



Universidade de Vigo

Traballo Fin de Máster

Modelos de optimización matemática para a planificación de citas de radioterapia

Jorge Rial Fondo

Máster en Técnicas Estatísticas

Curso 2025-2026

Proposta de Traballo Fin de Máster

Título en galego: Modelos de optimización matemática para a planificación de citas de radioterapia
Título en español: Modelos de optimización matemática para la planificación de citas de radioterapia
English title: Optimization models to schedule radiotherapy sessions on oncologic patients
Modalidade: Modalidade A
Autor/a: Jorge Rial Fondo, Universidade de Santiago de Compostela
Director/a: Ángel Manuel González Rueda, Universidade de Santiago de Compostela
<p>Breve resumo do traballo:</p> <p>Segundo a Organización Mundial da Saúde (OMS), o cancro é unha das principais causas de morte en todo o mundo. Existen diversos tipos de tratamento, sendo un dos máis habituais a radioterapia, que a miúdo combínase con quimioterapia ou outros tratamentos. Na radioterapia, os pacientes reciben, en cada sesión, unha dose de radiación co obxectivo de destruír células cancerosas, procedemento que se fai en máquinas coñecidas como aceleradores lineais. A planificación horaria de citas de radioterapia constitúe un problema complexo de optimización: procura asignar pacientes a aceleradores e determinar os horarios das súas sesións, respectando restricións clínicas (por exemplo, ventás de tempo preferidas, frecuencia de sesións e capacidade dos aceleradores) e buscando obxectivos como minimizar tempos de agarda, maximizar a utilización de recursos e garantir a continuidade do tratamento. Neste traballo realizarase unha revisión dos principais modelos de optimización matemática presentes na literatura para abordar este problema. Ademais da análise dos distintos tipos de modelos, propoñeranse posibles extensións, reformulacións e algoritmos que permitan resolver o problema de maneira máis eficiente. Dende un punto de vista aplicado, explorarase a implementación das propostas e validaranse computacionalmente empregando datos reportados na literatura.</p>
Recomendacións:
Outras observacións:

Don Ángel Manuel González Rueda, Profesor Axudante Doutor da Universidade de Santiago de Compostela, informa que o Traballo Fin de Máster titulado

Modelos de optimización matemática para a planificación de citas de radioterapia

foi realizado baixo a súa dirección por don Jorge Rial Fondo para o Máster en Técnicas Estatísticas. Estimando que o traballo está terminado, dan a súa conformidade para a súa presentación e defensa ante un tribunal. Ademais, Don Ángel Manuel González Rueda e don Jorge Rial Fondo

si non

autorizan a publicación da memoria no repositorio de acceso público asociado ao Máster en Técnicas Estatísticas.

En Santiago de Compostela, a 19 de maio de 2026.

O director:
Don Ángel Manuel González Rueda

O autor:
Don Jorge Rial Fondo

Declaración responsable. Para dar cumprimento á Lei 3/2022, de 24 de febreiro, de convivencia universitaria, referente ao plaxio no Traballo Fin de Máster (Artigo 11, [Disposición 2978 do BOE núm. 48 de 2022](#)), **o/a autor/a declara** que o Traballo Fin de Máster presentado é un documento orixinal no que se tiveron en conta as seguintes consideracións relativas ao uso de material de apoio desenvolto por outros/as autores/as:

- Todas as fontes usadas para a elaboración deste traballo foron citadas convenientemente (libros, artigos, apuntamentos de profesorado, páxinas web, programas, . . .)
- Calquera contido copiado ou traducido textualmente púxose entre comillas, citando a súa procedencia.
- Fíxose constar explicitamente cando un capítulo, sección, demostración, . . . sexa unha adaptación case literal dalgunha fonte existente.

E, acepta que, se se demostrara o contrario, se lle apliquen as medidas disciplinarias que correspondan.

Agradecementos

Agradécese a colaboración e inestimable axuda de meu director de Traballo de Fin de Máster, Ángel Manuel González Rueda, tanto polos seus abundantes coñecementos como a nivel de xestión e organización para poder levar a remate este traballo. Agradezo tamén ao Colexio Maior Rodríguez Cadarso, como medio que me fixo ser quen son e chegar a publicar este documento. Finalmente, agradezo a meus pais, amigos eternos e a toda persoa que se dea como aludida cun “te quero” meu.

Índice xeral

Resumo	XI
Prefacio	XIII
1. Estado do arte da optimización en medicina	3
1.1. Definición inicial do noso problema	3
1.2. Historia da optimización en citas médicas	5
1.3. Estado do arte da planificación en radioterapia	8
2. Modelo proposto para solucionar o RTSP	17
2.1. Primeira parte: asignación de día e <i>linac</i> para o tratamento	21
2.2. Segunda parte: priorizar preferencias temporais dos pacientes	25
3. Implementación e análise computacional	33
3.1. Creación de instancias	33
3.2. Implementación e requisitos técnicos	37
3.3. Análise computacional da primeira parte do modelo	41
3.3.1. Caso 1: $\omega_1 = 1000, \omega_2 = \omega_3 = 1, \gamma = 0$	42
3.3.2. Caso 2: $\omega_1 = 1000, \omega_2 = \omega_3 = 1, \gamma = 0.1$	46
3.3.3. Caso 3: $\omega_1 = \omega_3 = 1, \omega_2 = 1000, \gamma = 0$	48
3.4. Análise computacional da segunda parte do modelo	50
3.4.1. Caso 1: $\omega_4 = 60, \omega_5 = \omega_6 = 1$	51
3.4.2. Caso 2: $\omega_4 = \omega_5 = \omega_6 = 1$	53
4. Conclusión	55
A. Exemplo de formato das instancias iniciais: ins01	57
B. Código conxunto de ambas partes	61
C. Gráficas da primeira parte	75

Resumo

Resumo en galego

O gran aumento de pacientes oncolóxicos que requiren tratamento radioterapéutico xera unha gran tensión nos centros que atendan este servizo. Ao dispoñer dun número limitado de dispositivos que poden realizar esta terapia (aceleradores lineais) están na obriga de xestionar unha lista de agarda cada vez máis saturada con pacientes de diversos graos de urxencia.

O problema de planificación de citas de radioterapia (ou RTSP) procura dar solución a este reto. Na primeira fase do traballo dedicáronos a estudar como se relaciona con outros problemas de índole similar, introducir cales son os precedentes desta liña de investigación e, finalmente, dar paso a modelos xa propostos que o resolven.

A continuación, escollemos un dos modelos xa referendados que debemos analizar en detalle. Tu-San Pham, Louis-Martin Rousseau e Patrick De Causmaecker expoñen un enfoque en dúas fases. Na primeira parte búscase optimizar o tempo de atención dos pacientes; despois teranse en conta obxectivos para facilitar a atención continuada do paciente, como respectar as súas ventás temporais de preferencia de atención.

Finalmente, procederase a implementar en Python o modelo e ver como se comporta ante datos simulados baseados na realidade. O obxectivo é empregar estes datos para analizar se o modelo é realmente capaz de ser adaptativo a aquelas prioridades do centro e eficaz de cara a dar un horario de citación nun tempo razoable.

English abstract

The large increase in oncological patients requiring radiotherapy treatment is a highly volatile situation in any oncologic center. They have to manage an ever-increasing waiting list of patients, who have varying degrees of urgency, with a limited supply of devices that can perform this therapy (linear accelerators).

The radiotherapy scheduling problem (or RTSP) seeks to provide a solution to this challenge. First, we will study how it feeds from other problems of similar nature, introduce the precedents of this line of research and show models who have been built already for this purpose.

Subsequently, we analyze in detail one of those well-established models. Tu-San Pham, Louis-Martin Rousseau and Patrick De Causmaecker present a two-phase approach. The first objective is to reduce the waiting time of the patients; then, the quality of the patient's treatment will be taken into account, for example, by respecting their time window preferences.

In the end, we will implement the model and simulate data based on actual oncologic centers. The aim is to use this data to analyze whether the model is both capable of being adaptive to the center's priorities and also effective in providing a citation schedule in a reasonable time.

Prefacio

A orixe do traballo provén do meu interese xenuíno pola investigación operativa, xa dende os meus estudos no grao en Matemáticas. Xa daquela decanteime por este campo no meu traballo de fin de grao, baseándome nun estudo teórico da programación multiobxectivo. Desta rolda, preferín optar por un enfoque máis práctico en optimización discreta, traballando cunha problemática concreta e situada na súa totalidade na realidade.

A temática do proxecto conecta de maneira natural a calquera persoa; como xa se repetirá múltiples veces no traballo, hai tal incidencia dos casos de cancro que todos temos achegados afectados por esta enfermidade, ao cal hai que engadir a gran variedade de perfís das persoas afectadas. Este proxecto pretende por tanto aportar o seu gran de area axudando, dende a optimización, a poder ofrecer uns mellores servizos sanitarios aos pacientes oncolóxicos.

Esta investigación, en resumo, consiste en proporcionar un sistema para xestionar citas de radioterapia, pensando nun hospital con limitados recursos e horarios, e traballando en bloques de tempo fixos. Para isto, debemos simultaneamente ter en conta esta limitación material e cuestións sociais: os horarios preferentes dos pacientes, non modificar dun día para outro a súa cita cos efectos na vida laboral e familiar que implica,...

Con isto en mente, propoñemos un MILP (problema de programación lineal enteira mixta) que implementamos en Python a través do paquete Pyomo, baseándonos en modelos preexistentes na literatura. Neste proceso, vamos contextualizar as súas características para confirmar a súa correspondencia co dilema real. Finalizamos analizando a súa efectividade con diferentes bases de datos mostrando os horarios propostos e o tempo de execución necesario para a súa resolución.

Debemos salientar que foi ademais un proceso dinámico, xa que hai un conflito natural entre engadir maior interacción por parte do usuario ou engadir novas particularidades médicas e a súa efectividade á hora de ser operado. Por tanto, considero que presentar un modelo inspirado na investigación feita ata agora e observar como se comporta pode ser un punto de partida moi importante ante novas expansións ou modificacións necesarias ante novas realidades.

Introdución

As mortes por tumores malignos (cancro) son a principal causa de morte en España e a segunda principal en Galicia, só por detrás das mortes por causas cardiorrespiratorias (INE, 2024). Isto fai que sexa unha enfermidade prioritaria á hora tanto de investigala como de tratala, pero presenta varios retos a afrontar.

Un dos tratamentos máis comúns, xa sexa en conxunto con outras terapias como a cirurxía ou a quimioterapia ou de forma individual, é a **radioterapia**. En resumidas contas, consiste en aplicar radiación ás células cancerosas de forma local no corpo, destruíndo o seu ADN e evitando que se repliquen. Estímase que ata o 60 % dos casos de cancro en España requirirán deste método (Fernández *et al.*, 2020), e posúe múltiples finalidades: curativa (ao destruír estas células, redúcese o tamaño do tumor e impide o seu desenvolvemento), como paso previo a unha cirurxía para eliminar o tumor, con función paliativa para reducir dano en órganos afectados...

A subministración deste tratamento faise principalmente a través dos aceleradores lineais (*linacs* en inglés) (OIEA, 2026), sen necesidade, en xeral, de hospitalización. Son aparatos que emiten raios X de alta enerxía ou electróns, cuxa precisión e fiabilidade foi aumentando co paso do tempo ata os nosos días. O seu avance máis destacable foi a incorporación dun escáner e simulación 3D, co fin de reducir a cantidade de radiación que afecta a células sas. Tamén existen outros métodos de administración de radiación médica (Fernández *et al.*, 2020) como a braquiterapia ou a protonterapia.

Segundo o informe da AECC (Fernández *et al.*, 2020), estes aceleradores lineais son insuficientes (3.9 en 2020 por cada millón de habitantes fronte aos 7 propostos pola Sociedade Española de Oncoloxía Radioterápica) nos hospitais públicos, e, ademais, había un gran porcentaxe deles (42 %) que estaban obsoletos. Isto implica que hai moitos hospitais que non teñen suficientes aceleradores lineais para atender os seus pacientes (de feito, hai provincias que non dispoñen de ningún!).

Hai que engadir que o tratamento supón un elevado custo tanto para o paciente (múltiples sesións de radioterapia, transporte ata o hospital, efectos secundarios do tratamento) como para os hospitais, polo que procurase evitar que estean inutilizados os aceleradores lineais. Malia isto, debido ao mantemento cotián que hai que realizarlle ás máquinas e que moitos pacientes incorren nun custo moi elevado a nivel social, emocional e económico (Fernández *et al.*, 2020) que provoca que deixen o tratamento, temos situacións na que unha planificación a medio prazo do seu uso pódese ver interrompida facilmente.

Estes efectos fan que os tempos de agarda ata chegar a iniciar o tratamento sexan moi elevados, aos cales xa se engaden o tempo dende a propia detección da enfermidade en atención primaria (se non vén dun cribado), derivación ao hospital do paciente, e chegar ao diagnóstico correcto. De feito, vese en Sánchez Galindo *et al.* (2025), no caso concreto do cancro de mama (un dos máis comúns, pois representa aproximadamente unha cuarta parte dos tumores malignos), temos que o tempo de inicio de tratamento dende a confirmación do diagnóstico é, na maioría dos casos, superior aos 40 días, chegando a rexistrar en casos extremos 270 días de agarda.

Isto xa é preocupante e esta tendencia trasládase a varios tipos de tumores (Cañaveras León *et al.*, 2022), como o de vexiga, próstata, pulmón e colorrectal, sendo este último tamén dos máis frecuentes e facéndose múltiples cribados a varios sectores da poboación. Cos datos dispoñibles neste estudo, a gravidade dos tempos ata o tratamento do cancro de mama está diluída por estes prazos para os casos de cancro de próstata ou vexiga. Aquí son moito maiores, chegando a máis de 70 días de agarda. De feito, chegaron a detectar que un 25 por cento dos seus pacientes con cancro de próstata deben esperar

máis de 200 días en ser tratados.

Non debemos esquecer que a enfermidade, que se orixina por un tecido restrinxido nunha parte do corpo concreta, pero, se aumenta moito de tamaño, parte das células poden chegar ao torrente sanguíneo, derivando en metástases, fallo multiorgánico,... polo que se a un tempo de detección relativamente alto debido á in advertencia aparente da enfermidade lle engadimos un tempo de tratamento tardío, podemos estar falando de miles de persoas cuxa vida se pon en perigo debido a este punto de inflexión.

Ante esta situación, que se pode aportar dende a optimización? Vamos a intentar mellorar o máximo posible, cos recursos restrinxidos, o tempo de espera da intervención dun paciente oncolóxico, centrándonos no contexto das citas para sesións de radioterapia a través dun problema de programación matemática que se poida resolver de maneira eficiente. Isto nos permitiría atallar situacións diversas de recursos dispoñibles, evitar o desaproveitamento dos aceleradores lineais mentres que buscamos reducir este tempo total dende a detección da enfermidade ata o inicio do tratamento.

Este problema de planificación de citas de radioterapia (ou RTSP, (Pham *et al.*, 2022), *Radiotherapy Scheduling Problem*, en inglés) ten xa moito percorrido, polo que antes de nada debemos ver en que punto esta a investigación actual, presentar un modelo baseado nesta bibliografía de referencia e, finalmente, implementalo e ver como se comporta ante casos baseados na realidade.

Capítulo 1

Estado do arte da optimización en medicina

1.1. Definición inicial do noso problema

O problema xa presentado ten base dentro dos problemas de optimización matemática en medicina. Esta clase de problemas foron recollidos e estudados en múltiples fontes, como en [Hulshof *et al.* \(2012\)](#) ou en [Ahmadi-Javid *et al.* \(2017\)](#). Estas serán referencias moi importantes de cara a realizar un estudo de cales son as vías de traballo neste campo.

Partindo deste concepto, podemos enfocar os problemas de dúas formas que veremos a continuación. Primeiro, vamos a facer unha división temática segundo a área médica á que pertencen, seguindo o modelo proposto en [Hulshof *et al.* \(2012\)](#):

- Servizos ambulatorios de coidado. Poden ser tanto en clínicas externas ou parte dun hospital como tal, pero sempre implican a **non estancia do paciente no lugar de tratamento**. En xeral, trátanse de problemas que deben responder a situacións con limitación de persoal, de equipos ou recursos. Exemplos dentro desta categoría serían os problemas en servizos de endoscopia, radioloxía xeral e, o que nos preocupa, **radioterapia**, no que se inclúe a situación proposta na introdución.
- Servizos de emerxencia. O contexto pode ir dende xestionar o fluxo de pacientes en traumatoloxía ou urxencias dun hospital como á distribución de puntos de ambulancia co fin de atender rapidamente aos pacientes ([Hashemi *et al.*, 2022](#)).
- Servizos de cirurxía. Inclúese aquí tamén a optimización aplicada á anestesia. Hai que ter en conta as limitacións tanto das salas de operacións como dos horarios reservados a cada tipo de cirurxía diariamente ([Abbas *et al.*, 2026](#)).
- Servizos con hospitalización. Aquí debemos obter una maximización do número de pacientes atendidos coas camas e profesionais dispoñibles, sen intervir no tempo de atención. Sería un caso de xestión de camas de UCI, servizo en planta ante nacementos, e parte da literatura baséase en situación de crise, como na pandemia do COVID ([Silveira *et al.*, 2024](#)).
- Servizos de atención domiciliaria. Un posible contexto neste campo son problemas de rutas centrados en atender unha serie de pacientes que, mentres que están na súa casa, necesitan atención diaria, semanal,... para certas labores ou certos tratamentos na súa casa. Presentamos como exemplo [Vieira *et al.* \(2022\)](#).
- Servizos de atención continuada (residencial). Moi parecidos aos servizos de hospitalización, pero de maior duración. É aplicable a centros xeriátricos, por exemplo en [Kim e Kim \(2010\)](#).

Aproveitando esta primeira clasificación, na que xa establecemos que o noso interese vai ir dirixido aos problemas incluídos nos servizos ambulatorios de coidado, vamos establecer de forma paralela un segundo elemento de distinción que interseca con todos os subgrupos establecidos e que responde á seguinte pregunta: que se pretende solucionar, é dicir, que implicacións vai ter na realidade o feito de resolver o problema? Este formato ten maior aceptación na bibliografía, atopándose referencias en [Ahmadi-Javid *et al.* \(2017\)](#); [Hulshof *et al.* \(2012\)](#) ou [Marynissen e Demeulemeester \(2019\)](#):

- Problemas de decisións estratéxicas ou decisións de deseño. Son aquelas máis orientadas a longo prazo, de natureza estrutural á hora de outorgar o servizo ou que establezan a dimensión poboacional que debería alcanzar. Entre elas, estarían problemas de localización dun hospital/servizo, a disposición do distintos servizos na construción dun hospital, se aceptas ou non urxencias nun centro, recursos iniciais que outorgar á construción dun hospital...
- Problemas de decisións tácticas. Considerase un punto medio no que as decisións non teñen un impacto a longo prazo, pero que deben ser algo áxiles á hora de ser resoltas, sen ser de urxencia máxima. Para facer una analogía, un protocolo á hora de atender un paciente en radioterapia non é algo tan firme como a cantidade de aceleradores lineais que teña na súa construción, pero tampouco é tan urxente como decidir que paciente será atendido en que día.

Como exemplos, temos problemas como o de “rutas de paciente” (determinar en que orde se realizan as debidas probas e consultas ante pacientes con certos diagnósticos ou síntomas), os tempos entre citas e consultas, cal é a capacidade reservada para cada tipo de paciente nun determinado servizo, tamaño óptimo de poboación á que facer un cribado,...

- Problemas de decisións operacionais. Estes últimos xa son para decisións a curto prazo e da organización diaria, semanal, mensual,... de calquera actividade sanitaria. Pode ir dende o noso problema orixinal de planificación de citas dentro dun servizo de radioterapia cada semana, ou á organización de citas previas para pacientes que requiran consultas en función da súa gravidade.

Dentro desta categoría pódense facer dúas subclases: os casos nos que a solución xera un *plan* ideal a seguir, supoñendo saber os recursos e eventos que van acontecer no período de actuación (o que sería decisións operacionais *offline*); e os que aportan mecanismos ante imprevistos que xurdan na coidado dos pacientes (decisións operacionais *online*). Os exemplos anteriores pertencerían máis á primeira categoría, pero optimización da aceptación de pacientes ambulatorios cunha reserva para persoas que requiran atención inmediata, sería máis *online*.

Por último, en [Marynissen e Demeulemeester \(2019\)](#) emprégase un termo xa máis orientado a transmitir tanto información sobre a realidade do problema no servizo de saúde como a súa modelización e execución dentro da optimización matemática: os problemas de planificación de citas múltiples en hospitais (ou MASP, en inglés).

Etiquétanse nesta categoría a aquelas cuestións nas que unha serie de pacientes deben ser asignados a múltiples citas (facendo así unha planificación ou *schedule*) utilizando un conxunto de recursos médicos e integrando no problema a todos os axentes que interveñan na situación a resolver. Pódese interpretar como un concepto que aglutina tanto os chamados problemas combinatorios en hospital, nos que se busca planificar nun día varias citas a unha serie de pacientes; como os problemas de planificación en serie, nos que as citas consecutivas teñen que cumprir condicións como a existencia de planificacións previas, tempos entre citas,... cuxa descrición aparece en [Hulshof *et al.* \(2012\)](#).

Os MASP están inseridos completamente nos problemas de decisións operacionais, como se indica en [Marynissen e Demeulemeester \(2019\)](#), xa que se encargan de satisfacer as necesidades dos pacientes a nivel organizativo (tendo en conta as restricións dos pacientes), sen entrar en decidir cuestións a medio ou longo prazo. Como podemos supoñer, o noso problema inicial axústase aos MASP adecuadamente.

Veremos posteriormente na [Sección 1.2](#) que en xeral na literatura insírese o noso dilema radioterapéutico mais ben dentro dos problemas en serie. Hai que ter conta que será un particularización neste grupo, pois aplicaranse sobre este problema base condicións de continuidade das sesións segundo

os días en función do tratamento (todos os días de semana, descansos entre semana para recuperarse da radiación,...), entre outros engadidos necesarios.

Malia isto, de cara á categorización intermedia a nivel médico-matemática do problema inicial, vamos a quedarnos nos MASP, xa que si que hai casos nos que se poden aplicar dúas doses de radiación no mesmo día. Esta característica é propia dun problema combinatorio de hospital, mais que dun problema en serie.

Porén, hai moitos máis tipos de problemas que se poden incluír nos MASP de acordo a [Marynissen e Demeulemeester \(2019\)](#): xestión de citas en centros de rehabilitación, en servizos que necesitan operar citas para diagnosticar enfermidades, en servizos oncolóxicos (incluída a radioterapia, pero tamén quimioterapia), organización dunha sala de operacións.... pero tamén se poden propoñer incluso modelos a nivel xeral sen especificar en que parte dun hospital ou centro médico se puideran empregar. Isto xa implica unha certa desafección da realidade médica fronte ao modelado matemático como tal do problema á hora de definir os MASP.

Toda esta información a intentamos resumir de maneira gráfica na [Figura 1.1](#), aplicada ao noso dilema orixinal. Completada xa esta primeira parte na que encapsulamos onde se atopa o noso problema

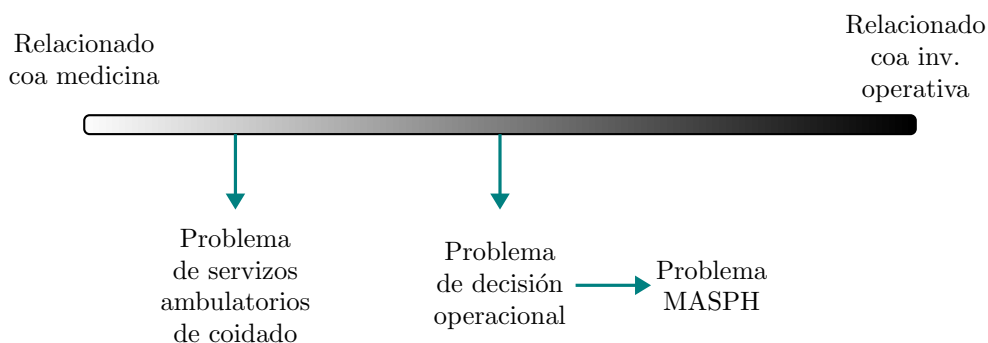


Figura 1.1: Gráfica informativa de como se encadra, dentro das clasificacións vistas en problemas de optimización na medicina, o problema de citas de radioterapia, segundo se son acordes máis á disciplina médica (esquerda) ou matemática (dereita).

dentro da optimización en citas aplicada á medicina, vamos agora a proceder a facer unha revisión histórica da liña de investigación que levou a pensar neste dilema.

1.2. Historia da optimización en citas médicas

Como antecedente primeiro de todo o campo de optimización aplicada á planificación de pacientes médicos temos o artigo de [Bailey \(1952\)](#). Primeiro, faise un estudo do tempo de agarda en servizos ambulatorios; logo, propónse e estímase os parámetros dun modelo do tempo que tardan en ser consultados os pacientes. Finalmente, expónse unha solución fixando o tempo de consulta medio do momento tendo en conta que se busca tamén simultaneamente que os médicos perdan tamén o menor tempo posible.

A partir de aquí, nos focalizaremos nas chamadas *outpatient clinics*, que inclúe aos servizos ambulatorios, pero tamén aos servizos cirúrxicos, pois é un servizo no que non temos que regular como tal a estancia dos pacientes. Se debemos considerar como factor restritivo a permanencia dos pacientes en observación ou hospitalización, denominamos este tipo de clínicas como *inpatient*. Estes conceptos son necesarios introducilos ao ser tan empregados na bibliografía: [Ahmadi-Javid et al. \(2017\)](#); [Cayirli e Veral \(2003\)](#);...

Todos estes problemas, en resumo, entran dentro da teoría de **problemas de colas**, xa que só nos centramos fundamentalmente nunha serie de pacientes que debemos atender a medida que van chegando, e que logo podemos esquecernos deles ([Cayirli e Veral, 2003](#)).

Aínda así, o que vemos neste contexto é que case toda a investigación céntrase exclusivamente na modelización de pacientes **nun só día** co fin de reducir o seu tempo de espera con respecto a unha **única consulta**. Unha primeira distinción xa recae se permitimos unha planificación dinámica ou estática. A primeira opción visualízase en [Fries e Marathe \(1981\)](#), na que se prevé con cadeas de Markov o posible custo en espera dos pacientes e dos facultativos a planificación en caso de incidencias (non asistencia de pacientes, por exemplo). As planificacións estáticas son as que se orixinan ao inicio do día/semana,... e continúa independentemente de imprevistos; considérase como a metodoloxía por defecto.

Neste primeiro momento estúdanse dilemas como o impacto no tempo de espera que os pacientes poidan ser atendidos por varios médicos posibles ou un médico habitual sempre ([Liu e Liu \(1998\)](#)) fronte a [Cox et al. \(1985\)](#)); modelar situacións como a falla de puntualidade dos pacientes ([Mercer, 1960](#)); comparar como afecta ao horario predeterminado a non asistencia de pacientes respecto doutros factores ([Ho e Lau, 1992](#)),...

Aínda que sexan asuntos de importancia, están enfocados a **simulacións de pacientes e a súa modelización** empregando modelos estocásticos (modelos de Poisson) ou de supervivencia (Weibull). Pretenden analizar, en xeral, *decisións tácticas*, non a proposición dun problema de optimización ao uso con certas restricións, o que sería unha *decisión operativa*.

Como se nos indica en [Froehle et al. \(2013\)](#), isto deixa de lado problemáticas como o feito de que pacientes que non sexan tratados (cunha cita ou múltiples) deberán ser atendidos días posteriores; polo que xa comezará a ter sentido traballar con **conxuntos de pacientes a asignar**. Esta idea é máis similar coa nosa visión proposta para o problema inicial. Seguindo esta referencia como modelo avanzaremos xa na **planificación de máis dun día**.

Os primeiros avances neste eido seguen estando orientados a pacientes de consultas médicas, sen restricións como tal para reincidencia nas citas, como en [Murray e Tantau \(1999\)](#) na que se propón o sistema de citas “*Advanced access*” ou acceso aberto. En resumidas contas, consiste na implantación de citas só durante os un/dous/tres primeiros días dende que se recibe unha chamada, e, en caso de estar completas, rexeitar a cita sen que haia unhas esperas moi elevadas.

Este efecto estúdase mediante simulación en [Kopach et al. \(2007\)](#), no que se obtén de conclusión que pode afectar tamén á continuidade da atención médica, ao non asegurar a posibilidade de realizar citas de seguimento, pero tamén que si presenta un efecto significativo a favor da cantidade de pacientes atendidos, pois reduciría os pacientes que non asistirían ás citas.

Antes de continuar, vamos a introducir brevemente o seguinte concepto:

- Un **proceso de decisión de Markov** consiste nun proceso estocástico en tempo discreto que supón unha extensión das cadeas de Markov. Temos un conxunto de estados S onde, en cada $s \in S$, teremos que tomar unha das decisións posibles no espazo A , que definirán a función de transición de estado de s a calquera outro estado $s' \in S$: $P_{a \in A}(s, s')$. Engadimos que a propiedade de Markov de carencia de memoria dende un estado presente s **modifícase requirindo tamén fixar unha acción a** .

Volvendo ao acceso aberto, aparecen artigos que se encargan de comparar esta metodoloxía fronte a modelos de programación lineal con estimacións para a probabilidade de que un paciente non asista á súa cita ([Robinson e Chen, 2009](#)). Tamén temos comparacións fronte a procesos de decisións de Markov, destacando [Liu et al. \(2010\)](#), onde ademais se establece que o modelo proposto por eles é preferente canto maior sexa a carga de pacientes fronte ao modelo de acceso aberto.

Neste último caso, conclúese ademais que debido ao gran tamaño do modelo, é necesario recorrer a métodos heurísticos, e este vai ser un gran problema recorrente ao afrontar citas en múltiples días. O proceso de decisións de Markov vai ter unha cantidade de estados absurdamente elevada, e, no caso de restricións de problemas de programación lineal, estaremos ante unha situación similar (maldición da dimensionalidade).

E agora si, con este contexto previo, podemos chegar á temática radioterapéutica, que ten sentido histórico ante o incremento dos casos detectados de tumores malignos, provocando un novo foco de saturación no sistema sanitario. Resumimos brevemente todo o visto ata agora na [Figura 1.2](#), pero

recordando que toda esta historia non debe ser lida como un transcurso lineal sen bifurcacións ou avances paralelos.

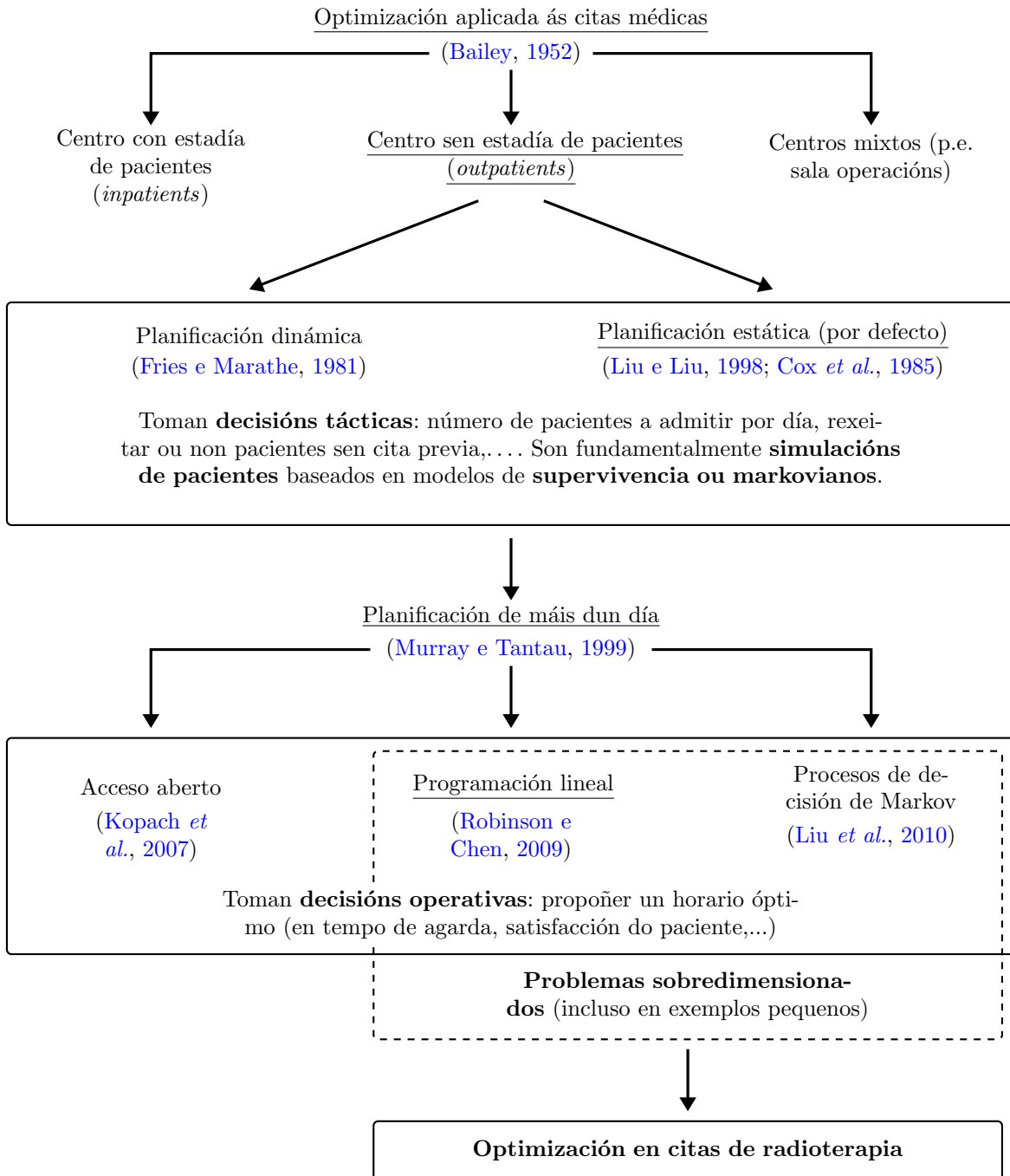


Figura 1.2: Gráfica informativa que resume dende a optimización aplicada a citacións médicas ata a súa aplicación concreta á radioterapia. Os avances predecesores directos do noso problema orixinal están subliñados.

