Sección de Matemáticas
Universidad de La Laguna

Nieves García Hernández

Una aproximación a los problemas de planificación estocástica

An approach to stochastic scheduling problems

Trabajo Fin de Grado
Grado en Matemáticas
La Laguna, Junio de 2019

Dirigido por
David Alcaide López de Pablo

David Alcaide López de Pablo
Departamento de Matemáticas, Estadística e Investigación Operativa
Universidad de La Laguna
38271 La Laguna, Tenerife

Agradecimientos

A mi tutor, David Alcaide López de Pablo, por su gran apoyo y tiempo. A mi hermana, por su apoyo durante la carrera.

Nieves García Hernández
La Laguna, 10 de junio de 2019

Resumen · Abstract

Resumen

Los problemas de Planificación y Secuenciación aparecen cuando deben realizarse determinadas actividades o tares (los trabajos) por ciertos entes (las máquinas) con el fin de alcanzar y optimizar ciertos objetivos (los criterios de optimización). Se tienen que asignar las máquinas a los trabajos para procesarlos con ciertas condiciones a fin de optimizar los criterios considerados.

Esta Memoria se dedica a hacer una introducción a los Problemas de Planificación con especial mención a los Problemas de Planificación Estocástica. También se dedica un capítulo a una aplicación práctica a un problema de ordenación secuencial estocástico jerárquico (Hierarchical stochastic sequential ordering problem (HSSOP)).

Palabras clave: *Investigación Operativa - Problemas de planificación - Planificación estocástica - Problema HSSOP*

Abstract

The Scheduling Problems arise when several tasks (the jobs) must be performed by several entities (the machines) in order to achieve and optimise some aims (the optimisation criteria). The machines must be assigned to the jobs to process them under some constraints with the purpose of optimising the considered criteria. This Report is devoted to introduce the Scheduling Problems making special mention to Stochastic Scheduling. A chapter devoted to a practical application to a Hierarchical Stochastic Sequential Ordering Problem (HSSOP) is also included.

Keywords: *Operational Research – Scheduling Problems – Stochastic Scheduling – Problem HSSOP*

Contenido

Introducción

Estando a punto de finalizar mis estudios de Grado en Matemáticas en la Universidad de La Laguna, y llegado el momento de decidir la línea para desarrollar el Trabajo de Fin de Grado, opté por la línea de Problemas de Planificación Estocástica ofrecida por el Área de Conocimiento de Estadística e Investigación Operativa y dirigida por el profesor David Alcaide López de Pablo.

Durante mis estudios de Grado he ido aprendiendo mucho y he elegido diferentes asignaturas optativas como Algebra Conmutativa, Topología Algebraica, Geometría Diferencial y Matemáticas para la enseñanza. Además, he realizado las Prácticas Externas en la empresa Carrefour. Este perfil de estudios, junto con el interés que despierta y la buena crítica que recibe me ha motivado a elegir la temática del Trabajo Fin de Grado en una de las líneas ofertadas por el Área de Conocimiento de Estadística e Investigación Operativa.

Con mi personalidad deductiva y entusiasmo me esmeré en encontrar un tutor que me pudiera proporcionar unas directrices y correciones adecuadas al tema, y que me permitiera avanzar en el desarrollo del Trabajo de Fin de Grado. En esa búsqueda encontré a mi tutor, el profesor David Alcaide, al cual quiero agradecer su labor.

En el proceso del Trabajo de Fin de Grado se ha trabajado con conceptos relacionados con la Planificación y Secuenciación de Tareas, que consiste en asignar recursos o máquinas a tareas que necesitan ser procesadas en unos tiempos y bajo ciertas condiciones. En concreto, con la presente memoria de Trabajo de Fin de Grado se pretende hacer “una aproximación a los problemas de planificación estocástica”. Por ello, en los primeros capítulos de esta memoria nos hemos esforzado en asimilar los conceptos fundamentales de los problemas de planificación estocástica. La memoria incluye también, en un último capítulo a modo ilustrativo, una aplicación práctica a un problema de ordenación secuencial jerárquico.

1

Introducción general a los problemas de planificación

A lo largo de la historia, el ser humano se ha cuestionado en diversas situaciones cómo plantear y resolver infinidad de problemas reales.
En el acontecimiento histórico de la Segunda Guerra Mundial se dieron una serie de problemas que propiciaron la aparición de un grupo de científicos que trabajarían con la armada británica para resolver dichas cuestiones. Dicho grupo, encargado de afrontar las peticiones del almirantazgo británico, fue más allá, y analizó varias cuestiones que ni el ejército se había planteado; pudiendo resolverlas con éxito. Este hecho concluyó en la formación de una unidad especial formada por un grupo de científicos con una gran variedad de conocimientos en diversos campos. En ese momento se creó la base de lo que hoy en día llamamos Investigación Operativa, llevada a cabo por estos científicos profesionales; y cuyo nombre proviene de la investigación sobre las operaciones militares del momento.

Una vez finalizada la guerra, los científicos se dispersaron a distintos puestos de trabajo tales como: la Universidad, economía, industria,... donde emplearon los resultados obtenidos en el grupo de Investigación Operativa para resolver problemas particulares y dar a conocer los proyectos creados.

La Investigación Operativa se introdujo poco a poco en la sociedad abarcando problemas de planificación o decisiones político-económico-sociales, cuyo auge surgió con la aparición de ordenadores electrónicos que facilitaron los cálculos imposibles de realizar a mano. La rapidez con la que evolucionó esta disciplina originó multitud de subdisciplinas reconocidas a nivel mundial entre las que se encuentran por ejemplo: Grafos, Teoría de Juegos, Teoría de Colas, Programación matemática,... cuya similitud es: la Investigación Operativa, la cual se ha definido en la Sociedad de I. O. de América de la siguiente forma: “La I. O. tiene por objeto decidir, mediante métodos científicos, sobre el diseño o modelo que optimice el funcionamiento de los sistemas hombre-máquina, generalmente bajo condiciones que implican la utilización de recursos escasos”. Entendemos por sistema un conjunto de unidades que trabajan en un ambiente de forma interactiva utilizando ciertos recursos (información, energía, vida,...) para llegar a un objetivo común; y ambiente o contexto las circunstancias de trabajo de las unidades. Cabe destacar de la definición que el objetivo principal de la Investigación Operativa es aportar ayuda objetiva a uno o más individuos en la toma de decisión de un problema real planteado (una descripción detallada del Proceso de Modelización puede verse en Sánchez-García (1982) [6]).

1.1. Generalidades sobre planificación

Uno de los campos desarrollados por la Investigación Operativa es la Planificación y Secuenciación (*"Scheduling and Sequencing"*) que intenta resolver modelos relacionados con la planificación de la producción, mantenimiento y reparación de productos, control y gestión de máquinas, etc. El objetivo principal consiste en asignar las máquinas disponibles a ciertas tareas o trabajos a lo largo del tiempo de manera que se cumplan los requisitos de optimalidad. En esta definición se observan los tres componentes esenciales de la planificación, los cuales son: los trabajos o tareas que se tienen que realizar, los recursos o máquinas disponibles y los objetivos que se pretenden conseguir. La tercera componente será clave a la hora de escoger la planificación o las planificaciones que entre las posibles cumplan los requisitos requeridos, en cuyo caso se denominarán óptimas.

Entre los modelos de planificación existentes se encuentra la planificación *determinística* (*"deterministic machine schedulling"*). Estos modelos se caracterizan por poseer datos de entrada conocidos. En cambio, cuando los datos de entrada se vuelven variables aleatorias, se habla de modelos de planificación *estocásticos* (*"stochastic machine schedulling"*). En ambos tipos de modelos, es más corriente encontrarnos con un único criterio de optimalidad, esto es, un único objetivo que determine si una planificación es óptima o no. Estos modelos reciben el nombre de *Modelos de Planificación Unicriterio*. Sin embargo, es posible que existan más criterios de optimalidad, siendo estos casos menos estudiados que los anteriores. A estos últimos se les denomina *Modelos de Planificación Multicriterio*.

1.2. Aspectos relevantes en los problemas de planificación determinística

La formulación general de los problemas de planificación (cuya descripción puede verse en Alcaide-López-de-Pablo (2008) [2]) se entiende como: se requiere ejecutar n trabajos o tareas J_j (con $j = 1, ..., n$), disponiendo de m máquinas M_i (con $i = 1, ..., m$). Teniendo en cuenta que en estos problemas la notación sugiere hablar del "trabajo j" en lugar de J_j y de la "máquina i" en vez de M_i. Al mismo tiempo, hay que tener en cuenta que cada máquina puede procesar únicamente un trabajo, y que en un instante dado, cada trabajo se puede procesar en a lo sumo una máquina. Además, en la literatura la mayor parte de los estudios hacen referencia a modelos unicriterio cuyo criterio es una función regular, esto es, una función no decreciente en cada uno de los tiempos de completación de los trabajos.

Cada uno de los trabajos del modelo tiene asociado los siguientes datos:

1. Una cantidad m_j de operaciones $O_{1j},...,O_{m_jj}$ en las que se divide el trabajo j. Completar el trabajo j es completar todas las operaciones $O_{1j},...,O_{m_jj}$ en las que se fracciona. Cada operación del trabajo j se procesa en una única máquina M_i. Cuando la operación O_{ij} del trabajo j se procesa en la máquina M_k se denota $\mu_{ij} = k$. En el caso de que exista algún trabajo j con una única operación $m_j = 1$ entonces se denota $\mu_{1j} := \mu_j = k$ si se realiza en la máquina M_k.
2. Un tiempo de procesamiento p_{ij} del trabajo j en la máquina M_i. Puede suceder que el trabajo j tenga un tiempo de procesamiento p_{ij} igual en todas las máquinas del sistema, en cuyo caso se denota p_j al tiempo de procesamiento de dicho trabajo j en cualquier máquina $M_i(i = 1,...,m)$. Si el trabajo j se divide en un número m_j de operaciones $O_{1j}, O_{2j},...,O_{m_jj}$ entonces se denota el tiempo de procesamiento de la operación O_{ij} como p_{ij}.
3. Una fecha de disponibilidad r_j, en la cual el trabajo j está disponible para empezar a ser procesado. Si $r_j = 0 \; \forall j$ todos los trabajos están disponibles desde el comienzo. Los problemas con esta característica se denominan *problemas estáticos*. Cuando los tiempos de disponibilidad están disponibles en instantes de tiempos distintos estamos considerando *problemas dinámicos*.
4. Una fecha de comienzo S_j que indica el instante de tiempo en el que el trabajo j ha empezado a ser procesado. Dicha fecha de comienzo S_j puede variar dependiendo de la planificación.
5. Una fecha límite d_j en la cual el trabajo j debería estar realizado.
6. Un peso o ponderación w_j^k que nos muestra la importancia del trabajo j en comparación con el resto de trabajos del modelo considerando el criterio k con $1 \leq k \leq K$, siendo K el número de criterios del problema.
7. Una función de costo f_j^k para cada criterio k del problema $(1 \leq k \leq K)$, donde $f_j^k(t)$ es el coste del trabajo j considerando el criterio k y $C_j = t$.

Notar que cuando el problema es *determinístico* todos estos datos son conocidos; sin embargo, ante un problema *estocástico* uno o varios de los datos anteriores son desconocidos, por lo que se definen como variables aleatorias.

A la hora de clasificar los problemas de planificación se emplean los tres campos $\alpha|\beta|\gamma$, nomenclatura propuesta por Graham et al. (1979) [4]. El primer campo hace referencia a las características de las m máquinas $M_i(i = 1,...,m)$, el segundo campo hace alusión a los n trabajos o tareas $J_j(j = 1, ..., n)$, y el tercer campo está dedicado a los criterios junto con el modo de optimización considerado.

A continuación, mostramos las diferentes características que pueden poseer las máquinas y las tareas, y los distintos criterios de optimalidad posibles. Dicho de otra forma, estudiaremos los campos $\alpha|\beta|\gamma$ descritos anteriormente (véase Alcaide-López-de-Pablo (1995) [1] y (2008) [2]).

1.2.1. Características de las máquinas: campo α

En los problemas de planificación es posible disponer de una o varias máquinas, siendo ésta la característica principal del parámetro α. Se denota entonces $\alpha = 1$ en el caso de considerar una única máquina.

Podemos referirnos a *paralelismo* cuando al considerar dos o más máquinas, varias de ellas ejecuten diferentes trabajos u operaciones de distintos trabajos al mismo tiempo.

Además, existen dos tipos de problemas según la funcionalidad de las máquinas. Los problemas donde todas las máquinas realizan las mismas tareas (*máquinas no especializadas*), y los problemas donde existen una o más máquinas que ejecutan ciertas tareas específicas y no pueden ejecutar cualquier otra tarea (*máquinas especializadas*).

A continuación, describimos las características de las máquinas al considerar máquinas especializadas o no especializadas.

Máquinas no especializadas

Al considerar máquinas no especializadas, se entiende que $\alpha \in \{1, P, Q, R\}$. Cada trabajo posee una única operación que se deba ejecutar, es decir, $m_j = 1$ para todo j. Cada trabajo J_j tiene un tiempo de procesamiento p_{ij} en la máquina M_i, cuyo valor puede diferir de una máquina a otra.

Cada valor que puede tomar α en esta sección constan de ciertas particularidades que se deben tener en cuenta:

- $\alpha = 1$. Se considera una única máquina ($m = 1$) para procesar los distintos trabajos $J_j(j = 1, ..., n)$. Los tiempos de procesamiento p_{ij} de los distintos trabajos se denotan $p_{1j} = p_j \; \forall j$, siendo p_j el tiempo de procesamiento del trabajo j en esta máquina.
- $\alpha = P$. Máquinas en paralelo e idénticas. Las máquinas son no especializadas, por lo que todas las máquinas tienen la capacidad de procesar cualquier trabajo. Todas las máquinas son idénticas: el tiempo necesario para procesar un trabajo cualquiera en una máquina es el mismo para cualquier otra máquina del sistema. Entonces, el tiempo de procesamiento p_{ij} de cada trabajo j es igual en todas las máquinas $M_i(i = 1, ..., m)$, por lo tanto, $p_{ij} = p_j \; \forall i$.

- $\alpha = Q$. Máquinas en paralelo y uniformes. Las máquinas no están especializadas y la velocidad de cada máquina se sabe que es constante (distintas en cada una) y no depende del trabajo que se esté procesando en ella. El tiempo de procesamiento del trabajo j en la máquina M_i es $p_{ij} = p_j/q_i$, siendo q_i la velocidad constante de la máquina M_i que no depende del trabajo que se esté procesando en dicha máquina y p_j es el tiempo de procesamiento del trabajo j en una máquina M que se toma como referencia en el sistema.

  Un ejemplo sigue de considerar un sistema con tres máquinas en paralelo y uniformes. Todas pueden procesar todos los trabajos del sistema, pero la primera máquina es capaz de procesar cualquier trabajo en la mitad de tiempo que la segunda pero tarda tres veces más que la tercera.
- $\alpha = R$. Máquinas en paralelo y no relacionadas. Las máquinas no están especializadas, y es imposible relacionar las velocidades de las máquinas entre si. La velocidad de la máquina depende del trabajo que se esté procesando y de la máquina en si.

Máquinas especializadas

Al considerar máquinas especializadas, entedemos que $\alpha \in \{F, O, J\}$. Una máquina especializada es capaz de realizar un conjunto de tareas pero no todas. Por lo tanto, es posible que el número de operaciones m_j del trabajo j sea distinto de 1. Cada trabajo se podría dividir en m_j operaciones $O_{1j}, O_{2j}, ...O_{m_j j}$ que podrían exigir más de una máquina para ser procesadas. Además, podría darse la situación de que $m \leq m_j$ en la que habrá alguna máquina que procesa más de una operación del trabajo j.

