Análisis y resolución del modelo de ruteo de pozos.

Xavier Ignacio González

# Resolución y objetivos

Cantidad mínima de cuadrillas según este análisis es **3** (2.0732). Se calcula en base a la necesidad de 85 medias jornadas de trabajo en 20.5 jornadas por mes.

Las rutas a transitar de cada cuadrilla en cada día se detallan en el archivo Excel adjunto ‘**Modelo Recolección Resuelto.xlsx**’

# Metodología

El problema expuesto es sumamente complejo de resolver por su carácter combinatorio. El mismo es una extensión del ‘**Traveling Salesman Problem**’ y considera múltiples salesman (mTSP). El problema TSP con 1 viajante es un problema muy estudiado dentro de la ciencias de la computación y pertenece al conjunto de problemas más complejos de resolver (se dice que es NP-Hard). Por supuesto que el mTSP es mucho mas complejo a su vez que el TSP.

Dada la complejidad, la resolución del problema en forma analítica (lo que sería programación lineal entera) es prácticamente inviable por el numero enorme de variables. Por lo tanto, lo que se emplea regularmente en la industria para resolver este tipo de problemas es métodos que implementan algoritmos meta-heurísticos que encuentran una solución buena, aunque no demuestran que sea la mejor solución posible.

Para la resolución de este modelo en particular se utilizó el método ‘**Local Search**’[[1]](#footnote-1) que busca iterativamente soluciones al problema modificando sucesivamente soluciones factibles que mejoran una función objetivo, en este caso, la cantidad de cuadrillas y el tiempo disponible. A su vez, se representó al problema en forma de grafo y se obtuvo la distancia mínima entre cada uno de los pozos (shortest path) usando el algoritmo **Dijkstra's**[[2]](#footnote-2).

La implementación del problema se programó en **R[[3]](#footnote-3)** (open source) y se utilizó la librería **igraph[[4]](#footnote-4)** para el modelado y la optimización del grafo. La corrida del script no demora mas de 20 minutos en una pc/notebook.

# Consideraciones del modelo y los datos

A continuación se mencionan algunas observaciones que los datos provistos presentan junto con su correspondiente tratamiento:

* Las líneas 1345 y 1426 en la pestaña de rutas están duplicadas con diferente información de nodos. Se procedió a generar dos líneas nuevas 1345b y 1426b.
* En la pestaña de puntos, existen 90 pozos con categoría ‘Fuera’ que no fueron considerados en el ruteo.
* El requerimiento de visitas provisto para los pozos categoría ‘X’ es de 0.5 veces por mes. Al ser éste un numero no entero, es imposible programar en forma correcta las visitas de la totalidad de los pozos ‘X’ en un horizonte de tiempo mensual. Para resolver esta cuestión, se seleccionó al azar la mitad de los posos categoría ‘X’ para la programación en un mes (una visita por mes). Otra opción hubiese sido programar la totalidad de los pozos en un horizonte bimestral (una visita por mes) sin mayores complicaciones más que haber hecho la codificación de la solución un poco mas compleja. El experimento podría repetirse con un horizonte bimestral de ser necesario.
* Los siguientes puntos están listados dentro de los requerimientos de programación pero los mismos no están presentes en ninguna ruta: S2394 S2505 S2650 S2651 S2669 S2679 S2684 S2686 S2691 S2693 S2703 S2705 S2710 S2712 S2713 S863. Los mismos no fueron considerados en el ruteo.
* Los siguientes puntos están desconectados de la base. Es decir, si bien están presentes como componentes en alguna de las rutas en la pestaña correspondiente, no existe ninguna trayectoria posible que conecte la base con los mismos. Por lo tanto no es posible considerarlos en el ruteo. S2023 S2216 S2335 S2384 S2456 S2493 S2515 S2530 S2540 S2556 S813 S855 S858 S900 S935

# Conclusiones

Dada la naturaleza del problema y las características de los datos, se asume que el cliente final encontrará esta solución satisfactoria, dada su simplicidad y su agilidad de procesamiento, aunque no haya garantía de que la misma sea la mejor solución posible. Para encontrar la solución teórica optima sería necesario implementar un modelo de programación lineal entera con un numero de variables exponencial al numero de nodos en la red de ruteo, lo que dificultaría mucho el modelado y su ejecución, y muy posiblemente la solución hallada no sea significativamente diferente a la presente solución.

El trabajo podría continuarse en tres direcciones no excluyentes.

* Con datos completos, podría hacerse un análisis de la solución actual y sensibilidad de las variables construyendo gráficos interactivos. Por ejemplo, si el cliente proveyese latitud y longitud de los pozos y de los nodos de la red (no necesitan ser exactos por temas de confidencialidad, podrían alterarse los datos sin perder la forma) sería posible la construcción de mapas interactivas que muestren los ruteos diarios sugeridos.
* Por otro lado, podría buscarse mejorar la solución con la metodología actual planteando una búsqueda mas eficiente y exhaustiva.
* Podría también implementarse una solución analítica (programación lineal entera) que garantice la mejor solución posible. Existe un riesgo asociado a esta dirección y es que no sea posible la implementación dada la naturaleza combinatoria del problema y el numero elevado de variables a considerar.

