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:

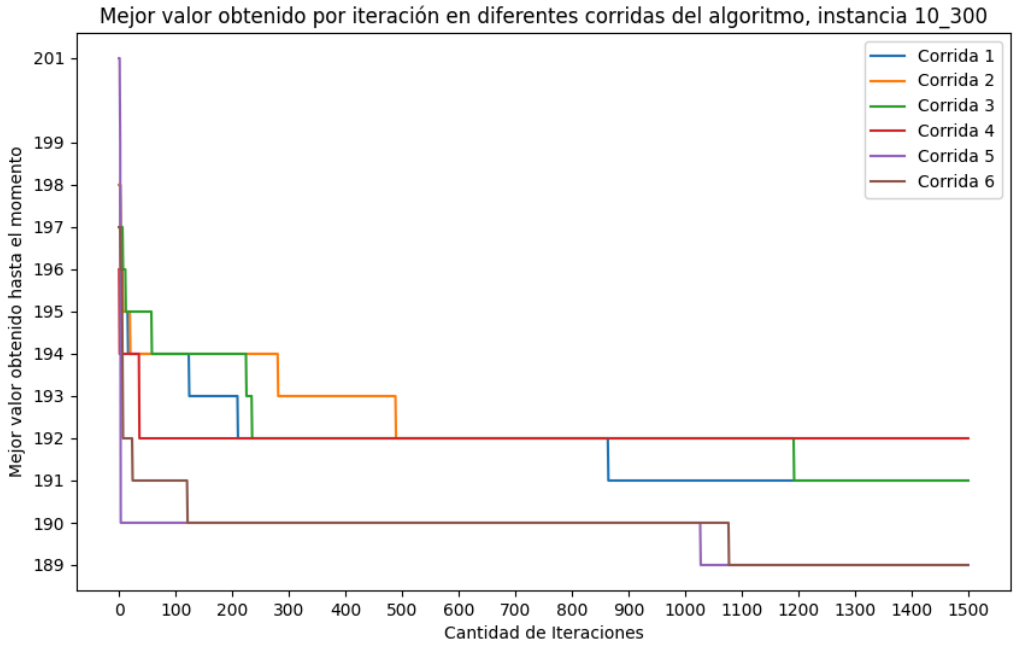
Además de la cantidad de iteraciones del GRASP, existen otros puntos críticos sobre los cuales se pueden experimentar y en base a los resultados proponer las implementaciones más convenientes. Por simplicidad de este trabajo y para mantener el foco sobre el punto crítico mencionado en la introducción, 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 decision:

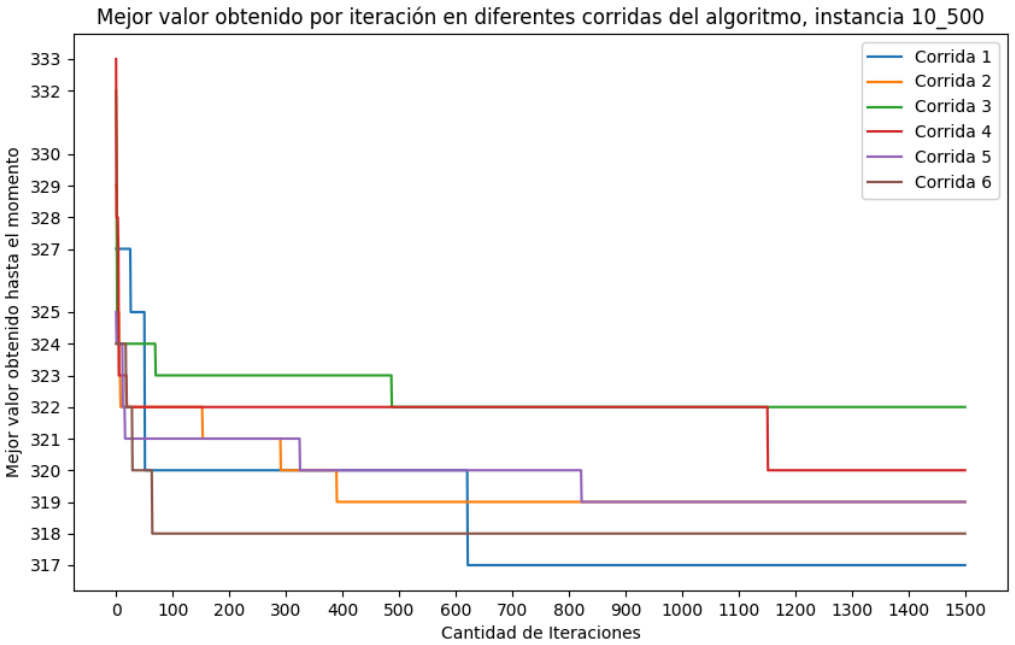
- Nivel de randomización de la heurística **Greedy-Randomizada**: El tamaño de la lista de candidatos de donde se elegirá el caracter que pasará a formar parte de la solución para cada posición de la misma, estará dado por la raíz cuadrada del tamaño del alfabeto utilizado, se optó por dicha solución 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.

