

Proyecto Fin de Carrera
Ingeniería Industrial

PROGRAMACIÓN DE PILOTOS DE UNA AEROLÍNEA

Autor: José Gabriel Rodríguez Gutiérrez

Tutor: José Manuel García Sánchez

Dpto. de Organización Industrial y Gestión de Empresas I
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2018



Proyecto Fin de Carrera

Ingeniería Industrial (Plan 98)

PROGRAMACIÓN DE PILOTOS DE UNA AEROLÍNEA

José Gabriel Rodríguez Gutiérrez

Tutor

José Manuel García Sánchez

Profesor titular

Departamento de Organización Industrial y Gestión de Empresas I

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2018

Proyecto Fin de Carrera: PROGRAMACIÓN DE PILOTOS DE UNA AEROLÍNEA

Autor: José Gabriel Rodríguez Gutiérrez

Tutor: José Manuel García Sánchez

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal

A mis padres, por todo su apoyo durante tantos años.

A mi tutor, por el interés en el proyecto, su paciencia y comprensión.

ÍNDICE

PLANTEAMIENTO DEL PROBLEMA	13
PLANTEAMIENTO	13
DEFINICIÓN DE CONCEPTOS	14
PROCESO DE PROGRAMACIÓN DE TRIPULACIONES	16
PERIODOS DE DESCANSO	20
RESTRICCIONES DEL PROBLEMA	21
PROGRAMACIÓN DE TRABAJOS EN INTERVALOS	24
INTRODUCCIÓN	24
Características de los trabajos o actividades	26
Características de los recursos o maquinas	27
Restricciones de procesamiento	28
CLASIFICACIÓN DE LOS TIPOS DE PROBLEMAS	29
ESCENARIOS	32
PROBLEMA FSP + TÁCTICO	34
PROBLEMA FSP TÁCTICO CON UNA CLASE	34
PROBLEMA FSP TÁCTICO CON VARIAS CLASES	35
REPRESENTACIÓN MATEMÁTICA DEL PROBLEMA	37
LINGO	43
INTRODUCCIÓN A LINGO	43
AMPLIACIÓN DE LINGO	47

INTERFAZ DE USUARIO.....	51
CÓDIGOS GENERADOS PARA LINGO PARA NUESTRO PROBLEMA:	54
EXPERIMENTACIÓN.....	62
CONCLUSIONES.....	72
BIBLIOGRAFÍA.....	74

CAPÍTULO 1

INTRODUCCIÓN

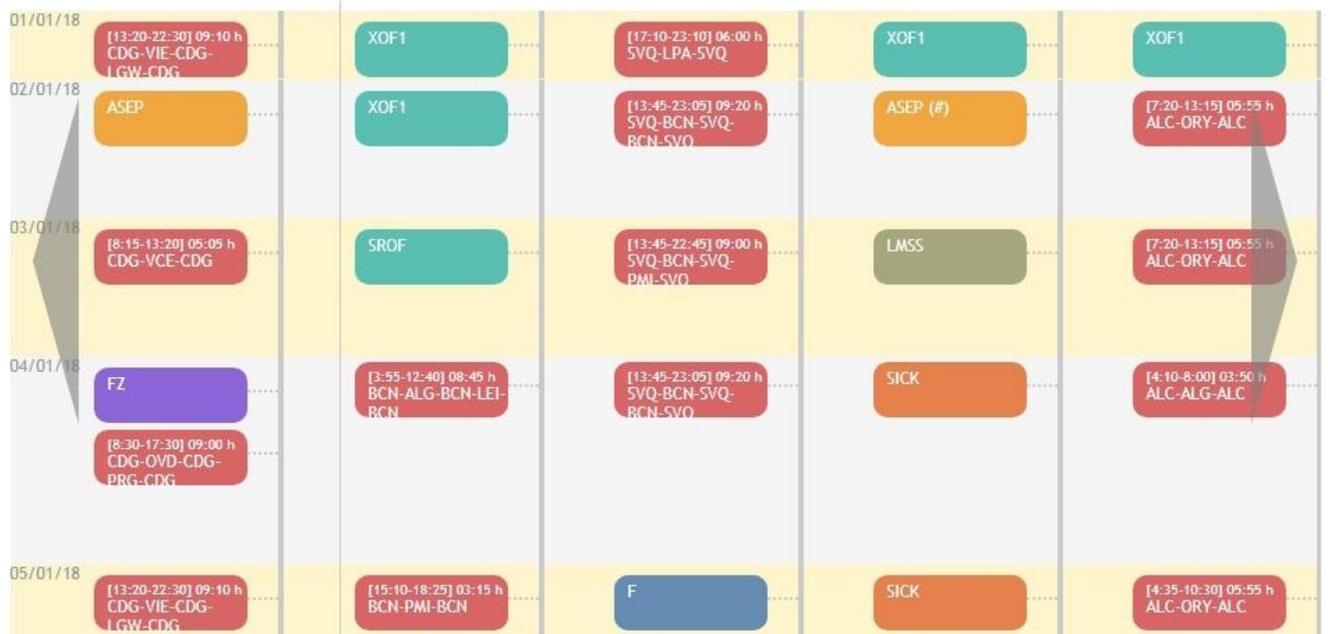
INTRODUCCIÓN

La idea de este proyecto surge de la importancia en los últimos tiempos de la optimización de la programación de tripulaciones para las aerolíneas, dado su gran tamaño de plantilla y sus numerosas restricciones aplicadas a su programación por normativa y convenios colectivos.

Programación compartida

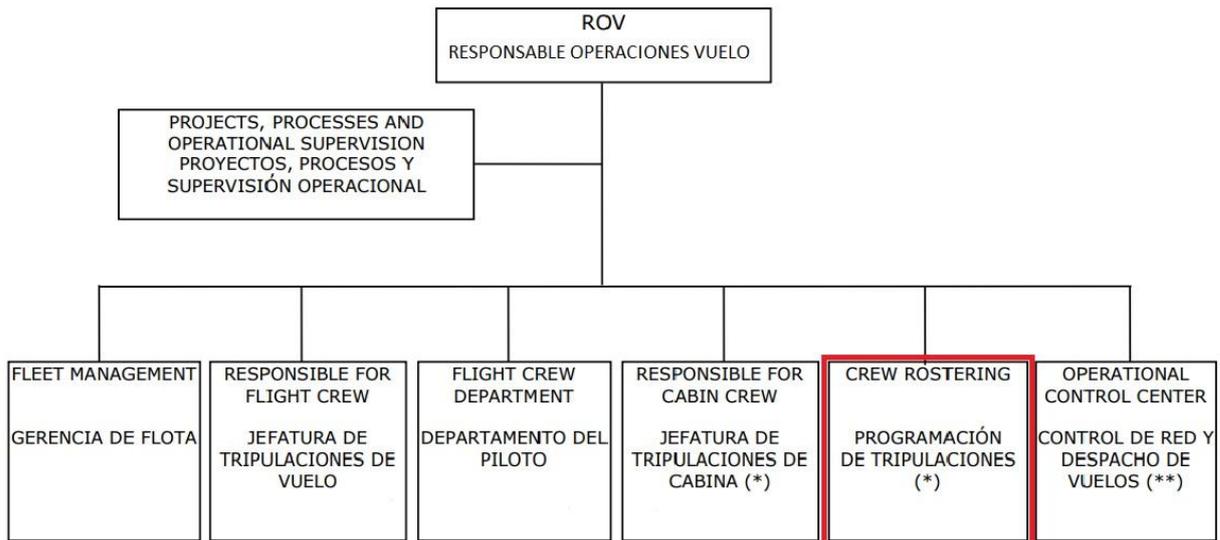
Para solicitar cambios de programación, vaya a la sección Solicitudes > [Pedir cambio de programación](#)

Desde: Hasta:



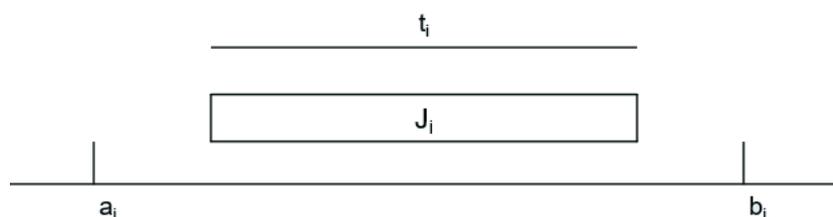
El departamento de programación, perteneciente a su vez al departamento de producción de la compañía, es uno de los más importantes. Interactúa bidireccionalmente y de forma continua con el departamento comercial y el departamento de recursos humanos, adaptando el número de vuelos al tamaño de la plantilla y viceversa, con retroalimentaciones periódicas.

Rev: 06



Nuestro proyecto serviría como ayuda para optimizar esta actividad de la programación de tripulaciones de vuelo.

Para ello, usamos la resolución de un problema de programación de trabajos en intervalos.



Los problemas de programación de trabajos en intervalos tratan la asignación de máquinas a trabajos que deben procesarse dentro de una ventana temporal. Es una variante de los problemas clásicos de scheduling que fuerzan el procesamiento dentro de unos límites temporales. En nuestro caso las máquinas, los recursos, serán los pilotos, y los trabajos los vuelos.

Para la resolución y búsqueda del óptimo del problema hemos empleado librerías de optimización de Lingo.

El documento está estructurado del siguiente modo:

- En primer lugar se realiza el planteamiento del problema mediante una descripción formal del escenario. Esto incluye la definición de los conceptos necesarios para entender el problema en su conjunto y sus restricciones. En este mismo apartado se describe el proceso completo de la programación de los pilotos, los períodos de descanso necesarios y todas las restricciones del problema completo.
- A continuación se desarrolla la teoría de la programación de trabajos en intervalos, junto con los diferentes tipos de problemas y escenarios.
- Nuestro caso será un problema de programación en intervalos fijos, buscando el minimizar el número de recursos necesarios, y teniendo en cuenta varias clases de recursos y trabajos. El desarrollo y explicación de este tipo de problemas lo haremos en un capítulo en concreto. En dicho capítulo se desglosará la representación matemática de nuestro problema.
- Dada la complejidad de una programación de una aerolínea completa, hemos centrado nuestro estudio en la programación de los vuelos de 28 días, dado que es la cifra usada en la normativa para los límites de actividad de vuelo acumulada, en la base concreta de Sevilla, y para la aerolínea Vueling, que posee 3 aviones basados en esta ciudad. Los datos de inicio y fin de cada vuelo para cada avión se sacaron de la intranet de la misma empresa.
- Dedicamos un capítulo a la teoría y uso de Lingo. El código usado para nuestro problema lo desarrollamos en este apartado.
- Finalmente, se realiza la experimentación con Lingo, y se muestran los resultados y conclusiones obtenidas.

CAPÍTULO 2

PLANTEAMIENTO DEL PROBLEMA

PLANTEAMIENTO DEL PROBLEMA

PLANTEAMIENTO

Este trabajo será específico para las tripulaciones técnicas de vuelo.

Consideraremos el trabajo para una aerolínea de gran tamaño y comercialmente llamada de bajo coste, pero que dentro del sector se suelen llamar de nueva generación.

Este tipo de aerolíneas tienen un conjunto de características diferenciadoras respecto de las aerolíneas tradicionales. Algunas de ellas, en cuanto a programación se refiere, son las siguientes:

- Los aviones se distribuyen geográficamente en varias bases.
- Los vuelos inicialmente siempre se programan con comienzo en una de las bases y finalización en la misma. Los tripulantes por tanto también comenzarán y acabarán los vuelos en dicha base al final del día.

Aparte de estas características, enumeramos otras propias de una aerolínea en sí, en cuanto a programación se refiere:

- Las tareas de mantenimiento se realizarán a lo largo de la parada nocturna del avión en cada base. Existen revisiones diarias, de una duración aproximada de una hora, y revisiones semanales, de algo más de una hora de duración. Aun así dicha parada tendrá duración suficiente para éstas revisiones, además de un margen holgado añadido, por si se encuentran tareas extras como cambios de ruedas, frenos, u otros componentes con tiempos de actuación más amplios que la propia revisión.

El resto de revisiones más profundas se programaran aparte, dejando al avión en estado no operativo mientras se realizan los trabajos de mantenimiento asociados.

