Trabajo Fin de Máster

# Problemas de rutas de vehículos estocásticos

Juan Carlos Gonçalves Dosantos

**Máster en Técnicas Estadísticas**

**Curso 2016-2017**

# Propuesta de Trabajo Fin de Máster

| |
|---|
| **Título en galego:** Problemas de rutas de vehículos estocásticos |
| **Título en español:** Problemas de rutas de vehículos estocásticos |
| **English title:** Problems of stochastic vehicle routes |
| **Modalidad:** Modalidad A |
| **Autor:** Juan Carlos Gonçalves Dosantos, Universidad de Santiago de Compostela |
| **Directora:** Balbina Virginia Casas Méndez, Universidad de Copostela |
| **Breve resumen del trabajo:**<br>En este Trabajo Fin de Máster se incluirá una introducción a la Programación Estocástica y se revisará el formato SMPS utilizado en el modelado de este tipo de problemas. A continuación, se estudiarán algunos modelos y soluciones existentes en la literatura de los problemas de rutas de vehículos estocásticos. |

Doña Balbina Virginia Casas Méndez, de la Universidad de Santiago de Copostela, informan que el Trabajo Fin de Máster titulado

**Problemas de rutas de vehículos estocásticos**

fue realizado bajo su dirección por don Juan Carlos Gonçalves Dosantos para el Máster en Técnicas Estadísticas. Estimando que el trabajo está terminado, dan su conformidad para su presentación y defensa ante un tribunal.

En Santiago de Compostela, a 3 de septiembre de 2017.

| La directora: | El autor: |
| --- | --- |
| Doña Balbina Virginia Casas Méndez | Don Juan Carlos Gonçalves Dosantos |

# Índice general

# Resumen

## Resumen en español

En la actualidad, existe una mejora significativa en la recogida de todo tipo de datos. Esto ha provocado que la mayoría de problemas tratados en investigación operativa pasen a ser más realistas y dinámicos.

Entre ellos destacamos los problemas de rutas de vehículos, que han incorporado nuevas restricciones aplicadas a datos en tiempo real y parámetros estocásticos. Un caso concreto de esta situación es considerar las demandas de los clientes como elementos aleatorios. Es por ello, que para comprender el modelado de este tipo de problemas es necesaria una Introducción a la Programación Estocástica, así como, algún lenguaje de programación que la soporte. Nosotros haremos mención del formato $SMPS$, *Stochastic Mathematical Programming System*, para modelar un problema de programación. Para así después resolver estos problemas mediante algún solver disponible en el conocido servidor de optimización $NEOS$.

Pero dado el elevado coste computacional que requieren este tipo de problemas, nuestra intención es utilizar algún método heurístico que nos ayude a obtener soluciones factibles para nuestros problemas en tiempos razonables. Para ello,vamos a considerar una adaptación de la heurística del caso determinista, el conocido algoritmo de ahorros de *Clarke* y $Wright$, a nuestro contexto estocástico. Mostraremos su funcionamiento en un problema real tomado del ámbito de la logística agrícola y programamos en $R$ ambos algoritmos (determinista y estocástico). Finalmente, diseñamos una interfaz gráfica en $R$ que facilita la introducción de los datos, la llamada a los algoritmos, la visualización de los resultados y la realización de análisis de post-optimalidad de manera rápida.

## English abstract

Nowadays, there is a significant improvement in the data collection of every kind. This has caused that the majority of problems treated in operative investigation pass to be more realistic and dynamic.

Between them we emphasize the problems of routes of vehicles which had incorporated new restrictions applied to data in real time and stochastic parameters. A specific case of this situation is to consider the clients´demands as random elements. For this reason, to understand the modelling of these kinds of problems it is necessary an Introduction to the Stochastic Programming, as well as, some programming language which can support it. We will mention the format $SMPS$, Stochastic Mathematical Programming System, in order to model a problem of programming. In this way, later we will be able to resolve these problems by some available solver in the well-known server of optimization $NEOS$.

But given the elevated computational cost that these kinds of problems require, our intention is to use some heuristic method which can help us to obtain feasible solutions to our problems in reasonable times. In order to do this, we will consider an adaptation of the heuristic from the deterministic case, the well-known savings algorithm of *Clarke* and $Wright$, to our stochastic context. We will show its working in a real problem taken from the field of the agricultural logistic and we programme in $R$ both algorithms (deterministic and stochastic). Finally, we design a graphic interface in $R$ which facilitates

the introduction of data, the call of the algorithms, the visualization of the results and the realization of an analysis of post-optimality in a faster way.

# Introducción

En 1959, [DR59], bajo el nombre de "El problema de despachos de camiones", propusieron una serie de problemas con el objetivo de diseñar una ruta óptima de una serie de camiones de reparto de gasolina entre un terminal y un gran número de estaciones.

A día de hoy, conocemos estos modelos como problemas de rutas de vehículos, $VRP$, y son una de las áreas más investigadas en programación entera, que es una clase de problemas de investigación operativa, dadas sus muchas aplicaciones en el mundo real, como el transporte y la logística de distribución.

El principal objetivo de este tipo de problemas es minimizar el coste total de una serie de rutas, diseñadas para que un conjunto de vehículos, desde un almacén, puedan satisfacer la demanda de una serie de clientes, regresando al almacén de partida.

A lo largo de la literatura se han propuesto múltiples versiones del modelo $VRP$ original, a fin de adaptarse lo más posible a la realidad, y los diversos problemas que nos podemos encontrar. Sin embargo, muchos de estos modelos, aún dadas una cantidad imponente de restricciones, no tienen en cuenta un aspecto importante, ya que no consideran el hecho de que los parámetros que constituyen el problema en la vida real son aleatorios.

A diferencia de los $VRP$ deterministas, los problemas de rutas de vehículos estocásticos cuentan con una literatura bastante dispersa y no organizada. Principalmente dado su alto coste computacional.

Es por ello, que en este trabajo nuestro interés es adentrarnos en los $SVRP$, problemas de rutas de vehículos estocásticos, con la finalidad de proponer un método de solución cuando las demandas de los clientes visitados son aleatorias.

De esta forma, la primera parte de este trabajo consistirá en una introducción a la programación estocástica, con una pequeña reseña a la optimización determinista. Además, incluiremos varios ejemplos, tanto con variables aleatorias discretas como continuas. Por último, revisaremos un algoritmo para resolver problemas de programación estocástica.

A continuación, hablaremos sobre una de las formas de implementar problemas lineales estocásticos con el fin de resolverlos mediante algún solver online almacenado en $NEOS$.

Para ya adentrarnos en el interés principal de este trabajo, los problemas de rutas de vehículos estocásticos, empezaremos con una revisión de los $VRP$ deterministas, presentando una heurística para resolverlos. Es entonces, cuando haremos una panorámica de los $VRP$ estocásticos, para centrarnos en los casos donde la demanda es aleatoria. Llegados a este punto, adaptaremos la heurística del caso determinista al modelo estocástico.

Una vez completado el capítulo referente a la metodología de los problemas de rutas de vehículos, continuaremos con una aplicación de los algoritmos estudiados a un ejemplo de la vida real. Completaremos el estudio del caso real con la realización de diversas simulaciones, mediante las cuales comparamos los resultados obtenidos en ambos casos, determinista y estocástico.

Finalizamos con una sección en la cual presentamos las conclusiones a este trabajo de fin de máster.

# Capítulo 1

# Introducción a la optimización estocástica

En este capítulo vamos a presentar los problemas de optimización estocástica, que son aquéllos en los que uno o varios parámetros del modelo son aleatorios. Entre las metas que pretende alcanzar nuestro estudio se haya el de encontrar una solución que sea factible en todos, o casi todos, los determinados escenarios posibles.

Para empezar, y para mayor claridad, haremos una breve introducción a los problemas de optimización deterministas, que son aquellos donde todos los parámetros del modelo son conocidos con anterioridad.

A continuación, presentaremos los problemas de optimización estocástica, así como algún ejemplo con variables discretas y continuas. Por último, presentaremos algunas propiedades básicas que serán útiles y necesarias en el algoritmo de resolución presentado.

## 1.1. Problemas de optimización determinista

En esta sección, nuestro interés es hacer un breve recordatorio de algunos elementos de la optimización determinista. Empezamos mostrando la formulación matemática de este tipo de problemas, y a continuación, mencionamos las condiciones necesarias y suficientes para la existencia de solución óptima. Para terminar, haremos referencia a los problemas duales en programación lineal.

Los problemas de optimización buscan minimizar (mín) o maximizar (máx) una función $f : S \subset \mathbb{R}^n \to \mathbb{R}$, sujeta a un conjunto de restricciones establecido. Dado que maximizar una función $f(x)$, es equivalente a minimizar $-f(x)$, podemos restringirnos a los problemas de minimización. El problema de minimización se expresa como:

$$\begin{aligned} \text{mín} \quad & f(x) \\ \text{Sujeto a} \quad & x \in S. \end{aligned}$$

En los problemas expresados de esta manera, la función $f(x)$ es conocida como función objetivo, y el conjunto $S$ como conjunto factible.

Un punto $\hat{x} \in S$ es una solución óptima global, si se verifica que para todo $y \in S$ se tiene que $f(\hat{x}) \leq f(y)$. Una solución óptima local, $\hat{x}$, es aquélla tal que existe $\epsilon > 0$ donde para todo $y \in B(\hat{x}, \epsilon) \cap S$, se tiene que $f(\hat{x}) \leq f(y)$. Aquí $B(\hat{x}, \epsilon)$ denota a todos los puntos que distan de $\hat{x}$ una distancia menor a $\epsilon$.

Pero en este tipo de problemas, normalmente, existe un conjunto de relaciones que se deben cumplir, llamadas restricciones. Estas son representadas por funciones $g_i : S \subset \mathbb{R}^n \to \mathbb{R}$, con

$i \in \{1, ..., m\}$, y $h_j : S \subset \mathbb{R}^n \to \mathbb{R}$, con $j \in \{1, ..., p\}$. Por tanto, nuestro problema de optimización se puede expresar de la forma:

$$\begin{aligned} \text{mín} \quad & f(x) \\ \text{Sujeto a} \quad & g_i(x) \leq 0 \quad i \in \{1, ..., m\} \\ & h_j(x) = 0 \quad j \in \{1, ..., p\}. \end{aligned}$$

Si en un problema de optimización la función objetivo es lineal, y las restricciones se expresan mediante un sistema de ecuaciones o inecuaciones también lineales, entonces estamos ante un denominado problema de programación lineal.

A continuación, el siguiente teorema nos aporta condiciones necesarias para la existencia de optimalidad.

**Teorema 1 ([KT51])** *Consideremos un problema de programación lineal en la forma anteriormente expuesta. Sean las funciones $f$, $g_i$ con $i \in \{1, ..., m\}$ y $h_j$ con $j \in \{1, ..., p\}$ continuamente diferenciables. Supongamos que $\hat{x}$ es una solución óptima local y los vectores gradiente $\nabla g_i(\hat{x})$, para todo $i$ tal que $g_i(\hat{x}) = 0$, y $\nabla h_j(\hat{x})$ son linealmente independientes. Entonces existen escalares $\lambda_i$, $i \in \{1, ..., m\}$, y $\mu_j$, $j \in \{1, ..., p\}$, tales que:*

$$\begin{aligned} & \nabla f(\hat{x}) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\hat{x}) + \sum_{j=1}^{p} \mu_j \nabla h_j(\hat{x}) = 0 \\ & \lambda_i g_i(\hat{x}) = 0 \quad i \in \{1, ..., m\} \\ & \lambda_i \geq 0 \quad i \in \{1, ..., m\} \\ & g_i(\hat{x}) \leq 0 \quad i \in \{1, ..., m\} \\ & h_j(\hat{x}) = 0 \quad j \in \{1, ..., p\}. \end{aligned}$$

Además, las siguientes condiciones son suficientes para la existencia de un óptimo global.

**Teorema 2 ([KT51])** *Consideremos un problema de programación lineal en la forma anteriormente expuesta. Sean las funciones $f$ y $g_i$ con $i \in \{1, ..., m\}$ continuamente diferenciables y convexas. Consideremos $\hat{x} \in \mathbb{R}^n$. Si existen escalares $\lambda_i$, $i \in \{1, ..., m\}$, y $\mu_j$, $j \in \{1, ..., p\}$, tales que:*

$$\begin{aligned} & \nabla f(\hat{x}) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\hat{x}) + \sum_{j=1}^{p} \mu_j \nabla h_j(\hat{x}) = 0 \\ & \lambda_i g_i(\hat{x}) = 0 \quad i \in \{1, ..., m\} \\ & \lambda_i \geq 0 \quad i \in \{1, ..., m\} \\ & g_i(\hat{x}) \leq 0 \quad i \in \{1, ..., m\} \\ & h_j(\hat{x}) = 0 \quad j \in \{1, ..., p\}, \end{aligned}$$

*entonces $\hat{x}$ es un óptimo global.*

Veamos ahora un ejemplo de optimización determinista para ilustrar lo visto en esta sección que además sirve de motivación a los problemas estocásticos.

### 1.1.1. El problema del granjero

El siguiente problema es introducido en [BL11]. Un granjero cultiva maíz, trigo y remolacha en sus 500 hectáreas de tierra. Durante el invierno, debe decidir cuanta tierra asignar a cada cultivo. Este sabe que necesita, al menos, 200 toneladas de trigo y 240 de maíz para alimentar su ganado. Estas cantidades pueden ser cultivadas en la granja o compradas a un mayorista. Los excesos de producción se venden al precio de 170 y 150 dólares por tonelada de trigo y maíz, respectivamente. Los precios

de compra son 238 y 210 dólares por tonelada, respectivamente. La remolacha se vende a 36 dólares por tonelada, aunque cualquier producción mayor a las 6000 toneladas será vendida por 10 dólares. Por su experiencia, sabe que la producción de cada cultivo por hectárea es de 2.5, 3 y 20 toneladas para el trigo, maíz y remolacha, respectivamente. Sabiendo que el coste por hectárea es de 150, 230 y 260 dólares por trigo, maíz y remolacha, respectivamente, nos interesa planificar la asignación a cada cultivo para obtener el mayor beneficio posible.

Consideremos las siguientes variables del problema de optimización:

$$
\begin{aligned}
x_1 &= \text{hectáreas de tierra dedicadas al trigo},\\
x_2 &= \text{hectáreas de tierra dedicadas al maíz},\\
x_3 &= \text{hectáreas de tierra dedicadas a la remolacha},\\
w_1 &= \text{toneladas de trigo vendidas},\\
w_2 &= \text{toneladas de maíz vendidas},\\
w_3 &= \text{toneladas de remolacha vendidas a 36 dólares},\\
w_4 &= \text{toneladas de remolacha vendidas a 10 dólares},\\
y_1 &= \text{toneladas de trigo compradas},\\
y_2 &= \text{toneladas de maíz compradas}.
\end{aligned}
$$

El problema queda formulado de la siguiente manera:

$$
\begin{aligned}
\text{mín} \quad & 150x_1 + 230x_2 + 260x_3 + 238y_1 - 170w_1 + 210y_2 - 150w_2 - 36w_3 - 10w_4\\
\text{Sujeto a} \quad & x_1 + x_2 + x_3 \leq 500\\
& 2.5x_1 + y_1 - w_1 \geq 200\\
& 3x_2 + y_2 - w_2 \geq 240\\
& w_3 + w_4 \leq 20x_3\\
& w_3 \leq 6000\\
& x_1, x_2, x_3, y_1, y_2, w_1, w_2, w_3, w_4 \geq 0.
\end{aligned}
$$

Si resolvemos este problema, véase Apéndice A, obtenemos los resultados que aparecen en el Cuadro 1.1:

| | Trigo | Maíz | Remolacha |
|---|---|---|---|
| Hectáreas Cultivadas | 120 | 80 | 300 |
| Toneladas Obtenidas | 300 | 240 | 6000 |
| Toneladas Vendidas | 100 | – | A 36\$: 6000<br>A 10\$: – |
| Toneladas Compradas | – | – | |
| Ganancias Netas: 118600 | | | |

Cuadro 1.1: Solución al problema del granjero.

Por tanto, de la solución deducimos que el granjero debe utilizar suficientes hectáreas para la remolacha de manera que no se sobrepase el límite de 6000 toneladas. El resto de hectáreas es repartida entre trigo y maíz de manera que se cubran los mínimos posibles, y las hectáreas sobrantes se utilizan en aquel producto que proporcione mayores beneficios.

### 1.1.2. Dualidad en programación lineal

En esta sección presentaremos el denominado problema dual, ya que la solución de este será utilizada en el algoritmo que presentaremos al final de este capítulo.

Aunque, entre el problema original y el dual, existen una serie de relaciones interesantes y de gran utilidad para la obtención de solución ante modelos aparentemente no factibles, en esta sección solo haremos mención de la transformación de nuestro problema en su versión dual, para así obtener la solución de este.

Consideremos el siguiente problema lineal, también llamado primal,

$$\begin{aligned} \text{mín} \quad & c^T x \\ \text{Sujeto a} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

donde $A$ es una matriz de dimensión $m \times n$, $c$ y $x$ vectores de $\mathbb{R}^n$, y $b \in \mathbb{R}^m$. Entonces el problema dual asociado al primal es de la forma

$$\begin{aligned} \text{máx} \quad & \pi^T b \\ \text{Sujeto a} \quad & \pi^T A \leq c^T \end{aligned}$$

donde las variables $\pi$ son llamadas variables duales, o en determinados contextos, precios sombra o multiplicadores.

El problema dual tiene solución óptima si y solo si el problema dual tiene solución óptima. Además, en ambos problemas la función objetivo, en el óptimo, tomará el mismo valor.

A continuación podemos observar en el Cuadro 1.2 las relaciones primal dual, que nos indica el tipo de desigualdad, o igualdad, que debemos tomar en cada restricción y variable al pasar del problema primal al dual.

## 1.2. Programación estocástica

En esta sección veremos la formulación del modelo general de un problema de programación estocástica. En este tipo de problemas alguno o todos los parámetros son variables aleatorias, pero se conoce una distribución de probabilidad asociada a los mismos.

En los programas estocásticos bietapa, presentados originalmente por [Dan55] y [Bea55], existen un conjunto de decisiones que deben tomarse sin la información completa de algunos eventos aleatorios, es decir, antes del experimento aleatorio. Estas decisiones son llamadas *decisiones de primera etapa* y las representamos por $x$, donde $x \in \mathbb{R}^n$, y el período donde estas decisiones son tomadas lo llamaremos *primera etapa*. Por ejemplo, en el problema del granjero, las decisiones de primera etapa pueden ser cuántas hectáreas dedicar a cada cultivo.

Más tarde, toda la información se recibe a través de algún vector aleatorio $\boldsymbol{\xi}$, se utilizará la letra negrita para denotas los vectores aleatorios y así distinguirlos de sus realizaciones. Tras esto, son tomadas las *decisiones de segunda etapa* $\mathbf{y}$, correspondientes al período de *segunda etapa*. Estas decisiones difieren como resultado del experimento aleatorio y de las decisiones de primera etapa. (Nótese que para cada realización de $\boldsymbol{\xi}$ el problema estocástico se convierte en determinista, por tanto, puede no existir un vector $x$ que sea óptimo, ni siquiera factible, para todas las realizaciones de los

| Problema de minimización | Problema de maximización |
|---|---|
| Restricciones | Variables |
| $\geq$ | $\geq 0$ |
| $=$ | Sin restricción |
| $\leq$ | $\leq 0$ |
| Variables | Restricciones |
| $\geq 0$ | $\leq$ |
| Sin restricción | $=$ |
| $\leq 0$ | $\geq$ |

Cuadro 1.2: Relaciones primal dual.

parámetros aleatorios). En el problema del granjero, el vector aleatorio es el conjunto de rendimientos y las decisiones de segunda etapa ("acciones correctivas") son las compras y ventas de los productos. A veces usaremos notación funcional $\xi(w)$ ó $y(s)$ para mostrar explícitamente la dependencia de algún elemento subyacente.

La formulación matemática del denominado problema de programación estocástica en dos etapas con recursos es como sigue:

$$\begin{aligned} \text{mín} \quad & c^T x + E_{\boldsymbol{\xi}} Q(x, \boldsymbol{\xi}) \\ \text{Sujeto a} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

donde $c \in \mathbb{R}^n$ es el vector de costes, $A$ la matriz de restricción de dimensión $m_1 \times n$ , $b \in \mathbb{R}^{m_1}$ el vector de recursos y $E_{\boldsymbol{\xi}}$ denota la esperanza matemática respecto a $\boldsymbol{\xi}$. Además:

$$\begin{aligned} Q(x, \boldsymbol{\xi}) = \text{mín} \quad & \boldsymbol{q}^T \boldsymbol{y} \\ \text{Sujeto a} \quad & W\boldsymbol{y} = \boldsymbol{h} - \boldsymbol{T}x \\ & y \geq 0 \end{aligned}$$

donde $\boldsymbol{\xi}$ está formado por las componentes $\boldsymbol{q}$, $\boldsymbol{h}$ y $\boldsymbol{T}$ de dimensiones $n_1$, $m_2$ y $m_2 \times n$, respectivamente, donde cada componente de $\boldsymbol{q}, \boldsymbol{h}$ y $\boldsymbol{T}$ es una realización de la variable aleatoria $\boldsymbol{\xi}$. Asumimos que $W$ es una matriz, de dimensión $m_2 \times n_1$, fija, de manera que $\boldsymbol{\xi}^T = \left(\boldsymbol{q}^T, \boldsymbol{h}^T, \boldsymbol{T}_{1\cdot}, ..., \boldsymbol{T}_{m_1\cdot}\right)$ tiene dimensión $N = n_1 + m_2 + (m_2 \times n)$, donde $\boldsymbol{T}_{i\cdot}$ es la fila $i-$ésima de $\boldsymbol{T}$ para $i \in \{1, ..., m_2\}$.

En el problema del granjero supondremos que el vector aleatorio es discreto con 3 valores distintos. Sólo $T$ es aleatorio. El problema de segunda etapa para un escenario particular $s$ se puede escribir así:

$$\begin{aligned} Q(x, \boldsymbol{\xi}) = \text{mín}\{ & 238y_1 - 170w_1 + 210y_2 - 150w_2 - 36w_3 - 10w_4 \\ & \text{Sujeto a } t_1(s)x_1 + y_1 - w_1 \geq 200; \quad t_2(s)x_2 + y_2 - w_2 \geq 240 \\ & w_3 + w_4 \leq t_3(s)x_3; \quad w_3 \leq 6000 \\ & y, w \geq 0\} \end{aligned}$$

donde $t_i(s)$ representa el rendimiento del cultivo $i$ bajo el escenario $s$ (en este caso, estado de la naturaleza $s$).

Si consideramos $\mathcal{Q}(x) = E_{\boldsymbol{\xi}} Q(x, \boldsymbol{\xi})$ como el valor de la función o función recurso, podemos considerar el problema estocástico de la siguiente forma, aún más condensada:

$$\begin{aligned} \text{mín} \quad & c^T x + \mathcal{Q}(x) \\ \text{Sujeto a} \quad & Ax = b \\ & x \geq 0 \end{aligned} \tag{1.1}$$

Cualquiera de los modelos que acabamos de introducir se denominan representaciones implícitas del problema estocástico.

