

Tópicos de Especialidad IV - Optimización Entrega 2 – Implementación

Carlos Manchego¹ David Molina²

^{1,2}*Universidad Andrés Bello, Facultad de Ingeniería, Santiago, Chile.*

27 de noviembre de 2018

Resumen. Uno de los mayores desafíos que se presentan a ciertas organizaciones es la gestión de distribución de sus productos mediante una flota de vehículos a un grupo determinado de localizaciones, es así como surge la pregunta: ¿cuál será el mejor conjunto de rutas que me permita abarcar y satisfacer todas las localizaciones?, entendiéndose como mejor, las rutas que entregan ciertas variables (tiempo, recursos, etc.) con mejores resultados. Este tipo de problema de optimización combinatoria es conocido como Problema de enrutamiento de vehículos (VRP, por sus siglas en inglés.). Dentro de este tipo de problema (VRP) existen distintas variantes, una de estas es Milk Collection Problem, la cual busca satisfacer la demanda de una planta procesadora yendo a buscar los distintos tipos de leche a las granjas existentes, estableciendo ciertas restricciones como el deterioro de la calidad al peor tipo de los mezclados, visitar cada granja a lo más una vez, entre otras.

1. Introducción

Las cooperativas de leche se enfrentan a la gran presión de los mercados, donde es muy complicado competir, por lo que se hace de vital importancia el aumento de la eficiencia en todos y cada uno de los aspectos de mercado para lograr recolectar la mayor cantidad de leche con la mejor calidad.

Las granjas productoras de leche se encuentran dispersas en áreas rurales extensas, donde estas producen distintos tipos de calidades o grados, por ello es necesario optimizar la recolección de leche de las distintas granjas que se encuentran distribuidas evitando así que la calidad de leche disminuya. Es un proceso en el cual intervienen distintos factores y restricciones, por lo que conseguir un conjunto de rutas para su recolección es un reto para cualquier industria lechera. Es por ello que para identificar la ruta que garantiza el mínimo costo y con la mejor calidad es un problema de VRP. Se han desarrollado innumerables tipos de aplicaciones que proveen soluciones factibles, con distintos tipos de arreglo en función del tipo de operación, el objetivo, y las restricciones a considerar.

Existen distintos tipos de enfoques para la recolección de leche, donde se abordan diferentes tipos de calidades, dentro de estos enfoques se considera la recolección por camiones separados o con compartimientos separados para las distintas calidades o un enfoque de mezcla de leche. El problema puede dividirse en dos fases, la primera es la fase de asignación, la cual consiste en hallar una asignación conveniente de granjas productoras. La segunda es la fase de ruteo y consiste en construir las rutas de forma tal que una todas las granjas asignadas. [7].

En el presente trabajo se resuelve el problema enfocado a la recolección de leche con mezcla (MB), mediante un algoritmo propuesto de manera particular y realizando la técnica de búsqueda incompleta Hill-climbing + Restart, la cual se enfocada en avanzar sólo en soluciones de mejor calidad, proporcionadas por su vecindario en un determinado óptimo local, además posee la particularidad de realizar Restart, es decir, reinicia la búsqueda en otras regiones del espacio de búsqueda. Este algoritmo resolutivo estará implementado con Mejora-mejora, vale decir, al momento de generarse el vecindario de una solución, éste tomará la mejor de tal vecindario.

2. Definición del Problema

El problema que se estudia en el presente informe consta de una variante de un Problema de enrutamiento de vehículos (VRP), conocido como Milk Collection Problem. Este problema busca establecer y dar mejor solución al inconveniente presentado al momento de recolectar la leche. Existen empresas que necesitan recolectar la leche en distintos puntos (granjas) para llevarlas a su planta procesadora, a su vez las granjas que proveen de esta leche están dispuestas geográficamente a distintas distancias, alejadas de la planta. Además, en cada granja se generan diferentes calidades de leche por lo que se complica la tarea de la recolección (para no mezclarlas), por lo cual en primera instancia se estableció un sistema de compartimientos para los vehículos recolectores con el fin de no mezclar las diferentes calidades de leche. Esta solución si bien es cierto operativamente logra su cometido, es muy costosa económicamente y de implementar.

Con el objetivo de maximizar los beneficios de las empresas es que surge el método de mezcla de las leches en la recolección. Este proceso tiene la particularidad que al mezclar las distintas leches (distintas calidades) de las granjas, éstas se deterioran a la de peor calidad, esto por lo tanto, implica una mayor planificación en la gestión ya que se debe cumplir con una cierta cuota de leches de distinta calidad para la planta procesadora. Otras restricciones presentes en este problema son la visita de cada granja a lo más una vez, la carga de toda la leche de la granja al momento de visitarla, y que cada vehículo de recolección debe salir de la planta procesadora al inicio y debe volver al final del día. Además, es posible deteriorar la calidad de la leche en la planta procesadora si fuese necesario para cumplir con las cuotas. Lo que en definitiva busca resolver este problema es realizar rutas eficientes, encontrando el óptimo que permita satisfacer las necesidades de la planta procesadora y obtener la mayor ganancia cumpliendo con las restricciones respectivas.

Existen otros tipos de variantes de este problema como por ejemplo “a milk collection

problem with incompatibility constraints” [5]. el cual plantea el problema de vehículos recolectores con compartimentos y sin usar mezcla, que deben recolectar distintos tipos de leche de granjas para las cuales a ciertos vehículos (grandes) le es difícil el acceso, este problema busca disminuir la flota de vehículos y minimizar la duración del recorrido. Otro muy parecido al problema que se aborda en este informe es “The milk collection problem with blending and collection points” [3] el cual además añade la posibilidad de luego de recoger la leche de los productores (granjas), entregar ésta en puntos de recolección. Lo interesante de este problema está en dónde es mejor posicionar esos puntos de recolección y cuáles serán las granjas asignadas a los puntos de recolección, todo esto con el propósito de encontrar rutas eficientes para minimizar los costos en la logística.

Además, este tipo de problema de ruteo de vehículos es posible de aplicar para otros contextos de la misma índole como en el caso de la recolección de desechos luego de una previa clasificación de los materiales reciclables. Bing et al. (2014). Es así, como el problema presente logra reunir las características necesarias para ser estudiado.