1.3. Estado do arte da planificación en radioterapia

Como xa adiantabamos na sección anterior, coa base que temos feita ata agora, xa podemos falar desta vertente de problemas coa vista posta na radioterapia. Debemos salientar de novo, que os dilemas aos que nos enfrontamos neste contexto serán moi similares aos de cirurxías (en particular, cando involucran tumores malignos) e con terapias de quimioterapia. Neste apartado, vamos a tomar como referencia principal a Vieira2020 (que é unha tesis doctoral que expande os contidos en [Vieira et al. \(2016\)](#)), no que só se revisan artigos despois do ano 2000. Neste documento faise unha clasificación inicial das múltiples metodoloxías que se poden encadrar na investigación de operacións aplicadas ás terapias radioterapéuticas:

- Simulación computacional e teoría de colas. Estas funcións xa as vimos anteriormente nun contexto xeral. Por simulación, nos referimos a imitar o comportamento que terían os pacientes de forma ideal e reproducilo múltiples veces ata ter un reflexo razoablemente similar á realidade.

Na teoría de colas non se estuda o comportamento dos pacientes na súa totalidade, focalízase en modelar o seu horario de chegada e o tempo de agarda. O obxectivo en ambos será ver os posibles colos de botella para así identificar os puntos de inflexión no fluxo de pacientes (e ver así onde poder mellorar de forma efectiva o servizo).

- Métodos heurísticos. Vamos a incluír aquí tanto os métodos heurísticos construtivos como as metaheurísticas. Antes de nada, debemos salientar que hai certa discusión con respecto á definición de ambos termos, por iso, indicamos que a fonte das descrições a continuación é [Vieira \(2020\)](#).

Os métodos heurísticos consisten en solucionar de forma aproximada o problema con algoritmos creados ad-hoc para o problema. No caso do tratamento de radioterapia, referiríanos a dar un horario aos pacientes que non é necesariamente óptimo, pero que si aseguramos que é “moi bo” en canto aos obxectivos que queiramos establecer (normalmente, tempo de agarda).

As metaheurísticas cumprirían a mesma idea, dar un horario razoablemente “bo” en función das prioridades que propoñamos. A diferenza radica en que os algoritmos propostos non son específicos do problema proposto, son xenéricos e están referendados como métodos efectivos para problemas con certa estrutura matemática. A virtude en xeral desta maneira de proceder é que a falla de precisión na solución supón unha maior eficiencia temporal fronte a unha solución exacta.

- Procesos de decisión de Markov. Partindo da [definición previa](#), no caso de planificar citas de radioterapia, consistiría, a termos xerais de propoñer un estado inicial de cara as citas necesarias a planificar, citas planificadas... e propoñer unha serie de accións (engadir máis pacientes a certa día, pospoñer citas,...) que, con certa probabilidade (onde cabe incerteza) pode xerar un novo estado no seguinte horizonte de planificación.
- Programación matemática. A gran maioría dos problemas de citas que veremos nesta categoría van ser lineais (usaremos a nomenclatura LP de *Linear Programming*), o que quere dicir que buscarase maximizar ou minimizar unha función lineal en función dun vector x con restricións que representamos con inecuacións lineais.

Tamén é frecuente traballar con variables enteiras (IP, *Integer Programming*). Cando coinciden ambas casuísticas temos programación lineal e enteira (ILP). Outra forma común de afrontar os problemas asociados ás citas en radioterapia será a programación lineal que mestura variables continuas e variables enteiras (MILP, *Mixed Integer Linear Programming*).

Ademais disto, tamén debemos falar da finalidade que teñen os diversos problemas: toma de [decisións estratéxicas e tácticas](#), organización de recursos limitados, redución do impacto temporal con priorización de pacientes e **planificación de tratamento de radioterapia**.

Só nos vamos a centrar neste último campo, no que tomaremos decisións a nivel [operativo offline e online](#), en consonancia cos obxectivos que se pretendan lograr.

En orde de comezar una liña cronolóxica, primeiro debemos concretar cales serían as primeiras instancias nas que se menciona como tal o problema de asignación de citas de radioterapia (RTSP, *radiotherapy scheduling problem*). Estes artigos van ser [Kapamara et al. \(2006\)](#) e [Petrovic et al. \(2006\)](#).

O primeiro artigo é moi importante pois nel establécese un paralelismo entre o RTSP e o “problema de planificación de tarefas e tendas” (JSSP, *job-shop scheduling problem*). O JSSP ([Pinedo, 2022](#)) trátase dunha problema de optimización moi recorrente que consiste nun conxunto de máquinas $M = \{M_1, M_2, \dots, M_m\}$ e de traballos $J = \{J_1, J_2, \dots, J_n\}$ finitos no que cada traballo debe seguir unha ruta específica entre as máquinas e permite a repetición de visitar múltiples veces a mesma máquina (repetindo a mesma operación). O obxectivo deste problema é **minimizar o tempo de remate de todos os traballos**.

Debemos salientar que este modelo ten moito sentido dentro da súa percepción do RTSP porque **inclúe as fases previas ao propio tratamento de radioterapia**. Estas etapas son a consulta inicial, a fase de preparación (escáner de visualización do tumor, simuladores en caso de dispoñelos, habitacións de moldes para bloquear o movemento do paciente no tratamento), o pre-tratamento (consulta na que un radiólogo decide que tipo de radiación debería recibir, tipo de aceleradores lineais aos que debería ir, calibración,...) e o tratamento en si (pasar polo acelerador lineal onde sofre radiación, usualmente varias veces).

Con este modelo, temos que cada paciente o podemos considerar un “traballo”, que terán rutas distintas en función da **priorización de cada paciente** (o cal xa é un engadido interesante con respecto ás citas médicas xerais que tiñamos anteriormente). Tamén xa se establece a posibilidade de “recirculación” na parte de tratamento, é dicir, permítese **repetir a terapia nos aceleradores lineais/linacs** cando sexan adecuados ao seu tratamento.

Tamén se propoñen múltiples funcións obxectivo (sen fixar ningún modelo como tal de programación matemática) distintas do JSSP base, que consistiría en minimizar o tempo que lle leva a todos os pacientes superar o tratamento. Isto é, minimizar número de pacientes que non chegan a comezar o tratamento antes de 31 días despois do seu fin de etapas previas. Outro dos obxectivos mostrados é minimizar a media de días entre completar tratamento e o primeiro día que se puidera iniciar.

Como último apuntamento, como está baseado nun JSSP, será un problema NP-duro (de feito o problema do viaxeiro é unha particularización do JSSP ([Pinedo, 2022](#))), o que implica que non se atopou a forma de resolvelo de forma exacta con tempo polinomial en función dos pacientes/linacs. Propón para resolver o problema métodos exactos (*branch and bound*) e metaheurísticas que se aplican aos JSSP: búsqueda tabú (que conclúe que sería o mellor en tempo de execución en xeral por experimentos en JSSP), GRASP, algoritmos xenéticos,...

Pasamos agora a falar de [Petrovic et al. \(2006\)](#), onde tamén se establece a comparación do problema de tarefas e tendas (JSSP) co RTSP: as máquinas, operacións e traballos do primeiro tipo de modelo pasan a ser aceleradores lineais, as sesións de radioterapia e os pacientes, respectivamente.

Ademais disto, tamén se aclara un detalle extra: que os tipos de aceleradores lineais (só presenta dous: de baixa intensidade e alta intensidade) pódense considerar **máquinas paralelas**. Isto nos permite transformar o RTSP ao **JSSP flexible** ([Pinedo, 2022](#)), no que hai dentro do fluxo de cada paciente conxuntos de máquinas que poden facer a mesma operación e, polo tanto, ser intercambiabes á hora de completar unha etapa do traballo.

Na proposta das variables do RTSP, moi similares a [Kapamara et al. \(2006\)](#), engade tamén a separación de pacientes en prioridades, un vector cos días que un paciente P_n ocupa unS *linacs* de potencia m_n dependente do paciente, e unha lista de restricións máis extensa que no artigo anterior. Entre elas, dúas que chaman a atención son que os pacientes de menor prioridade non deben ser atendidos en fin de semana, e que se permite unha interrupción do tratamento cando este supera 5 días e en función da categoría do paciente.

Para explicar os dous algoritmos propostos, debemos introducir o concepto de **release day** (día de lanzamento, o primeiro día que é posible que un paciente inicie tratamento) e o de **due day** (día de vencemento, o último día recomendable no que comezar tratamento, tratamentos que comecen despois dese día considéranse tardíos).

Aproveitando isto, tamén vamos a introducir o concepto de *slot*, que vén sendo a subdivisión de

espazo temporal para organizar as citas. Ata agora, asociamos unha cita a un *slot*, pero pasarase a considerar, posteriormente, que unha sesión pode durar varios *slots*. Isto nos permite traballar con divisións de tempo enteiras nun mesmo día e mantendo a estrutura de problema combinatorio.

Os dous algoritmos presentados en Petrovic *et al.* (2006) son aparentemente opostos conceptualmente. O primeiro, ASAP (*as soon as possible* ou “tan pronto como sexa posible”), procura dar a primeira cita do paciente no día de lanzamento. O segundo, JIT (*just in time* ou “xusto a tempo”), procura outorgar a primeira cita do tratamento no día anterior ao día de vencemento. Ambos algoritmos buscan respectar todas as condicións que se establecen na lista de restricións, pero seguimos sen modelo de programación a efectos prácticos.

Conclúe este traballo cunha **cunha comparación a partir de datos obtidos** polo servizo de saúde do Reino Unido (NHS), sen chegar a aplicar ningún algoritmo na realidade, e créase un índice de satisfacción específico para cada tipo de paciente. Despois de comparar tres instancias con menor a maior capacidade de saturación dos *linacs* con tratamentos xa planificados, conclúndo que se obteñen mellores resultados co algoritmo JIT (pero que sería interesante propoñer unha mestura de ambos algoritmos en función da urxencia do paciente).

Destaca varios avances que se solucionarán en artigos posteriores. Un deles é a adhesión de pacientes que teñen a radioterapia como terapia adxuvante (complementaria a outras como quimioterapia). Outro é que a planificación poida redistribuír espazos libres nos aceleradores lineais cando haia pacientes que renuncien ao tratamento (ter un obxectivo **operativo online**) e que se **teñan en conta as preferencias dos pacientes**.

Partindo destas primeiras instancias de localización, establecemento e enfoque de prioridades do problema, chegamos a unha división metodolóxica de como afrontar o problema. Entre as múltiples opcións está a de, por fin, **crear un modelo de programación matemática** ao uso, pero tamén hai moitas outras que vamos a describir brevemente:

- Metaheurísticas e algoritmos heurísticos propios para o RTSP. Aquí continuamos coa liña dos artigos de Petrovic *et al.* (2006). Un deles, que inclúe á mesma autora Petrovic e Leite-Rocha (2008), compara unha serie de algoritmos construídos a partir de varias regras de asignación de citas coa metaheurística GRASP (*greedy randomized adaptive search procedure*). GRASP é un procedemento heurístico semi-voraz aplicable a problemas de optimización combinatoria, que buscan a mellor combinación dun conxunto finito de posibilidades.

Dentro deste paradigma, avánzase moito na planificación non só do tratamento como tal de radioterapia, se non de todo o proceso pre-tratamento que mencionamos anteriormente: consulta inicial, fase de preparación con escáner do tumor, molde para inmovilización do paciente, planificación do número de sesións, calibración do acelerador lineal por parte do radiólogo,...

Os artigos que progresan nesta ruta son Kapamara e Petrovic (2009) e Petrovic *et al.* (2009). No primeiro utilízase unha heurística para cada una das partes de todo o proceso (consideran catro etapas: planificación do tratamento, consulta de radiólogos, pre-tratamento e tratamento) sendo a súa función obxectivo unha **ponderación en función da urxencia do tempo dos pacientes ata o seu primeiro día de tratamento**. Engade unha heurística voraz (*steepest hill climbing*) para rastrear posibles melloras para cada paciente (en función da súa urxencia) intentado facer que comece cada etapa un día antes.

No segundo artigo (Petrovic *et al.*, 2009), preséntanse algoritmos xenéticos para xestionar o mesmo proceso que en Kapamara e Petrovic (2009). Subliñamos que se ten en conta na función obxectivo tanto o tempo de agarda promedio como unha función que indica os días de **retraso que houbo respecto da data de vencemento de inicio de tratamento** (se comeza antes desta data límite, entón este valor é 0). O resultado é favorable para a maioría dos pacientes fronte aos datos reais que dispoñían, pero non é así para os casos máis urxentes.

Este traballo expándese en Petrovic *et al.* (2011), onde se compara o efecto que pode ter no algoritmo xenético o feito de considerar a todos os pacientes equivalentes, fronte a priorizalos na selección inicial de citas antes de facer o algoritmo xenético e durante os procesos de internos do

mesmo (modelo KB-GA). Por último tamén se compara estes dous modelos co modelo ponderado en función da categoría de urxencia (similar a [Petrovic et al. \(2009\)](#)).

- Procesos de decisión de Markov. Este proceso xa foi iniciado nas citas médicas xerais, pero estendeuse aos tratamentos oncolóxicos da man, sobre todo, de [Sauré et al. \(2012\)](#). Busca establecer un proceso de Markov de **horizonte infinito** (é dicir, as accións e estados posibles son, en teoría, infinitos) **descontado** (é dicir, que as recompensas de cada acción por pasar dun estado a outro sexan cada vez menores co paso do tempo).

O espazo de estados fíxao como un vector que contén o número de citas xa outorgadas en cada día, **o tempo empregado en citas fóra de horario** e as citas de cada tipo en espera para ser asignadas. O espazo de accións indicaría cantos pacientes vamos a asignar aos días seguintes e de que tipo e o tempo extra empregado en citas que teremos que cumprir nos vindeiros días. Estas accións tómanse ao final de cada día.

Os custos van asociados tanto aos tempos de agarda dos pacientes, os custos do tempo extra de atención e a posposición de pacientes que aínda quedan por asignar a outro día. A ecuación que necesitamos optimizar para cada posible estado s respecto do conxunto de posibles accións dende s A_s contempla dúas cousas. A primeira é o custo que comentamos respecto do estado s e calquera posible acción $a \in A_s$. A outra é unha serie dependente de todos os estados de transición dende posibles estados s' que poden chegar cunha acción a s , ponderado por un factor de desconto (que fará que esta parte vaia reducindo a súa importancia a maior distancia temporal dende a súa planificación).

A partir dos datos reais que dispoñen, cómpre reducir o espazo infinito de días posibles a citar a un espazo finito para que non afecte á planificación de pacientes. Se non, terían un conxunto de espazos inmenso e accións. Para isto, transfórmase o modelo de procesos por un modelo de programación lineal equivalente e logo aplicando un método de aproximación dinámica (*approximate dynamic programming*) permítelles chegar a un problema resoluble.

Este proceso continúa con [Gocgun e Puterman \(2014\)](#), que o aplica na quimioterapia, e engade distintos tipos de pacientes en función de urxencia, penalizacións en caso de cita anticipada e **niveis de tolerancia**. Compara este resultado con varias heurísticas, simulando a chegada de pacientes, e chegando á conclusión de que opera mellor en situacións de alta carga de capacidade (pois se os *linac* non se saturan, non se aproveita a capacidade de análise tan exhaustiva dos PDM).

- Optimización estocástica. Presentamos só un artigo con esta liña de investigación ([Legrain et al., 2015](#)). O obxectivo é presentar un modelo *online*, en concreto, que poida ter en conta a distribución de chegada de pacientes e eventos futuros (como a retirada dun paciente do tratamento). Primeiro, preséntase un modelo *offline* por programación lineal, para ter unha cota superior de optimalidade no caso do horario online (o custo de anticiparse a problemas repentinos que se dean co algoritmo online correspóndese cunha perda de optimalidade de horario).

A continuación, presenta un algoritmo voraz (empregado no lugar onde obteñen os datos), no que cada paciente será asignado ao día que menos custo provoque (que usualmente, será cando máis pronto sexa posible), cunha reserva de *slots* prefixada cada día para pacientes máis urxentes.

O terceiro modelo, estocástico, inclúe como función obxectivo o custo do plan de cada paciente j e a **esperanza, entre todos os posibles escenarios que se den, do custo e tempo de pacientes futuros ata o prazo máximo de fin de tratamento de j** . Para resolvelo é necesario empregar, entre outros técnicas, unha transformación primal-dual e o algoritmo de Benders (xeración de filas). De aquí obtense un día e *linac* a empregar por cada paciente.

Despois, combínase cun algoritmo de intención *online* (modelo clarividente) para dar aos pacientes urxentes as quendas máis recentes que estean dispoñibles. Os demais pacientes pasan ao modelo estocástico e, dentro da asignación de día e *linac*, asígnaselle aquel *slot* (neste caso, é equivalente á duración dunha sesión) que tivera máis días en desuso.

Conclúe que, aínda sendo un modelo que permita anticiparse á chegada de pacientes urxentes, non xera tempos de espera tan elevados como o modelo que estaba implantado (algoritmo voraz) para pacientes de urxencia alta e media, tanto nos datos simulados como cunha selección de datos reais. Nestes últimos, ademais, detéctase que se sobrepasa o tempo de agarda recomendado dos pacientes en menos ocasións co modelo clarividente, sen haber un aumento substancial de tempo extra nos aceleradores lineais.

Con isto, xa podemos presentar os documentos máis relevantes que estableceron un modelo de programación matemática para resolver o problema de asignación de citas de radioterapia. Os autores máis prolíficos neste aspecto van ser Domenico Conforti, Francesca Guerriero e Rosita Guido ([Conforti et al., 2008, 2010, 2011](#)), todos eles investigadores vinculados á Università della Calabria (Italia). Isto vai ser moi importante para nós, pois forman parte das referencias principais de cara ao resto do traballo.

En [Conforti et al. \(2008\)](#), establécese por primeira vez un **un modelo de programación matemática con restricións** ao uso, cunha función obxectivo ben definida, restricións expresadas como inecuacións (fronte a regras escritas nun algoritmo) e coa idea de obter un **horario óptimo concreto**, sen ser unha aproximación.

O RTSP proposto neste artigo céntrase exclusivamente na fase de tratamento radioterapéutico en pacientes oncolóxicos. Ademais, presenta unha estrutura de *slots*, ou subdivisións temporais de 15 minutos nas que cada sesión para cada paciente pode encadrarse. En xeral, cada sesión requirirá dunha subdivisión, menos a primeira, na que se necesitarán dúas para calibrar o acelerador lineal, facer o pre-tratamento, etc. Outro detalle é que neste primeiro documento suponse accesible só un *linac*, aínda que sexa posible estendelo a varios.

Un conxunto fundamental para entender o proceso levado a cabo é a lista de pacientes por citar (\mathcal{I}) fronte a lista de pacientes xa citados (\mathcal{L} , “lista de pacientes radioterapéuticos en tratamento” ou *Radiotherapy Treatment Patients List*, en [Conforti et al. \(2008\)](#)).

A primeira lista será actualizada a medida que van chegando os pacientes, e coa segunda, indican cunha matriz de dimensións (número de días a planificar) \times ($|\mathcal{L}|$) os *slots* que xa están asignados.

Consideran obrigatorio tratar a todos os pacientes citados **o número de veces que se lles fose recomendado (entre 1 e 6 días) de forma consecutiva, unha vez por día**, e procurase aumentar a cantidade de persoas novas atendidas á semana **tendo en conta a gravidade do paciente**. Esta gravidade está dividida en catro categorías, obtidas de [Lim et al. \(2005\)](#): pacientes radicais urxentes (intentar eliminar o tumor maligno e curar ao paciente), radicais non urxentes, paliativos urxentes (o obxectivo é evitar dor reducindo o tamaño do tumor), paliativos non urxentes.

Os dous modelos de programación presentados son ILPs, sendo un modelo básico e logo a ampliación do primeiro. O básico procura facer unha planificación *offline*, é dicir, non propón modificar pacientes que xa foran citados con anterioridade, semana a semana. As restricións que presenta aseguran tanto a citación dos pacientes en días consecutivos respectando a súa asignación outorgada (aplicando cortes para reducir o tamaño do problema), respectar a capacidade do *linac* cos pacientes xa en radioterapia e a duplicación de duración da sesión cando o paciente recibe a primeira cita.

A súa función obxectivo consiste de dous termos: no primeiro súmanse todos os novos pacientes engadidos á planificación (ponderados en función da súa prioridade) e no segundo termo búscase diferenciar os itinerarios cun mesmo número de pacientes reservados coa mesma prioridade mediante unha mentalidade FIFO (*first order, first server*; entre dúas persoas coa mesma prioridade de atención, aquela que estea antes na lista, servirá de desempate para o horario óptimo).

O segundo modelo ILP aumenta de tamaño ao ter os *slots* ocupados por pacientes xa citados como variables (no anterior, as súas citas, son un parámetro para inicializar o programa), pero, cunhas restricións similares ás aplicadas no primeiro modelo, isto permitiría modificar o horario asignado dunha semana a outra a este tipo de pacientes, convertendo a planificación proposta de *offline* a *online* (podemos traballar con inconvenientes como a chegada dun paciente urxente novo modificando as citas doutros).

Mostra a diferenza de ambos modelos con exemplos concretos simulados, salientando a efectividade do modelo *online* e despois centrándose nun caso concreto de organización feita nun hospital onde hai

un número fixo de pacientes a atender: o modelo estendido é capaz de xerar máis pacientes atendidos por día. En [Conforti et al. \(2011\)](#) esténdense estes modelos para horizontes de predición arbitrarios e incluíndo condicións de dispoñibilidade do paciente, de menor cantidade de citas na primeira semana de tratamento,...

Este primeiro caso serve de antecedente para afondar na planificación *offline* con programación matemática, pois temos xa un patrón para impoñer condicións máis estritas necesarias para un tratamento adecuado. Este é o caso de [Conforti et al. \(2010\)](#), no que o que máis salientamos é o cambio de paradigma: ata agora as sesións correspondíanse case na súa totalidade cos *slots* nos que subdividía o día. Esta metodoloxía denomínase “*block scheduling strategy*”. Este artigo desvincula esta relación entre bloques e subdivisións temporais: agora **cada paciente pode ter unha duración de sesión específica** (“*non-block scheduling strategy*”).

Como cambios respecto de [Conforti et al. \(2008\)](#), en concreto, sobre o programa *offline*, **xa se parte da existencia dun conxunto de linacs, como mínimo 1, que fan a mesma función**, pero cando se escolla unha máquina para un paciente, esta debe manterse durante todo o tratamento. Isto faise para evitar ter que actualizar a calibración nunha nova máquina na que se fixo a primeira sesión (porque considérase que implica un aumento de tempo de sesión, e non sería fiable).

Outra nova é a de considerar que os pacientes xa citados poden ter descansos aproveitando o fin de semana de máis dun día (considérase traballar de luns a sábado cada semana), polo que se debería poder comezar o tratamento da seguinte deixando días para descansar da semana de terapia e, á vez, respectar o número de días de sesión consecutivas que debe facer a semana seguinte a organizar.

Ademais, divídese cada día en quendas, nas que cada paciente pode estar dispoñible para ser tratado ou non. Isto permite **respectar as preferencias dos pacientes en canto a restricións horarias por traballo, distancias ao hospital,...** e restrinxir só as sesións na quenda que poidan asistir. Isto é coherente co feito de evitar o absentismo no tratamento.

Compárase a súa efectividade con datos coherentes coa realidade, pero ficticios, como se comporta; pero salientamos deste documento dous aspectos que recalca: **non temos unha distribución temporal dos slots nos que citaríamos a cada paciente**, só o día; o modelo **crece de maneira exponencial a medida que aumentan o número de aceleradores lineais dispoñibles, os pacientes e os días no horizonte de predición**.

Máis traballo de interese neste campo vén da man de [Jacquemin et al. \(2010\)](#) e [Jacquemin et al. \(2011\)](#). En [Jacquemin et al. \(2010\)](#) comparase o modelo de [Conforti et al. \(2010\)](#) con outros tres algoritmos, cunha planificación máis a longo prazo. No primeiro modelo novo, permítese a posibilidade aos pacientes que estaban xa citados, ante posibles infactibilidades, de poder ser citados en tempo extra. No segundo, ademais disto, engádese un novo parámetro para permitir que non só os pacientes xa citados estean no seu horario preferente, se non tamén os que entren en lista a ser planificados. No terceiro, a última adición é establecer a preferencia dun paciente de ser atendido polo seu “radioterapeuta usual”.

En [Jacquemin et al. \(2011\)](#) afóndase en permitir o comezo dun tratamento a mediados de semana e que poida haber días de descanso en medio de semana en pacientes en tratamento; todo isto xunto cunha redución no número de restricións totais. Este modelo compárase co modelo final en [Jacquemin et al. \(2010\)](#), obtendo resultados similares en canto a tempos de espera e prometedores en canto a utilización dos aceleradores lineais.

Queremos salientar que, malia centrarnos fundamentalmente nos modelos de programación matemática, a partir de agora será frecuente mesturar esta técnica, que dá horarios óptimos, con técnicas heurísticas, aprendizaxe automático,... entre outras. Isto é debido á elevada dimensión que poden tomar o RTSP, que fai necesario asumir esta perda de optimalidade.

O primeiro exemplo que vemos desta categoría é [Castro e Petrovic \(2012\)](#), onde se crea un modelo de programación matemática para tratar toda a fase previa ao tratamento. Propóñense tres funcións obxectivo a minimizar en orde decrecente de interese. A primeira é o número de pacientes que comezan tratamento fóra da cantidade de días recomendados, xunto cun factor de peso en función da súa prioridade. A segunda busca minimizar o máximo de tempo de espera fóra do recomendable entre todos os pacientes a citar. A última considera a ponderación destes días de espera fóra do recomendado ponderados respecto da súa prioridade.

Para levar a cabo o proceso propón unha técnica de optimización multiobxectivo: aplicar o método lexicográfico repetindo o problema con cada unha das funcións obxectivo en orde decrecente de prioridade e limitando a factibilidade en cada paso de solucións á que sexa óptima nos pasos anteriores. O problema é que o tempo necesario para resolver o problema é inabarcable (supera en xeral para casos reais as 3 horas de execución), polo que se recurren a regras de asignación para poder obter unha solución inicial na primeira etapa.

Isto tamén se ve aplicado en artigos posteriores como [Fredes *et al.* \(2024\)](#), onde se crea un planificación de toda ruta de tratamento oncolóxico (dende a consulta inicial) a partir dun problema MILP engadindo unha compoñente de limitación de recursos médicos (non so persoal, se non material). Despois de propoñer este MILP, compárase a súa efectividade con algoritmos heurísticos de citas en función do mínimo e máximo tempo recomendado entre etapas.

Outro problema que xa se viu en [Conforti *et al.* \(2010\)](#) é que, de momento, non presentamos aínda ningún sistema para avanzar na xestión de citas, non só a nivel de capacidade diaria, se non de indicar en que horario concreto de cada día débese colocar cada unha das persoas citadas. O obxectivo sería **ter en conta a preferencia horaria; intentando que se desvíen os horarios de citación entre días consecutivos o mínimo posible**. En caso contrario, existe risco de incompatibilidades coas súas actividades diarias, implicando unha redución da satisfacción.

Este matiz se comeza a traballar en [Vieira *et al.* \(2020\)](#), onde se indica que é a primeira vez que se propón un modelo de programación matemática capaz de establecer un horario con este nivel de detalle, aínda que con restriccións moi fortes de horizonte de planificación e de número de aceleradores lineais.

Aínda que sexa o primeiro caso no que se aplica esta preferencia temporal dentro de cada día para beneficio dos pacientes, xa houbo traballos anteriores como [Vogl *et al.* \(2019\)](#) e [Maschler e Raidl \(2020\)](#) onde xa se busca esta sincronización dentro do propio día a nivel de minutos. Neles, como se emprega terapia de partículas, un radio emite ions ou protóns a varios pacientes simultaneamente, polo que é necesario organizar as citas de forma que coincidan entre os distintos pacientes.

No primeiro, aínda que se establece un problema de programación enteira mixta, non se chega a resolver debido á elevada dimensión que chega a alcanzar. A solución que se toma en ambos artigos é instaurar unha serie de metaheurísticas para aumentar o emprego do uso dos raios de partículas, pero probar outros métodos de tipo heurístico como algoritmos xenéticos, métodos de optimización local,...

Volvendo a [Vieira *et al.* \(2020\)](#), aquí propónse un modelo de programación lineal enteira mixta, no que se busca minimizar só **a desviación temporal entre cada cita e a preferencia temporal de atención de cada paciente** en terapia radioterápica cada semana. Indícase que este proceso non se pode propoñer para un número elevado de *linacs*, e, en caso de seren 5 ou máis, recomendan aplicar un método heurístico específico para pre-asignar pacientes a aceleradores lineais; despois desta separación, aplicar o MILP proposto para “clústeres” ou agrupacións de aceleradores lineais que sexan idénticos.

O algoritmo necesita precalcular as sesións totais que terían que ser asignadas a cada paciente pendente de tratar na semana a planificar e, tendo en conta o número de *slots* por sesión que precisa (unha sesión pode durar máis dun *slot*, estamos nunha estratexia “non-block”, como en [Conforti *et al.* \(2010\)](#)), pódese calcular o seu volume de carga.

Con isto, e calculando tamén a capacidade total semanal de cada *linac* e a capacidade ocupada en cada un por pacientes xa en proceso de tratamento, vai asignando cada novo paciente ao *linac* con menor capacidade saturada que sexa adecuado para el, ata quedar espazo dispoñible. Isto só permite asignar cada novo paciente baixo a asunción de que hai suficiente capacidade entre todos os *linac* para atender a todos o conxunto de pacientes.

A partir de datos simulados con datos reais do Instituto de Cancro de Países Baixos (NKI), procura-se aplicar ambos métodos relaxando a condición de correspondencia entre paciente e tipo de acelerador lineal ao que asistir. Estas instancias simuladas mostran que o MILP non chega a obter resultado cun tempo de execución **inferior a 8 horas** cando estamos traballando xa con 99 pacientes e 3 *linacs*. Isto é preocupante, e **é onde debemos intervir**, pois este centro xa presenta unha chegada de pacientes de 260 diarios en media e 8 aceleradores lineais.

Co procesado feito co algoritmo, redúcese enormemente o tempo de agarda, obtendo resultados ata con 260 pacientes simulados e 8 *linacs*. Os resultados obtidos son bastantes razoables, rondado en todas as instancias en que hai un 5%, aproximadamente de pacientes tratados fóra da ventá preferente de horario. Ademais, non se supera, en media, entre todos os pacientes fóra de ventá os 30 minutos de desviación.

En casos nos que se especifique a condición de correspondencia *linac* adecuado - paciente, os clústeres afectarán moito á solución do MILP, aínda con heurística auxiliar aplicada. Isto volve ser debido ao crecemento exponencial que sobre o problema de localización temporal de citas de radioterapia dentro de cada día en función dos aceleradores lineais, pacientes, número de *slots* e días de predición.

Este modelo xa é bastante satisfactorio á hora de crear un horario razoable, que era o noso obxectivo, pero imos a tomar como referencia a [Pham et al. \(2022\)](#), que supón unha evolución sobre este artigo. Neste documento, o proceso asignación de citas consiste en dúas partes. Primeiro, aplícase a todos os pacientes novos un problema de programación enteira lineal (ILP) para asignar o día e o acelerador lineal a cada paciente, en función da capacidade da que dispoñan. Despois, coa saída deste proceso, créase un modelo de programación lineal enteira mixta (MILP) para a asignación concreta de *slots* aos pacientes dentro de cada día.

Non avanzaremos máis sobre este modelo, pois será o que empregaremos no [Capítulo 2](#) como proposición para axudar aos xestores hospitalarios a creación de horarios satisfactorios tanto para pacientes como para os profesionais, ao ser posible xa a súa computación cun número aceptable de aceleradores lineais e pacientes.

Con isto chegamos ao noso punto de partida para comezar a traballar na asignación de citas de radioterapia en pacientes oncolóxicos, pero antes queremos revisar unha última liña de investigación que, malia non ser básica nin obxecto do traballo, queremos salienta: **planificacións *online* con capacidade para prever a chegada de pacientes**.

Un deles é [Pham et al. \(2023\)](#), onde se propón un modelo de programación lineal (LP), moi similar á primeira fase de [Pham et al. \(2022\)](#) para a citación de pacientes en días e *linacs*. O novidoso xorde de analizar como se comportaría este método cunha lista dinámica de pacientes que se actualiza non en cada horizonte de predición, se non xusto no momento que van chegando os pacientes. Ante esta situación, considérase un modelo de aprendizaxe automático que permita predicir cal é a data que debería asignarse a cada novo paciente xa no momento de admisión no hospital.