Si las decisiones de primera etapa y/o segunda etapa son enteras, las restricciones $x \geq 0$ y/o $y \geq 0$ se sustituyen por:

$$x \in X, y \in Y$$

donde $X = \mathbb{Z}_+^n$ e $Y = \mathbb{Z}_+^{n_1}$.

Estos problemas son llamados programas de recurso, porque en ellos se pueden tomar decisiones o acciones de recurso después de que se divulgue la incertidumbre. En estos, tomamos $x$ con la presencia de incertidumbre sobre las realizaciones futuras, aun así, se tiene en cuenta sus efectos futuros a través de $\mathcal{Q}(x)$, que calcula el valor esperado de tomar la decisión $x$. En la segunda etapa, el valor real de $\xi$ es conocido, y se pueden tomar algunas acciones correctivas o decisiones de recurso.

Vistos los problemas de programación estocástica bietapa, podemos formular aquellos en los cuales existen un número finito de posibles decisiones futuras, de la manera siguiente:

$$\begin{aligned} \text{mín} \quad & c_1^T x^1 + E_{\boldsymbol{\xi^1}} \left[ \text{mín}\, \boldsymbol{c}_2^T \boldsymbol{x}^2 + ... + E_{\boldsymbol{\xi^{H-1}}} (\text{mín}\, \boldsymbol{c}_H^T \boldsymbol{x}^H) \right] \\ \text{Sujeto a} \quad & W^1 x^1 = h^1 \\ & \boldsymbol{T}^1 x^1 + W^2 \boldsymbol{x}^2 = \boldsymbol{h}^2 \\ & . \\ & . \\ & . \\ & \boldsymbol{T}^{H-1} x^{H-1} + W^H \boldsymbol{x}^H = \boldsymbol{h}^H \\ & x^1, \boldsymbol{x}^t \geq 0, t = 2, ..., H \end{aligned}$$

donde $c_1 \in \mathbb{R}^{n_1}$, $h^1 \in \mathbb{R}^{m_1}$ y $\boldsymbol{\xi^t}^T = \left( \boldsymbol{c_{t+1}}^T, \boldsymbol{h^{t+1}}^T, \boldsymbol{T^t}_{1\cdot}, ..., \boldsymbol{T^t}_{m_t \cdot} \right)$ para $t = 1, ..., H-1$. Las variables de la $t-$ésima etapa están indexados por el resultado de los eventos aleatorios ocurridos en etapas anteriores para todo $t = 2, ..., H$. Asumimos que $W^t$ es una matriz, de dimensión $m_t \times n_t$, fija para todo $t = 1, ..., H$.

Como se puede ver, en el modelo matemático del problema de programación estocástico multietapa, modelar problemas de gran escala puede llevar una dificultad computacional demasiado elevada. Es por ello, que muchas veces se reemplazan las variables aleatorias por sus valores esperados para resolverlos. Pero, este tipo de soluciones, puede alejarse del óptimo del problema estocástico. Con esto surgen dos conceptos asociados a las soluciones.

El *valor esperado de la información perfecta*, $(EVPI)$, representa la pérdida de beneficios debido a la presencia de incertidumbre. Este valor es la diferencia entre la *información perfecta*, $(WS)$, que es el valor óptimo que se obtiene cuando conoces lo que va a pasar, y la solución óptima, $(RP)$, obtenida al considerar la incertidumbre y resolver el problema. Por tanto,

$$EVPI = WS - RP.$$

Nótese que si el problema es de minimizar se invierten los elementos de la operación.

EL *valor de la solución estocástica*, $(VSS)$, representa las pérdidas por usar la *solución del valor esperado*, $(EEV)$, al utilizar la media de las variables aleatorias y no considerar la incertidumbre. Por tanto,

$$VSS = EEV - RP.$$

Nótese que si el problema es de minimizar se invierten los elementos de la operación.

Veamos ahora un ejemplo para poner en práctica los conceptos vistos en esta sección, para ello continuaremos con el ejemplo del granjero pero ahora vamos a considerar incertidumbre en los rendimientos de los diferentes cultivos. A tal fin, vamos a considerar dos casos, uno con variables discretas y otro con continuas.

### 1.2.1. Problema del granjero con incertidumbre

Consideremos entonces nuevamente el problema del granjero visto anteriormente. En este, como hemos dicho en la sección anterior, podemos considerar que las decisiones de primera etapa son las hectáreas dedicadas a cada cultivo, mientras que, las decisiones de segunda etapa son las compras y ventas de cada cultivo.

**1) Caso discreto.**

Supongamos que el rendimiento para cada cultivo puede darse en tres escenarios distintos, con igual probabilidad. En el primer escenario el rendimiento es un 20 % menos de lo que supone el granjero, en el segundo escenario los rendimientos son los supuestos por el granjero y por último, en el tercer escenario, son un 20 % más. Por tanto tenemos los rendimientos para cada tipo de escenario recogidos en el Cuadro 1.3:

| | Trigo | Maíz | Remolacha |
|---|---|---|---|
| Rendimientos primer escenario | 2 | 2.4 | 16 |
| Rendimientos segundo escenario | 2.5 | 3 | 20 |
| Rendimientos tercer escenario | 3 | 3.6 | 24 |

Cuadro 1.3: Rendimientos para cada cultivo en los diferentes escenarios.

Podemos calcular, de manera independiente, las soluciones para cada escenario. El código utilizado puede ser visto en el Apéndice A, y obtenemos las soluciones de los Cuadros 1.4 y 1.5.

| Primer escenario | Trigo | Maíz | Remolacha |
|---|---|---|---|
| Hectáreas | 100 | 25 | 375 |
| Toneladas Obtenidas | 200 | 60 | 6000 |
| Toneladas Vendidas | – | – | A 36$: 6000<br>A 10$: – |
| Toneladas Compradas | – | 180 | |
| Ganancias Netas: 59950 | | | |

Cuadro 1.4: Solución al problema del granjero para el primer escenario.

| Tercer escenario | Trigo | Maíz | Remolacha |
|---|---|---|---|
| Hectáreas | 183.33 | 66.67 | 250 |
| Toneladas Obtenidas | 550 | 240 | 6000 |
| Toneladas Vendidas | 350 | – | A 36$: 6000<br>A 10$: – |
| Toneladas Compradas | – | – | |
| Ganancias Netas: 167667 | | | |

Cuadro 1.5: Solución al problema del granjero para el tercer escenario.

Estas soluciones pueden ser calculadas si el granjero sabe con anterioridad lo que va a pasar. Sin embargo, debido a la incertidumbre de cada escenario, este, debe maximizar los beneficios sin saber que rendimientos obtendrá de su cosecha.

Para ello vamos a crear un nuevo índice, $s = 1, 2, 3$, correspondiente a cada escenario para las decisiones de segunda etapa. De manera que tenemos $w_{is}$ y $y_{js}$ con $i = 1, 2, 3, 4$, $j = 1, 2$ y $s = 1, 2, 3$, con significado análogo al utilizado en el apartado 1.1.1.

Entonces el problema a minimizar es:

$$\begin{aligned}
\text{mín}\quad & 150x_1 + 230x_2 + 260x_3 + \frac{1}{3}(+238y_{11} - 170w_{11} + 210y_{21} - 150w_{21} - 36w_{31} - 10w_{41}) \\
& + \frac{1}{3}(+238y_{12} - 170w_{12} + 210y_{22} - 150w_{22} - 36w_{32} - 10w_{42}) \\
& + \frac{1}{3}(+238y_{13} - 170w_{13} + 210y_{23} - 150w_{23} - 36w_{33} - 10w_{43}) \\
\text{Sujeto a}\quad & x_1 + x_2 + x_3 \leq 500 \\
& 2x_1 + y_{11} - w_{11} \geq 200, \quad 2.5x_1 + y_{12} - w_{12} \geq 200 \\
& 3x_1 + y_{13} - w_{13} \geq 200, \quad 2.4x_2 + y_{21} - w_{21} \geq 240 \\
& 3x_2 + y_{22} - w_{22} \geq 240, \quad 3.6x_2 + y_{23} - w_{23} \geq 240 \\
& w_{31} + w_{41} \leq 16x_3, \quad w_{31} \leq 6000, \quad w_{32} + w_{42} \leq 20x_3 \\
& w_{32} \leq 6000, \quad w_{33} + w_{43} \leq 24x_3, \quad w_{33} \leq 6000 \\
& x_1, x_2, x_3, y_{js}, w_{is} \geq 0, \quad j = 1, 2; \quad i = 1, 2, 3, 4; \quad s = 1, 2, 3;
\end{aligned}$$

Nótese, que esta forma de escribir el problema de programación estocástica, se llama forma extensiva ya que describe las decisiones de segunda etapa para todos los escenarios posibles.

La solución óptima, $(RP)$, se puede ver en el Cuadro 1.6, con los diferentes rendimientos obtenidos en cada escenario. El código utilizado se muestra en el Apéndice A.

| | | Trigo | Maíz | Remolacha |
|---|---|---|---|---|
| Decisiones de primera etapa | Hectáreas | 170 | 80 | 250 |
| Primer escenario | Toneladas Obtenidas | 340 | 192 | 4000 |
| Segundo escenario | Toneladas Obtenidas | 425 | 240 | 5000 |
| Tercer escenario | Toneladas Obtenidas | 510 | 288 | 6000 |
| Ganancias Netas: 108390 | | | | |

Cuadro 1.6: Solución al problema del granjero con incertidumbre, 3 escenarios.

Visto esto podemos calcular el valor esperado de la información perfecta, $(EVPI)$. Para ello necesitamos conocer la información perfecta, $(WS)$. Si suponemos que el clima es cíclico, el granjero puede tomar primero la solución del Cuadro 1.4, después la del Cuadro 1.1 y por último el Cuadro 1.5. Y así volver a empezar. Obteniendo unos beneficios, a largo plazo, de 115406. Por tanto $EVPI = 7016$, que equivalen a las pérdidas por no conocer lo que va a pasar.

Además, podemos calcular la solución del valor esperado. Para ello, utilizaremos las hectáreas obtenidas en la solución de la producción media, segundo escenario, con los diferentes rendimientos según el tipo de escenario. Estos resultados, así como, la ganancia media pueden ser vistos en el Cuadro 1.7.

| | | Trigo | Maíz | Remolacha |
|---|---|---|---|---|
| Decisiones de primera etapa | Hectáreas | 120 | 80 | 300 |
| Primer escenario | Toneladas Obtenidas | 240 | 192 | 4800 |
| Segundo escenario | Toneladas Obtenidas | 300 | 240 | 6000 |
| Tercer escenario | Toneladas Obtenidas | 360 | 288 | 7200 |
| Ganancias Netas: 107240 | | | | |

Cuadro 1.7: Solución del valor esperado al problema del granjero.

Por tanto el valor de la solución estocástica, $(VSS)$, es 1150, que equivalen a las perdidas por no calcular la solución al problema estocástico.

**2) Caso continuo.**

Ahora supongamos que, si los rendimientos para los diferentes cultivos son independientes, estos siguen una distribución uniforme para cada uno de ellos. El rango de cada cultivo viene dado por $[l_i, u_i]$ donde $i = 1, 2, 3$ con 1 trigo, 2 maíz y 3 remolacha, y que $l_i$ es el 80 % del rendimiento y $u_i$ el 120 %. Por ejemplo, para la remolacha el rendimiento seguirá una distribución uniforme de rango $[16, 24]$. Así, tenemos que la media de cada cultivo es la producción original, propuesta por el agricultor.

Como suponemos que los rendimientos son independientes, podemos considerar:

$$E_{\boldsymbol{\xi}} Q(x, \boldsymbol{\xi}) = \sum_{i=1}^{3} E_{\boldsymbol{\xi}} Q_i(x_i, \boldsymbol{\xi}) = \sum_{I=1}^{3} \mathcal{Q}_i(x_i)$$

donde $Q_i(x_i, \boldsymbol{\xi})$ es el valor óptimo de la segunda etapa de las compras y ventas del cultivo $i$. Podemos calcular la expresión analítica del valor $\mathcal{Q}_i(x_i)$ para cada $i$. Veamos, paso a paso, como obtener $\mathcal{Q}_3(x_3)$ para la remolacha. Tenemos:

$$\begin{aligned} Q(x_3, \boldsymbol{\xi}) = \text{mín} \quad & -36w_3(\boldsymbol{\xi}) - 10w_4(\boldsymbol{\xi}) \\ \text{Sujeto a} \quad & w_3(\boldsymbol{\xi}) + w_4(\boldsymbol{\xi}) \leq t_3(\boldsymbol{\xi})x_3 \\ & w_3(\boldsymbol{\xi}) \leq 6000 \\ & w_3(\boldsymbol{\xi}), w_4(\boldsymbol{\xi}) \geq 0 \end{aligned}$$

donde $t_3(\boldsymbol{\xi})$ es el rendimiento aleatorio de la producción de remolacha por hectárea. Sabemos que $w_3$ viene dado por la máxima venta a buen precio, mientras que, $w_4$ son las ventas al superar las 6000 unidades vendidas. Entonces:

$$\begin{aligned} w_3(\boldsymbol{\xi}) &= \text{mín}\{6000, t_3(\boldsymbol{\xi})x_3\} \\ w_4(\boldsymbol{\xi}) &= \text{máx}\{t_3(\boldsymbol{\xi})x_3 - 6000, 0\} \end{aligned}$$

Por tanto:

$$Q_3(x_3, \boldsymbol{\xi}) = -36\,\text{mín}\{6000, t_3(\boldsymbol{\xi})x_3\} - 10\,\text{máx}\{t_3(\boldsymbol{\xi})x_3 - 6000, 0\}$$

Visto esto, nos podemos encontrar en tres situaciones. Primero supongamos que:

$$l_3 x_3 \leq 6000 \leq u_3 x_3 \Leftrightarrow l_3 \leq \frac{6000}{x_3} \leq u_3$$

Entonces se tiene:

$$\begin{aligned}\mathcal{Q}_3(x_3) &= E_{\boldsymbol{\xi}} Q_3(x_3, \boldsymbol{\xi}_3) \\ &= -\int_{l_3}^{\frac{6000}{x_3}} 36 t x_3 f(t) dt \\ &= -\int_{\frac{6000}{x_3}}^{u_3} (216000 + 10 t x_3 - 60000) f(t) dt\end{aligned}$$

donde $f(t)$ es la función de densidad de $t_3(\boldsymbol{\xi})$. Como este sigue una distribución uniforme en el intervalo $[16, 24]$ se tiene que:

$$\mathcal{Q}_3(x_3) = -720 x_3 + \frac{13(24x_3 - 6000)^2}{8x_3}.$$

Ahora veamos que ocurre cuando $l_3 x_3 > 6000$. Las decisiones de segunda etapa son:

$$\begin{aligned}w_3(\boldsymbol{\xi}) &= 6000 \\ w_4(\boldsymbol{\xi}) &= t_3(\boldsymbol{\xi})x_3 - 6000\end{aligned}$$

para cualquier realización de $\boldsymbol{\xi}$. Por tanto:

$$Q_{\boldsymbol{\xi}}(x_3, \xi) = -216000 - 10(t_3(\boldsymbol{\xi})x_3 - 6000) = -156000 - 10 t_3(\boldsymbol{\xi}) x_3$$

y tenemos que el valor esperado es:

$$\mathcal{Q}_3(x_3) = -156000 - 200 x_3.$$

En la última situación, suponemos que $u_3 x_3 < 6000$. Por tanto las decisiones de segunda etapa son:

$$\begin{aligned}w_3(\boldsymbol{\xi}) &= t_3(\boldsymbol{\xi})x_3 \\ w_4(\boldsymbol{\xi}) &= 0\end{aligned}$$

para cualquier realización de $\boldsymbol{\xi}$. Por tanto:

$$Q_{\boldsymbol{\xi}}(x_3, \xi) = -36 t_3(\boldsymbol{\xi}) x_3$$

y tenemos que el valor esperado es:

$$\mathcal{Q}_3(x_3) = -720 x_3.$$

Por tanto tenemos que el valor esperado para el cultivo de la remolacha es:

$$\mathcal{Q}_3(x_3) = \begin{cases} -720 x_3 & \text{si } x_3 < 250 \\ -720 x_3 + \frac{13(24x_3 - 6000)^2}{8x_3} & \text{si } 250 \leq x_3 \leq 375 \\ -156000 - 200 x_3 & \text{si } x_2 > 375. \end{cases}$$

De igual manera, obtenemos para los otros dos cultivos sus valores esperados. Para el trigo:

$$\mathcal{Q}_1(x_1) = \begin{cases} 47600 - 595x_1 & \text{si } x_1 < 200/3 \\ 119\frac{(200-2x_1)^2}{x_1} - 85\frac{(200-3x_1)^2}{x_1} & \text{si } 200/3 \leq x_3 \leq 100 \\ 34000 - 425x_1 & \text{si } x_1 > 100. \end{cases}$$

Y para el maíz:

$$\mathcal{Q}_2(x_2) = \begin{cases} 50400 - 630x_2 & \text{si } x_2 < 200/3 \\ 87.5\frac{(240-2.4x_2)^2}{x_2} - 62.5\frac{(240-3.6x_2)^2}{x_2} & \text{si } 200/3 \leq x_2 \leq 100 \\ 36000 - 450x_2 & \text{si } x_2 > 100. \end{cases}$$

Entonces el problema de optimización es de la forma:

$$\begin{aligned} \text{mín} \quad & 150x_1 + 230x_2 + 260x_3 + \mathcal{Q}_1(x_1) + \mathcal{Q}_2(x_2) + \mathcal{Q}_3(x_3) \\ \text{Sujeto a} \quad & x_1 + x_2 + x_3 \leq 500 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Ante el problema que nos encontramos podemos destacar que es convexo y diferenciable, ya que las decisiones de primera etapa son lineales y las tres funciones, respectivas a la segunda etapa, son convexas, continuas y diferenciables. Por el Teorema 2, disponemos de condiciones que son necesarias y suficientes para la existencia de un óptimo global. Aplicando las condiciones de $Kuhn - Tucker$ tenemos:

$$\begin{aligned} c_i + \frac{\partial \mathcal{Q}_i(x_i)}{\partial x_i} + \lambda &= 0 \\ \lambda\left(x_1 + x_2 + x_3 - 500\right) &= 0 \\ x_1 + x_2 + x_2 &\leq 500 \\ \lambda, x_1, x_2, x_3 &\geq 0 \end{aligned}$$

donde $c_i$ es el coeficiente que representa el coste de producción de cada cultivo por hectárea y $\lambda$ el multiplicador asociada a la restricción. Si asumimos que la solución es tal que $x_1 \geq 100$, $\frac{200}{3} \leq x_2 \leq 100$ y $250 \leq x_3 \leq 375$ con $\lambda \neq 0$, entonces se tiene:

$$\begin{aligned} -275 + \lambda &= 0 \\ -76 - \frac{1.44 \cdot 10^6}{x_2^2} + \lambda &= 0 \\ 476 - \frac{5.85 \cdot 10^7}{x_3^2} + \lambda &= 0 \\ x_1 + x_2 + x_3 &= 0 \end{aligned}$$

Y obtenemos la solución del Cuadro 1.8.

| | Trigo | Maíz | Remolacha |
|---|---|---|---|
| Hectáreas | 135.83 | 85.07 | 279.10 |
| Ganancias Netas: 111237.4 | | | |

Cuadro 1.8: Solución al problema del granjero con incertidumbre.

### 1.2.2. Propiedades básicas

Las siguientes propiedades que vamos a presentar son especificas del caso discreto, es decir, cuando $\boldsymbol{\xi}$ es una variable aleatoria discreta. Además, vamos a considerar las decisiones de primera y segunda etapa variables continuas.

Estas propiedades enuncian condiciones importantes y necesarias para la solución algorítmica de nuestro problema que veremos más adelante.

Definimos el conjunto determinado por las restricciones fijas

$$K_1 = \{x/Ax = b, x \geq 0\}.$$

Véase que este conjunto no depende del valor aleatorio.

Para cualquier $\xi$ dado, definimos el conjunto de factibilidad elemental como

$$K_2(\xi) = \{x/ \text{ existe } y \geq 0 \text{ sujeto a } W(w)y = h(w) - T(w)x\}$$

donde $w \in \Omega$ es el conjunto de todos los resultados posibles del experimento aleatorio que se realizan en la segunda etapa. Por tanto, para cada realización $w$ tenemos $W(w), h(w)$ y $T(w)$ conocidos.

Dado que $\boldsymbol{\xi}$ es discreta, podemos definir el conjunto de factibilidad de segunda etapa como

$$K_2 = \bigcap_{\xi \in \Xi} K_2(\xi)$$

tal que $\Xi \subset \mathbb{N}$ es el soporte de $\boldsymbol{\xi}$, es decir, el conjunto más pequeño que verifica $\mathbb{P}\{\boldsymbol{\xi} \in \Xi\} = 1$.

A mayores, definimos la envoltura positiva de $W$ como

$$posW = \{t/Wy = t, y \geq 0\}$$

que representa el conjunto de lados derechos que pueden obtenerse mediante una combinación no negativa de $W$.

Después de introducir estos conjuntos, podemos enunciar los siguientes teoremas.

**Teorema 3 ([BL11])**

- *Para un valor de $\xi$ dado, el conjunto $K_2(\xi)$ es un poliedro convexo.*
- *Cuando $\boldsymbol{\xi}$ es una variable aleatoria discreta finita, $K_2$ es un poliedro convexo.*

**Teorema 4 ([BL11])** *Para un $\xi$ dado, el valor de $Q(x, \xi)$ es*

- *una función lineal convexa por partes en $(h, T)$,*
- *una función lineal concava por partes en $q$,*
- *una función lineal convexa por partes en $x$ para todo $x \in K_2$.*

*Cuando $\boldsymbol{\xi}$ es una variable aleatoria discreta finita, $\mathcal{Q}(x)$ es una función lineal convexa por partes en $K_2$.*

Tal y como se esperaría en la mayoría de los casos, supongamos que existe una solución óptima de nuestro problema de programación estocástica. De las condiciones vistas en los Teoremas 1 y 2 obtenemos el siguiente resultado.

**Teorema 5 ([BL11])** *Supongamos que (1.1) tiene un valor óptimo finito. Una solución $\hat{x} \in K_1$ es óptimo de (1.1) si y solo si existen $\hat{\lambda} \in \mathbb{R}^{m_1}$, $\hat{\mu} \in \mathbb{R}^n_+$ con $\hat{\mu}^T\hat{x} = 0$, tal que,*

$$+A^T\hat{\lambda} + \hat{\mu} \in \partial\mathcal{Q}(\hat{x}).$$

En algunos ejemplos, puede suceder que una o más restricciones no se mantengan casi seguro como hemos mencionado hasta ahora. En su lugar, pueden estar sujetas a alguna probabilidad o nivel de confianza. Estas restricciones son de la forma

$$\mathbb{P}\{g(x, \boldsymbol{y}, \boldsymbol{\xi}) \leq 0\} \geq \alpha$$

donde $0 < \alpha < 1$. Además, la restricción que $g(\cdot) \leq 0$ puede contener varias restricciones bajo una representación vectorial, tales como $g(x, \boldsymbol{y}, \boldsymbol{\xi}) = \boldsymbol{h} - Ax$, o $g(x, \boldsymbol{y}, \boldsymbol{\xi}) = \boldsymbol{h} - \boldsymbol{T}x - \boldsymbol{W}\boldsymbol{y}$.

