Documentación de la práctica
Búsqueda Local

IA - Subgrupo 12

Marc Nebot
Victor Teixidó
Florian Vogel

Marc Nebot
Victor Teixidó
Florian Vogel

Marc Nebot
Victor Teixidó
Florian Vogel

El problema

En este trabajo se nos planteaba el siguiente problema de planificación y distribución. Somos una empresa con diferentes centros de distribución repartidos por el mapa, todos los centros tendrán el mismo número de camiones. Además de los centros, también habrá distribuidas diferentes gasolineras a las que tendremos que abastecer cuando estas declaren una petición. Cada camión dispone de unas horas de trabajo diarias, una velocidad constante de desplazamiento y un límite de viajes diarios (todos estos datos son dados por el enunciado de la práctica). Cada cisterna será asignada cada día con las gasolineras que tendrá que visitar y en el orden en el que lo hará. Con toda esta información, y a través de algoritmos de búsqueda local, tenemos que encontrar una buena distribución de las gasolineras con tal de maximizar el beneficio de la empresa, cuanto más tardemos en resolver la petición de alguna de las gasolineras más barata le venderemos la gasolina. Hay que también tener en cuenta que, cada gasolinera puede tener entre 0 y 2 peticiones y que cada uno de los camiones deberá volver a su centro de distribución y no a uno cualquiera. Este proyecto es correcto basarlo en algoritmos de búsqueda local debido a que tenemos un problema cuya solución no es una en concreto, sino que estamos buscando una solución relativamente buena. Es decir, no buscamos una solución perfecta, sino que tratamos de buscar una que esté lo más cercana a esta hipotética solución ideal. En este tipo de problemas, los algoritmos de búsqueda local son los adecuados a utilizar para hallar estas soluciones.

Descripción de la implementación del estado

La representación del estado de nuestro problema, para ser una buena representación, tendría que mostrar para cualquier instante, cuál es el recorrido que van a hacer ese día todos los camiones de todos los centros del mapa. Es decir, que gasolineras visitará y en qué orden lo hará cada uno de los camiones.

Basándonos en esto y en toda la información que se nos daba en el enunciado decidimos hacer la siguiente implementación del estado. Tendremos una clase auxiliar Truck, que representará cada uno de los camiones de forma individual. Contendrá al centro al cual pertenece, la última gasolinera visitada, si se encuentra en el centro de distribución o en la última gasolinera, los kilómetros restantes que puede recorrer y los viajes realizados ese día. Además, y como es evidente, también contendrá la lista de gasolineras que visitará ese día, dicha lista, estará dispuesta en el orden en el que dichas gasolineras se visitarán. De forma adicional, cada camión guardará para cada gasolinera la petición solventada, ya que cada camión puede tener de hasta 2 peticiones.

Por otro lado, tendremos una clase que agrupará todo el problema, es decir desde los centros a todas las gasolineras. Esta clase representará el estado del problema en cualquiera de los instantes. Dicha clase, almacenará todas las peticiones que tenían inicialmente las gasolineras y las distancias entre dos puntos cualquiera del mapa, esto último se hace por un tema de eficiencia para no tener que estar calculando constantemente las distancias entre dos puntos del mapa. En relación con lo que hemos dicho antes, esta clase, contendrá una lista de los camiones ya mencionados. Por tanto, con esta

representación escogida, sabremos en todo momento que asignación tiene cada uno de los camiones.

Descripción de los operadores

Debido a que en este proyecto trabajamos con algoritmos de búsqueda local como puede ser el Hill Climbing, necesitaremos de operadores de cambio de estados adecuados. El trabajo de estos operadores será encontrar un máximo del beneficio lo más cercano posible al que sería el máximo real del problema. Para conseguir esto, y antes de pensar en qué operadores utilizaremos, hay que tener en cuenta lo siguiente. Nuestros operadores tendrán que cumplir con las restricciones del problema para asegurarnos de que nos mantienen dentro del espacio de soluciones. Por ejemplo, que no ocurra que al hacer un *swap* un camión se quede con una asignación de gasolineras que no podrá recorrer debido a la restricción de los kilómetros diarios. Para nuestro trabajo, y teniendo en cuenta lo ya mencionado, hemos dado con dos operadores de cambio de estado que a priori parecen ser unos buenos operadores.

El primero de estos, se trata de un *swap* que cumplirá la siguiente función. Dados dos camiones distintos, cada uno con su asignación correspondiente, intercambiaremos una gasolinera de cada uno entre ellos para obtener así un posible estado sucesor. Para esto, hemos cogido una gasolinera cualquiera de la asignación del primer camión y la hemos intercambiado por otra gasolinera cualquiera de la asignación del segundo camión. Si las rutas resultantes de este intercambio son fructíferas, es decir posibles, este *swap* se hará. En cualquier otro caso, el intercambio de gasolineras entre los camiones no se realizará. Tendrá que cumplirse que las rutas no excedan los kilómetros permitidos y, en caso de que fuera necesario, pueda regresar al centro de distribución sin ningún problema. El factor de ramificación de este operador son todas las gasolineras de algún camión diferente al propio y que, además, sean accesibles en la ruta de la asignación.