O outro artigo é o de [Aringhieri et al. \(2025\)](#). Primeiro, expón tres modelos de programación lineal enteira (ILPs) nos que os obxectivos son distintos (no primeiro minimiza o tempo de agarda dos pacientes; no segundo minimiza o tempo de non utilización dos aceleradores lineais; no terceiro minimiza o número de *slots* que quedarían “infactibles” para empregarse noutras citas). Logo, inclúe dúas adaptacións dos dous algoritmos de [Petrovic et al. \(2006\)](#) e tres algoritmos propios de asignación de pacientes con regras para evitar que non “encaixen” ben as sesións, evitando desaproveitamento de recursos.

Para aplicar o factor de “previsión” aos modelos exactos de programación lineal (que serían offline), traballase co concepto de “máscaras”, é dicir, créanse pacientes ficticios e planifícanse co modelo offline cun horizonte de predición alto. A medida que os pacientes reais chegan, iranse aloxando nestes espazos desocupados, que asegura que teñen aloxados os suficientes para que cada paciente poida realizar o seu tratamento..

Para rematar toda esta revisión integral do problema de planificación de citas de radioterapia (RTSP), presentamos a [Figura 1.3](#): unha gráfica informativa que pretende resumir as distintas técnicas de optimización que se empregaron para afrontar este dilema.

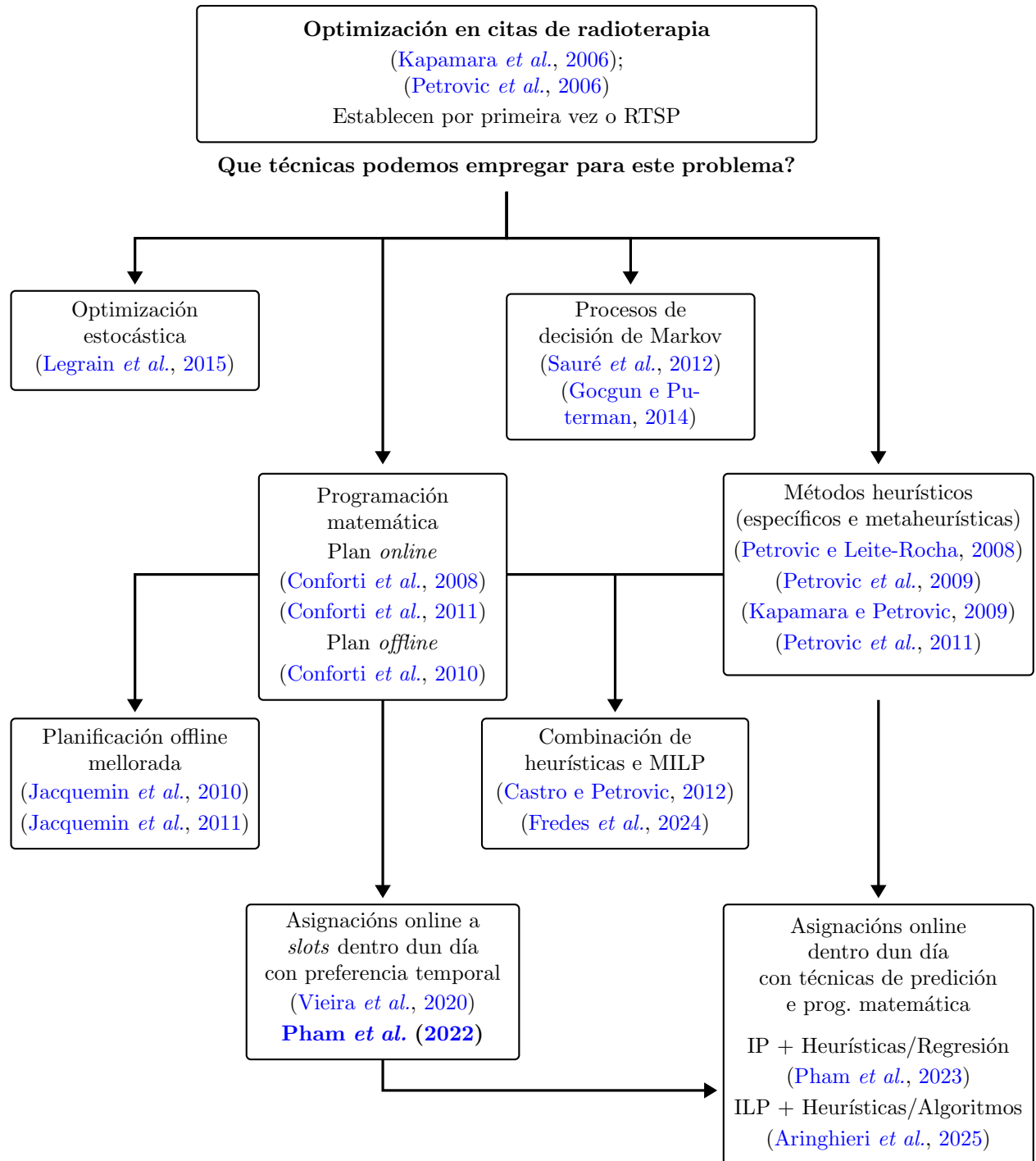


Figura 1.3: Gráfica informativa resumindo o estado do arte do RTSP. Salientamos en letra grosa o noso artigo de referencia de cara ao resto do traballo.

Capítulo 2

Modelo proposto para solucionar o RTSP

Como xa vimos no [Capítulo 1](#), estamos ante un problema combinatorio baseado noutro tipo de problema que é NP-duro, o que xa implica unha certa complicación á hora de resolvelo cando aumentamos moito a dimensión do problema. Ademais, vimos que hai moitas posibles configuracións dos pacientes: distintos tratamentos, prioridades, reservas de urxencia,... **Cómpre definir moi ben que queremos solucionar.**

Vamos a basearnos principalmente no artigo de [Pham *et al.* \(2022\)](#), que deixamos sen describir anteriormente. O contexto no que xorde este artigo é a raíz dos problemas de capacidade no CHUM (*Centre hospitalier de l'Université de Montréal*, Centro hospitalario da Universidade de Montreal, en Canadá). Para explicalos, antes de nada, indicamos cal é a ruta que toma un paciente cando é sospeitoso de padecer un tumor.

A ruta ata chegar á terapia radioterapéutica é moi similar ás que vimos no [Capítulo 1](#). O primeiro paso consiste na petición de consulta do potencial paciente, na que se describe tanto o estado do tumor como os posibles tratamentos e efectos secundarios.

Se o paciente opta pola radioterapia, procederase a “admitilo” (non debemos pensar necesariamente nunha hospitalización), prepararanlle o número e distribución das “fraccións”, e reservarán as citas para o tratamento. Estas “fraccións” consisten na división da cantidade total que debe recibir un paciente de radiación por sesión (usarémolo de aquí en adiante tamén como un sinónimo de sesión).

Antes de comezar a primeira fracción, hai que facer un recoñecemento do tumor (incluíndo posiblemente raios X, TACs, resonancias,...) e unha verificación por parte dun radiofísico, para comprobar a dose de radiación á que estaría sometido o paciente.

Con isto en mente, vamos a describir unhas características do CHUM. Os pacientes están divididos en catro categorías, en función da súa urxencia de atención:

- Pacientes P1: son pacientes paliativos (é dicir, búscase reducir unha dor intensa) e o día no que son admitidos (*admission day* ou día de admisión) e o día límite para ser tratados (*due day* ou día de vencemento) son consecutivos.
- Pacientes P2: son pacientes paliativos que se recomenda tratalos dentro dos 3 días seguintes ao día de admisión.
- Pacientes P3: son pacientes curativos (buscan reducir o tamaño do tumor, pero non están en situación de dor) e, dende que son admitidos, non deberían máis de 14 días ata a súa primeira sesión de radioterapia.
- Pacientes P4: pacientes curativos de menor urxencia. Considérase que dispoñen de 28 días dende admisión ata entrar fóra do período recomendado para iniciar tratamento.

Logo, debemos ter en conta que o CHUM dispón en total de 10 aceleradores lineais, dos cales tres están reservados para terapias específicas de radioterapia. Polo que o caso real contaría, a priori, con 7 *linacs* operativos e funcionais para calquera paciente usual (non consideramos a pacientes que requiran obrigatoriamente dos aceleradores lineais reservados).

Partindo desta situación, presentan en [Pham et al. \(2022\)](#) a [Táboa 2.1](#), na que se recollen a proporción de pacientes de cada categoría, o tempo medio de agarda dende a admisión de cada paciente e, o máis destacable, o porcentaxe (dentro de cada categoría) de dar un tratamento con atraso (é dicir, despois do día de vencemento dende a admisión) nos anos 2017-2018.

Categoría	Proporción sobre o total (%)	Días dende admisión ata vencemento	Porcentaxe de pacientes con atraso (%)	Tempo medio de tratamento dende admisión (días)
P1	0.44	1	14.29	1.09
P2	27.14	3	79.89	6.91
P3	41.36	14	74.55	18.11
P4	31.06	28	29.89	22.59

Táboa 2.1: Táboa de pacientes oncolóxicos atendidos no CHUM e seus tempos de atención

Como vemos, os pacientes de prioridade intermedia, P2 e P3 que, recordamos, polo menos os P2 foron categorizados como tratamento paliativos así que estarán en dor durante todo ese tempo, teñen un atraso moi importante á hora de ser tratados. De feito, os pacientes P2 teñen un tempo de agarda promedio que duplica o tempo recomendado de agarda recomendado de 3 días dende admisión.

Ante esta situación, que supón un perigo para a saúde dos pacientes, debemos escoller antes de nada como vamos a definila conceptualmente. Dende aquí, xa poderíamos poñer cales son os obxectivos que buscamos optimizar. Os conceptos cos que vamos a traballar, e a súa explicación correspondente son:

- Ao crear unha planificación das sesións, isto esixe establecer **como vamos a definir as nosas sesións**. Para isto, as sesións estarán definidas a partir **dunha estrutura de bloques**. En concreto, a súa subdivisión temporal sobre as que teremos que colocar as sesións será de 5 minutos.

Na vida real, as fraccións soen durar entre 10 e 165 minutos, polo que esta granulación é bastante adecuada, pero, evidentemente, a maior granulación, maior complexidade das restricións (ao haber máis bloques por día). En concreto, en todas as instancias que traballaremos propónse que os aceleradores lineais estean traballando 10 horas, polo que haberá 120 bloques por día (o mesmo para calquera acelerador lineal, pero deixamos a posibilidade de que sexa distinta, por se hai problemas técnicos ou restricións por mor do persoal dispoñible).

- Ademais, debemos traballar como xa dixemos coas diferentes categorías de pacientes. Cada paciente terá un plan de tratamento diferente no que se indican, á hora de admitilo a tratamento: o número de fraccións/sesións, o seu día de admisión, o día de lanzamento do tratamento a partir do cal pode ser tratado (*ready day*), o día de vencemento como límite de inicio recomendado de tratamento (*due day*) e a duración (en bloques de 5 minutos) de cada sesión.
- Por último, interézanos, se o paciente presenta unha hora de preferencia de atención, anotar entre que horas se atopa esta preferencia. **Só traballaremos cunha sección continua de horas preferentes**, é dicir, non se pode establecer unha preferencia de 10 da mañá a 12 da mañá e

despois de 4 da tarde a 6 da tarde. Isto bástanos ao ser as preferencias permitidas nos hospitais de turno de mañá ou tarde, en xeral.

- Todos os plans de atención esixen que cando un paciente inicie o tratamento, **débese facer unha sesión por día e de forma consecutiva**, deixando marxe se queremos aplicar descansos durante o fin de semana.

Para simplificar o modelo, non se consideran días sen actividade (festivos no medio da planificación) como unha característica adicional, pois sería cuestión de engadilos post-hoc de obter un calendario óptimo.

- En comparación doutros estudos que se revisaron no [Capítulo 1](#), vamos a permitir, nos casos onde hai múltiples aceleradores lineais, que se poida citar a un paciente nun *linac*, pero que este **non sexa fixo, pudiendo ser o próximo día citado noutro**.

O feito de ser permitido, na práctica, implicará un atraso temporal, xa que a xestión dos hospitais normalmente asigna a cada técnico en radioterapia a un acelerador lineal concreto, polo que se un cambio dun paciente de *linac* nesta configuración complicaría que os técnicos se axustaran á calibración necesaria para cada tratamento.

Ademais, se non se poden transferir facilmente as calibracións entre *linacs*, habería que axustalos de novo ante cada cambio. Ambas cousas conlevan unha perda de uso que pode ser un detrimento tanto para o número de pacientes a tratar como un atraso nas citas dos demais pacientes que veñan despois no día.

- En caso de emerxencias, **debemos ter a posibilidade de reservar parte da capacidade diaria conxunta para pacientes moi urxentes**.
- Debemos anotar tamén que hai pacientes que están xa en radioterapia no momento, e que haberá novos pacientes que hai que engadir ao sistema. **É obrigatorio respectar os días nos que cada paciente fixado é citado**, xa que non podemos interromper o tratamento (restrición), como xa indicamos anteriormente.

Malia isto, **si que se poden modificar as horas nas que os pacientes que xa estaban fixos son atendidos cada día**. Intentarase igualmente reducir esta casuística no posible, por mor do prexuízo para eles e mais para o hospital de notificar estes cambios.

Asemade, a nosas metas deberían ser:

- Unha prioridade evidente **é evitar que os pacientes comecen a radioterapia fóra dos seus días recomendados**. Esta sería a situación ideal, pero hai un problema: en xeral estamos nunha situación de imposibilidade de atender a todos pacientes dentro dos límites recomendados nos hospitais por falla de equipo e profesionais.

Propoñer isto como un límite férreo podería implicar, con horarios sen marxe a horas extra, obter problemas irresolubles. Polo tanto, en vez de ser unha restrición estrita, **cámbiase a unha restrición suave** que se poida sobrepasar, a cambio dunha penalización forte na función obxectivo.

- Os seguintes obxectivos en orde de prioridade son os que afectan á planificación temporal das sesións. Un deles é que, de ser posible, **hai que buscar a maior celeridade no tratamento**; non é suficiente que se comece dentro dos días recomendados.

Por exemplo, digamos que un paciente é admitido no día 1, pode comezar tratamento o día 3 e recoméndase que se inicie antes do día 6. Como buscamos rapidez no tratamento, se hai posibilidade de comezar o día 3 sen afectar a que outros pacientes sexan atendidos máis tarde, é mellor atendelo ese día que agardar máis adiante.

Ademais, outra restrición temporal de interese é o que xa comentamos nas características: **reducir o número de cambios de acelerador lineal durante a radioterapia**.

- Por último, vamos a salientar unha serie de obxectivos relacionados coa **preferencia horaria das citas**, obxectivo que xa fixo a súa aparición en [Vieira \(2020\)](#).

O máis prioritario sería, aínda que a demos como posibilidade, **evitar modificar horarios de sesións xa planificadas**, coa consumición de recursos administrativos e custos organizativos que implicaría tanto para o paciente como para o hospital.

Logo, debemos incluír o maior número posible de citas **dentro do intervalo preferente de atención** de cada novo paciente e, incluso dentro do seu horario, **que o rango de horarios nos que estea citado unha persoa sexa o mínimo posible** ao longo do tratamento.

Poñemos de novo un caso concreto: se podemos atender a un paciente que indicou a súa preferencia en quendas de mañá sempre ás 10 da mañá, isto será preferible a que algunha das citas sexan ás 9 fronte outras á 1 da tarde.

Por tanto, o que se suxire en [Pham et al. \(2022\)](#) é dividir o problema en dúas partes. Na primeira, procúrase a asignación de citas por día e priorízanse os obxectivos referentes a minimizar os atrasos, o tempo de agarda e os cambios de acelerador lineal durante o tratamento.

A segunda parte será a responsable de encadrar sobre o resultado da primeira as horas concretas de cada cita, e determinar que horas son as máis adecuadas tendo en conta como están distribuídos os pacientes xa fixos, preferencias diarias e rangos de horario nos que é atendido cada paciente pendente de ser aínda tratado.

Afianzamos esta división de prioridades en función de dúas etapas coa [Figura 2.1](#) e, en consecuencia, procedemos a explicar en detalle cada un dos subproblemas.

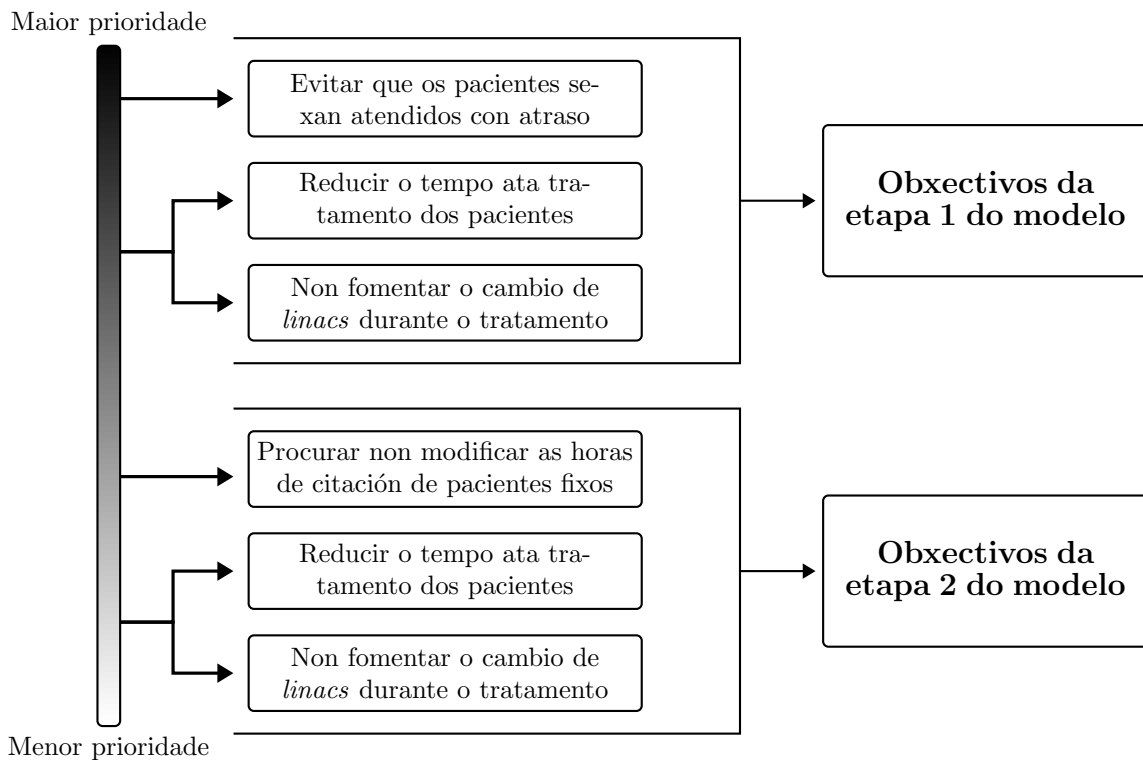


Figura 2.1: Distribución dos obxectivos en función da súa prioridade entre as etapas do modelo

2.1. Primeira parte: asignación de día e *linac* para o tratamento

Como xa dixemos, en cada momento hai un conxunto de pacientes **fixos** e un conxunto de pacientes **pendentes** ou novos, que aínda non comezaron a súa primeira sesión. Ademais, tamén debemos ter en conta que podemos dividir aos pacientes nas súas diversas prioridades.

Ademais, teremos que organizar unha serie de **días** no futuro, cada día co seu número de **bloques**, e dispoñemos dun conxunto de **aceleradores lineais ou linacs**.

Vamos a definir todos estes conxuntos, segundo as necesidades que precisemos para este problema na [Táboa 2.2](#). Debemos anotar antes de nada que todos estes conxuntos estarán compostos de índices numéricos que comezan en 0.

Conxunto	Definición do conxunto
\mathcal{P}	Conxunto de pacientes no sistema
$\hat{\mathcal{P}}$	Conxunto de pacientes que aínda non comezaron tratamento (novos)
$\bar{\mathcal{P}}$	Conxunto de pacientes que xa fixaron citas (fixos)
\mathcal{P}_i	Conxunto de pacientes con prioridade $i \in \{1, 2, 3, 4\}$
$\hat{\mathcal{P}}_i = \hat{\mathcal{P}} \cap \mathcal{P}_i$	Conxunto de pacientes novos con prioridade $i \in \{1, 2, 3, 4\}$
$\hat{\mathcal{P}}^c = \hat{\mathcal{P}}_3 \cup \hat{\mathcal{P}}_4$	Conxunto de pacientes novos curativos (non paliativos)
\mathcal{T}	Conxunto de días (unidade temporal) a planificar
\mathcal{L}	Conxunto de aceleradores lineais dispoñibles

Táboa 2.2: Conxuntos a empregar no modelo de asignación de días e *linacs*.

Ademais, por consecuencia lóxica, temos que: $\mathcal{P} = \hat{\mathcal{P}} \cup \bar{\mathcal{P}}$; $\hat{\mathcal{P}} \cap \bar{\mathcal{P}} = \emptyset$ e, da mesma forma $\hat{\mathcal{P}} = \hat{\mathcal{P}}_1 \cup \hat{\mathcal{P}}_2 \cup \hat{\mathcal{P}}_3 \cup \hat{\mathcal{P}}_4$; $\hat{\mathcal{P}}_i \cap \hat{\mathcal{P}}_j, i \neq j, i, j \in \{1, 2, 3, 4\}$. Non podemos ter pacientes que xa fixaron citas pero que acaban de entrar ao sistema, nin tampouco pacientes que presenten múltiples prioridades de atención simultaneamente.

A pesar disto, veremos cos datos que os pacientes fixos non comezan necesariamente o seu tratamento no día 0. Pode ser que lle engadamos un período de descanso dende o día 0 no que facemos a planificación, ou que esteamos aínda a agarda de que fagan a primeira sesión (que xa foi citada, pero non executada).

Debemos agora ter en conta unha serie de **parámetros**, que definimos en función do visto anteriormente:

- $a_p, p \in \mathcal{P}$. Día de admisión de cada paciente p . Non aparecerá no modelo como tal, pero é importante de cara aos datos, xa que se $a_p = 0$ implicará que o paciente aínda non entrou ao sistema e $a_p = -1$ que xa terá fixadas citas.
- $r_p, p \in \mathcal{P}$. Día de lanzamento (ready day), día no que pode comezar o tratamento do paciente p . Recordamos de novo, que **aínda que xa estean fixos, non ten porque ser $r_p = 0, p \in \bar{\mathcal{P}}$** .

- $d_p, p \in \mathcal{P}$. Día de vencemento (due day), día no que, como límite, debería comezar o tratamento do paciente p . **No caso dos pacientes fixos, teremos que $d_p = r_p, \forall p \in \bar{\mathcal{P}}$.**
- $f_p, p \in \mathcal{P}$. Duración, en bloques de 5 minutos, de cada fracción do paciente p (consideramos que sempre teñen a mesma duración).
- $n_p, p \in \mathcal{P}$. Número de fraccións (restantes no caso de pacientes fixos) de cada paciente p , de acordo ao seu plan proposto.
- $C_l^t, l \in \mathcal{L}, t \in \mathcal{T}$. Capacidade no día t no acelerador lineal l en número de bloques. En xeral, sen perda de xeneralidade, suporemos que esta sexa igual en todos os días (120 bloques).
- $\hat{C}_l^t, l \in \mathcal{L}, t \in \mathcal{T}$. Capacidade resultante de restar a C_l^t a cantidade de bloques ocupados polas citas fixadas dos pacientes $p \in \bar{\mathcal{P}}$ no día t no acelerador lineal l .
- $\omega_1, \omega_2, \omega_3$. Tres parámetros que determinan os pesos dos distintos termos da función obxectivo. Son libres do usuario o seu valor, pero deberían ser positivos. Os valores recomendados segundo [Pham et al. \(2022\)](#) serán $\omega_2 = \omega_3 = 1$ e $\omega_1 = 1000$ e comentaremos a motivación destes valores posteriormente.
- $\gamma \in [0, 1]$. Indica o porcentaxe de capacidade diaria do *linac* ($C_l^t, l \in \mathcal{L}, t \in \mathcal{T}$) que queremos reservar só para pacientes paliativos (categorías P1 e P2) que están pendentes de ser admitidos.

Pasamos agora a establecer as **variables de decisión** para este problema de optimización:

- $x_{pl}^t \in \{0, 1\}, p \in \hat{\mathcal{P}}, l \in \mathcal{L}, t \in \mathcal{T}$. Variable binaria que toma o valor 1 cando o paciente novo p é citado no día t no *linac* l , e 0 en caso contrario.
- $z_p^t \in \{0, 1\}, p \in \hat{\mathcal{P}}, t \in \mathcal{T}$. Variable binaria que toma o valor 1 cando o paciente novo p é citado por vez primeira no día t , e 0 en caso contrario.
- $w_{pl} \in \{0, 1\}, p \in \hat{\mathcal{P}}, l \in \mathcal{L}$. Variable binaria que toma o valor 1 cando o paciente novo p e asignado o acelerador lineal l durante o tratamento, e 0 en caso contrario.

De seguido, presentamos a **función obxectivo** e **as restriccións do modelo**. Debemos salientar que as metas alineadas con este subproblema van ser referentes principalmente á redución do atraso fóra do recomendado en pacientes e, logo de atender isto, á redución de tempo de atención e de cambios de acelerador lineal (por supoñer tamén un posible aumento de tempo na atención).

Hai varios obxectivos en mente, o que responde a unha situación propia de optimización multi-obxectivo, e, tendo en conta as diferencias de prioridade, unha técnica moi doada de aplicar será o método de suma ponderada ([Ehrgott, 2005](#)):

$$\text{minimizar} \quad \omega_1 \cdot \sum_{p \in \hat{\mathcal{P}}} \sum_{\substack{t \in \mathcal{T}, \\ t > d_p}} (t - d_p)^2 \cdot z_p^t + \omega_2 \cdot \sum_{p \in \hat{\mathcal{P}}} \sum_{t \in \mathcal{T}} (t - r_p)^2 \cdot z_p^t + \omega_3 \cdot \sum_{p \in \hat{\mathcal{P}}} \sum_{l \in \mathcal{L}} w_{pl} \quad (1.1)$$

Todas as variables están definidas para os pacientes novos, así que podemos adiantar que, cos parámetros anteriormente definidos, **non vamos a necesitar dos pacientes con citas prefixadas**, o cal reduce sensiblemente o tamaño do modelo. Hai tres termos nesta función obxectivo que representan cada un dos fins que xa mencionamos, multiplicados por cadanseu parámetro ω_i en función da orde.

- O primeiro termo vai acumular o cadrado dos días que tarde a atención de cada paciente novo, pero só a partir do día de vencemento (d_p) (sendo 0 o valor para o paciente $p \in \mathcal{P}$ se a súa primeira cita é antes de d_p).
- O segundo termo agrega o cadrado dos días que tarde cada paciente novo en ser atendido, a contar a partir do día de lanzamento (r_p).

- O último termo consiste en sumar para cada paciente novo o número de aceleradores lineais que necesita ao longo do seu tratamento.

Como vemos, de acordo a técnica do método da suma ponderada, o necesario para considerar os tres obxectivos será dar uns valores de ω_1, ω_2 e ω_3 positivos, e, se queremos dar prioridade ao primeiro sobre os demais, $\omega_1 > \omega_2$ e $\omega_1 > \omega_3$. Estas condicións son cumpridas polos valores propostos en [Pham et al. \(2022\)](#): $\omega_1 = 1000$ e $\omega_2 = \omega_3$.

Pasamos agora ás **restricións** do primeiro modelo e súas respectivas explicacións:

$$\sum_{l \in \mathcal{L}} x_{pl}^t \leq 1, \quad \forall p \in \hat{\mathcal{P}}, \forall t \in \mathcal{T} \quad (1.2)$$

A restrición (1.2) serve para impoñer que cada paciente $p \in \hat{\mathcal{P}}$ só pode, como moito, recibir unha única sesión por día.

$$\sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} x_{pl}^t \leq n_p, \quad \forall p \in \hat{\mathcal{P}} \quad (1.3)$$

A restrición (1.3) impide que un paciente $p \in \hat{\mathcal{P}}$ poida recibir máis fraccións das estipuladas no seu plan de tratamento.

$$\begin{aligned} \sum_{l \in \mathcal{L}} (x_{pl}^t - x_{pl}^{t-1}) &\leq \sum_{l \in \mathcal{L}} x_{pl}^k \quad \forall p \in \hat{\mathcal{P}}, t \in \mathcal{T}, x_{pl}^{-1} = 0, \\ &k = t + 1, \dots, \min\{|\mathcal{T}| - 1, t + n_p - 1\} \end{aligned} \quad (1.4)$$

A restrición (1.4) serve para cumprir a consecutividade das citas. Vamos a analizala cun maior detalle.

Se escollemos un paciente novo $p \in \mathcal{P}$ e un día $t \in \mathcal{T}$, teremos que a parte da esquerda da inecuación pode ter só tres valores, tendo en conta a limitación da restrición (1.2) e a binariedade da variable x_{pl}^t .

- $\sum_{l \in \mathcal{L}} (x_{pl}^t - x_{pl}^{t-1}) = \sum_{l \in \mathcal{L}} (x_{pl}^t) - \sum_{l \in \mathcal{L}} (x_{pl}^{t-1}) = 1$. Este caso xorde cando o paciente p ten unha cita no día $t > 0$, pero non foi asignado ningún *linac* o día anterior (iniciou tratamento). Tamén estaría incluído o caso de que inicie o tratamento en $t = 0$, grazas a que definimos $x_{pl}^{-1} = 0, \forall p \in \hat{\mathcal{P}}, \forall l \in \mathcal{L}$.
- $\sum_{l \in \mathcal{L}} (x_{pl}^t - x_{pl}^{t-1}) = 0$. Sería o caso de que non se iniciase tratamento do paciente p en $t = 0$ ou que en $t > 0$ e $t - 1$ o paciente p teña cita en ambos días ou ningún deles.
- $\sum_{l \in \mathcal{L}} (x_{pl}^t - x_{pl}^{t-1}) = -1$. Só hai un caso: que o paciente p estea citado o día $t - 1 \geq 0$, pero non en t .

En calquera caso, como $\sum_{l \in \mathcal{L}} x_{pl}^k \geq 0$, os dous últimos casos non son de interese (non forzan nada sobre a parte dereita da inecuación).

A primeira casuística forzará que os seguintes $n_p - 1$ días de t haia alomenos unha sesión nalgún acelerador lineal l , a non ser que o noso horizonte de predición remate antes ($|\mathcal{T}| - 1$ será o último día do conxunto de días \mathcal{T} , ao ter en conta o día $t = 0$).

Non vai haber máis que os n_p días de citación do paciente p pola restrición (1.3).

$$x_{pl}^t = 0 \quad \forall p \in \hat{\mathcal{P}}, l \in \mathcal{L}, t \in \{1, \dots, r_p - 1\} \quad (1.5)$$

A restrición (1.5) impide que o paciente p poida ser citado en calquera acelerador lineal l antes do seu día de lanzamento r_p .

$$\sum_{p \in \hat{\mathcal{P}}} f_p \cdot x_{pl}^t \leq \hat{C}_l^t \quad \forall t \in \mathcal{T}, l \in \mathcal{L} \quad (1.6)$$

A restrición (1.6) fai que non se exceda a capacidade máxima dispoñible tras quitar as citas do pacientes xa fixos do día t e en cada acelerador l .

$$\sum_{p \in \hat{\mathcal{P}}_3 \cup \hat{\mathcal{P}}_4} f_p \cdot x_{pl}^t \leq \hat{C}_l^t - \gamma \cdot C_l^t \quad \forall t \in \mathcal{T}, l \in \mathcal{L} \quad (1.7)$$

A restrición (1.7) fai que a capacidade adicada aos pacientes curativos (non paliativos) non pode ser maior que aquela que quede dispoñible para ese día t e *linac* l tras retirar as citas dos pacientes $p \in \hat{\mathcal{P}}$ e sen ocupar unha proporción γ (que poderá definir o usuario) da capacidade máxima do día t e acelerador l .

Esta reserva de espazo, xunto coa restrición (1.6), será para pacientes novos no sistema e paliativos (prioridades 1 e 2).

$$z_p^t \geq \sum_{l \in \mathcal{L}} x_{pl}^t - \sum_{l \in \mathcal{L}} x_{pl}^{t-1} \quad \forall p \in \hat{\mathcal{P}}, t \in \mathcal{T}, x_{pl}^{-1} = 0 \quad (1.8)$$

$$\sum_{l \in \mathcal{L}} x_{pl}^t \geq z_p^t \quad \forall p \in \hat{\mathcal{P}}, t \in \mathcal{T} \quad (1.9)$$

As restricións (1.8) e (1.9) definen a variable z_p^t , forzando a que teña valor 1 cando se produza no día t a primeira cita do paciente p e que vala 0 nos días t onde non haia ningunha cita do paciente t .