DEFINICIÓN DE CONCEPTOS

Para desarrollar las restricciones propias del problema, pasamos previamente a la definición de algunos conceptos:

- **Actividad:** Tarea que desempeña una tripulación para el operador, incluido la actividad de vuelo, trabajo administrativo, dar o recibir entrenamiento y verificación.
- **Periodo de actividad:** Periodo que comienza cuando el tripulante se presenta siguiendo las instrucciones del operador para desempeñar sus funciones y concluye cuando el tripulante queda relevado de ellas.
- **Periodo de actividad de vuelo:** Periodo que comienza cuando el tripulante es requerido para el servicio, incluyendo un sector o una serie de sectores, y finaliza cuando el avión se detiene completamente y se apagan los motores, al final del último sector en el que el tripulante actúa como miembro de la tripulación operativa.
- **Tiempo de vuelo:** Tiempo transcurrido desde que una aeronave comienza a moverse desde el sitio de estacionamiento con el fin de despegar hasta que se detiene en el puesto de estacionamiento y se apagan todos los motores.
- **Día local:** Período de 24 horas que se inicia a las 0.00 (hora local).
- **Noche local:** Período de ocho horas comprendido entre las 22.00 y las 8.00 horas (hora local).
- **Actividad nocturna:** Período de actividad que invade una parte del período desde 2.00 y a 4.59 de la zona horaria en que está aclimatada la tripulación.
- **Período de descanso:** Período ininterrumpido y definido de tiempo (antes de una actividad, o tras actividad) durante el cual el miembro de la tripulación queda relevado de todo servicio.
- **Sector:** Segmento de un período de actividad de vuelo transcurrido entre que una aeronave comienza a moverse con el fin de despegar hasta que esta se detiene después de

aterrizar en el puesto de estacionamiento designado.

- Día libre suelto: A efectos de cumplimiento de las disposiciones de CE 2000/79, un período libre de toda actividad y de imaginaria, constituido por un día y dos noches locales, comunicado con antelación. Un periodo de descanso puede ser incluido como parte de un día libre suelto.
- Actividad partida: Período de actividad de vuelo que consiste en dos sectores, separados por un descanso parcial en tierra.
- Imaginaria: Período de tiempo definido y notificado previamente durante el cual el tripulante debe estar a disposición para que se le asigne un vuelo, posicionamiento u otra actividad, sin que medie un período de descanso.
- Semana: Período de 7 días consecutivos, empezando a las 00:01 del lunes.
- Fase de ritmo circadiano de mínimo rendimiento: Período comprendido entre las 2.00 y las 5.59 horas de la zona horaria en la que está aclimatado el miembro de la tripulación.

PROCESO DE PROGRAMACIÓN DE TRIPULACIONES

Previo al comienzo de la programación se habrán completado los siguientes procesos de otros departamentos:

- 55 días antes del primer día del mes a publicar el departamento comercial debe congelar y entregar el programa comercial.
- 43 días antes del primer día del mes a publicar:
 - El departamento de entrenamiento de tripulaciones habrá insertado en el sistema las actividades de tierra, los cursos y las horas de simulador;
 - El departamento de operaciones vuelo habrá gestionado los escalafones correspondientes, cambios de base, ascensos a comandante, y renovaciones de contrato.
- El departamento de administración de Operaciones Vuelo, introduce las limitaciones, habilitaciones y autorizaciones de cada tripulante en el sistema. El sistema permite un control detallado de las limitaciones, proporcionando avisos y alarmas cuando corresponda.
- Una vez finalizados los procesos anteriores, el departamento de programación trabaja en 2 fases utilizando el sistema:
 - Generación de pairings, en la que se divide el programa comercial en rutas operables por las tripulaciones.
 - Rostering, en la que se asignan los pairings a tripulantes concretos.
- La generación de pairings con el sistema termina 43 días antes del primer día del mes que se publica, con la introducción definitiva de pairings en el sistema informático.
- En paralelo, los tripulantes introducen sus preferencias de vuelo y solicitudes de días libres a través de la plataforma web, según lo establecido en los convenios correspondientes.

- Una vez los pairings están en el sistema, junto con las preasignaciones de entrenamiento de tripulaciones, y las solicitudes de los tripulantes, el departamento de programación se responsabiliza de asignar las actividades, adaptándolas a las preferencias de turno solicitadas por los tripulantes.

ACTIVIDAD DE VUELO

La compañía asignará una base a cada tripulante, y esta base estará en el sistema.

La actividad en dicha base siempre comenzará a la hora de firma del tripulante, que consideraremos como estándar 45 minutos antes de la hora de salida del primer vuelo del día.

Pasamos a enumerar las restricciones por normativa que poseen los tripulantes técnicos en cuanto a programación de su actividad, vuelos y descansos, según en Anexo IV de la CE 216/2008, el Anexo III de la CE 965/2012, las especificaciones CS-FTL.1, la Directiva 2000/79/EC, y los reglamentos nacionales, RD1952 / 2009 y CO del 17 de diciembre de 2010.

El período máximo diario de actividad de vuelo para las tripulaciones debe cumplir los valores del siguiente cuadro:

Start of FDP at reference time	Maximum daily FDP – Acclimatised crew members								
	1-2 Sectors	3 Sectors	4 Sectors	5 Sectors	6 Sectors	7 Sectors	8 Sectors	9 Sectors	10 Sectors
0600-1329	13:00	12:30	12:00	11:30	11:00	10:30	10:00	09:30	09:00
1330-1359	12:45	12:15	11:45	11:15	10:45	10:15	09:45	09:15	09:00
1400-1429	12:30	12:00	11:30	11:00	10:30	10:00	09:30	09:00	09:00
1430-1459	12:15	11:45	11:15	10:45	10:15	09:45	09:15	09:00	09:00
1500-1529	12:00	11:30	11:00	10:30	10:00	09:30	09:00	09:00	09:00
1530-1559	11:45	11:15	10:45	10:15	09:45	09:15	09:00	09:00	09:00
1600-1629	11:30	11:00	10:30	10:00	09:30	09:00	09:00	09:00	09:00
1630-1659	11:15	10:45	10:15	09:45	09:15	09:00	09:00	09:00	09:00
1700-0459	11:00	10:30	10:00	09:30	09:00	09:00	09:00	09:00	09:00
0500-0514	12:00	11:30	11:00	10:30	10:00	09:30	09:00	09:00	09:00
0515-0529	12:15	11:45	11:15	10:45	10:15	09:45	09:15	09:00	09:00
0530-0544	12:30	12:00	11:30	11:00	10:30	10:00	09:30	09:00	09:00
0545-0559	12:45	12:15	11:45	11:15	10:45	10:15	09:45	09:15	09:00

TABLA 1

La primera columna se refiere a la hora de firma del tripulante en hora local del lugar de salida, y el resto de columnas a la actividad máxima según el número de sectores o saltos realizados. Una ida y vuelta son dos sectores.

Consideraremos a todos los efectos que los tripulantes se encuentran aclimatados al horario local de su base, ya que no se encontrarán de línea habiendo cruzado varios husos horarios.

Los períodos de actividad totales asignados a tripulante no pueden exceder:

- A. 60 horas de actividad en 7 días consecutivos;
- B. 110 horas de actividad en 14 días consecutivos, y
- C. 190 horas de actividad en 28 días consecutivos, repartidas tan uniformemente como sea posible, y
- D. 1800 horas en 1 año de calendario, aunque esta limitación puede excederse en 200 horas para tripulantes que también desarrollen cualquier otra función de gestión, administración o técnica que no incluya el transporte de pasajeros.

El tiempo total de vuelo (tiempo bloque) que se asigna a cada tripulante como miembro de la tripulación operativa no supera:

- A. 100 horas de tiempo de vuelo en 28 días consecutivos;
- B. 900 horas de tiempo de vuelo en 1 año de calendario;
- C. 945 horas de tiempo de vuelo en cualquier periodo de 12 meses consecutivos para miembros de tripulación de vuelo

La actividad post-vuelo cuenta como período de actividad. El período de tiempo mínimo de actividad post-vuelo es de 20 minutos.

El entrenamiento en vuelo o simulador, siempre que preceda a un vuelo, tiene un coeficiente de multiplicación de 1.5 a los efectos de cómputo de actividad.

PERIODOS DE DESCANSO

Todo período de actividad de vuelo está precedido de un periodo de descanso.

Con el fin de garantizar que el descanso de un tripulante inmediatamente posterior a un período de actividad en vuelo, no comienza hasta estar libre de todas las tareas asignadas por la compañía, y no contabilizan los tiempos de descanso hasta 20 minutos después de la hora de calzos para la tripulación de vuelo.

El período de descanso mínimo concedido antes de iniciar en la base un período de actividad de vuelo es como mínimo igual a la duración de la actividad precedente y, en cualquier caso, no inferior a 12 horas.

Para compensar la fatiga acumulada, se define un período de descanso de recuperación prolongado recurrente. El período de descanso de recuperación prolongado recurrente mínimo es de 36 horas, incluidas dos noches locales consecutivas, de forma que el tiempo comprendido entre el final de un período de descanso de recuperación prolongado y el comienzo del siguiente no sea nunca superior a 168 horas. El período de descanso de recuperación prolongado recurrente debe aumentarse a dos días locales dos veces al mes.

Habrán 12 días libres mensuales programados.

No se programará más de 6 días de actividad seguidos.

RESTRICCIONES DEL PROBLEMA

- Dados los tiempos comerciales de comienzo y finalización de los vuelos, el tripulante comenzará su actividad a su hora de firma, 45 minutos antes de la hora comercial de salida.
- Su actividad programada finalizará a la hora comercial de llegada del último vuelo del día.
- Dicha actividad programada no deberá exceder el máximo de la tabla 1(*), restando además una hora de margen por posibles demoras operativas.
- La actividad sólo podrá comenzar si el tripulante se encuentra descansado, para ello no habrá podido tener actividad alguna durante las 12 horas previas, o durante la actividad precedente, si ésta fuese mayor.
- El descanso programado comienza a contar desde 20 minutos después de la hora comercial de llegada del vuelo. Y éste, como hemos dicho, será de una duración mínima de 12 horas o la actividad precedente, lo que sea mayor.
- Iremos acumulando tanto actividad total programada, como horas bloque de vuelo. Las horas bloque de vuelo sólo contarán desde la hora de salida de cada vuelo hasta su hora de llegada, sin acumular las escalas y resto de paradas, al contrario de la actividad, que será de acumulación continua.
- La actividad acumulada no deberá superar:
 - 60 horas de actividad en 7 días consecutivos;
 - 110 horas de actividad en 14 días consecutivos, y
 - 190 horas de actividad en 28 días consecutivos, repartidas tan uniformemente como sea posible, y
 - 1800 horas en 1 año de calendario, aunque esta limitación puede excederse en 200 horas para tripulantes que también desarrollen cualquier otra función de gestión, administración o técnica que no incluya el transporte de pasajeros.

- El tiempo bloque de vuelo acumulado no deberá superar:
 - 100 horas de tiempo de vuelo en 28 días consecutivos;
 - 900 horas de tiempo de vuelo en 1 año de calendario;
 - 945 horas de tiempo de vuelo en cualquier periodo de 12 meses consecutivos para miembros de tripulación de vuelo.
- No deberán pasar más de 168 horas sin incluir un descanso de recuperación prolongado, 36 horas incluyendo dos noches locales (período de ocho horas comprendido entre las 22.00 y las 8.00 horas, hora local).
- Se programaran 12 días libres al mes, con al menos dos períodos de dos días libres consecutivos.
- No se programaran más de 6 días de actividad seguidos.

CAPÍTULO 3

PROGRAMACIÓN DE TRABAJOS EN INTERVALOS

PROGRAMACIÓN DE TRABAJOS EN INTERVALOS

INTRODUCCIÓN

En este capítulo explicaremos los conceptos y terminología asociados a la programación de trabajos en intervalos con el objetivo de situar el problema del proyecto.

En numerosos trabajos de planificación y programación de tareas, éstas poseen restricciones de tiempo motivadas por aspectos físicos de los materiales involucrados en la fabricación, por normas de proceso, por horarios de trabajo, o por otras imposiciones en los intervalos de procesamiento. En estos casos, la capacidad del sistema, es decir, la cantidad de recursos disponibles, juega un papel fundamental. En este tipo de sistemas, la planificación de la capacidad trata tanto de la asignación de los recursos a las tareas atendiendo a un criterio de optimización (planificación operacional) como al cálculo de la capacidad necesaria para completar todas las tareas (planificación táctica).