A continuación, definimos las características de cada posible valor de α al considerar máquinas especializadas:

- $\alpha = F$. Sistema *Flow Shop*. Cada trabajo j consta de $m_j = m$ operaciones $O_{1j}, O_{2j}, ..., O_{mj}$. La operación O_{ij} es procesada en la máquina M_i con un tiempo de procesamiento p_{ij}. El orden de procesamiento es siempre el mismo, es decir, está prefijado para cada trabajo del sistema. No tiene porque darse la circunstancia de que cada trabajo se tenga que procesar en cada máquina.
- $\alpha = O$. Sistema *Open Shop*. Cada trabajo j consta de una cantidad $m_j = m$ operaciones $O_{1j}, O_{2j}, ..., O_{mj}$. La operación O_{ij} se procesa en la máquina M_i con un tiempo de procesamiento p_{ij}. Todos los trabajos se procesan en todas las máquinas, y, se han ordenado las operaciones del trabajo j de tal forma que la operación O_{ij} se procesa en la máquina i, es decir, $\mu_{ij} = i \ \forall j$. El orden de procesamiento es irrelevante.
- $\alpha = J$. Sistema *Job Shop*. Cada trabajo j consta de una cantidad m_j de operaciones $O_{1j}, O_{2j}, ..., O_{mj}$. No tiene porque darse $m_j = m$. En este sistema m_j puede ser mayor que m, por lo que se puede dar el caso de que alguna máquina procese más de una operación de un trabajo. La trayectoria de la

máquinas de cada trabajo se conoce pero no tiene porque ser igual para todos los trabajos.

Cuando el parámetro α no toma el valor 1, estamos considerando un número m de máquinas que es arbitrario pero fijo. Por ejemplo, en un problema $P|\beta|\gamma$ el valor del número m de máquinas puede ser cualquier valor fijo que escojamos; sin embargo, en un problema $P2|\beta|\gamma$ la cantidad m de máquinas es 2.

1.2.2. Características de los trabajos: campo β

Los trabajos del sistema poseen una serie de condiciones para ser ejecutados. Estas condiciones se representan con el parámetro β, que a su vez se divide en varios parámetros: $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6$.

Interrupciones: $\beta_1 \in \{pmtn, \Box\}$

- $\beta_1 = pmtn$. Las interrupciones están permitidas. Se puede parar el procesamiento de cualquier trabajo u operación y retomarlo más tarde. Esta condición interesa solo si al retomar la operación no se tiene que repetir el trabajo que ya se había realizado.
- $\beta_1 = \Box$. No se permiten interrupciones.

Relaciones de precendencia: $\beta_2 \in \{prec, tree, \Box\}$

- $\beta_2 = prec$. Existen relaciones de precedencia general entre los trabajos. Dichas relaciones se representan con un grafo G dirigido acíclico (para cada vértice del grafo G no existe ningún camino que empiece y termine en dicho vértice) cuyos vértices simbolizan los trabajos $J_j (j = 1, ..., n)$ y los arcos $J_j \to J_k$ indican que el trabajo J_k no puede comenzar a procesarse hasta que el trabajo J_j se haya completado. Asimismo, si los trabajos se dividen en un número m_j de operaciones y dichas operaciones poseen relaciones de precedencia se pueden representar en un grafo G dirido acíclico donde los vértices simbolizan las operaciones $O_{ij} (i = 1, ..., m_j; j = 1, ..., n)$ y los arcos $O_{ij} \to O_{kl}$ indican que la operación O_{kl} del trabajo J_l no puede comenzar a procesarse hasta que la operación O_{ij} del trabajo J_j se haya completado.
- $\beta_2 = tree$. Existen relaciones de precedencia general entre los trabajos. Estas relaciones de precedencia se representan con un grafo G, el cual es un árbol. Es decir, cualesquiera dos vértices del grafo están conectados por un único camino.
  - $\beta_2 = intree$. De cada vértice del grafo sale como mucho un arco.
  - $\beta_2 = outtree$. A cada vértice del grafo llega como mucho un arco.
- $\beta_2 = \Box$. No existen relaciones de precedencia entre los trabajos.

Existencia de fechas de disponibilidad: $\beta_3 \in \{r_j, \Box\}$

- $\beta_3 = r_j$. En este caso las fechas de disponibilidad son distintas. Nos encontramos con los *problemas de planificación dinámicos*.
- $\beta_3 = \Box$. Todos los trabajos poseen la misma fecha de disponibilidad r_j. Si los trabajos están disponibles desde el inicio se denotará $r_j = 0\,\forall j$; en cambio cuando están disponibles desde un momento concreto $r_j = r\,\forall j$. Sin pérdida de generalidad se tomará $r_j = 0\,\forall j$. Nos encontramos con los *problemas de planificación estáticos*.

Cotas al número de operaciones: $\beta_4 \in \{m_j \leq mUB, \Box\}$

- $\beta_4 = m_j \leq mUB$. Dado un problema *Job Shop*, donde se cumple que el número de operaciones m_j de los trabajos $J_j (j = 1, ..., n)$ es mayor que el número de máquinas m, se acotan superiormente dichos valores m_j. Más en concreto, el número de operaciones m_j son acotados superiormente por un valor mUB.
- $\beta_4 = \Box$. No existe una cota superior para el número de operaciones m_j de los trabajos $J_j (j = 1, ..., n)$.

Tiempos de procesamiento: $\beta_5 \in \{p_{ij} = 1, \Box\}$

- $\beta_5 = p_{ij} = 1$. Las operaciones emplean el mismo tiempo de procesamiento.
- $\beta_5 = \Box$. Los tiempos de procesamiento p_{ij} son distintos.

Recursos adicionales: $\beta_6 \in \{res\lambda\sigma\rho, \Box\}$

- $\beta_6 = res\lambda\sigma\rho$. Los trabajos u operaciones poseen recursos adicionales. Están disponibles s clases de recursos adicionales, que denotaremos como RA_1, RA_2,..., RA_s en unas cantidades $ra_1, ra_2, ..., ra_s$, entendiendo que ra_i es la cantidad de recurso adicional RA_i. Entonces cada trabajo J_j u operación O_{ij} dispone de estos recursos, es decir, tiene asociado un vector de dichos recursos adicionales:
$$ra(J_j) = (ra_1(J_j), ra_2(J_j), ..., ra_s(J_j))$$
$$ra(O_{ij}) = (ra_1(O_{ij}), ra_2(O_{ij}), ..., ra_s(O_{ij}))$$
donde $ra_l(J_j) \in [0, ra_l]$ es la cantidad fijada del recurso RA_l para el trabajo J_j y $ra_l(O_{ij}) \in [0, ra_l]$ es la cantidad fijada del recurso RA_l para la operación O_{ij}.

  Cuando todas las cantidades anteriores son fijas se emplean los valores λ, σ y ρ. El valor λ representan el número de recursos adicionales, el valor σ la cantidad fija de cada recurso adicional y el valor ρ la cantidad fija de cada recurso que requiere cada trabajo.
- $\beta_6 = \Box$. No existen recursos adicionales.

1.2.3. Criterios de Optimalidad: campo γ

En un problema de planificación el parámetro γ representa los criterios de optimalidad, los cuales son: el número de funciones objetivos a tener en cuenta, las características de éstas que están relacionadas con la planificación, y la clase de optimización que debemos llevar a cabo (buscar puntos eficientes, puntos extremos, optimización simultánea u optimización jerárquica).

El número total de criterios del problema lo representamos con K, por lo tanto, cada función objetivo del problema se denota γ^k con $1 \leq k \leq K$. En general, las funciones objetivo son funciones crecientes en los tiempos de completación de los trabajos $C_1, C_2, ..., C_n$ (funciones regulares). Es decir, dada una función $\gamma^k(C_1, C_2, ..., C_n)$ de los tiempos de completación $C_1, C_2, ..., C_n$ de los n trabajos, se tiene que:

$\gamma^k(C_1, C_2, ..., C_n) \leq \gamma^k(C'_1, C'_2, ..., C'_n)$ si $C_1 \leq C'_1, C_2 \leq C'_2, ..., C_n \leq C'_n$.

Las funciones objetivos se definen según unas variables que a su vez se definen para cada trabajo $J_j (j = 1, ..., n)$ del modelo. Estas variables son las siguientes:

- Tiempo de completación C_j. El instante de tiempo en el que el trabajo j se ha terminado de procesar.
- La demora $L_j = C_j - d_j$. Una demora positiva nos proporciona el retraso o la *tardanza* en la completación de dicho trabajo. En cambio, una demora negativa nos indica que el trabajo j se ha completado antes de su fecha límite d_j, siendo el valor absoluto de esta demora la cantidad de tiempo de anticipación del trabajo a su fecha límite.
- La tardanza $T_j = max\{0, L_j\}$. Es la cantidad de tiempo de retraso en ejecutar el trabajo j, esto es, la demora si tuviese un valor positivo o cero en otro caso.
- Indicador de trabajo tardío U_j. Nos indica si el trabajo j se ha completado antes de su fecha límite d_j o no, es decir:
$$U_j = \begin{cases} 1, \, si \; C_j > d_j \\ 0, \, si \; C_j \leq d_j \end{cases}$$
Esta variable se utiliza para informarnos de cuántos trabajos han infringido sus fechas límites.
- El tiempo de permanencia $F_j = C_j - r_j$. Nos indica la cantidad de tiempo que transcurre desde que el trabajo j está disponible y se termina de procesar en el sistema.
- La anticipación $E_j = max\{0, -L_j = d_j - C_j\}$. Nos indica la cantidad de tiempo que el tiempo de completación C_j del trabajo j se ha anticipado a la fecha límite de dicho trabajo.
- La puntualidad $P_j = s_j - S_j$. Es la resta entre la fecha óptima de comienzo s_j del trabajo j y la fecha de comienzo S_j del trabajo j.
- Existen otro tipo de variables que nos proporcionan información sobre el estado del trabajo (si está siendo procesado en el instante actual, si se ha

terminado de procesar, si está esperando a ser procesado,...). Un ejemplo de variable es el siguiente:

$$\delta_j^p(t) = \begin{cases} 1, & \text{si el trabajo } j \text{ se esta procesando en el instante } t \\ 0, & \text{en otro caso} \end{cases}$$

Siendo t una variable en el intervalo $[0, C_{max}]$ con $C_{max} = \max_j\{C_j\} =$ "instante de finalización de todos los trabajos"

Normalmente se requiere minimizar las funciones objetivos. Las más estudiadas son las funciones objetivos de costo máximo

$$\gamma^k = f_{max} = \max_{j=1,\ldots,n}\{f_j(C_j)\}$$

donde $f_j(C_j) = C_j$ *o* L_j, esto es, se pretende minimizar el máximo tiempo de completación (*makespan*) o minimizar la máxima demora; o las funciones objetivos de costo total

$$\gamma^k = \sum f_j = \sum f_j(C_j)$$

donde $\sum f_j = \sum C_j, \sum T_j, \sum U_j, \sum w_j C_j, \sum w_j T_j, \sum w_j U_j$.

En ocasiones los criterios de planificación son equivalentes, es decir, la solución óptima para un criterio también es óptima para un criterio equivalente. Algunas de las equivalencias existentes son las siguientes:

- Como la suma de los tiempos de permanencia es
  $$\textstyle\sum_{j=1}^n F_j = \sum_{j=1}^n (C_j - r_j) = \sum_{j=1}^n C_j - \sum_{j=1}^n r_j$$
  y la suma de los tiempos de demora es
  $$\textstyle\sum_{j=1}^n L_j = \sum_{j=1}^n (C_j - d_j) = \sum_{j=1}^n C_j - \sum_{j=1}^n d_j$$
  si consideramos un problema de planificación donde las fechas de disponibilidad r_j y las fechas de vencimiento d_j son constantes para todos los trabajos obtenemos que minimizar $\sum C_j$ es equivalente a minimizar $\sum L_j$ *o* $\sum F_j$. Esto es, las funciones objetivos $\sum C_j, \sum L_j, \sum F_j$ son equivalentes bajo estas condiciones.

  Si además, estamos ante un problema estático, es decir, donde $r_j = 0\ \forall j$, entonces podemos apreciar que $F_j = C_j - r_j = C_j$ por lo tanto C_{max} y F_{max} son idénticos.
- Si nos encontramos con un problema de planificación donde todas las fechas de vencimiento son constantes e iguales $d_j = d$ $(j = 1, ..., n)$, obtenemos:
  $$L_{max} = \max_{j=1,\ldots,n}\{C_j - d_j\} = \max_{j=1,\ldots,n}\{C_j\} - d = C_{max} - d$$
  concluyendo que minimizar C_{max} es equivalente a minimizar L_{max}.
- Otra equivalencia se da en los problemas de planificación donde el objetivo es minimizar la demora máxima L_{max}, en cuyo caso también se minimizaría la tardanza máxima $T_{max} = max\{0, L_{max}\}$.
- En los problemas de planificación donde los tiempos de procesamiento son constantes (problemas determinísticos) se tiene que la suma del tiempo ocioso $\sum I_i$ de todas las máquinas y el máximo tiempo de completación C_{max} son

equivalentes. Esto sigue de considerar p_{ij} el tiempo de procesamiento del trabajo j en la máquina M_i. Entonces:

$$I_i = C_{max} - \sum_{j=1}^{n} p_{ij}$$

representa el tiempo ocioso en la máquina i. Ahora, operando:

$$\sum_{i=1}^{m} I_i = \sum_{i=1}^{m}(C_{max} - \sum_{j=1}^{n} p_{ij}) = mC_{max} - \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}$$

Donde finalmente observamos la equivalencia de dichas cantidades.

1.3. Aspectos relevantes y distintivos en los problemas de planificación estocástica

En Investigación Operativa, en concreto en los problemas de planificación sobre máquinas, a menudo surgen situaciones de incertidumbre en los datos del problema. Esto es debido, por ejemplo: a la oscilación de las condiciones del problema; a la variación que se presenta por fenómenos naturales; o por existir situaciones donde no se pueden medir con exactitud tales datos. Este tipo de problemas se denominan *problema de planificación estocásticos* y conllevan la utilización de un *modelo estocástico* que emplea variables aleatorias para enfrentar el estado de incertidumbre.

En muchos problemas de planificación la incertidumbre se puede omitir, convirtiéndose en un *problema de planificación determinístico*. En este caso se emplean estimaciones, medias o valores esperados de los datos del problema. Por lo tanto, partimos de un problema donde los datos son fijos y conocidos.

Como hemos mencionado, los *problemas de planificación estocástica* son problemas donde la incertidumbre es tratada con variables aleatorias. El uso de estas variables es esencial, en la mayoría de situaciones, para obtener un modelo más aceptable del problema y así resolverlo con más exactitud. Estas situaciones pueden ser, por ejemplo, problemas de planificación en los que las máquinas exigen mantenimiento, limpieza, reparación de posibles averías o discontinuidad en su funcionamiento. Otra posible situación viene de considerar un problema donde los tiempos de procesamiento de los trabajos sufren modificaciones conforme esperan a ser procesados.

Aún así, es posible considerar variables aleatorias en problemas determinísticos. Veámoslo a continuación.

1.3.1. Variables aleatorias en problemas determinísticos

En los problemas determinísticos ignoramos la incertidumbre y hacemos una estimación de los datos del problema, obteniendo valores fijos y conocidos. O simplemente nos encontramos con una situación donde no hay ninguna incertidumbre y podemos considerar datos fijos desde el principio. En ambas situaciones, los datos son conocidos y nos permiten representar un modelo fiable del problema.

