

Trabajo de Fin de Máster

Máster en Organización Industrial y Gestión de Empresas

Modelado y resolución de un sistema de planificación de turnos para la asistencia de pacientes dependientes

Autora: Ana Pegado Bardayo

Tutor: Pedro Luis González Rodríguez

Dpto. de Organización Industrial y Gestión de
Empresas I

Escuela Técnica Superior de Ingeniería

Sevilla, 2021



Trabajo de Fin de Máster
Máster en Organización Industrial y Gestión de Empresas

Modelado y resolución de un sistema de planificación de turnos para la asistencia de pacientes dependientes

Autora:

Ana Pegado Bardayo

Tutor:

Pedro Luis González Rodríguez

Catedrático de Universidad

Dpto. de Organización Industrial y Gestión de Empresas I

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2021

Trabajo de Fin de Máster: Modelado y resolución de un sistema de planificación de turnos para la asistencia de
pacientes dependientes

Autor: Ana Pegado Bardayo

Tutor: Pedro Luis González Rodríguez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2021

El Secretario del Tribunal

A mi familia

A mis maestros

Agradecimientos

Agradezco a los profesores del Máster en Organización Industrial y Gestión de Empresa la labor realizada durante este curso tan complejo para todos, así como su implicación y su adaptabilidad, pues pese a haber vivido un año de contratiempos y cambios en la enseñanza, nunca han dejado de motivarnos y generarnos curiosidad por una de las ramas más interesantes y fundamentales de la ingeniería.

Gracias a D. José Luis Caballano, Presidente de la Asociación de Ingenieros de Organización Industrial AINGOI (Asociación Profesional de Ingenieros de Organización de España) por habernos facilitado la identificación del problema y aportado las especificaciones del caso planteado.

En especial, agradezco a D. Pedro Luis González su implicación, sus ganas de enseñar -y de que aprendamos-, y la confianza depositada para sacar adelante este proyecto.

Ana Pegado Bardayo

Sevilla, 2021

Resumen

Este proyecto presenta una alternativa piloto al actual sistema de planificación de visitas en una empresa de Servicios de Atención Domiciliaria. Se diseña e implementa un modelo de programación entera adaptado a las diversas características de de este problema, con un resultado que cumple tres funciones: la asignación de pacientes a cada auxiliar médico, la programación de visitas para un periodo establecido, y la gestión de ausencias por parte de asistentes y demandantes.

Abstract

This project presents a pilot alternative to the current tour scheduling system in a Home Care Services company. An integer programming model adapted to the various characteristics of this problem is designed and implemented, with a result that fulfills three functions: the assignment of patients to each medical assistant, the scheduling of visits for a set period, and the management of patients and attendees' absences.

Índice

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Índice	xiv
Índice de ilustraciones	xvii
1 Justificación y objeto	11
1.1. <i>La importancia de la innovación en organizaciones no industriales</i>	11
1.2. <i>Justificación del Proyecto y objetivo general</i>	12
1.3. <i>Objetivos específicos</i>	12
1.4. <i>Estructura del documento</i>	13
2 Descripción del Problema y Revisión de la literatura	15
2.1 <i>Descripción del problema</i>	15
2.2 <i>Estudio de la problemática</i>	16
2.3 <i>Revisión bibliográfica y problemas relacionados</i>	16
2.4 <i>Conclusión de la revision</i>	19
3 Propuesta de resolución	21
4 Modelado matemático e implementación en Gurobi-Python	25
4.1. <i>Fase I: Modelo de asignación</i>	25
4.1.1. Definición de conjuntos	26
4.1.2. Parámetros	26
4.1.3. Variables	27
4.1.4. Restricciones	27
4.1.5. Función objetivo	30
4.1.6. Modelo completo para la fase I	31
4.2. <i>Fase II: Modelo de planificación</i>	32
4.2.1. Conjuntos	32
4.2.2. Parámetros	33
4.2.3. Variables	33
4.2.4. Restricciones	34
R1.1 <i>Horas servidas</i>	34
4.2.5. Función Objetivo	40
4.2.6. Modelo completo fase II	40
4.3. <i>Fase III: Modelo de replanificación diaria por ausencias.</i>	41
4.3.1. Conjuntos	41
4.3.2. Parámetros	42
4.3.3. Variables	43
4.3.4. Restricciones	43

4.3.5.	Función Objetivo	47
4.3.6.	Modelo complete fase III	48
5.	Pruebas y validación	51
5.1.	<i>Set de datos empleado</i>	51
5.1.1.	Datos de pacientes	51
5.1.2.	Datos de auxiliares	52
5.2.	<i>Evaluación de los resultados obtenidos</i>	53
5.2.1.	Ejecución del modelo básico (fases I y II)	53
5.3.	<i>Limitaciones del modelo</i>	58
6.	Conclusiones	59
6.1.	<i>Consecución de objetivos</i>	59
6.2.	<i>Valoración del desempeño</i>	60
6.3.	<i>Futuras líneas de trabajo</i>	60
	Bibliografía	63

Índice de ilustraciones

Ilustración 1. Clasificación de problemas de programación (Hüseyin E. et al.)	17
Ilustración 2. Ejemplo de distribución para 12 auxiliares y 23 pacientes	22
Ilustración 3. Asignación por distancia euclídea	22
Ilustración 4. Asignación por distancia euclídea	22
Ilustración 5. Asignación por distancia Manhattan	22
Ilustración 6. Secuencia de trabajo	23
Ilustración 7. Ejemplo restricción A3	37
Ilustración 8. Ejemplo de entrada Demanda.xlsx	52
Ilustración 9. Ejemplo de entrada Auxiliares.xlsx	52
Ilustración 10. Ejemplo salida Fase I	53
Ilustración 11. Asignación de auxiliares y pacientes	53
Ilustración 12. Ejemplo de planificación obtenida	54
Ilustración 13. Detalle de planificación obtenida	55
Ilustración 14. Horas mensuales asignadas a cada auxiliar	55
Ilustración 15. Resultados tras aplicar ausencias en Fase III	56
Ilustración 16. Resultados tras aplicar ausencias en Fase III	57

1 JUSTIFICACIÓN Y OBJETO

El proyecto objeto de estudio surge de la necesidad de agilizar la metodología de planificación de visitas de atención domiciliaria actualmente empleada en una organización de asistencia a personas dependientes.

Una persona es considerada dependiente funcional cuando por razones de edad, enfermedad o discapacidad precisa ayuda de otra en actividades cotidianas como vestirse, caminar, comer o asearse. Desde 2006 existe en España la Ley de Dependencia, y de ella deriva un sistema de soporte que, entre otras prestaciones, vela por la protección y atención de las personas dependientes a través de servicios públicos y privados.

Según la Organización Mundial de la Salud (OMS), la atención domiciliaria es aquella modalidad de asistencia programada que lleva al domicilio del paciente, los cuidados y atenciones biopsicosociales y espirituales. La atención se constituye en un servicio planificado ya que es concertado con el usuario.

El calificativo de servicios de ayuda a domicilio, también conocidos como Servicios de Atención Domiciliaria (SAD) consiste en prestar apoyo y cuidados en su domicilio a las personas cuando, por cualquier circunstancia, ya no son autónomas para realizar las actividades básicas de la vida diaria (por lo general son personas mayores, aunque también se presta este servicio a personas que por discapacidad temporal o permanente lo necesiten).

Con el aumento de la esperanza de vida en España y con una proporción de personas mayores de 60 años en alza, la demanda de este tipo de servicios está creciendo (Robinson, Shugure, Fortinsky, & Gruman, 2014), y los sistemas diseñados para la gestión del trabajo y programación de visitas en empresas SAD quedan obsoletos por no estar preparados para abordar grandes volúmenes de datos.

1.1. La importancia de la innovación en organizaciones no industriales

Hablar de innovación no abarca exclusivamente la creación de nuevos productos o servicios, y no debe vincularse únicamente con el panorama industrial. Existen numerosos aspectos dentro de una empresa que pueden verse beneficiados por un proyecto de innovación, y saber invertir en ellos aumentará los beneficios de una organización y dará ventaja frente a sus competidores.

En la edición de 2005 del Manual de Oslo, publicado por la OCDE (Organización para la Cooperación y el Desarrollo Económico) y referente bibliográfico en materia de innovación, se definen concretamente cuatro tipos de innovación:

- En el producto o servicio.

- En el proceso.
- En el marketing.
- A nivel organizacional.

Esta última se define como la implementación de nuevos métodos organizacionales en el negocio, en la organización del trabajo y/o en las relaciones hacia el exterior.

Por desconocimiento de los beneficios que reporta y en ocasiones por rechazo al cambio, la innovación en los aspectos organizacionales tiende a pasar desapercibida en las empresas, especialmente en aquellas de tamaño medio y bajo por contar con menor presupuesto. Esto lleva a encontrarnos numerosas empresas con una gestión del trabajo y de la cadena de valor poco eficientes, donde innovar puede aportar beneficios tanto económicos como sociales.

1.2. Justificación del Proyecto y objetivo general

La motivación de este proyecto surge tras conocer a fondo la metodología de trabajo de una empresa de Servicios de Atención Domiciliaria de una localidad cordobesa. El caso fue presentado D. José Luis Caballano, Presidente de la Asociación de Ingenieros de Organización Industrial AINGOI (Asociación Profesional de Ingenieros de Organización de España), el cual planteó el problema a mi Director del Trabajo Fin de Máster, exponiendo la situación actual. La valoración de la metodología empleada nos dio a entender que podría tratarse de un proceso fácilmente mejorable, lo que nos impulsó en la búsqueda de una primera aproximación a lo que puede ser un nuevo sistema de trabajo.

El objetivo general del proyecto es realizar un sistema de planificación de turnos de trabajo para asistir a personas dependientes, con una casuística muy variada y compleja. Este proyecto busca hallar una solución que simplifique la tarea de gestión de turnos de trabajo, pasando por una mejor asignación de personas dependientes y asistentes para reducir distancias recorridas durante la jornada laboral, una planificación mensual rápida y una gestión de las ausencias ágil.

Mediante un modelo piloto de la potencial alternativa, se pretende estudiar la viabilidad de la semi-automatización de esta tarea, así como ofrecer una propuesta de asignación óptima. El objetivo es ser capaz de crear un modelo fiel a la casuística real que permita evaluar la factibilidad de este nuevo sistema.

Intrínsecamente, se pretende se hace énfasis en la aplicación de la ingeniería de organización industrial en campos menos habituales como en el sector servicios, pero que igualmente pueden llegar a verse favorecidos por la aplicación de técnicas de organización industrial.

1.3. Objetivos específicos

Para conseguir este objetivo general del proyecto vamos a apoyarnos en la consecución de los siguientes objetivos específicos:

- OE1.** Valorar el problema, evaluar áreas de mejora propuestas y buscar similitudes con otros modelos existentes.
- OE2.** Realizar un análisis detallado del problema, simplificar los datos y elaborar una propuesta de mejora adaptada a sus características.
- OE3.** Modelar la solución propuesta e implementarla en Python.
- OE4.** Validar el modelo y evaluar los resultados obtenidos.
- OE5.** Valorar efectos positivos y limitaciones, así como su aplicación a nivel real.
- OE6.** Listar las acciones futuras.

1.4. Estructura del documento

El documento consta de 6 capítulos y un apartado dedicado a recopilar las referencias bibliográficas empleadas. Para ello se ha seguido la siguiente estructura:

- El capítulo 2. “Descripción del problema y revisión de la literatura” recoge las características esenciales de este problema, busca similitudes con modelos ya existentes y los diferentes enfoques de resolución.
- A continuación, el capítulo 3. “Propuesta de resolución” propone unas características de partida para la resolución de una versión piloto simplificada, así como una estructura de desglose y resolución del problema.
- El capítulo 4. “Modelado matemático e implementación en Gurobi-Python” modela los tres problemas a resolver: asignación de pacientes, planificación y re-planificación ante ausencias. Se incluyen para cada caso variables, restricciones, y función objetivo tenidas en cuenta.
- El capítulo 5. “Validación y pruebas” muestra los resultados obtenidos para la simplificación propuesta en el capítulo 3 y realiza las comprobaciones pertinentes para asegurar el correcto funcionamiento del programa.
- Por último, en el capítulo 6. “Conclusiones” se evalúa la consecución de los objetivos específicos propuestos en este capítulo y se plantean las líneas de trabajo que se espera seguir.

2 DESCRIPCIÓN DEL PROBLEMA Y REVISIÓN DE LA LITERATURA

2.1 Descripción del problema

Actualmente la empresa planifica las visitas a usuarios manualmente. Existen alrededor de 250 auxiliares (trabajadores) que prestan asistencia y 600 demandantes (pacientes), con diferentes necesidades y características.

Los demandantes tienen prescrito por técnicos de asuntos sociales entre 23 y 70 horas mensuales en función de sus necesidades, repartidas de lunes a viernes o de lunes a domingo. Cada demandante solicitará una, dos, o tres visitas diarias, que deben tener una duración mínima de 1 hora. Existen casos de pacientes que requieren dos auxiliares simultáneamente.

Algunos demandantes requieren que su auxiliar esté capacitado para el uso de grúas y sondas. Alrededor de 20 auxiliares son especialistas en manejo de grúas, y 30 lo son en alimentación por sonda. Estas especialidades pueden coincidir en el mismo auxiliar.

Existen limitaciones en las jornadas de los asistentes. El día se divide en tres turnos: de 8:00 a 14:00 (mañana), de 14:00 a 18:00 (tarde) y de 18:00 a 22:00 (noche). La jornada laboral no puede superar las 9 horas, trabajando como máximo durante dos turnos al día. Esto implica que, si un demandante requiere 3 visitas, una de ellas no podrá llevarla a cabo su auxiliar asignado. Además, entre el último turno trabajado de un día y el primero del día siguiente deben pasar al menos 12 horas. Es decir, si un asistente finaliza su jornada laboral a las 22:00 el día k , el primer turno del día $k + 1$ no podrá comenzar hasta las 10:00 de la mañana.

La empresa cuenta con cuatro modalidades de contrato para auxiliares:

- De lunes a viernes,
- De martes a sábado,
- Sábados, domingos, lunes y festivos,
- Suplencia de vacaciones.

Cada contrato oscila entre 1 y 35 horas semanales. Estas horas de contrato incluyen, además de las horas de asistencia, los descansos otorgados (15 minutos cada 6 horas consecutivas, 30 minutos en caso de ser 7 horas) y traslados (10 minutos si el paciente se encuentra dentro del pueblo, 15 minutos si reside a las afueras o en alguna pedanía).

Los auxiliares tienen 23 días de vacaciones y 5 de asuntos propios que deben ser previamente autorizados por la empresa. Además, por enfermedad u otras causas justificadas, los auxiliares pueden ausentarse durante uno o varios tramos. En caso de no poder realizar las visitas programadas a lo largo del día, será sustituido por otro auxiliar.

Para determinar los contratos se elabora un cuadrante base que excluye sustituciones y ausencias, y sumando las horas determinan las horas otorgadas a cada auxiliar. Sin embargo, este método acaba siendo inexacto e implica en muchos casos pagar horas que no han llegado a ser trabajadas, así como horas extras si se trabaja un domingo sin que su contrato incluya este día.

Este trabajo abordará tres aspectos mejorables de su actual gestión:

1. La asignación actual de pacientes a auxiliares es aleatoria, de manera que muchos auxiliares tienen que asumir grandes desplazamientos entre paciente y paciente.
2. Para la planificación de turnos, crean un cuadrante por cada auxiliar y tratan de asignar las visitas manualmente, lo que supone una tarea tediosa y automatizable.
3. La gestión de las ausencias puntuales resulta compleja, y el problema crece en periodos vacacionales. Se propone un sistema de suplencias puntuales, adaptable vacaciones y festivos.

2.2. Estudio de la problemática

Las tareas de planificación y gestión de los servicios de la empresa pueden resultar sencillas a primera vista, especialmente cuando tratamos con volúmenes de datos bajos y perfiles estandarizados.

Sin embargo, conforme aumenta el número de personas contratadas y clientes y se cuenta con perfiles diferenciados, acciones como hacer coincidir visitas simultáneas pueden volverse complejas para el planificador.

La casuística de este problema es extensa. Existen pacientes con prescripción de 5 y 7 días por semana. Cada paciente requerirá generalmente una o dos visitas, aunque se dan casos de pacientes que necesitan una tercera. En estos casos y debido a las limitaciones en la jornada laboral de los asistentes, un nuevo auxiliar realizará esta tercera visita.