Suponemos que $\boldsymbol{\xi}$ es una variable aleatoria discreta con un número finito de escenarios. Representamos las realizaciones por $\xi_1, ..., \xi_K$, donde cada escenario $k$ tiene probabilidad $p_k$, y $\sum_{k=1}^{K} p_k = 1$.

Considerando la función indicadora

$$\eta(a) = \begin{cases} 0 & \text{si } a \leq 0 \\ \\ 1 & \text{si } \exists \;\; a_i > 0, \quad i = 1, ..., n, \end{cases}$$

la restricción sujeta a probabilidad es equivalente a

$$\sum_{k=1}^{K} p_k \eta(g(x, y_k, \xi_k)) \leq 1 - \alpha.$$

Entonces, si existe para cada escenario $k$ un vector $u_k$ tal que $g(x, y_k, \xi_k) \leq u_k$ para todo $x, y_k$ factibles, se tiene que la restricción de probabilidad puede ser transformada en

$$\begin{aligned} \sum_{k=1}^{K} p_k w_k \leq & 1 - \alpha \\ g(x, y_k, \xi_k) \leq & u_k w_k && k = 1, ..., K \\ w_k \in & \{0, 1\} && k = 1, ..., K \end{aligned}$$

donde la variable binaria $w_k$ hace el rol de la función indicadora. Por tanto, hemos visto como transformar una restricción de probabilidad en un problema de programación entero mixto.

## 1.3. Métodos de solución

En la sección 1.2.1 hemos visto el problema del granjero con incertidumbre en el caso discreto. Este era un problema estocástico con dos etapas y un número finito de realizaciones para el vector aleatorio. A través de la forma extensiva, hemos convertido nuestro problema estocástico en un modelo determinista. El problema surge cuando el vector aleatorio está formado por muchas realizaciones, ya que se convierte en un problema demasiado extenso para resolver de manera directa.

Para evitar estos tiempos de computación, el método más utilizado se basa en aproximar el término no lineal en el objetivo, función de recurso, de estos problemas a través de una linealización externa. Debido a que la función de recurso implica una solución de todos los programas lineales de segunda etapa, queremos evitar numerosas evaluaciones de funciones en la misma. Por tanto, utilizaremos ese término para construir un problema maestro en $x$, pero solo evaluaremos la función de recurso como si se tratase de un subproblema.

Esta técnica de planos de corte se llama el método $L-shaped$ en programación estocástica.

Para aplicar este algoritmo vamos a suponer que $\boldsymbol{\xi}$ es una variable aleatoria discreta, donde $\xi_1, ..., \xi_K$ son las posibles realizaciones con probabilidades $p_k$, $\forall k \in \{1, ..., K\}$.

En el Cuadro 1.9 podemos ver el esquema de nuestro algoritmo, para ello, vamos a suponer un problema formulado como en la Sección 1.2 .

***Método L − shaped***

PASO 0

Tomamos $r = v = s = 0$.

PASO 1

Sea $v = v + 1$. Resolvemos el problema lineal (maestro)

$$\begin{aligned} \text{mín } z =& c^T x + \theta && && (1.2)\\ \text{Sujeto a } & Ax = b && && (1.3)\\ & D_l x \geq d_l && l = 1, ..., r && (1.4)\\ & E_l x + \theta \geq e_l && l = 1, ..., s && (1.5)\\ & x \geq 0 && \theta \in \mathbb{R} && (1.6) \end{aligned}$$

Obtenemos la solución óptima $(x^v, \theta^v)$. Si $s = 0$ tomamos $\theta^v = -\infty$ y resolvemos el problema lineal sin considerar esta variable.

PASO 2

Si $x^v \in K_2$ ir al PASO 3. En caso contrario, añadir una restricción (1.4) como sigue.

Para todo $k = 1, ..., K$ resolvemos el problema lineal

$$\begin{aligned} \text{mín } w^{'} =& e^T v^+ + e^T v^- \\ \text{Sujeto a } & Wy + Iv^+ - Iv^- = h_k - T_k x^v \\ & y \geq 0, v^+ \geq 0, v^- \geq 0 \end{aligned}$$

donde $e^T = (1, ..., 1)$ e $I$ es la matriz identidad, hasta que, para algún $k$, el valor óptimo, $w^{'}$, sea estrictamente mayor que cero. Tras esto, sean $\sigma^v$ los multiplicadores símplex asociados y definimos

$$D_{r+1} = (\sigma^v)^T T_k$$

$$d_{r+1} = (\sigma^v)^T h_k$$

y ya tenemos la restricción (1.4) creada, esta restricción se llama corte de factibilidad. Tomamos $r = r + 1$ y volvemos al PASO 1. Si para todo $k$, $w^{'} = 0$ pasamos al PASO 3.

PASO 3

Para todo $k = 1, ..., K$, resolvemos el problema lineal

$$\begin{aligned} \text{mín } w =& q_k^T y \\ \text{Sujeto a } & Wy = h_k - T_k x^v \\ & y \geq 0 \end{aligned}$$

Considerando $\pi_k^v$ los multiplicadores símplex asociados a la solución óptima del problema anterior para todo $k$, definimos

$$E_{s+1} = \sum_{k=1}^{K} p_k (\pi_k^v)^T T_k$$

$$e_{s+1} = \sum_{k=1}^{K} p_k (\pi_k^v)^T h_k$$

Sea $w^v = e_{s+1} - E_{s+1} x^v$. Si $\theta^v \geq w^v$, la solución óptima del problema estocástico es $x^v$. En caso contrario, $s = s + 1$, se añade una restricción (1.5) y volver al PASO 1.

Cuadro 1.9: Esquema del algoritmo $L - shaped$

Como podemos observar, el algoritmo busca aproximar el programa estocástico a través del programa maestro, utilizando una linealización externa de $\mathcal{Q}$. A través de los cortes de factibilidad (1.4) determinamos el conjunto $K_2$, y los cortes de optimalidad (1.5) buscan aproximar $\mathcal{Q}$ en su dominio de finitud.

Ahora, vamos a resolver el siguiente ejemplo a través del método $L-$ *shaped*. Con él, además, ilustramos los cortes de optimalidad.

$$\begin{aligned}
\text{mín } z =& 100x_1 + 150x_2 + E_{\boldsymbol{\xi}}(\boldsymbol{q_1 y_1} + \boldsymbol{q_2 y_2}) \\
\text{Sujeto a } & x_1 + x_2 \leq 120 \\
& 6\boldsymbol{y_1} + 10\boldsymbol{y_2} \leq 60x_1 \\
& 8\boldsymbol{y_1} + 5\boldsymbol{y_2} \leq 80x_2 \\
& \boldsymbol{y_1} \leq \boldsymbol{d_1} \\
& \boldsymbol{y_2} \leq \boldsymbol{d_2} \\
& x_1 \geq 40, x_2 \geq 20, \boldsymbol{y_1}, \boldsymbol{y_2} \geq 0
\end{aligned}$$

donde $\boldsymbol{\xi^T} = (\boldsymbol{d_1}, \boldsymbol{d_2}, \boldsymbol{q_1}, \boldsymbol{q_2})$ toma los valores $(500, 100, -24, -28)$ y $(300, 300, -28, -32)$ con probabilidades 0.4 y 0.6, respectivamente.

En nuestra situación, para todo $x_1 \geq 40$ y $x_2 \geq 20$ siempre existe un vector $\boldsymbol{y}$ factible. Como tenemos que $\boldsymbol{d} \geq 0$, podemos tomar los valores $y_1 = 0$ e $y_2 = 0$. Entonces, $x \in K_2$ para todo $x$ factible de nuestro problema. Por tanto, podemos omitir el PASO 2 del algoritmo.

Ahora el método sigue de la siguiente manera.

*Iteración* 1

Tomamos $r = v = s = 0$.

PASO 1

Sea $v = v + 1$. Dado que $s = 0$, $\theta^v = -\infty$, e ignoramos el valor de $\theta$ del problema maestro. Resolvemos

$$\begin{aligned}
\text{mín } z =& 100x_1 + 150x_2 \\
\text{Sujeto a } & x_1 + x_2 \leq 120 \\
& x_1 \geq 40, x_2 \geq 20
\end{aligned}$$

y obtenemos la solución $(x^v)^T = (40, 20)$.

PASO 3

Resolvemos para $k = 1$

$$\begin{aligned}
\text{mín } w =& -24y_1 - 28y_2 \\
\text{Sujeto a } & 6y_1 + 10y_2 \leq 60 \cdot 40 \\
& 8y_1 + 5y_2 \leq 80 \cdot 20 \\
& 0 \leq y_1 \leq 500 \\
& 0 \leq y_2 \leq 100
\end{aligned}$$

y obtenemos los multiplicadores símplex $\pi_1^T = (0, -3, 0, -13)$.

Resolvemos para $k = 2$

$$\begin{aligned}\text{mín } w = &-28y_1 - 32y_2\\ \text{Sujeto a } &6y_1 + 10y_2 \leq 60 \cdot 40\\ &8y_1 + 5y_2 \leq 80 \cdot 20\\ &0 \leq y_1 \leq 300\\ &0 \leq y_2 \leq 300\end{aligned}$$

y obtenemos la los multiplicadores símplex $\pi_2^T = (-2.32, -1.76, 0, 0)$.

Considerando $h_k^T = (0, 0, \boldsymbol{d_1}, \boldsymbol{d_2})$, calculamos

$$e_{s+1} = 0.4 \cdot \pi_1^T \cdot h_1 + 0.6 \cdot \pi_2^T \cdot h_2 = -520.$$

Sea la matriz

$$T = \begin{pmatrix} -60 & 0 \\ 0 & -80 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Entonces se tiene

$$E_{s+1} = 0.4 \cdot \pi_1^T \cdot T + 0.6 \cdot \pi_2^T \cdot T = (83.52, 180.48)$$

Por último, calculamos

$$w^v = -520 - (83.52, 180.48) \cdot x^v = -7470.4 > \theta^v = -\infty$$

Por tanto, consideramos $s = s + 1$ y volvemos al PASO 1.

*Iteración* 2
Tomamos $v = s = 1$ y $r = 0$.
PASO 1
Sea $v = v + 1$. Resolvemos el problema maestro

$$\begin{aligned}\text{mín } z = &100x_1 + 150x_2 + \theta\\ \text{Sujeto a } &x_1 + x_2 \leq 120\\ &x_1 \geq 40, x_2 \geq 20\\ &83.52x_1 + 180.48x_2 + \theta \geq -520\end{aligned}$$

y obtenemos la solución $(x^v)^T = (40, 80)$ y $\theta^v = -18299.2$.

PASO 3
Resolvemos para $k = 1$

$$\begin{aligned}\text{mín } w = &-24y_1 - 28y_2\\ \text{Sujeto a } &6y_1 + 10y_2 \leq 60 \cdot 40\\ &8y_1 + 5y_2 \leq 80 \cdot 80\\ &0 \leq y_1 \leq 500\\ &0 \leq y_2 \leq 100\end{aligned}$$

y obtenemos los multiplicadores símplex $\pi_1^T = (-4, 0, 0, 0)$.

Resolvemos para $k = 2$

$$\begin{aligned} \text{mín } w = &- 28y_1 - 32y_2 \\ \text{Sujeto a } &6y_1 + 10y_2 \leq 60 \cdot 40 \\ &8y_1 + 5y_2 \leq 80 \cdot 80 \\ &0 \leq y_1 \leq 300 \\ &0 \leq y_2 \leq 300 \end{aligned}$$

y obtenemos los multiplicadores símplex $\pi_2^T = (-3.2, 0, -8.8, 0)$.

Calculamos

$$e_{s+1} = 0.4 \cdot \pi_1^T \cdot h_1 + 0.6 \cdot \pi_2^T \cdot h_2 = -1584$$

y

$$E_{s+1} = 0.4 \cdot \pi_1^T \cdot T + 0.6 \cdot \pi_2^T \cdot T = (211.2, 0)$$

Por último, calculamos

$$w^v = -1584 - (211.2, 0) \cdot x^v = -100032 > \theta^v = -18299.2$$

Por tanto, consideramos $s = s + 1$ y volvemos al PASO 1.

*Iteración* 3

Tomamos $v = s = 2$ y $r = 0$.

PASO 1

Sea $v = v + 1$. Resolvemos el problema maestro

$$\begin{aligned} \text{mín } z = &100x_1 + 150x_2 + \theta \\ \text{Sujeto a } &x_1 + x_2 \leq 120 \\ &x_1 \geq 40, x_2 \geq 20 \\ &83.52x_1 + 180.48x_2 + \theta \geq -520 \\ &211.2x_1 + \theta \geq -1584 \end{aligned}$$

y obtenemos la solución $(x^v)^T = (66.828, 53.172)$ y $\theta^v = -15697.994$.

PASO 3

Resolvemos para $k = 1$

$$\begin{aligned} \text{mín } w = &- 24y_1 - 28y_2 \\ \text{Sujeto a } &6y_1 + 10y_2 \leq 60 \cdot 66.828 \\ &8y_1 + 5y_2 \leq 80 \cdot 53.172 \\ &0 \leq y_1 \leq 500 \\ &0 \leq y_2 \leq 100 \end{aligned}$$

y obtenemos los multiplicadores símplex $\pi_1^T = (0, -3, 0, -13)$.

Resolvemos para $k = 2$

$$\begin{aligned} \text{mín } w = &- 28y_1 - 32y_2 \\ \text{Sujeto a } &6y_1 + 10y_2 \leq 60 \cdot 66.828 \\ &8y_1 + 5y_2 \leq 80 \cdot 53.172 \\ &0 \leq y_1 \leq 300 \\ &0 \leq y_2 \leq 300 \end{aligned}$$

y obtenemos los multiplicadores símplex $\pi_2^T = (-3.2, 0, -8.8, 0)$.

Calculamos

$$e_{s+1} = 0.4 \cdot \pi_1^T \cdot h_1 + 0.6 \cdot \pi_2^T \cdot h_2 = -2104$$

y

$$E_{s+1} = 0.4 \cdot \pi_1^T \cdot T + 0.6 \cdot \pi_2^T \cdot T = (115.2, 96)$$

Por último, calculamos

$$w^v = -2104 - (115.2, 96) \cdot x^v = -14907.0976 > \theta^v = -15697.994$$

Por tanto, consideramos $s = s + 1$ y volvemos al PASO 1.

*Iteración* 4

Tomamos $v = s = 3$ y $r = 0$.

PASO 1

Sea $v = v + 1$. Resolvemos el problema maestro

$$\begin{aligned} \text{mín } z =& 100x_1 + 150x_2 + \theta \\ \text{Sujeto a } & x_1 + x_2 \leq 120 \\ & x_1 \geq 40, x_2 \geq 20 \\ & 83.52x_1 + 180.48x_2 + \theta \geq -520 \\ & 211.2x_1 + \theta \geq -1584 \\ & 115.2x_1 + 96x_2 + \theta \geq -2104 \end{aligned}$$

y obtenemos la solución $(x^v)^T = (40, 33.75)$ y $\theta^v = -9952$.

PASO 3

Resolvemos para $k = 1$

$$\begin{aligned} \text{mín } w =& -24y_1 - 28y_2 \\ \text{Sujeto a } & 6y_1 + 10y_2 \leq 60 \cdot 40 \\ & 8y_1 + 5y_2 \leq 80 \cdot 33.75 \\ & 0 \leq y_1 \leq 500 \\ & 0 \leq y_2 \leq 100 \end{aligned}$$

y obtenemos los multiplicadores símplex $\pi_1^T = (-2.08, -1.44, 0, 0)$.

Resolvemos para $k = 2$

$$\begin{aligned} \text{mín } w =& -28y_1 - 32y_2 \\ \text{Sujeto a } & 6y_1 + 10y_2 \leq 60 \cdot 40 \\ & 8y_1 + 5y_2 \leq 80 \cdot 33.75 \\ & 0 \leq y_1 \leq 300 \\ & 0 \leq y_2 \leq 300 \end{aligned}$$

y obtenemos los multiplicadores símplex $\pi_2^T = (-3.2, 0, -8.8, 0)$.

Calculamos

$$e_{s+1} = 0.4 \cdot \pi_1^T \cdot h_1 + 0.6 \cdot \pi_2^T \cdot h_2 = -1584$$

y

$$E_{s+1} = 0.4 \cdot \pi_1^T \cdot T + 0.6 \cdot \pi_2^T \cdot T = (165.12, 46.08)$$

Por último, calculamos

$$w^v = -1584 - (165.12, 46.08) \cdot x^v = -9744 > \theta^v = -9952$$

Por tanto, consideramos $s = s + 1$ y volvemos al PASO 1.

*Iteración* 5
Tomamos $v = s = 4$ y $r = 0$.
PASO 1
Sea $v = v + 1$. Resolvemos el problema maestro

$$\begin{aligned} \text{mín } z =& 100x_1 + 150x_2 + \theta \\ \text{Sujeto a } & x_1 + x_2 \leq 120 \\ & x_1 \geq 40, x_2 \geq 20 \\ & 83.52x_1 + 180.48x_2 + \theta \geq -520 \\ & 211.2x_1 + \theta \geq -1584 \\ & 115.2x_1 + 96x_2 + \theta \geq -2104 \\ & 165.12x_1 + 46.08x_2 + \theta \geq -1584 \end{aligned}$$

y obtenemos la solución $(x^v)^T = (46.667, 36.25)$ y $\theta^v = -10960$.
PASO 3
Resolvemos para $k = 1$

$$\begin{aligned} \text{mín } w =& -24y_1 - 28y_2 \\ \text{Sujeto a } & 6y_1 + 10y_2 \leq 60 \cdot 46.667 \\ & 8y_1 + 5y_2 \leq 80 \cdot 36.25 \\ & 0 \leq y_1 \leq 500 \\ & 0 \leq y_2 \leq 100 \end{aligned}$$

y obtenemos los multiplicadores símplex $\pi_1^T = (0, -3, 0, -13)$.
Resolvemos para $k = 2$

$$\begin{aligned} \text{mín } w =& -28y_1 - 32y_2 \\ \text{Sujeto a } & 6y_1 + 10y_2 \leq 60 \cdot 46.667 \\ & 8y_1 + 5y_2 \leq 80 \cdot 36.25 \\ & 0 \leq y_1 \leq 300 \\ & 0 \leq y_2 \leq 300 \end{aligned}$$

y obtenemos la los multiplicadores símplex $\pi_2^T = (-3.2, 0, -8.8, 0)$.
Calculamos

$$e_{s+1} = 0.4 \cdot \pi_1^T \cdot h_1 + 0.6 \cdot \pi_2^T \cdot h_2 = -2104$$

y

$$E_{s+1} = 0.4 \cdot \pi_1^T \cdot T + 0.6 \cdot \pi_2^T \cdot T = (115.2, 96)$$

Por último, calculamos

$$w^v = -2104 - (115.2, 96) \cdot x^v = -10960 = \theta^v$$

Por tanto el algoritmo finaliza. Y tenemos que la solución óptima a nuestro problema estocástico es $x^T = (46.667, 36.25)$.

# Capítulo 2

# El formato SMPS

En este capítulo veremos una de las formas de implementar programas lineales estocásticos, para después resolverlos mediante algún solver disponible en el servidor de $NEOS$, [CMM98]; [Dol01]; [GM97]. Podemos encontrar el enlace a internet en [NEO17]. $NEOS$ es un servicio, con acceso gratuito, que permite resolver problemas de optimización numérica, incluyendo programación lineal, entera y no lineal, a través de internet mediante sus más de 60 solvers.

EL formato $SMPS$ que es el que presentamos en este capítulo, [GK08], requiere de tres archivos de texto que debemos crear: un archivo principal, un archivo de tiempo y un archivo estocástico. Podemos ver ejemplos de utilización de este lenguaje en el Apéndice A.

La estructura es igual para los tres archivos:

| | |
|---|---|
| Registro de código | Columnas $2-3$ |
| Primer registro de letras | Columnas $5-12$ |
| Segunda registro de letras | Columnas $15-22$ |
| Primer registro numérico | Columnas $25-36$ |
| Tercer registro de letras | Columnas $40-47$ |
| Segundo registro numérico | Columnas $50-61$ |

Cuadro 2.1: Estructura de un archivo en lenguaje $SMPS$.

Además, se usa una cabecera con el nombre del programa.

## 2.1. Archivo principal

El archivo principal enumera toda la información determinista del problema. En el, incluimos los nombres de las filas y columnas, así como el tipo de cada restricción. Además, añadimos los valores de las variables que después serán reemplazados de manera estocástica, para lo cual le asignaremos un valor que no tiene que ser significativo.

La estructura es siempre de la siguiente forma:

**NAME.** El nombre del problema se da en el segundo registro de nombre. Este debe ser el mismo para los archivos de tiempo y estocástico.

**ROWS.** En esta sección indicamos el nombre de la función objetivo, el tipo de cada restricción y su nombre. En el registro de código indicamos el tipo de restricción y en el primer registro de letras el nombre de la fila. De manera que, $L$ indica restricciones de menor o igual, $E$ de igualdad y $G$ de mayor o igual. Indicamos $N$ para la función objetivo.

**COLUMNS.** Damos los coeficientes distintos de cero de la matriz de restricción en orden de aparición. El registro de código aparece vacío. Entonces, en el primer registro de letras se escribe el nombre de la variable, o columna, en el segundo registro de letras el nombre de la fila en la que aparece y en el primer registro numérico el valor del coeficiente. Podemos utilizar el tercer registro de letras para indicar otra fila, en la que aparezca, y darle el valor del coeficiente en el segundo registro numérico.

**RHS.** En esta sección indicamos el valor derecho de cada restricción. En el primer registro de nombre indicamos un nombre de identificación, ($RHS$), y en el segundo registro de nombre la fila a la cuál, en el primer registro numérico, damos el valor de su restricción. Al igual que en la sección **COLUMNS** podemos añadir otra fila, indicando su nombre en el tercer registro de letras y su valor en el segundo registro numérico. Nótese que las restricciones iguales a cero no es necesario indicarlas.

**ENDATA.** Se termina el archivo principal.

## 2.2. Archivo de tiempo

En el archivo de tiempo dividimos cada escenario en las distintas etapas del problema. La estructura es de la forma:

**TIME.** En el segundo registro de letras indicamos el nombre del problema, que debe coincidir con el del archivo principal y estocástico.

**PERIODS.** Como cabecera, escribimos en el primer registro de letras *PERIODS*. Para esta sección tenemos dos casos. Si en el archivo principal, las variables están en orden de asignación, entonces solo es necesario indicar en el primer registro de letras el nombre de la primera variable de cada etapa, así como la fila, a la que pertenece, en el segundo registro de nombre. Si estas no están en orden, debemos indicar en el segundo campo de letras de la cabecera la palabra *EXPLICIT*, e indicar en una lista completa de columnas y filas a la etapa que pertenecen.

**ENDATA.** Se termina el archivo de tiempo.

## 2.3. Archivo estocástico

En el archivo estocástico aportamos información sobre las variables aleatorias, para así construir un problema determinista equivalente. Tenemos diferentes formas, dependiendo del problema, de construir este archivo.