Sin embargo, una vez hallada una solución óptima del modelo empleando un algoritmo eficiente se considera el hecho de transformar los datos iniciales (uno o varios) en variables para mostrar la eficiencia del algoritmo y ofrecer un resultado más realista al problema.

Cuando consideramos distribuciones de los datos de un problema determinístico aparecen dos nociones asociadas a esta circunstancia:

- *Experiencia computacional.* Nos muestra los resultados obtenidos con el algoritmo para distintas cantidades de los n datos de entrada. A la hora de analizar los resultados se presta más atención a los tiempos obtenidos por el algoritmo para ofrecer una solución a los distintos casos.
- *Análisis probabilístico.* Este análisis consiste en examinar los resultados obtenidos por el algoritmo para distintos casos y proporcionar la complejidad del algoritmo.

En conclusión, cuando consideramos variables en un problema de planificación determinística estamos mejorando la fase de interpretación de las respuestas, sin perder el método inicial de resolución del problema. En cambio, en un problema estocástico inicial, las variables aleatorias influyen en el modelo y en la construcción del algoritmo que pueda ofrecer una solución óptima al problema.

1.3.2. Problemas de planificación determinísticos y estocásticos

En un problema de planificación determinística a la hora de especificar los tres campos: máquinas, trabajos y criterios de optimalidad, se emplea la notación $\alpha|\beta|\gamma$ respectivamente.

Esta notación también es utilizada en los problemas de planificación estocástica. Por ejemplo, en un problema de planificación de n trabajos con una máquina donde los tiempos de procesamiento X_j son variables aleatorias independientes siguiendo una distribución exponencial de parámetro $\lambda_j (j = 1, ..., n)$ y cuyo objetivo es minimizar el valor esperado de C_{max} la notación correspondiente sería: $1|X_j \sim exp(\lambda_j)|E[C_{max}]$. Como vemos, en el campo β se detallan las distribuciones que siguen las variables aleatorias del problema ya que están ligadas a las características de los trabajos. Además, podemos observar que las variables aleatorias se representan con letras mayúsculas y sus correspondientes distribuciones con letras minúsculas. Debido a esto, los tiempos de procesamiento se designan X_j para evitar posibles confusiones con la probabilidad P.

Si las variables aleatorias siguen distribuciones arbitrarias, entonces se emplea la notación $X_j \sim G_j$ siendo G_j la función de distribución de la variable

$X_j (j = 1, ..., n)$. Notar que esta notación sugiere que las variables aleatorias $X_j (j = 1, ..., n)$ siguen la misma clase de distribución difiriendo en los parámetros puesto que están relacionados con los trabajos.

En el campo γ se incluyen los criterios de optimalidad, esto es, el valor esperado de la función objetivo que se pretende minimizar. La notación general empleada es $E[\gamma(C_1, C_2, ..., C_n)]$ donde $\gamma(C_1, C_2, ..., C_n)$ es la función objetivo, la cual depende de los tiempos de completación $C_1, C_2, ..., C_n$ de los trabajos.

A continuación, mostramos un problema en versión determinśtica y estocástica para apreciar las diferencias existentes entre las soluciones obtenidas.

Ejemplo **1.1** *Obtener una planificación óptima del problema* $1||\sum C_j$, *donde* n *trabajos deben ser procesados en una única máquina.*

Caso Determinístico. La solución óptima del problema sigue de ordenar los n trabajos en una secuencia que siga la regla SPT (*shortest processing time*), ordenar los trabajos en orden no decreciente de sus tiempos de procesamiento. Es decir, la secuencia óptima cumple que:

$$p_{[1]} \leq p_{[2]} \leq p_{[3]} \leq ... \leq p_{[n]}$$

donde $p_{[i]}$ denota el tiempo de procesamiento del trabajo en la i-ésima posición de procesamiento en la máquina.

Caso Estocástico. En este caso los tiempos de procesamiento son variables aleatorias con $G_j (j = 1, ..., n)$ distribuciones arbitrarias y $\mu_j (j = 1, ..., n)$ medias conocidas. Al tratarse de un problema estocástico el problema inicialmente planteado se convierte en el problema con parámetros $1|X_j \sim G_j|E[\sum C_j]$.

La solución óptima se consigue empleando la regla SEPT (*shortest expected processing time*), la cual ordena los n trabajos en orden no decreciente del valor esperado de sus tiempos de procesamiento. En otras palabras, la secuencia óptima verifica lo siguiente:

$$E[p_{[1]}] \leq E[p_{[2]}] \leq E[p_{[3]}] \leq ... \leq E[p_{[n]}]$$

siendo $E[p_{[i]}]$ el valor esperado del tiempo de procesamiento del trabajo en la posición i-ésima en la máquina.

Vamos a emplear ahora dichas soluciones óptimas en un caso real. Sean dos trabajo J_1 y J_2 con tiempos de procesamiento variables aleatorias independientes:

$$X_1 = \begin{pmatrix} 2 & 6 \\ \frac{3}{4} & \frac{1}{4} \end{pmatrix} \qquad X_2 = \begin{pmatrix} 1 & 7 \\ \frac{1}{3} & \frac{2}{3} \end{pmatrix}$$

entendiendo que la primera fila representa los valores que pueden tomar dichas variables aleatorias y la segunda sus respectivas probabilidades. Ahora, obtenemos que $E[X_1] = 2 * (\frac{3}{4}) + 6 * (\frac{1}{4}) = 3$ y $E[X_2] = 1 * (\frac{1}{3}) + 7 * (\frac{2}{3})) = 5$. Según estos resultado en el caso estocástico la secuencia óptima sería procesar primero el trabajo J_1 y despues el trabajo J_2. Dado que las variables aleatorias son independientes, la probabilidad de que $X_1 = 2$ y $X_2 = 1$ es $P(X_1 = 2, X_2 = 1) = \frac{3}{4} * \frac{1}{3} = \frac{1}{4}$ y en este caso $\sum_{j=1}^{2} C_j = 5$. Sin embargo, si estuviesemos en el caso determinístico la secuencia óptima dada por la regla SPT consiste en procesar primero el trabajo J_2 con tiempo de procesamiento 1 y después el trabajo J_1 con tiempo de procesamiento 2, obteniendo que $\sum_{j=1}^{2} C_j = 4$.

Como vemos, la solución óptima al caso estocástico no es la solución más óptima posible, puesto que pueden existir otras secuencias que mejoren los resultados obtenidos. Esto es debido a que el caso estocástico busca una secuencia óptima en media. En cambio, en el caso determinístico se busca la secuencia óptima entre todos los casos posibles.

A continuación, explicamos algunas diferencias entre el caso determinístico y estocástico a tener en cuenta con respecto a los trabajos tardíos y a la dificultad de resolución.

En el caso determinístico un trabajo j es tardío si $C_j > d_j$, por lo tanto no puede ocurrir que un trabajo j sea tardío y, al mismo tiempo, este completado antes de su fecha límite. En contraste, en el caso estocástico un trabajo j puede cumplir que sea tardío $P(C_j > d_j) > 0$, y a la vez ser completado en su margen de tiempo $P(C_j \leq d_j) > 0$.

Además, en los problemas estocásticos puede ocurrir que la solución óptima consista en un resultado donde todos los trabajos del sistema cumplan que $P(C_j > d_j) > 0$, por ejemplo, en algunos problemas de planificación estocástica donde los tiempos de procesamiento X_j sean variables aleatorias cuyas funciones de distribución G_j estén definidas en el intervalo $[0, +\infty)$ y las fechas límites d_j sean valores conocidos y fijos. En consecuencia, se han definido dos nuevos conceptos relativos al número de trabajos tardíos: el número esperado de trabajos tardíos y el número de trabajos β-tardíos. Un trabajo es β-tardío si $P(C_j > d_j) > \beta$, siendo β una probabilidad permitida de tardanza en el problema.

A primera vista se puede pensar que un problema estocástico es más difícil de resolver debido a la variabilidad de los datos de entrada. Sin embargo, esto no es siempre cierto. Se puede dar el caso de que un problema en versión determinística y estocástica tengan la misma dificultad de resolución. O, incluso, que la versión estocástica sea fácil y la versión determinística sea *NP*-duro.

Además, existen varios motivos por los que un problema estocástico es más fácil de resolver que su versión determinística. Algunos de estos motivos son los siguientes:

- La distribución exponencial posee una propiedad denominada *propiedad de falta de memoria*. Si X es una variables aleatoria con función de distribución exponencial entonces se verifica
$$P(X \geq t + h / X \geq t) = P(X \geq h) \ \ \forall t, h > 0$$

En otras palabras, supongamos que el tiempo de procesamiento X de un trabajo está distribuido como una exponencial, entonces si el trabajo es interrumpido, el trabajo que falta por procesar también tendrá un tiempo de procesamiento X.

- En ocasiones es más fácil minimizar una función del tipo $E[\gamma(C_1, C_2, ..., C_n)]$ donde la función $\gamma(C_1, C_2, ..., C_n)$ contiene variables aleatorias, que del tipo $\gamma(C_1, C_2, ..., C_n)$ donde son conocidos todos los datos de entrada. Un ejemplo sigue de considerar el problema estocástico $P2|X_j \sim exp(\lambda_j)|E[C_{max}]$ donde el algoritmo de resolución finaliza en un tiempo $\mathcal{O}(n \text{ log } n)$ y su versión determinística $P2||C_{max}$ es NP-duro.
- La versión determinística de un problema además de poseer más información exacta debe proporcionar una solución óptima. En cambio, la versión estocástica debe proporcionar una solución óptima en media, pudiendo darse casos que mejoren la solución facilitada, como vemos en el *ejemplo* **1.1** .

2

Conceptos generales sobre planificación estocástica

En ocasiones en un problema de planificación puede observarse cierta incertidumbre en los datos de entrada. Es entonces cuando hablamos de problemas de planificación estocástica. En estos problemas no podemos obviar el hecho de que uno o varios datos no son fijos, por lo tanto, es esencial tratarlos como variables aleatorias para representar de forma fiel el problema real que estamos modelizando.

En este capítulo trataremos aspectos concretos relativos a variables aleatorias como ordenación estocástica, la importancia de la propiedad de pérdida de memoria, estrategias de toma de decisiones y métodos de resolución. Para esta descripción hemos tomado como base la publicación de Rodríguez-González (1999) [5].

2.1. Introducción a las distribuciones

Como hemos mencionado, los problemas de planificación estocástica poseen datos los cuales son variables aleatorias que tienen asociada una función de distribución. Por consiguiente, entender los conceptos básicos sobre distribuciones y propiedades útiles en dichos problemas nos facilitará la resolución de los mismos.

Definición **2.1** *La función de distribución* $F_X : \mathbb{R} \longrightarrow \mathbb{R}$ *se define como*

$$F_X(x) = P(X \leq x).$$

Es decir, a cada valor real x *le asigna la probabilidad de que una variable aleatoria* X *sea inferior o igual a* x.

A continuación, definimos variable aleatoria discreta y continua, y sus respectivas esperanzas o valores esperados.

Definición **2.2** *Una variables aleatoria* X *es discreta si sólo puede tomar valores* $x_1, x_2, ..., x_n$ *en un conjunto* D *finito o numerable.*

La función de probabilidad se define como:

$$P(X=x)=\begin{cases} p(x_i)=p_i & si\ x=x_i \\ 0 & si\ x\notin D \end{cases}$$

Además, se cumple que

$$\sum_{x_i\in D} p(x_i)=1.$$

La función de distribución en este caso, verifica lo siguiente

$$F(x)=P(X\leq x)=\sum_{x_i\leq x} p_i.$$

Definición **2.3** *La esperanza o valor esperado de una variable aleatoria discreta X consiste en el valor siguiente*

$$E[x]=\sum_i x_i p_i$$

siendo p_i las probabilidades representadas por la función de probabilidad.

Definición **2.4** *Una variable aleatoria X es continua si existe una función no negativa f(x) (denominada función de densidad) de forma que*

$$F(x)=\int_{-\infty}^{x} f(t)dt$$

donde $F(x)$ es la función de distribución de X.

Definición **2.5** *La esperanza o valor esperado de una variable aleatoria continua X es*

$$E[X]=\int_{-\infty}^{+\infty} xf(x)dx$$

A continuación, definimos la función de supervivencia y la tasa de fallos, conceptos útiles en los problemas de planificación estocásticos.

La tasa de fallos es un concepto relacionado con la fiabilidad del funcionamiento del sistema. La fiabilidad es la probabilidad de que el sistema funcione correctamente durante un tiempo determinado. En este caso, la variable aleatoria X representa el tiempo de duración o vida del dispositivo.

La *función de fiabilidad* o *función de supervivencia* se define como

$$\overline{F}(t)=P(X>t)=1-F(t)$$

y representa la probabilidad de que un dispositivo viva más del tiempo t. Por lo tanto, la función de distribución representa la probabilidad de que un dispositivo viva a lo sumo un tiempo t.

Definición **2.6** *Sea X una variable aleatoria no negativa con función de distribución F absolutamente continua. Se define la tasa de fallos de X o función de riesgo como*

$$r(t)=\frac{\partial}{\partial t}(-log(1-F(t))),\ t\geq 0$$

Otra forma de definir la función tasa de fallos es

$$r(t)=\lim_{\Delta t\to 0}\frac{P(t<X\leq t+\Delta t/X>t)}{\Delta t}=\frac{f(t)}{\overline{F}(t)},\ t\geq 0$$

La tasa de fallos se emplea en análisis de fiabilidad puesto que nos proporciona una aproximación de la probabilidad de que el sistema deje de funcionar en el intervalo $(t, t + \Delta t)$ sabiendo que ha funcionado correctamente hasta el instante t, sin embargo, hay que tener en cuenta que esta interpretación se da solo cuando t y Δt toman valores suficientemente pequeños. Se concluye que

$$P(X \leq t + \Delta t / X > t) = \frac{P(t < X \leq t + \Delta t)}{P(X > t)} = \frac{\int_t^{t+\Delta t} f(u)du}{\overline{F}(t)} \overset{\Delta t \to 0}{\approx} \frac{f(t)}{\overline{F}(t)} \Delta t = r(t)\Delta t$$

Definición **2.7** *Sea X una variable aleatoria discreta. Se define la tasa de fallos de X o función de riesgo como*

$$r(t) = \frac{P(X=t)}{P(X \geq t)},\ t \in \{0, 1, ...,\}$$

A continuación detallaremos una serie de propiedades de la variable aleatoria X relacionadas con la tasa de fallos, la función de supervivencia y la función de densidad:

- *Propiedad IFR (increasing failure rate).* Sea X una variable aleatoria no negativa. La tasa de fallos de X se dice que es *creciente* si $r(t)$ es creciente en t. En otras palabras, que la tasa de fallos sea creciente significa que la probabilidad de que el sistema deje de funcionar aumenta con el transcurso del tiempo. Un ejemplo de esta propiedad consistiría en un sistema donde los recursos se pueden agotar.
- *Propiedad DFR (decreasing failure rate).* Sea X una variable aleatoria no negativa. La tasa de fallos de X se dice que es *decreciente* si $r(t)$ es decreciente en t. Es decir, que la tasa de fallos sea decreciente es equivalente a que la probabilidad de que el sistema falle disminuye con el transcurso del tiempo. Un ejemplo de esta segunda propiedad podría verse en una empresa donde la variable aleatoria X represente su tiempo de vida o funcionamiento.
- *Propiedad NBU (new better than used).* Sea X una variable aleatoria no negativa. Se dice que X es NBU si $-log\,\overline{F}$ es superaditiva en el intervalo $(0, +\infty)$. Dicho de otra forma, si $\overline{F}(s+t) \leq \overline{F}(s)\overline{F}(t)\ \forall s, t \geq 0$.
- *Propiedad NWU (mew worse than used).* Sea X una variable aleatoria no negativa. Se dice que X es NWU si $-log\,\overline{F}$ es subaditiva en el intervalo $(0, +\infty)$. Dicho de otra forma, si $\overline{F}(s+t) \geq \overline{F}(s)\overline{F}(t)\ \forall s, t \geq 0$.
- *Propiedad ILR(increasing likelihood rate).* Sea X una variable aleatoria y $f(x)$ su función de densidad. Se dice que X es ILR si $logf(x)$ es cóncava.
- *Propiedad DLR(decreasing likelihood rate).* Sea X una variable aleatoria y $f(x)$ su función de densidad. Se dice que X es DLR si $logf(x)$ es convexa.