El segundo de los *swaps*, consistirá en intercambiar una gasolinera asignada a un camión por una gasolinera que en ese instante esté libre y tenga peticiones pendientes. Como hemos hecho en el anterior *swap*, tendremos que comprobar que esta nueva asignación generada es posible, habrá que asegurarse que el camión podrá recorrer todas las gasolineras que tiene asignadas. El factor de ramificación de este operador son todas las gasolineras del mapa con alguna petición pendiente y que sean accesibles por el camión.

Descripción de las estrategias del estado inicial

Las estrategias de generación del estado inicial son otras de las variables a tener en cuenta para la optimización del beneficio. Empezar con un estado o con otro, puede ser la diferencia entre encontrar un resultado aceptable o encontrar un resultado muy bueno, de hecho, así lo hemos podido comprobar nosotros mismos. Eso sí, hay que asegurarse que esta solución inicial cumple con las restricciones del problema y que por tanto, se encuentra

dentro del espacio de soluciones, podría si no darnos resultados finales que en verdad no deberían ser soluciones. Nosotros hemos ideado el proyecto a partir de dos tipos de estrategias de generación del estado inicial totalmente distintas para ver cómo incidían en la obtención de resultados dispares.

La primera de las estrategias, ha sido implementada utilizando un mecanismo pseudo-aleatorio para comprobar qué pasaría si partimos de un escenario totalmente desconocido. Para crear este estado, lo que hemos hecho ha sido ir añadiendo a las asignaciones iniciales de los distintos camiones las gasolineras por el mismo orden en que estas eran generadas por la semilla aleatoria dada al programa. Con tal de garantizar que esta solución inicial se encontraba dentro del marco de soluciones posibles, hemos aplicado un seguido de restricciones a cada gasolinera antes de añadirla a los distintos camiones. Dado un camión en un centro de distribución, este debería poder llegar a la gasolinera con petición sin superar el límite de distancia que puede recorrer, seguidamente, calcularemos que, en caso de que fuera necesario, dicho camión podría recorrer la distancia de vuelta al centro de distribución sin superar el límite de kilómetros impuesto. Si esto se cumple, la gasolinera será asignada. Este procedimiento lo haremos para cada camión tantas veces como viajes nos queden por realizar. De esta forma acabaremos obteniendo una ruta para cada camión con las gasolineras accesibles de una forma pseudo-aleatoria y sin ningún tipo de orden o algoritmo avaricioso. Además, en cuanto a coste esta estrategia es bastante eficiente ya que no requiere de ningún tipo de ordenación o estrategia elaborada.

Respecto a la segunda estrategia, está ha sido implementada utilizando un proceso más inteligente que en el primer caso, y esto ha sido posible aplicando un pensamiento enfocado a algoritmos ambiciosos o greedy. Para crear esta segunda implementación, hemos decidido asignar a cada centro de distribución las gasolineras más cercanas y que, además, fueran accesibles. Por lo tanto, para cada camión buscamos cuál es su gasolinera más cercana y una vez encontrada, si podemos llegar sin exceder nuestros kilómetros permitidos y, en caso de que sea necesario, volver al centro, la añadimos a la ruta. Posteriormente, buscamos la gasolinera más cercana a la que nos encontramos que tenga alguna petición pendiente y que cumpla con las restricciones dadas. Después de esta, restamos los kilómetros de vuelta al centro y volveremos a buscar más gasolineras cercanas con peticiones pendientes. Esto lo repetiremos hasta que alcancemos el número máximo de viajes diarios. A diferencia de nuestra primera estrategia, con esta generación de estados iniciales, obtendremos aquellas gasolineras más cercanas al centro de distribución del camión en cuestión. En cuanto al coste, esta estrategia es bastante más costosa ya que para cada asignación hay que iterar sobre todas las gasolineras del mapa para así encontrar la más cercana posible con alguna petición pendiente.

Descripción de las funciones heurísticas

Otra de las variables importantes que hay que tener en cuenta a la hora de realizar una búsqueda local es la función heurística. Esta determinará cuál de los estados sucesores generados por los operadores se convertirá en nuestro futuro estado. La función heurística se resolverá para cada uno de los estados posibles y a través de este valor se determinará

que sucesor se escogerá como futuro estado. Para este problema hemos ideado dos funciones heurísticas. Ambas dispondrán de implementaciones triviales y que no consideran muchos parámetros, pero que, sin embargo, tienen sentido y funcionan bien.

La primera de las heurísticas es más simple. Lo único que hará será calcular la suma de los costes de los kilómetros recorridos por cada uno de los camiones. De esta manera, buscamos optimizar que los estados recorran la menor distancia posible para abaratar el coste del desplazamiento.

La segunda de las heurísticas es algo más elaborada que la anterior. En esta, calcularemos el beneficio obtenido por las peticiones resueltas y a este, le restaremos el coste de lo viajado. Con este heurístico, pretendemos maximizar el beneficio obtenido por las peticiones y, a la vez, minimizar la distancia recorrida.