**STOCH.** En el segundo registro de letras indicamos el nombre del problema, que debe coincidir con el del archivo principal y de tiempo.

**SCENARIOS.** Este es el formato más utilizado debido a su flexibilidad, ya que permite modelar dependencia tanto entre periodos de tiempo como dentro de los mismos. Para ello, describimos un árbol de sucesos de manera explícita.

Cada escenario, representa un camino desde un nodo *raíz* hasta un nodo final, llamados *hojas*. Las ramas de un período pueden ser compartidas entre escenarios, lo que permite comprimir el tamaño del archivo estocástico permitiendo no repetir información redundante.

Para codificar la información, utilizamos dos tipos de registros. Primero indicamos en el registro de código, $SC$, después en el primer registro de letras el nombre del escenario. En el segundo registro de letras el escenario del cual se ramifica, y en el supuesto caso de ser el primer escenario, escribiremos la palabra *ROOT*. En el primer registro numérico, indicamos la probabilidad de que ocurra esa ruta. Por último, en el tercer registro de letras se añade el período de tiempo en el cual empieza la ramificación.

Dentro de cada escenario añadiremos, en una lista, de manera que en el primer registro de letras aparezca el nombre de la columna, en el segundo registro de letras el nombre de la fila y en el primer registro numérico el valor de esa variable.

**BLOCKS.** En esta sección, mostramos como añadir vectores aleatorios multidimensionales donde las componentes pueden estar correlacionadas pero son independientes de otros elementos aleatorios del problema. En la cabecera, en el primer registro de letras indicamos la palabra $BLOCKS$ y en el segundo registro de letras el tipo de bloque, por ejemplo, $DISCRETE$ o $MVNORMAL$.

En el caso de bloques del tipo $DISCRETE$, añadimos en el registro de código $BL$, en el primer registro numérico el nombre del bloque y en el segundo registro numérico el período en el que efectuamos los cambios, por último en el primer registro numérico la probabilidad de que suceda este conjunto de valores. Dentro de cada bloque añadiremos, en una lista, de manera que en el primer registro de letras aparezca el nombre de la columna, en el segundo registro de letras el nombre de la fila y en el primer registro numérico el valor de esa variable.

**ENDATA.** Se termina el archivo estocástico.

## 2.4. Un ejemplo

En esta sección vamos a detallar la forma de proceder en $NEOS$ una vez creados los 3 archivos de texto. Para ello, haremos uso del problema del granjero con incertidumbre en el caso discreto.

El archivo principal que vamos a utilizar es de la forma:

```
NAME          FARMER
ROWS
 N  WEALTH
 L  BUDGET
 G  BAL1
 G  BAL2
 L  BAL3
 L  BAL4
COLUMNS
    X1        BUDGET    1.00           BAL1      +2.5
    X2        BUDGET    1.00           BAL2      +3
    X3        BUDGET    1.00           BAL3      -20
    Y1        BAL1      1.00
    Y2        BAL2      +1.00
    W1        BAL1      -1.00
    W2        BAL2      -1.00
    W3        BAL3      +1.00          BAL4      +1.00
    W4        BAL3     +1.00
    X1        WEALTH    150
    X2        WEALTH    230
    X3        WEALTH    +260
    Y1        WEALTH    +238
    Y2        WEALTH    +210
    W1        WEALTH    -170
    W2        WEALTH    -150
    W3        WEALTH    -36
    W4        WEALTH    -10
RHS
    RHS       BUDGET    500            BAL1      200
RHS       BAL2      240            BAL4      6000
ENDATA
```

Como podemos ver, este está formado por el nombre del problema, el nombre de cada restricción, así como, si es de igualdad, menor e igual o mayor e igual. Además, la fila $WEALTH$ es la función objetivo ya que en el registro de código indicamos $N$. A continuación, se muestran las variables que componen cada restricción y el valor del coeficiente que les acompaña. Por último, indicamos el valor derecho de cada restricción y de no aparecer se considera el valor 0.

El archivo de tiempo es de la forma:

```
TIME          FARMER
PERIODS
    X1        BUDGET                  TODAY
    Y1        BAL1                    TOMORROW
ENDATA
```

Dado que las variables están ordenadas en el archivo principal, solo debemos indicar la primera variable de cada etapa. En este caso, desde $X1$ hasta $X3$ son las variables de la primera etapa, y a partir de $Y1$, todas las demás variables pertenecen a la segunda etapa.

Por último, el archivo estocástico es de la forma:

```
STOCH         FARMER
BLOCKS        DISCRETE
 BL BLOCK1    TOMORROW  0.33334
    X1        BAL1      2.5
    X2        BAL2      3
    X3        BAL3      -20
 BL BLOCK1    TOMORROW  0.33333
    X1        BAL1      3
    X2        BAL2      3.6
    X3        BAL3      -24
 BL BLOCK1    TOMORROW  0.33333
    X1        BAL1      2
    X2        BAL2      2.4
    X3        BAL3      -16
ENDATA
```

En este archivo hemos utilizado el tipo $BLOCKS$. Con igual probabilidad, hemos construido tres vectores aleatorios multidimensionales independientes, que aportan el valor de los rendimientos de los diferentes cultivos en los tres escenarios.

Una vez creados los tres archivos de texto, accederemos a $NEOS$ a través del enlace que nos proporciona [NEO17]. Una vez dentro, en la sección de solvers debemos buscar aquel relacionado con el tipo de problema que intentamos resolver, problemas lineales estocásticos. En la Figura 2.1 podemos ver los solvers disponibles en $NEOS$ para nuestro caso.

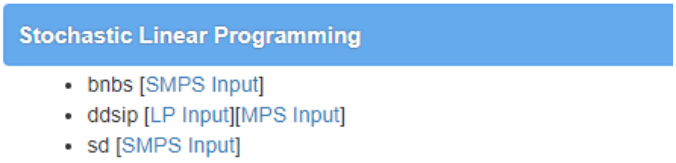


Figura 2.1: Solvers disponibles en $NEOS$ para resolver problemas de programación lineal estocástica.

Nosotros, en esta situación, utilizaremos el solver *bnbs* ya que soporta el formato $SMPS$ que estamos tratando en este capítulo. A continuación, en la Figura 2.2 apreciamos los elementos de

entrada que debemos añadir sobre nuestro problema, es decir, archivo principal, de tiempo y estocástico. Además, debemos indicar el $LP$-solver (el solver para problemas de programación lineal) que se va a utilizar, por ejemplo $CPLEX$.

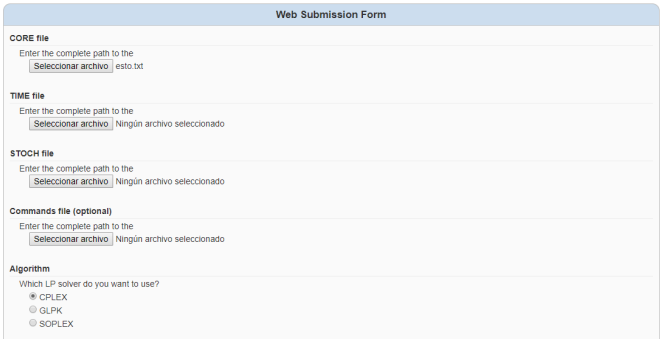


Figura 2.2: Formulario de elementos de entrada y algoritmos para el solver *bnbs*.

Una vez resuelto el problema, en la Figura 2.3 podemos apreciar los resultados que proporciona $NEOS$.

```
full iters 7, bounce iters 0
 value is -108390.009600
optimal value -108390.009600
node    1 prob 1.000000e+00 var                         X1 varno   0 : 1.700000000e+02
node    1 prob 1.000000e+00 var                         X2 varno   1 : 8.000000000e+01
node    1 prob 1.000000e+00 var                         X3 varno   2 : 2.500000000e+02
node    2 prob 3.333400e-01 var                         Y1 varno   3 : 0.000000000e+00
node    2 prob 3.333400e-01 var                         Y2 varno   4 : 0.000000000e+00
node    2 prob 3.333400e-01 var                         W1 varno   5 : 2.250000000e+02
node    2 prob 3.333400e-01 var                         W2 varno   6 : -0.000000000e+00
node    2 prob 3.333400e-01 var                         W3 varno   7 : 5.000000000e+03
node    2 prob 3.333400e-01 var                         W4 varno   8 : 0.000000000e+00
node    3 prob 3.333300e-01 var                         Y1 varno   3 : 0.000000000e+00
node    3 prob 3.333300e-01 var                         Y2 varno   4 : 0.000000000e+00
node    3 prob 3.333300e-01 var                         W1 varno   5 : 3.100000000e+02
node    3 prob 3.333300e-01 var                         W2 varno   6 : 4.800000000e+01
node    3 prob 3.333300e-01 var                         W3 varno   7 : 6.000000000e+03
node    3 prob 3.333300e-01 var                         W4 varno   8 : 0.000000000e+00
node    4 prob 3.333300e-01 var                         Y1 varno   3 : 0.000000000e+00
node    4 prob 3.333300e-01 var                         Y2 varno   4 : 4.800000000e+01
node    4 prob 3.333300e-01 var                         W1 varno   5 : 1.400000000e+02
node    4 prob 3.333300e-01 var                         W2 varno   6 : 0.000000000e+00
node    4 prob 3.333300e-01 var                         W3 varno   7 : 4.000000000e+03
node    4 prob 3.333300e-01 var                         W4 varno   8 : 0.000000000e+00
                                X1 mean :1.700000000e+02
                                X2 mean :8.000000000e+01
                                X3 mean :2.500000000e+02
                                Y1 mean :0.000000000e+00
                                Y2 mean :1.599984000e+01
                                W1 mean :2.250000000e+02
                                W2 mean :1.599984000e+01
                                W3 mean :5.000000000e+03
                                W4 mean :0.000000000e+00
```

Figura 2.3: Resultados obtenidos en $NEOS$ de nuestro problema.

Aquí podemos ver el valor de la función objetivo, optimal value, 108390. Por otra parte, en la columna superior derecha tenemos los valores de las variables $X1, X2$ y $X3$ comunes a los tres escenarios, y el valor de las variables $Y1, Y2, W1, W2, W3$ y $W4$ en cada escenario, es decir, según el tipo de rendimiento.

# Capítulo 3

# Problemas de rutas de vehículos (VRP)

En este capítulo hablaremos sobre los problemas de rutas de vehículos, [GRWE08], así como de su formulación matemática. Más adelante veremos un método heurístico para resolverlos, ya que, debido a la alta complejidad de estos problemas, los métodos exactos suelen ser demasiado lentos computacionalmente. Es por ello, que estas formas de resolución son procedimientos simples que mediante una aproximación intuitiva consiguen una solución de alta calidad. Por último, aplicaremos lo visto en este capítulo a algún ejemplo.

Los problemas de rutas de vehículos consisten en abastecer una demanda de servicios para un conjunto de clientes. Desde uno o varios depósitos, con una flota de vehículos, debemos encontrar rutas óptimas, es decir, con el menor coste posible, que comiencen y terminen en los depósitos, visitando a los diferentes clientes una única vez.

Las diferentes características de los problemas de rutas dan lugar a la siguiente clasificación:

| Demanda | Restricciones de capacidad | Nombre del problema |
|---|---|---|
| Nodos | No | Problema del viajante ($TSP$) |
| | Sí | Problema de rutas de vehículos ($VRP$) |
| Arcos | No | Problema del cartero chino o rural ($CPP$) o ($RPP$) |
| | Sí | Problema de rutas con capacidades ($CARP$) |

Cuadro 3.1: Clasificación de los problemas de rutas.

Además, existen otro tipo de restricciones, como un horario de servicio asociado a cada cliente, múltiples depósitos, entre otras. Nosotros vamos a centrarnos en los problemas de rutas de vehículos ($VRP$).

## 3.1. El problema de rutas de vehículos

Para empezar, debemos definir la red que une tanto a los clientes entre ellos mismos como con el depósito, así como los costes de circular por ella.

Sea la red $G = (V, E, C)$ donde $V$ es el conjunto de clientes, representados por nodos desde 1 hasta $n$ y 0 representa el depósito, y se tiene $d_i$ como la demanda del nodo $i$ para todo $i \in V\backslash\{0\}$. En esta situación $E$ indica el conjunto de caminos que unen cada par de nodos y que tienen unos costes asociados, es decir, $(i, j) \in E$ es el camino de $i$ hasta $j$ con coste $c_{ij}$. En nuestro caso, vamos a suponer que siempre existe un camino que une cualquier par de nodos, en cuyo caso denominaremos a $G$ completo. Por último, disponemos de una flota de $K$ vehículos con una capacidad conocida e igual a $C$.

Suponiendo que cada cliente ha de ser visitado una única vez, por un único vehículo, tal que todas las rutas deben empezar y acabar en el depósito y la suma de las demandas de los clientes visitados no puede exceder la capacidad del vehículo, nos interesa encontrar el conjunto de rutas de mínimo coste.

Este problema puede ser descrito de la siguiente manera:

$$\text{mín} \quad \sum_{i\in V}\sum_{j\in V} c_{ij} \sum_{k=1}^{K} x_{ijk} \tag{3.1}$$

$$\text{Sujeto a} \quad \sum_{k=1}^{K} y_{ik} = 1, \quad \forall i \in V\backslash\{0\} \tag{3.2}$$

$$\sum_{k=1}^{K} y_{0k} = K \tag{3.3}$$

$$\sum_{j\in V} x_{ijk} = \sum_{j\in V} x_{jik} = y_{ik}, \quad \forall i \in V, k = 1, ..., K \tag{3.4}$$

$$\sum_{i\in V} d_i y_{ik} \leq C, \quad \forall k = 1, ..., K \tag{3.5}$$

$$\sum_{i\in S}\sum_{j\notin S} x_{ijk} \geq y_{hk}, \quad \forall S \subseteq V\backslash\{0\}, h \in S, k = 1, ..., K \tag{3.6}$$

$$y_{ik} \in \{0, 1\}, \quad \forall i \in V, k = 1, ..., K \tag{3.7}$$

$$x_{ijk} \in \{0, 1\}, \quad \forall i, j \in V, k = 1, ..., K \tag{3.8}$$

donde $y_{ik} = 1$ si el vehículo $k$, con $k \in \{1, ..., K\}$, visita el cliente $i$, con $i \in \{0, ..., n\}$, en otro caso $y_{ik} = 0$. Y $x_{ijk} = 1$ si el vehículo $k$, con $k \in \{1, ..., K\}$, viaja desde el cliente $i$, con $i \in \{0, ..., n\}$, hasta el cliente $j$, con $j \in \{0, ..., n\}$, en otro caso $x_{ijk} = 0$.

Entonces se tiene que la función (3.1) es el coste total del problema, que buscamos minimizar. La restricción (3.2) indica que cada nodo será visitado por un único vehículo, y (3.3) que del depósito sale toda la flota. Las restricciones (3.4) aseguran que todo cliente es visitado una única vez en alguna ruta, y (3.5) que la demanda de los nodos visitados por un vehículo no supere su capacidad de carga total. Por último, las restricciones de la forma (3.6) eliminan cualquier posibilidad de subrutas, es decir, impide que se formen ciclos entre clientes.

Una vez visto el problema de rutas, proponemos el algoritmo de ahorros $Clarke$ y $Wright$, [CW64], que es una heurística para resolver problemas de rutas de vehículos.

Hay que tener en cuenta que el $VRP$ presentado es un problema de programación lineal entera. Los métodos de resolución clásicos como ramificación y acotación o planos de corte sólo son aplicables en problemas pequeños. No obstante, tiene interés considerar estos métodos para validar el modelo o poder hacer comparaciones cuando aplicamos métodos heurísticos tanto de su éxito (bondad de las soluciones) como de su eficiencia (rapidez). [GRWE08] es una referencia adecuada para saber más de estos métodos.

## 3.2. El algoritmo de ahorros

Este algoritmo es uno de los más utilizados para los problemas de rutas de vehículos, dada su sencillez y rapidez. Se basa en el calculo de los ahorros al unir dos nodos, clientes, y no tener que volver al depósito en medio de dicho viaje. El ahorro para cualquier par $i, j \in V\backslash\{0\}$ viene dado por:

$$S_{ij} = c_{i0} + c_{0j} - c_{ij}. \tag{3.9}$$

Entonces la ecuación (3.9) nos indica el coste ahorrado por viajar de $i$ hacia $j$, es decir, utilizar el arco $(i, j)$, en lugar de, utilizar los arcos $(i, 0)$ y $(0, j)$.

Por tanto, la descripción algorítmica del método está representada en el Cuadro 3.2.

***Algoritmo de Clarke y Wright***

INICIALIZACIÓN
Para cada cliente $i \in V\backslash\{0\}$ crear la ruta $(0, i, 0)$.

PASO 1
Calcular los ahorros $S_{ij} = c_{i0} + c_{0j} - c_{ij}$ para todo $i, j \in V\backslash\{0\}$.

PASO 2
Ordenar los ahorros de forma no decreciente.

PASO 3
Considerar $S_{\hat{i}\hat{j}} = \max S_{ij}$. Si $S_{\hat{i}\hat{j}} > 0$ y considerando $r_{\hat{i}}$ y $r_{\hat{j}}$, las rutas que contienen a $\hat{i}$ y $\hat{j}$, si en $r_{\hat{i}}$, $\hat{i}$ es el último cliente y en $r_{\hat{j}}$, $\hat{j}$ es el primer cliente y además, las demandas de dichos clientes no superan la capacidad del vehículo, la ruta es factible, eliminar los arcos $(i, 0)$ y $(j, 0)$ e implementar la ruta $(i, j)$. Eliminar los ahorros $S_{\hat{i}\hat{j}} = S_{\hat{j}\hat{i}}$.

PASO 4
Repetir PASO 3 hasta que no existan más fusiones posibles.

Cuadro 3.2: Esquema del algoritmo de Clarke y Wright

Nótese que solo es necesario calcular los ahorros una vez para todo el algoritmo, por tanto, deben guardarse para no realizar el PASO 1 constantemente. Además, tengamos en cuenta que ahorros iguales pueden dar lugar a soluciones distintas.

Veamos un ejemplo de problemas de rutas de vehículos para poner a prueba esta heurística.

### 3.2.1. Un ejemplo de rutas

Un vehículo tiene que visitar cuatro clientes, $(A, B, C, D)$, y la ruta debe empezar y terminar en el depósito, 0. Sabemos que el vehículo es único y tiene una capacidad de 10 unidades. No existe ningún otro tipo de restricciones.

Por tanto, en nuestro problema tenemos un grafo $G = (V, E, C)$, tal que, $V = \{0, A, B, C, D\}$ y $E$ corresponde a los arcos $(i, j)$ de viajar desde el nodo $i$ hasta el nodo $j$. Además, $K = 1$ y $C = 10$.

Ahora, nos dicen que los clientes $A$, $B$ y $D$ tienen una demanda igual a 2, mientras que $C$ tiene una demanda igual a 1.

En la Figura 3.1 podemos ver una representación gráfica de $G$, en la que no incluimos todos los arcos por motivos de orden.

Las distancias entre dos puntos $i, j \in V$ están representadas en la Tabla 3.3. En ella, podemos observar que se obtiene una matriz simétrica, por tanto, esto nos dice que la distancia es igual en las dos direcciones.

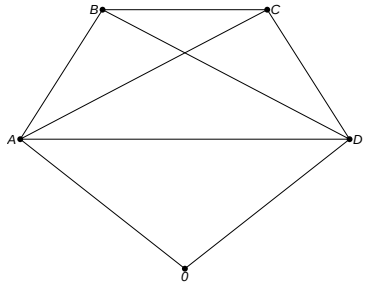


Figura 3.1: Representación del grafo $G$.

| | 0 | $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|---|---|
| 0 | – | 6 | 4 | 5 | 3 |
| $A$ | 6 | – | 4 | 8 | 4 |
| $B$ | 4 | 4 | – | 1 | 4 |
| $C$ | 5 | 8 | 1 | – | 3 |
| $D$ | 3 | 4 | 4 | 3 | – |

Cuadro 3.3: Distancias entre nodos.

Ahora que tenemos todos los datos del problema, procedamos a la resolución del mismo. Para ello utilizaremos el algoritmo de ahorros. Entonces implementamos las 4 rutas resultantes de ir desde el depósito a cada cliente y volver, de la forma $(0, i, 0)$ tal que $i \in V\backslash\{0\}$. Tenemos que, el coste total inicial de implementar este tipo de rutas es:

$$C_T = c_{01} + c_{10} + c_{02} + c_{20} + c_{03} + c_{30} + c_{04} + c_{40} = 6 \times 2 + 4 \times 2 + 5 \times 2 + 3 \times 2 = 36.$$

Procedamos con el PASO 1 y calculemos los ahorros. Al ser la matriz de distancias simétrica se tiene que $S_{ij} = S_{ji}$ para cualquier par $i, j \in V\backslash\{0\}$. Por tanto,

$$\begin{aligned}
S_{AB} &= S_{BA} = c_{A0} + c_{0B} - c_{AB} = 6 + 4 - 4 = 6 \\
S_{AC} &= S_{CA} = c_{A0} + c_{0C} - c_{AC} = 6 + 5 - 8 = 3 \\
S_{AD} &= S_{DA} = c_{A0} + c_{0D} - c_{AD} = 6 + 3 - 4 = 5 \\
S_{BC} &= S_{CB} = c_{B0} + c_{0C} - c_{BC} = 4 + 5 - 1 = 8 \\
S_{BD} &= S_{DB} = c_{B0} + c_{0D} - c_{BD} = 4 + 3 - 4 = 3 \\
S_{CD} &= S_{DC} = c_{C0} + c_{0D} - c_{CD} = 5 + 3 - 3 = 5
\end{aligned}$$

En el PASO 2 debemos ordenador los ahorros de manera no decreciente, es decir,

$$S_{CA} \leq S_{AC} \leq S_{BD} \leq S_{DB} \leq S_{AD} \leq S_{DA} \leq S_{DC} \leq S_{CD} \leq S_{BA} \leq S_{AB} \leq S_{BC} \leq S_{CB}$$

Y pasamos al PASO 3, donde debemos escoger el ahorro de mayor valor, que en nuestro caso es $S_{CB} = S_{BC} > 0$. Dado que el vehículo tiene una capacidad de 10 unidades y las demandas de $C$ y $B$ son 1 y 2, respectivamente, y ya que $C$ es último cliente en su ruta, $(0, C, 0)$, y $B$ primero, $(0, B, 0)$, tenemos una ruta factible. Por tanto, eliminamos los arcos $(C, 0)$ y $(0, B)$ e implementamos el arco $(C, B)$, con ruta resultante $(0, C, B, 0) = (0, B, C, 0)$, por ser la matriz de distancias simétrica. Por último, eliminamos los ahorros $S_{BC}$ y $S_{CB}$.