Cada paciente puede estar asignado a uno, dos, tres, cuatro o hasta 6 auxiliares, dependiendo del régimen de atención que demande:

- Si tiene prescripción de 5 días (de lunes a viernes), se le asignará un auxiliar, o dos si requiere tres visitas al día. Estas cifras se duplican si el paciente necesita ser atendido por dos auxiliares simultáneamente.
- Si tiene prescripción de 7 días, se le asignarán dos auxiliares que se irán alternando, cuatro si se demandan tres visitas al día. En caso de requerir asistencia de dos auxiliares simultáneamente, se le asignarán cuatro o 6.

La asignación de pacientes y auxiliares actual no tiene en cuenta ningún criterio, dándose casos de grandes desplazamientos en auxiliares.

A esto han de sumarse los casos de pacientes con necesidades especiales. Estos requerirán atención por parte de personal concreto, lo que restringe más el problema.

Todo esto, junto con las ausencias previstas, supone una tarea de planificación que, manualmente, requiere varias jornadas laborales y queda sujeta a errores humanos.

2.3. Revisión bibliográfica y problemas relacionados

La Investigación Operativa es una disciplina que se ocupa de la aplicación de métodos analíticos avanzados para ayudar a tomar mejores decisiones. Dentro de la tipología de problemas que aborda, aquellos relacionados con la planificación y asignación de recursos o *Scheduling Problems* tienen gran peso dentro de la literatura.

Los problemas de planificación y asignación de recursos consisten en, dados los recursos y personal de una empresa, lograr organizar el trabajo y los recursos disponibles con el objetivo, generalmente, de abastecer una demanda acordada.

(Hüseyin Özder, Evrencan, & Tamer, 2020) propone la siguiente clasificación de los problemas de planificación.

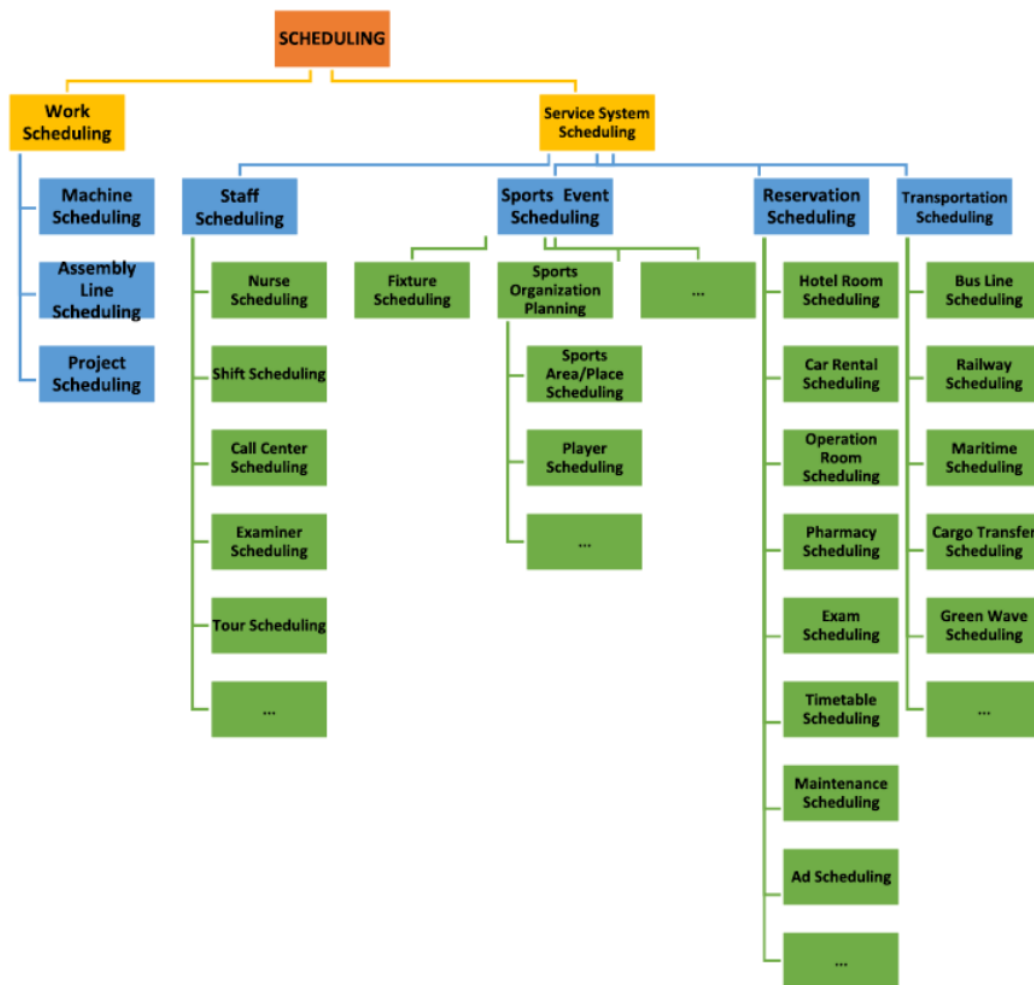


Ilustración 1. Clasificación de problemas de programación (Hüseyin E. et al.)

Dentro del sector servicios se identifican cuatro áreas principales: planificación de personal, planificación de eventos deportivos, planificación de citas y planificación de transporte.

Por las características del problema a abordar en este trabajo, nos centraremos en el primer subgrupo. Este tipo de problemas suele estar sujeto a restricciones horarias, como pueden ser jornadas laborales o duración de un servicio, traslados y elección del modo de transporte, y requisitos ligados a necesidades de la demanda y capacidades del trabajador. Entre las áreas de aplicación de este tipo de problemas destaca el sector sanitario, aunque otras como los servicios de protección y emergencias o la hostelería también pueden verse beneficiados (Ernst, Jiang, Krishnamoorthy, & Sier, 2004)

Dentro de los problemas de planificación de personal existen numerosas categorías u objetivos. Como recaba (Hüseyin Özder, Evrencan, & Tamer, 2020), algunos de ellos son:

- Planificación de días libres. Es el tipo de planificación más simple. Implica la asignación de personal a días concretos, de manera que la demanda (generalmente semanal) sea cubierta en cada periodo al menor coste posible.
- Planificación de turnos. Establecido un horizonte temporal (ciclo), se diseña un calendario que permita satisfacer los requerimientos acordados. Las horas de trabajo en los diferentes días de un ciclo pueden variar.

- Asignación cíclica de personal. Es uno de los problemas de planificación de personal más extendidos. Consiste en minimizar el personal asignado en la planificación de un ciclo de n periodos, de manera que haya suficientes trabajadores en cada periodo para cumplir la demanda y cada trabajador tenga tanto días asignados como libres (Bartholdi, 1980).
- Planificación de tripulación. Es empleado en el sector transportes, y especialmente en aerolíneas, donde el objetivo es seleccionar un conjunto de turnos rotantes de manera que cada segmento de vuelos, trenes, etc. esté cubierto por al menos uno de ellos a coste mínimo (Arabeyre, Fearnley, Steiger, & Teather, 1969).
- Planificación de operarios. El objetivo es especificar una serie de turnos y asignaciones para que las curvas de trabajo se correspondan lo máximo posible con la disponibilidad del operario. La solución a menudo está compuesta por sus horas de inicio y finalización y el número, duración y planificación en el calendario de los periodos no laborables.

En principio, por sus características, el problema objeto de estudio puede ser abordado como un problema de planificación de turnos. Dentro de esta categoría, dependiendo de los condicionantes del problema, los turnos de trabajo a planificar podrán estandarizarse o no. Si se da este segundo caso, será necesario crear un calendario y un horario específico para cada trabajador, sin la posibilidad de emplear una jornada laboral genérica. Esto implica que, conforme crezca el número de trabajadores y la variedad en la demanda, también lo hará la complejidad de resolución del problema, siendo generalmente problemas NP-hard (Hüseyin Özder, Evrencan, & Tamer, 2020).

Existen numerosos enfoques para este tipo de problema, los cuales Alfares H. resume en su artículo (Alfares, 2004). Relacionados con la búsqueda de una solución cuantitativa, podemos considerar las siguientes opciones:

1. Programación entera. En la mayoría de los casos prácticos, el tamaño del problema hace que la programación sea ineficaz para obtener la solución óptima. Para superar esta dificultad puede combinarse esta metodología con el uso de soluciones continuas de arranque avanzadas, cortes, el uso de reglas heurísticas, la adición secuencial de restricciones y la utilización de una estructura concreta para cada problema.
2. Modelado implícito. Para reducir el número de variables, (Bechtold & Jacobs, 1990) y (Thompson, 1995) utilizan modelos con descansos implícitos en la programación de turnos. En lugar de definir una variable para cada turno con un descanso específico, los tipos de turno se consideran variables. Un tipo de turno se identifica por una hora de inicio, una duración de turno y una ventana temporal en la que se establecerá el descanso. Esto implica la reducción en el tamaño del modelo, ya que se utiliza una sola variable para representar todos los turnos del mismo tipo, independientemente de la hora de inicio del descanso.
3. Descomposición. Este enfoque busca dividir un problema de programación entera en un número de subproblemas más sencillos y fáciles de resolver. La división más común consiste en una primera fase de creación de turnos, seguida de una fase de asignación de días libres.
4. Programación por objetivos. Se trata de una técnica de modelado matemático que permite la optimización de un modelo considerando múltiples objetivos priorizables. De esta manera, es posible establecer una primera fase de construcción, y una segunda fase de mejora, según los intereses en cada problema.
5. Generación de *sets* de trabajo. El problema de planificación de turnos se caracteriza por crecer rápidamente conforme se otorga flexibilidad a la programación. El modelado implícito, la descomposición y la generación de sets tratan de abordar el problema reduciendo su tamaño. El objetivo de este tercer enfoque es reducir el tamaño del problema seleccionando un subconjunto de variables de decisión del total de variables del problema.
6. Soluciones basadas en programación lineal. Consiste en aplicar heurísticas de redondeo a soluciones (fraccionales) obtenidas mediante programación lineal. Entre las opciones existentes en la literatura, (Brusco & Johns, 1995) proponen la siguiente: en primer lugar,

se obtiene una solución óptima mediante programación lineal. A continuación, se redondean las variables de decisión enteras y se van añadiendo trabajadores iterativamente hasta llegar a una solución factible. Por último, una vez obtenida una solución inicial, se realiza un proceso de búsqueda tratando de mejorar la solución inicial generada.

7. Construcción y mejora. Consiste en construir la planificación del trabajo de forma iterativa, comenzando por una solución “vacía” sin trabajadores asignados, e incluyéndolos progresivamente en la planificación con cada iteración hasta que se cumplan los requisitos establecidos. A continuación, en la fase de mejora se buscan mejores soluciones a través de heurísticas hasta que se cumpla un criterio de parada establecido.
8. Metaheurísticas. Se trata de un enfoque generalmente más efectivo que los métodos heurísticos en cuanto a la resolución de problemas combinatorios. La principal ventaja de este enfoque es que evita quedar estancado en un óptimo local prematuramente por permitir empeoramientos e infactibilidades en la solución. Las metaheurísticas más empleadas en este tipo de problemas son los Algoritmos Genéticos, el Recocido Simulado y la Búsqueda Tabú.

De esta clasificación de enfoques, el predominante en la literatura es el del modelado y resolución de problemas por programación entera. (Corominas, Lusa, & Pastor, 2004) propone dos modelos para planificar turnos con un horizonte temporal de un año, teniendo en cuenta la estacionalidad de la demanda. Esto implica el diseño de un modelo flexible, que permite abarcar épocas con mayor y menor producción. Más tarde, (Corominas & Pastor, 2009) adapta estos modelos para contemplar también periodos de vacaciones de la plantilla, incluyendo sustituciones y contrataciones temporales para abastecer la demanda en todo momento.

(Rekik, Cordeau, & Soumis, 2010) modela un problema de planificación permitiendo además descansos fraccionables haciendo uso de modelado implícito. Tiene en cuenta la flexibilidad en el comienzo de los turnos y sus pausas, lo cual puede llegar a resultar útil en nuestro caso.

Por último, (Guastaroba, Côté, & Coelho, 2020) propone un modelo MILP (Mixed-integer linear programming) de ruta mínima y programación de tareas de construcción y mantenimiento con características que nos recuerdan a nuestro caso: trabajadores asignados a tareas o máquinas, algunas de ellas requiriendo habilidades especiales, y teniendo en cuenta la distancia al punto de trabajo.

2.4. Conclusión de la revision

Existe un amplio abanico de artículos relacionados con la planificación de personal. El enfoque deductivo de esta revision ha tenido como objetivo clasificar el tipo de problema al que nos enfrentamos, para estudiar así qué soluciones han sido propuestas por los diferentes autores.

De la revisión concluimos que, generalmente, el método más habitual para resolver un problema de asignación y planificación es mediante modelos de programación lineal mixta-entera (MIP), obviamente quedando limitado por la viabilidad del modelo en cuanto al número de variables necesarias. Hay que tener en cuenta que nuestro problema de asignación puede llegar a implicar un gran número de variables y restricciones que pueden ralentizar el proceso de resolución, por lo que hay que tener muy en cuenta los posibles problemas de escalabilidad en la resolución del mismo. Por ello, dentro de los enfoques propuestos, resulta de especial interés la descomposición del problema. En nuestro caso, puede resultar sencillo dividir e individualizar el problema, dando lugar a sub-problemas de comprensión, diseño y resolución simple, por lo que puede suponer una solución eficaz.

Si se trabaja con contratos bien calculados, la planificación de servicios, y en especial la que ocupamos, no requiere la optimización de costes más allá de los relacionados con suplencias. Sin embargo, la planificación manual en sí tal y como se lleva a cabo actualmente en la empresa supone un proceso largo sujeto a errores y que, a fin de cuentas, no aporta valor al servicio ofrecido. Por ello, resulta interesante valorar la opción de automatizar este proceso.

3 PROPUESTA DE RESOLUCIÓN

Tras las conclusiones derivadas de la revisión de la literatura podemos decir que no hay ningún modelo que se adapte fielmente a las especificaciones planteadas. Parece interesante partir de un modelo MILP como en (Guastaroba, Côté, & Coelho, 2020) y tomando ideas de (Corominas & Pastor, 2009) y (Rekik, Cordeau, & Soumis, 2010)

Plantaremos el problema desde un punto de vista académico, pero tratando de ajustarnos al máximo a los condicionantes del problema real.

A diferencia de la situación inicial, este modelo tendrá en cuenta únicamente 3 tipos de contrato, prescindiendo del contrato de suplencia de vacaciones y eliminando del modelo los días festivos para simplificar su diseño. Por tanto, los tipos de contrato tenidos en cuenta son:

- Lunes a Viernes (contrato L-V)
- Martes a Domingo (contrato M-S)
- Sábado, Domingo y Lunes (contrato S-L)

Los pacientes que demanden 5 visitas semanales serán asignados a pacientes con contrato L-V, mientras que los pacientes que requieran 7 días estará asignados a un auxiliar con contrato M-S y a otro con contrato S-L.

Para llegar a soluciones fácilmente interpretables, partiremos de un caso con 44 auxiliares y 120 personas dependientes, manteniendo la razón auxiliares-pacientes similar a la original. En el capítulo 5 abordaremos un caso con un volumen de datos similar al original para evaluar el rendimiento del modelo diseñado con volúmenes mayores.

Además, se mantendrá el porcentaje de especialistas en grúas y sondas, siendo en este caso 4 especialistas en grúas y 5 en sonda.

A falta de datos reales, las ubicaciones de los diferentes auxiliares y pacientes son generadas aleatoriamente en una superficie de 10 unidades de distancia (por ejemplo km) de ancho y 10 de largo, situando a los auxiliares cerca del núcleo urbano y a los pacientes uniformemente distribuidos.

