



Objetivos de la prueba

- Entender el problema
- Determinar una técnica de optimización para resolverlo
- Modelizar el problema

Recogida y entrega de efectivo con blindados

Problema inicial

Una empresa de seguridad presta servicios de recogida y entrega de efectivo. Un servicio queda definido por una dirección física donde realizar el servicio, una petición de recogida o entrega y una cantidad (en €) a recoger o entregar.

Esta empresa tiene un único depósito donde dispone de k vehículos blindados, todos del mismo tamaño y capacidad. Cada blindado puede hacer un máximo de MAX_KM kilómetros al día y todos tienen que empezar y acabar sus rutas en el depósito.

En un día, esta empresa recibe del orden de n peticiones de entrega o recogida de efectivo y tiene que determinar las rutas de los distintos vehículos de modo que pueda realizar la mayor cantidad de servicios posibles.

El objetivo de la empresa es generar las rutas de los blindados minimizando:

- Servicios sin realizar
- Coste del transporte

Teniendo en cuenta disponibilidad y capacidad de blindados y la jornada laboral de los vigilantes.

Además, habrá que tener en cuenta que un servicio se considerará realizado sólo si se recoge o entrega el 100% de la cantidad definida a recoger o entregar, respectivamente. Por tanto, a la hora de realizar un servicio de recogida el equipo de vigilantes del blindado tiene que asegurarse que tiene suficiente capacidad en el vehículo para recoger la cantidad del servicio. Del mismo modo, para realizar un servicio de entrega, tendrán que asegurarse de que disponen de la cantidad demandada en el servicio.

Requisito adicional

La empresa puede almacenar un máximo de efectivo (medida en €) en el depósito. Esta cantidad es de MAX_STOCK €. En caso de verse obligados a almacenar más, tienen que hacer uso de almacenes



que no tienen en propiedad, lo cual les incrementa los costes de almacenaje. De este modo, la función de costes de almacenamiento es la siguiente:

$$COSTE_ALMACENAJE = \begin{cases} K_1 & \text{si } stock \leq MAX_STOCK \\ K_2(stock - MAX_STOCK) & \text{si } stock > MAX_STOCK \end{cases}$$

con $K_2 > K_1$.

De este modo, a la hora de definir las rutas, querrán que se minimice también el coste de almacenamiento.