Cualquier elección de otro ahorro que contenga a $C$ y $B$ es factible, ya que la capacidad sobrante del vehículo es 7. Por lo que ahora, el ahorro de mayor valor es $S_{AB} = S_{BA} > 0$. Como $B$ es último cliente de la ruta $(0, C, B, 0)$, y $A$ primero de $(0, A, 0)$, tenemos una ruta factible. Por tanto, eliminamos los arcos $(B, 0)$ y $(0, A)$ e implementamos el arco $(B, A)$, con ruta resultante $(0, C, B, A, 0) = (0, A, B, C, 0)$. Por último, eliminamos los ahorros $S_{AB}$ y $S_{BA}$.

La capacidad disponible de nuestro vehículo es de 5 unidades, por lo que podemos seguir visitando clientes. Ahora, el ahorro de mayor valor es $S_{CD} = S_{DC} > 0$. Como $C$ es último cliente de la ruta $(0, A, B, C, 0)$, y $D$ primero de $(0, D, 0)$, tenemos una ruta factible. Por tanto, eliminamos los arcos $(C, 0)$ y $(D, 0)$ e implementamos el arco $(C, D)$, con ruta resultante $(0, A, B, C, D, 0)$. Por último, eliminamos los ahorros $S_{CD}$ y $S_{DC}$.

Terminamos nuestro algoritmo, al no tener más rutas factibles a escoger. Las rutas resultantes pueden verse en la Figura 3.2, y ya que la matriz de distancias es simétrica, la dirección de las rutas es irrelevante. Tenemos un coste de $C_{vrp} = 6 + 4 + 1 + 3 + 3 = 17$.

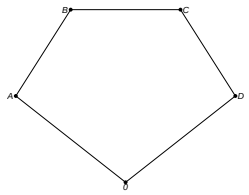


Figura 3.2: Solución del ejemplo del problema de rutas de vehículos.

Ahora, supongamos que la demanda del cliente $C$ es aleatoria. Es decir, vamos a suponer que la demanda de $C$ es 1 o 7 con probabilidad igual a $\frac{1}{2}$, en cada caso. Ya que la capacidad del vehículo es de 10 unidades, no podemos considerar la ruta obtenida con el algoritmo de ahorros, ya que cuando la demanda es de 7 unidades esa ruta no es factible. Por tanto, el tratamiento de la incertidumbre depende del momento en que esté disponible la información. Para ello, veamos tres formas de proceder que proporcionan diferentes soluciones.

La solución información perfecta, $(WS)$, es cuando se conoce la demanda antes de iniciar la ruta, y podemos hacer la elección de la ruta óptima después de obtener la información del nivel de la demanda.

- Si la demanda en $C$ es 1, el vehículo puede satisfacer la demanda en los 4 clientes. La solución optima es la calculada anteriormente de coste 17.

- Si la demanda en C es 7, es necesario recorrer dos rutas ya que el vehículo no puede llevar tanto en un sólo viaje. Las rutas resultantes son $(0, C, B, 0)$ y $(0, A, D, 0)$ con un coste total de 23.

Como ambas situaciones ocurren con la misma probabilidad, las rutas de costes 17 y 23 se recorren medio tiempo cada una. Por tanto, la media de distancia viajada es:

$$WS = \frac{1}{2} \times 17 + \frac{1}{2} \times 23 = 20.$$

La solución del valor esperado, $(EEV)$, es cuando la demanda no se conoce de antemano, solo al llegar a $C$. Si tenemos en cuenta la demanda esperada del cliente $C$, es decir, $\frac{1}{2} \times 1 + \frac{1}{2} \times 7 = 4$, tenemos que la capacidad del vehículo es suficiente para todos los clientes y la ruta óptima es $(0, A, B, C, D, 0)$, calculada anteriormente.

Pero esto no significa que no existe incertidumbre. La demanda de $C$ se revela al llegar al cliente, por tanto,

- Cuando la demanda de $C$ es 1, la demanda total es 7 y el vehículo puede satisfacer a todos los clientes. La solución óptima es la calculada anteriormente de coste 17.
- Si al llegar a $C$ la demanda es 7, la carga del vehículo es de 4 unidades por lo que el vehículo no puede satisfacer al cliente $C$, debe volver al depósito antes de continuar con su trayecto. Por lo que la solución vendrá dada de la forma $(0, A, B, C, 0, C, D, 0)$ con un coste de 27.

Al igual que en la situación anterior, ambos casos ocurren con la misma probabilidad, las rutas de costes 17 y 27 se recorren medio tiempo cada una. Por tanto, la media de distancia viajada es:

$$EEV = \frac{1}{2} \times 17 + \frac{1}{2} \times 27 = 22.$$

La solución óptima, $(RP)$, busca mejorar la elección de la ruta al considerar la aleatoriedad.

Observemos que, en esta situación, la ruta óptima obtenida $(0, A, B, C, D, 0)$ puede ser recorrida siguiendo el orden $(0, D, C, B, A, 0)$ sin cambiar el coste total. En esta situación,

- Si la demanda al llegar a $C$ es 1, el vehículo puede satisfacer a todos los clientes. La ruta óptima es la considerada anteriormente de coste 17.
- En otra situación, si la demanda de $C$ es 7, el vehículo puede satisfacer a $C$, ya que solo ha visitado $D$ con demanda 2. Una vez en $C$ el vehículo debe volver al depósito y continuar su trayecto. Por tanto, la ruta del vehículo es de la forma $(0, D, C, 0, B, A, 0)$ con un coste de 25.

Ya que ambas situaciones ocurren con igual probabilidad, la media de distancia viajada es:

$$\frac{1}{2} \times 17 + \frac{1}{2} \times 24 = 21.$$

Tener que decidir sobre la ruta planificada es una decisión de primera etapa, tomada antes de conocer los parámetros aleatorios. Cuando se revela la incertidumbre son posibles acciones adicionales o de segunda etapa.

Buscando nuevas soluciones, si planeamos la siguiente ruta $(0, C, B, A, D, 0)$, cuando la demanda de $C$ es 1 el coste total es de 17. Sin embargo, cuando la demanda es 7 recorremos las rutas $(0, C, B, 0)$ y $(0, A, D, 0)$ con ruta resultante $(0, C, B, 0, A, D, 0)$ de coste 23. Por tanto,

$$RP = \frac{1}{2} \times 17 + \frac{1}{2} \times 23 = 20.$$

De todo esto, se sigue que:

$$WS \leq RP \leq EEV$$

es decir, siempre es mejor obtener la información por adelantado, y en caso de no ser posible, es mejor resolver el problema estocástico que pretender que la incertidumbre no exista.

## 3.3. Panorámica sobre los problemas de rutas de vehículos estocásticos

En el ejemplo anterior, hemos visto que algunos parámetros del problema $VRP$ pueden ser aleatorios. Cuando nos encontramos en esta situación estamos ante problemas de rutas de vehículos estocásticos, $SVRP$. Es decir, consideramos uno o más parámetros como estocásticos o desconocidos durante el horizonte de planificación.

Además, estos pueden surgir cuando las rutas no se desean, o no se pueden, rediseñar todos los días y debemos estimar los valores futuros.

Este tipo de problemas se aborda a través de los $VRP$ clásicos, formulando lo que se considera una solución planificada a priori, y a continuación, se toman acciones correctivas. Es por tanto, una familia de problemas que combinan las características de la programación entera y estocástica de dos etapas. Dado que incluso muchas propiedades fundamentales de los $VRP$ ya no se mantienen, a menudo se consideran computacionalmente intratables. El método $L-shaped$ presentado en el Capítulo 1 se ha adaptado a problemas enteros [LL93] o a $SVRP$ con demandas estocásticas [SGL17].

Como se puede consultar en [EVR09], la literatura sobre los problemas de rutas de vehículos está creciendo exponencialmente a una tasa anual del 6.09 %. Esto conduce a una mejora en la metodología de recogida de datos, así como, en los métodos de resolución. Por tanto, esto ha provocado que los $VRP$ pasen de ser problemas estáticos a casos más dinámicos, incorporando datos en tiempo real y parámetros estocásticos.

La importancia de los problemas de rutas de vehículos estocásticos se debe a que en los modelos clásicos $VRP$, generalmente, no se considera un aspecto importante del transporte real y los problemas de distribución logística. Esto viene dado por el hecho de que varios parámetros del modelo, tales como, demanda, tiempo, distancia, entre otros, son estocásticos por naturaleza pero a menudo son simplificados y tratados como deterministas, [AE07].

En [BBKA14] podemos ver que las variantes más comunes de $SVRP$ son $VRP$ con demandas estocásticas de clientes, clientes estocásticos, cliente y demanda estocástica y el tiempo de servicio o viaje estocástico.

Y son principalmente aplicados a los problemas de rutas del autobús público urbano, recogida de residuos, recolección y entrega de bienes y para el sistema de transporte en condiciones climáticas adversas.

Además, la principal técnica de resolución es a través de algún método heurístico, adaptación de algún método originalmente diseñados para el caso determinista. Esto es debido principalmente, al alto coste computacional que conllevan estos problemas.

Sin embargo, en la literatura sobre $SVRP$ existen muchas áreas sin explorar. Entre ellas, nos encontramos la falta de $SVRP$ que maximiza la utilización de vehículos, número de clientes a los que se prestará servicio o los ingresos obtenidos.

Por otro lado, en la mayoría de modelos, los parámetros estocásticos considerados en el estudio para resolver problemas están tomados del mundo real. Es decir, casi todos hacen una suposición para simplificar y tratar de encajar en un modelo probabilístico un problema de la vida real.

Por último, destacar que el software comercial no está fácilmente disponible para cualquier modelo $VRP$. Esto dificulta poder hacer mejoras en los métodos de solución.

En conclusión, destacamos que el estudio de $SVRP$ es un área de investigación relativamente nueva y de rápido crecimiento que debería ganar importancia con el paso del tiempo. En comparación con el caso determinista, los problemas de rutas de vehículos estocásticos están bastante subdesarrollados. La cual dificulta dar una respuesta a la necesidad de un sistema de orientación en tiempo real de vehículos para proporcionar instrucciones actualizadas a los conductores, [GLS96].

A continuación, trataremos un método heurístico para resolver problemas de rutas de vehículos cuando la demanda es estocástica, $VRPSD$.

## 3.4. VRP con demanda estocástica

Los problemas que trataremos en esta sección son, sin duda, los más estudiados de todos los $SVRP$. En ellos, las demandas de los clientes son variables aleatorias, generalmente consideradas independientes.

Al igual que en el caso determinista, queremos determinar un conjunto fijo de rutas de longitud total esperada mínima, que corresponde a la longitud total del conjunto fijo de rutas más el valor esperado de la distancia extra que podría ser requerida por una realización particular de las variables aleatorias.

Las distancias adicionales se deben al hecho de que la demanda en las rutas puede exceder ocasionalmente la capacidad del vehículo y obligarlo a regresar al depósito antes de continuar su ruta. Esto es lo que denominaremos fallo de ruta, es decir, el retorno del vehículo al depósito en una ruta seleccionada a priori por exceso de demanda.

Aunque pueda parecer que estamos ante un problema estocástico de 2 etapas, cuando la demanda es aleatoria, esta se va conociendo gradualmente a medida que el vehículo completa su ruta. Por tanto, puede considerarse un problema multietapa donde el número de etapas no es necesariamente igual al número de clientes, sino que puede ser mayor y además desconocido a priori.

Antes de continuar,mencionamos posibles políticas de servicio.

- Entrega completa, es decir, si el vehículo llega al cliente, este debe satisfacer la demanda del mismo.

- División de entrega, cuando la demanda puede ser satisfecha por varios vehículos, o por el mismo, en varias partes.

Además, respecto a la demanda podemos considerar los siguientes casos.

- Información completa anticipada, que es cuando la demanda es conocida antes de planificar las rutas, $VRP$.

- Información tardía, que se produce cuando la demanda es conocida cuando el vehículo llega a la ubicación del cliente. Esto puede ocurrir inmediatamente antes o después de completar el servicio.

Los problemas $VRPSD$ que nosotros vamos a tratar, son aquellos en los que la entrega es completa y con información tardía inmediatamente antes de ofrecer el servicio, es por ello, que vamos a suponer que la probabilidad de que la demanda de un único cliente supere la capacidad del vehículo sea igual a 0.

En [DLT89] se obtiene que la política de servicio y la disponibilidad de la demanda determina el número de etapas en los problemas de rutas de vehículos. Si la información es anticipada estamos ante problemas de 1 etapa. Información tardía y entrega completa genera un problema de tantas etapas como clientes. Por último, información tardía y división de entregas puede generar más etapas de los clientes a visitar.

### 3.4.1. Modelado de VRP con demanda estocástica

El problema de rutas de vehículos con demanda estocástica que nosotros estamos tratando puede ser modelado como el de programación con restricciones aleatorias propuesto por [CC59], de la siguiente forma:

$$\begin{aligned} \text{mín} \quad & \sum_k \sum_{i,j} c_{ij} x_{ijk} \\ \text{Sujeto a } & \mathbb{P}\{\sum_{i,j} \boldsymbol{d_i} x_{ijk} \leq C\} \geq 1 - \alpha \\ & k = 1, ..., K \\ & x = [x_{ijk}] \in S_K \end{aligned}$$

donde $c_{ij}$ es el coste de viajar del cliente $i$ al cliente $j$, $x_{ijk} = 1$ si el vehículo $k$, con $k \in \{1, ..., K\}$, viaja desde el cliente $i$ hasta el cliente $j$, y $x_{ijk} = 0$ en otro caso. Tenemos que $\boldsymbol{d_i}$ representa la demanda aleatoria del cliente $i$ y $C$ la capacidad máxima del vehículo. Mientras que consideramos $\alpha$ como la probabilidad máxima de que la ruta falle.

Por último, destacar que $S_K$ es el conjunto de todas las soluciones factibles de los $K$ "agentes viajeros", problema $(K-TSP)$. El objetivo del problema $K-TSP$ es construir exactamente $K$ rutas, una para cada vehículo, de modo que cada cliente sea visitado una única vez por uno de los vehículos. Además, cada ruta debe comenzar y finalizar en el depósito y contener a lo sumo $p$ clientes. Este modelo puede ser visto con más detalle en [MTZ60].

También existe la posibilidad de modelar el problema como un modelo de programación estocástica con recurso en el cual existe una penalización para cuando falla una restricción. En [DT86] podemos ver dos tipos de estos modelos.

En el primer modelo, la penalización asociada con el no cumplimiento de una restricción es independiente del grado de incumplimiento, correspondiente al número de unidades por las cuales la demanda excede la capacidad del vehículo en esa ruta. Este modelo es de la forma:

$$\begin{aligned} \text{mín} \quad & \sum_k \sum_{i,j} c_{ij} x_{ijk} + \sum_k \lambda_k \mathbb{P}\{\sum_{i,j} \boldsymbol{d_i} x_{ijk} > C\} \\ \text{Sujeto a} \quad & x = [x_{ijk}] \in S_K \end{aligned}$$

donde $\lambda_k$ representa la penalización fija incurrida cada vez que la ruta $k$ falla.

El segundo modelo tiene una penalización proporcional al grado de incumplimiento de la restricción. El modelo es de la forma:

$$\begin{aligned} \text{mín} \quad & \sum_k \sum_{i,j} c_{ij} x_{ijk} + \sum_k \xi_k \mathbb{E}\left[l_k\right] \\ \text{Sujeto a} \quad & x = [x_{ijk}] \in S_K \end{aligned}$$

donde $\xi_k$ es la penalización por unidad de demanda que excede la capacidad $C$ del vehículo en la ruta $k$. Se tiene además que $\mathbb{E}\left[l_k\right] = \displaystyle\int_0^\infty l_k f(\sum_{i,j} \boldsymbol{d_i} x_{ijk} - C = l_k) dl_k$ es la esperanza del número de unidades de demanda que excede la capacidad total en la ruta $k$, donde $f$ es la función de densidad de la variable $l_k$, unidades en exceso de la ruta $k$.

Como podemos ver en ambos modelos, la función objetivo esta formada por dos términos. Por una parte, dada una ruta, las variables $x_{ijk}$ quedan especificadas por lo que inmediatamente tenemos calculado el primer término de la función objetivo. Mientas que, el segundo término expresa el coste esperado por cometer una infracción en la ruta. Debemos tener en cuenta, que está implícito en el objetivo que visitamos todos los clientes asignados a la ruta, pero en el caso de que el vehículo se llene este debe regresar al depósito, lo que requiere una modificación de la función objetivo, es decir, la visita de todos los clientes solo es posible si la probabilidad de fracaso hasta el penúltimo cliente es despreciable en comparación con la localización del último cliente.

### 3.4.2. Propiedades de las rutas con fallos

Consideremos una ruta $r$, de la forma $(0, 1, ..., n, 0)$ donde 0 es el depósito y 1 hasta $n$ son los clientes visitados. Haremos referencia al cliente $i$ de la ruta $r$ de la forma $r_i$, y denotaremos por $\mathbb{P}\{r_i\}$ la probabilidad de que la ruta falle en o antes del cliente $i$, es decir, que la demanda de los clientes visitados supere la capacidad del vehículo. Nótese que la probabilidad de fallo de una ruta es independiente de la capacidad del vehículo.

En [DT86] podemos ver las siguientes propiedades relacionadas con la probabilidad del fallo de ruta.

**Lema 1** *Si las demandas de los clientes son variables aleatorias independientes con media no negativa, entonces para cualquier ruta $r$, $\{\mathbb{P}(r_i)\}_{i=1}^{n}$ es una sucesión no decreciente.*

Antes de continuar, consideremos una sucesión finita o infinita de elementos $x_i$, con $i \in I$ tal que $I$ es el conjunto de elementos de la sucesión. Decimos que $\{x_i\}$, con $i \in I$, es convexa si para todo $i \in I$ se tiene que $x_{i-1} - x_i \geq x_i - x_{i+1}$, además, diremos que es estrictamente convexa si $x_{i-1} - x_i > x_i - x_{i+1}$ para todo $i \in I$.

Se tiene el siguiente resultado:

**Lema 2** *Si las demandas de los clientes de una ruta $r$ son independientes, idénticamente distribuidas con distribución normal de media $\mu$ y varianza $\sigma^2$, tal que, $\sigma \leq \mu$ y $\mu > 0$, y se tiene que $\mathbb{E}\left[\sum_{i=1}^{n} \boldsymbol{d_i}\right] + \epsilon \leq C$ para algún $\epsilon > 0$, entonces la sucesión $\{\mathbb{P}(r_i)\}_{i=1}^{n}$ es estrictamente convexa.*

**Ejemplo**

Ahora, veamos un ejemplo en el cual calcularemos la probabilidad de fallo de ruta en o hasta el cliente $i-$ésimo de una ruta. Para ello vamos a considerar una ruta formada por 20 clientes y un vehículo de capacidad $C = \frac{1}{2}$. La demanda de todos ellos son independientes e idénticamente distribuidas con distribución normal, de tal forma que:

$$d_i : N\left(\frac{C}{20}, \left(\frac{C}{40}\right)^2\right)$$

| Número clientes | $\mathbb{P}\{r_i\}$ | Número clientes | $\mathbb{P}\{r_i\}$ |
|---|---|---|---|
| 9 | 0.0000 | 15 | 0.0049 |
| 10 | 0.0000 | 16 | 0.0228 |
| 11 | 0.0000 | 17 | 0.0721 |
| 12 | 0.0000 | 18 | 0.1736 |
| 13 | 0.0001 | 19 | 0.3228 |
| 14 | 0.0007 | 20 | 0.5000 |

Cuadro 3.4: Probabilidades para el fallo de una ruta.

En el Cuadro 3.4 se ilustran las probabilidades de que falle una ruta formada por un número concreto de clientes, antes de llegar al último cliente. Por ejemplo, la probabilidad de que en una ruta

formada por 20 clientes, no visitemos el último cliente es de 0.3228, es decir, el coste de ida entre el penúltimo cliente y el último no ocurrirá en un 32.28 % de los casos.

### 3.4.3. Heurística para VRP con demanda estocástica

La heurística que nosotros vamos a tratar es propuesta por [DLT89]. Los autores proponen, que en caso de fallo de ruta se toma el valor del viaje de ida y vuelta al depósito desde cada uno de los clientes restantes de la ruta inicial. Este tipo de política elimina la opción de reoptimización, limitando el número de fallos a como máximo uno por ruta, y así, obligando al vehículo a ser vaciado en el punto de fallo.

Esta suposición, aunque no es la más realista, permite una comparación simple del coste esperado en el peor de los casos de proporcionar servicios de emergencia a través de entregas individuales.

Este método de solución se basa en el algoritmo de ahorros para $VRP$ determinista, [CW64]. En esta nueva heurística el término de ahorros es sustituido por $S_{ij} = c_i^{'} + c_j^{'} - c_{ij}^{'}$. Aquí, $c_i^{'}$ es el coste esperado de la ruta que contenga al cliente $i$, donde $i \in V\backslash\{0\}$, y $c_{ij}^{'}$ es el coste esperado de la ruta combinada donde $i$ precede inmediatamente a $j$, donde $i, j \in V\backslash\{0\}$.

Calculamos el coste esperado de una ruta, $(0, 1, ..., n, 0)$, de la siguiente forma:

$$c_{01} + \sum_{j=1}^{n-1} \left( \left[ \mathbb{P}\{\sum_{k=1}^{j} \boldsymbol{d_k} \leq C, \sum_{z=1}^{j+1} \boldsymbol{d_z} > C\} \right] \cdot \left[ \sum_{p=1}^{j} c_{pp+1} + c_{j+10} + 2 \cdot \left( \sum_{s=j+1}^{n} c_{0j} \right) \right] \right) +$$

$$+\mathbb{P}\{\sum_{k=1}^{n} \boldsymbol{d_k} \leq C\} \cdot \left[ \sum_{p=1}^{n-1} c_{pp+1} + c_{n0} \right]$$

donde $C$ es la carga máxima del vehículo y $\boldsymbol{d_i}$ es la demanda aleatoria del cliente $i \in V\backslash\{0\}$. Además

$$\mathbb{P}\{\sum_{k=1}^{j} \boldsymbol{d_k} \leq C, \sum_{z=1}^{j+1} \boldsymbol{d_z} > C\} = \mathbb{P}\{\sum_{z=1}^{j+1} \boldsymbol{d_z} > C\} - \mathbb{P}\{\sum_{z=1}^{j} \boldsymbol{d_z} > C\}.$$

**Ejemplo**

Veamos ahora un ejemplo para ilustrar el algoritmo. Para ello, vamos a considerar un depósito, 0, y cinco clientes, $\{1, 2, 3, 4, 5\}$. El depósito está situado en la coordenada $(0, 0)$ y los clientes en $(5, 0), (5, 5), (0, 5), (-5, 5)$ y $(-5, 0)$. Las distancias serán tomadas en linea recta. Ahora consideremos que los cliente $1, 2, 3$ y $4$ tienen demanda $0.15C$ y el cliente 5, situado en $(-5, 0)$, demanda $0.4C$ siendo $C$ la capacidad del vehículo.

Consideremos que la ruta $C^R$ es aquella que empieza en el depósito, pasa por $(5, 0), (5, 5), (0, 5), (-5, 5)$ y $(-5, 0)$ para ya por último volver al depósito. Denotamos por $C^L$ la ruta en sentido opuesto. Ambas rutas tienen una distancia de 30 unidades.