La programación de trabajos en intervalos consiste en la asignación en el tiempo de los recursos disponibles con el objetivo de optimizar una determinada medida de comportamiento. A partir de un determinado criterio se trata de establecer la selección de una serie de trabajos para su procesamiento sobre un conjunto de máquinas

Los elementos que componen este tipo de problemas son:

- Un conjunto de trabajos o actividades J con características definidas

$$J = \{J_1 \dots J_j \dots J_n\}$$

- Un conjunto de recursos o máquinas M capaces de realizar dichos trabajos

$$M = \{M_1 \dots M_j \dots M_m\}$$

- Restricciones de compatibilidad y temporales

- Matriz de compatibilidad

$$L_{D \times C} = L(d, c)$$

{1 si el trabajo de clase d puede ser procesado por la máquina c; 0 caso contrario}

- Instante de inicio del intervalo (a_i)
- Instante de fin del intervalo (b_i)
- Tiempo de proceso (t_i)

Todos los problemas tendrán en común el objetivo de maximizar un determinado comportamiento asignando en cada instante de tiempo nuestros recursos disponibles a los trabajos de la forma más conveniente.

Siempre cumpliendo la restricción de que los trabajos no pueden finalizar después de su fecha de entrega.

Características de los trabajos o actividades

Los problemas de programación en intervalos son aplicados a numerosos ámbitos y sectores de la industria buscando siempre el mayor beneficio para la empresa. No obstante todos los trabajos a los que aplicamos esta metodología viene definidos por una serie de parámetros.

Dichos parámetros o requisitos son los siguientes:

- Instante de inicio del intervalo (a_i)
- Instante de fin del intervalo (b_i)
- Tiempo de proceso (t_i)

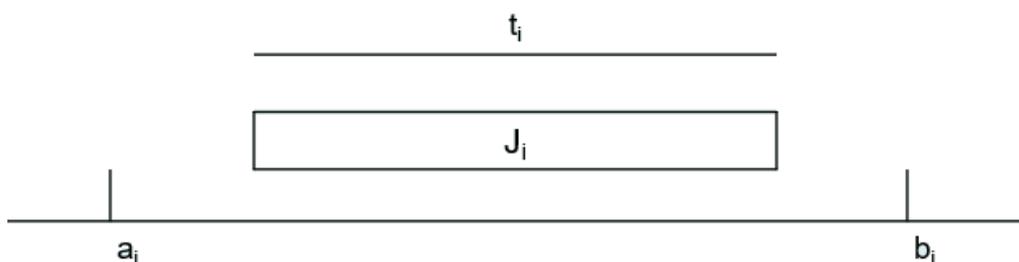
Siendo $b_i \geq a_i + t_i$

Estas características ubican el trabajo en el tiempo, pero los trabajos o actividades también cuentan con una característica que les otorga prioridad en caso de solapamiento:

- Peso del trabajo (w_i)

Esta variable es traducida en el coste de operación de la maquinaria o en el beneficio obtenido por la empresa al realizar el trabajo en función de si nuestra función objetivo resulte a maximizar beneficios o minimizar costes.

Con las características descritas en este punto podemos representar cada trabajo en un diagrama que nos servirá como herramienta visual de la siguiente manera:



En cada problema, estas variables deberán quedar definidas para cada uno de los n trabajos del conjunto $J = \{J_1 \dots J_j \dots J_n\}$.

Características de los recursos o maquinas

Los recursos de los que disponemos serán reflejados en el problema mediante los siguientes parámetros:

$M = \{M_1 \dots M_j \dots M_m\}$

- Clase de la máquina (c_j)
- Coste de uso de la máquina (u_j)
- Intervalo de disponibilidad [i_j, f_j]

Las máquinas básicamente se diferencian por ser de distinta clase dentro de un grupo de C clases de máquinas. Pero además pueden diferenciarse por tener distinto coste u_j o contar con un intervalo de disponibilidad [i_j, f_j] dentro del cual la máquina puede estar funcionando.

La factibilidad de uso del recurso para realizar una actividad vendrá definida en este caso por el intervalo de disponibilidad y la clase de máquina. Debiendo ser solapados los intervalos de disponibilidad de la máquina y de realización del trabajo en un espacio suficiente como para realizar el trabajo.

La clase de máquina en su caso, junto con la matriz de compatibilidad determinará qué actividades pueden ser realizadas por cada recurso. Esta limitación se traduce en la realidad con limitaciones físicas o por lo contrario intercambiabilidad entre máquinas que puedan realizar varios tipos de actividades.

Finalmente, una vez determinada la posibilidad de realizar el trabajo, la conveniencia de realizar dicho trabajo quedará influenciada por el coste de uso de dicho recurso. La equivalencia la podemos ver como costes de operación de un equipo o salario de un operario en realizar una operación.

En cada problema, estas variables deberán quedar definidas para cada uno de los n recursos del conjunto $J = \{J_1 \dots J_n\}$.

Restricciones de procesamiento

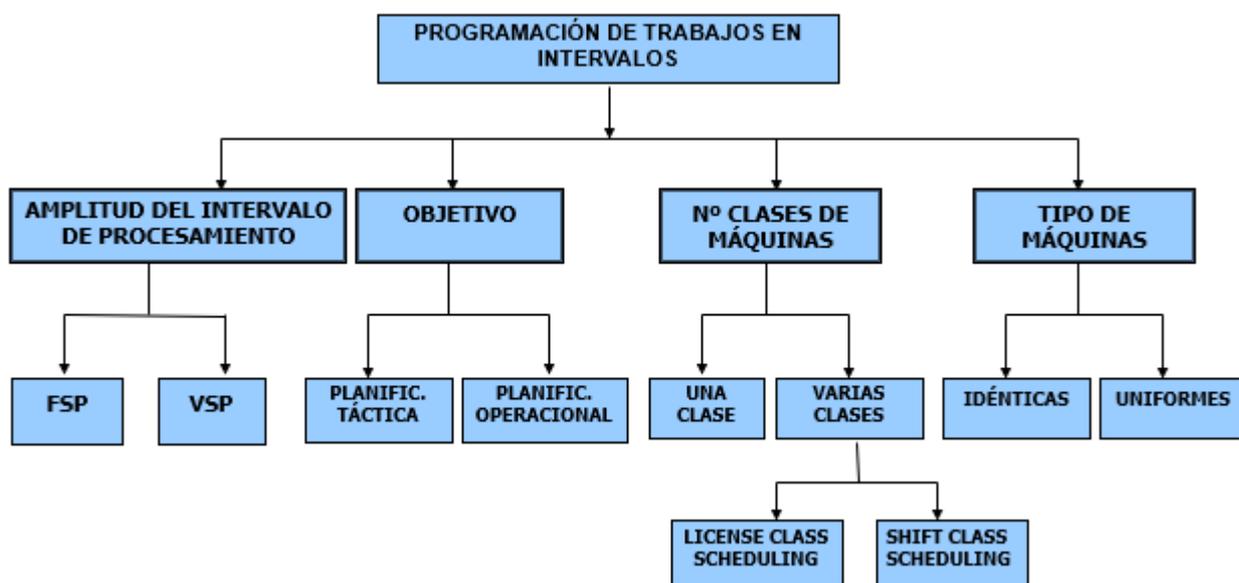
Teniendo definidos los conjuntos de actividades y recursos el siguiente paso lógico es tratar la existente interrelación entre ellos que será lo que realmente nos limite a la hora de realizar la asignación y encontrar la manera más conveniente de resolver el problema.

- Un solo trabajo al mismo tiempo en una máquina, es decir, una máquina no puede realizar dos trabajos simultáneamente. En el problema podrán existir varias máquinas en paralelo de distintas características pero no hay capacidad de realizar varios trabajos en el mismo instante de tiempo por una de las máquinas. Si por las características de nuestra planta el elemento máquina que queremos modelar fuera capaz de realizar varias actividades en el mismo instante de tiempo bastaría con añadir una máquina más de dichas características.
- Cada clase de máquina puede realizar un número limitado de clases de trabajos. Se trata de una restricción de compatibilidad entre los conjuntos actividad-recursos, la cual define la relación existente mediante una matriz de compatibilidad (L). Dicha matriz determina los trabajos que será capaz de realizar una máquina y de igual manera las máquinas que son capaces de realizar un determinado trabajo. Si definimos D clases de trabajo y C clases de máquinas y una matriz binaria $L_{D \times C}$ de forma que $L(d,c)$ vale 1 si la máquina de clase D es compatible con el trabajo de clase C y viceversa.

$$L_{D \times C} = L(d, c) = \begin{cases} 1 & \text{si un trabajo de clase } d \text{ puede ser procesado por una maquina de la clase } c \\ 0 & \text{en otro caso} \end{cases}$$

CLASIFICACIÓN DE LOS TIPOS DE PROBLEMAS

Un problema de programación de trabajos en intervalos se clasifica en función de las siguientes características.



En función de la amplitud del intervalo de procesamiento

- FSP (Fixed job Scheduling Problem)

Escenarios de programación de intervalos donde el tiempo de proceso de las actividades coincide con la amplitud del intervalo. Es decir, que la amplitud del intervalo coincide con el tamaño o la duración de las actividades

- VSP (Variable job Scheduling Problem)

Escenarios de programación de intervalos donde el tiempo de proceso de las tarea es menor o igual que la amplitud del intervalo. En este caso el tiempo de duración de los trabajos es menor que la amplitud del intervalo. De manera que existe cierta holgura a la hora de realizarse la actividad.

Otra característica es el objetivo del problema

- Planificación táctica operacional

No hay limitación de recursos. Se optimiza el uso de los recursos necesarios para atender todas las tareas. El objetivo es definir el número mínimo de recursos necesarios para realizar todas las tareas.

- Planificación operacional de la capacidad

Dado un conjunto de recursos, el objetivo de los problemas de planificación operacional es el de asignar las tareas o actividades con objeto de maximizar el peso total de las tareas completadas con el número de recursos disponibles. Es decir, determinar los trabajos que puedo realizar con los recursos disponibles.

Al realizar esta clasificación, en la planificación operacional partimos de la base de que no disponemos de recursos suficientes para acometer el total de las tareas ya que en ese caso la solución de planificación táctica y operacional nos daría los mismos resultados.

El siguiente criterio de clasificación de nuestro entorno es el número de clases de recursos que vamos a utilizar

- Una clase

Todas máquinas son iguales y por lo tanto todas pueden hacer los mismos tipos de trabajos.

- Varias clases

A la hora de asignar los recursos a los trabajos no todos tienen la misma aplicabilidad por lo que hay que tener en cuenta cómo realizamos la asignación.

En este criterio entra en juego la matriz L de compatibilidad. En esta matriz deberá quedar definida la relación entre recursos y tareas de forma inequívoca, es decir, cada tipo de recurso podrá realizar cierto tipo de trabajos. De esta forma, el entorno en el que todos los trabajos pueden ser realizados por todas las máquinas se podría entender como un caso puntual dentro del global de las combinaciones recursos-máquinas y podría quedar reflejado en una matriz de compatibilidad que determine dicha relación de intercambiabilidad.

Por último, la última clasificación que vamos a hacer para determinar el escenario de este proyecto va a ser dentro del ambiente marcado por recursos de varias clases haciendo la diferenciación entre:

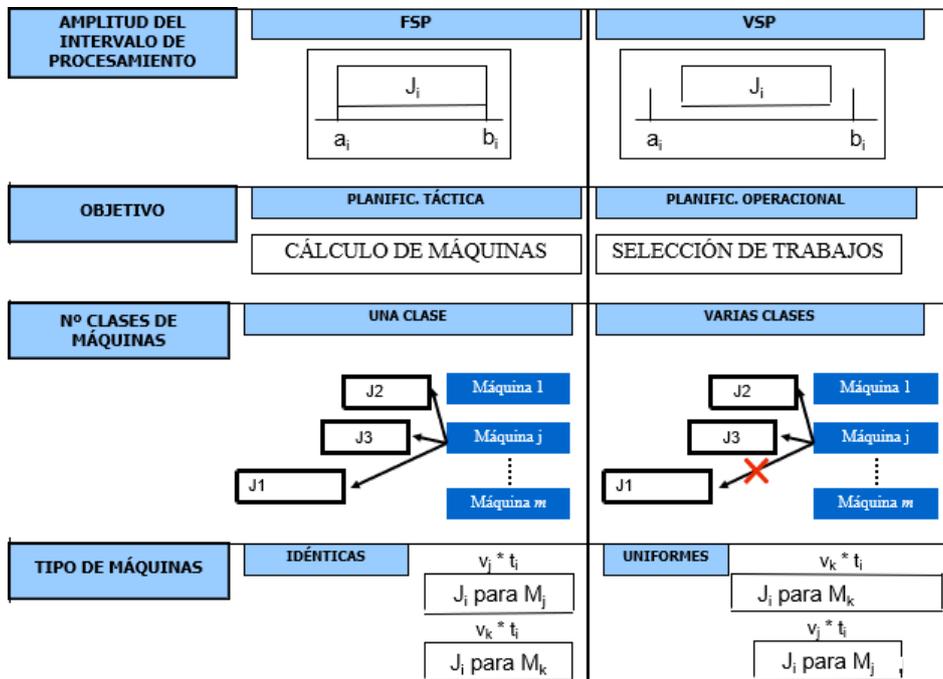
- License Class Scheduling (LCS)

El cual se caracteriza por mostrar disponibilidad de los recursos abierta durante todo el horizonte de trabajo, la compatibilidad está definida únicamente por aspectos técnicos.