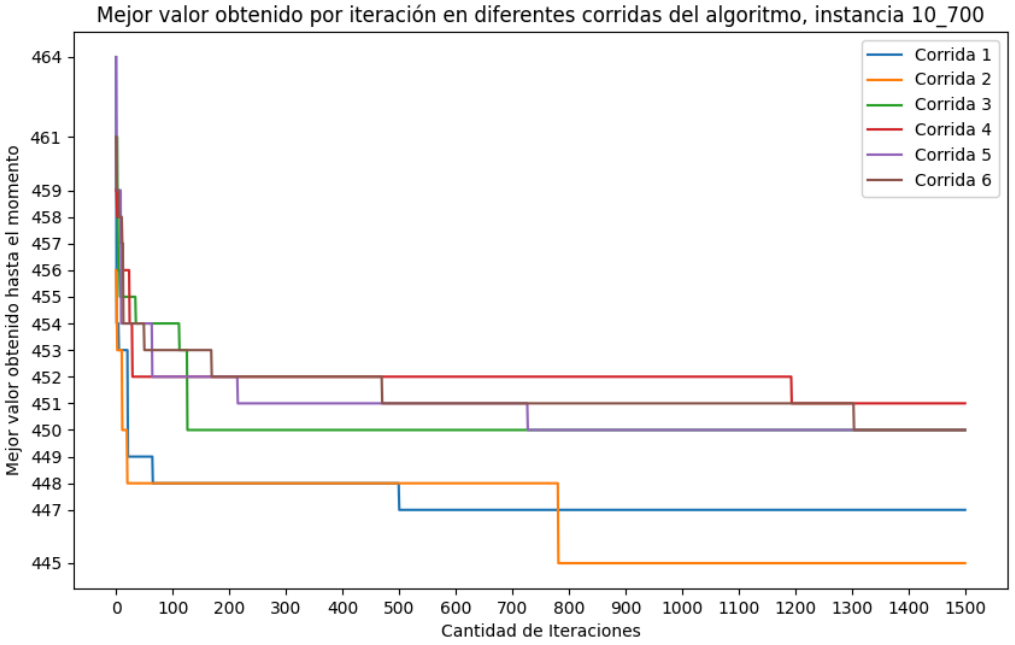
- Cantidad de iteraciones de la **Búsqueda Local**: Se decidió que la misma sea como máximo de 100 iteraciones, esto es así ya que realizando algunas pruebas sobre las instancias dadas, se notó que a la búsqueda local le tomaba en promedio unas 25/30 iteraciones encontrar el mínimo local.

