UNIVERSIDAD DE ZARAGOZA

MÁSTER EN MODELIZACIÓN E INVESTIGACIÓN MATEMÁTICA, ESTADÍSTICA Y COMPUTACIÓN.

Trabajo de fin de máster:

Modelos de optimización en la gestión de proyectos. Caso práctico aplicado a una empresa de servicios

Raúl Baigorri Martínez

Directoras del trabajo:
María del Carmen Galé Pola
Lidia Orellana Lozano

7 de julio de 2021

Prólogo

Durante los últimos años, la Industria del Software ha tenido un desarrollo vertiginoso y un gran impacto a nivel mundial. Las empresas ligadas a esta, desarrollan proyectos que abarcan un abanico amplio de tareas, las cuales deben ajustarse a las peticiones del cliente, por lo que necesitan de trabajadores con habilidades muy específicas. Por ello, a lo largo de las últimas décadas se han propuesto una serie de modelos de optimización matemática, denominados Problemas de Programación de Proyectos, bajo diferentes hipótesis para ajustarse a las variadas realidades de cada empresa.

La formulación matemática de un modelo que tome en consideración de forma fiel los objetivos, las limitaciones y el modo de operar de una empresa requiere de un conocimiento profundo de estas características para que la solución propuesta por el modelo sea de utilidad a la empresa. En primer lugar, se ha revisado la literatura sobre modelos de programación matemática en la planificación de proyectos para seleccionar aquellos que incorporasen alguna de las características del caso de estudio dado por la empresa. Además, se han formulado con una notación común para facilitar su comprensión.

En este TFM se formula un modelo matemático del problema de la empresa cuya resolución proporciona una planificación base de un conjunto de proyectos en un tiempo mucho menor que el que habitualmente emplea en realizar la planificación el responsable del departamento. En el tratamiento discreto del tiempo es importante el cálculo de cotas temporales sobre la duración de cada proyecto para reducir el tiempo de resolución del modelo matemático. Por ello, se utiliza el mismo modelo matemático de forma constructiva, para calcular estas cotas temporales primero para proyectos individuales, luego proyectos en paralelo del mismo tipo y finalmente, proyectos en paralelo de diferente tipo. Se analiza además la relación entre el tiempo de finalización de todos los proyectos y el número de horas totales que deben invertirse en las tareas.

La solución dada por el modelo matemático se presenta de forma visual al responsable del departamento e indica claramente los equipos que ejecutan cada tarea y la asignación temporal de empleados a las tareas. La planificación propuesta verifica todas las restricciones dadas por la empresa y establece un calendario claro para cada empleado, que conoce en qué tarea ha de trabajar cada semana. Esto genera agilidad en el desarrollo real del proyecto. Además, la formación de grupos productivos reduce el tiempo de ejecución de las tareas lo que permite ejecutar un número mayor de proyectos en un mismo intervalo temporal.

Abstract

The increasing demand on projects in software companies of the last decades has generated a growing interest in the cration of operational models which schedules an optimal assignment of resources to each job of the project, depending on the specific habilities of each resource. What's more, the specificeness of each resource who belongs to that companies makes those assignments even more difficult and involves the implementation of automaticed models to established te better solution requiring almost no time.

In the present work a cooperation with Efor Data & AI's departmen is establised in order to use optimization techniques to solve this problem. A deep knowldge is needed to create a model that simulates the software department performance and to be able to write all the restrictions needed. Although the departmen manages several types of project, our work focuses in closed projects, which are alrready budgeted and have an estimation of the number of hours to work on.

This projects are also divided in two different types. On one hand ones require data depuration and analitics and others need a predictive model creation using Artificial Intelligence. Therefore, a bunch of specific habilities is needed for the projects development.

In Chapter 1, a study is first carried out on the Project Scheduling Problem, including an extensive review of simmilar problems to the one presented in this work. This ones presented involve constraints on the resources which indicates if a resource is able to be assigned on a project task depending on its habilities or its capacities. An example of a problem allowing different ways in executing a task and other involving a set of projects are also introduced.

In Chapter 2, the different characteristics of the department which allow us to create the model are introduced. The features of all the employees which form part of the department are first presented. The projects structure is presented next with the different jobs which form part of them and the different roles needed to perform each of them. After that the model formulation and variables are defined with the algorithms needed to established the time windows in wich each project must be compleeted.

In Chapter 3, the general solution of the Department Project Schedulling Problem is presented, propving that it is the better one. Next to it an annalysis of a bunch of problems is analyced testing the variation of the Makespan. This annalysis consider some problems divided in different groups of projects depending on the thir type, the number of projects considered in each problem and the final date obtained when only one of them is solved. The bounds generated in the general problem are created with the results of this analysis.

The problems are solved using an operational reserarch file named MILP: Mix Integred Linear Problem incorporated in Python, whose code is shown in Appendix B.

Índice general

Capítulo 1

Programación matemática en la planificación de proyectos

1.1. Motivación y objetivos

Durante los últimos años, la Industria del Software ha tenido un desarrollo vertiginoso y un gran impacto a nivel mundial. A pesar de sus reconocidos logros, aún resulta significativo el número de proyectos que no culminan con éxito. Según Linberg [20] solo el 16 % de los proyectos de esta industria se terminan a tiempo y dentro del presupuesto. Una de las principales causas de estos malos resultados es la imprecisión en los cálculos sobre la planificación temporal de los proyectos y una asignación de personal con falta de formación adecuada para trabajar en ellos por lo que no opera de la forma más efectiva para la entrega de proyectos.

Una buena planificación y formación de los equipos para realizar los proyectos conlleva entregar los proyectos en un menor tiempo y con la máxima calidad posibles, lo que genera beneficios en la empresa al abarcar un número mayor de proyectos en un mismo tiempo y la dota de prestigio por la calidad de sus proyectos. Además, una planificación y asignación clara de los proyectos en un amplio periodo temporal facilita la organización del trabajo por parte de la empresa, ya que cada empleado dispone de un calendario con las tareas que debe realizar cada semana y las horas que deben invertir en cada una de ellas incentivando la motivación del empleado al proporcionales pequeños objetivos que cumplir cada día. A su vez, el establecimiento de tareas y metas claras y fijas elimina cualquier duda o incertidumbre a los trabajadores sobre su dedicación a lo largo de la jornada laboral, lo que supondría pérdidas de tiempo e ineficiencia en el desarrollo de los proyectos de la empresa.

Por todo esto, el conocimiento profundo de las cualidades del personal disponible y la planificación óptima de los proyectos son unos de los dos principales campos de interés y de estudio en las empresas de servicios a lo largo de estos últimos años.

Para alcanzar estas metas se utilizan modelos de optimización matemática, herramienta que abarca técnicas de modelización para encontrar la mejor solución maximizando el cumplimiento de los objetivos marcados. En este contexto se utilizan para determinar el número de empleados asignados a las tareas de cada proyecto, atendiendo a las habilidades de los mismos y cumpliendo la forma de operar de la empresa, que se traduce en distintas restricciones en el modelo implementado.

Los objetivos principales que se persiguen con la realización del presente TFM se pueden resumir en dos:

- Construcción de una planificación base para el departamento de Data & AI de una empresa de servicios. Gracias a esto se ha determinado el volumen de horas que dedica cada uno de los

trabajadores del departamento a cada una de las tareas a ejecutar dentro de los proyectos, con carácter semanal.

- Mejora de la herramienta de planificación de proyectos actual, a través de la cual el personal responsable de la planificación y carga del departamento puede monitorizar y analizar la planificación base de los proyectos.

Además, mediante el desarrollo del mismo se han llevado a cabo los siguientes objetivos secundarios:

- Anticipación ante posibles picos de trabajo y/o faltas de carga de trabajo en el departamento.
- Facilitar a los trabajadores el conocimiento de las tareas a llevar a cabo semanalmente.
- Adquisición de un conocimiento profundo del funcionamiento del departamento, así como de los diferentes trabajadores.
- Mejorar los tiempos de entrega.

1.2. Revisión de la literatura sobre modelos de planificación de proyectos

En este apartado se presentan algunos modelos de planificación de proyectos revisados en la literatura, comenzando desde el modelo más simple hasta modelos más complejos desarrollados para intentar representar de forma realista los comportamientos que surgen en las empresas. En los trabajos de Hartmann y Briskorn [14] y [15] se incluye una amplia revisión sobre los distintos modelos de programación matemática en la planificación de proyectos dependiendo de sus características y las restricciones a implementar intentando modelar el modo de trabajar de distintas empresas. En particular, se centra la atención en aquellos modelos con características similares y que son de interés para el caso de aplicación de este TFM.

El modelo más simple que resuelve la planificación de un proyecto considera un conjunto de tareas y un conjunto de máquinas capaces de ejecutar las tareas con distintos tiempos de procesamiento cada una de ellas. El problema más general supone un orden o secuencialidad entre tareas, situación también habitual en empresas del sector secundario dedicadas a la transformación de bienes como la industria y la construcción.

Seguidamente se cambia el concepto de máquina por recurso, pudiendo ser la capacidad de cada recurso superior a la unidad. En este modelo cada tarea puede necesitar más de una unidad de recurso por unidad de tiempo (u.t.). A su vez, cada recurso tiene una capacidad por u.t. . De esta manera aparecen nuevas formas de operar que no se contemplaban en los modelos anteriores como tareas que necesitan de un grupo de recursos para ser realizadas y recursos habilitados para estar implicados en varias tareas al mismo tiempo. Este modelo se utiliza en empresas del sector secundario en las que los recursos son empleados, los cuales puedan estar asignados a varias tareas en una misma semana o puedan trabajar en equipo. Estos modelos resuelven la planificación de un único proyecto.

En las empresas es habitual la realización de varios proyectos al mismo tiempo, siendo la función objetivo la minimización en el tiempo del último proyecto en finalizar. Por ello, en la revisión de la literatura se han considerado variantes del modelo más complejas que consideran la planificación de varios proyectos, o el uso de diferentes modos de realizar las tareas que implican diferente uso de los recursos y tiempos de ejecución.

Finalmente, la última variante que se introduce, más próxima al caso de estudio del TFM es un modelo que introduce un conjunto de habilidades necesarias en los recursos utilizados en la ejecución

de algunas tareas. Este modelo representa mejor el tipo de proyectos que se desarrollan en la industria del software ya que se desglosan en distintas tareas como el análisis y depurado de datos, creación de modelos, presentación visual de los resultados y puesta en producción o mantenimiento de los modelos que requieren del conocimiento de distintos programas o lenguajes de programación necesarios para realizarlas y que puede que no posean todos los trabajadores.

A continuación, se describe con más detalle los modelos de planificación de proyectos revisados en la literatura. En la formulación matemática se ha considerado una notación común y por tanto, no se incluye la formulación original. Se han elegido aquellos variantes del problema más próximas al caso de estudio de este TFM.

1.2.1. Problema de planificación de un proyecto

En 1967 se publicó un libro [12] "Theory of scheduling" escrito por los profesores en la Universidad de Cornell, Richard W.Conway, William L.Maxwell y Louis W.Miller sobre un abanico muy amplio de variantes del problema de planificación de tareas, en inglés, "Job Scheduling Problem" (JSP) introducido antes de los años 60. Este problema pertenece a la clase de problemas de optimización NP [6] según la teoría de la complejidad computacional, por lo que el problema no puede ser resuelto en tiempo polinómico.

En este problema se considera un conjunto de tareas, A y un conjunto M de máquinas que deberán realizar todas las tareas para considerar que el proyecto está terminado.En el modelo se asumen las siguientes hipótesis:

- Las tareas se realizan secuencialmente y no pueden solaparse entre ellas.
- Las tareas no pueden ser interrumpidas a lo largo de su ejecución.
- Cada máquina es renovable, esto es, cuando finaliza una tarea puede comenzar otra.
- El tratamiento del tiempo es discreto y una u.t. puede ser una hora, día, semana, mes, etc.
- El objetivo consiste en minimizar el tiempo total de ejecución del proyecto, en inglés *makespan*, esto es, el tiempo o instante temporal en el que termina la última tarea del proyecto.

A lo largo de los años se han ido desarrollando variantes del problema que han surgido en la industria y atienden a las restricciones específicas de cada sistema real. Estas variantes estudiadas por Hartman y Briskorn [15] se pueden clasificar atendiendo a los subconjuntos de tareas que puede realizar cada máquina, la relación cardinal entre tareas y máquinas, las relaciones de precedencia entre las tareas, la posibilidad de realizar tareas simultáneamente y parones en mitad de ejecución de una tarea.

La meta más habitual en las empresas consiste en determinar el máximo número de trabajos, posiblemente con prioridades, a completar antes de una fecha dada o el mínimo tiempo de ejecución de un conjunto dado de trabajos. En los modelos propuestos en la literatura se incluyen otras funciones objetivo como la maximización en la cantidad utilizada de productos biodegradables consumidos a lo largo de un proyecto que buscan abarcar las nuevas tendencias, finalidades o propósitos de empresas que se ajustan a su política interna intentando reducir la huella ecológica o su impacto medioambiental.

Formulación del Modelo

A continuación se introduce la formulación más común del modelo JSP, reemplazando los costes de asignación por la duración de cada tarea si la realiza una determinada máquina.

En la formulación matemática del problema se introducen dos conjuntos:

- A: Conjunto de tareas.
- M: Conjunto de máquinas.

Todas máquinas pueden realizar todas las tareas, pero el tiempo de procesamiento de cada tarea depende de la máquina que la lleve a cabo, por lo que se introduce el siguiente parámetro:

- d_m^a: Número de u.t. necesarias para realizar la tarea $a \in A$ si es ejecutada por la máquina $m \in M$.

Las variables de decisión son binarias y determinan la asignación de una tarea a una determinada máquina:

- x_m^a: Toma el valor 1 si la máquina $m \in M$ es asignada a la tarea $a \in A$.

El objetivo del problema consiste en minimizar el tiempo de finalización del proyecto, que coincide con la suma de los tiempos de ejecución de cada tarea ya que estas se realizan secuencialmente,

$$f(x) = min \sum_{a \in A} d_m^a x_m^a. \tag{1.1a}$$

El siguiente conjunto de restricciones garantizan que cada tarea ha de ser realizada solo por una máquina,

$$\sum_{m \in M} x_m^a = 1 \quad \forall a \in A. \tag{1.1b}$$

En la Figura 1.1 se muestran las tareas que puede realizar cada máquina y el tiempo requerido para su proceso. A continuación, se incluye el ejemplo ilustrativo propuesto por Blazeewicz et al. [6] de un problema con tres tareas y tres máquinas.

Las máquinas M_2 y M_3 son capaces de realizar las tres tareas, sin embargo, la máquina M_1 solo es capaz de realizar A_1 y A_3.

Tiempos de procesamiento de una instancia con 3 tareas y 3 máquinas

	A1	A2	A3
M1	3	--	3
M2	2	4	6
M3	3	3	2

Figura 1.1: Tiempos de proceso en un ejemplo con tres tareas y tres máquinas ([6])

Si se supone que cada máquina debe realizar una tarea y las tareas se han de ejecutar de forma secuencial, la solución óptima se indica en verde en la Figura 1.1: la máquina M_1 procesa la tarea A_1, M_2 la tarea A_2 y M_3 la tarea A_3 en un tiempo de $3+4+2=9$ u.t.

Sin embargo, si se permite que una máquina realice todas las tareas, el espacio de factibilidad aumenta y se puede obtener una mejor solución asignando a la máquina M_3 las tres tareas (en azul) con un procesamiento completo en $3+3+2=8$ u.t.

El ejemplo ilustrativo muestra la importancia de las restricciones en el valor de la función objetivo en el óptimo, de ahí la necesidad de conocer en profundidad las características de funcionamiento de una empresa para formular un modelo matemático que se ajuste lo mejor posible a la realidad de la misma.

1.2.2. Problema de planificación de un proyecto con restricciones en los recursos

Esta primera variante se denomina en inglés "Resource-Constrained Project Scheduling Problem" (RCPSP) y considera la planificación de un proyecto compuesto por un conjunto de tareas que han de ejecutarse sin interrupción una vez puestas en marcha.

A diferencia del modelo anterior en el que para procesar las tareas se utilizaban máquinas, en estos modelos se utiliza el término genérico recurso. Esto es debido a que la relación cardinal en el modelo anterior entre tarea y máquina era de uno uno y en un contexto más general se introduce el concepto de capacidad de un recurso.

En este modelo cada tarea requiere de un número de unidades de recurso por u.t. para que pueda ejecutarse, lo que implica que se formen grupos de recursos en la realización de tareas. A su vez, cada recurso tiene un número de unidades de capacidad por u.t., lo que permite que un mismo recurso, si tiene capacidad suficiente sea utilizado para trabajar en varias tareas en un mismo instante temporal.

El objetivo del modelo sigue siendo minimizar el tiempo en que se completa el proyecto.

Formulación del Modelo

Un primer modelo fue propuesto por Pritsker et al. en 1969 ([30]) y considera un tratamiento discreto del tiempo.

En el conjunto de tareas A se incluyen dos tareas ficticias, una al principio 0 y otra al final $|A|$, sin necesidad de recursos para su procesado y que se utilizan para indicar el comienzo y la finalización de un proyecto. Las tareas tienen una relación de precedencia, de manera que no se inicia una tarea hasta que no han finalizado antes todas las que la preceden. Dada una tarea $a \in A$ se denota por P_a al conjunto de tareas que preceden a la tarea a y sus características se introducen a partir de los siguientes parámetros:

- d^a: Número de unidades de tiempo necesarias para procesar la tarea.
- ES^a (*Earliest Starting Time*): Primer instante de tiempo en el que puede empezar la tarea.
- LS^a (*Latest Starting Time*): Último instante de tiempo en el que puede empezar la tarea $a \in A$.
- EF^a (*Earliest Finishing Time*): Primer instante de tiempo en el que puede terminar la tarea.
- LF^a (*Latest Finishing Time*): Último instante de tiempo en el que puede terminar la tarea $a \in A$.

Notemos que, $EF^a = ES^a + d^a$ y $LF^a = LS^a + d^a$. La ventana temporal de una tarea $a \in A$ puede estar determinada por la empresa o se puede estimar de la forma más precisa posible a partir de la información disponible. La inclusión de estos parámetros en el modelo, tal y como se verá más adelante, contribuyen a reducir el tamaño del problema, a resolverlo en un menor tiempo computacional y a mejorar la calidad de las soluciones obtenidas.
Dado un recurso $w \in W$ se conoce:

- H_w: Número de unidades del recurso $w \in W$.
- c_w^a: Número de unidades del recurso $w \in W$ necesarias cada u.t. durante la ejecución de la tarea $a \in A$.

Finalmente, sea T_{MAX} el máximo número de unidades temporales disponibles para realizar el proyecto y T el conjunto de instantes temporales, $T = \{1, 2, \ldots, T_{MAX}\}$.

Las variables de decisión son binarias y determinan el instante en el que se comienza cada tarea del proyecto:

- $I^{at} \in \{0, 1\}$: Toma el valor 1 si la tarea $a \in A$ empieza en el instante $t \in \{ES^a, ..., LS^a\}$.

El objetivo del problema consiste en minimizar el tiempo de proceso del proyecto que coincide con el fin de ejecución de la última tarea, cuyo tiempo de proceso es 0:

$$f(x) = min \sum_{t \in T} t I^{|A|t} \tag{1.2a}$$

Las restricciones del problema son de dos tipos ([36]):

a) Restricciones en la secuencia de las tareas. Cada tarea puede empezar solo una vez:

$$\sum_{t=ES^a}^{t=lS^a} I^{at} = 1 \quad \forall a \in A. \tag{1.2b}$$

Se garantiza la relación de precedencia de las tareas, esto es, una tarea comienza en un instante posterior a la finalización de todas las tareas de su conjunto de precedencia:

$$\sum_{t=ES^a}^{LS^a} t I^{at} \geq \sum_{t=ES^\alpha}^{LS^\alpha} t I^{\alpha t} + d^\alpha \quad \forall a \in A, \forall \alpha \in P_a. \tag{1.2c}$$

b) Restricciones de capacidad de los recursos. Se garantiza que el número de unidades requeridas de un recurso dado cada u.t para relizar las tareas que se estén ejecutando en dicha u.t. sea inferior a la capacidad del recurso.

Como se pueden realizar varias tareas a la vez, el sumatorio considera todas las tareas,

$$\sum_{a \in A} \sum_{q=max\{t, EF^a\}}^{min\{t+d^a-1, LF^a\}} c_w^a I^{aq} \leq H_w \quad \forall w \in W, \forall t \in T. \tag{1.2d}$$

Una representación gráfica de un proyecto denominada "Activity-on-Node Network" introduce un grafo $\mathscr{G}(\mathscr{N}, \mathscr{E})$ donde cada nodo representa una tarea y cada arco la relación de precedencia directa entre dos tareas. En la Figura 1.2 se muestra el grafo correspondiente a un proyecto con cinco tareas y un recurso de capacidad 2 dado por Schwindt y Zimmerman ([31]).

Las tareas 0 y 6 son las tareas ficticias. Junto a cada nodo representado con un cuadrado se incluye, en la parte inferior, la cantidad de recurso necesaria para procesar la tarea y, en la parte superior, las unidades de tiempo necesarias. Por ejemplo, la tarea 5 requiere procesarse durante una unidad de tiempo por dos unidades de recurso y el conjunto $P_5 = \{3, 4\}$, luego la tarea 5 solo se comenzará cuando las tareas 3 y 4 hayan finalizado.

En la Figura 1.3 se ha representado una solución factible del problema ya que se verifica las relaciones de precedencia, el tiempo de ejecución de cada tarea y las unidades de recurso necesarias y no se excede la capacidad del recurso. Sin embargo, esta solución no es óptima ya que hay instantes intermedios en los que no se realiza ninguna tarea y algunas de las tareas sin relación de precedencia se pueden ejecutar en el mismo instante temporal ya que no superan la capacidad del recurso. En la Figura 1.4 se ha representado una mejor solución.

La solución factible de la Figura 1.4 se puede mejorar ya que las tareas 1 y 5 se pueden solapar al no tener relación de precedencia entre ellas. La Figura 1.5 representa la solución óptima al problema que ejecuta el proyecto en 5 u.t.

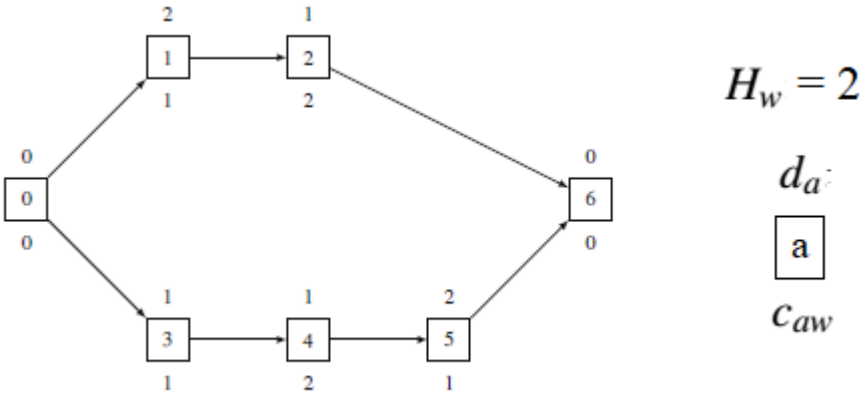


Figura 1.2: RCPSP con cinco tareas y un recurso con capacidad 2.

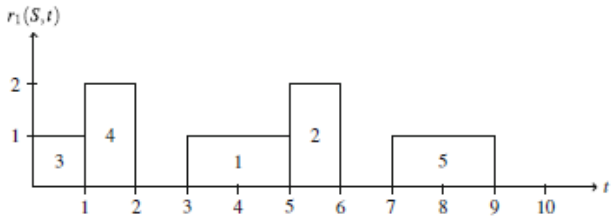


Figura 1.3: Solución factible en 9 u.t. del RCPSP con cinco tareas y un recurso con capacidad 2 unidades ([31]).

1.2.3. Problema de planificación de un proyecto con restricciones en los recursos multi-modales

Una extensión del modelo RCPSP consiste en considerar que cada tarea admite realizarse de diferentes modos y se denomina en inglés, *Multi-Mode Resource Constrained Project Scheduling Problem* (MM-RCPSP). La formulación matemática del MM-RCPSP más utilizada fue introducida por Tablot en 1982 y está basada en el modelo de RCPSP de Pritsker et al. de 1969 ([30]). El modelo presentado a continuación se ha formulado atendiendo al modelo introducido por Schwindt y Zimmermann en su manual de gestión y programación de proyectos ([31]).

En este nuevo modelo se introduce un conjunto de modos de ejecución M que viene representado por el tiempo de proceso de la tarea y el número de unidades de recurso requeridas por u.t. . Al tratarse de una extensión del RCPSP, el MM-RCPSP es NP-completo en sentido estricto ([17]).

Formulación del Modelo

En la formulación matemática del MM-RCPSP se introduce para cada tarea $a \in A$, el conjunto M_a de modos de procesado de la tarea. En esta variante el tiempo de proceso de una tarea d_a, depende del modo de ejución $m \in M_a$, por ello se introducen para cada tarea $a \in A$ y recurso $w \in W$ dos nuevos parámetros que sustituyen a d_a y c_a^w del RCPSP:

- d_m^a: Tiempo de proceso de la tarea a en el modo de ejecución $m \in M_a$.
- c_{mw}^a: Unidades de recurso w necesarios para realizar la tarea a en el modo de ejecución $m \in M_a$.

Análogamente, dada una tarea $a \in A$ y recurso $w \in W$, las variables de decisión I^{at} del RCPSP se sustituyen por:

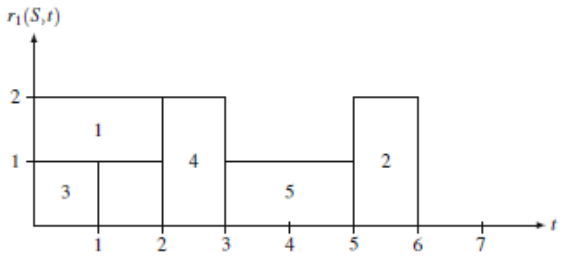


Figura 1.4: Solución factible en 6 u.t. del RCPSP con cinco tareas y un recurso con capacidad 2 unidades ([31]).

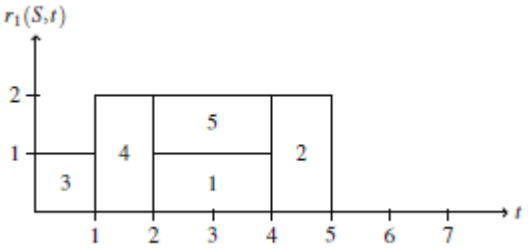


Figura 1.5: Solución óptima del RCPSP con cinco tareas y un recurso con capacidad 2 unidades ([31]).

- $I_m^{at} \in \{0,1\}$: toma el valor 1 si la tarea a empieza en el instante $t \in \{ES^a,...,LS^a\}$ en el modo de ejecución $m \in M_a$.

La función objetivo del MM-RCPSP consiste en minimizar el tiempo total de ejecución del proyecto,

$$f(x) = min \sum_{t \in T} I_1^{(|A|)t}. \tag{1.3a}$$

Las restricciones han de garantizar que cada tarea puede empezar solo una vez y ejecutarse de un único modo,

$$\sum_{m=1}^{M_a} \sum_{t=ES^a}^{LS^a} I_m^{at} = 1 \quad \forall a \in A. \tag{1.3b}$$

Si existe relación de precedencia, el tiempo de inicio de una tarea debe ser mayor o igual que el tiempo de inicio más el tiempo de ejecución de todas las tareas precedentes a ella,

$$\sum_{m=1}^{M_a} \sum_{t=ES^a}^{LS^a} tI_m^{at} \geq \sum_{m=1}^{M_\alpha} \sum_{t=ES^\alpha}^{LS^\alpha} (tI_m^{\alpha t} + d_m^\alpha) \quad \forall a \in A, \forall \alpha \in P_a. \tag{1.3c}$$

Las siguientes restricciones garantizan que cada instante temporal el número de recursos asignados a las tareas no excede la capacidad del recurso,

$$\sum_{a \in A} \sum_{m=1}^{M_a} \sum_{\tau=max\{t,ES^a\}}^{min\{t+d_m^a-1,LS^a\}} c_{wm}^a I_m^{a\tau} \leq H_w \quad \forall w \in W, \forall t \in T. \tag{1.3d}$$

Otra formulación matemática alternativa al modelo 1.3 está basada en el concepto de subconjuntos factibles de tareas de Vittorio et al. ([24]), formados por tareas posibles de ejecutrse paralelamente, sin relación de precedencia y que no superen las capacidades de los distintos recursos. Esta formulación no se incluye en esta memoria ya que fue propuesta para modelos con un solo proyecto donde cada tareas tiene un número alto de tarea precedentes y no se corresponde con el caso de estudio considerado en

este TFM.

En este modelo, cada modo viene definido por el número de recursos que están implicados en una tarea dada. Aunque teóricamente al aumentar el número de recursos asociado a una tarea se reducirá su tiempo de procesamiento, lo que significará una relación inversamente proporcional entre estos dos parámetros, en la realidad esta propiedad no se tiene por qué cumplir siempre, ya que al aumentar el número de empleados que trabajan en una misma tarea se puede reducir la efectividad del grupo. Por ello algunos autores como Lova et al. ([21]) han centrado su interés en la formulación y resolución del MM-RCPSP a partir del cálculo de la duración de una tarea d_m^a como función discreta del número de unidades de recurso c_{mw}^a que se emplean para realizarla.

1.2.4. Problema de planificación de múltiples proyectos con restricciones en los recursos

Esta variante del RCPSP, en inglés *Multi-Project Resource Constrained Project Scheduling Problem* (MP-RCPSP) ([32]), se consideran varios proyectos que pueden ser ejecutados de forma simultánea algunos de ellos. Las variantes de este problema son inmensas. En este apartado se asumen las restricciones del RCPSP y que las relaciones de precedencia de tareas solo se establecen entre tareas que pertenecen al mismo proyecto. Este modelo se conoce como modelo en entorno estático y fue ya estudiado por Fendley en 1968 ([13]) y Pritsker et al. ([30]) entre otros. Al tratarse de una extensión del problema RCPSP, MP-RCPSP también es NP-completo en sentido estricto ([17]).

Formulación del Modelo

En el MP-RCPSP se introduce un nuevo conjunto B de proyectos. Para cada proyecto $b \in B$, se define el conjunto A_b de tareas a realizar y los siguientes parámetros que coinciden en algún caso con parámetros del RCPSP salvo el índice adicional que indica el proyecto:

- d_b^a: Tiempo de proceso de la tarea $a \in A_b$.
- g_b: Instante de finalización deseada del proyecto.
- E_b: Primer instante de tiempo en el que puede terminar el proyecto.
- G_b: Último instante de tiempo en el que puede terminar el proyecto.
- ES_b^a: Primer instante de tiempo en el que se puede empezar la tarea $a \in A_b$.
- LS_b^a: Último instante de tiempo en el que se puede empezar la tarea $a \in A_b$.
- EF_b^a: Primer instante de tiempo en el que puede terminar la tarea $a \in A_b$.
- LF_b^a: Último instante de tiempo en el que puede terminar la tarea $a \in A_b$.
- c_{bw}^a: Unidades del recurso $w \in W$ necesarias por u.t. para realizar la tarea $a \in A_b$.

En cuanto a las variables de decisión, en este caso se centran en el instante de finalización de una tarea, en lugar del instante de inicio:

- $F_b^{at} \in \{0, 1\}$: toma el valor 1 si la tarea $a \in A_b$ del proyecto $b \in B$ finaliza en el tiempo $t \in \{EF_b^a, ..., LS_b^a\}$.
- $F_b^t \in \{0, 1\}$: toma el valor 1 si todas las tareas del proyecto b han sido completadas para el periodo t.
- $F^t \in \{0, 1\}$: toma el valor 1 si todos los proyectos han sido completados en el instante t.

Pritsker et al. ([30]) incluyen tres posibles funciones objetivo:

1. Minimizar el tiempo total de ejecución de todos los proyectos, de forma que se maximiza la siguiente función objetivo:

$$f(x) = \sum_{t=max\{E_b\}}^{max\{G_b\}} F^t \tag{1.4a}$$

Notemos que, dadas las restricciones sobre la finalización de los proyectos dadas por los parámetros E_b y G_b, $b \in B$, el primer instante más temprano en el que pueden estar completados todos los proyectos es $max\{E_b\}$ y el último instante en el que todos los proyectos pueden de estar completados es $max\{G_b\} \quad \forall b \in B$.

2. Minimizar el tiempo de ejecución de cada proyecto, lo que se hace maximizando la siguiente función objetivo:

$$f(x) = \sum_{b \in B} \sum_{t=E_b}^{G_b} F_b^t. \tag{1.4b}$$

La variable F_b^t toma el valor 1 a partir del instante de tiempo en el que finaliza el proyecto $b \in B$ y no puede ser anterior a E_b ni posterior a G_b.

3. Minimizar el retraso total. Dado que un proyecto tiene una fecha deseada de finalización, g_b, se define el retraso de un proyecto $b \in B$ como el número de u.t. por encima de g_b que tarda un proyecto en finalizarse:

$$\sum_{b \in B} \sum_{t=g_b+1}^{G_b} (1 - F_b^t). \tag{1.4c}$$

Se produce un retraso si la variable F_b^t es 0 para los instantes $t \in [g_b + 1, G_b]$ ya que si finaliza en g_b lo hace a tiempo. Si la importancia de entregar a tiempo los proyectos no es la misma se multiplica el retraso individual de cada proyecto por una constante positiva γ_b que penaliza que el proyecto se retrase, obteniendo una función objetivo de la siguiente forma:

$$\sum_{b \in B} \sum_{t=g_b+1}^{G_b} \gamma_b (1 - F_b^t). \tag{1.4d}$$

La constante de penalización puede depender de la magnitud del retraso, γ_b^t, $b \in B, t \in [g_b + 1, G_b - 1]$, ya que que no es lo mismo terminar un proyecto una, dos o tres semanas tarde, esto es, $\gamma_b^t \geq \gamma_b^{t+1}$. En este caso, la función objetivo es:

$$f(x) = \sum_{b \in B} \sum_{t=g_b+1}^{G_b} \gamma_b^t F_b^t. \tag{1.4e}$$

El conjunto de restricciones se modifica para incorporar todos los proyectos. Las siguientes restricciones garantizan que cada tarea finaliza en un único instante temporal:

$$\sum_{t=EF_b^a}^{LF_b^a} F_b^{at} = 1 \quad \forall b \in B, \forall a \in A_b \tag{1.5a}$$

Un proyecto está finalizado solo cuando todas las tareas se han finalizado:

$$F_b^t = 1 \Leftrightarrow \sum_{\tau=EF_b^a}^{t-1} F_b^{a\tau} = 1 \quad \forall a \in A_b$$

$$F_b^t \leq \frac{1}{|A_b|} \sum_{a \in A_b} \sum_{\tau=EF_b^a}^{t-1} F_b^{a\tau} \quad \forall b \in B, \forall t \in [e_b, G_b]. \tag{1.5b}$$

Donde $|A_b|$ indica la función cardinal. La finalización de todos los proyectos se formula de forma análoga:

$$F^t \leq \frac{1}{|B|} \sum_{b \in B} F_b^{\tau}, \quad \forall t \in [max(e_b), max(G_b)]. \tag{1.5c}$$

Notemos que, si en t no se han finalizado todas las tareas, el lado derecho de la ecuación es estrícta-mente menor que 1 y al ser la variable F_b^t binaria, tomará el valor 0. En otro caso, el lado derecho toma el valor 1 y al ser la función objetivo de máximo, F_b^t tomará el valor 1.

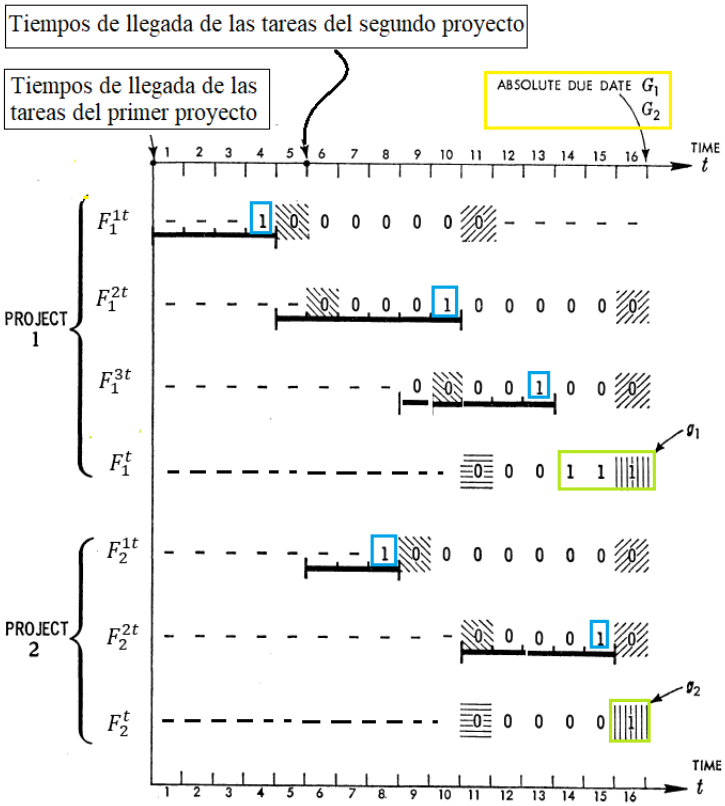
En este modelo, las restricciones sobre la precedencia de tareas son:

$$\sum_{t=EF_b^{\alpha}}^{LF_b^a} tF_b^{\alpha t} + d_b^{\alpha} \leq \sum_{t=EF_b^a}^{LF_b^a} tF_b^{at} \quad \forall b \in B, \forall a \in A_b, \forall \alpha \in P_a. \tag{1.5d}$$

Y las restricciones de capacidad de los recursos:

$$\sum_{b \in B} \sum_{a \in A_b} \sum_{\tau=t}^{min\{t+d_b^a-1\}} c_{bw}^a F_b^{a\tau} \leq H_w, \quad \forall w \in W, \quad \forall t \in \{1, ..., max(G_b)\}. \tag{1.5e}$$

En la Figura 1.6 se muestra un ejemplo ilustrativo dado en [30] con el calendario de dos proyectos con 3 y 2 tareas, respectivamente.



PROYECTO	TAREA	d	ES	EF	LF	E	G
1	1	4	0	5	11	11	11
1	2	6	0	6	16	11	16
1	3	5	5	9	16	11	16
2	1	3	6	9	16	11	16
2	2	5	6	10	16	11	16

Figura 1.6: Planificación de 2 proyectos ([30])

Las dos primeras tareas del primer proyecto se pueden comenzar a ejecutar en el instante 0, ya que vemos que la fecha para la que pueden estar completadas coincide con el instante temporal con una unidad mayor al valor de sus respectivas duraciones. Esto significa que cualquiera de las dos tareas puede ejecutarse antes que la otra y que por tanto no tienen relación de precedencia entre ellas. Además, como se observa en la Figura 1.6, estas no pueden realizarse en paralelo, lo que significa que requieren de un número de recursos mayor que el disponible.

Por el contrario, la tercera tarea de dicho proyecto tiene una relación de precedencia con la primera, ya que su fecha más temprana para la cual puede estar finalizada es una unidad mayor que el valor de la suma de las duraciones de ambas tareas. Por otra parte, el hecho de que las fechas de finalización más tardías de las tareas dos y tres de este proyecto sean iguales indica que estas no tienen relación de precedencia y como sus ejecuciones coinciden en dos instantes temporales la suma de recursos necesarios para realizar ambas tareas es menor que el total disponible.

Las tareas del segundo proyecto pueden empezar a realizarse a partir del sexto instante temporal. Al igual que en el caso del primer proyecto, estas no tienen relación de precedencia, ya que la fecha para la que pueden estar completadas coincide con el instante temporal con una unidad mayor al valor de sus respectivas duraciones más la fecha más temprana en la que pueden comenzar a ejecutarse.

En la solución se decide ejecutar la primera de este proyecto paralelamente a la segunda del primero y la segunda junto a la tercera del primer proyecto debido a que no superan la cantidad de recursos disponible.

1.2.5. Problema de planificación de un proyecto con restricciones en los recursos con múltiples habilidades

Esta variante del problema se conoce en inglés por *Multi-Skilled Resource Constrained Project Scheduling Problem* (MS-RCPSP) ([11]) y surge de forma natural en consultoras en las que los recursos son sus empleados. En este caso no se introduce el concepto de capacidad de recurso porque cada recurso tiene capacidad 1, esto es: $H_w = 1 \quad \forall w \in W$, siendo en este caso W el conjunto de empleados. En estas empresas se han de planificar varios proyectos en los que las tareas suelen requerir de una serie de competencias para poder ser realizadas y cada empleado posee una serie de habilidades que le permiten realizar aquellas tareas que requieran dichas competencias.

