

Optimización de la producción de concentrados de plaquetas en bancos de sangre



Gorka Labata Lezaun

Trabajo de fin del Máster en Modelización e
Investigación Matemática, Estadística y
Computación
Universidad de Zaragoza

Directores del trabajo: Carlos Gorria

Francisco Javier López

18 de julio de 2017

Agradecer al Banco de Sangre y Tejidos de Aragón
su apoyo y participación en todo el proyecto y en especial a
Ana Isabel Perez por explicarme todo el funcionamiento y
atender mis innumerables dudas.

Prólogo

En esta memoria se presenta un problema real de gestión de inventarios propuesto por el Banco de Sangre y Tejidos de Aragón. En concreto se trata de proponer estrategias óptimas para la producción de concentrados de plaquetas que proceden de donaciones voluntarias. Partiendo de un trabajo realizado por miembros de la Universidad de País Vasco en colaboración con un profesor de esta universidad, decidimos extenderlo al caso del Banco de Sangre y Tejidos de Aragón. Tras una reunión con miembros de dicha institución, las diferencias en el sistema de producción con respecto a lo anteriormente estudiado nos obligan a realizar una recapitulación sobre distintas formas de resolución en la literatura.

Puesto que diariamente se transfunden un considerable número de unidades de este producto que cuenta con un corto periodo de caducidad de entre 5 y 7 días. Este importante condicionante temporal, junto con el elevado coste económico de procesamiento, alrededor de 300 euros / unidad, hace recomendable el planteamiento de una metodología matemática de optimización.

Inicialmente el problema matemático presenta algunas dificultades como el tamaño de las bases de datos y la incertidumbre sobre la demanda diaria de concentrados, que se puede representar como una variable aleatoria. Del análisis exploratorio de los datos se deduce cierta estacionalidad en la serie temporal de unidades transfundidas y un comportamiento que se aproxima a una variable aleatoria normal en la demanda de cada uno de los 7 días de la semana. Estas características facilitan la utilización de técnicas estudiadas en algunas asignaturas del Máster en Modelización e Investigación Matemática, Estadística y Computación, como Series Temporales y Modelización Estadística. Por otra parte el problema propio de optimización de inventarios puede ser planteado mediante técnicas propuestas en las asignaturas de Modelos de Logística y Técnicas de Computación Evolutiva y Algoritmos Bioinspirados. Dicha optimización tratará de encontrar una estrategia óptima para la producción.

Para validar dicha estrategia se utilizarán tanto datos reales como simulaciones. Dichas simulaciones, así como la resolución con el algoritmo bioinspirado se programará en R.

Todo el estudio será recogido en una herramienta Excel que se entregará al Banco de Sangre y Tejidos de Aragón para que pueda servir de apoyo a la hora de la toma de decisiones en la producción.

Resumen

Blood derivatives are a crucial resource in medicine and no one doubts they save lives. However, they cannot be produced out of nothing and its production depends on the number of blood donations available. Also, each type of blood derivative has its own self life, being platelet concentrates (also called platelet pools) the ones with shortest self time. The high costs derived from their production and the uncertainties in the future demand make platelet concentrate's production a really interesting field to answer mathematically. As the restrictions of the problem are different for each case and depend on the logistics of the production center and the supply chain, a specific study of our case is needed.

We will focus in the case of the "Banco de Sangre y tejidos de Aragón". A revealing meeting with members of the direction revealed us the particularities of this institution. At the beginning of year 2016, they started to implement a new treatment to the concentrates that allowed to increment their self life in 2 days, from 5 to 7 days by irradiation. Due to budget limitations, they started producing 25% of those irradiated pools while they expect to be able to produce a 50% in 2017 and end producing all of them in a close future. Therefore, irradiation will increase the production costs and could be also optimized. The current production strategy will be explained in detail to be compared with the strategies studied in this paper.

Along the work, we will use the data obtained since the beginning of 2016 to April 2017. Based on the data compiled by the blood bank during 2016, we will build a mathematical model explaining the demand depending on each day of the week. After a simple transformation to place the start of production at the beginning of the day, we will consider those demands normally distributed and independent from other days as it will be corroborated by Shapiro-Wilk test and auto correlation function.

Once modelled the demand, we propose a modified FIFO (*first in first out*) where the first leaving are the closest to the expiry day, as the sending policy and a model of reordering up to a level for the production policy. This level will be different for each day of the week and will be obtained as the mean demand plus 3 times its standard deviation. This will allow us an average stockout of 1 every 740 days (less than once every two years), when in 2016 there were 36 stockout.

After considering the number of concentrates that will expire in the non-producing days, we will test our production policy for 5 and 7 self life concentrates for production models with 1 and 2 non-producing days. We do it in 3 different ways: using data from 2016, using 4 months of 2017 and by simulating data. With data from 2016 and 2017, we find that for models with 6 producing days per week we could get zero expirations, much better than the 2.28% that they are obtaining currently. With 5 producing days, we can obtain a 1.26% expiry rate with 5 self life concentrates. However, with 7 self life concentrates, we can obtain 0% expirations, what makes us think that a combined model will reach the perfect strategy.

In order to obtain such strategy, we will translate our model into an optimization problem using as decision variables the percentage of 5 self life and 7 self life concentrates that we will be produced each day of the week. Due to the size of the problem, that has 7371 variables and 18227 restrictions, we will not be able to find a global solution, so we will use heuristics. We will propose a bio-inspired computing such as Particle Swarm Optimization and we will explain it in detail. Using penalties for the use of 7 days concentrates, we will be able to reach the optimum strategy with 2016 data and check it with 2017 and simulations with all others strategies. Producing all 5 days concentrates everyday but the days before non producing days when 30% of 7 days concentrates are produced, we will reach similar results as with only 7 days concentrates.

Once we will have studied different cases and obtained the answers we needed, we will turn to the main subject of the study: producing a tool that will help the blood bank to produce the most accurate number of concentrates of each type. In order to make a handy and easy to use tool, we will choose Excel as the format to implement it. We will include some instructions in a PDF that we will also hand in and attach those same instructions to the Excel file.

The excel will be composed of 4 sheets. The first and principal one will collect all the data from the production and sending decisions. It will allow the manager to introduce whether a day is non-producing or the maximum productions for each day. It will also contain the number of concentrates produced and sent each day and for each type. The second one will contain some interesting statistics as the average age of the pools when they are sent to the hospitals or the demand per day of the week. The third one keeps a record of the stock available every day and the type and ages of each concentrate. The last sheet will be an explanation of the other sheets.

Índice general

| | |
|---|------------|
| Prólogo | V |
| Resumen | VII |
| 1. | 1 |
| 1.1. Introducción | 1 |
| 1.2. El primer modelo | 3 |
| 1.2.1. Estimación de unidades caducadas | 5 |
| 1.2.2. Simulaciones | 7 |
| 2. | 11 |
| 2.1. Cambios en el modelo | 11 |
| 2.1.1. Estrategia óptima | 11 |
| 3. | 17 |
| 3.1. La Herramienta | 17 |
| 3.1.1. Hoja a hoja | 17 |
| 3.1.2. Modo de Empleo | 19 |
| 3.2. Conclusiones y trabajos futuros | 21 |
| Anexos | 23 |
| 3.3. Anexo I | 23 |
| 3.4. Anexo II | 26 |
| 3.5. Anexo III | 41 |
| Bibliografía | 45 |

Capítulo 1

1.1. Introducción

Los bancos de sangre, aparte de a la extracción de sangre, se dedican a la producción y almacenamiento de 3 tipos de componentes sanguíneos derivados de ésta: glóbulos rojos, plasma y plaquetas. Esto es porque a los pacientes nunca se les trasfunde la sangre completa, sino uno de estos derivados dependiendo de sus necesidades. La caducidad de los concentrados de glóbulos rojos es de 1 mes o mes y medio y se utilizan en pacientes con anemia. La caducidad de los de plasma es de en torno a 3 años, ya que se irradian para eliminar patógenos y después se congelan, y se suelen utilizar en pacientes con trastornos en la coagulación. Esto hace que el riesgo de que se caduque un concentrado de estos sea muy bajo y, de hecho, pocas son las veces que tienen que desechar un concentrado de estos tipos.

Los concentrados de plaquetas se utilizan en pacientes que tienen bajos niveles de estas, de tal forma que podrían tener una hemorragia. Su fecha de caducidad es mucho más temprana y su demanda bastante alta puesto que se utilizan en numerosos tratamientos, en la mayoría de los casos para enfermos graves. Durante mucho tiempo los concentrados de plaquetas se volvían inservibles a los 5 días de ser producidos a partir de sangre de donantes pero ahora, gracias a un sistema por el que irradian los concentrados para eliminar otros microorganismos, este tiempo puede extenderse hasta a los 7 días.

Aunque los procesos de producción y transporte de los concentrados varían dependiendo del lugar, encuestas a nivel español realizadas en 2012 indican que los índices de caducidad en los centros de producción son del 3.49% para centros que fabrican concentrados de 5 días y del 1.62% para centros que fabrican concentrados de 7 días [9]. Si embargo, tanto estos porcentajes de caducidades como la vida media con la que son transfundidos, con respecto a la fecha de donación, pueden ser mejorados utilizando modelos matemáticos que optimicen el número de pools producidos y su gestión al ser transfundidos como se demuestra en [6].

El Banco de Sangre y Tejidos de Aragón, que será la institución en la que centraremos mayormente nuestro estudio, implantó el sistema de irradiación en 2016, irradiando un 25% de concentrados ese año y ampliando ese porcentaje al 50% en 2017. Prevén poco a poco ser capaces de irradiar el 100% de los concentrados en unos años. Todo esto hace que en 2016, el Banco de Sangre y Tejidos de Aragón desechase 143 por haberse caducado de los 6285 producidos, el 2,28%, un porcentaje mucho menor que el estudiado en [10], pero que, debido a las menores restricciones que se dan en nuestro caso, creemos es susceptible de mejora.

Las restricciones vienen derivadas del sistema de producción de los concentrados. El banco de sangre realiza extracciones de lunes a sábado. De lunes a viernes esta sangre se analiza y si da negativo en un test para diferentes enfermedades, se declara apta para transfusión y puede ser utilizada para producir concentrados. Esto es ya al día siguiente y por tanto las plaquetas tienen ya 1 día. Se juntan 4 bolsas del mismo tipo sanguíneo (a veces 5 si hay poca concentración o 3 si hay mucha) y se les somete a un centrifugado que separa las plaquetas. Al elemento resultante de la unión de varias bolsas se le denomina pool. También pueden obtenerse mediante aféresis de forma que todas las plaquetas provienen de un mismo donante, algo útil en determinados casos. Puesto que el proceso de producción se realiza en el mismo lugar donde se almacenan los concentrados y desde donde se envían, una vez separadas las plaquetas, estos están ya disponibles para ser enviados, de forma que se reduce el *lead time*

a 0. La hora que comienzan a producir es en torno a las 2-3 del mediodía y producen hasta completar el número que han decidido hacer.

Con la sangre extraída el sábado se actúa de distinta forma. Como el laboratorio de análisis cierra los domingos, la sangre no puede ser analizada hasta el lunes. De esta forma, si se tiene mucha necesidad de concentrados (actualmente el nivel de referencia es tener menos de 50 unidades), la sangre se utiliza para producirlos. Sin embargo, estos concentrados no pueden ser enviados hasta que son analizados el lunes o, en caso de ser enviados, no pueden ser utilizados hasta que no estén disponibles los resultados el lunes por la mañana.

La unidad de producción cierra los lunes, aparte de los domingos si hay un número suficiente de concentrado guardados en el almacén.

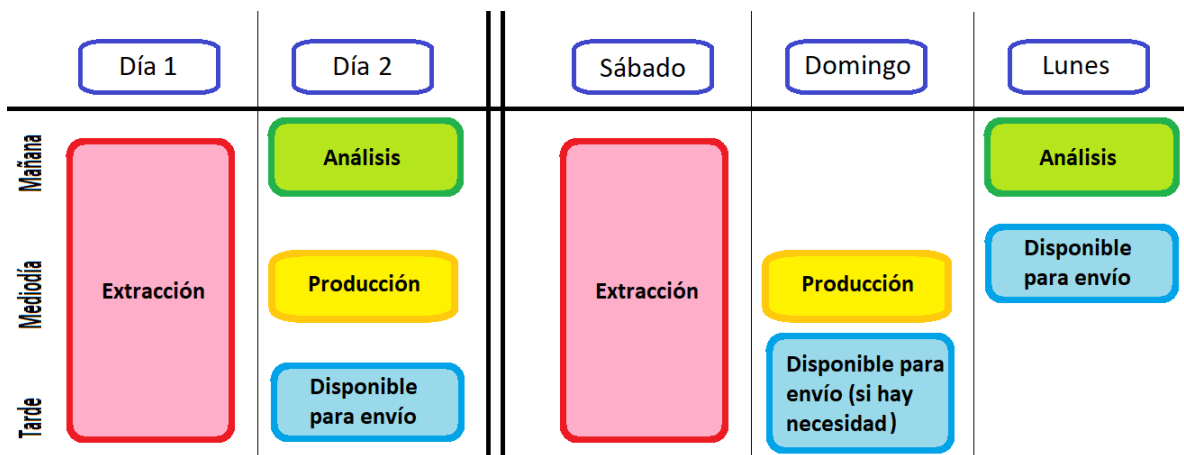


Figura 1.1: Dibujo explicativo de las principales fases por las que pasa la sangre hasta convertirse en concentrado de plaquetas dependiendo del día de la semana.

A la hora de enviar los pedidos de concentrados que piden los hospitales, se reparten primero los más cercanos a la fecha de caducidad. Esta política sería un FIFO (*first in first out*) modificado, puesto que en el caso en el que solo existiera un tipo de concentrado se trataría de esta política. Como hay 2 tipos, irradiados y no irradiados, en caso de empate se reparten primero los más viejos, los de 7 días. De esta forma el banco de sangre fabrica concentrados para cubrir los pedidos del día y tratar de no quedarse sin stock. Hay hospitales que realizan pedidos todos los días y hospitales que no. A veces un hospital puede pedir concentrados de plaquetas de un tipo específico, pero esto ocurre en contadas ocasiones, como cuando una embarazada necesita una transfusión, que se utiliza Rh-.

Los centros de producción suelen tener conflictos con los hospitales, debido a que los primeros tratan de evitar despilfarros mientras que los segundos pretenden mantener una reserva bastante elevada para salvar posibles problemas. Aparte del motivo económico (producir un concentrado sin irradiar cuesta en torno a 300 euros), existe el motivo ético, por el cual no parece correcto desperdiciar la sangre que los donantes han proporcionado con toda su ilusión y cariño. En la actualidad, el número de concentrados fabricados cada día se decide por parte de los responsables del banco sin ninguna herramienta matemática o estadística, sino por el conocimiento que tienen del problema. Lo que nuestro modelo busca controlar es que determinados días se produzcan demasiados concentrados, lo que provoque que no puedan ser consumidos y terminen caducando. Existen numerosas formas de encarar este problema, muchas de ellas recogidas en [4]. En nuestro caso buscaremos fijar unos niveles hasta los que rellenar el almacén de forma que se cubra con un cierto margen de seguridad la demanda prevista procurando, a la vez, que el número de unidades caducadas sea mínimo.

Un estudio similar al que haremos fue realizado en 2014 por Mikel Lezaun, Carlos Gorria y Javier López [10] para optimizar la producción en el País Vasco. Pese a la similitudes, existen suficientes diferencias como para considerar nuestro trabajo aparte. En su caso, la producción se organizaba de diferente forma, lo que provocaba que el número de concentrados que se debían producir un día se eligiese el día anterior. Los fines de semana no se producía nada y no se consideraron topes en la

producción de cada día. Además, la producción se realizaba por la noche y concentrados se encontraban disponibles a primera hora de la mañana y solo se utilizaba un tipo de ellos, los de 5 días de caducidad.

1.2. El primer modelo

Para fijar estos límites en la producción comenzamos realizando un estudio de la serie temporal que representa el número de envíos realizados en el año 2016 por el Banco de Sangre. Nuestra hipótesis será que a los hospitales no se les caducan los concentrados de plaquetas o que, en todo caso, un incremento en la producción del número de concentrados de plaquetas de 7 días no afectará a la tendencia de la demanda global, que tiende a crecer en 500 unidades cada año. De esta forma sin mucho problema seremos capaces de utilizar los datos de 2016 para inferir sobre la producción del resto de años.

También consideraremos que los días empiezan a las 3 del mediodía y que duran hasta las 3 del mediodía del día siguiente. Así, un pool enviado el 27/01/2016 del 2016 a las 13:24 contará como enviado el 26/01/2016 y sin embargo uno enviado a las 15:24 contará para el 27/01/2016. Esto nos permitirá trasladar el momento de producción al comienzo del día y considerar solo la demanda predicha para ese día.

De esta forma, una vez filtrado y transformado el conjunto de datos para obtener un serie temporal en R, calculamos media y varianza para cada día de la semana.

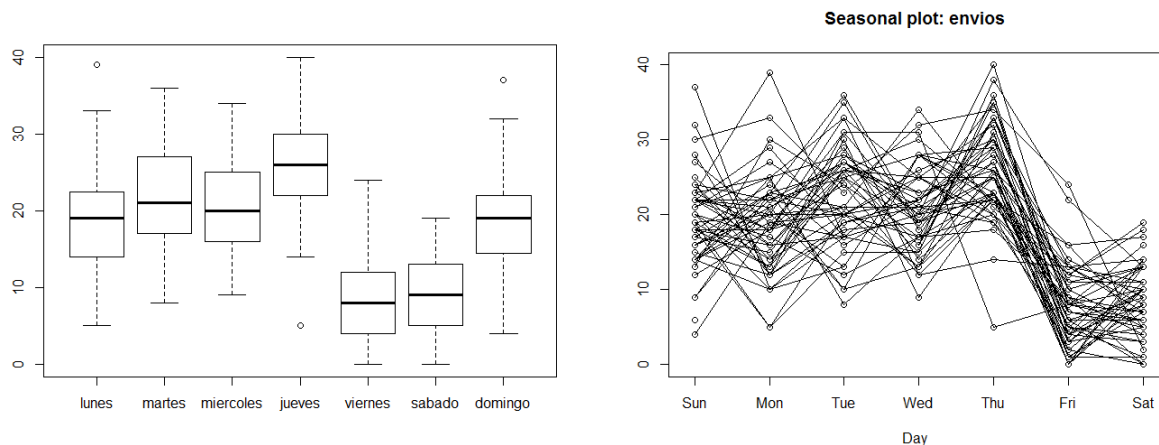


Figura 1.2: Boxplot (izquierda) y Sesonalplot (derecha) de la demanda según el día de la semana.