Hasta ahora hemos hablado de tiempo de duración o vida de un sistema. No obstante, cuando el tiempo de procesamiento de algún trabajo del sistema es una variable aleatoria que sigue una función de distribución hablamos de *vida residual* del sistema. La vida residual de un sistema nos informa de si un trabajo con las condiciones anteriores se ha completado o cuánto falta para que termine de procesarse en el sistema.

Definición **2.8** *Sea X una variable aleatoria positiva que representa el tiempo de vida o duración del sistema. Se define la vida residual del sistema en el momento* *t como la variable aleatoria* $X_t = (X - t/X > t)$.

La vida residual X_t *representa el tiempo de vida del sistema teniendo en cuenta que ha funcionado correctamente hasta el momento* t.

La variable aleatoria vida residual X_t *sigue una función de distribución:*

$$F_{X_t}(x) = P(X - t \leq x/X > t) = \frac{P(t<X\leq x+t)}{P(X>t)} = \frac{F_X(x+t)-F_X(t)}{1-F_X(t)}$$

La vida residual X_t es una variable aleatoria definida en el momento t y que depende de la variablea aleatoria X tiempo de vida del sistema. En consecuencia, es posible comparar las vidas residuales de la variable X con respecto al tiempo t. La vida residual nos permitirá saber si dado un trabajo cuyo tiempo de procesamiento es una variable aleatoria y es interrumpido mientras se procesa, cuánto falta para terminar de ser procesado puesto que la vida residual equivale al tiempo de procesamiento restante al ser interrumpido.

2.2. Ordenación estocástica

En los problemas de planificación estocástica a menudo es necesario comparar variables aleatorias independientes entre sí. Para abordar este asunto se emplean diferentes órdenes estocásticos basados en propiedades conocidas.

2.2.1. Comparación de esperanzas

Una comparación muy útil en problemas de planificación estocásticos entre dos variables aleatorias es la comparación de sus esperanzas o valores esperados.

Dadas dos variables aleatorias X e Y, se dice que X es *menor en esperanza* que Y si

$$E[X] \leq E[Y]$$

y se denota $X \leq_\mu Y$.

Este orden es muy eficaz puesto que en muchas ocasiones es necesario minimizar la esperanza de una o más funciones objetivos de los tiempos de completación de los trabajos. Por lo tanto, el problema precisa un método para comparar esperanzas.

2.2.2. Orden casi seguro

Cuando una probabilidad es *casi seguro* uno entendemos que el suceso puede no suceder pero es casi seguro que ocurra.

Ejemplo **2.1** *Un ejemplo sencillo para ilustrar una probabilidad casi seguro con valor uno sigue de suponer que alguien tira a diana con los ojos cerrados y queremos saber cuál es la probabilidad de que no haga pleno. Obviamente existe una posibilidad muy pequeña de que acierte, sin embargo, lo más probable es que falle por lo que estamos ante una probabilidad casi seguro con valor uno.*

Esta propiedad se emplea para comparar dos variables aleatorias, donde el suceso es precisamente el orden entre ellas.

Dadas dos variables aleatorias X e Y, se dice que X es *casi seguro* menor o igual a Y si

$$P(X \leq Y) = 1$$

y se denota $X \leq_{as} Y$.

2.2.3. Orden estocástico usual

Dadas dos variables aleatorias X e Y, se dice que X es *estocásticamente menor* que Y si

$$P(X > t) \leq P(Y > t),\ \forall t \in (-\infty, \infty)$$

es decir, si

$$\overline{F}_X(t) \leq \overline{F}_Y(t),\ \forall t \in (-\infty, +\infty)$$

y se denota $X \leq_{st} Y$.

El orden estocástico usual implica orden en media. Es decir, dadas dos variables aleatorias X e Y que verifican $X \leq_{st} Y$, entonces se cumple $E[X] \leq E[Y]$.

Ahora exponemos un teorema que relaciona el orden estocástico entre las vidas residuales de una variable con propiedad *IFR, DFR, NBU o NWU.*

Teorema 2.1. *Sea X una variable aleatoria. Entonces se cumplen los siguientes casos:*

1. *X es IFR $\Leftrightarrow (X - t/X > t) \geq_{st} (X - t'/x > t')$, si $t \leq t'$.*
2. *X es DFR $\Leftrightarrow (X - t/X > t) \leq_{st} (X - t'/x > t')$, si $t \leq t'$.*
3. *X es NBU $\Leftrightarrow X \geq_{st} (X - t/x > t)$, si $t > 0$ (con X no negativa).*
4. *X es NWU $\Leftrightarrow X \leq_{st} (X - t/x > t)$, si $t > 0$ (con X no negativa).*

Si el tiempo de procesamiento de un trabajo está definido como una variable aleatoria X y se decide interrumpir el trabajo en un momento t, el tiempo necesario para procesar el trabajo restante es equivalente a la vida residual de X en el momento t, es decir, es equivalente a X_t. De manera que el tiempo restante para completar el trabajo desde el momento t en el que se interrumpió es $(X - t/X > t)$. Si X verifica alguna de las propiedades *IFR, DFR* entonces se puede hacer una comparación estocástica usual entre sus vidas residuales con respecto al momento t de interrupción. Si X verifica alguna de las propiedades *NBU, NWU* entonces se puede hacer una comparación estocástica usual entre la variable aleatoria X y el conjunto de sus vidas residuales.

2.2.4. Orden en tasa de fallos

Sean X e Y variables aleatorias no negativas cuyas funciones de distribución son absolutamente continuas. Además, r y q son las funciones de tasa de fallos de X e Y respectivamente. Se dice que X es *menor en tasa de fallos* que Y si

$$r(t) \geq q(t),\ t \geq 0$$

y se denota $X \leq_{hr} Y$.

Otra forma de definir el orden en tasa de fallos es X es *menor en tasa de fallos* que Y si

$$\frac{\overline{F}(t)}{\overline{G}(t)} \text{ es decreciente en } t \in \mathbb{R}$$

siendo $\overline{F}(t)$ la función de supervivencia de X y $\overline{G}(t)$ la función de supervivencia de Y.

Como propiedades interesantes relativas al orden en tasa de fallos podemos destacar dos. El orden en tasa de fallos implica orden estocástico usual y, el orden en tasa de fallos implica una ordenación estocástica usual entre las vidas residuales de las variables. Entendiendo que las ordenaciones usuales siguen el mismo "sentido" de orden que la que se sigue en la ordenación en tasa de fallos.

El orden en tasa de fallos puede emplearse para definir la propiedad que verifica una variable aleatoria, ya sea la propiedad *IFR o DFR.*

Teorema 2.2. *1. La variable aleatoria X es IFR si, y solo si, una de las siguientes condiciones equivalentes se cumplen:*
   a) $(X - t/X > t) \geq_{hr} (X - t'/X > t')$, *si* $t \leq t'$.
   b) $X \geq_{hr} (X - t/X > t)$, $\forall t \geq 0$, *siendo X no negativa.*
   c) $X + t \leq_{hr} X + t'$, *si* $t \leq t'$.
2. La variable aleatoria X es DFR si, y solo si, una de las siguientes condiciones equivalentes se verifica:
   a) $(X - t/X > t) \leq_{hr} (X - t'/X > t')$, *si* $t \leq t'$.
   b) $X \leq_{hr} (X - t/X > t)$ $\forall t \geq 0$, *siendo X no negativa.*

2.2.5. Orden en razón de verosimilitud

Sean X e Y variables aleatorias continuas con funciones de densidad f y g respectivamente. Se dice que X es *menor en razón de verosimilitud* que Y si

$$\frac{f(t)}{g(t)} \text{ es decreciente } \forall\ t \in \{R_X, R_Y\}.$$

y se denota $X \leq_{lr} Y$.

Si X e Y son variables aleatorias discretas, se dice que X es *menor en razón de verosimilitud* que Y si

$$\frac{P(X=t)}{P(Y=t)} \text{ es decreciente } \forall\ t \in \{R_X, R_Y\}$$

Una propiedad muy importante relativa al orden en razón de verosimilitud es la siguiente. El orden en razón de verosimilitud implica orden en tasa de fallos.

A continuación, exponemos un teorema que expone condiciones basadas en comparaciones en orden de razón de verosimilitud para caracterizar que X posee la propiedad *IFR.*

Teorema 2.3. *La variable aleatoria X es ILR si, y solo si una de las siguientes condiciones equivalentes se verifica:*
1. $(X - t/X > t) \geq_{lr} (X - t'/X > t')$, *si* $t \leq t'$.
2. $X \geq_{lr} (X - t/X > t)$, *con* $t \geq 0$ *y* X *no negativa.*
3. $X + t \leq_{lr} X + t'$ *si* $t \leq t'$.

2.2.6. Órdenes convexo y monótono convexo

En primer lugar, definimos un orden que compara dos variables aleatorias empleando funciones reales convexas.

Dadas dos variables aleatorias X e Y, se dice que X es *menor en orden convexo* que Y si

$$E[\phi(X)] \leq E[\phi(Y)] \text{ con } \phi : \mathbb{R} \longrightarrow \mathbb{R} \text{ función convexa}$$

y se denota $X \leq_{cx} Y$.

Si en la definición anterior sustituimos la función convexa ϕ por una función ϕ' cóncava nos encontramos ante el orden cóncavo.

La propiedad característica de este orden es que el orden convexo implica la igualdad entre las medias de X e Y.

En segundo lugar, definimos otro orden que relaciona dos variables aleatorias aplicando funciones reales crecientes y convexas.

Dadas dos variables aleatorias X e Y, se dice que X es *menor en orden convexo creciente* que Y si

$$E[\phi(X)] \leq E[\phi(Y))] \text{ con } \phi : \mathbb{R} \longrightarrow \mathbb{R} \text{ función creciente y convexa}$$

y se denota $X \leq_{icx} Y$.

Si sustituimos la función ϕ creciente y convexa por la función ϕ' creciente y cóncava conseguimos el orden monótono cóncavo.

La propiedad fundamental de este orden es la siguiente. Si X e Y se pueden ordenar mediante un orden monótono convexo creciente de tal forma que $X \leq_{icx} Y$ entonces se verifica que $E[X] \leq E[Y]$. En otras palabras, el orden monótono convexo implica orden en media.

2.3. Características destacables de los problemas de planificación estocástica

2.3.1. Distribución exponencial

En los problemas de planificación estocásticos, en muchas ocasiones, es útil emplear la distribución exponencial debido a que las circunstancias de incertidumbre siguen dicha distribución y, además, resulta más sencillo trabajar con ella puesto que posee la propiedad de ausencia de memoria.

Como sabemos, si X es una variable aleatoria continua que sigue una distribución exponencial de parámetro λ la función de distribución es:

$$F_X(x) = \begin{cases} 0, & si\ x < 0 \\ 1 - e^{-\lambda x} & si\ x \geq 0 \end{cases}$$

con función de densidad

$$f(x) = \begin{cases} 0, & si\ x < 0 \\ \lambda e^{-\lambda x}, & si\ x \geq 0 \end{cases}$$

y con valor esperado $E[X] = \frac{1}{\lambda}$ y varianza $V(X) = \frac{1}{\lambda^2}$.

La distribución exponencial representa la distribución del tiempo entre dos sucesos consecutivos que siguen una distribución de Poisson. Recordemos que la distribución de Poisson se emplea para calcular la probabilidad de que se den una cierta cantidad k de sucesos con probabilidad pequeña.

La distribución más importante que se da con la exponencial es la distribución del tiempo necesario para finalizar un trabajo interrumpido. Como ya mencionamos en el capítulo anterior, la exponencial posee la propiedad de ausencia de memoria la cual es muy útil para este tipo de circunstancias. Asimismo, se emplea para distribuir el tiempo que tarda en fallar una máquina, en cuyo caso se verifica que dicho tiempo no depende del tiempo que la máquina ha estado en funcionamiento, es decir, trata con máquinas cuyo envejecimiento no afecta a su tiempo de vida.

Recordemos que la propiedad de ausencia de memoria se resume en: dada una variable aleatoria X que representa el tiempo de vida de un sistema, la probabilidad de que dicho sistema funcione correctamente al menos un tiempo $t+s$ si ha funcionado durante un tiempo s es igual a la probabilidad de que dicho sistema funcione al menos un tiempo t.

$$P(X \geq t + s / X \geq s) = P(X \geq t)$$

En otras palabras, la probabilidad de que el sistema funcione correctamente al menos un tiempo t no es influenciado por el hecho de haber funcionado correctamente un tiempo s. La vida del sistema no depende del tiempo en funcionamiento o del envejecimiento del mismo.

A la hora de emplear esta propiedad de ausencia de memoria en la interrupción de trabajos se sigue de la siguiente forma. Sea X_j el tiempo de procesamiento de un trabajo j que sigue una distribución exponencial, y supongamos que dicho trabajo comienza en un tiempo fijo S_j y se interrumpe en un instante h, entonces se cumple que el tiempo X'_j que falta para finalizar el trabajo sigue la misma distribución que el tiempo inicial X_j:

$$P(X'_j > t) = P(X_j - (h - S_j) > t / X_j + S_j > h) =$$
$$= P(X_j > t + (h - S_j) / X_j > (h - S_j)) = P(X_j > t)$$

2.3.2. Políticas de toma de decisiones

En este apartados nos centraremos en definir las cuatro estrategias posibles de toma de decisiones en los problemas estocásticos. Estas políticas dependerán de si podemos obtener la información relativa al procesamiento de los trabajos en tiempo real o, en cambio, debemos realizar una lista de trabajos desde el comienzo debido a nuestra ignorancia con respecto al procesamiento.

- *Política estática de lista sin interrupciones.* Está pensada para los problemas estáticos donde las máquinas pueden ser: una, máquinas idénticas y sistemas flow shop. En esta estrategia de actuación los trabajos son ordenados en una lista desde el instante inicial teniendo en cuenta las características de cada uno, de forma que en el sistema el trabajo a ser procesado en cada instante es el que encabeza dicha lista. Durante el procesamiento de los trabajos el orden considerado en la lista no se puede cambiar.
- *Política estática de lista con interrupciones.* Esta clase de política considera trabajos con diferentes tiempos de disponibilidad que pueden ser interrumpidos. Ordena los trabajos en una lista desde el instante inicial teniendo en cuenta sus características. Dicha lista no se puede cambiar una vez se comiencen a procesar los trabajos.