- Shift Class Scheduling (SCS)

Donde cada recurso solamente está disponible en un rango determinado de tiempo, la compatibilidad entre máquina y trabajo está afectada por la disponibilidad de la máquina.

Nuevamente esta determinación será fijada por el entorno al que nos enfrentemos y aunque siempre exista la posibilidad de mejorar nuestra capacidad de recursos solemos encontrarnos la limitación impuesta.



ESCENARIOS

Combinando las características mencionadas previamente podemos encontrar los siguientes escenarios de problemas de asignación de intervalos.

FSP + Táctico + Una clase

FSP + Operacional + Una clase

FSP + Táctico + Varias clases

FSP + Operacional + Varias clases

VSP + Táctico + Una clase

VSP + Operacional + Una clase

VSP + Táctico + Varias clases

VSP + Operacional + Varias clases

Este proyecto se centra en el caso FSP + Táctico + Varias clases.

CAPÍTULO 4

PROBLEMA FSP + TÁCTICO

PROBLEMA FSP + TÁCTICO

PROBLEMA FSP TÁCTICO CON UNA CLASE

Para la resolución de problemas en este escenario se pueden emplear el siguiente modelo de resolución. Se trata de un problema clase P.

Algoritmo de carácter Greedy

Procedimiento Asignación_FSP

$m = n$

$J = \{J_i, i = 1..n; s_i \leq s_{i+1}\}$

$MA = \emptyset$ {conjunto de máquinas utilizadas}

$a = |MA| = 0$

$t_j = 0, j = 1..m$ {instante de fin de trabajo de cada máquina}

Para cada $J_i \in J$

Si $t_j > s_i \forall j \in MA$ or $MA = \emptyset$ entonces

$a = a + 1$

$MA = MA + \{M_a\}$

{Asigna J_i a máquina M_a }

$t_a = f_i$

si no

$k = \{j \in MA / t_j \leq s_i\}$

{Asigna J_i a máquina k }

$t_k = f_i$

Fin Si

Fin Para

Fin

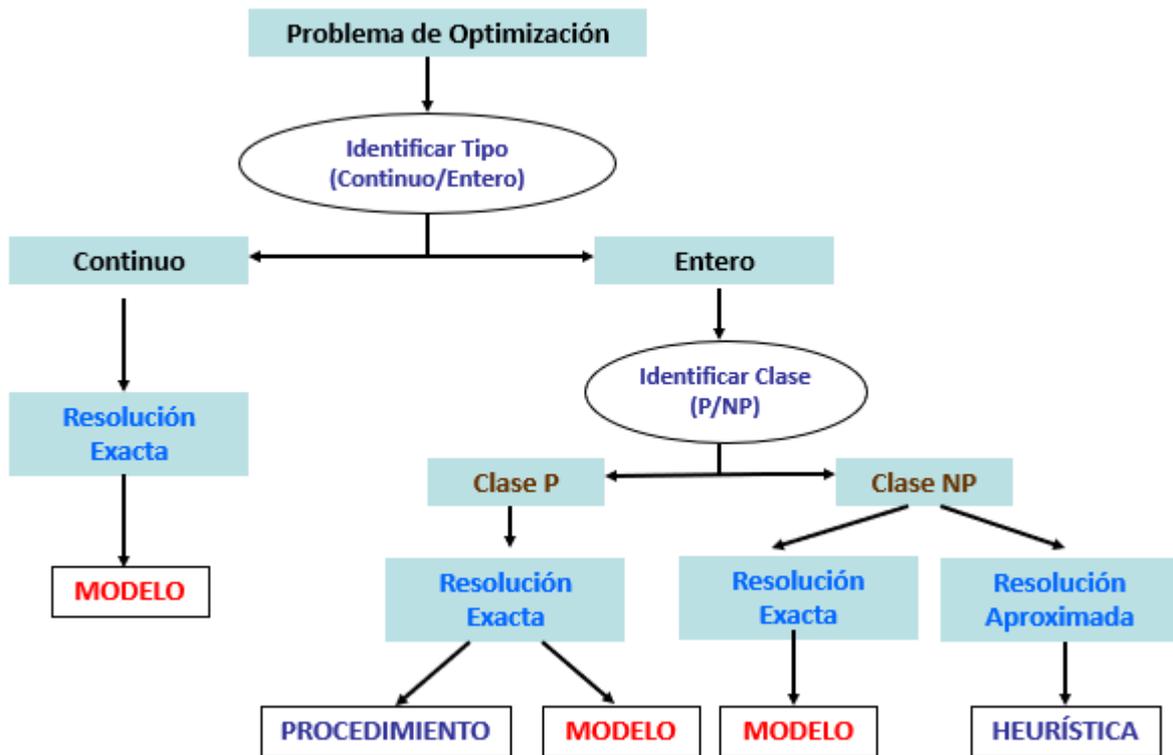
Tiene como objetivo determinar el grado de simultaneidad. Es decir el número mínimo de máquinas necesarias para realizar todos los trabajos.

Siendo s_i el instante de comienzo y f_i al instante final. Donde m es el número de máquinas.

PROBLEMA FSP TÁCTICO CON VARIAS CLASES

Para esta clase de problemas es necesario definir la matriz de compatibilidad entre clases de trabajos y clases de máquinas.

Se trata de problemas de tipo NP (problemas complejos que requieren múltiples iteraciones para el cálculo del valor del óptimo) . Para su resolución en este proyecto se hará uso de librerías de optimización.



Modelo:

Modelo:

$$\text{Min} \sum_{c=1}^C c_c y_c$$

y_c = Número de máquinas utilizadas de la clase c

x_{ic} = Si Trabajo i se procesa con una máquina de clase c

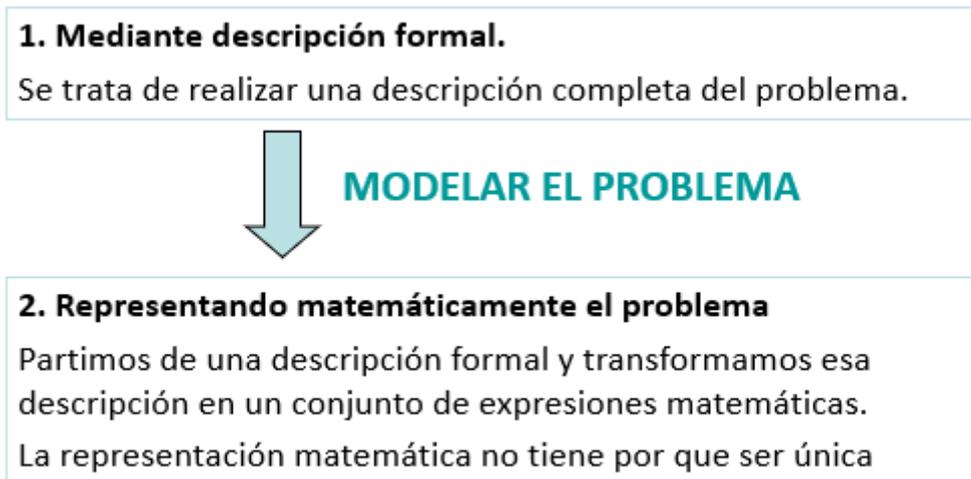
$$\sum_{c/L(d_i,c)=1} x_{ic} = 1 \quad \forall i \quad \text{Todo trabajo debe ser realizado por alguna clase}$$

$$\sum_{\{i/s_i \leq t \leq f_i \wedge L(d_i,c)=1\}} x_{ic} \leq y_c \quad c=1 \dots C; t = s_{\bar{k}} \forall J_{\bar{k}} \in J / L(d_{\bar{k}},c) = 1$$

El número de trabajos procesados por una clase en cualquier instante de comienzo de un trabajo compatible con esa clase, no puede ser superior al nº de máquinas de esa clase

REPRESENTACIÓN MATEMÁTICA DEL PROBLEMA

Una vez planteado el problema mediante una descripción formal del escenario y las variables y restricciones procedemos a la representación matemática del problema.



- Conjunto de variables.

Referencia a la actividad que se desarrolla en el sistema que se quiere optimizar.

En nuestro caso:

Vuelos(j): Conjunto de vuelos a programar a los pilotos. Tendrá 4 argumentos: día, inicio, fin, y tipo de vuelo.

Pilotos(i): Conjunto de pilotos a los que programaremos los vuelos. Incluirá a pilotos que no serán programados, por no ser necesario debido a la optimización.

Días(d): Conjunto de días de programación. Debido a restricciones de actividad máxima por normativa, consideraremos conjuntos de 7, 14 y 28 días.

Patrones(p): Conjunto de distintos patrones posibles de programación. Considerando el patrón estándar de 5 días de trabajo-4 días libres, tendremos 5 posibles patrones de programación a superponer entre tripulantes:

T-T-T-T-L-L-L-L,

L-T-T-T-T-L-L-L,

L-L-T-T-T-T-L-L,

L-L-L-T-T-T-T-L,

L-L-L-L-T-T-T-T.

$X(i,j)$: Variable binaria que nos indicará si un vuelo j lo realiza un piloto i .

$C(i,j)$: Matriz de compatibilidad tipo de vuelo-tipo de piloto. En nuestro caso existirán dos tipos de vuelos y dos tipos de pilotos.

Vuelos Tipo 1: Vuelos que no requieren experiencia.

Vuelos Tipo 2: Vuelos que requieren al menos 1000 horas de vuelo en el tipo de aeronave.

Pilotos Tipo 1: Pilotos con menos de 1000 horas en el tipo de aeronave,

Pilotos Tipo 2: Pilotos con más de 1000 horas en el tipo de aeronave.

$Beta(i,d)$: Variable binaria que nos indica si el piloto i trabaja el día d .

$T(p,d)$: Matriz que relaciona y desarrolla los patrones con los días.

$Mx(i,d)$: Hora de finalización del último vuelo del día d por el piloto i .

$m(i,d)$: Hora de comienzo del primer vuelo del día d por el piloto i .

$delta(i,p)$: Variable que relaciona al piloto con un patrón.

- Conjunto de restricciones

Expresan las relaciones entre el consumo de recursos y las limitaciones de los mismos, así como toda clase normas y características que hay que imponer asociadas a la actividad que se realiza en el sistema.