En el MS-RCPSP, algunas hipótesis coinciden con los modelos anteriores:

- Cada empleado es renovable, esto es, cuando finaliza una tarea puede comenzar otra tarea.
- Las tareas no pueden ser interrumpidas.
- El tratamiento del tiempo es discreto y cada u.t. puede ser una hora, día, semanas, mes, etc.
- Los recursos requeridos para cada tarea son conocidos y no varían en el tiempo.

Además, se asumen algunas hipótesis adicionales a los modelos anteriores sobre los recursos y las tareas:

- Cada empleado tiene capacidad uno.
- Cada empleado puede contribuir con solo una habilidad en cada tarea, aunque posea otra habilidad requerida en la tarea.

En este caso, cada tarea, en lugar de requerir un número de recursos cada u.t. como en los modelos anteriores va a requerir de un número de unidades de una o varias habilidades. Como cada empleado tiene una capacidad 1, cada unidad de habilidad será cubierta por un solo empleado y se deberán de cubrir al completo para poder ejecutar la tarea. Un empleado debe de tener al menos una de las habilidades requeridas por una tarea para poder estar implicado en esta tarea y aunque posea varias habilidades que se requieran por la tarea solo desempeñará una.

Para una tarea dada, supongamos la tarea 1, se pueden requerir tres habilidades diferentes y distintas unidades de habilidad para cada una de ellas, lo que se traduce en un número de recursos. De este modo la tarea 1 necesita de 4 empleados de los cuales dos de ellos trabajarán desarrollando la habilidad 1, uno de ellos desempeñando la habilidad 3 y otros dos con la habilidad 4.

Formulación del Modelo

En el MS-RCPSP se mantienen los conjuntos de tareas A y se añade el conjunto R de habilidades.

Otros conjuntos que interesa definir para la formulación matemática del problema son:

- $R_a \subseteq R$: Conjunto de habilidades requeridas para realizar la tarea $a \in A$.

- $R_w \subseteq R$: Conjunto de habilidades del empleado $w \in W$.
- $A_w \subseteq A$: Conjunto de tareas en las que puede participar el empleado $w \in W$ al disponer de al menos una de las habilidades necesarias.
- $W_a \subseteq R$: Conjunto de recursos que pueden contribuir con al menos una habilidad a la tarea $a \in A$.
- $W_r \subseteq W$: Conjunto de recursos con la habilidad $r \in R$.

Además de los parámetros ya definidos en los modelos anteriores relativos a las posibles fechas de iniciación y finalización de proyectos y los tiempos de procesamiento de cada tarea, se define un parámetro equivalente a los definidos en los modelos anteriores que especifica la relación de cardinalidad entre tarea y empleado:

- c_r^a: Número de recursos con la habilidad $r \in R$ necesarios para realizar la tarea $a \in A$.

Al igual que en modelos anteriores las variables de decisión son binarias y determinan el instante en el que comienza cada tarea del proyecto, la asignación de los empleados a las distintas tareas y la habilidad que desarrollan en ella en caso de estar asignados:

- $I^{at} \in \{0,1\}$: toma el valor 1 si la tarea $a \in A$ empieza en el tiempo $t \in \{ES^a, ..., LS^a\}$.
- $x_{wr}^{at} \in \{0,1\}$: toma el valor 1 si el empleado w empieza la tarea a en el tiempo t ejecutando la habilidad r. Donde $a \in A$, $w \in W_a$, $r \in R_a$ y $t \in \{ES^a, ..., LS^a\}$.

El objetivo sigue siendo minimizar el tiempo de ejecución del proyecto, lo que equivale a minimizar el tiempo de comienzo de la tarea ficticia $|A|$ que será igual que el tiempo de finalización de la última tarea real,

$$f(x) = min \sum_{t \in T} tI^{|A|t}. \tag{1.6a}$$

Las restricciones han de garantizar que cada tarea empiece solo una vez,

$$\sum_{t=ES^a}^{LS^a} I^{at} = 1 \quad \forall a \in A. \tag{1.6b}$$

Además deben cumplir las restricciones de precedencia entre tareas, cumpliéndose que el tiempo de inicio de una tarea sea mayor o igual que el tiempo de inicio más el de duración de todas las tareas pertenecientes a su conjunto de precedencia,

$$\sum_{t=ES^a}^{LS^a} tI^{at} \geq \sum_{t=ES^\alpha}^{LS^\alpha} tI^{\alpha t} + d_\alpha \quad \forall a \in A, \forall \alpha \in P_a. \tag{1.6c}$$

Cada tarea a del proyecto debe tener asignados a todos los recursos necesarios para que pueda ejecutarse.

$$\sum_{t=ES_a}^{LS_a} \sum_{w \in W_r} x_{wr}^{at} = c_r^a \quad \forall a \in A \setminus \{|A|\}, \forall r \in R_a. \tag{1.6d}$$

Cada tarea tiene una fecha de inicio a partir de la cual se incorporan todos los empleados asignados a ella, por lo que si un trabajador está implicado en una tarea a desarrollando una habilidad r, la comenzará en el tiempo de inicio de dicha tarea, siendo este el mismo para todos los implicados correspondientes,

$$\sum_{r \in R_w \cap R_a} x_{wr}^{at} \leq I^{at} \quad \forall w \in W, \quad \forall a \in A_w, \forall t \in \{ES_a, ..., LS_a\}. \tag{1.6e}$$

Por último, cada empleado solo puede estar asignado en una sola tarea y ejecutando una sola habilidad en un mismo instante temporal,

$$\sum_{a \in A_w} \sum_{t=max\{ES_a, t-d_a+1\}}^{min\{LS_a, t\}} \sum_{r \in R_w \cap R_a} x_{wr}^{at} \leq 1 \quad \forall w \in W, \forall t \in T. \tag{1.6f}$$

1.3. Tratamiento de la evolución temporal de un proyecto

En este apartado se van a incluir tres tratamientos alternativos en tiempo discreto y un tratamiento en tiempo continuo.

La planificación de la ejecución de las tareas de uno o más proyectos requiere proporcionar información sobre cuándo se ejecutan las tareas y qué recursos se utilizan en su ejecución. De ahí que el tiempo es un elemento clave en la formulación matemática de cualquier modelo. En la sección anterior se ha realizado el mismo tratamiento discreto del tiempo en los modelos presentados introducida en [30], denominada P_{DT} del inglés *Discrete-Time Problem*.

1.3.1. Tratamiento discreto desagregado

Este modelo fue propuesto por Konéet al. ([19]) y las restricciones desagregadas por instante de tiempo que sustituyen al conjunto de restricciones 1.6c son:

$$\sum_{\tau=t}^{LS^\alpha} I^{\alpha\tau} + \sum_{\tau=ES^a}^{min\{LS^a, t+d^\alpha-1\}} I^{a\tau} \leq 1 \quad \forall a \in A, \quad \forall \alpha \in P_a, \quad \forall t \in \{ES^\alpha, ..., LS^\alpha\}. \tag{1.7}$$

Dada una tarea $a \in A$, una tarea $\alpha \in P_a$ y un instante temporal $t \in \{ES^\alpha, ..., LS^\alpha\}$ en el que la tarea α pueda comenzar, las restricciones 1.7 indican solo uno de los dos sumatorios puede tomar valor 1.

Si el primer sumatorio toma el valor 1, significa que la tarea $\alpha \in P_a$ empieza en un instante de tiempo $t_0 \in [t, LS^\alpha]$, por lo que la tarea a podrá empezar en un instante superior a $t + d_\alpha$ que supondremos que está dentro del intervalo de tiempos en el que puede empezar la tarea a: $t + d_\alpha \in [ES^a, LS^a]$, y el segundo sumatorio se obliga a que tome el valor 0.

Por el contrario, si el primer sumatorio vale 0, significa que la tarea $\alpha \in P_a$ empieza en un tiempo anterior a t, por lo que finalizará antes del instante temporal $t + d_\alpha$. Así el segundo sumatorio podrá ser 1, suponiendo que la tarea a podrá empezar en un instante $t_a \in [ES^a, t + d_\alpha]$, suponiendo de nuevo que: $t + d_\alpha \in [ES^a, t + d_\alpha]$.

Esta formulación alternativa aumenta el número de restricciones en el modelo final, de ahí su denominación como Problema Desagregado en Tiempo Discreto, en inglés *Disagregated Discrete-Time Problem* (P_{DDT})

En la Figura 1.7 incluye un ejemplo con dos tareas para ilustrar la aplicación de esta restricción, en la que la tarea 1 precede a la 2.

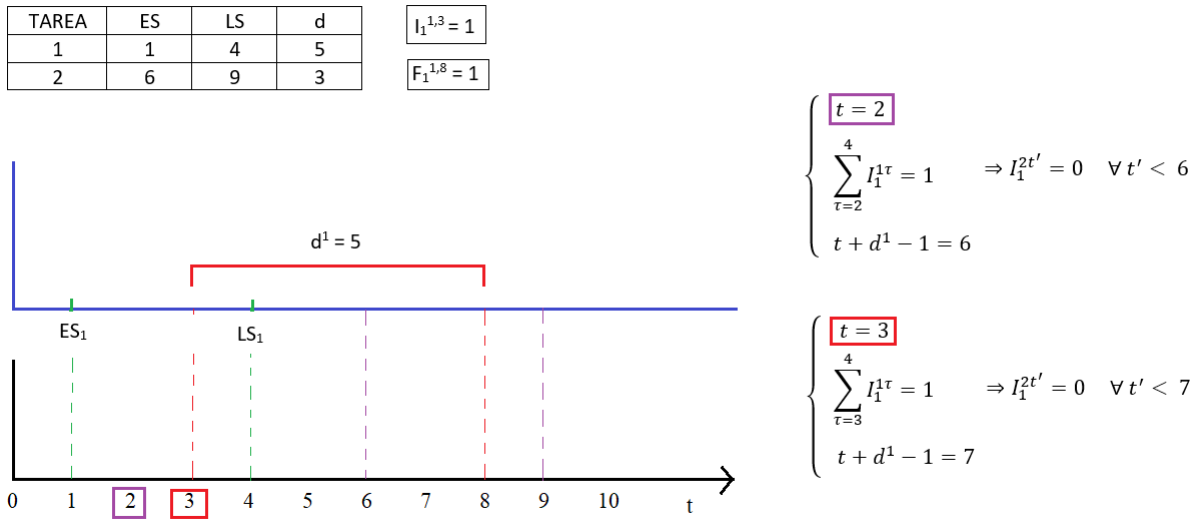


Figura 1.7: Visualización de la restricción desagregada que impone la precedencia entre tareas.

En el ejemplo, la tarea 1 puede comenzar como pronto en el instante de tiempo 1 y como tarde en el instante de tiempo 4, además tiene una duración de cinco u.t. . Supongamos que la tarea 1 comienza en $t_0 = 3$, o equivalentemente que $I_1^{1,3} = 1$, entonces termina en el instante temporal 8 y que hasta ese instante no se puede comenzar la tarea 2.

Como la primera tarea empieza en $t_0 = 3$, si se fija el instante temporal en $t = 3$ o en cualquier instante anterior $t \in [ES_\alpha, t_0]$ el primer sumatorio de la ecuación 1.7 tomará el valor 1 imponiendo que la tarea 2 no se puede empezar hasta un instante $t' \geq t_0 + d^\alpha$, siendo en este caso 8.

En el trabajo de Almeida et al. [1] se demuestra que la formulación desagregada disminuye el tiempo de resolución de la formulación original P_{DDT}.disminuye y es posible la resolución exacta de problemas que con la formulación anterior no se conseguía en un tiempo razonable.

1.3.2. Tratamiento discreto con tres índices

Esta segunada alternativa de fromulación se denota por P_M y trata de evitar el uso de variables binarias con cuatro índices. Montoya et al. ([27]) introducen un nuevo modelo basado en variables binarias con, a lo más, entres índices. Estas variables especifican por separado el instante de tiempo en el que empieza a trabajar un empleado en una tarea y la habilidad que utiliza en la tarea.

Así, las variables de decisión son:

- $I^{at} \in \{0,1\}$: toma el valor 1 si la tarea $a \in A$ empieza en el tiempo $t \in \{ES^a, ..., LS^a\}$.
- $x_w^{at} \in \{0,1\}$: toma el valor 1 si el empleado $w \in W_a$ empieza la tarea $a \in A$ en el instante t.
- y_{wr}^a toma el valor 1 si el empleado $w \in W_a$ utiliza la habilidad $r \in R_a$ para ejecutar la tarea $a \in A$.

La función objetivo del modelo sigue siendo la misma que 1.6a.

Las restricciones con las nuevas variables siguen imponiendo que cada empleado empiece una tarea en un único instante temporal,

$$\sum_{t=ES_a}^{LS_a} x_w^{at} = 1 \quad \forall a \in A, \forall w \in W_a. \tag{1.8a}$$

Las siguientes restricciones garantizan que cuando un empleado inicia una tarea, ha de utilizar exactamente una y solo una de las habilidades requeridas por la tarea. En otro caso no utiliza ninguna y las variables toman todas el valor 0,

$$\sum_{t=ES_a}^{LS_a} x_w^{at} = \sum_{r \in R_w \cap R_a} y_{wr}^a \quad \forall a \in A \setminus \{|A|\}, \forall w \in W_a. \tag{1.8b}$$

Al igual que se ha indicado en el modelo anterior 1.6 con las restricciones 1.6b y 1.6e, se debe cumplir que cuando un empleado esté asignado a una tarea comenzará a trabajar en ella en su fecha de inicio,

$$x_w^{at} + 1 \geq I^{at} + \sum_{r \in R_w \cap R_a} y_{wr}^a \quad \forall a \in A \setminus \{|A|\}, \forall w \in W_a, \forall t \in \{ES_a, ..., LS_a\}. \tag{1.8c}$$

Cuando un empleado esté asignado a una tarea, equivale a que desarrolle una única habilidad en esta y por tanto: $\sum_{r \in R_w \cap R_a} y_{wr}^a = 1$, lo que implicará que $x_w^{at} = 1$ cuando $I^{at} = 1$.

Las restricciones 1.6d garantizan que a todas las tareas se le asignan el número de recursos con las habilidades necesarias para la ejecución de una tarea, lo que se consigue con las siguientes restricciones,

$$\sum_{w \in W_r} y_{wr}^a = c_r^a \quad \forall a \in A \setminus \{|A|\}, \forall r \in R_a. \tag{1.8d}$$

Los empleados solo pueden utilizar una habilidad por u.t. Esto se garantiza sustituyendo las restricciones1.6f del modelo 1.6 por,

$$\sum_{a \in A_w} \sum_{\tau = max\{ES_a, t - p_\alpha + 1\}}^{min\{LS_\alpha, t\}} x_w^{a\tau} \leq 1 \quad \forall w \in W, \quad \forall t \in T. \tag{1.8e}$$

Al igual que en el modelo P_{DT} 1.6, de Pritsker et al. ([30]), en la formulación 1.8 del modelo P_M de Montoya et al.([27]) se utiliza el conjunto de restricciones 1.6c para las relaciones de precedencia entre tareas, o bien las restricciones 1.7, las cuales generan un nuevo problema con una formulación desagregada denominado P_{MDDT}.

1.3.3. Tratamiento discreto combinado

El tercer modelo de tratamiento discreto fue propuesto por Almeida et al. ([1]) y selecciona las mejores características de los dos modelos anteriores. Este modelo se denota por P_N y utiliza las mismas variables de decisión que la formulación 1.8 del modelo P_M.

En el modelo P_N 1.6a también es la función objetivo, las restricciones 1.6b garantizan que cada tarea empiece solo una vez, las restricciones 1.6c que las relaciones de precedencia entre tareas se verifican, las restricciones 1.6e consiguen que cada empleado empiece a trabajar en la tarea en la que ha sido asignado en su fecha de comienzo, las restricciones 1.8b y 1.8d aseguran que las habilidades que requiere cada tarea para ser realizada estén completas por los empleados asignados a ellas y la 1.8e restringe la asignación de un empleado a una tarea en cada período de tiempo como máximo.

La formulación P_N también se puede desagregar sustituyendo las restricciones 1.6c por la restricción 1.7 creando un nuevo modelo denotado como P_{NDDT} el cual disminuye los tiempos de resolución del problema.

1.3.4. Tratamiento continuo

En 2012 Almeida et al. ([11]) proponen un modelo de programación entera mixta que utiliza, además de las variables binarias asociadas a la secuenciación de tareas sin relación de precedencia y a la asignación de recursos a tareas, otras variables de tipo continuo para representar el tiempo en que comienza cada tarea del proyecto. Este modelo se conoce con el nombre de Problema en Tiempo Contínuo: P_{CT}, en inglés *Continuous Time Problem*.

El tiempo computacional de resolución de los modelos con tratamiento discreto del tiempo tienen una gran dependencia de la magnitud del horizonte temporal. Por ello la propuesta de un modelo con variables continuas que especifiquen el tiempo en el que comienza cada tarea. De este modo, el tiempo resolución del modelo pasa a depender del número de tareas, de las relaciones de precedencia entre ellas, del número de recursos y del número de habilidades, como demuestran en el trabajo de Almeida et al. ([1]).

En este modelo se definen las siguientes variables de decisión:

- $I^a \geq 0$ variable contínua que indica el comienzo de la tarea $a \in A$.
- $q_{a\alpha} \in \{0,1\}$: toma el valor 1 si la tarea a termina antes de que comience la α, donde $\alpha \notin P_a$.
- $y^a_{wr} \in \{0,1\}$: toma el valor 1 si el empleado $w \in W_a$ contribuye con la habilidad $r \in R_a$ en la tarea $a \in A$.

La función objetivo del problema, al igual que en el resto de modelos, consiste en minimizar el tiempo de comienzo de la tarea ficticia, que será igual al de finalización de la última tarea real:

$$f(x) = min \quad I^{|A|}. \tag{1.9a}$$

Al igual que en los modelos anteriores una tarea podrá comenzar dentro de su ventana temporal, de manera que no empezará ni antes que la fecha de inicio más temprana ni después de la fecha de inicio más tardía,

$$ES^a \leq I^a \leq LS^a \quad \forall a \in A. \tag{1.9b}$$

Las restricciones han de garantizar las relaciones de precedencia entre tareas,

$$I^a \geq I^\alpha + d^\alpha \quad \forall a \in A, \forall \alpha \in P_a. \tag{1.9c}$$

Se establece un orden entre tareas sin relaciones de precedencia,

$$I_a \geq I_\alpha + d_\alpha - M(1 - q_{\alpha a}) \quad \forall a \in A, \forall \alpha \notin P_a. \tag{1.9d}$$

El valor de $q_{\alpha a}$ tomará el valor 1 cuando las tareas a y α no tengan relación de precedencia pero se decida realizar la tarea α antes que la a por lo que se tendrá que el tiempo de comienzo de la tarea a será

mayor o igual que el de finalización de α. En caso de haber decidido que la tarea α no va a realizarse antes que la a: $q_{\alpha a} = 0$, no habrá restricción para el comienzo de a.

Si se decide realizar la tarea α antes que la a es imposible realizar la a antes que la α, por lo que $q_{\alpha a} = 1 \Rightarrow q_{a\alpha} = 0$, lo que significa que el sumatorio de estas dos variables será 1 como máximo,

$$q_{\alpha a} + q_{a\alpha} \leq 1 \quad \forall a \in A, \forall \alpha \in P_a. \tag{1.9e}$$

Se pueden realizar tareas simultáneamente si no superan el número de empleados disponibles. Además, en este caso un empleado solo podrá participar en una de ellas,

$$\sum_{r \in R_w \cap R_\alpha} y^{\alpha}_{wr} + \sum_{r \in R_w \cap R_a} y^{a}_{wr} \leq q_{\alpha a} + q_{a\alpha} + 1 \quad \forall w \in W, \forall a \in A_w, \forall \alpha \in P_a. \tag{1.9f}$$

En el caso en el que dos tareas a y α se realicen simultáneamente ambas variables binarias serán igual a cero: $q_{\alpha a} = 0$ y $q_{a\alpha} = 0$.

El número de empleados seleccionados entre los capacitados para realizar la tarea a deberá de ser igual que el requerimiento de recursos para realizar dicha tarea,

$$\sum_{w \in W_a} y^{a}_{wr} = c^{a}_{r} \quad \forall a \in A \setminus \{|A|\}, \forall r \in R_w \cap R_a. \tag{1.9g}$$

Cada empleado contribuye como mucho con una habilidad a cada tarea,

$$\sum_{r \in R_w \cap R_a} y^{a}_{wr} \leq 1 \quad \forall w \in W, \forall a \in A_w. \tag{1.9h}$$

1.3.5. Comparación Modelos

En el trabajo de Almeida et al. ([1]) se introduce una colección de problemas para comparar la efectividad de los modelos anteriores. Estos problemas están diseñados para ver cómo las diferencias en las características del sistema influyen en la complejidad del problema y los tiempos de resolución. En particular se definen tres índices:

- Factor de habilidad (SF, del ingles *Skill Factor*): Cociente entre la media de habilidades o competencias requerida por cada tarea y el número total de habilidades.

- Complejidad de la red (NC, del inglés *Network Complexity*): Número medio de tareas precedentes en la red de precedencia.

- Fuerza de recursos modificada (MRS, del inglés *Modified Resource Strength*): Cociente entre el número de empleados y el número total de unidades de empleados necesarios para llevar a cabo todas las tareas.

Todos los problemas se resolvieron con CPLEX indicando un tiempo máximo de resolución de 5 horas. La de esta experiencia computacional fue que el modelo en tiempo continuo proporciona mejores resultados, en el sentido que obtiene un mayor número de problemas resueltos a optimalidad en las 5 horas, un 83 % de los problemas. En relación con el tiempo de resolución fue el segundo más rápido.

El número de soluciones óptimas obtenidas por los modelos en tiempo discreto fue muy similar, entre un 65 % para el modelo P_{DT} y un 67 % para el modelo P_{DDT}.

Por último, dentro de los modelos discretos, los más rápidos en encontrar solución fueron los modelos P_{MDDT} y P_{NDDT}, además de ser dos de los que proporcionaron el menor número de problemas en los que el modelo no era capaz de proporcionar una solución factiblee.

1.3.6. Variables temporales discretas

A continuación se describen conjuntos de variables binarias que se clasifican según la característica que indica la variable binaria ([31]).

Variables pulsadas

Si se considera S_a como el instante de inicio de cada una de las tareas $a \in A$ del proyecto podemos saber el estado que se encuentra cada una de ellas el cual depende del instante actual t y del instante en el que se empieza la ratea S_a. El hecho de que una tarea empiece en S_a quiere decir

- La tarea comienza en S_a si y solo si: $I^{aS_a} = 1$.
- La tarea está en progreso si se cumple que: $t > S_a$ y $t < S_a + d^a$.
- La tarea termina en el instante de tiempo t si y solo si : $t = S_a + d^a$.

Al igual que la variable pulsada de comienzo I^{at}, hay variables pulsadas que indican la finalización de tareas, F^{at} y toman el valor 1 únicamente si la tarea a termina en el instante t.

Variables escalonadas

Una variable escalonada de comienzo, sea ξ^{at} a partir del instante en el que se empieza una determinada tarea. Esto es, dada una tarea a, si $I^{aS_a} = 1$, ξ^{at} tal que $t < S_a$ será 0 y ξ^{at} tal que $t \geq S_a$ será 1. Podemos ver una representación gráfica de esta variable en la Figura 1.8.

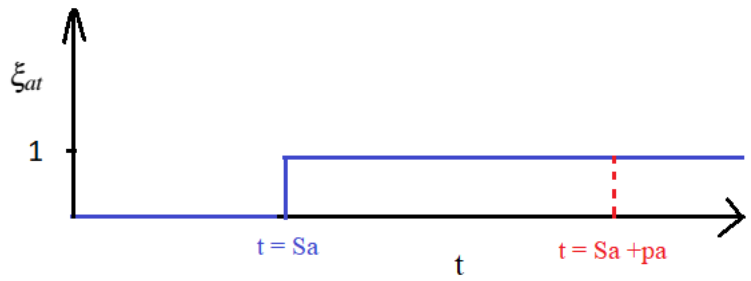


Figura 1.8: Representación gráfica de la variable de decisión binaria escalonada.

Si se conoce el valor de las variables escalonadas, el valor de la variable pulsade se puede obtener como sigue:

$$\begin{cases} \xi^{a(t-1)} = 0 \\ \xi^{at} = 1 \\ \xi^{a(t+1)} = 1 \end{cases} \Rightarrow I^{at} = \xi^{at} - \xi^{a(t-1)} \tag{1.10}$$

De esta manera, la transformación inversa es: $\xi^{at} = \sum_{\tau=0}^{t} I^{at}$, esto permite formular todos los modelos introducidos utilizando variables escalonadas de comienzo en lugar de variables pulsadas de inicio. Además se pueden añadir otras restricciones:

- Cada tarea tiene que empezar antes que su último instante de inicio:

$$\xi^{aLS^a} = 1 \quad \forall a \in A. \tag{1.11a}$$

- Si la tarea ha empezado en un instante t, habrá empezado también en cualquier instante posterior a este:

$$\xi^{at} - \xi^{a(t-1)} \geq 0 \quad \forall a \in A, \forall t \in T. \tag{1.11b}$$

- Cada tarea no puede empezar antes que su instante de inicio más temprano:

$$\xi^{at} = 0 \quad \forall a \in A, t \in \{0, ...ES^a - 1\} \tag{1.11c}$$

- Cada tarea empezará en un instante interior a sus cotas temporales:

$$\xi^{at} \in \{0,1\} \quad \forall a \in A, t \in \{ES^a - 1, ..., LS^a\} \tag{1.11d}$$

Variables de encendido/apagado

Las variables vinarias de encendido/apagado, O^{at} toman el valor 1 si la tarea $a \in A$ está siendo ejecutada en el instante t.

De esta forma, si una tarea $a \in A$ empieza en el instante S_a , en el instante $S_a + d^a$ habrá acabado, por lo que $O^{at} = 1, t \in \{S_a, ..., S_a + d^a\}$. Una representación gráfica para ver la relación entre la variable escalonada y la variable de encendido se proporciona en la Figura 1.9. Así, la relación de estas variables con las anteriores es:

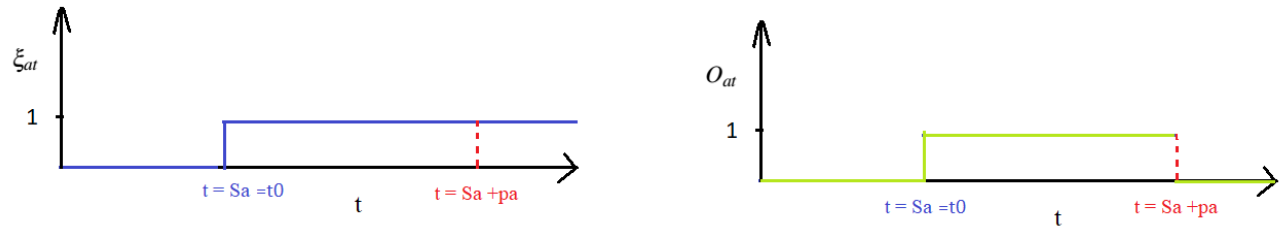


Figura 1.9: Representación gráfica de las variable de decisión binaria escalonada y de encendido/apagado.

$$\begin{cases} O^{a(S_a-1)} = 0 \\ O^{aS_a} = 1 \\ O^{a(S_a+d^a)} = 1 \\ O^{a(S_a+d^a+1)} = 0 \end{cases} \Rightarrow O^{at} = \xi^{at} - \xi^{a(t-d^a)} \tag{1.12}$$

$$O^{at} = \sum_{\tau=t-d^a+1}^{t} I^{a\tau}. \tag{1.13}$$

1.3.7. Comparación de modelos según las variables binarias de decisión

Koné et al. muestran una comparación experimental en su artículo ([19]), que compara las cotas inferiores obtenidas en la relajación lineal de las formulaciones de los modelos según las variables binarias utilizadas. Cuanto mayor sea el valor de la cota inferior, más próxima estará la solución obtenida de la solución óptima del problema. En ([19]) se pone de manifiesto la importancia de disponer de ventanas temporales lo más precisas posibles para la ejecución de las tareas, ya que estos modelos son muy sensibles al horizonte temporal. Koné et al. ([19]) señalan "*El hecho de acotar las posibles soluciones del índice temporal y que estas cotas estén bien determinadas nos ayuda a mejorar el tiempo de resolución y el análisis de nuestros resultados*. En el tratamiento discreto del tiempo tiene un gran relevancia el cálculo de cotas temporales que reduzcan la dimensión del espacio en el que se pueden mover las variables binarias utilizadas en los modelos de optimización matemática y esto contribuye a disminuir el tiempo de resolución del problema.

En el artículo [19] primero se reduce el horizonte temporal global, T_{MAX}, con un algoritmo heurístico desarrollado por Kolisch([17]) el cual encuentra un límite en el tiempo de finalización de todas las posibles últimas tareas y establece un límite global que cumpla que cualquier solución factible del problema la fecha de finalización del proyecto no lo supere.

Después de encontrar ese límite temporal, denominado T_M, se empieza con una ventana temporal de $[0, T_M]$ para cada tarea $a \in A$. A partir del conjunto de tareas precedentes y la duración de cada una de las tareas de este conjunto, se reduce la ventana temporal establecida inicialmente de cada tarea a.

Así mostró que la formulación basada en variables pulsadas de comienzo y su correspondiente disagregada eran las que mejor límites inferiores relajados obtienen y mejores soluciones en menor tiempo de compilación.

Además, Almeida et al. ([1]) resaltan la importancia de acotar en el problema tanto en la fecha de iniciación como en la de finalización de las tareas del proyecto.

Sistejkovic ([35]) presenta una revisión bibliográfica extensa de los distintos métodos heurísticos implementados para establecer cotas temporales. La primera característica que deben cumplir estas cotas es que sean verificadas por al menos una solución óptima del problema. Esto conlleva demostrar que existe alguna solución óptima del problema cuya planificación verifica los instantes de inicio y finalización establecidos por las cotas.

Por otra parte, las cotas temporales son más efectivas cuanto menos amplio es el intervalo entre cotas más tempranas y más tardías, ya que esto indica menor rango de factibilidad del índice tiempo y por tanto mayores restricciones para los valores de las variables de decisión.

Una primera forma trivial de calcular una cota para el horizonte temporal del problema puede ser el tiempo que cuesta realizar todas las tareas secuencialmente. Sin embargo, esta cota es muy pobre, ya que precisamente lo que se persigue para hacer un uso eficiente de los recursos es considerar la superposición de tareas en el tiempo.

Almeida et al.([11]) proponen calcular la longitud del camino más largo entre dos nodos de la red de precedencia. Sean i y j dos nodos de la red, se denota por $\mu(i,j)$ esta longitud máxima. Estos autores introducen algunos resultados conocidos en la gestión de proyectos que permiten proporcionar una cota superior válida del valor óptimo del problema de planificación. Esta cota se denota por UB.

A continuación, se incluyen algunos de estos resultados, todos para el problema RCPSP:

- Una tarea j no puede comenzar antes del instante dado por $\mu(0,j)$.
- Cuando una tarea j termina se ha de esperar al menos $\mu(j,|A|)$ u. t. hasta que el proyecto finalice. Por tanto, la tarea j no podrá empezar después de $UB-\mu(j,|A|)$.

Así se tiene que:

i) $ES_j = \mu(0,j) \Rightarrow EF_j = \mu(0,j) + p_j$

ii) $LS_j = UB - \mu(j,n+1) \Rightarrow LF_j = UB - \mu(j,n+1) - p_j$

Estos resultados se ilustran en la Figura 1.10

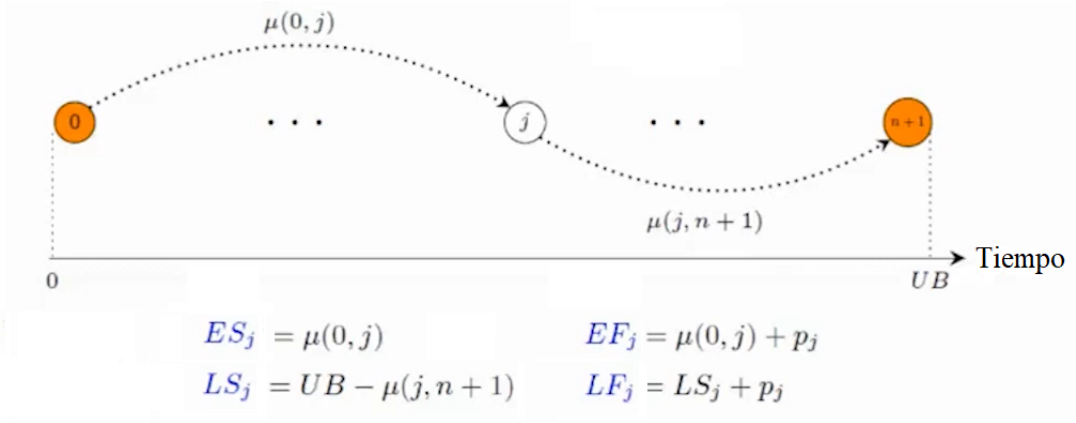


Figura 1.10: Visualización de la longitud del camino más largo entre dos nodos para establecer las ventanas temporalesde una tarea [16].

A lo largo de los últimos años se han realizado numerosos estudios sobre el cálculo de estas cotas, como los realizados por Bruker y Knust([8]). A continuación, se presenta un resumen de los dos métodos más populares e incluidos en el manual de Schwindt y Zimmerman ([31]):

Método constructivo:

Estas cotas se obtienen resolviendo el problema con algunas restricciones relajadas. Para el problema RCPSP, por ejemplo, una cota inferior se obtiene relajando las restricciones de los recursos e imponiendo solo restricciones de precedencia entre tareas. Esta cota mínima se denomina LB_0

Una segunda cota inferior propuesta por Stinson y Davis ([34]) aplica el siguiente procedimiento:

Dada una tarea $a \in A$, debe finalizarse antes de su fecha de finalización tardia LF^a y empieza como pronto después de su fecha de iniciación temprana ES^a, lo que significa que si se cumple que su duración es menor que este intervalo: $d_a < LF_a - ES_a$ el problema no será factible y deberemos aumentar la

ventana temporal en $d_a^+ = d_a - (LF_a - ES_a)$ unidades.

Este cálculo se realiza para cada tarea del proyecto y finalmente se obtiene otra cota mínima: $LB_S = LB_0 + max(d_a^+)$

Método destructivo:

Estos métodos, están basados en estudiar las soluciones factibles del problema e ir reduciendo el intervalo temporal. El procedimiento aplica los siguientes pasos:

1- Se resuelve el problema en el tiempo máximo dado T_{MAX}.

2- Si el problema es factible, se escoge como número de intervalos temporales $T_{MAX} = tI^{|A|t}$ y ya habremos terminado el problema.

3- Si el problema es no factible, se aumenta el límite temporal en una unidad y se itera hasta encontrar una solución factible.

1.4. Contribución del TFM

La realización de las prácticas del Máster en una empresa consultora que desarrolla proyectos de software ha permitido conocer el funcionamiento de una empresa de este tipo y, en particular, cómo se realiza la planificación y asignan las tareas a los empleados cuando se ha de trabajar de forma simultánea en varios proyectos con fechas de inicio y finalización. La disponibilidad de un caso de estudio real es una ventaja porque proporciona una motivación clara al establecer un problema de interés que hay que resolver y los resultados se podrán validar al comparar si la solución propuesta introduce una mejora en el sistema. Como inconveniente, todos los aspectos ligados a un caso real que precisa de un conocimiento profundo y preciso de los objetivos y restricciones particulares del mismo. Por un lado, al tratarse de un problema muy específico, las hipótesis habituales asumidas en los modelos propuestos en la literatura pueden no verificarse. Por otro lado, los parámetros de estos modelos pueden no conocerse en el caso de estudio o la información no ser completa.

Una primera contribución del TFM es la revisión de la literatura por un abanico muy amplio de modelos matemáticos para la planificación de proyectos que incorporaran alguna de las características del sistema real. Una vez elegidos se han formulado utilizando una notación común que ha facilitado identificar las diferencias entre las formulaciones propuestas y determinar cuál o qué combinación de ellas se ajusta mejor al caso de estudio. En la planificación de proyectos es esencial el tratamiento del tiempo. Por ello, tiene especial interés conocer más de un tratamiento de la evolución temporal, para de nuevo, elegir el más adecuado. En esta revisión se ha presentado cuatro tratamientos discretos del tiempo y uno continuo, junto con tres tipos de variables binarias que se diferencian en la característica temporal que expresan. Además de estas alternativas en la modelización, otro aspecto de gran relevancia en relación con los tiempos computacionales de resolución es el cálculo de ventanas temporales o cotas en el tiempo para la ejecución de las tareas. Este primer capítulo ha incluido en los apartados anteriores un resumen de la revisión realizada.

La segunda contribución es el modelo matemático propuesto en el segundo capítulo de la memoria. Este modelo se ha diseñado adhoc para el caso de estudio e incluye aspectos de varios modelos de planificación de la literatura, ya que ninguno de los revisados incluía todas las características específicas de los proyectos y modo de operación de la empresa. Por ello, en primer lugar se ha descrito con detalle el

tipo de proyectos que afronta el departamento, las habilidades o roles de los empleados y modo habitual de trabajo. Las decisiones sobre cómo modelar el problema ha tomado en consideración los objetivos propuestos por la empresa. Un primer objetivo, a largo plazo, es el cálculo de una planificación base en un plazo de varios meses de un número amplio de proyectos que guíe el calendario del departamento con una distribución horaria semanal del personal. A corto plazo, el objetivo de la empresa es que los empleados del departamento completen su jornada laboral semanal con una clara definición de las tareas que han de completar.

El modelo matemático propuesto determina la planificación base requerida por la empresa. Esta planificación proporciona una primera estimación sobre las posibles fechas de entrega de cada uno de los proyectos, los empleados asignados a cada tarea de cada proyecto con una distribución horaria semanal. A partir de esta planificación, semana a semana se puede ver modificada atendiendo a los imprevistos que influyen en el calendario semanal de los empleados, como algunas bajas, vaciones, peticiones de última hora o algún incumplimiento en la ejecución de los plazos. Por ello, el modelo matemático se puede ejecutar semanalmente actualizando los datos de los parámetros del modelo.

La tercera contribución se refiere al cálculo de cotas temporales. En el tratamiento discreto del tiempo tiene un gran relevancia el cálculo de cotas temporales que reduzcan la dimensión del espacio en el que se pueden mover las variables binarias utilizadas en los modelos de optimización matemática y esto contribuye a disminuir el tiempo de resolución del problema. En el segundo capítulo se ha descrito el procedimiento desarrollado adhoc para el caso de estudio. Un primer análisis considera los proyectos de forma individual y calcula la duración de cada proyecto cuando todos los empleados del departamento están disponibles a jornada completa para trabajar exclusivamente en el proyecto. De esta forma se obtiene una estimación del horizonte temporal de cada proyecto según su fecha de inicio y obtener cotas inferiores en las fehcas de finalización de cada proyecto. A partir de estos resultados, se realiza un segundo análisis cuando se consideran varios proyectos. En este caso, las fechas de finalización se retrasan, ya que los empleados se distribuyen entre las tareas de los distintos proyectos, lo que modifica el valor de las cotas temporales mínimas y máximas de finalización para cada uno de los proyectos.

Todo el procedimiento se ha implementado en el lenguaje de programación Python y puede ser utilizado por la empresa para realizar planificaciones futuras, de ahí que se pueda considerar como otra contribución de este TFM. En el tercer capítulo de la memoria se muestran los resultados obtenidos al aplicar el modelo matemático y los procedimientos para el cálculo de cotas temporales.

Con la formulación y resolución del modelo matemático en el caso de estudio se comprueba que se han alcanzado los objetivos planteados inicialmente y especificados en 1.1. El paso de una tarea manual como la que se realiza por parte del responsable de la planificación, a un resultado automatizado y optimizado teniendo en cuenta la casuística del departamento, supone para la empresa una mejora significativa de los tiempos y la calidad de la planificación. Además se ha mejorado la herramienta de planificación de proyectos actual incluyendo nuevas visualizaciones basadas en los resultados del trabajo. Los resultados obtenidos han derivado en otras mejoras tales como las planteadas en los objetivos secundarios concluyendo así, con satisfacción, el resultado del presente trabajo para la empresa.

