

Problema del texto más parecido:

Introducción:

Dado el problema del texto más parecido, cuyo enunciado dice:

“Dada una lista de N textos, de longitud M de caracteres, generar un texto de M caracteres con la menor distancia posible a los textos dados. Donde la distancia entre dos textos está dada por la cantidad de caracteres diferentes que tienen entre sí.”

Se nos pidió proponer una metaheurística **GRASP** cuya complejidad sea polinómica, su tiempo de ejecución no sea excesivamente largo y que retorne una solución válida, pero no necesariamente óptima. Para lograr esto, es de suma importancia definir correctamente ciertos puntos críticos de la metaheurística mencionada, uno de ellos es la cantidad de iteraciones necesarias del **GRASP**.

Este informe se centra en la experimentación, recolección y la visualización de datos, para poder así, ser capaces de determinar una función que en base a el N y M de cualquier instancia del problema, nos indique cual es la cantidad de iteraciones de GRASP, necesarias para que este retorne una solución con las características deseadas y lo haga dentro de los términos ya mencionados.

Otros puntos críticos que valen la pena mencionar:

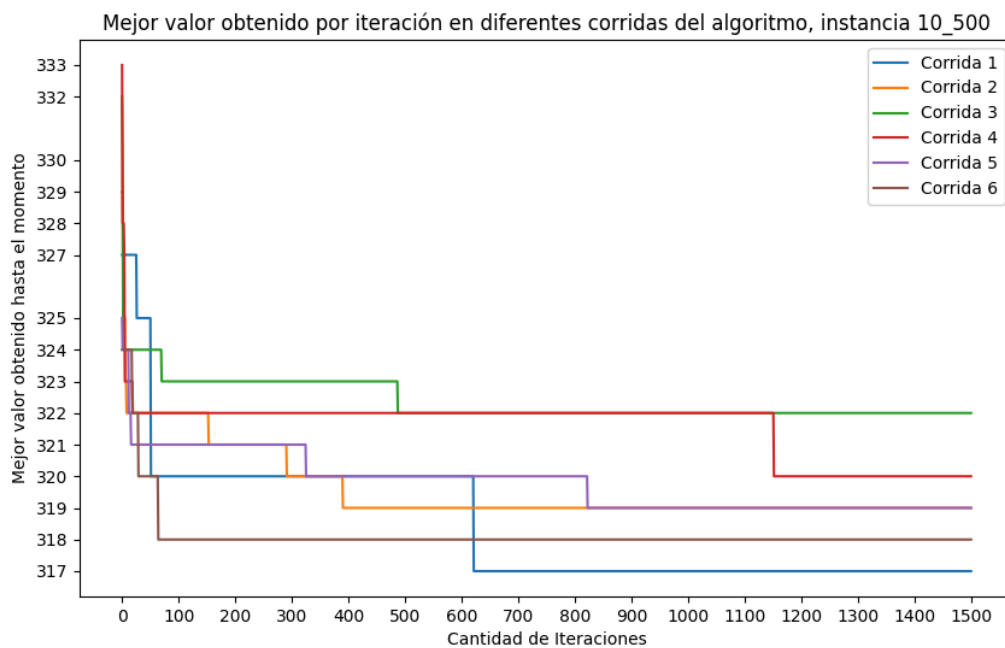
Además de la cantidad de iteraciones del GRASP, existen otros puntos críticos sobre los cuales se pueden experimentar y en base a los resultados proponer las implementaciones más convenientes. Por simplicidad de este trabajo y para mantener el foco sobre el punto crítico mencionado en la introducción, las demás partes críticas quedaron a criterio del alumno y de su implementación. A continuación se listan dichas secciones críticas, la decisión que se tomó con respecto a ellas y una breve explicación del por que de dicha decisión:

- Nivel de randomización de la heurística **Greedy-Randomizada**: El tamaño de la lista de candidatos de donde se elegirá el caracter que pasará a formar parte de la solución para cada posición de la misma, estará dado por la raíz cuadrada del tamaño del alfabeto utilizado, se optó por dicha solución ya que la pendiente de la curva generada por dicha función no es ni tan plana ni tan pronunciada, lo que significa que para los distintos posibles candidatos tendremos una cantidad razonable de candidatos posibles.

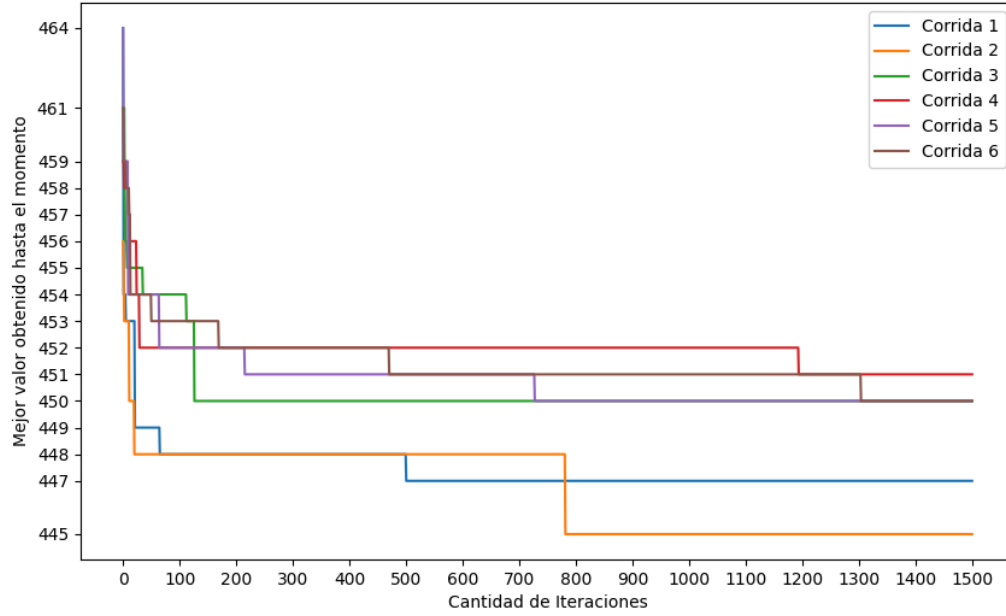
- Cantidad de iteraciones de la **Búsqueda Local**: Se decidió que la misma sea como máximo de 100 iteraciones, esto es así ya que realizando algunas pruebas sobre las instancias dadas, se notó que a la búsqueda local le tomaba en promedio unas 25/30 iteraciones encontrar el mínimo local.