Como puede observarse en la figura 1, la demanda del viernes y sábado es bastante menor que la del resto de días. Sin embargo, también la demanda del resto de días es diferente y los que tienen medias similares difieren en la varianza. Esto nos obliga a considerar una demanda distinta para cada día de la semana.

Tras un estudio de los *outliers*, incluido en el Anexo I, comprobamos la normalidad de las demandas según el día de la semana mediante un test de Shapiro-Wilk, uno de los más potentes para muestras pequeñas como las nuestras (tendremos 52 datos para cada día de la semana). Todos los días pasan el test sin ningún problema excepto el viernes. Nótese que este es el día con menor demanda media, siendo esta bastante cercana a 0 teniendo en cuenta la varianza, lo que hace que valores negativos en una distribución normal semejante a la real no encuentren semejantes en la distribución empírica. Sin embargo esto hace también que el número de casos afectados por este hecho es bastante pequeño. Para un p-valor de 0.01 también los viernes pasan el test, luego consideramos la demanda distribuida normalmente.

Para poder realizar predicciones sobre la demanda en el futuro, debemos ajustar un modelo. Ajustaremos un modelo simple, que tendrá en cuenta únicamente la estacionalidad. La tendencia la consideraremos cero ya que, aunque se produzca un incremento de unas 500 unidades cada año, si tenemos

| Valores del test de Shapiro-Wilk | | |
|----------------------------------|---------|---------|
| | W | p-valor |
| Lunes | 0.97542 | 0.3536 |
| Martes | 0.98308 | 0.6634 |
| Miércoles | 0.97413 | 0.3137 |
| Jueves | 0.96808 | 0.1751 |
| Viernes | 0.94458 | 0.01726 |
| Sábado | 0.97828 | 0.4427 |
| Domingo | 0.97629 | 0.3824 |

Tabla 1.1: Valores obtenidos por los datos para el test de normalidad de Shapiro-Wilk.

en cuenta que esto se produce en 365 días, nos da un incremento de la demanda de 1.37 en 365 días. Es decir, cada día se demandan en media 0.00375 concentrados más, que nosotros tomamos por 0. La estacionalidad la calcularemos simplemente calculando la media para cada día de la semana. Una vez eliminada estacionalidad y tendencia obtenemos ruido blanco, lo que nos indica que nuestro ajuste ha terminado. Comprobamos calculand los PACF y ACF obteniendo que no hay correlación con lo ocurrido en días anteriores, como se puede ver en la figura 1.2.

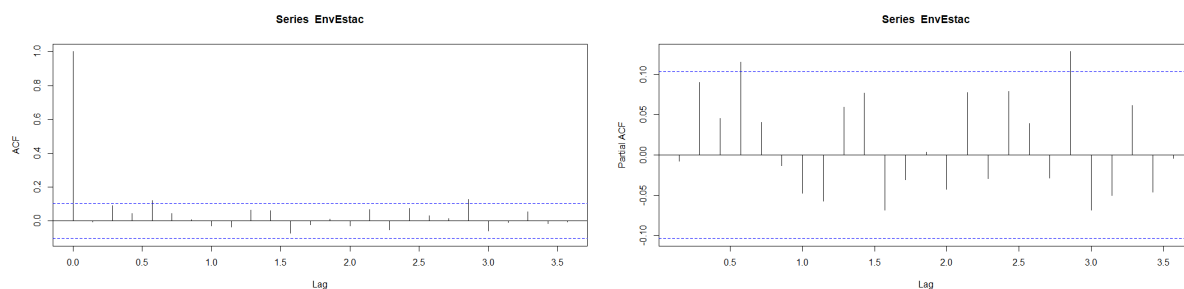


Figura 1.3: ACF (izquierda) y PACF (derecha) de la demanda una vez eliminada la estacionalidad y la tendencia.

Otros modelos pueden diseñarse utilizando medias móviles y distintas ventanas o usando modelos ARMA, pero complican las cosas pues nos darían una predicción para cada día del año en vez de cada día de la semana.

Lo que en este trabajo se propone es un modelo de revisión periódica con estrategia *up-to*. Una vez tenemos las predicciones, nuestro modelo establecerá unos niveles óptimos para el almacén para cada día, que constituyen un stock de seguridad para garantizar el abastecimiento en un porcentaje muy alto según las estadísticas, basándose en dichas predicciones y en la cantidad de concentrados de las que disponemos en el almacén. Estos niveles serán únicamente para la producción y si ya disponemos de más concentrados que los que indica dicho nivel simplemente no produciremos ninguno más. Hace falta para ello fijarlos a una cantidad. Para los días en los que el siguiente día también se produce el tope será:

$$T_i = \lceil \mu_i + k_i \sigma_i \rceil, \quad (1.1)$$

siendo μ_i y σ_i la media y la desviación típica de la demanda del i -ésimo día de la semana. k_i es una constante a elegir, que en nuestro caso fijamos a 3, como se hace en [10]. Esta constante nos permite reducir la probabilidad de que se rompa la cadena de suministro, es decir, de que nos quedemos sin concentrados para abastecer a los hospitales. En el caso de $k_i = 3$ en situaciones normales se nos rompería el stock con probabilidad 0.0013.

Si nos encontramos a día i y no se va a volver a producir hasta el día $j + 1$, la expresión es de la forma:

$$T_{i,j} = \lceil \mu_{i,j} + k_{i,j} \sigma_{i,j} \rceil, \quad (1.2)$$

donde $\mu_{i,j}$ es la suma de las demandas de los días entre i y j (ambos incluidos), $\sigma_{i,j}$ la desviación típica de esos días (obtenida como $\sigma_{i,j} = \sqrt{(\sigma_i)^2 + \dots + (\sigma_j)^2}$) y $k_{i,j}$ la constante que nos permita obtener el nivel de seguridad que deseemos para esos días.

Desde luego, como solo se puede fabricar un número entero de concentrados, elegimos como tope el techo de las cantidades.

Puesto que los domingos a veces se produce y a veces no, tenemos 2 posibles escenarios. Con los datos de 2016 y $k = 3$, los topes son:

| | | | | | | |
|-------------------|-------|-------|-------|-------|-----------|-----------|
| | T_2 | T_3 | T_4 | T_5 | $T_{6,1}$ | |
| Cerrando Domingos | 43 | 39 | 45 | 25 | 77 | |
| | T_2 | T_3 | T_4 | T_5 | T_6 | $T_{7,1}$ |
| Abriendo Domingos | 43 | 39 | 45 | 25 | 24 | 65 |

Así por ejemplo, si nos encontramos a martes y en el almacén disponemos de 20 concentrados, deberemos fabricar $43 - 20 = 23$ más.

1.2.1. Estimación de unidades caducadas

La solución expuesta corresponde a la política (R,S) de revisión periódica de productos no perecederos. Sin embargo, en nuestro caso estamos tratando con productos perecederos. El análisis de los modelos de producto perecedero es mucho más complicado que el de productos no perecederos y no se conocen soluciones óptimas excepto en modelos muy simplificados. La solución que venimos a adoptar en este trabajo es la denominada EWA [13] en la que a la producción de cada día $T_{i,j}$ anterior se le suman las cantidades que se estima van a caducar entre los días i y j . Para poder realizar esta estimación es necesario tener en cuenta no solo la caducidad de unidades en el stock sino también las edades de estas. Así, para la producción de martes a viernes (o a sábado si hay producción el domingo) la caducidad no afecta a la hora de producir ya que el *lead time* es 0. En el caso de los domingos (o del sábado si no hay producción los domingos) la cantidad que va a caducar es igual a $E(a - X)^+$, donde a es el número de unidades que caducan el domingo y X la demanda del domingo. Como hemos dicho, X nos es desconocido, luego esta cantidad también, y su valor esperado es:

$$\begin{aligned} E(a - X)^+ &= \int_{-\infty}^a (a - x)f(x) dx = a \int_{-\infty}^a f(x) dx - \int_{-\infty}^a xf(x) dx + \int_a^{\infty} xf(x) dx = \\ &= aF(a) - \mu + \int_a^{\infty} xf(x) dx \end{aligned} \quad (1.3)$$

siendo f la función de densidad de X , F su función de distribución acumulada y μ su media. En nuestro caso, tratándose de una distribución normal de media μ y desviación típica σ tenemos:

$$\begin{aligned} \int_a^{\infty} xf(x) dx &= \int_a^{\infty} \frac{x}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = \sigma \int_a^{\infty} \frac{x - \mu}{\sqrt{2\pi}\sigma^2} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx + \mu \int_a^{\infty} \frac{x}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = \\ &= \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{(a-\mu)^2}{2\sigma^2}} + \mu(1 - F(a)), \end{aligned}$$

y por tanto:

$$E(a - X)^+ = (a - \mu)F(a) + \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{(a-\mu)^2}{2\sigma^2}}. \quad (1.4)$$

En el caso de que no haya producción el domingo, entonces la producción del sábado tiene que tener en cuenta la caducidad de sábado y domingo. Si se trata de un puente y hay 3 días festivos, pongamos que el sábado tampoco se produce. Llamaremos X a la demanda del viernes, Y a la demanda del sábado y Z a la demanda del domingo. De la misma forma utilizaremos los subíndices X para lo referente al viernes, Y para lo referente al sábado y Z para lo referente al domingo. Así, μ_X será la media de la

demanda del sábado y f_Y la función de densidad de la variable Y . Además a , b y c serán el número de concentrados con 0, 1 y 2 días de vida restantes respectivamente.

De este modo, siendo hoy viernes gastaré $\min\{a, X\}$ unidades a punto de caducar, $\min\{b, (X - a)^+\}$ unidades a las que les queda un día de vida y $\min\{c, (X - (a + b))^+\}$ unidades de las que les quedan 2 días de vida. Por tanto mañana sábado tendré $(b - (X - a)^+)^+$ de las que hoy les queda 1 día de vida y $(c - (X - (a - (a + b))^+)^+$ de las de 2 y así tiraré $((b - (X - a)^+)^+ - Y)^+$. De la misma forma que antes, esta cantidad no la conocemos y por tanto produciremos:

$$E((b - (X - a)^+)^+ - Y)^+ = \int_{-\infty}^a \int_{-\infty}^b ((b - (x - a)^+)^+ - y)^+ f_X(x) f_Y(y) dx dy \quad (1.5)$$

por ser X e Y independientes. Como:

$$(b - (x - a)^+)^+ = \begin{cases} b & \text{si } x \leq a \\ a + b - x & \text{si } a < x \leq a + b \\ 0 & \text{si } x > a + b \end{cases}$$

tenemos

$$\begin{aligned} E((b - (X - a)^+)^+ - Y)^+ &= \int_{-\infty}^a f_X(x) \left(\int_{-\infty}^b (b - y) f_Y(y) dy \right) dx + \\ &\int_a^{a+b} f_X(x) \left(\int_{-\infty}^{a+b-x} (a + b - x - y) f_Y(y) dy \right) dx + \int_{a+b}^{\infty} f_X(x) \left(\int_{-\infty}^0 -y f_Y(y) dy \right) dx. \end{aligned} \quad (1.6)$$

El primer término se puede escribir como $F_X(a)E(b - Y)^+$ y por tanto sabemos calcularlo. El último término es $(1 - F_X(a + b))E(-Y)^+$, que es prácticamente 0 en nuestro caso porque la probabilidad de que Y tome valor negativo es casi nula (de hecho en el caso real es nula) y por tanto $(-Y)^+$ vale casi con total probabilidad 0. El término de en medio es más complicado. En nuestro caso, como la demanda se distribuye normalmente podemos sustituir y obtener:

$$\begin{aligned} \int_a^{a+b} f_X(x) \left(\int_{-\infty}^{a+b-x} (a + b - x - y) f_Y(y) dy \right) dx &= \int_a^{a+b} f_X(x) E(a + b - x - Y)^+ dx = \\ &= \int_a^{a+b} f_X(x) \left((a + b - x - \mu_Y) F_Y(a + b - x) + \frac{\sigma_Y}{\sqrt{2\pi}} e^{-\frac{(a+b-x-\mu_Y)^2}{2\sigma_Y^2}} \right) dx. \end{aligned}$$

Esta última igualdad puede descomponerse en distintos sumandos. Si nos centramos en los que contienen a $F_Y(a + b - x)$ tenemos:

$$\begin{aligned} \int_a^{a+b} f_X(x) F_Y(a + b - x) dx &= \int_a^{a+b} \int_{-\infty}^{a+b-x} f_X(x) f_Y(y) dy dx = \\ &\int_0^b \int_a^{a+b-y} f_X(x) f_Y(y) dx dy + \int_0^{\infty} \int_a^{a+b} f_X(x) f_Y(y) dx dy = \\ &= \int_0^b \int_a^{a+b-y} f_X(x) f_Y(y) dx dy + F_X(a + b) - F_X(a). \end{aligned} \quad (1.7)$$

Sin embargo el primer sumando de esta última igualdad no tiene expresión en términos de forma elemental por tratarse de la función de distribución acumulada de una distribución normal bivalente en una región no rectangular.

Algo similar ocurre con la cantidad de concentrados que se tirarán pasado mañana, es decir, el domingo. A principios del sábado tendremos en el stock $(b - (X - a)^+)^+$ unidades de las que caducan ese día y $(c - (X - (a + b))^+)^+$ de las que caducan el domingo. De esta forma, razonando como antes, a principios del domingo tendremos

$$\left((c - (X - (a + b))^+)^+ - (Y - (b - (X - a)^+)^+)^+ \right)^+ \quad (1.8)$$

concentrados que caducan ese día. Por tanto la cantidad esperada que deberemos producir para suplir las caducadas ese día es:

$$E \left(\left((c - (X - (a + b))^+)^+ - (Y - (b - (X - a))^+)^+ \right)^+ - Z \right)^+ . \quad (1.9)$$

De nuevo esta esperanza no tiene una expresión matemática y solo se puede aproximar numéricamente para cada valor de (a, b, c) y cada conjunto de distribuciones (X, Y, Z) . Puesto que hacer esto para cada día supondría un coste computacional muy alto (que por ejemplo no nos permitiría realizar simulaciones luego para validar el modelo), optamos por algo más sencillo: introducimos la esperanza dentro de la parte positiva. De esta forma por ejemplo sustituimos $E(a - X)^+$ por $(a - EX)^+$ y $E((b - (X - a))^+ - Y)^+$ por $((b - (EX - a))^+ - EY)^+$.

Aunque evidentemente se comete un error al realizar las sustituciones, la parte entera de la cantidad anterior, con los valores a , b y c (cantidades a punto de caducar) que aparece en la aplicación real de nuestro modelo, casi nunca difieren y si o hacen, es en una unidad, lo que no tiene influencia en los resultados finales. La sustitución permite sin embargo una mayor rapidez en las simulaciones que se usan para validar el modelo. Además, esta sustitución permitirá la estimación de las cantidades que van a caducar en situaciones más complejas, como el caso de puentes o vacaciones de Navidad o Semana Santa, donde el periodo sin producción es mayor que dos días.

1.2.2. Simulaciones

Con el modelo ya diseñado, conviene ponerlo a prueba para los distintos escenarios y ver si hemos obtenido alguna mejoría sustancial respecto a los resultados que se vienen obteniendo en el Banco de Sangre con la política actual. A la hora de ponerlo a prueba centraremos nuestra atención en 3 valores importantes: vida media de los concentrados enviados, cantidad de concentrados caducados y número de concentrados faltantes. Hay varias formas de poner a prueba nuestro modelo.

La primera forma consiste en utilizar los mismos datos de 2016 para validar el modelo. Como sabemos, este tipo de validación resulta demasiado optimista y probablemente nos diga que nuestro modelo es mejor de lo que realmente es, pero sin embargo nos puede resultar muy útil para comparar los resultados con los que fue obteniendo el banco de sangre a lo largo de ese año. Puesto que en el banco no se sigue ningún criterio específico para decidir si se fabrica un pool de 5 o de 7 días (a excepción de los que marcan los presupuestos), compararemos los resultados del año con los que obtendríamos fabricando concentrados de 7 y 5 días de caducidad. Consideraremos también que los domingos se trabaja. Los resultados se recogen en la tabla 1.2.

| | Modelo actual | Modelo 7 días | Modelo 5 días |
|---------------------------|---------------|---------------|---------------|
| Número de pools caducados | 143 | 0 | 0 |
| % pools caducados | 2.28 | 0 | 0 |
| Pools faltantes | 34 | 0 | 0 |
| Vida media pools enviados | 2.498 | 2.336 | 2.336 |

Tabla 1.2: Resultados obtenidos con los datos de 2016 para el modelo de producción con domingos.

Nótese que abriendo los domingos, los resultados nos indican que no hay ninguna ganancia irradiando los concentrados que no irradiándolos. También se puede observar que los resultados en todos los casos son mejores que los obtenidos actualmente. Sin embargo recordemos también que estos son los datos con los que se ha desarrollado el modelo y que por tanto pueden ser demasiado optimistas.