En nuestro caso:

Cada vuelo lo realizará un único piloto:

$$\forall j : \quad x_{ij} = 1 ; \quad x_{ij} = \text{matriz compatibilidad vuelos pilotos} \\ i / c_{ij}=1$$

Restricción de máximo número de horas de vuelo en 28 días:

$$\forall i : \quad \sum_j [fin(j) - inicio(j)] \cdot x_{ij} \leq Tope 2 \cdot y_i ; \quad Tope 2 = 100 ; \\ \Rightarrow \text{Limite de tiempo bloque de vuelo acumulado en 28 días}$$

Cálculos de hora de comienzo y fin del primer y último vuelo del día, respectivamente:

$$\forall i, \forall d, \forall j / \text{día}(j) = d : m(i, d) \leq inicio(j) \cdot x(i, j) + E \cdot [1 - x(i, j)] ; \\ \forall i, \forall d, \forall j / \text{día}(j) = d : Mx(i, d) \geq fin(j) \cdot x(i, j) + E \cdot [1 - x(i, j)] ;$$

Restricción del límite diario de actividad:

$$\forall i, \forall d : [Mx(i, d) - m(i, d)] + 0.75 \leq Tope 1 ; \quad Tope 1 = 12 : \text{límite diario de actividad}$$

Restricción de no superposición de vuelos:

$$\forall i, \forall j, \forall k / j \neq k, dia(j) = dia(k), inicio(j) \leq inicio(k), fin(j) > inicio(k) \\ : x(i, j) + x(i, k) \leq 1;$$

Restricción de descanso mínimo antes del vuelo del día siguiente:

$$\forall i, \forall j, \forall s / j \neq s, dia(j) = dia(s) - 1, [24 - fin(j) + inicio(s)] \leq (12 + 0.33 + 0.75) \\ : x(i, j) + x(i, s) \leq 1;$$

Restricción de actividad máxima acumulada:

$$\forall i : \sum_d M x(i, j) - m(i, d) + 0.75 \cdot beta(i, d) \leq Tope 3 ;$$

⇒ *Tope 3 : Actividad total acumulada = 6h en 7 días ; 110h en 14 días ; 190h en 28 días*

Restricciones auxiliares de cálculo:

$$\forall i : \sum_j x_{ij} \leq Mtope \cdot y_i ; Mtope = \text{número elevado} ; x_{ij} \\ = 1 \text{ si vuelo } j \text{ lo hace piloto } i ; 0 \text{ si no}$$

$$\forall i, \forall d : \sum_{j/dia(j)=d} x(i, j) \leq Mtope \cdot beta(i, d) ;$$

$$\forall i, \forall d : \sum_{j/dia(j)=d} x(i, j) \geq beta(i, d) ;$$

Restricciones de asignación de un patrón a cada piloto:

$$\forall i : \sum_p delta(i, p) \leq 1 ;$$

$$\forall i, \forall p, \forall d : delta(i, p) \leq 1 - T(p, d) + [2 \cdot T(p, d) \cdot beta(i, d)] - beta(i, d) ;$$

- Función objetivo

Criterio que se desea optimizar.

En nuestro caso será el minimizar el número total de pilotos necesarios:

$$\text{Min } \sum_{i=1}^n y_i ; y_i = 1 \text{ si trabaja piloto } i ; 0 \text{ si no } ; n = n^{\circ} \text{ pilotos}$$

CAPÍTULO 5

LINGO

LINGO

INTRODUCCIÓN A LINGO

Lingo es una herramienta informática utilizada para resolver modelos de optimización tanto lineales como no lineales automáticamente y de forma rápida, ofreciendo una hoja de resultados muy detallada que nos permitirá interpretar estos de forma cómoda.

Para programar en Lingo es necesario utilizar su propio lenguaje, el cual describiremos brevemente a continuación, así como el formato necesario para escribir el modelo.

El formato consta de 5 partes:

- **Encabezado**

Sirve para indicarle a Lingo que vamos a escribir un modelo.

MODEL:

- **Función Objetivo**

Maximizar o minimizar con los valores de los costes de todas las variables.

$MIN = 3 * X1 + 4 * X2 + X3 - 7 * X4 + 2 * X5 + 7;$

Es importante indicar que, a fin de evitar errores, es necesario poner punto y coma (;) al final de cada línea, de modo que indiquemos al programa que en la siguiente línea se va a escribir otra indicación diferente a la anterior.

- **Restricciones del problema**

Al igual que en todos los modelos de optimización, existen una serie de restricciones a cumplir que hay que reflejarlas con el siguiente formato.

$$X1+3*X4 \leq 7;$$

$$X3-7*X2 \geq 1;$$

...

- Restricciones asociadas al tipo de variable

- En Lingo, por defecto, las variables son continuas ($\geq 0 \geq 0$)
- Las variables enteras se escriben como @GIN(X1);
- Las variables binarias se escriben como @BIN(X2);
- Las variables libres se escriben como @FREE(X3);

- Fin

Para indicarle a Lingo que hemos finalizado la escritura del modelo.

END

Hay que tener en cuenta ciertas consideraciones como por ejemplo:

- Lingo ignora las líneas en blanco.
- Si queremos insertar un comentario dentro de la misma programación del modelo debemos preceder éste del símbolo ¡ y al final de un punto y coma (;).
- Lingo no distingue entre mayúsculas y minúsculas.
- La longitud máxima de una línea es de 512 caracteres.

A continuación se muestra un modelo de optimización, el cual será programado en lenguaje Lingo de forma que tengamos un ejemplo de modelo completo para poder ver más claramente las partes de las que constan los modelos en este lenguaje.

$$\text{Max } x_1 + x_2 + y$$

s.a.

$$x_1 \leq 2$$

$$x_1 + y \leq 3$$

$$y + 2x_2 \leq 4$$

$$x_1 \geq 0, x_2 \geq 0 \text{ Entera}$$

y binaria

Si escribimos el modelo anterior en lenguaje Lingo quedaría como lo siguiente:

MODEL:

[OBJETIVO] MAX = x1 + x2 + y ;

[PRIMERA] x1 <=2; !ejemplo de comentario;

[SEGUNDA] x1 + y <=3;

[TERCERA] y + 2*x2 <=4;

[Signo_x1] x1 >=0;

[Signo_x2] x2 >=0;

@GIN(x2);

@BIN(y);

END

Aparece una función objetivo seguida de cinco restricciones, de las cuales dos de ellas sólo sirven para indicar que X1 y X2 son positivos.

También aparecen dos restricciones asociadas al tipo de variable que nos indican que X_2 es una variable entera y que y representa una variable binaria.

En el caso de no haber indicado nada en las restricciones asociadas al tipo de variable, Lingo asumirá que éstas pueden tomar cualquier valor real.

AMPLIACIÓN DE LINGO

USO DE CONJUNTOS

Cuando queremos expresar un modelo de forma simplificada (abstracta), es necesario el uso de conjuntos. Los conjuntos tienen una equivalencia directa con los elementos del problema, con una diferencia. LINGO distingue dos tipos de conjuntos:

- Conjuntos primitivos:

Recogen los elementos conjunto del problema y sus variables asociadas independientes o relacionadas con elementos no conjuntos

- Conjuntos derivados:

Recogen las variables asociadas relacionadas entre elementos conjunto. Están formados, por tanto, por los elementos primitivos ya definidos.

Las variables asociadas sólo se definen, por tanto, una única vez.

Los elementos concepto o unitarios simples no es necesario definirlos. Sus variables asociadas independientes se definen directamente en la sección de valores (DATA).

LINGO posee dos secciones para la definición de los conjuntos.

- Sección **SET**: Se definen los nombres y número de elementos de los conjuntos
- Sección **DATA**: Se definen los valores de las variables asociadas. Es posible también definir aquí el número de elementos del conjunto.

Por ejemplo, en el caso de un grafo, la tabla de elementos estaría compuesta por nodos y arcos. Es esta tabla la que habría que definir en la sección **SETS**.

Si suponemos un grafo con 9 nodos cuyas variables asociadas son ser terminal o no, y 13 arcos cuyas variables asociadas son un coste y un valor de tránsito, su modelado en Lingo sería el siguiente:

SETS :

Nodos/1..9/:Ti;

Arcos (Nodos,Nodos)/1 2,2 4,2 5,4 5,4 7,7 5,8 5,7 8,5 3,3 6,5 6,5 9,6 9/:C,X;

ENDSETS

Podemos ver cómo hemos definido un conjunto de 9 nodos (/1..9/) y posteriormente hemos indicado la característica como Ti.

Lo mismo hemos hecho con los arcos, aunque aquí hemos indicado que estos están formados por dos valores dados por dos nodos (Nodos,Nodos), especificando posteriormente qué nodos forman cada arco. Por ejemplo, para el primer arco (1 2), su origen sería el nodo 1 y el destino sería el nodo 2, y así para todos los arcos.

Finalmente hemos definido sus variables asociadas que son C para el coste de cada arco y X para el valor de tránsito.

Si queremos fijar valores para las variables asociadas usaremos la sección DATA. En el caso de los problemas que se tratan en este trabajo, el enunciado nos facilita cuáles son los nodos terminales, así como el coste de cada arco. Por lo tanto, tendríamos que fijar estos valores para trabajar con ellos.

Podemos considerar la variable Ti como un valor binario que valdrá 1 si el nodo es terminal y 0 si no. El valor de coste de cada arco es un valor entero que nos proporciona el enunciado. Todo esto lo definiríamos en Lingo de la siguiente forma:

DATA :

Ti=0,1,1,1,0,1,1,0,1;

C=2,1,3,2,1,3,2,3,1,1,3,2,3;

ENDDATA

Ahora tenemos 9 valores de T_i . El nodo 1, por ejemplo, no es terminal, ya que es el primer valor de T_i el que corresponde al nodo 1.

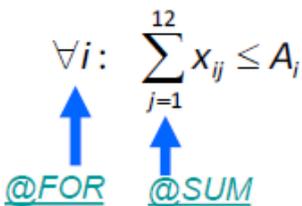
Este mismo orden se sigue para el coste. El coste del primer arco definido (1 2) es de 2 y así sucesivamente.

RESTRICCIONES

A la hora de representar restricciones en Lingo usamos las funciones @FOR y @SUM.

Función	Uso
@FOR	@FOR es principalmente usado par generar restricciones sobre los miembros de un conjunto.
@SUM	@SUM calcula la suma de una expresión sobre los miembros de un conjunto.

Aplicación:

$$\forall i: \sum_{j=1}^{12} x_{ij} \leq A_i$$


El formato utilizado en las restricciones para los signos es el siguiente:

- #EQ# =
- #NE# ≠
- #GE# ≥
- #GT# >
- #LT# <
- #LE# ●

Por poner un ejemplo para aclarar un poco más estos conceptos trataremos un sistema con 100 proveedores a los que compro unidades de un producto.

Hay que imponer que a cada proveedor no se le compren más de 2500 unidades.

SETS:

Proveedores/1..100/:X;

Producto;

ENDSETS

@FOR(Proveedores(i):X(i)<=2500);

Como se puede ver, hemos definido los 100 proveedores con una variable X que indica las unidades que se compran a cada proveedor.

El *@FOR* indica que para cada proveedor no podemos comprar más de 2500 unidades.

Un ejemplo para la función *@SUM* sería:

SETS:

Clientes/1..12/: Demanda;

ENDSETS

DATA:

Demanda = 20, 25, 25, 30, 10, 10, 16, 23, 25, 35, 30, 13;

ENDDATA

DemandaTotal = @SUM(Clientes(j): Demanda(j));

En este último ejemplo se han definido doce clientes cada cual tiene una demanda que se fija en la sección DATA.

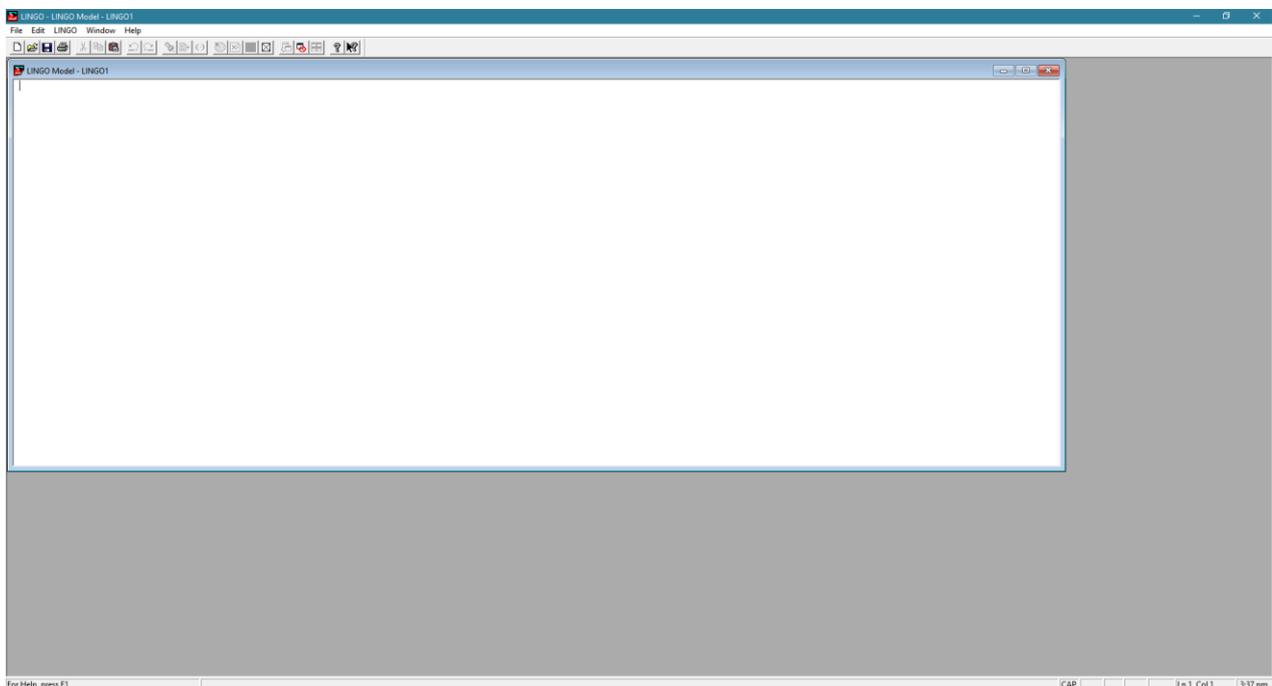
Al final definimos un valor “DemandaTotal” que es la suma de las demandas de todos los clientes,

$$\sum_{j \in \text{Clientes}} D_j$$

lo que se definiría matemáticamente así:

INTERFAZ DE USUARIO

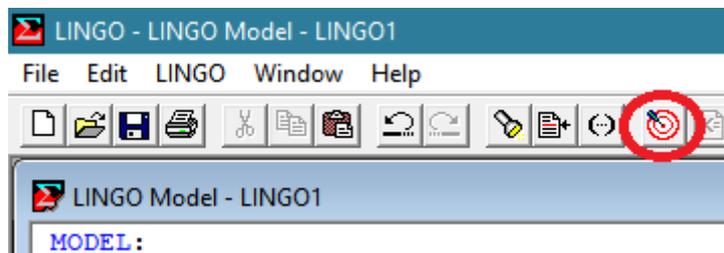
Una vez ejecutamos la aplicación Lingo lo primero que nos debe aparecer es la siguiente ventana:



La misma ventana de la aplicación contiene a otra que nos deja ver una página en blanco. Es ahí donde escribiremos el código del modelo para Lingo lo resuelva.