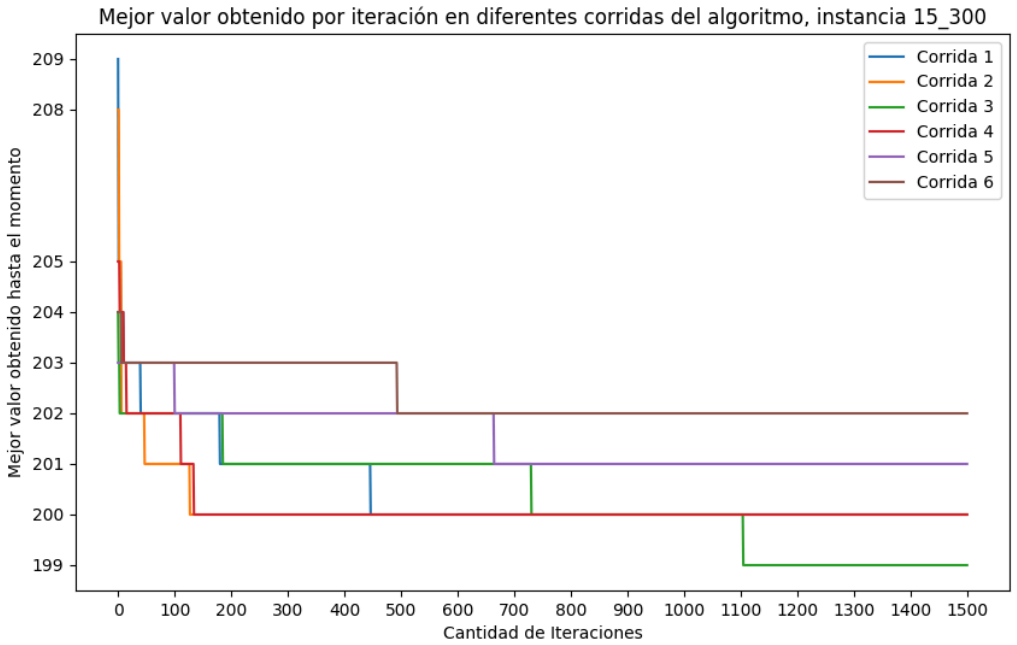
Gráficos Iniciales:

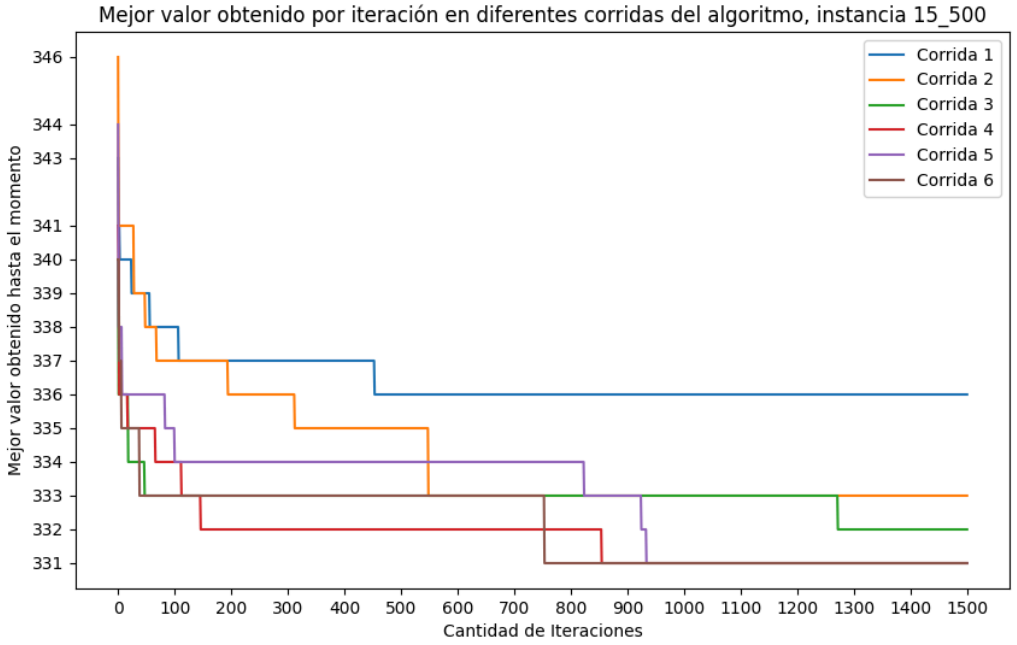
A continuación se presentan 9 gráficos realizados sobre 9 instancias distintas del problema, los cuales muestran la relación de la **cantidad de iteraciones** y el **mejor valor obtenido hasta el momento**, para varias ejecuciones sobre cada una de las instancias utilizadas.