Para corregir esto la segunda opción es realizar simulaciones. Para obtenerlas, en la literatura se citan ciertas herramientas o paquetes diseñados ya específicamente para ello [6],[1], mientras que otros optan por programarlas ellos mismos en distintos lenguajes [2],[11], [12]. En nuestro caso utilizaremos una combinación de Excel y R. Partiendo de un conjunto de premisas podemos simular distintas situaciones de demandas y observar cómo se comporta el modelo ante cada situación. Nuestras premisas serán que

las demandas se distribuyen normalmente con la media y la varianza específica de cada día de la semana. Al distribuirse normalmente, puede ser que las demandas tomen valores negativos, algo que resulta irreal y sustituiremos estos valores por 0. También ocurrirá que las demandas sean número reales no enteros. Para evitarlo tomaremos el entero superior más cercano a estos valores. No tomaremos la parte entera del valor para evitar ponérselo fácil al modelo con valores pequeños. Realizamos 50 simulaciones y obtenemos las medias y varianzas de las cantidades en las que estamos interesados. Los resultados se muestran en la tabla 1.3:

| | | Modelo 7 días | Modelo 5 días |
|-------------------------------|--------|---------------|---------------|
| Número de pooles caducados | Mínimo | 0 | 0 |
| | Media | 0 | 0.56 |
| | S.D. | 0 | 2.331 |
| | Maximo | 0 | 14 |
| Pooles faltantes | Mínimo | 0 | 0 |
| | Media | 0.88 | 0.86 |
| | S.D. | 1.965 | 1.938 |
| | Maximo | 8 | 8 |
| Vida media pooles enviados | Mínimo | 2.147 | 2.152 |
| | Media | 2.248 | 2.253 |
| | S.D. | 0.05 | 0.05 |
| | Maximo | 2.377 | 2.38 |

Tabla 1.3: Resultados obtenidos realizando simulaciones con el modelo de producción con domingos.

Como se puede observar, para modelos con concentrados que caducan a los 7 días la caducidad es nula. Para modelos con concentrados de 5 días la caducidad es de media 0.56, pero teniendo en cuenta que en una simulación caducan 14, esto quiere decir que entre todas las otras 49 caducan solo otros 14 concentrados. Las vidas medias con las que se envían los concentrados parecen mantenerse estables puesto que dependen del número de concentrados caducados, que hace que se desechen concentrados viejos, y del aumento de las rupturas de stock (que satisface demanda con concentrados de vida 0).

La tercera forma para validar el modelo es utilizar los datos de 2017 para comprobar el modelo. Existen dos inconvenientes: el primero es que como existe una tendencia creciente en el número de concentrados demandados los nuevos pedidos sean de cantidades bastante superiores a la media de cada día y que por tanto se produzcan más rupturas de stock y la segunda que puesto que estamos en el mismo año 2017, la cantidad de datos sea insuficiente. Como dijimos la demanda aumenta en torno a 500 unidades, lo que de un año a otro haría aumentar la demanda media diaria en 1.37 unidades. Podríamos corregir este efecto sumando esta cantidad a los niveles óptimos del almacén o calculando un nuevo nivel para cada día del año en base a la demanda experimentada el año anterior. Sin embargo, como número es muy pequeño (poco más de una unidad al día), no lo haremos. Respecto a la segunda pega, comentar que disponemos de los datos de los 4 primeros meses, 120 días. Con estos datos se obtienen los resultados de la tabla 1.4.

| | Modelo actual | Modelo 7 días | Modelo 5 días |
|----------------------------------|---------------|---------------|---------------|
| Número de concentrados caducados | 78 | 0 | 0 |
| % concentrados caducados | 5.65 | 0 | 0 |
| Pooles faltantes | 4 | 2 | 2 |
| Vida media concentrados enviados | 3.88 | 2.194 | 2.194 |

Tabla 1.4: Resultados obtenidos con los datos de 2017 para el modelo de producción con domingos.

El aumento en las rupturas del stock está debido principalmente al aumento de la demanda anual, que también contribuye a reducir la vida media con la que los concentrados son enviados a los hospitales.

Conviene también considerar el caso en el que se cierra los domingos pues esto permitiría a algún

empleado librar un día más y no encender la máquina un día, con el consiguiente ahorro económico. Por contra, esto implicaría aumentar el nivel óptimo del almacén de los sábados para incluir la demanda de 3 días. La tabla 1.5 recoge los valores más relevantes.

| | Modelo 7 días | Modelo 5 días |
|---------------------------|---------------|---------------|
| Número de pools caducados | 0 | 83 |
| % pools caducados | 0 | 1.26 |
| Pools faltantes | 0 | 0 |
| Vida media pools enviados | 2.823 | 2.758 |

Tabla 1.5: Resultados obtenidos con los datos de 2016 para el modelo de producción sin domingos.

Aparte de un aumento en la caducidad (antes no se caducaba ninguno), se observa un aumento en la vida media de los concentrados al ser enviados, provocado al arrastrar la sobreproducción algunos sábados.

De la misma forma que antes, se pueden realizar simulaciones para la producción si no se fabrican concentrados los domingos, obteniendo los resultados de la tabla 1.6.

| | | Modelo 7 días | Modelo 5 días |
|---------------------------|--------|---------------|---------------|
| Número de pools caducados | Mínimo | 0 | 10 |
| | Media | 0.36 | 54.62 |
| | S.D. | 1.711 | 28.51 |
| | Maximo | 10 | 123 |
| Pools faltantes | Mínimo | 0 | 0 |
| | Media | 0.8 | 0.8 |
| | S.D. | 2.213 | 2.213 |
| | Maximo | 11 | 25 |
| Vida media pools enviados | Mínimo | 2.583 | 2.55 |
| | Media | 2.732 | 2.691 |
| | S.D. | 0.065 | 0.054 |
| | Maximo | 2.908 | 2.81 |

Tabla 1.6: Resultados de las simulaciones obtenidos para el modelo de producción sin domingos.

También puede compararse el modelo con los datos de 2017.

| | Modelo 7 días | Modelo 5 días |
|----------------------------------|---------------|---------------|
| Número de concentrados caducados | 0 | 6 |
| % concentrados caducados | 0 | 0.27 |
| Pools faltantes | 7 | 7 |
| Vida media concentrados enviados | 2.561 | 2.547 |

Tabla 1.7: Resultados obtenidos con los datos de 2017 para el modelo de producción sin domingos.

Observese que los resultados obtenidos para 2016 con el modelo de producción sin domingos son mejores incluso que los obtenidos para ese año con el modelo que aplica actualmente el banco de sangre. Se caducan 60 concentrados menos, lo que hace reducirse el porcentaje de caducados de un 2.28 % a un 1.26 % y se producen 0 rupturas del stock frente a las 34 del modelo actual. Sin embargo, la vida media con la que los concentrados son transferidos aumenta en 0.26 días (aunque este aumento puede deberse a la demanda que se cubre con concentrados de 0 días de vida que se produce cuando se produce una ruptura del stock).

Una vez presentado y estudiado ya el modelo básico procedemos a observar los parámetros del problema. Es importante comprender cada uno de ellos para determinar si son susceptibles de mejora.

Los datos del problema son las demandas (o las distribuciones de probabilidad de las demandas, según se estudie el problema de optimización con las demandas de 2016 o el problema general) y las vidas de los concentrados. Por tanto, sobre estos datos no hay nada a hacer exceptuando quizás en el caso del problema general encontrar una distribución de probabilidad que se ajuste mejor que la distribución normal a los datos. Recordemos sin embargo que nuestras distribuciones superaban el test de Shapiro-Wilk.

El indicador principal que debe ser optimizado es el número de concentrados faltantes. Como hemos comentado en la sección 2.1, partiendo de la hipótesis de que las demandas se distribuyen normalmente, este indicador depende casi exclusivamente del k que elijamos. Al mismo tiempo, no puede hacerse cero por tratarse de una distribución definida en todo \mathbb{R} . De esta forma solo podemos acotarla por una cierta cantidad que sea el porcentaje de veces que queremos en media que se nos rompa la cadena de suministro como máximo. En nuestro caso elegimos que esta cantidad sea un 0.13 % (casi una vez cada 2 años), lo que nos hace elegir un k igual a 3. Depende ligeramente también del número de concentrados caducados. Los concentrados caducan al final del día y se producen al principio con lo que una posible caducidad de gran parte del stock sería corregida al principio del día siguiente. Sin embargo, los días festivos este hecho no puede ser corregido porque no es posible producir y por tanto puede desencadenar rupturas. En la sección 2.2 tratamos de mitigar esta situación pero de nuevo el carácter aleatorio de las demandas nos impide eliminarlo del todo.

En las simulaciones se puede comprobar que para igual escenario e igual número de concentrados caducados se obtienen iguales números de concentrados caducados.

El siguiente indicador más importante es la caducidad. Dado que cada concentrado de plaquetas cuesta en torno a 400 euros, aunque fuera solo por motivos económicos convendría evitar este derroche. Unido además al componente ético que supone desechar una muestra de altruismo que otra persona ha donado este indicador se sitúa como el segundo más importante. Depende del valor k pero una vez fijado este de forma que evite que se nos rompa la cadena de suministro, depende principalmente de la vida máxima que puedan tener los concentrados, aunque también de nuestra habilidad para determinar la distribución de probabilidad que sigue la demanda de cada día.

La vida media con la que se envían los concentrados a los hospitales depende también de k . La vida máxima de los concentrados y las rupturas de stock pueden también influir en ella haciéndola reducirse pues en un caso se eliminan pooles viejos quedando los más nuevos y en otro directamente la producción de un número menor al que se suele demandar haría que los pooles enviados se acabaran de producir. Sin embargo sería un sinsentido elegir no producir lo suficiente para que todos los pooles enviados fueran nuevos. De este modo la vida media la optimizaremos mediante la optimización de los otros parámetros.

Capítulo 2

2.1. Cambios en el modelo

En la literatura y en la práctica existen numerosas variaciones que pueden afectar a la solución o a la forma misma de plantear el problema. En algunas situaciones, como en la citada en [6] la demanda es tan grande que se deben agrupar las posibles demandas en múltiplos de 5 para poder hacer simulaciones. En nuestro caso esto no es así y, con la potencia de los ordenadores actuales, no nos preocupamos por ello. También puede ser que sea posible producir concentrados de varios tipos y, quizás, para diferentes propósitos. Por ejemplo, a veces los concentrados se dividen en otros de menor volumen para trasfudir a la población pediátrica.

En el caso de estrategias mezclando concentrados de 5 y 7 días (o en un caso más general concentrados de 2 tipos) existe un parámetro sobre el que optimizar: el número de concentrados producidos de cada tipo cada día. Podría darse el caso de que algunos de los concentrados que inactivamos no necesitaran realmente este tratamiento, pues fuesen consumidos antes de que se caducasen. El hecho de que la vida media sea de 2.49 en el modelo implementado actualmente y que la desviación típica de las simulaciones sea en torno a 0.05 nos hace intuir que esto pueda ser así. Como el número de concentrados que tenemos en el almacén un cierto día depende de las demandas de los días anteriores, esta cantidad es también aleatoria. Por tanto no podemos darla como un número fijo y lo haremos como un porcentaje sobre el número de concentrados producidos.

2.1.1. Estrategia óptima

En nuestro caso, el Banco de Sangre y Tejidos de Aragón se encuentra en una transición entre la fabricación de concentrados de 5 días y concentrados de 7 días. Como hemos podido observar en el apartado de simulaciones, lo ideal sería fabricar todos los concentrados de 7 días, lo que haría que en media se caducaran 1.36 concentrados al año si se cierran los domingos y que en ningún caso, si las distribuciones de las demandas se ajustan a lo que hemos considerado, se caduquen concentrados si se decide producir los domingos. Sin embargo, por motivos principalmente presupuestarios, no es posible todavía fabricar todos los concentrados de este tipo y, aunque esperan un incremento en el presupuesto para los próximos años, de momento solo son capaces de fabricar el 50% de los concentrados de 7 días.

Aunque calculan que en unos años si serán capaces de producir el 100%, como hemos comentado anteriormente, puede darse el caso de que esto no sea necesario puesto que muchos de los concentrados serán enviados a los hospitales mucho antes de que lleguen a los 5 días de vida.

Por tanto, debemos plantearnos cuál es el porcentaje óptimo de concentrados de cada tipo que se deben fabricar cada día de la semana. De nuevo como hemos observado en las simulaciones, para el caso en el que se mantiene la producción los domingos, la cantidad media de concentrados caducados al año es de tan solo 0.56 y por tanto no parece interesante estudiar este caso. Si se decide producir todos los domingos no resultaría necesario producir ningún concentrado de 7 días y bastaría con los de 5.

El caso en el que no se trabaja los domingos es más interesante de estudiar. La media de 54.62 concentrados caducados al año es claramente mejorable como demuestran los resultados obtenidos para concentrados de 7 días. Para comenzar escribiremos el problema en forma de un problema de optimización que satisfaga las condiciones de nuestro problema original. Para ello nos basaremos en el modelo

propuesto en [7].

Como en todo problema de optimización, lo primero será definir las variables de decisión y las restricciones. Comenzaremos introduciendo las constantes, que son valores que conocemos. Utilizaremos los datos del año 2016 para la optimización, para así poder comprobar la estrategia con los datos de 2017. Nótese que este año era bisiesto y tenía por tanto 366 días. Por tanto, las constantes corresponden a estos datos y son:

- T_j : Tope de fabricación del j -ésimo día del año. En nuestro caso $T_j = 0$ para lunes y domingos. Los demás días no hay restricciones pero podría modificarse para tener en cuenta topes debidos a falta de materias primas u otras cuestiones.
- d_j : Demanda del j -ésimo día del año.

Puesto que tratamos de encontrar la estrategia óptima de producción, nuestras variables de decisión serán las proporciones α_i de concentrados de 5 días sobre el total fabricados. Toma valores en $[0, 1]$ y al haber 5 días de producción tenemos $i = 1, 2, \dots, 5$. Sin embargo, como nos basaremos en los datos de un año usaremos $\alpha_j = \alpha_i$ si $7|(j - i)$.

Para elegir cuanta cantidad de los 2 tipos se produce cada día introducimos las variables:

- s_j^k : Número de concentrados a los que les quedan k días restantes el día j del año.
- z_j^k : Cantidad de demanda sin cubrir utilizando concentrados de k días restantes el día j .

Con las variables y constantes definidas procedemos a introducir las restricciones que nos permitan definir una cadena de suministro. Deben controlar la cantidad de concentrados que pasan al día siguiente de cada edad y los que se consumen:

$$\begin{aligned}
 s_{j+1}^{-1} &= (s_j^0 - d_j)^+ & \forall j = 1, \dots, 366 \\
 z_j^0 &= (d_j - s_j^0)^+ & \forall j = 1, \dots, 366 \\
 s_{j+1}^0 &= (s_j^1 - z_j^0)^+ & \forall j = 1, \dots, 366 \\
 z_j^1 &= (z_j^0 - s_j^1)^+ & \forall j = 1, \dots, 366 \\
 s_{j+1}^1 &= (s_j^2 - z_j^1)^+ & \forall j = 1, \dots, 366 \\
 z_j^2 &= (z_j^1 - s_j^2)^+ & \forall j = 1, \dots, 366 \\
 s_{j+1}^2 &= (s_j^3 - z_j^2)^+ & \forall j = 1, \dots, 366 \\
 z_j^3 &= (z_j^2 - s_j^3)^+ & \forall j = 1, \dots, 366 \\
 s_{j+1}^3 &= (s_j^4 - z_j^3)^+ & \forall j = 1, \dots, 366 \\
 z_j^4 &= (z_j^3 - s_j^4)^+ & \forall j = 1, \dots, 366 \\
 s_{j+1}^4 &= (s_j^5 - z_j^4)^+ + \alpha_j(T_j - s_{j+1}^0 - s_{j+1}^1 - s_{j+1}^2 - s_{j+1}^3 - s_{j+1}^5 - (s_j^5 - z_j^4)^+)^+ & \forall j = 1, \dots, 366 \\
 z_j^5 &= (z_j^4 - s_j^5)^+ & \forall j = 1, \dots, 366 \\
 s_{j+1}^5 &= (s_j^6 - z_j^5)^+ & \forall j = 1, \dots, 366 \\
 z_j^6 &= (z_j^5 - s_j^6)^+ & \forall j = 1, \dots, 366 \\
 s_{j+1}^6 &= (1 - \alpha_j)(T_j - s_{j+1}^0 - s_{j+1}^1 - s_{j+1}^2 - s_{j+1}^3 - s_{j+1}^5 - (s_j^5 - z_j^4)^+)^+ & \forall j = 1, \dots, 366
 \end{aligned} \tag{2.1}$$

Las restricciones del tipo $s_{j+1}^i = (s_j^i - z_j^i)^+$ permiten controlar los concentrados que siguen en el almacén el día siguiente, un día más próximos a su fecha de caducidad (si la demanda sin cubrir es mayor que los que hay de esa proximidad a la caducidad, se gastan todos y sino el resto pasan al día

siguiente) mientras que las restricciones del tipo $z_j^i = (z_j^{i-1} - s_j^i)^+$ controlan que los concentrados se vayan consumiendo en el orden requerido. La variable s_{j+1}^{-1} controla los concentrados caducados y la variable z_j^6 las rupturas de stock. La función objetivo será de la forma:

$$\text{mín} \quad \sum_{j=1}^{366} c_1 s_j^{-1} + c_2 s_j^6 + c_3 z_j^6 \quad (2.2)$$

Donde las constantes c_1 , c_2 , y c_3 representan los costes de producción de un concentrado, de inactivación de este y de ruptura del stock respectivamente.

Como puede observarse, $s_1^k \forall k = 1, \dots, 7$ no está definido y deberemos inicializar estas variables a un valor. Estos valores pueden ser casi cualquiera siempre que su suma sea del orden de los topes y no haya demasiados viejos, para que puedan ser consumidos antes de caducar. Si además quisiéramos asegurarnos que en la solución no hubiera rupturas de stock, podríamos hacerlo añadiendo una restricción $z_j^6 = 0 \quad \forall i = 1, \dots, 366$.