3. Estado del Arte

Los estudios relacionados con el Milk Collection Problem van desde el desarrollo de modelos para su resolución como algoritmos para resolverlos, y procedimientos heurísticos. En los últimos años se han visto avances en materia de documentación respecto a esta variante VRP con lo que se ha logrado plasmar el problema y aplicar según metodologías las soluciones encontradas. Además, luego de una búsqueda y según ciertos estudios señalan, el Milk Collection Problem with blending no había sido analizado anteriormente en la literatura sino hace un par de años atrás. Uno de los estudios más relevantes en esta materia es el de [3], el cual enfoca el problema a la situación actual en Chile y cómo se han ido ciñendo a este cambio de utilizar mezcla en la distribución de la leche, producido principalmente por los costos que conlleva el proceso.

En la figura (1) vemos como distintas rutas de 2 camiones (Camion 1: línea continua y Camión 2: línea discontinua) realizan el recorrido de las siguientes maneras: (a) Proceso de recolección sin mezcla y (b) Proceso de recolección con mezcla. Para este ejemplo que presenta este estudio se deja en evidencia el beneficio de ocupar el proceso de mezcla, aunque en este caso al ser sólo dos calidades de leche, A y B no se podría mezclar en los camiones, si le logra ver beneficiado el caso (b) ya que la mezcla se produce en la planta. Es así, como aportando la capacidad de mezcla más que una limitante en el proceso, sólo sirve como beneficio en el caso que se requiera. De esta imagen, vemos que luego finalizado el proceso, en el caso (b) se obtuvo menores costos y mayores ingresos, por lo tanto mayor ganancia. Pero a su vez esto no siempre es así, lo que busca este proceso al final del día es obtener mayores ganancias y para ello hay veces en las cuales no necesariamente se obtienen costos más bajos que otro proceso o ingresos más altos. En la sección 5.3 Figura 1.

El método propuesto por este estudio establece 2 casos principales para la resolución del problema:

- Establecer un modelo entero mixto y aplicar algoritmos basados en branch-and cut.

- Aplicar una heurística de 3 etapas para las instancias grandes.

Este método de resolución propuesto, permite en primer lugar y para instancias de tamaño medio, definir el modelo de resolución, y en este caso plantea la función objetivo en la sección 5.2 ecuación 1.

La primera parte de la fórmula busca maximizar las ganancias, que es el ingreso percibido de la leche que ingresa a la planta procesadora y la segunda parte resta el costo de transporte, para obtener mayor rentabilidad en el proceso.

Además, en ocasiones el visitar algunas granjas distantes de baja producción podría ser más costoso que el beneficio obtenido de la recolección de su leche. En este caso, si existen acuerdos entre las granjas y la planta, estas granjas podrían ser pagadas por su leche, pero no visitadas. Este caso es una variante del problema de enrutamiento de la recopilación de precios [1].

Luego de definir el modelo lo que se plantea mediante los estudios es generar diferentes algoritmos basados en Branch-and-cut para ir realizando cortes (cumplimiento de cierta desigualdad) generando el cumplimiento de otras restricciones del modelo.

Luego para cada corte se opera con algoritmos de separación, estos algoritmos de separación se utilizan en cada corte para encontrar qué restricciones y cortes agregar en cada iteración. Una vez que ya no se encuentran nuevos cortes, el procedimiento continúa con otra rama y un paso, después de lo cual los algoritmos de separación se aplican nuevamente, y así sucesivamente. La figura(2) permite mostrar más que una comparación frente a otros métodos de solución, una verificación de la solución del problema para instancias de tamaño medio. Estas instancias fueron provistas del estudio de [2]. y permiten mostrar los resultados en granjas de tamaño desde 32 a 80.

En este caso de prueba se utilizaron 3 camiones con capacidad suficiente para transportar la leche requerida, y el ingreso por litro es de 1.0, 0.7 y 0.3 unidades para las calidades A, B y C, respectivamente.

Como vemos la mezcla en los camiones se utiliza en 11/22 veces y en la planta 8/22, y en 14/22 casos al menos una de estas mezclas fueron utilizadas, por lo tanto el porcentaje de utilización de esta técnica es alto, y permite lograr un beneficio en la oportunidad de recolectar en contra de una recolección separada.

Por otro lado, cuando las instancias son grandes (aproximadamente mayor a 100) se hace necesario para lograr mejores resultados en términos de tiempo, el uso de heurísticas para lograr una solución del problema de manera eficiente. En el estudio analizado de Paredes-Belmar se hace mención a otro realizado por [4] en el cual se utiliza la heurística The Tabu-Search (TSH) para resolver el problema de recolección de leche pero con puntos de recolección. Lo que realiza esta heurística es dividir las grandes instancias en subproblemas y resolver estos por separado. Este método de segmentación de un problema es diseccionado de mejor manera en el estudio de [8]. Para lograr un agrupamiento de las granjas en cada segmento se recomienda el uso de K-means intentando como se mencionaba anteriormente que cada grupo no exceda de las 100 granjas (nodos), ya que localmente se vería igualmente influenciado por el tiempo en su resolución.

Lo que permite este método de agrupamiento K-Means es asignar a cada granja a un

segmento o clúster mediante sus distintas distancias, este es un método rápido, eficiente y muy utilizado en la literatura de problemas de optimización [6].

Por otra parte, la limitación que tiene este método de agrupamiento es que no toma en cuenta ciertas situaciones que se dan en la realidad y que son muy comunes, como podría ser el paso por carreteras en mal estado, montañas u otras variantes, ya que la distancia euclidiana con la que trabaja este método podría ser corta pero en la realidad no. Es así como este análisis debe realizarse caso por caso y para algunos estudios se mantiene como constante (al igual que los contratos con cada productor).

La figura 3 muestra el resultado del proceso de división de una región en 25 áreas independientes.

Luego de esta etapa ya dividida se utiliza un modelo para asignar camiones y asignar los requisitos de leche a los grupos, siguiendo con el algoritmo branch-and-cut para encontrar la solución óptima a cada área.

La figura (4) muestra el resultado del caso real de una empresa Chilena que recolecta leche de 500 granjas distribuidas en una región geográfica (la misma de la figura anterior) de aproximadamente 9600 km². Este resultado se muestra en base a la comparación del uso de esta técnica de mezcla frente a la recolección por separado (sin mezcla) y actual de la empresa, y el proceso del VRP sin mezcla.