En la parte superior de la ventana aparecen una serie de herramientas, algunas necesarias para la resolución del modelo y otras que pueden servir de ayuda.

Una vez escribimos el código sólo tenemos que pulsar en el botón “solve” para que Lingo lo resuelva automáticamente:



Es entonces cuando se pueden dar dos situaciones:

- Lingo nos indica que existe algún fallo en nuestro código, ya sea porque no hemos seguido correctamente la ortografía del lenguaje Lingo, porque estamos usando variables no definidas, etc.
- Lingo resuelve el problema y nos abre una nueva ventana donde nos indica el número total de variables, número de restricciones, valor de la función objetivo, etc.

Solver Status Model Class: IIP State: Global Opt Objective: 18 Infeasibility: 0 Iterations: 0		Variables Total: 39 Nonlinear: 0 Integers: 26	
Extended Solver Status Solver Type: B-and-B Best Obj: 18 Obj Bound: 18 Steps: 0 Active: 0		Constraints Total: 36 Nonlinear: 0	
		Nonzeros Total: 90 Nonlinear: 0	
		Generator Memory Used (K) 30	
		Elapsed Runtime (hh:mm:ss) 00 : 00 : 00	
Update Interval: 2		<input type="button" value="Interrupt Solver"/> <input type="button" value="Close"/>	

Finalmente se nos abre otra ventana que contiene todos los datos obtenido en la solución como la siguiente:

```

Global optimal solution found.
Objective value:                18.00000
Extended solver steps:          0
Total solver iterations:        0

Variable      Value      Reduced Cost
-----
CS            5.000000      0.000000
TI( 1)        0.000000      0.000000
TI( 2)        1.000000      0.000000
TI( 3)        1.000000      0.000000
TI( 4)        1.000000      0.000000
TI( 5)        0.000000      0.000000
TI( 6)        1.000000      0.000000
TI( 7)        1.000000      0.000000
TI( 8)        0.000000      0.000000
TI( 9)        1.000000      0.000000
R( 1)         0.000000      0.000000
R( 2)         1.000000      0.000000
R( 3)         0.000000      0.000000
R( 4)         0.000000      0.000000
R( 5)         0.000000      0.000000
R( 6)         0.000000      0.000000
R( 7)         0.000000      0.000000
R( 8)         0.000000      0.000000
R( 9)         0.000000      0.000000
C( 1, 2)      2.000000      0.000000
C( 2, 4)      1.000000      0.000000
C( 2, 5)      3.000000      0.000000
C( 4, 5)      2.000000      0.000000
C( 4, 7)      1.000000      0.000000
C( 7, 5)      3.000000      0.000000
C( 8, 5)      2.000000      0.000000
C( 7, 8)      3.000000      0.000000
C( 5, 3)      1.000000      0.000000
C( 3, 6)      1.000000      0.000000

```

A partir de los datos que obtenemos de esta última imagen podemos interpretar las soluciones, ya que no sólo nos muestra el valor de la función objetivo en el óptimo, sino que también nos ofrece la posibilidad de saber el valor de todas las variables en todas las situaciones posibles.

CÓDIGOS GENERADOS PARA LINGO PARA NUESTRO PROBLEMA:

Declaración de variables y asignación de valores y atributos:

MODEL:

SETS:

VUELOS/1..280/:dia,inicio,fin,tipovuelo;

PILOTOS/1..20/:tipopiloto,Y;

DIAS/1..28/;

PATRONES/1..5/;

PILOTOS_VUELOS(PILOTOS,VUELOS):x;

tiposvuelos/1,2/;

tipospilotos/1,2/;

tiposvuelos_tipospilotos(tipospilotos,tiposvuelos):c;

pilotos_dias(pilotos,dias):Mx,m,beta;

pilotos_patrones(pilotos,patrones):delta;

dias_patrones(patrones,dias):T;

ENDSETS

DATA:

DIA,INICIO,FIN,TIPOVUELO= !Inicios y fines en horas de los vuelos de 28 días de la base de Vueling en Sevilla, con 3 aviones, considerando los vuelos a Bilbao y Tenerife Norte como vuelos Tipo 2, que requieren a un piloto con más de 1000 horas en el tipo de aeronave;

1,4.75,8.75,1, 1,9.5,12.83,2, 1,13.58,17.67,1, 1,18.25,22.25,1, 1,5.00,10.08,2, 1,11.08,14.50,1,
1,15.08,20.58,1, 1,5.75,9.75,1, 1,10.5,15.5,2, 1,16.17,21.17,1, 2,4.67,9.5,1, 2,10.67,14.42,1,
2,16.42,21.33,1, 2,05.75,09.75,1, 2,10.5,15.5,2, 2,16.92,21.58,1, 2,4.75,8.75,1, 2,9.42,12.83,1,

2,13.58,17.67,1, 2,18.25,22.25,1, 3,5,10.5,1, 3,11.67,15.58,1, 3,16.17,20.75,1, 3,5.75,9.75,1,
 3,10.5,15.5,2, 3,16.17,21.17,1, 3,4.75,8.75,1, 3,9.5,12.83,2, 3,13.58,17.67,1, 3,18.25,22.25,1,
 4,4.75,8.75,1, 4,9.5,12.83,2, 4,13.58,17.67,1, 4,18.33,21.58,1, 4,5.75,9.75,1, 4,10.5,15.5,2,
 4,16.17,21.17,1, 4,5,10.08,1, 4,10.67,14.42,1, 4,15.08,20.58,1, 5,5,10.08,2, 5,11.08,14.5,1,
 5,15.08,20.58,1, 5,5.75,9.75,1, 5,10.5,15.5,2, 5,16.17,21.17,1, 5,4.75,8.75,1, 5,9.5,12.83,2,
 5,13.58,17.67,1, 5,18.25,22.25,1, 6,4.67,9.5,1, 6,10.17,14.08,1, 6,15,19.92,1, 6,5.33,9.92,1,
 6,10.5,15.5,2, 6,16.17,21.17,1, 6,4.92,8.5,1, 6,9.08,12.33,2, 6,12.92,16.5,1, 6,17.58,22.08,1,
 7,4.75,8.75,1, 7,9.5,12.83,2, 7,13.58,17.67,1, 7,18.25,22.25,1, 7,5,10.08,2, 7,10.67,14.42,1,
 7,15.08,20.58,1, 7,5.75,9.75,1, 7,10.5,15.5,2, 7,16.17,21.17,1, 8,4.75,8.75,1, 8,9.5,12.83,2,
 8,13.58,17.67,1, 8,18.25,22.25,1, 8,5.00,10.08,2, 8,11.08,14.50,1, 8,15.08,20.58,1, 8,5.75,9.75,1,
 8,10.5,15.5,2, 8,16.17,21.17,1, 9,4.67,9.5,1, 9,10.67,14.42,1, 9,16.42,21.33,1, 9,05.75,09.75,1,
 9,10.5,15.5,2, 9,16.92,21.58,1, 9,4.75,8.75,1, 9,9.42,12.83,1, 9,13.58,17.67,1, 9,18.25,22.25,1,
 10,5,10.5,1, 10,11.67,15.58,1, 10,16.17,20.75,1, 10,5.75,9.75,1, 10,10.5,15.5,2, 10,16.17,21.17,1,
 10,4.75,8.75,1, 10,9.5,12.83,2, 10,13.58,17.67,1, 10,18.25,22.25,1, 11,4.75,8.75,1, 11,9.5,12.83,2,
 11,13.58,17.67,1, 11,18.33,21.58,1, 11,5.75,9.75,1, 11,10.5,15.5,2, 11,16.17,21.17,1, 11,5,10.08,1,
 11,10.67,14.42,1, 11,15.08,20.58,1, 12,5,10.08,2, 12,11.08,14.5,1, 12,15.08,20.58,1,
 12,5.75,9.75,1, 12,10.5,15.5,2, 12,16.17,21.17,1, 12,4.75,8.75,1, 12,9.5,12.83,2, 12,13.58,17.67,1,
 12,18.25,22.25,1, 13,4.67,9.5,1, 13,10.17,14.08,1, 13,15,19.92,1, 13,5.33,9.92,1, 13,10.5,15.5,2,
 13,16.17,21.17,1, 13,4.92,8.5,1, 13,9.08,12.33,2, 13,12.92,16.5,1, 13,17.58,22.08,1, 14,4.75,8.75,1,
 14,9.5,12.83,2, 14,13.58,17.67,1, 14,18.25,22.25,1, 14,5,10.08,2, 14,10.67,14.42,1,
 14,15.08,20.58,1, 14,5.75,9.75,1, 14,10.5,15.5,2, 14,16.17,21.17,1, 15,4.75,8.75,1, 15,9.5,12.83,2,
 15,13.58,17.67,1, 15,18.25,22.25,1, 15,5.00,10.08,2, 15,11.08,14.50,1, 15,15.08,20.58,1,
 15,5.75,9.75,1, 15,10.5,15.5,2, 15,16.17,21.17,1, 16,4.67,9.5,1, 16,10.67,14.42,1, 16,16.42,21.33,1,
 16,05.75,09.75,1, 16,10.5,15.5,2, 16,16.92,21.58,1, 16,4.75,8.75,1, 16,9.42,12.83,1,
 16,13.58,17.67,1, 16,18.25,22.25,1, 17,5,10.5,1, 17,11.67,15.58,1, 17,16.17,20.75,1,
 17,5.75,9.75,1, 17,10.5,15.5,2, 17,16.17,21.17,1, 17,4.75,8.75,1, 17,9.5,12.83,2, 17,13.58,17.67,1,
 17,18.25,22.25,1, 18,4.75,8.75,1, 18,9.5,12.83,2, 18,13.58,17.67,1, 18,18.33,21.58,1,
 18,5.75,9.75,1, 18,10.5,15.5,2, 18,16.17,21.17,1, 18,5,10.08,1, 18,10.67,14.42,1, 18,15.08,20.58,1,
 19,5,10.08,2, 19,11.08,14.5,1, 19,15.08,20.58,1, 19,5.75,9.75,1, 19,10.5,15.5,2, 19,16.17,21.17,1,
 19,4.75,8.75,1, 19,9.5,12.83,2, 19,13.58,17.67,1, 19,18.25,22.25,1, 20,4.67,9.5,1, 20,10.17,14.08,1,
 20,15,19.92,1, 20,5.33,9.92,1, 20,10.5,15.5,2, 20,16.17,21.17,1, 20,4.92,8.5,1, 20,9.08,12.33,2,