Aunque la función parte positiva no sea lineal, las restricciones que la contienen pueden dividirse en 4, añadiendo una variable binaria y convirtiéndose así en lineales. De esta forma, $z = (a)^+$ podría escribirse como:

$$\begin{aligned} z &\geq 0 \\ z &\geq a \\ z &\leq 1000t \\ z &\leq 1000(1-t) + a \end{aligned} \quad (2.3)$$

con $t \in \{0, 1\}$ una variable binaria. De esta forma $t = 1 \Leftrightarrow a = 0$ y $t = 0 \Leftrightarrow a = 1$. Se toma 1000 por la dimensión del problema, pero basta con un número más grande que el mayor de los topes. También las restricciones de s_{j+1}^4 deben ser partidas en 2 para salvar la no linealidad de la parte positiva. Denominando s_{j+1}^e a las recién producidas de 5 días y s_{j+1}^v a las procedentes de concentrados de 7 días envejecidos tenemos:

$$\begin{aligned} s_{j+1}^v &= (s_j^5 - z_j^4)^+ \\ s_{j+1}^e &= \alpha_j (T_j - s_{j+1}^0 - s_{j+1}^1 - s_{j+1}^2 - s_{j+1}^3 - s_{j+1}^5 - s_{j+1}^v)^+ \\ s_{j+1}^4 &= s_{j+1}^e + s_{j+1}^v \end{aligned} \quad (2.4)$$

Sin embargo, el problema sigue siendo intratable con los medios actuales puesto que se trata de un problema no lineal debido a la componente α_j que multiplica algunos coeficientes en las restricciones de s_{j+1}^4 y s_{j+1}^6 .

De hecho, aunque se considerara otro problema más simple como el de obtener los niveles óptimos del almacén, uno para cada tipo de concentrado, el problema seguiría siendo intratable debido al número de restricciones que tiene y a tratarse de un problema mixto. Obsérvese que solo por cada restricción de las de (2.1) existen 2 variables, una entera (que puede considerarse real si los topes son enteros y las demandas también) y otra binaria. Incluso para casos seleccionando menos días en lugar de un año o introduciendo restricciones que acerquen el problema relajado a la región de factibilidad, el ordenador no es capaz de obtener una solución. Para hacernos una idea, el problema de encontrar los niveles óptimos del almacén considerando 351 días tiene 7371 variables (3855 enteras y 3504 binarias) y 18277 restricciones, algo que herramientas informáticas para este tipo de problemas, como Gurobi o Gusek, no son capaces de resolver actualmente.

Es por ello que se hace necesario obtener otro método para lograr una solución. Una forma de trabajar con problemas de gran dimensión o no lineales es con los denominados algoritmos bioinspirados. Aprovechamos el programa escrito en R para realizar simulaciones para evaluar las posibles soluciones. Esto nos permitirá olvidarnos de las restricciones (2.1), intrínsecas en el propio programa, para centrarnos únicamente en una población cuyos elementos sean tuplas de los 5 α_i que tomarán valores entre 0 y 1. Reducimos así enormemente el tamaño del problema que pasa a ser un problema de 5 variables.

Por contra, los costes computacionales de evaluar una solución aumentan ya que para cada una el algoritmo debe obtener toda la secuencia de producciones, consumos y caducidades y, tras esto, aplicar una función *fitness* al objetivo.

Puesto que nuestro programa está diseñado para aceptar rupturas de stock y caducidades, deberemos utilizar una función *fitness* que las tenga en cuenta y trate de minimizarlas al máximo. Lo principal es que no se produzcan rupturas de stock, luego asociaremos un coste a esto bastante grande con respecto al coste de caducarse un pool, que será de 300 (la cantidad en euros que cuesta producir un pool). Nosotros elegimos que el coste de ruptura sea de 1000 unidades. Por último, una vez minimizadas las rupturas y la caducidad, minimizaremos el número de concentrados de 7 días fabricados. Penalizamos la producción de estos con 100 unidades.

Para el desarrollo del bioinspirado elegimos el algoritmo de enjambre de partículas o PSO por sus siglas en inglés (*Particle Swarm Optimizatio*) descrito en [8],[5]. Este algoritmo considera el conjunto de posibles soluciones como un espacio donde un número determinado de partículas se mueve con una velocidad variable. Estas, en cada iteración, ocupan una posición obtenida de sumar a su posición anterior la velocidad que poseen. Tras ello actualizan su velocidad atendiendo a 3 factores: la inercia, la memoria de individuo y la memoria colectiva. La inercia se obtiene como un porcentaje de la velocidad a la que se desplazaba la partícula anteriormente. La memoria de cada individuo permite a cada partícula recordar la mejor posición en la que ha estado y desplazarse con una intensidad aleatoria hacia este punto. Las partículas también son capaces de comunicarse, recordando así la mejor posición de todas las partículas a lo largo de las iteraciones y se desplazan en esta dirección también con intensidad aleatoria. Una vez actualizada su velocidad, evalúan la posición en que se encuentran.

Lo elegimos porque permite, con un pequeño número de individuos, obtener resultados bastante buenos, sobre todo cuando el número de dimensiones es reducido. El pequeño número de individuos permite un ahorro en memoria y, sobre todo, un ahorro en el coste computacional que conllevaría realizar una evaluación del fitness de un individuo demasiadas veces.

Un pequeño cambio en una variable α_i puede provocar que se produzca una unidad menos de 7 días, causando quizás así una caducidad en un determinado momento y si este momento es cuando al día siguiente no se produce, una ruptura de stock. Esto indica que la función fitness es muy irregular y escalonada. Para una región del espacio $(a_1, b_1] \times (a_2, b_2] \times (a_3, b_3] \times (a_4, b_4] \times (a_5, b_5]$ la función toma un valor constante y para un intervalo contiguo puede tomar cualquier otro valor. Esto puede hacer que los algoritmos bioinspirados se atasquen en un óptimo local y no sean capaces de salir de allí. De hecho, eso ocurre en nuestro caso.

Existen gran cantidad de modificaciones para evitarlo como las expuestas en [3] pero nosotros utilizaremos la mas sencilla. Añadiremos un nuevo tipo de partículas que se mueven de forma aleatoria por el espacio, sin atender a mejores posiciones. Sin embargo, sí que comunican al resto de partículas cual ha sido la mejor posición que han visitado.

Con este algoritmo, en un número bastante pequeño de iteraciones, obtenemos la solución óptima. Como hemos dicho, esta no será un único valor sin una región. Con el fin de facilitarle las cosas al Banco de Sangre, nos quedaremos con un elemento representativo de esta: $\alpha_1 = 1$, $\alpha_2 = 1$, $\alpha_3 = 1$, $\alpha_4 = 1$ y $\alpha_5 = 0,7$, es decir, producir de martes a viernes todos los concentrados de 5 días y los sábados un 70% de concentrados de 5 días y un 30% de 7 días

Con ella no se producen rupturas ni caducidades y producimos únicamente 894 concentrados de 7 días y el resto de 5 días, lo que hace un 13.93%. Por tanto podría comenzar a aplicarse en la actualidad sin ningún tipo de problema presupuestario y, de hecho, se ahorraría.

Obtener la solución para un escenario concreto no implica que esta solución sea la óptima en un caso más general. Es por ello que, para comprobar como se comporta en otros casos (siempre que las demandas sean normales y con igual media y varianza que en 2016), realizamos simulaciones para estos valores de α_i . De nuevo realizamos 50 simulaciones en los 50 mismos escenarios que en los casos anteriores. Los resultados aparecen reflejados en la tabla 2.1.

Como se puede observar comparando con la tabla de resumen de simulaciones sin producción los domingos del capítulo anterior, los resultados obtenidos con este modelo son iguales respecto a número

| | | Modelo (1,1,1,1,0.7) |
|----------------------------------|--------|----------------------|
| Número de Concentrados caducados | Minimo | 0 |
| | Media | 2.48 |
| | S.D. | 5.052 |
| | Maximo | 18 |
| Concentrados faltantes | Minimo | 0 |
| | Media | 0.8 |
| | S.D. | 2.213 |
| | Maximo | 11 |
| Vida media Concentrado enviados | Minimo | 2.583 |
| | Media | 2.730 |
| | S.D. | 0.065 |
| | Maximo | 2.906 |

Tabla 2.1: Resultados obtenidos realizando simulaciones con el modelo de estrategia óptima.

de rupturas de cadena y vida media de los concentrados enviados que los modelos de producción de 6 o 7 días. Respecto a número de concentrados caducados se observa que la cantidad es en media una unidad mayor que en el modelo de 6 días y 2 unidades mayor que la del modelo de 7 días. Sin embargo, la varianza en el número de caducados en un año es similar a la de 6 días y el máximo es 8 unidades menor.

También comprobamos los resultados utilizando las demandas de 2017 obteniendo la tabla 2.2.

| | Caducados | % Caducados | Faltantes | Vida media |
|----------------------|-----------|-------------|-----------|------------|
| Modelo (1,1,1,1,0.7) | 0 | 0 | 7 | 2.560 |

Tabla 2.2: Resultados obtenidos con el modelo de estrategia óptima para los datos de 2017.

Debemos mencionar que pese a nuestros cálculos para seleccionar el k más adecuado, que nos indicaban que con $k = 3$ asegurábamos una ruptura cada 740 días, los datos de las simulaciones nos muestran algo distinto. En media en un año se producen en torno a 0.8 rupturas de stock, lo que equivale a una cada 457.5 días, casi el doble de veces. Se trata de un problema meramente computacional. Puesto que nuestro proceso para obtener las demandas simuladas era muestrear una distribución normal centrada en la media de cada día de la semana de los datos de 2016 y luego transformar los resultados negativos en ceros, las distribuciones resultantes tienen medias superiores a las de los datos de 2016 y por tanto, los escenarios de las simulaciones son más exigentes en cuanto a las rupturas y menos en cuanto a las caducidades. El hecho de que tomemos el techo de las cantidades que el algoritmo recomienda fabricar también contribuye a amplificar este efecto.

No obstante, si queremos ser más precavidos, siempre podemos aumentar el valor k . Esto no hará variar prácticamente nada la cantidad de concentrados producidos, pero sí el tiempo que estos pasen almacenados. Por tanto, la vida media con la que los concentrados son enviados aumentará, haciendo aumentar la probabilidad de caducarse de los concentrados y reducir la probabilidad de que se rompa el stock. Así por ejemplo, utilizando $k = 3,5$ se obtienen los resultados de la tabla 2.3.

| | | Modelo (1,1,1,1,0.7) con $k = 3,5$ |
|-------------------------------------|--------|------------------------------------|
| Número de Concentrados caducados | Minimo | 0 |
| | Media | 5.560 |
| | S.D. | 8.119 |
| | Maximo | 33 |
| Concentrados faltantes | Minimo | 0 |
| | Media | 0.18 |
| | S.D. | 0.661 |
| | Maximo | 3 |
| Vida media Concentrados enviados | Minimo | 2.799 |
| | Media | 2.945 |
| | S.D. | 0.066 |
| | Maximo | 3.129 |

Tabla 2.3: Resultados obtenidos con el modelo de estrategia óptima utilizando un parámetro $k = 3,5$.

Capítulo 3

3.1. La Herramienta

Puesto que nuestro objetivo es ayudar al Banco de Sangre y Tejidos de Aragón a tomar las decisiones sobre cuantos concentrados de plaquetas producir, necesitamos traducir nuestro trabajo de la forma más clara y sencilla a una herramienta fácil de interpretar. Elegimos la hoja Excel por su extensión en diversos ámbitos y su facilidad de uso. En ella incluiremos diversas hojas que permitirán seguir un registro de las operaciones realizadas en el almacén, así como diversas estadísticas útiles para la/el gerente.

3.1.1. Hoja a hoja

Nuestra Hoja de Cálculo de Microsoft Excel se llamará “Optimizacion Pooles Plaquetas Banco de Sangre” y constará de 5 hojas:

Registro

| Orden | Fecha | Día Semana | Stock | Demanda media | Desv. Típica | Demanda | Tope almacenado | Recomend. Fabric. 5 días | Recomend. Fabric. 7 días | Recomend. extra 7 días | Fabricados 5 días | Fabricados 7 días | Enviados 5 días | Enviados 7 días | resto | Tope fabricación | Faltante | Caducados |
|-------|------------|------------|-------|---------------|--------------|-------------|-----------------|--------------------------|--------------------------|------------------------|-------------------|-------------------|-----------------|-----------------|-------|------------------|----------|-----------|
| 347 | 11/12/2016 | Domingo | 43 | 37,86538 | 6,498985 | 64,62213987 | 22 | 0 | 0 | 0 | 1 | 0 | 15 | 7 | 200 | 0 | 0 | 0 |
| 348 | 12/12/2016 | Lunes | 26 | 18,80769 | 6,498985 | 38,304645 | 0 | 0 | 0 | 0 | 0 | 0 | 22 | 7 | SI | 200 | 0 | 0 |
| 349 | 13/12/2016 | Martes | 18 | 22,13462 | 6,680463 | 42,178009 | 27 | 0 | 0 | 0 | 23 | 4 | 18 | 1 | 200 | 0 | 0 | 0 |
| 350 | 14/12/2016 | Miércoles | 26 | 20,61538 | 5,928177 | 38,399911 | 33 | 0 | 0 | 0 | 27 | 8 | 13 | 2 | 200 | 0 | 0 | 1 |
| 351 | 15/12/2016 | Jueves | 45 | 26,03846 | 6,258991 | 44,815433 | 0 | 0 | 0 | 0 | 17 | 5 | 30 | 6 | 200 | 0 | 0 | 0 |
| 352 | 16/12/2016 | Viernes | 31 | 7,846154 | 5,602467 | 24,653555 | 0 | 0 | 0 | 0 | 14 | 2 | 1 | 3 | 200 | 0 | 0 | 3 |
| 353 | 17/12/2016 | Sábado | 46 | 8,54717 | 4,841988 | 23,073134 | 0 | 0 | 0 | 0 | 12 | 2 | 4 | 0 | 200 | 0 | 0 | 0 |
| 354 | 18/12/2016 | Domingo | 46 | 37,86538 | 8,918919555 | 64,62213987 | 15 | 0 | 0 | 0 | 7 | 0 | 15 | 8 | 200 | 0 | 0 | 0 |
| 355 | 19/12/2016 | Lunes | 34 | 18,80769 | 6,498985 | 38,304645 | 0 | 0 | 0 | 0 | 10 | 0 | 36 | 3 | SI | 200 | 0 | 0 |
| 356 | 20/12/2016 | Martes | 14 | 22,13462 | 6,680463 | 42,178009 | 29 | 0 | 0 | 0 | 23 | 5 | 12 | 5 | 200 | 0 | 0 | 0 |
| 357 | 21/12/2016 | Miércoles | 25 | 20,61538 | 5,928177 | 38,399911 | 14 | 0 | 0 | 0 | 26 | 9 | 19 | 6 | 200 | 0 | 0 | 0 |
| 358 | 22/12/2016 | Jueves | 26 | 26,03846 | 6,258991 | 44,815433 | 16 | 0 | 0 | 0 | 21 | 12 | 25 | 7 | 200 | 0 | 0 | 0 |
| 359 | 23/12/2016 | Viernes | 34 | 7,846154 | 5,602467 | 24,653555 | 0 | 0 | 0 | 0 | 8 | 19 | 3 | 0 | 200 | 0 | 0 | 0 |
| 360 | 24/12/2016 | Sábado | 58 | 8,54717 | 4,841988 | 23,073134 | 0 | 0 | 0 | 0 | 8 | 10 | 9 | 1 | 200 | 0 | 0 | 0 |
| 361 | 25/12/2016 | Domingo | 67 | 37,86538 | 8,918919555 | 64,62213987 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 200 | 0 | 0 | 0 |
| 362 | 26/12/2016 | Lunes | 61 | 18,80769 | 6,498985 | 38,304645 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 8 | SI | 200 | 0 | 0 |
| 363 | 27/12/2016 | Martes | 47 | 22,13462 | 6,680463 | 42,178009 | 0 | 0 | 0 | 0 | 2 | 0 | 2 | 17 | 200 | 0 | 0 | 0 |
| 364 | 28/12/2016 | Miércoles | 30 | 20,61538 | 5,928177 | 38,399911 | 0 | 0 | 0 | 0 | 10 | 8 | 14 | 13 | 200 | 0 | 0 | 0 |
| 365 | 29/12/2016 | Jueves | 25 | 26,03846 | 6,258991 | 44,815433 | 20 | 0 | 0 | 0 | 19 | 14 | 24 | 6 | 200 | 0 | 0 | 1 |
| 366 | 30/12/2016 | Viernes | 32 | 7,846154 | 5,602467 | 24,653555 | 0 | 0 | 0 | 0 | 7 | 14 | 15 | 6 | 200 | 0 | 0 | 4 |
| 367 | 31/12/2016 | Sábado | 45 | 8,54717 | 4,841988 | 23,073134 | 0 | 0 | 0 | 0 | 11 | 17 | 3 | 0 | 200 | 0 | 0 | 0 |
| 368 | 01/01/2017 | Domingo | 70 | 37,86538 | 8,918919555 | 64,62213987 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0 | 0 | 0 |
| 369 | 02/01/2017 | Lunes | 78 | 18,80769 | 6,498985 | 38,304645 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SI | 200 | 0 | 0 |
| 370 | 03/01/2017 | Martes | 70 | 22,13462 | 6,680463 | 42,178009 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0 | 0 | 0 |
| 371 | 04/01/2017 | Miércoles | 57 | 20,61538 | 5,928177 | 38,399911 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0 | 0 | 0 |
| 372 | 05/01/2017 | Jueves | 32 | 26,03846 | 6,258991 | 44,815433 | 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0 | 0 | 0 |
| 373 | 06/01/2017 | Viernes | 17 | 7,846154 | 5,602467 | 24,653555 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0 | 0 | 0 |
| 374 | 07/01/2017 | Sábado | 0 | 8,54717 | 4,841988 | 23,073134 | 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0 | 0 | 0 |
| 375 | 08/01/2017 | Domingo | 0 | 37,86538 | 8,918919555 | 64,62213987 | 65 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0 | 0 | 0 |
| 376 | 09/01/2017 | Lunes | 0 | 18,80769 | 6,498985 | 38,304645 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SI | 200 | 0 | 0 |
| 377 | 10/01/2017 | Martes | 0 | 22,13462 | 6,680463 | 42,178009 | 43 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0 | 0 | 0 |
| 378 | 11/01/2017 | Miércoles | 0 | 20,61538 | 5,928177 | 38,399911 | 39 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0 | 0 | 0 |
| 379 | 12/01/2017 | Jueves | 0 | 26,03846 | 6,258991 | 44,815433 | 45 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0 | 0 | 0 |
| 380 | 13/01/2017 | Viernes | 0 | 7,846154 | 5,602467 | 24,653555 | 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0 | 0 | 0 |
| 381 | 14/01/2017 | Sábado | 0 | 8,54717 | 4,841988 | 23,073134 | 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0 | 0 | 0 |
| 382 | 15/01/2017 | Domingo | 0 | 37,86538 | 8,918919555 | 64,62213987 | 65 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0 | 0 | 0 |

Figura 3.1: Captura de pantalla de la hoja Registro

La hoja principal es la denominada “Registro”. En ella se recogen día a día los concentrados producidos y enviados de cada tipo. También indica las recomendaciones de producción dependiendo del día de la semana y de los posibles topes en la producción de futuros días. Estos topes, así como los posibles días festivos, se pueden especificar en la misma tabla.