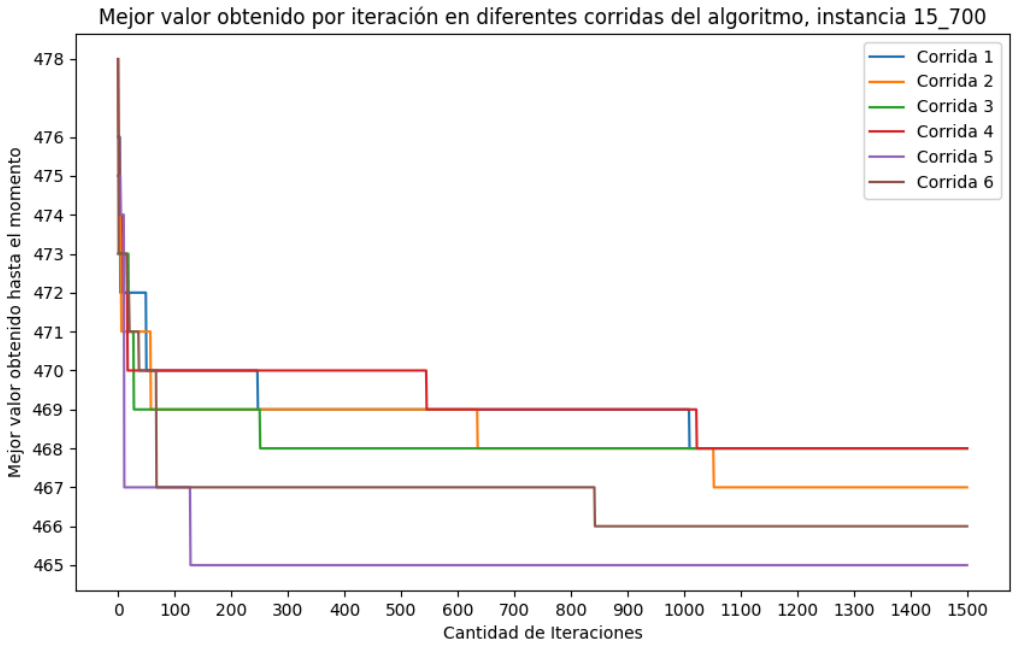


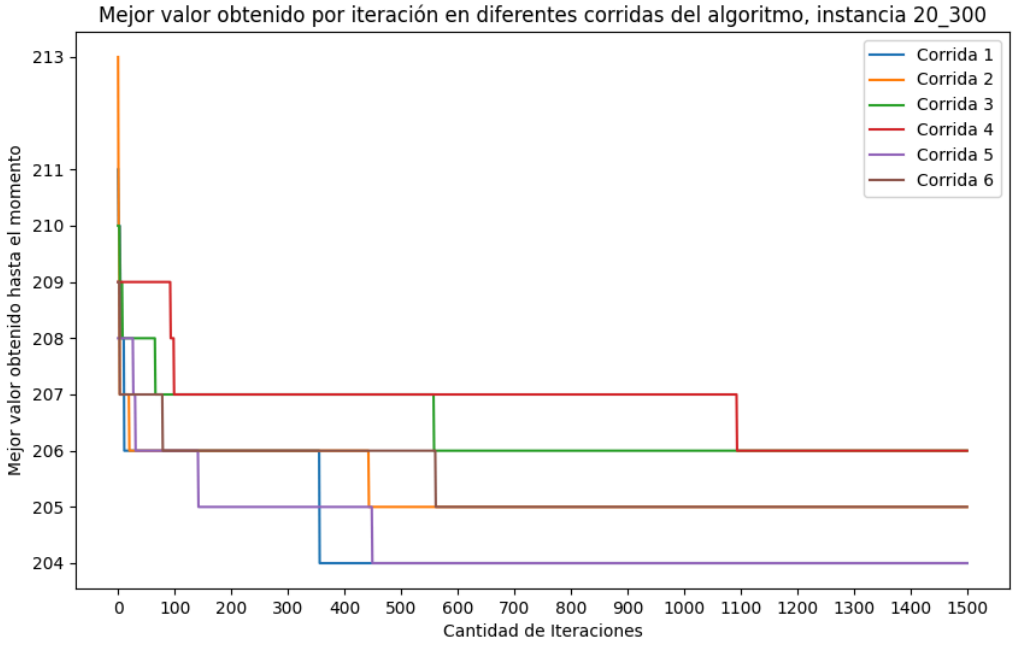

Mejor valor obtenido por iteración en diferentes corridas del algoritmo, instancia 10_500
Mejor valor obtenido hasta el momento
Cantidad de Iteraciones
Corrida 1
Corrida 2
Corrida 3
Corrida 4
Corrida 5
Corrida 6
333
332
330
329
328
327
325
324
323
322
321
320
319
318
317
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500