Non pode haber casos de tratamentos que se interrompan e volvan a recomenzarse en días non consecutivos por mor de (1.2),(1.3) e (1.4), así que, en consecuencia $z_p^t \leq 1$ (aínda que xa o tínamos por definición) para calquera paciente $p \in \hat{\mathcal{P}}$ e día $t \in \mathcal{T}$.

$$\sum_{t \in \mathcal{T}} z_p^t = 1 \quad \forall p \in \hat{\mathcal{P}} \quad (1.10)$$

A restrición (1.10) forza a que todos os paciente novos comecen o seu tratamento nalgún día $t \in \mathcal{T}$.

$$w_{pl} \geq x_{pl}^t \quad \forall p \in \hat{\mathcal{P}}, \forall t \in \mathcal{T}, l \in \mathcal{L} \quad (1.11)$$

A restrición (1.11) forza a que $w_{pl} = 1$ cando haia algún día $t \in \mathcal{T}$ no que o paciente p estea empregando o acelerador lineal l .

$$x_{pl}^t, z_p^t, w_{pl} \in \{0, 1\} \quad \forall p \in \hat{\mathcal{P}}, \forall t \in \mathcal{T}, l \in \mathcal{L} \quad (1.12)$$

As restricións (1.12) son as que definen a o dominio de todas as variables; neste caso, todas son binarias.

Para construír o calendario óptimo (respecto ás metas previamente establecidas) de asignación de días e aceleradores lineais nos pacientes novos mediante os valores de x_{pl}^t no óptimo. Se $x_{pl}^t = 1$, isto implicará que o paciente p ten unha cita asignada nese día t no acelerador lineal l .

Ademais, debemos salientar que a formulación matemática presentada do problema pertence á programación enteira (IP). En concreto, tamén sería lineal (ILP), porque o grao da función obxectivo, con respecto ás variables, sería de grao 1 (están ao cadrado só os coeficientes que acompañan ás variables).

Para maior claridade para o lector, resumimos a continuación a primeira parte do modelo, incluíndo só a función obxectivo e as restricións sen as explicacións intercaladas:

$$\text{minimizar } \omega_1 \cdot \sum_{p \in \hat{\mathcal{P}}} \sum_{\substack{t \in \mathcal{T}, \\ t > d_p}} (t - d_p)^2 \cdot z_p^t + \omega_2 \cdot \sum_{p \in \hat{\mathcal{P}}} \sum_{t \in \mathcal{T}} (t - r_p)^2 \cdot z_p^t + \omega_3 \cdot \sum_{p \in \hat{\mathcal{P}}} \sum_{l \in \mathcal{L}} w_{pl} \quad (1.1)$$

$$\text{suxeito a } \sum_{l \in \mathcal{L}} x_{pl}^t \leq 1, \quad \forall p \in \hat{\mathcal{P}}, \forall t \in \mathcal{T} \quad (1.2)$$

$$\sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} x_{pl}^t \leq n_p, \quad \forall p \in \hat{\mathcal{P}} \quad (1.3)$$

$$\sum_{l \in \mathcal{L}} (x_{pl}^t - x_{pl}^{t-1}) \leq \sum_{l \in \mathcal{L}} x_{pl}^k, \quad \forall p \in \hat{\mathcal{P}}, t \in \mathcal{T}, x_{pl}^{-1} = 0, \quad (1.4)$$

$$k = t + 1, \dots, \min\{|\mathcal{T}| - 1, t + n_p - 1\}$$

$$x_{pl}^t = 0, \quad \forall p \in \hat{\mathcal{P}}, l \in \mathcal{L}, t \in \{1, \dots, r_p - 1\} \quad (1.5)$$

$$\sum_{p \in \hat{\mathcal{P}}} f_p \cdot x_{pl}^t \leq \hat{C}_l^t, \quad \forall t \in \mathcal{T}, l \in \mathcal{L} \quad (1.6)$$

$$\sum_{p \in \hat{\mathcal{P}}_3 \cup \hat{\mathcal{P}}_4} f_p \cdot x_{pl}^t \leq \hat{C}_l^t - \gamma \cdot C_l^t, \quad \forall t \in \mathcal{T}, l \in \mathcal{L} \quad (1.7)$$

$$z_p^t \geq \sum_{l \in \mathcal{L}} x_{pl}^t - \sum_{l \in \mathcal{L}} x_{pl}^{t-1}, \quad \forall p \in \hat{\mathcal{P}}, t \in \mathcal{T}, x_{pl}^{-1} = 0 \quad (1.8)$$

$$\sum_{l \in \mathcal{L}} x_{pl}^t \geq z_p^t, \quad \forall p \in \hat{\mathcal{P}}, t \in \mathcal{T} \quad (1.9)$$

$$\sum_{t \in \mathcal{T}} z_p^t = 1, \quad \forall p \in \hat{\mathcal{P}} \quad (1.10)$$

$$w_{pl} \geq x_{pl}^t, \quad \forall p \in \hat{\mathcal{P}}, \forall t \in \mathcal{T}, l \in \mathcal{L} \quad (1.11)$$

$$x_{pl}^t, z_p^t, w_{pl} \in \{0, 1\}, \quad \forall p \in \hat{\mathcal{P}}, \forall t \in \mathcal{T}, l \in \mathcal{L} \quad (1.12)$$

Feita xa a asignación de días dos pacientes que non tiñas ningunha cita previa adxudicada, pasamos a intentar buscar en cada día un horario adecuado no que poidan ser atendidos, tendo en conta os espazos xa ocupados por pacientes que xa foron planificados con anterioridade.

2.2. Segunda parte: priorizar preferencias temporais dos pacientes

A segunda fase do modelo completo consiste en facer que os pacientes novos, cuxos días de atención xa están adxudicados da primeira parte, teñan un horario de cita concreto, dividido dentro dos bloques que xa indicamos.

Partindo desta información, vamos a traballar cos seguintes **conxuntos** (volvemos a refrescar os conxuntos básicos, que se reciclan do primeiro subprograma) que aparecen na [Táboa 2.3](#).

Igual que nos conxuntos da primeira fase, a partir destes conxuntos, vamos a definir unha serie de **parámetros de interese**:

- $f_p, p \in \mathcal{P}$. Duración da fracción do paciente p .
- $C_l^t, t \in \mathcal{T}, l \in \mathcal{L}$. Capacidade máxima en bloques do acelerador lineal l no día t .
- $t_p^{min}, t_p^{max} \in [0, C_l^t], p \in \hat{\mathcal{P}}^C$. Intervalo de tempo preferente para atención do paciente p . Nos datos que dispoñemos, a capacidade máxima de cada *linac* é constante no tempo e son todos iguais entre si, polo que os límites de t_p^{min} e t_p^{max} son claros.

\mathcal{P}	Conxunto de pacientes no sistema
\mathcal{B}	Conxunto de bloques (ordenados temporalmente) por día
\mathcal{T}	Conxunto de días (unidade temporal) a planificar.
\mathcal{L}	Conxunto de aceleradores lineais dispoñibles
$\hat{\mathcal{P}}$	Conxunto de pacientes novos (sen citas asignadas antes da fase previa)
$\bar{\mathcal{P}}$	Conxunto de pacientes fixos (con citas asignadas antes da primeira parte)
\mathcal{P}_i	Conxunto de pacientes con prioridade $i \in \{1, 2, 3, 4\}$
$\hat{\mathcal{P}}_i = \mathcal{P}_i \cap \hat{\mathcal{P}}$	Conxunto de pacientes novos con prioridade $i \in \{1, 2, 3, 4\}$
$\hat{\mathcal{P}}^c = \hat{\mathcal{P}}_3 \cup \hat{\mathcal{P}}_4$	Conxunto de pacientes novos curativos (non paliativos)
\mathcal{P}^t	Conxunto de pacientes que teñen cita no día $t \in \mathcal{T}$
\mathcal{P}_l^t	Conxunto de pacientes que están asignados ao acelerador lineal $l \in \mathcal{L}$ no día $t \in \mathcal{T}$
$\hat{\mathcal{P}}_l^t = \mathcal{P}_l^t \cap \hat{\mathcal{P}}$	Conxunto de pacientes que non tiñan citas previas na primeira fase (novos) que teñen cita no día $t \in \mathcal{T}$ no <i>linac</i> $l \in \mathcal{L}$
$\bar{\mathcal{P}}_l^t = \mathcal{P}_l^t \cap \bar{\mathcal{P}}$	Conxunto de pacientes que si tiñan citas previas na primeira fase (fixos) que teñen cita no día $t \in \mathcal{T}$ no <i>linac</i> $l \in \mathcal{L}$
$(\hat{\mathcal{P}}^c)^t = \hat{\mathcal{P}}^c \cap \mathcal{P}^t$	Conxunto de pacientes novos curativos que están citados no día t .
$(\hat{\mathcal{P}}^c)_l^t = \hat{\mathcal{P}}^c \cap \mathcal{P}_l^t$	Conxunto de pacientes novos curativos que están citados no día t e <i>linac</i> l .

Táboa 2.3: Conxuntos a empregar no modelo de planificación horaria.

En caso de ter diferenzas neste aspecto, recomendaríamos usar a definición $t_p^{min}, t_p^{max} \in [0, \max\{C_l^t, \forall t \in \mathcal{T}, \forall l \in \mathcal{L}\}]$, $p \in \hat{\mathcal{P}}^c$.

Todos os pacientes $p \in \mathcal{P}$ posúen en xeral un tempo de preferencia de atención, pero no caso dos pacientes fixos entendemos que xa se tivo en conta anteriormente, e nos pacientes paliativos priorízase a atención rápida, polo que nos abstemos de levar a cabo unha preferencia deste tipo.

- \bar{b}_p^t , $p \in \bar{\mathcal{P}}$, $t \in \mathcal{T}$. Primeiro bloque no que se atendeu ao paciente fixo p na cita corresponde ao día t . Recordamos que podemos mover a cita deste paciente, pero queremos evitalo na medida do posible.
- k_p^t , $t \in \mathcal{T}$, $p \in (\hat{\mathcal{P}}^c)^t$. Acelerador lineal onde está atendido o paciente curativo novo p no día t .

Esta información pódese obter revisado os distintos conxuntos $(\hat{\mathcal{P}}^c)_l^t$ en función do *linac*.

Digamos que $p \in (\hat{\mathcal{P}}^c)_{l'}$, cun l' concreto, entón: $k_p^t = l'$. Será único: un paciente non pode facer dúas fraccións distintas no mesmo día nin cambiar de acelerador lineal durante unha sesión.

- $\omega_4, \omega_5, \omega_6$. Parámetros que se empregaran na función obxectivo para ponderar os diversos obxectivos deste subproblema, similar ao que xa vimos na primeira parte. Serán todos termos positivos e, a maior valor en comparación aos demais, maior prioridade do “obxectivo” que acompañe.
- Proponse dende [Pham et al. \(2022\)](#) os seguintes valores: $\omega_4 = 60$ e $\omega_5 = \omega_6 = 1$.

Cos parámetros xa expresados, pasamos a definir as **variables**, onde xa anotamos que, xunto aos conxuntos, debemos ter coidado á hora de definilas para que inabarcables en termos de dimensionalidade. Trataremos este punto tomando a primeira variable como exemplo.

- $y_{pb}^{tl} \in \{0, 1\}$, $p \in \mathcal{P}$, $b \in \mathcal{B}$, $t \in \mathcal{T}$, $l \in \mathcal{L}$. Variable binaria que toma o valor 1 cando o paciente p comeza a fracción do día t no acelerador lineal l no bloque b .

Un problema que xorde é que se tomamos en conta como aumenta o número de variables do tipo y_{pb}^{tl} en función do tamaño dos conxuntos, vemos que consiste en multiplicar a cardinal de $\mathcal{B}, \mathcal{L}, \mathcal{T}$ e \mathcal{P} . Poñendo por exemplo un caso hipotético, non moi complexo, con 120 bloques, 40 días de predición, 2 aceleradores lineais e 100 pacientes, xa estaríamos falando de case 50 millóns de variables...

- $\bar{\phi}_p^t, p \in \bar{\mathcal{P}}^c, t \in \mathcal{T}$. Variable enteira non negativa que representa en cantos bloques se adiantou a cita do paciente fixo p no día t .
- $\underline{\phi}_p^t, p \in \bar{\mathcal{P}}^c, t \in \mathcal{T}$. Variable enteira non negativa que representa en cantos bloques se atrasou a cita do paciente fixo p no día t .
- $\bar{\Delta}_p^t, p \in \hat{\mathcal{P}}^c, t \in \mathcal{T}$. Desviación, en número de bloques, do inicio de sesión que se lle asignou ao paciente p no día t , respecto do límite inferior do seu tramo de atención preferido, t_p^{min} .
- $\underline{\Delta}_p^t, p \in \hat{\mathcal{P}}^c, t \in \mathcal{T}$. Desviación, en número de bloques, do inicio de sesión que se lle asignou ao paciente p no día t , respecto do límite superior do seu tramo de atención preferido, t_p^{max} .
- $\bar{\theta}_p, p \in \hat{\mathcal{P}}^c$. Bloque que máis **tarde** se lle asignou ao paciente novo p para iniciar tratamento, de entre todos os días e aceleradores lineais.
- $\underline{\theta}_p, p \in \hat{\mathcal{P}}^c$. Bloque que máis **cedo** se lle asignou ao paciente novo p para iniciar tratamento, de entre todos os días e aceleradores lineais.

Tras isto, vamos primeiro a establecer cales son os **obxectivos** deste modelo e formular a función obxectivo como tal, na que aplicaremos de novo a técnica multiobxectivo da suma ponderada.

O máis prioritario para nós vai ser evitar cambiar as citas dos pacientes fixos, a non ser que supoña moito beneficio a nivel de preferencia horaria nos pacientes novos, xa que supón un problema administrativo, ademais dun inconveniente para o paciente que xa axustou seu horario a unha cita programada.

Logo, estarán dous obxectivos referentes aos pacientes novos: intentar que as súas citas non estean fóra dos seus horarios preferentes e que o rango, en bloques, das citas non sexan moi grande durante o tratamento.

Creamos a seguinte función obxectivo para intentar satisfacer todos estes requirimentos:

$$\text{minimizar } \omega_4 \cdot \sum_{p \in \bar{\mathcal{P}}^c} \sum_{t \in \mathcal{T}} (\bar{\phi}_p^t + \underline{\phi}_p^t) + \omega_5 \cdot \sum_{p \in \hat{\mathcal{P}}^c} \sum_{t \in \mathcal{T}} (\bar{\Delta}_p^t + \underline{\Delta}_p^t) + \omega_6 \cdot \sum_{p \in \hat{\mathcal{P}}^c} (\bar{\theta}_p - \underline{\theta}_p) \quad (2.1)$$

Cada termo volve representar un dos obxectivos:

- O primeiro termo acumula para cada paciente fixo $p \in \bar{\mathcal{P}}$ os bloques de cambio (tanto por adiantar como por atrasar) que sufriu nas súas citas previamente fixadas.
- O segundo suma nos pacientes novos a desviación (tanto por cedo como por tarde) con respecto da ventá de tempo preferente de cada paciente novo curativo $p \in \hat{\mathcal{P}}^c$ en todos os seus días $t \in \mathcal{T}$ onde está programada a citación.
- O último termo representa a diferenza para cada paciente novo curativo $p \in \hat{\mathcal{P}}^c$ do bloque no que máis tarde comezou a ser atendido respecto do que máis cedo foi atendido.

Ao darlle máis importancia ao primeiro obxectivo, os cales queremos ter en conta aínda que con distinta valoración, deberíamos tomar como ponderacións $\omega_4 > \omega_5 = \omega_6 > 0$; as propostas do artigo orixinal, polo tanto, teñen sentido para esta finalidade: $\omega_4 = 60, \omega_5 = \omega_6 = 1$.

Finalmente, daremos forma a este subproblema cunha serie de **restricións**:

$$\sum_{b \in \mathcal{B}} y_{pb}^{tl} = 1 \quad \forall t \in \mathcal{T}, l \in \mathcal{L}, p \in \mathcal{P}_l^t \quad (2.2)$$

Exiximos con (2.2), para todos os pacientes en $p \in \mathcal{P}_l^t$, é dicir, que teñen cita o día $t \in \mathcal{T}$ no acelerador lineal $l \in \mathcal{L}$, haia algún bloque no que comece a fracción do paciente p (no día t e acelerador lineal l).

$$\sum_{p \in \mathcal{P}_l^t} y_{pb}^{tl} \leq 1 \quad \forall b \in \mathcal{B}, t \in \mathcal{T}, l \in \mathcal{L} \quad (2.3)$$

A restrición (2.3) indica que non hai ningún bloque $b \in \mathcal{B}$ concreto dun día $t \in \mathcal{T}$ e acelerador lineal $l \in \mathcal{L}$ onde poidan comezar simultaneamente dous pacientes a súa fracción.

$$\begin{aligned} y_{pb}^{tl} &\leq 1 - y_{p'b'}^{tl} \quad \forall t \in \mathcal{T}, l \in \mathcal{L}, p, p' \in \mathcal{P}_l^t, p \neq p' \\ b &= 0, \dots, |\mathcal{B}| - f_p \\ b' &= s + 1, \dots, s + f_p - 1 \end{aligned} \quad (2.4)$$

Para interpretar a restrición (2.4) temos que partir dun día $t \in \mathcal{T}$, un acelerador lineal $l \in \mathcal{L}$ e un paciente $p \in \mathcal{P}_l^t$ (citado o día t no *linac* l).

Se hai outro paciente $p' \in \mathcal{P}_l^t$, entón, o feito de que o paciente p comece a súa sesión no bloque $b \in \mathcal{B}$ (é dicir, $y_{pb}^{tl} = 1$) forza a que ningún bloque $b' \in \mathcal{B}$ sexa o comezo no día t e *linac* l para outro paciente p' durante os $f_p - 1$ seguintes bloques.

Como é evidente, isto só ten sentido cando a primeira sesión do paciente p dá marxe a que remate o tratamento (esta imposibilidade a valoraremos no (2.5)), polo que só temos que considerar esta restrición nos bloques s que van dende 0 (o primeiro turno) ata o bloque $|S| - 1 - (f_p - 1)$ (o último bloque do día restándolle a duración da sesión menos 1, que debe caber entre a sesión de inicio e a propia sesión de inicio de sesión), ou, de forma equivalente, ata o bloque $|S| - f_p$.

Anotamos adicionalmente que o conxunto de restricións (2.2) a (2.5) son moi necesarias polos motivos que indicamos á hora de definir y_{pb}^{tl} : reducen o espazo factible desta variable, o cal é enorme.

$$\begin{aligned} y_{pb}^{tl} &= 0 \quad \forall t \in \mathcal{T}, l \in \mathcal{L}, p \in \mathcal{P}_l^t, \\ b &= |S| - f_p + 1, \dots, |S| - 1 \end{aligned} \quad (2.5)$$

Xa indicamos que a restrición (2.5) impón que un paciente $p \in \mathcal{P}_l^t$ non pode ser comezado a atender cando non quedan bloques no día para completar a sesión.

Neste caso, se a súa fracción é de f_p bloques, entón non comezar dende $|S| - 1 - (f_p - 1) + 1 = |S| - f_p + 1$ (se inicias neste día, só se che quedan $f_p - 2$ para completar a fracción), ata o último día posible $|S| - 1$.

$$\bar{\phi}_p^t \geq \bar{b}_p^t - \sum_{b \in \mathcal{B}} b \cdot y_{pb}^{tl}, \quad \forall t \in \mathcal{T}, l \in \mathcal{L}, p \in \bar{\mathcal{P}}_l^t \quad (2.6)$$

A restrición (2.6) define $\bar{\phi}_p^t$ para cada paciente $p \in \bar{\mathcal{P}}$ fixo e día $t \in \mathcal{T}$.

Limita $\bar{\phi}_p^t$ inferiormente pola diferenza entre o bloque no que estaba previamente inicializada a cita do paciente fixo p no día t (\bar{b}_p^t) e o bloque polo que se modificou a inicialización da cita polo modelo. Como a función obxectivo é de minimización, e $\bar{\phi}_p^t$ ten signo positivo en (2.1), entón $\bar{\phi}_p^t$ será igual ao dito límite inferior.

Poñemos un exemplo de cal sería o efecto de adiantar a cita dun paciente fixo. Se unha sesión dun paciente fixo en principio debería comezar no bloque b e pretendemos inicializala no bloque $b-3$, entón $\bar{\phi}_p^t \geq 3$ (que aparece con signo positivo na función obxectivo).

Se a cita fose, polo contrario, atrasada, como a variable $\bar{\phi}_p^t$ non toma valores non negativos, a restrición será redundante.

$$\bar{\phi}_p^t \geq \sum_{b \in \mathcal{B}} b \cdot y_{pb}^{tl} - \bar{b}_p^t, \quad \forall t \in \mathcal{T}, l \in \mathcal{L}, p \in \bar{\mathcal{P}}_l^t \quad (2.7)$$

A restrición (2.7) define $\underline{\phi}_p^t$ para cada paciente $p \in \bar{\mathcal{P}}$ fixo e día $t \in \mathcal{T}$.

Limita $\underline{\phi}_p^t$ inferiormente pola diferenza entre o bloque polo que se modificou a inicialización da cita polo modelo e o bloque no que estaba previamente inicializada a cita do paciente fixo p no día t (\bar{b}_p^t). Como a función obxectivo é de minimización, e $\underline{\phi}_p^t$ ten signo positivo en (2.1), entón $\underline{\phi}_p^t$ será igual ao dito límite inferior.

Vamos a traballar de novo cun exemplo. Se unha sesión dun paciente fixo en principio debería comezar no bloque b e pretendemos inicializala no bloque $b+3$, entón $\underline{\phi}_p^t \geq 3$ (que aparece con signo positivo na función obxectivo).

Se a cita fose, polo contrario, adiantada, como a variable $\underline{\phi}_p^t$ non toma valores non negativos, a restrición será redundante.

$$\bar{\Delta}_p^t \geq y_{pb}^{tl} \cdot (t_p^{min} - b), \quad \forall b \in \mathcal{B}, t \in \mathcal{T}, l \in \mathcal{L}, p \in (\hat{\mathcal{P}}^c)_l^t \quad (2.8)$$

Con (2.8) limitamos inferiormente $\bar{\Delta}_p^t$ para cada cita t de cada paciente novo curativo p .

Cando o paciente comece a súa fracción nun bloque anterior ao límite inferior de preferencia do paciente (t_p^{min}), o límite inferior de $\bar{\Delta}_p^t$ aumenta (nunha unidade por bloque antes do preferido), de forma similar á restrición (2.6).

$$\underline{\Delta}_p^t \geq y_{pb}^{tl} \cdot (b - t_p^{max}), \quad \forall b \in \mathcal{B}, t \in \mathcal{T}, l \in \mathcal{L}, p \in (\hat{\mathcal{P}}^c)_l^t \quad (2.9)$$

Con (2.9) limitamos inferiormente $\underline{\Delta}_p^t$ para cada cita t de cada paciente novo curativo p .

Cando o paciente comece súa fracción nun bloque posteriormente ao límite inferior de preferencia do paciente (t_p^{max}), o límite inferior de $\underline{\Delta}_p^t$ aumenta (nunha unidade por bloque despois do preferido), de forma similar á restrición (2.7).

$$\bar{\theta}_p \geq \sum_{b \in \mathcal{B}} b \cdot y_{pb}^{tk}, \quad t \in \mathcal{T}, p \in (\hat{\mathcal{P}}^c)^t, k = k_p^t \quad (2.10)$$

Limitamos superiormente $\bar{\theta}_p$, que na función obxectivo ten signo negativo. As inecuacións (2.10) fan que non poida ser menor que o bloque co que comezou tratamento calquera día $t \in \mathcal{T}$ o paciente $p \in (\hat{\mathcal{P}}^c)^t$ (que é curativo, novo e tiña cita ese día t), no acelerador lineal que lle correspondía, que é k_p^t .

$$\underline{\theta}_p \leq \sum_{b \in \mathcal{B}} b \cdot y_{pb}^{tk}, \quad t \in \mathcal{T}, p \in (\hat{\mathcal{P}}^c)^t, k = k_p^t \quad (2.11)$$

Limitamos inferiormente $\underline{\theta}_p$, que na función obxectivo ten signo negativo. As inecuacións (2.11) fan que non poida ser maior que o bloque co que comezou tratamento calquera día $t \in \mathcal{T}$ o paciente $p \in (\hat{\mathcal{P}}^c)^t$, no acelerador lineal que lle correspondía, que é k_p^t .

$$y_{pb}^{tl} \in \{0, 1\}, \quad \forall t \in \mathcal{T}, l \in \mathcal{L}, i \in \mathcal{P}_l^t \quad (2.12)$$

$$\bar{\theta}_p^t, \underline{\theta}_p^t \in [0, C_l^t], \quad \forall t \in \mathcal{T}, i \in \bar{\mathcal{P}} \quad (2.13)$$

$$\bar{\Delta}_p^t, \underline{\Delta}_p^t, \bar{\theta}_p, \underline{\theta}_p \in [0, C_l^t], \quad \forall t \in \mathcal{T}, i \in \hat{\mathcal{P}}^c \quad (2.14)$$

Este conxunto de restricións serve para definir a binariedade das variables y_{pb}^{tl} -(2.12)-, o rango das variables aplicadas aos pacientes fixos -(2.13)- e o rango das variables aplicadas aos pacientes novos curativos -(2.14)-.

De igual forma que na definición de t_p^{max} e t_p^{min} para $p \in \hat{\mathcal{P}}^c$, recomendamos substituír o intervalo $[0, C_l^t]$ por $[0, \max\{C_l^t, l \in \mathcal{L}, t \in \mathcal{T}\}]$ en caso de que non sexan iguais as capacidades para todos os aceleradores lineais todos os días.

Este modelo, como está definido, é un problema de programación lineal enteira mixta (MILP), xa que só hai restrición de binariedade (2.12) explícita nas variables y_{pb}^{tl} , mentres que as demais variables teñen dominio nos reais, \mathbb{R} .

Malia isto, por mor das limitacións da súa definición e o seu efecto na función obxectivo, a efectos prácticos, sempre tomarán todas valores enteiros en calquera solución óptima.

Volvemos a, igual que Sección 2.1, resumir todo a segunda parte do modelo escribindo só a función obxectivo e as restricións para que estea recollido de forma completa e concisa:

$$\text{minimizar} \quad \omega_4 \cdot \sum_{p \in \bar{\mathcal{P}}} \sum_{t \in \mathcal{T}} (\bar{\phi}_p^t + \underline{\phi}_p^t) + \omega_5 \cdot \sum_{p \in \hat{\mathcal{P}}^c} \sum_{t \in \mathcal{T}} (\bar{\Delta}_p^t + \underline{\Delta}_p^t) + \omega_6 \cdot \sum_{p \in \hat{\mathcal{P}}^c} (\bar{\theta}_p - \underline{\theta}_p) \quad (2.1)$$

$$\text{suxeito a} \quad \sum_{b \in \mathcal{B}} y_{pb}^{tl} = 1, \quad \forall t \in \mathcal{T}, l \in \mathcal{L}, p \in \mathcal{P}_l^t \quad (2.2)$$

$$\sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} x_{pl}^t \leq n_p, \quad \forall p \in \hat{\mathcal{P}} \quad (2.3)$$

$$y_{pb}^{tl} \leq 1 - y_{p'b'}^{tl}, \quad \begin{aligned} &\forall t \in \mathcal{T}, l \in \mathcal{L}, p, p' \in \mathcal{P}_l^t, p \neq p', \\ &b = 0, \dots, |\mathcal{B}| - f_p, \\ &b' = s + 1, \dots, s + f_p - 1 \end{aligned} \quad (2.4)$$

$$y_{pb}^{tl} = 0, \quad \begin{aligned} &\forall t \in \mathcal{T}, l \in \mathcal{L}, p \in \mathcal{P}_l^t, \\ &b = |\mathcal{B}| - f_i + 1, \dots, |\mathcal{B}| - 1 \end{aligned} \quad (2.5)$$

$$\bar{\phi}_p^t \geq \bar{b}_p^t - \sum_{b \in \mathcal{B}} b \cdot y_{pb}^{tl}, \quad \forall t \in \mathcal{T}, l \in \mathcal{L}, p \in \bar{\mathcal{P}}_l^t \quad (2.6)$$

$$\underline{\phi}_p^t \geq \sum_{b \in \mathcal{B}} b \cdot y_{pb}^{tl} - \bar{b}_p^t, \quad \forall t \in \mathcal{T}, l \in \mathcal{L}, p \in \bar{\mathcal{P}}_l^t \quad (2.7)$$

$$\bar{\Delta}_p^t \geq y_{pb}^{tl} \cdot (t_p^{min} - b), \quad \forall b \in \mathcal{B}, t \in \mathcal{T}, l \in \mathcal{L}, p \in (\hat{\mathcal{P}}^c)_l^t \quad (2.8)$$

$$\underline{\Delta}_p^t \geq y_{pb}^{tl} \cdot (b - t_p^{max}), \quad \forall b \in \mathcal{B}, t \in \mathcal{T}, l \in \mathcal{L}, p \in (\hat{\mathcal{P}}^c)_l^t \quad (2.9)$$

$$\bar{\theta}_p \geq \sum_{b \in \mathcal{B}} b \cdot y_{pb}^{tk}, \quad t \in \mathcal{T}, p \in (\hat{\mathcal{P}}^c)^t, k = k_p^t \quad (2.10)$$

$$\underline{\theta}_p \leq \sum_{b \in \mathcal{B}} b \cdot y_{pb}^{tk}, \quad t \in \mathcal{T}, p \in (\hat{\mathcal{P}}^c)^t, k = k_p^t \quad (2.11)$$

$$y_{pb}^{tl} \in \{0, 1\}, \quad \forall t \in \mathcal{T}, l \in \mathcal{L}, i \in \mathcal{P}_i^t \quad (2.12)$$

$$\bar{\theta}_p^t, \underline{\theta}_p^t \in [0, C_l^t], \quad \forall t \in \mathcal{T}, i \in \bar{\mathcal{P}} \quad (2.13)$$

$$\bar{\Delta}_p^t, \underline{\Delta}_p^t, \bar{\theta}_p, \underline{\theta}_p \in [0, C_l^t], \quad \forall t \in \mathcal{T}, i \in \hat{\mathcal{P}}^c \quad (2.14)$$

Con isto rematamos este apartado no que estudamos con detalle o modelo completo creado por [Pham et al. \(2022\)](#), e que no seguinte apartado implantaremos e analizaremos a nivel de computación.

Debemos asegurarnos que é aceptable a nivel técnico con datos baseados na realidade para o seu obxectivo principal: crear un horario de atención de pacientes de radioterapia atendendo ás distintas prioridades que vimos neste capítulo.

Capítulo 3

Implementación e análise computacional

Chegamos á parte final do traballo coa que pretendemos empregar de forma directa o modelo que vimos no [Capítulo 2](#).

O primeiro paso para lograr ver a súa utilidade vai ser a creación de escenarios de paciente fixos e novos (instancias) que permitan dar uns datos realistas para a aplicación do modelo.

A continuación, debemos describir os recursos técnicos que dispoñemos e a ruta de procesamento que seguen os nosos datos ata chegar a unha planificación de citas aceptable.

Finalmente, faremos unha análise dos resultados de diversas instancias a través de varias medidas de interese tanto a nivel de efectividade do código como de comparación de calidade da atención ao paciente.

3.1. Creación de instancias

Como xa indicamos na introdución a este capítulo, o noso primeiro paso vai ser propoñer uns escenarios coherentes coa realidade da atención radioterapéutica actualmente.

Para isto, tomaremos as instancias creadas dende [Pham *et al.* \(2022\)](#), veremos cales nos interesan (dentro das 20 á nosa disposición), tendo en conta os recursos dalgún dos hospitais de Galicia e explicaremos como se levou a cabo a súa creación.

O primeiro vai ser fixar que recursos queremos que aspiren a ter os nosos “centros” aos que van ir chegando os pacientes que necesitan radioterapia. Isto nos permitirán elixir as instancias que sexan unha mellor referencia para nós.

O máis crucial vai ser **decidir o número de aceleradores lineais** dos que dispón esta simulación. Para isto vamos a ver cantos aceleradores lineais posúen diversos hospitais e clínicas de Galicia, de acordo a [SEOM \(2020\)](#) (datos publicados no 2020):

- Hospital Meixoeiro, Vigo: 4 aceleradores lineais.
- Complexo Hospitalario Universitario de Vigo (CHUVI), Vigo: 3 aceleradores lineais.
- Hospital Povisa, Vigo: 1 acelerador lineal.
- Complexo Hospitalario de Santiago (CHUS), Santiago de Compostela: 3 aceleradores lineais.
- Centro Oncolóxico de Galicia, A Coruña: 2 aceleradores lineais.
- Complexo Hospitalario Universitario de Ourense (CHUO), Ourense: 2 aceleradores lineais.

De acordo a esta información, aspiramos que as nosas instancias teñan **entre 1 e 4 aceleradores lineais**, para poder traballar con todo o posible rango dos centros galegos.

Á parte, como nos baseamos no artigo [Pham et al. \(2022\)](#), que pretende resolver a problemática no CHUM, que ten 7 *linacs*, incluiremos na análise unha instancia de 7 aceleradores.

Isto tamén cubrirá o caso dos hospitais españois, xa que, segundo [SEOM \(2020\)](#), os que teñen maior cantidade de *linacs* en España chegan so ás 5 unidades.

Continuando coa intención de establecer unha certa conexión coa realidade, deberíamos tamén definir **durante cantos días deberemos planificar as sesións dos pacientes e cantos bloques temporais consideraremos que haia en cada un deses días**.

Propoñemos, en consonancia co artigo orixinal e coas súas instancias, un período de planificación de **40 días e 120 bloques** para cada día laborable. Como xa dixemos na [Sección 2.1](#), isto último coincidirá coa capacidade de cada *linac* en calquera día.