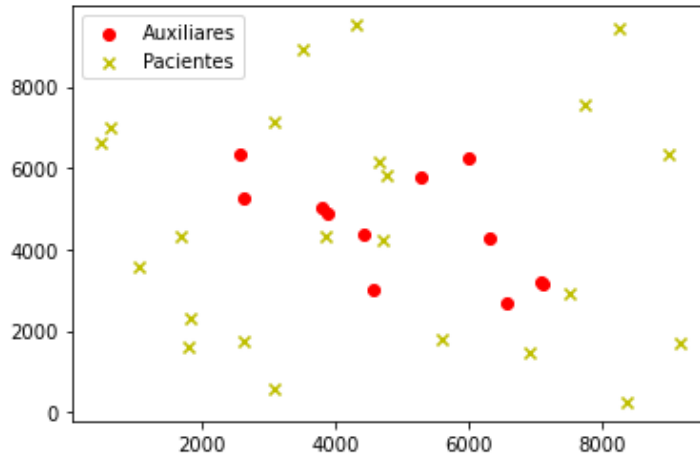


Ilustración 2. Ejemplo de distribución para 12 auxiliares y 23 pacientes

Para la asignación de pacientes se emplearán distancias Manhattan por asemejarse más que las distancias euclídeas a la distancia real que recorre cada trabajador, no obstante quedará habilitada también la posibilidad de trabajar con distancias euclídeas. Como se observa en las ilustraciones 3 y 4, el uso de una u otra condicionará parte de las asignaciones.

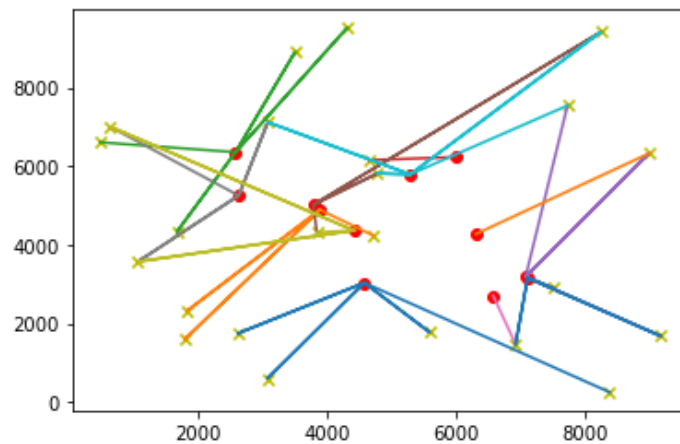


Ilustración 3. Asignación por distancia euclídea

Ilustración 4. Asignación por distancia euclídea

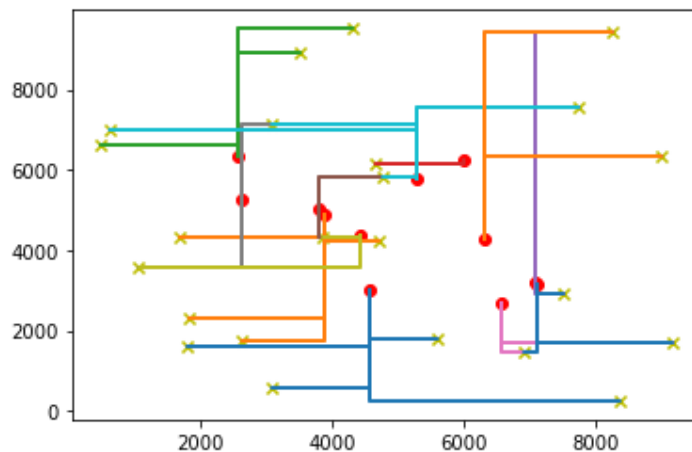


Ilustración 5. Asignación por distancia Manhattan

Actualmente, la empresa gestiona la asignación y planificación simultáneamente y a cuatro meses vista, dando lugar a soluciones poco eficientes.

Tratar de asumir un problema de esta embergadura como un todo puede suponer un número de variables y restricciones excesivo que, si bien puede ser resuelto con un número de datos limitados, puede implicar su no resolubilidad cuando trabajemos con cantidades de datos reales.

Por ello, desglosaremos el problema, secuenciando y simplificando los diferentes subproblemas a los que nos enfrentamos. En este caso abordaremos el problema en tres fases, tal y como representa el esquema de la Ilustración 6.

- La fase I distribuye los pacientes entre los diferentes auxiliares, minimizando la distancia recorrida por cada uno de ellos. Actualmente, cada auxiliar tiene asignados entre uno y cuatro pacientes.
- La fase II crea una planificación de visitas. Por tratarse de un modelo piloto, se establecerá un horizonte temporal de 1 mes, y no 4 como actualmente organiza la empresa.
- Una vez planificado y puesto en marcha en plan mensual, gracias a la fase III se podrán gestionar las ausencias dadas durante el transcurso del mes mediante la replanificación de días concretos en los que se de la ausencia.



Ilustración 6. Secuencia de trabajo

En resultado de la fase I es un listado de dos columnas representando todas las relaciones existentes, indicándose cada una de ellas en una fila diferente.

El resultado de la fase II es un calendario mensual indicando qué visitas hará cada auxiliar durante cada turno de trabajo. Aunque este proyecto no la incluye, una cuarta y última fase será útil para, dentro de cada turno, hacer una programación detallada con el horario y orden de visitas, incluyendo información concreta de horas de comienzo y finalización, traslados y descansos. Existe un segundo output que recoge el total de horas que computarán en el contrato de cada asistente, que puede ser empleado para la generación de contratos.

El resultado de la fase III mantendrá la misma estructura que el calendario mensual obtenido en la fase II, indicando en este caso cuándo se dan sustituciones. Pueden darse ausencias por parte de trabajadores y demandantes, y puntuales (uno o dos turnos) o de día completo.

- Cuando un auxiliar se ausenta un día completo, el resto de trabajadores (respetando especialidades) deben atender sus visitas.
- Cuando un auxiliar se ausenta durante uno o dos turnos, sus visitas se organizan en el turno disponible. Si no fuera posible completar todas, se reparten las visitas con un segundo trabajador.
- Cuando un paciente se ausenta durante uno o dos turnos pero sigue siendo posible satisfacer la demanda de visitas, estas se reorganizan en los turnos disponibles.
- Cuando un paciente se ausenta un día completo o no es posible satisfacer el total de visitas de ese día, las horas correspondientes a dicha visita se pierden.

Pese a no tener en cuenta días festivos y vacaciones, la inclusión de una fase de reprogramación para días concretos puede dar los primeros pasos en la resolución de este problema.

4 MODELADO MATEMÁTICO E IMPLEMENTACIÓN EN GUROBI-PYTHON

Este capítulo recoge los modelos diseñados para cada fase propuesta. En primer lugar, ha de resolverse el primer modelo, que asigna pacientes a auxiliares bajo un criterio dado. A continuación, del segundo modelo se obtendrá la programación de turnos para el mes seleccionado. Por último, si fuera necesario reprogramar un día concreto por ausencias, se aplicará el tercer modelo, que mantendrá la demanda planificada para ese día.

La resolución de los modelos se llevará a cabo mediante el solver comercial Gurobi, que permite la optimización de modelos de programación lineal, programación cuadrática, programación lineal con variables enteras y programación cuadrática con variables enteras. Se hará uso de la interfaz Gurobi Python (gurobipy) por la versatilidad que ofrece en la construcción del modelo frente a las interfaces para otros lenguajes. La interfaz gurobipy permite trabajar tanto con variables y restricciones individuales como con matrices, además de contar con una construcción que permite crear modelos usando una sintaxis muy similar a la empleada en un lenguaje de modelado tradicional.

En cada restricción se incluyen las líneas de código correspondientes a la implementación en Python.

La lectura de datos y escritura de resultados se realizará haciendo uso de las librerías *xlsxwriter* y *openpyxl*, y la posibilidad de trabajar con fechas reales se debe al uso de la librería *calendar*, que permite, entre otras opciones, obtener la estructura semanal para un mes y un año dado. La visualización de asignaciones en el primer modelo se realiza mediante la librería *matplotlib*. El traspaso de resultados de un modelo a otro se lleva a cabo gracias a la librería *pickle*, que permite serializar (convertir en una secuencia de bytes) y deserializar objetos de Python.

4.1. Fase I: Modelo de asignación

Por acuerdo con la empresa, el modelo de asignación minimiza la distancia total que recorre cada auxiliar. Concretamente, se tiene en cuenta la distancia desde el domicilio del auxiliar hasta cada uno de los pacientes, sin tener en cuenta recorridos entre pacientes.

En su implementación, se incluye la posibilidad del cálculo de distancias euclídea y Manhattan, siendo la segunda la empleada en nuestro modelo. Para cada auxiliar i y paciente j , esta distancia la calcularemos como:

$$d_M(i, j) = |x_i - x_j| + |y_i - y_j|$$

Los inputs para la implementación en Python serán dos documentos Excel, recogiendo la información necesaria en columnas para demandantes y auxiliares.

Concretamente, los campos que se incluyen son:

- Auxiliares.xlsx: ID del auxiliar, tipo de contrato, especialidad en grúa, especialidad en sonda.
- Demanda.xlsx: ID del paciente, horas mensuales demandadas, días demandados, visitas diarias demandadas, tiempo de desplazamiento, necesidad de grúa, necesidad de sonda.

4.1.1. Definición de conjuntos

Se trata de un modelo básico, con los siguientes conjuntos y subconjuntos:

$I = \{0, 1, \dots, i, \dots, n_I - 1\}$ el conjunto de trabajadores de la empresa.

$I_{LV} = \{0, 1, \dots, i, \dots, n_{I_{LV}} - 1\} \subset I$ el conjunto de trabajadores con contrato L-S.

$I_{MS} = \{0, 1, \dots, i, \dots, n_{I_{MS}} - 1\} \subset I$ el conjunto de trabajadores con contrato M-S.

$I_{SL} = \{0, 1, \dots, i, \dots, n_{I_{SL}} - 1\} \subset I$ el conjunto de trabajadores con contrato S-L.

$I_{grúa} = \{0, 1, \dots, i, \dots, n_{I_{grúa}} - 1\} \subset I$ el conjunto de trabajadores especialistas en manejo de grúa.

$I_{sonda} = \{0, 1, \dots, i, \dots, n_{I_{sonda}} - 1\} \subset I$ el conjunto de trabajadores especialistas en alimentación por sonda.

$J = \{0, 1, \dots, j, \dots, n_J - 1\}$ el conjunto de pacientes.

$J_5 = \{0, 1, \dots, j, \dots, n_{J_5} - 1\} \subset J$ el conjunto de pacientes que demanda 5 visitas semanales.

$J_7 = \{0, 1, \dots, j, \dots, n_{J_7} - 1\} \subset J$ el conjunto de pacientes que demanda 7 visitas semanales.

$J_{v3} = \{0, 1, \dots, j, \dots, n_{J_{v3}} - 1\} \subset J$ el conjunto de pacientes que demanda 3 visitas diarias.

$J_{grúa} = \{0, 1, \dots, j, \dots, n_{J_{grúa}} - 1\} \subset J$ el conjunto de pacientes que requiere especialistas en manejo de grúa.

$J_{sonda} = \{0, 1, \dots, j, \dots, n_{J_{sonda}} - 1\} \subset J$ el conjunto de pacientes que requiere especialistas en alimentación por sonda.

4.1.2. Parámetros

Los parámetros con los que trabaja el modelo son los siguientes.

4.1.2.1. Parámetros relacionados con el paciente

$Horas_demandadas_j \in \mathbb{R}$ las horas de asistencia demandadas por el paciente j .

$Días_demandados_j = \{5, 7\}$ el número de días que un paciente j tiene que ser atendido cada semana.

$Auxiliares_demandado_j = \{1, 2\}$ el número de auxiliares que deben visitar a un paciente j simultáneamente.

$Visitas_j = \{1, 2, 3\}$ el número de visitas diarias demandadas por el paciente j .

$Grúa_j = \begin{cases} 1 & \text{si el paciente requiere la especialidad manejo de grúa} \\ 0 & \text{e. o. c.} \end{cases}$

$Sonda_j = \begin{cases} 1 & \text{si el paciente requiere la especialidad alimentación por sonda} \\ 0 & \text{e. o. c.} \end{cases}$

$Localización_j = (x_j, y_j)$ las coordenadas para cada paciente j .

4.1.2.2. Parámetros relacionados con el auxiliar

$Contrato_i = \{ 'L - V', 'M - S', 'S - L' \}$ el tipo de contrato que tiene cada auxiliar.

$$Grúa_i = \begin{cases} 1 & \text{si el el auxiliar } i \text{ es especialista en manejo de grúa} \\ 0 & \text{e. o. c.} \end{cases}$$

$$Sonda_i = \begin{cases} 1 & \text{si el auxiliar } i \text{ es especialista en alimentación por sonda} \\ 0 & \text{e. o. c.} \end{cases}$$

$$\alpha_j = \begin{cases} 1 & \text{si el el paciente } j \text{ demanda 3 visitas} \\ 0 & \text{e. o. c.} \end{cases}$$

$Localización_i = (x_i, y_i)$ las coordenadas para cada auxiliar i .

4.1.2.3. Otros parámetros

$Distancia_{i,j} \in \mathbb{R}$ la distancia entre el domicilio del auxiliar i y el domicilio del paciente j .

4.1.3. Variables

$$x_{i,j} = \begin{cases} 1 & \text{si el auxiliar } i \text{ atiende al paciente } j \\ 0 & \text{e. o. c.} \end{cases}$$

$Distancia_aux_i$ es la distancia a todos los pacientes asignados a i .

$Distancia_total$ es la suma para todos los auxiliares i de las distancias a sus pacientes asignados.

$Horas_aux_i$ son las horas mensuales de servicio asignadas a cada auxiliar. La signación total no puede superar las 130 horas mensuales.

4.1.4. Restricciones

Las restricciones se representan en tres grupos:

- R: Restricciones genéricas, incluyen relaciones entre variables y acotaciones del problema.
- A: Restricciones relacionadas exclusivamente con auxiliares.
- P: Restricciones relacionadas exclusivamente con pacientes.

R1. Asignación y cumplimiento de la demanda según el contrato de cada paciente

Los pacientes tienen prescritas visitas durante 5 o 7 días por semana. Los pacientes que requieren 5 días son atendidos por auxiliares con contrato L-V, mientras que los pacientes han de ser visitados durante toda la semana tendrán asignados dos auxiliares: uno con contrato M-S y otro con contrato S-L, que se irán alternando en las visitas. Las restricciones que modelan esta condición son las siguientes:

$$\sum_{i \in I_{LV}} x_{i,j} = Demanda_auxiliares_j + \alpha_j \quad \forall j \in J_5 \quad (1-1)$$

$$\sum_{i \notin I_{LV}} x_{i,j} = 0 \quad \forall j \in J_5 \quad (1-2)$$

$$\sum_{i \in I_{MS}} x_{i,j} = Demanda_auxiliares_j + \alpha_j \quad \forall j \in J_7 \quad (1-3)$$

$$\sum_{i \in I_{SL}} x_{i,j} = Demanda_auxiliares_j + \alpha_j \quad \forall j \in J_7 \quad (1-4)$$

$$\sum_{i \in I_{LV}} x_{i,j} = 0 \quad \forall j \in J_7 \quad (1-5)$$

```

for j in pat:
    if Demanda[j]['Días demandados']==5:
        m.addConstrs(x[i,j]==0 for i in aux if data_aux[i]['contrato']!='L-V')
        m.addConstr(quicksum(x[i,j] for i in aux)==(Demanda[j]['Auxiliares
demandados']+alpha[j]))
    elif Demanda[j]['Días demandados']==7:
        m.addConstrs(x[i,j]==0 for i in aux if data_aux[i]['contrato']=='L-V')
        m.addConstr(quicksum(x[i,j] for i in aux if data_aux[i]['contrato']=='M-
S')==Demanda[j]['Auxiliares demandados']+alpha[j]))
        m.addConstr(quicksum(x[i,j] for i in aux if data_aux[i]['contrato']=='S-
L')==Demanda[j]['Auxiliares demandados']+alpha[j]))

```

R2. Contabilizar de distancia asignada a cada auxiliar

El aspecto evaluado en la función objetivo bajo demanda de la empresa es la distancia que separa a los auxiliares de cada uno de sus pacientes asignados. La variable $Distancia_aux_i$ recoge esta distancia para cada auxiliar i , siendo esta el resultado de sumar todos los productos del parámetro $Distancia_{i,j}$ por su correspondiente variable de asignación.

$$Distancia_aux_i = \sum_{j \in J} Distancia_{i,j} \cdot x_{i,j} \quad \forall i \in I \quad (1-6)$$

```

m.addConstrs((distancia_aux[i]==quicksum(distancia[i,j]*x[i,j] for j in pat) for i in aux),
name='R2')

```

R3. Respetar la especialidad requerida

Los pacientes pueden requerir características especiales. Las tenidas en cuenta por la empresa son el manejo de grúas y la alimentación por sonda. Un paciente que requiera alguna de estas características no podrá ser asignado a un auxiliar que no sea especialista en ello.

$$x_{i,j} = 0 \quad \forall i \notin I_{grúa}; j \in J_{grúa} \quad (1-7)$$

$$x_{i,j} = 0 \quad \forall i \notin I_{sonda}; j \in J_{sonda} \quad (1-8)$$

```
for j in pat:
    for i in aux:
        if Demanda[j]['grua']==1 and data_aux[i]['grua']==0:
            m.addConstr((x[i,j]==0))
        if Demanda[j]['sonda']==1 and data_aux[i]['sonda']==0:
            m.addConstr((x[i,j]==0))
```

R4. Cada auxiliar atiende al menos a un paciente, y como máximo a 4.

