Reporte práctica 10

Algoritmo genético

Introducción

La práctica actual está basada en el problema de la mochila, donde se busca la optimización basados en la selección de un grupo de objetos, de modo que no exceda la capacidad de la mochila y se obtenga el valor máximo posible.

En el presente caso, se adapta el problema al caso de un genoma. Para buscar la mejor solución (generación) se utilizó un algoritmo pseudo-polinomial para luego utilizar un algoritmo genético y buscar a lo largo de varias generaciones, donde interfieren dos mecanismos de posible mejora (reproducción y mutación) en la nueva generación creada acercarse lo mejor posible a la solución óptima.

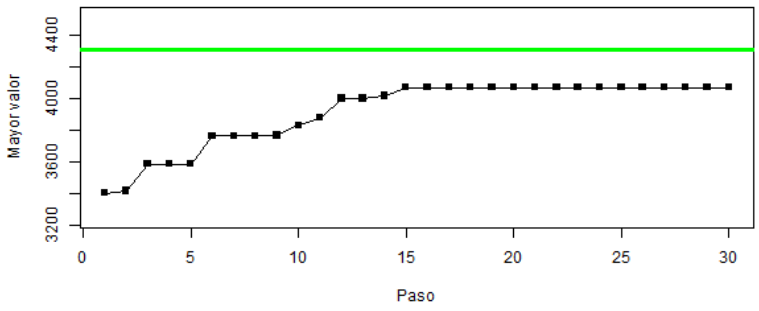

Figura 1. Solución del código pseudo-polonimial y genético.

Objetivos

Paralelizar el algoritmo genético y estudiar los efectos en el tiempo de ejecución.

Cambiar la selección de padres para la reproducción a una selección de ruleta. Estudiar si este cambio produce una mejora estadista significativa en la calidad de la solución.

Simulación y Resultados

Para poder demostrar si existe alguna mejora en los tiempos de ejecución de la simulación del algoritmo genético, primero se implementó la librería doParallel, como se ha hecho en prácticas pasadas y por ende se está familiarizado, y con ello la creación de ciertas funciones para la agilización de nuestra simulación. Logrando esto, se buscó la comparación de ambas simulaciones con distintos números de población inicial *init*, pero manteniendo los demás parámetros intactos, como lo son el número máximo de pasos o generaciones *tmax*.

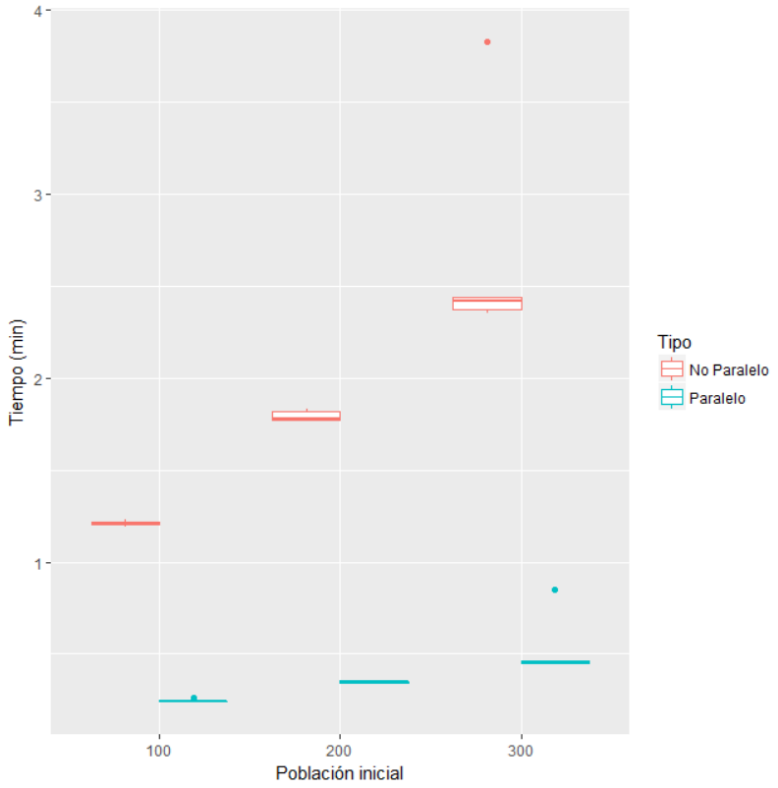

Figura 2. Efecto en el tiempo de ejecución de la simulación.

Como se muestra en la figura 2, existe una notable diferencia al utilizar alguna librería para paralelizar nuestra simulación, quedando demostrado que sin importar mucho el tamaño de población inicial *init*, nuestra simulación tarda poco menos de medio minuto en el caso paralelo, mientras que en la simulación base se llega a extender dicho tiempo hasta casi dos minutos y medio, existiendo un aumento considerable al momento de aumentar el tamaño de población inicial *init*.