4. Modelo Matemático

A continuación se presenta el modelo matemático el modelado para resolver el problema de recolección de leche con mezcla. El modelo consiste en definir las variables, parámetros y restricciones que se presentan en el problema. Se tiene que uno o más vehículos deben partir y terminar en la planta procesadora. Cada granja puede ser visitada a lo más 1 vez, siendo la distancia la menos posible con la mejor calidad. Cuando una granja es visitada se carga toda la leche. Se debe cumplir con el mínimo de cuota de cada tipo de leche requerida por la planta procesadora. La calidad de una leche puede ser degradada cuando se combina con una de peor calidad, es decir si el vehículo carga con leche tipo A y se visita una granja de leche con tipo B, la leche resultante es B. Se tiene un Grafo no dirigido, sea G (2) el conjunto de nodos, donde G_0 es el vértice que identifica a la planta productora. i es el conjunto de vehículos y j es el conjunto de tipos de leches. $C(i)$ (6) indica la capacidad que posee el vehículo, $L_c(j)$ (7) es la cuota de la leche de tipo j , $L_v(j)$ (8) Valor de la leche tipo j , $G_c(k,j)$ (9) es la cuota que posee la granja k de leche tipo j , $D_c(k,j)$ (10) es la distancia desde el nodo x al nodo y .

El objetivo (11) maximizar la ganancia, donde se calcula el ingreso de la leche recibida por calidad de cada vehículo i , menos el costo de la ruta realizada por el vehículo i . La restricción (12) verifica que la capacidad del vehículo no sea sobrepasada. La restricción (13) valida que las granjas no sean visitadas más de una vez. La restricción (14) modifica el valor de la cuota de la granja a cero siempre y cuando esta haya sido visitada. La restricción (15) valida que la cuota de cada tipo solicitado por la planta cumpla con el mínimo requerido.

5. Propuesta

Nuestra propuesta de solución se enmarca en la utilización de un algoritmo de resolución, utilizando una técnica de búsqueda incompleta llamada Hill-climbing + Restart. Esta técnica se enfoca en avanzar siempre a soluciones de mejor calidad por ciertas regiones del espacio de búsqueda. No tiene la capacidad de empeorar soluciones por lo que al momento de no poder avanzar se realiza un Restart para que aleatoriamente vuelva a caer en alguna nueva región, y así comenzar con una nueva búsqueda local.

Por otro lado, esta solución será implementada con Hill-climbing versión Mejora-mejora, es decir, al momento de avanzar por el espacio de búsqueda, se avanza por medio de los vecinos, de los cuales se elegirá la mejor solución de éstos, que a su vez mejoren la solución actual. Luego, el mejor vecino pasa a ser la solución actual y así hasta encontrar siempre una mejor solución con la cual seguir avanzando en el espacio de búsqueda.

A continuación se presenta la representación de las soluciones y la función de evaluación con la cual se guía el algoritmo de resolución.

5.1. Representación

La representación de las soluciones estará dada por un número de 3 Tours (Rutas) de nodos (Granjas), a través de arreglos unidimensionales de datos (Vectores). Cada uno de estos arreglos contienen en la primera y última posición el número “1” el cual representa a la planta procesadora.

En la Figura 5, se muestra un ejemplo de 15 Granjas, numeradas desde 2 hasta 16, para un caso de recolección de 3 distintas categorías de leches. Cada fila representa una ruta para una categoría de leche específica, y el conjunto de filas, una solución. Al comienzo y al final de cada fila se localiza el id de la planta procesadora.

Esta representación en arreglos nos proporciona un correcto manejo tanto de las posiciones, al permitir controlar el orden de visita de cada granja, como de los valores al realizar Swap entre éstos.

5.2. Función de evaluación

La función de evaluación se encarga principalmente del: Costo total de transporte en las rutas. Este costo de transporte determinado por el trayecto del camión recolector, establece la suma de los costos de ir de granja en granja. A su vez, al realizar el proceso de búsqueda local con la solución actual (random), ésta irá moviéndose siempre a rutas de mejor calidad, es decir, con un menor costo de transporte en ella.

El siguiente ejemplo muestra en la figura 6 el cambio en los costos de transporte al momento de la búsqueda local en una determinada ruta.

6. Descripción del algoritmo

La propuesta de resolución mediante el algoritmo Hill-climbing se basa en el siguiente algoritmo genérico, de éste se establece el procedimiento general de resolución y la lógica principal. Cada uno de los secciones que componen al algoritmo general siguiente es explicado en detalle para la resolución particular realizada:

Algorithm 1 Procedure hill-climbing + Restart

```

 $t \leftarrow 0$ 
initialize Sbest
repeat
   $local \leftarrow FALSE$ 
   $Sc \leftarrow$  select a point at random
  evaluate Sc
  repeat
    select  $Sn$  the best quality point in  $N(Sc)$ 
    if  $f(Sn)$  is better than  $f(Sc)$  then
       $Sc \leftarrow Sn$ 
    else
       $local \leftarrow TRUE$ 
    end if
  until  $local$ 
   $t \leftarrow t + 1$ 
  if  $Sc$  is better than Sbest then
     $Sbest \leftarrow Sc$ 
  end if
until  $t = MAX$ 

```

6.1. Sección 1

En esta sección comenzamos inicializando las variables involucradas en la resolución, la lógica principal del algoritmo está centralizada dentro de un procedimiento llamado “principal()”. Posteriormente, para el ciclo de búsqueda local de soluciones (está limitado por la variable MAX, la cual se ingresa por consola como argumento) se debe inicializar en cada iteración una solución actual de manera aleatoria. La elección de esta solución está determinada mediante una función “GeneratorsolutionsRandom()”, la cual se encarga de generar tales soluciones a través de ciertas restricciones, las más importantes en primera instancia es que la cuota de leche para una categoría particular no supere la capacidad del vehículo.

A su vez, la asignación aleatoria de granjas es realizada en un vector, seleccionando estos valores dentro de un dominio de valores posibles para cada ruta, es decir: Sólo valores de clase A para la leche de clase A, valores de clase A y B para la leche de clase B, y valores de

clase A, B y C para la leche de clase C. Luego de esto se va realizando el cálculo del costo de la ruta, asignando el costo del vecino anterior a cada nodo cuando se tiene la ruta completa.

Por último, se verifica que el total de la cuota que lleva el vehículo sea mayor o igual a lo exigido por la planta y que además tal cuota sea inferior o igual a la capacidad que tiene el vehículo. Si se cumple esta condición quiere decir que tenemos una solución actual generada de manera aleatoria, sino se realiza una nueva iteración en el bucle hasta encontrarla.

6.2. Sección 2

Esta sección se encarga de iterar mediante un bucle hasta que la variable local sea distinta de False. En esta iteración lo que se realiza es el avance del algoritmo (la búsqueda local), es decir, la selección del mejor vecino dentro del vecindario generado. Esta selección se realiza mediante la función “BestNeighborhood()”, la cual se encarga en primera instancia de recorrer la solución actual mediante un ciclo for, en este recorrido se produce un movimiento Swap (intercambio), en los valores entre la primera posición de la solución actual (arreglo) y las demás posiciones restantes. Este intercambio entre posiciones de la solución actual nos permite saber desde ya que las soluciones candidatas que se generarán cumplen con las cuotas, ya que son las mismas granjas dispuestas diferentemente en la ruta, facilitando la tarea de discriminar por los costos de transporte. A su vez, se va calculando la distancia de estos vecinos y seleccionando cuál es el mejor de ellos. Luego retornará el mejor de todos los vecinos generados.