Por demanda de la empresa, los auxiliares tienen que asistir al menos a un demandante, y como máximo a cuatro.

$$\sum_{j \in J} x_{i,j} \geq 1 \quad \forall i \in I \quad (1-9)$$

$$\sum_{j \in J} x_{i,j} \leq 4 \quad \forall i \in I \quad (1-10)$$

```
m.addConstrs(x.sum(i,'*')>=1 for i in aux)
m.addConstrs(x.sum(i,'*')<=4 for i in aux)
```

R5. Definición de Horas_ aux_i.

La variable $Horas_aux_i$ acota las horas asignadas a un auxiliar mensualmente, con el objetivo de evitar contratos excesivos.

Las horas que computan en un contrato están compuestas por las horas servidas a los pacientes, los traslados realizados y los posibles descansos, como se detallará en el siguiente modelo. Se estima que, en los peores casos, 75 minutos de cada jornada se dedican a posibles descansos y traslados, lo que a la semana suponen algo más de 6 horas. Teniendo en cuenta un contrato máximo de 35 horas semanales sin contar posibles sustituciones, el

total de horas asignadas no puede ser mayor a 29 horas semanales. Esto se traduce en una acotación superior de 130 horas mensuales en esta variable.

- Al contabilizar las horas tenemos que tener en cuenta que, si se trata de un paciente que requiere 3 visitas, las horas de demanda serán 'compartidas' con otro auxiliar. Por ello, supondremos un reparto equitativo y multiplicaremos por 0.5 las horas dedicadas a este tipo de pacientes.
- La definición de horas varía según el contrato del auxiliar. Para auxiliares con contrato L-V, las horas asignadas será el total de horas que demandan sus pacientes asignados.
- En cuanto a los auxiliares con contrato M-S y S-L, supondremos, al igual que con los pacientes que requieren 3 visitas, un reparto equitativo, aunque pueda no corresponderse con la futura asignación de turnos.

Vistos estos tres puntos, las horas contabilizadas en cada caso se modelarán como:

$$Horas_aux_i = \sum_{j \notin J_{v3}} x_{i,j} \cdot Horas_demandadas_j + 0.5 \cdot \sum_{j \in J_{v3}} x_{i,j} \cdot Horas_demandadas_j \quad \forall i \in I_{LV} \quad (1-11)$$

$$Horas_aux_i = 0.5 \cdot \sum_{j \in J_{v3}} x_{i,j} \cdot Horas_demandadas_j + 0.5 \cdot \sum_{j \notin J_{v3}} x_{i,j} \cdot Horas_demandadas_j \quad \forall i \notin I_{LV} \quad (1-12)$$

```
for i in aux:
    if data_aux[i]['contrato']=='L-V':
        m.addConstr(horas_aux[i]==quicksum(x[i,j]*Demanda[j]['Horas demandadas'] for j in pat if
Demanda[j]['visitas']!=3)+0.5*quicksum(x[i,j]*Demanda[j]['Horas demandadas'] for j in pat if
Demanda[j]['visitas']==3))
    elif data_aux[i]['contrato']=='M-S'or data_aux[i]['contrato']=='S-L':
        m.addConstr(horas_aux[i]==quicksum(x[i,j]*Demanda[j]['Horas demandadas']*0.5 for j in
pat if Demanda[j]['visitas']!=3)+0.5*quicksum(x[i,j]*Demanda[j]['Horas demandadas'] for j in pat
if Demanda[j]['visitas']==3))
```

4.1.5. Función objetivo

El objetivo planteado por la organización es la minimización de la suma para cada auxiliar de las distancias a todos sus pacientes asignados.

Para ello, creamos la variable auxiliar *Distancia_total*:

$$Distancia_total = \sum_{i \in I} Distancia_aux_i \quad (1-13)$$

Siendo por tanto nuestra función objetivo:

$$Min\ Distancia_total \quad (1-14)$$


```
m.addConstr(distancia_total==distancia_aux.sum('*'))
```

4.1.6. Modelo completo para la fase I

$$\text{Min Distancia_total} \quad (1-14)$$

s.a:

$$\sum_{i \in I_{LV}} x_{i,j} = \text{Demanda_auxiliares}_j + \alpha_j \quad \forall j \in J_5 \quad (1-1)$$

$$\sum_{i \in I_{LV}} x_{i,j} = 0 \quad \forall j \in J_5 \quad (1-2)$$

$$\sum_{i \in I_{MS}} x_{i,j} = \text{Demanda_auxiliares}_j + \alpha_j \quad \forall j \in J_7 \quad (1-3)$$

$$\sum_{i \in I_{SL}} x_{i,j} = \text{Demanda_auxiliares}_j + \alpha_j \quad \forall j \in J_7 \quad (1-4)$$

$$\sum_{i \in I_{LV}} x_{i,j} = 0 \quad \forall j \in J_7 \quad (1-5)$$

$$\text{Distancia_aux}_i = \sum_{j \in J} \text{Distancia}_{i,j} \cdot x_{i,j} \quad \forall i \in I \quad (1-6)$$

$$x_{i,j} = 0 \quad \forall i \notin I_{grúa}; j \in J_{grúa} \quad (1-7)$$

$$x_{i,j} = 0 \quad \forall i \notin I_{sonda}; j \in J_{sonda} \quad (1-8)$$

$$\sum_{j \in J} x_{i,j} \geq 1 \quad \forall i \in I \quad (1-9)$$

$$\sum_{j \in J} x_{i,j} \leq 4 \quad \forall i \in I \quad (1-10)$$

$$\text{Horas_aux}_i = \sum_{j \in J_{v3}} x_{i,j} \cdot \text{Horas_demandadas}_j + 0.5 \cdot \sum_{j \in J_{v3}} x_{i,j} * \text{Horas_demandadas}_j \quad \forall i \in I_{LV} \quad (1-11)$$

$$Horas_aux_i = 0.5 * \sum_{j \in J_{v3}} x_{i,j} \cdot Horas_demandadas_j + 0.5 \cdot \sum_{j \in J_{v3}} x_{i,j} * Horas_demandadas_j \quad \forall i \notin I_{LV} \quad (1-12)$$

$$Distancia_total = \sum_{i \in I} Distancia_aux_i \quad (1-13)$$

El output obtenido tras la ejecución de este modelo es un nuevo documento Excel con el listado de asignaciones paciente-demandante.

4.2. Fase II: Modelo de planificación

El modelo, planteado para un horizonte temporal de un mes, planifica las visitas a cada paciente cumpliendo con la demanda mensual y sin tener en cuenta posibles ausencias que se den a lo largo del mes. Debe cumplir así mismo con restricciones de jornada laboral y disponibilidad.

Los inputs que recibe son los libros Excel empleados en la fase I, así como su output.

4.2.1. Conjuntos

$I = \{0, 1, \dots, i, \dots, n_I - 1\}$ el conjunto de auxiliares de la empresa.

$I_{LV} = \{0, 1, \dots, i, \dots, n_{LV} - 1\} \subset I$ el conjunto de auxiliares con contrato L-S.

$I_{MS} = \{0, 1, \dots, i, \dots, n_{MS} - 1\} \subset I$ el conjunto de auxiliares con contrato M-S.

$I_{SL} = \{0, 1, \dots, i, \dots, n_{SL} - 1\} \subset I$ el conjunto de auxiliares con contrato S-L.

$I_{grúa} = \{0, 1, \dots, i, \dots, n_{Igrúa} - 1\} \subset I$ el conjunto de auxiliares especialistas en manejo de grúa.

$I_{sonda} = \{0, 1, \dots, i, \dots, n_{sonda} - 1\} \subset I$ el conjunto de auxiliares especialistas en alimentación por sonda.

$J = \{0, 1, \dots, j, \dots, n_J - 1\}$ el conjunto de pacientes.

$J_5 = \{0, 1, \dots, j, \dots, n_{J5} - 1\} \subset J$ el conjunto de pacientes que demanda 5 visitas semanales.

$J_7 = \{0, 1, \dots, j, \dots, n_{J7} - 1\} \subset J$ el conjunto de pacientes que demanda 7 visitas semanales.

$J_{grúa} = \{0, 1, \dots, j, \dots, n_{grúa} - 1\} \subset J$ el conjunto de pacientes que requiere especialistas en manejo de grúa.

$J_{sonda} = \{0, 1, \dots, j, \dots, n_{sonda} - 1\} \subset J$ el conjunto de pacientes que requiere especialistas en alimentación por sonda.

$Y_i = \{0, 1, \dots, j, \dots, n_{Yi} - 1\} \subset J$ el conjunto de pacientes asignados al auxiliar i .

$K = \{0, 1, \dots, j, \dots, n_K\}$ el conjunto de días del mes a planificar.

$K_w = \{0, 1, \dots, j, \dots, n_{KW}\} \subset K$ el conjunto de fines de semana del mes a planificar.

$Semana = \{0, 1, \dots, j, \dots, n_{Semana}\} \subset K$ el conjunto de semanas dentro del mes a planificar

$K_S = \{0, 1, \dots, j, \dots, n_{K_S}\} \subset \text{Semana}$ el conjunto de días dentro de una semana.

$L = \{0, 1, 2\}$ el conjunto de slots o turnos dentro de un día.

4.2.2. Parámetros

Los parámetros con los que trabaja el modelo son los siguientes.

4.2.2.1. Parámetros relacionados con el paciente

Horas_demandadas $_j \in \mathbb{R}$ las horas de asistencia demandadas por el paciente j .

Días_demandados $_j = \{5, 7\}$ el número de días que un paciente j tiene que ser atendido cada semana.

Auxiliares_demandados $_j = \{1, 2\}$ el número de auxiliares que deben visitar a un paciente j simultáneamente.

Desplazamiento $_j = \{10, 15\}$ los minutos de desplazamiento para visitar al paciente j .

Visitas $_j = \{1, 2, 3\}$ el número de visitas diarias demandadas por el paciente j .

Grúa $_j = \begin{cases} 1 & \text{si el paciente requiere la especialidad manejo de grúa} \\ 0 & \text{e. o. c.} \end{cases}$

Sonda $_j = \begin{cases} 1 & \text{si el paciente requiere la especialidad alimentación por sonda} \\ 0 & \text{e. o. c.} \end{cases}$

4.2.2.2. Parámetros relacionados con el auxiliar

Contrato $_i = \{ 'L - V', 'M - S', 'S - L' \}$ el tipo de contrato que tiene cada auxiliar.

Grúa $_i = \begin{cases} 1 & \text{si el auxiliar } i \text{ es especialista en manejo de grúa} \\ 0 & \text{e. o. c.} \end{cases}$

Sonda $_i = \begin{cases} 1 & \text{si el auxiliar } i \text{ es especialista en alimentación por sonda} \\ 0 & \text{e. o. c.} \end{cases}$

Asignaciones $_i = \{j \in Y_i\}$

4.2.2.3. Otros parámetros

Mes = $\{1, 2, \dots, 12\}$ el mes para el que se planifican las visitas.

Año $\in \mathbb{N}$ el año para el que se planifican las visitas.

M = 10^3 parámetro auxiliar para el modelado de restricciones condicionales

4.2.3. Variables

$x_{i,j,k,l} = \begin{cases} 1 & \text{si el auxiliar } i \text{ atiende al paciente } j \text{ el día } k \text{ durante el slot } l \text{ si } j \in Y_i \\ 0 & \text{e.o.c.} \end{cases}$

Horas $_a_{i,k,l}$ las horas de contrato contabilizadas para un auxiliar i el día k en el turno l .

Horas $_servidas_{i,j,k,l}$ las horas servidas por el auxiliar i , al paciente j , el día k durante el turno l .

$Horas_p_{j,k,l}$ las horas que percibe un paciente j , el día k durante el turno l .

$Traslados_{i,k,l}$ los traslados realizados por el auxiliar i , el día k durante el turno l en minutos.

$$d_{i,k,l} = \begin{cases} 1 & \text{si el auxiliar } i \text{ realiza un descanso durante el turno } l \text{ del día } k \\ 0 & \text{e.o.c} \end{cases}$$

Las variables $x_{i,j,k,l}$ y $Horas_servidas_{i,j,k,l}$ son generadas únicamente cuando un paciente está asignado a un auxiliar.

4.2.4. Restricciones

R1. Definición de horas de contrato de un auxiliar

La variable $Horas_a_{i,k,l}$ representa el total de horas contabilizadas en el contrato. Estas horas son la suma de las horas servidas a cada paciente dentro de un slot, los tiempos de traslados y los posibles descansos.

Contabilizaremos descansos de 15 minutos, por lo que al trabajar en horas dividiremos el tercer término de la restricción entre 60 para cambiar sus unidades.

$$Horas_a_{i,k,l} = \sum_{j \in J} Horas_servidas_{i,j,k,l} + Traslados_{i,k,l} + d_{i,k,l} \cdot 15/60 \quad \forall i \in I, k \in K, l \in L \quad (2-1)$$

```
m.addConstrs((horas_a[i,k,l]==horas_servidas.sum(i,'*',k,l)+traslados[i,k,l]+d1[i,k,l]*round(15/60,3) for (i,k,l) in ha), name='R1')
```

R1.1 Horas servidas

Las siguientes restricciones modelan y restringen la variable $Horas_servidas_{i,j,k,l}$.

R1.1.1 Relación entre horas servidas y horas percibidas por un paciente

La restricción (2-2) relaciona las variables $Horas_servidas_{i,j,k,l}$ y $Horas_p_{j,k,l}$: cuando la variable binaria de asignación toma valor 1, las horas que dedica el auxiliar i al paciente j son las mismas que el paciente j percibe.

Los casos de pacientes que requieran 2 auxiliares simultáneamente se representan con la restricción (2-3), que fuerza que el sumatorio (para todos los auxiliares) de las horas servidas a j en k y l sea igual al número de horas percibidas por j multiplicado por el número de auxiliares que j demanda.

$$Horas_servidas_{i,j,k,l} = Horas_p_{j,k,l} \cdot x_{i,j,k,l} \quad \forall i \in I, j \in J, k \in K, l \in L \quad (2-2)$$

$$\sum_{i \in I} Horas_servidas_{i,j,k,l} = Horas_p_{j,k,l} \cdot Auxiliares_demandados_j \quad \forall j \in J, k \in K, l \in L \quad (2-3)$$

```
m.addConstrs((horas_servidas[i,j,k,l]==horas_p[j,k,l]*x[i,j,k,l] for (i,j,k,l) in lista))
m.addConstrs((horas_servidas.sum('*',j,k,l)==horas_p[j,k,l]*Demanda[j]['Auxiliares demandados']
for (j,k,l) in hp), name="R1.1.1")
```

R1.1.2 Relación horas servidas y variable de asignación

La variable $Horas_servidas_{i,j,k,l}$ solo puede tomar valores mayores que cero para unos valores (i,j,k,l) cuando el auxiliar i visita al paciente j el día k en el turno l .

$$Horas_servidas_{i,j,k,l} \leq M \cdot x_{i,j,k,l} \quad \forall i \in I, j \in J, k \in K, l \in L \quad (2-4)$$

```
m.addConstrs((horas_servidas[i,j,k,l]<=x[i,j,k,l]*BIG_M for (i,j,k,l) in lista ),name='R1.1.2')
```

R1.1.3. Visita mínima de una hora

Si se realiza una visita, la duración mínima ha de ser una hora. A partir de ahí cualquier duración es válida, siempre que no supere la jornada laboral del auxiliar.

$$Horas_servidas_{i,j,k,l} \geq x_{i,j,k,l} \quad \forall i \in I, j \in J, k \in K, l \in L \quad (2-5)$$

```
m.addConstrs((x[i,j,k,l]<=horas_servidas[i,j,k,l] for (i,j,k,l) in lista), name='R1.1.3')
```

R1.2. Definición de traslados

La variable traslados es igual a la suma de todos los desplazamientos que un auxiliar haga durante cada slot y día. Los datos aportados por la empresa se expresan en minutos, por lo que esta restricción incluye además un cambio de unidades.

$$Traslados_{i,k,l} = \sum_{j \in J} x_{i,j,k,l} \cdot Desplazamiento_j / 60 \quad \forall i \in I, k \in K, l \in L \quad (2-6)$$

```
m.addConstrs((traslados[i,k,l]==quicksum (x[i,j,k,l]*round(Demanda[j]['desplazamiento']/60,3)
for j in pat) for (i,k,l) in ha), name='R1.2')
```

R1.3. Definición de descansos

Cada 6 horas de trabajo consecutivas se otorgan 15 minutos de descanso. Dado que en esta segunda fase del

modelo no se obtienen las horas exactas de inicio y finalización de cada visita, se otorgará una holgura correspondiente a este descanso. Sería en una hipotética cuarta fase propuesta cuando esta holgura se materializa o no, quedando fuera del alcance de este proyecto.