Capítulo 2

Caso estudio: Planificación de proyectos en una empresa de servicios

2.1. Descripción de los proyectos, tareas y roles de empleados

El primer paso en el tratamiento de un problema consiste en la descripción detallada del mismo. En este caso de estudio implica una descripción de las características de un departamento de una empresa de servicios que ha de planificar el desarrollo de varios proyectos asignando para ello los recursos necesarios. Además, es preciso identificar todas la restricciones que han de verificarse y los criterios que el departamento utiliza para decidir entre varias planificaciones posibles. De ahora en adelante, al Problema de Planificación de Proyectos del caso de estudio se denota por P3D.

Los proyectos que se abordan desde el departamento se clasifican en cuatro tipos según si el presupuesto y/o las tareas a desarrollar están fijadas en el momento de contratar el proyecto y si es requerida o no la presencia del cliente:

- Cerrado: Proyectos para los que existe un presupuesto fijado y una especificación del alcance del proyecto completamente determinada.
- Bolsa: Proyectos para los que existe un presupuesto fijado en bolsa de horas. A priori no se conocen las tareas exactas a realizar, dado que el cliente las va indicando durante la duración del contrato.
- Mantenimiento: Proyectos para los que no se conocen todas las tareas exactas a realizar y no hay un presupuesto fijado, sino una tarifa por hora. Cada mes se facturan las horas consumidas.
- Formación: Proyecto acotado y predefinido en cuanto al alcance. La planificación es establecida por el cliente, ya que se requiere su disponibilidad y presencia durante todo el desarrollo del proyecto.

En el caso de estudio de este TFM se han considerado exclusivamente proyectos cerrados. En estos proyectos las tareas a realizar están claramente definidas. En general, hay tres tipos de tareas que deben realizarse de forma secuencial, lo que implica una relación de precedencia entre ellas. La primera tarea es de análisis y diseño del proyecto, la segunda tarea se encarga de la construcción del proyecto y la tercera tarea consiste en la puesta en producción del proyecto. En general, la tarea 1 requiere el 20 % de las horas estimadas de duración de un proyecto, la tarea 2 el 60 % y la tarea 3, el 20 % restante.

En la Figura 2.1 se incluye el código numérico de 1 a 7 de cada una de las tareas posibles en un proyecto cerrado. Se ha añadido el tipo 7, tarea Dummy, que es una tarea ficticia y que al igual que en los modelos de la literatura permite indiciar el inicio y final de un proyecto. De forma similar, en la

Figura 2.2 se incluye el código numérico de 1 a 6 de los roles posibles en el departamento.

id_tar	Tarea
1	Análisis y diseño
2	ConstrucciónAI
3	ConstrucciónBI
4	Puesta en producción
5	Formación
6	Mantenimiento
7	Dummy

Figura 2.1: Distintas tareas con sus identificadores

En los departamentos de software es habitual que los empleados del departamento posean distintas competencias que les hagan hábiles para desempeñar distintos roles. En el caso del departamento bajo estudio los roles posibles son:

- Jefe de proyecto: Responsable de la gestión del proyecto, sus entregables y la relación con el cliente.
- Arquitecto de datos: Responsable de la definición, despliegue y mantenimiento de las herramientas de los proyectos.
- Ingeniero de datos: Responsable de la adquisición, el almacenamiento, la transformación y la gestión de los datos para el posterior tratamiento analítico.
- Consultor AI: Responsable de la definición y construcción de los modelos matemáticos de Inteligencia Artificial.
- Consultor BI: Desarrolladores de los modelos y de las visualizaciones a incorporar en los cuadros de mando.
- Formador: Responsable de impartir sesiones de capacitación sobre las diferentes tecnologías con las que se trabaja en el departamento.

Aunque todos los proyectos cerrados tengan la misma estructura en cuanto a las tareas, estos se pueden subdividir en dos clases, dependiendo de las características de la segunda tarea. Los Proyectos BI, en inglés *Business Intelligence Proyects* que precisan de empleados que puedan adoptar el rol de consultor BI, y Proyectos AI, en inglés *Artificial Intelligence Proyects* que requieren de empleados que puedan adoptar el rol de consultor AI.

En la Figura 2.3 se describe de forma más detallada para cada tarea de un proyecto cerrado los roles de los empleados que pueden ser requeridos. La primera tarea de análisis y diseño del proyecto solo puede ser ejecutada por un Jefe de proyecto. El resto de tareas pueden ser ejecutadas por varios empleados, aunque cada empleado solo puede desempeñar un rol para cada tarea y es el Jefe del proyecto quien lo determina. En el departamento del caso de estudio, los empleados con el rol de Formador solo realizan proyectos de tipo Formación, por ello no se van a considerar en el caso de estudio que está centrado en Proyectos cerrados.

id	Roles
1	Jefe de proyecto
2	Consultor IA
3	Consultor BI
4	Arquitecto de datos
5	Ingeniero de datos
6	Formador

Figura 2.2: Roles con sus respectivos identificadores

id_tar	Tarea	id_rol	Rol
1	Análisis y diseño	1	Responsable de proyecto
2	ConstrucciónAI	1	Responsable de proyecto
2	ConstrucciónAI	3	Consultor IA
3	ConstrucciónBI	1	Responsable de proyecto
3	ConstrucciónBI	2	Consultor BI
4	Puesta en producción	1	Responsable de proyecto
4	Puesta en producción	4	Arquitecto de datos
4	Puesta en producción	5	Ingeniero de datos
5	Formación	1	Responsable de proyecto
5	Formación	6	Formador
6	Mantenimiento	1	Responsable de proyecto
6	Mantenimiento	4	Arquitecto de datos
7	Dummy	1	Responsable de proyecto

Figura 2.3: Relación Tarea-Rol

Las distintas tareas pertenecientes a los proyectos cerrados se pueden definir de la siguiente forma:

- Análisis y diseño del proyecto: Primera tarea en ejecutarse que abarca el 20 % de las horas estimadas al proyecto total aproximadamente y solo puede ser ejecutada por el Jefe de proyecto.
- Tarea de Construccion: Abarca generalmente el 60 % de las horas totales estimadas a un proyecto y puede ser asignada a todo el personal que sea capaz de/ apto para desempeñar el rol de consultor BI o consultor AI.
- Puesta en producción: Tarea que abarca el 20 % de las horas estimadas, llevada a cabo por un arquitecto o ingeniero de datos.

2.2. Descripción del funcionamiento en la planificación de proyectos cerrados

Para asegurar la confidencialidad y atendiendo a la política de privacidad de la empresa, a partir de los datos de proyectos reales se han transformado los valores numéricos de forma que mantienen la estructura original de los datos y se han etiquetado con nombres ficticios a los proyectos y empleados.

El departamento está formado por 12 empleados. En la primera columna de la Figura 2.4 se ha incluido el código numérico que identifica cada empleado. En la segunda columna se muestra el nombre del empleado y en la tercera columna un valor numérico entre 1 y 3, que representa un factor de rendimiento. Cada tarea requiere un número de horas que son estimadas. Este factor permite calcular para cada empleado, según su experiencia, el número de horas que precisará para realizar la tarea. Por ejemplo, un empleado recién incorporado al equipo y que no disponga de experiencia previa tendrá asignado un factor 3, el cual indica que, en caso de realizar una determinada tarea él solo, necesita tres veces las horas estimadas inicialmente para la ejecución de la tarea.

Por otra parte, cada empleado puede desempeñar uno o varios roles, dependiendo principalmente de su formación previa, aunque el nivel de experiencia o años trabajados en la empresa también pueden permitir la adquisición de nuevos roles. En la última columna de la Figura 2.4 se muestra para cada empleado los roles que puede desempeñar. De los 12 empleados, 8 empleado solo tienen un rol y los otros 4, pueden desempeñar dos roles. Solo hay un empleado con el rol de Arquitecto de datos (4) y con el rol de Consultor AI (2).

En la organización actual del departamento, el Jefe de proyecto divide el proyecto en tareas e indica los roles que han de asignarse para realizar cada tarea.

En la Figura 2.5 se muestra la información relativa a 19 proyectos desarrollados en el departamento. En la primera columna se incluye el identificador numérico del proyecto y en la segunda columna su título. La tercera columna muestra el tipo de proyecto, de AI o BI. A continuación, se indica la fecha de inicio del mismo y el número estimado de horas totales de dedicación de un empleado con factor 1. La duración final de un proyecto dependerá del número de empleados y de su factor de rendimiento asignados a un proyecto.

Además se incluye una representación de una planificación real de 5 proyectos cerrados en la Figura 2.6, donde se puede apreciar que las tareas de un mismo proyecto se realizan secuencialmente, aunque hay algunas fechas de solapamiento entre algunas tareas consecutivas de los proyectos. Además, en el modelo planteado se han unifiacado las dos primeras tareas de cada proyecto en una sola, ya que el encargado de realizarlas es el mismo Jefe de proyecto.

id	Nombre	Factor
1	Eloy Salcedo	1.5
2	Alex Esteban	1.0
3	Raúl Baigorri	1.0
4	Ricardo Canudas	1.0
5	Antonio Gonzalez-Aller	1.0
6	Axier Iñiguez	1.5
7	Ignacio Tortajada	3.0
8	Adrián Lopez	1.0
9	Mario Cereza	3.0
10	Samuel Lacueva	2.0
11	Adrián Martínez	3.0
12	Javier Baigorri	1.5

id_per	id_rol
1	3
1	1
2	3
2	1
3	1
4	1
5	5
5	4
6	2
7	3
8	6
9	3
10	3
11	3
12	3
12	5

Figura 2.4: Datos de los empleados del departamento

La planificación de los proyectos del departamento persigue un objetivo a largo plazo que consiste en desarrollar el mayor número de proyectos, para ello le interesa minimizar el tiempo de ejecución de los proyectos. A medio plazo, la planificación de un conjunto de proyectos, asegura que todos los empleados del departamento tienen unas tareas asignadas cada semana para completar su jornada laboral.

Cualquier planificación que se considere ha de verificar las directrices establecidas por el departamento que se resumen en los siguientes puntos:

D1 La planificación ha de minimizar el tiempo de entrega del conjunto de proyectos proporcionados por el departamento, esto es, minimizar la fecha de entrega del último proyecto en finalizar. De esta forma se pueden incorporar cuanto antes nuevos proyectos.

D2 Se ha de establecer una fecha de finalización para cada proyecto:

El departamento debe realizar una estimación de la fecha de entrega de cada proyecto para poder negociar su precio con el cliente. En la planificación del departamento se proporciona una fecha de entrega para cada proyecto con la asignación de los empleados, el rol que desempeñan y las horas de dedicación semanales al proyecto. Esta planificación proporciona una guía para que el equipo cumpla la fecha estimada de finalización del proyecto.

D3 Se busca una continuidad en las tareas asignadas a un empleado y que en cada una de las tareas trabaje el menor número de empleados posible, estableciendo un máximo de cuatro empleados por tarea.

D4 Se deben cumplir las jornadas laborables de cada empleado. La disponibilidad de un empleado depende de si es contratado, personal en prácticas o con jornada reducida. En el caso de estudio todos los empleados tienen un contrato de jornada completa.

D5 Cada proyecto tiene asignado un Jefe responsable, el cual realiza su análisis y diseño pudiendo participar en otras tareas si ‘posee el rol requerido.

D6 Es conveniente que cada empleado esté asignado a más de un proyecto cada semana:

id_proy	Titulo	Fecha inicio	Horas totales	TIPO
1	PROYECTO 1GENERAL	2021-07-01	75	AI
2	PROYECTO 2GENERAL	2021-08-01	280	AI
3	PROYECTO 3GENERAL	2021-09-01	150	AI
4	PROYECTO 4GENERAL	2021-10-01	600	AI
5	PROYECTO 5GENERAL	2021-09-01	310	AI
6	PROYECTO 6GENERAL	2021-07-01	100	BI
7	PROYECTO 7GENERAL	2021-08-01	100	BI
8	PROYECTO 8GENERAL	2021-07-15	134	BI
9	PROYECTO 9GENERAL	2021-08-15	55	BI
10	PROYECTO 10GENERAL	2021-07-01	50	BI
11	PROYECTO 11GENERAL	2021-06-30	250	BI
12	PROYECTO 12GENERAL	2021-08-01	280	BI
13	PROYECTO 13GENERAL	2021-07-15	30	BI
14	PROYECTO 14GENERAL	2021-08-31	80	BI
15	PROYECTO 15GENERAL	2021-09-01	760	BI
16	PROYECTO 16GENERAL	2021-09-01	100	BI
17	PROYECTO 17GENERAL	2021-09-01	500	BI
18	PROYECTO 18GENERAL	2021-09-15	34	BI
19	PROYECTO 19GENERAL	2021-08-15	1250	BI

Figura 2.5: Proyectos del departamento

Aunque es preferible la continuidad de los empleados en una misma tarea hasta su ejecución completa, es deseable que todos los empleados tengan una tarea alternativa para dedicar horas en caso de que el proyecto en el que esté trabajando una semana determinada se paralice. Por ello, hay que encontrar el equilibrio entre la continuidad en una tarea y la dedicación a varias tareas cada semana sin provocar el estrés del empleado.

Utilizando todos estos datos se realiza una primera aproximación a largo plazo del tiempo de finalización de los distintos proyectos a trabajar en un número determinado de semanas, de horas a dedicar a cada tarea, del grupo y Jefe implicados y semanas seleccionadas para trabajar en cada proyecto.

Esta aproximación servirá al responsable de departamento para hacerse una idea aproximada de como van a ir evolucionando los proyectos en el tiempo y una vez realizada esta aproximación se irá viendo el desarrollo de los distintos proyectos realizando modificaciones a la planificación original dependiendo de las distintas necesidades e imprevistos que puedan surgir.

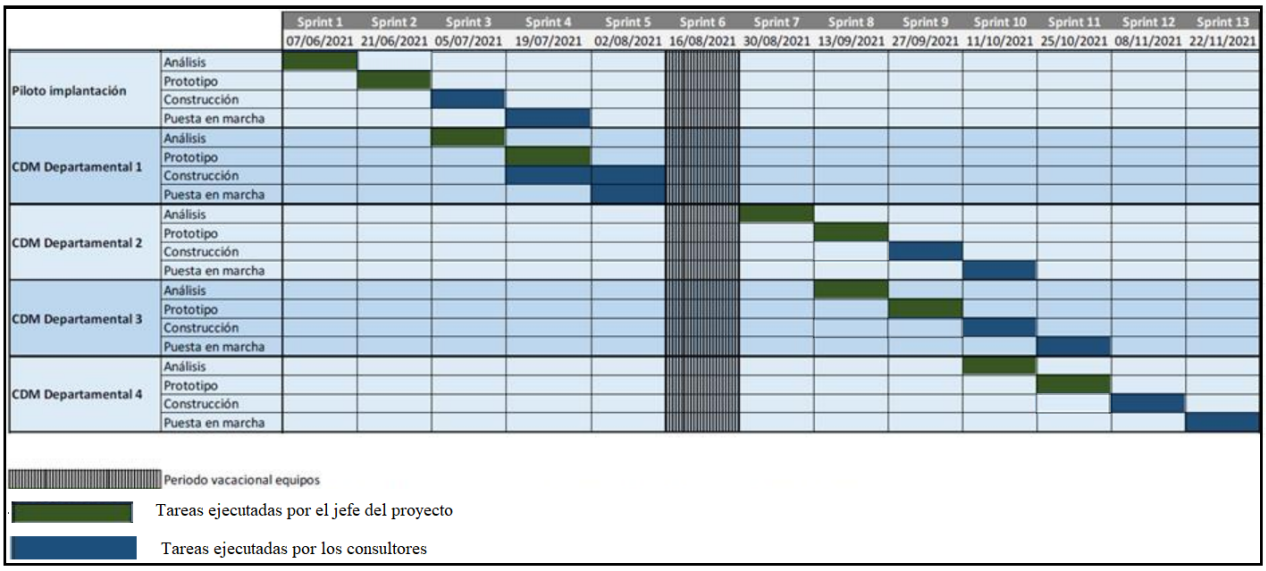


Figura 2.6: Planificación de 5 proyectos

id_proy	id_tar	Horas	id_rol
1	1	15	1
1	3	45	2
1	4	15	5
1	7	0	1
2	1	40	1
...	...	...	...
18	7	0	1
19	1	50	1
19	2	150	3
19	4	50	5
19	7	0	1

Figura 2.7: Desglose de proyectos en tareas, horas estimadas por tarea y rol correspondiente para realizar la tarea

2.3. Tratamiento del tiempo y definición de las variables

En la revisión de la literatura, una de las principales diferencias entre los modelos es el tratamiento del tiempo y el tipo de variables de decisión que se introducen para representar la evolución temporal. El análisis de las diferentes alternativas y las características del tipo de proyectos del departamento para su conduce al modelo matemático formulado más adelante.

Una de las primeras diferencias del P3D con los modelos revisados en la literatura es que el número de recursos asignados a una misma tarea es una variable de decisión. Para cada tarea se conoce la du-

ración en horas requeridas de un empleado con factor 1 y el conjunto de tareas precedentes. El número de horas que requiere una tarea depende del número de empleados asignados como en el modelo Multi-Modal1.3. Sin embargo, en el P3D la duración de una tarea depende además del factor del empleado y del número de horas semanales que invierte cada empleado.

Una segunda diferencia es que no se asume que todos los empleados empiecen a trabajar en una tarea en el momento en que se inicia, ni que deban invertir horas en ella en todos los instantes temporales desde su inicio hasta su finalización. Aunque se toma como unidad temporal la semana se debe especificar la hora en la que se termina cada tarea para que ningún empleado pueda trabajar en una tarea antes de terminar las tareas precedentes. Para ello se define una nueva variable entera que especifica la hora de finalización de cada tarea que es igual al número de horas que dedica el empleado que más horas invierte en esa tarea esa semana.

La falta de un un parámetro fijo que especifique la duración de cada tarea provoca que no haya una relación única entre los instantes de inicio de dos tareas consecutivas. Por tanto, el modelo matemático ha de incorporar las variables necesarias para controlar cada semana las horas invertidas en cada tarea de cada proyecto y establecer la finalización de la tarea cuando se hayan cubierto las horas estimadas. En particular, va a implicar la definición de una variable binaria asociada a cada instante de tiempo y cada empleado, que indique si un empleado está implicado en una tarea de un determinado proyecto en ese instante de tiempo, y otra variable contínua que indique el número de horas que dedica a cada tarea en la que está asignado. El hecho de que la duración de cualquier tarea sea una variable de decisión más implica que no es posible utilizar el modelo con tratamiento del tiempo contínuo ni las formulaciones desagregadas de los modelos en tiempo discreto vistas en el apartado 1.3 del Capítulo1.

Otra diferencia del resto de modelos vistos en el Capítulo 1 es que no se utiliza una tarea ficticia para determinar el comienzo de la primera tarea de cada proyecto debido a la introducción de variables binarias tanto de inicio como de finalización de las tareas de todos los proyectos.

La relación cardinal entre habilidad y tarea es uno a uno. Esto es, cada empleado debe ser capaz de asumir una único rol para resolver cada tarea, por lo que se utiliza una formulación discreta semejante al modelo discreto P_{DT} utilizado por Pritsker et al. ([30]).

Tras considerar un tratamiento discreto del tiempo, se van a introducir variables temporales de dos tipos. Una variable de tiempo escalonada determina el inicio de una tarea y toma el valor 1 a partir del instante en el que se inicia la tarea. La finalización de un proyecto se representa a través de una variable pulsada de finalización. Esta variable binaria toma el valor 1 cuando ya se han invertido el total de horas requerido por una tarea. El uso de variables pulsadas facilita la formulación de algunas restricciones del modelo.

Además, la asignación de empleados a tareas en cada instante temporal se realiza utilizando variables de encendido/apagado, que indican si una tarea se está ejecutando, esto es, que se ha iniciado y aún no ha finalizado.

2.4. Formulación matemática del P3D

En la formulación matemática del P3D se definen algunos conjuntos ya introducidos en la revisión de la literatura del Capítulo 1.

- W: Conjunto de los empleados del departamento.

- R: Conjunto de roles. El máximo rol es el de Jefe de proyecto que se define como el primer rol, esto es, $r = 1$.
- A: Conjunto de tareas posibles a realizar en un proyecto. La última tarea, $a = |A|$ es ficticia y requiere 0 horas. Esta tarea identifica cuándo finaliza un proyecto.
- $R_w \subset R$: Conjunto de roles del empleado $w \in W$.
- $W_r \subset W$: Conjunto de empleados con rol $r \in R$. Con esta notación, el conjunto de Jefes de proyecto es W_1.
- $R_a \subset R$: Conjunto de roles que pueden desarrollar la tarea a, $a \in A$. Se define $R_{|A|} = 1$, esto es, la tarea ficticia siempre ha de ser desarrollada por un Jefe de proyecto. En todo proyecto interviene un único Jefe de proyecto. Si la tarea ficticia $|A|$ se pudiera asignar a otro rol, cabe la posibilidad de que se asigne a otro empleado que no desarrolle ninguna otra tarea del proyecto y que sea miembro del equipo del proyecto sin invertir horas en él.
- $A_r \subset A$: Conjunto de tareas que pueden desarrollar el rol r, $r \in R$.
- B: Conjunto de proyectos.
- $A_b \subset A$: Conjunto de tareas a desarrollar en el proyecto $b \in B$. Las actividades son secuenciales, es decir, sea $A_b = \{a_b^1, \cdots, a_b^{|A_b|} = |A|\}$ entonces si $i < j$ entonces la tarea a_b^j no se inicia hasta que la tarea a_b^i se haya finalizado.
- T: Conjunto de semanas a la vista para establecer la planificación.

Los parámetros que describen las características de los proyectos, las tareas y los empleados:

- L_w: Horas laborables semanales del trabajador $w \in W$.
- γ_w: Factor de rendimiento del empleado $w \in W$.
- H_b^{ar}: Horas necesarias en el proyecto b de la actividad a por un empleado con rol r y un factor de rendimiento igual a 1, $b \in B$, $a \in A_b$, $r \in R_a$.

  Si un proyecto b incluye la actividad $a \in A$ se proporciona las horas necesarias de cada rol r que puede realizar dicha tarea. Si un determinado proyecto no requiere de un determinado rol para esa tarea se introduce $H_b^{ar} = 0$.
- E_b: Primera semana en la que se puede iniciar el proyecto $b \in B$.
- C_b: Última semana en la que puede finalizar el proyecto $b \in B$.

Las variables de decisión:

- $z_{wb}^a \in \{0, 1\}$: toma el valor 1 si el empleado w realiza la tarea a del proyecto b, $b \in B$, $a \in A_b, w \in W_a$.
- $x_{wb}^{at} \in \{0, 1\}$: toma el valor 1 si el empleado w realiza la tarea a del proyecto b en la semana t, $b \in B, a \in A_b, w \in W_a, t \in T$.
- u_{wb}^{at}: Número de horas que invierte el empleado w en la tarea a del proyecto b durante la semana t, $b \in B, a \in A_b, w \in W_a, t \in T$.
- $y_{wb} \in \{0, 1\}$: toma el valor 1 si el empleado w está implicado en el proyecto b, $b \in B, w \in W$.
- H_w^t: Número de horas trabajadas por el empleado w en la semana t, $w \in W, t \in T$.

- M_b^{at}: Número de horas que invierte el empleado que más dedicación le presta a la tarea a del proyecto b la semana t, $b \in B, a \in A_b, t \in T$.
- $I_b^{at} \in \{0, 1\}$ toma el valor 1 si la tarea a del proyecto b ha empezado a realizarse en la semana t o en una semana anterior, $b \in B, a \in A_b, t \in T$.
- $O_b^{at} \in \{0, 1\}$ toma el valor 1 si la tarea a del proyecto b está en transcurso en la semana t, $b \in B, a \in A_b, t \in T$.
- $F_b^{at} \in \{0, 1\}$ toma el valor 1 si la semana t se finaliza la tarea a del proyecto b, $b \in B, a \in A_b, t \in T$.
- F Variable entera que indica la fecha en la que se finaliza el último proyecto.

La función objetivo del P3D consiste en minimizar el tiempo de finalización de todos los proyectos:

$$f(x) = \text{mín} \qquad F \tag{2.1a}$$

A continuación, se presentan las restricciones agrupadas según el tipo de limitación que introducen.

Definir el instante de tiempo en el que han finalizado todos los proyectos:

$$F_b^{|A|t} \leq F \qquad b \in B, t \in T \tag{2.1b}$$

Asignación de un empleado a un proyecto:

Un empleado está asignado a un proyecto si realiza alguna de sus tareas. Una cota válida para el número de tareas en las que puede participar es el número de tareas del proyecto, $|A_b|$:

$$\sum_{a \in A_b} z_{wb}^a \leq |A_b| y_{wb} \qquad w \in W, b \in B. \tag{2.1c}$$

$$\sum_{a \in A_b} z_{wb}^a \geq y_{wb} \qquad w \in W, b \in B. \tag{2.1d}$$

$$z_{wb}^a \leq y_{wb} \qquad b \in B, a \in A_b, w \in W_a. \tag{2.1e}$$

Un empleado asignado a un proyecto en alguna tarea, ha de trabajar al menos una semana en esa tarea,

$$z_{wb}^a \leq \sum_{t \in T} x_{wb}^{at} \qquad b \in B, a \in A_b, w \in W_a \tag{2.1f}$$

$$x_{wb}^{at} \leq z_{wb}^a \qquad b \in B, a \in A_b, w \in W_a \tag{2.1g}$$

En cada tarea de un proyecto a lo más participan cuatro empleados,

$$\sum_{w \in W} z_{wb}^a \leq 4 \qquad b \in B, a \in A_b \tag{2.1h}$$

Cada empleado no puede estar asignado a más de 3 proyectos en una misma semana. SEn esta restricción se puede aumentar el valor de 3, si la planificación considera un número elevado de proyectos,

$$\sum_{b \in B, a \in A_b} x_{wb}^{at} \leq 3 \qquad w \in W, t \in T \tag{2.1i}$$

Un empleado puede desempeñar una tarea solo si tiene el rol correspondiente,

$$x_{wb}^{at} = 0 \qquad b \in B, a \in A_b, R_w \cap R_a = \emptyset, t \in T \tag{2.1j}$$

Todas las tareas de un proyecto han de estar asignadas al menos a un empleado,

$$\sum_{w \in W} z_{wb}^{a} \geq 1 \qquad b \in B, a \in A_b \tag{2.1k}$$

En cada proyecto se asigna uno y solo un Jefe de proyecto:

$$\sum_{w \in W_1} y_{wb} = 1 \qquad b \in B \tag{2.1l}$$

La última tarea de todos los proyectos se asigna a un Jefe de proyecto,

$$\sum_{w \in W_1} z_{wb}^{|A|} = 1 \qquad b \in B \tag{2.1m}$$

$$\sum_{w \in W \setminus W_1} z_{wb}^{|A|} = 0 \qquad b \in B \tag{2.1n}$$

La dedicación en horas de los empleados a las tareas:

Un empleado invierte horas una semana en una tarea solo si está asignado a dicha tarea esa semana. Además se garantiza que al menos trabaje durante un día completo en dicha tarea,

$$u_{wb}^{at} \leq L_w x_{wb}^{at} \qquad b \in B, a \in A_b, w \in W_a, t \in T \tag{2.1ñ}$$

$$u_{wb}^{at} \geq 8 x_{wb}^{at} \qquad b \in B, a \in A_b, w \in W_a, t \in T \tag{2.1o}$$

El total de horas trabajadas a la semana por un empleado se define como la suma de horas trabajadas esa semana en todas las tareas que le han asignado y esta suma no superará la jornada laboral de cada empleado,

$$\sum_{b \in B, a \in A_b} u_{wb}^{a} = H_w^t \qquad w \in W, t \in T \tag{2.1p}$$

$$\sum_{b \in B, a \in A_b} u_{wb}^{a} \leq L_w \qquad w \in W, t \in T \tag{2.1q}$$

La suma de todas las horas dedicadas por todos los empleados a una tarea, corregidas por el factor de rendimiento del empleado, ha de ser al menos el número de horas que requiere dicha tarea para ser completada:

$$\sum_{w \in W_r, t \in T} \frac{u_{wb}^{at}}{\gamma_w} \geq H_b^{ar} \qquad b \in B, a \in A_b, r \in R_a \tag{2.1r}$$

La secuenciación de las tareas:

Una tarea acaba cuando todos los empleados asignados terminan su trabajo. Para determinar este instante, se define para cada tarea y semana, el máximo de horas de dedicación entre los empleados asignados a la tarea,

$$M_b^{at} \geq u_{wb}^{at} \qquad b \in B, a \in A_b, w \in W_a, t \in T \tag{2.2a}$$

Un trabajador asignado a una tarea a_b^i de un proyecto b no puede dedicar horas a dicha tarea hasta que no finalice la tarea anterior del proyecto. Si esta finaliza la semana t, lo hará justamente en la hora $M_b^{a_b^{i-1}t}$, o el día $\frac{M_b^{a_b^{i-1}t}}{8}$ y las horas que le podrá dedicar esa semana a la siguiente actividad serán las horas laborables restantes: $40 - M_b^{a_b^{i-1}t}$.

$$M_b^{a_b^i t} + \sum_{\alpha \in A_b \setminus \{a_b^i\}} u_{wb}^{a_b^{i+1}t} \leq L_w \qquad b \in B,\, a_b^i \in A_b \setminus \{|A|\},\, w \in W_a,\, t \in T \tag{2.2b}$$

Cuando tenemos varios proyectos tendremos en cuenta la implicación de los empleados en cada proyecto. Las horas dedicadas por un trabajador a cada tarea de cada proyecto una misma semana no pueden sobrepasar sus horas laborables y debe respetar la secuenciación de tareas,

$$M_b^{at} + \sum_{\alpha_1 \in A_b \setminus \{a\}} u_{wb}^{\alpha_1 t} + \sum_{\beta \in B \setminus \{b\}, \alpha_2 \in A_\beta} u_{w\beta}^{\alpha_2 t} \leq L_w \qquad b \in B,\, a \in A_b,\, w \in W_a,\, t \in T \tag{2.2c}$$

Una tarea empieza a partir de la semana en la que se puede iniciar,

$$I_b^{at} = 0, \qquad b \in B,\, a \in A_b, t < E_b \tag{2.2d}$$

Si una tarea empieza la semana t, la semana siguiente ya habrá empezado, por lo que su variable indicadora de inicialización toma el valor 1 a partir de dicha semana,

$$I_b^{at} \leq I_b^{a(t+1)}, \qquad b \in B,\, a \in A_b,\, t > E_b, t \in T \tag{2.2e}$$

La semana de inicio de una tarea es siempre igual o posterior a las semanas de inicio de las tareas precedentes,

$$I_b^{at} \geq I_b^{(a+1)t}, \qquad b \in B,\, a \in A_b \setminus \{|A|\},\, t \in T \tag{2.2f}$$

Como las tareas se realizan secuencialmente, una tarea está en proceso una semana dada t si se ha empezado con esa tarea alguna semana anterior a t y todavía no ha empezado con la tarea siguiente:

$$O_b^{at} = I_b^{at} - I_b^{(a+1)t}, \qquad b \in B,\, a \in A_b \setminus \{|A|\}, t \in T. \tag{2.2g}$$

La actividad ficticia de finalización de un proyecto no requiere procesarse,

$$O_b^{|A|t} = 0, \qquad b \in B, t \in T \tag{2.2h}$$

Las tareas solo finalizan una vez y la semana ha de ser anterior a la semana de finalización límite proyecto. Esta estará entre la ventana temporal de la fecha de finalización mínima considerando que solo tenemos ese proyecto en el departamento y que todos los empleados se pueden dedicar a él exclusivamente y entre una fecha de finalización máxima o más tardía que la calcularemos mediante un algoritmo construido después de analizar varios ejemplos en los que teníamos proyectos coincidentes en el tiempo,

$$\sum_{t=C_b^{Min}}^{C_b^{Max}} F_b^{at} = 1, \qquad b \in B,\, a \in A_b \tag{2.2i}$$

$$\sum_{t=0}^{C_b^{Min}} F_b^{at} = 0, \qquad b \in B,\, a \in A_b \tag{2.2j}$$

$$\sum_{t=C_b^{Max}}^{T_{Max}} F_b^{at} = 0, \qquad b \in B, a \in A_b \tag{2.2k}$$

Una tarea finaliza cuando se han completado todas sus horas y comienza la siguiente tarea del mismo proyecto,

$$F_b^{(a-1)t} = I_b^{at} - I_b^{a(t-1)}, \qquad b \in B, a \in A_b \setminus \{a_b^1\}, t \in T \tag{2.2l}$$

La tarea ficticia terminará la misma semana que la última tarea real,

$$F_b^{(|A_b|-1)t} = F_b^{(|A_b|)t}, \qquad b \in B, t \in T \tag{2.2m}$$

Una tarea se puede asignar a un empleado una semana si en esa semana está en proceso o finaliza:

$$x_{wb}^{at} \leq O_b^{at} + F_b^{at}, \qquad b \in B, a \in A_b \setminus \{|A|\}, t \in T \tag{2.2n}$$

2.5. Cotas temporales para P3D

En el cálculo de cotas temporales se realiza un primer análisis de los proyectos de forma individual, suponiendo que todo el departamento de software está disponible para trabajar en en el proyecto. De este forma, se calcula una estimación del horizonte temporal de cada proyecto dependiendo de su fecha de inicio. Los resultados obtenidos se expresan como función de las horas estimadas de cada tarea y sirven para establecer cotas en el P3D. Se denota por C_b^{Min} la fecha de finalización más temprana para el proyecto $b \in B$.

Se considera un proyecto $b \in B$ y todos los empleados disponibles para ser asignados a las tareas de b.

Antes de presentar la realización de estimaciones para las fechas de finalización de proyectos individuales se expone una descripción de las variables que se utilizan en su cálculo:

- SFT_a: Variable que indica la semana en la que finaliza la tarea $a \in A$.
- SDT_a: Variable que indica la duración en semanas de la tarea $a \in A$.
- HET_a: Variable que indica las horas estimadas para la ejecución de la tarea $a \in A$ de un proyecto dado, independientemente de su clase.
- HSL: Variable que indica las horas semanales laborables de los empleados.
- HES_w: Variable que indica las horas semanales efectivas del empleado $w \in W$.
- HT_a: Variable que indica las horas reales que se van a dedicar para la ejecución de la tarea $a \in A$.
- FL_w: Variable que indica el factor de rendimiento del empleado $w \in W$.
- EF_{AI}: Estimación de la fecha de finalización de un proyecto de Inteligencia Artificial en semanas.
- EF_{BI}: Estimación de la fecha de finalización de un proyecto de Inteligencia Empresarial en semanas.

El procedimiento se inicia con la primera tarea y se elige entre los empleados con el rol necesario el que más rápido puede realizar la tarea. De esta forma la duración de la tarea se obtiene:

$$SDT_1 = \frac{HET_1}{HSL} = \frac{HET_1}{40}$$

Se continúa con las sucesivas tareas, considerando en cada caso la asignación de empleados, con el rol y factor de rendimiento correspondiente que permiten completar la tarea lo antes posible, STD_2 y STD_3. La fecha más temprana de finalización del proyecto $b \in B$:

$$C_b^{Min} = E_b + SDT_1 + SDT_2 + SDT_3$$

En el caso del P3D, el cálculo de SDT_1 y SDT_2 depende de si el proyecto es AI o BI, ya que los roles necesarios para su desarrollo son diferentes.

El algoritmo para la obtención de la fecha mínima de finalización de cada proyecto es el siguiente:

Algorithm 1 Cota mínima para la finalización de cada proyecto.

Input: Conjunto de proyectos B; Fecha de iniciación de cada proyecto E_b; Horas estimadas de las tareas de cada proyecto H_b^{ar}, $a \in A_b$;

Output: Cota mínima para la fecha de finalización de cada proyecto C_b^{Min}, $b \in B$;

1: $i \leftarrow 0$
2: Resolver el problema individualmente para cada proyecto:
3: **while** $i < |B|$ **do**
4: Resuelve el problema para el proyecto i.
5: Devuelve el resultado $C_i^{Min} \leftarrow E_i + Makespan[i]$
6: **end while**
7: **return** C^{Min};

Cuando se han de ejecutar varios proyectos simultáneamente, es preciso realizar un segundo análisis que modifique las cotas temporales obtenidas con el procedimiento temporal anterior aplicado a cada proyecto de forma individual. Notemos que, la duración de un proyecto aumenta cuando se ha de desarrollar varios proyectos en paralelo.

La cotas temporales de los proyectos dependen de la duración estimada de cada proyecto y de los otros proyectos que el departamento ejecuta de forma paralela. En el Anexo B se incluye el código en Python del algoritmo implementado para su cálculo. Los datos de entrada se refieren a los proyectos y los empleados y se consideran conocidas las cotas temporales de cada proyecto cuando se trata de forma individual, C_b^{Min}.

Seguidamente, con las fechas más tempranas de cada proyecto, E_b, y la duración mínima, $C_b^{Min} - E_b$ de cada uno se realiza un análisis de los proyectos cuyas ventanas temporales intersecan y así determinar para cada proyecto el número de proyectos con los que coincide en el tiempo y el número de semanas en las que coincide. Con toda la información se amplia la ventana temporal del proyecto. Una descripción más detallada se incluye en el Algoritmo 2.

Algorithm 2 Cota máxima para la finalización de cada proyecto.

Input: Conjunto de proyectos B; Fecha de iniciación de cada proyecto E_b; Duración minima o Makespan de cada proyecto $Makespan_b$, $b \in B$; Cota mínima para la fecha de finalización de cada proyecto C_b^{Min}, $b \in B$; Clase de cada proyecto $Class_b$, $b \in B$; Estimación Horaria para cada tarea de cada proyecto H_b

Output: Cota máxima para la fecha de finalización de cada proyecto C_b^{Max}, $b \in B$;

$i \leftarrow 0$
Coincidencia en semanas del proyecto i con el resto:
while $i < |B|$ **do**
  $Coinciden \leftarrow 0$
  $j \leftarrow 0$
  while $j < |B|$ **do**
    if $(j \neq i)$ & $(Class_j = Class_i)$ **then**
      if $(C_i^{Min} < E_j) \mid (E_i > C_j^{Min})$ **then**
        $Suma \leftarrow 0$
      else
        if $E_i < E_j$ **then**
          if $C_i^{Min} < C_j^{Min}$ **then**
            $Suma \leftarrow C_i^{Min} - E_j$
          end if
          if $C_i^{Min} \geq C_j^{Min}$ **then**
            if $Class_i =$'AI' **then**
              $Suma \leftarrow \frac{H_j^{21} \times 1.5}{40}$
            else
              $Suma \leftarrow \frac{H_j^{12}}{66.67}$
            end if
          end if
        end if
        if $E_i \geq E_j$ **then**
          if $C_i^{Min} > C_j^{Min}$ **then**
            $Suma \leftarrow C_j^{Min} - E_i$
          end if
          if $C_i^{Min} \leq C_j^{Min}$ **then**
            if $Class_i =$'AI' **then**
              $Suma \leftarrow \frac{H_i^{21} \times 1.5}{40}$
            else
              $Suma \leftarrow \frac{H_i^{12}}{66.67}$
            end if
          end if
        end if
      end if
    else
      $Suma \leftarrow 0$
    end if
    $Coinciden \leftarrow Coinciden + Suma$
  end while
  $C_i^{Max} \leftarrow C_i^{Min} + Coinciden$
end whilereturn C^{Max};

Capítulo 3

Caso de estudio: Resolución y conclusiones

3.1. Introducción

Tras analizar la solución general, se describen en los siguientes apartados los análisis intermedios realizados para calcular las cotas temporales de proyectos individuales y en paralelo, del mismo y diferente tipo.

Dado el modelo matemático, para su aplicación en la empresa es preciso utilizar un solucionador de programación matemática de alto rendimiento que calcule las soluciones al problema en cada instancia en un tiempo computacional razonable. Dado el contexto de aplicación del modelo, es necesario transferir las soluciones al entorno de trabajo del departamento para que las soluciones se examinen de forma rápida y lo más fácilmente posible. Por ello, es importante elegir el entorno tecnológico que facilite la creación del modelo y la visualización de los resultados. La justificación de esta decisión se incluye en el siguiente apartado.

A continuación, se muestran los resultados obtenidos para el P3D y se realiza un análisis para verificar que se cumplen las restricciones impuestas, en particular, los requerimientos dados por los responsables de la planificación del departamento a la hora de establecer los calendarios de los empleados.

Una vez analizada la solución general se incluyen los análisis intermedios necesarios para fijar las cotas temporales de la solución general.