Ya retornado el mejor vecino generado a partir de la solución actual (la cual llamaremos “candidata”) que se envía a “BestNeighborhood()”. En el procedimiento “principal()” se procede a realizar una comparación entre estos. Esta comparación se produce en los costos de estos dos arreglos, si la solución candidata es mejor que la actual en términos de costos de transporte, la solución candidata pasará a ser la actual y el algoritmo seguirá iterando en el ciclo do while hasta encontrar siempre una mejor solución candidata que la actual, y con ello avanzar hacia un óptimo local. Si eso no sucede, es decir la solución candidata actual es mejor que la candidata, el algoritmo no podrá avanzar más en el espacio de búsqueda, por lo tanto, se saldrá del bucle (local = True), y es tiempo de realizar un Restart.

6.3. Sección 3

En esta última sección del algoritmo lo que se busca es dar control tanto a la cantidad de Reset como a las mejores soluciones que se irán obteniendo. En primera instancia, cuando se sale del bucle do while ya que (local = True), es decir no se encontró una solución candidata vecina mejor que la actual, se procede a incrementar la variable “t”, la cual se encarga de posteriormente comparar con la variable “MAX” si el número de Restart es menor o igual a la cantidad máxima de Restart permitidos. Si no se ha superado este número, se procede a guardar la solución actual como la mejor solución hasta ahora.

Luego de estos pasos, se procede a realizar un Restart, el cual volverá a realizar todo el procedimiento descrito desde la sección 1. Es decir, generar una nueva solución aleatoria e ir avanzando en busca de mejores soluciones vecinas.

Cuando se terminan los Restart se actualizan las granjas (un arreglo para saber cuáles han sido visitadas y cuáles no, hasta este punto). Posteriormente, en las variables globales donde se guarda la información de las mejores rutas, se procede a realizar la actualización correspondiente. Hasta este punto ya podríamos tener una solución, pero eventualmente podrían sobrar granjas que no han sido visitadas, ya que con las actuales si se cumple con la cuota requerida.

Por último, fuera ya de todos los ciclos se procede a obtener la lista de las granjas hasta ahora no visitadas, las cuales tendrán que ser añadidas a alguna ruta correspondiente. Por medio de la función “insertarNoVisitado()”. Esta función se encarga de ir añadiendo cada una de estas granjas de manera iterativa, en las rutas donde cumpla las restricciones dependiendo de su clase de leche, y actualizando los costos de estas nuevas rutas. Se podría dar el caso donde no es posible añadir todas las granjas ya que la capacidad del vehículo se podría encontrar limitada para una clase particular, y ésta tampoco podría entrar en otro tipo de vehículo por el mismo motivo. Finalmente, se procede a escribir estas variables globales contenedoras de las rutas y sus costos en un archivo de texto.

7. Experimentos

Los siguientes experimentos tienen por objetivo comprobar inferencias, esto a partir de los resultados que se obtienen. El problema VRP a resolver mediante Hill-climbing, la cual es una técnica incompleta, y por lo tanto no determinista, provee a través de una entrada distintos resultados en su salida, los cuales se comparan y se analizan en la sección de resultados. Los experimentos que se tratan tienen que ver en base a variaciones. Estos experimentos son:

7.1. Mayor restart - Mejores resultados

Lo que se trata de comprobar con este experimento es qué tanto podrían mejorar los resultados al otorgarle un mayor número de reinicios al algoritmo. ¿Se podría dar el caso en que para la misma instancia un resultado sea mejor con menor cantidad de reinicios que otro? Es una posibilidad, considerando que la solución inicial es random, y por lo tanto, podría caer en un “mejor” óptimo local. Sin embargo, la lógica nos dice que al tener una mayor cantidad de oportunidades de reiniciarse (mayor Restart), tendrá mayor oportunidad de encontrar mejores resultados.

7.2. Mismas Instancias y número de Restart - Variación de resultados

Para una misma cantidad de instancias de entrada y número de Restart, lo que se espera comprobar es qué tanto es el cambio, la variabilidad de los resultados. Es decir, cómo al caer en otros puntos de inicio de manera aleatoria puede implicar en obtener mejores resultados, y qué tan bueno serían éstos.

8. Resultados

Los resultados en las tablas 1 y 2 están asociados a los experimentos antes descritos, cada uno contiene una explicación y conclusión particular. Además, se expresan sus gráficos y tablas correspondientes de los datos.

La tabla 1, entrega la información referente a la variación de la ganancia respecto a distintos reinicios al momento de la búsqueda. Se entrega la información para 3 instancias distintas (36,65 y 76), cada ejemplo implementado con la misma variación en la cantidad de Restart.

Como se puede desprender de esta información, existe una notoria relación de variación en cada una de las 3 instancias (Figura 7,8 y 9), esta variación corresponde a una tendencia en alto en la ganancia total, vale decir, a mayor cantidad de restart, existe una mayor probabilidad de encontrar mejores resultados en la ganancia total. Por otra parte, esta tendencia no es estrictamente regular siempre ya que existen casos en los cuales la ganancia baja, esto tiene sentido ya que una nueva solución podría eventualmente no caer en la misma región y/o caer en regiones peores.

Los resultados en la tabla 2 muestran la variabilidad de los datos frente a una misma instancia y número de Restart. Para este ejemplo se tomaron resultados de 2 distintas instancias (36, 75), de los cuales se realizó la ejecución en 5 ocasiones, con un número de 5 Restart.

Como se percibe en la tabla 2 no existe una alta diferencia para la ganancia total, entre los distintos resultados para una misma instancia y número de Restart, se aprecian variaciones pero éstas no superan el 6.5 % en cualquiera de los dos casos. Por otra parte, podemos ver que si existe una alta variación para distintas instancias (a36 , tai75A) en la ganancia total. Esto se produce principalmente por el incremento de combinaciones que toma el problema al crecer el número de granjas. De todas maneras se aprecia una ganancia por leche similar en cualquiera de las instancias y en cualquier caso de éstos, esto dado principalmente por la restricción de la capacidad del vehículo.

