

Tarea 10

Simulación de Sistemas

Beatriz Alejandra García Ramos

A 17 de Octubre de 2017

1. Algoritmo genético

Para esta práctica se utilizará un algoritmo genético, el cual sirve de apoyo para cuando no existen algoritmos exactos que sean eficientes. Un algoritmo genético incluye posibles soluciones de un problema que se tiene en términos de un genoma, el cual en este caso será un vector que contenga verdaderos y falsos, los cuales indicarán qué objetos se deben incluir y cuáles no para construcción de los hijos tomando en cuenta que los padres deben reproducirse de manera que se toma el inicio del primero con el fin del segundo ó el fin del primero con el inicio del segundo.

Además de que las soluciones puedan hacer una reproducción entre padres para crear hijos éstas pueden crear réplicas de sí mismas pero cambiando uno de sus valores, así se crea una solución distinta pero manteniendo con mayor probabilidad su factibilidad.

Así se puede tener entonces un algoritmo que crea una población inicial con cierta probabilidad de ocurrencia y después se encarga de hacer mutaciones y reproducciones para lograr tener una mejor población. Al final del algoritmo se toma la mejor de las soluciones factibles y éste es el resultado que se arroja como el cercano al óptimo.

2. Solución

La idea principal en ésta práctica es lograr que el algoritmo tome menos tiempo en ejecutarse para lograr conseguir respuestas en menor tiempo y además poder hacer modificaciones en la selección de padres para que la reproducción que se haga nos ayude a tomar las mejores combinaciones posibles y obtener muchas soluciones factibles muy cercanas al óptimo e incluso el óptimo mismo. Además se desea modificar la manera en que se seleccionan a los hijos que quedarán vivos en la fase de supervivencia, ya que cada vez que el algoritmo hace un paso éste selecciona a ciertos hijos que lograron tener mayor factibilidad, sin embargo ciertas combinaciones de hijos factibles con hijos no factibles podrían hacer que al terminar los pasos logren mejores combinaciones que nos ayuden a encontrar una mejor solución, incluso una mejor que al combinar sólo las soluciones factibles.

2.1. Tarea base

Dado que el algoritmo es secuencial en un inicio se hace un proceso un poco tardado, por lo que es mejor paralelizar ciertas funciones que nos ayuden a reducir el tiempo de ejecución del algoritmo, la tarea base es sobre el cambio que se debe hacer en el código para lograr paralelizarlo y lograr esta mejora.

Lo que se realizó fue un cambio en la manera en que se crean las nuevas soluciones al reproducirse y mutarse las soluciones actuales, además de hacer un cambio en los objetivos que se toman para hacer las soluciones nuevas y los que se eligen como factibles.

El cambio que se realizó se hizo así ya que estas funciones cada vez que se hace un paso se realizaban muchas veces, dependientes de las repeticiones, del tamaño y de la longitud del vector que asigna los pesos, los cuales se asignaron al algoritmo originalmente.

Dado que se ha visto anteriormente que la función `foreach` de la librería `doParallel` sí paraleliza sin ningún problema pero no reduce tiempos de ejecución se realizaron cambios con la librería `parallel` utilizando de ella la función `parSapply`, la cual nos ayuda a darle una mejora en tiempos a nuestro algoritmo, lo que se realizaron fueron funciones nuevas de reproducción, mutación, objetivos y factibles que nos ayuda a llamar las funciones creadas originalmente, en éstas se tiene como proceso lo que en el algoritmo original se llevaba a cabo dentro del proceso, es decir, aquellas funciones que se mandaban llamar dentro del algoritmo y hacían algún proceso ahora son funciones nuevas que se paralelizan dentro del proceso del algoritmo y las funciones regresan aquellos datos que se requieren para lograr tener buenas soluciones y entre ellas elegir el mejor, como se hace en el algoritmo original.

Primeramente, cuando se hizo la comparación de las mejores soluciones del algoritmo secuencial con el algoritmo paralelizado se hace notorio que la solución óptima aumenta pero que la mejor de las soluciones factibles en cada paso del proceso no tiene una cercanía a la solución óptima, como se observa en la figura 2.1, lo cual no es algo bueno para el objetivo que tiene el algoritmo que es acercarse cada vez más a ésta.