Ahora tomemos las demandas de los clientes como variables aleatorias normales independientes con media la demanda dada y coeficiente de variación $\frac{\sigma}{\mu} = 0.5$. Entonces el coste esperado para $C^R$ es de la forma,

$$c_{01} + \mathbb{P}\{d_1 \leq C, d_1 + d_2 > C\} \cdot [c_{12} + c_{20} + 2(c_{02} + c_{03} + c_{04} + c_{05})] +$$

$$+\mathbb{P}\{d_1 + d_2 \leq C, d_1 + d_2 + d_3 > C\} \cdot [c_{12} + c_{23} + c_{30} + 2(c_{03} + c_{04} + c_{05})] +$$

$$+\mathbb{P}\{d_1 + d_2 + d_3 \leq C, d_1 + d_2 + d_3 + d_4 > C\} \cdot [c_{12} + c_{23} + c_{34} + c_{40} + 2(c_{04} + c_{05})] +$$

$$+\mathbb{P}\{d_1 + d_2 + d_3 + d_4 \leq C, d_1 + d_2 + d_3 + d_4 + d_5 > C\} \cdot [c_{12} + c_{23} + c_{34} + c_{45} + c_{50} + 2c_{05}] +$$

$$+\mathbb{P}\{d_1 + d_2 + d_3 + d_4 + d_5 \leq C\} \cdot [c_{12} + c_{23} + c_{34} + c_{45} + c_{50}]$$

de igual manera calculamos el coste esperado de la ruta $C^L$. De esta forma obtenemos que el coste esperado de la ruta $C^R$ es igual a 40.5563 y 48.8362 para $C^L$.

Como se ha podido observar, a diferencia del caso determinista, $S_{ij} \neq S_{ji}$. Es decir, en un contexto estocástico, la selección de ahorros depende de la dirección del viaje. Y dado que las demandas no son conocidas hasta ser alcanzado el cliente, las orientaciones deben ser seleccionadas a priori.

A continuación, la descripción algorítmica del método puede verse en el Cuadro 3.5.

***Algoritmo estocástico de Clarke y Wright***

INICIALIZACIÓN
Tomar $W$ conjunto de todos los posibles pares $(i, j)$ tal que $i, j \in V\backslash\{0\}$. Para cada cliente $i \in V\backslash\{0\}$ crear la ruta $(0, i, 0)$.

PASO 1
Calcular los ahorros $S_{ij} = c_i^{'} + c_j^{'} - c_{ij}^{'}$ para todo par $(i, j) \subset W$.

PASO 2
Ordenar los ahorros de forma no decreciente.

PASO 3
Considerar $S_{\hat{i}\hat{j}} = \max S_{ij}$. Si $S_{\hat{i}\hat{j}} > 0$ y considerando $r_{\hat{i}}$ y $r_{\hat{j}}$, las rutas que contienen a $\hat{i}$ y $\hat{j}$, si en $r_{\hat{i}}$, $\hat{i}$ es el último cliente y en $r_{\hat{j}}$, $\hat{j}$ es el primer cliente y además, las demandas de dichos clientes no superan la capacidad del vehículo, la ruta es factible, eliminar los arcos $(i, 0)$ y $(j, 0)$ e implementar la ruta $(i, j)$. Eliminar los ahorros $S_{\hat{i}\hat{j}} = S_{\hat{j}\hat{i}}$. Sea $W = W\backslash\{(\hat{i}, \hat{j}), (\hat{j}, \hat{i})\}$.

PASO 4
Repetir PASO $1, 2$ y 3 hasta que no existan más fusiones posibles.

Cuadro 3.5: Esquema del algoritmo estocástico de Clarke y Wright

# Capítulo 4

# Una aplicación

En este capítulo, nuestro interés es hacer uso de todo lo tratado en los anteriores capítulos. Para ello, vamos a considerar el problema de minimizar los costes de transporte, al optimizar las rutas de los vehículos utilizados por una cooperativa agrícola que distribuye piensos entre sus socios.

## 4.1. Descripción del problema

Esta aplicación está motivada por un problema de carácter real, asociado a una empresa que se dedica a la fabricación y distribución de pienso. *AIRA* es una cooperativa agrícola ubicada en *Taboada*, un localidad gallega situada a medio camino entre las ciudades de *Lugo* y *Ourense*, en el noroeste de España. Produce cuatro tipos diferentes de alimentos que se distribuyen a sus clientes. La compañía tiene aproximadamente 1500 agricultores (clientes) dispersos en 60 municipios adyacentes, cubriendo una amplia área de la zona.

Los agricultores realizan pedidos para distintos tipos de alimentos, que pueden ser urgentes o tienen un plazo de entrega específico. Para pedidos urgentes, el plazo se reduce a un solo día, y tales órdenes suponen un precio más alto para el cliente. Los agricultores suelen emitir órdenes una o dos veces al mes y, en condiciones normales, sólo requieren un tipo de alimento.

La distribución de los bienes es realizada por una serie de camiones que se encuentran disponibles en la cooperativa, y cada conductor es pagado de acuerdo con la distancia recorrida y la carga. Los camiones tienen diferentes tamaños y los requisitos legales limitan la masa transportable y la distancia recorrida cada día. Los vehículos están compartimentados en varias tolvas (tres a cinco), cada una de las cuales tiene diferentes capacidades. Además, por razones técnicas, cada contenedor puede contener sólo un tipo de alimento y no puede ser compartido entre los agricultores. Además, dado el tamaño y las características de cada camión, algunos camiones no pueden transitar por ciertas carreteras o acceder a ciertas granjas, por lo que para cada vehículo hay una lista de clientes que no pueden ser atendidos.

En cuanto a los datos utilizados, conocemos las capacidades de cada contenedor, las demandas de diferentes clientes y las urgencias de los pedidos, así como, las distancias desde la cooperativa a los clientes y entre los mismos. También sabemos si un camión puede acceder a cada granja y su carga máxima. Además, utilizamos la información sobre el precio de cada descarga y el coste variable de transportar una cierta cantidad de alimentos a una distancia determinada.

## 4.2. Resultados en el caso determinista

En esta sección, vamos a optimizar las rutas de los vehículos utilizadas cuando las demandas son conocidas. Pero antes de continuar, haremos unas pruebas con el algoritmo de ahorros en el caso determinista programado en *R*, ver Apéndice B.

Para trabajar con más comodidad, hemos realizado una interfaz gráfica a través de *Shiny Dashboard*, que es un procedimiento posible para trabajar con *Shiny*, un paquete de *R* para crear webs interactivas [CCA+15], a través de diferentes paneles interactivos.

En la Figura 4.1 podemos ver los diferentes elementos que conforman la interfaz en el contexto determinista. En la misma, encontramos un panel interactivo para simular problemas de rutas de vehículos aleatorios con único elemento de entrada el número de clientes. Además, podemos resolver un problema propio indicando directamente la capacidad del vehículo, y añadiendo luego dos archivos de texto con las demandas y distancias entre clientes.

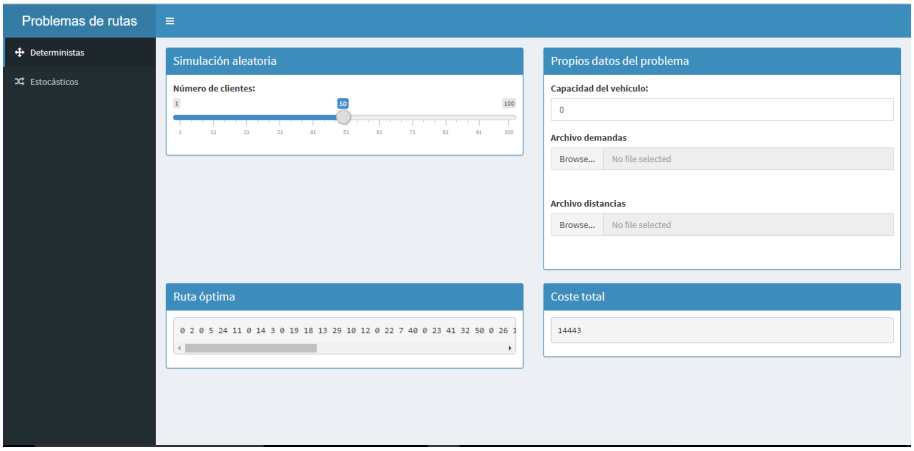


Figura 4.1: Interfaz gráfica para $VRP$ en el caso determinista.

Para hacer uso de estas simulaciones, en el Cuadro 4.2 podemos ver las demandas asociadas a 100 clientes generadas aleatoriamente. Ahora, con estos datos vamos a resolver problemas de rutas de vehículos. Los resultados pueden ser vistos en el Cuadro 4.1. Para ello, la capacidad del vehículo se ha tomado como el doble de la media de la demanda de los clientes utilizados. Las distancias entre depósito y clientes, y entre ellos mismos, han sido generadas de manera aleatoria.

Como podemos apreciar el tiempo de computación de nuestro algoritmo crece de manera exponencial según aumenta el número de clientes utilizados. Esto puede ser visto de igual manera en la Figura 4.2.

| Número clientes | Capacidad vehículo | Coste total | Tiempo computación |
|---|---|---|---|
| 10 | 102 | 920 | 0.02 segundos |
| 25 | 104 | 1194 | 0.14 segundos |
| 50 | 106 | 3602 | 2.12 segundos |
| 75 | 102 | 4901 | 11.65 segundos |
| 100 | 104 | 6615 | 39.11 segundos |

Cuadro 4.1: Resultados de los problemas de rutas.

| Cliente | Demanda | Cliente | Demanda | Cliente | Demanda | Cliente | Demanda |
|---|---|---|---|---|---|---|---|
| 1 | 27.00 | 26 | 39.00 | 51 | 48.00 | 76 | 90.00 |
| 2 | 38.00 | 27 | 2.00 | 52 | 87.00 | 77 | 87.00 |
| 3 | 58.00 | 28 | 39.00 | 53 | 44.00 | 78 | 39.00 |
| 4 | 91.00 | 29 | 87.00 | 54 | 25.00 | 79 | 78.00 |
| 5 | 21.00 | 30 | 35.00 | 55 | 8.00 | 80 | 97.00 |
| 6 | 90.00 | 31 | 49.00 | 56 | 10.00 | 81 | 44.00 |
| 7 | 95.00 | 32 | 60.00 | 57 | 32.00 | 82 | 72.00 |
| 8 | 67.00 | 33 | 50.00 | 58 | 52.00 | 83 | 40.00 |
| 9 | 63.00 | 34 | 19.00 | 59 | 67.00 | 84 | 33.00 |
| 10 | 7.00 | 35 | 83.00 | 60 | 41.00 | 85 | 76.00 |
| 11 | 21.00 | 36 | 67.00 | 61 | 92.00 | 86 | 21.00 |
| 12 | 18.00 | 37 | 80.00 | 62 | 30.00 | 87 | 72.00 |
| 13 | 69.00 | 38 | 11.00 | 63 | 46.00 | 88 | 13.00 |
| 14 | 39.00 | 39 | 73.00 | 64 | 34.00 | 89 | 25.00 |
| 15 | 77.00 | 40 | 42.00 | 65 | 66.00 | 90 | 15.00 |
| 16 | 50.00 | 41 | 83.00 | 66 | 26.00 | 91 | 24.00 |
| 17 | 72.00 | 42 | 65.00 | 67 | 48.00 | 92 | 6.00 |
| 18 | 100.00 | 43 | 79.00 | 68 | 77.00 | 93 | 65.00 |
| 19 | 39.00 | 44 | 56.00 | 69 | 9.00 | 94 | 88.00 |
| 20 | 78.00 | 45 | 53.00 | 70 | 88.00 | 95 | 78.00 |
| 21 | 94.00 | 46 | 79.00 | 71 | 34.00 | 96 | 80.00 |
| 22 | 22.00 | 47 | 3.00 | 72 | 84.00 | 97 | 46.00 |
| 23 | 66.00 | 48 | 48.00 | 73 | 35.00 | 98 | 42.00 |
| 24 | 13.00 | 49 | 74.00 | 74 | 34.00 | 99 | 82.00 |
| 25 | 27.00 | 50 | 70.00 | 75 | 48.00 | 100 | 61.00 |

Cuadro 4.2: Demandas para los 100 clientes.

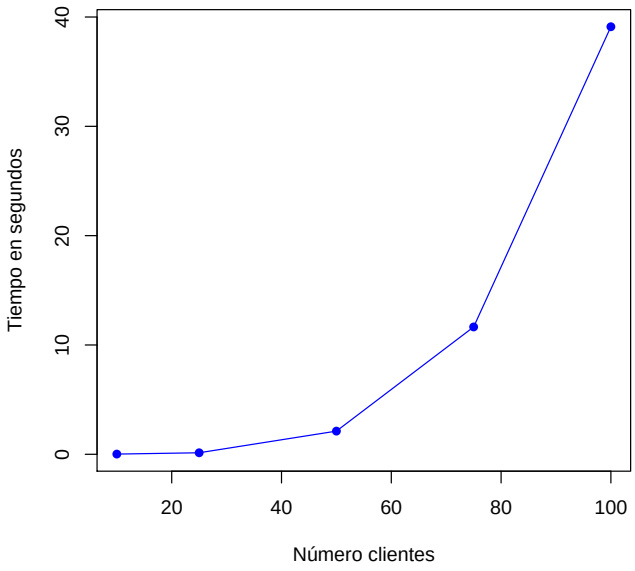


Figura 4.2: Representación del tiempo de computación.

Además podemos resolver el ejemplo de la Sección 3.2.1. En el, tenemos cuatro clientes con demandas asociadas $(2, 2, 1, 2)$ y matriz de distancias el Cuadro 3.3. Si la capacidad del vehículo es de 10 unidades, entonces obtenemos la siguiente ruta óptima:

$$(0, D, C, B, A, 0)$$

con un coste total de 17 unidades de distancia. Obteniendo el mismo resultado que habríamos calculado anteriormente pero tomando la ruta en sentido opuesto.

Ahora continuamos con los datos del problema real. Estos se dividen en dos días de diferentes pedidos. Además vamos a considerar exclusivamente, para evitar restricciones adicionales, la capacidad máxima de masa transportable, permitiendo a los agricultores el uso compartido de las tolvas. Se toma esa carga máxima como $C = 15300$ kilogramos.

En el primer día, 19 clientes hacen un pedido de un único tipo de alimento. Las demandas en kilogramos de estos 19 clientes son, respectivamente,

$$3300, 6041, 5959, 2951, 4885, 3003, 3016, 4478, 5413, 3490$$

$$5283, 15078, 10017, 10176, 3954, 10013, 4526, 5119, 5000.$$

Tales demandas pueden ser vistas en la Figura 4.3.

En el Cuadro 4.3 podemos ver las distancias entre el depósito, 0, y clientes, así como, entre los clientes.

Una vez resuelto nuestro problema, con el algoritmo creado, en el Cuadro 4.4 podemos ver los resultados obtenidos, donde obtenemos un coste total de 354 (que es la distancia total recorrida). Además, las rutas óptimas escogidas son

$$(0, 3, 2, 1, 0, 9, 8, 7, 0, 10, 5, 4, 6, 0, 12, 0, 13, 11, 0, 15, 0, 17, 16, 0, 18, 14, 0, 19, 0).$$

Podemos ver la representación en la Figura 4.4.

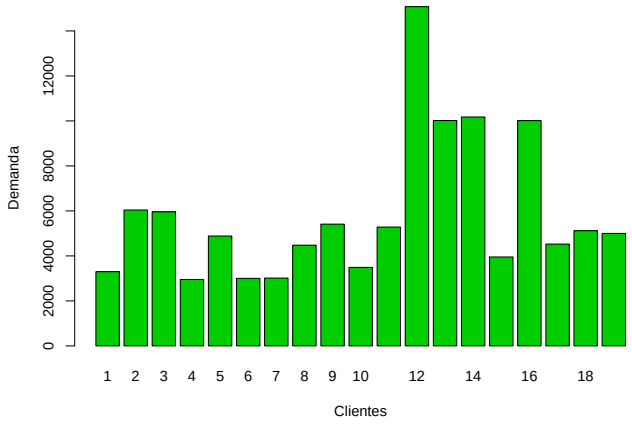


Figura 4.3: Demandas asociadas a los clientes del primer día.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 21 | 20 | 17 | 65 | 63 | 60 | 19 | 22 | 24 | 60 | 5 | 29 | 6 | 4 | 7 | 8 | 6 | 4 | 0 |
| 1 | 21 | 0 | 4 | 6 | 60 | 58 | 55 | 15 | 18 | 20 | 55 | 26 | 39 | 27 | 25 | 21 | 19 | 20 | 24 | 21 |
| 2 | 20 | 4 | 0 | 4 | 59 | 56 | 53 | 13 | 8 | 12 | 53 | 24 | 37 | 26 | 23 | 19 | 17 | 18 | 22 | 19 |
| 3 | 17 | 6 | 4 | 0 | 57 | 54 | 52 | 11 | 13 | 16 | 52 | 21 | 35 | 23 | 21 | 18 | 15 | 16 | 20 | 17 |
| 4 | 65 | 60 | 59 | 57 | 0 | 3 | 7 | 66 | 69 | 71 | 6 | 66 | 90 | 61 | 66 | 62 | 60 | 62 | 69 | 65 |
| 5 | 63 | 58 | 56 | 54 | 3 | 0 | 4 | 64 | 66 | 69 | 3 | 64 | 88 | 58 | 63 | 60 | 57 | 59 | 66 | 63 |
| 6 | 60 | 55 | 53 | 52 | 7 | 4 | 0 | 61 | 64 | 66 | 2 | 61 | 85 | 56 | 60 | 57 | 55 | 57 | 63 | 60 |
| 7 | 19 | 15 | 13 | 11 | 66 | 64 | 61 | 0 | 3 | 5 | 61 | 24 | 33 | 25 | 23 | 24 | 25 | 23 | 22 | 19 |
| 8 | 22 | 18 | 8 | 13 | 69 | 66 | 64 | 3 | 0 | 7 | 64 | 26 | 35 | 28 | 26 | 27 | 28 | 26 | 25 | 22 |
| 9 | 24 | 20 | 12 | 16 | 71 | 69 | 66 | 5 | 7 | 0 | 66 | 29 | 38 | 30 | 28 | 29 | 30 | 28 | 27 | 24 |
| 10 | 60 | 55 | 53 | 52 | 6 | 3 | 2 | 61 | 64 | 66 | 0 | 61 | 85 | 56 | 60 | 57 | 55 | 57 | 63 | 60 |
| 11 | 5 | 26 | 24 | 21 | 66 | 64 | 61 | 24 | 26 | 29 | 61 | 0 | 33 | 7 | 1 | 12 | 5 | 4 | 8 | 5 |
| 12 | 29 | 39 | 37 | 35 | 90 | 88 | 85 | 33 | 35 | 38 | 85 | 33 | 0 | 35 | 33 | 35 | 36 | 34 | 30 | 29 |
| 13 | 6 | 27 | 26 | 23 | 61 | 58 | 56 | 25 | 28 | 30 | 56 | 7 | 35 | 0 | 7 | 13 | 14 | 12 | 10 | 6 |
| 14 | 4 | 25 | 23 | 21 | 66 | 63 | 60 | 23 | 26 | 28 | 60 | 1 | 32 | 7 | 0 | 11 | 5 | 3 | 7 | 4 |
| 15 | 7 | 21 | 19 | 18 | 62 | 60 | 57 | 24 | 27 | 29 | 57 | 12 | 35 | 13 | 11 | 0 | 8 | 8 | 10 | 7 |
| 16 | 8 | 19 | 17 | 15 | 60 | 57 | 55 | 25 | 28 | 30 | 55 | 5 | 36 | 14 | 5 | 8 | 0 | 3 | 11 | 8 |
| 17 | 6 | 20 | 18 | 16 | 62 | 59 | 57 | 23 | 26 | 28 | 57 | 4 | 34 | 12 | 3 | 8 | 3 | 0 | 9 | 6 |
| 18 | 4 | 24 | 22 | 20 | 69 | 66 | 63 | 22 | 25 | 27 | 63 | 8 | 30 | 10 | 7 | 10 | 11 | 9 | 0 | 3 |
| 19 | 0 | 21 | 19 | 17 | 65 | 63 | 60 | 19 | 22 | 24 | 60 | 5 | 28 | 6 | 4 | 7 | 8 | 6 | 3 | 0 |

Cuadro 4.3: Distancias entre clientes y depósito (primer día de planificación).

A mayores, en el Cuadro 4.4 podemos ver los resultados para diferentes capacidades del vehículo. Como podemos apreciar, a mayor carga máxima menor distancia total recorrida.

Además, las rutas óptimas escogidas para una carga máxima de $C = 18000$ son

$$(0, 3, 2, 1, 0, 10, 5, 4, 6, 0, 12, 0, 13, 0, 14, 11, 0, 15, 9, 8, 7, 0, 17, 16, 0, 19, 18, 0)$$

y para $C = 20000$,

$$(0, 3, 2, 1, 0, 12, 0, 13, 0, 14, 11, 0, 15, 10, 5, 4, 6, 0, 17, 16, 0, 18, 9, 8, 7, 0, 19, 0).$$

| Número clientes | Capacidad vehículo | Coste total | Tiempo computación |
|---|---|---|---|
| 19 | 15300 | 354 | 0.08 segundos |
| 19 | 18000 | 348 | 0.09 segundos |
| 19 | 20000 | 340 | 0.07 segundos |

Cuadro 4.4: Resultados del primer día.

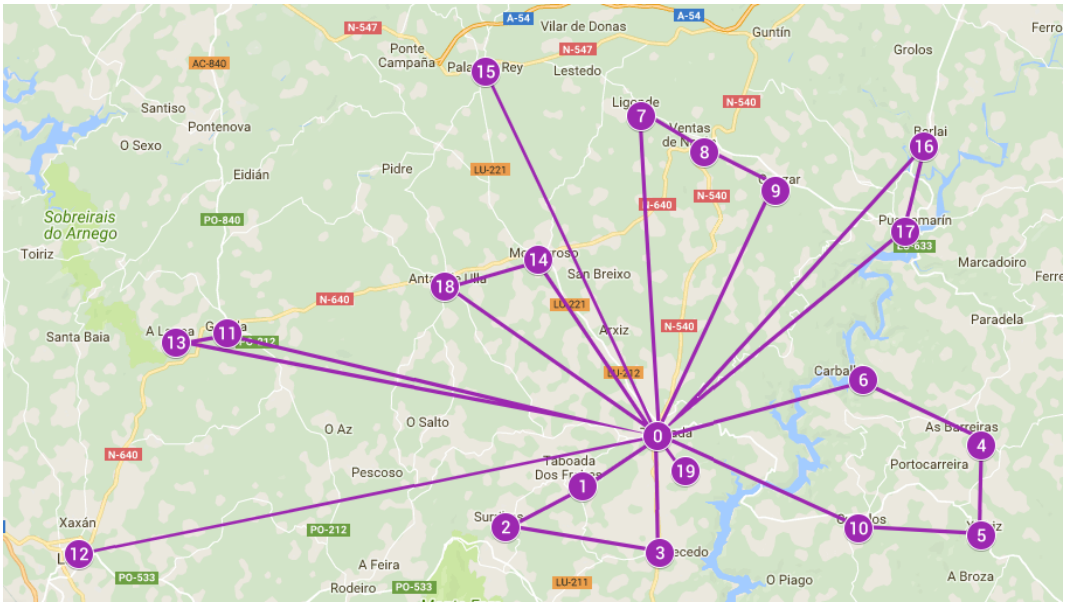


Figura 4.4: Representación de las rutas óptimas para el primer día en el contexto determinista.