9. Conclusiones

Para los estudios analizados existe la concepción transversal de establecer un modelo matemático que cumpla con una función objetivo, la cual debe garantizar que los beneficios sean mayores que los costos y con ello obtener una rentabilidad superior al método actual de recolección. A su vez, que las restricciones sean representativas del problema y abarquen cada punto.

Por otro lado, mediante la propuesta de solución expuesta se puede ratificar una alternativa real de solución, relativamente rápida para los descritos anteriormente y con un buen desempeño en la ganancia total. Además, se identifica y comprueba la ventaja de Hill-climbing+Restart sobre sólo Hill-climbing, ya que el primero guarda una mejor solución y abarca más regiones del espacio de búsqueda otorgando por ende mejores resultados. La implementación si bien es cierto no es tarea fácil para este problema, se entiende que

es posible de adaptar a muchos problema de optimización combinatoria y obtener buenos resultados. También, se encuentran buenas y mejores soluciones en la medida que damos la posibilidad de hacer más Restart al algoritmo, ayuda también en este sentido que se elige el mejor vecino del vecindario generado de la solución y con esto obtener una ventaja en término de los costos.

Una de las ventajas que no es aprovechada por esta solución, es la capacidad de poder tomar soluciones que empeoren la función de evaluación, obviamente controlada mediante un análisis de memoria, ya que se podrían recorrer muchas más regiones del espacio de soluciones y acrecentar las posibilidades de encontrar una mejor solución.

Finalmente, una de las estrategias más prometedoras son el uso de metaheurísticas como el caso de TSH ya que provee y beneficia de gran manera el correcto uso de los movimientos entre nodos, mediante una memoria. Además, para que sea ocupado en diferentes rubros y en tamaños de nodos diversos, un método de agrupación más sensible sería perfecto.

Referencias

- [1] E.Balas. The prize collecting traveling salesman problem. networks. pages 621–636, 1989.
- [2] Augerat et al. Computational results with a branch and cut code for the capacitated vehicle routing problem technical report rr 949-m. *University Joseph Fourier, Grenoble, France*, 1995.
- [3] G. Paredes-Belmar. The milk collection problem with blending and collection points. 2017.
- [4] K. Sethanan and R. Pitakaso. Differential evolution algorithms for scheduling raw milk transportation. 2016.
- [5] M. Caramia y F. Guerriero. A milk collection problem with incompatibility constraints. pages 130–143, 2009.
- [6] J.A. Hartigan y M.A Wong. Algorithm as 136: a k-means clustering algorithm. pages 100—108, 1979.
- [7] S. Fojo y V. García. Clustering aplicado al problema vrp con múltiples depósitos y ventanas de tiempo. 2001.
- [8] D. Xu y Y. Tian. A comprehensive survey of clustering algorithms. ann. data sci. pages 165–193, 2015.

Anexos

9.1. Ecuaciones

La primera parte de la fórmula (1) busca maximizar las ganancias, que es el ingreso percibido de la leche que ingresa a la planta procesadora y la segunda parte resta el costo de transporte

$$Z = \text{Max} \sum_{t \in T} \sum_{r \in T} \alpha^r v^{tr} - \sum_{(i,j,k) \in AK} c_{ij}^k x_{ij}^k \quad (1)$$

Las ecuaciones para el modelo matemático son las siguientes:

k = 0,...,k-1 Granjas y Planta procesadora.
i = 0,...,i-1 Conjunto de vehículos.
j = A,B,C ó 1,2,3 Tipo de leche.

Variables:

$$G_k \begin{cases} 1 & \text{Si la granja k fue visitada.} \\ 0 & \text{Sino.} \end{cases} \quad (2)$$

$$V_{gik} \begin{cases} 1 & \text{Si Si el vehículo i visita la granja k.} \\ 0 & \text{Sino.} \end{cases} \quad (3)$$

$$A_{ixy} \begin{cases} 1 & \text{Si el vehículo i viaja desde al nodo x al nodo y; } x, y \in k \quad ; x \neq y \\ 0 & \text{Sino.} \end{cases} \quad (4)$$

$$V_{l_{ij}} \begin{cases} 1 & \text{Si el vehículo i carga leche del tipo j.} \\ 0 & \text{Sino.} \end{cases} \quad (5)$$

Parámetros

$$C_i: \text{Capacidad del vehículo i.} \quad (6)$$

$$Lc_j: \text{Cuota de la leche tipo j.} \quad (7)$$

$$Lv_j: \text{Valor de la leche tipo j.} \quad (8)$$

$$G_{C_{kj}}: \text{Cuota de la granja k de leche tipo j.} \quad (9)$$

$$D_{xy}: \text{Distancia desde el nodo x al nodo y ; } x \neq y \quad (10)$$

$$\text{Max} \sum_{1 \in j} \left(\sum_{1 \in k} G_{C_{kj}} * V_{gik} \right) * lv_j * Cl_{ij} - \sum_{1 \in x} \sum_{1 \in y} A_{ixy} * D_{xy}; \forall i \quad (11)$$

Restricciones

$$X = \sum_{1 \in k} \sum_{1 \in j} G c_{kj} * V g_{ik} \leq C_i; \forall i \quad (12)$$

$$\sum_{1 \in k} G_k \leq k; \forall i \quad (13)$$

$$G c_{kj} = 0 \text{ Si } G_k = 1 \quad (14)$$

$$\sum_{1 \in i} X_i \geq G c_{0j}; \forall j \quad (15)$$

9.2. Figuras

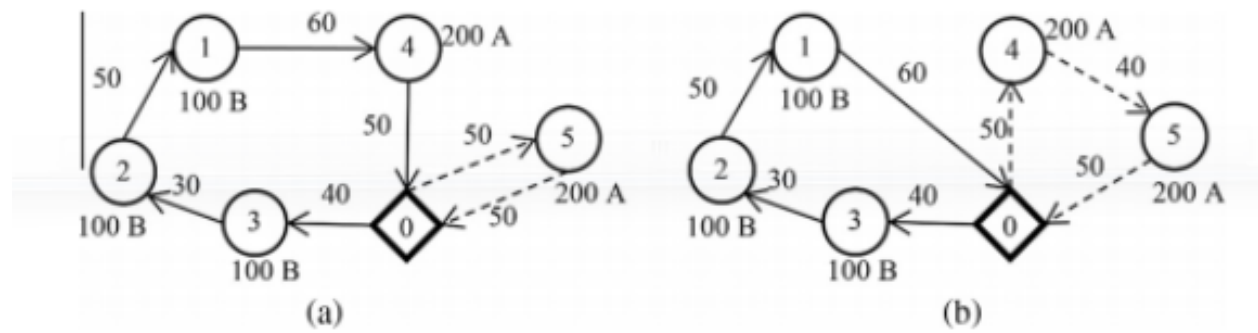


Figura 1: Efectos de la mezcla en la planta procesadora.