Experimentos

Experimento 1

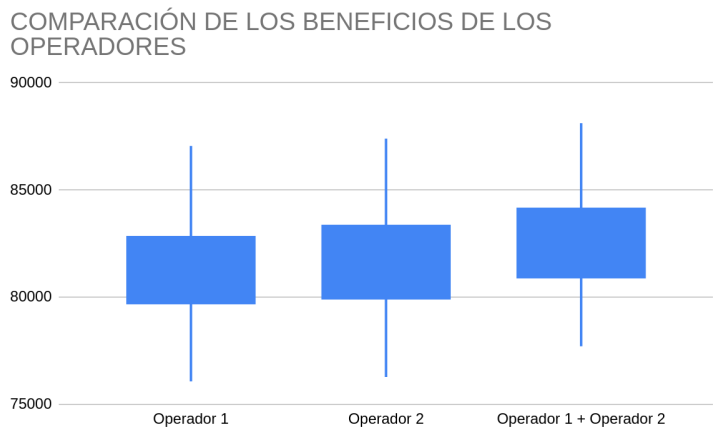
Observamos que el resultado del beneficio que nos otorgan los recorridos de los camiones pueden variar según diversos parámetros, uno de estos parámetros son los operadores de búsqueda. Hay una amplia gama de operadores de los que podemos disponer, pero a nosotros nos interesa seleccionar aquellos que nos proporcionen unos mejores resultados. Partiendo de esta base, seleccionamos dos operadores que nos parecían bastante correctos. El *swap* entre gasolineras de diferentes camiones y el *swap* entre la gasolinera de un camión y una gasolinera con peticiones pendientes. Hay que ver entonces, cuál de estos operadores nos irá mejor o, incluso, si ambos podrían servirnos.

Condiciones

En todos nuestros experimentos dispondremos del mismo número de centros de distribución y de gasolineras. En total tendremos 10 centros de distribución con 1 camión cada uno y de un total de 100 gasolineras. Dependiendo de la *seed* las ubicaciones y las peticiones de las gasolineras y los centros irán variando. Hay que destacar que para este experimento estaremos haciendo uso de Hill Climbing para la obtención de los resultados y la opción pseudo-aleatoria para la generación de estados iniciales. Respecto a la heurística utilizaremos la más elaborada, la que tiene en cuenta, además de los kilómetros, las peticiones resueltas.

Marc Nebot
Victor Teixidó
Florian Vogel

Resultados



Hemos decidido comparar la obtención de beneficios según los operadores utilizados en nuestros experimentos. Podemos observar que el espacio en el que están comprendidos los beneficios según los operadores escogidos varían. Siendo la opción de ambos operadores la que da mejores resultados.

¿Qué esperábamos y que hemos obtenido?

Para las 3 posibilidades ya mencionadas, esperábamos unos resultados medianamente satisfactorios, ya que todas tienen sentido en nuestro problema y, sobre el papel, parecen resolver nuestras necesidades y funcionar de forma bastante buena. Aun así, no estábamos seguros de si el primer operador, el swap entre camiones, iba a funcionar tan bien como los otros dos. Eso sí, apostábamos a que la combinación de ambos operadores iba a ser la que mejor resultados nos iba a proporcionar.

Después de ver los resultados, podemos corroborar que nuestras sospechas sobre utilizar ambos operadores simultáneamente si nos han dado los mejores resultados y que, por tanto, serán los operadores que utilizaremos a partir de ahora. Debemos decir también, que pese a nuestra pequeña desconfianza, el primer operador dio casi tan buenos resultados como el resto sin que hubiera una diferencia lo suficientemente grande como para descartar el swap entre camiones como un posible operador.

Comparaciones

Como ya hemos mencionado, las diferencias por separado entre ambos operadores apenas es mencionable. Estamos hablando de operadores que, en las condiciones en las que trabajan, ambos dan beneficios de entre 80000 y 90000 y que si nos ponemos a comparar ambos operadores en cada *seed* las diferencias que existen entre ambos son de un beneficio menor a 1000. Podemos afirmar entonces que ninguno de los dos operadores probados es mucho peor ni mejor que el otro. Si tenemos en cuenta el uso de ambos de forma simultánea podemos comprobar que la mejora es aún mayor que antes, pero tampoco algo exagerado.