Para poder conseguir cambiar el método de selección de padres, se optó por la implementación del parámetro sugerido en clase *prob* de *sample*, además de la creación de una nueva función que nos ayude a lograr dicho objetivo. Con esto se consigue que cada padre posea una probabilidad de ser elegido que es directamente proporcional a su valor de función objetivo y su factibilidad.

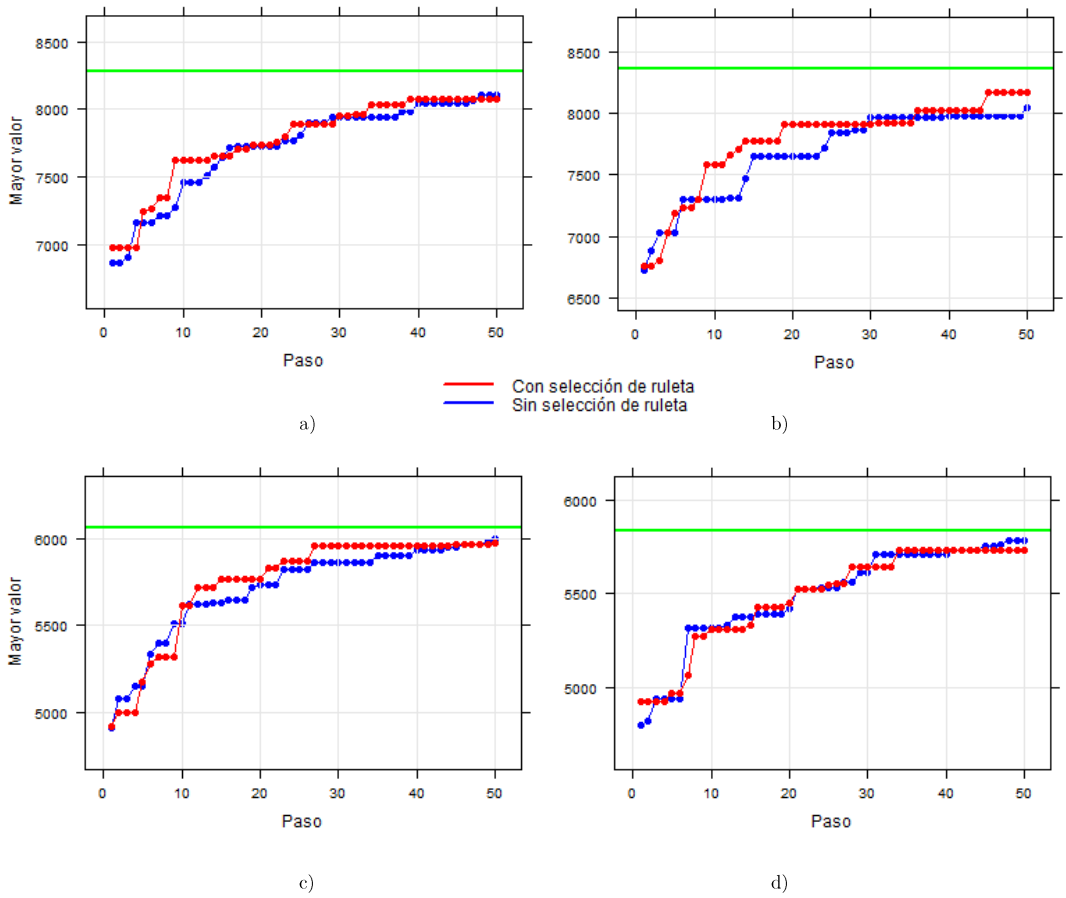

Figura 3. Efecto en generaciones futuras modificando el método de selección.

Se puede observar en la figura 3 el efecto que tuvo el cambio del método de selección, mostrando en la mayoría de los casos, como lo son a), b) y c) una mayor eficiencia para acercarse a la solución óptima, consiguiendo mejoras en una menor cantidad de generaciones en comparación con el método aleatorio. Sin embargo, dicha mejora termina perdiendo relevancia, ya que en las últimas generaciones simuladas, ambos métodos consiguen una aproximación similar, inclusive en el caso d), al implementar el método aleatorio se logró una aproximación más certera.

Conclusiones

Se demostró de manera muy notoria la mejoría en los tiempos de ejecución al implementar alguna librería para realizar múltiples funciones en paralelo, quedando la reducción del tiempo a poco menos de medio minuto para la simulación.

Además, se logró mostrar el impacto que causa el efecto al utilizar otro método en la selección de padres para nuevas generaciones, dejando claro que este nuevo método logra acercarse a la solución óptima en menos generaciones, pero pierde valor al avanzar a generaciones futuras, donde es alcanzado por el método aleatorio.