20,12.92,16.5,1, 20,17.58,22.08,1, 21,4.75,8.75,1, 21,9.5,12.83,2, 21,13.58,17.67,1,
 21,18.25,22.25,1, 21,5,10.08,2, 21,10.67,14.42,1, 21,15.08,20.58,1, 21,5.75,9.75,1, 21,10.5,15.5,2,
 21,16.17,21.17,1, 22,4.75,8.75,1, 22,9.5,12.83,2, 22,13.58,17.67,1, 22,18.25,22.25,1,
 22,5.00,10.08,2, 22,11.08,14.50,1, 22,15.08,20.58,1, 22,5.75,9.75,1, 22,10.5,15.5,2,
 22,16.17,21.17,1, 23,4.67,9.5,1, 23,10.67,14.42,1,23,16.42,21.33,1, 23,05.75,09.75,1,
 23,10.5,15.5,2, 23,16.92,21.58,1, 23,4.75,8.75,1, 23,9.42,12.83,1, 23,13.58,17.67,1,
 23,18.25,22.25,1, 24,5,10.5,1, 24,11.67,15.58,1, 24,16.17,20.75,1, 24,5.75,9.75,1, 24,10.5,15.5,2,
 24,16.17,21.17,1, 24,4.75,8.75,1, 24,9.5,12.83,2, 24,13.58,17.67,1, 24,18.25,22.25,1,
 25,4.75,8.75,1, 25,9.5,12.83,2, 25,13.58,17.67,1, 25,18.33,21.58,1, 25,5.75,9.75,1, 25,10.5,15.5,2,
 25,16.17,21.17,1, 25,5,10.08,1, 25,10.67,14.42,1, 25,15.08,20.58,1, 26,5,10.08,2, 26,11.08,14.5,1,
 26,15.08,20.58,1, 26,5.75,9.75,1, 26,10.5,15.5,2, 26,16.17,21.17,1, 26,4.75,8.75,1, 26,9.5,12.83,2,
 26,13.58,17.67,1, 26,18.25,22.25,1, 27,4.67,9.5,1, 27,10.17,14.08,1, 27,15,19.92,1, 27,5.33,9.92,1,
 27,10.5,15.5,2, 27,16.17,21.17,1, 27,4.92,8.5,1, 27,9.08,12.33,2, 27,12.92,16.5,1, 27,17.58,22.08,1,
 28,4.75,8.75,1, 28,9.5,12.83,2, 28,13.58,17.67,1, 28,18.25,22.25,1, 28,5,10.08,2, 28,10.67,14.42,1,
 28,15.08,20.58,1, 28,5.75,9.75,1, 28,10.5,15.5,2, 28,16.17,21.17,1;

TIPOPILOTO= 2,2,2,2,2,2,2,2,2,2,2,1,1,1,1,1,1,1;

T= 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1

0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0

0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0

0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0

0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0;

C= 1 0 1 1;

Mtope=1000;

E=1000;

Tope1=12; !LIMITE ACTIVIDAD DIARIA;

Tope2=100; !LIMITE HORAS DE VUELO EN 28 DIAS. El tiempo bloque de vuelo acumulado no deberá superar:100 horas de tiempo de vuelo en 28 días consecutivos;

Tope3=190; !La actividad acumulada no deberá superar:

60 horas de actividad en 7 días consecutivos, 110 horas de actividad en 14 días consecutivos, y 190 horas de actividad en 28 días consecutivos;

ENDDATA

Función objetivo:

$Min \sum_{i=1}^n y_i$; $y_i = 1$ si trabaja piloto i ; 0 si no ; $n = n^o$ pilotos

MIN=@SUM(PILOTOS:Y);

Restricciones:

$$\forall i : \sum_j x_{ij} \leq Mtope \cdot y_i ; Mtope = \text{número elevado} ; x_{ij} = 1 \text{ si vuelo } j \text{ lo hace piloto } i ; 0 \text{ si no}$$

@FOR(pilotos(i):@SUM(vuelos(j):X(i,j))<=Mtope*Y(i));

$$\forall j : \sum_{i/c_{ij}} x_{ij} = 1 ; x_{ij} = \text{matriz compatibilidad vuelos pilotos}$$

@FOR(vuelos(j):@SUM(pilotos(i)|c(tipopiloto(i),tipovuelo(j))#eq#1:X(i,j))=1);

$$\forall i : \sum_j [fin(j) - inicio(j)] \cdot x_{ij} \leq Tope\ 2 \cdot y_i ; Tope\ 2 = 100 ;$$

\Rightarrow Limite de tiempo bloque de vuelo acumulado en 28 días

@FOR(pilotos(i):@SUM(vuelos(j):(fin(j)-inicio(j))*X(i,j))<=Tope2);

$$\forall i, \forall d, \forall j / día(j) = d : n(i, d) \leq inicio(j) \cdot x(i, j) + E \cdot [1 - x(i, j)];$$

@FOR(pilotos(i):@FOR(dias(d):@FOR(vuelos(j)|dia(j)#eq#d:m(i,d)<=inicio(j)*X(i,j)+E*(1-X(i,j))));

$$\forall i, \forall d, \forall j / día(j) = d : Mx(i, d) \geq fin(j) \cdot x(i, j) + E \cdot [1 - x(i, j)];$$

@FOR(pilotos(i):@FOR(dias(d):@FOR(vuelos(j)|dia(j)#eq#d:Mx(i,d)>=fin(j)*X(i,j)-E*(1-X(i,j))));

$$\forall i, \forall d : [Mx(i, d) - m(i, d)] + 0.75 \leq Tope\ 1 ; Tope\ 1 = 12 : \text{límite diario de actualidad}$$

@FOR(dias(d):@FOR(pilotos(i):(Mx(i,d)-m(i,d))+0.75<=Tope1));

$$\forall i, \forall j, \forall k / j \neq k, dia(j) = dia(k), inicio(j) \leq inicio(k), fin(j) > inicio(k) \\ : x(i, j) + x(i, k) \leq 1;$$

@FOR(pilotos(i): @for(vuelos(j):@for(vuelos(k)|j#ne#k #and# dia(j)#eq#dia(k) #and# inicio(j)#le#inicio(k) #and# fin(j)#gt#inicio(k) : x(i,j)+x(i,k)<=1));

$$\forall i, \forall j, \forall s / j \neq s, \text{dia}(j) = \text{dia}(s) - 1, [24 - \text{fin}(j) + \text{inicio}(s)] \leq (12 + 0.33 + 0.75) \\ : x(i, j) + x(i, k) \leq 1;$$

@FOR(pilotos(i): @for(vuelos(j):@for(vuelos(s)|j#ne#s #and# dia(j)#eq#(dia(s)-1) #and#((24-
fin(j))+inicio(s))#le#(12+0.33+0.75)) : x(i,j)+x(i,s)<=1));

$$\forall i : \quad \sum_d M x(i, j) - n(i, d) + 0.75 \cdot \text{beta}(i, d) \leq \text{Tope 3};$$

⇒ *Tope 3 : Actividad total acumulada = 6h en 7 días ; 110h en 14 días ; 190h en 28 días*

@FOR(pilotos(i):@sum(dias(d):Mx(i,d)-m(i,d)+(0.75*beta(i,d)))<=Tope3);

$$\forall i, \forall d : \quad \sum_{j/\text{dia}(j)=d} x(i, j) \leq M_{\text{tope}} \cdot \text{beta}(i, d);$$

@FOR(pilotos(i):@FOR(dias(d):@SUM(vuelos(j)|Dia(j)#eq#d:X(i,j))<=Mtope*beta(i,d)));

$$\forall i, \forall d : \quad \sum_{j/\text{dia}(j)=d} x(i, j) \geq \text{beta}(i, d);$$

@FOR(pilotos(i):@FOR(dias(d): @SUM(vuelos(j)|Dia(j)#eq#d:X(i,j))>=beta(i,d)));

$$\forall i : \quad \sum_p \text{delta}(i, p) \leq 1;$$

@for(Pilotos(i):@SUM(patrones(p):delta(i,p))<=1);

$$\forall i, \forall p, \forall d : \text{delta}(i, p) \leq 1 - T(p, d) + [2 \cdot T(p, d) \cdot \text{beta}(i, d)] - \text{beta}(i, d);$$

```
@FOR(pilotos(i):@FOR(patrones(p):@FOR(dias(d):delta(i,p)<=1-T(p,d)+(2*T(p,d)*beta(i,d))-
beta(i,d))));
```

Restricciones y declaraciones auxiliares:

```
@for(Pilotos(i):@BIN(Y(i)));
```

```
@for(Pilotos_vuelos(i,j):@BIN(X(i,j)));
```

```
@for(Pilotos_dias(i,d):@BIN(beta(i,d)));
```

```
@for(Pilotos_patrones(i,p):@BIN(delta(i,p)));
```

END

CAPÍTULO 6

EXPERIMENTACIÓN

EXPERIMENTACIÓN

Dado que la actividad acumulada no deberá superar:

60 horas de actividad en 7 días consecutivos,

110 horas de actividad en 14 días consecutivos, y

190 horas de actividad en 28 días consecutivos.

Realizaremos 3 problemas, la programación de pilotos para 7, 14 y 28 días.

PROGRAMACIÓN PARA 7 DÍAS.

Los datos para Lingo serán:

VUELOS/1..70/:dia,inicio,fin,tipovuelo;

PILOTOS/1..20/:tipopiloto,Y;

DIAS/1..7/;

Ajustaremos el conjunto Vuelos a los 7 primeros días y la matriz T para considerar 7 días en cada patrón.

No tendremos en cuenta el Tope 2:

!Tope2=100; !LIMITE HORAS DE VUELO EN 28 DIAS. El tiempo bloque de vuelo acumulado no deberá superar: 100 horas de tiempo de vuelo en 28 días consecutivos;

Definiremos el Tope 3 para este caso en concreto:

Tope3=60; !La actividad acumulada

Solución mostrada:



Reporte:

Feasible solution found.

Objective value: 6.000000

Extended solver steps: 27204

Total solver iterations: 6516811

Row	Slack or Surplus	Dual Price
1	6.000000	-1.000000
2	0.000000	0.000000
8270	0.000000	0.000000
8271	0.000000	0.000000

PROGRAMACIÓN PARA 14 DÍAS.

Los datos para Lingo serán:

VUELOS/1..140/:dia,inicio,fin,tipovuelo;

PILOTOS/1..20/:tipopiloto,Y;

DIAS/1..14/;

Ajustaremos el conjunto Vuelos a los 14 primeros días y la matriz T para considerar 14 días en cada patrón.

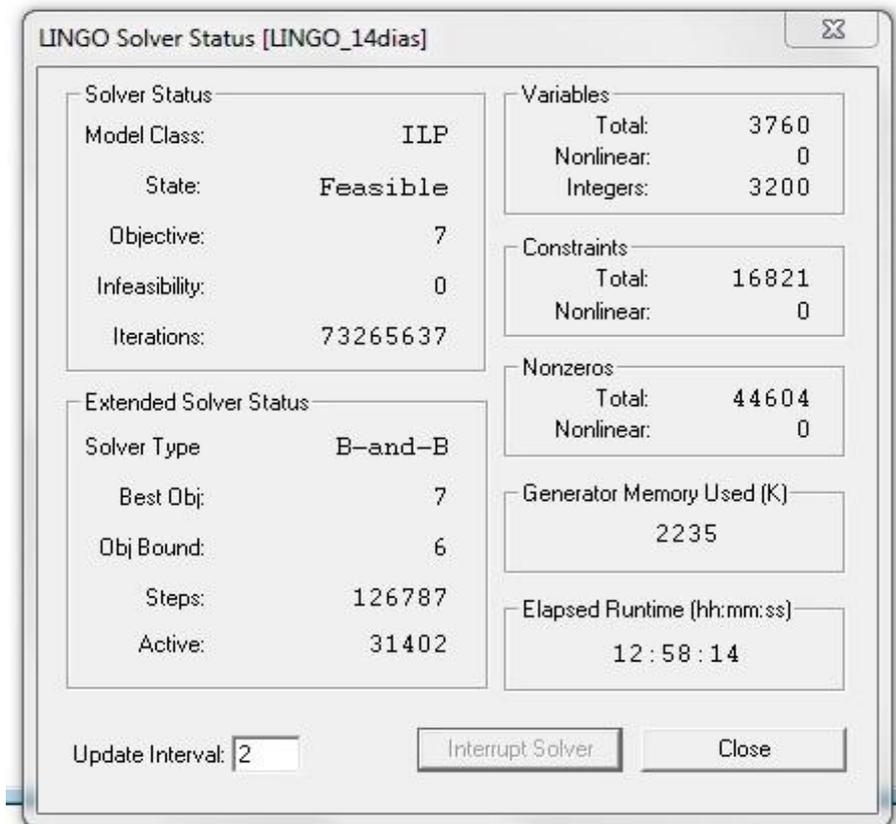
No tendremos en cuenta el Tope 2:

!Tope2=100; !LIMITE HORAS DE VUELO EN 28 DIAS. El tiempo bloque de vuelo acumulado no deberá superar: 100 horas de tiempo de vuelo en 28 días consecutivos;

Definiremos el Tope 3 para este caso en concreto:

Tope3=110; !La actividad acumulada

Solución mostrada:



Reporte:

Feasible solution found.