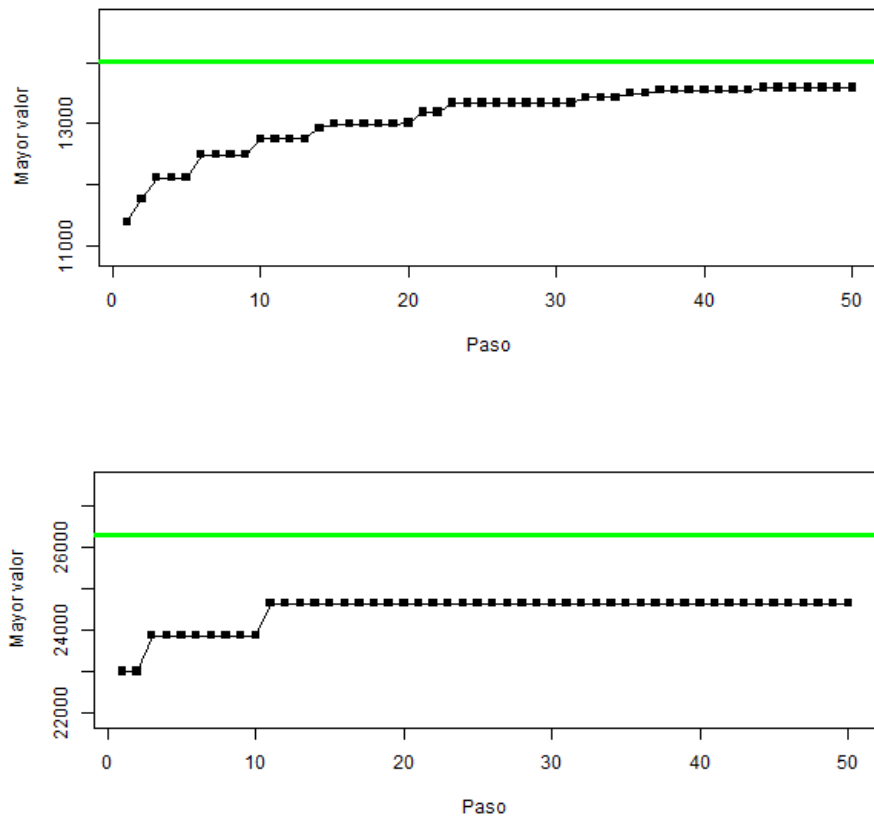


Figura 1: Soluciones factibles en cada paso que ocurre en el algoritmo.

Sin embargo, aunque no hay mejoría en la cercanía de la solución óptima sí existe una mejora muy notoria en el tiempo de ejecución del algoritmo, como se puede ver en la figura 2.1 tenemos una gran diferencia en tiempos entre el algoritmo secuencial, que es el que se maneja con color rojo y el algoritmo paralelizado que es el que se encuentra con color azul. Como se puede ver, el algoritmo paralelizado es aproximadamente un minuto de diferencia, que es una gran ventaja cuando se tienen que hacer procesos de pasos extensos. Se realizaron cinco repeticiones de cada tipo de proceso (secuencial y paralelo) para verificar si el tiempo era similar cada vez que se realizaba el algoritmo, en el algoritmo secuencial fue similar en cuatro ocasiones y en una aumentó, sin embargo en el algoritmo paralelizado los tiempos fueron muy parecidos, y como antes se mencionó, existe casi un minuto de diferencia entre ambos procesos, por lo cual se puede decir que el paralelizado sí le gana por mucho al secuencial.

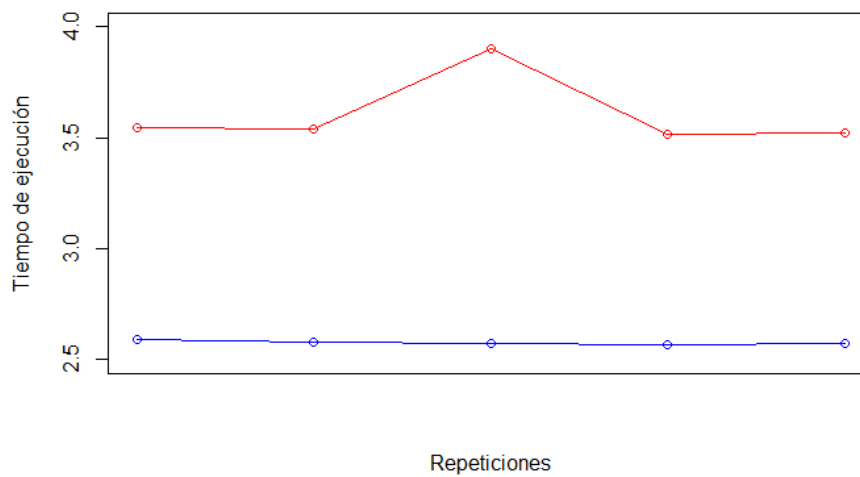


Figura 2: Tiempos de ejecución del algoritmo secuencial y el algoritmo paralelizado.

Usando la función `Shapiro` se sabe que los datos de los algoritmos son normales, lo cual permite realizar una prueba Wilcoxon en R que nos muestra si las medias son iguales o no, al momento de realizar esta prueba con los datos resultantes arrojó un valor $p = 9,762e - 07$ que es un valor pequeño, menor al valor $p = 0,05$ que debe tener la prueba para aceptar la hipótesis nula, como es menor entonces la hipótesis nula de que son iguales se rechaza, por lo tanto tenemos como resultado que existe diferencia entre los datos, es decir que existe diferencia entre los tiempos de ejecución.