Gráficos Iniciales:

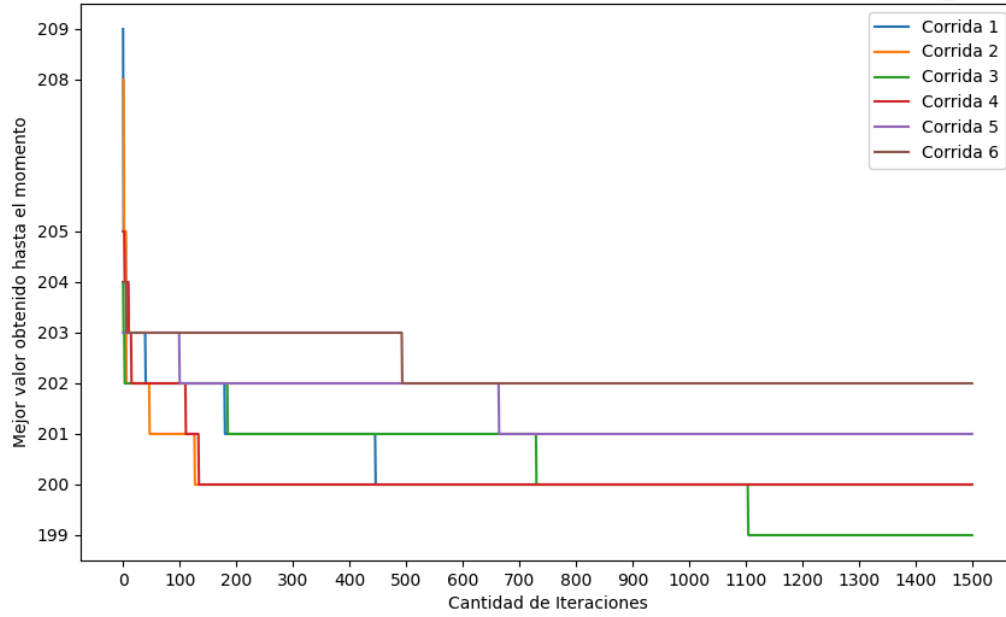
A continuación se presentan 9 gráficos realizados sobre 9 instancias distintas del problema, los cuales muestran la relación de la **cantidad de iteraciones** y el **mejor valor obtenido** hasta esa iteración, para varias ejecuciones sobre cada una de las instancias utiliza



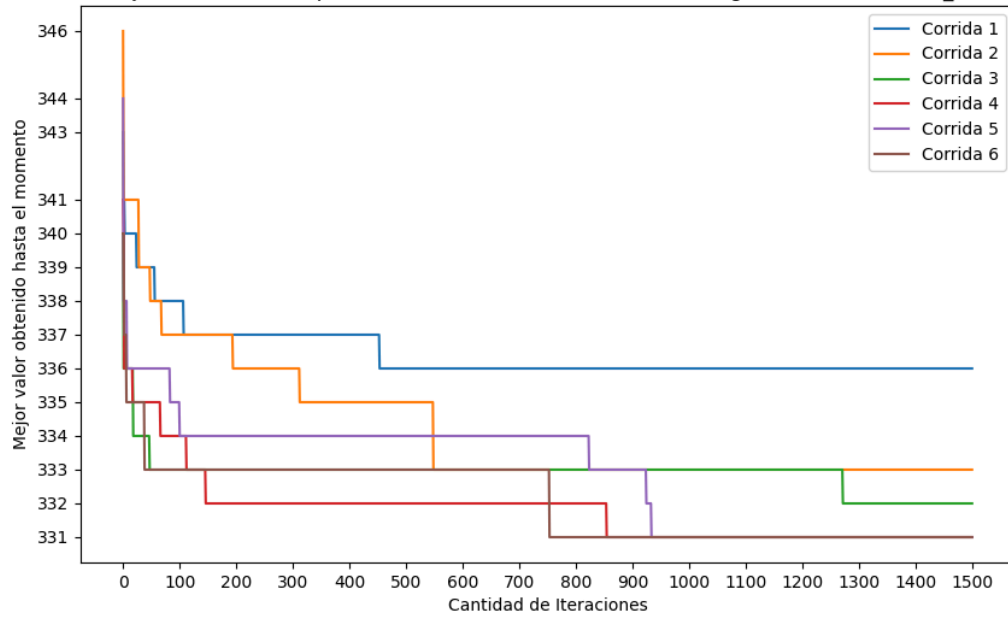
Mejor valor obtenido por iteración en diferentes corridas del algoritmo, instancia 10_700



Mejor valor obtenido por iteración en diferentes corridas del algoritmo, instancia 15_300



Mejor valor obtenido por iteración en diferentes corridas del algoritmo, instancia 15_500



Mejor valor obtenido por iteración en diferentes corridas del algoritmo, instancia 20_300