  En esta política los trabajos son ordenados desde el principio en una lista pero siempre que el trabajo que encabece la lista no esté disponible se elegirá el siguiente hasta llegar a uno que si esté disponible. Si se llega a la fecha de disponibilidad de un trabajo que no pudo empezar a ser procesado cuando encabezaba la lista entonces se interrumpe el trabajo actual y se comienza a procesar éste.
- *Política dinámica sin interrupciones.* En esta estrategia se eligen los trabajos a ser procesados a lo largo del tiempo, sin embargo, no se pueden interrumpir los trabajos. El siguiente trabajo a ser procesado se elegirá teniendo en cuenta la información de la que se dispone en cada momento.
- *Política dinámica con interrupciones.* Al igual que en la política anterior, se puede elegir el trabajo siguiente a ser procesado en el momento que se precise y, además, se pueden interrumpir trabajos si así fuera necesario.

2.4. Algunos métodos de resolución de problemas estocásticos

En esta sección exponemos algunos métodos de resolución de problemas de planificación estocástica pertenecientes a otras ramas de la Investigación Operativa. Estos son los siguientes:

- *Teoría de Colas.* La teoría de colas se encarga del estudio de las colas en un sistema afrontando objetivos como medir la capacidad del sistema para evitar colapsos o fallos. Por lo tanto, si consideramos los problemas estocásticos como un problema de colas donde los trabajos juegan el papel de la cola y se debe asignar a una máquina que lo pueda procesar entonces podemos aplicar técnicas relativas a esta ciencia.

  Sin embargo, cuando el objetivo es minimizar la esperanza de una función de costo total y los tiempos de procesamiento de los trabajos se distribuyen con una distribución no estudiada en Teoría de Colas entonces es imposible proporcionar una solución óptima al problema inicial.

- *Índices dinámicos.* Es un método donde se le asigna a cada trabajo del sistema un índice que representa la prioridad que tiene cada trabajo en la planificación y posteriormente se procesan en orden decreciente de índices. Es un método dinámico ya que asignamos índices a los trabajos al inicio pero podemos hacer cambios pertinentes para cumplir los requisitos de los trabajos.
- *Simulación.* El problema estocástico es representado virtualmente con una serie de valores numéricos que sustituyen la incertidumbre del problema. Para estudiar los diferentes casos y hallar la mejor solución posible se emplean diferentes valores que pertenecen al conjunto de valores posibles de las variables aleatorias del problema.

3

Clasificación computacional

Cuando intentamos resolver un problema mediante un algoritmo computacional nos surgen de entrada varias preguntas: ¿cuál es la dificultad de dicho problema?, ¿es posible encontrar un algoritmo que lo resuelva?, ¿será dicho algoritmo de alguna manera eficiente?.

Podemos entender que un algoritmo es eficiente cuando al ser ejecutado en una máquina los recursos que utiliza son considerablemente pequeños. Entendemos por recursos:

- Tiempo: cantidad de pasos de ejecución dados por el algoritmo para resolver un problema.
- Espacio: cantidad de memoria empleada por la máquina donde se ejecuta el algoritmo para resolver el problema.

El recurso espacio se denota con $S(n)$ y depende del tamaño n de los datos de entrada. Representa la cantidad de memoria que necesita la máquina para ejecutar el algoritmo. En otras palabras, la cantidad de datos informáticos que debe guardar la máquina mientras procesa el algoritmo.

La necesidad de considerar el recurso tiempo en función de los pasos realizados por el algoritmo en lugar del tiempo real de ejecución es debido a que un mismo algoritmo en computadoras distintas puede obtener un tiempo de ejecución diferente. Por lo tanto, no se puede considerar el tiempo real de ejecución. El tiempo necesario para resolver un problema utilizando un algoritmo depende de los datos de entrada. Por esta razón, el tiempo $T(n)$ puede ser expresado en función del tamaño n de los datos de entrada.

En computación se emplea la $\mathcal{O}-$grande para definir una cota superior de $T(n)$. Se dice que $T(n)$ es $\mathcal{O}(f(n))$, con $f(n)$ función que depende de los n datos de entrada, si $\exists c, n_0 \in \mathbb{N} : T(n) \leq cf(n) \; \forall n \geq n_0$. Intuitivamente lo que representa la $\mathcal{O}-$grande es, de forma abreviada, la rapidez con la que tiende a infinito el tiempo $T(n)$ al considerar infinitos datos de entrada.

Ejemplo **3.1** *Consideramos dos algoritmos A_1 y A_2. El tiempo $T_1(n)$ es $n^3 + 24n^2$. El tiempo $T_2(n)$ es $n^5 + n$. Si metemos un número infinito de datos n en el primer algoritmo A_1 entonces el tiempo $T_1(n)$ crecerá con una velocidad de n^3, sin embargo, en el algoritmo A_2 el tiempo $T_2(n)$ crecerá con una velocidad de n^5. Entonces se dice que el tiempo $T_1(n)$ tiene un orden de complejidad $\mathcal{O}(n^3)$, y el tiempo $T_2(n)$ tiene un orden de complejidad $\mathcal{O}(n^5)$.*

Al comparar los tiempos de ejecución de los algoritmos se comparan sus $\mathcal{O}(f(n))$ sin tener en cuenta las constantes de proporcionalidad de $f(n)$. Así, si consideramos un algoritmo $\mathcal{O}(n^2)$ y otro algoritmo $\mathcal{O}(n^3)$ decimos que el primer algoritmo es más rápido que el segundo, puesto que la cantidad de pasos del segundo a medida que aumentamos los datos de entrada se dispara con una velocidad de n^3, en cambio, en el primero a medida que aumentamos los datos la cantidad de pasos a realizar aumenta con una velocidad de n^2.

Calcular el tiempo de ejecución de un algoritmo se resume en dar una cota de $T(n)$, es decir, en proporcionar el valor de $\mathcal{O}-$grande de $T(n)$. En concreto, se dice que un algoritmo es de *tiempo polinomial* si $\mathcal{O}(p(n))$ con $p(n)$ función polinómica . En este caso se afirma que el problema está bien resuelto, obteniendo un algoritmo muy eficiente. En caso contrario, el algoritmo es de *tiempo exponencial* y se considera que es muy ineficiente.

Un problema es una cuestión que debemos responder. El problema es planteado mediante unos datos de entrada los cuales se relacionan para formular la cuestión. Por otro lado, el método de resolución del problema es mediante un algoritmo o procedimiento que nos permite responder, empleando transformaciones en los datos de entrada, a la cuestión correctamente.

Todo problema de optimización tiene asociado un problema de decisión y otro problema de reconocimiento del lenguaje. Para estudiar la complejidad de un problema de optimización se procede así: si encontramos un algoritmo que resuelva sus problemas asociados entonces el problema estará resuelto.

Los problemas en Complejidad Computacional se pueden clasificar como:

- *Clase P*. Problemas cuyos problemas de reconocimiento del lenguaje asociados se pueden resolver por una máquina de Turing determinista en tiempo polinomial. Para los problemas de clase *P* existe un algoritmo que los resuelva de forma rápida.
- *Clase NP*. Problemas cuyos problemas de reconocimiento del lenguaje asociados se pueden resolver por una máquina de Turing no determinista en tiempo polinomial. Para resolver los problemas de clase *NP* se emplean algoritmos heurísticos para hallar una solución aproximada del problema. Dichas soluciones se comprueban a un ritmo polinomial.
- *Clase NP-Duro*. Un problema es *NP*-duro si el problema de reconocimiento de su lenguaje asociado es *NP*-duro. El problema de reconocimiento del lenguaje es *NP*-duro si todos los problemas de reconocimiento del lenguaje de la clase *NP* se reducen polinomialmente a él.

- *NP-Completos.* Conjunto de problemas que son a la vez de la clase *NP* y *NP*-duros.

Es una pregunta abierta a día de hoy si la clase *P* es igual a la clase *NP*. Los problemas de la clase *P* se pueden resolver en un tiempo polinomial, por tanto, dada una solución comprobar si verifica las condiciones del problema se puede realizar de manera rápida, con lo cual $P \subset NP$. Sin embargo, los problemas de la clase *NP* (en los cuales se puede verificar rápidamente si un resultado es una solución del problema) se desconoce si se pueden resolver fácilmente. Si se pudiesen resolver en tiempo polinomial entonces se daría la igualdad $P = NP$.

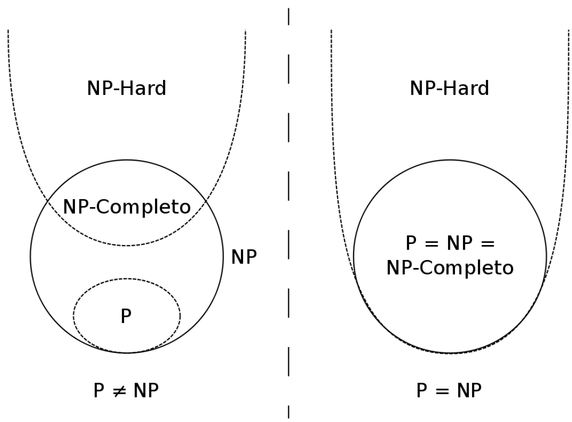

Figura 3.1: Distinción $P \subset NP$ y $P = NP$

4

Aplicación a un problema de ordenación secuencial estocástica jerárquico

En esta sección vamos a plantear la resolución de un *problema de ordenación secuencial estocástico jerárquico* (*HSSOP*) para posteriormente exponer los resultados de una situación concreta. Este problema fue investigado por Alcaide et al. (2003) y originó una publicación científica en la revista Omega , the International Journal of Management Science [3].

4.1. Definición y formulación del problema

Primero consideramos el *problema de ordenación secuencial* (*SOP*) definido como sigue: dado un grafo $G = (V, A)$ donde V es el conjunto de vértices con costo p_j para cada vértice $j \in V$ y A es el conjunto de arcos con costo $c_{i,j}$ para cada arco $(i, j) \in A$. Este problema estará resuelto si encontramos un camino hamiltoniano S en G de tal forma que minimice el costo total. Es decir, el camino S debe verificar

$$min_s\{\sum_{j\in V} p_j + \sum_{(i,j)\in S} c_{i,j}\}. \tag{4.1}$$

También se pueden añadir condiciones de relaciones de precedencia entre los vértices y cotas a la suma de costos de subcaminos en S.

Un caso particular de este problema es el famoso *problema del viajante de comercio* donde cada vértice es una ciudad con costo el tiempo que se tarda en recorrerla y los arcos representan el camino que une las ciudades con costo el tiempo que se tarda en llegar de una a otra.

Nosotros nos centraremos en el problema *SOP* aplicado a los problemas de planificación. En estos problemas, los vértices representan los trabajos y los arcos representan la secuencia de trabajos. El costo c_j representa el tiempo de procesamiento del trabajo j, y el costo c_{ij} representa el time-setup desde el trabajo i al trabajo j. Existe un arco $(i,j) \in A$ si es posible empezar a procesar el trabajo j justo despues del trabajo i. En los problemas SOP pueden suceder varias circunstancias que obliguen a considerar los tiempos de procesamiento de los trabajos como variables aleatorias. Esto podría deberse a situaciones que afecten a las máquinas (averías, falta de materia prima,...) o a trabajadores (falta de preparación, falta de concentración en el trabajo,...).

El problema a estudiar se formula como sigue: se desean procesar n trabajos en una única máquina de forma que se cumpla la factibilidad y se minimice el makespan. Los tiempos de procesamiento son variables aleatorias y los tiempos de disponibilidad son diferentes. Las fechas de vencimiento son también distintas, por lo que cada trabajo dispone de una ventana temporal en la que debe ser procesado. Las interrupciones no están permitidas. Además, los trabajos deben satisfacer unas relaciones de precedencia entre ellos. Al tratarse de un problema estocástico la factibilidad se afronta hallando la secuencia que maximice la probabilidad de que sea factible, y la minimización del makespan con la minimización del valor esperado del makespan.

La solución final consistirá en una ordenación secuencial de los n trabajos puesto que disponemos de una máquina y no se permiten las interrupciones de los trabajos. Por lo tanto, se proporcionará una secuencia donde los trabajos están ordenados de tal forma que la posición i-ésima en la secuencia equivale al puesto i-ésimo a procesar en la máquina, teniendo en cuenta que un trabajo no puede empezar a ser procesado hasta que el trabajo anterior se haya terminado de procesar. En esta secuencia los trabajos están ordenados según los criterios de optimalidad: maximizar la probabilidad factible y minimizar el máximo tiempo de completación. Sin embargo, nos encontramos ante un problema jerárquico puesto que el criterio de factibilidad tiene mayor importancia debido a no estar permitidos los trabajos tardíos. Este problema se denomina $HSSOP$ o problema de ordenación secuencial estocástico jerárquico.

Atendiendo a lo mencionado anteriormente, nos encontramos ante un problema de planificación $1|prec, r_j|(P(Sfactible), C_{max})$.

En este problema cada trabajo j $(j = 1, ..., n)$ tiene que cumplir una ventana temporal $[r_j, d_j]$, donde r_j es el tiempo de disponibilidad del trabajo j y d_j es la fecha límite del trabajo j. Los tiempos de disponibilidad de los trabajos son distintos, con lo que nos encontramos ante un *problema de planificación dinámico*.

Las interrupciones no están permitidas. Si en la máquina empieza a procesarse un trabajo entonces el siguiente trabajo no podrá comenzar hasta haber finalizado el anterior.

Además, la ordenación de los trabajos está condicionada por relaciones de precedencia entre ellos. Estas relaciones se representan con un grafo $R = (V, P)$ acíclico donde el conjunto V de vértices representa el conjunto de los n trabajos del problema y donde el conjunto de arcos P nos indica que cualquier solución factible S debe verificar que si $(i, j) \in P$ entonces el trabajo j solo se puede empezar a procesar una vez se haya terminado de procesar el trabajo i, aunque no necesariamente despues de este.

Decimos que una secuencia S es factible si verifica las relaciones de precedencia entre los trabajos dadas por el grafo $R = (V, P)$ y las ventanas temporales $[r_j, d_j]$ para todo trabajo j $(j = 1, ..., n)$. Es decir, los tiempos de comienzo cumplen $S_j \geq r_j (j = 1, ..., n)$ y los tiempos de completación cumplen $C_j \leq d_j$. Si S verifica las relaciones de precedencia y las fechas de disponibilidad r_j $(j = 1, ..., n)$ entonces decimos que $S \in \Omega$.

Los tiempos de procesamiento de los trabajos son desconocidos y, por lo tanto, la incertidumbre se afronta con la variabilidad de los datos. Que los tiempos de procesamiento p_j $(j = 1, ..., n)$ sean variables aleatorias implica que los tiempos de completación C_j $(j = 1, ..., n)$ también son variables aleatorias. Las condiciones que debe verificar S cuando pertenece a Ω no están sujetas a la variabilidad de los tiempos de procesamiento de los trabajos. En cambio, la condición $C_j \leq d_j$ $(j = 1, ..., n)$ están sujetas a la variabilidad de los datos. Si $S \in \Omega$ entonces la probabilidad de que S sea factible es

$$P(S \text{ factible}) = P(C_1 \leq d_1, C_2 \leq d_2, ..., C_n \leq d_n).$$

Por lo tanto, este problema estará resuelto si encontramos al menos una secuencia de tal forma que verifique

$$max\{P(S\ factible)\ /S \in \Omega\} \tag{4.2}$$

y

$$min\{E[C_{max}(S)]\ /S \in \Omega\ y\ P(S\ factible) = P^*\} \tag{4.3}$$

donde $P^* = max\{P(S \text{ factible})\ /S \in \Omega\}$.

Debemos notar que, en general, si los tiempos de procesamiento de los trabajos son variables aleatorias entre $[0, +\infty)$ entonces es improbable que $P(S$ factible$) = 1$. Y, además, si una secuencia S no verifica las relaciones de precedencia entonces $P(S \text{ factible}) = 0$.