3.2. Entorno Tecnológico

3.2.1. Python

El problema teórico planteado en el apartado 2.4, es un problema de programación matemática entera mixta, por tanto se ha de elegir un programa que sea capaz de calcular la solución óptima en un tiempo razonable. Este programa se puede ejecutar utilizando un lenguaje de programación que adapte las instancias al formato adecuado. En este TFM se ha elegido Python.

Los principales motivos por los que se decide utilizar este lenguaje es porque es un lenguaje de código abierto, que utiliza la empresa para el desarrollo de sus proyectos y dispone de un gran número de paquetes con herramientas para la creación de modelos de programación lineal, desde paquetes libres como Pulp, Pyomo, Scipy y otros comerciales como CPLEX y Gurobi.

Por otra parte, Python es uno de los tres lenguajes de programación más utilizados a lo largo de los últimos años, apareciendo en el informe de 2021 de Octoverse realizado por GitHub en segundo lugar, por debajo de JavaScript [28] y se espera además que en el resultado para el presente año Python ya supere a JavaScript como principal lenguaje de programación:

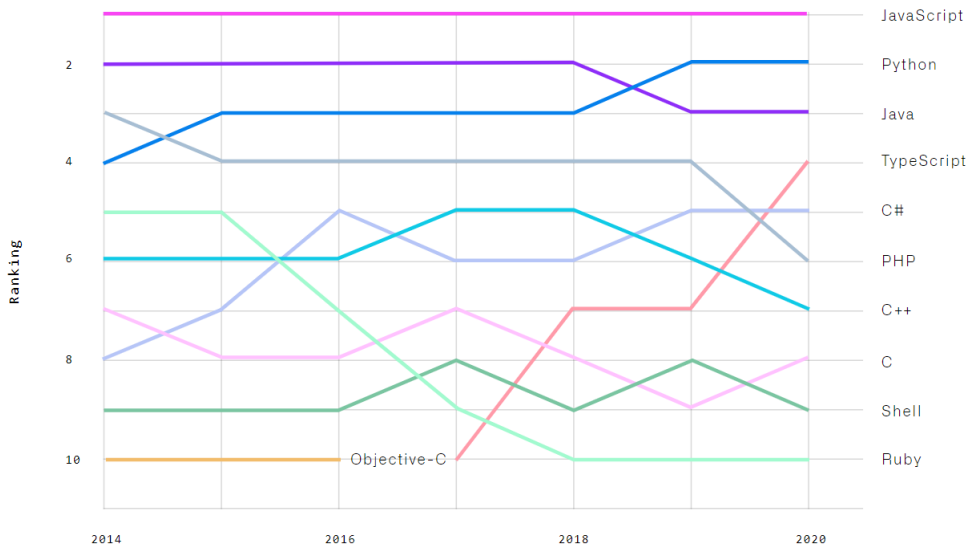


Figura 3.1: Gráfica de la evolución de la posición de los lenguajes más populares en los últimos años [28].

Este lenguaje ofrece código abierto utilizando una sintaxis sencilla y fácil de entender [3]. También cuenta con frameworks de gran calibre e incorpora un gran abanico de bibliotecas, las que hacen que tenga un gran número de aplicaciones, desde el análisis de datos y la inteligencia artificial hasta el desarrollo de páginas web.

Aunque el lenguaje incorpora un gran número de paquetes con herramientas para la creación de modelos de programación lineal, en este TFM se utiliza la librería que incorpora Python de Programación Lineal Mixta, Python MIP [25], que permite utilizar simultáneamente variables continuas y enteras. Esta librería está profundamente integrada con las bibliotecas C del solver de Branch-and-Cut CBC y CLP de COIN-OR, el cual es el recurso abierto más rápido actualmente [26].

Para la ejecución y el entendimiento de todos los pasos previos que se realizan hasta la ejecución del modelo se utiliza la aplicación web de código abierto *Jupyter Notebook*. Este entorno permite el dla introducción de los datos que dan valor a los parámetros del modelo y su clasificación, la creación de conjuntos y funciones que se utilizan para la resolución del problema y la visualización de los resultados.

Esta aplicación posibilita crear documentos o cuadernos que permiten incorporar elementos de código con texto narrativo, con el que podemos facilitar la explicación y proporcionar documentación. Además es capaz de mostrar resultados utilizando medios enriquecidos como Latex, PNG, HTML y adjuntos al código que los genera, lo que la hacen muy apropiada para utilizarla en la memoria.

En el caso de la resolución de P3D y los problemas correspondientes al procedimiento desarrollado para el cálculo de cotas temporales se utiliza IDLE, en inglés *Integrated DeveLopment Environment for Python*, el cual es un entorno gráfico de desarrollo elemental que permite editar y ejecutar programas en Python.

3.2.2. Power BI

Los resultados obtenidos por el modelo de optimización necesitan ser fácilmente interpretados por los responsables del área de Data & AI y así que los empleados puedan conocer a simple vista cuáles son las tareas a ejecutar cada semana.

Para facilitar toda esta gestión, se utiliza como herramienta de visualización de datos el servicio de creación de cuadros de mando Microsoft Power BI. Con esta herramienta, se genera una plataforma en la que se apoyan los empleados para realizar el seguimiento semanal de los proyectos y de las tareas.

Algunas de las características principales de este visualizador son:

- Permite la conexión a un gran número de orígenes de datos, que pueden ser modelizados de forma simplificada para posteriormente poder realizar un análisis de los mismos.

  En este TFM, los orígenes de datos son ficheros Excel, que registran los datos de los resultados del modelo de optimización exportados, los cuales son la planificación general del proyecto en semanas dividida por tareas y el desglose en horas asignadas a cada empleado por semana divididas a su vez en tareas de distintos proyectos.

- Facilita la creación de informes altamente visuales, los cuales pueden publicarse y ser compartirlos a través de Web y en dispositivos móviles. Dentro del alcance del presente TFM se han desarrollado distintas páginas de analítica con diferentes objetivos dependiendo del usuario consumidor (empleado, responsable, etc.)

- Creación de paneles personalizados al alcance de todos, con una perspectiva empresarial única individualizada según la función de cada empleado. Además, se incorpora la seguridad de que cada empleado tiene acceso únicamente a sus datos, y los responsables tienen acceso a la gestión de todo el área.

- Capacidad de filtrado y personalización en la visualización de los datos, de forma que el responsable del departamento tenga acceso a la gestión de todo el área y determine las gráficas que cada empleado pueda visualizar, permitiendole visualizar únicamente los paneles en los que esté implicado para que tenga claro y se focalice en el desarrollo de los proyectos en los que participa, sus tiempos y las horas que debe invertir en cada uno de ellos.

Un cuadro de mando es una herramienta interactiva que te permite conocer en un golpe de vista la situación global actualizada a través de la visualización de los datos. Además, está compuesto por gráficas que contribuyen a visualizar el comportamiento de los indicadores. Un ejemplo de un cuadro de mando lo podemos ver en la siguiente figura:

Figura 3.2: Ejemplo de cuadro de mando.

3.3. Visualización de la planificación general de los proyectos

En este TFM se plantea transmitir claramente los resultados proporcionados al resolver el modelo matemático de planificación de proyectos y los requisitos se plantean en dos ejes:

i) Responsable de la planificación del departamento: Este perfil necesita de una visión global del conjunto de proyectos del departamento.

ii) Los empleados: Necesitan de una visión personalizada con respecto a los proyectos que les han sido asignados.

Tras la realización de distintas sesiones de trabajo para determinar cuál es la mejor forma de mostrar la información en cada caso se han construido, con la herramienta de Power BI, las siguientes visualizaciones de los resultados sobre la planificación.

La Figura 3.3 muestra la planificación general del P3D. En ella se indica, para cada uno de los proyectos y cada una de sus tareas el equipo asignado, el volumen de horas total y las fechas de inicio y finalización.

En esta se puede observar como el proyecto 4 es el último en acabar, por lo que su fecha de finalización es el valor de la función objetivo en el óptimo, la cual es el 28/04/2022 o la semana 43 en términos numéricos. Además el proyecto 17 es el último en acabar entre los pertenecientes al tipo BI, por lo que el conjunto de proyectos de Inteligencia Empresarial están finalizados para Febrero de 2022, lo que daría pie a la posibilidad de incorporar nuevos proyectos de ese tipo para esas fechas.

Titulo proyecto	Equipo de proyecto	Total horas	Fecha comienzo	Fecha fin
PROYECTO 1GENERAL				
Análisis y diseño	Alex Esteban	15	15/07/2021	29/07/2021
Construccion2	Axier Iñiguez	68	29/07/2021	12/08/2021
Puesta en producción	Javier Baigorri, Antonio Gonzalez-Aller	18	12/08/2021	12/08/2021
PROYECTO 2GENERAL				
Análisis y diseño	Alex Esteban	40	12/08/2021	12/08/2021
Construccion2	Axier Iñiguez	300	19/08/2021	28/10/2021
Puesta en producción	Javier Baigorri, Antonio Gonzalez-Aller	43	28/10/2021	02/12/2021
PROYECTO 3GENERAL				
Análisis y diseño	Ricardo Canudas	30	09/09/2021	09/09/2021
Construccion2	Axier Iñiguez	135	09/09/2021	25/11/2021
Puesta en producción	Antonio Gonzalez-Aller	30	25/11/2021	16/12/2021
PROYECTO 4GENERAL				
Análisis y diseño	Raúl Baigorri	120	07/10/2021	21/10/2021
Construccion2	Axier Iñiguez	540	28/10/2021	10/02/2022
Puesta en producción	Javier Baigorri, Antonio Gonzalez-Aller	133	10/02/2022	28/04/2022
PROYECTO 10GENERAL				
Análisis y diseño	Eloy Salcedo	15	15/07/2021	15/07/2021
Construcción	Eloy Salcedo, Adrián Martínez, Ignacio Tortajada	65	15/07/2021	22/07/2021
Puesta en producción	Javier Baigorri	15	22/07/2021	22/07/2021
PROYECTO 11GENERAL				
Análisis y diseño	Raúl Baigorri	50	08/07/2021	15/07/2021
Construcción	Adrián Martínez, Javier Baigorri, Samuel Lacueva, Mario Cereza	290	15/07/2021	19/08/2021
Puesta en producción	Javier Baigorri, Antonio Gonzalez-Aller	68	19/08/2021	23/09/2021
PROYECTO 12GENERAL				
Análisis y diseño	Alex Esteban	56	05/08/2021	30/09/2021
Construcción	Alex Esteban, Javier Baigorri, Samuel Lacueva, Mario Cereza	272	30/09/2021	14/10/2021
Puesta en producción	Javier Baigorri, Antonio Gonzalez-Aller	66	14/10/2021	11/11/2021
PROYECTO 13GENERAL				
Análisis y diseño	Eloy Salcedo	9	22/07/2021	22/07/2021
Construcción	Eloy Salcedo, Javier Baigorri	27	22/07/2021	29/07/2021
Puesta en producción	Javier Baigorri	9	29/07/2021	29/07/2021
PROYECTO 14GENERAL				
Análisis y diseño	Alex Esteban	16	09/09/2021	09/09/2021
Construcción	Alex Esteban, Javier Baigorri, Ignacio Tortajada	56	09/09/2021	30/09/2021
Puesta en producción	Javier Baigorri, Antonio Gonzalez-Aller	19	30/09/2021	21/10/2021
PROYECTO 15GENERAL				
Análisis y diseño	Alex Esteban	50	16/09/2021	21/10/2021
Construcción	Alex Esteban, Javier Baigorri, Ignacio Tortajada, Samuel Lacueva	212	21/10/2021	02/12/2021
Puesta en producción	Javier Baigorri, Antonio Gonzalez-Aller	55	02/12/2021	16/12/2021
PROYECTO 16GENERAL				
Análisis y diseño	Eloy Salcedo	30	16/09/2021	23/09/2021
Construcción	Eloy Salcedo, Adrián Martínez, Samuel Lacueva, Mario Cereza	122	23/09/2021	28/10/2021
Puesta en producción	Javier Baigorri, Antonio Gonzalez-Aller	26	28/10/2021	28/10/2021
PROYECTO 17GENERAL				
Análisis y diseño	Alex Esteban	56	09/09/2021	23/09/2021
Construcción	Alex Esteban, Adrián Martínez, Javier Baigorri, Samuel Lacueva	255	23/09/2021	11/11/2021
Puesta en producción	Javier Baigorri, Antonio Gonzalez-Aller	80	18/11/2021	13/01/2022
PROYECTO 18GENERAL				
Análisis y diseño	Alex Esteban	8	30/09/2021	30/09/2021
Construcción	Alex Esteban	18	30/09/2021	30/09/2021
Puesta en producción	Antonio Gonzalez-Aller	8	30/09/2021	14/10/2021
PROYECTO 19GENERAL				
Análisis y diseño	Eloy Salcedo	75	19/08/2021	30/09/2021
Construcción	Eloy Salcedo, Adrián Martínez, Javier Baigorri, Ignacio Tortajada	258	30/09/2021	25/11/2021
Puesta en producción	Javier Baigorri, Antonio Gonzalez-Aller	60	25/11/2021	02/12/2021

Figura 3.3: Resumen de la planificación general del departamento.

En la Figura 3.4 podemos ver la planificación de las tareas de los 9 primeros proyectos y los 6 últimos del conjunto total, en los que sus tareas vienen diferenciadas y estructuradas secuencialmente. Se presenta tal y como lo recibe el responsable de la planificación de proyectos para que disponga de una visualización general de ellos. Podemos ver como esta planificación abarca un marco temporal de 10 meses, pudiendo servir para construir una primera planificación del calendario anual de los empleados. Para esta visualización se ha seleccionado el diagrama de Gantt.

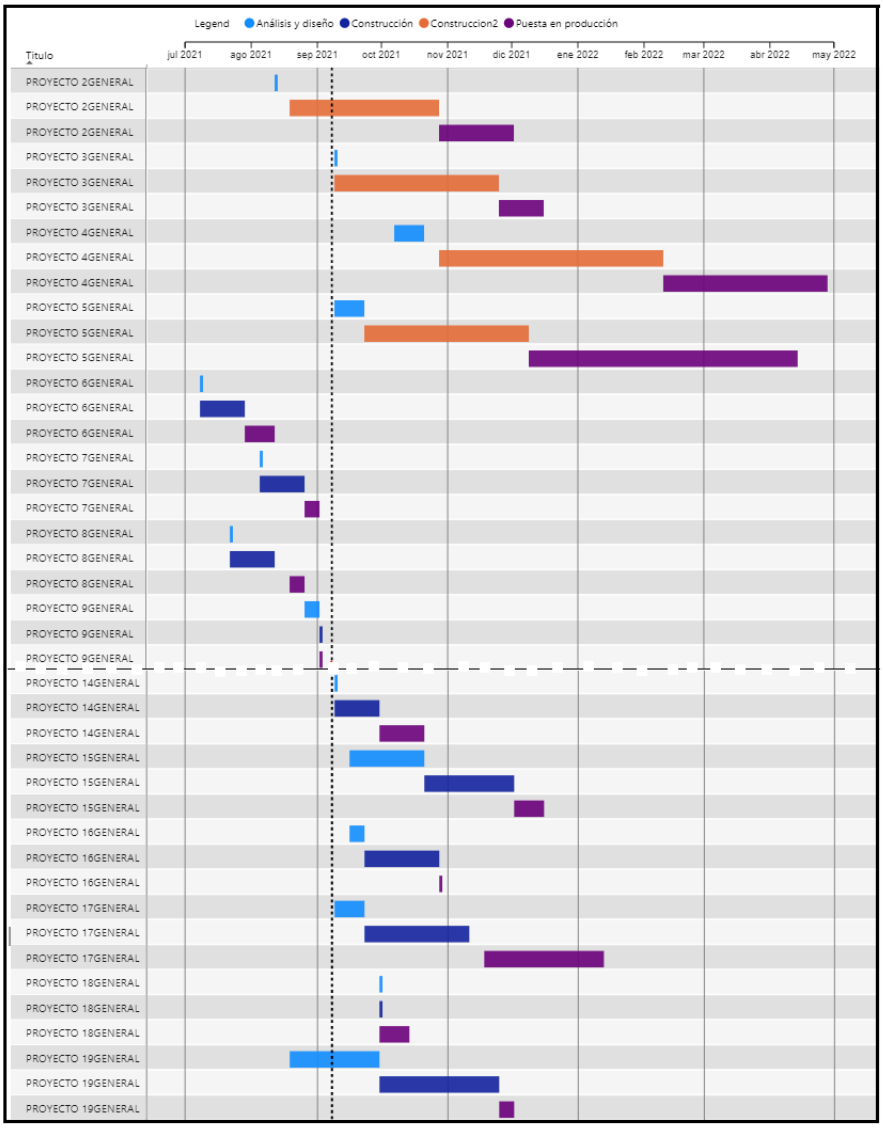


Figura 3.4: Ejemplo de cuadro de mando.

Se puede observar en la figura anterior que, aunque el último proyecto en terminar pertenezca al conjunto de Proyectos AI, el valor de la función objetivo depende de todos los proyectos, ya que existe solapamiento entre las fechas de ejecución de la tercera tarea de algunos de ellos, por lo que los empleados responsables de la puesta en producción deben repartirse el trabajo de todos ellos para finalizar lo antes posible.

Por último, la Figura 3.5 muestra información más detallada indicando para cada semana el número de horas que cada empleado va a dedicar a cada tarea. Además, la figura incorpora el número de horas totales planificadas para cada tarea finalmente, después de haber formado los equipos de los proyectos, lo que se puede comparar con el número de horas estimadas inicialmente y que, debido a los factores de rendimiento puede no coincidir.

Esta información es de gran ayuda para establecer un calendario laboral para cada empleado del departamento y la planificación de los distintos proyectos.

Titulo proyecto	Semana 10	Semana 11	Semana 12	Semana 13	Semana 15	Semana 16	Semana 17	Semana 18	Semana 19	Semana 20	Semana 22	Semana 23	Semana 27	Total
PROYECTO 14GENERAL	**24**		**48**	**8**	**11**									**91**
Análisis y diseño	**16**													**16**
Alex Esteban	16													**16**
Construcción	**8**		**48**											**56**
Alex Esteban			40											**40**
Ignacio Tortajada			8											**8**
Javier Baigorri	8													**8**
Puesta en producción				**8**	**11**									**19**
Antonio Gonzalez-Aller					11									**11**
Javier Baigorri				8										**8**
PROYECTO 15GENERAL		**8**			**11**	**57**	**48**	**18**			**120**	**55**		**317**
Análisis y diseño		**8**			**11**	**31**								**50**
Alex Esteban		8			11	31								**50**
Construcción						**26**	**48**	**18**			**120**			**212**
Alex Esteban						9	17	9			40			**75**
Ignacio Tortajada							23							**23**
Javier Baigorri						8		9			40			**57**
Samuel Lacueva						9	8				40			**57**
Puesta en producción												**55**		**55**
Antonio Gonzalez-Aller												40		**40**
Javier Baigorri												15		**15**
PROYECTO 17GENERAL	**24**	**32**	**80**		**29**	**11**	**55**	**40**	**40**	**40**			**40**	**391**
Análisis y diseño	**24**	**32**												**56**
Alex Esteban	24	32												**56**
Construcción			**80**		**29**	**11**	**55**	**40**	**40**					**255**
Adrián Martínez					29									**29**
Alex Esteban							23	8	40					**71**
Javier Baigorri			40				23							**63**
Samuel Lacueva			40			11	9	32						**92**
Puesta en producción										**40**			**40**	**80**
Antonio Gonzalez-Aller										8				**8**
Javier Baigorri										32			40	**72**

Figura 3.5: Ejemplo de cuadro de mando.

Aunque el ejemplo muestra esta información únicamente para los proyectos 14, 15 y 17, el modelo proporciona esta planificación para todo el conjunto de proyectos, lo que permite establecer un calendario detallado para cada empleado.

Las instancias proporcionadas por la empresa contienen proyectos con unas horas estimadas totales que se suelen encontrar entre las 100 y 150, aunque abarcan un rango desde 30 hasta 1250 horas estimadas. Como se ha comentado en el apartado 2.1 solo se consideran proyectos cerrados, los cuales poseen una estructura definida y similar entre todos ellos, como se ha explicado en 2.2. Estos datos van a permitir generar nuevas instancias para mostrar el funcionamiento del modelo propuesto.

3.3.1. Tiempos de ejecución de P3D

Las Figura 3.6 presenta los tiempos de ejecución obtenidos, en segundos, para la solución del P3D. Para establecer estas cotas se ha utilizado el algoritmo mostrado en el apartado 2.5 del segundo capítulo.

P3DC2	P3DC	P3D
2197	11103	30301

Figura 3.6: Tiempos de ejecución en segundos para el P3D

Para los tres problemas la solución de la función objetivo ha sido la misma, lo que quiere decir que las tres soluciones son equivalentes. Estos se diferencian dependiendo de las cotas establecidas. El problema denominado P3D corresponde al Problema de Planificaxión del Departamento sin acotar estableciendo un límite temporal de 37 semanas. En el problema denominado P3DC se han establecido las cotas obtenidas con el algoritmo presentado en el apartado 2.5 del segundo capítulo y para el problema

denominado P3DC2 se han establecido las cotas obtenidas con el algoritmo solo para los dos últimos proyectos en terminar, restringiendo la fecha de finalización del resto a la cota máxima proporcionada por el algoritmo.

3.4. Cálculo de cotas temporales en proyectos individuales

La dificultad de resolver P3D reside en la dimensión del conjunto de proyectos y en la determinación de fechas límite cuando debemos realizar varios proyectos simultáneamente, por lo que se sigue el capítulo con unos análisis de distintas instancias que han servido para determinar el algoritmo del establecimiento de estas cotas de la solución general.

3.4.1. Proyectos AI

Para realizar las estimaciones en la fecha de finalización de estos proyectos se han generado un conjunto de 10 proyectos que abarcan desde las 50 a las 600 horas, ya que los proyectos reales tienen estas caracterírticas.

La definición y características de las distintas tareas de estos proyectos se puede ver en el apartado 2.1, la cual se ha seguido para la generación de los proyectos cuyas características se muestran en la Figura 3.7.

	id_proy	Fecha inicio	Horas totales
0	1	2021-07-01	50
1	2	2021-07-01	75
2	3	2021-07-01	100
3	4	2021-07-14	150
4	5	2021-07-21	200
5	6	2021-06-30	250
6	7	2021-08-01	300
7	8	2021-07-15	400
8	9	2021-08-31	500
9	10	2021-09-01	600

Horas primera tarea	Horas Construcción	Horas produccion
10	30	10
15	45	15
20	60	20
30	90	30
40	120	40
50	150	50
60	180	60
80	240	80
100	300	100
120	360	120

Figura 3.7: Características de los proyectos de AI con el desglose en horas por tarea.

A continuación, se muestra un ejemplo de aplicación del procedimiento propuesto.

Por definición, la primera tarea siempre la realiza un único jefe de proyecto. Además, al considerar el proyecto de forma individual, para minimizar el tiempo de finalización, la tarea del diseño del proyecto la realiza el jefe de proyecto que no invierta más horas de las estimadas, es decir, un empleado que puede desempeñar el rol de jefe de proyecto con un factor de rendimiento de 1, los cuales son los empleados 2, 3 y 4, como se puede comprobar viendo la Figura 2.4 del apartado 2.2.

Así, las semanas dedicadas a la primera tarea son:

$$SFT_1 = \frac{HET_1}{HSL} = \frac{HET_1}{40}. \tag{3.1}$$

De todos los empleados de la figura 2.4 solo el 6, llamado Axier Iñiguez posee el rol 2 capaz de realizar las tareas de un proyecto AI. Este empleado tiene un factor de 1.5, lo que significa las horas que debe invertir en realizar la tarea son las horas estimadas por su factor de rendimiento,

$$HT_2 = HET_2 \times FL. \tag{3.2}$$

Por tanto, al dividir estas horas entre las horas semanales laborables se obtiene la duración de la tarea,

$$SDT_2 = \frac{HT_2}{HSL} = \frac{HET_2 \times 1.5}{40} = \frac{HET_2}{26.6}. \tag{3.3}$$

También se puede calcular el tiempo que un empleado va a necesitar dedicar a una tarea con las horas laborables efectivas semanales, las cuales serían las horas laborables entre el factor de rendimiento del empleado,

$$\frac{HT_2}{HSL} = \frac{HET_2}{HES_2} \Rightarrow HES_2 = \frac{HSL}{FL_6}. \tag{3.4}$$

Por otra parte, los empleados capacitados para realizar la última tarea son el 5 y el 12, con unos factores de 1 y 1.5 respectivamente. En este caso, al tratar con un único proyecto y disponer de ambos empleados, se puede suponer que los dos empleados empezarán a trabajar en cuanto acabe la tarea anterior, por lo que las horas efectivas semanales serán:

$$HES_3 = \frac{HSL}{FL_5} + \frac{HSL}{FL_{12}} = \frac{40}{1} + \frac{40}{1.5} = 40 + 26.6 = 66.6. \tag{3.5}$$

Así la duración de esta tarea en semanas será:

$$SDT_3 = \frac{HET_3}{HES_3} = \frac{HET_3}{66.6}. \tag{3.6}$$

Finalmente, la ecuación para establecer la fecha de finalización de un proyecto es la suma de las duraciones de las tres tareas:

$$EF_{AI} = SFT_1 + SDT_2 + SDT_3 = \frac{HET_1}{40} + \frac{HET_2}{26.6} + \frac{HET_3}{66.6}. \tag{3.7}$$

En la Figura 3.8 se ha representado para todos los proyectos AI generados la semana de finalización estimada frente a las horas estimadas de cada proyecto. Notemos que, las estimaciones coinciden con los valores obtenidos para la finalización de los proyectos de AI pudiendo utilizarse para establecer las cotas para resolver los problemas individuales de AI del departamento reduciendo al máximo el tiempo de ejecución del problema.

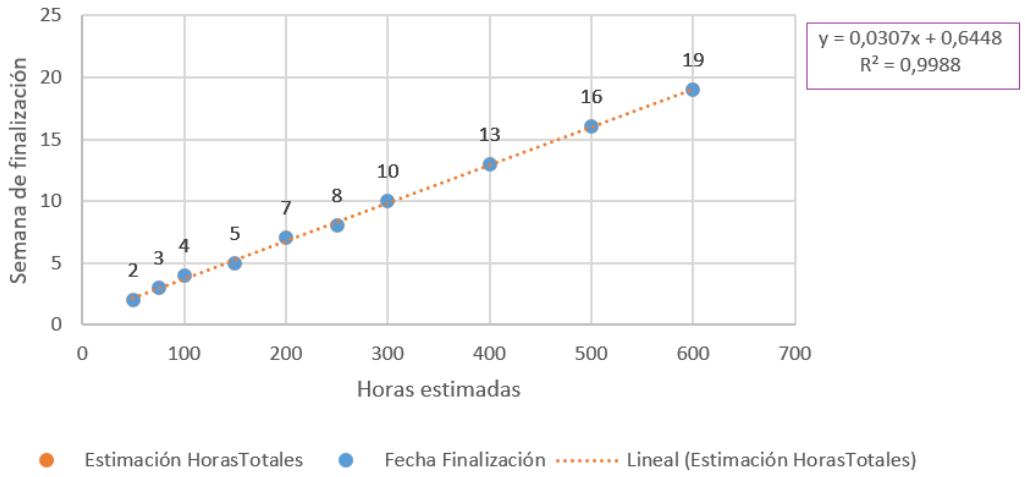


Figura 3.8: Para cada proyecto AI, la semana de finalización calculada frente al número de horas totales del proyecto.

En el apartado 3.6 se presenta una tabla de los tiempos de compilación al resolver todos los proyectos del problema uno a uno comparando el caso en el que se utilizan cotas del que no se utilizan.

3.4.2. Proyectos BI

En este caso se han generado un conjunto de 15 proyectos que abarcan un rango horario de 50 a 1250 horas. El rango horario escogido abarca a todos los proyectos proporcionados por el departamento de este tipo y su estructura es similar a la de los proyectos descritos en el apartado anterior. En la Figura 3.9 se describe con detalle las características de los proyectos generados.

En estos proyectos, la primera tarea también se realiza por un jefe de proyecto con un factor de rendimiento igual a la unidad. Por ello se obtiene de nuevo la ecuación 3.1 para su fecha de finalización.

De los 12 empleados del departamento, 7 poseen el rol necesario para realizar la segunda tarea. Como una de las restricciones establecidas es que no pueden trabajar más de 4 empleados en esta, la mitad del grupo se quedarán sin trabajar. Al resolver un único proyecto se escoge a los cuatro consultores más rápidos y se puede calcular las horas efectivas semanales del grupo.

El jefe de proyecto escogido para realizar estos proyectos es el 2, ya que además de realizar la tarea de análisis puede participar en la la construcción del proyecto y posee el factor de rendimiento más bajo de todos los empleados que pueden asumir el rol correspondiente a dicha tarea. Los dos siguientes consultores asignados a esta tarea son el 10 y el 12, ya que sus factores respectivos son 2 y 1.5. Como el resto de consultores capacitados para participar en la ejecución de esta tarea tienen un valor de 3 en el factor de rendimiento se escoge a uno de ellos indistintamente para cerrar el grupo.

	id_proy	Fecha inicio	Horas totales
0	1	2021-07-01	50
1	2	2021-07-01	75
2	3	2021-07-01	100
3	4	2021-07-14	125
4	5	2021-07-21	150
5	6	2021-06-30	175
6	7	2021-08-01	200
7	8	2021-07-15	225
8	9	2021-08-31	250
9	10	2021-09-01	300
10	11	2021-09-01	500
11	12	2021-09-01	600
12	13	2021-09-15	750
13	14	2021-08-15	1000
14	15	2021-03-15	1250

Horas primera tarea	Horas Construcción	Horas produccion
10	30	10
15	45	15
20	60	20
25	75	25
30	90	30
35	105	35
40	120	40
45	135	45
50	150	50
60	180	60
100	300	100
120	360	120
150	450	150
200	600	200
250	750	250

Figura 3.9: Table que muestra los proyectos utilizados para establecer las predicciones en la fecha de finalización de proyectos individuales de BI y desglose en horas por tarea.

Así, las horas efectivas semanales del grupo formado por dichos consultores es:

$$HES_2 = \frac{HSL}{FL_2} + \frac{HSL}{FL_{12}} + \frac{HSL}{FL_{10}} + \frac{HSL}{FL_7} = \frac{40}{1} + \frac{40}{1.5} + \frac{40}{2} + \frac{40}{3} = 40 + 26.67 + 20 + 13.33 = 100. \quad (3.8)$$

Obteniendo la siguiente ecuación para la duración en semanas en la tarea de construcción de proyectos BI:

$$SDT_2 = \frac{HET_2}{HES_2} = \frac{HET_2}{100}. \quad (3.9)$$

Por último, para la duración de la última tarea se sigue cumpliendo la ecuación 3.8, quedando una ecuación para la fecha de finalización de este tipo de proyectos:

$$EF_{BI} = SFT_1 + SDT_2 + SDT_3 = \frac{HET_1}{40} + \frac{HET_2}{100} + \frac{HET_3}{66.6}. \quad (3.10)$$

La Figura 3.10 muestra los datos obtenidos con el programa y las predicciones realizadas:

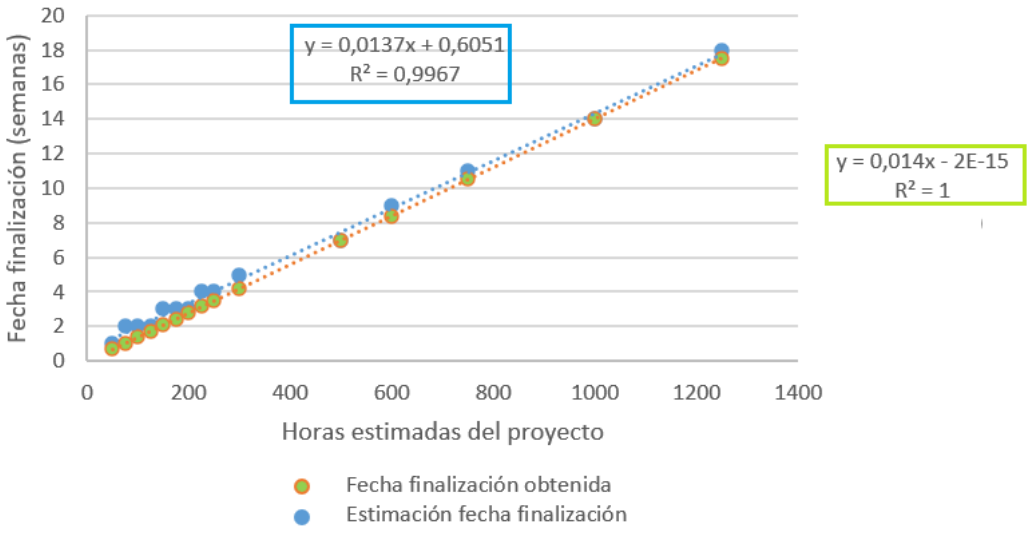


Figura 3.10: Figura en la que vemos los resultados obtenidos por el programa y las predicciones realizadas de los proyectos BI, estableciendo una línea de tendencia para realizar predicciones en los proyectos de P3D.

Tiempos de ejecución de proyectos individuales

Las Figura 3.11 presenta los tiempos de ejecución obtenidos, en segundos, para la solución del problema presentado en los dos apartados anteriores, resolviendo cada proyecto individualmente. Esta diferencia la solución del problema sin acotar del acotado. Para establecer estas cotas se ha utilizado el procedimiento mostrado en el apartado 2.5 del segundo capítulo.

TIEMPOS DE EJECUCIÓN					
Grupos proyectos individuales AI				Grupos proyectos individuales BI	
Sin cotas	Acotados			Sin cotas	Acotados
22	7			166	73

Figura 3.11: Tiempos de ejecución en segundos para el conjunto de proyectos resueltos individualmente

3.5. Cálculo de cotas temporales en grupos de proyectos

En este apartado se analiza el comportamiento en la fecha de finalización cuando varios proyectos del mismo tipo se ejecutan en paralelo, con el fin de establecer las mejores cotas temporales para el P3D con todos los proyectos. Dado que en el problema existen proyectos que se deben empezar en fechas cercanas, hay instantes temporales en los que el departamento debe tener varios proyectos en desarrollo.

En el apartado anterior se ha visto que los resultados de proyectos de distinta clase proporcionan distintos resultados, ya que el subconjunto de empleados capacitados para desempeñar las tareas de construcción difieren de uno a otro, por lo que este análisis se vuelve a dividir según la clase.

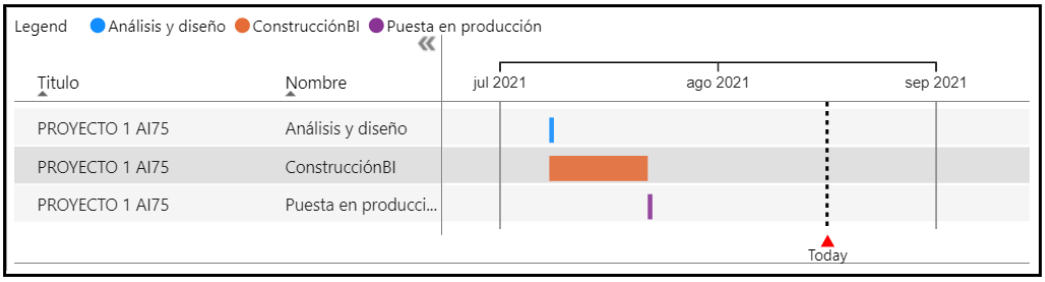
Por último, se realiza un análisis en la variación de las soluciones al introducir proyectos de distinto tipo ya que estas se pueden ver afectadas debido a que ambas clases de proyecto comparten tareas.

3.5.1. Proyectos AI en paralelo

Este apartado comienza con la resolución y análisis de un proyecto de Inteligencia Artificial. Después se sigue con la resolución de dos y tres proyectos con las mismas horas estimadas a invertir en su ejecución y de la misma clase comparando los resultados obtenidos con el resultado obtenido en el problema de un único proyecto. En el análisis de los resultados se verifica que se cumplen las restricciones establecidas y se hace un seguimiento de la variación del valor de la función objetivo dependiendo del número de proyectos.

Se resuelven proyectos con 75 horas estimadas, lo que significa que el proyecto se puede ejecutar en 3 semanas. En el Anexo A se realiza el mismo análisis con proyectos de 150 y 200 horas estimadas y con todos ellos se construyen la tabla del apartado 3.6. Esta nos proporciona el algoritmo para establecer las cotas temporales presentado en la sección 2.5 del segundo capítulo.

El resultado para el primer problema es el siguiente:



id_proy, Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
1, PROYECTO 1, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
1, PROYECTO 1, ConstrucciónBI	08/07/2021 0:00:00	22/07/2021 0:00:00
1, PROYECTO 1, Puesta en producción	22/07/2021 0:00:00	22/07/2021 0:00:00

Figura 3.12: Planificación de un proyecto de AI con 75 horas totales estimadas.

Con un desglose de horas por empleado por semana:

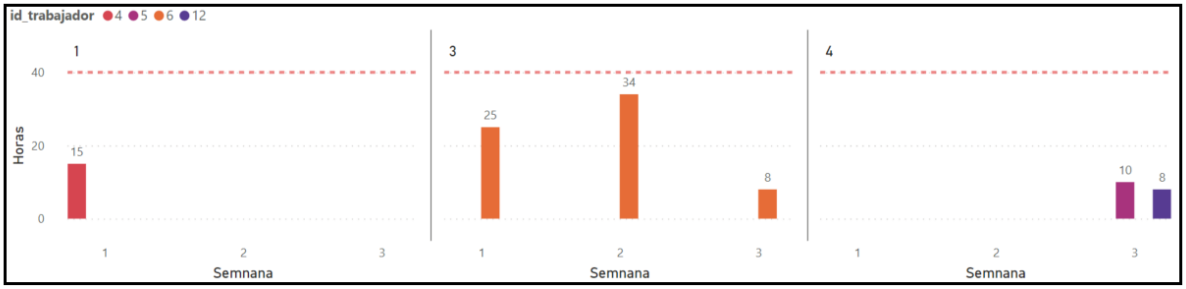


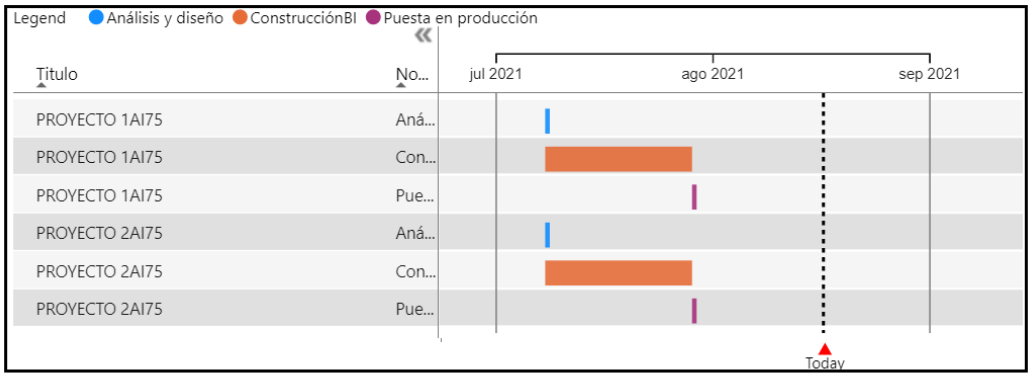
Figura 3.13: Dedicación por tareas y semanas de cada empleado.

En la primera semana, en cuanto el jefe de proyecto 4 acaba la primera tarea, el consultor 6 empieza su tarea rellenando al completo las horas de una semana laboral. La segunda semana, el empleado 6 no competa su jornada laborable y termina la tercera tarea la tercera semana, dedicándole 8 horas.

El empleado 6 no puede completar la jornada laborable de la segunda semana ya que si lo hiciera debería dedicar las 2 primeras horas de la tercera semana a terminarla, lo que no es posible ya que el mínimo de horas que un empleado debe dedicar a una tarea por semana es 8. Además, en el caso de completar su jornada laboral la segunda semana, la solución de la función objetivo seguiría siendo la misma.

Por último, los empleados 5 y 12 son los responsables de ejecutar la última tarea, terminándola la tercera semana.

Para el problema con dos proyectos la resolución del modelo proporciona una planificación como muestra la siguiente figura:



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1AI75, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 2AI75, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 1AI75, ConstrucciónBI	08/07/2021 0:00:00	29/07/2021 0:00:00
PROYECTO 2AI75, ConstrucciónBI	08/07/2021 0:00:00	29/07/2021 0:00:00
PROYECTO 1AI75, Puesta en producción	29/07/2021 0:00:00	29/07/2021 0:00:00
PROYECTO 2AI75, Puesta en producción	29/07/2021 0:00:00	29/07/2021 0:00:00

Figura 3.14: Planificación de dos proyecto de AI con 75 horas totales estimadas.