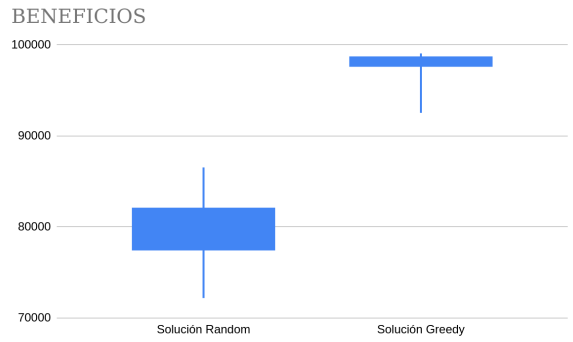
Experimento 2

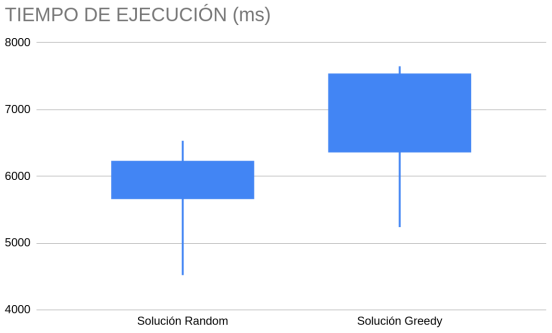
Como ya hemos observado antes, el resultado del beneficio que nos otorgan los recorridos de los camiones pueden variar según diversos parámetros, además de los operadores vistos en el experimento 1, otro de los parámetros a tener en cuenta es la estrategia de generación de la solución inicial. Para crear esta solución inicial hay tantas posibilidades como ideas tengamos, y a nosotros nos interesa seleccionar una estrategia lo suficientemente buena para que nos lleve a máximos locales satisfactorios. Partiendo de esta base, hemos ideado dos estrategias que nos parecían idóneas para el problema. La primera consistirá en asignar todas las gasolineras posibles a los camiones en orden de aparición en el mapa, podemos verla como una estrategia pseudoaleatoria. La segunda estrategia está basada en un algoritmo greedy que asignará el máximo de gasolineras posibles a cada uno de los camiones con la condición que estos camiones serán los más cercanos posibles al centro en el que se encuentre.

Condiciones

En todos nuestros experimentos dispondremos del mismo número de centros de distribución y de gasolineras. En total tendremos 10 centros de distribución con 1 camión cada uno y de un total de 100 gasolineras. Dependiendo de la *seed* las ubicaciones y las peticiones de las gasolineras y los centros irán variando. Hay que destacar que para este experimento estaremos haciendo uso de Hill Climbing para la obtención de los resultados. En cuánto a los operadores, estaremos usando la combinación de los dos operadores vistos en el experimento anterior. Respecto a la heurística utilizaremos la más elaborada, la que tiene en cuenta, además de los kilómetros, las peticiones resueltas.

Resultados





Hemos decidido realizar dos boxplots analizando la diferencia entre beneficios y tiempos de ejecución de cada una de nuestras soluciones iniciales. Como podemos ver en la primera gráfica, hay una notable diferencia entre los beneficios obtenidos por las diversas estrategias. Aún así, como es observable en la segunda gráfica, este algoritmo avaricioso, supondrá un aumento del coste temporal del programa.

¿Qué esperábamos y que hemos obtenido?

Sin haber aún probado las diferentes estrategias, tenemos buenas expectativas para ambas, ya que ambas parecen ser correctas y generan estados dentro del marco del

espacio de soluciones. Aun así, tenemos más expectativas de un buen rendimiento de la segunda estrategia ya que hacemos uso de un algoritmo greedy, por lo que de primeras, debería darnos mejores resultados. Básicamente, la primera estrategia puede llevarnos a situaciones no deseadas como que el camión haga viajes que, con sentido común, se podrían clasificar de inútiles. Se podría dar el caso que el vehículo tuviese que ir de punta a punta del mapa saltándose muchas otras gasolineras por el camino. En la segunda estrategia, apostamos por una forma más intuitiva, cercana al mundo real y con más sentido común. en la que cada centro se ocupará de las gasolineras más cercanas que tengan.

Después de haber realizado la experimentación, podemos corroborar que nuestras expectativas se han cumplido y que la idea del funcionamiento que teníamos sobre las estrategias estaba en lo correcto. A pesar de que ambas formas de crear soluciones iniciales dan unos buenos resultados, la segunda estrategia da unos beneficios mayores y por tanto, podemos concluir que es mejor. Probablemente, y como decíamos previamente, usando la primera estrategia los camiones den más vueltas saltándose así posibles gasolineras a tratar por el camino.

Comparaciones

Como mencionamos anteriormente, el aumento de beneficio respecto a las dos estrategias es bastante notable y podemos afirmar que nos saldrá más a cuenta hacer uso del algoritmo greedy para generar la solución inicial. Es observable en los boxplots que la media de la primera estrategia está alrededor de los 80000 de beneficio, lo cual es un resultado satisfactorio, pero si nos fijamos en la otra estrategia podemos observar medias de beneficios que rozan los casi 100000. Aún así, cabe destacar que al hacer uso de un algoritmo greedy el tiempo de ejecución es más elevado en esta segunda estrategia como es observable en los boxplots, a pesar de esto, el coste en tiempo es asumible para la mejora que proporciona esta segunda estrategia.