En el segundo día, 17 clientes hacen un pedido de un único tipo de alimento. Las demandas de estos clientes son, respectivamente,

$$4523, 5087, 3017, 4981, 4507, 3612, 4551, 3492,$$

$$3645, 10200, 10055, 10091, 5200, 5100, 11005, 1998, 2501.$$

Podemos ver estas demandas representadas en la Figura 4.5.

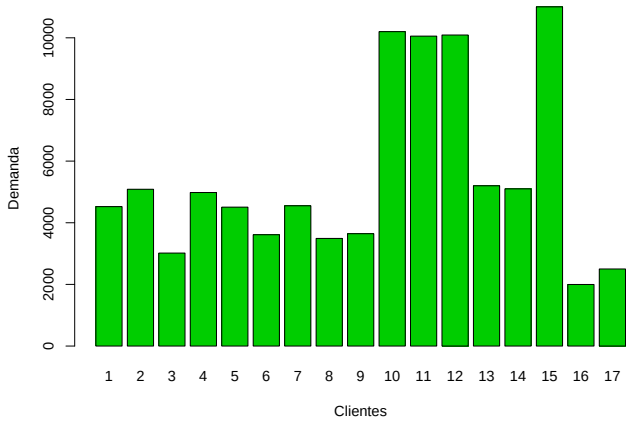


Figura 4.5: Demandas asociadas a los clientes del segundo día.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 24 | 27 | 23 | 5 | 21 | 11 | 55 | 12 | 10 | 3 | 25 | 9 | 7 | 4 | 8 | 25 | 41 |
| 1 | 24 | 0 | 6 | 2 | 23 | 6 | 15 | 53 | 17 | 23 | 27 | 39 | 32 | 24 | 26 | 18 | 39 | 50 |
| 2 | 27 | 6 | 0 | 4 | 26 | 9 | 18 | 47 | 20 | 26 | 30 | 42 | 35 | 27 | 29 | 21 | 42 | 53 |
| 3 | 23 | 2 | 4 | 0 | 21 | 4 | 14 | 51 | 15 | 22 | 26 | 38 | 30 | 22 | 25 | 17 | 37 | 48 |
| 4 | 5 | 23 | 26 | 21 | 0 | 20 | 14 | 60 | 11 | 13 | 8 | 30 | 13 | 10 | 7 | 7 | 29 | 46 |
| 5 | 21 | 6 | 9 | 4 | 20 | 0 | 9 | 46 | 12 | 17 | 24 | 36 | 29 | 17 | 17 | 16 | 36 | 47 |
| 6 | 11 | 15 | 18 | 14 | 14 | 9 | 0 | 43 | 4 | 9 | 14 | 39 | 18 | 9 | 9 | 18 | 39 | 51 |
| 7 | 55 | 53 | 47 | 51 | 60 | 46 | 43 | 0 | 47 | 52 | 54 | 77 | 63 | 52 | 51 | 56 | 76 | 81 |
| 8 | 12 | 17 | 20 | 15 | 11 | 12 | 4 | 47 | 0 | 9 | 15 | 37 | 20 | 9 | 9 | 7 | 37 | 53 |
| 9 | 10 | 23 | 26 | 22 | 13 | 17 | 9 | 52 | 9 | 0 | 7 | 34 | 17 | 10 | 8 | 16 | 34 | 50 |
| 10 | 3 | 27 | 30 | 26 | 8 | 24 | 14 | 54 | 15 | 7 | 0 | 28 | 11 | 10 | 7 | 11 | 28 | 44 |
| 11 | 25 | 39 | 42 | 38 | 30 | 36 | 39 | 77 | 37 | 34 | 28 | 0 | 22 | 31 | 29 | 32 | 2 | 13 |
| 12 | 9 | 32 | 35 | 30 | 13 | 29 | 18 | 63 | 20 | 17 | 11 | 22 | 0 | 15 | 12 | 16 | 21 | 32 |
| 13 | 7 | 24 | 27 | 22 | 10 | 17 | 9 | 52 | 9 | 10 | 10 | 31 | 15 | 0 | 5 | 13 | 31 | 47 |
| 14 | 4 | 26 | 29 | 25 | 7 | 17 | 9 | 51 | 9 | 8 | 7 | 29 | 12 | 5 | 0 | 10 | 28 | 44 |
| 15 | 8 | 18 | 21 | 17 | 7 | 16 | 18 | 56 | 7 | 16 | 11 | 32 | 16 | 13 | 10 | 0 | 32 | 48 |
| 16 | 25 | 39 | 42 | 37 | 29 | 36 | 39 | 76 | 37 | 34 | 28 | 2 | 21 | 31 | 28 | 32 | 0 | 12 |
| 17 | 41 | 50 | 53 | 48 | 46 | 47 | 51 | 81 | 53 | 50 | 44 | 13 | 32 | 47 | 44 | 48 | 12 | 0 |

Cuadro 4.5: Distancias entre clientes y depósito (segundo día de planificación).

En el Cuadro 4.5 podemos ver las distancias entre el depósito, 0, y clientes, así como, entre los clientes.

Una vez resuelto nuestro problema, con el algoritmo creado, en el Cuadro 4.6 podemos ver los resultados obtenidos, donde obtenemos un coste total de 344, que hace referencia a la distancia total recorrida. Además, las rutas óptimas escogidas son

$$(0, 3, 2, 1, 0, 7, 5, 6, 0, 9, 8, 13, 0, 10, 0, 11, 17, 16, 0, 12, 0, 14, 4, 0, 15, 0).$$

Podemos ver representadas estas rutas en la Figura 4.6.

A mayores, en el Cuadro 4.6 podemos ver los resultados para diferentes capacidades del vehículo. Como podemos apreciar, al aumentar la carga máxima menor distancia total recorrida, aunque obtenemos valores similares para las capacidades 18000 y 20000.

Además, las rutas óptimas escogidas para las carga máxima de $C = 18000$ y $C = 20000$ son

$$(0, 5, 3, 2, 1, 0, 9, 8, 7, 6, 0, 10, 0, 11, 17, 16, 0, 12, 0, 14, 13, 0, 15, 4, 0).$$

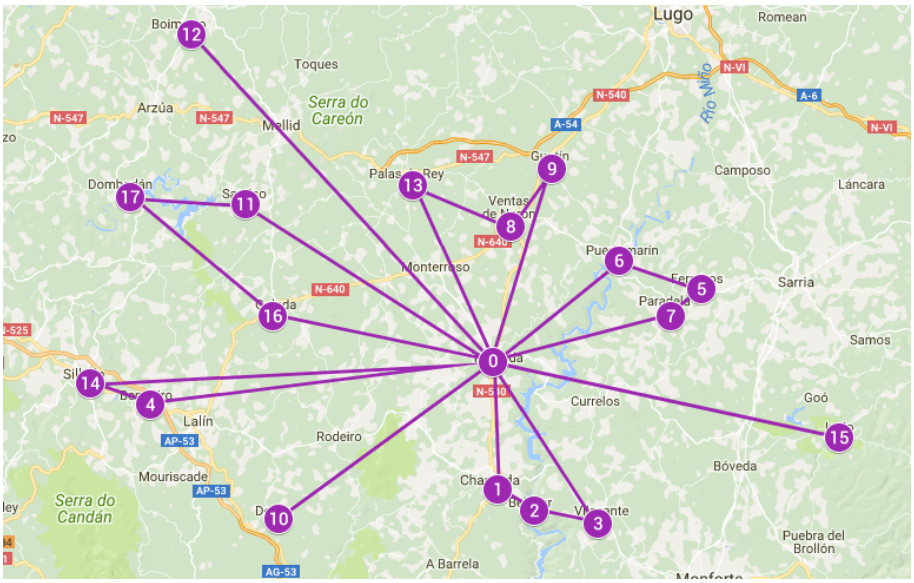


Figura 4.6: Representación de las rutas óptimas para el segundo día en el contexto determinista.

| Número clientes | Capacidad vehículo | Coste total | Tiempo computación |
|---|---|---|---|
| 17 | 15300 | 344 | 0.06 segundos |
| 17 | 18000 | 314 | 0.06 segundos |
| 17 | 2000 | 314 | 0.08 segundos |

Cuadro 4.6: Resultados del segundo día.

A continuación, haremos un estudio de selección de rutas óptimas para la planificación de ambos días en conjunto. En el Cuadro 4.7 podemos ver los resultados obtenidos. A medida que la carga

máxima transportable por vehículo aumenta, la distancia total recorrida disminuye. Además, las rutas óptimas escogidas para $C = 15300$ son

$$(0, 9, 8, 7, 0, 12, 11, 10, 0, 14, 2, 1, 0, 15, 16, 3, 0, 18, 5, 4, 6, 0, 20, 0, 21, 34, 33, 0, 23,$$

$$0, 24, 0, 25, 0, 27, 19, 0, 30, 26, 13, 0, 31, 28, 22, 0, 32, 17, 0, 35, 29, 0, 36, 0)$$

para $C = 18000$,

$$(0, 3, 2, 1, 0, 13, 9, 8, 7, 0, 14, 12, 11, 10, 0, 18, 5, 4, 6, 15, 0, 20, 0, 24, 0, 25, 22, 0,$$

$$27, 19, 0, 30, 26, 17, 16, 0, 31, 28, 0, 32, 0, 33, 21, 0, 34, 23, 0, 35, 29, 0, 36, 0)$$

y para $C = 20000$,

$$(0, 12, 11, 10, 2, 0, 15, 14, 1, 3, 0, 17, 9, 8, 7, 0, 18, 5, 4, 6, 16, 0, 20, 0, 21, 0, 24, 0, 20, 0, 21, 0,$$

$$24, 0, 25, 22, 0, 27, 19, 28, 0, 31, 30, 26, 0, 32, 13, 0, 33, 34, 23, 0, 35, 29, 0, 36, 0).$$

| Número clientes | Capacidad vehículo | Coste total | Tiempo computación |
|---|---|---|---|
| 36 | 15300 | 681 | 0.59 segundos |
| 36 | 18000 | 653 | 0.53 segundos |
| 36 | 2000 | 547 | 0.54 segundos |

Cuadro 4.7: Resultados del primer y segundo día en conjunto.

## 4.3. Resultados en el caso estocástico

A continuación, en esta sección, nuestro interés es optimizar las rutas de los vehículos utilizados cuando las demandas son aleatorias. Para ello, utilizaremos el algoritmo de ahorros en el caso estocástico programado en *R*, ver Apéndice B. Al igual que en el apartado anterior haremos uso de una interfaz gráfica, ver Figura 4.7.

Como podemos observar, disponemos de un panel de entrada para realizar simulaciones aleatorias indicando el número de clientes deseados. En otro caso, podemos resolver nuestros propios problemas de rutas de vehículos añadiendo la capacidad del vehículo y distancias entre clientes. Además, podemos resolverlos según sigan una distribución normal o discreta. En el caso normal, para el archivo demandas subiremos la media de estas y en el archivo probabilidades o desviaciones típicas las diferentes desviaciones típicas. Por último, en el caso discreto, en el archivo demandas subiremos las diferentes demandas que pueden tomar los distintos clientes, y así, en el archivo probabilidades o desviaciones típicas la probabilidad que puede tomar cada demanda.

Antes de utilizar los datos de nuestro ejemplo real, trabajaremos con los datos simulados utilizados en la sección anterior, y haremos simulaciones con los mismos números de clientes pero considerando la demanda como una variable aleatoria, para así, comprobar los diferentes cambios que se obtienen en la resolución de este tipo de problemas.

En esta situación vamos a suponer que las demandas de nuestros clientes son independientes con distribución normal, donde la media son las demandas del Cuadro 4.2. Al igual que en el caso determinista, utilizaremos la misma capacidad del vehículo y distancias entre depósito y clientes, así como entre ellos.

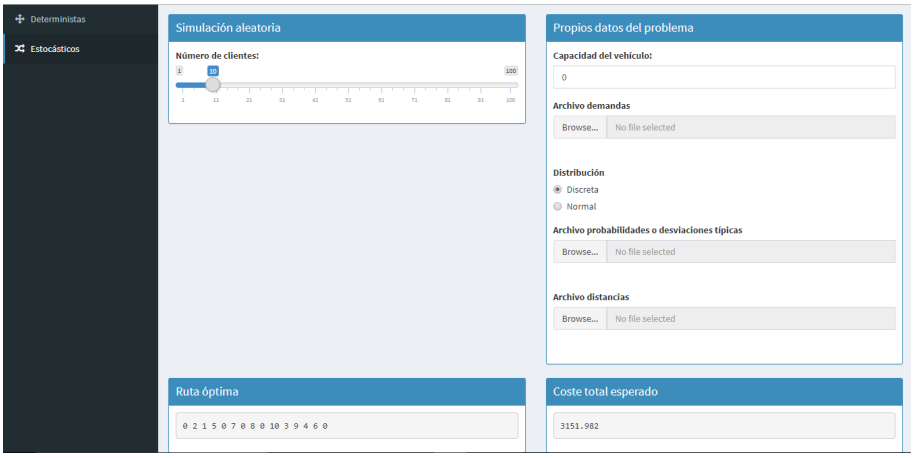


Figura 4.7: Interfaz gráfica para $VRP$ en el caso estocástico.

Los resultados ante este problema pueden ser vistos en el Cuadro 4.8. En el mismo, hemos variado la desviación típica de las diferentes demandas entre los valores $2, 5$ y $10$. Se puede observar que, los costes de computación son más elevados que en el caso determinista, y de igual manera, aumentan de forma exponencial.

Sobre los costes esperados de las rutas escogidas, en comparación con los costes totales del caso determinista, estos no distan demasiado con un número bajo de clientes, es decir, según aumenta el número de clientes el coste esperado de optimización de rutas se hace mayor en comparación al coste real.

Ahora trabajaremos con los datos reales aunque suponiendo que las demandas son aleatorias. Para ello, utilizaremos las distancias y capacidad del vehículo de la anterior sección. Pero supondremos que las demandas son independientes y siguen una distribución normal.

En el primer día, las medias de estas serán las demandas utilizadas en el caso determinista. Sin embargo, trabajaremos con diferentes desviaciones típicas y observaremos el efecto en el resultado final.

Rutas para desviación típica 10:

$$(0, 2, 1, 3, 0, 9, 8, 7, 0, 10, 5, 4, 6, 0, 12, 0, 13, 0, 14, 0, 15, 18, 11, 0, 17, 16, 0, 19, 0)$$

Rutas para desviación típica 50:

$$(0, 2, 1, 3, 0, 9, 8, 7, 0, 10, 5, 4, 6, 0, 12, 0, 13, 0, 14, 0, 15, 18, 11, 0, 16, 17, 0, 19, 0)$$

Rutas para desviación típica 100:

$$(0, 2, 1, 3, 0, 9, 8, 7, 0, 10, 5, 4, 6, 0, 11, 14, 0, 12, 0, 13, 0, 16, 17, 0, 19, 18, 15, 0)$$

Como podemos observar en el Cuadro 4.9 los costes esperados para las diferentes desviaciones típicas no distan mucho entre sí, siendo los resultados similares. Respecto al coste total del caso determinista, en el contexto estocástico el coste esperado es mayor aunque, en el peor de los casos, no supera el $6\,\%$.

Las rutas óptimas para las diferentes desviaciones típicas son prácticamente iguales cuando estas toman los valores 10 y 50, cambiando el sentido de la ruta $(0, 17, 16, 0)$, para 10, por $(0, 16, 17, 0)$, para 50. Respecto al valor 100, aunque tenemos rutas similares ya existen mayores diferencias con respecto a los otros dos valores. Las diferencias más significativas entre las rutas óptimas en el caso determinista y estocástico vienen dadas por el sentido de las mismas, así como, la existencia de selecciones diferentes de clientes en distintas rutas.

| Número clientes | Capacidad vehículo | Desviación típica | Coste esperado total | Tiempo computación |
|---|---|---|---|---|
| 10 | 102 | 2 | 893.4684 | 0.11 segundos |
| | | 5 | 904.8284 | |
| | | 10 | 925.9974 | |
| 25 | 104 | 2 | 1913.994 | 1.28 segundos |
| | | 5 | 1985.387 | |
| | | 10 | 2002.881 | |
| 50 | 106 | 2 | 3754.281 | 11.32 segundos |
| | | 5 | 3901.149 | |
| | | 10 | 3884.235 | |
| 75 | 102 | 2 | 5151.255 | 42.49 segundos |
| | | 5 | 5314.362 | |
| | | 10 | 5457.355 | |
| 100 | 104 | 2 | 6800.154 | 111.08 segundos |
| | | 5 | 7122.145 | |
| | | 10 | 7217.029 | |

Cuadro 4.8: Resultados de los problemas de rutas.

| Número clientes | Capacidad vehículo | Desviación típica | Coste esperado total | Tiempo computación |
|---|---|---|---|---|
| 19 | 15300 | 10 | 375 | 1.53 segundos |
| | | 50 | 375.0001 | |
| | | 100 | 374.2069 | |

Cuadro 4.9: Resultados de los problemas de rutas para el primer día.

| Número clientes | Capacidad vehículo | Desviación típica | Coste esperado total | Tiempo computación |
|---|---|---|---|---|
| 17 | 15300 | 10 | 344 | 1.11 segundos |
| | | 50 | 344 | |
| | | 100 | 344.3224 | |

Cuadro 4.10: Resultados de los problemas de rutas para el segundo día.

En el segundo día, al igual que en el caso anterior, tomaremos como demandas las medias utilizadas en el caso determinista. Trabajaremos también con diferentes desviaciones típicas para ver el efecto en el resultado final.

Rutas para desviación típica 10:

$$(0, 3, 2, 1, 0, 7, 5, 6, 0, 9, 8, 13, 0, 10, 0, 11, 17, 16, 0, 12, 0, 14, 4, 0, 15, 0)$$

Rutas para desviación típica 50:

$$(0, 3, 2, 1, 0, 7, 5, 6, 0, 9, 8, 13, 0, 10, 0, 11, 17, 16, 0, 12, 0, 14, 4, 0, 15, 0)$$

Rutas para desviación típica 100:

$$(0, 3, 2, 1, 0, 7, 5, 6, 0, 9, 8, 13, 0, 10, 0, 11, 17, 16, 0, 12, 0, 14, 4, 0, 15, 0)$$

Como podemos observar en el Cuadro 4.10, para las diferentes desviaciones típicas que hemos tomado el coste esperado total no ha variado. Así mismo, las rutas seleccionadas son las mismas tanto para el contexto determinista como en los 3 casos del contexto estocástico.

Ahora por último haremos una planificación para ambos días en conjunto. En el Cuadro 4.11 podemos ver las diferentes distancias esperadas totales recorridas para cada una de las desviaciones típicas escogidas. No apreciamos grandes diferencias entre los costes esperados totales y el coste total en el caso determinista, siendo este mayor para una desviación típica igual a 100.

| Número clientes | Capacidad vehículo | Desviación típica | Coste esperado total | Tiempo computación |
|---|---|---|---|---|
| 36 | 15300 | 10 | 680 | 10.79 segundos |
| | | 50 | 681.4759 | |
| | | 100 | 684.6841 | |

Cuadro 4.11: Resultados de los problemas de rutas para el primer y segundo día juntos.

Rutas para desviación típica 10:

$$(0, 9, 8, 7, 0, 12, 11, 10, 0, 14, 2, 1, 0, 15, 16, 3, 0, 18, 5, 4, 6, 0, 20, 0, 21, 34, 33, 0,$$

$$23, 0, 24, 0, 25, 0, 27, 19, 36, 0, 29, 35, 0, 30, 26, 13, 0, 31, 28, 22, 0, 32, 17, 0)$$

Rutas para desviación típica 50:

$$(0, 9, 8, 7, 0, 12, 11, 10, 0, 13, 28, 22, 0, 14, 2, 1, 0, 15, 16, 3, 0, 17, 32, 0, 18, 5, 4, 6, 0,$$

$$20, 0, 21, 34, 33, 0, 23, 0, 24, 0, 25, 0, 27, 19, 36, 0, 29, 35, 0, 30, 26, 31, 0)$$

Rutas para desviación típica 100:

$$(0, 2, 1, 14, 0, 9, 8, 7, 0, 12, 11, 10, 0, 13, 28, 22, 0, 15, 16, 3, 0, 17, 32, 0, 18, 5, 4, 6, 0,$$

$$20, 0, 21, 34, 33, 0, 23, 0, 24, 0, 25, 0, 27, 19, 0, 29, 0, 30, 26, 31, 0, 36, 35, 0)$$

Las diferentes rutas óptimas tomadas para las distintas desviaciones típicas son muy similares, a pesar de tener ciertas rutas con una selección diferente de clientes. Respecto a las rutas en el caso determinista existen mayores diferencias, aunque podemos apreciar rutas similares.

# Capítulo 5

# Conclusiones

En este trabajo, hemos tratado un conjunto de técnicas de optimización que dan lugar a una de las muchas aplicaciones que tienen las matemáticas en la industria, más concretamente, en la logística. El avance de la investigación ha provocado que cada vez los problemas de optimización pueden resolver situaciones más próximas a las planteadas en la realidad, llegando a imitarla con pequeños errores no significativos. Aunque para ello, hemos tenido que crear nuevos modelos, aumentando considerablemente el número de restricciones, así como, el coste de computación.

Es por ello, que para solucionar este tipo de problemas los métodos heurísticos son cada vez más habituales, ya que resolverlos de forma exacta puede dar lugar a grandes dificultades. Nosotros hemos presentado un algoritmo para resolver problemas de rutas de vehículos, y su adaptación al contexto estocástico, concretamente, al caso en el que las demandas de los clientes servidos son variables aleatorias.

Aunque existen algoritmos con un diseño más sofisticado, el que nosotros hemos tratado puede llegar a ser de gran utilidad como solución inicial, y a partir del resultado que proporciona, se pueden implementar mejoras, si es el caso, haciendo uso de algún algoritmo de búsqueda local.

Aún considerando las posibles limitaciones, con el algoritmo estudiado hemos comprobado las diferentes soluciones que se obtienen al resolver modelos estocásticos y las hemos comparado con las obtenidas al no considerar la aleatoriedad de las variables demandas. Aún siendo la matriz de distancias simétrica, es decir, el viaje de ida o vuelta supone un mismo coste, nos hemos encontrado que el sentido de la ruta es importante en un contexto estocástico, ya que con el denominado fallo de ruta nuestra política de ida y retorno al depósito provoca un mayor coste en aquellos clientes más alejados del depósito.

Cabe señalar, en cualquier caso, que se obtienen unos resultados alentadores, ya que, al menos, con los datos considerados, la presencia de aleatoriedad no implica un incremento grande del coste respecto a una situación determinista, cuando hacemos uso de los algoritmos estudiados en este trabajo. En el caso tratado, esto se interpreta como que si en lugar de esperar a recibir órdenes exactas por parte de los clientes, se trabajase con predicciones suficientemente fiables, el precio a pagar no sería alto y en cambio, sería ventajoso para la planificación de poder hacerlas con antelación.