Tabla 1: Instancias de prueba a36, a65 y eil76.

Instancia de Prueba	Cantidad de Resetart (MAX)	Ganancia Total	Costo total de Transporte	Ganancia por leche
a36	5	38539.1	960.875	39500
	50	39771.2	728.826	40500
	500	40139.3	660.696	40800
	1000	39618.1	681.924	40300
	5000	39410.3	689.687	40100
	10000	40910.2	689.83	41600
a65	5	58899.1	2405.91	61305
	50	61276.1	2078.89	63355
	500	59753.9	1551.05	61305
	1000	61401.9	1738.1	63140
	5000	61759.1	1595.93	63355
	10000	59845.1	1459.93	61305
eil76	5	135113	1286.92	136400
	50	135235	1164.58	136400
	500	135281	1119.03	136400
	1000	135402	997.563	136400
	5000	135430	969.6	136400
	10000	135472	928.21	136400

Tabla 2: Instancias de prueba a36 y tai75A.

Instancia	Resetart (MAX)	Ganancia Total	Costo Total de Transporte	Ganancia Total por Leche
a36	5	40260,6	438926	41300
		40216	285492	41200
		39044,3	368,59	40000
		39994,3	333842	41000
		39415,4	490688	40600
	1000	38730,9	267742	39500
		41530,3	270537	42300
		39720,7	190277	40400
		38829,1	250729	39500
		39927,8	315037	40700
tai75A	5	60629,7	1306,62	63355
		61154,5	992139	63355
		59304,1	1168,72	61305
		60799,9	1427,11	63355
		60993,9	1353,41	63355
	1000	59709	889143	61305
		61695,5	931452	63355
		61681,5	982901	63355
		61731	776734	63355
		61500,6	891384	63355

Net	Q	P	$ N_0 $	f	Z	VA	VB	VC	T	TB	PB
a32	[10; 15; 20]	[12; 10; 8]	32	1	26,660	16,200	10,000	14,800	23	No	Yes
a33	[15; 20; 25]	[15; 8; 6]	33	100	29,417	17,600	11,400	15,600	62	No	No
a34	[20; 20; 25]	[10; 12; 14]	34	100	30,496	15,900	16,000	14,000	40	No	Yes
a36	[20; 15; 15]	[10; 12; 14]	36	100	29,233	16,000	14,200	14,000	110	No	Yes
a37	[20; 15; 10]	[10; 8; 6]	37	100	24,837	10,000	16,200	14,500	45	Yes	No
a38	[20; 20; 10]	[10; 15; 15]	38	100	28,596	10,000	20,000	18,100	570	Yes	No
a39	[20; 20; 20]	[10; 12; 14]	39	100	30,808	14,600	17,300	16,600	110	No	No
a44	[25; 20; 15]	[20; 16; 12]	44	100	38,771	23,400	16,000	17,600	101	Yes	Yes
a45	[25; 20; 20]	[20; 18; 18]	45	100	40,282	23,300	18,000	18,000	136	Yes	No
a46	[30; 25; 20]	[16; 17; 18]	46	100	40,696	22,400	19,900	18,000	66	No	No
a48	[30; 25; 20]	[20; 20; 20]	48	100	39,800	20,100	20,000	22,500	230	Yes	Yes
a53	[30; 30; 30]	[20; 20; 20]	53	30	46,662	23,490	24,870	23,250	183	No	No
a54	[15; 15; 15]	[5; 5; 5]	54	50	22,414	12,600	11,650	9200	304	No	No
a55	[15; 15; 20]	[5; 10; 15]	55	50	24,694	11,900	12,250	17,800	270	No	No
a60	[20; 10; 20]	[8; 12; 16]	60	50	25,041	11,800	13,650	16,000	3,565	Yes	Yes
a61	[35; 35; 35]	[30; 20; 10]	61	100	60,644	30,400	34,600	23,500	561	No	No
a62	[15; 15; 15]	[10; 11; 12]	62	50	22,917	12,500	11,000	13,150	1022	Yes	No
a63	[20; 20; 20]	[5; 10; 20]	63	50	24,447	10,050	13,600	20,000	2930	Yes	Yes
a64	[20; 20; 20]	[5; 10; 20]	64	50	24,100	11,750	10,650	20,000	5395	Yes	Yes
a65	[15; 15; 15]	[10; 12; 14]	65	50	28,046	14,350	15,000	14,500	478	Yes	No
a69	[20; 20; 20]	[10; 15; 15]	69	50	25,822	11,750	15,500	15,000	1552	Yes	No
a80	[20; 20; 20]	[16; 10; 16]	80	50	29,977	16,250	14,650	16,200	5626	No	No

Figura 2: Resultados para 22 instancias de prueba para diferentes métodos de recolección.



Figura 3: División de la región en 25 áreas y agrupación de granjas mediante K-Means.

	MB	VRP	Current procedure
Trucks	81	99	100
Revenue [MU]	23,266	24,273	22,804
Costs [MU]	10,093	12,319	18,659
Profit [MU]	13,173	11,954	4145
Milk A [l]	1,278,815	1,435,168	1,051,791
Milk B [l]	306,060	268,564	627,043
Milk C [l]	193,332	74,475	99,373
A → B [l]	0	31,436	–
A → C [l]	0	25,525	–
B → C [l]	0	–	627
CPU time (dock time) [s]	23,836 (5959)	6241 (1560)	–

Figura 4: Comparativa de resultados para un caso real de recolección.