Con un desglose de horas por empleado por semana:

Figura 3.15: Dedicación por tareas y semanas de cada empleado.

El valor del tiempo de finalización de los dos proyectos en este caso es 4, superando en una unidad el valor obtenido para el problema de un proyecto.

Los cálculos para predecir la fecha de finalización mínima para el proyecto que acabe más tarde, teniendo en cuenta tres aspectos:

i) Para ejecutar la primera tarea de ambos proyectos debemos invertir 15 horas.

ii) Para ejecutar la segunda tarea de ambos proyectos debemos invertir 67 horas, lo que quiere decir que ambas tareas las terminaremos a las 134 horas.

iii) Para ejecutar la tercera tarea de ambos proyectos debemos invertir 15 horas repartidas entre los empleados 5 y 12, lo que significa que su duración mínima será de 9 horas: $\frac{9}{1} + \frac{9}{1.5} = 15$.

El total de semanas que requiere el proyecto más tardío para finalizarse es de $\frac{15+67\times2+9}{40} = 3.95$

Así podremos concluir estableciendo que el último proyecto en finalizar lo hará la cuarta semana.

La solución presentada en la Figura 3.15 cumple todas las restricciones impuestas en el modelo. El empleado 6 invierte tiempo en ambos proyectos durante las semanas $\{1,2,3,4\}$ dejando un hueco en la última semana para que los empleados 5 y 12 terminen ambos proyectos.

La interpretación del final de ambos proyectos proporciona dos opciones posibles:

i) El empleado 6 empieza la cuarta semana terminando el primer proyecto en las 20 primeras horas e invierte las próximas 10 horas semanales en terminar el segundo proyecto.

   De este modo, la última tarea del primer proyecto empieza a las 20 horas y se termina las 29 horas semanales, ya que se supone que los empleados 5 y 12 trabajan a la vez. Así, el empleado 6 empieza a trabajar en la penúltima tarea del proyecto 2 a las 20 horas y la termina en las siguientes 10 horas. Esto significa que empieza con la última tarea del segundo proyecto a las 30 horas semanales y la termina a las 39 horas.

ii) El empleado 6 empieza la cuarta semana terminando el segundo proyecto a las 10 primeras horas e invierte las próximas 20 horas semanales en terminar el primer proyecto.

De este modo, la última tarea del segundo proyecto empieza a las 10 horas y se termina a las 29 horas semanales. Así, el empleado 6 empieza a trabajar en la penúltima tarea del proyecto 1 a las 10 horas y la termina en las siguientes 20 horas Esto conlleva que empieza con la última tarea del primer proyecto a las 30 horas semanales y la termina a las 39 horas semanales.

Como se sabe que se puede terminar uno de los dos proyectos la tercera semana, se puede fijar esa fecha de finalización para uno de los proyectos y ver como modifica esta restricción impuesta a la solución general. El resultado correspondiente lo vemos en la Figura 3.16.

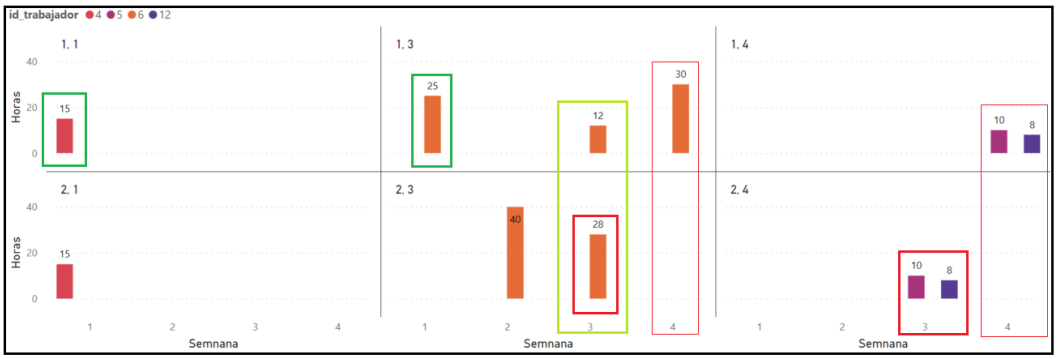


Figura 3.16: Dedicación por tareas y semanas de cada empleado asignado al los proyectos.

Con estas gráficas vemos que el resultado del tiempo de finalización de los dos proyectos sigue siendo 4 aun imponiendo que el segundo de los proyectos finalice la misma semana que el resultado obtenido para el problema de un proyecto individual.

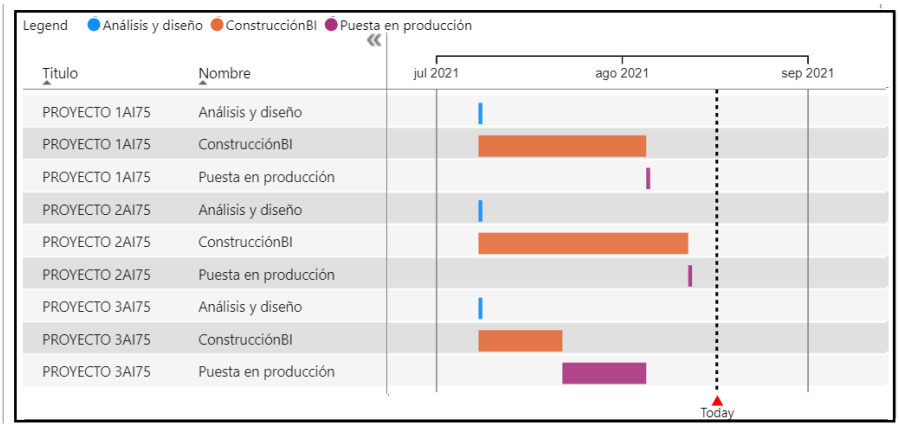
La Figura 3.16 presenta una solución que cumple todas las restricciones impuestas en el modelo.

La reducción de la ventana temporal para uno de los proyectos a una sola semana conlleva una reducción considerable del tiempo de resolución del problema, como se puede apreciar en el apartado 3.6. Además el hecho de obtener el mismo valor para la función objetivo del problema ambas soluciones serán equivalentes en este caso.

Ambas soluciones representan bien la realidad dentro de una empresa. Por un lado, la primera solución puede proporcionar más libertad y desahogo a la hora de entregar ambos proyectos. Estos deben entregarse la cuarta semana por lo que los empleados pueden repartirse más el trabajo entre ellos y, en caso de que alguna semana ocurra una incidencia en el proyecto en el que se planea trabajar, tienen la opción de trabajar en el otro.

En el segundo caso, el empleado responsable de la tarea de construcción debe estar más enfocado en el segundo proyecto durante las primeras semanas para cumplir con las fechas de entrega. El hecho de estar trabajando en menos proyectos simultáneamente le puede ayudar a ser más eficiente, pero si surge cualquier incidencia su fecha de finalización se retrasará y no cumplirá con las fechas acordadas con el cliente.

A continuación se presentan los resultados obtenidos para el problema con tres proyectos:



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1AI75, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 2AI75, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 3AI75, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 1AI75, ConstrucciónBI	08/07/2021 0:00:00	05/08/2021 0:00:00
PROYECTO 2AI75, ConstrucciónBI	08/07/2021 0:00:00	12/08/2021 0:00:00
PROYECTO 3AI75, ConstrucciónBI	08/07/2021 0:00:00	22/07/2021 0:00:00
PROYECTO 1AI75, Puesta en producción	05/08/2021 0:00:00	05/08/2021 0:00:00
PROYECTO 2AI75, Puesta en producción	12/08/2021 0:00:00	12/08/2021 0:00:00
PROYECTO 3AI75, Puesta en producción	22/07/2021 0:00:00	05/08/2021 0:00:00

Figura 3.17: Planificación de tres proyectos de AI con 75 horas totales estimadas.

Con un desglose de horas por empleado por semana:

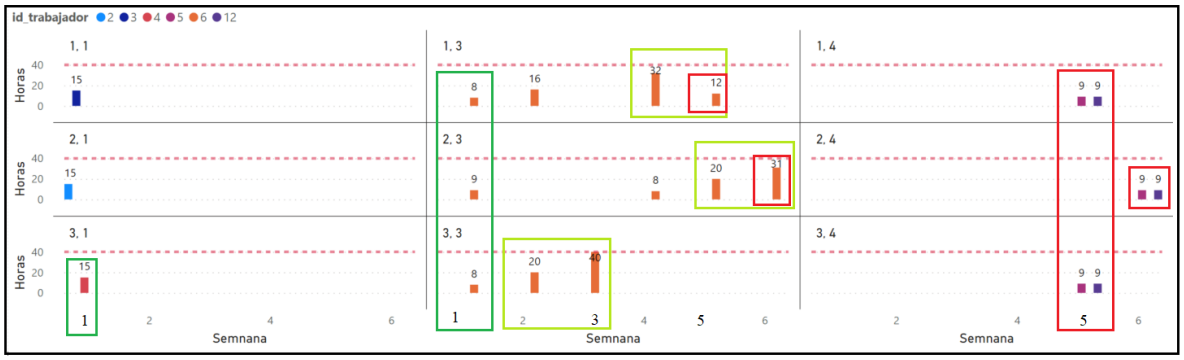


Figura 3.18: Dedicación por tareas y semanas de cada empleado asignados a los proyectos.

El valor de la función objetivo es 6, superando en tres unidades el valor obtenido para el problema de un proyecto.

Teniendo en cuenta los mismos puntos que en el apartado anterior, el total de semanas que requiere el proyecto más tardío para finalizarse en este caso es de $\frac{15+67\times3+9}{40} = 5.63$, lo que significa que el último proyecto en finalizar lo hará la sexta semana.

Como es conocido que uno de los tres proyectos puede finalizar la tercera semana y otro lo puede hacer la cuarta, se pueden fijar estas fechas de finalización para reducir el tiempo de compilación del problema de optimización. La reducción en este tiempo se puede ver en el apartado 3.6 y la Figura 3.19 proporciona su resultado:

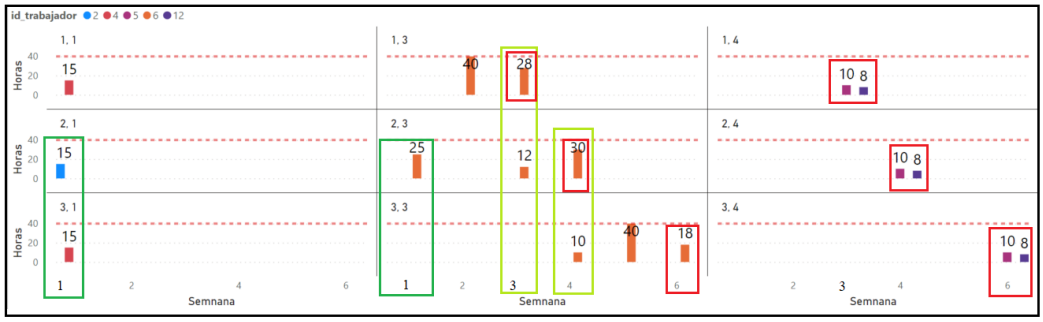


Figura 3.19: Dedicación por tareas y semanas de cada empleado.

Al acotar los dos primeros proyectos a su fecha de finalización mínima y dejando libre la fecha de finalización del tercero el valor de la función objetivo no varía, por lo que los dos resultados vistos del problema con tres proyectos son equivalentes.

A modo de resumen, la Figura 3.20 muestra las cotas temporales calculados con uno, dos y tres proyectos AI. Además, el hecho de restringir las ventanas temporales de una semana a dos de los tres proyectos implica una reducción considerable del tiempo de ejecución, pasando de 54 segundos a 1, como muestra la tabla 3.21. El hecho de ir añadiendo proyectos de la misma clase hace que aumente el valor de la función objetivo. En concreto, para proyectos de AI, este aumento es igual al número de semanas en los que la tarea de construcción de un proyecto dado coincida con la misma tarea del resto de proyectos en una planificación que suponga la fecha de finalización mínima para cada uno de ellos. Este algoritmo se ha presentado en la sección 2.5 del capítulo anterior. Los problemas de la gráfica 3.21 vienen clasificados según el número de proyectos que considera, por su clase y por el número de horas estimadas para su realización.

COTAS TEMPORALES PROYECTOS AI					
1 PROYECTO		2 PROYECTOS		3 PROYECTOS	
Cota mínima	Duración Tarea 2	Cota mínima	Cota máxima	Cota mínima	Cota máxima
3	2	3	4	3	6

Figura 3.20: Cotas para los proyectos de AI

2AI75	2AI75C	2AI75C	3AI75	3AI75C	3AI75CS
7	1	1	54	1	1

Figura 3.21: Tiempos de ejecución en segundos para los proyectos de AI

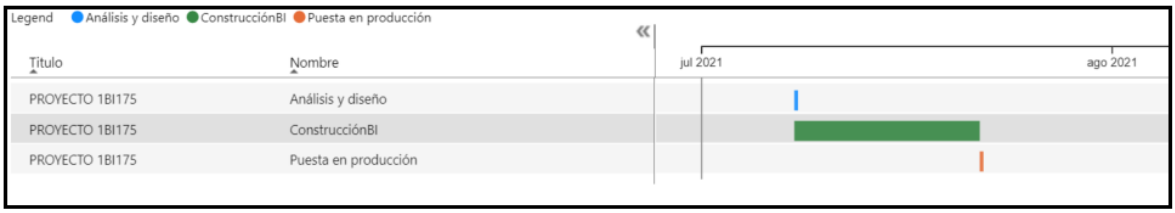
3.5.2. Proyectos BI en paralelo

En este apartado se realiza un análisis similar que en el apartado anterior utilizando proyectos de BI. Se comienza con la resolución y análisis de un proyecto con 175 horas estimadas, lo que significa que el proyecto se puede ejecutar en 3 semanas. Después se sigue con la resolución de dos y tres proyectos con las mismas horas estimadas y se comparan los resultados obtenidos con el resultado del problema de un único proyecto. En el análisis de los resultados se verifica que se cumplen las restricciones establecidas

y se hace un seguimiento de la variación del valor de la función objetivo dependiendo del número de proyectos.

En el AnexoA se realiza el mismo análisis con proyectos de 300 y 500 horas estimadas y con todos ellos se construyen la tabla del apartado 3.6. Esta nos proporciona el algoritmo para establecer las cotas temporales presentado en la sección 2.5 del segundo capítulo.

El resultado para el primer problema es el siguiente:



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1BI175, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 1BI175, ConstrucciónBI	08/07/2021 0:00:00	22/07/2021 0:00:00
PROYECTO 1BI175, Puesta en producción	22/07/2021 0:00:00	22/07/2021 0:00:00

Figura 3.22: Planificación de un proyecto de BI con 175 horas totales estimadas.

Con un desglose de horas por empleado por semana:

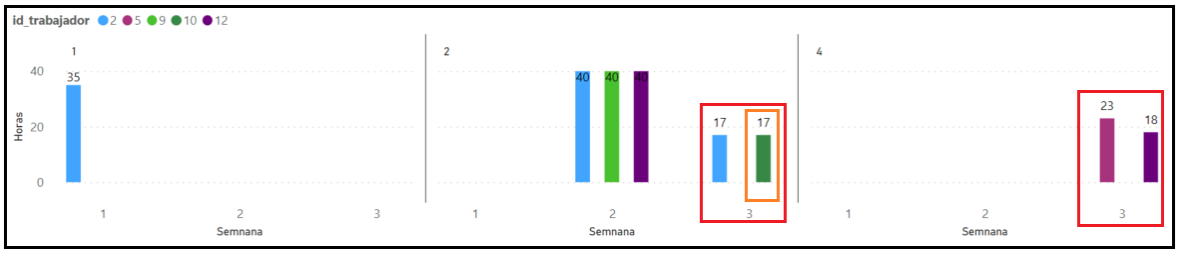


Figura 3.23: Dedicación por tareas y semanas de cada empleado.

El jefe de proyecto designado a este es el empleado 2 y además interviene en el desarrollo de la construcción junto a los empleados 9, 10 y 12. Esta segunda tarea finaliza a las 17 horas de la tercera semana, por los empleados 2 y 10, dando tiempo así a finalizar la última tarea y el proyecto en las 23 horas siguientes y por tanto, la misma semana.

Se invierten 105 horas en la segunda tarea:

$$\frac{40+17}{1} + \frac{40}{3} + \frac{40}{1.5} + \frac{17}{2} = 105$$

Para la tercera tarea, los empleados 5 y 12 se pueden repartir las 30 horas estimadas. Los factores de rendimiento para los empleados 12 y 5 son 1.5 y 1 respectivamente, por lo que para minimizar el tiempo de ejecución de la última tarea se deben cumplir las siguientes ecuaciones:

- Para terminarla cuanto antes ambos empleados deben invertir el mismo número de horas. Sea x el número de horas que debe invertir el empleado 5 e y el número de horas que debe invertir el empleado 12 se tiene que:

$$x = y$$

- La suma de las horas que inviertan ambos entre sus respectivos factores de rendimiento son igual a las horas estimadas de la tarea:

$$\frac{x}{1} + \frac{y}{1.5} = 35.$$

Por lo que se tiene finalmente que la tercera tarea se puede terminar en:

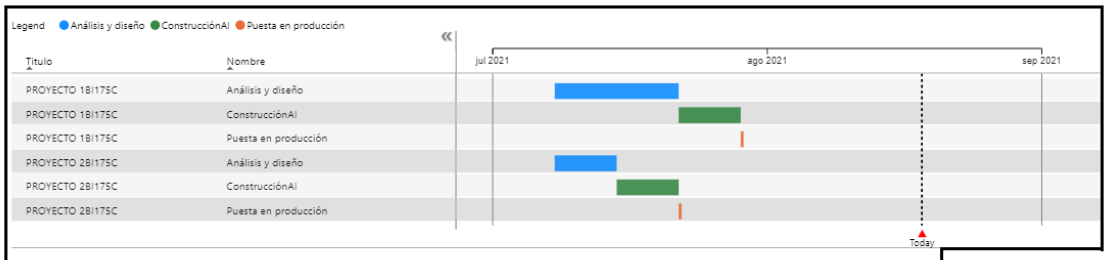
$$\frac{x}{1} + \frac{y}{1.5} = 35 \Rightarrow 2.5y = 52,5 \Rightarrow y = 21 \text{ horas.}$$

En la tarea de construcción, aunque en total intervienen 4 empleados, la segunda semana solo invierten horas tres de ellos, por lo que se puede pensar que la solución proporcionada se puede mejorar. Para verificar que esta es la solución óptima se puede imponer que intervengan todos los empleados desde la segunda semana. Teniendo en cuenta que la segunda tarea se debe empezar la segunda semana y que la duración mínima de la tercera tarea son 21 horas, para mejorar el valor de la función objetivo la segunda tarea se debe finalizar antes de las $40 - 21 = 19$ horas de la segunda semana. Así, la solución puede mejorar si se cumple que al invertir 19 horas todos los empleados capacitados en la segunda tarea, el resultado de la suma de las horas efectivas es mayor que 105:

$$\frac{19}{1} + \frac{19}{3} + \frac{19}{1.5} + \frac{19}{2} = 47.5 < 105$$

Como no se cumple, el valor óptimo de la función objetivo es 3.

El resultado obtenido para el problema con dos proyectos es el siguiente:



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1BI175C, Análisis y diseño	08/07/2021 0:00:00	22/07/2021 0:00:00
PROYECTO 2BI175C, Análisis y diseño	08/07/2021 0:00:00	15/07/2021 0:00:00
PROYECTO 1BI175C, ConstrucciónAI	22/07/2021 0:00:00	29/07/2021 0:00:00
PROYECTO 2BI175C, ConstrucciónAI	15/07/2021 0:00:00	22/07/2021 0:00:00
PROYECTO 1BI175C, Puesta en producción	29/07/2021 0:00:00	29/07/2021 0:00:00
PROYECTO 2BI175C, Puesta en producción	22/07/2021 0:00:00	22/07/2021 0:00:00

Figura 3.24: Planificación de dos proyectos de BI con 175 horas totales estimadas.

Con un desglose de horas por empleado por semana:
El valor de la función objetivo es 4.

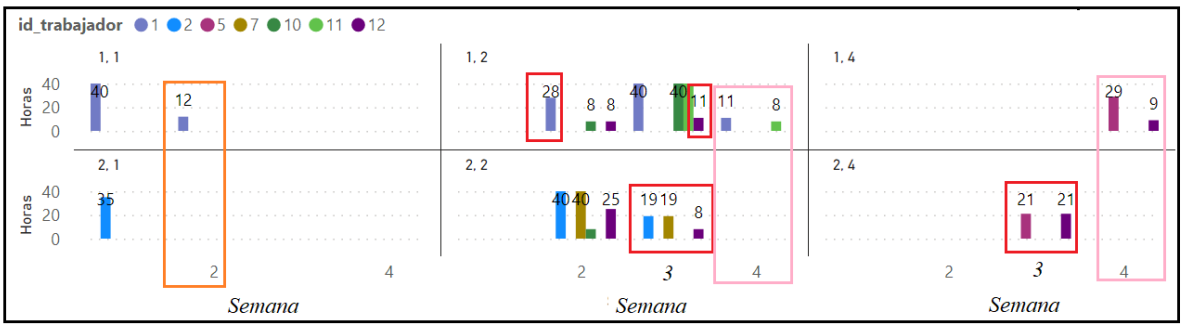


Figura 3.25: Dedicación por tareas y semanas de cada empleado.

Los resultados de la Figura 3.25 cumplen las restricciones establecidas. A continuación se presenta una interpretación de la solución:

- El jefe de proyecto del primer proyecto, la segunda semana termina el diseño de este, a las 12 horas y seguidamente se dedica a la tarea de construcción completando su jornada laboral.

- La tercera semana, los empleados del segundo proyecto invierten las horas necesarias dejando horas suficientes a los empleados encargados de la puesta en producción para finalizar el proyecto la misma semana. La tarea de consturucción de este proyecto se termina a las 19 horas y seguidamente se empieza con la última tarea, después de 21 horas y finalizándola en la tercera semana.

  Un caso particular interesante de esta semana es el desglose en horas del empleado 12:

  Este empiezaá trabajando 8 horas junto a los empleados 2 y 7 en la segunda tarea del segundo proyecto. Seguidamente, se incorporará a los empleados 10 y 11 para ayudarles en la ejecución de la tarea de construcción del primer proyecto durante 11 horas, hasta que a las 19 horas semanales, junto al consultor 5 ejecuta la puesta en producción del segundo proyecto hasta el final de la semana.

- La cuarta semana el empleado 1 finaliza la tarea de construcción a las 11 horas y la puesta en producción 29 horas después.

- Aunque ambos proyectos fueran capaces de realizar su tarea de construcción en semana y media, uno de ellos deberá finalizar una semana más tarde que el otro debido a la tarea de puesta en producción.

Las restricciones horarias y de secuencialidad impiden que el primer proyecto se finalice la tercera semana, aun siendo posible realizar su tarea de construcción en semana y media en paralelo con el segundo proyecto.

Para finalizar el primer proyecto la tercera semana se debe cumplir que el sumatorio de horas efectivas en ese periodo de tiempo sea mayor o igual que 105. Optimizando el resultado al máximo, para la segunda tarea del primer proyecto, el empleado 10, en vez de 8 horas, podrá trabajar 28 horas la segunda semana, completando así su jornada laboral. El 11 podrá trabajar 28 horas, desde que se finaliza la tarea de análisis hasta el final de la semana, el 12 podrá trabajar 15 horas, desde que finaliza con el segundo

proyecto. La tercera semana deberán invertir todos 19 horas como máximo, para que les diera tiempo a finalizarla la tercera semana. Así obtenemos el siguiente resultado:

$$\frac{28+19}{1.5} + \frac{28+19}{2} + \frac{28+19}{3} + \frac{15+11}{1.5} = 87.83 < 105$$

Lo que indica que con ese grupo y teniendo en cuenta las horas que invierten en el segundo proyecto es imposible que el primero se termine la tercera semana.

Como se sabe que se puede terminar uno de los dos proyectos la tercera semana, se puede fijar esa fecha de finalización para uno de los proyectos y ver como modifica esta restricción impuesta a la solución general. El resultado correspondiente lo vemos en la Figura 3.26. Este es muy similar al obtenido anteriormente, ya que las fechas de finalización son las mismas, pero ha sido obtenido en un tiempo de ejecución mucho menor, pasando de 211 a 66 segundos, como se puede ver en el apartado 3.6. El resultado del reparto de horas para el problema con ventanas temporales es:

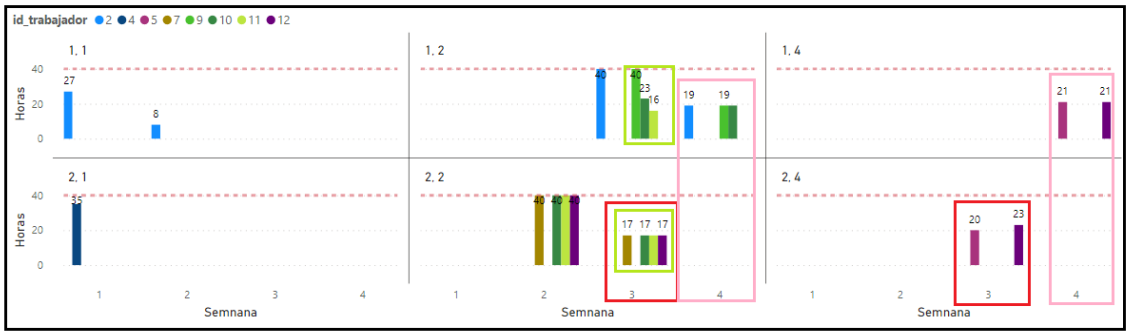


Figura 3.26: Dedicación por tareas y semanas de cada empleado.

Para el problema con tres proyectos tenemos una planificación como muestra la siguiente figura:
Con un desglose de horas por empleado por semana:
El valor de la función objetivo es 5.
El primer proyecto es asignado al jefe de proyecto 2, el cual realiza también casi toda la tarea de construcción y es ayudado por los consultores 7, 9 y 11. La puesta en producción de este proyecto la realizan entre los empleados 5 y 12.

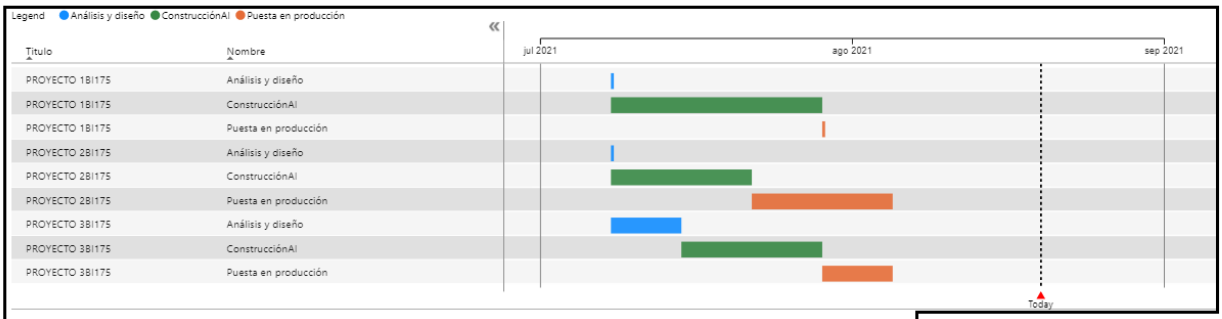
El segundo proyecto es asignado al jefe de proyecto 3, el cual solo realiza el análisis y diseño de este, ya que no está capacitado para realizar más tareas, como indica la Figura 2.4. La tarea de construcción se ejecuta por los consultores 7, 9, 10 y 11 todos con una distribución de horas similar y la puesta en producción del proyecto la lleva a cabo el empleado 5.

El tercer proyecto es asignado al jefe de proyecto 1, el cual, coopera en la tarea de construcción junto a los consultores 7 y 12.

Los resultados de la Figura 3.28 cumplen las restricciones establecidas. A continuación se presenta una interpretación de la solución:

- Los tres proyectos empiezan la tarea de construcción la segunda semana:

  En el tercer proyecto, esta tarea se empieza la segunda semana después de que el jefe de proyecto termine el diseño del proyecto, lo que sucede a las 12 horas laborables de esta semana. El consul-



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1BI175, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 2BI175, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 3BI175, Análisis y diseño	08/07/2021 0:00:00	15/07/2021 0:00:00
PROYECTO 1BI175, ConstrucciónAI	08/07/2021 0:00:00	29/07/2021 0:00:00
PROYECTO 2BI175, ConstrucciónAI	08/07/2021 0:00:00	22/07/2021 0:00:00
PROYECTO 3BI175, ConstrucciónAI	15/07/2021 0:00:00	29/07/2021 0:00:00
PROYECTO 1BI175, Puesta en producción	29/07/2021 0:00:00	29/07/2021 0:00:00
PROYECTO 2BI175, Puesta en producción	22/07/2021 0:00:00	05/08/2021 0:00:00
PROYECTO 3BI175, Puesta en producción	29/07/2021 0:00:00	05/08/2021 0:00:00

Figura 3.27: Planificación de tres proyectos de BI con 175 horas totales estimadas.

tor 12 sigue con la segunda tarea invirtiendo 28 horas, o sea, hasta el final de la semana.

- Tanto el primer como el tercer proyecto acaban la tarea de construcción la cuarta semana cuando llevan trabajando 13 horas. Seguidamente, el proyecto 1 comienza la tarea de procesamiento, ejecutada por los empleados 5 y 12. Esta finaliza la misma cuarta semana, cuando el empleado 12 ha invertido 25 horas después de las 13 que se invierten en la construcción. El empleado 5 termina a las 29 horas e invierte el resto de horas laborables que le quedan de la semana en adelantar la producción del tercer proyecto. Finalmente, este tercer proyecto acaba la quinta semana con una inversión de 40 horas laborables del empleado 12.

  La tarea de construcción del segundo proyecto está la tercera semana. Aunque este podría acabar la cuarta semana, el empleado 5 no lo finaliza hasta la semana siguiente, ya que la cuarta semana está asignado en los proyectos 1 y 3 y no dispone de horas laborables.

- Ningún empleado supera ninguna semana las 40 horas laborables.

  En el Anexo A se sigue con este análisis de proyectos BI ejecutados en paralelo con mayores dimensiones. Los resultados resumidos de la variación en las fechas de finalización se pueden ver en el apartado 3.6, mediante los que se ha construido el algoritmo de ventanas temporales mostrado en el apartado 2.5 del capítulo anterior.

  Al acotar los dos primeros proyectos a su fecha de finalización mínima y dejando libre la fecha de finalización del tercero el valor de la función objetivo no varía, por lo que los dos resultados vistos del problema con tres proyectos son equivalentes.

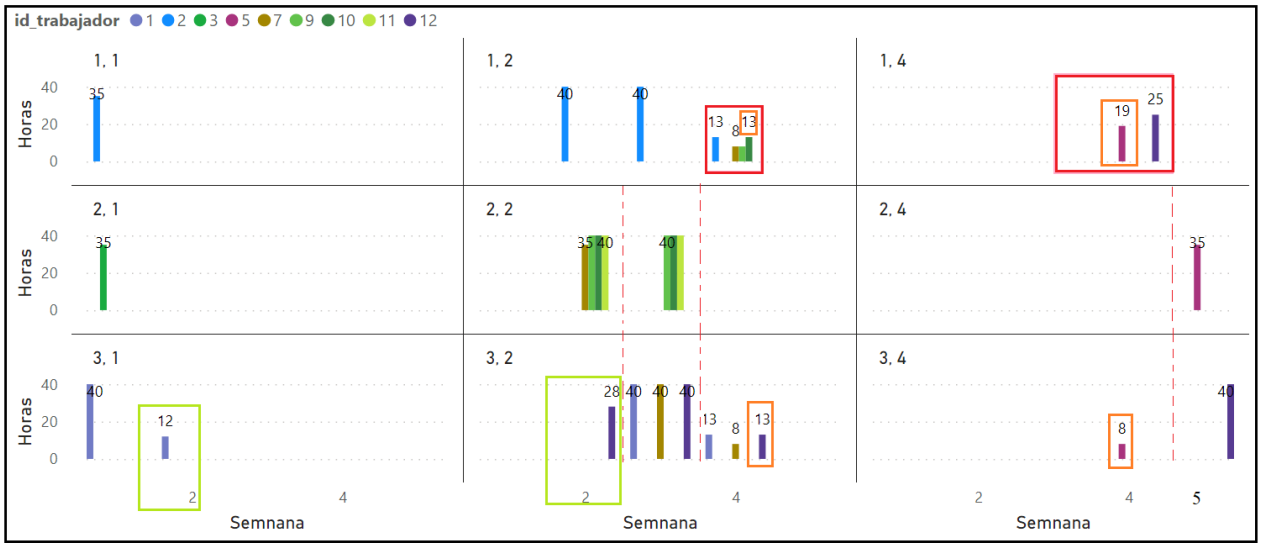


Figura 3.28: Dedicación por tareas y semanas de cada empleado.

A modo de resumen, la Figura 3.29 muestra las cotas temporales calculados con uno, dos y tres proyectos AI. Además, el hecho de restringir las ventanas temporales de una semana a dos de los tres proyectos implica una reducción considerable del tiempo de ejecución, pasando de 211 segundos a 34, como muestra la tabla 3.30.

Para proyectos de Inteligencia Empresarial, este aumento en semanas depende mucho de la duración de cada proyecto. Como para el caso de los proyectos con mas horas estimadas analizados este aumento es igual al número de semanas en los que la tarea de construcción de un proyecto dado coincide con la misma tarea del resto de proyectos cuando estos se solucionan de manera individual, el algoritmo a seguir es igual que el construido para los proyectos de Inteligencia Artificial presentado en la sección 2.5 del capítulo anterior.

Los problemas de la gráfica 3.21 vienen clasificados según el número de proyectos que considera, por su clase y por el número de horas estimadas para su realización.

COTAS TEMPORALES PROYECTOS BI					
1 PROYECTO		2 PROYECTOS		3 PROYECTOS	
Cota mínima	Duración Tarea 2	Cota mínima	Cota máxima	Cota mínima	Cota máxima
3	2	3	4	4	5

Figura 3.29: Cotas para los proyectos de BI

2BI175	2BI175C	2BI175CS	3BI175	3BI175C	3BI175CS
50	37	14	211	66	34

Figura 3.30: Tiempos de ejecución en segundos para los proyectos de BI

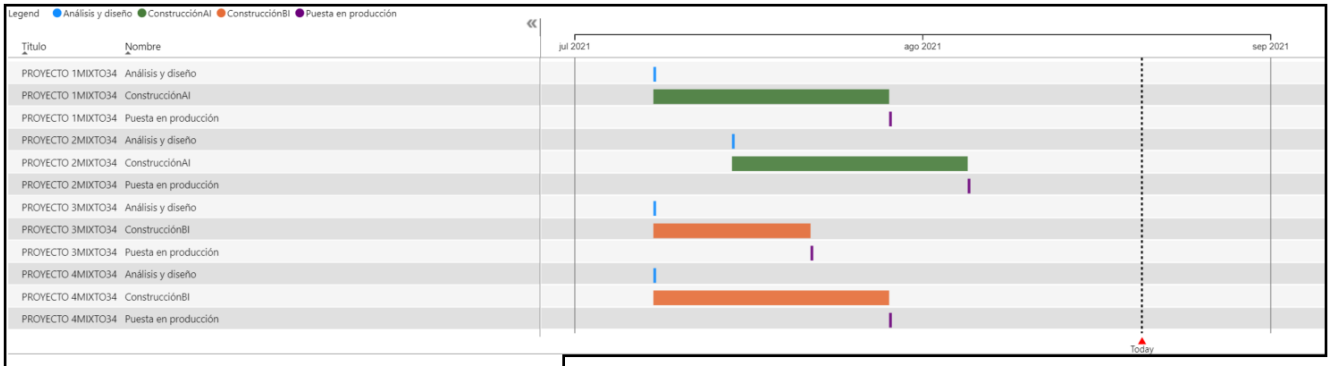
3.5.3. Proyectos AI y BI en paralelo

Como las dos clases de proyectos comparten tareas y por tanto empleados, en este apartado se realiza un análisis de un grupo de proyectos incorporando ambas clases de proyectos. Se consideran los mismos proyectos que en los apartados anteriores para ver si la fecha de finalización de cada proyecto se ve afectada. Se analiza un caso con dos proyectos de Inteligencia Artificial y dos proyectos de Inteligencia Empresarial, los cuales tienen la misma fecha de iniciación y las dimensiones que los proyectos utilizados en los apartados 3.5.1 y 3.5.2.

Aunque se trabaja con las dos clases de proyectos, cuando dos proyectos de distinta clase tengan la misma fecha de finalización estimada, la tarea que no permita terminarlos la misma semana será la puesta en producción, ya que las dos tareas de construcción no comparten empleados.

Otro posible aspecto que puede afectar a la fecha de finalización de un proyecto dado al considerar proyectos de ambos tipos puede ser empezar la misma semana con un número de proyectos mayor que el número de jefes de proyecto del departamento, en nuestro caso 4. Esto supone que alguno de ellos empiece más tarde o que uno de los jefes se divida el trabajo entre dos proyectos aumentando la duración de la tarea de su diseño.

Los resultados para la fecha de finalización de ambas parejas de proyectos utilizados es $[3,4]$ si los ejecutamos solos. Por ello se fijan estas fechas de finalización para los proyectos de Inteligencia Artificial y se establece una ventana temporal para los proyectos de Inteligencia Empresarial de $[3-4,4-5]$. Los resultados obtenidos para este problema han sido los siguientes:



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1MIXTO34, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 2MIXTO34, Análisis y diseño	15/07/2021 0:00:00	15/07/2021 0:00:00
PROYECTO 3MIXTO34, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 4MIXTO34, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 1MIXTO34, ConstrucciónAI	08/07/2021 0:00:00	29/07/2021 0:00:00
PROYECTO 2MIXTO34, ConstrucciónAI	15/07/2021 0:00:00	05/08/2021 0:00:00
PROYECTO 3MIXTO34, ConstrucciónBI	08/07/2021 0:00:00	22/07/2021 0:00:00
PROYECTO 4MIXTO34, ConstrucciónBI	08/07/2021 0:00:00	29/07/2021 0:00:00
PROYECTO 1MIXTO34, Puesta en producción	29/07/2021 0:00:00	29/07/2021 0:00:00
PROYECTO 2MIXTO34, Puesta en producción	05/08/2021 0:00:00	05/08/2021 0:00:00
PROYECTO 3MIXTO34, Puesta en producción	22/07/2021 0:00:00	22/07/2021 0:00:00
PROYECTO 4MIXTO34, Puesta en producción	29/07/2021 0:00:00	29/07/2021 0:00:00

Figura 3.31: Figura en la que vemos la planificación de dos proyectos de BI con 175 horas y otros dos proyectos de AI con 75 horas estimadas que se ejecutan en paralelo.

Con un desglose de horas por empleado por semana como muestra la figura 3.32.

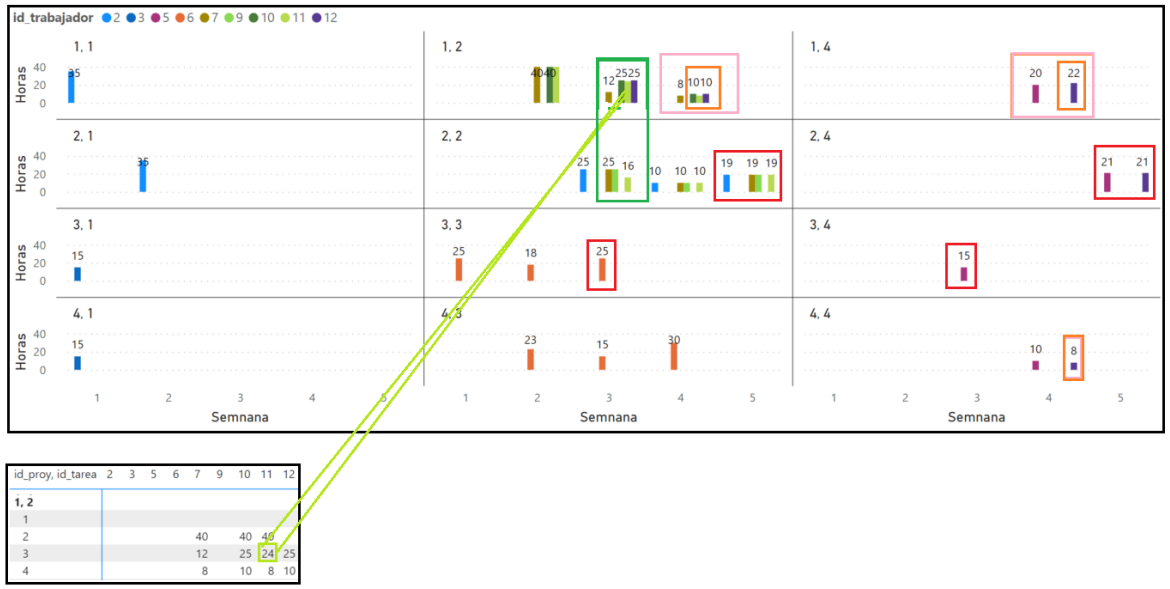


Figura 3.32: Figura en la que vemos la dedicación en horas por tareas y semanas de cada empleado asignado a cada proyecto.