Objective value: 7.000000

Extended solver steps: 126787

Total solver iterations: 73265637

Row	Slack or Surplus	Dual Price
1	7.000000	-1.000000
2	975.0000	0.000000
16820	1.000000	0.000000
16821	0.000000	0.000000

PROGRAMACIÓN PARA 28 DÍAS.

Los datos para Lingo serán:

VUELOS/1..280/:dia,inicio,fin,tipovuelo;

PILOTOS/1..20/:tipopiloto,Y;

DIAS/1..28/;

Ajustaremos el conjunto Vuelos a los 28 días y la matriz T para considerar 28 días en cada patrón.

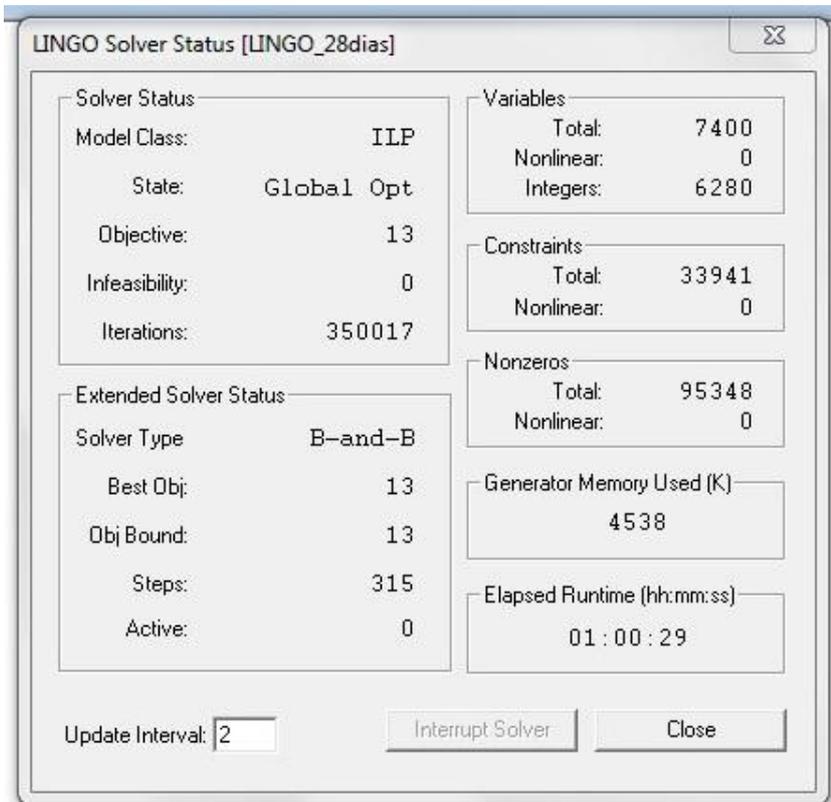
Si tendremos en cuenta el Tope 2:

Tope2=100; !LIMITE HORAS DE VUELO EN 28 DIAS. El tiempo bloque de vuelo acumulado no deberá superar: 100 horas de tiempo de vuelo en 28 días consecutivos;

Definiremos el Tope 3 para este caso en concreto:

Tope3=190; !La actividad acumulada

Solución mostrada:



Reporte:

Feasible solution found.

Objective value: 13.00000

Extended solver steps: 331

Total solver iterations: 340785

Row	Slack or Surplus	Dual Price
1	13.00000	-1.000000
2	978.0000	0.00000
33940	0.000000	0.000000
33941	1.000000	0.000000

ANÁLISIS DE RESULTADOS CON LINGO

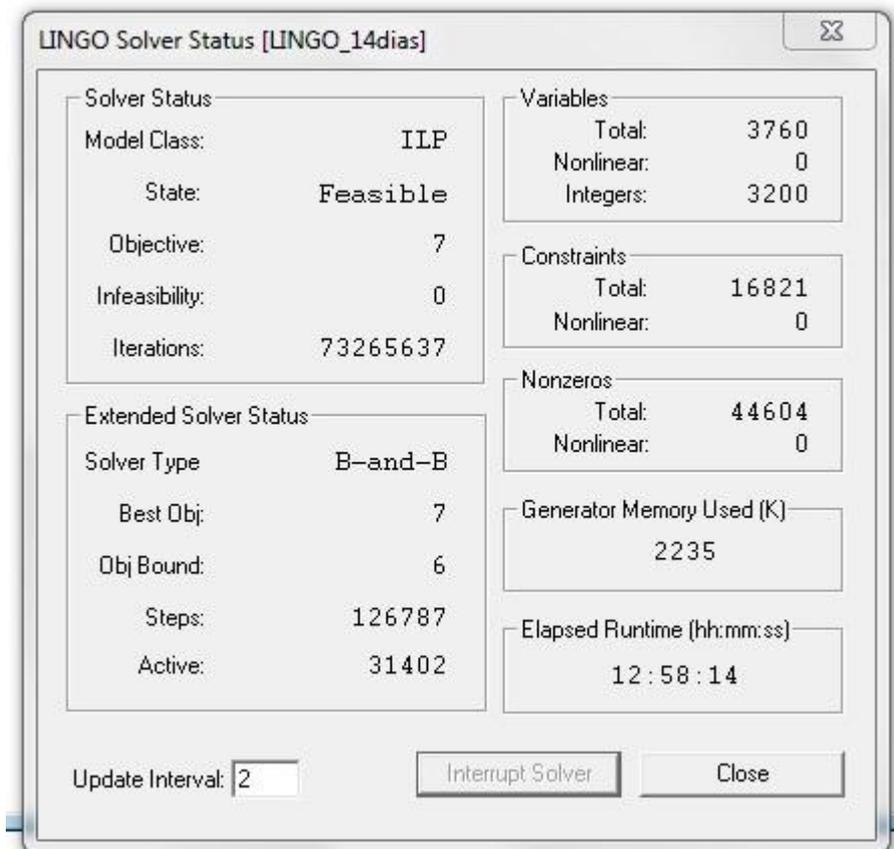
Se trata de un problema de programación de trabajos en intervalos con varias clases de máquinas, de manera que como se comentó en apartados anteriores nos encontramos en un problema de clase NP. Los problemas NP son problemas de optimización complejos que requieren múltiples iteraciones para obtener el óptimo.

En nuestro primer problema suponemos una programación de 7 días, y los resultados fueron los siguientes:



En casi 37 minutos, después de 6516811 iteraciones, Lingo consigue una solución óptima de **6 pilotos** necesarios para dicha programación de 7 días, 70 vuelos y 3 aviones.

En el segundo problema, de mayor envergadura, con 14 días a programar, es decir, 140 vuelos de 3 aviones, con las restricción de actividad máxima adaptada a esos días, después de más de 12 horas calculando, no se alcanzó el óptimo, pero la función objetivo no cambio de valor desde antes de la primera hora de cálculo:



Después de parar los cálculos justo antes de que se cumpliesen 13 horas de procesamiento, vemos como el objetivo encontrado y mantenido durante varias horas fué de **7 pilotos** necesarios para dicha programación. En todo ese tiempo Lingo realizó 73265637 iteraciones.

Para nuestro tercer y último caso, teniendo en cuenta el conjunto completo de los 28 días de vuelo de los 3 aviones en Sevilla, actualizando la actividad máxima de trabajo para cada piloto en esos 28 días y añadiendo esta vez la restricción del número máximo de horas de vuelo en ese mismo período, se alcanzó una solución óptima:



El valor óptimo en este caso de la función objetivo fué de **13 pilotos** necesarios y suficientes para realizar todos los vuelos, cumpliendo todas las restricciones.

Lingo tardó 1 hora en encontrarla, y realizó 350017 iteraciones.

CAPÍTULO 7

CONCLUSIONES

CONCLUSIONES

En este proyecto se han unido los conocimientos teóricos de programación de trabajos en máquinas, adquiridos principalmente en la asignatura de Secuenciación de la titulación de Ingeniero Industrial, y la modelación matemática adquirida en varias asignaturas durante toda la carrera, junto con el sector que profesionalmente conozco, los vuelos comerciales. Para ello ha sido también necesario el adquirir los conocimientos de las librerías de optimización en Lingo, a través de la documentación de la asignatura de Métodos Cuantitativos de Gestión y el Manual de Usuarios facilitado por Lindo Systems.

Después de introducir y plantear nuestro problema, hemos hecho un resumen de la teoría en la que basamos nuestra posterior manera de resolver, la programación de trabajos en intervalos. Y concretamos nuestro problema para sus características de intervalos fijos, minimización de los recursos a usar (los pilotos), y las dos clases de vuelos y pilotos que íbamos a tener en cuenta.

Hemos realizado también una introducción y explicación de Lingo, además del código generado para la resolución de nuestros 3 problemas.

La mayor dificultad del proyecto ha sido la modelación matemática de todas las restricciones y su correspondiente programación en Lingo, siendo esto último totalmente novedoso para el alumno.

Como hemos visto, se han conseguido soluciones óptimas en dos de los 3 problemas, habiéndose alcanzado igualmente el objetivo en el otro, sin prácticamente variación, para cálculos de menos de una hora.

La diferencia entre los problemas, aparte del tamaño, estaba en la restricción de actividad máxima de vuelo (horas de trabajo), que era distinta para cada uno de ellos. Además, la restricción del número máximo de horas bloque de vuelo sólo se aplicaba para el tercer problema (28 días de vuelos), dado que no existen requisitos por normativa para períodos inferiores, como sí pasa con la actividad. Es por todo esto por lo que surgen las diferencias en los resultados.

Igualmente los resultados son muy coherentes:

- 6 pilotos necesarios para 3 aviones, 70 vuelos y 7 días.
- 7 pilotos necesarios para 3 aviones, 140 vuelos y 14 días.
- 13 pilotos necesarios para 3 aviones, 280 vuelos y 28 días.

Para comparar los resultados con la realidad, podemos decir que la base actual de Vueling en Sevilla tiene 15 pilotos de cada especialidad (comandante y segundo), lo que es un valor muy próximo a nuestro último cálculo.

Nosotros no tuvimos en cuenta vacaciones, posibles bajas médicas, posibles reducciones por paternidad, cuidados de familiares u otros menesteres, etc. Ésto habría acercado aún más nuestro cálculo a la plantilla real.

Las aerolíneas actuales manejan un ratio de entre 4 y 6 pilotos de cada especialidad por avión, estando lo más común de las grandes compañías de nueva generación en el entorno de 4.8 pilotos por avión. Dicho valor es muy próximo a nuestro resultado para la programación completa de 28 días, **4.33 pilotos por avión**.

Como podemos ver, programas como el nuestro, a partir de unos vuelos a programar, nos podrían dar una estimación inicial de la plantilla necesaria para hacerlos. Dicho proceso en una aerolínea comienza por parte del departamento comercial, que pasa los vuelos vendidos al departamento de programación de tripulaciones, y este a su vez al departamento de recursos humanos, con la plantilla necesaria para hacerlos. Aunque estas relaciones siempre son bidireccionales.

Actualmente en Vueling el cálculo de pilotos necesarios se ve apoyado por estimaciones basadas en históricos sobre bajas médicas, peticiones de reducción de jornada, etc.

Igualmente, siempre es necesaria una idea inicial del valor mínimo de pilotos a tener en plantilla para un cierto número de vuelos, y para esa función sería ideal una versión similar, pero ampliada, de nuestro programa.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

- *Documentación de la asignatura “Secuenciación”*

Profesor: José Manuel García Sánchez

Universidad de Sevilla, 2016.

- *Documentación de la asignatura “Métodos Cuantitativos de Gestión”.*

Profesor: José Manuel García Sánchez

Universidad de Sevilla, 2016.

- *Lingo Users Manual*

Lindo Systems Inc.

Chicago, Illinois.

- *Scheduling. Theory, Algorithms and Systems.*

Autor: M.Pinedo

Editorial: Prentice-Hall, 1995