# Comentarios sobre el modelo Recorredores

## Correcciones:

* En el output del modelo, si bien todas las cuadrillas trabajan la misma cantidad de días, durante un turno, hay cuadrillas que se repiten y otras cuadrillas que no aparecen.

**Corregido.**

* Verificar el comportamiento del modelo al modificar los sets de datos (agregar o quitar pozos activos)

**Se incorporaron mensajes donde el script puede dar error para hacerlo mas interpretable. También se unificó la librería que lee los archivos Excel para poder reproducir futuros errores mas fácilmente. Los errores de los diferentes casos de uso podrán ser corregidos mas fácilmente.**

* Verificar que si bien dadas las categorías y frecuencias planteadas, la cantidad de puntos a recorrer da 814, el output nos dio una cantidad total de 813 puntos recorridos.

**Se verificó que el numero de pozos el la solución sea el correcto. Recordar que los pozos que no están en las rutas, y los que están en las rutas sin conexión no son considerados.**

## Requerimientos:

* Permitir la modificación de la cantidad de visitas mensuales que se deben realizar por Categoría (sólo números enteros).  Por ejemplo, cambiar la categoría A y en vez de tener 3 visitas mensuales, tener que realizar 4.

**Incorporado, se lee en la pestaña ‘categorías’.**

* Ampliar a n categorías de pozo.

**Incorporado, se lee en la pestaña ‘categorías’. Un comentario importante es que el script esta diseñado para valores de cantidad de visitas no mayores a 5 visitas (5 es el máximo) y también tiene que existir un numero considerable de pozos con exactamente 1 visita para hacer el balanceo al final de la optimización.**

* En un recorrido se deben visitar las baterías asociadas a los pozos que recorre (todos los pozos tienen una batería asociada). Para minimizar el recorrido se deberían visitar varios pozos que dependen de la misma batería para reducir las visitas a baterías. Adicionalmente, una batería no debería ser visitada por cuadrillas diferentes en el mismo turno. (en el archivo ‘xls’ servicio Recorredores que te enviamos inicialmente están detalladas las relaciones pozo batería. Esta oculta la columna pero cuando se importa a R se cargan todos estos datos)

**Este requerimiento será incorporado en proceso separado y sus resultados serán fusionados con los resultados de la optimización de ruteo standard. Para la implementación necesitaría determinar cuantas visitas a cada pozo para este esquema visitas pozo-batería? Se divide por categoría o es un numero fijo?**

* Sería deseable que la misma cuadrilla sea la que haga las visitas a un mismo pozo en un mismo período.

**Luego de implementado el punto anterior, podemos evaluar diferentes opciones de zonificación de las instalaciones.**

# Comentarios 22-julio.

## Lectura de datos:

1. Verificar que los nombres de todas columnas existan, no tengan acentos ni caracteres raros y no estén duplicados, tanto las columnas visibles como las ocultas.
2. Una vez que se fijan los nombres de las columnas en el script, los mismos no pueden ser cambiados. Ejemplo: en la pestaña categorías, visitas x Visitas, cat x categoría, etc. Puntos x Puntos de operacion
3. En esta versión del script, todas las categorías con valores de frecuencia deben tener al menos un pozo asociado. Ejemplo Categoría E, sin pozos. Se borró la línea en la pestaña correspondiente.

## Parámetros:

1. Si bien el modelo permite setear el numero de iteraciones, es recomendable que se usen valores alrededor de 20000, y si no se encuentra el optimo, repetir el ciclo tipiando la opción correspondiente. Con 12 ciclos de 20,000, la exploración es mas eficiente que con 1 ciclo de 250,000.
2. La cantidad de pozos con numero de visitas igual a 1, debe superar un mínimo (aproximadamente la mitad) similar a los primeros datos enviados. Los pozos con una visita se usan para balancear las visitas ya que los mismos no requieren verificación para ser cambiados del día de visita. En caso de requerirse casos diferentes, por ejemplo todos las categorías con mas de una visita, debe cambiarse la lógica del balanceo.

## Otros comentarios:

1. Se corrigió un problema que se ocasionaba con el redondeo de los valores de tiempos que daba un error en la ejecución.

1. https://en.wikipedia.org/wiki/Local\_search\_(optimization) [↑](#footnote-ref-1)
2. https://en.wikipedia.org/wiki/Dijkstra%27s\_algorithm [↑](#footnote-ref-2)
3. https://www.r-project.org/ [↑](#footnote-ref-3)
4. http://igraph.org/r/ [↑](#footnote-ref-4)