Experimento 3

Condiciones

En todos nuestros experimentos dispondremos del mismo número de centros de distribución y de gasolineras. En total tendremos 10 centros de distribución con 1 camión cada uno y de un total de 100 gasolineras. Dependiendo de la *seed* las ubicaciones y las peticiones de las gasolineras y los centros irán variando. Hay que destacar que para este experimento estaremos haciendo uso Simulated Annealing para la obtención de los resultados. En cuánto a los operadores, estaremos usando la combinación de los dos operadores vistos anteriormente. Respecto a la heurística utilizaremos la más elaborada, la que tiene en cuenta, además de los kilómetros, las peticiones resueltas.

Resultados

Para encontrar los mejores parámetros para nuestro Simulated Annealing hemos tenido que hacer muchas pruebas e ir testeando los resultados que nos daban los diferentes parámetros combinados entre sí. Hay que decir, que por mucho que cambiáramos los

parámetros, el beneficio resultante siempre era muy parecido, apenas difería nunca de más de 1000 de beneficio. Primero fuimos testeando el cambio en el número de iteraciones y cuando encontrábamos el mejor íbamos a por los siguientes parámetros. Entonces lo hicimos para todos las variables hasta dar con el mejor set de valor para nuestros parámetros. En ese set, el número total de iteraciones es 1000, las iteraciones por cambio de temperatura es 100, el parámetro *k* es 5 y el parámetro *lambda* es 0.1. Volver a destacar que, por mucho que variáramos los parámetros, el beneficio obtenido nunca variaba exageradamente.

Experimento 4

Condiciones

En todos nuestros experimentos dispondremos del mismo número de centros de distribución y de gasolineras. En total tendremos 10 centros de distribución con 1 camión cada uno y de un total de 100 gasolineras. Dependiendo de la *seed* las ubicaciones y las peticiones de las gasolineras y los centros irán variando. Hay que destacar que para este experimento estaremos haciendo uso tanto del Hill Climbing como del Simulated Annealing para la obtención de los resultados. En cuánto a los operadores, estaremos usando la combinación de los dos operadores vistos anteriormente. Respecto a la heurística utilizaremos la más elaborada, la que tiene en cuenta, además de los kilómetros, las peticiones resueltas.

Resultados

Los resultados obtenidos han sido bastante más elevados a lo esperado en lo que a tiempo de ejecución se refiere. Para el Hill Climbing fue posible hacer los experimentos hasta la iteración 5, en la que teníamos 50 centros de distribución y 500 gasolineras, este experimento tardó en ejecutarse alrededor de unas 3 horas. Viendo el crecimiento del tiempo, podemos observar que se forma una tendencia exponencial. Respecto al Simulated Annealing nos ha ocurrido algo similar pero incluso más exagerado.

¿Qué esperábamos y que hemos obtenido?

Teníamos las expectativas algo dispersas respecto a este experimento. El hecho de que a cada iteración aumentamos en 10 el número de centros y en 100 el número de gasolineras, nos hacía dudar, y temer, en que los tiempos de disparan exponencialmente. Diferenciando entre los algoritmos, suponíamos que los tiempos de Hill Climbing iban a ser más inferiores que los tiempos obtenidos con el Simulated Annealing. En resumen, pensábamos que a partir de una cierta iteración, el programa iba a ser prácticamente imposible de ejecutar debido a los largos tiempos de espera a los que nos íbamos a enfrentar

Después de obtener los primeros resultados, podemos afirmar que nuestras suposiciones no iban mal encaminadas. Respecto al Hill Climbing, las primeras iteraciones nos daban tiempos razonables. 10 centros nos comportó una espera de 6 segundos, 20

centros fueron algo más de 120 segundos, para 3 centros nos tardó 28 minutos, 4 centros el programa tardó alrededor de 1 hora y cuarto y en la quinta iteración, la última que hicimos, nos tardó poco más de 3 horas. A partir de esta, se nos hizo posible seguir testeando debido a los largos tiempo de espera, pero fue suficiente para poder ver el comportamiento exponencial de estos tiempos. Con el Simulated Annealing nos ocurrió algo parecido, solo que en menos iteraciones pudimos comprobar los altos tiempos de espera que se producían, con tan solo 3 iteraciones ya tardaba en calcular el beneficio alrededor de 3 horas.

Comparaciones

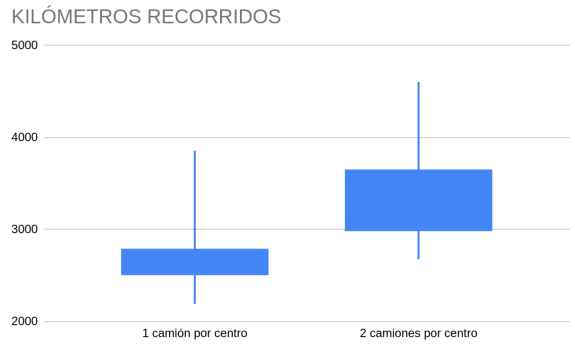
Hemos podido comprobar que tanto un algoritmo como el otro, tienen tiempos de ejecución bastante elevados a la hora de tratar con un gran número de centros de distribución y número de gasolineras. Aún así, los tiempos del Simulated Annealing evolucionaron de una forma más exponencial y rápida que los tiempos del Hill Climbing, ya que con no muchas iteraciones el tiempo aumentaba mucho.