Por otro lado, introducimos un nuevo concepto que nos ayudará a obtener las secuencias con una alta probabilidad factible. Si la secuencia S verifica las relaciones de precedencia entonces esta secuencia es α-factible si $P(S \text{ factible}) \geq 1 - \alpha$ con $0 \leq \alpha \leq 1$. Fijamos un nivel de tolerancia $\overline{\alpha}$ cercano a cero, intentando obtener una secuencia S α-factible con $\alpha \leq \overline{\alpha}$. Es decir, estamos haciendo tender a cero α.

Por otra parte, debido a que los tiempos de completación $C_1, C_2, ..., C_n$ son variables aleatorias dependientes, el cálculo computacional de $P(S \text{ factible}) = P(C_1 \leq d_1, ..., C_n \leq d_n)$ puede ser complicado. Este problema se afronta considerando que la probabilidad de factibilidad sea

$$\min_{j=1,...,n} P(C_j \leq d_j) \tag{4.4}$$

donde j se refiere a la posición j-ésima de la solución S. Dado que si aumentamos (4.4) entonces también aumentará $P(C_1 \leq d_1, ..., C_n \leq d_n)$.

El conjunto de probabilidades $P(C_j \leq d_j)$ $(j = 1, ..., n)$ es necesario aproximarlo con variables aleatorias independientes, las cuales serán los tiempos de procesamiento p_j. Se estiman empleando límites superiores e inferiores. Dichos límites surgen de la fórmula recursiva de los tiempos de completación siguiente:

$$C_j = max\{r_j + p_j, r_{j-1} + p_{j-1} + c_{j-1,j} + p_j, ..., r_1 + \sum_{i=1}^{j} p_i + \sum_{i=1}^{j-1} c_{i,i+1}\} \quad (4.5)$$

entonces escogemos los siguientes límites inferior y superior de (4.5):

$$UB(C_j) = \max_{i=1,...,j}\{r_i\} + \sum_{i=1}^{j} p_i + \sum_{i=1}^{j-1} c_{i,i+1} \quad (4.6)$$

y

$$LB(C_j) = r_1 + \sum_{i=1}^{j} p_i + \sum_{i=1}^{j-1} c_{i,i+1} \quad (4.7)$$

que verifican $LB(C_j) \leq C_j \leq UB(C_j)$. Y, por lo tanto, $P(UB(C_j) \leq d_j) \leq P(C_j \leq d_j) \leq P(LB(C_j) \leq d_j)$ para $j = 1, ..., n$. Es decir, $P(C_j \leq d_j) \in [P(UB(C_j) \leq d_j), P(LB(C_j) \leq d_j)]$. A dicho intervalo se le denomina intervalo probabilístico del trabajo j.

La estimación $\hat{P}(C_j \leq d_j)$ se aproxima con una suma ponderada de los extremos del intervalo probabilístico. En concreto,

$$\hat{P}(C_j \leq d_j) = w_{1j}P(UB(C_j) \leq d_j) + w_{2j}P(LB(C_j) \leq d_j) \quad (4.8)$$

con $w_{1j}, w_{2j} \geq 0, w_{1j} + w_{2j} = 1$ para $j = 1, ..., n$.

Con respecto a minimizar el makespan, el valor real también es aproximado. Si en (4.6) y (4.7) hacemos $j = n$ y tomamos esperanzas obtenemos $E[UB(C_n)] = E[UB(C_{max})]$ y $E[LB(C_n)] = E[LB(C_{max})]$, que verifican $E[LB(C_{max})] \leq E[C_{max}] \leq E[UB(C_{max})]$. Incluso podemos acotar más el valor inferiormente si consideramos la proposición: dadas las variables aleatorias $X_1, X_2, ..., X_n$ entonces $E[max\{X_1, ..., X_n\} \geq max\{E[X_1], ..., E[X_n]\}$. Ahora, empleando la expresión (4.5) con $j = n$ y esta proposición obtenemos otra cota inferior nueva,

$$max\{r_n + E[p_n], r_{n-1} + E[p_{n-1}] + c_{n-1,n} + E[p_n], ..., r_1 + \sum_{i=1}^{n} E[p_i] + \sum_{i=1}^{n-1} c_{i,i+1}\} \quad (4.9)$$

que mejora la cota inferior $E[LB(C_{max})]$. Si denotamos LB_2 a esta segunda cota inferior (4.9) y $UB_1 := E[UB(C_{max})]$ logramos la estimación de $E[C_{max}]$ dada por

$$\hat{E}[C_{max}] = w'_1 LB_2 + w'_2 UB_1 \tag{4.10}$$

con $w'_1, w'_2 \geq 0, w'_1 + w'_2 = 1$.

4.2. Complejidad Computacional

Relativo a la complejidad computacional contamos con el siguiente teorema y su respectiva demostración.

Teorema 4.1 *El problema HSSOP es NP-duro.*

Demostración. Como el *problema del viajante de comercio* es *NP*-duro, y es un caso particular del problema (4.1), se tiene que el problema (4.1) es también *NP*-duro.

Por otro lado, consideramos además el problema de minimizar el makespan en el caso determinístico:

$$min\{C_{max}(S)/S \in \Omega \text{ y } S factible\}. \tag{4.11}$$

Por otra parte, el makespan del problema (4.11) es

$$C_{max}(S) = \sum_{j=1}^{n} p_j + \sum_{(i,j)\in S} c_{i,j} + I(r_1, r_2, ..., r_n, d_1, d_2, ..., d_n)$$

donde $S \in \Omega$ es una solución y $I(r_1, r_2, ..., r_n, d_1, d_2, ..., d_n)$ el tiempo ocioso debido a las fechas de disponibilidad y las fechas límite de los trabajos. Podemos observar, que si $r_j = 0$ y $d_j = \infty$ $(j = 1, ..., n)$ entonces el problema 4.1 es un caso particular de (4.11). Acabamos de ver que el problema (4.1) es *NP*-duro entonces se tiene que el problema (4.11) es *NP*-duro.

En el problema de ordenación secuencial estocástico jerárquico (HSSOP) dado por (4.2) y (4.3) se trabaja con el conjunto $\{S \in \Omega/P(S factible) = P^*\}$. Dicho conjunto en el caso determinístico equivale al conjunto $\{S \in \Omega/S factible\}$ siendo S factible si $C_j \leq d_j$, $\forall j$. Si nos fijamos en el problema (4.3) podemos ver que el caso determinístico equivale al problema (4.11). Por lo tanto, tenemos que el problema (4.3) es *NP*-duro puesto que el problema (4.11) es *NP*-duro. Finalmente, hemos llegado a que el problema HSSOP es *NP*-duro.

4.3. Métodos de resolución del problema

Para resolver este problema de planificación hacemos uso de dos algoritmos.

El primer algoritmo es un algoritmo exacto que nos devuelve la solución óptima al problema. El pseudocódigo de este algoritmo se puede ver en **Algorithm** 2. En él se emplean las siguientes variables:

1. U: conjunto de soluciones no probadas
2. $\sum$: conjunto de mejores soluciones encontradas relativas al primer criterio
3. P^*: probabilidad factible de todas las soluciones en $\sum$
4. $M \subseteq \sum$: contiene las mejores soluciones del segundo criterio
5. E^*: makespan esperado de las soluciones en M.

Al finalizar el procedimiento, obtenemos el conjunto M de soluciones óptimas cuya probabilidad factible es P^* y makespan esperado es E^*.

El segundo algoritmo es un algoritmo heurístico creado por **David et al.** [3]. Pretende conseguir una solución aproximada del problema. Este algoritmo es más eficiente que el algoritmo exacto ya que si aumentamos el valor de n realizar el cálculo exacto requiere muchos recursos. El pseudocódigo puede verse en **Algorithm** 1.

El algoritmo comienza con una *fase de prepocesamiento* donde se pretende reducir el tamaño de la ventana temporal de cada trabajo teniendo en cuenta las relaciones de precedencia entre ellos. Sea j un trabajo con un conjunto de predecesores en el grafo $R = (V, P)$, y sea θ_j el instante de tiempo en el cual si el trabajo j comienza a procesarse antes de dicho valor entonces alguno de sus predecesores no pueden cumplir su fecha de vencimiento. Entonces actualizamos la fecha de disponibilidad del trabajo j a $r'_j = max\{r_j, \theta_j\}$. De la misma forma, dado el trabajo i con un conjunto de sucesores en el grafo $R = (V, P)$, y sea δ_i el instante de tiempo en el cual si el trabajo i finaliza despues de este valor entonces alguno de sus sucesores no puede cumplir su fecha de vencimiento. Por lo tanto, actualizamos la fecha de vencimiento del trabajo i a $d'_i = min\{d_i, \delta_i\}$. Una vez terminado este proceso se ordenan los trabajos en una secuencia S según la regla *EDD (Earliest Due Date)*. Dicha secuencia satisface las relaciones de precedencia de los trabajos.

En la primera fase se pretende encontrar la secuencia S con mayor factibilidad. Buscamos aumentar el valor de (4.8) para todos los trabajos, es decir, aumentar el valor de (4.4) considerando las aproximaciones dadas por (4.8). En esta fase recibimos la secuencia $S \in \Omega$ dada por la fase anterior y un valor α inicial. Si para dicha secuencia S y valor α la secuencia es α-factible y $\alpha \leq \overline{\alpha}$ entonces pasamos a la segunda fase.

En caso contrario, si $\alpha > \overline{\alpha}$ activamos el módulo de reordenamiento y comenzamos desde la posición inicial buscando el primer trabajo que no sea α-factible.

Si la secuencia S cumple la factibilidad para todos los trabajos entonces reducimos α en una cantidad ε. Si este proceso se repite hasta que $\alpha = 0$ o $\alpha \leq \overline{\alpha}$ entonces pasamos a la segunda fase. En cambio, si $\alpha > \overline{\alpha}$ y encontramos un trabajo j que no sea α-factible (es decir, $\hat{P}(C_j \leq d_j) < 1 - \alpha$) entonces hacemos un reordenamiento de la secuencia. En la secuencia parcial desde el trabajo inicial k al trabajo $y(j)$ anterior al trabajo j, buscamos un trabajo i tal que si cambiamos el trabajo j justo después del trabajo i entonces conseguimos una secuencia parcial factible desde el trabajo k hasta el trabajo $y(j)$. Si hay varios candidatos a i entonces escogemos el trabajo tal que haciendo el cambio maximiza el mínimo de las probabilidades de factibilidad de cada trabajo de dicha secuencia parcial. A continuación, reducimos el valor de α y continuamos con el mismo proceso desde $y(j)$ hasta haber estudiado toda la secuencia. Seguimos con el módulo de intercambio. En esta subfase buscamos 4-uplas de trabajos consecutivos que no pertenezcan a la misma familia de trabajos (e.d. existe time setup entre dichos trabajos) y realizamos un intercambio entre los dos trabajos intermedios siempre y cuando la nueva secuencia sea α-factible, no afecte a las relaciones de precedencia y, sobre todo, si la suma de los times set-up se reduce. Si podemos realizar el intercambio entonces reducimos α y comenzamos de nuevo la fase 1. Si no podemos realizar el intercambio pasamos a la fase 2.

La fase 2 consiste en reducir los tiempos ociosos de la secuencia S recibida de la fase 1. Primero se ordenan los trabajos en una lista en orden no decreciente de los tiempos de disponibilidad. En segundo lugar, calculamos las probabilidades $P(I_i > 0)$ donde $I_i = max\{0, r_i - (C_{i-1} + c_{i-1,1})\}$ es la variable aleatoria que mide el tiempo ocioso entre el trabajo $i-1$ y el trabajo i en S. Para calcular esta probabilidad, tenemos en cuenta que $P(I_i > 0) = P(C_{i-1} < r_i - c_{i-1.i})$ y empleando los límites inferiores y superiores dados en (4.7) y (4.6) obtenemos un intervalo $P(UB(I_i) > 0) \leq P(I_i > 0) \leq P(LB(I_i) > 0)$. Finalmente conseguimos la estimación,

$$\hat{P}(I_i > 0) = w_1 P(UB(I_i) > 0) + w_2 P(LB(I_i) > 0) \tag{4.12}$$

con $w_1, w_2 \geq 0, w_1 + w_2 = 1$. Ahora, se intenta anticipar el trabajo j (seleccionado en orden no decreciente de los tiempos de disponibilidad) a la posición i (seleccionada en orden no creciente de $P(I_i > 0)$). Si existen varias posiciones a las que mover el trabajo j, elegimos la posición que aumenta la probabilidad de factibilidad de la secuencia S.

Por último, en la tercera fase minimizamos el makespan esperado teniendo especial cuidado en no empeorar el nivel de factibilidad conseguido en las dos primeras fases. En esta fase empleamos la estimación conseguida en 4.10. El algoritmo intenta anticipar el trabajo j (seleccionado en orden no creciente de la anchura $d_j - r_j$) a la posición i (seleccionada en orden no creciente de $P(I_i > 0)$). Antes de realizar ningún cambio se verifica que dicha modificación minimiza el makespan esperado y que el nivel de factibilidad no empeora.

4.4. Experiencia Computacional

Con el fin de analizar el algoritmo exacto y el algoritmo heurístico, ambos se implementaron en lenguaje de programación C y, posteriormente, se realizó la experiencia computacional de los mismos.

Se desean procesar n trabajos en una máquina. La cantidad de trabajos varían entre $\{5, 10, 20, 50, 100\}$. Los tiempos de procesamiento son variables aleatorias independientes distribuidas con distribución exponencial $p_j \sim exp(\lambda_j)$ con el mismo parámetro λ_j, donde $E[p_j] = \frac{1}{\lambda_j}$ es igual para todos los trabajos. El resto de datos son determinísticos, incluidos los tiempos de disponilidad r_j y las fechas de vencimiento d_j.
Para cada valor de n se han generado 20 instancias o batería de problemas. En cada instancia las fechas de disponibilidad r_j se han generado a partir de $U[0, 50]$, los time setup $c_{i,j}$ se han generado de $U[0, 20]$ con $i \neq j$ y $c_{i,i} = 0$, y las fechas de vencimiento d_j se han generado de $U[r_j + L_1, r_j + L_2]$ donde $L_1 = n\frac{1}{\lambda}(2 - TF - \frac{RDD}{2})$ y $L_2 = n\frac{1}{\lambda}(2 - TF + \frac{RDD}{2})$. Vemos que L_1 y L_2 dependen de $\frac{1}{\lambda}$ generado a partir de $U[0, 100]$, TF = "tardiness factor" o factor de tardanza que varía entre $\{0, 0.2, 0.4, 0.6, 0.8\}$, n que es la cantidad de trabajos, y RDD = "range of due date" o rango de fechas límite que varía entre $\{0.2, 0.4, 0.6, 0.8\}$. Cada instancia está caracterizada por el intervalo de variación de las fechas de vencimiento d_j al combinar los valores de TF con los valores de RDD. Las 20 instancias se han sido numeradas variando primero las filas y luego las columnas de la tabla 4.1.

En la tabla 4.1 podemos observar el intervalo que generan los valores de RDD y TF.