Aunque la fecha de comienzo de todos los proyectos sugerida en un principio es la misma, el resultado indica que el análisis y diseño del segundo proyecto se realiza la segunda semana.Las fechas de finalización para los proyectos AI pueden seguir siendo las mismas, pero el P3D requiere de la adición de una quinta semana para poder terminar los proyectos de BI.

Cada empleado cumple las restricciones de su horario laboral y las tareas siguen la restricción de secuencialidad.

- La tercera semana se termina el primer proyecto de AI finalizando la tarea de construcción a las 25 horas semanales y la puesta en producción, por el empleado 5, en las 15 horas restantes.

- La cuarta semana se terminan el primer proyecto de BI y el segundo de AI.

  Primero se termina la tarea de construcción del proyecto BI, en la que participa el empleado 12, a las 10 horas semanales. Seguidamente se empieza con su puesta en producción, en la que participan el empleado 5 con 20 horas y el empleado 12 con 22 horas.En cuanto acaban con esta tarea se pasan a la puesta en producción del proyecto de AI, la cual se empieza a las 30 horas semanales, completando sus jornada laboral.

- Por último, la quinta semana se termina el segundo proyecto de BI.

3.6. Resumen de las cotas temporales y los tiempos de ejecución del modelo matemático

En este apartado se presentan los datos obtenidos para las cotas temporales y los tiempos de ejecución de los problemas anslizados a lo largo del capítulo y los presentados en el Anexo A.

3.6.1. Cotas temporales

En este apartado se presentan los datos obtenidos para las cotas temporales, las cuales ayudan a reducir el tiempo de ejecución del problema y sirven de guía para construir el algoritmo de construcción de cotas del problema ganaral visto en el apartado 2.5 del segundo capítulo.

COTAS TEMPORALES PROYECTOS AI					
1 PROYECTO		2 PROYECTOS		3 PROYECTOS	
Cota mínima	Duración Tarea 2	Cota mínima	Cota máxima	Cota mínima	Cota máxima
3	2	3	4	3	6
5	3	5	8	5	12
7	4	7	11	7	15

Figura 3.33: Cotas para los proyectos de AI

COTAS TEMPORALES PROYECTOS BI					
1 PROYECTO		2 PROYECTOS		3 PROYECTOS	
Cota mínima	Duración Tarea 2	Cota mínima	Cota máxima	Cota mínima	Cota máxima
3	2	3	4	4	5
5	3	5	6	7	8
7	4	8	11	9	15

Figura 3.34: Cotas para los proyectos de BI

2AI75	2AI75C	2AI75C	3AI75	3AI75C	3AI75CS
7	1	1	54	1	1
2AI150	2AI150C	2AI150CS	3AI150	3AI150C	3AI150CS
11	7	1	600	1673	1
2AI200	2AI200C	2AI200CS	3AI200	3AI200C	3AI200CS
34	18	10	6000	2535	34

Figura 3.35: Tiempos de ejecución en segundos para los proyectos de AI

3.6.2. Tiempos de ejecución de grupos de proyectos

En esta sección se presentan los tiempos de compilación para los problemas presentados en el apartado 3.5. Los nombres de los problemas están etiquetados según el número de proyectos del problema, la clase de proyecto, el número de horas y la cota establecida, de manera que si un problema incluye dos proyectos de Inteligencia Artificial con 150 horas estimadas y acotado se denomina como *2AI150C*.

Los problemas se han resuelto dejando un límite en el tiempo de compilación de 6000 segundos. Además para los problemas sin acotar se ha fijado un horizonte temporal de 20 semanas.

La Figura 3.36 presenta los tiempos de ejecución obtenidos, en segundos, para la solución de los problemas presentados en el apartado 3.4.1 del tercer capítulo y del apartado A.1 del Anexo A para problemas con 1, 2 y 3 proyectos y con distintas horas de ejecución estimadas.

Estos se recsuelven sin acotar, esto es, dejando un horizonte temporal de 20 semanas y acotados, utilizando las cotas mínima y máxima obtenidas de la tabla 3.6.1.

Para el caso de los problemas llamados CS se obliga a que la fecha de finalización del primer proyecto sea igual a la cotamínima obtenida, dejando únicamente una ventana temporal para el último proyecto del grupo.

2AI75	2AI75C	2AI75C	3AI75	3AI75C	3AI75CS
7	1	1	54	1	1
2AI150	2AI150C	2AI150CS	3AI150	3AI150C	3AI150CS
11	7	1	600	1673	1
2AI200	2AI200C	2AI200CS	3AI200	3AI200C	3AI200CS
34	18	10	6000	2535	34

Figura 3.36: Tiempos de ejecución en segundos para los proyectos de AI

Para el problema 3AI200, esto es, el problema con 3 proyectos de clase AI con 200 horas estimadas, no se ha podido encontrar el óptimo de la función obtenida en el límite temporal establecido. El programa ha sido capaz de devolver la solución óptima del problema pero no le ha dado tiempo de comprobarla.

La Figura 3.37 presenta los tiempos de ejecución obtenidos, en segundos, para la solución de los problemas presentados en el apartado 3.4.2 del tercer capítulo y del apartado A.2 del Anexo A para

problemas con 1, 2 y 3 proyectos y con distintas horas de ejecución estimadas.

El procedimiento para el establecimiento de cotas es el mismo pero escogiendo los datos proporcionados de la Figura 3.34

2BI175	2BI175C	2BI175CS	3BI175	3BI175C	3BI175CS
50	37	14	211	66	34
2BI300	2BI300C	2BI300CS	3BI300	3BI300C	3BI300CS
412	170	150	6000	3763	618
2BI500	2BI500C	2BI500CS	3BI500	3BI500C	3BI500CS
87	311	87	6000	5374	963

Figura 3.37: Tiempos de ejecución en segundos para los proyectos de BI

Para los problemas 3BI300 Y 3BI500 no se ha podido encontrar el óptimo de la función obtenida en el límite temporal establecido. En este tiempo, el programa ha proporcionado soluciones factibles pero no óptimas.

3.7. Líneas futuras de investigación

Después de obtener las soluciones y ver que los objetivos del TFM se han cumplido satisfactoriamente, se pantean líneas futuras por las que se podría seguir ampliando el modelo propuesto, que generen planificaciones más detalladas o extensiones.

- Ampliar el alcance a otras áreas de la empresa:

  Además del departamento de Data & AI, EFOR está constituida por otra serie de departamentos como Marketing y estrategia digital, Soluciones de Software y Gestión del Talento entre otros, los cuales están formados por empleados y desarrollan proyectos los cuales habrá que analizar para proponer un medelo equivalente que devuelva una planificación óptima del departamento.

- Cambiar algunos parámetros por variables de decisión:

  Otra extensión que se puede realizar al problema es introducir una variable de decisión que indique cuáles de los proyectos pendientes de realizar se van a realizar, intentando maximizar el beneficio que obtenga la empresa de ellos.

  Además, suponiendo que todavía no esté fijado un acuerdo con el cliente, se puede modificar la fecha de iniciación de cada proyecto, de manera que sirva para que los empleados estén implicados en un número menor de proyectos por instante temporal o que se mejore la solución al problema principal, pudiendo adelantar la fecha de iniciación de algunos de ellos si se dispone de los empleados suficientes.

- Modificar secuenciación entre tareas:

  Aunque en el presente trabajo se ha supuesto que una tarea dada no podrá empezarse hasta la finalización de su tarea predecesora este aspecto no se da en la totalidad de los proyectos, pudiendo

haber semanas en las que se trabaje en dos tareas sucesivas al mismo tiempo.

Este aspecto se puede modelizar con la definición de un parámetro que indique el número de semanas que pueden ejecutarse en paralelo dos tareas consecutivas o que especifique a partir de que hora invertida en una tarea dada podremos empezar con su tarea sucesiva.

Anexos

Anexo A

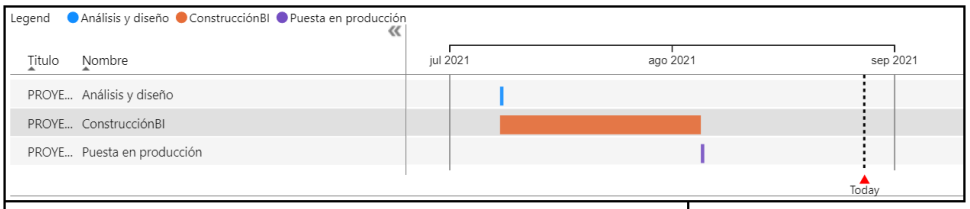
Resultados de la planificación de proyectos para 5 y 7 semanas

En este anexo se sigue con el análisis presentado en el apartado 3.5 del tercer capítulo con proyectos con mayor número de horas estimadas. Estos se han ejecutado tanto estableciendo cotas temporales como sin ellas, presentando los tiempos de compilación de cada uno de ellos en las figuras del apartado 3.6 del capítulo 3.

A.1. Análisis Proyectos AI

A.1.1. Proyectos AI con 150 horas estimadas

El resultado para el problema en el que se resuelve la planificación de un proyecto individual de Inteligencia Artificial con 150 horas estimadas es:



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1AI150, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 1AI150, ConstrucciónBI	08/07/2021 0:00:00	05/08/2021 0:00:00
PROYECTO 1AI150, Puesta en producción	05/08/2021 0:00:00	05/08/2021 0:00:00

Figura A.1: Planificación de un proyecto de AI con 150 horas totales estimadas.

Con un desglose de horas por trabajador por semana:

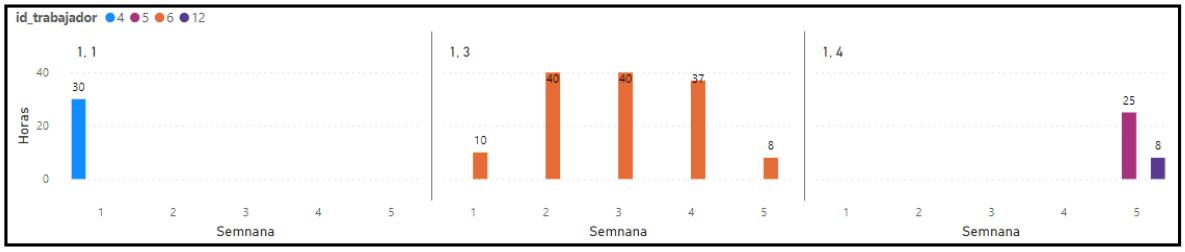


Figura A.2: Dedicación por tareas y semanas de cada empleado.

El valor de la función objetivo es 5. En cuanto se acaba la primera tarea la primera semana ejecutada por el jefe de proyecto 4, el empleado 6 empieza su tarea rellenando al completo las horas de una semana laboral. Este continua con ella tres semanas más finalizándola en las ocho primeras horas de la quinta semana.

Se puede comprobar que la solución es correcta, ya que se puede calcular el mínimo de la función objetivo:
Para terminar la primera tarea el jefe de proyecto debe invertir 30 horas.

Para terminar la segunda, como el empleado 6 tiene un factor de rendimiento de 1.5 debe invertir un total de $90 \times 1.5 = 135$ horas.

Para la tercera tarea, los trabajadores 5 y 12 se pueden repartir las 30 horas estimadas. Como los factores de rendimiento de los empleados 12 y 5 son 1.5 y 1 respectivamente, se puede minimizar la duración de esta tarea si ambos dedican 18 horas en ella:

- Para terminarla cuanto antes, ambos empleados deberán invertir el mismo número de horas. Sea x el número de horas que debe invertir el empleado 5 e y el número de horas que debe invertir el empleado 12. Por tanto tendremos que:

  $$x = y$$

- La suma de las horas que inviertan ambos entre sus respectivos factores de lentitud serán igual a las horas estimadas de la tarea:

  $$\frac{x}{1} + \frac{y}{1.5} = 30.$$

Por lo que tendremos finalmente que la tercera tarea se podrá terminar en:

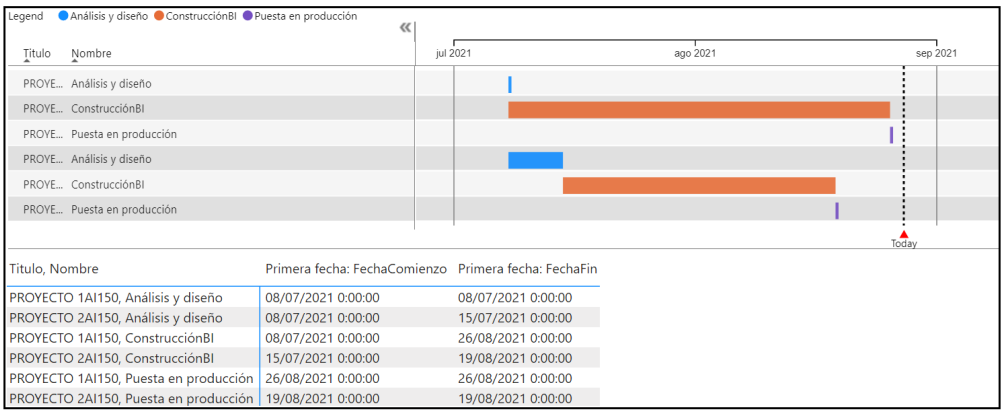
$$\frac{x}{1} + \frac{y}{1.5} = 30 \Rightarrow 2.5y = 45 \Rightarrow y = 18 \text{ horas.}$$

Obteniendo un resultado para el Makespan de:

$$Makespan = \frac{30+135+18}{40} = 4.58.$$

Lo que indica que es imposible terminar el proyecto antes de la quinta semana, por lo que el valor mínimo de la función objetivo es 5.

La Figura A.3 muestra la planificación para el problema con dos proyectos:



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1AI150, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 2AI150, Análisis y diseño	08/07/2021 0:00:00	15/07/2021 0:00:00
PROYECTO 1AI150, ConstrucciónBI	08/07/2021 0:00:00	26/08/2021 0:00:00
PROYECTO 2AI150, ConstrucciónBI	15/07/2021 0:00:00	19/08/2021 0:00:00
PROYECTO 1AI150, Puesta en producción	26/08/2021 0:00:00	26/08/2021 0:00:00
PROYECTO 2AI150, Puesta en producción	19/08/2021 0:00:00	19/08/2021 0:00:00

Figura A.3: Planificación de dos proyecto de AI con 150 horas totales estimadas.

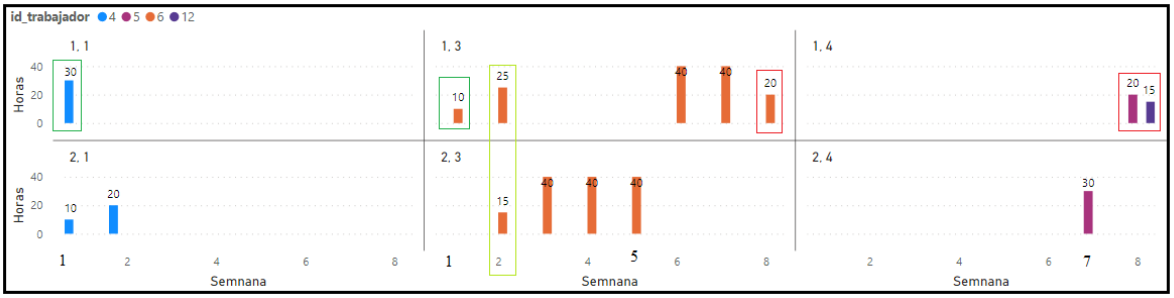


Figura A.4: Dedicación por tareas y semanas de cada empleado.

Con un desglose de horas por trabajador por semana:

El valor de la función objetivo en el mínimo en este caso es 8, superando en tres unidades el valor obtenido para el problema de un proyecto.

Los cálculos para obtener el valor del mínimo son:

$$Makespan = \frac{30+2\times135+18}{40} = 7.95.$$

Lo que indica que es imposible terminar el proyecto antes de la quinta semana, por lo que el valor mínimo de la función objetivo será 8.

Como se ha hecho en el apartado 3.5.1 se puede restringir la fecha de finalización para el primer proyecto a la obtenida en el problema con un solo proyecto, quinta semana, y ver si cambia la solución de nuestro problema. La Figura A.5 muestra el reparto de horas.

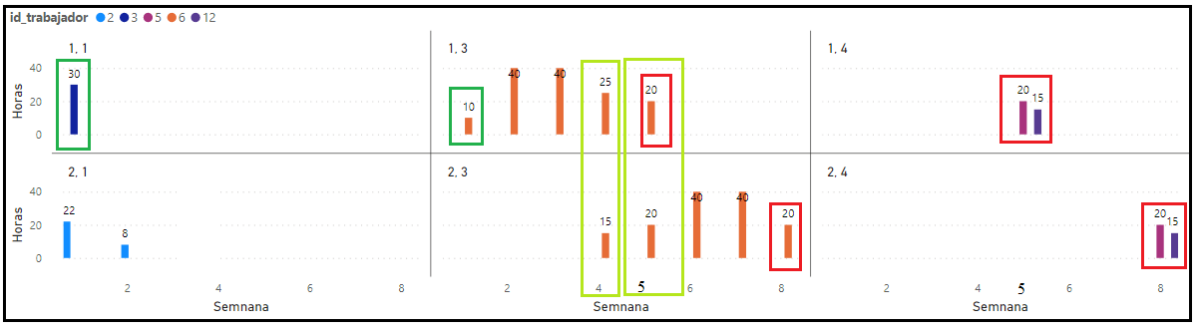
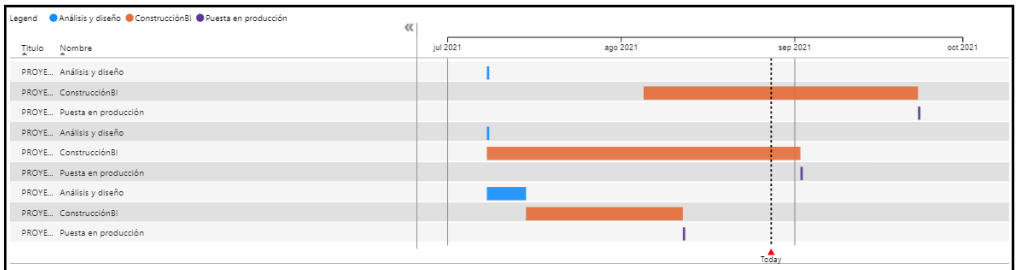


Figura A.5: Dedicación por tareas y semanas de cada empleado.

El valor de la función objetivo es el mismo que el obtenido en el caso anterior, lo que significa que las soluciones son equivalentes. Sin embargo, al restringir al mínimo la fecha de finalización de uno de los proyectos, disminuye el tiempo de ejecución del programa, como podemos ver en la Figura 3.36 del apartado 3.6 del Capítulo 3.

Un aspecto interesante de la Figura A.5 es que, para que su solución sea real, en la quinta semana el empleado 6 debe empezar con la construcción del primer proyecto, para que este pueda finalizar dicha semana. Al finalizar con esa tarea comenzará con la construcción del siguiente proyecto.

La FiguraA.6 muestra la planificación para el problema con tres proyectos:



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1AI150, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 2AI150, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 3AI150, Análisis y diseño	08/07/2021 0:00:00	15/07/2021 0:00:00
PROYECTO 1AI150, ConstrucciónBI	05/08/2021 0:00:00	23/09/2021 0:00:00
PROYECTO 2AI150, ConstrucciónBI	08/07/2021 0:00:00	02/09/2021 0:00:00
PROYECTO 3AI150, ConstrucciónBI	15/07/2021 0:00:00	12/08/2021 0:00:00
PROYECTO 1AI150, Puesta en producción	23/09/2021 0:00:00	23/09/2021 0:00:00
PROYECTO 2AI150, Puesta en producción	02/09/2021 0:00:00	02/09/2021 0:00:00
PROYECTO 3AI150, Puesta en producción	12/08/2021 0:00:00	12/08/2021 0:00:00

Con un desglose de horas por trabajador por semana:

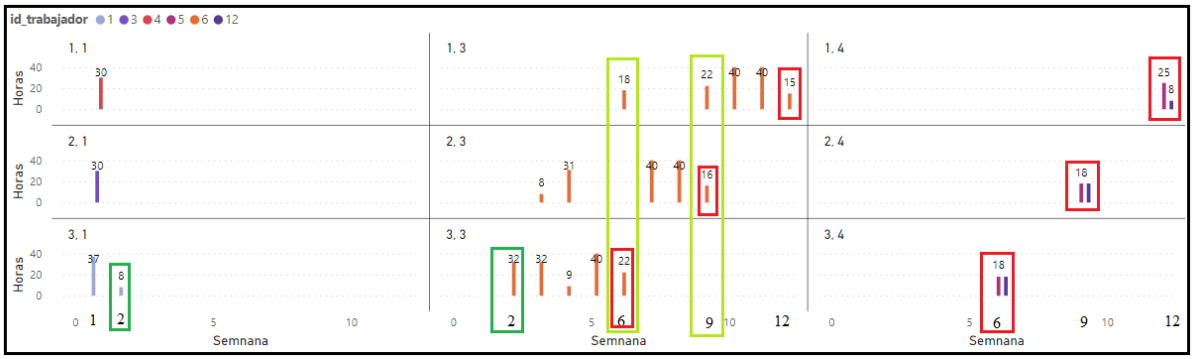


Figura A.7: Dedicación por tareas y semanas de cada empleado.

El valor de la función objetivo en el mínimo es 12, superando en siete unidades el valor obtenido para el problema de un proyecto y en tres al valor obtenido con dos proyectos.

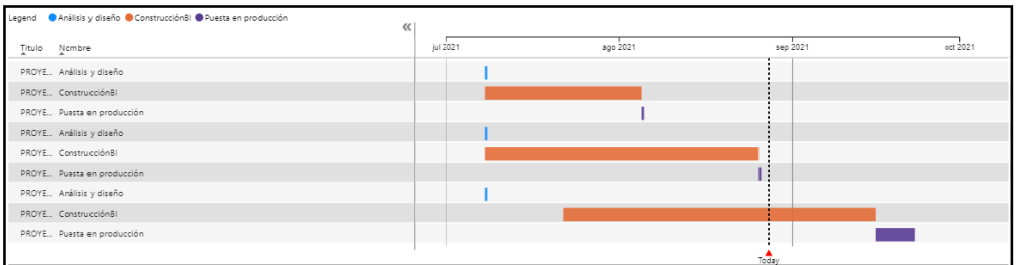
Los cálculos para obtener el valor del mínimo son:

$$Makespan = \frac{30+3\times135+18}{40} = 11.32.$$

Lo que nos indica que es imposible terminar el proyecto antes de la quinta semana y que por tanto el valor mínimo de la función objetivo será 12.

Como se ha hecho en el apartado 3.5.1 se puede restringir la fecha de finalización para los dos primeros proyectos a la obtenida en el problema con dos proyectos, lo que quiere decir fijar la fecha de finalización del primer proyecto a la quinta y la del segundo a la octaba semana, y ver si cambia la solución de nuestro problema.

Los resultados obtenidos han sido los siguientes:



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1AI150C, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 2AI150C, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 3AI150C, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 1AI150C, ConstrucciónBI	08/07/2021 0:00:00	05/08/2021 0:00:00
PROYECTO 2AI150C, ConstrucciónBI	08/07/2021 0:00:00	26/08/2021 0:00:00
PROYECTO 3AI150C, ConstrucciónBI	22/07/2021 0:00:00	16/09/2021 0:00:00
PROYECTO 1AI150C, Puesta en producción	05/08/2021 0:00:00	05/08/2021 0:00:00
PROYECTO 2AI150C, Puesta en producción	26/08/2021 0:00:00	26/08/2021 0:00:00
PROYECTO 3AI150C, Puesta en producción	16/09/2021 0:00:00	23/09/2021 0:00:00

Figura A.8: Figura en la que vemos la planificación de dos proyectos de AI con 150 horas totales estimadas, donde fijamos la fecha de finalización de los dos primeros proyectos.

Con un desglose de horas por trabajador por semana:

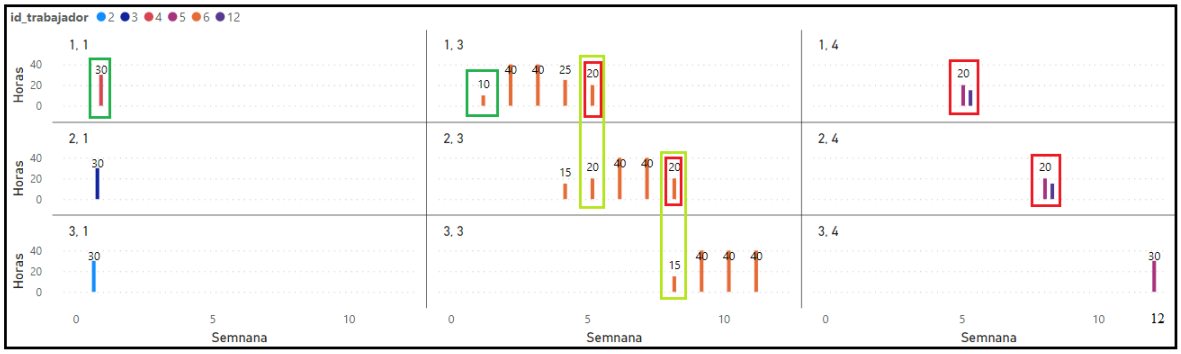


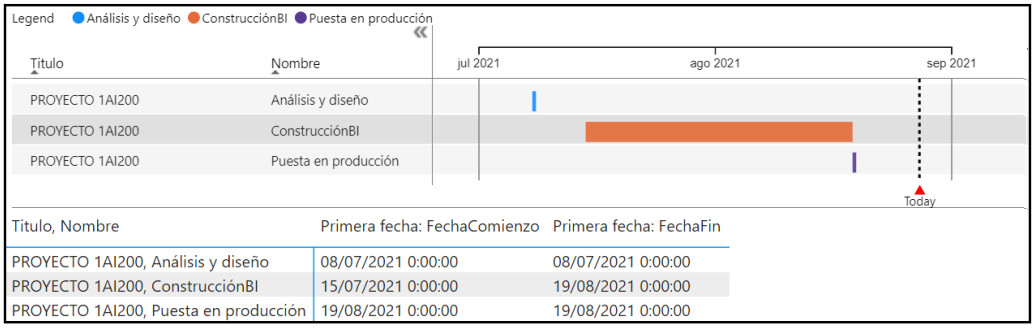
Figura A.9: Figura en la que vemos la dedicación por tareas y semanas de cada tranajador asignado al proyecto.

El valor de la función objetivo es el mismo que el obtenido en el caso anterior, lo que significa que las soluciones son equivalentes. Sin embargo, al restringir al mínimo la fecha de finalización de dos de los tres proyectos, disminuye el tiempo de ejecución del programa, como podemos ver en la Figura 3.36 del apartado 3.6 del Capítulo 3.

Un aspecto interesante de la figura A.5 es que, para que su solución sea real, en la quinta y octaba semana, el empleado 6 debe empezar con los proyectos 1 y 2 respectivamente, ya que de otro modo no dará tiempo a finalizar los proyectos dichas semanas.

A.1.2. Proyectos AI con 200 horas estimadas

El resultado para el problema en el que se resuelve la planificación de un proyecto individual de Inteligencia Artificial con 200 horas estimadas es:



Titulo	Nombre
PROYECTO 1AI200	Análisis y diseño
PROYECTO 1AI200	ConstrucciónBI
PROYECTO 1AI200	Puesta en producción

Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1AI200, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 1AI200, ConstrucciónBI	15/07/2021 0:00:00	19/08/2021 0:00:00
PROYECTO 1AI200, Puesta en producción	19/08/2021 0:00:00	19/08/2021 0:00:00

Figura A.10: Planificación de un proyecto de AI con 200 horas totales estimadas.

Con un desglose de horas por trabajador por semana:

El valor de la función objetivo en el mínimo es 7. Se puede comprobar que la solución es correcta, ya que se puede calcular el mínimo de la función objetivo:

Para terminar la primera tarea el jefe de proyecto debe invertir 40 horas.Para terminar la segunda, como el empleado 6 tiene un factor de rendimiento de 1.5 debe invertir un total de $120 \times 1.5 = 180$ horas.Para la tercera tarea, los trabajadores 5 y 12 se pueden repartir las 40 horas estimadas, pudiendo minimizar la duración de esta tarea a 24 horas y obteniendo así un resultado para el Makespan de:

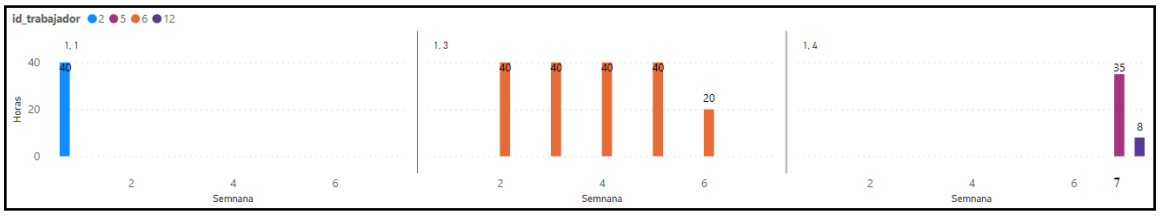
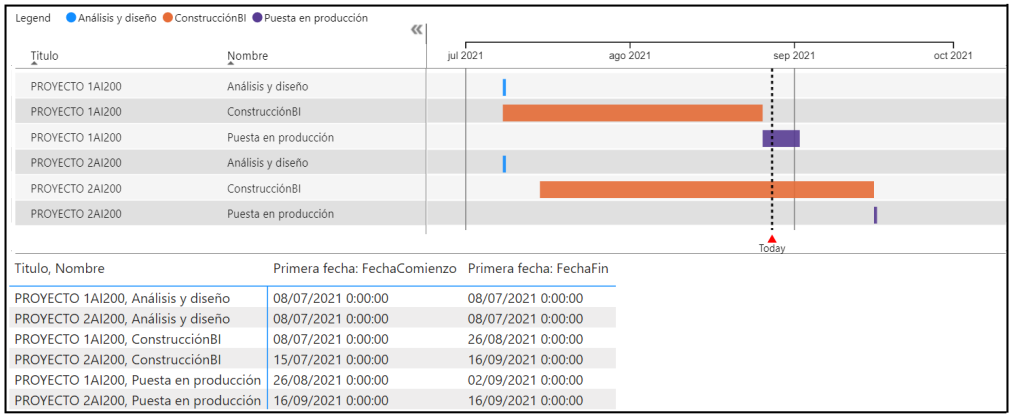


Figura A.11: Dedicación por tareas y semanas de cada trabajador.

$Makespan = \frac{40+180+24}{40} = 6.1$ semanas.

Lo que indica que es imposible terminar el proyecto antes de la sexta semana y que por tanto el valor mínimo de la función objetivo es 7.

La Figura A.12 muestra la planificación para el problema con dos proyectos:



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1AI200, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 2AI200, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 1AI200, ConstrucciónBI	08/07/2021 0:00:00	26/08/2021 0:00:00
PROYECTO 2AI200, ConstrucciónBI	15/07/2021 0:00:00	16/09/2021 0:00:00
PROYECTO 1AI200, Puesta en producción	26/08/2021 0:00:00	02/09/2021 0:00:00
PROYECTO 2AI200, Puesta en producción	16/09/2021 0:00:00	16/09/2021 0:00:00

Figura A.12: Planificación de dos proyectos de AI con 200 horas totales estimadas.

Con un desglose de horas por trabajador por semana:

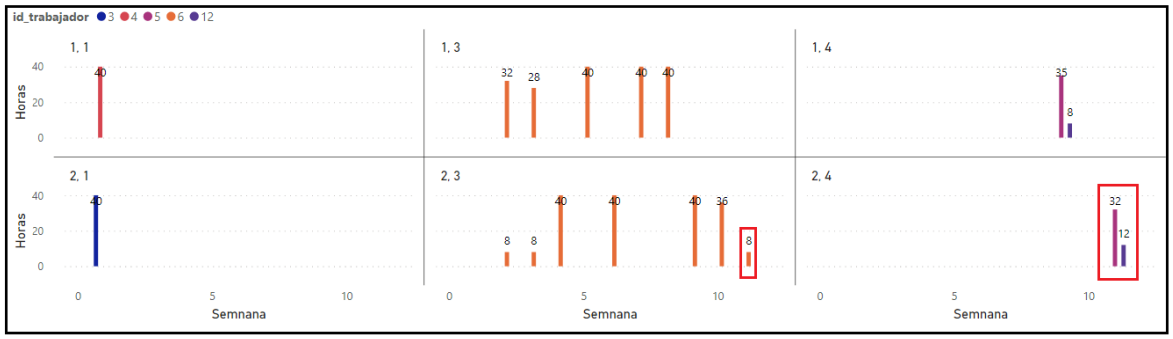


Figura A.13: Dedicación por tareas y semanas de cada trabajador asignado al proyecto.

El valor de la función objetivo en el mínimo en este caso es 11, superando en cuatro unidades el valor obtenido para el problema de un proyecto.

Los cálculos para obtener el valor del mínimo son:

$$Makespan = \frac{40+2\times180+24}{40} = 10.6.$$

Lo que indica que es imposible terminar el proyecto antes de la décima semana, por lo que el valor mínimo de la función objetivo será 11.

Como se ha hecho en el apartado anterior, se puede restringir la fecha de finalización para el primer proyecto a la obtenida en el problema con un solo proyecto y ver si cambia la solución de nuestro problema. La Figura A.14 muestra el reparto de horas.

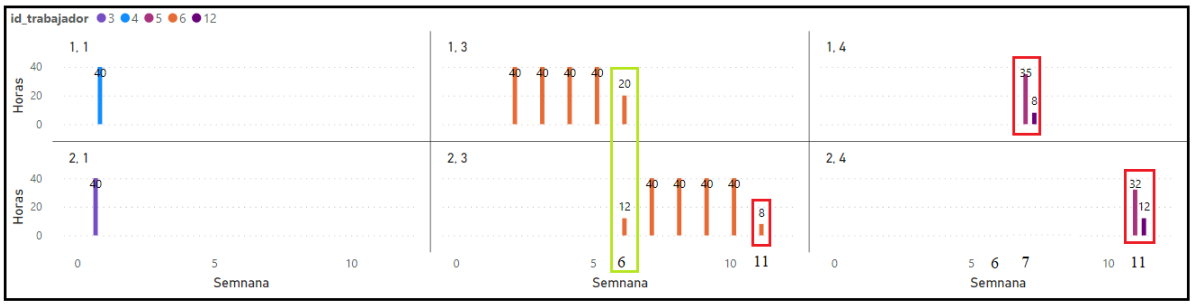
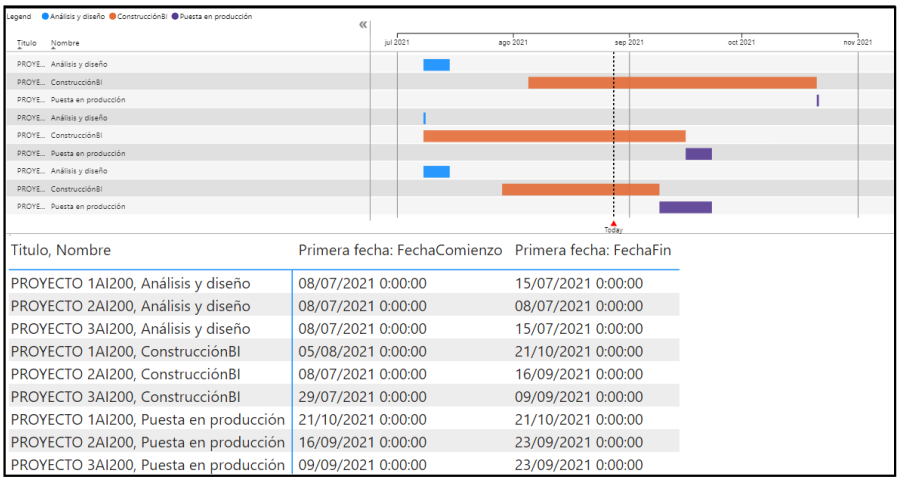


Figura A.14: Dedicación por tareas y semanas de cada trabajador asignado al proyecto.

El valor de la función objetivo es el mismo que el obtenido en el caso anterior, lo que significa que las soluciones son equivalentes. Sin embargo, al restringir al mínimo la fecha de finalización de un proyecto, disminuye el tiempo de ejecución del programa, como podemos ver en la Figura 3.36 del apartado 3.6 del Capítulo 3.

Para el problema con tres proyectos tenemos una planificación como muestra la siguiente figura:



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1AI200, Análisis y diseño	08/07/2021 0:00:00	15/07/2021 0:00:00
PROYECTO 2AI200, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 3AI200, Análisis y diseño	08/07/2021 0:00:00	15/07/2021 0:00:00
PROYECTO 1AI200, ConstrucciónBI	05/08/2021 0:00:00	21/10/2021 0:00:00
PROYECTO 2AI200, ConstrucciónBI	08/07/2021 0:00:00	16/09/2021 0:00:00
PROYECTO 3AI200, ConstrucciónBI	29/07/2021 0:00:00	09/09/2021 0:00:00
PROYECTO 1AI200, Puesta en producción	21/10/2021 0:00:00	21/10/2021 0:00:00
PROYECTO 2AI200, Puesta en producción	16/09/2021 0:00:00	23/09/2021 0:00:00
PROYECTO 3AI200, Puesta en producción	09/09/2021 0:00:00	23/09/2021 0:00:00

Figura A.15: Planificación de tres proyectos de AI con 200 horas totales estimadas.

Con un desglose de horas por trabajador por semana:

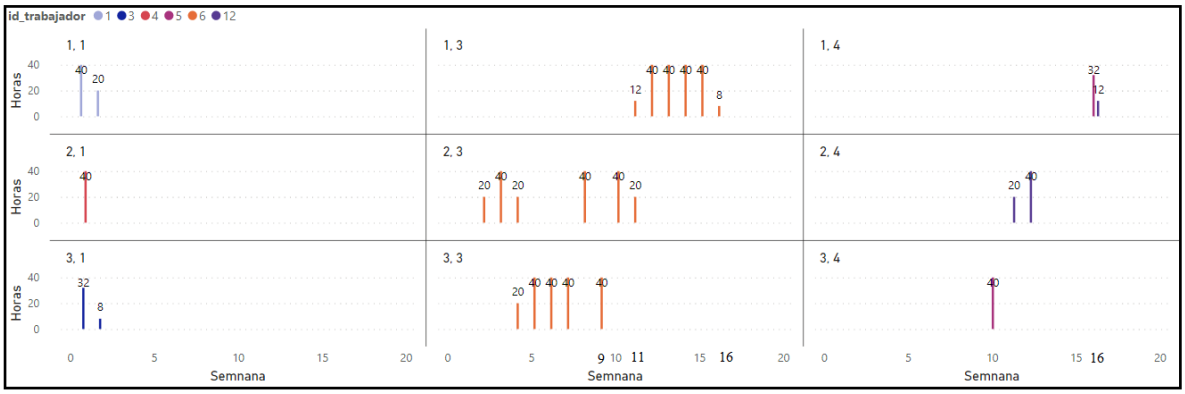


Figura A.16: Dedicación por tareas y semanas de cada trabajador asignado a cada proyecto.

El valor de la función objetivo en el mínimo es 16. Se puede comprobar que la solución es correcta, ya que se puede calcular el mínimo de la función objetivo. Si realizamos los cálculos como en los casos anteriores obtenemos un resultado para el Makespan de:

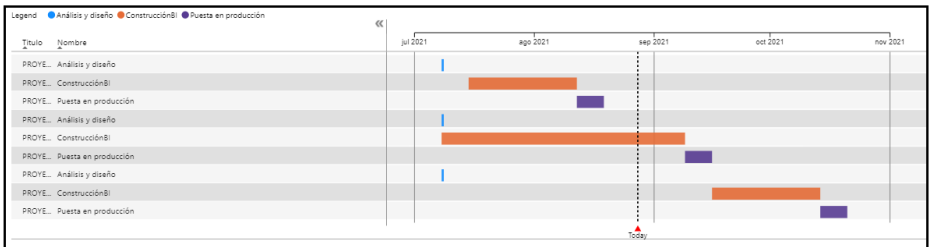
$$Makespan = \frac{40+3\times 180+24}{40} = 15.1.$$

Lo que indica que es imposible terminar el proyecto antes de la semana 15 y que por tanto el valor mínimo de la función objetivo es 16.