O número de días é razoable para conseguir un planificación dunha duración a medio prazo, que realmente sirva de asistencia e referencia para os pacientes e profesionais médicos. O horario por día sería equivalente, como cada bloque son 5 minutos, a un uso diario de 10 horas por cada acelerador lineal. Podería verse representado na realidade por un horario continuo entre as 10 da mañá e as 8 da tarde.

Só nos queda un último aspecto necesario para crear un escenario efectivo: **unha distribución inicial de pacientes** que inclúa persoas que xa iniciaron tratamento e teñen citas fixas pendentes por realizar como pacientes novos que aínda non foron admitidos a radioterapia.

Realizouse un proceso con certo carácter estocástico, co fin de simular a aleatoriedade cotiá. Seguíuse a seguinte metodoloxía: primeiro, créanse pacientes fixos que, en función do día que poidan comezar o tratamento, lles serán asignadas citas ata que se chegue a sobrepassar certo límite de capacidade; logo se completará a instancia con pacientes pendentes de citación.

O primeiro paso, vai ser a simulación do paciente como tal. Independentemente da súa natureza (fixo ou novo), antes de engadirse a calquera instancia, deberá **asignárselle un plan**. En [Pham et al. \(2022\)](#) creouse unha base que contén ata 5000 posibles plans, dos que se escollera un ao azar para cada paciente.

Isto determinará o seu número de sesións, a duración en bloques de cada sesión e a súa categoría de prioridade ($n_p, f_p, p \in \mathcal{P}$, e os conxuntos $\mathcal{P}_i, i \in \{1, 2, 3, 4\}$ respectivamente, tal e como se introduciron na [Sección 2.1](#)).

Como xa indicamos, os **primeiros pacientes sempre van ser fixos** e nos centraremos por agora só neles. Considerouse por defecto que o seu día de admisión é -1 (o cal quere dicir, tomando como defecto o día “actual” no que queremos facer a planificación como día “0”, que foron admitidos antes deste día).

Quedan por determinar os días de lanzamento (*ready day* ou *release day*, denominación utilizada exclusivamente nas súas bases de datos), días de vencemento (*due day*) e, de cara á seguinte fase, as súas preferencias temporais á hora de ser atendido ($t_p^{min}, t_p^{max}, \forall p \in \mathcal{P}$ que aparecen na [Sección 2.2](#)).

- Os días de lanzamento, para os pacientes fixos, terán como valores días entre 0 e 10, cunha gran cantidade de pacientes fixos con data de comezo o día 0 e sen moi grandes diferenzas entre os demais días.

Isto consideramos que será debido a que se quere emular a existencia de pacientes fixos que están no medio do seu tratamento (por tanto terán que seguir sendo tratados de forma continuada) ou que aínda están por iniciar o seu tratamento (pero teñen cita fixada con anterioridade).

- O día de vencemento tómase o mesmo que o de lanzamento. Como son citas xa fixadas, cuxa asignación de día e acelerador lineal non vamos a modificar, non ten sentido deixar unha marxe de varios días para que poidan ser atendidos por primeira vez nesta programación.
- Con respecto ás quendas de preferencia, traballaremos con valores entre 0 e 120. Isto é porque ao haber 120 bloques, podemos asumir que o bloque 0 (o primeiro bloque) dun día $t \in \mathcal{T}$ comeza no primeiro instante “0” do día t e remata no instante “1”.

Cos datos que dispoñemos, detectamos sen indicación previa do autor os intervalos de preferencia que aparecen en (Táboa 3.1), independentemente do tipo de paciente segundo a súa admisión. Aproveitamos para poñer estes intervalos temporais como categorías que se lle poderían atribuír de cara a facilitar ao paciente establecer este patrón de preferencia de atención.

(t_p^{min}, t_p^{max})	(0,120)	(0,30)	(30,90)	(90,120)
Possible contexto	Sen preferencia de horario	Preferencia horario mañá	Horario de mediodía ou media tarde	Preferencia horario tarde/noite

Táboa 3.1: Posibles horarios de preferencia de atención.

Xa tendo toda esta información, agora si, iremos engadindo pacientes fixos á instancia, un tras outro. Deterase este proceso cando o seguinte paciente a engadir faga que se supere o **90 % da capacidade dos aceleradores lineais do primeiro día**.

A partir de aquí, todos os pacientes a engadir despois serán novos. Para saber cantos ían a necesitar, fixeron unha análise preliminar de cantos se poderán incluír. Isto é porque se saturamos a instancia de pacientes novos, implicaría a existencia de atrasos moi grandes. Todos estes pacientes serán admitidos no noso día actual, ou “0”.

Logo de que sexan identificados por un plan de tratamento, igual que os pacientes fixos (e coa mesma información obtida que estes últimos), encargáronse de definir o día de lanzamento. Presentamos a referencia que se tomou directamente de [Pham et al. \(2022\)](#), pois en función da prioridade de atención do paciente, se asignará un día aleatorio de atención de entre os rangos que mostramos na [Táboa 3.2](#).

Prioridade	P1	P2	P3	P4
Rango de valores para <i>ready day</i>	$r_p = 0$	$r_p \in [0, 2]$	$r_p \in [5, 7]$	$r_p \in [5, 7]$

Táboa 3.2: Día de lanzamento ($r_p, p \in \hat{\mathcal{P}}$) para pacientes novos.

O día de vencemento, en contraste do que cabería esperar, que sería que emulase o visto na [Táboa 2.1](#), comprobamos que seguiu o patrón da [Táboa 3.3](#) (sen indicacións aparentes no artigo).

Prioridade	P1	P2	P3	P4
Rango de valores para <i>ready day</i>	$v_p = 1$	$v_p = 3$	$v_p = 10$	$v_p = 20$

Táboa 3.3: Día de vencemento ($v_p, p \in \hat{\mathcal{P}}$, notación na [Sección 2.1](#)) para pacientes novos nos nosos datos.

Por último, os tempos de preferencia serán escollidos entre as mesmas posibilidades xa explicadas na [Táboa 3.1](#). Só resta explicar un último detalle para completar o escenario inicial: a distribución horaria para as citas que lles quedan aos pacientes fixos.

Segundo se indicou en [Pham et al. \(2022\)](#), foise engadindo aos pacientes a citas a medida que se ían integrando na instancia. Aínda así, nos datos non se pode comprobar este proceso, pois tamén se

indica no artigo que se aplicaron “factores estocásticos” para intercambiar entre pacientes sesións e ou aceleradores lineais, adiantar/atrasar citas,... co fin de asemellarse máis á realidade.

Isto completaría o proceso para crear un marco inicial sobre o que traballar. Engadimos no [Apéndice A](#) unha estrutura esquematizada de como sería este documento de cara á posterior implementación do modelo e un enlace para a descarga dos arquivo que conteñen as instancias. Mostramos agora un resumo das súas características na [Táboa 3.4](#).

Nome instancia	$ \mathcal{L} $	$ \mathcal{T} $	$ \mathcal{B} $	$ \mathcal{P} $	$ \mathcal{P} $	$ \hat{\mathcal{P}}_1 $	$ \hat{\mathcal{P}}_2 $	$ \hat{\mathcal{P}}_3 $	$ \hat{\mathcal{P}}_4 $
ins01	1	40	120	36	10	0	2	4	4
ins02	1	40	120	55	15	0	4	6	5
ins03	1	40	120	54	17	1	6	6	4
ins04	2	40	120	96	15	0	5	5	5
ins05	2	40	120	94	16	0	4	11	1
ins06	3	40	120	121	30	0	4	20	6
ins07	3	40	120	137	33	0	8	13	12
ins08	3	40	120	147	35	0	8	17	10
ins09	4	40	120	172	42	0	13	19	10
ins10	4	40	120	226	47	0	20	15	12
ins20*	7	40	120	332	82	0	23	39	20

Táboa 3.4: Descrición resumo de cada instancia. Marcamos ins20 con *, pois ten unha diferenza grande respecto das demais instancias en aceleradores lineais. Os significado de cada un dos conxuntos expresados está na [Sección 2.1](#).

Como vemos, traballaremos en total con 11 instancias, todas elas entre 1 e 4 aceleradores lineais, salvo a ins20, que dispón de 7. Polo que se indica na [Táboa 3.4](#), non buscamos que a ins20 centre a nosa análise xa que é só coherente co número de *linacs* do CHUM, mencionado en [Pham et al. \(2022\)](#).

De cara á análise computacional, só empregaremos esta instancia nalgúns ocasións, e, sobre todo, co fin de comparar os nosos resultados cos obtidos en [Pham et al. \(2022\)](#), sobre os datos que recolle do CHUM.

Continuando coa descrición das 10 primeiras instancias, o número de pacientes rexistrados en cada instancia tomará valores entre 36 pacientes totais e 226; o número de paciente novos estará entre 10 e 47 pacientes, en tendencia crecente ambos valores en función dos aceleradores lineais da instancia.

De cara ás categorías de prioridade deste tipo de pacientes, temos que en xeral hai un gran maioría de pacientes P3, seguidos dos das categorías P2 e P4. Só hai un paciente de prioridade P1 na ins03. Isto é debido a que, ao estar a base de datos dos plans recollida no CHUM, terá sentido que os porcentaxes de pacientes de cada categoría se asemllen aos datos da [Táboa 2.1](#).

Con isto rematamos a xeración de escenarios e veremos que ruta deben recorrer estas instancias ata chegar a obter uns calendario de citas de acordo ao noso modelo principal.

3.2. Implementación e requisitos técnicos

A implementación do modelo debe ser a seguinte tarefa a realizar. Explicaremos, en liñas xerais, como funciona e como se vai propagando a información. Finalmente, explicaremos que recursos foron empregados na aplicación do código que está no [Apéndice B](#) e [Apéndice C](#). Enumeremos e describamos en detalle cada etapa deste procedemento:

1. Definir o “input” (datos de entrada) da primeira parte. Aquí incluímos o arquivo que describe o escenario inicial, cuxa creación xa comentamos na [Sección 3.1](#) e cun exemplo ilustrativo no [Apéndice A](#).

Ademais, no propio código deberemos definir os valores do parámetros $\omega_1, \omega_2, \omega_3$ e γ .

Facemos un recordatorio do seu significado (que está tamén na [Sección 2.1](#)):

- ω_1 . Parámetro estritamente positivo que canto maior sexa relativamente a ω_2 e ω_3 , maior prioridade terá para nós a redución dos atrasos no inicio da atención dos pacientes novos (fronte aos demais obxectivos).
 - ω_2 . Parámetro estritamente positivo que canto maior sexa relativamente a ω_1 e ω_3 , maior prioridade terá o tempo ata chegar a ser tratados os pacientes novos por primeira vez (fronte aos demais obxectivos).
 - ω_3 . Parámetro estritamente positivo que canto maior sexa relativamente a ω_1 e ω_2 , maior prioridade terá para nós evitar que os pacientes novos teñan cambios durante o tratamento de acelerador lineal (fronte aos demais obxectivos).
 - γ . Proporción de capacidade reservada para pacientes paliativos novos.
2. Código da primeira parte do modelo, datos de saída, con gráfica adicional. Esta instancia e os parámetros de entrada descritos anteriormente son suficientes para aplicar a primeira fase do modelo, cuxo código atópase no [Apéndice B](#).

O código é, de forma conceptual, unha implementación fidedigna da estrutura presentada na primeira fase do modelo, polo que non destacamos ningunha particularidade ao respecto. Só salientamos que está limitado o tempo de execución máximo a 3600 segundos (1 hora); se se sobrepasa, se devolve a mellor solución atopada para o problema obtida ata ese momento.

Como resultado de resolver este problema devólvenos varios obxectos:

- Unha serie de arquivos .csv, que representa cada un os aceleradores lineais da instancia. En cada un deles, teremos a información sobre canta capacidade é ocupada en cada *linac* en cada día por cada paciente novo.
- Unha saída por pantalla dunha serie de medidas relativas ao tempo de execución e os obxectivos máis importantes a optimizar na función obxectivo; neste subproblema van ser a agarda media e o atraso medio da primeira sesión que padecen os pacientes novos.

Un engadido adicional que fixemos foi, tomando como “input” a serie de arquivos .csv e a instancia orixinal, empregar o código do [Apéndice C](#) para obter unha visualización de como se distribuirían os pacientes novos nos distintos días, de cara a asistir neste decisión. Na [Figura 3.1](#) temos un exemplo para a ins06.

Este calendario de citas a nivel diario e de *linac* xérase a partir das variables nas que se acada o óptimo na primeira fase do modelo.

Indicamos en cor gris cales serían os días onde debería comezar o tratamento sen incorrer en atrasos para cada paciente, e, en verde, o tratamento realizado en cada acelerador lineal. Os números denotan o inicio e fin en días dos tramos continuados de días con sesión para cada paciente no mesmo acelerador. Se só hai un número no tramo, o paciente só asistiu ese día illado a ese acelerador lineal.

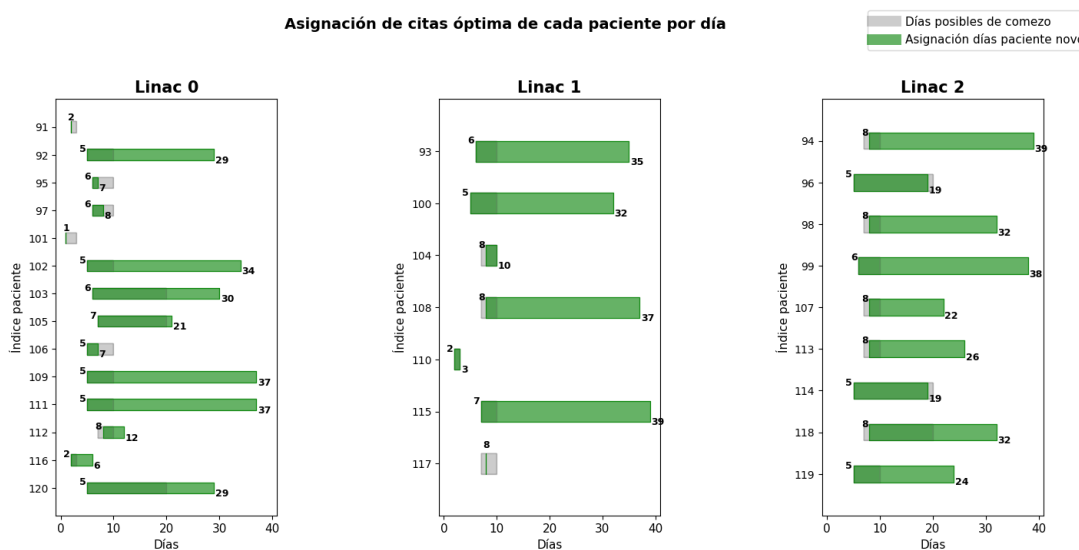


Figura 3.1: Calendario de citas segundo a asignación óptima da primeira fase do modelo para a ins06 (escolléronse como parámetros $\omega_1 = 1000$, $\omega_2 = \omega_3 = 1$ e $\gamma = 0$).

3. Definir os datos de entrada da segunda parte. Necesitamos tanto a instancia como o arquivo .csv da anterior fase. Ademais, debemos definir de novo unha serie de parámetros de entrada, que recordamos o seu significado que aparece na [Sección 2.2](#):
 - ω_4 . Parámetro estritamente positivo que, canto maior sexa en relación a ω_5 e ω_6 , maior impacto terá na función obxectivo a redución dos cambios horarios das citas dos pacientes fixos.
 - ω_5 . Parámetro estritamente positivo que, canto maior sexa en relación a ω_4 e ω_6 , maior impacto terá na función obxectivo incumprir os intervalos de atención preferente dos pacientes novos.
 - ω_6 . Parámetro estritamente positivo que, canto maior sexa en relación a ω_4 e ω_5 , maior impacto terá na función obxectivo que o rango horario do inicio das sesións ao longo do tratamento dos pacientes novos sexa o menor posible.
4. Código da segunda parte do modelo e datos de saída. Tamén temos un tempo de execución límite de 1 hora (3600 segundos), pero, nesta ocasión, debemos facer unha larga anotación sobre o código do [apéndice B](#): non podemos traballar exclusivamente cos conxuntos que aparecen na segunda parte do modelo.

Se só temos acceso a estes conxuntos, por exemplo, a restrición (2.4) requiriría revisar todos os días e aceleradores lineais, e, dentro disto, todos os posibles pares de pacientes nese conxunto, e analizar todos as posibles combinacións de bloques para cada par de pacientes.

Esta problemática tamén se traslada á restrición (2.5), na que teríamos que estudar cada día e *linac*, comprobar se cada paciente se está dentro dos citados nese día e analizar todos os bloques para cada paciente, para ver cales cumpren as condicións da restrición.

Como vemos, ambos casos fan un gran número de operacións que non levarían a ningún lado, polo que vémonos na obrigar de crear os seguintes conxuntos “auxiliares” que aparecen na [Táboa 3.5](#).

$(\mathcal{P}_l^t)_\mathcal{P}$	<p>Conxunto cuxas elementos conteñen son ternas (p, t, l) tales que desagregan cada \mathcal{P}_l^t para cada paciente.</p> <p>É dicir, para cada posible $p \in \mathcal{P}_l^t$, haberá unha tupla (p, t, l) como elemento de $(\mathcal{P}_l^t)_\mathcal{P}$.</p> <p>É equivalente a \mathcal{P}_l^t, só que indicando cos dous elementos finais da terna a que día e acelerador lineal pertence o paciente.</p>
$(\mathcal{P}_l^t)_{\mathcal{P}^2}$	<p>Para cada par ordenado do conxunto $(\mathcal{P}_l^t)_\mathcal{P}$ da forma $(\{p, t, l\}, \{p', t, l\}), p \neq p'$, créase un elemento de $(\mathcal{P}_l^t)_{\mathcal{P}^2}$ como unha 5-tupla da forma:</p> (p, p', t, l, f_p) <p>sendo f_p a duración en bloques de cada fracción do paciente p.</p>
\mathcal{R}_4	<p>Para cada elemento $(p, p', t, l, t_p) \in (\mathcal{P}_l^t)_{\mathcal{P}^2}$, configuramos un elemento en \mathcal{R}_3 da seguinte forma:</p> $(p, p', b_1, b_2, t, l), \quad \forall (p, p', t, l, t_p) \in (\mathcal{P}_l^t)_{\mathcal{P}^2},$ $\forall b_1 \in \{0, \dots, \mathcal{B} - f_p\}, \forall b_2 \in \{b_1 + 1, \dots, b_1 + f_p - 1\}$
\mathcal{R}_5	<p>Para cada elemento de $(\mathcal{P}_l^t)_\mathcal{P}$, creamos a seguinte 4-tupla como elemento de \mathcal{R}_5:</p> $(p, t, l, b), \quad \forall (p, t, l) \in (\mathcal{P}_l^t)_\mathcal{P}$ $\forall b \in \{ \mathcal{B} - f_p + 2, \dots, \mathcal{B} - 1\}$

Táboa 3.5: Conxuntos fóra do modelo necesarios para a súa implementación.

Con estes conxuntos, podemos cambiar a expresión das restricións (2.4) e (2.5):

$$y_{pb}^{tl} \leq 1 - y_{p'b'}^{tl} \quad (p, p', b, b', t, l) \in \mathcal{R}_4 \quad (2.4 \text{ alt})$$

$$y_{pb}^{tl} = 0 \quad (p, t, l, b) \in \mathcal{R}_5 \quad (2.5 \text{ alt})$$

Traballar con (2.4 alt) e (2.5 alt) fai que o modelo se cree máis rápido, pois terán menos índices que comprobar ao estar definidas sobre \mathcal{R}_4 e \mathcal{R}_5 , que se traballáramos directamente cos conxuntos e restricións base.

De feito, este cambio considerámolo imprescindible, ao ser en caso contrario **inviabile en termos de tempo** construír a segunda parte do modelo (tarda máis dunha hora en establecerse o modelo, e máis aínda executalo con calquera das nosas instancias).

Á parte, e de forma similar á primeira fase, a saída contén: unha táboa resumo con medidas relacionadas co tempo de execución e con respecto aos obxectivos que xa mencionamos; e un conxunto de .csv. Cada un deles representa un acelerador lineal e un día e serán matrices que tomarán valores 0 ou 1 dependendo do valor de y_{pb}^{tl} .

Se o paciente $p \in \mathcal{P}_l^t$ citado en $t \in \mathcal{T}$ e $l \in \mathcal{L}$ comezou a sesión no bloque $b \in \mathcal{B}$, entón temos que no arquivos correspondente a ese día e *linac*, a fila b e columna do paciente p ten valor 1; en caso contrario, terá valor 0.

Resumimos toda esta ruta de execución na [Figura 3.2](#) para deixar claro as distintas partes do proceso.

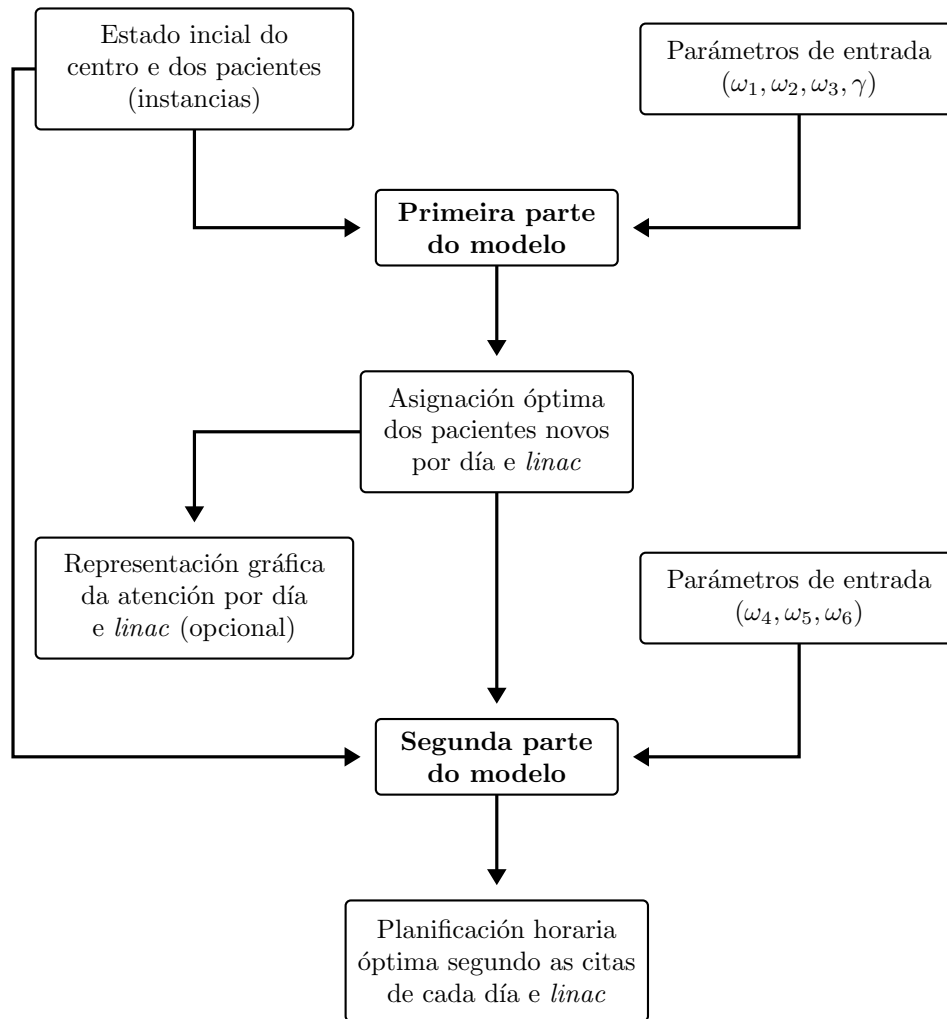


Figura 3.2: Fluxo de información durante a implementación do modelo.

Finalmente, comentamos brevemente as características técnicas da implementación para fomentar a fiabilidade da reprodución do código. Está escrito integramente na linguaxe de programación Python, cunha serie de paquetes que vamos a expresar de forma rápida tanto as versións empregadas como a súa utilidade:

- Numpy 2.3.1, ([Harris et al., 2020](#)). Paquete básico de vectores e matrices matemáticas xunto outras funcións matemáticas.
- pandas 2.3.0, ([The pandas development team, 2026](#); [Wes McKinney, 2010](#)). Paquete básico de manipulación de análise de datos en Python.
- Matplotlib 3.10.3, ([Hunter, 2007](#)). É un paquete que só empregaremos na saída gráfica opcional da primeira parte.
- Pyomo 6.9.2, ([Hart et al., 2011](#)). Paquete que pertence ao COIN-OR (Computational Infrastructure for Operation Research), unha comunidade que xera código aberto en relación a software

para a optimización. É unha linguaxe para modelar problemas de programación matemática moi útil, similar en natureza a outras como AMPL, pero do entorno de Python.

- Gurobi 12.0.2 ([Gurobi Optimization, LLC, 2026](#)). Como é un solucionador (denomínase tamén “solver”) comercial, necesitamos tamén dunha licenza (no noso caso, solicitouse unha licencia académica temporal) para aplicar Gurobi ao modelo de programación.

Hai múltiples *solvers* que se poden empregar, en función, normalmente, do tipo de problemas de optimización a resolver (LP, MILP, restricións polinómicas de calquera grao, ...). No noso caso temos dous problemas de programación lineal; malia isto, o primeiro pertence á programación enteira; e o segundo á programación enteira mixta.

Gurobi pode resolver este conxunto de problemas, polo que é adecuado o seu emprego. Aínda que Gurobi sexa un dos solucionadores máis coñecidos para afrontar problemas MILP e ILP, recordamos que hai outros que tecnicamente poden afrontar este tipo de formulacións:

- IBM ILOG CPLEX, ([IBM Corporation, 2022](#)). Emprégase no artigo de [Pham et al. \(2022\)](#) e foi creado por IBM Corporation. Tamén se require unha licenza para o seu emprego.
- CBC, ([Forrest e Lougee-Heimer, 2005](#)). *Solver* de código aberto mantido pola organización COIN-OR. Está dedicado fundamentalmente á programación enteira.
- HiGHS, ([Huangfu e Hall, 2018](#)). Tamén é de código aberto e o empregaremos ocasionalmente na análise computacional para contrastar a efectividade de Gurobi, que vai ser o noso *solver* preferente. Traballaremos coa versión 1.10.0 ([HiGHS Development Team, 2025](#)).

A execución dos datos foi feita nun ordenador portátil cun procesador AMD Ryzen 7 de 2,3 GHz, con 16GB de memoria RAM e sistema operativo Windows 10 de 64 bits. Ademais, empregouse o IDE (entorno de desenvolvemento integrado) PyCharm 2025.1.2 para traballar con Python.

3.3. Análise computacional da primeira parte do modelo

Recordamos brevemente os obxectivos que presenta a fase 1 do modelo proposto para resolver o problema de citas de radioterapia.

Ante todo, queremos evitar que os pacientes sexan tratados fóra do tempo recomendado en función da súa prioridade. Ademais, tamén buscaremos que os pacientes sexan tratados o antes posible. Malia querer celeridade no tratamento, permitiremos que poida haber cambios nos aceleradores lineais asociados para cada paciente, pero coa intención de que sexa o menor posible.

Traballaremos coas instancias vistas na [Sección 3.1](#). Como xa indicamos na [Sección 3.2](#), teremos que indicar, ademais da instancia de traballo, os valores asignados aos parámetros $\omega_1, \omega_2, \omega_3$ e γ .

Co fin de mostrar unha selección das utilidades que pode ter o noso código, propoñemos os seguintes casos de estudo:

- A nosa primeira configuración de parámetros de entrada mostrará os resultados obtidos tomando $\omega_1 = 1000, \omega_2 = 1, \omega_3 = 1$ e $\gamma = 0$, que priorizan evitar atrasos na atención fronte aos demais obxectivos. Esta proposta vén de [Pham et al. \(2022\)](#) (non está indicado o valor de γ como tal). Implicaría tamén, a efectos prácticos, ignorar a restrición (2.7) da primeira parte do modelo.

Esta casuística vai ser a única na que analizaremos tamén a ins20, pois podemos comparar algún dos datos que aparecen do CHUM en [Pham et al. \(2022\)](#). Tomaremos como referencia esta configuración de parámetros para afondar nos efectos que poidan ter nas medidas relacionadas coa atención ao paciente.

Adicionalmente, empregaremos a saída gráfica opcional da ins04 para mostrar en que *linacs* e días se dispoñerían os pacientes novos, en concreto, na [Figura 3.3](#). Será de utilidade, pois tamén mostraremos como se dispuxeron os pacientes nesta instancia graficamente, tamén con fins comparativos.

- Outro caso vai ser $\omega_1 = 1000, \omega_2 = 1, \omega_3 = 1$ e $\gamma = 0.1$; é dicir, hai un reserva do 10% en cada día para pacientes paliativos. Veremos como son os efectos a nivel xeral en todas as instancias, e, en concreto, compararemos que pacientes se viron afectados comparando os resultados gráficos da [Figura 3.3](#) cos da [Figura 3.4](#).
- Co fin de completar esta análise, propoñemos tamén os seguintes valores para os parámetros: $\omega_1 = 1, \omega_2 = 1000, \omega_3 = 1, \gamma = 0$.

Esta situación implica que o que se busca é que os pacientes novos inicien tratamento o antes posible; sen ter moito en consideración os “atrasos” que se calculan en función da súa prioridade. Volveremos a revisar o calendario óptimo para a ins04 dentro desta configuración para ver se detectamos algún cambio concreto fronte á [Figura 3.3](#).

3.3.1. Caso 1: $\omega_1 = 1000, \omega_2 = \omega_3 = 1, \gamma = 0$

Seguindo este esquema, vamos a comezar cos parámetros propostos por [Pham et al. \(2022\)](#). Dividiremos os datos que obtemos do código no [Apéndice B](#) en dúas táboas. Recordamos que na [Táboa 3.4](#) están recollidas as características principais de cada instancia.

Nesta primeira configuración analizaremos primeiro os resultados da [Táboa 3.6](#), que conterá o nome de cada instancia, os tempos de execución con Gurobi e con HiGHS, o *gap* de optimalidade da función obxectivo, os aceleradores lineais promedio polos que pasan os pacientes novos e os tempos de agarda para cada tipo de prioridade.

Cremos que convén advertir de como se realiza o cálculo do *gap* (relativo) de optimalidade: como estamos ante un problema de minimización, teremos que será valor igual a diferenza entre o límite superior (*upper bound*) e límite inferior (*lower bound*) atopados ao intentar resolver o problema, dividida polo *upper bound*.

O límite superior, neste caso, coincidirá coa menor solución obtida durante a execución, que vai chegar ao valor do límite inferior (cunha tolerancia de 10^{-4} unidades), ou ata que remata o tempo de execución de 3600 segundos.

Nome ins.	Tempo execución (HiGHS) (s)	Tempo execución (Gurobi) (s)	Gap	Media <i>linacs</i> paciente	Agarda media (días)				
					\hat{P}_1	\hat{P}_2	\hat{P}_3	\hat{P}_4	\hat{P}
ins01	3.09	0.34	0.0	1.0	-	1.0	6.25	8.0	5.9
ins02	6.63	0.95	0.0	1.0	-	1.25	10.16	14.4	9.2
ins03	6.53	1.05	0.0	1.0	0.0	3.16	10.5	14.75	8.29
ins04	7.09	0.79	0.0	1.07	-	1.0	7.6	10.0	6.2
ins05	15.26	1.71	0.0	1.13	-	1.25	7.36	11.0	6.06
ins06	52.4	3.5	0.0	1.0	-	1.75	6.5	6.0	5.77
ins07	431.42	10.36	0.0	1.06	-	1.0	8.38	10.92	7.52
ins08	603.3	8.64	0.0	1.06	-	1.0	7.64	10.1	6.83

ins09	439.56	19.4	0.0	1.07	-	0.69	7.74	10.9	6.31
ins10	2276.47	75.01	0.0	1.13	-	1.65	8.13	12.5	6.49
ins20	3600*	764.02	0.000083	1.13	-	1.13	8.56	10.85	7.04

Táboa 3.6: Resumo de medidas referentes ao tempo de execución, cambios de *linac* e agarda media por categoría de paciente na primeira fase do modelo con parámetros $\omega_1 = 1000, \omega_2 = \omega_3 = 1, \gamma = 0$.

Marcamos con * cando se acadou o límite do tempo de execución e con “-” cando non temos pacientes con prioridade P1 novos.

Un gran punto que vemos nestes resultados é que, en xeral, o solver Gurobi é claramente o máis adecuado para afrontar este problema: os tempos con HiGHS son moito máis elevados de forma relativa que os de Gurobi. Isto o visualizamos na [Táboa 3.7](#) onde facemos unha comparación relativa de ambos valores para cada unha das primeiras dez instancias (desbotamos a ins20 por non acadar o óptimo no tempo de execución).

Nome instancia	ins01	ins02	ins03	ins04	ins05
Aumento relativo do tempo de execución (%)	808.82 %	597.89 %	521.9 %	797.47 %	792.4 %

Nome instancia	ins06	ins07	ins08	ins09	ins10
Aumento relativo do tempo de execución (%)	1397.14 %	4064.29 %	6882.64 %	2165.77 %	2934.89 %

Táboa 3.7: Aumento relativo do tempo de execución de HiGHS respecto a Gurobi para cada instancia.