2.2. Reto 1

Para realizar el primer reto y hacer cambios en la selección de la solución que será tomada como un padre se debe modificar la probabilidad en que estos son tomados en la función de reproducción, dado que se van a modificar todas las probabilidades en que se toma la solución para ser candidato a ser padre se realizó un proceso de `for` que va guardando cada probabilidad para cada solución cada solución se toma de la función objetivo, al cual le damos mayor probabilidad de ocurrencia a los factibles pero no se deja de lado que los no factibles sean tomados en cuenta ya que al hacer combinaciones de éstos después de ciertos pasos pueden llegar a hacer mejores soluciones como hijos. Entonces, tendremos una probabilidad de ocurrencia distinta en cada paso que da el algoritmo, tendremos también que para crear esta probabilidad se debe tomar el objetivo i y multiplicarlo por una probabilidad de ocurrencia de 0,1 a 0,9, y ésta probabilidad se toma en cuenta en la manera en que se van a tomar los padres en la función de reproducción. Además se realizaron más pasos en el proceso para lograr ver con mayor claridad el efecto que puede tener el cambio de probabilidad en la selección de padres.

Como se puede observar en la figura 2.2 tenemos una mejoría en las soluciones factibles mejores al momento de hacer el cambio de la selección de padres, cuando se tenía solamente el cambio en la paralelización (como se encuentra en la primer imagen de la figura) se puede notar que la mejor factible en cada caso no se acerca mucho a la solución óptima que se tiene en el algoritmo, pero cuando se tiene el cambio en la paralelización y además el cambio en la manera en que se toman los padres se tiene una mejoría en el acercamiento a la solución óptima del algoritmo (como se puede ver en la segunda imagen de la figura).

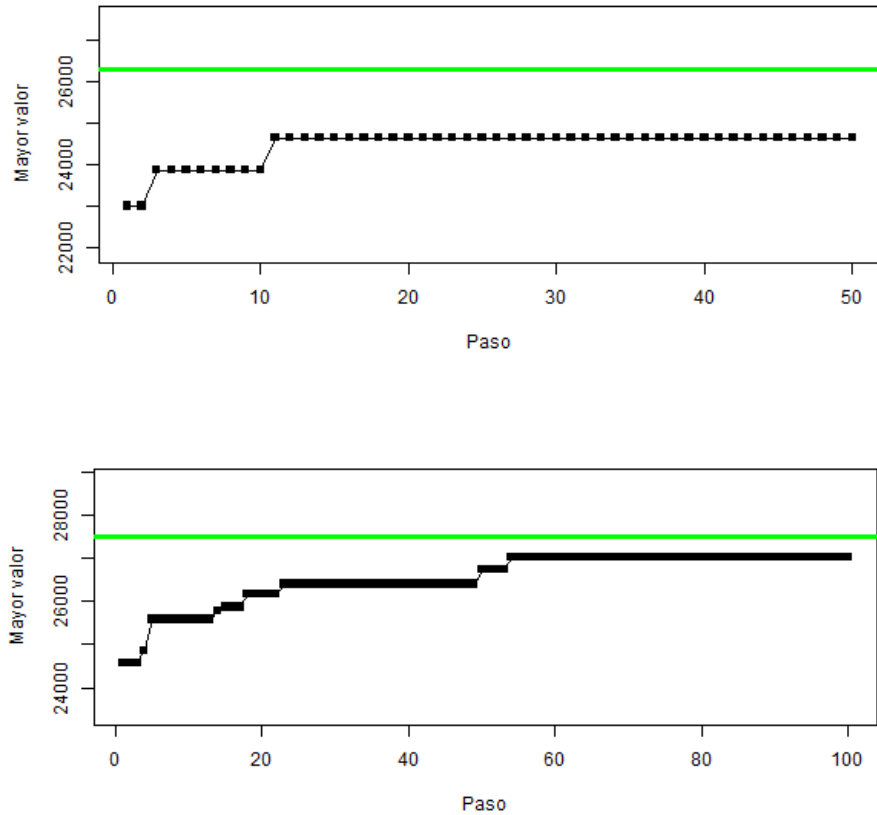


Figura 3: Soluciones factibles en cada paso que ocurre en el algoritmo.

Además haciendo la prueba estadística de Wilcoxon en R se tiene un valor de $p = 2,2e - 16$, el cual es menor a $p = 0,05$ que nos indica que la hipótesis nula de que los datos son iguales, por lo cual tenemos entonces que los datos son diferentes y sí hay una diferencia significativa que nos permite decir que las modificaciones que se hicieron en el código son favorables a nuestro algoritmo.

2.3. Conclusiones

En la práctica se obtuvieron buenos resultados, el cambiar la paralelización aporta mucho tiempo de ahorro dado que los procesos se hacen al mismo tiempo y al hacerse secuenciales toma tiempo en cada una de las funciones para lograr obtener los resultados requeridos, además el cambiar la manera en que los padres son seleccionados al momento de tomar en cuenta la función de reproducción tenemos que las soluciones factibles van mejorando al hacer distintas combinaciones entre soluciones buenas y malas dependientes de la probabilidad de ocurrencia tomada en la nueva forma en que las probabilidades son asignadas. Éstas son buenas formas de cambiar la manera en que el algoritmo genético se lleva a cabo.