En principio tiene espacio para 3000 días, hasta el 17/03/2024 pero puede extenderse cuanto se quiera (Excel permite 1048576 filas).

A cada día le asocia un identificador (“Orden”) y el día de la semana al que corresponde. Estos datos son fijos y de identificación, y se encuentran coloreados en azul oscuro.

Los datos en un azul un poco más claro son los resultantes del proceso que desarrolla internamente el algoritmo pero que se ha considerado hacer visibles. Estos son:

- Stock: Cuenta el número de concentrados de plaquetas que hay en el almacén al principio del día.
- Demanda media: Muestra la demanda media seleccionada (puede elegirse desde la hoja “Datos Estadísticos”) para ese día de la semana.
- Desv. Típica demanda: Muestra la desviación típica seleccionada para ese día de la semana.
- Tope almacenadas: Es el número de concentrados que deberemos tener en el almacén para asegurarnos llegar al próximo día de producción con una probabilidad de 1/1000 de ruptura del stock.
- Recomend. Fabric. 5 días: Muestra la recomendación que hace el algoritmo para la producción de concentrados de 5 días de caducidad para satisfacer la demanda de ese día.
- Recomend. Fabric. 7 días: Muestra la recomendación que hace el algoritmo para la producción de concentrados de 7 días de caducidad para satisfacer la demanda de ese día.
- Recomend. Extra 7 días: Muestra la recomendación que hace el algoritmo para la producción de concentrados de 7 días para satisfacer la demanda de futuros días en los que se prevea que no va a haber suficientes bolsas de sangre para fabricar concentrados.
- Faltantes: Muestra el número de concentrados que han faltado ese día.
- Caducados: Muestra el número de concentrados que han caducado al final de ese día.

Las celdas en el azul más claro son aquellas en las que debe escribirse y así pues son las únicas que no están protegidas. En las 2 primeras (“Fabricados 5 días” y “Fabricados 7 días”) se deben escribir el número de concentrados de cada tipo que se han fabricado realmente ese día y en las 2 siguientes las enviadas de cada tipo. La columna “Festivo” sirve para indicar si ese día se pueden producir concentrados o no. Por defecto están colocados todos los Lunes como festivos pero si hay alguno más, hay que escribir “SI” en la fila de ese día un día antes de que llegue el día. Si el día no es festivo no hay que escribir nada. La columna “Tope fabricación” sirve para introducir en el algoritmo posibles topes en la fabricación producidos por bajas expectativas de donaciones en esos días (y por tanto falta de materia prima para producir los concentrados). Por defecto están situados en 200 pero deben actualizarse tan pronto como lleguen las predicciones para la siguiente quincena, especialmente en el caso de niveles muy bajos.

Datos Estadísticos

La hoja “Datos Estadísticos” permite obtener estadísticas importantes para el Banco de Sangre hasta ahora imposibles de obtener con las herramientas con que disponían. Aparte del número de pooles producidos, caducados y faltantes, que ya podían ser obtenidos, la herramienta permite obtener la vida media con la que los concentrados son enviados a los hospitales, así como la media y la desviación típica de la demanda dependiendo del día de la semana.

Consta de 2 tablas. La tabla verde permite obtener resúmenes estadísticos procedentes de un periodo de tiempo concreto. Solo se encuentran desprotegidas las celdas “Día Inicio”, para indicar a partir de que día queremos las estadísticas, y “Número últimos días”, para indicar cuantos días en adelante a partir del día de inicio queremos para nuestra estadística. El resto de casillas, como indican sus nombres, nos

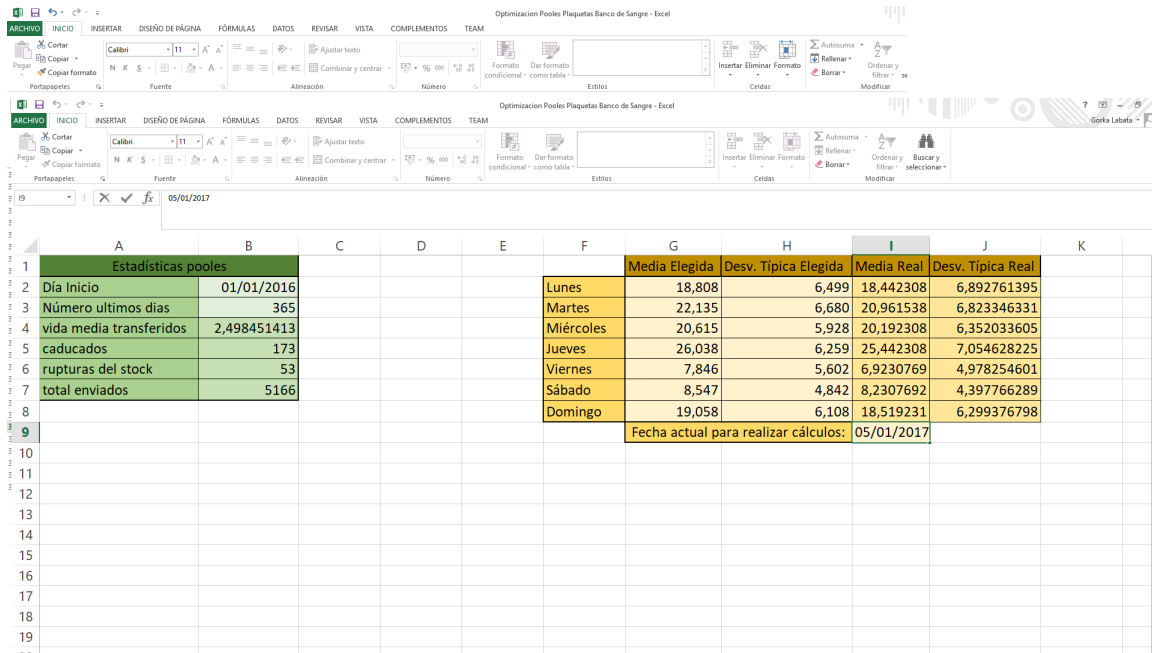


Figura 3.2: Captura de pantalla de la hoja Datos Estadísticos

dan la vida media con la que son enviados los concentrados a los hospitales y el número de concentrados caducados, faltantes y enviados en ese periodo de tiempo.

La tabla naranja tiene 4 columnas, agrupadas por colores en 2 pares. Las celdas más claritas sirven para introducir los datos de medias y desviaciones típicas de cada día de la semana elegidas para ser utilizadas en la tabla “Registro”. Las más oscuras son los valores reales obtenidos para el último año a partir de la fecha indicada en "Fecha actual para realizar cálculos".

vidas pools

La hoja “vidas pools” lleva un registro detallado del número de concentrados que hay cada día en el almacén según el tipo y el número de días que les restan para caducar.

Información de uso

Las explicaciones de cómo utilizar la hoja, aparte de ser entregadas en un PDF, se incluyen en la hoja “Información de Uso”, con el fin de que aunque se perdiera el PDF, se siguiera teniendo acceso a las instrucciones.

Cálculos

Existe una última hoja, denominada “Cálculos”, que realiza los cálculos necesarios para a hoja “Datos Estadísticos”, aunque se encuentra oculta y protegida. Los cálculos se llevaron a esta hoja aparte con el fin de mejorar la presentación.

Muchas de las hojas se encuentran total o parcialmente protegidas para evitar posibles borrados de fórmulas o alteraciones que provoquen errores de cálculo o pérdidas de funcionamiento.

3.1.2. Modo de Empleo

Como hemos comentado al comienzo de este capítulo, al archivo se le pueden dar varios usos:

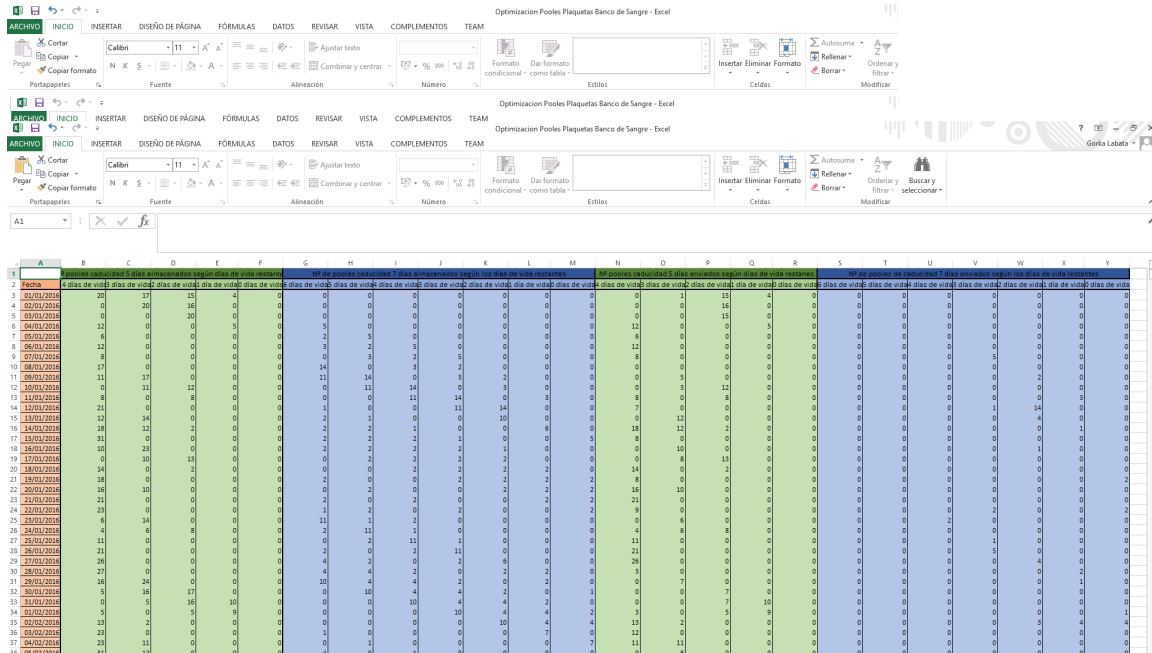


Figura 3.3: Captura de pantalla de la hoja vidas pools

Realizar una predicción sobre el número de concentrados a fabricar

La función principal de la aplicación es servir como herramienta para ayudar en la toma de decisión de cuántos concentrados fabricar cada día. Esto se realiza principalmente sobre la hoja Registro".

Para empezar conviene cercionarse que las columnas “Demanda media” y “Desv. Típica demanda” están actualizadas y se asemejan a las reales. Esto puede comprobarse en la hoja “Datos Estadísticos” en la tabla naranja. Salvo en casos excepcionales conviene que los parámetros elegidos se parezcan a los reales. Para ello se selecciona en la casilla “Fecha actual para realizar cálculos” la fecha en la que nos encontramos y se observan los valores de las columnas. Si algún valor elegido difiere mucho del valor real conviene actualizarlo.

Con los parámetros ya actualizados, el siguiente paso es comprobar que las celdas “Festivo” están actualizadas para el próximo día, esto es, si mañana es festivo, asegurarse de que en la casilla “Festivo” de mañana hay escrito un “SI” (ha de poner “SI” ambas con mayúscula). Si se da el caso y hay más días festivos seguidos a mañana, han de escribirse en este momento, para que el algoritmo pueda tenerlos en cuenta y producir suficiente para esos días.

También ha de comprobarse que las casillas “Tope fabricación” están actualizadas al menos para los 2 próximos días.

Con las restricciones actualizadas, se procede a introducir lo que son propiamente los datos del día. Estos datos deben introducirse justo antes de comenzar la producción del día.

Se deben rellenar las 4 casillas amarillas de la Figura 3.1. Se comienza introduciendo los datos de envíos en las casillas correspondientes al día anterior. Deberemos introducir el número de concentrados de cada tipo enviados desde el día anterior justo después de empezar a producir hasta el día actual justo antes de empezar a producir (hasta el momento en el que nos encontremos). Si el día anterior no se produjo porque era festivo, se eligen los concentrados enviados desde una hora que usualmente sea a la que se producen.

Tras esto, el algoritmo ofrecerá una previsión del número de concentrados que va a ser necesario producir de cada tipo y tomándolo como ayuda o referencia, elegimos el número de concentrados de cada tipo que queremos producir y los anotamos en las casillas de fabricados de hoy.



Figura 3.4: Captura de pantalla de la hoja Información de Uso

Obtener estadísticas de los envíos

Una función secundaria del archivo es obtener estadísticas de nuestros envíos. Estas se obtienen en la hoja "Datos Estadísticos". Aparte de poder obtener las demandas medias reales y la desviación típica real de las demandas para cada día de la semana como hemos indicado en el apartado anterior de esta misma sección, la tabla verde nos permite calcular otros valores para periodos de tiempo a definir.

Seleccionando el "Día inicio" como el primer día a contar en las estadísticas y el número de días en adelante que queremos que entren en estas, podemos obtener el número de concentrados faltantes, caducados y producidos en ese periodo de tiempo, así como la vida media con la que los pooles son enviados a los hospitales.

Así, por ejemplo, si queremos conocer estos datos para el periodo de tiempo comprendido entre el 2 de enero de 2016 hasta el 27 de enero de ese mismo año escribiremos en la casilla "Día inicio" el valor "02/01/2016" y en la casilla "Número últimos días" el valor 25.

3.2. Conclusiones y trabajos futuros

Numerosos trabajos se realizan en este campo cada año, pese al desconocimiento de los bancos de sangre del país. En España, puesto que cada comunidad autónoma tiene su sistema de gestión con sus características específicas, los modelos deben adaptarse al caso que se dé.

En el caso de regiones grandes donde el envío de concentrados desde el centro de producción a los hospitales resulta demasiado costoso para realizarse todos los días o cuando en caso de emergencia, enviar un pool a un determinado hospital requiriese demasiado tiempo se construyen almacenes intermedios. No se trata del caso de Aragón pero sí de otras comunidades como el País Vasco, donde existen 5 almacenes y se ha estudiado este caso. Desde el centro de producción no se suministra directamente a los hospitales sino que todos los concentrados pasan primero por un almacén y de allí, como en el caso de Aragón a los hospitales, quienes piden según la necesidad de cada día. Los hospitales piden los concentrados a un determinado almacén y los almacenes al centro productor.

La forma de considerar este nuevo problema es similar al caso sin almacenes. Lo único que cambia es la división de las demandas. A parte de considerar la demanda al centro productor por parte de los almacenes, consideramos también las demandas a los almacenes por parte de los Hospitales a cada

almacén. Así pues existen 6 series de demandas a considerar.

Cada día el centro produce la demanda media total de ese día mas 3 veces su desviación típica. Cada almacén se rellena con concentrados nuevos procedentes del centro productor hasta un tope de la demanda media que recibe el almacén mas 1,5 veces su desviación típica si se trata de un almacén grande (con mucha demanda) o 0 veces su desviación típica si se trata de un almacén pequeño. Puesto que suma de distribuciones normales se distribuye normalmente con media la suma de las medias y varianza la suma de las varianzas, es muy probable que sobre algún concentrado. Este se guarda en un pequeño almacén situado en el hospital.

A la hora de repartir, se sigue la siguiente política. Los almacenes se llenan cada día hasta su respectivo tope con concentrados recién producidos. Sin embargo, si un almacén se queda sin concentrados en algún momento y necesita más, se los pide al centro de producción. De esta forma, aunque haya algún concentrado sin usar y a punto de caducar en otro almacén, se utilizan los del pequeño almacén del centro productor (pero la política de distribución y los topes bajos de los almacenes hacen que esta última situación casi no se dé). Aunque se caduquen más concentrados, logísticamente es mucho más simple. Si en el almacén del centro productor no quedan concentrados, estos son pedidos a otros almacenes, buscando el más viejo de entre ellos.

Los sistemas con almacenes no logran tan buenos resultados como los sin almacenes respecto a caducidades debido en gran parte a la desconexión entre almacenes. Puesto que la mayoría de las caducidades se concentran en el almacén del centro de producción, una forma de mejorar esto sería el permitir que a los almacenes se enviaran concentrados más viejos, de los del almacén del centro. Sin embargo, esto podría acarrear tiranteces con los gerentes de los almacenes, que podrían adoptar estrategias para evitar que les tocaran los viejos (como pedir de los últimos, para que los viejos ya se hubiesen repartido) obligando a otros a recibirlos siempre.

Este tipo de cuestiones y tiranteces entre productores, almacenes y hospitales, aunque no han sido consideradas en el trabajo son importantes y pueden llegar a dar lugar a conflictos.

Quedan para el futuro también posibles extensiones del estudio de los topes en la producción impuestos por la falta de donaciones determinados días. En nuestro caso, tras preveer que se iba a dar el caso, incrementábamos la producción el día anterior con concentrados de 7 días. Sin embargo, un estudio de otras estrategias posibles podría realizarse para determinar en que casos bastaría con concentrados de 5 días o qué ocurriría si se produce un largo periodo con escasez de donaciones.

Otra posible extensión podría ser calcular un nivel de almacén exacto para el cual decidir producir los domingos, algo que actualmente el Banco de Sangre y Tejidos de Aragón hace basandose en la experiencia e intuición.

Un posible estudio de las incidencias que producen los casos en los que es necesaria sangre de un determinado tipo sanguíneo también podría realizarse.

Sin embargo nuestro modelo es un modelo lo bastante complejo como para ajustarse a la realidad con bastante fidelidad. Permite reducir las caducidades a 0 tanto si se desea producir todos los concentrados de 7 días (por motivo médicos) como si se quiere comenzar a utilizar con el presupuesto actual (menos del 25% de los concentrados se producirían de 7 días). Las rupturas de stock se reducirían al mínimo deseado, un mínimo que podría ser mucho menor que el valor actual con total facilidad. Además, la vida media con la que los concentrados se enviarían a los hospitales sería menor, mejorando la calidad de las transfusiones y así del servicio médico.