Para finalizar, creemos que, a nivel global, si bien los modelos $SVRP$ son cada vez más estudiados, existen cuestiones pendientes por resolver tanto a nivel literatura como software. De esta forma, el interés de nuestro trabajo es aportar una revisión bibliográfica que puede constituir una introducción a investigaciones teóricas futuras o a la resolución de otros problemas logísticos.

# Apéndice A

# Código en lenguaje SMPS

Código para resolver el problema del granjero sin incertidumbre para los rendimientos supuestos por el granjero (segundo escenario):

```
Archivo principal:
NAME          FARMER
ROWS
 N  WEALTH
 L  BUDGET
 G  BAL1
 G  BAL2
 L  BAL3
 L  BAL4
COLUMNS
    X1        BUDGET    1.00           BAL1      +2.5
    X2        BUDGET    1.00           BAL2      +3
    X3        BUDGET    1.00           BAL3      -20
    Y1        BAL1      1.00
    Y2        BAL2      +1.00
    W1        BAL1      -1.00
    W2        BAL2      -1.00
    W3        BAL3      +1.00          BAL4      +1.00
    W4        BAL3     +1.00
    X1        WEALTH    150
    X2        WEALTH    230
    X3        WEALTH    +260
    Y1        WEALTH    +238
    Y2        WEALTH    +210
    W1        WEALTH    -170
    W2        WEALTH    -150
    W3        WEALTH    -36
    W4        WEALTH    -10
RHS
    RHS       BUDGET    500            BAL1      200
RHS       BAL2      240            BAL4      6000
ENDATA

Archivo de tiempo:
TIME          FARMER
```

```
PERIODS
    X1        BUDGET                  TODAY
    Y1        BAL1                    TOMORROW
ENDATA

Archivo estocástico:
STOCH         FARMER
BLOCKS        DISCRETE
 BL BLOCK1    TOMORROW  1
    X1        BAL1      2.5
    X2        BAL2      3
    X3        BAL3      -20
ENDATA
```

Aunque añadimos el archivo estocástico, en el mismo, solo ocurre un suceso.

Código para resolver el problema del granjero sin incertidumbre para los rendimientos en el primer escenario:

```
Archivo principal:
NAME          FARMER
ROWS
 N  WEALTH
 L  BUDGET
 G  BAL1
 G  BAL2
 L  BAL3
 L  BAL4
COLUMNS
    X1        BUDGET    1.00          BAL1      +2.5
    X2        BUDGET    1.00          BAL2      +3
    X3        BUDGET    1.00          BAL3      -20
    Y1        BAL1      1.00
    Y2        BAL2      +1.00
    W1        BAL1      -1.00
    W2        BAL2      -1.00
    W3        BAL3      +1.00         BAL4      +1.00
    W4        BAL3     +1.00
    X1        WEALTH    150
    X2        WEALTH    230
    X3        WEALTH    +260
    Y1        WEALTH    +238
    Y2        WEALTH    +210
    W1        WEALTH    -170
    W2        WEALTH    -150
    W3        WEALTH    -36
    W4        WEALTH    -10
RHS
    RHS       BUDGET    500           BAL1      200
RHS       BAL2      240           BAL4      6000
ENDATA

Archivo de tiempo:
```

```
TIME          FARMER
PERIODS
    X1        BUDGET                  TODAY
    Y1        BAL1                    TOMORROW
ENDATA

Archivo estocástico:
STOCH         FARMER
BLOCKS        DISCRETE
 BL BLOCK1    TOMORROW  1
    X1        BAL1      2
    X2        BAL2      2.4
    X3        BAL3      -16
ENDATA
```

Aunque añadimos el archivo estocástico, en el mismo, solo ocurre un suceso.

Código para resolver el problema del granjero sin incertidumbre para los rendimientos en el tercer escenario:

```
Archivo principal:
NAME          FARMER
ROWS
 N  WEALTH
 L  BUDGET
 G  BAL1
 G  BAL2
 L  BAL3
 L  BAL4
COLUMNS
    X1        BUDGET    1.00          BAL1      +2.5
    X2        BUDGET    1.00          BAL2      +3
    X3        BUDGET    1.00          BAL3      -20
    Y1        BAL1      1.00
    Y2        BAL2      +1.00
    W1        BAL1      -1.00
    W2        BAL2      -1.00
    W3        BAL3      +1.00         BAL4      +1.00
    W4        BAL3     +1.00
    X1        WEALTH    150
    X2        WEALTH    230
    X3        WEALTH    +260
    Y1        WEALTH    +238
    Y2        WEALTH    +210
    W1        WEALTH    -170
    W2        WEALTH    -150
    W3        WEALTH    -36
    W4        WEALTH    -10
RHS
    RHS       BUDGET    500           BAL1      200
RHS       BAL2      240           BAL4      6000
ENDATA
```

```
Archivo de tiempo:
TIME          FARMER
PERIODS
    X1        BUDGET                    TODAY
    Y1        BAL1                      TOMORROW
ENDATA

Archivo estocástico:
STOCH         FARMER
BLOCKS        DISCRETE
 BL BLOCK1    TOMORROW  1
    X1        BAL1      3
    X2        BAL2      3.6
    X3        BAL3      -24
ENDATA
```

Aunque añadimos el archivo estocástico, en el mismo, solo ocurre un suceso.

# Apéndice B

# Código en lenguaje R

Algoritmo de ahorros contexto determinista:

```
Ahorrosalgorithm<-function(vector.demandas,matriz.distancia,capacidad.vehiculo){

n<-dim(matriz.distancia)[1] #número de clientes más depósito

c<-numeric(n) #vector costes de rutas
R<-matrix(0,nrow=n,ncol=3) #matriz de rutas

##########Paso 1: calcular rutas ir y volver

c<-matriz.distancia[1,]*2 #Coste ir desde el depósito al cliente i y volver
R[,2]<-1:n #Generamos las rutas (0,i,0) donde 0 es depósito
R[1,2]<-0

ctotal<-sum(c) #Coste total de ir desde cada depósito al cliente

S<-matrix(0,nrow=n,ncol=n) #Matriz ahorros

############Paso 2: calcular los ahorros


for(i in 2:n){
for(j in 2:n){
{
if(i!=j){
S[i,j]<-matriz.distancia[i,1]+matriz.distancia[1,j]-matriz.distancia[i,j]
}
}
}
}

############Paso 3: optimizar rutas

indicar<-1
Sm<-1 #Valores de entrada del primer while
```

```
while(Sm>0){ #Mientras existan ahorros mayores que cero buscamos rutas factibles

Sm<-max(S) #Ahorro máximo
if(Sm>0){
{ #Posición ahorro máximo

if(order(S,decreasing=TRUE)[1]%%n==0){
Positionfilas<-n
Positioncolumnas<-order(S,decreasing=TRUE)[1]%/%n
}

else{
Positionfilas<-order(S,decreasing=TRUE)[1]%%n
Positioncolumnas<-order(S,decreasing=TRUE)[1]%/%n + 1
}

}

#Demandas de los clientes i y j
CargaT<-vector.demandas[Positionfilas]+vector.demandas[Positioncolumnas]

{ #Indicamos a que cliente visitamos antes de ir a i y después de ir a j

if(R[Positionfilas,3]==0 && R[Positioncolumnas,1]==0 &&  CargaT<=capacidad.vehiculo){
newPositionfilas<-Positionfilas
newPositioncolumnas<-Positioncolumnas
x<-0 #Evitamos ciclos

while(R[newPositionfilas,1]!=0){ #Sumamos la carga de los clientes anteriores a i
CargaT<-CargaT+vector.demandas[R[newPositionfilas,1]]
newPositionfilas<-R[newPositionfilas,1]
if(newPositionfilas==Positioncolumnas) x<-x+1
}

while(R[newPositioncolumnas,3]!=0){ #Sumamos la carga de los clientes posteriores a j
CargaT<-CargaT+vector.demandas[R[newPositioncolumnas,3]]
newPositioncolumnas<-R[newPositioncolumnas,3]
if(newPositioncolumnas==Positionfilas) x<-x+1
}

if(CargaT<=capacidad.vehiculo && x==0){ #Añadimos la ruta si es factible
R[Positionfilas,3]<-Positioncolumnas
R[Positioncolumnas,1]<-Positionfilas
}
S[Positionfilas,Positioncolumnas]<-0
S[Positioncolumnas,Positionfilas]<-0 #Borramos ahorros utilizados para evitar ciclos
}
}
S[Positionfilas,Positioncolumnas]<-0 #Si no es factible tambien lo borramos
}
```

```
} #Fin del while

rutas<<-numeric() #Vector de rutas FINAL
rutas[1]<<-0 #Empezamos en el deposito
indicador<-2 #Nos movemos por el vector de rutas

#Creamos la ruta final

for(i in 2:n){

if(R[i,1]==0){
rutas[indicador]<<-i

while(rutas[indicador]!=0){
rutas[indicador+1]<<-R[rutas[indicador],3]
indicador<-indicador+1

}
indicador<-indicador+1
}
}

rutas[which(rutas==0)]<-1
coste.total<-0
for(i in 1:(length(rutas)-1)){
coste.total<-coste.total+matriz.distancia[rutas[i],rutas[i+1]]
}

rutas<-rutas-1

cat("Tomamos las rutas",":\n")
cat(rutas,"\n")
cat("Con un coste total de",":\n")
cat(coste.total,"\n")

} #Fin de la función
```

**Ejemplo:**

```
capacidad.vehiculo<-10
matriz.distancia<-matrix(c(0,6,4,5,3,6,0,4,8,4,4,4,0,1,4,5,8,1,0,3,3,4,4,3,0),nrow=5)
vector.demandas<-c(0,2,2,1,2)

Ahorrosalgorithm(vector.demandas,matriz.distancia,capacidad.vehiculo)

Tomamos las rutas :
0 4 3 2 1 0
Con un coste total de :
17
```

Algoritmo de ahorros contexto estocástico:

```
Stochasticahorrosalgorithm<-function(matriz.distancia,capacidad.vehiculo,
demandas,probs,VARIABLES){

n<-dim(matriz.distancia)[1] #número de clientes más depósito

c<-numeric(n) #vector costes de rutas
R<-matrix(0,nrow=n,ncol=3) #matriz de rutas

##########Paso 1: calcular rutas ir y volver

R[,2]<-1:n #Generamos las rutas (0,i,0) donde 0 es depósito
R[1,2]<-0


############Paso 2: calcular los ahorros

S<-matrix(0,nrow=n,ncol=n) #Matriz ahorros

########################
#####Función probabilidad cliente i con variables discretas o normales
########################

{
if(VARIABLES=="DISCRETAS"){

probabilidad<-function(j){

probabanterior<-probab
d2<-demandas[rutasi[j],]
probs2<-probs[rutasi[j],]

{
if(j==3){
d1<-demandas[rutasi[j-1],]
d12<<-matrix(0,ncol=length(d2),nrow=length(d1))
probs12<-matrix(0,ncol=length(d2),nrow=length(d1))
probs1<-probs[rutasi[j-1],]
}
else{
d1<-d12[1:length(d12)]
d12<<-matrix(0,ncol=length(d2),nrow=length(d1))
probs1<-probs12[1:length(probs12)]
probs12<-matrix(0,ncol=length(d2),nrow=length(d1))
}
}


for(i in 1:length(d1)){
```

```
d12[i,1:length(d2)]<-d1[i]+d2[1:length(d2)]
probs12[i,1:length(probs2)]<-probs1[i]*probs2[1:length(probs2)]

}
d12<<-d12
probs12<<-probs12
probab<<-sum(probs12[which(d12>capacidad.vehiculo)])-probabanterior
probab2<<-sum(probs12[which(d12<=capacidad.vehiculo)])
return(probab)
}
}
else if(VARIABLES=="NORMALES"){

probabilidad<-function(j){

probabanterior<-probab

{
if(j==3){

d12medias<<-demandas[rutasi[j]] + demandas[rutasi[j-1]]
d12sigma<<-probs[rutasi[j]] + probs[rutasi[j-1]]
}
else{
d12medias<<-d12medias + demandas[rutasi[j]]
d12sigma<<-d12sigma + probs[rutasi[j]]
}
}

probab<<-pnorm(capacidad.vehiculo,mean=d12medias,sd=d12sigma,lower.tail=F) - probabanterior
probab2<<-pnorm(capacidad.vehiculo,mean=d12medias,sd=d12sigma,lower.tail=T)
return(probab)

}
}
}

########################
#####Función valor coste esperado para un cliente i
########################

coste.esperado<-function(i){

inew<-i
rutasi<-numeric()
rutasi[1]<-1
indicador<-n-1
rutasi[n]<-i

while(R[inew,1]!=0){ #Buscamos los clientes que preceden a i
rutasi[indicador]<-R[inew,1]
inew<-R[inew,1]
```

```
indicador<-indicador-1
}

rutasii<-numeric() #Eliminamos los NA
rutasii[1:length(na.omit(rutasi))]<-na.omit(rutasi)[1:length(na.omit(rutasi))]
rutasi<-rutasii

inew<-i
indicador<-length(rutasi)+1

while(R[inew,3]!=0){ #Buscamos los clientes que siguen a i
rutasi[indicador]<-R[inew,3]
inew<-R[inew,3]
indicador<-indicador+1
}
rutasi[indicador]<-1
rutasi<<-rutasi

#Calculamos el coste esperado una vez conocida la ruta que contiene a i
costeesperado<-matriz.distancia[rutasi[1],rutasi[2]]
distancia.visitas<-0
probab<<-0
{
if(length(rutasi)>=4){
for(j in 3 :(length(rutasi)-1)){

distancia.visitas<-distancia.visitas+matriz.distancia[rutasi[j-1],rutasi[j]]
costeesperado<-costeesperado+probabilidad(j)*(distancia.visitas+matriz.distancia[rutasi[j],1]+
2*(sum(matriz.distancia[1,rutasi[j:(length(rutasi)-1)]])))

}

costeesperado<-costeesperado+probab2*
(distancia.visitas+matriz.distancia[rutasi[length(rutasi)-1],1])
}

else{
costeesperado<-costeesperado+matriz.distancia[rutasi[1],rutasi[2]]
}
}
return(costeesperado)
}
################################################
#####Función valor coste esperado para que un cliente i precede a uno j
################################################

coste.unido.esperado<-function(i,j){

inew<-i
rutasij<-numeric()
rutasij[1]<-1
indicador<-n-1
```

```
rutasij[n]<-i

while(R[inew,1]!=0){ #Buscamos los clientes que preceden a i
rutasij[indicador]<-R[inew,1]
inew<-R[inew,1]
indicador<-indicador-1
}

rutasii<-numeric() #Eliminamos los NA
rutasii[1:length(na.omit(rutasij))]<-na.omit(rutasij)[1:length(na.omit(rutasij))]
rutasij<-rutasii

inew<-i
jnew<-j
rutasij[length(rutasij)+1]<-j
indicador<-length(rutasij)+1

while(R[jnew,3]!=0){ #Buscamos los clientes que siguen a j
rutasij[indicador]<-R[jnew,3]
jnew<-R[jnew,3]
indicador<-indicador+1
}
rutasij[indicador]<-1
rutasi<<-rutasij

#Calculamos el coste esperado una vez conocida la ruta que contiene a i
costeesperado<-matriz.distancia[rutasij[1],rutasij[2]]
distancia.visitas<-0
probab<<-0
for(j in 3:(length(rutasij)-1)){

distancia.visitas<-distancia.visitas+matriz.distancia[rutasij[j-1],rutasij[j]]
costeesperado<-costeesperado+probabilidad(j)*(distancia.visitas+matriz.distancia[rutasij[j],1]+
2*(sum(matriz.distancia[1,rutasij[j:(length(rutasij)-1)]])))

}
costeesperado<-costeesperado+probab2*
(distancia.visitas+matriz.distancia[rutasij[length(rutasij)-1],1])
return(costeesperado)
}


######################
###Calculo de los ahorros
######################

for(i in 2:n){
for(j in 2:n){
{
if(i!=j & R[i,3]==0 & R[j,1]==0){
S[i,j]<-coste.esperado(i)+coste.esperado(j)-coste.unido.esperado(i,j)
}
```

```
}
}
}

##################
###Calculo ruta óptima
##################

indicar<-1
Sm<-1 #Valores de entrada del primer while

while(Sm>0){ #Mientras existan ahorros mayores que cero buscamos rutas factibles

Sm<-max(S) #Ahorro máximo
if(Sm>0){

{ #Posición ahorro máximo

if(order(S,decreasing=TRUE)[1]%%n==0){
Positionfilas<-n
Positioncolumnas<-order(S,decreasing=TRUE)[1]%/%n
}

else{
Positionfilas<-order(S,decreasing=TRUE)[1]%%n
Positioncolumnas<-order(S,decreasing=TRUE)[1]%/%n + 1
}

}

R[Positionfilas,3]<-Positioncolumnas
R[Positioncolumnas,1]<-Positionfilas

##############################
######Calculo de los ahorros nuevamente
##############################

S<-matrix(0,nrow=n,ncol=n) #Matriz ahorros

for(i in 2:n){
for(j in 2:n){
{
if(i!=j & R[i,3]==0 & R[j,1]==0){
S[i,j]<-coste.esperado(i)+coste.esperado(j)-coste.unido.esperado(i,j)
}
}
}
}

S[Positionfilas,Positioncolumnas]<-0
S[Positioncolumnas,Positionfilas]<-0 #Borramos ahorros utilizados para evitar ciclos
}
```

```
} #Fin del while

rutas<<-numeric() #Vector de rutas FINAL
rutas[1]<<-0 #Empezamos en el deposito
indicador<-2 #Nos movemos por el vector de rutas

#Creamos la ruta final

for(i in 2:n){

if(R[i,1]==0){
rutas[indicador]<<-i

while(rutas[indicador]!=0){
rutas[indicador+1]<<-R[rutas[indicador],3]
indicador<-indicador+1

}
indicador<-indicador+1
}
}

rutas[which(rutas==0)]<<-1
coste.total<-0
ind<-which(rutas==1)[2:length(which(rutas==1))]
for(i in ind){
coste.total<-coste.total+coste.esperado(rutas[i-1])
}

rutas<-rutas-1
cat("Tomamos las rutas",":\n")
cat(rutas,"\n")
cat("Con un coste esperado total de",":\n")
cat(coste.total,"\n")

} #Fin función algoritmo de ahorros estocástico
```

**Ejemplo caso discreto:**

```
capacidad.vehiculo<-10
matriz.distancia<-matrix(c(0,2,4,4,1,2,0,3,4,2,4,3,0,1,3,4,4,1,0,3,1,2,3,3,0),nrow=5)
demandas<-matrix(c(0,0,2,0,2,0,1,7,2,0),nrow=5,byrow=TRUE)
probs<-matrix(c(0,0,1,0,1,0,1/2,1/2,1,0),nrow=5,byrow=TRUE)
VARIABLES<-"DISCRETAS"

Stochasticahorrosalgorithm(matriz.distancia,capacidad.vehiculo,demandas,probs,VARIABLES)

Tomamos las rutas :
0 3 2 1 4 0
Con un coste esperado total de :
13.5
```

**Ejemplo caso continuo:**

```
capacidad.vehiculo<-10
matriz.distancia<-matrix(c(0,2,4,4,1,2,0,3,4,2,4,3,0,1,3,4,4,1,0,3,1,2,3,3,0),nrow=5)
demandas<-c(0,2,2,4,2)
probs<-c(0,0.1,0.2,2,0.1)
VARIABLES<-"NORMALES"

Stochasticahorrosalgorithm(matriz.distancia,capacidad.vehiculo,demandas,probs,VARIABLES)

Tomamos las rutas :
0 3 2 1 0 4 0
Con un coste esperado total de :
13.0107
```

# Bibliografía


[AE07] Aykagan Ak and Alan L Erera. A paired-vehicle recourse strategy for the vehicle-routing problem with stochastic demands. *Transportation Science*, 41(2):222–237, 2007.

[BBKA14] Eshetie Berhan, Birhanu Beshah, Daniel Kitaw, and Ajith Abraham. Stochastic vehicle routing problem: A literature survey. *Journal of Information & Knowledge Management*, 13(03):12 pages, 2014.

[Bea55] Evelyn ML Beale. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society. Series B (Methodological)*, 17(2):173–184, 1955.

[BL11] John R Birge and Francois Louveaux. *Introduction to Stochastic Programming*. Springer Science & Business Media, 2011.

[CC59] Abraham Charnes and William W Cooper. Chance-constrained programming. *Management science*, 6(1):73–79, 1959.

[CCA+15] Winston Chang, Joe Cheng, J Allaire, Yihui Xie, Jonathan McPherson, et al. Shiny: Web application framework for R, 2015. *URL http://CRAN. R-project. org/package= shiny. R package version 0.11*, 2015.

[CMM98] Joseph Czyzyk, Michael P Mesnier, and Jorge J Moré. The neos server. *IEEE Journal on Computational Science & Engineering*, 5(3):68–75, 1998.

[CW64] GU Clarke and John W Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581, 1964.

[Dan55] George B Dantzig. Linear programming under uncertainty. *Management Science*, 1(3-4):197–206, 1955.

[DLT89] Moshe Dror, Gilbert Laporte, and Pierre Trudeau. Vehicle routing with stochastic demands: Properties and solution frameworks. *Transportation Science*, 23(3):166–176, 1989.

[Dol01] Elizabeth D Dolan. Neos server 4.0 administrative guide. *arXiv preprint cs/0107034*, 2001.

[DR59] George B Dantzig and John H Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.

[DT86] Moshe Dror and Pierre Trudeau. Stochastic vehicle routing with modified savings algorithm. *European Journal of Operational Research*, 23(2):228–235, 1986.

[EVR09] Burak Eksioglu, Arif Volkan Vural, and Arnold Reisman. The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4):1472–1483, 2009.

[GK08] Horand I Gassmann and Bjarni Kristjánsson. The smps format explained. *IMA Journal of Management Mathematics*, 19(4):347–377, 2008.

[GLS96] Michel Gendreau, Gilbert Laporte, and René Séguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88(1):3–12, 1996.

[GM97] William Gropp and Jorge Moré. Optimization environments and the neos server. *In: Approximation theory and optimization, M.D. Buhmann and A. Iserles, eds. Cambridge University Press Cambridge, UK*, pages 167–182, 1997.

[GRWE08] Bruce L Golden, Subramanian Raghavan, and Edward A Wasil (Eds.). *The vehicle routing problem: latest advances and new challenges*, volume 43. Springer Science & Business Media, 2008.

[KT51] Harold W. Kuhn and Albert W. Tucker. Nonlinear programming. In *: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492. The Regents of the University of California, 1951.

[LL93] Gilbert Laporte and François V Louveaux. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133–142, 1993.

[MTZ60] Clair E Miller, Albert W Tucker, and Richard A Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4):326–329, 1960.

[NEO17] NEOS WEB. `https://neos-server.org/neos/`, 2017.

[SGL17] Juan José Salazar González and François V Louveaux. Exact approach for the vehicle routing problem with stochastic demands and preventive returns. *Manuscrito de los autores. Presentado al congreso de la Real Sociedad Matemática Española (RSME), Zaragoza*, Enero de 2017.