El trabajador 6 debe empezar en la semana 11 con la construcción del segundo proyecto, ya que en caso contrario el trabajador 12 no podrá trabajar en su producción las 20 horas establecidas dicha semana.

Se puede restringir la fecha de finalización para dos de los tres proyectos, igualándolas a las obtenidas en el problema de dos proyectos anterior con el objetivo de reducir el tiempo de ejecución del problema.

Los resultados obtenidos han sido los siguientes:



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1AI200CS, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 2AI200CS, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 3AI200CS, Análisis y diseño	08/07/2021 0:00:00	08/07/2021 0:00:00
PROYECTO 1AI200CS, ConstrucciónBI	15/07/2021 0:00:00	12/08/2021 0:00:00
PROYECTO 2AI200CS, ConstrucciónBI	08/07/2021 0:00:00	09/09/2021 0:00:00
PROYECTO 3AI200CS, ConstrucciónBI	16/09/2021 0:00:00	14/10/2021 0:00:00
PROYECTO 1AI200CS, Puesta en producción	12/08/2021 0:00:00	19/08/2021 0:00:00
PROYECTO 2AI200CS, Puesta en producción	09/09/2021 0:00:00	16/09/2021 0:00:00
PROYECTO 3AI200CS, Puesta en producción	14/10/2021 0:00:00	21/10/2021 0:00:00

Figura A.17: Figura en la que vemos la planificación de dos proyectos de AI con 200 horas totales estimadas, donde fijamos la fecha de finalización de los dos primeros proyectos.

Con un desglose de horas por trabajador por semana:

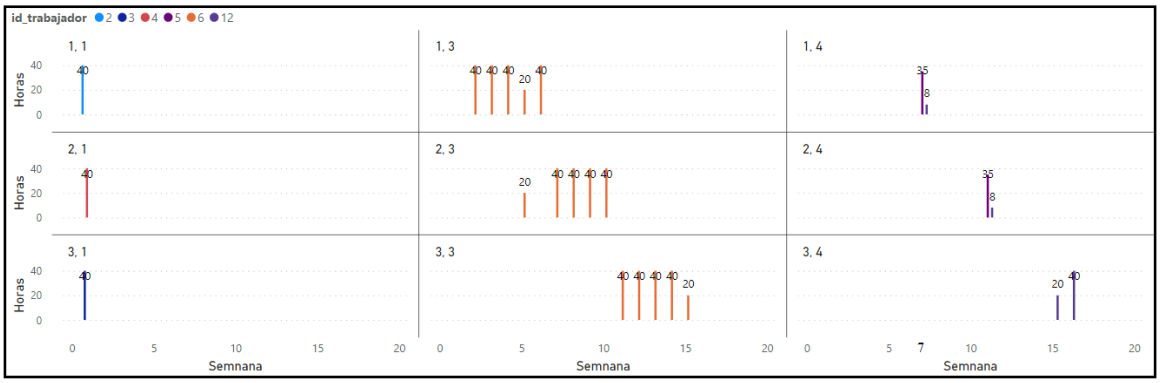


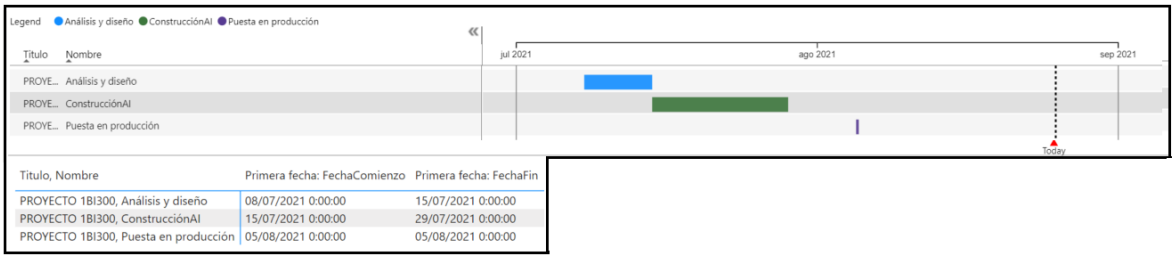
Figura A.18: Figura en la que vemos la dedicación por tareas y semanas de cada tranajador asignado al proyecto.

El valor de la función objetivo es el mismo que el obtenido en el caso anterior, lo que significa que las soluciones serán equivalentes.

A.2. Análisis Proyectos BI

A.2.1. Proyectos BI con 300 horas estimadas

El resultado para el problema en el que se resuelve la planificación de un proyecto individual de Inteligencia Empresarial con 300 horas estimadas es:



Título, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1BI300, Análisis y diseño	08/07/2021 0:00:00	15/07/2021 0:00:00
PROYECTO 1BI300, ConstrucciónAI	15/07/2021 0:00:00	29/07/2021 0:00:00
PROYECTO 1BI300, Puesta en producción	05/08/2021 0:00:00	05/08/2021 0:00:00

Figura A.19: Planificación de un proyecto de BI con 300 horas totales estimadas.

Con un desglose de horas por trabajador por semana:

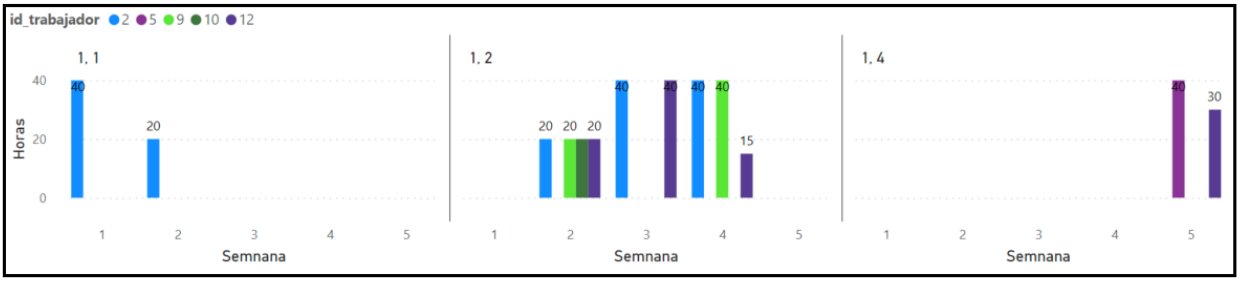
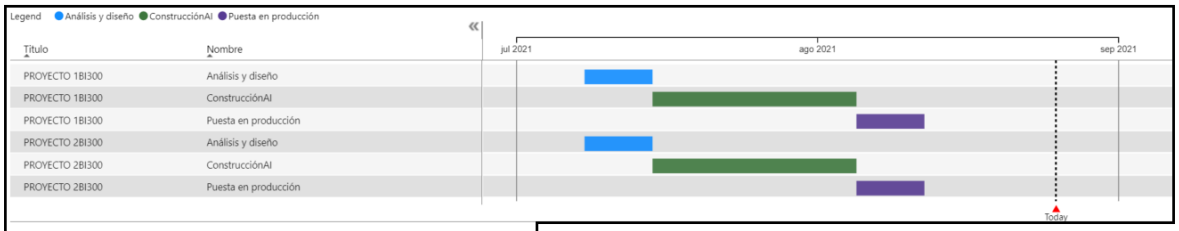


Figura A.20: Dedicación por tareas y semanas de cada trabajador asignado al proyecto.

La ejecución del proyecto solo, necesita de 5 semanas para su realización.

La Figura A.21 muestra la planificación para el problema con dos proyectos:



Título, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1BI300, Análisis y diseño	08/07/2021 0:00:00	15/07/2021 0:00:00
PROYECTO 2BI300, Análisis y diseño	08/07/2021 0:00:00	15/07/2021 0:00:00
PROYECTO 1BI300, ConstrucciónAI	15/07/2021 0:00:00	05/08/2021 0:00:00
PROYECTO 2BI300, ConstrucciónAI	15/07/2021 0:00:00	05/08/2021 0:00:00
PROYECTO 1BI300, Puesta en producción	05/08/2021 0:00:00	12/08/2021 0:00:00
PROYECTO 2BI300, Puesta en producción	05/08/2021 0:00:00	12/08/2021 0:00:00

Figura A.21: Planificación de dos proyecto de BI con 300 horas totales estimadas.

Con un desglose de horas por trabajador por semana:

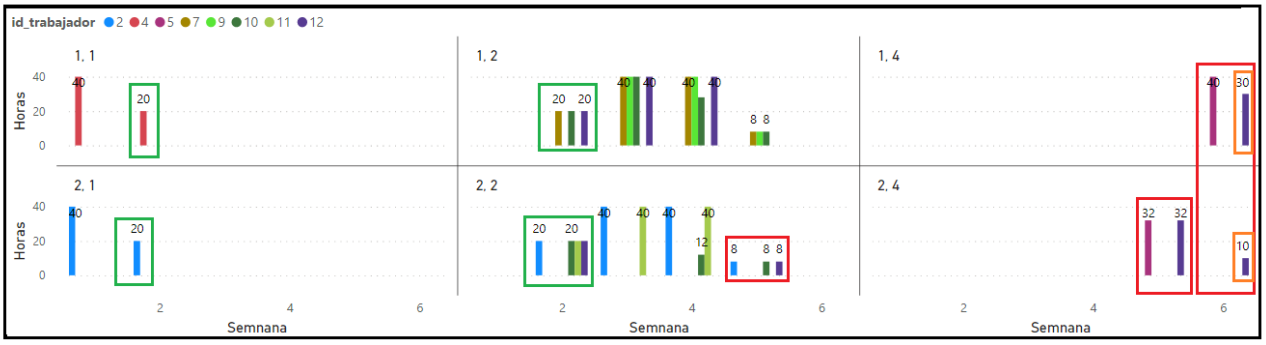


Figura A.22: Dedicación por tareas y semanas de cada trabajador asignado al proyecto.

El valor de la función objetivo en el mínimo es 6, aumentando en una semana el resultado obtenido para el problema con un solo proyecto de las mismas características. En esta solución ambos proyectos terminan la sexta semana. Se puede resolver el problema fijando la fecha de finalización de uno de los proyectos a la obtenida en el problema de un proyecto individual. Los resultados obtenidos son los siguientes:

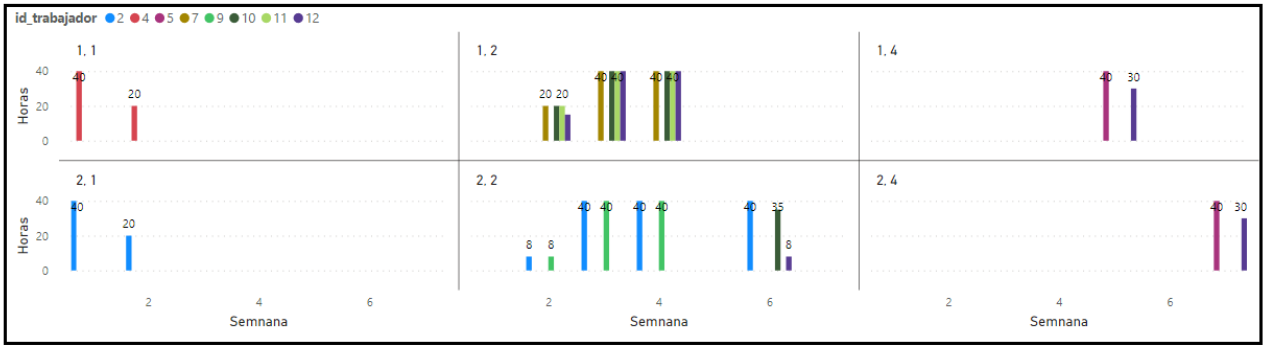
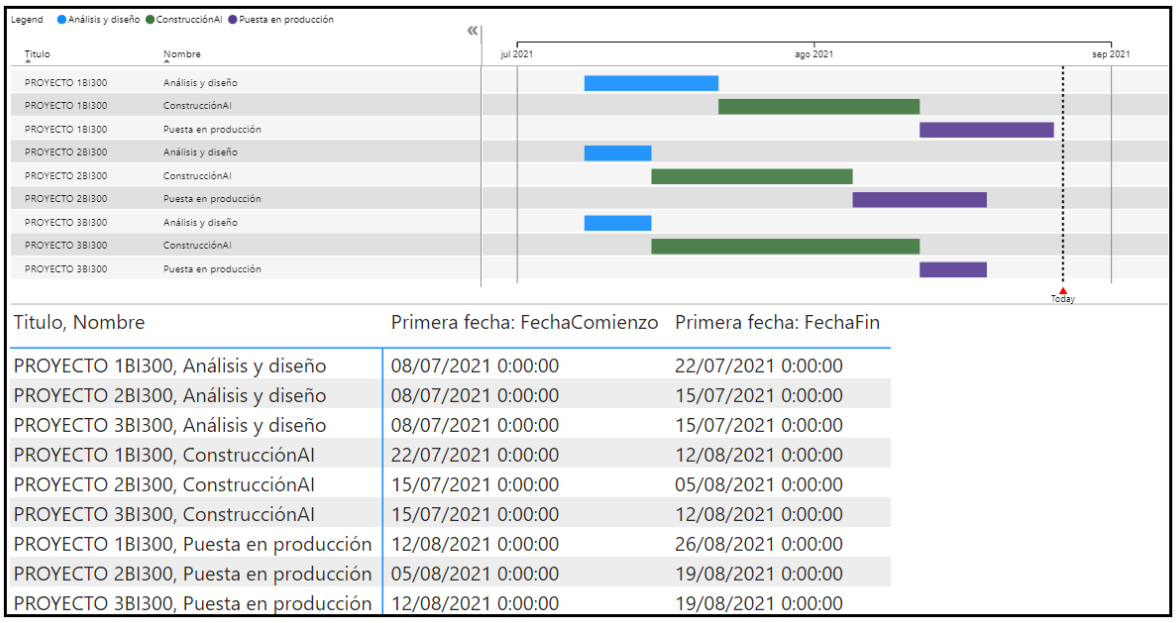


Figura A.23: Dedicación por tareas y semanas de cada trabajador asignado al proyecto.

Al contrario que con la clase de proyectos AI, al intentar resolver el problema fijando la fecha de entrega de uno de los proyectos a la fecha mínima obtenida con el problema de un solo proyecto, el valor de la función objetivo aumenta en una unidad, por lo que las soluciones obtenidas no serán equivalentes en este caso. Esto significa que cuando se ejecutan dos proyectos BI en paralelo, la fecha mínima obtenida para cada uno de ellos de forma individual debe aumentar, como muestra la Figura 3.34 del apartado3.6 del Capítulo 3.

Por ello, en este caso la ventana temporal la introduciremos en ambos proyectos permitiéndoles terminar tanto la quinta como la sexta semana. Los resultados de los tiempos de compilación para ambos casos lo podemos ver en la Figura 3.37.

Para el problema con tres proyectos tenemos una planificación como muestra la siguiente figura:



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1BI300, Análisis y diseño	08/07/2021 0:00:00	22/07/2021 0:00:00
PROYECTO 2BI300, Análisis y diseño	08/07/2021 0:00:00	15/07/2021 0:00:00
PROYECTO 3BI300, Análisis y diseño	08/07/2021 0:00:00	15/07/2021 0:00:00
PROYECTO 1BI300, ConstrucciónAI	22/07/2021 0:00:00	12/08/2021 0:00:00
PROYECTO 2BI300, ConstrucciónAI	15/07/2021 0:00:00	05/08/2021 0:00:00
PROYECTO 3BI300, ConstrucciónAI	15/07/2021 0:00:00	12/08/2021 0:00:00
PROYECTO 1BI300, Puesta en producción	12/08/2021 0:00:00	26/08/2021 0:00:00
PROYECTO 2BI300, Puesta en producción	05/08/2021 0:00:00	19/08/2021 0:00:00
PROYECTO 3BI300, Puesta en producción	12/08/2021 0:00:00	19/08/2021 0:00:00

Figura A.24: Planificación de tres proyectos de BI con 300 horas totales estimadas.

Con un desglose de horas por trabajador por semana:

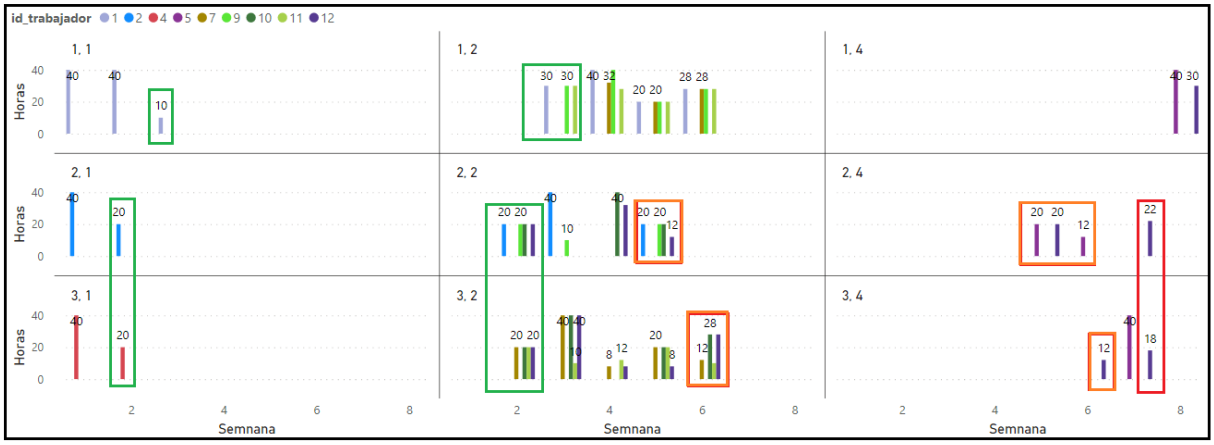
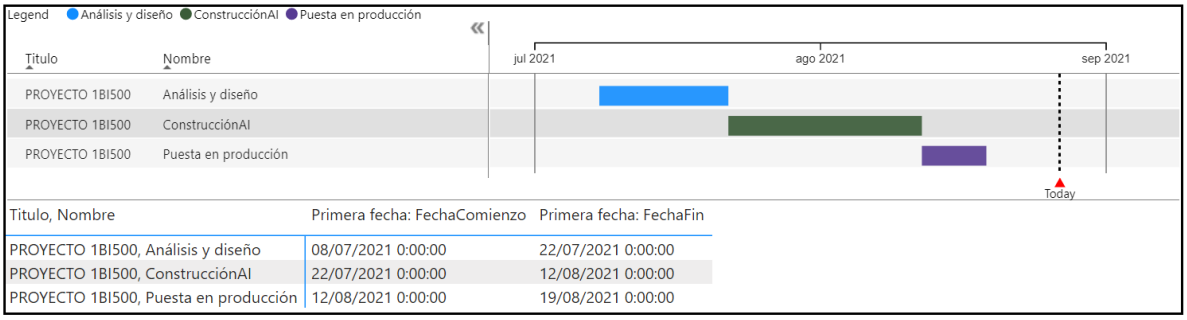


Figura A.25: Dedicación por tareas y semanas de cada trabajador asignado al proyecto.

El valor de la función objetivo en el mínimo es 8, aumentando en una semana el resultado obtenido para el problema con un solo proyecto de las mismas características. Dos de los proyectos terminan la séptima semana para que el tercero en terminar lo haga la octava, lo que vuelve a indicar que en el caso de ejecutar proyectos de BI en paralelo, para minimizar el Makespan global se deben retrasar las fechas de entrega de todos los proyectos cuyas tareas de construcción coincidan en el tiempo en la solución individual de cada uno de ellos.

A.2.2. Proyectos BI con 500 horas estimadas

El resultado para el problema en el que resolvemos la planificación de un proyecto es:



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1BI500, Análisis y diseño	08/07/2021 0:00:00	22/07/2021 0:00:00
PROYECTO 1BI500, ConstrucciónAI	22/07/2021 0:00:00	12/08/2021 0:00:00
PROYECTO 1BI500, Puesta en producción	12/08/2021 0:00:00	19/08/2021 0:00:00

Figura A.26: Planificación de un proyecto de BI con 500 horas estimadas.

Con un desglose de horas por trabajador por semana:

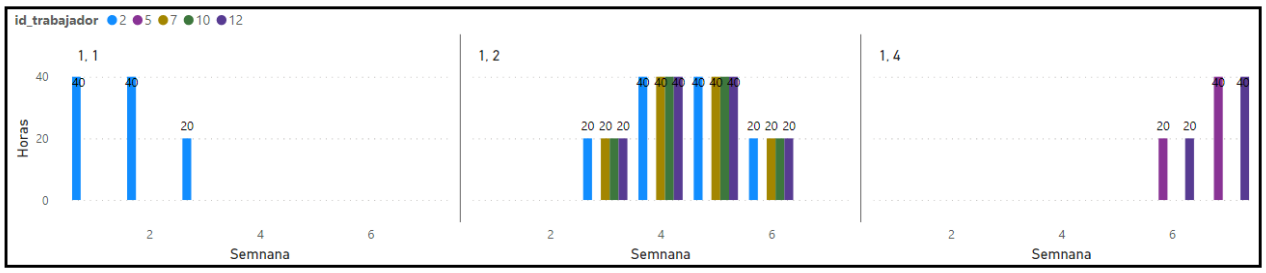
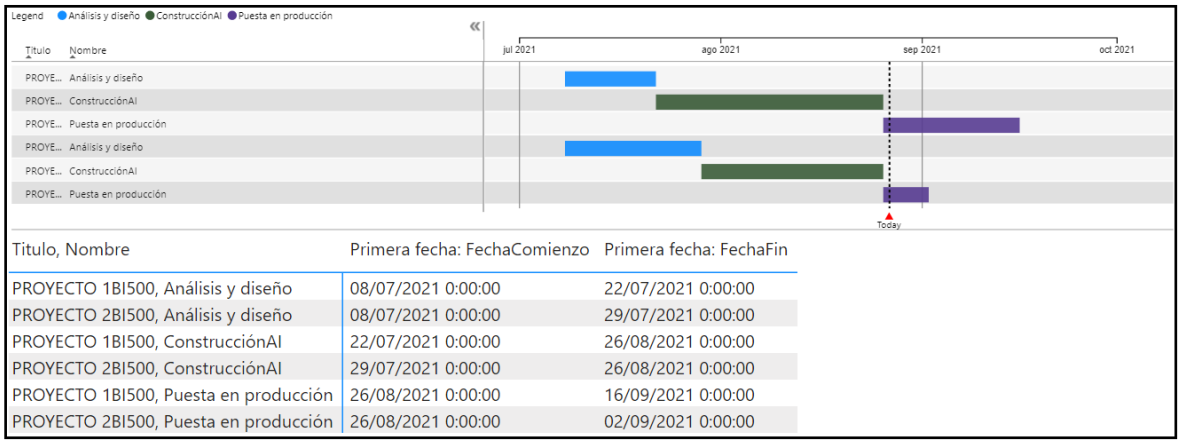


Figura A.27: Dedicación por tareas y semanas de cada trabajador asignado al proyecto.

La ejecución del proyecto necesita de 7 semanas para su realización.

La Figura A.28 muestra la planificación para el problema con dos proyectos:



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1BI500, Análisis y diseño	08/07/2021 0:00:00	22/07/2021 0:00:00
PROYECTO 2BI500, Análisis y diseño	08/07/2021 0:00:00	29/07/2021 0:00:00
PROYECTO 1BI500, ConstrucciónAI	22/07/2021 0:00:00	26/08/2021 0:00:00
PROYECTO 2BI500, ConstrucciónAI	29/07/2021 0:00:00	26/08/2021 0:00:00
PROYECTO 1BI500, Puesta en producción	26/08/2021 0:00:00	16/09/2021 0:00:00
PROYECTO 2BI500, Puesta en producción	26/08/2021 0:00:00	02/09/2021 0:00:00

Figura A.28: Planificación de dos proyecto de BI con 500 horas totales estimadas.

Con un desglose de horas por trabajador por semana:

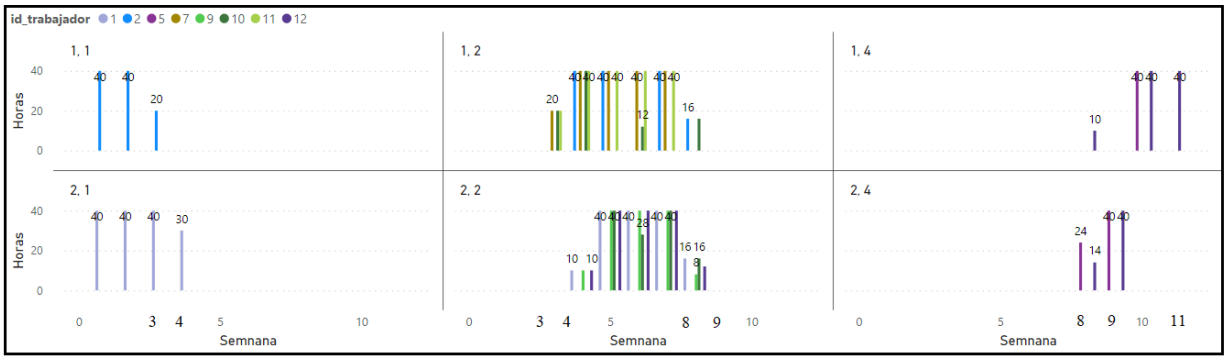


Figura A.29: Dedicación por tareas y semanas de cada trabajador asignado al proyecto.

El valor de la función objetivo en el mínimo es 11, aumentando en cuatro semanas el resultado obtenido para el problema con un solo proyecto de las mismas características. Para que el segundo proyecto acabe la semana 11, el primero debe hacerlo la novena semana, aumentando en dos unidades la el valor obtenido para el problema con un proyecto.Si resolvemos el problema fijando la fecha de finalización de uno de los proyectos a la obtenida en el problema de un proyecto individual el valor de la función objetivo aumenta como se ve en la FiguraA.30, lo que indica de nuevo que la cota de finalización mínima debe ser mayor que la obtenida en el problema de un solo proyecto.

Los resultados obtenidos son los siguientes:

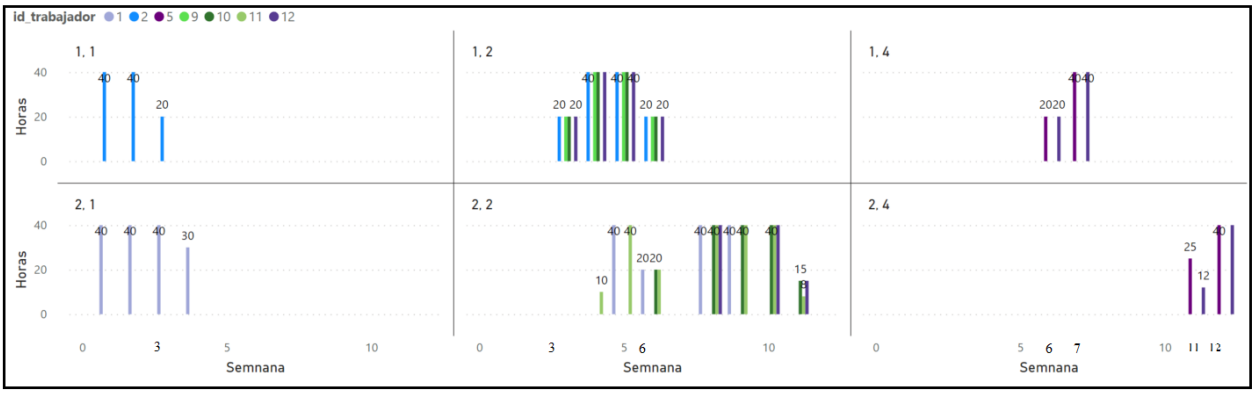
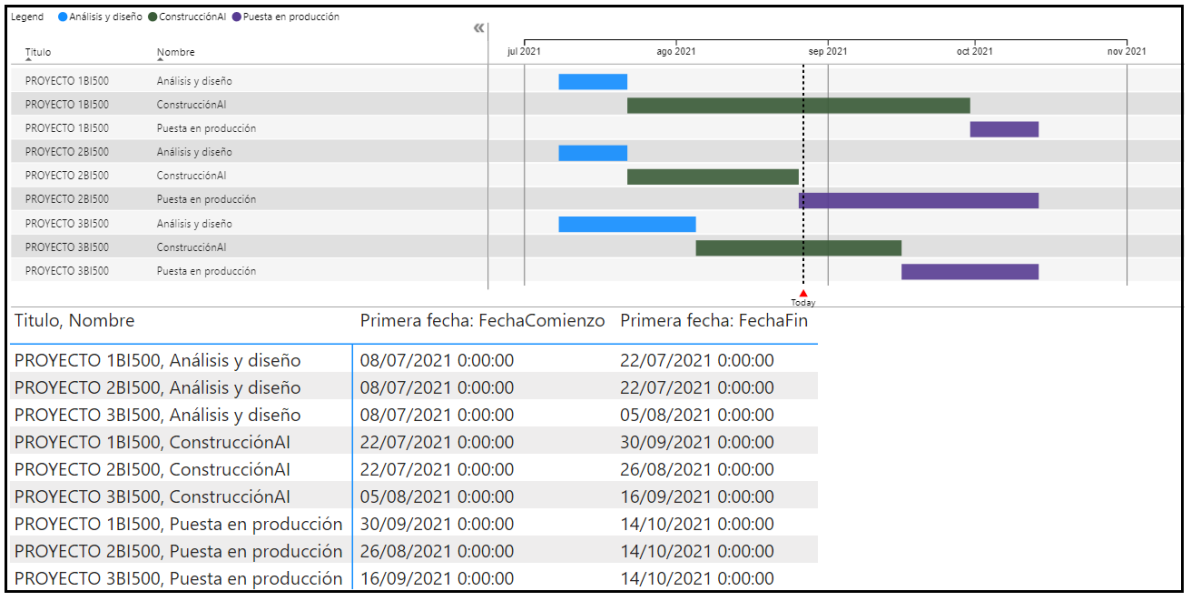


Figura A.30: Dedicación por tareas y semanas de cada trabajador asignado al proyecto.

La Figura A.31 muestra la planificación para el problema con tres proyectos:



Titulo, Nombre	Primera fecha: FechaComienzo	Primera fecha: FechaFin
PROYECTO 1BI500, Análisis y diseño	08/07/2021 0:00:00	22/07/2021 0:00:00
PROYECTO 2BI500, Análisis y diseño	08/07/2021 0:00:00	22/07/2021 0:00:00
PROYECTO 3BI500, Análisis y diseño	08/07/2021 0:00:00	05/08/2021 0:00:00
PROYECTO 1BI500, ConstrucciónAI	22/07/2021 0:00:00	30/09/2021 0:00:00
PROYECTO 2BI500, ConstrucciónAI	22/07/2021 0:00:00	26/08/2021 0:00:00
PROYECTO 3BI500, ConstrucciónAI	05/08/2021 0:00:00	16/09/2021 0:00:00
PROYECTO 1BI500, Puesta en producción	30/09/2021 0:00:00	14/10/2021 0:00:00
PROYECTO 2BI500, Puesta en producción	26/08/2021 0:00:00	14/10/2021 0:00:00
PROYECTO 3BI500, Puesta en producción	16/09/2021 0:00:00	14/10/2021 0:00:00

Figura A.31: Planificación de tres proyectos de BI ejecutados en paralelo con 500 horas totales estimadas.

Con un desglose de horas por trabajador por semana:

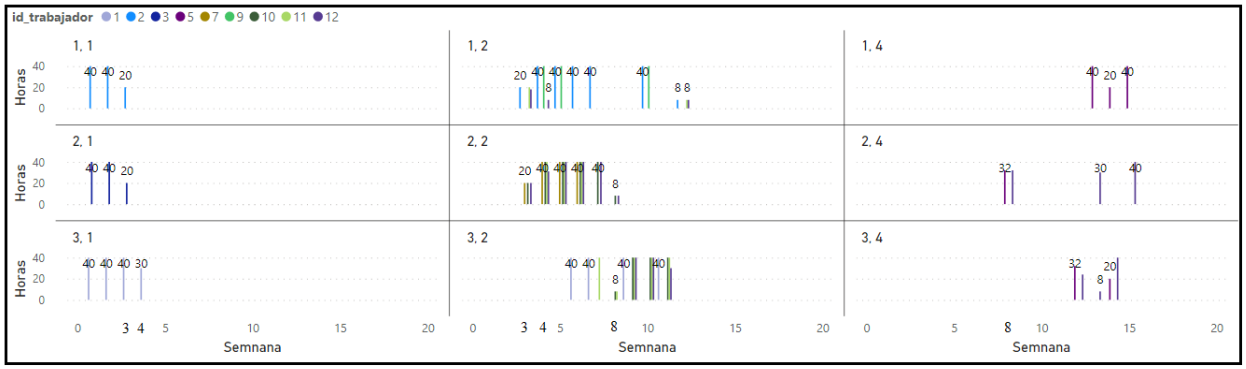


Figura A.32: Dedicación por tareas y semanas de cada trabajador asignado al proyecto.

El valor de la función objetivo en el mínimo es 15, aumentando en ocho semanas el resultado obtenido para el problema con un solo proyecto y en cuatro semanas el resultado obtenido para el problema con dos proyectos con las mismas horas estimadas. En la solución obtenida de la Figura A.32, dos de los tres proyectos terminan la semana 15 y uno la 14. Como la tarea de construcción del segundo proyecto termina al principio de la octava y la puesta en producción puede ejecutarse en menos de dos semanas y el segundo proyecto sería capaz de acabar la novena semana.

La Figura A.33 muestra los resultados obtenidos al acotar la fecha de finalización de uno de los proyectos a la novena semana y la de otro de ellos a la doceaba:

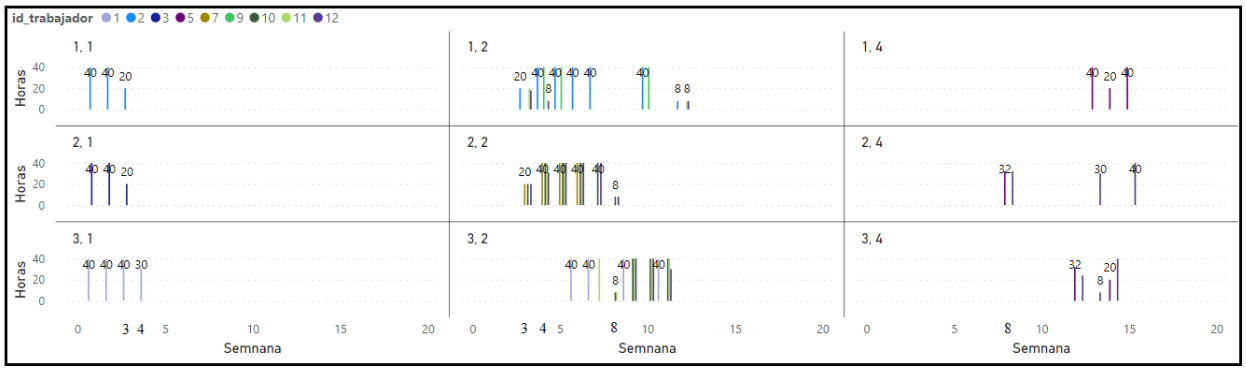


Figura A.33: Dedicación por tareas y semanas de cada trabajador asignado al proyecto.

Como el valor de la función objetivo en el óptimo sigue siendo el mismo, esta solución y la anterior son equivalentes, pero reduciendo el tiempo de compilación.

Anexo B

Código de programación

B.1. Tratamiento de datos en Jupyter Notebook

En este primer apartado del anexo se incorpora el código escrito en Jupyter Notebook con el que se ha realizado el tratamiento de datos, creación de los distintos subconjuntos del modelo y clasificación y guardad o de resultados.

Para ver como se han obtenido los datos como el tiempo de ejecución empleado y las fechas de finalización se incorpora también el modelo de un ejemplo sencillo con el que se pueden ver estos valores.