A conclusión obtida é que, a medida que aumentan os pacientes e aceleradores lineais, preséntase cada vez unha maior diferenza relativa entre cada solver, sendo todas elas superiores ao 500 % de cambio relativo a favor de Gurobi (é dicir, Gurobi atopa unha solución 5 veces máis rápido). Ademais, en casos de instancias máis grandes, como a ins10, ou as que aparecen en [Pham et al. \(2022\)](#), os tempos chegan a ser inadmisibles con HiGHS: estamos falando de máis de media hora de tempo de execución, cando a etapa realmente pesada vai ser a segunda parte.

Con isto confirmamos a eficacia de Gurobi para o problema que temos proposto. Aínda así, en todas as instancias chegamos a solución antes de rematar o tempo límite (3600 minutos, 1 hora). Por facer un leve comentario do *gap*, só vemos que a ins20 é a única na que non se chegou a pecha totalmente *gap*, pero este é practicamente nulo de igual maneira.

Analicemos agora as outras medidas obtidas na primeira parte do modelo. Vemos que, en xeral, incluso en instancias que teñen múltiples aceleradores lineais, non hai unha tendencia xeral a realizar moitos cambios de *linacs* durante o tratamento. A maioría dos pacientes realizan todo o seu plan nun único acelerador lineal. Enténdese que só se fan cando supoñen realmente un problema para que un paciente non chegue a padecer un atraso na súa primeira sesión, que ten a maior penalización na función obxectivo.

Pasamos agora a analizar o tempo de agarda medio, que é similar en todas as instancias e é coherente coa prioridade que ten cada categoría de cada paciente: a maior prioridade, menor tempo de agarda medio en cada instancia (con leves excepcións como a ins06).

Aproveitaremos para facer unha comparación dos tempos de agarda da ins20 cos tempos que se rexistraron no CHUM nos anos 2017-2018 (que ambos dispoñen tecnicamente dos mesmos aceleradores lineais) e que están anotados na [Táboa 2.1](#):

- Pacientes P1: Non temos instancias con pacientes P1 na ins20.
- Pacientes P2: 6.91 (CHUM) vs 1.13 (na ins20)
- Pacientes P3: 18.11 (CHUM) vs 8.56 (na ins03)
- Pacientes P4: 22.59 (CHUM) vs 10.85 (na ins03)
- Pacientes no seu conxunto: obtemos este valor para o CHUM tendo en conta os porcentaxes de pacientes por prioridade e o tempo medio de agarda para cada grupo.

Este valor é 16.39 días, moito maior que os 7.04 días da ins20.

Poderíamos concluír que este modelo supón unha mellora a nivel de tempo medio de agarda respecto do procedemento orixinal do CHUM, vista a gran mellora a nivel xeral.

Ademais destes datos, tamén queremos analizar algunha medida relacionada co obxectivo que queremos priorizar: a redución no atraso de atención de pacientes novos. Na [Táboa 3.8](#) ensinamos, para cada tipo de prioridade nos pacientes novos (e tamén para a totalidade dos pacientes novos), **os días de demora medios e a porcentaxe de pacientes que sofren atrasos**.

Os resultados da [Táboa 3.8](#) vemos que os días de atraso medio en xeral son moi baixos en todas as instancias, só salientado as ins02 e ins03 a nivel xeral. Só padecen días de atraso os pacientes de categoría P2 e P3, que ten lóxica pois son maioritarios en todas as instancias.

Ademais, se observamos os porcentaxes de atrasos, vemos que é moi preocupante o ratio de atrasos a nivel xeral tanto na ins02 como na ins03 (supera o 25 % da totalidade dos pacientes). De feito a porcentaxe de pacientes que non son atendidos nun tempo razoable da prioridade P2 (que seguen sendo paliativos e teñen dor intensa durante ese período) é do 50 % no ins03.

Un motivo desta saturación destas dúas instancias pode ser debido a que teñen moito máis número de pacientes que ins01. Como se ve na [Táboa 3.4](#), pasamos de 36 pacientes na ins01 a 55 e 54 respectivamente para ins02 e ins03.

Obviando estas instancias concretas, as demais presentan datos máis razoables. De feito, grazas ás saídas gráficas, puidemos detectar que moitos dos pacientes novos que sofren atraso, é só dun día (por iso coinciden en moitas ocasións os porcentaxes de atraso de cada categoría cos días de atraso en cada unha).

Vamos a comparar de novos os datos do CHUM nos anos 2017-2018 da [Táboa 2.1](#) fronte aos da ins20, pero desta vez, con respecto a porcentaxes de atrasos en cada categoría:

- Pacientes P1: Non temos instancias con pacientes P1 na ins20.
- Pacientes P2: 14.29 % (CHUM) vs 0 % (na ins20)
- Pacientes P3: 79.89 % (CHUM) vs 28.21 % (na ins03)
- Pacientes P4: 74.55 % (CHUM) vs 0 % (na ins03)
- Pacientes no seu conxunto: obtemos este valor para o CHUM tendo en conta os porcentaxes de pacientes por prioridade e os de ser atendido con atraso.

Con isto temos que o porcentaxe de pacientes atendidos fóra do período recomendado no CHUM foi de 61.86 % sobre a totalidade dos pacientes, mentres que na ins20 foi de 13.41 %.

Identificamos entón que a mellora sobre os atrasos de atención no CHUM é moi salientable, sobre todo porque, aínda que tomáramos como referencia a ins20, todas as instancias comparten esta lectura, exceptuando ins02 e ins03.

Nome ins.	Atraso medio (%)					Porcentaxe de atrasos (%)				
	$\hat{\mathcal{P}}_1$	$\hat{\mathcal{P}}_2$	$\hat{\mathcal{P}}_3$	$\hat{\mathcal{P}}_4$	$\hat{\mathcal{P}}$	$\hat{\mathcal{P}}_1$	$\hat{\mathcal{P}}_2$	$\hat{\mathcal{P}}_3$	$\hat{\mathcal{P}}_4$	$\hat{\mathcal{P}}$
ins01	-	0.0	0.0	0.0	0.0	-	0.0 %	0.0 %	0.0 %	0.0 %
ins02	-	0.0	1.83	0.0	0.73	-	0.0 %	66.67 %	0.0 %	26.67 %
ins03	0.0	0.83	0.5	0.0	0.47	0.0 %	50.0 %	33.33 %	0.0 %	29.41 %
ins04	-	0.0	0.2	0.0	0.07	-	0.0 %	20.0 %	0.0 %	6.66 %
ins05	-	0.0	0.27	0.0	0.19	-	0.0 %	27.27 %	0.0 %	18.75 %
ins06	-	0.0	0.0	0.0	0.0	-	0.0 %	0.0 %	0.0 %	0.0 %
ins07	-	0.0	0.08	0.0	0.03	-	0.0 %	7.69 %	0.0 %	3.03 %
ins08	-	0.0	0.0	0.0	0.0	-	0.0 %	0.0 %	0.0 %	0.0 %
ins09	-	0.0	0.26	0.0	0.12	-	0.0 %	26.32 %	0.0 %	11.90 %
ins10	-	0.05	0.33	0.0	0.13	-	5.0 %	33.33 %	0.0 %	12.77 %
ins20	-	0.05	0.33	0.0	0.13	-	0.0 %	28.21 %	0.0 %	13.41 %

Táboa 3.8: Táboa cos días de demora medio e porcentaxe de pacientes novos con atraso dividido en categorías de prioridade resultantes da primeira fase do modelo con parámetros $\omega_1 = 1000, \omega_2 = \omega_3 = 1, \gamma = 0$.

Por último, para ter unha mostra de cara a comparar os resultados doutras configuracións de parámetros de entrada, visualizamos o calendario de asignación de días e *linacs* da ins04 na [Figura 3.3](#).

Xa vimos un exemplo coa ins06 na [Figura 3.1](#), pero recordamos como se debe realizar a súa lectura: as barras verdes representan as sesións de radioterapia e as grises os períodos recomendados para iniciar tratamento. Cada unha das barras estará no ordenada reservada a cada paciente e terá a anchura que lle corresponda en función dos días que foi citado o paciente **de forma continuada**.

Para maior claridade, fíxemonos no paciente con índice “89” na [Figura 3.1](#), que aparece no eixo esquerda das gráficas que representan a cada un dos dous aceleradores lineais desta instancia. Comeza o seu tratamento no “linac 0” o día 7 e remata no día 39.

A primeira sesión é no “linac 1” o día 7, pero as seguintes sesións nos días 8 e 9 son no “linac 1”. Esta primeira sesión a indicamos cun segmento verde (barra de que comeza e remata no día 7) coincidindo co 7 no eixo de abscisas e co índice “89” no eixo de coordenadas. Como é unha sesión illada, indicamos que se realiza o día 7 marcándoo enriba do segmento.

Por outra banda, as sesións 8 e 9 fanse no “linac 1”, pero a 10 volve a ser no “linac 0”. Isto indicamos cunha barra no eixo de ordenadas con índice “89”. Abranguerá unha anchura dunha unidade (comeza no día 8 e remata no día 9) e aparecen na parte superior esquerda e na inferior dereita os límites no eixo de abscisas.

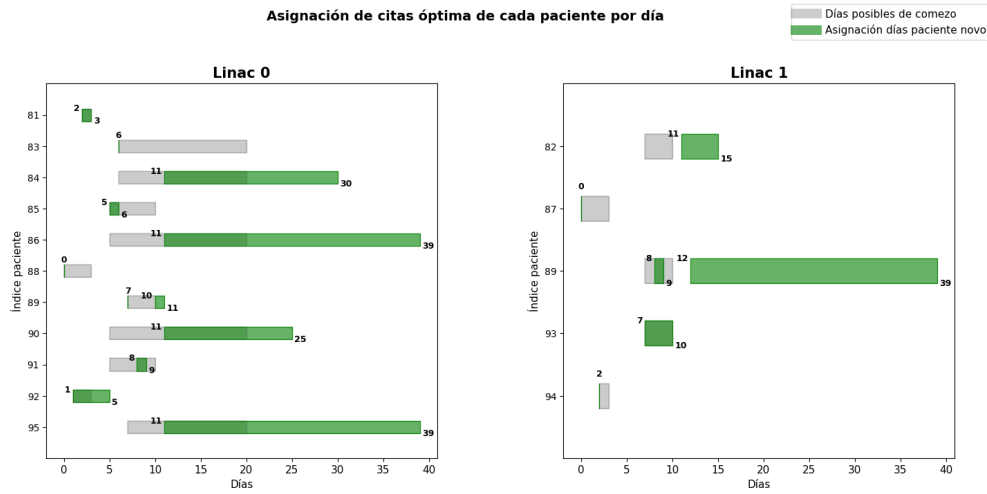


Figura 3.3: Asignación de días e *linacs* aos pacientes novos de ins04 na primeira fase do modelo con parámetros $\omega_1 = 1000, \omega_2 = \omega_3 = 1, \gamma = 0.1$.

3.3.2. Caso 2: $\omega_1 = 1000, \omega_2 = \omega_3 = 1, \gamma = 0.1$

Esta situación sería máis adecuada nun hospital público por exemplo, ao facer que haia un espazo equivalente ao 10% da capacidade total reservado para pacientes paliativos, que necesitan atención máis urxente.

Queremos avaliar con esta nova selección de parámetros que impacto teñen nos obxectivos referentes ao tempo de atención esta reserva de bloques, e ver se hai algún cambio no tempo de execución polo que vamos a construír dúas táboas coa seguinte información:

- **Táboa 3.9.** Conterá información sobre o tempo de execución (a partir de agora só empregaremos Gurobi), o *gap* de optimalidade e a agarda media dos pacientes novos por categoría de prioridade de cada instancia.
- **Táboa 3.10.** Será idéntica á **Táboa 3.8**; mostrará os días de atraso medio e porcentaxes de pacientes atendidos con demora por categoría.

Nome ins.	Tempo execución (s)	Gap	Agarda media (días)				
			\hat{P}_1	\hat{P}_2	\hat{P}_3	\hat{P}_4	\hat{P}
ins01	0.65	0.0	-	1.0	7.5	6.5	5.8
ins02	0.59	0.000059	-	1.25	13.17	16.0	10.93
ins03	0.86	0.0	0.0	3.17	10.33	15.25	8.35
ins04	0.59	0.0	-	1.0	10.4	12.8	8.07
ins05	1.88	0.000076	-	1.25	9.36	13.0	7.56

ins06	39.61	0.0	-	1.75	7.15	9.0	6.8
ins07	44.29	0.000095	-	1.0	11.46	12.58	9.33
ins08	5.03	0.0	-	1.0	9.94	11.9	8.46
ins09	9.09	0.000073	-	0.69	10.95	12.8	8.21
ins10	22.05	0.000045	-	1.65	10.8	14.42	7.83

Táboa 3.9: Tempo de execución e agarda media das ins01 a ins10 da primeira parte do modelo con parámetros $\omega_1 = 1000, \omega_2 = \omega_3 = 1, \gamma = 0.1$.

Unha lectura que podemos facer é que o *gap* en moitas instancias non é 0. Isto sucede porque finalizou a execución do programa cun *gap* é inferior á tolerancia por defecto (10^{-4}). Aínda así, consideramos este límite relativo suficiente para non ter que preocuparnos pola optimalidade das solucións.

A nivel de tempos de execución, non podemos sacar ningunha conclusión forte: hai instancias moito máis rápidas (ins09 ou ins10) e instancias moito máis lentas (ins05 ou ins06). Aínda así, en ambos casos, teñen tempos inferiores a 45 segundos.

Comparando cos datos da [Táboa 3.6](#), vemos que houbo en leve empeoramento xeral respecto tempos medios de agarda, pero nos tranquiliza que, en termos xerais, non é moi transcendental. Aínda que era previsible, malia o anterior, os tempos de agarda media de pacientes paliativos P2 redúcense ou permanecen estables (salvo na ins06 e ins10). Isto é lóxico pois non se vén afectados estes pacientes pola restrición (2.7) da primeira parte do modelo.

Pasamos agora a comentar coa [Táboa 3.10](#) o atraso na atención dos pacientes novos con esta configuración de parámetros de entrada.

Nome ins.	Atraso medio (días)					Porcentaxe de atrasos (%)				
	\hat{P}_1	\hat{P}_2	\hat{P}_3	\hat{P}_4	\hat{P}	\hat{P}_1	\hat{P}_2	\hat{P}_3	\hat{P}_4	\hat{P}
ins01	-	0.0	0.25	0.0	0.1	-	0.0 %	25.0 %	0.0 %	10.0 %
ins02	-	0.0	3.17	0.0	1.27	-	0.0 %	100.0 %	0.0 %	40.0 %
ins03	0.0	0.83	1.0	0.0	0.65	0.0 %	50.0 %	33.33 %	0.0 %	29.41 %
ins04	-	0.0	1.4	0.0	0.47	-	0.0 %	80.0 %	0.0 %	26.67 %
ins05	-	0.0	0.82	0.0	0.56	-	0.0 %	63.64 %	0.0 %	43.75 %
ins06	-	0.0	0.0	0.0	0.0	-	0.0 %	0.0 %	0.0 %	0.0 %
ins07	-	0.0	2.15	0.0	0.84	-	0.0 %	76.92 %	0.0 %	30.30 %
ins08	-	0.0	0.65	0.0	0.31	-	0.0 %	47.06 %	0.0 %	22.86 %

ins09	-	0.0	1.26	0.0	0.57	-	0.0 %	89.47 %	0.0 %	40.48 %
ins10	-	0.05	1.33	0.0	0.45	-	5.0 %	86.67 %	0.0 %	29.79 %

Táboa 3.10: Táboa cos días de demora medio e porcentaxe de pacientes novos con atraso dividido en categorías de prioridade resultantes da primeira fase do modelo con parámetros

$$\omega_1 = 1000, \omega_2 = \omega_3 = 1, \gamma = 0.1.$$

Destacamos un empeoramento xeneralizado da porcentaxe de atrasos a nivel global, sen mellorar aqueles de pacientes paliativos (tanto a ins03 como a ins10 seguen tendo o mesmo número de pacientes paliativos novos P2 atendidos con atraso).

Se miramos máis polo miúdo, podemos detectar que sobre todo son os pacientes P3 (curativos) os que sofren o aumento da porcentaxe de atrasos, o cal é lóxico, pois hai menor capacidade cada día para seren atendidos. Este porcentaxe estabilízase en torno ao 80 % nas instancias máis complexas.

Tamén é moi rechamante que na ins02 todos os pacientes novos P2 sexan atendidos con atraso, e, ademais, en media sexan atendidos con máis de 3 días de atraso respecto do que sería recomendable (valor moi similar aos datos da [Táboa 2.1](#)).

Con isto podemos concluír que esta reserva, malia non afectar tanto ao tempo de agarda absoluta, pode ser unha característica suficiente para que moitos pacientes atendidos dentro do marxe recomendado o deixen de estar. Finalmente mostramos a [Figura 3.4](#) co calendario de días e *linacs* para os pacientes da ins04.

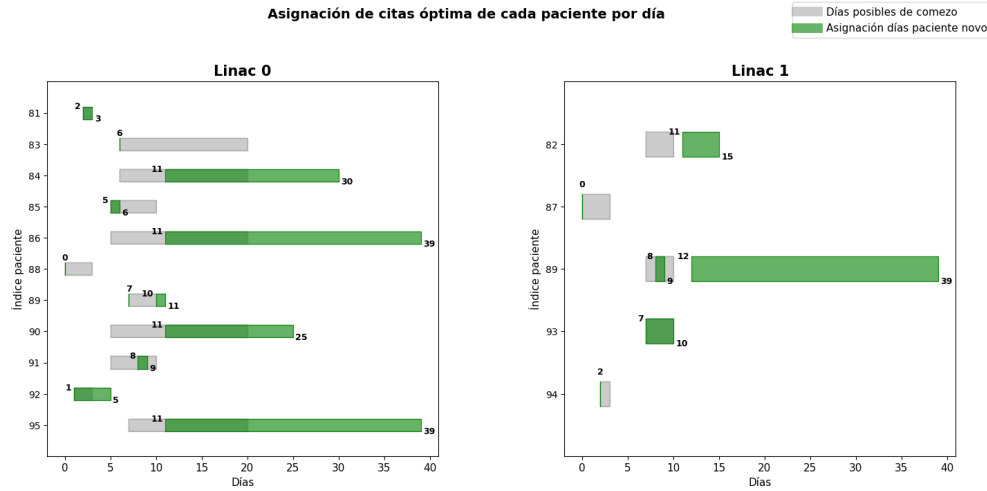


Figura 3.4: Asignación de días e *linacs* aos pacientes novos de ins04 na primeira fase do modelo con parámetros $\omega_1 = 1000, \omega_2 = \omega_3 = 1, \gamma = 0.1$.

3.3.3. Caso 3: $\omega_1 = \omega_3 = 1, \omega_2 = 1000, \gamma = 0$

A derradeira configuración que vamos a probar vai ser a que, fronte á situación base proposta en [Pham et al. \(2022\)](#), priorizaremos a atención rápida dos pacientes canto antes sexa posible. Este enfoque, ademais da non reserva dos pacientes paliativos, pode provocar atrasos na súa atención.

Este contexto sería máis propio dunha clínica privada na que nos dé igual a transferencia de información entre *linacs* e na que prima máis a satisfacción de cada paciente individual que a de evitar que todos os pacientes sexan atendidos dentro do tempo recomendado.

Igual que na anterior configuración, traballaremos coa [Táboa 3.11](#) na que ilustraremos información sobre o tempo de execución e tempo de agarda medio. Tamén teremos a [Táboa 3.12](#), que expón información sobre os atrasos nos pacientes en cada instancia.

Nome ins.	Tempo execución (s)	Gap	Agarda media (días)				
			\hat{P}_1	\hat{P}_2	\hat{P}_3	\hat{P}_4	\hat{P}
ins01	0.32	0.0	-	1.0	7.25	5.5	5.3
ins02	1.22	0.000059	-	1.25	10.83	13.6	9.2
ins03	1.61	0.0	0.0	3.17	11.17	11.5	7.76
ins04	1.16	0.000061	-	1.0	9.4	5.8	5.4
ins05	1.89	0.000082	-	1.25	7.36	11.0	6.06
ins06	3.57	0.0	-	1.75	6.5	6.0	5.77
ins07	15.35	0.000005	-	1.0	9.23	8.75	7.06
ins08	14.79	0.0	-	1.0	8.29	8.6	6.71
ins09	22.02	0.000033	-	0.69	8.89	8.1	6.17
ins10	545.44	0.000012	-	1.7	9.2	10.08	6.23

Táboa 3.11: Tempo de execución e agarda media das ins01 a ins10 da primeira parte do modelo con parámetros $\omega_2 = 1000, \omega_1 = \omega_3 = 1, \gamma = 0$.

Dentro do que cabería agardar, os tempos de agarda media a nivel conxunto redúcense respecto dos que temos na [Táboa 3.6](#), pero non cunha gran diferenza en moitas ocasións.

Na maioría das instancias detéctase que a mellora de tempo de agarda dáse sobre todo nos pacientes P4, de menor prioridade, o cal ten sentido pois, nun contexto de prioridade de evasión de atrasos, son o que máis marxe tiñan para ser pospostos sen implicar atrasos.

Con respecto aos tempos de execución, só salientamos o aumento que se vai producindo en todas as instancias e que se vai amplificando a medida que aumentamos o número de pacientes e aceleradores lineais (ins10 pasa de 75 segundos de execución na [Táboa 3.6](#) a máis de 500 na [Táboa 3.11](#)).

Pasamos agora á [Táboa 3.12](#) para analizar os efectos que tivo esta configuración no atrasos dos pacientes. En resumo, detectamos que hai aumento na cantidade de días de atraso nos pacientes P3 en certas instancias, e este aumento dáse, sobre todo, entre a ins07 e a ins10.

A nivel de pacientes paliativos só preocupa o leve aumento de atrasos nos pacientes P2 na ins10. Aínda así, esta configuración ten un efecto bastante moderado a nivel xeral sobre tanto os porcentaxes de atrasos como os días de atraso, pois seguen sendo bastante baixos ambos.

Por último, presentamos a derradeira referencia gráfica da ins04 na [Figura 3.5](#) para rematar a análise computacional da primeira fase do modelo. O seguinte paso será ver que posibilidades nos ofrece a segunda parte do modelo.

Nome ins.	Días de atraso medio					Porcentaxe de atrasos (%)				
	$\hat{\mathcal{P}}_1$	$\hat{\mathcal{P}}_2$	$\hat{\mathcal{P}}_3$	$\hat{\mathcal{P}}_4$	$\hat{\mathcal{P}}$	$\hat{\mathcal{P}}_1$	$\hat{\mathcal{P}}_2$	$\hat{\mathcal{P}}_3$	$\hat{\mathcal{P}}_4$	$\hat{\mathcal{P}}$
ins01	-	0.0	0.25	0.0	0.1	-	0.0 %	25.0 %	0.0 %	10.0 %
ins02	-	0.0	2.5	0.0	1.0	-	0.0 %	66.67 %	0.0 %	26.67 %
ins03	0.0	1.0	1.67	0.0	0.94	0.0 %	50.0 %	50.0 %	0.0 %	35.29 %
ins04	-	0.0	0.8	0.0	0.27	-	0.0 %	60.0 %	0.0 %	20.0 %
ins05	-	0.0	0.27	0.0	0.19	-	0.0 %	27.27 %	0.0 %	18.75 %
ins06	-	0.0	0.0	0.0	0.0	-	0.0 %	0.0 %	0.0 %	0.0 %
ins07	-	0.0	1.0	0.0	0.39	-	0.0 %	46.15 %	0.0 %	18.18 %
ins08	-	0.0	0.24	0.0	0.11	-	0.0 %	23.53 %	0.0 %	11.43 %
ins09	-	0.0	0.68	0.0	0.31	-	0.0 %	52.63 %	0.0 %	23.81 %
ins10	-	0.15	0.8	0.0	0.32	-	15.0 %	53.33 %	0.0 %	23.40 %

Táboa 3.12: Táboa cos resultados relacionados co atraso de atención dos pacientes da ins01 á ins10 na primeira fase do modelo con parámetros fixados $\omega_2 = 1000, \omega_1 = \omega_3 = 1, \gamma = 0$.

3.4. Análise computacional da segunda parte do modelo

Como indicamos na [Sección 3.2](#), o propósito agora é obter un horario dos distintos pacientes que son citados ao mesmo día e acelerador lineal. Un punto de partida desta sección é recordar cales son os diversos obxectivos que procuramos nesta planificación final.

O primordial, de acordo a [Pham et al. \(2022\)](#), será evitar dentro do posible modificar as citas xa fixadas para pacientes que xa estaban admitidos anteriormente. Os outros dous obxectivos son respectar as ventás temporais de preferencia na que desexan ser atendidos os pacientes novos e a redución do rango de horas ás que son citados.

Tendo en conta que ω_4, ω_5 e ω_6 son as ponderacións a ditos obxectivos, cuxa explicación tamén está na [Sección 3.2](#), propoñemos as seguintes casuísticas:

- O primeiro caso exposto tomará como parámetros de entrada $\omega_4 = 60, \omega_5 = \omega_6 = 1$. Esta configuración é a proposta en [Pham et al. \(2022\)](#) e ten como consecuencia a priorización da redución de cambios das citas dos pacientes fixos.
- Unha configuración alternativa que vamos a revisar é $\omega_4 = \omega_5 = \omega_6 = 1$. Isto indicaría que consideramos todos as metas que representan igual de importantes. Queremos ver se ten un efecto positivo para mellorar a calidade de atención dos pacientes novos, quitando esa prioridade aos pacientes fixos.

En ambos casos traballaremos coas instancias descritas na [Sección 3.1](#); aínda que só empregaremos a ins20 na primeira das configuracións para ver como se comporta a nivel de tempos de execución a segunda fase do modelo. Só empregaremos o solver Gurobi, pola xustificación indicada na [Sección 3.3](#).

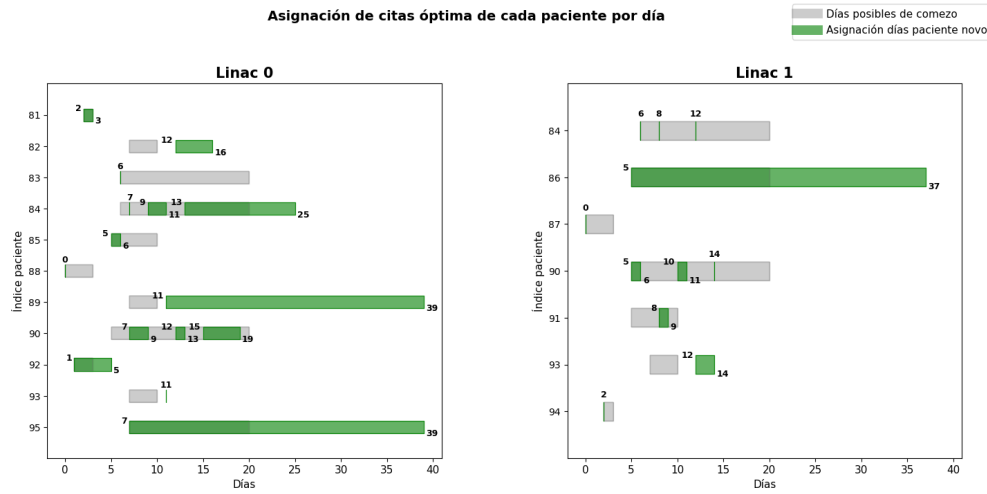


Figura 3.5: Asignación de días e *linacs* aos pacientes novos de ins04 na primeira fase do modelo con parámetros $\omega_2 = 1000, \omega_1 = \omega_3 = 1, \gamma = 0$.

Ademais, en calquera dos casos propostos teremos que as instancias pasaron pola primeira fase do modelo tomando como parámetros de entrada $\omega_1 = 1000, \omega_2 = \omega_3 = 1, \gamma = 0$, cuxa análise, se se precisa, está detallada na [Sección 3.3](#).

3.4.1. Caso 1: $\omega_4 = 60, \omega_5 = \omega_6 = 1$

Co fin de resumir os resultados obtidos, creamos a [Táboa 3.13](#), que está composta do nome da instancia, o tempo de execución, o *gap* de optimalidade, o valor da función obxectivo na solución óptima (ou na mellor atopada ata chegar ao límite de tempo de execución de 3600 segundos ou ata chegar ao límite de tolerancia do *gap*) e unha serie de medidas descritivas que explicamos a continuación:

- **Media cambios fixos.** Valor medio entre todos os pacientes fixos da desviación total (medida en bloques) das citas previas dentro dos días de planificación:

$$\frac{\sum_{p \in \hat{\mathcal{P}}} \sum_{t \in \mathcal{T}} (\bar{\phi}_p^t + \underline{\phi}_p^t)}{|\hat{\mathcal{P}}|}$$

- **Media non preferencia.** Valor medio entre todos os pacientes novos e os días de planificación da distancia en bloques das citas respecto do seu intervalo de atención de preferencia:

$$\frac{\sum_{p \in \hat{\mathcal{P}}} \sum_{t \in \mathcal{T}} (\bar{\Delta}_p^t + \underline{\Delta}_p^t)}{|\mathcal{T}| \cdot |\hat{\mathcal{P}}|}$$

- **Media rango temporal.** Valor medio entre todos os pacientes novos do rango temporal que teñen entre todas as súas citas dentro do período de planificación:

$$\frac{\sum_{p \in \hat{\mathcal{P}}} (\bar{\theta}_p - \underline{\theta}_p)}{|\hat{\mathcal{P}}|}$$

Con todo isto, xa podemos dar paso á análise dos resultados obtidas na [Táboa 3.13](#).

Nome ins.	Tempo execución (s)	Gap	F.obx en óptimo	Media cambios fixos	Media non preferencia	Media rango temporal
ins01	84.73	0.0	366.0	0.0	0.5	16.7
ins02	293.93	0.0	1076.0	0.05	1.06	21.47
ins03	220.08	0.0	1062.0	0.24	0.4	14.65
ins04	273.59	0.0	1132.0	0.1	0.4	27.33
ins05	385.27	0.0	2428.0	0.31	0.71	33.19
ins06	542.77	0.0	2343.0	0.02	0.96	35.8
ins07	583.76	0.0	3636.0	0.18	0.98	36.55
ins08	472.48	0.0	4074.0	0.410714	0.40	21.71
ins09	709.92	0.0	5628.0	0.35	0.99	28.81
ins10	1072.23	0.0	10916.0	0.79	0.72	22.34
ins20	2567.14	0.0	9184.0	0.324	0.64	27.12

Táboa 3.13: Táboa resumo das características técnicas e operativas das ins01 a ins20 tras aplicar a segunda parte do modelo con parámetros de entrada $\omega_4 = 60, \omega_5 = \omega_6 = 1$.

Podemos observar que os tempos de execución, aínda sendo realizables nun día, son elevados, o cal xa indica que o problema non é doado de escalar ao crecer exponencialmente a partir do número de pacientes e aceleradores lineais.

Tamén vemos que, aínda sendo os cambios de citas de pacientes fixos moi penalizados, en todos os casos (menos o primeiro) dáse algún cambio das súas citas. Parece reforzar que a posibilidade de que permitir cambiar citas a pacientes fixos ás veces favorece en gran medida á preferencia temporal e rango horario das citas dos pacientes novos. Malia isto son cambios moi pequenos en número de bloques: todos os valores rondan os 0.4 bloques de cambio de cita por paciente fixo durante toda a planificación.

En xeral, tamén detectamos que as violacións das preferencias de atención quedan en xeral por debaixo dun bloque por día de planificación e paciente novo, polo que, a priori, parece que si se teñen en conta estas preferencias. Debemos ser cautos aínda así con esta lectura, xa que hai moitos pacientes que non teñen preferencia establecida, o que implica que esta desviación é máis pronunciada para os pacientes que si esixen ser tratados en tramos diarios como os que vimos na [Táboa 3.1](#).

Por último, a media do rango temporal está entre 14 e 37 bloques. Se facemos unha conversión a intervalos temporais de 5 minutos, isto implicaría que para os pacientes novos, en media, pode variar o seu inicio de citas nun día a outro, como moito, de 70 a 185 minutos (dependendo da instancia). Considerámolo positivo, pois fará que, máis ou menos, os pacientes estean nun rango moi similar aos períodos de preferencia horaria (o máis pequenos na [Táboa 3.1](#) eran de 30 bloques).

Con isto rematamos o estudo desta configuración e pasamos a outra na que todos os obxectivos teñen a mesma prioridade e así comparar os efectos que levou esta decisión.

3.4.2. Caso 2: $\omega_4 = \omega_5 = \omega_6 = 1$

A configuración inicial proposta fará, a priori, que ningunha das metas presentes na función obxectivo da segunda parte do modelo teñan prioridade sobre a outra.

Vexamos na [Táboa 3.14](#) cales son os resultados que nos aporta cada unha das instancias saídas da primeira fase con argumentos $\omega_1 = 1000, \omega_2 = \omega_3 = 1, \gamma = 0$, coas mesmas medidas de interese que xa comentamos na [Táboa 3.13](#).

Engadimos desta vez o límite inferior (*lower bound*) da función obxectivo atopado polo programa porque desta vez si vamos ter instancias que sobrepasan o límite do tempo de execución.