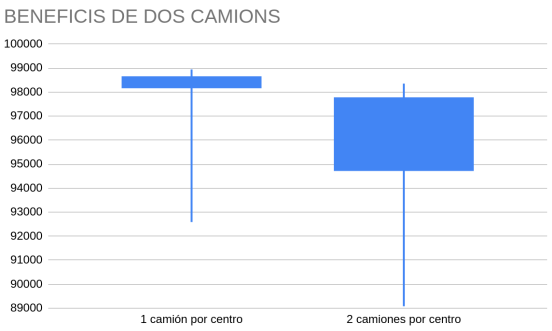
Experimento 5

Condiciones

En este experimento las condiciones difieren un poco del resto. El número de gasolineras seguirá siendo 100 y los operadores, la estrategia de generación del estado inicial y la heurística seguirán siendo las mismas que las que hemos estado utilizando hasta ahora. La diferencia estará en que los números de centros irán variando entre 5 y 10 al igual que lo hará la multiplicidad de estos, que variará entre 1 y 2. Es decir siempre tendremos 10 camiones en nuestro estado, la diferencia estará en que estarán repartidos entre 10 centros o entre 5. Además, dependiendo de la *seed* las ubicaciones y las peticiones de las gasolineras y los centros irán variando.

Resultados





En la primera de las gráficas podemos ver que, cuando los centros tienen dos camiones cada uno, los kilómetros recorridos es mayor a cuándo tienen solo un único camión. Esto también va relacionado con que los beneficios de tener un único centro son

más elevados cuando se tiene una multiplicidad mayor a uno. En este último caso, los beneficios difieren más y son menores.

¿Qué esperábamos y que hemos obtenido?

Lo que esperábamos en este experimento era lo siguiente. El hecho de tener más de un camión en un centro es contraproducente y más teniendo en cuenta en como está pensada nuestra estrategia de solución inicial. La idea óptima de este problema es que cada centro se ocupe de las gasolineras de sus alrededores pero teniendo dos camiones, lo único que se consigue es obligar a uno de ellos a tener que recorrer más kilómetros. Por tanto, nuestra suposición era que los kilómetros recorridos serían mayores al tener más de un camión por centro. Respecto al beneficio no estábamos del todo seguros, ya que podría ser que accedieramos a gasolineras más o menos cercanas que nos generaran un beneficio notable.

Los resultados vistos en los experimentos iban en camino a nuestras suposiciones previas. Los kilómetros recorridos por los camiones de los mismos centros es notablemente superior y esto se ve claramente reflejado en la primera de las gráficas. Respecto a los beneficios, los resultados obtenidos van también en relación a los kilómetros recorridos y tienen bastante sentido. Es lógico pensar que al recorrer más kilómetros esto se verá reflejado de una forma u otra en una reducción de los beneficios. Esto es comprobable en la segunda gráfica, donde podemos ver que los beneficios obtenidos son menores por los centros con una multiplicidad de camiones mayor a uno.

Comparaciones

Como ya hemos mencionado, el hecho de tener más de un camión en el mismo centro es contraproducente ya que se “quitarán” las gasolineras entre ellos y se obligarán a recorrer más kilómetros para poder abastecer peticiones. Lo óptimo en un problema como este sería que cada centro se encargará de abastecer una zona del mapa. Podemos ver como la media de kilómetros recorridos es de casi 1000 kilómetros más cuando los camiones no están bien distribuidos. Además de eso, los beneficios son mucho más dispares e inferiores en estas situaciones no deseadas. En otras palabras, mantener la multiplicidad a uno y poner más centros, siempre será mucho más rentable que poner menos centros con una mayor multiplicidad.

Experimento 6

Una de las variables que afectan al beneficio obtenido y que, tenemos en cuenta en el cálculo del heurístico para cribar a los sucesores, es la distancia recorrida. Por defecto, el enunciado nos indica que cada kilómetro recorrido tiene un coste de 2 pero durante este experimento este valor irá variando. Con este experimento veremos en cuanto afecta el coste por kilómetro a la hora de computar el beneficio y de encontrar máximos locales adecuados.

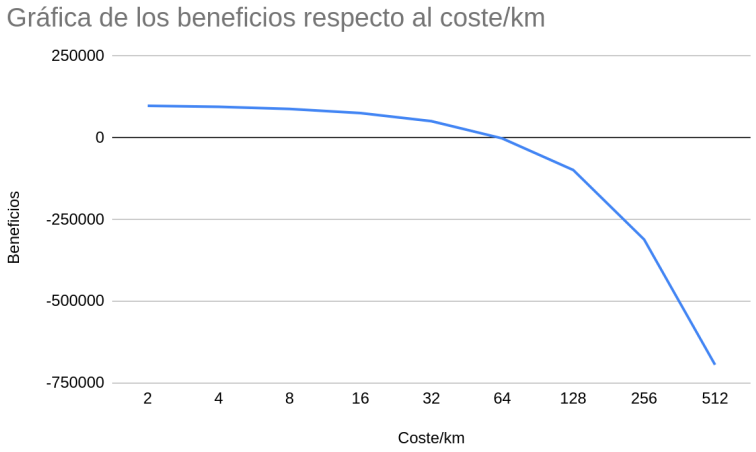
Marc Nebot
Victor Teixidó
Florian Vogel

Condiciones

En todos nuestros experimentos dispondremos del mismo número de centros de distribución y de gasolineras. En total tendremos 10 centros de distribución con 1 camión