TF \ RDD	0.2	0.4	0.6	0.8
0.0	1.9-2.1	1.8-2.2	1.7-2.3	1.6-2.4
0.2	1.7-1.9	1.6-2.0	1.5-2.1	1.4-2.2
0.4	1.5-1.7	1.4-1.8	1.3-1.9	1.2-2.0
0.6	1.3-1.5	1.2-1.6	1.1-1.7	1.0-1.8
0.8	1.1-1.3	1.0-1.4	0.9-1.5	0.8-1.6

Tabla 4.1: Intervalos (2-TFF-RDD/2)-(2-TFF-RDD/2)

En las figuras 4.1 y 4.2 se pueden ver los resultados obtenidos con el algoritmo exacto para $n = 5$. Se ha resuelto únicamente para 5 trabajos debido a que el problema pertenece a la clase NP-duro. El algoritmo exacto tardará mucho más tiempo en proporcionar una respuesta a medida que se aumenta el número de trabajos. Además, podemos comparar los resultados obtenidos con el algoritmo heurístico. En la figura 4.1 comparamos los criterios de $P(Sfactible)$ y $E[C_{max}]$. En la figura 4.2 comparamos los criterios de $min_{j=1,\ldots,n}P(C_j \leq d_j)$ y $E[C_{max}]$. Todos los criterios se han calculado a partir de fórmulas exactas.

En todas las figuras la columna "Inst." representa las veinte instancias de cada valor n. La columna "Imp." hace referencia a la cantidad de mejoras (Improvement). Una mejora se da en el algoritmo heurístico cuando en la fase 1 α se reduce, en la fase 2 cuando se reducen los tiempos ociosos o cuando se activa la fase 3. En las figuras 4.1 y 4.2 la columna de asterisco nos informa cuándo la solución dada por el algoritmo heurístico coincide con la solución proporcionada por el algoritmo exacto. En la figura 4.1 podemos ver que en el experimento con $n = 5$ el algoritmo heurístico ha proporcionado en 17 de las 20 instancias una respuesta óptima. En las tres instancias restantes, las respuestas obtenidas con el algoritmo heurístico son peores que las soluciones óptimas aunque siguen siendo resultados buenos. Por otro lado, los tiempos CPU necesarios para aportar una solución con el algoritmo heurístico es insignificante, no superior a $10^{-4}s$.

En la figura 4.2 podemos observar que en 17 de las 20 instancias el algoritmo heurístico nos proporciona unas respuestas aproximadas que coinciden con las respuestas óptimas. En las 3 estancias restantes, las respuestas obtenidas son peores aunque no muy alejadas del óptimo. Además, el tiempo requerido para aportar dichas soluciones no supera los $10^{-4}s$.

En la figura 4.3 nos centramos en resultados obtenidos por el algoritmo heurístico para $n = 5$. Esta tabla nos proporciona información relativa a la factibilidad y al makespan.

La factibilidad se divide en 3 columnas. La primera columna (Min. Est. Probability) nos facilita el valor

$$\min_{j=1,\ldots,n} \{\hat{P}(C_j \leq d_j)\}$$

que representa el valor de la mínima probabilidad de factibilidad de los trabajos estimada de la solución secuencial S. La segunda columna (Average of Est. Probabilities) nos aporta el valor

$$\frac{1}{n}\sum_{j=1}^{n} \hat{P}(C_j \leq d_j)$$

es decir, el promedio de las probabilidades de factibilidad de los trabajos estimadas. La tercera columna (Interval Width Average) nos da el valor

$$\frac{1}{n}\sum_{j=1}^{n}[P(LB(C_j \leq d_j) - P(UB(C_j) \leq d_j))]$$

nos proporciona la anchura promedio de los intervalos que contienen probabilidades de factibilidad de los trabajos estimadas. La última columna nos da el valor exacto de la mínima probabilidad de factibilidad de los trabajos.

Las columnas relativas al makespan son cuatro. La primera y segunda columna nos ofrecen el límite inferior y el límite superior del valor esperado del makespan, respectivamente. La tercera columna nos da el valor esperado del makespan estimado. La cuarta columna nos da el valor exacto del valor esperado del makespan. La última columna nos informa del tiempo CPU que tardó el algoritmo en alcanzar la solución final.

Si observamos la columna "Interval Width Average" vemos que los valores son muy pequeños. Por lo tanto, cuando escogemos pesos $w_{1j} = w_{2j} = 0.5$ para todo j la aproximación $min_{j=1,\ldots,n}\hat{P}(C_j \leq d_j)$ (Min. Est. Probability) es bastante aceptable si la comparamos con la columna $min_{j=1,\ldots,n}P(C_j \leq d_j)$. Además, también se experimentó con diferentes pesos para distintos valores de n y se concluyó que los valores obtenidos eran similares.

En las figuras 4.4-4.8 se muestran los mismos parámetros que en la figura 4.3 pero sin aportar los valores exactos de la probabilidad factible ni del makespan esperado puesto que estamos trabajando con valores elevados de n. Podemos ver que los valores de la mínima probabilidad estimada son cercanos a uno en su mayoría y el tiempo CPU que se ha tardado en aportar la solución final es insignificante. Por ejemplo, en la figura 4.4 17 de 20 instancias el valor de la prob. fact. es mayor que 0.8 y 9 de 20 instancias es mayor que 0.9. En la figura 4.5 15 de 20 instancias es mayor que 0.9 y para 18 de 20 instancias son mayores que 0.7. En la figura 4.6 18 de 20 instancias es mayor que 0.96. En la figura 4.7 18 de 20 instancias son mayores que 0.99.
La última figura 4.8 nos muestra una comparativa con la figura 4.7 con respecto al valor $\hat{E}[C_{max}]$ con respecto a distintos pesos w'_1 y w'_2. En la figura 4.7 se han escogidos pesos $w'_1 = w'_2 = 0.5$ y en la figura 4.8 $w'_1 = 0.75$ y $w'_2 = 0.25$. Vemos que en ambas el valor $\hat{E}[C_{max}]$ es similar.

En cuanto al tiempo empleado por el algoritmo heurístico podemos apreciar que en la figura 4.7 donde tenemos $n = 100$ no se superan los 6 segundos, solo dos instancias superan los 5 segundos y 3 superan los 3 segundos.

En resumen, se ha obtenido un algoritmo que proporciona niveles altos de probabilidad de factibilidad y minimiza el makespan esperado en un tiempo aceptable.

Figura 4.1: $P(S factible)$ y $E[C_{max}]$ para $n = 5$

Inst.	Imp.	Approximate solution				Optimal solution		
		Feasibility probability	Expected makespan		CPU time	Feasibility probability	Expected makespan	CPU time
1	1	0.977568	137.32	*	0.0000	0.977568	137.32	0.0000
2	1	0.976137	501.42	*	0.0000	0.976137	501.42	0.0125
3	1	0.988682	459.38		0.0000	0.988972	450.96	0.0000
4	0	0.991375	236.21	*	0.0000	0.991375	236.21	0.0000
5	1	0.971392	127.74	*	0.0000	0.971392	127.74	0.0125
6	0	0.966587	477.72	*	0.0000	0.966587	477.72	0.0000
7	1	0.978070	304.21	*	0.0000	0.978070	304.21	0.0000
8	1	0.980944	83.07	*	0.0000	0.980944	83.07	0.0000
9	1	0.939837	146.36		0.0000	0.939871	146.36	0.0000
10	0	0.924841	469.36	*	0.0000	0.924841	469.36	0.0000
11	1	0.937822	193.00		0.0000	0.947114	178.96	0.0000
12	0	0.961276	519.96	*	0.0000	0.961276	519.96	0.0000
13	0	0.872667	133.95	*	0.0000	0.872667	133.95	0.0000
14	1	0.874701	186.07	*	0.0000	0.874701	186.07	0.0000
15	0	0.915177	197.14	*	0.0000	0.915177	197.14	0.0000
16	0	0.872564	285.10	*	0.0000	0.872564	285.10	0.0000
17	0	0.904001	66.29	*	0.0000	0.904001	66.29	0.0000
18	0	0.798292	102.23	*	0.0000	0.798292	102.23	0.0000
19	1	0.728907	531.02	*	0.0000	0.728907	531.02	0.0125
20	0	0.832689	499.08	*	0.0000	0.832689	499.08	0.0000

Figura 4.2: $max_{S \in \Omega}\{min_{j=1,\ldots,n}\{P(C_j(S) \leq d_j)\}\}$ y $E[C_{max}]$ para $n = 5$

Inst.	Imp.	Approximate solution				Optimal solution		
		$\min_{j=1,\ldots,n}$ $P(C_j \leqslant d_j)$	Expected makespan		CPU time	$\min_{j=1,\ldots,n}$ $P(C_j \leqslant d_j)$	Expected makespan	CPU time
1	0	0.977819	137.32	*	0.0000	0.977819	137.32	0.0000
2	1	0.976314	501.42	*	0.0000	0.976314	501.42	0.0125
3	1	0.988904	459.38		0.0000	0.989145	453.24	0.0000
4	1	0.992257	236.19	*	0.0000	0.992257	236.19	0.0000
5	1	0.972078	127.74	*	0.0000	0.972078	127.74	0.0125
6	1	0.967023	477.70	*	0.0000	0.967023	477.70	0.0000
7	2	0.981689	304.08		0.0000	0.982988	304.21	0.0000
8	1	0.987305	83.07	*	0.0000	0.987305	83.07	0.0000
9	1	0.946307	146.36	*	0.0000	0.946307	146.36	0.0000
10	0	0.934321	469.36	*	0.0000	0.934321	469.36	0.0000
11	1	0.942745	193.00		0.0000	0.957715	178.96	0.0000
12	1	0.963066	519.11	*	0.0000	0.963066	519.11	0.0000
13	0	0.886076	133.95	*	0.0000	0.886076	133.95	0.0000
14	1	0.902853	186.07	*	0.0000	0.902853	186.07	0.0000
15	0	0.927503	197.14	*	0.0000	0.927503	197.14	0.0000
16	0	0.901269	285.10	*	0.0000	0.901269	285.10	0.0000
17	0	0.926692	66.29	*	0.0000	0.926692	66.29	0.0000
18	1	0.809492	101.49	*	0.0000	0.809492	101.49	0.0000
19	1	0.767242	531.02	*	0.0000	0.767242	531.02	0.0125
20	2	0.853313	487.74	*	0.0000	0.853313	487.74	0.0000

Figura 4.3: Resultados algoritmo heurístico con $n = 5$

Inst.	Imp.	Feasibility				Makespan				CPU time
		Min. est. probability	Average of est. probabilities	Interval width average	$\min_{j=1,\dots,n} P(C_j \leqslant d_j)$	Lower bound	Upper bound	$\hat{E}[C_{\max}]$	Expected makespan	
1	0	0.956243	0.987017	0.012354	0.977819	137	173	155.00	137.32	0.0000
2	1	0.972140	0.991352	0.002707	0.976314	500	546	523.00	501.42	0.0000
3	1	0.987074	0.996134	0.001364	0.988904	450	494	472.00	459.38	0.0000
4	1	0.990751	0.996545	0.000688	0.991708	240	252	246.00	240.00	0.0000
5	1	0.948509	0.984074	0.014740	0.972078	126	163	144.50	127.74	0.0000
6	0	0.962161	0.987889	0.003099	0.967022	476	514	495.00	477.73	0.0000
7	2	0.979866	0.989191	0.002412	0.981689	304	331	317.50	304.08	0.0000
8	0	0.950884	0.977422	0.034319	0.983418	86	113	99.50	89.41	0.0000
9	0	0.899797	0.964837	0.024370	0.945945	146	191	168.50	147.01	0.0000
10	0	0.928178	0.972996	0.004152	0.934321	469	494	481.50	469.36	0.0000
11	1	0.911421	0.968586	0.019250	0.935172	181	211	196.00	183.11	0.0000
12	1	0.957275	0.984997	0.004252	0.963030	519	562	540.50	519.96	0.0000
13	0	0.779686	0.924489	0.052997	0.886076	133	181	157.00	133.95	0.0000
14	1	0.859094	0.927784	0.042545	0.902853	186	227	206.50	186.07	0.0000
15	0	0.884796	0.957855	0.025063	0.927503	197	245	221.00	197.14	0.0000
16	0	0.885067	0.939344	0.015832	0.901269	284	310	297.00	285.10	0.0000
17	0	0.683158	0.876036	0.199314	0.926692	62	105	83.50	66.29	0.0000
18	0	0.718081	0.894601	0.067586	0.805706	100	126	113.00	102.23	0.0000
19	1	0.741762	0.867985	0.020363	0.767242	531	572	551.50	531.02	0.0000
20	1	0.834477	0.927047	0.014539	0.853261	487	529	508.00	487.96	0.0000

Figura 4.4: Resultados algoritmo heurístico con $n = 10$

Inst.	Imp.	Feasibility			Makespan			CPU time
		Min. est. probability	Average of est. probabilities	Interval width average	Lower bound	Upper bound	$\hat{E}[C_{\max}]$	
1	1	0.992330	0.998735	0.001879	181	243	212.0	0.010
2	1	0.990884	0.997805	0.003667	591	819	705.0	0.000
3	1	0.994677	0.998423	0.002230	875	1127	1001.0	0.000
4	1	0.992006	0.998135	0.002125	933	1124	1028.5	0.000
5	1	0.969763	0.993503	0.008426	746	959	852.5	0.010
6	1	0.978068	0.995867	0.005963	338	444	391.0	0.000
7	2	0.502317	0.785209	0.426056	52	72	62.0	0.000
8	2	0.983110	0.994400	0.007407	357	436	396.5	0.000
9	1	0.818833	0.957699	0.064642	301	416	358.5	0.000
10	2	0.893957	0.973512	0.031177	572	719	645.5	0.000
11	3	0.985514	0.995602	0.001825	673	740	706.5	0.000
12	1	0.879168	0.954294	0.074076	555	783	669.0	0.000
13	1	0.874632	0.964231	0.031238	697	831	764.0	0.000
14	2	0.888569	0.965053	0.047179	570	750	660.0	0.000
15	0	0.850564	0.959727	0.035759	1028	1294	1161.0	0.010
16	1	0.955033	0.985940	0.017288	217	270	243.5	0.000
17	3	0.279625	0.635151	0.381234	286	455	370.5	0.000
18	2	0.554438	0.842926	0.192960	265	360	312.5	0.010
19	2	0.827840	0.927486	0.082519	731	975	853.0	0.000
20	1	0.828546	0.947658	0.062604	869	1124	996.5	0.000

Figura 4.5: Resultados algoritmo heurístico con $n = 20$

Inst.	Imp.	Feasibility			Makespan			CPU time
		Min. est. probability	Average of est. probabilities	Interval width average	Lower bound	Upper bound	$\hat{E}[C_{\max}]$	
1	4	0.999731	0.999969	0.000050	514	617	565.5	0.030
2	4	0.999842	0.999980	0.000033	1240	1482	1361.0	0.020
3	5	0.999959	0.999994	0.000010	1848	2230	2039.0	0.020
4	1	0.999951	0.999989	0.000015	1185	1361	1273.0	0.010
5	4	0.997484	0.999808	0.000269	1104	1312	1208.0	0.010
6	4	0.997901	0.999550	0.000810	402	506	454.0	0.020
7	4	0.998176	0.999684	0.000576	507	637	572.0	0.030
8	2	0.999737	0.999956	0.000076	1871	2335	2103.0	0.020
9	3	0.953885	0.993836	0.010699	228	289	258.5	0.040
10	6	0.521306	0.640145	0.719650	110	117	113.5	0.040
11	2	0.994686	0.999080	0.001445	1510	1821	1665.5	0.020
12	0	0.935827	0.988085	0.021819	389	505	447.0	0.020
13	1	0.926181	0.987313	0.013629	1710	1953	1831.5	0.030
14	4	0.969222	0.994191	0.005489	879	974	926.5	0.020
15	3	0.442813	0.757506	0.422128	231	324	277.5	0.020
16	1	0.998506	0.999773	0.000399	1152	1439	1295.5	0.040
17	0	0.716047	0.942063	0.072209	491	592	541.5	0.020
18	4	0.822506	0.950648	0.060464	1543	1859	1701.0	0.020
19	3	0.781767	0.942206	0.060684	1953	2337	2145.0	0.020
20	2	0.941873	0.977480	0.029512	945	1144	1044.5	0.030