Nome ins.	Tempo execución (s)	Gap	F.obx en óptimo	Límite inferior	Media cambios fixos	Media non preferencia	Media rango temporal
ins01	136.39	0.0	165.0	165.0	5.38	0.03	1.2
ins02	811.35	0.0	405.0	405.0	7.23	0.07	5.13
ins03	372.08	0.0	241.0	241.0	5.03	0.04	1.71
ins04	2658.46	0.0	213.0	213.0	1.75	0.0	4.73
ins05	1570.72	0.0	480.0	480.0	2.56	0.02	16.81
ins06	3288.96	0.0	1281.0	1281.0	7.11	0.16	14.83
ins07	3600.0*	0.00744	1344.0	1334.0	8.12	0.1	11.27
ins08	2108.54	0.0	746.0	746.0	4.62	0.05	4.6
ins09	3600.0*	0.008906	1572.0	1558.0	5.52	0.26	9.98
ins10	3600.0*	0.00939	1491.0	1477.0	5.27	0.07	8.66

Táboa 3.14: Táboa resumo das características técnicas e operativas das ins01 a ins20 tras aplicar a segunda parte do modelo con parámetros de entrada $\omega_4 = \omega_5 = \omega_6 = 1$. Marcamos con * aquelas instancias que chegan ao límite do tempo de execución.

O primeiro a destacar é que chegamos a varias instancias que non foron capaces de resolver a optimalidade global esta etapa cunha hora de tempo de execución (ins07, ins09 e ins10).

Queda patente que hai marxe de mellora, xa sexa a nivel de modelo ou de implementación para que os tempos de execución de instancias con máis pacientes e aceleradores lineais poidan resolverse nun tempo razoable (recordamos que non consideramos instancias de 7 aceleradores lineais, como ins20).

Este problema tamén se vía en [Pham et al. \(2022\)](#): no seu caso, a ins09 e ins10 tiñan un *gap* do 7.7% e 43.68%, respectivamente, cunha hora de execución. De feito, o noso *gap* é menor que o do artigo. Unha explicación a estas diferenzas é que aquí estase empregando como *solver* Gurobi, mentres que en [Pham et al. \(2022\)](#) empregaban CPLEX.

Ademais destas anécdotas, vemos que hai un incremento moi grande en todas as instancias do tempo de execución no caso 2 respecto ao caso 1. Visualizamos este aumento relativo [Táboa 18](#). Desbotamos ins07, ins09 e ins10 ao non acadar o óptimo no tempo de execución.

Esta aumento podería ser debido a que, en caso de haber unha preferencia moi grande sobre un dos

Nome instancia	ins01	ins02	ins03	ins04
Aumento relativo do tempo de execución (%)	60.97 %	176.04 %	69.07 %	1107.95 %

Nome instancia	ins05	ins06	ins08
Aumento relativo do tempo de execución (%)	474.11 %	505.96 %	346.27 %

Táboa 3.15: Aumento relativo do tempo de execución de cada instancia na segunda parte do modelo ao pasar ω_4 de 60 a 1 na primeira configuración.

obxectivos (como é o caso da [Táboa 3.13](#)), os demais termos da función obxectivo toman un segundo plano fronte a minimizar só o termo coa ponderación máis alta.

Por mor desta situación, pode ser moito máis doado minimizar un termo que tres, especialmente cando a redución dun deles pode implicar o aumento doutro.

Seguimos comparando os datos da [Táboa 3.13](#) cos da [Táboa 3.18](#). Como $\omega_4 = \omega_5 = \omega_6$, o obxectivo de reducir os cambios nas citas fixas dos pacientes perdeu a prioridade que tiña no caso 1 sobre os demais obxectivos.

Analizando os nosos datos, vemos que no caso 2 cada paciente fixo tivo cambios nas súas citas fixas de máis de 4 bloques de duración, en xeral (salvo ins04 e ins05, onde se rexistran menores valores). É dicir, que un paciente medio terá, entre todas as súas citas que quedaban por planificar, cambios nas súas horas de citación que, acumulando as diferenzas de tempo en cada cita con respecto ás que tiña previamente asignadas, sumarán máis de 20 minutos (4 bloques) de cambios temporais.

Na anterior configuración indicamos que en ningunha instancia se superaba, en media, 1 bloque de cambio en calquera das súas citas, polo que houbo un cambio substancial.

Como contraparte, as outras dúas medidas melloran os resultados de maneira moi clara. No caso de preferencia horaria, temos que, en xeral, hai menor distancia en bloques respecto dos límites daqueles pacientes novos que teñan quendas preferentes.

Con respecto á media do rango temporal, estamos entre 1 e 17 bloques como intervalo temporal onde poden variar, en media, para cada paciente novo a súas citas. Isto é só un período entre 5 e 85 minutos, o cal, se estamos falando dun tratamento de varias semanas, é bastante consecuente que as citas varíen ao moito unha hora e media.

Por tanto, podemos corroborar que si houbo un efecto positivo para os pacientes novos ao relaxar a prioridade que se aplica ás citas dos pacientes fixos.

Con isto, rematamos a análise global do modelo, tras facer unha lectura das posibles instancias, das diferentes etapas da implementación do código e do efecto que producen os parámetros de entrada tanto nas asignacións de *linacs* e días como na xestión do horario posterior.

Capítulo 4

Conclusión

Este documento expón o problema de planificación de citas de radioterapia (ou RTSP). Tras un revisión bibliográfica, escollemos entre os múltiples enfoques que relatamos o abordado en [Pham *et al.* \(2022\)](#). A súa característica principal é que afronta o RTSP mediante un proceso en dúas etapas. Isto permite satisfacer obxectivos de eficiencia temporal, pero tamén facilita a calidade de atención do paciente.

En primeiro lugar, realízase a planificación de pacientes novos no sistema a días e aceleradores lineais que están parcialmente ocupados por outros pacientes xa fixos. Posteriormente, en función da solución do problema anterior, asignaranse espazos temporais como tal dentro destes días e *linacs*.

O que denominamos como primeira fase, coa axuda de Python e do solver Gurobi, permítenos obter resultados moi rápidos para a asignación de citas de pacientes a días con centros de pacientes de ata 7 aceleradores lineais, o cal son condicións máis que suficientes para aplicarse ao sistema sanitario galego e español.

A comparación desta primeira fase con respecto aos resultados do CHUM fai ver que temos un avance fronte ao sistema actual de citas, que, cos mesmos recursos, fai un mellor aproveitamento; reducindo en gran medida a duración media de atención do paciente en casos de gravidade intermedia, e os porcentaxes de atrasos na atención.

Durante a análise computacional da primeira fase tamén vemos que unha reserva do 10% da capacidade para pacientes paliativos pode provocar un maior porcentaxe de atrasos, pero concentrado en pacientes de urxencia media-baixa.

A segunda parte, malia a súa maior complexidade, é manexable en duración ata instancias con 4 aceleradores lineais, non sobrepasando os 15 minutos de tempo de execución. Tamén é capaz de executalo para instancias de ata 7 *linacs* con tempo límite dunha hora, obtendo un resultado cun *gap* moi baixo .

Ademais de permitirnos crear un horario completo que determina as horas de atención a cada paciente, tamén nos axuda a xustificar os beneficios de permitir o leve cambio de citas de pacientes fixos para mellorar as condicións das citas dos pacientes que entren no sistema.

Como era previsible, aínda así, temos que ter en conta que, sobre todo, con esta segunda fase, hai un aumento pronunciado do tempo de execución, polo que entendemos que segue habendo marxe de mellora. Isto vese acentuado cando non queremos establecer ningunha prioridade entre os termos da función obxectivo.

Asemade, poderíanse considerar outros tratamentos radioterapéuticos que non se tiveron en conta: pacientes con sesións con descanso dun día fóra do fin de semana (tratamento alterno), doses máis pequenas de radiación en dúas ou máis sesións por día (tratamento hiperfraccionado), ...

Coa vista posta en posibles melloras e modificacións, consideramos, aínda así, que logramos chegar a obter un código cun resultados que melloran a situación real e que supoñen un investimento temporal moi asumible.

Damos remate, por tanto a esta revisión, explicación e aplicación dun modelo con datos baseados

na realidade para resolver o problema de asignación de citas de radioterapia (RTSP), agardando que sexa útil a nivel sanitario, investigador ou divulgativo.

Apéndice A

Exemplo de formato das instancias inicias: ins01

Mostramos a continuación a [Figura A.1](#), na que se aparecen os elementos de interese de cara á aplicación do modelo na instancia 1. Están dispoñibles para a súa descarga en http://hanalog.polymtl.ca/wp-content/uploads/2020/10/RTSP_dataset.zip. Aínda que todas as instancias estean recollidas en arquivos .csv, están convertidas ao formato .xlsx para maior claridade.

Como están en inglés, identificamos nunha lista os elementos de interese. Primeiro, aparecen características xerais da instancia:

- “Name”. Nome da instancia
- “K.” Número de aceleradores lineais dispoñibles.
- “S”. Número de bloques por día (os límites de cada bloque b serán entre b e $b + 1$).
- “T”. Número de días que é necesario planificar (é dicir, entre o día 0 e o 39 serán os índices de cada día).
- “Lambda”. Valor que intervéñ na xeración de pacientes novos. Non o empregamos.
- “scope in days”. Límite superior da duración total de todos os tratamentos (cun plan eficiente).
- “noSimulationDays”. Rango de días que poden ser admitidos os pacientes novos (só 1, o día 0).
- “currentday”. Día actual (en índice e data concreta); sempre é 0 así que non o empregamos.
- “nopatients”. Número de pacientes totais.
- “sumduration”. Suma da duración dunha sesión individual do conxunto total de pacientes. Non o empregamos.

Logo está a lista de pacientes, primeiro os fixos e logo os novos. Cada fila está composta de:

- “index”. Índice de cada paciente.
- “treatmentID”. ID, dentro dos 5000 plans dispoñibles, do plan asignado (non o empregamos).
- “careplan”. Nome do plan asignado (non o empregamos).
- “priority”. Prioridade de atención do paciente.
- “noSections”. Número de fraccións/sesións restantes do paciente para completar tratamento.

Name	ins01											
K	1											
S	120											
Lambda	20											
T	40											
scope in days	66											
noSimulationDays	1											
current day	0	01/03/2021										
no patients	36	sum duration 289										
index	treatmentID	patID	careplan	priority	noSections	admissionDay	releaseDay	dueDay	duration	TWMin	TWMax	
0	3717	3348	GYN Col pelvie P3		27	-1	0	0	5	90	120	
1	3486	3126	PAL Pal 800 IM P2		1	-1	10	10	6	0	120	
2	2039	1842	PMN S PMN S P3		3	-1	8	8	9	0	120	
3	163	149	ORL Glandes s P3		15	-1	9	9	5	90	120	
4	3432	3072	SEI Sein partie P4		18	-1	0	0	8	0	120	
5	3191	2861	PAL Pal 800 IM P2		5	-1	0	0	5	0	120	

↓

A continuación do conxunto de pacientes

↓

fixed appointment	180											
day	linac	index	appointmenttime	appointmentendtime								
0	0	14	0	6								
0	0	9	6	21								
0	0	23	21	33								
0	0	16	33	39								
0	0	4	39	47								
0	0	5	47	52								
0	0	21	52	61								
0	0	25	61	66								
0	0	11	66	71								
0	0	17	71	80								
0	0	10	80	86								
0	0	0	103	108								
0	0	20	108	113								
0	0	15	113	120								

Figura A.1: Estrutura de cada instancia de traballo. Na primeira imaxe, na esquina superior esquerda preséntanse datos xerais sobre a instancia, seguido, na parte inferior dalgúns datos sobre os pacientes que están no sistema. A segunda imaxe contén, despois da lista de pacientes, a asignación de citas dos que sexan fixos.

- “admissionDay”. Día no que foi admitido o paciente. Se é -1 o paciente é fixo; en caso contrario, novo.
- “releaseDay”. Día de lanzamento do paciente.

- “dueDay”. Día de vencemento do paciente.
- “duration”. Duración en bloques de cada fracción de cada paciente.
- “TWMin”. Límite inferior da ventá temporal preferente de cada paciente.
- “TWMin”. Límite inferior da ventá temporal preferente de cada paciente.

E, concluindo a descrición do arquivo, as citas dos pacientes fixos conterán a seguinte información:

- “fixed appointment”. Número de citas para pacientes fixos. Non o empregamos.
- “day”. Día da cita para paciente fixo.
- “linac”. Acelerador lineal para a cita do paciente fixo no día da primeira columna.
- “index”. Índice do paciente.
- “appointmenttime”. Recordamos que os bloques están definidos polo tempos onde “comezan” e onde “finalizan” (por exemplo, o bloque 64 estaría entre os instantes 63 e 64).
Este valor representará o momento onde comeza a sesión dun paciente fixo nun día e *linac* determinados.
- “appointmentendtime”. Este valor representará o momento onde finaliza a sesión dun paciente fixo nun día e *linac* determinados.

Apéndice B

Código conxunto de ambas partes

```
#PAQUETES
import pandas as pd
import numpy as np
import time
import sys

pathfile=("C:\\Users\\Asus\\PycharmProjects\\pythonProject\\
Radioterapia\\ins01.csv")
#dirección do input
#LECTURA DOS DATOS
def lecturedata(pathdata):
    df = pd.read_csv(pathdata)
    name=df.columns[1] #NOME DA INSTANCIA
    nlinac=df.iloc[0,1] #NÚMERO DE LINACS
    nlinac=int(nlinac)
    ntimslots=df.iloc[1,1] #NÚMERO DE BLOQUES DE TEMPO POR DÍA
    ntimslots=int(ntimslots)
    ndays=df.iloc[3,1] #NÚMERO DE DÍAS COS QUE TRABALLAMOS
    ndays=int(ndays)
    nscope=df.iloc[4,1] #NÚMERO DE DÍAS DE LÍMITE DE EXTENSIÓN
    nscope=int(nscope)
    npat=df.iloc[7,1] #NÚMERO DE PACIENTES
    npat=int(npat)
    datapatients=df.iloc[list(range(9,9+npat)),:].copy()
    #Os datos dos pacientes como tal
    datapatients.columns = df.iloc[8]
    datapatients[['index','noSections','treatmentID','patID',
        'admissionDay','releaseDay','dueDay','duration','TWMin',
        'TWMax']]=datapatients[['index','noSections','treatmentID','patID',
        'admissionDay','releaseDay','dueDay','duration','TWMin',
        'TWMax']].apply(pd.to_numeric)
    print("Datos de entrada:\n")
    print(datapatients.to_string())
    noSectionstotal=sum(list(datapatients['noSections']))
    #Calculase número total de bloques
    que se van ocupar
    print("\n")
    print(f"Número de seccións a asignar: {noSectionstotal}")
    fixedplan=df.iloc[range(9+npat+2,df.shape[0]),:].copy()
```

```

#Estes pacientes non se van a
tramitar, porque son fixos
fixedplan.columns = df.iloc[9+npat+1]
fixedplan[['day','linac','index','appointmenttime',
'appointmentendtime']]=(fixedplan[['day','linac','index',
'appointmenttime','appointmentendtime']].apply(pd.to_numeric))
dict={"name":name,"nlinac":nlinac,"ntimslots":ntimslots,"ndays":ndays,
"nscope":nscope,"npat":npat,"noSectionstotal":noSectionstotal,
"datapatients":datapatients,"fixedplan":fixedplan}
return dict
data=lecturedata(pathfile)

#CARGA DE PYOMO E GUROBI PARA FORMULAR E RESOLVER MODELOS
import pyomo.environ as pyo
solver = "gurobi"
SOLVER = pyo.SolverFactory(solver)
assert SOLVER.available(), f"Solver {solver} non empregable."
SOLVER.options['TimeLimit'] = 3600

#EN CASO DE EMPREGAR HIGHS
#import pyomo.environ as pyo
#solver = "appsi_highs"
#SOLVER = pyo.SolverFactory(solver)
#assert SOLVER.available(), f"Solver {solver} non empregable."
#SOLVER.options['time_limit'] = 3600

#PRIMEIRA PARTE DO MODELO: parámetros a definir polo usuario
omega_overdue=1000
omega_tardiness=1
omega_diflinac=1
gamma_reserves=0

#FUNCIÓN QUE DEFINE A PRIMEIRA PARTE DO MODELO
def
#Precálculos sobre o noso input
radio_asignation(data,omega_overdue,omega_tardiness,
omega_diflinac,gamma_reserves):
fixedpatindex = np.hstack(data['datapatients'].loc[data[
'datapatients'].index[data['datapatients'][
'admissionDay'] <=-1].tolist(),['index']].values)
#conx. pacientes fixos
newpatindex = np.hstack(data['datapatients'].loc[data[
'datapatients'].index[data['datapatients'][
'admissionDay'] >-1].tolist(),['index']].values)
#conx. pacientes novos
#pacientes P3
indpacP3= np.hstack(data['datapatients'].loc[data['datapatients'].index[
data['datapatients']['priority'] =='P3'].tolist(),['index']].values)
#pacientes P3 novos
indP3=list(set(newpatindex).intersection(set(indpacP3)))
#pacientes P4
indpacP4= np.hstack(data['datapatients'].loc[data['datapatients'].index[
data['datapatients']['priority'] =='P4'].tolist(),['index']].values)
#pacientes P4 novos
indP4 = list(set(newpatindex).intersection(set(indpacP4)))

```

```

#Nome do modelo
m = pyo.ConcreteModel("Radio therapy asignation")

#Definición dos conxuntos con Pyomo
m.PAT_NEW = pyo.Set(initialize=newpatindex)
m.LINACS = pyo.Set(initialize=range(data['nlinac']))
m.DAYS = pyo.Set(initialize=range(data['ndays']))

#Definición dos parámetros con Pyomo
#Parámetro omega_1
m.w_1 = pyo.Param(mutable=False, default=omega_overdue,
    domain=pyo.NonNegativeReals)
#Parámetro omega_2
m.w_2 = pyo.Param( mutable=False, default=omega_tardiness,
    domain=pyo.NonNegativeReals)
#Parámetro omega_3
m.w_3 = pyo.Param(mutable=False, default=omega_diflinac,
    domain=pyo.NonNegativeReals)
#Parámetro gamma
m.gamma = pyo.Param(mutable=False, default=gamma_reserves,
    domain=pyo.NonNegativeReals)

#Capacidade total por linac e día
@m.Param(m.DAYS, m.LINACS, domain=pyo.Integers)
def cap_general(m, day, linac):
    return data['ntimslots']

#Capacidade libre por cada linac e día (quitando pacientes fixos)
@m.Param(m.DAYS, m.LINACS, domain=pyo.Integers)
def cap(m, day, linac):
    x=(data['fixedplan']['day'] == day).to_list()
    y =(data['fixedplan']['linac'] == linac).to_list()
    fixedtime=data['fixedplan'].loc[list(map(lambda a, b: a and b, x, y))]
    return data['ntimslots']-sum(fixedtime['appointmentendtime']-
        fixedtime['appointmenttime'])

#Días de lanzamento por paciente novo
@m.Param(m.PAT_NEW, domain=pyo.Integers)
def ready_new(m, pat):
    return data['datapatients'].loc[[9+pat], ['releaseDay']].values.item()

#Días de vencemento por paciente novo
@m.Param(m.PAT_NEW, domain=pyo.Integers)
def due_new(m, pat):
    return data['datapatients'].loc[[9+pat], ['dueDay']].values.item()

#Número de fraccións por paciente novo
@m.Param(m.PAT_NEW, domain=pyo.Integers)
def noSections_new(m, pat):
    return data['datapatients'].loc[[9+pat], ['noSections']].values.item()

#Duración de cada fracción por paciente novo
@m.Param(m.PAT_NEW, domain=pyo.Integers)
def duration_new(m, pat):
    return data['datapatients'].loc[[9+pat], ['duration']].values.item()

```

```

#Definición das variables con Pyomo
m.xikt_new = pyo.Var(m.PAT_NEW,m.LINACS,m.DAYS, domain=pyo.Binary)
m.zit_new = pyo.Var(m.PAT_NEW,m.DAYS, domain=pyo.Binary)
m.wik_new = pyo.Var(m.PAT_NEW,m.LINACS, domain=pyo.Binary)
#Xa van incluídas aquí as restricións 1.12
#ao definir todas as variables implicadas como binarias
#xa estamos aplicando a restricción (1.12)

#Definición da función obxectivo (1.1) con Pyomo
@m.Objective(sense=pyo.minimize)
def minimize_overdue(m):
    return (m.w_1 * sum(((t - m.due_new[i]) ** 2) *
        m.zit_new[i, t] for i in m.PAT_NEW for t in m.DAYS if t>m.due_new[i])+
        m.w_2 * sum(((t-m.ready_new[i])**2)*
        m.zit_new[i,t] for i in m.PAT_NEW for t in m.DAYS)+
        m.w_3 * sum(m.wik_new[i,k] for i in m.PAT_NEW for k in m.LINACS))

#Definición das restricións con Pyomo
#Restrición (1.2)
@m.Constraint(m.PAT_NEW,m.DAYS)
def limit_day_new(m,pat,day):
    return sum(m.xikt_new[pat,k,day] for k in m.LINACS) <= 1

#Restrición (1.3)
@m.Constraint(m.PAT_NEW)
def limit_nosections_new(m,pat):
    return sum(m.xikt_new[pat,k,t] for k in m.LINACS
        for t in m.DAYS)<= m.noSections_new[pat]

#Restrición (1.4)
@m.Constraint(m.PAT_NEW,m.DAYS,m.DAYS)
def consecutive_new(m,pat,day,n):
    if day>0:
        if day < n <= data['ndays'] - 1 and n <= day +
            m.noSections_new[pat] - 1:
            return (sum(m.xikt_new[pat, k, day] - m.xikt_new[pat,
                k, day - 1] for k in m.LINACS) <=
                sum(m.xikt_new[pat,k,n] for k in m.LINACS))
        else:
            return pyo.Constraint.Skip
    else:
        if day < n <= data['ndays'] - 1 and n <= day +
            m.noSections_new[pat] - 1:
            return (sum(m.xikt_new[pat, k, day] for k in m.LINACS) <=
                sum(m.xikt_new[pat,k,n] for k in m.LINACS))
        else:
            return pyo.Constraint.Skip

#Restrición (1.5)
@m.Constraint(m.PAT_NEW,m.LINACS)
def before_new(m,pat,linac):
    if (m.ready_new[pat]>0):
        return sum(m.xikt_new[pat, linac, t] for t in
            m.DAYS if t<m.ready_new[pat])<=0
    else:
        return pyo.Constraint.Skip

```

```

#Restrición (1.6)
@m.Constraint(m.DAYS,m.LINACS)
def limit_capacity_linac_day(m,day,linac):
    return sum(m.duration_new[j] * m.xikt_new[j, linac, day]
               for j in m.PAT_NEW)<=m.cap[day,linac]

#Restrición (1.7)
@m.Constraint(m.DAYS,m.LINACS)
def reserve_capacity(m,day,linac):
    return (sum(m.duration_new[j]*m.xikt_new[j, linac, day] for j in
               indP3)+ sum(m.duration_new[j]*m.xikt_new[j, linac, day] for j in
               indP4)<=m.cap[day,linac]-m.gamma*m.cap_general[day,linac])

#Restrición (1.8)
@m.Constraint(m.PAT_NEW,m.DAYS)
def no_start_forces_new(m,pat,day):
    if (day>0):
        return sum(m.xikt_new[pat, k, day]- m.xikt_new[pat, k, day-1]
                   for k in m.LINACS)<=m.zit_new[pat,day]
    else:
        return sum(m.xikt_new[pat, k, day] for k in
                   m.LINACS)<=m.zit_new[pat,day]

#Restrición (1.9)
@m.Constraint(m.PAT_NEW,m.DAYS)
def start_forces_new(m,pat,day):
    return sum(m.xikt_new[pat, k, day]for k in
               m.LINACS)>=m.zit_new[pat,day]

#Restrición (1.10)
@m.Constraint(m.PAT_NEW)
def limit_starts_new(m,pat):
    return sum(m.zit_new[pat,t] for t in m.DAYS)== 1

#Restrición (1.11)
@m.Constraint(m.PAT_NEW,m.DAYS,m.LINACS)
def linac_used_new(m,pat,day,linac):
    return m.wik_new[pat,linac]>=m.xikt_new[pat, linac, day]

#Restrición (1.12) aplicada na def. das variables
return m

#Cálculo do tempo de execución
starttime1=time.time()
m = radio_asignation(data,omega_overdue,omega_tardiness,
                    omega_diflinac,gamma_reserves)
endtimeformula1 = time.time()
res=SOLVER.solve(m,tee=True)
endtimesolving1 = time.time()

#Valor da función obxectivo
print('Valor da función obxectivo no óptimo:\n', pyo.value(
    m.minimize_overdue))

#Índices dos pacientes novos

```

```

newpatindex = np.hstack(data['datapatients'].loc[
    data['datapatients'].index[data['datapatients']
    'admissionDay'] > -1].tolist(),['index']].values)

#Diccionario para organizar a planificación
#que se obtén como saída da primeira fase
#Serán que pacientes van cada día a cada linac
dict_planings={}
for k in m.LINACS:
    daily_planning = pd.DataFrame(index=newpatindex)
    for pat in m.PAT_NEW:
        for t in m.DAYS:
            if round(pyomo.value(m.xikt_new[pat,k,t]),1)==1:
                daily_planning.loc[pat, t] = m.duration_new[pat]
            else:
                daily_planning.loc[pat, t] = 0
    dict_planings[k]=daily_planning

#Convertir a csv os diccionarios de solución
for i in range(len(dict_planings)):
    dict_planings[i].to_csv(data['name']+'linac' + str(i)+'.csv', sep='\t')

#Imprimir por pantalla os diccionarios obtidos
for i in range(len(dict_planings)):
    print("Emprego do linac", i ,
        "respecto de paciente e día, indicando número de blocks empregados")
    print(dict_planings[i].to_string())

#Función para obter táboa con resultados emulando Pham
def tableresults(data,m,res,endtimeformula1,endtimesolving1):
    K = data['nlinac'] #número linacs
    P = data['npat'] #número pacientes totais
    #Pacientes novos total
    x=(data['datapatients']['admissionDay'] > -1).to_list()
    hat_P = sum(x)
    #Pacientes novos P1
    y1=(data['datapatients']['priority'] == 'P1').to_list()
    t1=list(map(lambda a, b: a and b, x, y1))
    hat_P1 = sum(t1)
    #Pacientes novos P2
    y2=(data['datapatients']['priority'] == 'P2').to_list()
    t2=list(map(lambda a, b: a and b, x, y2))
    hat_P2 = sum(t2)
    #Pacientes novos P3
    y3=(data['datapatients']['priority'] == 'P3').to_list()
    t3=list(map(lambda a, b: a and b, x, y3))
    hat_P3 = sum(t3)
    #Pacientes novos P4
    y4=(data['datapatients']['priority'] == 'P4').to_list()
    t4=list(map(lambda a, b: a and b, x, y4))
    hat_P4 = sum(t4)
    #Tempo de execución de modelo
    runtime=round(endtimesolving1-endtimeformula1,4)
    #Gap de resolución
    gap=(res.problem[0]['Upper bound']- res.problem[0][

```

```

'Lower bound')]/res.problem[0]['Upper bound']
#Tempo promedio de atención dos ^P1
averwait1=np.mean([[int(round(x)) for x in pyo.value(
    m.zit_new[k, :])] .index(1) for k in data['datapatients']['index'][t1]])
#Tempo promedio de atraso dos ^P1
prov1=np.subtract([[int(round(x)) for x in pyo.value(
    m.zit_new[k, :])] .index(1) for k in data['datapatients']['index'][t1]],
    [m.due_new[k] for k in data['datapatients']['index'][t1]])
averover1=np.mean([0 if i < 0 else i for i in prov1])
#Porcentaxe de atrasos en pacientes ^P1
perover1=np.mean(prov1>0)*100
#Tempo promedio de atención dos ^P2
averwait2=np.mean([[int(round(x)) for x in pyo.value(
    m.zit_new[k, :])] .index(1) for k in data['datapatients']['index'][t2]])
#Tempo promedio de atraso dos ^P2
prov2=np.subtract([[int(round(x)) for x in pyo.value(
    m.zit_new[k, :])] .index(1) for k in data['datapatients']['index'][t2]],
    [m.due_new[k] for k in data['datapatients']['index'][t2]])
averover2=np.mean([0 if i < 0 else i for i in prov2])
#Porcentaxe de atrasos en pacientes ^P2
perover2=np.mean(prov2>0)*100
#Tempo promedio de atención dos ^P3
averwait3=np.mean([[int(round(x)) for x in pyo.value(
    m.zit_new[k, :])] .index(1) for k in data['datapatients']['index'][t3]])
#Tempo promedio de atraso dos ^P3
prov3=np.subtract([[int(round(x)) for x in pyo.value(
    m.zit_new[k, :])] .index(1) for k in data['datapatients']['index'][t3]],
    [m.due_new[k] for k in data['datapatients']['index'][t3]])
averover3=np.mean([0 if i < 0 else i for i in prov3])
#Porcentaxe de atrasos en pacientes ^P3
perover3=np.mean(prov3>0)*100
#Tempo promedio de atención dos ^P4
averwait4=np.mean([[int(round(x)) for x in pyo.value(
    m.zit_new[k, :])] .index(1) for k in data['datapatients']['index'][t4]])
#Tempo promedio de atraso dos ^P4
prov4=np.subtract([[int(round(x)) for x in pyo.value(
    m.zit_new[k, :])] .index(1) for k in data['datapatients']['index'][t4]],
    [m.due_new[k] for k in data['datapatients']['index'][t4]])
averover4=np.mean([0 if i < 0 else i for i in prov4])
#Porcentaxe de atrasos en pacientes ^P4
perover4=np.mean(prov4>0)*100
#Tempo promedio de atención dos ^P
averwait=np.mean([[int(round(x)) for x in pyo.value(
    m.zit_new[k, :])] .index(1) for k in data['datapatients']['index'][x]])
#Tempo promedio de atraso dos ^P
prov=np.subtract([[int(round(x)) for x in pyo.value(
    m.zit_new[k, :])] .index(1) for k in data['datapatients']['index'][x]],
    [m.due_new[k] for k in data['datapatients']['index'][x]])
averover=np.mean([0 if i < 0 else i for i in prov])
#Porcentaxe de atrasos en pacientes ^P
perover=np.mean(prov>0)*100
#Promedio entre os pacientes novos de linacs empregados
averlinac=np.mean([sum([x for x in pyo.value(m.wik_new[k,
    :])) for k in data['datapatients']['index'][x]])
#Índices do dicionario
temp_dict={'K':K, 'P':P, '^P':hat_P, '^P_1':hat_P1, '^P_2':hat_P2,

```

```

    '^P_3':hat_P3,'^P_4':hat_P4,'runtime':runtime,'gap':gap,
    'await:^P_1':await1,'await:^P_2':await2,
    'await:^P_3':await3,'await:^P_4':await4,
    'await:overall':await,'over:^P_1':over1,
    'over:^P_1':over1,'over:^P_2':over2,
    'over:^P_2':over2,'over:^P_3':over3,
    'over:^P_3':over3,'over:^P_4':over4,
    'over:^P_4':over4,'over:overall':over,
    'over:overall':over,'linac:overall':linac}
complete_results = pd.DataFrame(data=temp_dict,index=[data['name']])
print(complete_results.to_string())
return complete_results

#Impresión da táboa por pantalla co resumo
print("Táboa resultados fase 1:\n")
complete_results=tableresults(data,m,res,endtimeformula1,endtimesolving1)

#SEGUNDA PARTE DO MODELO: parámetros a definir polo usuario
omegachanges=60
omegatimewindow=1
omegadeviation=1
#FUNCIÓN QUE DEFINE A SEGUNDA PARTE DO MODELO
def schedule(dict_planings,data,omega4,omega5,omega6):
    #Precálculos para os conxuntos
    #Índices dos pacientes fixos
    fixedpatindex = np.hstack(
        data['datapatients'].loc[data['datapatients'].index[
            data['datapatients']['admissionDay'] <=
            -1].tolist(),['index']].values)
    #Índices dos pacientes novos
    newpatindex = np.hstack(
        data['datapatients'].loc[data['datapatients'].index[data[
            'datapatients']['admissionDay'] > -1].tolist(),['index']].values)
    #Pacientes P3
    indpacP3 = np.hstack(
        data['datapatients'].loc[data['datapatients'].index[data[
            'datapatients']['priority'] == 'P3'].tolist(),['index']].values)
    #Pacientes ^P3
    indP3 = list(set(newpatindex).intersection(set(indpacP3)))
    #Pacientes P4
    indpacP4 = np.hstack(
        data['datapatients'].loc[data['datapatients'].index[data[
            'datapatients']['priority'] == 'P4'].tolist(),['index']].values)
    #Pacientes ^P4
    indP4 = list(set(newpatindex).intersection(set(indpacP4)))
    #Pacientes curativos novos
    curative_new = indP3+indP4
    curative_new.sort()
    #Para cada día e linac, indicamos os pacientes atendidos
    #Combina os resultados da primeira parte do modelo
    #coas citas xa asignadas para os pacientes fixos
    plan_diario_linac={(t,k):[] for t in range(data[
        'ndays']) for k in range(data['nlinac']) }
    for i in range(data['npat']):
        if i in fixedpatindex:
            lista=data['fixedplan'][(data['fixedplan']][

