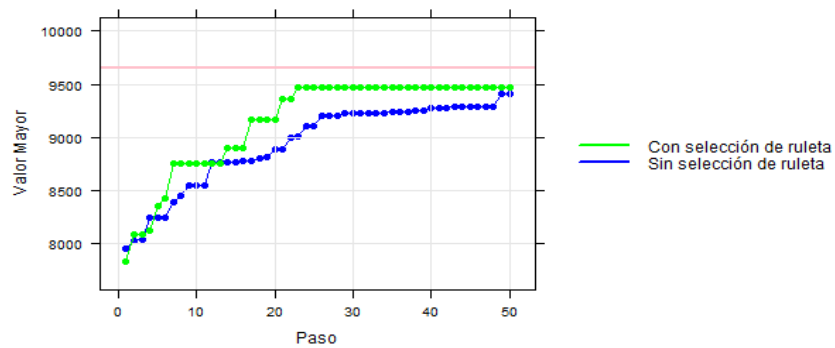
## Selección por Ruleta

El primer reto consta de cambiar el método de selección de los individuos a reproducirse, este método consta en asignar una probabilidad de selección a cada padre que dependa directamente de su valor objetivo.

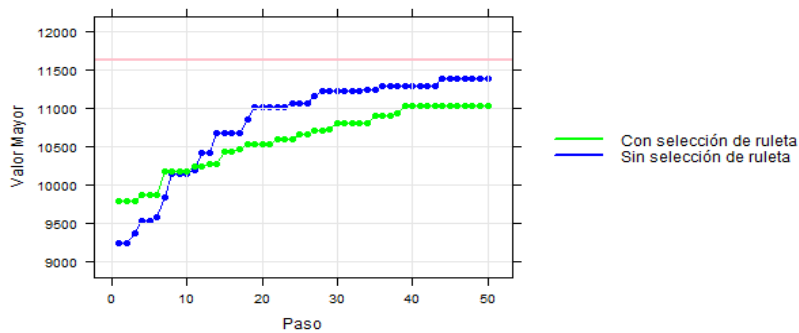
Para implementar el método de selección por ruleta se introduce en la simulación una nueva distribución de probabilidad con prob de sample.



A)



B)



C)

Figura 2 Eficacia del método de selección por ruletas

En la figura 2 aparece la gráfica de eficiencia del método de selección por ruletas. La gráfica A) muestra un comportamiento muy similar ambos se acercan al óptimo, sin embargo, para el experimento B) si se observa una diferencia, desde el paso 13 empieza a optimizarse hasta llegar a tener un comportamiento lineal a partir del paso 22 hasta el fin de la simulación. Finalmente la gráfica C) se aprecia notablemente que el método sin reproducción tuvo una eficacia muy notable comparado con las otras graficas y con el método por ruletas.

## **Conclusiones**

En base a las simulaciones realizadas es factible paralelizar el algoritmo ya que los tiempos de ejecución son mucho menores si se comparan con los tiempos obtenidos de manera secuencial, sin importar si el tamaño de la población es chico o es grande. Respecto al reto 1, el método de la reproducción por medio de ruleta no siempre será la mejor opción, pues como se mostro en la figura 2C puede que la reproducción sin ruletas se acerque más al óptimo.

## **Bibliografía**

- [1] Fayard, D. and Plateau G. (1994). An exact algorithm for the O-I collapsing knapsack problem. Discrete Applied Mathematics 49, 175-187
- E. Balas and E. Zemel, An algorithm for large zero-one knapsack problems, Oper. Res. 28 (1980) 1130-1154.