Anexos

3.3. Anexo I

Este anexo recoge el código del análisis estadístico de los datos proporcionados por el Banco de Sangre y Tejidos de Aragón.

```
summary(SerieEnvios[,3])
var(SerieEnvios[,3])
shapiro.test(SerieEnvios[,3])
sum(SerieEnvios[,3]) #son 5 menos que el numero de columnas de
#"Datos" porque 5 caen en 2015
envios<-ts(SerieEnvios[,3],start=2016, end = 2017,frequency = 365)
plot(envios)#parece ruido
acf(envios, lag.max = 50)#mucha relaci?n con lo ocurrido la semana
#anterior. Tambien con lo ocurrido hace 2, 3, 4 y 5 dias. Sugiere
#estacionalidad.
envios<-ts(SerieEnvios[3:362,3],start=1, end = 52,frequency = 7)
library(forecast)
seasonplot(envios)#los viernes y s?bados se piden menos y los jueves
#mas (hay que tener en cuenta que en este modelo el jueves llega
#hasta las 15:00 del viernes)
acf(envios, lag.max = 50)
sabados<-vector(mode = 'numeric', length = 53)
for(i in 1:53){
  sabados[i]=2+(i-1)*7
}
domingos<-vector(mode = 'numeric', length = 52)
for(i in 1:52){
  domingos[i]=3+(i-1)*7
}
lunes<-vector(mode = 'numeric', length = 52)
for(i in 1:52){
  lunes[i]=4+(i-1)*7
}
martes<-vector(mode = 'numeric', length = 52)
for(i in 1:52){
  martes[i]=5+(i-1)*7
}
miercoles<-vector(mode = 'numeric', length = 52)
for(i in 1:52){
  miercoles[i]=6+(i-1)*7
}
```

```

jueves<-vector(mode = 'numeric', length = 52)
for(i in 1:52){
  jueves[i]=7+(i-1)*7
}
viernes<-vector(mode = 'numeric', length = 52)
for(i in 1:52){
  viernes[i]=1+(i-1)*7
}
summary(SerieEnvios[sabados,3])
var(SerieEnvios$nEnvios[sabados])
summary(SerieEnvios[domingos,3])
var(SerieEnvios$nEnvios[domingos])
summary(SerieEnvios[lunes,3])
var(SerieEnvios$nEnvios[lunes])
summary(SerieEnvios[martes,3])
var(SerieEnvios$nEnvios[martes])
summary(SerieEnvios[miercoles,3])
var(SerieEnvios$nEnvios[miercoles])
summary(SerieEnvios[jueves,3])
var(SerieEnvios$nEnvios[jueves])
summary(SerieEnvios[viernes,3])
var(SerieEnvios$nEnvios[viernes])
boxplot(nEnvios~DiaSemana,data=SerieEnvios,at=c(7,4,1,2,3,6,5))#Lo
#que ocurre los viernes y sabados es claramente distinto de lo que
#ocurre el resto de la semana. Quizás considerar diferencias tambien
#en toda la semana, sobre todo diferenciando el jueves. Aunque las
#medias son bastante parecidas (la mas alta y la mas baja difieren en
#4 puntos)las varianzas de lunes y martes si que son bastante mayores
#que las del resto de dias.Puede que estas diferencias se deban a
#puentes. Viernes y sábado se diferencian sobre todo en la varianza.
which((SerieEnvios[lunes,3])>35)
SerieEnvios$Dias[51*7+4]#lunes despues de navidad, probablemente
#muchos hospitales el 25 no pidieran.
SerieEnvios$nEnvios[51*7+2]
SerieEnvios$nEnvios[51*7+3]#el domingo se pidió muy poo. De echo, es
#el mínimo de la categoría.
which((SerieEnvios[domingos,3])>35)
SerieEnvios$Dias[17*7+3]#dia del trabajo? Quizás dieron fiesta el 2?
SerieEnvios$nEnvios[17*7+2]#en la media
SerieEnvios$nEnvios[17*7+4]#en la media igual
which((SerieEnvios[jueves,3])<10)
SerieEnvios$Dias[7*4]#Ni idea
SerieEnvios$nEnvios[4*7+1]#en la media
SerieEnvios$nEnvios[3*7+6]#al parecer se pidieron muchos el
#miércoles, no se la causa.
shapiro.test(SerieEnvios$nEnvios[lunes])
shapiro.test(SerieEnvios$nEnvios[martes])
shapiro.test(SerieEnvios$nEnvios[miercoles])
shapiro.test(SerieEnvios$nEnvios[jueves])
shapiro.test(SerieEnvios$nEnvios[viernes])
shapiro.test(SerieEnvios$nEnvios[sabados])

```

```

shapiro.test(SerieEnvios$Envios[domingos])
#parece que dia a dia tambien se distribuyen normalmente. Prestar
#atención al viernes, con un p-valor pequeño.

#parece que dia a dia tambien se distribuyen normalmente. Prestar
#atención al viernes, con un p-valor pequeño.
medias<-c(mean(SerieEnvios$Envios[lunes]),
          mean(SerieEnvios$Envios[martes]),
          mean(SerieEnvios$Envios[miercoles]),
          mean(SerieEnvios$Envios[jueves]),
          mean(SerieEnvios$Envios[viernes]),
          mean(SerieEnvios$Envios[sabados]),
          mean(SerieEnvios$Envios[domingos]))
EnvEstac<-envios-c(mean(SerieEnvios$Envios[domingos]),
                  rep(medias,51))
plot(EnvEstac)#parece ruido blanco
acf(EnvEstac)
pacf(EnvEstac)

a<-1/14
filtro<-rep(a,14)
filCR<-filter(envios, filtro,sides=2)
plot(envios)
lines(filCR,col=2)
abline(v=seq(1,52,by=2),lty=2,col=3)# Quizas una estacionalidad cada
#2 meses
decompose(envios)
plot(decompose(envios))
#no parece observarse una tendencia. No podemos usar la tendencia que
#sale, ni la estacionalidad por ser dependientes del tiempo. Busco
#otra:
plot(stl(envios, s.window="periodic", t.window=14))#sigue sin valer,
#queremos una tendencia lineal, o al menos no "aleatoria".
m_envios = t(matrix(data = envios[1:357], nrow = 7))
seasonal_envios = colMeans(m_envios, na.rm = T)#Calculo la
#estacionalidad como la media de los días de la semana.
plot(as.ts(rep(seasonal_envios,10)))
EnvSinEstac<-envios-c(rep(seasonal_envios,51),seasonal_envios[1])
plot(EnvSinEstac)
plot(decompose(EnvSinEstac))#como se puede observar, la
#estacionalidad se reduce casi a 0.
acf(EnvSinEstac)
pacf(EnvSinEstac)#Sobrepasan el 0,10 el 4º y el 20º.
##Problema:
acf(decompose(envios)$random[4:355])#Si que tiene dependencia de
#otras variables, cuando en cambio el calculado por nosotros no

```

3.4. Anexo II

Este anexo incluye el código en R del programa utilizado para realizar las simulaciones:

```

for(i in 1:53){
  sabados[i]=2+(i-1)*7
}
domingos<-vector(mode = 'numeric', length = 52)
for(i in 1:52){
  domingos[i]=3+(i-1)*7
}
lunes<-vector(mode = 'numeric', length = 52)
for(i in 1:52){
  lunes[i]=4+(i-1)*7
}
martes<-vector(mode = 'numeric', length = 52)
for(i in 1:52){
  martes[i]=5+(i-1)*7
}
miercoles<-vector(mode = 'numeric', length = 52)
for(i in 1:52){
  miercoles[i]=6+(i-1)*7
}
jueves<-vector(mode = 'numeric', length = 52)
for(i in 1:52){
  jueves[i]=7+(i-1)*7
}
viernes<-vector(mode = 'numeric', length = 52)
for(i in 1:52){
  viernes[i]=1+(i-1)*7
}

topes<-vector(mode="numeric", length = 9)
topes[1]=mean(SerieEnvios[lunes,3])+3*sqrt(var(SerieEnvios[lunes,3]))
topes[2]=mean(SerieEnvios[martes,3])+3*sqrt(var(SerieEnvios[martes,3]))
topes[3]=mean(SerieEnvios[miercoles,3])+3*sqrt(var(SerieEnvios[miercoles,3]))
topes[4]=mean(SerieEnvios[jueves,3])+3*sqrt(var(SerieEnvios[jueves,3]))
topes[5]=mean(SerieEnvios[viernes,3])+3*sqrt(var(SerieEnvios[viernes,3]))
topes[6]=mean(SerieEnvios[sabados,3])+3*sqrt(var(SerieEnvios[sabados,3]))
topes[7]=mean(SerieEnvios[domingos,3])+3*sqrt(var(SerieEnvios[domingos,3]))
topes[9]=mean(SerieEnvios[lunes,3])+mean(SerieEnvios[sabados,3])+
  mean(SerieEnvios[domingos,3])+
  3*sqrt(var(SerieEnvios[lunes,3])+var(SerieEnvios[sabados,3])+
    var(SerieEnvios[domingos,3]))
topes[8]=mean(SerieEnvios[lunes,3])+mean(SerieEnvios[domingos,3])+
  3*sqrt(var(SerieEnvios[lunes,3])+var(SerieEnvios[domingos,3]))
topes=ceiling(topes)
sigma1=var(SerieEnvios[domingos,3])
mu1=mean(SerieEnvios[domingos,3])
sigma2=var(SerieEnvios[sabados,3])
mu2=mean(SerieEnvios[sabados,3])

```

```

ntipos<-2
vidapool<-c(5,7)
alfa<-c(1,1,1,1,0.7)
semilla<-17
set.seed(semilla)
SerieEnvios$NEnvios<-env
medias<-c(mean(SerieEnvios[lunes,3]),mean(SerieEnvios[martes,3]),
          mean(SerieEnvios[miercoles,3]),mean(SerieEnvios[jueves,3]),
          mean(SerieEnvios[viernes,3]),mean(SerieEnvios[sabados,3]),
          mean(SerieEnvios[domingos,3]))
sds<-sqrt(c(var(SerieEnvios[lunes,3]),var(SerieEnvios[martes,3]),
            var(SerieEnvios[miercoles,3]),var(SerieEnvios[jueves,3]),
            var(SerieEnvios[viernes,3]),var(SerieEnvios[sabados,3]),
            var(SerieEnvios[domingos,3])))
nSimula<-50
Caducados<-rep(0,nSimula)
VidaMedia<-rep(0,nSimula)
Rupturas<-rep(0,nSimula)
Enviados<-rep(0,nSimula)
fileConn<-file("SimulacionesAlfas.txt",open = "at")
writeLines(sprintf("OPTIMIZACION POOLES DE PLAQUETAS: \n NÚMERO DE
SIMULACIONES: \t%3d, VIDAS DE LOS POOLES: \t(%2d,%2d).\n
                PRODUCCIONES: \t(%2f,%2f,%2f,%2f,%2f)",
                nSimula, vidapool[1], vidapool[2], alfa[1],
                alfa[2], alfa[3], alfa[4], alfa[5]), fileConn)
Domingos<-F

for (k in 1:nSimula){
  pedido<-rep(0,366)
  for (i in 1:366){
    if ((i+4)%7==0){
      pedido[i]=ceiling(max(c(rnorm(1,medias[7],sds[7]),0)))
    }else{pedido[i]=ceiling(max(c(rnorm(1,medias[(i+4)%7],
                                  sds[(i+4)%7]),0)))}
  }
  SerieEnvios$NEnvios<-pedido
  caducados<-vector(mode = "numeric", length=366)
  faltantes<-rep(-0.5,367)
  fabricados<-matrix(rep(0,366*ntipos),nrow=ntipos, ncol = 366,
                    dimnames = list(c("5 días", "7 días"),
                    as.character(SerieEnvios$Dias)))
  colapooles1<-c(3,rep(2,5))
  colapooles2<-c(rep(3,2),rep(2,6))
  npoolescola<-c(6,8)
  rupturas<-0
  demanda<-0
  vidaAcumulada<-0
  i=1
  for(i in 1:366){
    if(SerieEnvios[i,1]=="martes"){

```

```

if(faltantes[i]==-0.5&&npoolescola[1]!=0&&npoolescola[2]!=0){
  fabricados[1,i]=ceiling(max(topes[2]-sum(npoolescola),0)
    *alfa[1])
  fabricados[2,i]=floor(max(topes[2]-sum(npoolescola),0)
    *(1-alfa[1]))
  colapooles1<-c(colapooles1,rep(1,fabricados[1,i]))
  npoolescola[1]=npoolescola[1]+fabricados[1,i]
  colapooles2<-c(colapooles2,rep(1,fabricados[2,i]))
  npoolescola[2]=npoolescola[2]+fabricados[2,i]
}else if(faltantes[i]==-0.5&&npoolescola[1]==0){
  fabricados[1,i]=ceiling(max(topes[2]-sum(npoolescola),0)
    *alfa[1])
  fabricados[2,i]=floor(max(topes[2]-sum(npoolescola),0)
    *(1-alfa[1]))
  colapooles1<-rep(1,fabricados[1,i])
  npoolescola[1]=npoolescola[1]+fabricados[1,i]
  colapooles2<-c(colapooles2,rep(1,fabricados[2,i]))
  npoolescola[2]=npoolescola[2]+fabricados[2,i]
}else if(faltantes[i]==-0.5&&npoolescola[2]==0){
  fabricados[1,i]=ceiling(max(topes[2]-sum(npoolescola),0)
    *alfa[1])
  fabricados[2,i]=floor(max(topes[2]-sum(npoolescola),0)*
    (1-alfa[1]))
  colapooles1<-c(colapooles1,rep(1,fabricados[1,i]))
  npoolescola[1]=npoolescola[1]+fabricados[1,i]
  colapooles2<-rep(1,fabricados[2,i])
  npoolescola[2]=npoolescola[2]+fabricados[2,i]
}else{
  fabricados[1,i]=ceiling(topes[2]*alfa[1])
  fabricados[2,i]=floor(topes[2]*(1-alfa[1]))
  colapooles1<-rep(1,fabricados[1,i])
  npoolescola[1]=fabricados[1,i]
  colapooles2<-rep(1,fabricados[2,i])
  npoolescola[2]=fabricados[2,i]
}
if(sum(npoolescola)-SerieEnvios$nEnvios[i]<=0){
  faltantes[i+1]=sum(npoolescola)-SerieEnvios$nEnvios[i]
  rupturas=rupturas+SerieEnvios$nEnvios[i]-sum(npoolescola)
  colapooles1<-colapooles1[-c(1:npoolescola[1])]
  colapooles2<-colapooles2[-c(1:npoolescola[2])]
  npoolescola[1]=0
  npoolescola[2]=0
}else{
  demanda<-SerieEnvios[i,3]
  for(j in 1:7){
    while(!is.na(colapooles1[1]))&&
      (vidapool[1]-colapooles1[1]+1)==j&&demanda>0){
      colapooles1<-colapooles1[-1]
      demanda<-demanda-1
      npoolescola[1]<-npoolescola[1]-1
      vidaAcumulada=vidaAcumulada+vidapool[1]-j+1
    }
  }
}

```

```

}
while((!is.na(colapooles2[1]))&&
      (vidapool[2]-colapooles2[1]+1)==j&&demanda>0){
  colapooles2<-colapooles2[-1]
  demanda<-demanda-1
  npoolescola[2]<-npoolescola[2]-1
  vidaAcumulada=vidaAcumulada+vidapool[2]-j+1
}
}
for(j in 1:npoolescola[1]){
  colapooles1[j]=colapooles1[j]+1
  if((is.numeric(colapooles1[j]))&&npoolescola[1]>0&&
      colapooles1[j]==(vidapool[1]+1)){
    caducados[i]=caducados[i]+1
  }
}
if(caducados[i]>0){
  colapooles1<-colapooles1[-c(1:caducados[i])]
  npoolescola[1]=npoolescola[1]-caducados[i]
}
cadu<-caducados[i]
for(j in 1:npoolescola[2]){
  colapooles2[j]=colapooles2[j]+1
  if((is.numeric(colapooles2[j]))&&npoolescola[2]>0&&
      colapooles2[j]==(vidapool[2]+1)){
    caducados[i]=caducados[i]+1
  }
}
if(caducados[i]-cadu>0){
  colapooles2<-colapooles2[-c(1:(caducados[i]-cadu))]
  npoolescola[2]=npoolescola[2]-caducados[i]+cadu
}
}
}
if(SerieEnvios[i,1]=="miercoles"){
  if(faltantes[i]==-0.5&&npoolescola[1]!=0&&npoolescola[2]!=0){
    fabricados[1,i]=ceiling(max(topes[3]-sum(npoolescola),0)
                           *alfa[2])
    fabricados[2,i]=floor(max(topes[3]-sum(npoolescola),0)
                          *(1-alfa[2]))
    colapooles1<-c(colapooles1,rep(1,fabricados[1,i]))
    npoolescola[1]=npoolescola[1]+fabricados[1,i]
    colapooles2<-c(colapooles2,rep(1,fabricados[2,i]))
    npoolescola[2]=npoolescola[2]+fabricados[2,i]
  }else if(faltantes[i]==-0.5&&npoolescola[1]==0){
    fabricados[1,i]=ceiling(max(topes[3]-sum(npoolescola),0)
                           *alfa[2])
    fabricados[2,i]=floor(max(topes[3]-sum(npoolescola),0)
                          *(1-alfa[2]))
    colapooles1<-rep(1,fabricados[1,i])
    npoolescola[1]=npoolescola[1]+fabricados[1,i]
  }
}