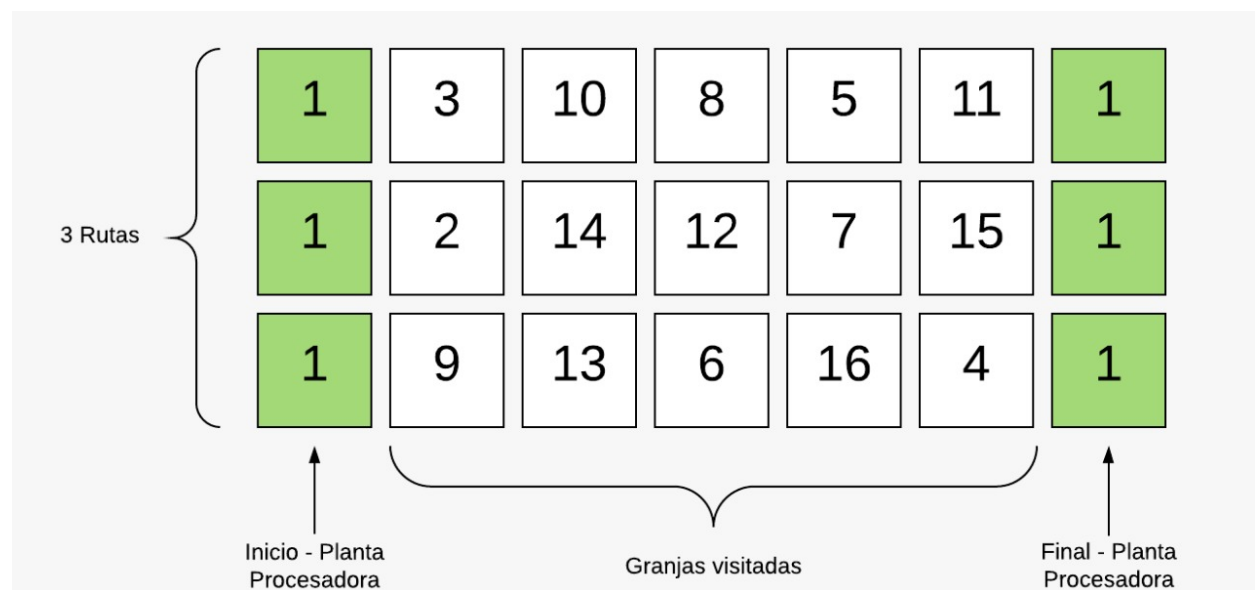


Figura 5: Vector rutas generadas.

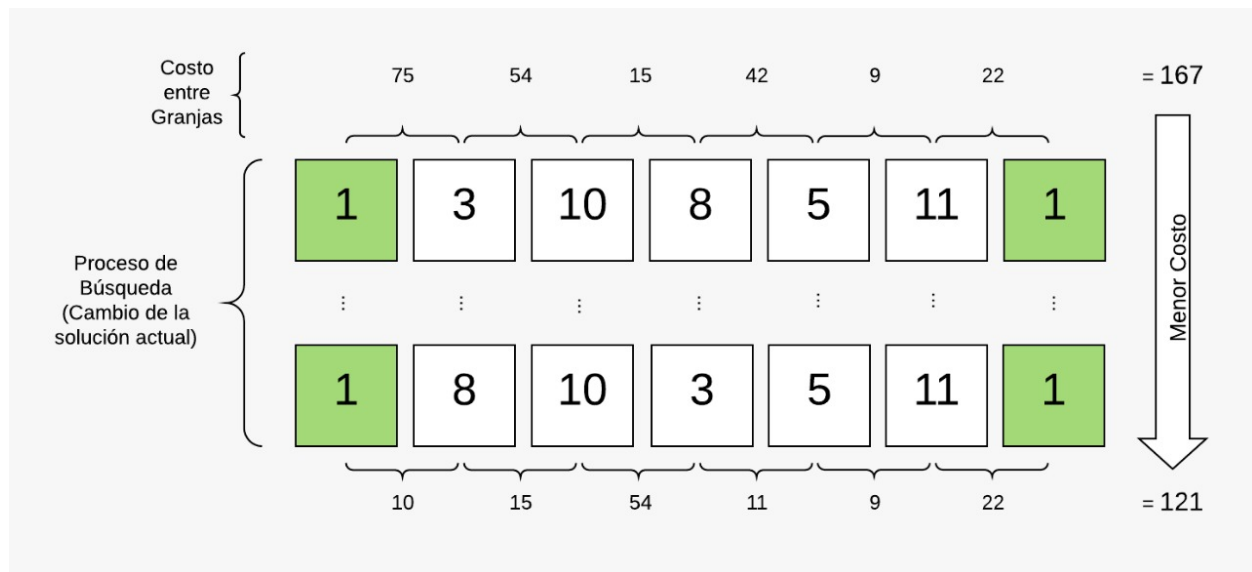


Figura 6: Vector rutas generadas.

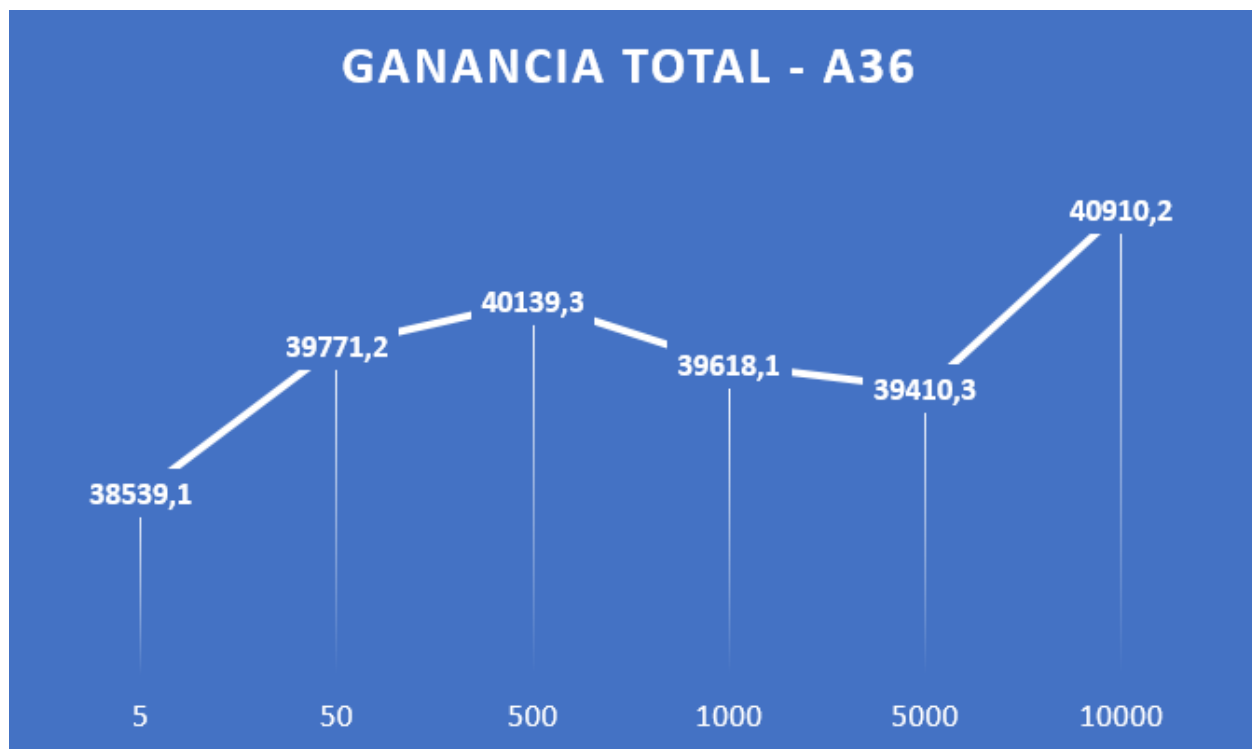


Figura 7: Instancia de prueba a36 - Ganancia Total.