cada uno y de un total de 100 gasolineras. Dependiendo de la *seed* las ubicaciones y las peticiones de las gasolineras y los centros irán variando. Hay que destacar que para este experimento estaremos haciendo uso de Hill Climbing para la obtención de los resultados. En cuánto a los operadores, estaremos usando la combinación de los dos operadores vistos anteriormente. Respecto a la heurística utilizaremos la más elaborada, la que tiene en cuenta, además de los kilómetros, las peticiones resueltas.

Resultados



En esta gráfica hemos querido dejar representada la relación directa que hay entre el coste por kilómetro recorrido y el beneficio total obtenido. Podemos observar cómo a medida que recorrer kilómetros se vuelve más costoso, el beneficio obtenido si hiciéramos la misma ruta se vuelve mucho menor, hasta el punto en el que sale más rentable visitar muchas menos gasolineras.

¿Qué esperábamos y que hemos obtenido?

Los resultados que esperamos de este experimento son los siguientes. Al ir aumentando poco a poco el coste por kilómetro recorrido, al seguir haciendo el mismo recorrido que hacíamos, el beneficio será, o al menos eso creemos, mucho menor. Es decir, sería mucho más importante recorrer poca distancia que no cubrir un gran número de peticiones pendientes. Por tanto, nuestra suposición es que el beneficio irá disminuyendo acorde al aumento del coste por kilómetro.

Los resultados obtenidos por el experimento confirman nuestras suposiciones respecto a la relación entre el coste de cada kilómetro con el beneficio total obtenido. Cuanto más cueste recorrer un kilómetro, menos distancia no saldrá rentable recorrer y, por tanto, menos beneficio obtendremos al hacer las entregas de las peticiones asignadas. Llegará a extremos en los que el beneficio será negativo y, por tanto, ese recorrido con ese

coste por kilómetro no sería para nada deseable. Así que, tal y como habíamos pensado, cuánto menor sea el coste por kilómetro, mejor será para el aumento de los beneficios.

Comparaciones

En el gráfico lineal podemos observar que cada vez que duplicamos el coste por kilómetro obtenemos menos beneficio llegando a tener pérdidas. Por lo tanto, al ver nuestro gráfico, podemos llegar a la conclusión de que si tenemos demasiadas pérdidas sale más a cuenta no coger peticiones muy lejanas al centro y que, por tanto, los camiones no aceptarán peticiones de las gasolineras lejanas ya que estas no generarían ningún beneficio.

Experimento 7

En este experimento veremos como afecta el aumentar en una hora el horario de trabajo de los camiones, o en otras palabras, qué pasa si en vez de recorrer un máximo de 640 kilómetros diarios, pudieran recorrer 720 kilómetros. Recordemos que suponemos que los camiones viajan a una velocidad constante de kilómetros por hora y que por tanto aumentar una hora supone poder recorrer 80 kilómetros de más.

Condiciones

En todos nuestros experimentos dispondremos del mismo número de centros de distribución y de gasolineras. En total tendremos 10 centros de distribución con 1 camión cada uno y de un total de 100 gasolineras. Dependiendo de la *seed* las ubicaciones y las peticiones de las gasolineras y los centros irán variando. Hay que destacar que para este experimento estaremos haciendo uso de Hill Climbing para la obtención de los resultados. En cuánto a los operadores, estaremos usando la combinación de los dos operadores vistos anteriormente. Respecto a la heurística utilizaremos la más elaborada, la que tiene en cuenta, además de los kilómetros, las peticiones resueltas.

Resultados

En este experimento no ha sido necesario hacer uso de gráficas ya que, tanto si aumentamos una hora el trabajo de los camiones como si les reducimos una hora, el beneficio no cambiaba en absoluto. Es por eso, que representar estos valores en una gráfica o en un boxplot no nos parecía necesario en absoluto. Los valores de los beneficios se han mantenido en los lindares vistos hasta ahora, entre los 95000 y los 100000.

¿Qué esperábamos y que hemos obtenido?

Teniendo en cuenta que la variable que irá cambiando de valor constantemente serán las horas de trabajo diarias, deducimos lo siguiente. Nos parece lógico pensar que, a cuantas más horas trabajen, más kilómetros podrá recorrer y, por tanto, más gasolineras podrán cubrir. Por otra parte, si estas horas son reducidas, los beneficios de los camiones deberían disminuir. En otras palabras, aumentando las horas que pueden conducir los

camiones aumentamos las peticiones servidas y, por tanto, aumentaremos los beneficios obtenidos. Por otro lado, si lo que hacemos es disminuir las horas, es decir decrementamos los kilómetros diarios recorridos, es de sentido común pensar que menos gasolineras podrán ser visitadas y que, por tanto, menos dinero ganaremos entregando gasolina. Si estas suposiciones se cumplen, podremos afirmar la relación directa entre los kilómetros diarios viajados y el beneficio obtenido por la empresa.