```

```

    colapooles2<-c(colapooles2,rep(1,fabricados[2,i]))
    npoolescola[2]=npoolescola[2]+fabricados[2,i]
  }else if(faltantes[i]==-0.5&&npoolescola[2]==0){
    fabricados[1,i]=ceiling(max(topes[3]-sum(npoolescola),0)
      *alfa[2])
    fabricados[2,i]=floor(max(topes[3]-sum(npoolescola),0)
      *(1-alfa[2]))
    colapooles1<-c(colapooles1,rep(1,fabricados[1,i]))
    npoolescola[1]=npoolescola[1]+fabricados[1,i]
    colapooles2<-rep(1,fabricados[2,i])
    npoolescola[2]=npoolescola[2]+fabricados[2,i]
  }else{
    fabricados[1,i]=ceiling(topes[3]*alfa[2])
    fabricados[2,i]=floor(topes[3]*(1-alfa[2]))
    colapooles1<-rep(1,fabricados[1,i])
    npoolescola[1]=fabricados[1,i]
    colapooles2<-rep(1,fabricados[2,i])
    npoolescola[2]=fabricados[2,i]
  }
  if(sum(npoolescola)-SerieEnvios$nEnvios[i]<=0){
    faltantes[i+1]=sum(npoolescola)-SerieEnvios$nEnvios[i]
    rupturas=rupturas+SerieEnvios$nEnvios[i]-sum(npoolescola)
    colapooles1<-colapooles1[-c(1:npoolescola[1])]
    colapooles2<-colapooles2[-c(1:npoolescola[2])]
    npoolescola[1]=0
    npoolescola[2]=0
  }else{
    demanda<-SerieEnvios[i,3]
    for(j in 1:7){
      while(!is.na(colapooles1[1]))
        &&(vidapool[1]+1-colapooles1[1])==j&&demanda>0){
          colapooles1<-colapooles1[-1]
          demanda<-demanda-1
          npoolescola[1]<-npoolescola[1]-1
          vidaAcumulada=vidaAcumulada+vidapool[1]-j+1
        }
      while(!is.na(colapooles2[1]))&&
        (vidapool[2]+1-colapooles2[1])==j&&demanda>0){
          colapooles2<-colapooles2[-1]
          demanda<-demanda-1
          npoolescola[2]<-npoolescola[2]-1
          vidaAcumulada=vidaAcumulada+vidapool[2]-j+1
        }
    }
  }
  for(j in 1:npoolescola[1]){
    colapooles1[j]=colapooles1[j]+1
    if((is.numeric(colapooles1[j]))&&
      npoolescola[1]>0&&colapooles1[j]==(vidapool[1]+1)){
      caducados[i]=caducados[i]+1
    }
  }
}

```



```

if(caducados[i]>0){
  colapooles1<-colapooles1[-c(1:caducados[i])]
  npoolescola[1]=npoolescola[1]-caducados[i]
}
cadu<-caducados[i]
for(j in 1:npoolescola[2]){
  colapooles2[j]=colapooles2[j]+1
  if((is.numeric(colapooles2[j]))&&npoolescola[2]>0
    &&colapooles2[j]==(vidapool[2]+1)){
    caducados[i]=caducados[i]+1
  }
}
if(caducados[i]-cadu>0){
  colapooles2<-colapooles2[-c(1:(caducados[i]-cadu))]
  npoolescola[2]=npoolescola[2]-caducados[i]+cadu
}
}
}
if(SerieEnvios[i,1]=="jueves"){
  if(faltantes[i]==-0.5&&npoolescola[1]!=0&&npoolescola[2]!=0){
    fabricados[1,i]=ceiling(max(topes[4]-sum(npoolescola),0)
      *alfa[3])
    fabricados[2,i]=floor(max(topes[4]-sum(npoolescola),0)
      *(1-alfa[3]))
    colapooles1<-c(colapooles1,rep(1,fabricados[1,i]))
    npoolescola[1]=npoolescola[1]+fabricados[1,i]
    colapooles2<-c(colapooles2,rep(1,fabricados[2,i]))
    npoolescola[2]=npoolescola[2]+fabricados[2,i]
  }else if(faltantes[i]==-0.5&&npoolescola[1]==0){
    fabricados[1,i]=ceiling(max(topes[4]-sum(npoolescola),0)
      *alfa[3])
    fabricados[2,i]=floor(max(topes[4]-sum(npoolescola),0)
      *(1-alfa[3]))
    colapooles1<-rep(1,fabricados[1,i])
    npoolescola[1]=npoolescola[1]+fabricados[1,i]
    colapooles2<-c(colapooles2,rep(1,fabricados[2,i]))
    npoolescola[2]=npoolescola[2]+fabricados[2,i]
  }else if(faltantes[i]==-0.5&&npoolescola[2]==0){
    fabricados[1,i]=ceiling(max(topes[4]-sum(npoolescola),0)
      *alfa[3])
    fabricados[2,i]=floor(max(topes[4]-sum(npoolescola),0)
      *(1-alfa[3]))
    colapooles1<-c(colapooles1,rep(1,fabricados[1,i]))
    npoolescola[1]=npoolescola[1]+fabricados[1,i]
    colapooles2<-rep(1,fabricados[2,i])
    npoolescola[2]=npoolescola[2]+fabricados[2,i]
  }else{
    fabricados[1,i]=ceiling(topes[4]*alfa[3])
    fabricados[2,i]=floor(topes[4]*(1-alfa[3]))
    colapooles1<-rep(1,fabricados[1,i])
    npoolescola[1]=fabricados[1,i]
  }
}

```

```

colapooles2<-rep(1,fabricados[2,i])
npoolescola[2]=fabricados[2,i]
}
if(sum(npoolescola)-SerieEnvios$nEnvios[i]<=0){
  faltantes[i+1]=sum(npoolescola)-SerieEnvios$nEnvios[i]
  rupturas=rupturas+SerieEnvios$nEnvios[i]-sum(npoolescola)
  colapooles1<-colapooles1[-c(1:npoolescola[1])]
  colapooles2<-colapooles2[-c(1:npoolescola[2])]
  npoolescola[1]=0
  npoolescola[2]=0
}else{
  demanda<-SerieEnvios[i,3]
  for(j in 1:7){
    while(!is.na(colapooles1[1]))&&
      (vidapool[1]+1-colapooles1[1])==j&&demanda>0){
      colapooles1<-colapooles1[-1]
      demanda<-demanda-1
      npoolescola[1]<-npoolescola[1]-1
      vidaAcumulada=vidaAcumulada+vidapool[1]-j+1
    }
    while(!is.na(colapooles2[1]))&&
      (vidapool[2]+1-colapooles2[1])==j&&demanda>0){
      colapooles2<-colapooles2[-1]
      demanda<-demanda-1
      npoolescola[2]<-npoolescola[2]-1
      vidaAcumulada=vidaAcumulada+vidapool[2]-j+1
    }
  }
  for(j in 1:npoolescola[1]){
    colapooles1[j]=colapooles1[j]+1
    if((is.numeric(colapooles1[j]))&&
      npoolescola[1]>0&&colapooles1[j]==(vidapool[1]+1)){
      caducados[i]=caducados[i]+1
    }
  }
  if(caducados[i]>0){
    colapooles1<-colapooles1[-c(1:caducados[i])]
    npoolescola[1]=npoolescola[1]-caducados[i]
  }
  cadu<-caducados[i]
  for(j in 1:npoolescola[2]){
    colapooles2[j]=colapooles2[j]+1
    if((is.numeric(colapooles2[j]))&&
      npoolescola[2]>0&&colapooles2[j]==(vidapool[2]+1)){
      caducados[i]=caducados[i]+1
    }
  }
  if(caducados[i]-cadu>0){
    colapooles2<-colapooles2[-c(1:(caducados[i]-cadu))]
    npoolescola[2]=npoolescola[2]-caducados[i]+cadu
  }
}

```

```

}
}
if(SerieEnvios[i,1]=="viernes"){
  if(faltantes[i]==-0.5&&npoolescola[1]!=0&&npoolescola[2]!=0){
    fabricados[1,i]=ceiling(max(topes[5]-sum(npoolescola),0)
      *alfa[4])
    fabricados[2,i]=floor(max(topes[5]-sum(npoolescola),0)
      *(1-alfa[4]))
    colapooles1<-c(colapooles1,rep(1,fabricados[1,i]))
    npoolescola[1]=npoolescola[1]+fabricados[1,i]
    colapooles2<-c(colapooles2,rep(1,fabricados[2,i]))
    npoolescola[2]=npoolescola[2]+fabricados[2,i]
  }else if(faltantes[i]==-0.5&&npoolescola[1]==0){
    fabricados[1,i]=ceiling(max(topes[5]-sum(npoolescola),0)
      *alfa[4])
    fabricados[2,i]=floor(max(topes[5]-sum(npoolescola),0)
      *(1-alfa[4]))
    colapooles1<-rep(1,fabricados[1,i])
    npoolescola[1]=npoolescola[1]+fabricados[1,i]
    colapooles2<-c(colapooles2,rep(1,fabricados[2,i]))
    npoolescola[2]=npoolescola[2]+fabricados[2,i]
  }else if(faltantes[i]==-0.5&&npoolescola[2]==0){
    fabricados[1,i]=ceiling(max(topes[5]-sum(npoolescola),0)
      *alfa[4])
    fabricados[2,i]=floor(max(topes[5]-sum(npoolescola),0)
      *(1-alfa[4]))
    colapooles1<-c(colapooles1,rep(1,fabricados[1,i]))
    npoolescola[1]=npoolescola[1]+fabricados[1,i]
    colapooles2<-rep(1,fabricados[2,i])
    npoolescola[2]=npoolescola[2]+fabricados[2,i]
  }else{
    fabricados[1,i]=ceiling(topes[5]*alfa[4])
    fabricados[2,i]=floor(topes[5]*(1-alfa[4]))
    colapooles1<-rep(1,fabricados[1,i])
    npoolescola[1]=fabricados[1,i]
    colapooles2<-rep(1,fabricados[2,i])
    npoolescola[2]=fabricados[2,i]
  }
}
if(sum(npoolescola)-SerieEnvios$nEnvios[i]<=0){
  faltantes[i+1]=sum(npoolescola)-SerieEnvios$nEnvios[i]
  rupturas=rupturas+SerieEnvios$nEnvios[i]-sum(npoolescola)
  colapooles1<-colapooles1[-c(1:npoolescola[1])]
  colapooles2<-colapooles2[-c(1:npoolescola[2])]
  npoolescola[1]=0
  npoolescola[2]=0
}else{
  demanda<-SerieEnvios[i,3]
  for(j in 1:7){
    while(!is.na(colapooles1[1]))&&
      ((vidapool[1]+1-colapooles1[1])==j)&&demanda>0){
      colapooles1<-colapooles1[-1]
    }
  }
}
}
}

```

```

    demanda<-demanda-1
    npoolscola[1]<-npoolscola[1]-1
    vidaAcumulada=vidaAcumulada+vidapool[1]-j+1
  }
  while(!(is.na(colapooles2[1]))&&
        ((vidapool[2]+1-colapooles2[1])==j)&&demanda>0){
    colapooles2<-colapooles2[-1]
    demanda<-demanda-1
    npoolscola[2]<-npoolscola[2]-1
    vidaAcumulada=vidaAcumulada+vidapool[2]-j+1
  }
}
for(j in 1:npoolscola[1]){
  colapooles1[j]=colapooles1[j]+1
  if((is.numeric(colapooles1[j]))&&
      npoolscola[1]>0&&colapooles1[j]==(vidapool[1]+1)){
    caducados[i]=caducados[i]+1
  }
}
if(caducados[i]>0){
  colapooles1<-colapooles1[-c(1:caducados[i])]
  npoolscola[1]=npoolscola[1]-caducados[i]
}
cadu<-caducados[i]
for(j in 1:npoolscola[2]){
  colapooles2[j]=colapooles2[j]+1
  if((is.numeric(colapooles2[j]))&&
      npoolscola[2]>0&&colapooles2[j]==(vidapool[2]+1)){
    caducados[i]=caducados[i]+1
  }
}
if(caducados[i]-cadu>0){
  colapooles2<-colapooles2[-c(1:(caducados[i]-cadu))]
  npoolscola[2]=npoolscola[2]-caducados[i]+cadu
}
}
}
if(SerieEnvios[i,1]=="sabado"){
  if(Domingos){
    if(faltantes[i]==-0.5&&npoolscola[1]!=0&&
        npoolscola[2]!=0){
      fabricados[1,i]=ceiling(max(topes[6]-sum(npoolscola),0)
                              *alfa[5])
      fabricados[2,i]=floor(max(topes[6]-sum(npoolscola),0)
                             *(1-alfa[5]))
      colapooles1<-c(colapooles1,rep(1,fabricados[1,i]))
      npoolscola[1]=npoolscola[1]+fabricados[1,i]
      colapooles2<-c(colapooles2,rep(1,fabricados[2,i]))
      npoolscola[2]=npoolscola[2]+fabricados[2,i]
    }else if(faltantes[i]==-0.5&&npoolscola[1]==0){
      fabricados[1,i]=ceiling(max(topes[6]-sum(npoolscola),0)

```

```

        *alfa[5])
fabricados[2,i]=floor(max(topes[6]-sum(npoolescola),0)
        *(1-alfa[5]))
colapooles1<-rep(1,fabricados[1,i])
npoolescola[1]=npoolescola[1]+fabricados[1,i]
colapooles2<-c(colapooles2,rep(1,fabricados[2,i]))
npoolescola[2]=npoolescola[2]+fabricados[2,i]
}else if(faltantes[i]==-0.5&&npoolescola[2]==0){
fabricados[1,i]=ceiling(max(topes[6]-sum(npoolescola),0)
        *alfa[5])
fabricados[2,i]=floor(max(topes[6]-sum(npoolescola),0)
        *(1-alfa[5]))
colapooles1<-c(colapooles1,rep(1,fabricados[1,i]))
npoolescola[1]=npoolescola[1]+fabricados[1,i]
colapooles2<-rep(1,fabricados[2,i])
npoolescola[2]=npoolescola[2]+fabricados[2,i]
}else{
fabricados[1,i]=ceiling(topes[6]*alfa[5])
fabricados[2,i]=floor(topes[6]*(1-alfa[5]))
colapooles1<-rep(1,fabricados[1,i])
npoolescola[1]=fabricados[1,i]
colapooles2<-rep(1,fabricados[2,i])
npoolescola[2]=fabricados[2,i]
}
}else{
if(faltantes[i]==-0.5&&npoolescola[1]!=0&&
npoolescola[2]!=0){
a=sum(colapooles1==(vidapool[1]-1))+
sum(colapooles2==(vidapool[2]-1))
b=sum(colapooles1==(vidapool[1]-2))+
sum(colapooles2==(vidapool[2]-2))
if(is.na(a)){a=0}
if(is.na(b)){b=0}
w=(a-mu2)*pnorm(a, mean=mu2, sd=sqrt(sigma2))+
sigma2*dnorm(a,mean=mu2, sd=sqrt(sigma2))+
+max(0,max(0,b-max(0,mean(SerieEnvios[sabados,3])-a))
-mean(SerieEnvios[domingos,3]))
fabricados[1,i]=ceiling((max(topes[9]-sum(npoolescola),0)
+ceiling(round(w,1)))*(alfa[5]))
fabricados[2,i]=floor((max(topes[9]-sum(npoolescola),0)
+ceiling(round(w,1)))*(1-alfa[5]))
colapooles1<-c(colapooles1,rep(1,fabricados[1,i]))
npoolescola[1]=npoolescola[1]+fabricados[1,i]
colapooles2<-c(colapooles2,rep(1,fabricados[2,i]))
npoolescola[2]=npoolescola[2]+fabricados[2,i]
}else if(faltantes[i]==-0.5&&npoolescola[1]==0){
a=sum(colapooles1==(vidapool[1]-1))+
sum(colapooles2==(vidapool[2]-1))
b=sum(colapooles1==(vidapool[1]-2))+
sum(colapooles2==(vidapool[2]-2))
if(is.na(a)){a=0}

```

```

if(is.na(b)){b=0}
w=(a-mu2)*pnorm(a, mean=mu2, sd=sqrt(sigma2))+
  sigma2*dnorm(a,mean=mu2, sd=sqrt(sigma2))+
  +max(0,max(0,b-max(0,mean(SerieEnvios[sabados,3])-a))
      -mean(SerieEnvios[domingos,3]))
fabricados[1,i]=ceiling((max(topes[9]-sum(npoolescola),0)
                        +ceiling(round(w,1)))*(alfa[5]))
fabricados[2,i]=floor((max(topes[9]-sum(npoolescola),0)
                       +ceiling(round(w,1)))*(1-alfa[5]))
colapooles1<-rep(1,fabricados[1,i])
npoolescola[1]=npoolescola[1]+fabricados[1,i]
colapooles2<-c(colapooles2,rep(1,fabricados[2,i]))
npoolescola[2]=npoolescola[2]+fabricados[2,i]
}else if(faltantes[i]==-0.5&&npoolescola[2]==0){
a=sum(colapooles1==(vidapool[1]-1))+
  sum(colapooles2==(vidapool[2]-1))
b=sum(colapooles1==(vidapool[1]-2))+
  sum(colapooles2==(vidapool[2]-2))
if(is.na(a)){a=0}
if(is.na(b)){b=0}
w=(a-mu2)*pnorm(a, mean=mu2, sd=sqrt(sigma2))+
  sigma2*dnorm(a,mean=mu2, sd=sqrt(sigma2))+
  +max(0,max(0,b-max(0,mean(SerieEnvios[sabados,3])-a))
      -mean(SerieEnvios[domingos,3]))
fabricados[1,i]=ceiling((max(topes[9]-sum(npoolescola),0)
                        +ceiling(round(w,1)))*(alfa[5]))
fabricados[2,i]=floor((max(topes[9]-sum(npoolescola),0)
                       +ceiling(round(w,1)))*(1-alfa[5]))
colapooles1<-c(colapooles1,rep(1,fabricados[1,i]))
npoolescola[1]=npoolescola[1]+fabricados[1,i]
colapooles2<-rep(1,fabricados[2,i])
npoolescola[2]=npoolescola[2]+fabricados[2,i]
}else{
fabricados[1,i]=ceiling((topes[9])*alfa[5])
fabricados[2,i]=floor((topes[9])*(1-alfa[5]))
colapooles1<-rep(1,fabricados[1,i])
npoolescola[1]=fabricados[1,i]
colapooles2<-rep(1,fabricados[2,i])
npoolescola[2]=fabricados[2,i]
}
}
}
if(sum(npoolescola)-SerieEnvios$nEnvios[i]<=0){
faltantes[i+1]=sum(npoolescola)-SerieEnvios$nEnvios[i]
rupturas=rupturas+SerieEnvios$nEnvios[i]-sum(npoolescola)
colapooles1<-colapooles1[-c(1:npoolescola[1])]
colapooles2<-colapooles2[-c(1:npoolescola[2])]
npoolescola[1]=0
npoolescola[2]=0
}else{
demanda<-SerieEnvios[i,3]
for(j in 1:7){