Esta holgura se otorgará en situaciones en las que la suma de las horas asignadas en dos slots consecutivos sume 6 o más. Como se define en el problema, los trabajadores pueden trabajar un máximo de dos turnos diarios, por lo que el total de descansos otorgados cada día a cada paciente será mejor o igual a uno.

Para simplificar la restricción, definiremos las horas servidas y los traslados como $Horas_trabajadas_{i,k,l}$.

$$Horas_trabajadas_{i,k,l} = \sum_{j \in J} Horas_servidas_{i,j,k,l} + Traslados_{i,k,l} \quad \forall i \in I, k \in K, l \in L \quad (2-7)$$

La restricción de descansos quedaría por tanto como:

$$Horas_trabajadas_{i,k,l-1} + Horas_trabajadas_{i,k,l} \geq 6 - M \cdot (1 - d_{i,k,l}) \quad \forall i \in I, k \in K, l = 1,2 \quad (2-8)$$

```
m.addConstrs((d1.sum(i,k,'*')<=1 for i in aux for k in days))
m.addConstrs(((horas_servidas.sum(i,'*',k,l-1)+traslados[i,k,l-1])+(horas_servidas.sum(i,'*',k,l)+traslados[i,k,l])>=6-BIG_M*(1-d1[i,k,l]) for (i,k,l) in ha if l>0), name='R1.3.1')
m.addConstrs(((horas_servidas.sum(i,'*',k,l-1)+traslados[i,k,l-1])+(horas_servidas.sum(i,'*',k,l)+traslados[i,k,l])<=6+BIG_M*d1[i,k,l] for (i,k,l) in ha if l>0), name='R.1.3.2')
```

A1. Jornada laboral

Por convenio, la jornada laboral de los auxiliares no puede superar las 9 horas. Por ello, incluimos una restricción que limite a este valor la suma de las horas trabajadas a lo largo de todos los turnos por un trabajador cada día.

$$\sum_{l \in L} Horas_a_{i,k,l} \leq 9 \quad \forall i \in I, \quad k \in K \quad (2-9)$$

```
m.addConstrs((horas_a.sum(i,k,'*')<=9 for i in aux for k in days), name="A1")
```

A2. Trabajar como máximo 2 turnos diarios

La jornada laboral de cada auxiliar debe organizarse como máximo en dos turnos. Las siguientes restricciones modelan este límite.

Para ello crearemos la variable auxiliar $\gamma_{i,k,l}$, representando los turnos en los que trabaja i .

$$\gamma_{i,k,l} = \begin{cases} 1 & \text{si el auxiliar } i \text{ trabaja durante el turno } l \text{ del día } l \\ 0 & \text{si no} \end{cases}$$

Cuando un trabajador atiende a al menos un paciente durante un slot, $\gamma_{i,k,l}$ debe valer 1:

$$\sum_{j \in J} x_{i,j,k,l} \geq 1 - M \cdot (1 - \gamma_{i,k,l}) \quad \forall i \in I, k \in K, l \in L \quad (2-10)$$

Si por el contrario no se atiende a ningún paciente, el valor de $\gamma_{i,k,l}$ debe ser forzosamente 0:

$$\sum_{j \in J} x_{i,j,k,l} \leq M \cdot \gamma_{i,k,l} \quad \forall i \in I, k \in K, l \in L \quad (2-11)$$

Una vez modelada la relación entre γ y slots, limitamos el valor de la suma para todos los slots de un día a 2:

$$\sum_{l \in L} \gamma_{i,k,l} \leq 2 \quad \forall i \in I, k \in K, l \in L \quad (2-12)$$

```
m.addConstrs((x.sum(i, '*', k, l) >= 1 - BIG_M * (1 - gamma[i, k, l]) for (i, k, l) in ha), name='A2.1')
m.addConstrs((x.sum(i, '*', k, l) <= BIG_M * gamma[i, k, l] for (i, k, l) in ha), name='A2.2')
m.addConstrs((gamma.sum(i, k, '*') <= 2 for i in aux for k in days), name='A2.3')
```

A3. Entre dos jornadas tienen que pasar al menos 12 horas

El tercer criterio impuesto referente a la jornada laboral es que entre el final de la jornada laboral del día k y el inicio de la jornada $k+1$ deben transcurrir al menos 12 horas. Esto quiere decir que, si una jornada finaliza a las 21:00, la jornada del día siguiente no podrá comenzar hasta las 9:00 de la mañana.

El último turno de trabajo dura 4 horas (de 18:00 a 22:00), mientras que el primer turno son 6 horas (de 8:00 a 14:00). Entre estos dos turnos pasan 10 horas, por lo que la suma de las horas trabajadas en el último turno de k y el primero de $k+1$ debe ser 4 (turno 2, día k) + 6 (turno 0, día $k+1$) - 2 (para respetar la restricción).

La Ilustración 7 ejemplifica este concepto. La parte izquierda representa las horas del último turno de k , y la derecha las correspondientes al primer turno de $k+1$. El sumatorio de ambas no podrá ser nunca superior a 8 si queremos respetar esta restricción.

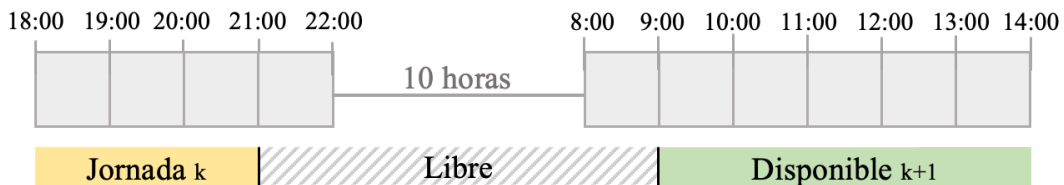


Ilustración 7. Ejemplo restricción A3

Visto esto, el grupo de restricciones correspondientes lo modelaremos como:

$$Horas_{a_{i,k,2}} + Horas_{a_{i,k+1,0}} \leq 8 \quad \forall i \in I, k = 1 \dots K - 1 \quad (2-13)$$

Este modelo resuelve la planificación para un mes, por lo que en la práctica, al planificar las visitas del primer día de un mes ha de tenerse en cuenta las horas trabajadas durante el último turno del último día del mes anterior. Para evitar conflictos en este modelo generamos esta restricción para el conjunto de días excepto el último.

```
m.addConstrs((horas_a[i,k,2]+horas_a[i,k+1,0]<=8 for i in aux for k in days[:-1]), name='A3')
```

A4. Cada auxiliar debe tener programadas entre 1 y 35 horas cada semana

Según el criterio de la empresa, cada trabajador debe tener cada semana asignadas entre 1 y 35 horas de contrato.

$$\sum_{k_s \in Semana} \sum_{l \in L} Horas_{a_{i,k_s,l}} \leq 35 \quad \forall i \in I, k_s \in Semana \quad (2-14)$$

$$\sum_{k_s \in Semana} \sum_{l \in L} Horas_{a_{i,k_s,l}} \geq 1 \quad \forall i \in I, k_s \in Semana \quad (2-15)$$

```
for week in range(len(c1)):
    week_days=[]
    for d in c1[week]:
        if d>0:
            week_days.append(d)
    m.addConstrs((quicksum(horas_a[i,k,l] for k in week_days for l in slots)>=1 for i in aux),
name='A4')
    m.addConstrs((quicksum(horas_a[i,k,l] for k in week_days for l in slots)<=35 for i in aux),
name='A4')
```

P1. Demanda horaria de cada paciente

La demanda mensual de cada paciente es un dato conocido en este problema. La suma por tanto de todas las horas de atención que ha recibido un paciente debe ser igual a este dato.

$$\sum_{k \in K} \sum_{l \in L} Horas_{p_{j,k,l}} = Demanda_j \quad \forall j \in J \quad (2-16)$$


```
m.addConstrs((horas_p.sum(j,'*','*')==Demanda[j]['Horas demandadas'] for j in pat), name='P1')
```

P2. Demanda de auxiliares y número de visitas

La siguiente restricción modela que, cada día y para cada paciente, debe cumplirse la demanda de auxiliares y visitas. Para pacientes con asistencia únicamente en días laborables, hay que hacer una breve distinción. La demanda de visitas y asistentes ha de satisfacerse únicamente durante los cinco primeros días de la semana, siendo el valor de la variable $x_{i,j,k,l}$ nulo en fines de semana.

Pacientes que demandan 7 días:

$$\sum_{i \in I} \sum_{l \in L} x_{i,j,k,l} = Demanda_auxiliares_j \cdot Demanda_visitas_j \quad \forall j \in J, k \in K \quad (2-17)$$

```
for j in pat:
    if Demanda[j]['Días demandados']==7:
        m.addConstrs((x.sum('*','j,k','*')==Demanda[j]['visitas']*Demanda[j]['Auxiliares
demandados'] for k in days), name='P2.1')
```

Pacientes que demandan 5 días:

$$\sum_{i \in I} \sum_{l \in L} x_{i,j,k,l} = Demanda_auxiliares_j \cdot Demanda_visitas_j \quad \forall j \in J, k \notin K_w \quad (2-18)$$

$$\sum_{i \in I} \sum_{l \in L} x_{i,j,k,l} = 0 \quad \forall j \in J, k \in K_w \quad (2-19)$$

$$\sum_{l \in L} Horas_p_{j,k,l} = 0 \quad \forall j \in J, k \in K_w \quad (2-20)$$

```
for j in pat:
    if Demanda[j]['Días demandados']==5:
        m.addConstrs((x.sum('*','j,k','*')==Demanda[j]['visitas']*Demanda[j]['Auxiliares
demandados'] for k in days if k not in weekend), name='P3.2')
        m.addConstrs((x.sum('*','j,k','*')==0 for k in weekend), name='P2.3')
        m.addConstrs((horas_p.sum(j,k,'*')==0 for k in weekend), name='P2.4')
```

4.2.5. Función Objetivo

De acuerdo con la empresa, el objetivo real de este problema no es más que cuadrar las visitas de todos los pacientes a lo largo de un mes, por lo que en este aspecto no será necesaria una búsqueda de una mejor solución: en principio, y ante las necesidades de la organización, cualquier solución que cumpla con los requisitos impuestos es una solución válida y aceptable para ellos.

En un futuro, tanto en esta fase como en la anterior pueden establecerse otros objetivos adicionales como el equilibrado de horas asignadas o de número de pacientes.

4.2.6. Modelo completo fase II

$$Horas_{a_{i,k,l}} = \sum_{j \in J} Horas_{servidas_{i,j,k,l}} + Traslados_{i,k,l} + d_{i,k,l} \cdot 15/60 \quad \forall i \in I, k \in K, l \in L \quad (2-1)$$

$$Horas_{servidas_{i,j,k,l}} = Horas_{p_{j,k,l}} \cdot x_{i,j,k,l} \quad \forall i \in I, j \in J, k \in K, l \in L \quad (2-2)$$

$$\sum_{i \in I} Horas_{servidas_{i,j,k,l}} = Horas_{p_{j,k,l}} \cdot Auxiliares_{demandados_j} \quad \forall j \in J, k \in K, l \in L \quad (2-3)$$

$$Horas_{servidas_{i,j,k,l}} \leq M \cdot x_{i,j,k,l} \quad \forall i \in I, j \in J, k \in K, l \in L \quad (2-4)$$

$$Horas_{servidas_{i,j,k,l}} \geq x_{i,j,k,l} \quad \forall i \in I, j \in J, k \in K, l \in L \quad (2-5)$$

$$Traslados_{i,k,l} = \sum_{j \in J} x_{i,j,k,l} \cdot Desplazamiento_j / 60 \quad \forall i \in I, k \in K, l \in L \quad (2-6)$$

$$Horas_{trabajadas_{i,k,l}} = \sum_{j \in J} Horas_{servidas_{i,j,k,l}} + Traslados_{i,k,l} \quad \forall i \in I, k \in K, l \in L \quad (2-7)$$

$$Horas_{trabajadas_{i,k,l-1}} + Horas_{trabajadas_{i,k,l}} \geq 6 - M \cdot (1 - d_{i,k,l}) \quad \forall i \in I, k \in K, l = 1, 2 \quad (2-8)$$

$$\sum_{l \in L} Horas_{a_{i,k,l}} \leq 9 \quad \forall i \in I, k \in K \quad (2-9)$$

$$\sum_{j \in J} x_{i,j,k,l} \geq 1 - M \cdot (1 - \gamma_{i,k,l}) \quad \forall i \in I, k \in K, l \in L \quad (2-10)$$

$$\sum_{j \in J} x_{i,j,k,l} \leq M \cdot \gamma_{i,k,l} \quad \forall i \in I, k \in K, l \in L \quad (2-11)$$

$$\sum_{l \in L} \gamma_{i,k,l} \leq 2 \quad \forall i \in I, k \in K, l \in L \quad (2-12)$$

$$Horas_{a_{i,k,2}} + Horas_{a_{i,k+1,0}} \leq 8 \quad \forall i \in I, \quad (2-13)$$

$$k = 1 \dots K - 1$$

$$\sum_{ks \in Semana} \sum_{l \in L} Horas_{a_{i,ks,l}} \leq 35 \quad \forall i \in I, k_s \in Semana \quad (2-14)$$

$$\sum_{ks \in Semana} \sum_{l \in L} Horas_{a_{i,ks,l}} \geq 1 \quad \forall i \in I, k_s \in Semana \quad (2-15)$$

$$\sum_{k \in K} \sum_{l \in L} Horas_{p_{j,k,l}} = Demanda_j \quad \forall j \in J \quad (2-16)$$

$$\sum_{i \in I} \sum_{l \in L} x_{i,j,k,l} = Demanda_{auxiliares_j} \cdot Demanda_{visitas_j} \quad \forall j \in J, k \in K \quad (2-17)$$

$$\sum_{i \in I} \sum_{l \in L} x_{i,j,k,l} = Demanda_{auxiliares_j} \cdot Demanda_{visitas_j} \quad \forall j \in J, k \notin K_w \quad (2-18)$$

$$\sum_{i \in I} \sum_{l \in L} x_{i,j,k,l} = 0 \quad \forall j \in J, k \in K_w \quad (2-19)$$

$$\sum_{l \in L} Horas_{p_{j,k,l}} = 0 \quad \forall j \in J, k \in K_w \quad (2-20)$$

Los outputs de esta segunda fase serán dos libros Excel:

- Calendario.xlsx tiene tantas hojas como días tenga el mes planificado. En cada una de ellas se recoge la programación de visitas, indicando auxiliar, paciente al que visita, turno en el que lo hace y duración de la misma.
- Contratos.xlsx recoge en una hoja el total, para cada auxiliar, de horas asignadas durante el mes planificado.

Además, se generan datos necesarios para la Fase III.

4.3. Fase III: Modelo de replanificación diaria por ausencias.

Con el calendario obtenido en la fase II quedan planificadas todas las visitas para un mes dado. Sin embargo, una vez puesto en marcha, auxiliares y demandantes podrán ausentarse en momentos puntuales. Este modelo reorganiza las visitas para un día dado, teniendo en cuenta las ausencias previstas y las visitas a realizar.