El código correspondiente a la lectura de los parámetros del problema y su manipulación para introducirlos en el modelo son:

Codigo Jupyter Notebook

September 7, 2021

0.1 IMPORTAR LOS DATOS DESDE EXCELL

[1]:
```
#PARA ELLO DEBEMOS IMPORTAR LA LIBRERIA PANDAS
import pandas as pd
import numpy as np
from itertools import product
from sys import stdout as out
from mip import*

#Importamos los datos desde un excella un dataframe
excel=pd.ExcelFile('/Users/rbaigorri/Documents/TFM/DOCUMENTOS/EJEMPLOS PYTHON/
 ↪TRABAJO/IDLE/FINAL/Pruebas/CoincidenAI.xlsx')
leerProy = 2

personas = pd.read_excel(excel,'Personas',usecols=['id','Nombre','Factor'])
roles = pd.read_excel(excel,'Roles',usecols=['id_rol','Roles'])
rolper = pd.read_excel(excel,'Roles_Personas',usecols=['id_per','id_rol'])
tareas = pd.read_excel(excel,'Tareas',usecols=['id_tar','Tarea'])
proyectos = pd.read_excel(excel,'Proyectos', usecols=['id_proy','Titulo','Fecha␣
 ↪inicio','Horas totales'],nrows=leerProy)
proytareas = pd.read_excel(excel,'Proy_Tareas',␣
 ↪usecols=['id_proy','id_tar','id_rol','Horas'])

nrowsProyTarea=0
for i in range(len(proytareas)):
    if(proytareas['id_proy'][i] <= leerProy):
        nrowsProyTarea = nrowsProyTarea+1

proytareas = pd.read_excel(excel,'Proy_Tareas',␣
 ↪usecols=['id_proy','id_tar','id_rol','Horas'], nrows = nrowsProyTarea)
```

[2]:
```
#Ejemplo de un dataframe
personas
```

[2]:

	id	Nombre	Factor
0	1	Eloy Salcedo	1.5
1	2	Alex Esteban	1.0
2	3	Raúl Baigorri	1.0
3	4	Ricardo Canudas	1.0
4	5	Antonio Gonzalez-Aller	1.0
5	6	Axier Iñiguez	1.5
6	7	Ignacio Tortajada	3.0
7	8	Adrián Lopez	1.0
8	9	Mario Cereza	3.0
9	10	Samuel Lacueva	2.0
10	11	Adrián Martínez	3.0
11	12	Javier Baigorri	1.5

[3]:
```
proytareas
```

[3]:

	id_proy	id_tar	Horas	id_rol
0	1	1	30	1
1	1	3	90	2
2	1	4	30	5
3	1	7	0	1
4	2	1	30	1
5	2	3	90	2
6	2	4	30	5
7	2	7	0	1

[4]:
```
#Inicializacion de datos
RolPersona = np.zeros((len(roles),len(personas)))
ProyTareaRol = np.zeros((len(proyectos),len(tareas),len(roles)))
HorasProy = np.zeros(len(proyectos))
HorasProyTarea = np.zeros((len(proyectos),len(tareas)))
HorasProyTareaRol = np.zeros((len(proyectos),len(tareas),len(roles)))
```

[5]:
```
#Identificación del proyecto más temprano en empezar a ejecutarse
IndiceIni=0
for f in range(len(proyectos)-1):
    if(0 < np.round_((proyectos.iloc[IndiceIni][2]-proyectos.iloc[f+1][2]).days/
 ↪7)):
        IndiceIni=f+1
IndiceIni
```

[5]: 0

[6]:
```
#DEFINICIÓN DE PARÁMETROS, LISTAS POR PROYECTOS Y MATRICES

#Horas totales
for h in range(len(proyectos)):
    HorasProy[h] = int(proyectos['Horas totales'][h])

#Clase de cada proyecto
Clase = []
#for p in range(len(proyectos)):
    #Clase.append(proyectos['TIPO'][p])

#Rol que puede asumir cada trabajador
for n in range(len(rolper)):
    RolPersona[rolper.iloc[n][1]-1][rolper.iloc[n][0]-1]=1

#Factor de ponderación de cada trabajador
Mult=[personas.iloc[n][2] for n in range(len(personas))]

#Horas laborables semanales de cada trabajador
HorasLabSem=[40 for n in range(len(personas))]

#Matriz indicadora de qué rol debe a ejecutar qué tarea de cada proyecto
for n in range(nrowsProyTarea):
    ProyTareaRol[proytareas.iloc[n][0]-1][proytareas.iloc[n][1]-1][proytareas.
 ↪iloc[n][3]-1] = 1

for n in range(nrowsProyTarea):
    HorasProyTareaRol[proytareas.iloc[n][0]-1][proytareas.
 ↪iloc[n][1]-1][proytareas.iloc[n][3]-1] = proytareas['Horas'][n]

Horas=0
for n in range(nrowsProyTarea):
    if (n > 0) & ((proytareas.iloc[n][0] == proytareas.iloc[n-1][0]) &␣
 ↪(proytareas.iloc[n][1] == proytareas.iloc[n-1][1])):
        Horas = Horas + proytareas['Horas'][n]
    else:
        HorasProyTarea[proytareas.iloc[n][0]-1][proytareas.iloc[n][1]-1] =␣
 ↪proytareas['Horas'][n]
        Horas = proytareas['Horas'][n]

    HorasProyTarea[proytareas.iloc[n][0]-1][proytareas.iloc[n][1]-1] = Horas

FechaInicio = [proyectos.iloc[n][2]-proyectos.iloc[IndiceIni][2] for n in␣
 ↪range(len(proyectos))]
FechaInicioSemanas = [int(np.round_(FechaInicio[n].days/7)) for n in␣
 ↪range(len(proyectos))]
```

[7]:
```
#Ejemplo de una matriz creada
print(HorasProyTareaRol[0])
```

```
[[30.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.]
 [ 0. 90.  0.  0.  0.  0.]
 [ 0.  0.  0.  0. 30.  0.]
 [ 0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.]]
```

0.2 Creación de listas representativas de los subconjuntos del modelo

[8]:
```
#subconjunto de roles que puede asumir cada trabajador
RolP=[]
Roles=set()
personas
roles
for p in range(len(personas)):
    Roles=set()
    for r in range(len(roles)):
        if(RolPersona[r][p]==1):
            Roles.add(r)
    RolP.append(Roles)
```

[9]:
```
#Subconjunto de trabajadores que pueden asumir cada rol
PersonaR=[]
Persona=set()

for r in range(len(roles)):
    Persona=set()
    for p in range(len(personas)):
        if(RolPersona[r][p]==1):
            Persona.add(p)
    PersonaR.append(Persona)

PersonaR
```

[9]: `[{0, 1, 2, 3}, {5}, {0, 1, 6, 8, 9, 10, 11}, {4}, {4, 11}, {7}]`

[10]:
```
#Subconjunto de actividades/ tareas por proyecto
Actividad=set()
ActividadB=[]
#ACTIVIDADES DE CADA PROYECTO:
for n in range(len(proyectos)):
    Actividad=set()
    for a in range(len(tareas)):
        Suma=0
        for r in range(len(roles)):
            Suma = Suma + ProyTareaRol[n][a][r]
        if(Suma >= 1):
            Actividad.add(a)
    ActividadB.append(Actividad)

ActividadB
```

[10]:
```
[{0, 2, 3, 6}, {0, 2, 3, 6}]
```

[11]:
```
#Lista de los subconjuntos de  tareas por proyecto, para poder acceder a ellas
 ↪en las restricciones del modelo
ListaActividadB=[]
Act=[]
for n in range(len(ActividadB)):
    Act=[]
    for i in ActividadB[n]:
        Act.append(i)
    ListaActividadB.append(Act)

ListaActividadB
```

[11]:
```
[[0, 2, 3, 6], [0, 2, 3, 6]]
```

[12]:
```
#Subconjunto de roles por tarea
RolA=[]
Rol=set()

for a in range(len(tareas)):
    Rol=set()
    for b in range(len(proyectos)):
        for r in range(len(roles)):
            if(ProyTareaRol[b][a][r] == 1):
                Rol.add(r)
    RolA.append(Rol)
```

[13]:
```
#Subconjunto de roles por tarea de cada proyecto
RolPA=[]
Rol=set()
RolAct=[]

for b in range(len(proyectos)):
    RolAct=[]
    for a in range(len(tareas)):
        Rol=set()
        for r in range(len(roles)):
            if(ProyTareaRol[b][a][r] == 1):
                Rol.add(r)
        RolAct.append(Rol)
    RolPA.append(RolAct)
```

[14]:
```
#Se crea una lista para poder acceder a los datos en algunas restricciones.
ListaRolPA=[]
Actividad=[]
Rol=[]
for n in range(len(proyectos)):
    Rol=[]
    for i in RolPA[n]:
        Actividad=[]
        for j in i:
            Actividad.append(j)
        Rol.append(Actividad)
    ListaRolPA.append(Rol)
```

[15]:
```
#También se crea el subconjunto de trabajadores aptos para realizar cada tarea␣
 ↪de cada proyecto.
PersonaPA=[]
Persona=set()
PersonaAct=[]
PersonaProy=[]

for b in range(len(proyectos)):
    PersonaAct=[]
    for a in range(len(tareas)):
        Persona=set()
        for w in range(len(personas)):
            if(RolPA[b][a] & RolP[w] != set()):
                Persona.add(w)
        PersonaAct.append(Persona)
    PersonaPA.append(PersonaAct)
```

[16]:
```
#Se crea los subconjuntos de los roles aptos o escogidos para cada tartea de␣
 ↪cada proyecto, ya que estos son específicos
RolPAR=[]
Rolpa=[]
Rol=set()
RolAct=[]

for b in range(len(proyectos)):
    Rolpa=[]
    for a in range(len(tareas)):
        RolAct=[]
        for r in range(len(roles)):
            Rol=set()
            if(ProyTareaRol[b][a][r] == 1):
                Rol.add(r)
                RolAct.append(Rol)
            else:
                RolAct.append(Rol)
        Rolpa.append(RolAct)
    RolPAR.append(Rolpa)
```

[17]:
```
#También el subconjunto de trabajadores aptos para realizar cada tarea de cada␣
 ↪proyecto.
PersonaPAR=[]
Persona=set()
PersonaAct=[]
PersonaProy=[]

for b in range(len(proyectos)):
    PersonaProy=[]
    for a in range(len(tareas)):
        PersonaAct=[]
        for r in range(len(roles)):
            Persona=set()
            for w in range(len(personas)):
                if(RolPAR[b][a][r] & RolP[w] != set()):
                    Persona.add(w)
            PersonaAct.append(Persona)
        PersonaProy.append(PersonaAct)
    PersonaPAR.append(PersonaProy)
```

[18]:
```
#Subconjunto de roles que pueden realizar cada tarea
ActividadR = []
Actividad =set()

for r in range(len(roles)):
    Actividad =set()
    for b in range(len(proyectos)):
        for a in range(len(tareas)):
            if(ProyTareaRol[b][a][r] == 1):
                Actividad.add(a)
    ActividadR.append(Actividad)
```

0.3 Conjuntos del modelo

[19]:
```
ntrabajadores=len(personas)
nroles=len(roles)
ntareas=len(tareas)
nproyectos=len(proyectos)

njefes=0
for n in range(len(rolper)):
    if rolper['id_rol'][n]==1:
        njefes = njefes + 1

workers = range(ntrabajadores)
managers = range(njefes)
roles = range(nroles)
tasks = range(ntareas)
projects = range(nproyectos)
nsemanas=20
weeks = range(nsemanas)

#CONJUNTOS DEL MODELO
CWORKERS = set()
for w in range(len(personas)):
    CWORKERS.add(w)

CTASKS =set()
for a in range(len(tasks)):
    CTASKS.add(a)

CROLES =set()
for r in range(len(roles)):
    CROLES.add(r)

CPROJECTS =set()
for p in range(len(projects)):
    CPROJECTS.add(p)
```

0.4 Definición de funciones para incorporar en las restricciones

[20]:
```
def ListaProyectosDistintos(arg1,arg2):
    """Lista de todos los proyectos arg1 sin el introducido: arg2"""
    i=0
    Lista = []
    for i in range(arg1):
        if(i != arg2):
            Lista.append(i)
    return Lista
```

[21]:
```
def ListaTareasDistintas(arg1,arg2):
    """Lista de todos los proyectos arg1 sin el introducido: arg2"""
    i=0
    Lista = []
    for i in arg1:#range(len(arg1)):
        if(i != arg2):
            Lista.append(i)

    return Lista
```

[22]:
```
#Lista de tareas que sean posteriores a una dada

def ListaPosteriorTareas(arg1,arg2):
    """Lista de todos los proyectos arg1 sin el introducido: arg2
    Como la lista viene ordenada, hasta que i no sea mayor que arg2
    no introduciremos la tarea en la lista"""
    i=0
    Lista = []
    for i in arg1:#range(len(arg1)):
        if(i > arg2):
            Lista.append(i)

    return Lista
```

[23]:
```
#Tarea posterior a una dada
def TareaSiguiente(arg1,arg2):
    """Introduciendo la lista de tareas y una tarea dada, la función nos␣
 ↪devolverá la tarea siguiente en la lista"""
    i=0
    j=0
    Tarea = 0
    for i in arg1:#range(len(arg1)):
        if((i > arg2) & (j<1)):
            Tarea = i
            j = j+1

    return Tarea
```

[24]:
```
#Lista de tareas que sean anteriores a una dada
def ListaAnteriorTareas(arg1,arg2):
    """Lista de todos los proyectos arg1 sin el introducido: arg2
    Como la lista viene ordenada, hasta que i no sea mayor que arg2
    no introduciremos la tarea en la lista"""
    i=0
    Lista = []
    for i in arg1:
        if(i < arg2):
            Lista.append(i)

    return Lista
```

[25]:
```
#Última tarea de una lista
def ListaUltimaTarea(arg1):
    """Lista de la última tarea real de la lista"""
    '''Introduciremos una lista de tareas de un proyecto dado, donde la última␣
 ↪siempre será la tarea ficticia, por tanto '''
    i=0
    Lista = []
    for i in range(arg1):
        if(i != arg2):
            Lista.append(i)

    return Lista
```

0.5 Ventanas temporales

[26]:
```
FechaMaximaSemanas= [5, 8]
FechaMinimaSemanas= [5, 8]
```

1 MODELO

[27]:
```
model = Model()
```

[28]:
```
#Definición de variables binarias
z = [[[model.add_var('z({},{},{})'.format(p+1, t+1, w+1),var_type=BINARY) for w␣
 ↪in workers] for t in tasks] for p in projects]

y = [[model.add_var('y({},{})'.format(p+1, w+1), var_type=BINARY) for w in␣
 ↪workers] for p in projects]

x = [[[[model.add_var('x({},{},{},{})'.format( p+1, t+1, w+1,␣
 ↪s+1),var_type=BINARY) for s in weeks] for w in workers] for t in tasks] for p␣
 ↪in projects]

I = [[[model.add_var('I({},{},{})'.format(p+1, t+1, s+1),var_type=BINARY) for s␣
 ↪in weeks] for t in tasks] for p in projects]

O = [[[model.add_var('O({},{},{})'.format(p+1, t+1, s+1),var_type=BINARY) for s␣
 ↪in weeks] for t in tasks] for p in projects]

F = [[[model.add_var('F({},{},{})'.format(p+1,t+1,s+1),var_type=BINARY) for s in␣
 ↪weeks] for t in tasks] for p in projects]
```

[29]:
```
#Definición de variables enteras
u = [[[[model.add_var('HATS({},{},{},{})'.format(p+1, t+1, w+1, s+1)) for s in␣
 ↪weeks] for w in workers] for t in tasks] for p in projects]

H = [[model.add_var('HTS({},{})'.format(w+1, s+1)) for s in weeks] for w in␣
 ↪workers]

M = [[[model.add_var('HMax({},{},{})'.format(p+1, t+1, s+1)) for s in weeks] for␣
 ↪t in tasks] for p in projects]

Makespan = model.add_var('Makespan')
```

[30]:
```
#Función objetivo
model.objective = minimize(Makespan)
```

[31]:
```
#MINIMIZARMAKESPAN
for p in projects:
    for s in weeks:
        model += (s+1)*F[p][ntareas-1][s] <= Makespan
```

[32]:
```
#IMPLICACIÓN DEL TRABAJADOR EN EL PROYECTO
for p in projects:
    for w in workers:
        model += xsum(z[p][a][w] for a in ActividadB[p]) <= ntareas*y[p][w]

for p in projects:
    for w in workers:
        model += xsum(z[p][a][w] for a in ActividadB[p]) >= y[p][w]

#Si está asignado, trabajará una semana o más.
#Además solo podrá trabajar las semanas después de su fecha de iniciación y␣
 ↪antes que la de entrega
for p in projects:
    for a in ActividadB[p]:
        for w in workers:
            model += z[p][a][w] <= xsum(x[p][a][w][s] for s in weeks)

for p in projects:
    for a in ActividadB[p]:
        for w in workers:
            for s in weeks:
                model += x[p][a][w][s] <= z[p][a][w]

#Todas las actividades de los proyectos que no deban realizarse no serán␣
 ↪asignadas a ningún empleado
for p in projects:
    for a in (CTASKS-ActividadB[p]):
        for w in workers:
            for s in weeks:
                model += x[p][a][w][s]  == 0

#Máximo de 4 personas por tarea de proyecto:
for p in projects:
    for a in ActividadB[p]:
        model += xsum(z[p][a][w] for w in workers) <= 4
```

[33]:
```
#RESTRICCIONES DE ROL:
#Deberá de haber solo un jefe de proyecto por cada proyecto
for p in projects:
    model += xsum(y[p][w] for w in PersonaR[0])   == 1

#La última tarea, que es ficticia la asignaremos al jefe de proyecto
for p in projects:
    model += xsum(z[p][ntareas-1][w] for w in PersonaR[0])   == 1

for p in projects:
    model += xsum(z[p][ntareas-1][w] for w in (CWORKERS - PersonaR[0]))   == 0

#RESTRICCIONES COMPATIBILIDAD
#El trabajador podrá desempeñar una tarea solo si está capacitado.
for p in projects:
    for a in ActividadB[p]:
        for s in weeks:
            for w in workers:
                if (RolPA[p][a] & RolP[w] ==  set()):
                    model += x[p][a][w][s] == 0

#Todas actividades asignadas al menos a un empleado:
for p in projects:
    for a in ActividadB[p]:
        model += xsum(z[p][a][w] for w in workers) >= 1
            #model += xsum(x[p][t][w] for w in workers)  >=
 ↪ProyTareaRol[5][t][r]
```

[34]:
```
#RESTRICCIONES HORARIAS
#Inviertirá horas esa semana si trabaja
for p in projects:
    for a in range(len(ListaActividadB[p])-1):
        for w in workers:
            for s in weeks:
                model += u[p][ListaActividadB[p][a]][w][s] <=␣
 ↪HorasLabSem[w]*x[p][ListaActividadB[p][a]][w][s]

#Si trabaja una semana en una tarea de un proyecto que invierta más de 8 horas/␣
 ↪un dia al menos
for p in projects:
    for a in range(len(ListaActividadB[p])-1):
    #for a in ListaActividadB[p]:
        for w in workers:
            for s in weeks:
                model += u[p][ListaActividadB[p][a]][w][s] >=␣
 ↪8*x[p][ListaActividadB[p][a]][w][s]

for w in workers:
    for s in weeks:
        model += xsum(u[p][a][w][s] for a in ActividadB[p] for p in projects) <=␣
 ↪HorasLabSem[w]

#Para todos los trabajadores, todas las semanas, todos los proyectos, el tiempo␣
 ↪invertido máximo de una actividad + la siguiente sucesiva
#del mismo proyecto + el resto de horas invertidas deberá ser menor que 40
for w in workers:
    for s in weeks:
        for p in projects:
            #Para todas las tareas menos para la última
            for a in ListaTareasDistintas(ActividadB[p],6):
                model +=  M[p][a][s]  + xsum(u[p][alpha][w][s] for alpha in␣
 ↪ListaPosteriorTareas(ActividadB[p],a)) + xsum(u[b][alpha][w][s] for alpha in␣
 ↪ListaPosteriorTareas(ActividadB[p],a) for b in␣
 ↪ListaProyectosDistintos(len(projects),p)) <= 40

for w in workers:
    for s in weeks:
        model += xsum(u[p][a][w][s] for a in ActividadB[p] for p in projects) ==␣
 ↪H[w][s]
```

[35]:
```
#RESTRICCIONES SOBRE EL DESARROLLO DE LOS PROYECTOS
for p in projects:
    for a in ActividadB[p]:
        for r in ListaRolPA[p][a]:
            model += xsum(u[p][a][w][s]/ Mult[w] for s in weeks for w in␣
 ↪PersonaPAR[p][a][r]) == HorasProyTareaRol[p][a][r]
```

[36]:
```
#SECUENCIACION DE TAREAS EN UNA MISMA SEMANA
for p in projects:
    for a in ActividadB[p]:
        for w in workers:
            for s in weeks:
                model += M[p][a][s] >= u[p][a][w][s]

for p in projects:
    for s in weeks:
        model += xsum(M[p][a][s] for a in ActividadB[p]) <= 40
```

[37]:
```
#FLUJO DE TAREAS
#Restricción del inicio de la tarea:
for p in projects:
    if (FechaInicioSemanas[p] > 0):
        model += I[p][0][FechaInicioSemanas[p]-1] == 0

for p in projects:
    for t in range(ntareas-1):
        for s in weeks:
            model += I[p][t][s] >= I[p][t+1][s]

for p in projects:
    for t in tasks:
        #for s in range(nsemanas-1):
        for s in range(nsemanas-1):
            model += I[p][t][s] <= I[p][t][s+1]


for p in projects:
    for t in range(ntareas-1):
        for s in weeks:
            model += O[p][t][s] == I[p][t][s]-I[p][t+1][s]

for p in projects:
    for s in weeks:
        model += O[p][ntareas-1][s] == I[p][ntareas-1][s]
```

```
#Solo acaba una vez
#Y lo hace la semana correspondiente
for p in projects:
    for t in tasks:
        model += xsum(F[p][t][s] for s in weeks) == 1

#Restringimos la fecha de finalización con las cotas encontradas anteriormente
for p in projects:
    model += xsum(F[p][ntareas-1][s] for s in␣
 ↪range(FechaMaximaSemanas[p],nsemanas)) == 0

#Con esta segunda y la anterior de la anterior valdría
for p in projects:
    model += xsum(F[p][ntareas-1][s] for s in␣
 ↪range(FechaMinimaSemanas[p]-1,FechaMaximaSemanas[p])) == 1

for p in projects:
    for t in range(1,ntareas):
        for s in range(1,nsemanas):
            model += F[p][t-1][s] == I[p][t][s]-I[p][t][s-1]

#La tarea ficticia terminará la misma semana que la última tarea real
for p in projects:
    for s in weeks:
        #model += F[p][ntareas-1][s] == F[p][ntareas-2][s]
        model += F[p][ntareas-1][s] ==␣
 ↪F[p][ListaActividadB[p][len(ListaActividadB[p])-2]][s]

for p in projects:
    for w in workers:
        for t in range(ntareas-1):
            for s in weeks:
                model += x[p][t][w][s] <= O[p][t][s]+F[p][t][s]

for p in projects:
    for w in workers:
        for s in weeks:
            model += x[p][ntareas-1][w][s] <= O[p][ntareas-1][s]
```

[38]:
```
print('Model has {} vars, {} constraints and {} nzs'.format(model.num_cols,␣
 ↪model.num_rows, model.num_nz))
```

```
Model has 8273 vars, 16246 constraints and 43285 nzs
```

2 Resolución del problema

[39]:
```
from time import time
# optimizing
start_time = time()
status = model.optimize(max_seconds = 6000)#300 900)3000
#Máximo de 10 mins
elapsed_time = np.round(time() - start_time)

elapsed_time
```

[39]: 1.0

[40]:
```
if status == OptimizationStatus.OPTIMAL:
    print('optimal solution cost {} found'.format(model.objective_value))
elif status == OptimizationStatus.FEASIBLE:
    print('sol.cost {} found, best possible: {}'.format(model.objective_value,␣
 ↪model.objective_bound))
elif status == OptimizationStatus.NO_SOLUTION_FOUND:
    print('no feasible solution found, lower bound is: {}'.format(model.
 ↪objective_bound))

#if status == OptimizationStatus.OPTIMAL or status == OptimizationStatus.
 ↪FEASIBLE:
    #print('solution:')
    #for v in model.vars:
        #if abs(v.x) > 1e-6: # only printing non-zeros
            #print('{} : {}'.format(v.name, v.x))
```

```
optimal solution cost 8.0 found
```

3 Obtención de datos

3.1 Fin y comienzo de proyectos

[41]:
```
arcocomienzo =[]
#arcofin tiene todas las finclizacionesen semanas de todas las tareas de los␣
 ↪proyectos
ComienzoTareaProyecto =[]

for v in model.vars:#if (abs(v.x) > 1e-6) & (v.name == 'Semanas(3,1)'): # only␣
 ↪printing non-zeros
    if (v.name[:1] == 'I') & (abs(v.x) > 1e-6):
        #print(v.name)
        indiceComa1 = v.name.index(',')
        indiceComa2 = v.name.index(',',indiceComa1+1)
        indice3 = v.name.index(')',indiceComa2+1)
        arcocomienzo.append((int(v.name[2:indiceComa1]),int(v.name[indiceComa1+1:
 ↪indiceComa2]),int(v.name[indiceComa2+1:indice3])))
```

[42]:
```
ComienzoTareaProyecto =[]
for i in range(len(arcocomienzo)):
    if((arcocomienzo[i][1] != arcocomienzo[i-1][1])|(arcocomienzo[i][1] !=␣
 ↪arcocomienzo[i-1][1])|(i == 0)): #& (arcocomienzo[i][1] != 7)):
        print(arcocomienzo[i])
        ComienzoTareaProyecto.append([arcocomienzo[i][0], arcocomienzo[i][1],␣
 ↪arcocomienzo[i][2]])
```

```
(1, 1, 1)
(1, 2, 1)
(1, 3, 1)
(1, 4, 5)
(1, 5, 5)
(1, 6, 5)
(1, 7, 5)
(2, 1, 1)
(2, 2, 2)
(2, 3, 2)
(2, 4, 8)
(2, 5, 8)
(2, 6, 8)
(2, 7, 8)
```

[43]:
```
ComienzoTareaProyectoB=[]
ComienzoTarea=[]
p = 0
for i in range(len(ComienzoTareaProyecto)):
    if((ComienzoTareaProyecto[i][0] == ComienzoTareaProyecto[i-1][0]) | (i ==␣
 ↪0)):
        for a in ListaActividadB[p]:
            if(ComienzoTareaProyecto[i][1]-1 == a):
                ComienzoTarea.append(ComienzoTareaProyecto[i])
                #print(ComienzoTarea)

    else:
        ComienzoTareaProyectoB.append(ComienzoTarea)
        ComienzoTarea=[]
        ComienzoTarea.append(ComienzoTareaProyecto[i])
        p = p + 1
ComienzoTareaProyectoB.append(ComienzoTarea)

ComienzoTareaProyectoB
```

[43]:
```
[[[1, 1, 1], [1, 3, 1], [1, 4, 5], [1, 7, 5]],
 [[2, 1, 1], [2, 3, 2], [2, 4, 8], [2, 7, 8]]]
```

[44]:
```
ComienzoTareaProyecto =[]
p = 0
i = 0
for p in projects:
    for i in range(len(ComienzoTareaProyectoB[p])):
        if(ComienzoTareaProyectoB[p][i][1] != 7):
            ComienzoTareaProyecto.append(ComienzoTareaProyectoB[p][i])
```

[45]:
```
arcofin =[]
FinTareaProyecto =[]

for v in model.vars:#if (abs(v.x) > 1e-6) & (v.name == 'Semanas(3,1)'): # only␣
 ↪printing non-zeros
    if (v.name[:1] == 'F') & (abs(v.x) > 1e-6):
        indiceComa1 = v.name.index(',')
        indiceComa2 = v.name.index(',',indiceComa1+1)
        indice3 = v.name.index(')',indiceComa2+1)
        arcofin.append((int(v.name[2:indiceComa1]),int(v.name[indiceComa1+1:
 ↪indiceComa2]),int(v.name[indiceComa2+1:indice3])))
        FinTareaProyecto.append([int(v.name[2:indiceComa1]),int(v.
 ↪name[indiceComa1+1:indiceComa2]),int(v.name[indiceComa2+1:indice3])])
```

```
        for i in (ComienzoTareaProyecto):
            if(int(v.name[indiceComa1+1:indiceComa2]) == i[1]):
                if(int(v.name[indiceComa1+1:indiceComa2]) == 7):
                    FinProyecto.append(int(v.name[indiceComa2+1:indice3]))
```

[46]:
```
FinTareaProyectoB=[]
FinTarea=[]
FinProyecto =[]
p = 0
for i in range(len(FinTareaProyecto)):
    if((FinTareaProyecto[i][0] == FinTareaProyecto[i-1][0]) | (i == 0)):
        for a in ListaActividadB[p]:
            if(FinTareaProyecto[i][1]-1 == a):
                FinTarea.append(FinTareaProyecto[i])
                #print(ComienzoTarea)

        if(FinTareaProyecto[i][1]-1 == 6):
            FinProyecto.append(FinTareaProyecto[i][2])

    else:
        FinTareaProyectoB.append(FinTarea)
        FinTarea=[]
        FinTarea.append(FinTareaProyecto[i])
        p = p + 1

FinTareaProyectoB.append(FinTarea)
```

[47]:
```
print(FinProyecto)
```

```
[5, 8]
```

[48]:
```
FinTareaProyecto =[]
for p in projects:
    for i in range(len(FinTareaProyectoB[p])):
        if(FinTareaProyectoB[p][i][1] != 7):
            FinTareaProyecto.append(FinTareaProyectoB[p][i])
FinTareaProyecto
```

[48]:
```
[[1, 1, 1], [1, 3, 5], [1, 4, 5], [2, 1, 2], [2, 3, 8], [2, 4, 8]]
```

3.2 FECHAS

[49]:
```
import datetime
Inicio = proyectos.iloc[IndiceIni][2]
```

[50]:
```
FechaComienzoTareaProy=[]
FechaFinTareaProy=[]

for p in projects:
    for a in range(len(ComienzoTareaProyecto)):
        #print(Inicio + datetime.timedelta(days = ComienzoTareaProyecto[a][2]))
        FechaComienzoTareaProy.append(Inicio + datetime.timedelta(weeks =␣
 ↪ComienzoTareaProyecto[a][2]))
        #FechaFinTareaProy.append(Inicio + datetime.timedelta(days =␣
 ↪FinTareaProyecto[a][2]))
        FechaFinTareaProy.append(Inicio + datetime.timedelta(weeks =␣
 ↪FinTareaProyecto[a][2]))
```

Horas de cada trabajador por semanas

[51]:
```
arcoTraSem0 =[(i+1,j+1) for i in workers for j in weeks]
HorasTraSem ={(i, j): 0 for i,j in arcoTraSem0}
arcoTraSem = []

for v in model.vars:#if (abs(v.x) > 1e-6) & (v.name == 'Semanas(3,1)'): # only␣
 ↪printing non-zeros
    if (v.name[:3] == 'HTS') & (abs(v.x) > 1e-6):
        indice1 = v.name.index('(')
        indiceComa1 = v.name.index(',')
        indice2 = v.name.index(')')
        arcoTraSem.append((int(v.name[indice1+1:indiceComa1]),int(v.
 ↪name[indiceComa1+1:indice2])))
        HorasTraSem[(int(v.name[indice1+1:indiceComa1]),int(v.name[indiceComa1+1:
 ↪indice2]))] = np.round(v.x)
```

Desglose semanal de proyectos por tareas y horas invertidas por cada trabajador

[52]:
```
arcoHWTS =[]
valor = []
#HWTS: Proy,Tarea,Trabajador,Semana
dictHWTS={}

for v in model.vars:
    #x viene dada como tarea-trabajador-proyecto
    if (v.name[:4] == 'HATS') & (abs(v.x) > 1e-6):
        indiceComa1 = v.name.index(',')
        indiceComa2 = v.name.index(',',indiceComa1+1)
        indiceComa3 = v.name.index(',',indiceComa2+1)
        indice4 = v.name.index(')',indiceComa3+1)
        arcoHWTS.append((int(v.name[indiceComa3+1:indice4]),int(v.name[5:
 ↪indiceComa1]),int(v.name[indiceComa1+1:indiceComa2]),int(v.name[indiceComa2+1:
 ↪indiceComa3])))
        valor.append(np.round(v.x))
        dictHWTS[(int(v.name[indiceComa3+1:indice4]),int(v.name[5:
 ↪indiceComa1]),int(v.name[indiceComa1+1:indiceComa2]),int(v.name[indiceComa2+1:
 ↪indiceComa3]))]=np.round(v.x)
```

3.2.1 COPIA ORDENADA

[53]:
```
arcoHWTS.sort(key = lambda x: x[0])
#arcoHWTS
arcoHWTS[0]
```

[53]:
```
(1, 1, 1, 2)
```

[54]:
```
copiadictHWTS={}

for i in range(len(arcoHWTS)):
    copiadictHWTS[arcoHWTS[i]]=dictHWTS[arcoHWTS[i]]


dictHWTS = copiadictHWTS
```

3.3 Se escriben las soluciones en un excel

[55]:
```
import openpyxl
wb = openpyxl.Workbook()
```

[56]:
```
hoja1 = wb.create_sheet("Trabajadores",2)
hoja2 = wb.create_sheet("ProyPrint",3)
hoja3 = wb.create_sheet("Desglose",4)
```

[57]:
```
StringTitulo = proyectos.iloc[p][1]
```

[58]:
```
hoja1.append(["id_trabajador","Semana","Horas","Horas Lab"])
for i in range(len(arcoTraSem0)):
    hoja1.
 ↪append([arcoTraSem0[i][0],arcoTraSem0[i][1],HorasTraSem[arcoTraSem0[i]],42])
```

[59]:
```
from datetime import datetime
from datetime import timedelta

hoja2.append(["id_proy","Titulo","id_tarea","Nombre","FechaComienzo","FechaFin"])
for i in range(len(FinTareaProyecto)):
    p = FinTareaProyecto[i][0]-1
    a = FinTareaProyecto[i][1]-1
    StringTitulo = proyectos.iloc[p][1]
    StringTarea = tareas.iloc[a][1]
    hoja2.append([FinTareaProyecto[i][0], StringTitulo, FinTareaProyecto[i][1],␣
 ↪StringTarea, FechaComienzoTareaProy[i],FechaFinTareaProy[i]])
```

[60]:
```
hoja3.append(["Semnana","id_proy","id_tarea","id_trabajador","Horas"])
for i in range(len(dictHWTS)):
    hoja3.
 ↪append([arcoHWTS[i][0],arcoHWTS[i][1],arcoHWTS[i][2],arcoHWTS[i][3],dictHWTS[arcoHWTS[i]]])
```

[61]:
```
#wb.save('Paralelo3BI175B.xlsx')
```

B.2. Código en Python

En este segundo apartado se incorpora el código utilizado para para el establecimiento de las distintas cotas temporales.Como se ha dicho anteriormente, primero se resuelven tantos problemas como número de proyectos pertenecientes al problema general. Para cada uno de ellos se obtienen una duración y una fecha de finalización, las cuales se utilizan para el establecimiento de las ventanas temporales del problema general.

```
DuracionMaxima = []
FechaMaxima = []
FechaMinima = []
i=0
for i in range(len(proyectos)):
    Coinciden = 0
    j=0

    for j in range(len(proyectos)):
        if((j!=i) & (Clase[j] == Clase [i])):
            if((FechaInicioSemanas[i] + DuracionMinima[i] < FechaInicioSemanas[
    j]) | (FechaInicioSemanas[i] > FechaInicioSemanas[j] + DuracionMinima[j])):
                Suma = 0
            else:
                if(FechaInicioSemanas[i] < FechaInicioSemanas[j]):
                    if(FechaInicioSemanas[i] + DuracionMinima[i] <
    FechaInicioSemanas[j] + DuracionMinima[j]):
                        Suma = FechaInicioSemanas[i] + DuracionMinima[i] -
    FechaInicioSemanas[j]
                    if(FechaInicioSemanas[i] + DuracionMinima[i] >=
    FechaInicioSemanas[j] + DuracionMinima[j]):
                        if(Clase[i] == 'AI'):
                            Suma = HorasProyTareaRol[j][2][1]/26.6
                        else:
                            Suma = HorasProyTareaRol[j][1][2]/66.67
                    #Coinciden = Coinciden + Suma
                if(FechaInicioSemanas[i] >= FechaInicioSemanas[j]):
                    if(FechaInicioSemanas[i] + DuracionMinima[i] >
    FechaInicioSemanas[j] + DuracionMinima[j]):
                        Suma = FechaInicioSemanas[j] + DuracionMinima[j] -
    FechaInicioSemanas[i]
                    if(FechaInicioSemanas[i] + DuracionMinima[i] <=
    FechaInicioSemanas[j] + DuracionMinima[j]):
                        if(Clase[i] == 'AI'):
                            Suma = HorasProyTareaRol[i][2][1]/26.6
                        else:
                            Suma = HorasProyTareaRol[i][1][2]/66.67
        else:
            Suma = 0
        Coinciden = Coinciden + Suma

    Coinciden = Coinciden/5*3
    Minima = Coinciden/3


    if(Clase[i] == 'AI'):
        Minima = 0
    else:
        Minima = Coinciden/3


    coinciden = int(Coinciden +0.5)
    minima = int(Minima  +0.5)
    DuracionMaxima.append(DuracionMinima[i] + int(Coinciden+0.5)) #coinciden)
```

```
48      FechaMaxima.append(FechaMinimaSemanas [i] + DuracionMinima[i] + coinciden)
    #int(Coinciden+0.5))
49      FechaMinima.append(FechaMinimaSemanas [i] + minima) #int(Coinciden+0.5))
```

Bibliografía

[1] BERNARDO F. ALMEIDA, ISABEL CORREIA AND FRANCISCO SALDANHA-DA-GAMA, *Modeling frameworks for the multi-skill resource-constrained project scheduling problem: a theoretical and empirical comparison*, International Transactions in Operational Research, págs 946

[2] P. BAPTISTE, C. LEPAPE, *Constraint propagation and decomposition techniques for highly disjunctive and highly cumulative project scheduling problems.* Constraints, 2000, pp. 119

[3] *7 RAZONES PARA PROGRAMAR EN PYTHON*, `https://www.bejob.com/7-razones-para-programar-en-python/`, 04/08/2021.

[4] J. BRUNA Y J. CUFÍ, *Anàlisi complexa*, Colección Manuals, Universitat Autònoma de Barcelona, Bellaterra, 2008.

[5] J. BLAZEEWICZ, J. K. LENSTRA, A. H. G. RINNOOY KAN, *Scheduling subject to resource constraints: classification and complexity*, Discret Appl Math 5:11

[6] J. BLAZEEWICZ, W. DOMSCHKE, E. PESCH, *The job shop scheduling problem: Conventional and new solution techniques*, European Journal of Operational Research, págs 1-33, 1996.

[7] P. BRUCKER, S. KNUST, *A linear programming and constraint propagation-based lower bound for the RCPSP*, European Journal of Operational Research, 2000, pp. 355

[8] P. BRUCKER, S. KNUST, *Complex Scheduling*, Springer, GOR-Publications 2012, pp. 355

[9] J. CARLIER, E. NÉRON, *On linear lower bounds for resource constrained project scheduling problem*, European Journal of Operational Research, 2003, pp. 314

[10] N. CHRISTOFIDES, R.ALVAREZ-VALDES, J.M. TAMARIT, *Project scheduling with resource constraints: A branch and bound approach*, European Journal of Operational Research, 1987, pp. 262-273.

[11] I. CORREIA, L.L. LOURENÇO, F. SALDANHA-DA-GAMA, *Project scheduling with flexible resources: formulation and inequalities*, OR Spectrum, 2012, pp. 635

[12] W.CONWAY, L.MAXWELL Y W.MILLER, *THEORY OF SCHEDULING*, ADDISON-WESLEY PUBLISHING COMPANY, Cornell University, 1967.

[13] L.G. FENDLEY, *Towards the Development of a Complete Multi Project Scheduling System*, Journal of Industrial Engineering, 1968, pp. 505

[14] SÖNKE HARTMANN, DIRK BRISKORN, *A Survey of Deterministic Modeling Approaches for Modeling Scheduling under Resource Constraints*, HSBA HAMBURG SCHOOL OF BUSINESS ADMINISTRATION. University of Applied Sciences, 2008.

[15] SÖNKE HARTMANN, DIRK BRISKORN, *A survey of variants and extensions of the resource-constrained project scheduling problem*, European Journal of Operational Research,

[16] ICIOUMH, 10/05/2021, *Seminario Francisco Saldanha-da-Gama: Project Scheduling with Flexible Resources*, Universidad Miguel Hernandez, Instituto Çentro de Investigación Operativa"(CIO), https://www.youtube.com/watch?v=odIKNihqmbc.

[17] R. KOLISCH, *Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation*, European Journal of Operational Research, 1996, pp. 320

[18] R. KOLISCH, A. SPRECHER, *PSPLIB: A project scheduling library*, European Journal of Operational Research, 1997, pp. 205

[19] O. KONÉ, C. ARTIGUES, P. LOPEZ, M. MONGEAU, *Event-based MILP models for resource-constrained project scheduling problems*, Computers and Operations Research, Elsevier, 2011, pp.3

[20] KURT R. LINBERG, *Software developer perceptions about software project failure: a case study*, The Journal of Systems and Software, págs 177-192, 1999.

[21] A. LOVA, P. TORMOS, F. BARBER, *Multi-Mode Resource Constrained Project Scheduling: Scheduling Schemes, Priority Rules and Mode Selection Rules*, Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial. Universidad Politécnica de Valencia, 2006, pp. 69Computers & Industrial Engineering, págs 1333

[22] J. M. LÓPEZ, *Los lenguajes de programación más populares, según los expertos*, 10 Abril 2021. https://hipertextual.com/2021/04/lenguajes-programacion-populares-tiobe

[23] J. C. LOPEZ, J. A. GIRALDO Y J. A. ARANGO, *Reducción del Tiempo de Terminación en la Programación de la Producción de una Línea de Flujo Híbrida Flexible (HFS)*, Información Tecnológica Vol26, 2015, pp. 157

[24] V. MANIEZZO A. MINGOZZI, *A HEURISTIC PROCEDURE FOR THE MULTI-MODE PROJECT SCHEDULING PROBLEM BASED ON BENDERS' DECOMPOSITION*, University of Bologne, 1999.

[25] *The Python-MIP package*, https://www.python-mip.com/, O4/08/2021.

[26] HAROLDO G. SANTOS, TÚLIO A.M. TOFFOLO, *Mixed Integer Linear Programming with Python*, Computational Infrastructure for Operational Research, 2020.

[27] CARLOS MONTOYA ODILE BELLENGUEZ-MORINEAU ERIC PINSON DAVID RIVREAU, *Branch-and-price approach for the multi-skill project scheduling problem*, Optim Lett, págs 1721

[28] THE 2020 STATE OF THE OCTOVERSE, https://octoverse.github.com/, 04/08/2021.

[29] L.D. OTERO, G. CENTENO, ALEX J. RUIZ-TORRES, C.E. OTERO, *A systematic approach for resource allocation in software projects*, Computers & Industrial Engineering, págs 1333

[30] A. ALAN B. PRITSKER, LAWRENCE J. WAITERS AND PHILIP M. WOLFE, *Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach* Manage Sci 16:93

[31] CHRISTOPH SCHWINDT JÜRGEN ZIMMERMANN, *Handbook on Project Management and Scheduling Vol.1* International Handbooks on Information Systems, 2015.

[32] CHRISTOPH SCHWINDT JÜRGEN ZIMMERMANN, *Handbook on Project Management and Scheduling Vol.2* International Handbooks on Information Systems, 2015.

[33] J. SNAUWAERT Y M. VANHOUCKE, *A new algorithm for resource-constrained project scheduling with breadth and depth of skills* European Journal of Operational Research, págs 143

[34] J.P. STINSON, E.W. DAVIS Y B.M. KHUMAWALA, *Multiple Resource Constrained Scheduling Using Branch and Bound* A I I E Transactions, 1978, pp. 252

[35] D. SISTEJKOVIC,*EVOLUTION OF SCHEDULING HEURISTICS FOR THE RESOURCE CONSTRAINED SCHEDULING PROBLEM* Universuty of Zagreb, Faculty of electrical engineering and computing, 2016.

[36] F.A. VILLAFÁÑEZ, D. POZA, A. LÓPEZ-PAREDES, J. PAJARES, *Una nomenclatura unificada para problemas de programación de proyectos (RCPSP and RCMPSP)*, INSISOC

[37] J. WEGLARZ, J. JÓZEFOWSKA, M. MIKA, G. WALIGÓRA, *Project scheduling with finite or infinite number of activity processing modes: A survey*, European Journal of Operational Research, págs 177

[38] LENGUAJES DE PROGRAMACIÓN MÁS USADOS SEGÚN EL TIPO DE DESARROLLO, `https://www.yeeply.com/blog/lenguajes-de-programacion-mas-usados/`, 04/08/2021.