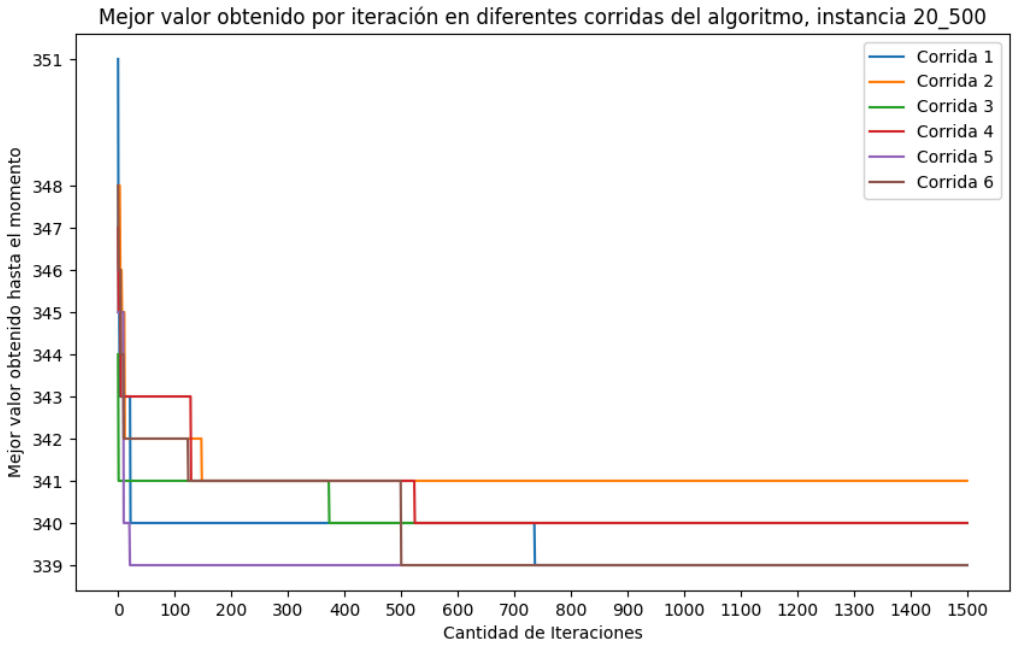

Mejor valor obtenido por iteración en diferentes corridas del algoritmo, instancia 10_700
Mejor valor obtenido hasta el momento
Cantidad de Iteraciones
Corrida 1
Corrida 2
Corrida 3
Corrida 4
Corrida 5
Corrida 6
464
461
459
458
457
456
455
454
453
452
451
450
449
448
447
445
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500