El modelo es una adaptación del empleado en la fase II, fijando un día concreto y teniendo en cuenta que la demanda a satisfacer en este caso será aquella planificada inicialmente para este día. Para agilizar la lectura, en el apartado 4.3.4. comentaremos únicamente las modificaciones realizadas. Para otras descripciones, se señalan en **negrita** las nuevas modificaciones

4.3.1. Conjuntos

Los conjuntos tenidos en cuenta coinciden con los descritos en la Fase II. A pesar de que en este caso K tendrá un valor único, tendremos en cuenta los conjuntos Semana y K_w para en las restricciones saber cuándo aplicar

una condición u otra.

$I = \{0, 1, \dots, i, \dots, n_I - 1\}$ el conjunto de auxiliares de la empresa.

$I_{LV} = \{0, 1, \dots, i, \dots, n_{I_{LV}} - 1\} \subset I$ el conjunto de auxiliares con contrato L-S.

$I_{MS} = \{0, 1, \dots, i, \dots, n_{I_{MS}} - 1\} \subset I$ el conjunto de auxiliares con contrato M-S.

$I_{SL} = \{0, 1, \dots, i, \dots, n_{I_{SL}} - 1\} \subset I$ el conjunto de auxiliares con contrato S-L.

$I_{grúa} = \{0, 1, \dots, i, \dots, n_{I_{grúa}} - 1\} \subset I$ el conjunto de auxiliares especialistas en manejo de grúa.

$I_{sonda} = \{0, 1, \dots, i, \dots, n_{I_{sonda}} - 1\} \subset I$ el conjunto de auxiliares especialistas en alimentación por sonda.

$K = J = \{0, 1, \dots, j, \dots, n_J - 1\}$ el conjunto de pacientes.

$J_5 = \{0, 1, \dots, j, \dots, n_{J_5} - 1\} \subset J$ el conjunto de pacientes que demanda 5 visitas semanales.

$J_7 = \{0, 1, \dots, j, \dots, n_{J_7} - 1\} \subset J$ el conjunto de pacientes que demanda 7 visitas semanales.

$J_{grúa} = \{0, 1, \dots, j, \dots, n_{J_{grúa}} - 1\} \subset J$ el conjunto de pacientes que requiere especialistas en manejo de grúa.

$J_{sonda} = \{0, 1, \dots, j, \dots, n_{J_{sonda}} - 1\} \subset J$ el conjunto de pacientes que requiere especialistas en alimentación por sonda.

$K = \{0, 1, \dots, j, \dots, n_K\}$ el conjunto de días del mes a planificar.

$K_w = \{0, 1, \dots, j, \dots, n_{K_w}\} \subset K$ el conjunto de fines de semana del mes a planificar.

$Semana = \{0, 1, \dots, j, \dots, n_{Semana}\} \subset K$ el conjunto de semanas dentro del mes a planificar

$K_s = \{0, 1, \dots, j, \dots, n_{K_s}\} \subset Semana$ el conjunto de días dentro de una semana.

4.3.2. Parámetros

Se mantienen los parámetros de la Fase II, indicándose en **negrita** los nuevos parámetros incluidos.

4.2.2.4. Parámetros relacionados con el paciente

Demanda_j $\in \mathbb{R}$ las horas de asistencia planificadas inicialmente para el paciente j .

Disponibilidad_j $= \{0, 1, 2, 3\}$ numero de slots disponibles el día k .

Dias_demandados_j $= \{5,7\}$ Días demandados _{j} $= \{5,7\}$ el número de días que un paciente j tiene que ser atendido cada semana.

Auxiliares_demandados_j $= \{1, 2\}$ el número de auxiliares que deben visitar a un paciente j simultáneamente.

Desplazamiento_j $= \{10,15\}$ los minutos de desplazamiento para visitar al paciente j .

Visitas_j $= \{1, 2, 3\}$ el número de visitas diarias demandadas por el paciente j .

Grúa_j $= \begin{cases} 1 & \text{si el paciente requiere la especialidad manejo de grúa} \\ 0 & \text{e. o. c.} \end{cases}$

Sonda_j $= \begin{cases} 1 & \text{si el paciente requiere la especialidad alimentación por sonda} \\ 0 & \text{e. o. c.} \end{cases}$

Ausencia_{p_{j,k,l}} $= \begin{cases} 1 & \text{si el paciente } j \text{ se ausenta el día } k \text{ durante el slot } l \\ 0 & \text{e. o. c.} \end{cases}$

$$Ausencia_día_{j,k,l} = \begin{cases} 1 & \text{si el paciente } j \text{ se ausenta el día completo} \\ 0 & \text{e. o. c.} \end{cases}$$

4.2.2.5. Parámetros relacionados con el auxiliar

$Contrato_i = \{ 'L - V', 'M - S', 'S - L' \}$ el tipo de contrato que tiene cada auxiliar.

$$Grúa_i = \begin{cases} 1 & \text{si el el auxiliar } i \text{ es especialista en manejo de grúa} \\ 0 & \text{e. o. c.} \end{cases}$$

$$Sonda_i = \begin{cases} 1 & \text{si el auxiliar } i \text{ es especialista en alimentación por sonda} \\ 0 & \text{e. o. c.} \end{cases}$$

$$Asignaciones_i = \{ j \in Y_i \}$$

$$Ausencia_a_{i,k,l} = \begin{cases} 1 & \text{si el auxiliar } i \text{ se ausenta el día } k \text{ durante el slot } l \\ 0 & \text{e. o. c.} \end{cases}$$

$Horas_inicial_{i,k,l} \in \mathbb{R}$ las horas asignadas inicialmente a cada paciente i , el día k en el turno l .

4.2.2.6. Otros parámetros

$Mes = \{1,2, \dots, 12\}$ el mes para el que se planifican las visitas.

$Año \in \mathbb{N}$ el año para el que se planifican las visitas.

$M = 10^3$ parámetro auxiliar para el modelado de restricciones condicionales.

$x_inicial_{i,j,l} = \{0,1\}$ el valor de la variable $x_{i,j,k,l}$ correspondiente a la fase II en cada k a reprogramar.

4.3.3. Variables

$$x_{i,j,k,l} = \begin{cases} 1 & \text{si el auxiliar } i \text{ atiende al paciente } j \text{ el día } k \text{ durante el slot } l \text{ si } j \in Y_i \\ 0 & \text{e.o.c.} \end{cases}$$

$Horas_a_{i,k,l}$ las horas de contrato contabilizadas para un auxiliar i el día k en el turno l .

$Horas_servidas_{i,j,k,l}$ las horas servidas por el auxiliar i , al paciente j , el día k durante el turno l .

$Horas_p_{j,k,l}$ las horas el percibe un paciente j , el día k durante el turno l .

$Traslados_{i,k,l}$ los traslados realizados por el auxiliar i , el día k durante el turno l en minutos.

$$d_{i,k,l} = \begin{cases} 1 & \text{si el auxiliar } i \text{ realiza un descanso durante el turno } l \text{ del día } k \\ 0 & \text{e. o. c} \end{cases}$$

Desviación_calendario la diferencia en valor absoluto entre $x_{i,j,k,l}$ y el parámetro $x_inicial_{i,j,l}$.

Penalización la penalización aplicada a la función objetivo por sustituciones realizadas.

4.3.4. Restricciones

El modelo diseñado para la fase II es aplicable en este caso, si bien requiere añadir nuevas restricciones y modificar antiguas. En primer lugar, se describirán las adaptaciones realizadas sobre las restricciones ya existentes (en concreto, sobre las restricciones A3 y P2), y finalmente se describen dos nuevos grupos de restricciones (P3 y R2). Asimismo, como se comenta en la descripción del problema, no se tendrán en cuenta en

esta fase el límite inferior y superior de horas semanales de contrato, correspondiente a las restricciones A4.

Modificación de A3. Entre dos jornadas tienen que pasar al menos 12 horas

Al ser k un parámetro fijo en este modelo, esta restricción ha de ser modificada. En este caso, las horas del día anterior y posterior son parámetros fijos, por lo que aseguraremos que se cumple esta restricción dividiendo la restricción inicial (2-13) en dos restricciones:

Partiendo de la restricción inicial:

$$Horas_{a_{i,k,2}} + Horas_{a_{i,k+1,0}} \leq 8 \quad \forall i \in I, k = 1 \dots K - 1 \quad (2-13)$$

- En la restricción (3-1) sumamos las horas del primer turno del día que reasigno y las horas del último slot del día anterior (dato).
- En la restricción (3-2) sumamos las horas del último turno del día que reasigno y las del primer turno del día siguiente.

$$Horas_{a_{i,k,0}} + Horas_{inicial_{i,k-1,2}} \leq 8 \quad \forall i \in I, k \neq 1 \quad (3-1)$$

$$Horas_{a_{i,k,2}} + Horas_{inicial_{i,k+1,0}} \leq 8 \quad \forall i \in I, k \neq k_n \quad (3-2)$$

Al tratarse de un modelo planteado para un único mes, en la primera restricción no se tendrá en cuenta el primer día (al tener que hacer la comprobación con el día anterior), y en la segunda omitiremos el último día (por no existir un día posterior a este).

```
if k!=1:
    m.addConstrs((horas_inicial[i][k-1][2]+horas_a[i,k,0]<=8 for i in aux), name='A3.1')
if k!=days[-1]:
    m.addConstrs((horas_inicial[i][k+1][0]+horas_a[i,k,2]<=8 for i in aux), name='A3.2')
```

Modificación de P2. Demanda de auxiliares y número de visitas

La siguiente restricción modela que, cada día y para cada paciente, debe cumplirse la demanda de auxiliares y visitas. Las ausencias de día completo se tienen en cuenta en esta restricción con el término $(1 - Ausencia_día_j)$, que como se comenta en apartados anteriores, es una variable binaria que se activa en caso de ausencia, ya sea por día completo o porque el número de slots disponibles es inferior al número de visitas requeridas. Por tanto, este término anulará la parte derecha de la restricción en caso de ausentarse el paciente.

Pacientes que demandan 7 días:

$$\sum_{i \in I} \sum_{l \in L} x_{i,j,k,l} = (1 - Ausencia_día_j) \cdot Demanda_auxiliares_j \cdot Demanda_visitas_j \quad \forall j \in J_7, k \in K \quad (3-3)$$

Pacientes que demandan 5 días:

$$\sum_{i \in I} \sum_{l \in L} x_{i,j,k,l} = (1 - \text{Ausencia_día}_j) \cdot \text{Demanda_auxiliares}_j \cdot \text{Demanda_visitas}_j \quad \forall j \in J_5, k \notin K_W \quad (3-4)$$

$$\sum_{i \in I} \sum_{l \in L} x_{i,j,k,l} = 0 \quad \forall j \in J_5, k \in K_W \quad (3-5)$$

$$\sum_{l \in L} \text{Horas_p}_{j,k,l} = 0 \quad \forall j \in J_5, k \in K_W \quad (3-6)$$

Para programar esta restricción en Python comprobaremos inicialmente si k es un día laboral o fin de semana, para diferenciar la demanda de los diferentes pacientes.

```

for j in pat:
    if k in weekend:
        if Demanda[j]['Días demandados']==7:
            m.addConstr((x.sum('*',j,k,'')== (1-
ausencia_dia[j])*Demanda[j]['visitas']*Demanda[j]['Auxiliares demandados']), name='P2.1')

        else:
            m.addConstr((x.sum('*',j,k,'')==0), name='P2.1')

    else:
        m.addConstr((x.sum('*',j,k,'')== (1-
ausencia_dia[j])*Demanda[j]['visitas']*Demanda[j]['Auxiliares demandados']), name='P2.2')

```

P3. Si un paciente requiere el uso de grúa, sonda, o ambas, el auxiliar que le atienda debe contar con esta especialidad.

El listado de asignaciones obtenidos en el primer modelo ya tiene en cuenta los servicios especiales requeridos por cada paciente. No obstante, de ser necesaria una sustitución o una tercera visita, estos requisitos deben ser tenidos en cuenta. Por ello, la siguiente restricción fuerza a que auxiliares no especialistas no puedan atender a pacientes que requieren manejo de grúa no formación en alimentación por sonda.

$$x_{i,j,k,l} = 0 \quad \forall i \in I_{grúa}, j \in J_{grúa}, k \in K, l \in L \quad (3-7)$$

$$x_{i,j,k,l} = 0 \quad \forall i \in I_{sonda}, j \in J_{sonda}, k \in K, l \in L \quad (3-8)$$

```
m.addConstrs((x[i,j,k,l]==0 for (i,j,k,l) in lista if data_aux[i]['grua']<Demanda[j]['grua']),
name='P1.1')
m.addConstrs((x[i,j,k,l]==0 for (i,j,k,l) in lista if data_aux[i]['sonda']<Demanda[j]['sonda']),
name='P1.2')
```

R2. Ausencias

Este grupo de restricciones está pensado para suplir ausencias puntuales, con el objetivo de cubrir sustituciones y de esbozar una posible solución para planificar en un futuro periodos de vacaciones y festivos.

R2.1. Ausencia de auxiliares

En los turnos en los que un auxiliar se ausente, la correspondiente variable de asignación valdrá 0. Los turnos ausentes son un dato conocido a priori. En este modelo, se ha definido la variable $x_{i,j,k,l}$ para cualquier combinación auxiliar-paciente, esté o no asignado previamente.

$$\sum_{j \in J} x_{i,j,k,l} = 0 \quad \forall (i, k, l) \mid Ausencia_a_{i,k,l} = 1 \quad (3-9)$$

```
m.addConstrs((x.sum(i,'*',k,l)==0 for (i,j,k,l) in lista if ausencia_a[i][k-1][1]==1),
name='R2.1')
```

R2.2. Ausencia de pacientes

Al igual que en la restricción anterior, la variable de asignación será forzada a valer 0 en aquellos slots afectados.

$$\sum_{i \in I} x_{i,j,k,l} = 0 \quad \forall (i, k, l) \mid Ausencia_p_{j,k,l} = 1 \quad (3-10)$$

Si el paciente se ausenta durante un número de slots tal que no sea posible abastecer el total de visitas demandadas para ese día, la variable $Ausencia_día_j$ valdrá 1, afectando a las restricciones de demanda que veremos en los siguientes apartados.

Para la implementación en Python de estas restricciones hemos creado dos listas de ceros de tres dimensiones (auxiliar o paciente, día y slot):

- Ausencias_a, con dimensiones I, K, L
- Ausencias_p, con dimensiones J, K, L

Como se comenta en el punto 4.2.1. Conjuntos, los conjuntos de auxiliares, pacientes y turnos comienzan por 0. Sin embargo, el conjunto días tiene como primer valor el día 1 de cada mes.

En Python, las posiciones o índices en una lista comienzan a enumerarse desde el 0. Por ello, para trabajar con la matriz de ausencias podemos usar el mismo valor de i, j y l como índices, y para referirnos a los días tendremos que hacerlo como $k-1$.

Los datos de ausencia pueden cargarse para cualquier día, el modelo se ejecuta para un día seleccionado, por lo que tomará los datos correspondientes a la k seleccionada.

```
#Ausencia un slot
m.addConstrs((x.sum('*',j,k,l)==0 for (i,j,k,l) in lista if ausencia_p[j][k-1][l]==1),name='R2.2')

#Ausencia día completo
m.addConstrs((ausencia_dia[j,k]==1 for (j,k) in pd if disponibilidad[j][k-1]<Demanda[j]['visitas']), name='R2.3')
m.addConstrs((ausencia_dia[j,k]==0 for (j,k) in pd if disponibilidad[j][k-1]>=Demanda[j]['visitas']), name='R2.4')
```

4.3.5. Función Objetivo

En este modelo existen dos objetivos: en primer lugar reducir al mínimo la alteración de las visitas ya programadas para ese día, y en segundo lugar minimizar la penalización por asistencia a pacientes no asignados.