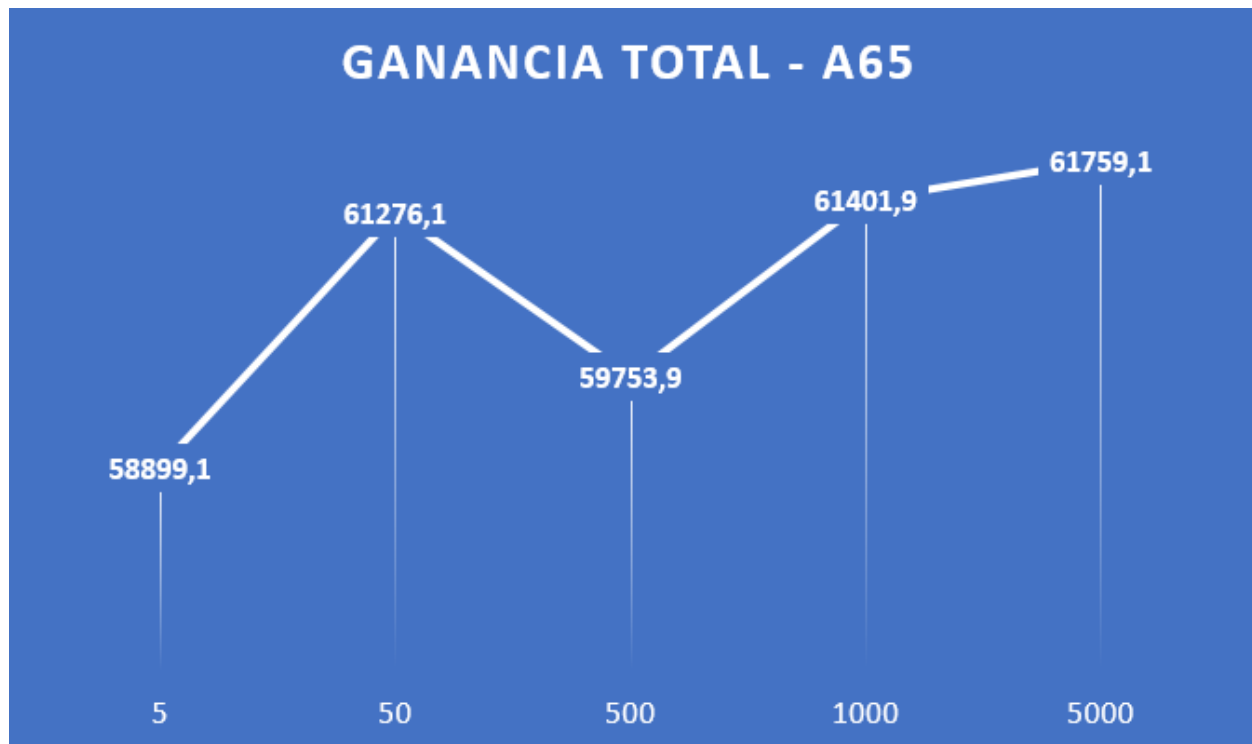


Figura 8: Instancia de prueba a65 - Ganancia Total.

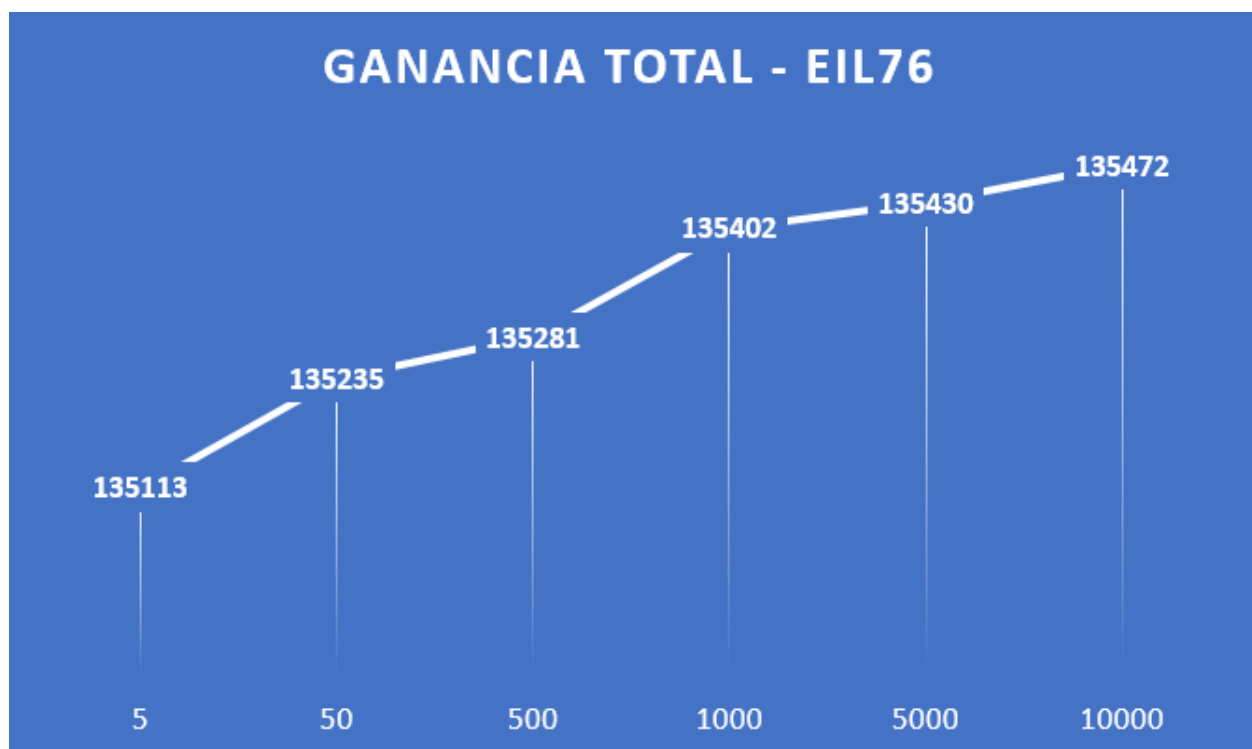


Figura 9: Instancia de prueba eil76 - Ganancia Total.