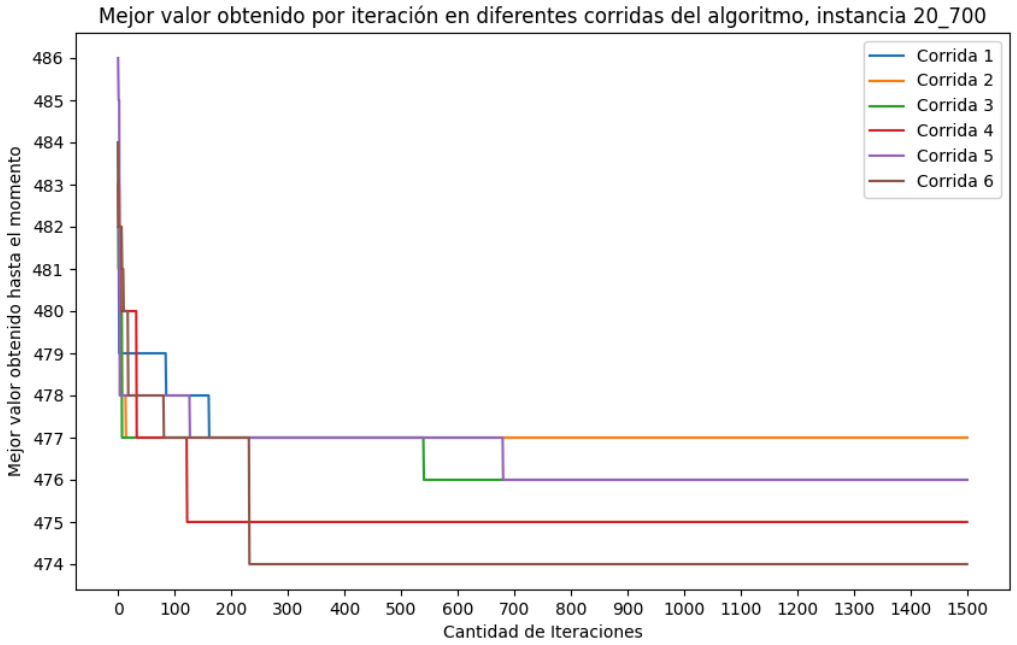

Mejor valor obtenido por iteración en diferentes corridas del algoritmo, instancia 15_300
Mejor valor obtenido hasta el momento
Cantidad de Iteraciones
Corrida 1
Corrida 2
Corrida 3
Corrida 4
Corrida 5
Corrida 6
209
208
205
204
203
202
201
200
199
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500


Mejor valor obtenido por iteración en diferentes corridas del algoritmo, instancia 15_500
Mejor valor obtenido hasta el momento
Cantidad de Iteraciones
Corrida 1
Corrida 2
Corrida 3
Corrida 4
Corrida 5
Corrida 6
346
344
343
341
340
339
338
337
336
335
334
333
332
331
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500


Mejor valor obtenido por iteración en diferentes corridas del algoritmo, instancia 15_700
Mejor valor obtenido hasta el momento
Cantidad de Iteraciones
Corrida 1
Corrida 2
Corrida 3
Corrida 4
Corrida 5
Corrida 6
478
476
475
474
473
472
471
470
469
468
467
466
465
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500


Mejor valor obtenido por iteración en diferentes corridas del algoritmo, instancia 20_300
Mejor valor obtenido hasta el momento
Cantidad de Iteraciones
Corrida 1
Corrida 2
Corrida 3
Corrida 4
Corrida 5
Corrida 6
213
211
210
209
208
207
206
205
204
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500


Mejor valor obtenido por iteración en diferentes corridas del algoritmo, instancia 20_500
Mejor valor obtenido hasta el momento
Cantidad de Iteraciones
Corrida 1
Corrida 2
Corrida 3
Corrida 4
Corrida 5
Corrida 6
351
348
347
346
345
344
343
342
341
340
339
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500

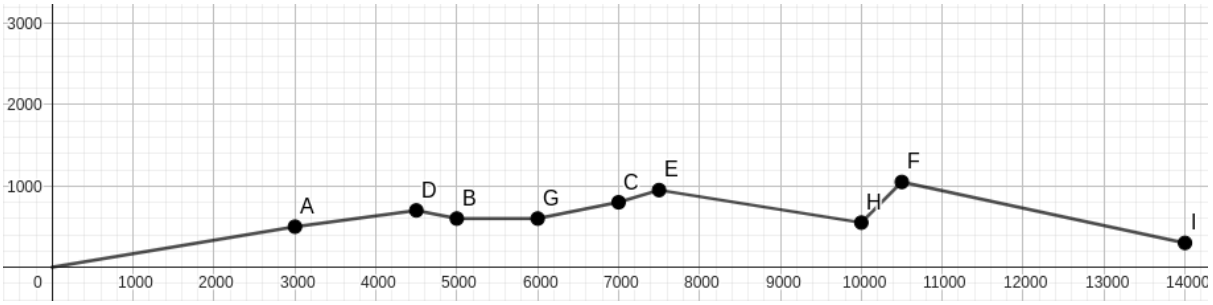

Mejor valor obtenido por iteración en diferentes corridas del algoritmo, instancia 20_700
Mejor valor obtenido hasta el momento
Cantidad de Iteraciones
Corrida 1
Corrida 2
Corrida 3
Corrida 4
Corrida 5
Corrida 6
486
485
484
483
482
481
480
479
478
477
476
475
474
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500