Al reasignar las visitas para un día concreto por ausencia de un único auxiliar o paciente, debemos evitar que el resto de visitas se vean afectadas. Para ello, creamos la variable *desviación_calendario*, que representa la diferencia en valor absoluto entre los valores $x_{i,j,k,l}$ del calendario base ($x_{inicial_{i,j,l}}$) y las $x_{i,j,k,l}$ de la reasignación (nótese que k es un valor fijo). Como vemos, se trata de una ecuación no lineal, pero es admitida sin problemas por el solver empleado.

$$Desviación_calendario = \sum_{i \in I} \sum_{k \in K} \sum_{l \in L} (x_{inicial_{i,j,l}} - x_{i,j,k,l})^2 \quad (3-11)$$

```
m.addConstr(desviacion_calendario==quicksum((x[i,j,k,l]-x_inicial[i,j,l])**2 for (i,j,k,l) in lista ))
```

Lo ideal es que cada paciente sea atendido en cada momento por asistentes asignados. El segundo término que definirá la función objetivo trata de reducir estos casos. Además, los domingos cada hora de sustitución que realiza un auxiliar con contrato L-V duplican su coste, por lo que será aquí donde se tiene en cuenta. El cálculo de la penalización será el siguiente:

$$Penalización = Sustitución_normal\ LV + Sustitución_doble\ LV + Sustitución_otros \quad (3-12)$$

Donde:

Sustitución_normal LV \equiv Sustituciones realizadas por un auxiliar con contrato L-V de lunes a sábado.

Sustitución_doble LV \equiv Sustituciones realizadas por un auxiliar con contrato L-V un domingo.

Sustitución_otros \equiv Sustituciones realizadas por un auxiliar con otro tipo de contrato cualquier día.

```

sust_normal_LV=quicksum(x[i,j,k,l] for (i,j,k,l) in lista if data_aux[i]=='L-V' if j not in
asignaciones_aux[i] if k not in Domingo)

sust_doble_LV=2*quicksum(x[i,j,k,l] for (i,j,k,l) in lista if data_aux[i]=='L-V' if j not in
asignaciones_aux[i] if k in Domingo)

sust_otros=quicksum(x[i,j,k,l] for (i,j,k,l) in lista if data_aux[i]!='L-V' if j not in
asignaciones_aux[i])

m.addConstr((penalizacion==sust_doble_LV+sust_normal_LV+sust_otros))

```

Por parte de la empresa no se han expresado preferencias entre los términos desviación y penalización, por lo que, al presentar mismas magnitudes e importancia relativa, la función objetivo para esta tercera fase será la siguiente:

$$\text{Min (Desviación_calendario + Penalización)} \quad (3-13)$$

4.3.6. Modelo complete fase III

$$\text{Min (Desviación_calendario + Penalización)} \quad (3-13)$$

s.a:

$$\text{Horas_a}_{i,k,l} = \sum_{j \in J} \text{Horas_servidas}_{i,j,k,l} + \text{Traslados}_{i,k,l} + d_{i,k,l} \quad \forall i \in I, k, l \in L \quad (2-1)$$

$$\cdot 15/60$$

$$\text{Horas_servidas}_{i,j,k,l} = \text{Horas_p}_{j,k,l} \cdot x_{i,j,k,l} \quad \forall i \in I, j \in J, k, l \in L \quad (2-2)$$

$$\sum_{i \in I} \text{Horas_servidas}_{i,j,k,l} = \text{Horas_p}_{j,k,l} \cdot \text{Auxiliares_demandados}_j \quad \forall j \in J, k, l \in L \quad (2-3)$$

$$\text{Horas_servidas}_{i,j,k,l} \leq M \cdot x_{i,j,k,l} \quad \forall i \in I, j \in J, k, l \in L \quad (2-4)$$

$$\text{Horas_servidas}_{i,j,k,l} \geq x_{i,j,k,l} \quad \forall i \in I, j \in J, k, l \in L \quad (2-5)$$

$$\text{Traslados}_{i,k,l} = \sum_{j \in J} x_{i,j,k,l} \cdot \text{Desplazamiento}_j / 60 \quad \forall i \in I, k, l \in L \quad (2-6)$$

$$\begin{aligned} \text{Horas_trabajadas}_{i,k,l} & \quad \forall i \in I, k, l \in L \quad (2-7) \\ & = \sum_{j \in J} \text{Horas_servidas}_{i,j,k,l} + \text{Traslados}_{i,k,l} \end{aligned}$$

$$\begin{aligned} \text{Horas_trabajadas}_{i,k,l-1} + \text{Horas_trabajadas}_{i,k,l} & \geq 6 - M \cdot (1 - d1_{i,k,l}) & \forall i \in I, k, l = 1,2 & (2-8) \end{aligned}$$

$$\sum_{l \in L} \text{Horas_a}_{i,k,l} \leq 9 \quad \forall i \in I, k \quad (2-9)$$

$$\sum_{j \in J} x_{i,j,k,l} \geq 1 - M \cdot (1 - \gamma_{i,k,l}) \quad \forall i \in I, k, l \in L \quad (2-10)$$

$$\sum_{j \in J} x_{i,j,k,l} \leq M \cdot \gamma_{i,k,l} \quad \forall i \in I, k, l \in L \quad (2-11)$$

$$\sum_{l \in L} \gamma_{i,k,l} \leq 2 \quad \forall i \in I, k, l \in L \quad (2-12)$$

$$\text{Horas_a}_{i,k,0} + \text{Horas_inicial}_{i,k-1,2} \leq 8 \quad \forall i \in I, k \neq 1 \quad (3-1)$$

$$\text{Horas_a}_{i,k,2} + \text{Horas_inicial}_{i,k+1,0} \leq 8 \quad \forall i \in I, k \neq k_n \quad (3-2)$$

$$\sum_{k \in K} \sum_{l \in L} \text{Horas_p}_{j,k,l} = \text{Demanda}_j \quad \forall j \in J \quad (2-16)$$

$$\sum_{i \in I} \sum_{l \in L} x_{i,j,k,l} = (1 - \text{Ausencia_día}_j) \cdot \text{Demanda_auxiliares}_j \cdot \text{Demanda_visitas}_j \quad \forall j \in J_7, k \quad (3-3)$$

$$\sum_{i \in I} \sum_{l \in L} x_{i,j,k,l} = (1 - \text{Ausencia_día}_j) \cdot \text{Demanda_auxiliares}_j \cdot \text{Demanda_visitas}_j \quad \forall j \in J_5, k \notin K_w \quad (3-4)$$

$$\sum_{i \in I} \sum_{l \in L} x_{i,j,k,l} = 0 \quad \forall j \in J_5, k \in K_w \quad (3-5)$$

$$\sum_{l \in L} \text{Horas_p}_{j,k,l} = 0 \quad \forall j \in J_5, k \in K_w \quad (3-6)$$

$$\sum_{i \in I} \sum_{l \in L} x_{i,j,k,l} = 0 \quad \forall j \in J, k \in K_w \quad (2-19)$$

$$\sum_{l \in L} \text{Horas_p}_{j,k,l} = 0 \quad \forall j \in J, k \in K_w \quad (2-20)$$

$$x_{i,j,k,l} = 0 \quad \forall i \in I_{grúa}, j \in J_{grúa}, k, l \in L \quad (3-7)$$

$$x_{i,j,k,l} = 0 \quad \forall i \in I_{sonda} \quad (3-8)$$

$$j \in J_{sonda}, k, l \in L$$

$$\sum_{j \in J} x_{i,j,k,l} = 0 \quad \forall (i, k, l) \text{ Ausente} \quad (3-9)$$

$$\sum_{i \in I} x_{i,j,k,l} = 0 \quad \forall (i, k, l) \text{ Ausente} \quad (3-10)$$

$$\text{Desviación_calendario} = \sum_{l,j,k} (x_{inicial_{i,j,l}} - x_{i,j,k,l})^2 \quad (3-11)$$

$$\text{Penalización} = \text{Sustitución_normal LV} + \text{Sustitución_doble LV} + \text{Sustitución_otros} \quad (3-12)$$

Finalmente, de este modelo se obtendrá como salida un documento Excel con la reprogramación del día modificado.

5. PRUEBAS Y VALIDACIÓN

Este capítulo realiza un análisis sobre los outputs obtenidos de las fases I, II y III del modelo, con el objetivo de comprobar tanto el funcionamiento básico como la respuesta en algunas situaciones conflictivas. El capítulo se organiza en tres apartados: descripción del set de datos empleado, evaluación de los resultados obtenidos para este set de datos y finalmente la evaluación de la capacidad del modelo para resolver un modelo con un volumen de datos similar al real.

Las pruebas se han realizado en un equipo con una memoria RAM de 8 GB y una CPU Intel Core i5 de doble núcleo.

5.1. Set de datos empleado

El set de datos elaborado para este modelo piloto incluye 120 pacientes y 44 auxiliares. Datos como la localización de cada auxiliar y paciente o las especialidades requeridas se han vinculado a pacientes y auxiliares de manera aleatoria, tratando de mantener características similares a la reales.

5.1.1. Datos de pacientes

La Ilustración 8 muestra las primeras líneas del documento excel empleado. Se han asignado 120 pacientes, con las siguientes características base:

- 80 pacientes demandan visitas durante 5 días a la semana, 40 lo hacen durante 7.
- El 82,5% de los pacientes que requieren 5 días demandan 1 visita y 23 horas mensuales.
- El 15% de los pacientes que requieren 5 días demandan 2 visitas y 46 horas mensuales.
- El 2'5% de los pacientes que requieren 5 días demandan 3 visitas y 69 horas mensuales.
- El 80% de los pacientes que requieren 7 días demandan 1 visita y 31 horas mensuales.
- El 15 % de los pacientes que requieren 7 días demandan 2 visitas y 31 horas mensuales.
- 5 pacientes requieren a un auxiliar especialista en grúas y 5 a un especialista en alimentación por sonda. El paciente 1 requiere ambas.
- Los pacientes 3 y 17 (5 y 7 días demandados respectivamente) precisan ser visitados por dos auxiliares simultáneamente.

Paciente	Horas demandadas	Días demandados	Auxiliares demandados	visitas	desplazamiento	grua	sonda
0	23	5	1	1	10	0	0
1	46	5	1	2	10	1	1
2	46	5	1	2	10	0	0
3	23	5	2	1	10	0	0
4	23	5	1	1	15	0	0
5	62	7	1	2	15	0	0
6	31	7	1	1	15	0	1
7	31	7	1	1	15	1	0

Ilustración 8. Ejemplo de entrada Demanda.xlsx

El hecho de tomar demandas mensuales múltiplos de 23 o 31 se debe a que, en el peor de los casos, un mes tendrá 23 días laborables (para casos de pacientes que demanden 5 días a la semana), y 31 días naturales (para pacientes que demanden 7 días). Cada visita debe tener una duración de al menos 1 hora.

Nótese que, bajo las condiciones impuestas por la empresa, existen combinaciones no factibles (por ejemplo, demandantes de 7 días con 3 visitas diarias, siendo la visita mínima de una hora, supondrían una demanda mensual de 93 horas).

El objetivo de esta demostración es asegurar el funcionamiento de la asignación, planificación y sustituciones bajo las restricciones planteadas, por lo que no repararemos en representar una demanda de horas heterogénea.

5.1.2. Datos de auxiliares

La estructura de los datos de entrada relativos a los auxiliares disponibles se muestra en la Ilustración 9. Se han tenido en cuenta 44 auxiliares con las siguientes características:

- 22 auxiliares tienen contrato tipo L-V.
- 11 auxiliares tienen contrato tipo M-S.
- 11 auxiliares tienen contrato tipo S-L.
- 4 auxiliares están capacitados para manejar grúas, y 5 para alimentar por sonda. Estas dos especialidades coinciden en los auxiliares 0 y 1, que tiene contrato L-V.

Aux	contrato	grua	sonda
0	L-V	1	1
1	L-V	1	1
2	L-V	0	1
3	L-V	0	0
4	L-V	0	0
5	L-V	0	0

Ilustración 9. Ejemplo de entrada Auxiliares.xlsx

5.2. Evaluación de los resultados obtenidos

5.2.1. Ejecución del modelo básico (fases I y II)

De la ejecución del primer modelo obtenemos el Excel Asignaciones.xlsx, con tantas filas como asignaciones sean necesarias. La siguiente ilustración recoge algunas de las asignaciones.

Paciente	Auxiliar
0	3
1	0
2	4
3	12
3	21
4	10
5	23
5	39
6	23
6	34
7	22
7	33

Ilustración 10. Ejemplo salida Fase I

Observamos por ejemplo que el paciente 3 tendrá 2 auxiliares asignados, ya que como muestra la Ilustración 8 este requerirá la visita de 2 auxiliares simultáneamente. Asimismo, comprobamos que los pacientes 6 y 7, al demandar 7 días, tienen cada uno dos auxiliares asignados.

Por último, volviendo a la Ilustración 8, observamos que el paciente 1 requiere a especialistas en grúa y sonda. En el apartado 5.1.2. se detalla que los auxiliares que poseen estas características son los identificados como 0 y 1, y en el ejemplo de la asignación podemos comprobar que, en efecto, el paciente 1 estará asignado al auxiliar 0. La Ilustración 11 muestra la representación de todos los usuarios.

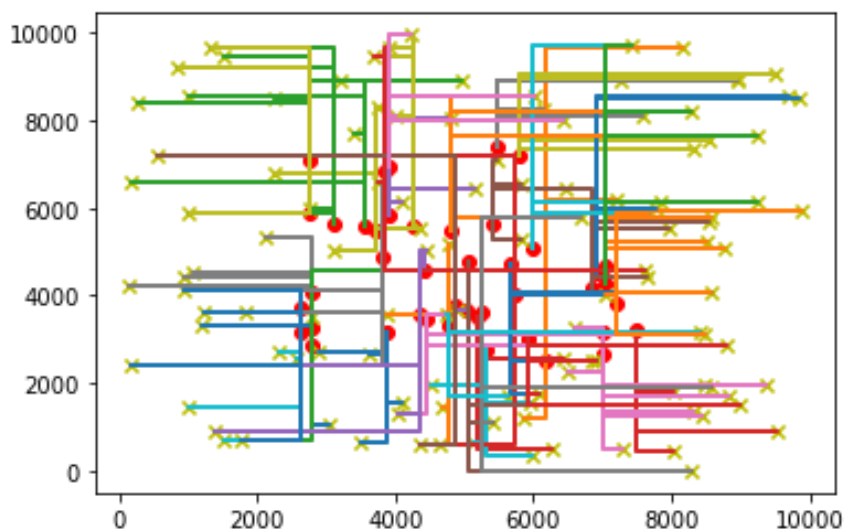


Ilustración 11. Asignación de auxiliares y pacientes

El segundo modelo, correspondiente a la Fase II, requiere tres entradas: las dos mencionadas anteriormente (Demanda.xlsx y Auxiliares.xlsx), y el Excel obtenido como salida de la Fase I.

Para una mejor comprensión de los datos, ejecutaremos el modelo para agosto de 2022. Por contar con 31 días naturales y 23 días laborables, se adapta de manera exacta al set de datos diseñado.

La Ilustración 12 muestra un ejemplo del output obtenido. Observamos que el paciente 1, que estaba asignado al asistente 0 y requería dos visitas, es visitado dos veces el día 1 de agosto.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Día	Lunes										
2	Auxiliar	Paciente	Turno	Horas								
3	0	1	Mañana	1								
4	0	1	Tarde	1								
5	0	11	Mañana	1								
6	0	15	Mañana	1								
7	0	71	Mañana	1								
8	1	19	Mañana	1								
9	1	79	Mañana	1								
10	1	113	Mañana	1								
11	1	113	Tarde	1								
12	1	118	Mañana	1								
13	1	118	Tarde	1								
14	2	10	Mañana	1								
15	2	29	Tarde	1								
16	3	0	Mañana	1								
17	3	30	Mañana	1								
18	3	58	Mañana	1								
19	3	61	Mañana	1								
20	4	2	Mañana	1								
21	4	2	Tarde	1								
22	4	80	Tarde	1								
23	4	87	Tarde	1								
24	4	98	Mañana	1								
25	5	49	Noche	1								
26	5	56	Noche	1								
27	5	85	Noche	1								
28	5	107	Mañana	1								

Ilustración 12. Ejemplo de planificación obtenida

La Ilustración 13 muestra las visitas planificadas para el paciente 3. Como mencionábamos anteriormente, este paciente requiere la visita de sus dos auxiliares asignados simultáneamente y comprobamos que, en efecto, esto se cumple.

	A	B	C	D	E	F
1	Día	Lunes				
2	Auxiliar	Paciente	Turno	Horas		
56	12	3	Noche	1		
97	21	3	Noche	1		
147						
148						
149						
150						
151						
152						
153						

1-August 2-August 3-August

Ilustración 13. Detalle de planificación obtenida

Las horas que contabilizarán en el contrato de cada auxiliar (horas de servicio, traslados y descansos) son las mostradas en la Ilustración 14.