Viendo los resultados obtenidos por los experimentos, podemos contrastar nuestras previas suposiciones, ya que estas eran erróneas, al menos en nuestra implementación. Lo que delimita el beneficio en este experimento, no son el máximo de kilómetros diarios, sino los viajes diarios máximos. Si se diera el caso en el que algún camión se quedará sin kilómetros y, por tanto, le quedarán viajes libres por hacer, este incremento de los kilómetros se vería reflejado en un aumento del beneficio. En las condiciones de nuestro experimento, todos los camiones han alcanzado el límite de viajes y, por tanto, un aumento de kilómetros no tendrá ningún tipo de repercusión. Por otro lado, el decremento de una única hora de trabajo tampoco tiene ningún tipo de impacto en el beneficio por los mismos motivos ya mencionados. Si este decremento fuera mayor, entonces los beneficios disminuirían.

Comparaciones

No podemos realizar ningún tipo de comparación ya que los resultados obtenidos en las diferentes situaciones planteadas son exactamente iguales.

Comparación Hill Climbing/Simulated Annealing

En este proyecto hemos estado trabajando con algoritmos de búsqueda local, concretamente con el algoritmo de Hill Climbing y el de Simulated Annealing. Para ambos algoritmos, las estrategias de inicialización del estado, los operadores y la heurística, serán exactamente las mismas. Tanto con Hill Climbing como con Simulated Annealing, hemos obtenido resultados bastante buenos, siempre entre los 95000 y los 100000 de beneficio, lo que representa que las variables seleccionadas son eficaces y trabajan bien.

Lo que esperábamos y no hemos visto, ha sido una mejora en el Simulated Annealing respecto al Hill Climbing. En los primeros experimentos, trabajamos constantemente con este segundo y los resultados eran buenos, aún así, pensamos que al empezar a utilizar el otro algoritmo los resultados mejorarían aún más, aunque fuera un poco. Pero esto no fue así. Aunque estuvimos largos ratos probando diferentes parámetros para este, los beneficios siempre se quedaban un poco por debajo del Hill Climbing, lo que nos llevó a la siguiente conclusión. Probablemente, lo que está ocurriendo es que nuestra solución inicial, aquella que asigna las gasolineras más cercanas a los camiones, sea una estrategia *greedy* bastante eficaz lo que deja en una clara ventaja al algoritmo de Hill Climbing a la hora de buscar optimizar la solución. Cabe destacar, que el tiempo de ejecución del Simulated Annealing es mucho mayor que el otro algoritmo, estamos hablando de un incremento de más del 1500%.

A pesar de esto, y como ya hemos dicho, ambos algoritmos nos dan resultados muy buenos y ambos podrían ser utilizados obteniendo así buenos beneficios, siempre y cuando el tiempo de ejecución no sea un factor importante a tener en cuenta. Aunque si tuviéramos que elegir uno con el que quedarnos, este sería sin duda el algoritmo de Hill Climbing, ya que aunque el beneficio no sea exageradamente mayor, es verdad que siempre nos otorga un mejor beneficio y en un tiempo muy inferior al Simulated Annealing.

Trabajo de Innovación

Tema

Hemos decidido escoger como trabajo de innovación *Google Lens,* la inteligencia artificial que hay detrás de este proyecto. *Google Lens* es una cámara de la compañía *Google* que analiza lo que ves en una fotografía proporcionándole al usuario información sobre ella haciendo uso de *Machine Learning.*

Reparto

Usamos los apartados redactados del trabajo para elaborar un esquema en un documento de texto. Una vez realizado este esquema, decidimos que cada persona fuese buscando información sobre el tema escogido para después reunirse y comparar entre nosotros qué habíamos encontrado.

Referencias

1- **These Machine Learning Techniques Make Google Lens A Success,**
https://analyticsindiamag.com/these-machine-learning-techniques-make-google-lens-a-success/

2- **Why Google Lens Is The Perfect Answer For Content Based Image Recognition**
https://analyticsindiamag.com/why-google-lens-is-the-perfect-answer-for-content-based-image-recognition/

3- **Wikipedia: ¿Qué es Google Lens?**
https://es.wikipedia.org/wiki/Google_Lens

4- **Deep Learning: Architecture And Algorithms**
https://www.jimsindia.org/techbyte2k19/insight/Deep%20Learning%20Architecture%20And%20Algorithms.html

Dificultades

Encontrar información sobre el algoritmo de reconocimiento de Google Lens ha sido dificultoso debido a que parece ser que es una variante del método que se implementa en el reconocimiento de imágenes mediante *Machine Learning*.