Conclusiones sobre los gráficos iniciales:

A partir de un análisis de los resultados de las pruebas, representados en los gráficos iniciales, se puede concluir para cada instancia, cual fue la cantidad de iteraciones necesarios para obtener resultados buenos:

- **Instancia 10_300:** 500 iteraciones (A).
- **Instancia 10_500:** 600 iteraciones (B).
- **Instancia 10_700:** 800 iteraciones (C).
- **Instancia 15_300:** 700 iteraciones (D).
- **Instancia 15_500:** 950 iteraciones (E).
- **Instancia 15_700:** 1050 iteraciones (F).
- **Instancia 20_300:** 600 iteraciones (G).
- **Instancia 20_500:** 550 iteraciones (H).
- **Instancia 20_700:** 300 iteraciones (I).

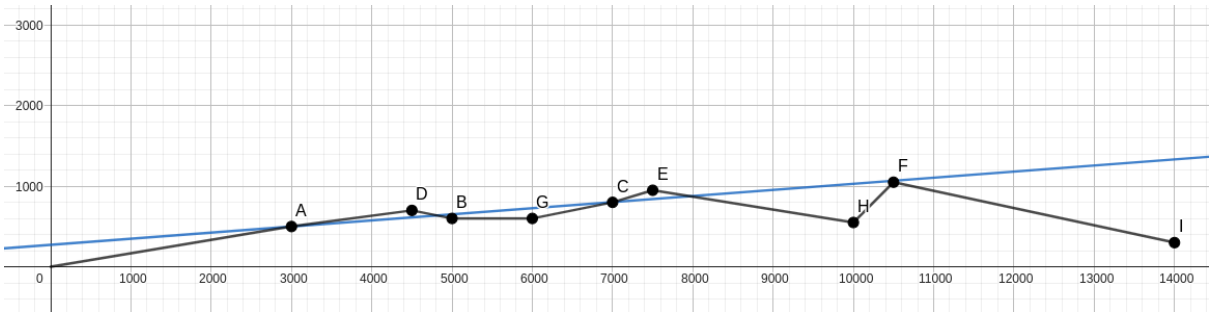
Los datos anteriores, son representados en el siguiente gráfico, donde el **eje x** representa el tamaño de la instancia (**M*N**) y el **eje y** representa la cantidad de iteraciones:



A partir del gráfico anterior, se puede hacer una aproximación lineal para obtener una función de dicho tipo lineal, que luciría como y = m x + b, donde el input de dicha función es la cantidad de textos y el largo de dichos textos y el resultado de la misma es la cantidad de iteraciones necesarias para obtener resultados óptimos globales, o con valores muy próximos a estos.
Dicha función resultante de la aproximación lineal es

$$f(m,n) = 0.0758 * (m * n) +271.7741$$

Y se vería de la siguiente manera con respecto al gráfico anterior:



Conclusión:

En este informe, se presentó el enunciado del problema del texto más parecido, y las características que debía tener nuestra solución GRASP, se mencionó también cuáles eran los puntos críticos dignos de análisis de dicha solución, pero que por simplicidad del informe no iban a ser tenidos en cuenta. Se presentaron los gráficos generados con los resultados de las pruebas sobre diferentes instancias y para cada uno de ellos, se determinó la cantidad de iteraciones más adecuada para alcanzar una solución buena, por último con dichas estimaciones se elaboró un último gráfico que nos sirvió para determinar una función que en base al tamaño de la instancia nos retorne la cantidad de iteraciones más conveniente para obtener un buen resultado.