```

```

while(!(is.na(colapooles1[1]))&&
      (vidapool[1]+1-colapooles1[1])==j&&demanda>0){
  colapooles1<-colapooles1[-1]
  demanda<-demanda-1
  npoolescola[1]<-npoolescola[1]-1
  vidaAcumulada=vidaAcumulada+vidapool[1]-j+1
}
while(!(is.na(colapooles2[1]))&&
      (vidapool[2]+1-colapooles2[1])==j&&demanda>0){
  colapooles2<-colapooles2[-1]
  demanda<-demanda-1
  npoolescola[2]<-npoolescola[2]-1
  vidaAcumulada=vidaAcumulada+vidapool[2]-j+1
}
}
for(j in 1:npoolescola[1]){
  colapooles1[j]=colapooles1[j]+1
  if((is.numeric(colapooles1[j]))&&npoolescola[1]>0&&
      colapooles1[j]==(vidapool[1]+1)){
    caducados[i]=caducados[i]+1
  }
}
if(caducados[i]>0){
  colapooles1<-colapooles1[-c(1:caducados[i])]
  npoolescola[1]=npoolescola[1]-caducados[i]
}
cadu<-caducados[i]
for(j in 1:npoolescola[2]){
  colapooles2[j]=colapooles2[j]+1
  if((is.numeric(colapooles2[j]))&&npoolescola[2]>0&&
      colapooles2[j]==(vidapool[2]+1)){
    caducados[i]=caducados[i]+1
  }
}
if(caducados[i]-cadu>0){
  colapooles2<-colapooles2[-c(1:(caducados[i]-cadu))]
  npoolescola[2]=npoolescola[2]-caducados[i]+cadu
}
}
}
if(SerieEnvios[i,1]=="domingo"){
  if(Domingos){
    if(faltantes[i]==-0.5&&npoolescola[1]!=0&&npoolescola[2]!=0){
      a=sum(colapooles1==(vidapool[1]-1))+
        sum(colapooles2==(vidapool[2]-1))
      if(is.na(a)){a=0}
      w=(a-mu2)*pnorm(a, mean=mu2, sd=sqrt(sigma2))+
        sigma2*dnorm(a, mean=mu2, sd=sqrt(sigma2))
      fabricados[1,i]=ceiling((max(topes[8]-sum(npoolescola),0)
        +ceiling(round(w,1)))*(alfa[6]))
      fabricados[2,i]=floor((max(topes[8]-sum(npoolescola),0)

```

```

                                +ceiling(round(w,1))*(1-alfa[6]))
  colapooles1<-c(colapooles1,rep(1,fabricados[1,i]))
  npoolescola[1]=npoolescola[1]+fabricados[1,i]
  colapooles2<-c(colapooles2,rep(1,fabricados[2,i]))
  npoolescola[2]=npoolescola[2]+fabricados[2,i]
}else if(faltantes[i]==-0.5&&npoolescola[1]==0){
  a=sum(colapooles1==(vidapool[1]-1))+
    sum(colapooles2==(vidapool[2]-1))
  if(is.na(a)){a=0}
  w=(a-mu2)*pnorm(a, mean=mu2, sd=sqrt(sigma2))+
    sigma2*dnorm(a,mean=mu2, sd=sqrt(sigma2))
  fabricados[1,i]=ceiling((max(topes[8]-sum(npoolescola),0)
    +ceiling(round(w,1))*(alfa[6])))
  fabricados[2,i]=floor((max(topes[8]-sum(npoolescola),0)
    +ceiling(round(w,1))*(1-alfa[6])))
  colapooles1<-rep(1,fabricados[1,i])
  npoolescola[1]=npoolescola[1]+fabricados[1,i]
  colapooles2<-c(colapooles2,rep(1,fabricados[2,i]))
  npoolescola[2]=npoolescola[2]+fabricados[2,i]
}else if(faltantes[i]==-0.5&&npoolescola[2]==0){
  a=sum(colapooles1==(vidapool[1]-1))+
    sum(colapooles2==(vidapool[2]-1))
  if(is.na(a)){a=0}
  w=(a-mu2)*pnorm(a, mean=mu2, sd=sqrt(sigma2))+
    sigma2*dnorm(a,mean=mu2, sd=sqrt(sigma2))
  fabricados[1,i]=ceiling((max(topes[8]-sum(npoolescola),0)
    +ceiling(round(w,1))*(alfa[6])))
  fabricados[2,i]=floor((max(topes[8]-sum(npoolescola),0)
    +ceiling(round(w,1))*(1-alfa[6])))
  colapooles1<-c(colapooles1,rep(1,fabricados[1,i]))
  npoolescola[1]=npoolescola[1]+fabricados[1,i]
  colapooles2<-rep(1,fabricados[2,i])
  npoolescola[2]=npoolescola[2]+fabricados[2,i]
}else{
  fabricados[1,i]=ceiling((topes[8])*alfa[6])
  fabricados[2,i]=floor((topes[8])*(1-alfa[6]))
  colapooles1<-rep(1,fabricados[1,i])
  npoolescola[1]=fabricados[1,i]
  colapooles2<-rep(1,fabricados[2,i])
  npoolescola[2]=fabricados[2,i]
}
}
}
if(sum(npoolescola)-SerieEnvios$nEnvios[i]<=0){
  faltantes[i+1]=sum(npoolescola)-SerieEnvios$nEnvios[i]
  rupturas=rupturas+SerieEnvios$nEnvios[i]-sum(npoolescola)
  colapooles1<-colapooles1[-c(1:npoolescola[1])]
  colapooles2<-colapooles2[-c(1:npoolescola[2])]
  npoolescola[1]=0
  npoolescola[2]=0
}else{
  demanda<-SerieEnvios[i,3]

```



```

for(j in 1:7){
  while(!is.na(colapooles1[1]))&&
    (vidapool[1]+1-colapooles1[1])==j&&demanda>0){
    colapooles1<-colapooles1[-1]
    demanda<-demanda-1
    npoolescola[1]<-npoolescola[1]-1
    vidaAcumulada=vidaAcumulada+vidapool[1]-j+1
  }
  while(!is.na(colapooles2[1]))&&
    (vidapool[2]+1-colapooles2[1])==j&&demanda>0){
    colapooles2<-colapooles2[-1]
    demanda<-demanda-1
    npoolescola[2]<-npoolescola[2]-1
    vidaAcumulada=vidaAcumulada+vidapool[2]-j+1
  }
}
for(j in 1:npoolescola[1]){
  colapooles1[j]=colapooles1[j]+1
  if((is.numeric(colapooles1[j]))&&
    npoolescola[1]>0&&colapooles1[j]==(vidapool[1]+1)){
    caducados[i]=caducados[i]+1
  }
}
if(caducados[i]>0){
  colapooles1<-colapooles1[-c(1:caducados[i])]
  npoolescola[1]=npoolescola[1]-caducados[i]
}
cadu<-caducados[i]
for(j in 1:npoolescola[2]){
  colapooles2[j]=colapooles2[j]+1
  if((is.numeric(colapooles2[j]))&&
    npoolescola[2]>0&&colapooles2[j]==(vidapool[2]+1)){
    caducados[i]=caducados[i]+1
  }
}
if(caducados[i]-cadu>0){
  colapooles2<-colapooles2[-c(1:(caducados[i]-cadu))]
  npoolescola[2]=npoolescola[2]-caducados[i]+cadu
}
}
}
if(SerieEnvios[i,1]=="lunes"){
  if(faltantes[i]!=-0.5||sum(npoolescola)==0){
    faltantes[i+1]=SerieEnvios$Envios[i]
    rupturas=rupturas+SerieEnvios$Envios[i]
  }else if(SerieEnvios$Envios[i]>=sum(npoolescola)){
    rupturas=rupturas+(SerieEnvios$Envios[i]-sum(npoolescola))
    faltantes[i+1]=SerieEnvios$Envios[i]-sum(npoolescola)
    colapooles1<-colapooles1[-c(1:npoolescola[1])]
    colapooles2<-colapooles2[-c(1:npoolescola[2])]
    npoolescola[1]=0
  }
}

```

```

    npoolescola[2]=0
  }else{
    demanda<-SerieEnvios[i,3]
    for(j in 1:7){
      while(!is.na(colapooles1[1]))&&
        (vidapool[1]+1-colapooles1[1])==j&&demanda>0){
        colapooles1<-colapooles1[-1]
        demanda<-demanda-1
        npoolescola[1]<-npoolescola[1]-1
        vidaAcumulada=vidaAcumulada+vidapool[1]-j+1
      }
      while(!is.na(colapooles2[1]))&&
        (vidapool[2]+1-colapooles2[1])==j&&demanda>0){
        colapooles2<-colapooles2[-1]
        demanda<-demanda-1
        npoolescola[2]<-npoolescola[2]-1
        vidaAcumulada=vidaAcumulada+vidapool[2]-j+1
      }
    }
    for(j in 1:npoolescola[1]){
      colapooles1[j]=colapooles1[j]+1
      if((is.numeric(colapooles1[j]))
        &&npoolescola[1]>0&&colapooles1[j]==(vidapool[1]+1)){
        caducados[i]=caducados[i]+1
      }
    }
    if(caducados[i]>0){
      colapooles1<-colapooles1[-c(1:caducados[i])]
      npoolescola[1]=npoolescola[1]-caducados[i]
    }
    cadu<-caducados[i]
    for(j in 1:npoolescola[2]){
      colapooles2[j]=colapooles2[j]+1
      if((is.numeric(colapooles2[j]))&&
        npoolescola[2]>0&&colapooles2[j]==(vidapool[2]+1)){
        caducados[i]=caducados[i]+1
      }
    }
    if(caducados[i]-cadu>0){
      colapooles2<-colapooles2[-c(1:(caducados[i]-cadu))]
      npoolescola[2]=npoolescola[2]-caducados[i]+cadu
    }
  }
}
}
Caducados[k]=sum(caducados)
Enviados[k]=sum(SerieEnvios$Envios)
VidaMedia[k]=vidaAcumulada/Enviados[k]
Rupturas[k]=rupturas
}
writeLines(sprintf("-----"), fileConn)

```

```

writeLines(sprintf("VIDA MEDIA: \n"), fileConn)
writeLines(sprintf("Minimo: %3f \t Maximo: %3f \t Media: %3f \t S.D:
                  %3f \n", min(VidaMedia), max(VidaMedia),
                  mean(VidaMedia), sqrt(var(VidaMedia))), fileConn)
writeLines(sprintf("CADUCADOS: \n"), fileConn)
writeLines(sprintf("Minimo: %3f \t Maximo: %3f \t Media: %3f \t S.D:
                  %3f \n", min(Caducados), max(Caducados),
                  mean(Caducados), sqrt(var(Caducados))), fileConn)
writeLines(sprintf("RUPTURAS: \n"), fileConn)
writeLines(sprintf("Minimo: %3f \t Maximo: %3f \t Media: %3f \t S.D:
                  %3f \n", min(Rupturas), max(Rupturas),
                  mean(Rupturas), sqrt(var(Rupturas))), fileConn)
close(fileConn)

```

De forma muy similar, eliminando o sustituyendo las partes referentes a la simulación e introduciendo los datos de 2016 obtenidos desde las hojas excel que nos proporcionó el banco de sangre se obtienen ls estadísticas reales. No se incluyen en este anexo los pasos seguidos para obtener desde una tabla donde se citan una auna todas las bolsas de concentrados estos datos por resultar tediosos y repetitivos.

3.5. Anexo III

En este anexo se incluye el código utilizado para obtener la solución eurística mediante el algoritmo de enjambre de partículas. La función evaluadora es la utilizada para obtener las estadísticas obtenida como se describe al final del anexo anterior.

```

source("EvaluadorEstrategia.R")

semilla<-53
set.seed(semilla)

nIter<-3000
nPartcl<-50
nExplor<-20
w<-0.7
c1<-1.2
c2<-1.2
bloqueo<-150
n<-nPartcl
nbloqueo<-0
nvariables <- 5
l<-rep(0,nvariables)
u<-rep(1,nvariables)
k<-0.5
vmax<-k*(u-1)/2
#Movimiento:
#Inicio la población:
particulas<-matrix(nrow=nPartcl,ncol = nvariables)
for(i in 1:nPartcl){
  for(j in 1:nvariables){
    particulas[i,j]=runif(1,1[j],u[j])
  }
}

```

```

}
}
velocidades<-matrix(nrow=nPartcl,ncol = nvariables)
for(i in 1:nPartcl){
  for(j in 1:nvariables){
    velocidades[i,j]=runif(1,(-vmax[j])/3,vmax[j]/3)
  }
}
mparticula<-matrix(nrow=nPartcl,ncol = nvariables+1)
for(i in 1:nPartcl){
  for(j in 1:nvariables){
    mparticula[i,j]=runif(1,l[j],u[j])
  }
}
mparticula[,nvariables+1]=evaluadora(particulas)
mtotal<-rep(NA,nvariables)
mtotal<-mparticula[which.min(mparticula[,nvariables+1]),]
fitpos<-vector(mode="numeric",length = nPartcl)

for(k in 1:nIter){
  for(i in 1:nPartcl){
    particulas[i,]=particulas[i,]+velocidades[i,]
    for(j in 1:nvariables){
      if(particulas[i,j]<l[j]){
        particulas[i,j]=l[j]
        velocidades[i,j]=0
      }
      if(particulas[i,j]>u[j]){
        particulas[i,j]=u[j]
        velocidades[i,j]=0
      }
    }
  }
}
for(i in 1:(nPartcl-nExplor)){
  velocidades[i,]=w*velocidades[i,]+c1*runif(1,min = 0,max = 1)
  *(mparticula[i,1:nvariables]-particulas[i,])+
  +c2*runif(1,min = 0,max = 1)*(mtotal[1:nvariables]
  -particulas[i,])
  for(j in 1:nvariables){
    if(velocidades[i,j]<(-vmax[j])) velocidades[i,j]=(-vmax[j])
    if(velocidades[i,j]>vmax[j]) velocidades[i,j]=vmax[j]
  }
}
for(i in (nPartcl-nExplor+1):nExplor){
  velocidades[i,]=w*velocidades[i,]+c1*runif(1,min = -1,max = 1)
  for(j in 1:nvariables){
    if(velocidades[i,j]<(-vmax[j])) velocidades[i,j]=(-vmax[j])
    if(velocidades[i,j]>vmax[j]) velocidades[i,j]=vmax[j]
  }
}
fitpos=evaluadora(particulas)

```

```
for(i in 1:nPartcl){
  if(fitpos[i]<mparticula[i,(nvariables+1)]){
    mparticula[i,1:nvariables]=particulas[i,1:nvariables]
    mparticula[i,(nvariables+1)]=fitpos[i]
    if(fitpos[i]<mtotal[(nvariables+1)]){
      mtotal=mparticula[i,]
      nbloqueo=0
    }
  }
}
nbloqueo=nbloqueo+1
if (nbloqueo>=bloqueo){
  break
}
}
mtotal
```


Bibliografía

- [1] ASLLANI, A., CULLER, E., & ETTKIN, L., *A simulation-based apheresis platelet inventory management model*. Transfusion, 2014, 54: 2730-2735.
- [2] BAESLER, F., NEMETH, M., MARTÍNEZ, C., & BASTÍAS, A., *Analysis of inventory strategies for blood components in a regional blood center using process simulation*. Transfusions, 2014, 54: 323-330.
- [3] BANDARUA, S., DEBB, K., *Metaheuristic Techniques*. Decision Sciences, Theory and Practice, 2016 chapter 11: 693-750.
- [4] BELIËN, J., & FORCÉ, H., *Supply chain management of blood products: a literature review*. Eur J Oper Res 2012, 217:116.
- [5] BLONDIN, J., *Particle Swarm Optimization: A Tutorial*. 2009. Disponible en: http://cs.armstrong.edu/saad/csci8100/pso_tutorial.pdf.
- [6] DE KORT, W., JANSSEN, M., KORTBEEK, N., JANSEN, N., VAN DER WAL, J., & VAN DIJK, N *Platelet pool inventory management: theory meets practice*. Transfusions, 2011, 51: 2295-2303.
- [7] HAIJEMA, R., VAN DER WAL, J., & VAN DIJK, N. M., *Blood platelet production: Optimization by dynamic programming and simulation*. Computers & Operations Research, 2007, 34: 760-779.
- [8] PEREIRA, G., *Particle Swarm Optimization*. INESCID & Instituto Superior Tecnico, Porto Salvo, Portugal, 2011, 15.
- [9] PÉREZ VAQUERO, M.A., CARRIÓN ESPEJO, J., & CASADO CALDERÓN, M.S., *Resultados de una encuesta sobre gestión de existencias de componentes sanguíneos a responsables de distribución de los centros de transfusión de España*. Boletín SETS, 2012, 23:82.
- [10] PÉREZ VAQUERO, M. Á., GORRIA, C., LEZAUN, M., LÓPEZ, F. J., MONGE, J., EGUIZABAL, C., & VESGA, M. A., *Optimization of the management of platelet concentrate stocks in the Basque Country using mathematical simulation*. Vox Sanguinis, 2016, 110: 369-375.
- [11] ROUTROY, S., & BHAUSAHEB, N. A., *Evaluation of inventory performance for perishable products through simulation*. IUP Journal of Operations Management, 2010, 9: 71-80.
- [12] RYTIÄ, J. S. & SPENS, K. M., *Using simulation to increase efficiency in blood supply chains*. Management Research News, 2006, 29: 801-829.
- [13] BROEKMEULEN, R. A., & VAN DONSELAAR, K. H. *A heuristic to manage perishable inventory with batch ordering, positive lead-times, and time-varying demand*. Computers Operations Research, 2009, 36(11): 3013-3018.