Figura 4.6: Resultados algoritmo heurístico con $n = 50$

Inst.	Imp.	Feasibility			Makespan			CPU time
		Min. est. probability	Average of est. probabilities	Interval width average	Lower bound	Upper bound	$\hat{E}[C_{\max}]$	
1	15	1.000000	1.000000	0.000000	1206	1254	1230.0	0.250
2	10	1.000000	1.000000	0.000000	1120	1168	1144.0	0.160
3	0	1.000000	1.000000	0.000000	1212	1261	1236.5	0.210
4	16	1.000000	1.000000	0.000000	3524	3570	3547.0	0.170
5	14	1.000000	1.000000	0.000000	3643	3693	3668.0	0.180
6	5	1.000000	1.000000	0.000000	2464	2514	2489.0	0.130
7	9	1.000000	1.000000	0.000000	3569	3616	3592.5	0.240
8	13	1.000000	1.000000	0.000000	1960	2005	1982.5	0.300
9	0	0.998299	0.999916	0.000043	2138	2187	2162.5	0.010
10	9	0.999997	1.000000	0.000000	2050	2098	2074.0	0.230
11	3	1.000000	1.000000	0.000000	4582	4629	4605.5	0.170
12	6	0.999996	1.000000	0.000000	987	1033	1010.0	0.390
13	10	0.864857	0.925874	0.013844	4325	4867	4596.0	0.250
14	6	0.999797	0.999988	0.000003	4198	4248	4223.0	0.210
15	12	0.999967	0.999998	0.000001	1772	1822	1797.0	0.360
16	14	0.999990	0.999998	0.000000	4567	4614	4590.5	0.250
17	7	0.966218	0.997264	0.000863	2319	2367	2343.0	0.490
18	16	0.493719	0.886292	0.167731	294	342	318.0	0.690
19	10	0.993988	0.999324	0.000395	1570	1620	1595.0	0.490
20	7	0.999464	0.999897	0.000058	1879	1929	1904.0	0.390

Figura 4.7: Resultados algoritmo heurístico con $n = 100$ y $w'_1 = w'_2 = 0.5$

Inst.	Imp.	Feasibility			Makespan			CPU time
		Min. est. probability	Average of est. probabilities	Interval width average	Lower bound	Upper bound	$\hat{E}[C_{\max}]$	
1	18	1.000000	1.000000	0.000000	5204	5249	5226.5	1.180
2	14	1.000000	1.000000	0.000000	5087	5137	5112.0	1.610
3	18	1.000000	1.000000	0.000000	7446	7495	7470.5	1.580
4	11	1.000000	1.000000	0.000000	9471	9520	9495.5	1.590
5	19	1.000000	1.000000	0.000000	2810	2859	2834.5	1.750
6	23	1.000000	1.000000	0.000000	9285	9330	9307.5	1.270
7	14	1.000000	1.000000	0.000000	1939	1988	1963.5	2.380
8	10	1.000000	1.000000	0.000000	8969	9019	8994.0	1.230
9	9	1.000000	1.000000	0.000000	9190	9240	9215.0	0.190
10	13	1.000000	1.000000	0.000000	7192	7242	7217.0	1.240
11	27	1.000000	1.000000	0.000000	5865	5914	5889.5	1.180
12	14	1.000000	1.000000	0.000000	8880	8930	8905.0	1.300
13	16	0.999991	1.000000	0.000000	8183	8233	8208.0	2.660
14	11	1.000000	1.000000	0.000000	9178	9227	9202.5	1.320
15	9	1.000000	1.000000	0.000000	5196	5246	5221.0	1.420
16	14	1.000000	1.000000	0.000000	6652	6701	6676.5	1.300
17	10	0.929023	0.996038	0.002936	1519	1545	1532.0	5.470
18	20	0.427526	0.914740	0.055715	1015	1062	1038.5	3.050
19	9	0.997986	0.999912	0.000082	1650	1700	1675.0	5.430
20	16	0.999937	0.999997	0.000001	3825	3875	3850.0	2.910

Figura 4.8: Resultados algoritmo heurístico con $n = 100$, $w'_1 = 0.75$ y $w'_2 = 0.25$

Inst.	Imp.	Feasibility			Makespan			CPU time
		Min. est. probability	Average of est. probabilities	Interval width average	Lower bound	Upper bound	$\hat{E}[C_{\max}]$	
1	16	1.000000	1.000000	0.000000	5250	5295	5272.5	1.060
2	14	1.000000	1.000000	0.000000	5087	5137	5112.0	1.590
3	18	1.000000	1.000000	0.000000	7446	7495	7470.5	1.580
4	11	1.000000	1.000000	0.000000	9471	9520	9495.5	1.580
5	19	1.000000	1.000000	0.000000	2810	2859	2834.5	1.760
6	20	1.000000	1.000000	0.000000	9296	9339	9317.5	1.230
7	16	1.000000	1.000000	0.000000	1935	1985	1960.0	1.750
8	10	1.000000	1.000000	0.000000	8969	9019	8994.0	1.240
9	9	1.000000	1.000000	0.000000	9190	9240	9215.0	0.200
10	15	1.000000	1.000000	0.000000	7161	7211	7186.0	1.390
11	28	1.000000	1.000000	0.000000	5864	5914	5889.0	1.180
12	20	1.000000	1.000000	0.000000	8792	8839	8815.5	1.700
13	14	0.999990	1.000000	0.000000	8199	8249	8224.0	2.130
14	11	1.000000	1.000000	0.000000	9191	9241	9216.0	1.080
15	9	1.000000	1.000000	0.000000	5196	5246	5221.0	1.420
16	9	1.000000	1.000000	0.000000	6651	6701	6676.0	1.350
17	15	0.954087	0.997469	0.001683	1435	1485	1460.0	7.780
18	20	0.362039	0.899107	0.057361	1015	1062	1038.5	4.140
19	7	0.997540	0.999892	0.000082	1650	1700	1675.0	5.920
20	16	0.999929	0.999997	0.000001	3825	3875	3850.0	2.910

4.5. Conclusiones

En conclusión, se ha afrontado un problema de planificación estocástico jerárquico bicriterio. Los tiempos de procesamiento son variables aleatorias con lo que obtenemos un problema de planificación estocástico. La incertidumbre en los tiempos de procesamiento implica variabilidad en los tiempos de completación de los trabajos. Cada uno de los datos restantes son conocidos. Se pretende maximizar la probabilidad de factibilidad y minimizar el makespan esperado. Debido a que los trabajos tardíos no están permitidos, el criterio de máximo es más relevante. Intervienen relaciones de precedencia e intervalos de tiempo para cada trabajo. Se ha demostrado que el problema es NP-duro, y se ha presentado un algoritmo para aproximar la solución. La experiencia computacional nos muestra que el algoritmo propuesto proporciona resultados aceptables en poco tiempo.

Como futuros proyectos, se puede considerar hacer un cambio en el segundo criterio de minimizar el makespan esperado. Este cambio puede derivar en otro problema bicriterio donde se pretende maximizar la probabilidad de factibilidad y minimizar, por ejemplo, el valor esperado de la máxima demora, la máxima tardanza o el tiempo de permanencia. También se podría considerar incertidumbre en otros datos, por ejemplo, en las fechas límites o los tiempos de disponibilidad.

5

Conclusiones

El trabajo realizado con la realización de la presente memoria de Trabajo Fin de Grado nos ha permitido utilizar lo aprendido en los estudios de Grado para entender y asimilar mejor los conceptos más relevantes en los Problemas de Planificación Estocástica y así aproximarnos a ellos siendo consciente de la dificultad que tienen y del gran potencial y aplicabilidad para la resolución de problemas reales que pueden presentar.

A

Apéndice

A.1. Pseudocódigo del algoritmo heurístico

Algorithm 1 Algorithm HSSOP

```
/* Pre-processing phase */
UpdateJobTimeWindows();
/* Phase 1 */
Initial solution S;
Initial α − value  α;
while α > ᾱ do
    RearrangeModule();
    if we find S feasible with parameter α then
        Reduce α
    end if
    else
        ExchangeModule();
        If we find S feasible with parameter α
            Reduce α
        end if
    end else
end while
/* Phase 2 */
ReduceIdleTime();
/* Phase 3 */
MinimizeExpectedMakespan();
STOP
```

A.2. Pseudocódigo del algoritmo exacto

Algorithm 2 Exact Procedure

/ Initialization Phase */*
$U = \Omega$;
Choose a sequence $S_1 \in U$ *as initial solution*;
$P^* = P(S_1 factible)$;
$\sum = \{S_1\}$;
$U = U - \{S_1\}$;
/ Feasibility probability phase */*
while $U \neq \emptyset$ **do**
    Choose a sequence $S \in U$;
    if $P(S feasible) > P^*$ **then**
        $P^* = P(S feasible)$;
        $\sum = \{S\}$,
    end if
    else if $P(S feasible) = P^*$ **then**
        $\sum = \sum \cup \{S\}$;
    end else if
    $U = U - \{S\}$;
end while
/ Expected makespan phase */*
$U = \sum$;
Choose a sequence $S_2 \in U$ *as initial solution in this phase*;
$E^* = E[C_{max}(S_2)]$;
$M = \{S_2\}$;
$U = U - \{S_2\}$;
while $U \neq \emptyset$ **do**
    Choose a sequence $S \in U$;
    if $E[C_{max}(S)] < E^*$ **then**
        $E^* = E[C_{max}(S)]$;
        $M = \{S\}$;
    end if
    else if $E[C_{max}(S)] = E^*$ **then**
        $M = M \cup \{S\}$;
    end else if
    $U = U - \{S\}$;
end while

Bibliografía

[1] Alcaide López de Pablo, David (1995). *Problemas De Planificación Y Secuenciación Determinística : Modelización Y Técnicas De Resolución.* Tesis Doctoral. Departamento de Estadística, Investigación Operativa y Computación. Universidad de La Laguna (1995). Publicado también por el Servicio de Publicaciones de la Universidad de La Laguna. Soportes Audiovisuales e Informáticos. Serie Tesis Doctorales. Curso 1995/96. Ciencias y Tecnologías. Vol. 10. Servicio de Publicaciones de la Universidad de La Laguna, 2004. I.S.B.N.: 84-7756-582-1 (Obra completa). I.S.B.N.: 84-7756-584-8 (Vol.10) Depósito Legal: TF 1115/2004.

[2] Alcaide López de Pablo, David. (2008). *On scheduling models. BEIO, Boletín de la Sociedad Española de Estadística E Investigación Operativa, 24(2), 11-21.*

[3] Alcaide, D., Rodríguez-González, A., Sicilia, J. (2003) *An approach to solve a hierarchical stochastic sequential ordering problem* Omega, the International Journal of Management Science, vol. 31, 169-187.

[4] Graham, R.L.; E.L. Lawler; J.K. Lenstra; A.H.G. Rinnooy Kan (1979) *Optimization and approximation in deterministic sequencing and scheduling: a survey* Annals of discrete mathematics, vol. 5, pp. 287-326.

[5] Rodríguez González, A. (1999). *Cuestiones notables sobre problemas de planificación estocática.* Memoria de Licenciatura. Departamento de Estadística, Investigación Operativa y Computación. Universidad de La Laguna. Tenerife. España.

[6] Sánchez García, M. (1982). *El proceso de modelización en la investigación actual : Discurso inaugural del curso 1981-1982.* Servicio de publicaciones. Universidad de La Laguna. Depósito Legal: TF 1939-82.

An approach to stochastic scheduling problems

Sección de Matemáticas
Universidad de La Laguna

Nieves García Hernández
Facultad de Ciencias · Sección de Matemáticas
Universidad de La Laguna
alu0100776782@ull.edu.es

Abstract

The Scheduling Problems arise when several tasks (the jobs) must be performed by several entities (the machines) in order to achieve and optimise some aims (the optimisation criteria). The machines must be assigned to the jobs to process them under some constraints with the purpose of optimising the considered criteria. This Report is devoted to introduce the Scheduling Problems making special mention to Stochastic Scheduling. A chapter devoted to a practical application to a Hierarchical Stochastic Sequential Ordering Problem (HSSOP) is also included.

1. Introduction

This Report introduces the Scheduling Problems with special mention to Stochastic Scheduling problems. The Scheduling Problems appear when there are activities or tasks to process (" what? ", the jobs) by some entitiess ("who? ", the machines) in order to achieve some aims ("For what?", the optimisation criteria). These problems are usually relatively easy to set but very difficult to solve. The chapters of the Report are dedicated to introduce these problems and different models to solve them.

2. Chapter 1

The basic models to solve Scheduling Problems are described, taking into account the three-parameter classification $\alpha|\beta|\gamma$ initially proposed by Graham et al. (1979), in which α refers to the characteristics of the machines, β to the jobs and γ to the optimisation criteria. We also distinguish and compare the Stochastic Scheduling Problems with the Deterministic Scheduling Problems.

3. Chapter 2

In the second chapter, we present general concepts about Stochastic Scheduling. The uncertainty of these problems is confronted using random variables, therefore, we make a reminder of the distribution functions and describe some methods of comparison among them. In addition, the decision-making policies and some resolution methods are explained.

4. Chapter 3

In the third chapter we have expected to compile the most important facets related to the Computational Complexity.

5. Chapter 4

This chapter confront a concrete practical application studied by Alcaide et al. (2003) related to a Hierarchical Stochastic Sequential Ordering Problem (HSSOP).

6. Chapter 5

The last chapter includes a brief conclusion of this Dissertation.

References

[1] Alcaide López de Pablo, David (1995). *Problemas De Planificación Y Secuenciación Determinística : Modelización Y Técnicas De Resolución.* Tesis Doctoral. Departamento de Estadística, Investigación Operativa y Computación. Universidad de La Laguna (1995). Publicado también por el Servicio de Publicaciones de la Universidad de La Laguna. Soportes Audiovisuales e Informáticos. Serie Tesis Doctorales. Curso 1995/96. Ciencias y Tecnologías. Vol. 10. Servicio de Publicaciones de la Universidad de La Laguna, 2004. I.S.B.N.: 84-7756-582-1 (Obra completa). I.S.B.N.: 84-7756-584-8 (Vol.10) Depósito Legal: TF 1115/2004.

[2] Alcaide López de Pablo, David. (2008). *On scheduling models. BEIO, Boletín de la Sociedad Española de Estadística E Investigación Operativa, 24(2), 11-21.*

[3] Alcaide, D., Rodríguez-González, A., Sicilia, J. (2003) *An approach to solve a hierarchical stochastic sequential ordering problem* Omega, the International Journal of Management Science, vol. 31, 169-187.

[4] Graham, R.L.; E.L. Lawler; J.K. Lenstra; A.H.G. Rinnooy Kan (1979) *Optimization and approximation in deterministic sequencing and scheduling: a survey* Annals of discrete mathematics, vol. 5, pp. 287-326.

[5] Rodríguez González, A. (1999). *Cuestiones notables sobre problemas de planificación estocática.* Memoria de Licenciatura. Departamento de Estadística, Investigación Operativa y Computación. Universidad de La Laguna. Tenerife. España.

[6] Sánchez García, M. (1982). *El proceso de modelización en la investigación actual : Discurso inaugural del curso 1981-1982.* Servicio de publicaciones. Universidad de La Laguna. Depósito Legal: TF 1939-82.