	A	B	C	D
1	Auxiliar	Tipo de contrato	Horas mensuales	
2	0	L-V	109,27	
3	1	L-V	136,11	
4	2	L-V	134,21	
5	3	L-V	134,21	
6	4	L-V	134,21	
7	5	L-V	136,11	
8	6	L-V	136,11	
9	7	L-V	113,09	
10	8	L-V	109,27	
11	9	L-V	107,36	
12	10	L-V	145,68	
13	11	L-V	136,11	
14	12	L-V	127,61	
15	13	L-V	109,27	
16	14	L-V	107,36	
17	15	L-V	134,21	
18	16	L-V	134,21	
19	17	L-V	53,68	
20	18	L-V	143,77	

Ilustración 14. Horas mensuales asignadas a cada auxiliar

El tiempo de ejecución de la Fase II ha sido de 9 segundos.

5.2.2. Ejecución del modelo básico con ausencias

En la siguiente prueba verificaremos el modelo correspondiente a la fase III. Incluiremos los siguientes supuestos:

Supuesto 1: el paciente 1 requiere a un especialista en grúa y sonda durante dos visitas. Actualmente le atiende el auxiliar 0, que se ausentará el 1 de agosto al completo.

Supuesto 2: el paciente 2 requiere 2 visitas, comprobaremos si al ausentarse durante 2 turnos el día 1 de agosto

las visitas para ese día quedan canceladas

Supuesto 3: el paciente 3 requiere la visita simultánea de los auxiliares 12 y 21. El día 1 el auxiliar 12 se ausentará durante por la tarde y noche (quedando disponible únicamente por la mañana), y ese mismo día el auxiliar 21 se ausentará el día completo.

Supuesto 4: El paciente 7 requiere el uso de grúa, tiene un régimen de visitas de 7 días a la semana y es atendido actualmente por los auxiliares 22 (contrato M-S) y 33 (contrato S-L). En nuestro set de datos los únicos especialistas en sonda con contrato el domingo son ellos dos. El domingo 7 de agosto se ausentarán ambos, forzando a que algún auxiliar especialista y con contrato L-V tenga le atiende.

Ante estos datos el tiempo de ejecución permanece bajo: 6 segundos.

La Ilustración 15 recoge las modificaciones en el calendario por parte del programa para los supuestos 1, 2 y 3.

	A	B	C	D	E
1	Día	Lunes			
2	Auxiliar	Paciente	Turno	Horas	
3	3	0	Mañana	1	Sustitución
4	1	1	Mañana	1	Sustitución
5	1	1	Tarde	1	
6	4	3	Mañana	2	
7	12	3	Mañana	2	
8	10	4	Noche	1	
9	39	5	Mañana	1	
10	39	5	Tarde	1	
11	23	6	Mañana	1	
12	22	7	Mañana	1	
13	33	8	Mañana	1	
14	34	9	Mañana	1	
15	2	10	Mañana	1	
16	33	11	Tarde	1	Sustitución
17	27	12	Mañana	1	
18	35	13	Mañana	1	Sustitución
19	30	14	Mañana	1	
20	33	15	Mañana	1	
21	34	16	Noche	1	
22	22	17	Noche	2	

Ilustración 15. Resultados tras aplicar ausencias en Fase III

Comprobamos que el auxiliar 0 ya no atiende este día al paciente 1. Como se comenta en apartados anteriores, el paciente 1 tan solo puede ser atendidos por especialista en grúa y sonda. La sustitución la llevará a cabo el auxiliar 1, que cumple con estos requisitos, por lo que podemos validar el supuesto 1.

El paciente 2 se ha ausentado durante mañana y tarde, quedando disponible únicamente en el tercer turno. Sin embargo, una de las condiciones impuestas era que, de no ser posible recibir todas las visitas, ese día pasaría a estar ausente. El paciente 2 demanda 2 visitas y tan solo está disponible durante un turno, por lo que este día no será atendido.

Por último, respecto al paciente 3 comprobamos dos aspectos. En primer lugar, uno de sus auxiliares demandados tan solo tendrá disponible el primer turno, por lo que la visita tendrá que ser a esta hora. Asimismo, el auxiliar 21, que se ausentaría la jornada completa, será sustituido por el auxiliar 4.

La Ilustración 16 hace referencia al supuesto 4. El paciente 7 requiere especialistas en grúa y sonda un domingo, no obstante, todos los los auxiliares con un contrato que incluya el domingo y estas especialidades están ausentes. Por ello, la visita será efectuada por el auxiliar 0 pese a tener contrato tipo L-V, cumpliendo las restricciones de especialidad, pero penalizando la función objetivo. Como describíamos en el problema, las horas de trabajo

efectuadas un domingo sin que este día este incluido en el contrato contabilizarán el doble. El valor del término penalización dentro de la función objetivo es de 2.

	A	B	C	D	E
1	Día	Domingo			
2	Auxiliar	Paciente	Turno	Horas	
3	0	7	Mañana	1	Sustitución
4	0	8	Mañana	1	Sustitución
5	23	16	Mañana	1	
6	24	68	Mañana	1	
7	25	69	Mañana	1	
8	26	53	Mañana	1	
9	26	76	Noche	1	
10	27	47	Mañana	1	
11	28	37	Mañana	1	
12	28	78	Mañana	1	
13	29	38	Mañana	1	
14	30	39	Mañana	1	
15	30	83	Mañana	1	Sustitución
16	30	83	Noche	1	Sustitución
17	31	42	Noche	1	
18	31	46	Noche	1	
19	31	72	Mañana	1	

Ilustración 16. Resultados tras aplicar ausencias en Fase III

Hechas estas comprobaciones, podemos afirmar que la programación de ausencias, tanto por pacientes como por asistentes, es posible, lo que puede suponer el inicio del diseño de un modelo para la programación de vacaciones y festivos.

5.2.3. Ejecución del modelo con un volumen de datos realista

El objetivo de este proyecto es crear un modelo piloto escalable -en un futuro- a los datos empleados en la organización. Por ello, es importante confirmar que el sistema en sus tres fases tiene un comportamiento correcto en volúmenes de datos superiores.

En este caso, ante las comprobaciones del apartado previo, no nos detendremos en crear situaciones especiales como números de visitas o asistentes demandados: nuestro objetivo en este caso es confirmar que el sistema es válido cuando el número de variables y restricciones crece. A priori, se contemplan empeoramientos en los tiempos de ejecución.

Utilizaremos un set de datos estandarizado, con 630 demandantes y 250 auxiliares:

- 510 pacientes demandan 5 días a la semana, 1 visita, 1 auxiliar por visitas y 23 horas mensuales.
- 120 pacientes demandan 7 días a la semana, 1 visita, 1 auxiliar por visita y 31 horas mensuales.
- 170 auxiliares con contrato L-V.
- 40 auxiliares con contrato M-S.
- 40 auxiliares con contrato S-L.
- No se demandan especialidades.

Ante estos datos, los datos para las fases I y II han sido:

- Fase I: se han generado 158006 variables continuas y 157750 variables enteras binarias. El tiempo de resolución ha sido de 12 segundos.
- Fase II: se han generado 174842 variables continuas y 116250 variables enteras binarias. El tiempo de resolución ha sido de 42 segundos.

Para la fase III, aplicamos las mismas ausencias propuestas en el apartado 5.2.2, obteniendo 475892 variables continuas, 474630 variables enteras y un tiempo de resolución de 92 segundos.

A la vista de los resultados en los apartados de este capítulo, podemos considerar el modelo efectivo y rápido, convirtiéndose en una potencial alternativa a la metodología actual de gestión de visitas.

5.3. Limitaciones del modelo

El modelo no presenta limitaciones de funcionamiento para las consideraciones tenidas en cuenta durante su diseño.

Sin embargo, por tratarse de un modelo piloto, existen limitaciones a la hora de incluir y modificar parámetros. En algunos aspectos como el mes de planificación o las ausencias será necesario manipular el código para realizar modificaciones.

Asimismo, es necesario partir de un set de datos factible. Se entiende por set de datos factible aquel que da lugar al menos a una solución admisible en la fase I del modelo. Es decir, debe comprobarse previamente, o tenerse la certeza, de que el número de auxiliares es suficiente para cubrir todas las visitas demandadas, teniendo en cuenta contratos y especialidades.

6. CONCLUSIONES

El presente proyecto perseguía la creación de un sistema piloto para la asignación y planificación de visitas de trabajadores sociales a personas dependientes. El trabajo ha pasado por estudiar detalladamente la problemática, conocer los tipos de perfiles que se ven involucrados en la asignación, plantear una solución, diseñar un modelo adaptado a esta y su posterior implementación y validación. Ante los resultados obtenidos y como se expone a continuación, podemos considerar exitoso el diseño de este sistema piloto, validando su posible adaptación e implantación tanto en la empresa como en organizaciones similares.

6.1. Consecución de objetivos

A continuación, se evalúan los seis objetivos específicos marcados al inicio del proyecto:

1. Se ha analizado el problema actual dentro de la empresa, y se han planteado como aspectos de mejora el método actual de planificación de visitas y la asignación de pacientes y demandantes vigente. Se ha efectuado una revisión de la literatura, que ha llevado a conocer problemas similares ya planteados y modelados.
2. Se ha realizado un análisis detallado del problema, detallando aspectos relacionados a la jornada laboral de los auxiliares, los tipos de contrato y diferentes requisitos por parte de los demandantes. Una vez descritas las características del problema, se han simplificado sus características -manteniendo todas las funciones principales a tener en cuenta-, y se ha propuesto una solución compuesta por tres fases consecutivas que solventarán los diferentes aspectos.
3. Se han modelado las tres fases del problema: asignación, planificación de visitas y replanificación por ausencias. Para ello, se han descrito variables, restricciones y función objetivo en cada caso. Se han incluido los fragmentos de código Python correspondientes a cada restricción descrita.
4. Se han realizado pruebas en el modelo para verificar su efectividad y escalabilidad. Para ello, se ha diseñado un set de datos de tamaño medio, y un set de datos básico con un volumen de datos similar al manejado actualmente por la empresa.
5. Se han detectado los aspectos positivos del modelo (rapidez, versatilidad y posible aplicación a gestión de festivos y vacaciones), así como las limitaciones del sistema diseñado (el modelo requiere una combinación de datos factibles, y ciertos parámetros solo podrán ser modificados manipulando el código).
6. Se propone en el apartado 6.3. la línea de trabajo futura para el proyecto.

6.2. Valoración del desempeño

El modelado matemático y su programación requieren un estudio exhaustivo del problema a abordar, incluyendo el análisis de las variables, restricciones y funciones objetivo que entran en juego y la finalidad de este: en qué tipo de equipo se ejecutará, con qué frecuencia, qué tiempo de ejecución se espera, etc.

En cuanto al aspecto temporal, es importante establecer un límite acorde al propósito del modelo. Por ello, cuando se trata de problemas con un volumen considerable de datos, es necesario evaluar su rendimiento a escalas similares a las reales.

Del diseño de este modelo obtenemos como conclusiones generales:

- La importancia de un buen estudio previo de las características del problema, y en especial de la estructura de la solución propuesta, pues de ella dependerá que el modelo sea escalable o no. Concretamente, el hecho de dividir el problema en tres fases ha facilitado significativamente la comprensión y resolución del caso, además de aportar flexibilidad en la reprogramación de días.
- Asimismo, la simplificación de variables juega un papel fundamental. Nos encontramos ante un modelo que, si no es simplificado, puede superar los 25 millones de variables si se trabaja con datos cercanos a los reales: aquellas variables que cuenten con los subíndices i , j , k y l combinarían 600 auxiliares, 250 pacientes, 31 días y 3 slots. Esto hace que su resolución sea compleja incluso para ordenadores de una potencia muy superior a la de un ordenador de oficina. Por lo que prestar atención al volumen de variables y a la posibilidad de simplificación es imprescindible. De este modo evitamos crear variables innecesarias, que si bien no afectan demasiado con volúmenes bajos, a escalas mayores puede resultar decisivo.

Con estas consideraciones y ante los resultados obtenidos durante la fase de pruebas, podemos dar por concluido satisfactoriamente el diseño del modelo piloto para la asignación, planificación y replanificación de visitas a pacientes dependientes

6.3. Futuras líneas de trabajo

El Proyecto ha permitido confirmar la viabilidad de la semi-automatización del proceso de planificación de visitas a personas dependientes, ofreciendo una solución cuantitativa rápida de una eficiencia potencialmente superior a la actual. Dado este primer paso, se abre un listado de acciones futuras si, vistos los resultados, se desea implementar la metodología en una empresa.

1. Como se menciona en el capítulo 3, este proyecto ha sido enfocado desde un punto de vista académico y con el objetivo de comprobar si es posible automatizar la gestión de visitas. Por ello no se han tenido en cuenta días festivos, vacaciones, y los contratos que suplen las ausencias en estos acontecimientos. Confirmada la factibilidad de la implantación del sistema, el siguiente paso consistirá en completar el modelo incluyendo estos supuestos descartados, así como la implementación de una bolsa de horas que tenga en cuenta ausencias y sustituciones de cada auxiliar.
2. Alineado con el primer punto, con el objetivo de desarrollar un sistema realista debe incluirse la posibilidad de que los demandantes expresen sus preferencias en turnos para la programación de las visitas.
3. Asimismo, es importante desarrollar una interfaz que permita al usuario interactuar con el modelo sin necesidad de modificar el código.
4. Informatizar la gestión de visitas conlleva como beneficio colateral la posibilidad de generación almacenamiento de datos como horas trabajadas o porcentajes de ausencia, que podrán ser tratados con fines estadísticos y analíticos. Existe por tanto la posibilidad de realizar análisis y previsiones en pro de los intereses de la empresa.
5. Resulta interesante la implementación de los modelos en un software gratuito. El solver empleado requiere una licencia costosa, que puede llegar a anular los beneficios obtenidos con este Proyecto. Por

ello, se propone la implementación de los diferentes modelos planteados en un solver gratuito como GLPSOL o lp_sol.

6.

BIBLIOGRAFÍA

- Alfares, H. K. (2004). Survey, Categorization, and Comparison of Recent Tour Scheduling Literature. *Annals of Operations Research*, 127, 145–175.
- Arabeyre, J., Fearnley, J., Steiger, F., & Teather, W. (1969). The Airline Crew Scheduling Problem: A Survey. *Transportation Science*, 3(2), 140-163.
- Bartholdi, J. J. (1980). Cyclic Scheduling via Integer Programs with Circular Ones. *Operations Research*.
- Bechtold, S., & Jacobs, L. (1990). Implicit Modeling of Flexible Break Assignment in Optimal Shift Scheduling. *Management Science*(36), 1339–1351.
- Brusco, M., & Johns, T. (1995). Improving the Dispersion of Surplus Labor in Personnel Scheduling Solutions. *Computers and Industrial Engineering*(28), 745–754.
- Corominas, A., & Pastor, R. (2009). Replanning working time under annualised working hours. *International Journal of Production Research*, 1493-1515.
- Corominas, A., & Pastor, R. (2009). Replanning working time under annualised working hours. *International Journal of Production Research 2009*, 1493-1515.
- Corominas, A., Lusa, A., & Pastor, R. (2004). Using MILP to plan annualised working hours. *Journal of the Operational Research Society*(53), 1101-1108.
- Ernst, A., Jiang, H., Krishnamoorthy, M., & Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*(153), 3-27.
- Guastaroba, G., Côté, J.-F., & Coelho, L. (2020). The Multi-Period Workforce Scheduling and Routing Problem. *Omega*.
- Hüseyin Özder, E., Evrencan, Ö., & Tamer, E. (2020). A Systematic Literature Review for Personnel Scheduling Problems. *International Journal of Information Technology & Decision Making*, Vol. 19, No. 06, pp. 1695-1735.
- Rekik, M., Cordeau, J., & Soumis, F. (2010). Implicit shift scheduling with multiple breaks and work stretch duration restrictions. *Journal of Scheduling*, 49-75.
- Robinson, J., Shugure, N., Fortinsky, R., & Gruman, C. (2014). Long-Term Supports and Services Planning for the Future: Implications From a Statewide Survey of Baby Boomers and Older Adults. *The Gerontologist*, 297-313.
- Thompson, G. (1995). Labor Scheduling Using NPV Estimates of the Marginal Benefit of Additional Labor Capacity. *Journal of Operations Management*, 67-86.