```

```

    'day']<data['ndays']]) & (data['fixedplan']['index']==i)]
for t in lista['day']:
    for k in lista['linac']:
        plan_diario_linac[(t,k)]=plan_diario_linac[(t,k)]+[i]
        break
else:
    tempdays=[]
    templinac=[]
    for k in range(data['nlinac']):
        temp=(dict_planings[k].loc[[i]] > 0).values[0].tolist()
        if any(temp):
            tempdays.extend( [a for a in range(
                len(temp)) if temp[a]==True])
            templinac.extend([k for i in range(len(tempdays))])
    for t in tempdays:
        for k in templinac:
            plan_diario_linac[(t,k)]=plan_diario_linac[(t,k)] + [i]
            break

#Nome do modelo
m = pyo.ConcreteModel("Radio therapy schedule")

#Definición dos conxuntos con Pyomo
m.PAT= pyo.Set(initialize=range(data['npat']))
m.LINACS = pyo.Set(initialize=range(data['nlinac']))
m.DAYS = pyo.Set(initialize=range(data['ndays']))
#Conxunto de bloques
m.SLOTS= pyo.Set(initialize=range(data['ntimslots']))
#Pacientes fixos
m.PAT_FIXED =[p1 for p1 in m.PAT if p1 in fixedpatindex]
#Pacientes curativos novos
m.PAT_CURATIVE_NEW = [p1 for p1 in m.PAT if p1 in curative_new]
#Cada elemento do seguinte conxunto é un subconxunto de pacientes.
#En concreto, terá os índices de día-linac, onde
#cada subconxunto serán os pacientes tratados nese índice.
#É equivalente a  $P^t_l$  no modelo visto no Capítulo 2
m.PTK= plan_diario_linac

#Conxuntos necesarios para o funcionamento
#do código: por mor da cantidade de
#restricións que se xeran, necesitamos
#preinicializar, sobre todo para as restricións
#(2.4) e (2.5) os conxuntos nos que traballa
#en vez de ir iterando a totalidade dos conxuntos
#de pacientes, días, linacs, bloques (x2 no caso de (2.4))...

#Este conxunto está definido por subconxuntos de tres elementos
#{paciente atendido, día, linac}
#por cada vez que apareza en m.PTK
m.PTKP1= [(p1,t,k) for (t,k) in m.PTK for p1 in m.PTK[t,k]]

#Definición dos parámetros con Pyomo
#Só definimos a capacidade total
#Os demais os introduciremos ao rematar cos conxuntos
#Pero necesitamos a capacidade total

```

```

#Capacidade total de cada linac
@m.Param(m.DAYS, m.LINACS, domain=pyo.Integers)
def cap_general(m, day, linac):
    return data['ntimslots']

#Crea un conxunto de subconxuntos de 5 elementos
#Cada un é, para cada paciente atendido nun día e linac
#escolle outro atendido nese día e linac distinto
#e o último valor é a duración de cada fracción
#en bloques para o primeiro dos pacientes mencionados
m.PTKP1P2= [(p1,p2,t,k,list(data['datapatients']['duration'])[p1])
    for (t,k) in m.PTK for p1 in
        m.PTK[t,k] for p2 in m.PTK[t,k] if p2!=p1]

#Este conxunto vai definir todos posibles índices que hai para (2.4)
#Sobre m.PTKP1P2, vamos a tomar todos os posibles s1 e s2
#coas condicións que había en (2.4)
m.REST4= [(p1, p2, s1, s2, t, k) for (p1, p2, t, k, dur1) in
    m.PTKP1P2 for s1 in m.SLOTS if s1 <= m.cap_general[t,k] -
    dur1 for s2 in m.SLOTS if s1 < s2 < s1 + dur1]

#Este conxunto define todos os posibles índices que hai para (2.5)
#Colle para cada día e linac un paciente da súa lista
#e só considera os bloques nas condicións da
#restrición (2.5)
m.REST5=[(p1,t,k,s) for (p1,t,k) in m.PTKP1 for s in m.SLOTS
    if m.cap_general[t,k]-list(data[
    'datapatients']['duration'])[p1]+2<=s<m.cap_general[t,k]]

#Definición dos parámetros con Pyomo
#(salvo capacidade total, que xa a definimos
#por necesidade antes)
#Parámetro omega_4
m.w_4= pyo.Param(mutable=False,default=omega4,
    domain=pyo.NonNegativeReals)
#Parámetro omega_5
m.w_5 = pyo.Param( mutable=False,default=omega5,
    domain=pyo.NonNegativeReals)
#Parámetro omega_6
m.w_6 = pyo.Param(mutable=False,default=omega6,
    domain=pyo.NonNegativeReals)

#Límite inferior para todos os pacientes da ventá
#de preferencia temporal
@m.Param(m.PAT, domain=pyo.Integers)
def slot_appointment_min(m, pat):
    return data['datapatients']['TWMin'].iloc[pat]

#Límite inferior para todos os pacientes da ventá
#de preferencia temporal
@m.Param(m.PAT, domain=pyo.Integers)
def slot_appointment_max(m, pat):
    return data['datapatients']['TWMax'].iloc[pat]

#Definición das variables con Pyomo
#Xunto coa restrición (2.12) de binariedade

```

```

#nas variables  $y_{\{pb\}}^{\{tl\}}$ .
#Variables  $y_{\{pb\}}^{\{tl\}}$ 
m.y_istk = pyo.Var(m.PAT,m.SLOTS,m.DAYS,m.LINACS,
    domain=pyo.Binary)
#Variables  $\phi_{\{p\}}^{\{t\}}$ 
m.phi_it_upper= pyo.Var(m.PAT_FIXED,m.DAYS,
    domain=pyo.NonNegativeIntegers)
m.phi_it_lower = pyo.Var(m.PAT_FIXED, m.DAYS,
    domain=pyo.NonNegativeIntegers)
#Variables  $\delta_{\{p\}}^{\{t\}}$ 
m.delta_it_upper= pyo.Var(m.PAT_CURATIVE_NEW,m.DAYS,
    domain=pyo.NonNegativeIntegers)
m.delta_it_lower= pyo.Var(m.PAT_CURATIVE_NEW,m.DAYS,
    domain=pyo.NonNegativeIntegers)
#Variables  $\theta_{\{p\}}$ 
m.theta_i_upper=pyo.Var(m.PAT_CURATIVE_NEW,
    domain=pyo.NonNegativeIntegers)
m.theta_i_lower= pyo.Var(m.PAT_CURATIVE_NEW,
    domain=pyo.NonNegativeIntegers)

#Definición da función obxectivo (2.1) con Pyomo
@m.Objective(sense=pyo.minimize)
def minimize_incompatibility(m):
    return (m.w_4 * sum(m.phi_it_upper[i,t]+m.phi_it_lower[i,t]
        for i in m.PAT_FIXED for t in m.DAYS)+
        m.w_5 * sum(m.delta_it_upper[i,t]+m.delta_it_lower[i,t]
        for i in m.PAT_CURATIVE_NEW for t in m.DAYS)+
        m.w_6 * sum(m.theta_i_upper[i]-m.theta_i_lower[i]
        for i in m.PAT_CURATIVE_NEW))

#Restrición (2.2)
@m.Constraint(m.PTKP1)
def force_appointment(m,pat,day,linac):
    return sum(m.y_istk[pat,slot,day,linac] for slot in m.SLOTS)==1

#Restrición (2.3)
@m.Constraint(m.SLOTS,m.DAYS,m.LINACS)
def limit_slot(m,slot,day,linac):
    suma=sum(m.y_istk[pat,slot,day,linac]
        for pat in m.PTK[day,linac])
    if isinstance(suma,int):
        return pyo.Constraint.Skip
    else:
        return sum(m.y_istk[pat,slot,day,linac]
            for pat in m.PTK[day,linac])<=1

#Restrición (2.4)
#Ao traballar cos índices de m.REST4, queda moito máis concentrada
#Aínda así, como referencia, para a ins01
#Chegamos a obter 2 millóns de restricións
@m.Constraint(m.REST4)
def reserve_consecutive_slots(m, p1,p2,s1,s2,a,k):
    return m.y_istk[p1,s1,a,k]<=1-m.y_istk[p2,s2,a,k]

#Restrición (2.5)
#Ao traballar cos índices de m.REST5, queda moito máis concentrada

```

```

@m.Constraint(m.REST45)
def no_chance_to_start(m,pat,day,linac,slot):
    return m.y_istk[pat,slot,day,linac]==0

#Restrición (2.6)
@m.Constraint(m.PTKP1)
def linking_upperbound_phi(m,pat,day,linac):
    if pat in m.PAT_FIXED:
        s_bar_it=data['fixedplan'].loc[(data['fixedplan'][['
            'day', 'linac', 'index']]==[day, linac, pat]).all(
            axis=1)]['appointmenttime'].item()
        return m.phi_it_upper[pat,day] >=s_bar_it-
            sum(s*m.y_istk[pat,s,day,linac] for s in m.SLOTS)
    else:
        return pyo.Constraint.Skip

#Restrición (2.7)
@m.Constraint(m.PTKP1)
def linking_lowerbound_phi(m,pat,day,linac):
    if pat in m.PAT_FIXED:
        s_bar_it=data['fixedplan'].loc[(data['fixedplan'][['
            'day', 'linac', 'index']]==[day, linac, pat]).all(
            axis=1)]['appointmenttime'].item()
        return m.phi_it_lower[pat,day] >=sum(s*m.y_istk[pat,s,day,
            linac] for s in m.SLOTS)-s_bar_it
    else:
        return pyo.Constraint.Skip

#Restrición (2.8)
@m.Constraint(m.PTKP1,m.SLOTS)
def linking_upperbound_delta(m,pat,day,linac,slot):
    if pat in m.PAT_CURATIVE_NEW:
        return m.delta_it_upper[pat,day]>=m.y_istk[pat,slot,day,linac]*
            (m.slot_appointment_min[pat]-slot)
    else:
        return pyo.Constraint.Skip

#Restrición (2.9)
@m.Constraint(m.PTKP1,m.SLOTS)
def linking_lowerbound_delta(m,pat,day,linac,slot):
    if pat in m.PAT_CURATIVE_NEW:
        return m.delta_it_lower[pat,day]>=m.y_istk[pat,slot,day,linac]*
            (slot-m.slot_appointment_max[pat])
    else:
        return pyo.Constraint.Skip

#Restrición (2.10)
@m.Constraint(m.PAT_CURATIVE_NEW,m.DAYS)
def linking_upperbound_theta(m,pat,day):
    for linac in m.LINACS:
        if pat in m.PTK[day,linac]:
            return m.theta_i_upper[pat]>=sum(s*m.y_istk[pat,s,day,linac]
                for s in m.SLOTS)
        else:
            return pyo.Constraint.Skip

```

```

#Restricción (2.11)
@m.Constraint(m.PAT_CURATIVE_NEW,m.DAYS)
def linking_lowerbound_theta(m,pat,day):
    for linac in m.LINACS:
        if pat in m.PTK[day,linac]:
            return m.theta_i_lower[pat]<=sum(s*m.y_istk[pat,s,day,linac]
            for s in m.SLOTS)
        else:
            return pyo.Constraint.Skip

#Xa aplicamos a restricción (2.12) ao definir as variables

#Restricións (2.13)
@m.Constraint(m.PAT_FIXED,m.DAYS) #restricción de cota
def upperbound_phi_upper(m,pat,day):
    return m.phi_it_upper[pat,day]<= max(
        list(m.cap_general[day,linac] for linac in m.LINACS) )

@m.Constraint(m.PAT_FIXED,m.DAYS) #restricción de cota
def upperbound_phi_lower(m,pat,day):
    return m.phi_it_lower[pat,day]<= max(
        list(m.cap_general[day,linac] for linac in m.LINACS) )

#Restricións (2.14)
@m.Constraint(m.PAT_CURATIVE_NEW,m.DAYS) #restricción de cota
def upperbound_delta_upper(m,pat,day):
    return m.delta_it_upper[pat,day]<= max(
        list(m.cap_general[day,linac] for linac in m.LINACS) )

@m.Constraint(m.PAT_CURATIVE_NEW,m.DAYS) #restricción de cota
def upperbound_delta_lower(m,pat,day):
    return m.delta_it_lower[pat,day]<= max(
        list(m.cap_general[day,linac] for linac in m.LINACS) )

@m.Constraint(m.PAT_CURATIVE_NEW) #restricción de cota
def upperbound_theta_upper(m,pat):
    return m.theta_i_upper[pat]<= max(
        list(m.cap_general[day,linac] for day in
        m.DAYS for linac in m.LINACS))

@m.Constraint(m.PAT_CURATIVE_NEW) #restricción de cota
def upperbound_theta_lower(m,pat):
    return m.theta_i_lower[pat]<= max(
        list(m.cap_general[day,linac] for day in
        m.DAYS for linac in m.LINACS))

return m

#Ejecución e cálculo tempo execución
starttime2=time.time()
m = schedule(dict_planings,data,
    omegachanges,omegatimewindow,omegadeviation)
endtimeformula2 = time.time()
#Limitamos o tempo de execución a 6000 segundos
#é dicir, 1 hora e 40 minutos (100 min)
res=SOLVER.solve(m,tee=True,timelimit=6000)

```

```

endtimesolving2 = time.time()

#Táboa resumo con resultados
def tableresults2(data,m,res,endtimeformula,endtimesolving):
    #Número de aceleradores lineais
    K = data['nlinac']
    #Número de pacientes a atender
    P = data['npat']
    #Número de pacientes novos a atender
    x=(data['datapatients']['admissionDay'] > -1).to_list()
    hat_P = sum(x)
    #Valor da función obxectivo
    obj=pyo.value(m.minimize_incompatibility)
    runtime=endtimesolving-endtimeformula
    #Límite inferior (maior valor da función obxectivo)
    #atopado polo solver
    bound=res.problem[0]['Lower bound']
    #Gap sobre o límite superior e inferior
    do valor óptimo da función obxectivo
    gap=(res.problem[0]['Upper bound'] -
        bound)/res.problem[0]['Upper bound']
    #Número de días a planificar
    ndays=data['ndays']
    #Promedio por día sobre os pacientes novos
    #de violación da ventá de preferencia de atención
    avgTWviolation=(sum(pyo.value(m.delta_it_lower[:, :])) +
        sum(pyo.value(m.delta_it_upper[:, :])))/ hat_P /ndays
    #Promedio sobre os pacientes novos
    #de rango de bloques de atención entre todas as súas citas
    avgdeviation = (sum(pyo.value(m.theta_i_upper[:]))+
        sum(pyo.value(m.theta_i_lower[:])))/ hat_P
    #Promedio sobre os pacientes fixos
    #de bloques de cambio sobre as súas
    #citas orixinais
    avgchanges = (sum(pyo.value(m.phi_it_upper[:, :]))-
        sum(pyo.value(m.phi_it_lower[:, :])))/ (P-hat_P)
    temp_dict={'K':K,'P':P,'^P':hat_P,'obj':obj, 'runtime': runtime,
        'gap':gap,'bound':bound,'avgTWviolation':avgTWviolation,
        'avgdeviation':avgdeviation,'avgchanges':avgchanges}
    complete_results = pd.DataFrame(data=temp_dict,index=[data['name']])
    print(complete_results.to_string())
    return complete_results

#Imprimir táboa resumo
print("Táboa resultados fase 2:\n")
schedule_results=tableresults2(data,m,res,endtimeformula2,endtimesolving2)

```

Apéndice C

Gráficas da primeira parte

```
#Paquetes
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D

#Lectura de datos (misma que no código completo)
pathfile=("C:\\Users\\Asus\\PycharmProjects\\
pythonProject\\Radioterapia\\ins01.csv")
#dirección do arquivo con \\
def lecturedata(pathdata):
    df = pd.read_csv(pathdata)
    name=df.columns[1]
    nlinac=df.iloc[0,1]
    nlinac=int(nlinac)
    ntimslots=df.iloc[1,1]
    ntimslots=int(ntimslots)
    ndays=df.iloc[3,1]
    ndays=int(ndays)
    nscope=df.iloc[4,1]
    nscope=int(nscope)
    npat=df.iloc[7,1]
    npat=int(npat)
    datapatients=df.iloc[list(range(9,9+npat)),:].copy()
    datapatients.columns = df.iloc[8]
    datapatients[['index','noSections','treatmentID','patID',
'admissionDay','releaseDay','dueDay','duration','TWMin',
'TWMax']]=datapatients[['index','noSections','treatmentID','patID',
'admissionDay','releaseDay','dueDay','duration','TWMin',
'TWMax']].apply(pd.to_numeric)
    noSectionstotal = sum(list(datapatients['noSections']))
    fixedplan=df.iloc[range(9+npat+2,df.shape[0]),:].copy()
    fixedplan.columns = df.iloc[9+npat+1]
    fixedplan[['day','linac','index','appointmenttime',
'appointmentendtime']]=(fixedplan[['day','linac','index',
'appointmenttime','appointmentendtime']].apply(pd.to_numeric))
    dict={"name":name,"nlinac":nlinac,"ntimslots":ntimslots,
"ndays":ndays,"nscope":nscope,"npat":npat,
"noSectionstotal":noSectionstotal,"datapatients":datapatients,
"fixedplan":fixedplan}
```

```

    return dict
data=lecturedata(pathfile)

#Busca da solução obtida da primeira fase no código completo
dict_planings={}
for i in range(data["nlinac"]):
    file=data['name']+'linac'+str(i)+'.csv'
    dict_planings[i]= pd.read_csv(file,sep="\t",index_col=0)

#Precálculo do número de "cadeas" de atenção, é dizer,
#as citas en días continuados que foi atendido polo mesmo linac
def streaks_list(examplelist):
    n=len(examplelist)
    s=False
    streaks_start_end=[]
    if n>1:
        if examplelist[0]!=0:
            streaks_start_end.append(0)
            s=True
        ant=examplelist[0]
        for i in range(1,n-1):
            new=examplelist[i]
            if new!=ant:
                if s==True:
                    streaks_start_end.append(i-1)
                    s=False
                else:
                    streaks_start_end.append(i)
                    s=True
            ant=new
        if examplelist[n-1]!=0:
            if s==True:
                streaks_start_end.append(n-1)
            else:
                streaks_start_end.append(n-1)
                streaks_start_end.append(n-1)
        else:
            if s==True:
                streaks_start_end.append(n-2)
    else:
        if n==1 and examplelist[0]!=0:
            streaks_start_end=[0,0]
    return streaks_start_end

#VISUALIZATION VIA MATPLOTLIB
def gantt(data,dict_planings):
    w = 0.2 # Anchura barra
    n=len(dict_planings)
    plt.rcParams.update({"font.size": 11})
    fig, ax = plt.subplots(1, n, sharex= True ,squeeze=False)
    for linac in range(data['nlinac']):
        k=0
        yticks=[]
        for pat in dict_planings[0].index:
            r = data['datapatients'].iloc[pat,7]
            d = data['datapatients'].iloc[pat,8]

```

```

streaks = streaks_list(dict_planings[linac].loc[pat, :])
if(streaks!=[]):
    k=k+1
    yticks.append(str(pat))
    nstreaks= int(len(streaks)/2)
    # Mostrar cada cadea de atención
    ax[0,linac].fill_between([r, d], [-k - w, -k - w],
        [-k + w, -k + w], lw=1, color="grey", alpha=0.4)
    ax[0,linac].plot([r, r, d, d, r], [-k - w, -k + w,
        -k + w, -k - w, -k - w],lw=0.3, color="grey")
    for j in range(nstreaks):
        ax[0, linac].fill_between([streaks[2 * j], streaks[2 * j + 1]],
            [-k - w, -k - w],[-k + w, -k + w], color='green',alpha=0.6)
        ax[0, linac].plot([streaks[2 * j], streaks[2 * j],
            streaks[2 * j + 1],streaks[2 * j + 1], streaks[2 * j]],
            [-k - w, -k + w, -k + w, -k - w, -k - w],
            lw=0.5, color="green")
        if (streaks[2*j]==streaks[2*j+1]):
            ax[0,linac].text(streaks[2*j], -k+0.35, str(streaks[2*j]),
                color="black",weight="bold", size=9,
                ha="center", va="center")
        else:
            ax[0,linac].text(streaks[2*j]-0.25, -k+0.2,
                str(streaks[2*j]),color="black",
                weight="bold", size=9, ha="right", va="center")
            ax[0,linac].text(streaks[2*j+1]+0.25,
                -k-0.2, str(streaks[2*j+1]),color="black",
                weight="bold", size=9,ha="left", va="center")
    ax[0,linac].set_ylabel("Índice paciente")
    ax[0,linac].set_xlabel("Días")
    ax[0,linac].set_title('Linac '+str(linac),size=15,weight="demibold")
    ax[0,linac].set_ylim(-len(yticks)-1,0)
    ax[0,linac].set_yticks(range(-1,-len(yticks)-1,-1),yticks,size=10)
    custom_lines = [
        Line2D([0], [0], c="grey", lw=10, alpha=0.4),
        Line2D([0], [0], c="green", lw=10, alpha=0.6),
    ]
    fig.legend(custom_lines, ["Días posibles de comezo",
        "Asignación días paciente novo"], loc='upper right',fontsize=11)
    fig.suptitle('Asignación de citas óptima de cada paciente por día',
        fontsize=14, weight="bold")
    plt.tight_layout()
    plt.show()

gantt(data,dict_planings)

```


Bibliografía

- Abbas, A., Saleh, I., Wong, P., Larouche, J., Abouali, J., Park, S., Chan, T. C., Sarhangian, V., e Toor, J. (2026). Optimizing daily surgical scheduling improves operative time consumption: A retrospective study. *Arthroplasty Today*, p. 101926.
- Ahmadi-Javid, A., Jalali, Z., e Klassen, K. J. (2017). Outpatient appointment systems in healthcare: A review of optimization studies. *European Journal of Operational Research*, 258(1):3–34.
- Aringhieri, R., Duma, D., e Squillace, G. (2025). Online algorithms with foresight for radiotherapy patient scheduling. *Computers and Operations Research*, 182:107132.
- Bailey, N. T. J. (1952). A study of queues and appointment systems in hospital out-patient departments, with special reference to waiting-times. *Journal of the Royal Statistical Society. Series B (Methodological)*, 14(2):185–199.
- Cañaveras León, E., Cano Fuentes, G., Dastis Bendala, C., Terrón Dastis, P., e Almeida González, C. V. (2022). Demora en el diagnóstico y tratamiento de 5 tipos de cáncer en 2 centros de salud urbanos. *Atención Primaria*, 54(3).
- Castro, E. e Petrovic, S. (2012). Combined mathematical programming and heuristics for a radiotherapy pre-treatment scheduling problem. *Journal of Scheduling*, 15(3):333–346.
- Cayirli, T. e Veral, E. (2003). Outpatient scheduling in health care: A review of literature. *Production and Operations Management*, 12(4):519–549.
- Conforti, D., Guerriero, F., e Guido, R. (2008). Optimization models for radiotherapy patient scheduling. *4OR*, 6:263–278.
- Conforti, D., Guerriero, F., e Guido, R. (2010). Non-block scheduling with priority for radiotherapy treatments. *European Journal of Operational Research*, 201(1):289–296.
- Conforti, D., Guerriero, F., Guido, R., e Veltri, M. (2011). An optimal decision-making approach for the management of radiotherapy patients. *OR Spectr.*, 33(1):123–148.
- Cox, T. F., Birchall, J. P., e Wong, H. (1985). Optimising the queuing system for an ear, nose and throat outpatient clinic. *Journal of Applied Statistics*, 12(2):113–126.
- Ehrgott, M. (2005). *The Weighted Sum Method and Related Topics*, pp. 65—95. Springer, Berlin, Heidelberg, Heidelberg.
- Fernández, B., Rodríguez, A., Grande, I., Fernández, A., Contreras, J., De Haro, D., Castañeda, D., González de Aledo, A., e Maye, A. (2020). *Acceso al tratamiento de radioterapia en España*. Observatorio del Cáncer de la AECC, Madrid.
- Forrest, J. e Lougee-Heimer, R. (2005). Cbc user’s guide. <https://coin-or.github.io/Cbc/>. Acceso en: 03-06-2026.

- Fredes, M., Dávila, S., e Quezada, F. (2024). Online multi-appointment scheduling under demand for oncology patient treatment. En *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 1310–1316.
- Fries, B. E. e Marathe, V. P. (1981). Determination of optimal variable-sized multiple-block appointment systems. *Operations Research*, 29(2):324–345.
- Froehle, C. M., Magazine, M. J., e Denton, B. T. (2013). Improving scheduling and flow in complex outpatient clinics. En *Handbook of Healthcare Operations Management*, International Series in Operations Research and Management Science, pp. 411–413. Springer New York, New York, NY.
- Gocgun, Y. e Puterman, M. L. (2014). Dynamic scheduling with due dates and time windows: an application to chemotherapy patient appointment booking. *Health Care Management Science*, 17(1):60–76.
- Gurobi Optimization, LLC (2026). Gurobi optimizer reference manual. <https://docs.gurobi.com/projects/optimizer/en/12.0/>. Acceso en: 03-06-2026.
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., e Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.
- Hart, W. E., Watson, J.-P., e Woodruff, D. L. (2011). Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation*, 3(3):219–260.
- Hashemi, S. E., Jabbari, M., e Yaghoubi, P. (2022). A mathematical optimization model for location emergency medical service (ems) centers using contour lines. *Healthcare Analytics*, 2:100026.
- HiGHS Development Team (2025). Highs - high performance optimization software. <https://ergo-code.github.io/HiGHS/v1.10/#>. Acceso en: 03-06-2026.
- Ho, C.-J. e Lau, H.-S. (1992). Minimizing total cost in scheduling outpatient appointments. *Management Science*, 38(12):1750–1764.
- Huangfu, Q. e Hall, J. A. J. (2018). Parallelizing the dual revised simplex method. *Mathematical Programming Computation*, 10(1):119–142.
- Hulshof, P. J. H., Kortbeek, N., Boucherie, R. J., Hans, E. W., e Bakker, P. J. M. (2012). Taxonomic classification of planning decisions in health care: a structured review of the state of the art in or/ms. *Health Systems*, 1(2):129–175.
- Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.
- IBM Corporation (2022). User’s manual for cplex. <https://www.ibm.com/docs/en/icos/22.1.0?topic=optimizers-users-manual-cplex>. Acceso en: 03-06-2026.
- INE (2024). Estadística de defunciones según la causa de muerte. Datos definitivos publicados el 17 de diciembre de 2025 polo Instituto Nacional de Estadística.
- Jacquemin, Y., Marcon, E., e Pommier, P. (2010). Towards an improved resolution of radiotherapy scheduling. En *2010 IEEE Workshop on Health Care Management, WHCM 2010*, Venice, Italy. cited By 5.
- Jacquemin, Y., Marcon, E., e Pommier, P. (2011). A pattern-based approach of radiotherapy scheduling. *IFAC Proceedings Volumes*, 44(1):6945–6950. 18th IFAC World Congress.

- Kapamara, T. e Petrovic, D. (2009). A heuristics and steepest hill climbing method to scheduling radiotherapy patients. En *Proceedings of the 35th International Conference on Operational Research Applied to Health Services, Catholic University of Leuven, Belgium*.
- Kapamara, T., Sheibani, K., Haas, O., Reeves, C., e Petrovic, D. (2006). A review of scheduling problems in radiotherapy. En Burnham, K. e Haas, O., editores, *Proceedings of the Eighteenth International Conference on Systems Engineering (ICSE2006)*, volumen 1, pp. 201–207, United Kingdom. Coventry University. This paper was given at the Eighteenth International Conference on Systems Engineering (ICSE 2006), Coventry: Coventry University, 2006.; ICSE - International Conference on Systems Engineering, 18th ICSE ; Conference date: 05-09-2006 Through 07-09-2006.
- Kim, D.-G. e Kim, Y.-D. (2010). A branch and bound algorithm for determining locations of long-term care facilities. *European Journal of Operational Research*, 206(1):168–177.
- Kopach, R., DeLaurentis, P.-C., Lawley, M., Muthuraman, K., Ozsen, L., Rardin, R., Wan, H., Intrevado, P., Qu, X., e Willis, D. (2007). Effects of clinical characteristics on successful open access scheduling. *Health Care Management Science*, 10(2):111–124.
- Legrain, A., Fortin, M.-A., Lahrichi, N., e Rousseau, L.-M. (2015). Online stochastic optimization of radiotherapy patient scheduling. *Health Care Management Science*, 18(2):110–123.
- Lim, K. S. H., Vinod, S. K., Bull, C., O'Brien, P., e Kenny, L. (2005). Prioritization of radiotherapy in australia and new zealand. *Australasian Radiology*, 49(6):485–488.
- Liu, L. e Liu, X. (1998). Block appointment systems for outpatient clinics with multiple doctors. *Journal of the Operational Research Society*, 49(12):1254–1259.
- Liu, N., Ziya, S., e Kulkarni, V. G. (2010). Dynamic scheduling of outpatient appointments under patient no-shows and cancellations. *Manufacturing & Service Operations Management*, 12(2):347–364.
- Marynissen, J. e Demeulemeester, E. (2019). Literature review on multi-appointment scheduling problems in hospitals. *European Journal of Operational Research*, 272(2):407–419.
- Maschler, J. e Raidl, G. R. (2020). Particle therapy patient scheduling with limited starting time variations of daily treatments. *International Transactions in Operational Research*, 27(1):458–479.
- Mercer, A. (1960). A queueing problem in which the arrival times of the customers are scheduled. *Journal of the Royal Statistical Society. Series B (Methodological)*, 22(1):108–113.
- Murray, M. e Tantau, C. (1999). Redefining open access to primary care. *Managed Care Quarterly*, 7(3):45–55.
- OIEA (2026). Tratamiento del cáncer: Radioterapia. <https://www.iaea.org/es/temas/tratamiento-del-cancer-radioterapia>. Información revisada a 26 de febreiro de 2026 na páxina da OIEA (Organización Internacional da Enxerxía Atómica).
- Petrovic, D., Morshed, M., e Petrovic, S. (2009). Genetic algorithm based scheduling of radiotherapy treatments for cancer patients. En Combi, C., Shahar, Y., e Abu-Hanna, A., editores, *Artificial Intelligence in Medicine*, pp. 101–105, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Petrovic, D., Morshed, M., e Petrovic, S. (2011). Multi-objective genetic algorithms for scheduling of radiotherapy treatments for categorised cancer patients. *Expert Systems with Applications*, 38(6):6994–7002.
- Petrovic, S. e Leite-Rocha, P. (2008). Constructive and grasp approaches to radiotherapy treatment scheduling. En *Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*, pp. 192–200.

- Petrovic, S., Leung, W., Song, X., e Sundar, S. (2006). Algorithms for radiotherapy treatment booking. En *25th Workshop of the UK planning and scheduling special interest group*, pp. 105–112.
- Pham, T. S., Legrain, A., De Causmaecker, P., e Rousseau, L.-M. (2023). A prediction-based approach for online dynamic appointment scheduling: A case study in radiotherapy treatment. *INFORMS Journal on Computing*, 35(4):844–868.
- Pham, T.-S., Rousseau, L.-M., e De Causmaecker, P. (2022). A two-phase approach for the radiotherapy scheduling problem. *Journal of Scheduling*, 25(2):191–207.
- Pinedo, M. L. (2022). *Deterministic Models: Preliminaries*, pp. 13–33. Springer International Publishing, Cham.
- Robinson, L. W. e Chen, R. R. (2009). The effects of patient no-shows on traditional and open-access appointment scheduling policies. Working Paper UC Davis Graduate School of Management Research No. 16-09; Johnson School Research Paper No. 43-09, University of California, Davis, Graduate School of Management and Cornell University, Johnson School.
- Sánchez Galindo, B., López-Torres Hidalgo, J., López González, Á., Sánchez Martínez, C. M., Castaño Díaz, M., e Rabanales Sotos, J. (2025). Intervalos de tiempo para el diagnóstico y tratamiento en pacientes con cáncer de mama. *Atención Primaria*, 57(3).
- Sauré, A., Patrick, J., Tyldesley, S., e Puterman, M. L. (2012). Dynamic multi-appointment patient scheduling for radiation therapy. *European Journal of Operational Research*, 223(2):573–584.
- SEOM (2020). Categoría: Centros. <https://seor.es/category/centros/>. Acceso en: 29-05-2026.
- Silveira, I. M., de Freitas Almeida, J. F., Pinto, L. R., Epaminondas, L. A. R., Conceição, S. V., e Machado, E. L. (2024). A multi-stage optimization model for managing epidemic outbreaks and hospital bed planning in intensive care units. *Healthcare Analytics*, 5:100342.
- The pandas development team (2026). User guide. Acceso en: 03-06-2026.
- Vieira, B. (2020). *Logistical Optimization of Radiotherapy Treatments*. Phd thesis, University of Twente, Enschede.
- Vieira, B., de Armas, J., e Ramalhinho, H. (2022). Optimizing an integrated home care problem: A heuristic-based decision-support system. *Engineering Applications of Artificial Intelligence*, 114:105062.
- Vieira, B., Demirtas, D., van de Kamer, J. B., Hans, E. W., Rousseau, L.-M., Lahrichi, N., e van Harten, W. H. (2020). Radiotherapy treatment scheduling considering time window preferences. *Health Care Management Science*, 23(4):520–534.
- Vieira, B., Hans, E. W., van Vliet-Vroegindewij, C., van de Kamer, J., e van Harten, W. (2016). Operations research for resource planning and -use in radiotherapy: a literature review. *BMC medical informatics and decision making*, 16(1):149–.
- Vogl, P., Braune, R., e Doerner, K. F. (2019). Scheduling recurring radiotherapy appointments in an ion beam facility. *Journal of Scheduling*, 22(2):137–154.
- Wes McKinney (2010). Data Structures for Statistical Computing in Python. En Stéfan van der Walt e Jarrod Millman, editores, *Proceedings of the 9th Python in Science Conference*, pp. 56 – 61.